

Chapter 9

Topics in Time Series Analysis

This chapter presents several topics in time series analysis. These include several of the popular time series models which are special cases of the $\text{ARMA}(p, q)$, including the software required for fitting these models. The first section surveys the probability models and statistical methods associated with autoregressive models, more specifically the $\text{AR}(1)$, $\text{AR}(2)$, and $\text{AR}(p)$ models. The second section surveys the probability models and statistical methods associated with autoregressive models, more specifically the $\text{MA}(1)$, $\text{MA}(2)$, and $\text{MA}(q)$ models. It is important to know the properties of these special cases of the $\text{ARMA}(p, q)$ model in order to successfully fit such a model to a realization of a time series. This will allow us to build an inventory of population autocorrelation and partial autocorrelation functions for these models that can be matched to their statistical counterparts for building a time series model. Time series analysts tend to use the smallest possible p and q values that adequately describe a time series. For this reason, separate subsections are devoted to the $\text{AR}(1)$, $\text{AR}(2)$, $\text{MA}(1)$, and $\text{MA}(2)$ time series models.

9.1 Autoregressive Models

Autoregressive models for a time series $\{X_t\}$ will be considered in this section. An autoregressive model of order p is a special case of an $\text{ARMA}(p, q)$ model with no moving average terms (that is, $q = 0$), specified as

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + Z_t,$$

where $\phi_1, \phi_2, \dots, \phi_p$ are real-valued parameters and $\{Z_t\}$ is a time series of white noise with population mean zero and population variance σ_Z^2 . The formulation of the $\text{AR}(p)$ time series model looks quite similar to that of a multiple linear regression model with p independent variables. These independent variables are also known as predictors, regressors, or covariates in regression analysis. That is the genesis of the term *autoregressive* to describe this model. The prefix *auto* means *self*, indicating that this model has the current value of the time series $\{X_t\}$ written as a linear function of the p previous versions of itself plus a white noise term Z_t . The white noise term is critical to the model because without it, there would be no randomness in the model.

Rather than diving right into an $\text{AR}(p)$ model, we first introduce the $\text{AR}(1)$ and $\text{AR}(2)$ models in separate sections because the mathematics are somewhat easier than the general case and some important geometry and intuition can be developed in these restricted models. In addition, an $\text{AR}(1)$ or $\text{AR}(2)$ model is often an adequate time series model in a particular setting. We always want a

model with the fewest possible number of parameters that adequately approximates the underlying time series probability model. In the sections that follow, we will

- define the time series model for $\{X_t\}$,
- determine the values of the parameters associated with a stationary model,
- derive the population autocorrelation and partial autocorrelation functions,
- develop algorithms for simulating observations from the time series,
- inspect simulated realizations to establish patterns,
- estimate the parameters from a time series realization $\{x_t\}$,
- assess the adequacy of the time series model, and
- forecast future values of the time series using both point and interval estimates.

The purpose of deriving the population autocorrelation and partial autocorrelation functions is to build an inventory of shapes and patterns for these functions that can be used to identify tentative time series models from their sample counterparts by making a visual comparison between population and sample versions. This inventory of shapes and patterns plays an analogous role to knowing the shapes of various probability density functions (for example, the bell-shaped normal probability density function or the rectangular-shaped uniform distribution) in the analysis of univariate data in which the shape of the histogram is visually compared to the inventory of probability density function shapes.

In each section that follows, a single example of a time series will be carried through the various statistical procedures given in the list above. Stationarity plays a critical role in time series analysis because we are not able to forecast future values of the time series without knowing that the probability model is stable over time. This is why the visual assessment of a plot of the time series is always a critical first step in the analysis of a time series.

9.1.1 The AR(1) Model

The autoregressive model of order 1 is defined next. It has a closed-form expression for the population autocorrelation function and is frequently used in applications.

Definition 9.1 A first-order autoregressive time series model, denoted by AR(1), for the time series $\{X_t\}$ is defined by

$$X_t = \phi X_{t-1} + Z_t,$$

where ϕ is a real-valued parameter and $\{Z_t\}$ is a time series of white noise:

$$Z_t \sim WN(0, \sigma_Z^2).$$

No subscript is necessary on the ϕ parameter because there is only one ϕ parameter in the AR(1) model. So there are two parameters that define an AR(1) model: the coefficient ϕ and the population variance of the white noise σ_Z^2 .

The current value in the time series, X_t , is given by the parameter ϕ multiplied by the previous observed value in the time series, ϕX_{t-1} , plus the current white noise term Z_t . This model has the

form of a simple linear regression model forced through the origin in which X_t is being predicted by the previous value of the time series X_{t-1} . The parameter ϕ plays the role of the slope of the regression line. Thinking about an AR(1) model as a simple linear regression model suggests a statistical graphic that can be helpful in determining whether it is an appropriate model for a particular time series. A plot of x_t on the vertical axis against x_{t-1} on the horizontal axis should be approximately linear if the AR(1) model is appropriate. The slope of the regression line on this plot corresponds to ϕ , and the magnitude of the variability of the points about the regression line is determined by the population variance of the white noise σ_Z^2 .

Some authors prefer to parameterize the AR(1) model as

$$X_t = \phi_1 X_{t-1} + \phi_0 Z_t,$$

where ϕ_0 and ϕ_1 are real-valued parameters. We avoid this parameterization because the ϕ_0 parameter is redundant in the sense that the population variance of the white noise σ_Z^2 is absorbed into the ϕ_0 parameter. Also, some authors use a $-$ rather than a $+$ between the terms on the right-hand side of the model.

To illustrate the thinking behind the AR(1) model in a specific context, let X_t represent the closing price of a particular stock on day t . The AR(1) model indicates that today's closing price, denoted by X_t , equals ϕ multiplied by yesterday's closing price (ϕX_{t-1}), plus today's random white noise term Z_t .

Stationarity

One initial important question concerning the AR(1) model is whether or not the model is stationary. Consider a thought experiment that determines whether an AR(1) model is stationary for specific values of ϕ . For one particular instance, consider $\phi = 0$. In this case the AR(1) time series model reduces to

$$X_t = Z_t,$$

which is a time series model consisting solely of white noise. We know from Example 7.15 that a time series model of white noise terms is stationary. Now consider another instance, $\phi = 1$. In this case the AR(1) time series model reduces to

$$X_t = X_{t-1} + Z_t,$$

which indicates that each value in the time series is the previous value plus the current white noise. In this case the population variance of the process is increasing with time because the number of white noise terms accumulate over time (see Example 7.8), so the AR(1) model with $\phi = 1$ violates one of the stationarity conditions given in Definition 7.6. The AR(1) model with $\phi = 1$ can be recognized as a random walk model from Example 7.4, and it was determined to be nonstationary in Example 7.17. So we have established that the AR(1) time series model is stationary for $\phi = 0$ and nonstationary for $\phi = 1$. We now try to determine general restrictions on ϕ associated with a stationary AR(1) time series model. We take four different approaches to establishing the values of the coefficient ϕ that lead to a stationary model. The four approaches provide a review of several concepts defined previously.

Approach 1: Causality. In the derivations concerning the AR(1) time series model that follow, it will be beneficial to write the time series value X_t as a linear combination of the current and previous white noise values. This will allow us to use the definition of causality in Definition 8.2 to determine the values of ϕ associated with a stationary AR(1) model. To begin, recall that the AR(1) model given by

$$X_t = \phi X_{t-1} + Z_t$$

can be shifted in time and is equally valid for other t values, for example,

$$\begin{aligned} X_{t-1} &= \phi X_{t-2} + Z_{t-1} \\ X_{t-2} &= \phi X_{t-3} + Z_{t-2} \\ &\vdots \end{aligned}$$

Using successive substitutions into the AR(1) model results in

$$\begin{aligned} X_t &= \phi X_{t-1} + Z_t \\ &= \phi (\phi X_{t-2} + Z_{t-1}) + Z_t \\ &= \phi^2 X_{t-2} + \phi Z_{t-1} + Z_t \\ &= \phi^2 (\phi X_{t-3} + Z_{t-2}) + \phi Z_{t-1} + Z_t \\ &= \phi^3 X_{t-3} + \phi^2 Z_{t-2} + \phi Z_{t-1} + Z_t \\ &\vdots \\ &= Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2} + \phi^3 Z_{t-3} + \cdots \end{aligned}$$

This can be recognized as an MA(∞) time series model. Representing an AR(1) model as an MA(∞) model is known as *duality*. We now determine the constraints on the parameter ϕ which are required for stationarity. This is the form that is required for causality from Definition 8.2. The coefficients $\psi_1, \psi_2, \dots$ for the AR(1) model from Definition 8.2 are

$$\psi_1 = \phi, \psi_2 = \phi^2, \psi_3 = \phi^3, \dots,$$

or in general, $\psi_j = \phi^j$, for $j = 1, 2, \dots$. Definition 8.2 requires that

$$\sum_{j=1}^{\infty} \psi_j^2 = \sum_{j=1}^{\infty} \phi^{2j} = \phi^2 + \phi^4 + \phi^6 + \cdots < \infty$$

for the time series model to be written in causal form. This summation is a geometric series that converges when $|\phi| < 1$, or equivalently, when $-1 < \phi < 1$, so these are the values of ϕ for which the AR(1) model is causal, which also implies that the model is stationary. Expressing the AR(1) model as an MA(∞) model will also be helpful in the subsequent derivation of the population autocovariance and autocorrelation functions.

Approach 2: Backshift operator. Although the purely algebraic derivation of the causal form of the AR(1) time series model using standard algebra techniques from Approach 1 works fine for establishing stationarity, there is an alternative approach which is slightly more elegant that exploits the backshift operator B . The AR(1) model

$$X_t = \phi X_{t-1} + Z_t$$

can be rewritten as

$$X_t - \phi X_{t-1} = Z_t,$$

which can be expressed using the backshift operator as

$$(1 - \phi B)X_t = Z_t.$$

The first-order polynomial $\phi(B) = 1 - \phi B$ generalizes to a polynomial in B of order p for an $AR(p)$ model. Dividing both sides of this equation by $1 - \phi B$ gives

$$X_t = \frac{Z_t}{1 - \phi B}.$$

For values of ϕ satisfying $-1 < \phi < 1$, this can be recognized as a geometric series in B :

$$X_t = (1 + \phi B + \phi^2 B^2 + \phi^3 B^3 + \cdots) Z_t.$$

Executing the B operator converts this to the form

$$X_t = Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2} + \phi^3 Z_{t-3} + \cdots,$$

which is the same form that we encountered using the successive substitutions in the causality approach.

Approach 3: Unit roots analysis. Theorem 8.3 indicates that all $AR(1)$ models are invertible and they are stationary when the root of

$$\phi(B) = 1 - \phi B = 0$$

lies outside of the unit circle in the complex plane. The solution to this equation is

$$B = \frac{1}{\phi}.$$

This root falls on the real axis in the complex plane and lies outside of the unit circle when

$$-1 < \phi < 1,$$

which is consistent with Approaches 1 and 2.

Approach 4: Definition of stationarity. We can also return to first principles to establish the values of ϕ associated with a stationary $AR(1)$ model. This approach also results in the derivation of the population autocorrelation function. Recall from Definition 7.6 that a time series model is stationary if (a) the expected value of X_t is constant for all t , and (b) the population covariance between X_s and X_t depends only on the lag $|t - s|$. Using the causal formula for the $AR(1)$ time series model expressed as an $MA(\infty)$ time series model from Approach 1, the expected value of X_t is

$$\begin{aligned} E[X_t] &= E[Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2} + \phi^3 Z_{t-3} + \cdots] \\ &= E[Z_t] + \phi E[Z_{t-1}] + \phi^2 E[Z_{t-2}] + \phi^3 E[Z_{t-3}] + \cdots \\ &= 0 \end{aligned}$$

for all values of the parameters ϕ and σ_Z^2 , and all values of t . Again using the causal formula for the $AR(1)$ time series model expressed as an $MA(\infty)$ time series model,

$$\begin{aligned} \gamma(s, t) &= \text{Cov}(X_s, X_t) \\ &= \text{Cov}(Z_s + \phi Z_{s-1} + \phi^2 Z_{s-2} + \cdots, Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2} + \cdots) \\ &= \text{Cov}(Z_s, \phi^{|t-s|} Z_s) + \text{Cov}(\phi Z_{s-1}, \phi^{|t-s|+1} Z_{s-1}) + \text{Cov}(\phi^2 Z_{s-2}, \phi^{|t-s|+2} Z_{s-2}) + \cdots \\ &= \phi^{|t-s|} \sigma_Z^2 + \phi^{|t-s|+2} \sigma_Z^2 + \phi^{|t-s|+4} \sigma_Z^2 + \cdots \\ &= (\phi^{|t-s|} + \phi^{|t-s|+2} + \phi^{|t-s|+4} + \cdots) \sigma_Z^2 \\ &= \phi^{|t-s|} \left(\frac{1}{1 - \phi^2} \right) \sigma_Z^2 \quad |t - s| = 0, 1, 2, \dots \end{aligned}$$

for $-1 < \phi < 1$. Since $E[X_t] = 0$ for all values of t and the population autocovariance function depends only on the lag $|t - s|$, we conclude that the AR(1) process is stationary when $-1 < \phi < 1$. So the population autocovariance function can be expressed in terms of the lag k as

$$\gamma(k) = \left(\frac{\phi^k}{1 - \phi^2} \right) \sigma_Z^2 \quad k = 0, 1, 2, \dots$$

Dividing by the population autocovariance function by

$$\gamma(0) = \left(\frac{1}{1 - \phi^2} \right) \sigma_Z^2$$

gives the population autocorrelation function

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)} = \phi^k \quad k = 0, 1, 2, \dots$$

Based on the four approaches, we now know beyond a shadow of doubt that an AR(1) model is stationary for values of the parameter ϕ satisfying $-1 < \phi < 1$. This derivation constitutes a proof of the following result, which will be stated for just the nonnegative lags. Many authors list the lags as $k = \pm 1, \pm 2, \dots$, but we appeal to Theorem 7.1 to cover the negative lags and only report the nonnegative lags in all of the population autocorrelation functions given in this chapter.

Theorem 9.1 The AR(1) time series model for $\{X_t\}$ is stationary for values of the parameter ϕ satisfying $-1 < \phi < 1$ and σ_Z^2 satisfying $\sigma_Z^2 > 0$ with population autocorrelation function

$$\rho(k) = \phi^k \quad k = 0, 1, 2, \dots$$

The derivation of $\rho(k) = \phi^k$ for $k = 0, 1, 2, \dots$ provides still further evidence of the restriction that $-1 < \phi < 1$. If ϕ were equal to a value outside of this range, say $\phi = 2$, this would result in population correlation values outside of the range $-1 \leq \rho(k) \leq 1$.

For all admissible values of ϕ on the interval $-1 < \phi < 1$, we see from the formula $\rho(k) = \phi^k$ for $k = 0, 1, 2, \dots$ that there will be a geometric decline in the *magnitude* of the values in the population autocorrelation function as the lag k increases. There are two distinct cases for ϕ , however, which will result in population autocorrelation functions with distinctly different shapes. The first case is $0 < \phi < 1$, which gives positive population autocorrelation values at all lags. This is associated with a time series that lingers on one side of the mean. How long it lingers depends on the magnitude of ϕ . Larger values of ϕ indicate that nearby observations will tend to be more likely to be on the same side of the mean, and therefore the time series will tend to linger longer on one side of the mean. The second case is $-1 < \phi < 0$, which gives population autocorrelation function values which alternate in sign and is associated with a time series that is likely to jump from one side of the mean to the other for adjacent observations. These two cases are illustrated in Figure 9.1 for the first 8 lags of the population autocorrelation function for $\phi = 0.8$ and $\phi = -0.8$.

Population Partial Autocorrelation Function

We now determine the population partial autocorrelation function for an AR(1) model. By Definition 7.4, the population lag 0 partial autocorrelation value is $\rho^*(0) = 1$. The population lag 1

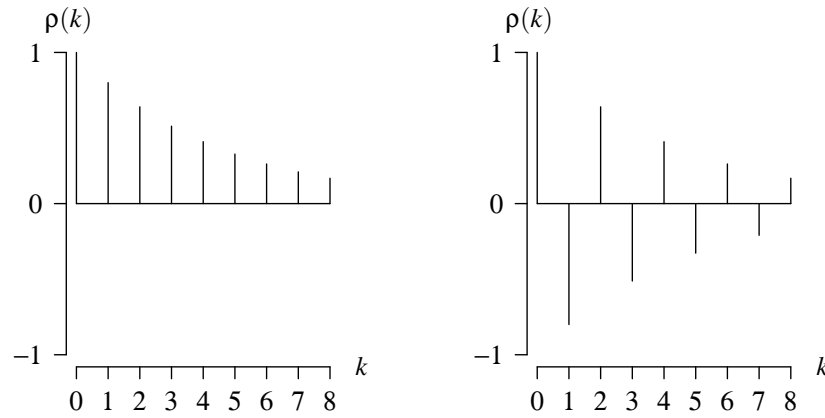


Figure 9.1: AR(1) population autocorrelation functions for $\phi = 0.8$ (left) and $\phi = -0.8$ (right).

partial autocorrelation value is $\rho^*(1) = \rho(1) = \phi$. The population lag 2 partial autocorrelation is

$$\rho^*(2) = \frac{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & \rho(2) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & 1 \end{vmatrix}} = \frac{\begin{vmatrix} 1 & \phi \\ \phi & \phi^2 \end{vmatrix}}{\begin{vmatrix} 1 & \phi \\ \phi & 1 \end{vmatrix}} = 0.$$

This is consistent with the result from Example 7.22 from first principles. Notice that the second column of the matrix in the numerator is a multiple of the first column of the matrix in the numerator. This is why the determinant of the numerator is zero. The population lag 3 partial autocorrelation is

$$\rho^*(3) = \frac{\begin{vmatrix} 1 & \rho(1) & \rho(2) \\ \rho(1) & 1 & \rho(3) \\ \rho(2) & \rho(1) & \rho(3) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) & \rho(2) \\ \rho(1) & 1 & \rho(1) \\ \rho(2) & \rho(1) & 1 \end{vmatrix}} = \frac{\begin{vmatrix} 1 & \phi & \phi^2 \\ \phi & 1 & \phi^3 \\ \phi^2 & \phi & \phi^3 \end{vmatrix}}{\begin{vmatrix} 1 & \phi & \phi^2 \\ \phi & 1 & \phi \\ \phi^2 & \phi & 1 \end{vmatrix}} = 0.$$

Again, the determinant in the numerator is zero because the third column is a multiple of the first column. This pattern continues for the lag k population partial autocorrelation function, which has a first column of the numerator matrix $[1, \phi, \phi^2, \dots, \phi^{k-1}]'$ and last column $[\phi, \phi^2, \phi^3, \dots, \phi^k]'$. Since the last column of the numerator matrix is a multiple of the first column of the numerator matrix, the determinant of the numerator matrix is zero. This constitutes a proof of the following result.

Theorem 9.2 The population partial autocorrelation function for a stationary AR(1) process is

$$\rho^*(k) = \begin{cases} 1 & k = 0 \\ \phi & k = 1 \\ 0 & k = 2, 3, \dots \end{cases}$$

Figure 9.2 shows the first 8 lags of the population partial autocorrelation function for $\phi = 0.8$ and $\phi = -0.8$. These are the same parameter settings as in Figure 9.1. Unlike the population autocorrela-

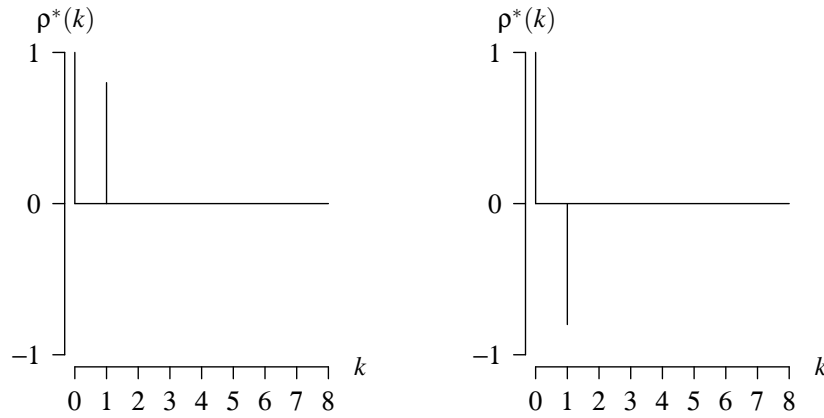


Figure 9.2: Population partial autocorrelation functions for $\phi = 0.8$ (left) and $\phi = -0.8$ (right).

tion function which tails off in magnitude for increasing lags, the population partial autocorrelation cuts off after lag 1. When plotting the corresponding sample analogs, it is typically easier to visually assess a function cutting off rather than tailing off, particularly if there is significant random sampling variability in the observed time series.

The Shifted AR(1) Model

The population mean function for the AR(1) model is $E[X_t] = 0$. This model is not of much use in practice because most real-world time series are not centered around zero. Adding a third parameter μ overcomes this shortcoming. Since population variance and covariance are unaffected by a shift, the associated population autocorrelation and partial autocorrelation functions remain the same as those given in Theorems 9.1 and 9.2. Likewise, the condition for stationarity is unchanged.

Theorem 9.3 A shifted first-order autoregressive model for the time series $\{X_t\}$ is defined by

$$X_t - \mu = \phi(X_{t-1} - \mu) + Z_t,$$

where ϕ , μ , and $\sigma_Z^2 > 0$ are real-valued parameters and $\{Z_t\}$ is a time series of white noise. This model is stationary when $-1 < \phi < 1$. The expected value of X_t is $E[X_t] = \mu$. The population autocorrelation function is

$$\rho(k) = \phi^k \quad k = 0, 1, 2, \dots$$

and the population partial autocorrelation function is

$$\rho^*(k) = \begin{cases} 1 & k = 0 \\ \phi & k = 1 \\ 0 & k = 2, 3, \dots \end{cases}$$

Simulation

An AR(1) time series can be simulated by appealing to the defining formula for the AR(1) model.

Iteratively applying the defining formula for an AR(1) model

$$X_t = \phi X_{t-1} + Z_t$$

results in the simulated values $X_1, X_2, \dots, X_n$. The difficult aspect of this algorithm is how to generate the first value X_1 because there is no X_0 available. For simplicity, assume that the white noise terms are Gaussian white noise. Since the expected value of X_t is $E[X_t] = 0$, the population variance of X_t is

$$V[X_t] = \gamma(0) = \left(\frac{1}{1 - \phi^2} \right) \sigma_Z^2,$$

and linear combinations of mutually independent normally distributed random variables are normal, then the first simulated observation

$$X_1 \sim N\left(0, \left(\frac{1}{1 - \phi^2} \right) \sigma_Z^2\right).$$

The algorithm given as pseudocode below generates an initial time series observation X_1 as indicated above, and then uses an additional $n - 1$ Gaussian white noise terms $Z_2, Z_3, \dots, Z_n$ to generate the remaining time series values $X_2, X_3, \dots, X_n$ using the AR(1) defining formula from Definition 9.1. Indentation denotes nesting in the algorithm.

```

t ← 1
generate  $X_t \sim N\left(0, \left(\frac{1}{1 - \phi^2}\right) \sigma_Z^2\right)$ 
while ( $t < n$ )
  t ← t + 1
  generate  $Z_t \sim N(0, \sigma_Z^2)$ 
   $X_t \leftarrow \phi X_{t-1} + Z_t$ 

```

The three-parameter shifted AR(1) time series model which includes a population mean parameter μ can be simulated by simply adding μ to each time series observation generated by this algorithm. The next example implements this algorithm in R.

Example 9.1 Generate a realization of $n = 100$ observations from an AR(1) time series model with $\phi = 0.8$ and Gaussian white noise error terms with $\sigma_Z^2 = 9$.

Since $\phi = 0.8$ lies in the interval $-1 < \phi < 1$, this is a stationary AR(1) time series model via Theorem 9.1. The first (optional) statement in the R code below uses the `set.seed` function to establish the random number seed. The second statement sets the AR(1) coefficient to $\phi = 0.8$. The third statement sets the standard deviation of the Gaussian white noise to $\sigma_Z = 3$. The fourth statement sets the number of simulated values to $n = 100$. The fifth statement defines the vector `x` of length $n = 100$ to hold the simulated time series values. The sixth statement generates the first simulated time series observation X_1 with a call the `rnorm` function. Finally, the `for` loop iterates through the defining formula for the AR(1) model generating the remaining observations $X_2, X_3, \dots, X_{100}$.

```

set.seed(3)
phi = 0.8
sigz = 3
n = 100
x = numeric(n)
x[1] = rnorm(1, 0, sigz / sqrt(1 - phi ^ 2))
for (t in 2:n) x[t] = phi * x[t - 1] + rnorm(1, 0, sigz)

```

Using the `plot.ts` function to make a plot of the time series contained in `x`, the `acf` function to plot the associated correlogram, the `pacf` function to plot the associated sample partial autocorrelation function, and the `layout` function to arrange the graphs as in Example 7.24, the resulting trio of graphs are displayed in Figure 9.3. The points that have been added to the time series plot can be helpful in identifying patterns. Consistent with an AR(1) model with $\phi = 0.8$ the time series plot shows that the observations tend to have runs of observations that linger above and below the population mean of 0, which is indicated by a horizontal line. The associated sample autocorrelation function tails off as expected from Figure 9.1. The associated sample partial autocorrelation function has a statistically significant spike at lag 1 with $r_1^* = 0.8187$, and then cuts off after lag 1 as expected from Figure 9.2. The spike at lag 1 on both autocorrelation graphs is approximately $\phi = 0.8$, as expected. The 95% confidence intervals indicated by the dashed lines show that the values of the sample partial autocorrelation function do not significantly differ from zero at lags beyond lag 1.

We recommend running the simulation code from the previous example several dozen times in a

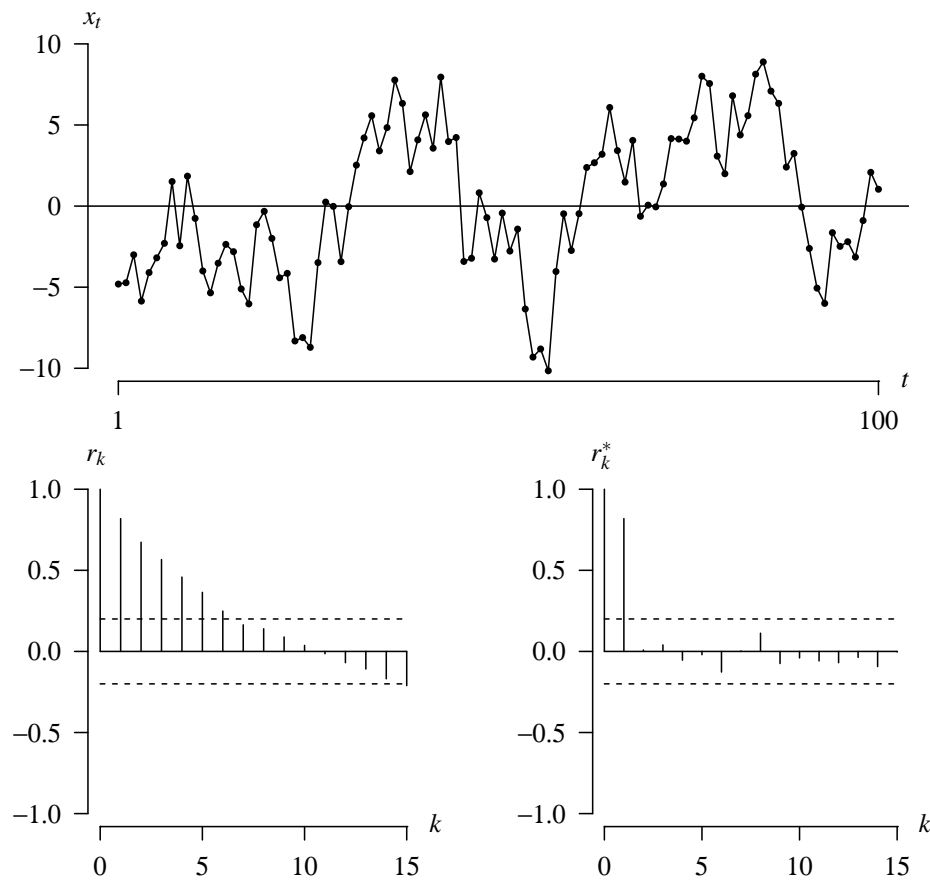


Figure 9.3: Time series plot, r_k , and r_k^* for $n = 100$ simulated values from an AR(1) model.

loop and viewing the associated plots of x_t , r_k , and r_k^* in search of patterns. A call to the R function `Sys.sleep` between the displays of the trio of plots can be used to include an artificial time delay to allow you to inspect the plots. This will allow you to see how various realizations of a simulated AR(1) time series model vary from one realization to the next. So when you then view a *single* realization of a real-life time series, you will have a better sense of how far these plots might deviate from their expected patterns.

There is a second way to simulate observations from an AR(1) time series. This second technique starts the time series at an initial arbitrary value, and then allows the time series to “warm up” or “burn in” for several time periods before producing the first observation X_1 . A reasonable initial arbitrary value for the standard AR(1) model is 0; a reasonable initial arbitrary value for the shifted AR(1) model is μ . This is the approach taken by the built-in R function named `arima.sim` (for autoregressive moving average simulation), which simulates a realization of a time series. Using the `arima.sim` function saves a few keystrokes over the approach taken in the previous example, as illustrated next.

Example 9.2 Generate a realization of $n = 100$ observations from a shifted AR(1) time series model with $\phi = -0.8$, Gaussian white noise error terms with $\sigma_Z^2 = 9$, and mean value $\mu = 10$.

Since there is now a nonzero population mean value, the shifted AR(1) model is

$$X_t - \mu = \phi(X_{t-1} - \mu) + Z_t,$$

where $\mu = 10$, $\phi = -0.8$, and $\sigma_Z = 3$. Since $\phi = -0.8$ lies in the interval $-1 < \phi < 1$, this is a stationary AR(1) time series model. The `model` argument in the `arima.sim` function is a list containing the value of ϕ . Although the default probability distribution for the white noise is normal (that is, Gaussian white noise) with population variance σ_Z^2 , the function allows for other distributions. The second argument to `arima.sim` is `n`, the number of time series observations to be generated. The `sd` argument defines the standard deviation of the white noise. The `n.start` argument gives the number of observations in the warm-up period, which we specify here as 50. The R code to generate $n = 100$ values from the shifted AR(1) model is given below.

```
set.seed(10)
x = 10 + arima.sim(model = list(ar = -0.8), n = 100, sd = 3, n.start = 50)
```

Figure 9.4 shows the three plots associated with the simulated values using the `plot.ts`, `acf`, and `pacf` functions. The time series plot shows a radically different pattern than the time series in the previous example in two manners. First, this simulated time series is centered around $\mu = 10$ (indicated by a horizontal line) rather than $\mu = 0$. Second, adjacent observations in the time series tend to jump from one side of the population mean to other side of the population mean, which is consistent with the population autocorrelation function from the right-hand plot in Figure 9.1. Consistent with the time series plot, the values in the sample autocorrelation function alternate in sign and decrease in magnitude. The sample partial autocorrelation function has a statistically significant spike at lag 1 of $r_1^* = -0.8330$, and nonsignificant spikes thereafter. This is consistent with the right-hand plot in Figure 9.2. Type

```
getAnywhere(arima.sim)
```

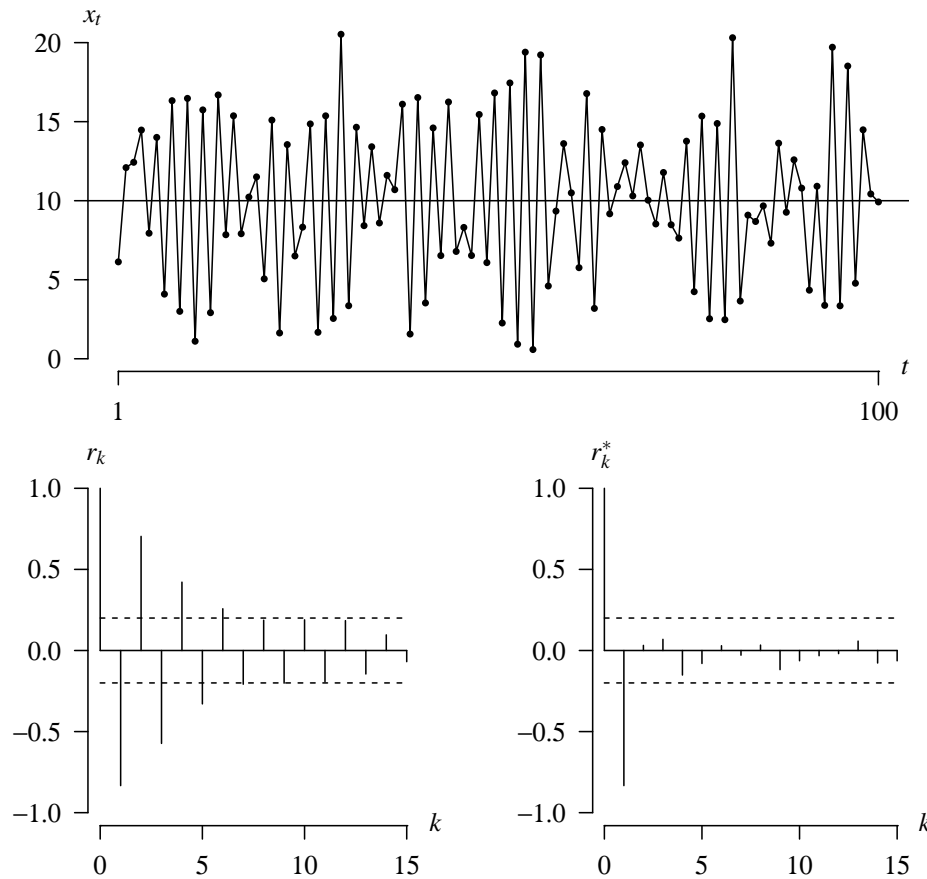


Figure 9.4: Time series plot, r_k , and r_k^* for $n = 100$ simulated values from an AR(1) model.

in order to view the R source code associated with the `arma.sim` function. This function can simulate a realization of any $\text{ARMA}(p, q)$ time series model, and we will use it for simulations subsequently.

The remaining topics associated with the AR(1) time series model are statistical in nature: parameter estimation, model assessment, model selection, and forecasting. A sample time series that will be revisited throughout these topics will be introduced next.

Example 9.3 The temperature in degrees Celsius of a beaver (*Castor canadensis*) in Wisconsin was taken every ten minutes by telemetry on November 3–4, 1990. The resulting time series of $n = 100$ observations is contained in the built-in data frame named `beaver2` in R. The data frame includes columns that contain the temperatures recorded in degrees Celsius (`beaver2$temp`) and an indicator variable (`beaver2$activ`) that reports whether or not the beaver was active outside of its lodge at the associated observation time. The data frame can be viewed by typing `beaver2`. More information about the data set can be viewed by typing `help(beaver2)`. The R statement

```
plot.ts(beaver2$temp)
```

generates the time series plot of the temperature readings given in Figure 9.5. A vertical dashed line has been added between x_{38} and x_{39} to signify when the beaver transitioned from an inactive state to an active state. It is clear that a stationary time series model is *not* appropriate for the entire time series because the population mean appears to increase significantly between the inactive and active periods. So we limit our modeling effort to just the temperatures that were recorded while the beaver was in the active state. The $n = 62$ beaver temperatures during the active period, ordered row-wise, are given in Table 9.1.

37.98	38.02	38.00	38.24	38.10	38.24	38.11	38.02	38.11
38.01	37.91	37.96	38.03	38.17	38.19	38.18	38.15	38.04
37.96	37.84	37.83	37.84	37.74	37.76	37.76	37.64	37.63
38.06	38.19	38.35	38.25	37.86	37.95	37.95	37.76	37.60
37.89	37.86	37.71	37.78	37.82	37.76	37.81	37.84	38.01
38.10	38.15	37.92	37.64	37.70	37.46	37.41	37.46	37.56
37.55	37.75	37.76	37.73	37.77	38.01	38.04	38.07	

Table 9.1: Beaver temperatures at ten-minute intervals in the active state.

The question posed in this example is whether an AR(1) model is appropriate time series model for the 62 temperatures taken during the active period.

The time series of temperatures of the beaver in the active state, the sample autocorrelation function, and the sample partial autocorrelation function can be graphed with the R statements

```
x = beaver2$temp[beaver2$activ == 1]
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
```

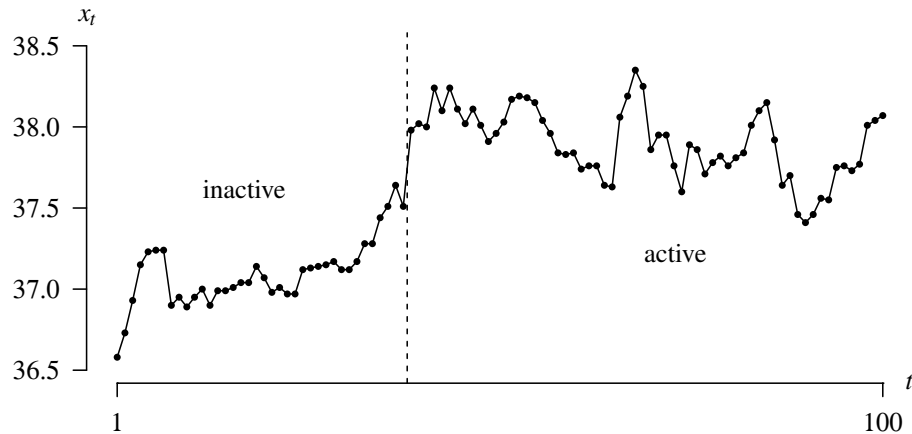


Figure 9.5: Wisconsin beaver temperatures recorded at 10-minute intervals.

```
plot.ts(x)
abline(h = mean(x))
acf(x)
pacf(x)
```

The trio of graphs is displayed in Figure 9.6. A horizontal line has been added to the time series plot at $\bar{x} = 37.9$. A visual assessment of the $n = 62$ observations from the time series indicates that the mean value does not appear to be systematically increasing or decreasing over the time period. From the documentation of the time series, one can see that the active period for the beaver began at 3:50 PM on November 3, 1990 and ended at 2:00 AM on November 4, 1990. The ambient temperature might have an effect on the beaver's temperature, but this will not be pursued further. For now, we will assume that there is no systematic, secular trend in the mean value. The variance of the observations in the time series also seems to be stable over time. Based on this cursory analysis, it appears plausible that the beaver's temperature during the active period could have been drawn from a stationarity time series model. Now we turn to the

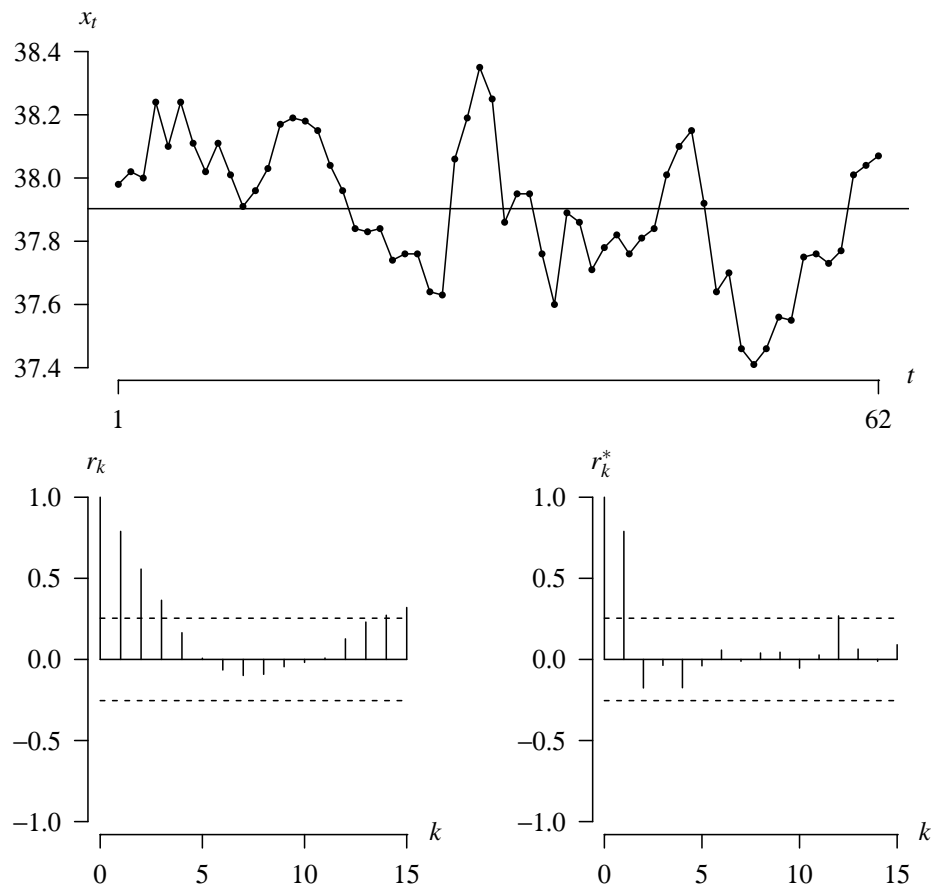


Figure 9.6: Time series plot, r_k , and r_k^* for $n = 62$ temperatures of an active beaver.

interpretation of the sample autocorrelation function and the sample partial autocorrelation function. The sample autocorrelation function has three initial positive statistically significant spikes which decrease in magnitude. The correlogram tails off. This is consistent with an AR(1) model. We discount the barely significant spikes at lags $k = 14$ and $k = 15$, although we could pursue other studies concerning beaver temperatures over time to see if there might be a cyclic variation component present. Furthermore, the partial autocorrelation function has a positive statistically significant spike at lag 1, then cuts off after lag 1. This is also consistent with an AR(1) model. The fact that $r_1 = r_1^* = 0.79$ is positive is consistent with the time series plot of the beaver's temperature, which tends to linger above and below the sample mean value $\bar{x} = 37.9$ for significant periods of time. So far, the evidence points to the AR(1) time series model being a reasonable model for the beaver's temperature during its active period.

The AR(1) model gives us a secondary manner to visually assess whether or not it is an appropriate time series model for the beaver temperatures. Since the shifted AR(1) model is

$$X_t - \mu = \phi (X_{t-1} - \mu) + Z_t,$$

the aforementioned interpretation of this time series model as a simple linear regression model means that a plot of $x_{t-1} - \bar{x}$ versus $x_t - \bar{x}$ (or x_{t-1} versus x_t) should be approximately linear. The plot displayed in Figure 9.7 contains the $n - 1 = 62 - 1 = 61$ pairs (x_{t-1}, x_t) of adjacent points on a set of axes, which is generated by the R statements

```
x = beaver2$temp[beaver2$activ == 1]
n = length(x)
plot(x[2:n], x[1:(n - 1)])
```

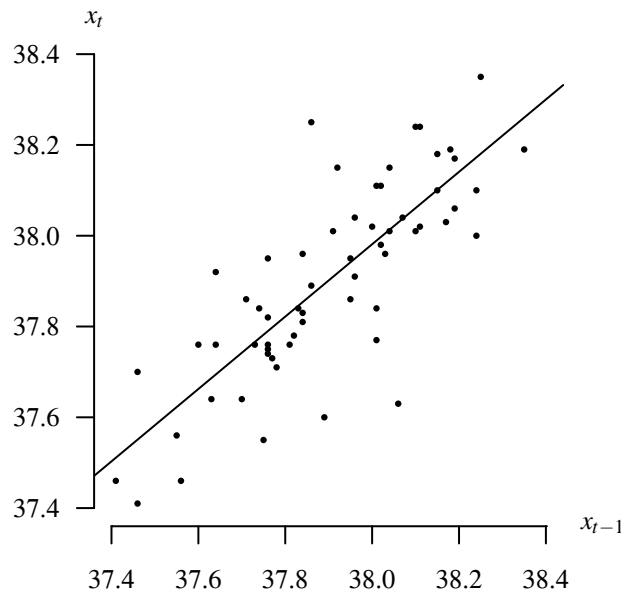


Figure 9.7: Scatterplot of adjacent pairs of temperatures of an active beaver.

```
abline(lm(x[2:n] ~ x[1:(n - 1)]))
```

The paucity of points in the upper-left and lower-right portions of the scatterplot and the approximately linear relationship between the adjacent observations lends further evidence that the AR(1) model might be an appropriate time series model. The `lm` function fits a linear model to the data pairs and the `abline` function plots the associated regression line. The additional R statement

```
summary(lm(x[2:n] ~ x[1:(n - 1)]))
```

indicates that the slope of the regression line differs statistically from 0 with p -value $2 \cdot 10^{-14}$, indicating a strong linear relationship between adjacent observations.

In conclusion, a preliminary graphical analysis of the $n = 62$ temperatures of the beaver in the active state indicates that an AR(1) time series model should be on the short list of potential time series models. The next step is to estimate the parameters in the model.

Parameter Estimation

There are two parameters, ϕ and σ_Z^2 , to estimate in the standard AR(1) model

$$X_t = \phi X_{t-1} + Z_t.$$

There are three parameters, μ , ϕ , and σ_Z^2 , to estimate in the shifted AR(1) model

$$X_t - \mu = \phi (X_{t-1} - \mu) + Z_t.$$

The three parameter estimation techniques outlined in Section 8.2.1 are applied to the shifted AR(1) time series model next.

Approach 1: Method of moments. In the case of the shifted AR(1) model, we match the population and sample (a) first-order moments, (b) second-order moments, and (c) lag 1 autocorrelation. Placing the population moments on the left-hand side of the equation and the associated sample moments on the right-hand side of the equation results in three equations in three unknowns:

$$\begin{aligned} E[X_t] &= \frac{1}{n} \sum_{t=1}^n X_t \\ E[X_t^2] &= \frac{1}{n} \sum_{t=1}^n X_t^2 \\ \rho(1) &= r_1 \end{aligned}$$

or

$$\begin{aligned} \mu &= \bar{X} \\ V[X_t] + E[X_t]^2 &= \gamma(0) + \mu^2 = \left(\frac{1}{1 - \phi^2} \right) \sigma_Z^2 + \mu^2 = \frac{1}{n} \sum_{t=1}^n X_t^2 \\ \phi &= r_1. \end{aligned}$$

These equations can be solved in closed form for the three unknown parameters μ , ϕ , and σ_Z^2 yielding the method of moments estimators

$$\hat{\mu} = \bar{X}, \quad \hat{\phi} = r_1, \quad \hat{\sigma}_Z^2 = (1 - \hat{\phi}^2) \left(\frac{1}{n} \sum_{t=1}^n X_t^2 - \hat{\mu}^2 \right) = (1 - r_1^2) \left(\frac{1}{n} \sum_{t=1}^n X_t^2 - \bar{X}^2 \right).$$

This constitutes a proof of the following result.

Theorem 9.4 The method of moments estimators of the parameters in a shifted AR(1) model are

$$\hat{\mu} = \bar{X} \quad \hat{\phi} = r_1 \quad \hat{\sigma}_Z^2 = (1 - r_1^2) \left(\frac{1}{n} \sum_{t=1}^n X_t^2 - \bar{X}^2 \right).$$

These point estimators are random variables and have been written as a function of the random time series values $X_1, X_2, \dots, X_n$. For observed time series values $x_1, x_2, \dots, x_n$, the lowercase versions of the formulas will be used.

Example 9.4 For the time series of $n = 62$ temperature observations of the beaver in the active state, find the method of moments estimators of μ , ϕ , and σ_Z^2 for the AR(1) model.

The R code below calculates and prints the point estimates of the μ , ϕ , and σ_Z^2 parameters.

```
x      = beaver2$temp[beaver2$activ == 1]
muhat  = mean(x)
phihat = acf(x, plot = FALSE)$acf[2]
sig2hat = (1 - phihat ^ 2) * (mean(x ^ 2) - muhat ^ 2)
print(c(muhat, phihat, sig2hat))
```

The point estimates for the unknown parameters computed by this code are

$$\hat{\mu} = 37.90 \quad \hat{\phi} = 0.7894 \quad \hat{\sigma}_Z^2 = 0.01734.$$

These point estimates are reported to four digits because the data values were given to four-digit accuracy. The positive value for $\hat{\phi}$ is consistent with the fact that the beaver's temperature lingers above and below the sample mean in the time series plot in Figure 9.6. The estimated standard deviation of the white noise error terms,

$$\hat{\sigma}_Z = \sqrt{0.01734} \cong 0.1317,$$

reflects the dispersion of the observations in Figure 9.7 about the regression line.

Approach 2: Least squares. Consider the shifted stationary AR(1) model

$$X_t - \mu = \phi(X_{t-1} - \mu) + Z_t.$$

For least squares estimation, we first establish the sum of squares S as a function of the parameters μ and ϕ and use calculus to find the least squares estimators of μ and ϕ . This will result in a slight difference between the usual pattern of using the sample mean $\bar{x}$ to estimate the population mean μ . Once these least squares estimators have been determined, the population variance of the white noise σ_Z^2 will be estimated.

The sum of squared errors is

$$S = \sum_{t=2}^n Z_t^2 = \sum_{t=2}^n [X_t - \mu - \phi(X_{t-1} - \mu)]^2.$$

The partial derivatives of S with respect to μ and ϕ are

$$\frac{\partial S}{\partial \mu} = \sum_{t=2}^n 2[X_t - \mu - \phi(X_{t-1} - \mu)](-1 + \phi)$$

and

$$\frac{\partial S}{\partial \phi} = \sum_{t=2}^n -2[X_t - \mu - \phi(X_{t-1} - \mu)](X_{t-1} - \mu).$$

Equating the first of the partial derivatives to zero yields

$$\sum_{t=2}^n [X_t - \mu - \phi(X_{t-1} - \mu)] = 0$$

or

$$\sum_{t=2}^n X_t - \phi \sum_{t=2}^n X_{t-1} - (n-1)\mu(1-\phi) = 0$$

or

$$\bar{X}_2 - \phi \bar{X}_1 - \mu(1-\phi) = 0$$

or

$$\hat{\mu} = \frac{\bar{X}_2 - \hat{\phi} \bar{X}_1}{1 - \hat{\phi}},$$

where

$$\bar{X}_1 = \frac{1}{n-1} \sum_{t=1}^{n-1} X_t \quad \text{and} \quad \bar{X}_2 = \frac{1}{n-1} \sum_{t=2}^n X_t.$$

Equating the second of the partial derivatives to zero yields

$$\sum_{t=2}^n (X_t - \mu)(X_{t-1} - \mu) - \phi \sum_{t=2}^n (X_{t-1} - \mu)^2 = 0$$

or

$$\hat{\phi} = \frac{\sum_{t=2}^n (X_t - \hat{\mu})(X_{t-1} - \hat{\mu})}{\sum_{t=2}^n (X_{t-1} - \hat{\mu})^2}.$$

So the ordinary least squares estimators for μ and ϕ can be determined by numerically solving the simultaneous equations

$$\hat{\mu} = \frac{\bar{X}_2 - \hat{\phi} \bar{X}_1}{1 - \hat{\phi}} \quad \text{and} \quad \hat{\phi} = \frac{\sum_{t=2}^n (X_t - \hat{\mu})(X_{t-1} - \hat{\mu})}{\sum_{t=2}^n (X_{t-1} - \hat{\mu})^2}$$

for $\hat{\mu}$ and $\hat{\phi}$.

The last parameter to estimate is σ_Z^2 . Since

$$\gamma(0) = \left(\frac{1}{1 - \phi^2} \right) \sigma_Z^2$$

for an AR(1) time series model, the population variance of the white noise can be expressed as

$$\sigma_Z^2 = (1 - \phi^2) \gamma(0).$$

Replacing ϕ by the estimator r_1 because $\rho(1) = \phi$, and replacing $\gamma(0) = V[X_t]$ by the estimator $c_0 = \frac{1}{n} \sum_{t=1}^n (X_t - \bar{X})^2$ gives the point estimator

$$\hat{\sigma}_Z^2 = (1 - r_1^2) c_0,$$

which matches the method of moments estimator from Theorem 9.4. This derivation constitutes a proof of the following result.

Theorem 9.5 The least squares estimators of the parameters in a shifted AR(1) model are the $\hat{\mu}$ and $\hat{\phi}$ values that satisfy

$$\hat{\mu} = \frac{\bar{X}_2 - \hat{\phi} \bar{X}_1}{1 - \hat{\phi}} \quad \text{and} \quad \hat{\phi} = \frac{\sum_{t=2}^n (X_t - \hat{\mu})(X_{t-1} - \hat{\mu})}{\sum_{t=2}^n (X_{t-1} - \hat{\mu})^2},$$

where $\bar{X}_1 = \frac{1}{n-1} \sum_{t=1}^{n-1} X_t$ and $\bar{X}_2 = \frac{1}{n-1} \sum_{t=2}^n X_t$. The least squares estimator of σ_Z^2 is

$$\hat{\sigma}_Z^2 = (1 - r_1^2) c_0.$$

We now apply these techniques to the beaver temperature data set from Example 9.3.

Example 9.5 Find the least squares estimators of μ , ϕ , and σ_Z^2 for the time series of $n = 62$ beaver temperatures from Example 9.3.

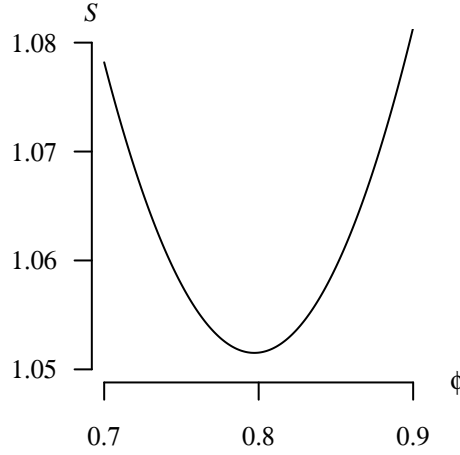
The code below contains a function `s` which calculates the sum of squares, and then uses the R `optim` function to minimize the sum of squares using the method of moments estimates as initial estimates. The `optim` function minimizes the objective function by default.

```
x = beaver2$temp[beaver2$activ == 1]
n = length(x)
s = function(parameters) {
  mu = parameters[1]
  phi = parameters[2]
  sum((x[2:n] - mu - phi * (x[1:(n - 1)] - mu)) ^ 2)
}
optim(c(37.90, 0.7894), s)
r1 = acf(x, plot = FALSE)$acf[2]
sig2hat = (1 - r1 ^ 2) * mean((x - mean(x)) ^ 2)
```

The point estimates for the unknown parameters computed by this code are

$$\hat{\mu} = 37.91 \quad \hat{\phi} = 0.7972 \quad \hat{\sigma}_Z^2 = 0.01762.$$

Figure 9.8 shows the sum of squares for fixed $\hat{\mu} = 37.91$ as a function of ϕ . The sum of squares is minimized at $\hat{\phi} = 0.7972$. The least squares parameter estimates are very close to the method of moments parameter estimators. We now consider why the two estimators are so close to one another.

Figure 9.8: Sum of squares as a function of ϕ .

Since

$$\bar{X}_1 = \frac{1}{n-1} \sum_{t=1}^{n-1} X_t \quad \text{and} \quad \bar{X}_2 = \frac{1}{n-1} \sum_{t=2}^n X_t$$

contain the $n-2$ common values $X_2, X_3, \dots, X_{n-1}$, one approximation that can be applied to the least squares estimates is to assume that $\bar{X}_1 \cong \bar{X}_2 \cong \bar{X}$ for large values of n , which allows for closed-form approximate least squares estimators:

$$\hat{\mu} = \bar{X} \quad \text{and} \quad \hat{\phi} = \frac{\sum_{t=2}^n (X_t - \bar{X})(X_{t-1} - \bar{X})}{\sum_{t=2}^n (X_{t-1} - \bar{X})^2}.$$

As a secondary additional approximation, the denominator of $\hat{\phi}$ with the first approximation in place,

$$\sum_{t=2}^n (X_{t-1} - \bar{X})^2,$$

is approximately equal to

$$\sum_{t=1}^n (X_t - \bar{X})^2$$

for large values of n . With this additional assumption, the least squares estimate for ϕ reduces to the approximate least squares estimate

$$\hat{\phi} = \frac{c_1}{c_0} = r_1,$$

which is the method of moments estimator of ϕ because $\rho(1) = \phi$ for an AR(1) model. With both approximations in place, the least squares estimators exactly match the method of moments estimators. This is why the estimates from the two techniques are so close.

Approach 3: Maximum likelihood estimation. The likelihood function is the joint probability density function of the observed values in the time series $x_1, x_2, \dots, x_n$ in a shifted AR(1) model is

$$L(\mu, \phi, \sigma_Z^2) = f(x_1, x_2, \dots, x_n),$$

where the $x_1, x_2, \dots, x_n$ arguments on L and the μ, ϕ , and σ_Z^2 arguments on f have been dropped for brevity. It is not possible to simply multiply the marginal probability density functions because the values in the AR(1) time series model are correlated. In order to use maximum likelihood estimation, we make the additional assumption that the white noise terms $Z_1, Z_2, \dots, Z_n$ are in fact Gaussian white noise terms:

$$f_{Z_t}(z_t) = \frac{1}{\sqrt{2\pi\sigma_Z^2}} e^{-z_t^2/(2\sigma_Z^2)} \quad -\infty < z_t < \infty$$

for $t = 1, 2, \dots, n$, which is the probability density function of a $N(0, \sigma_Z^2)$ random variable. Ignoring Z_1 temporarily, the joint probability density function of the mutually independent white noise random variables $Z_2, Z_3, \dots, Z_n$ is

$$f_{Z_2, Z_3, \dots, Z_n}(z_2, z_3, \dots, z_n) = (2\pi\sigma_Z^2)^{-(n-1)/2} e^{-\sum_{t=2}^n z_t^2/(2\sigma_Z^2)}$$

for $(z_2, z_3, \dots, z_n) \in \mathcal{R}^{n-1}$. The shifted AR(1) model

$$X_t - \mu = \phi(X_{t-1} - \mu) + Z_t$$

applies for all values of t , so

$$\begin{aligned} X_2 - \mu &= \phi(X_1 - \mu) + Z_2 \\ X_3 - \mu &= \phi(X_2 - \mu) + Z_3 \\ &\vdots \\ X_n - \mu &= \phi(X_{n-1} - \mu) + Z_n. \end{aligned}$$

Solving these equations for $X_2, X_3, \dots, X_n$, consider the transformation of the $Z_2, Z_3, \dots, Z_n$ values

$$\begin{aligned} X_2 &= \mu + \phi(X_1 - \mu) + Z_2 \\ X_3 &= \mu + \phi(X_2 - \mu) + Z_3 \\ &\vdots \\ X_n &= \mu + \phi(X_{n-1} - \mu) + Z_n \end{aligned}$$

conditioned on $X_1 = x_1$, which is a one-to-one transformation from $\mathcal{R}^{n-1}$ to $\mathcal{R}^{n-1}$ with inverse transformation

$$\begin{aligned} Z_2 &= X_2 - \mu - \phi(X_1 - \mu) \\ Z_3 &= X_3 - \mu - \phi(X_2 - \mu) \\ &\vdots \\ Z_n &= X_n - \mu - \phi(X_{n-1} - \mu) \end{aligned}$$

and Jacobian

$$J = \begin{vmatrix} 1 & -\phi & 0 & \dots & 0 & 0 \\ 0 & 1 & -\phi & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -\phi \\ 0 & 0 & 0 & \dots & 0 & 1 \end{vmatrix} = 1.$$

By the transformation technique, the joint probability density function of $X_2, X_3, \dots, X_n$ conditioned on $X_1 = x_1$ is

$$f_{X_2, X_3, \dots, X_n | X_1}(x_2, x_3, \dots, x_n | X_1 = x_1) = (2\pi\sigma_Z^2)^{-(n-1)/2} e^{-\sum_{t=2}^n [x_t - \mu - \phi(x_{t-1} - \mu)]^2 / (2\sigma_Z^2)}$$

for $(x_2, x_3, \dots, x_n) \in \mathcal{R}^{n-1}$ and $x_1 \in \mathcal{R}$. The final step in the derivation of the likelihood function involves determining the marginal distribution of X_1 . Since

$$X_1 \sim N\left(\mu, \frac{\sigma_Z^2}{1 - \phi^2}\right),$$

the probability density function of X_1 is

$$f_{X_1}(x_1) = \sqrt{\frac{1 - \phi^2}{2\pi\sigma_Z^2}} e^{-(1 - \phi^2)(x_1 - \mu)^2 / (2\sigma_Z^2)} \quad -\infty < x_1 < \infty.$$

The joint probability density function of $X_1, X_2, \dots, X_n$ is the product of the conditional probability density function and the marginal probability density function:

$$f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = f_{X_2, X_3, \dots, X_n | X_1}(x_2, x_3, \dots, x_n | X_1 = x_1) f_{X_1}(x_1)$$

for $(x_1, x_2, \dots, x_n) \in \mathcal{R}^n$. So the likelihood function is

$$L(\mu, \phi, \sigma_Z^2) = (2\pi\sigma_Z^2)^{-n/2} \sqrt{1 - \phi^2} e^{-S(\mu, \phi) / (2\sigma_Z^2)},$$

where the *unconditional sum of squares* is

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

The associated log likelihood function is

$$\ln L(\mu, \phi, \sigma_Z^2) = -\frac{n}{2} \ln(2\pi\sigma_Z^2) + \frac{1}{2} \ln(1 - \phi^2) - \frac{S(\mu, \phi)}{2\sigma_Z^2}.$$

The maximum likelihood estimators $\hat{\mu}$, $\hat{\phi}$, and $\hat{\sigma}_Z^2$ satisfy

$$\begin{aligned} \frac{\partial \ln L(\mu, \phi, \sigma_Z^2)}{\partial \mu} &= \frac{(1 - \phi^2)(x_1 - \mu) + (1 - \phi) \sum_{t=2}^n [(x_t - \mu) - \phi(x_{t-1} - \mu)]}{\sigma_Z^2} = 0 \\ \frac{\partial \ln L(\mu, \phi, \sigma_Z^2)}{\partial \phi} &= -\frac{\phi}{1 - \phi^2} + \frac{\phi(x_1 - \mu)^2 + \sum_{t=2}^n [(x_t - \mu) - \phi(x_{t-1} - \mu)](x_{t-1} - \mu)}{\sigma_Z^2} = 0 \\ \frac{\partial \ln L(\mu, \phi, \sigma_Z^2)}{\partial \sigma_Z^2} &= -\frac{n}{2\sigma_Z^2} + \frac{S(\mu, \phi)}{2\sigma_Z^4} = 0. \end{aligned}$$

Although the third equation satisfies

$$\hat{\sigma}_Z^2 = \frac{S(\hat{\mu}, \hat{\phi})}{n},$$

numerical methods are required to solve the equations.

Theorem 9.6 The maximum likelihood estimators of the parameters in a shifted AR(1) model with Gaussian white noise are the $\hat{\mu}$, $\hat{\phi}$, and $\hat{\sigma}_Z^2$ values that satisfy

$$\begin{aligned}\frac{\partial \ln L(\mu, \phi, \sigma_Z^2)}{\partial \mu} &= \frac{(1 - \phi^2)(x_1 - \mu) + (1 - \phi) \sum_{t=2}^n [(x_t - \mu) - \phi(x_{t-1} - \mu)]}{\sigma_Z^2} = 0 \\ \frac{\partial \ln L(\mu, \phi, \sigma_Z^2)}{\partial \phi} &= -\frac{\phi}{1 - \phi^2} + \frac{\phi(x_1 - \mu)^2 + \sum_{t=2}^n [(x_t - \mu) - \phi(x_{t-1} - \mu)](x_{t-1} - \mu)}{\sigma_Z^2} = 0 \\ \frac{\partial \ln L(\mu, \phi, \sigma_Z^2)}{\partial \sigma_Z^2} &= -\frac{n}{2\sigma_Z^2} + \frac{S(\mu, \phi)}{2\sigma_Z^4} = 0.\end{aligned}$$

Maximum likelihood estimation will be illustrated in the next example.

Example 9.6 Find the maximum likelihood estimators of μ , ϕ , and σ_Z^2 for the time series of $n = 62$ beaver temperatures from Example 9.3.

The R code below again uses the `optim` function to maximize the likelihood function. Since the default for `optim` is to minimize, the likelihood function is negated within the `L` function. The method of moments estimators are used as initial estimates. The `optim` function uses the Nelder–Mead method by default to maximize the likelihood function.

```
x = beaver2$temp[beaver2$activ == 1]
n = length(x)
L = function(parameters) {
  mu = parameters[1]
  phi = parameters[2]
  sig2 = parameters[3]
  -(2 * pi * sig2) ^ (- n / 2) * (1 - phi ^ 2) ^ (1 / 2) *
  exp(-((1 - phi ^ 2) * (x[1] - mu) ^ 2 +
  sum(((x[2:n] - mu) - phi * (x[1:(n - 1)] - mu)) ^ 2) / (2 * sig2)))
}
optim(c(37.90, 0.7894, 0.01734), L)
```

The point estimates for the unknown parameters computed by this code are

$$\hat{\mu} = 37.91 \quad \hat{\phi} = 0.7850 \quad \hat{\sigma}_Z^2 = 0.01697.$$

Table 9.2 summarizes the point estimators that have been calculated in the previous three examples for the $n = 62$ beaver temperatures. The point estimators associated with the three methods are quite close for this particular time series.

Method	$\hat{\mu}$	$\hat{\phi}$	$\hat{\sigma}_Z^2$
Method of moments	37.90	0.7894	0.01734
Ordinary least squares	37.91	0.7972	0.01762
Maximum likelihood estimation	37.91	0.7850	0.01697

Table 9.2: Point estimators for the AR(1) parameters for the $n = 62$ beaver temperatures.

The R function `ar` fits autoregressive models. The parameter estimates from the three previous examples could have been calculated with the following four R statements.

```
x = beaver2$temp[beaver2$activ == 1]
ar(x, order.max = 1, aic = FALSE, method = "yule-walker")
ar(x, order.max = 1, aic = FALSE, method = "ols")
ar(x, order.max = 1, aic = FALSE, method = "mle")
```

Table 9.3 contains the point estimates returned by the `ar` function. The tiny differences between some of the entries in Tables 9.2 and 9.3 might be due to slightly different approximations and/or roundoff in the optimization routines.

Method	$\hat{\mu}$	$\hat{\phi}$	$\hat{\sigma}_Z^2$
Method of moments (Yule–Walker)	37.90	0.7894	0.01792
Ordinary least squares	37.90	0.7972	0.01724
Maximum likelihood estimation	37.90	0.7865	0.01699

Table 9.3: Point estimators for the $n = 62$ beaver temperatures via the `ar` function.

We have now derived and illustrated the three point estimation techniques, the method of moments, least squares, and maximum likelihood estimation, for the parameters in an AR(1) model from a realization of a time series consisting of n observations. Which of these techniques provides the best point estimators? This is not an easy question to answer because there are a large number of factors, such as the sample size n , the values of the parameters in the model, and the fact that there are three parameters to estimate. There will not necessarily be one universal answer to the question. We do a focused evaluation on the point estimator for ϕ because it typically differs for the three methods of point estimation. The mean square error associated with the point estimate for ϕ is

$$E \left[(\hat{\phi} - \phi)^2 \right].$$

The following R code conducts a Monte Carlo simulation experiment which estimates the mean square error of the three point estimators for ϕ for 40,000 replications. We selected the time series model with $\mu = 38$, $\phi = 0.8$, $\sigma_Z = 0.13$, and $n = 62$, which are parameters that are near the estimated parameters in the last three examples involving the time series of beaver temperatures.

```
nrep = 40000
mse.mom = 0
mse.ols = 0
mse.mle = 0
for (i in 1:nrep) {
  x = 38 + arima.sim(model = list(ar = 0.8, sd = 0.13), n = 62, n.start = 10)
  mse.mom = mse.mom + (ar(x, order.max = 1, method = "yw")$ar[1] - 0.8) ^ 2
  mse.ols = mse.ols + (ar(x, order.max = 1, method = "ols")$ar[1] - 0.8) ^ 2
  mse.mle = mse.mle + (ar(x, order.max = 1, method = "mle")$ar[1] - 0.8) ^ 2
}
print(c(mse.mom, mse.ols, mse.mle) / nrep)
```

After a call to `set.seed(4)` to establish the random number stream, three runs of this simulation yielded the following estimated mean squared error values:

Method of moments : 0.0135 Least squares : 0.0117 Maximum likelihood : 0.0113.

Furthermore, confidence intervals for the three methods do not overlap. Since small values of the mean square error are preferred, we conclude that the maximum likelihood estimator is the preferred estimator for these parameter settings, followed by the least squares estimator, followed by the method of moments estimator in a distant third place.

The focus on estimation thus far has been on point estimation techniques. We also want to report some indication of the *precision* associated with these point estimators. In the previous example, the sampling distributions of $\hat{\mu}$, $\hat{\phi}$, and $\hat{\sigma}_Z^2$ in the AR(1) model are too complicated to derive analytically. As an illustration of a confidence interval for one of the parameters, we use the asymptotic normality of the maximum likelihood estimator of ϕ in the result:

$$\hat{\phi} \xrightarrow{D} N\left(\phi, \frac{1-\phi^2}{n}\right).$$

This result leads to an asymptotically exact two-sided $100(1-\alpha)\%$ confidence interval for ϕ .

Theorem 9.7 For a stationary AR(1) time series model, an asymptotically exact two-sided $100(1-\alpha)\%$ confidence interval for ϕ is given by

$$\hat{\phi} - z_{\alpha/2} \sqrt{\frac{1-\hat{\phi}^2}{n}} < \phi < \hat{\phi} + z_{\alpha/2} \sqrt{\frac{1-\hat{\phi}^2}{n}},$$

where $\hat{\phi}$ is the maximum likelihood estimator of ϕ and $z_{\alpha/2}$ is the $1-\alpha/2$ fractile of the standard normal distribution.

This asymptotically exact confidence interval will now be illustrated with the time series of active beaver temperatures from the three previous examples.

Example 9.7 Find an approximate 95% confidence interval for ϕ for the AR(1) model associated with the $n = 62$ beaver temperature time series values from Example 9.3.

Recall from Table 9.3 that the maximum likelihood estimator returned by the `ar` function is $\hat{\phi} = 0.7865$. The following R code calculates a 95% confidence interval for ϕ .

```
x      = beaver2$temp[beaver2$activ == 1]
n      = length(x)
mle    = ar(x, order.max = 1, aic = FALSE, method = "mle")$ar
alpha  = 0.05
crit   = qnorm(1 - alpha / 2)
lo     = mle - crit * sqrt((1 - mle ^ 2) / n)
hi     = mle + crit * sqrt((1 - mle ^ 2) / n)
print(c(lo, hi))
```

This code returns the approximate 95% confidence interval

$$0.6328 < \phi < 0.9402$$

which does not contain $\phi = 0$, giving further evidence that the AR(1) model is justified. An AR(1) model with $\phi = 0$ reduces to just white noise, and the beaver temperature time series is clearly not comprised of mutually independent observations.

Model Assessment

Now that techniques for point and interval estimates for the parameters in the AR(1) model have been established, we are interested in assessing the adequacy of the AR(1) time series model. This will involve an analysis of the residuals. Recall from Section 8.2.3 that the residuals are defined by

$$[\text{residual}] = [\text{observed value}] - [\text{predicted value}]$$

or

$$\hat{Z}_t = X_t - \hat{X}_t.$$

Since $\hat{X}_t$ is the one-step-ahead forecast from the time origin $t - 1$, this is more clearly written as

$$\hat{Z}_t = X_t - \hat{X}_{t-1}(1).$$

Therefore, for the time series $x_1, x_2, \dots, x_n$ and the fitted AR(1) model with parameter estimates $\hat{\mu}$ and $\hat{\phi}$, the residual at time t is

$$\hat{Z}_t = x_t - [\hat{\mu} + \hat{\phi}(x_{t-1} - \hat{\mu})]$$

for $t = 2, 3, \dots, n$ via Example 8.12. The next example shows the steps associated with assessing the adequacy of the AR(1) model for the active beaver temperature time series.

Example 9.8 Fit the AR(1) model to the active beaver temperatures from Example 9.3 using the sample mean to estimate μ and the maximum likelihood estimators for ϕ and σ^2 . Assess the fitted AR(1) model by the following five methods.

- Calculate and plot the residuals, their sample autocorrelation function, and their sample partial autocorrelation function.
 - Conduct a test of independence on the residuals using the number of sample autocorrelation function values for the first $m = 40$ lags which fall outside of $\pm 1.96/\sqrt{n}$.
 - Conduct the Box–Pierce and Ljung–Box tests for independence of the residuals.
 - Conduct the turning point test for independence of the residuals.
 - Plot a histogram and a QQ plot of the standardized residuals in order to assess the normality of the residuals.
- The following R commands calculate the $n - 1 = 61$ residuals and plot them as a time series, along with the associated sample autocorrelation function and sample partial autocorrelation function.

```
x      = beaver2$temp[beaver2$activ == 1]
n      = length(x)
m      = 40
muhat  = mean(x)
phihat = ar(x, order.max = 1, aic = FALSE, method = "mle")$ar
zhat   = x[2:n] - (muhat + phihat * (x[1:(n - 1)] - muhat))
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(zhat)
acf(zhat, lag.max = m)
pacf(zhat, lag.max = m)
```

The results are displayed in Figure 9.9. The residuals do not appear to have any cyclic variation, trend, or serial correlation.

- (b) There is just one sample autocorrelation function value that falls outside of the limits $\pm 1.96/\sqrt{n}$ (at lag 15) in the plot in Figure 9.9 of the first 40 sample autocorrelation function values associated with the residuals. Since we expect $40 \cdot 0.05 = 2$ values to fall outside of these limits in the case of a good fit, we fail to reject H_0 in this case. The adequacy of the fit of the AR(1) model is not rejected by this test.
- (c) The additional R code below calculates the Box–Pierce and Ljung–Box test statistics and the associated p -values.

```
n      = length(zhat)
r      = acf(zhat, lag.max = m, plot = FALSE)$acf[2:(m + 1)]
boxpierce = n * sum(r ^ 2)
1 - pchisq(boxpierce, m - 2)
ljungbox  = n * (n + 2) * sum(r ^ 2 / seq(n - 1, n - m))
1 - pchisq(ljungbox, m - 2)
```

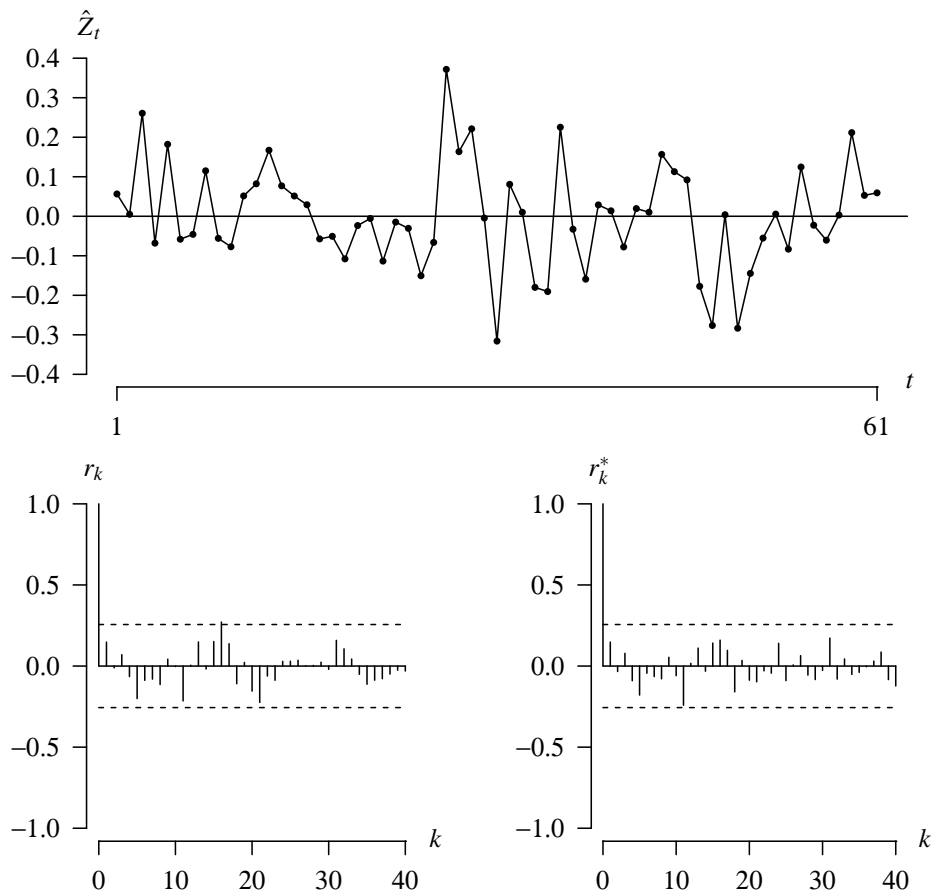


Figure 9.9: Time series plot, r_k , and r_k^* for $n - 1 = 61$ residuals from AR(1) fitted model.

The Box–Pierce test statistic is 27.9 and the associated p -value is $p = 0.89$. The Ljung–Box test statistic is 41.8 and the associated p -value is $p = 0.31$. We fail to reject H_0 in both tests based on the chi-square critical value with $40 - 2 = 38$ degrees of freedom. Some keystrokes can be saved by using the built-in `Box.test` function in R as shown below.

```
Box.test(zhat, lag = 40, type = "Box-Pierce", fitdf = 2)
Box.test(zhat, lag = 40, type = "Ljung-Box", fitdf = 2)
```

The `Box.test` function delivers identical test statistics and p -values. The adequacy of the fit of the AR(1) model is not rejected by these tests.

- (d) The following additional R statements calculate the test statistic and the p -value for the turning point test applied to the time series consisting of the $n - 1 = 61$ residual values for the AR(1) fit to the beaver temperatures in the active state.

```
n = n - 1
m = (2 / 3) * (n - 2)
v = (16 * n - 29) / 90
T = 0
for (i in 2:(n - 1)) {
  if ((zhat[i - 1] < zhat[i] && zhat[i] > zhat[i + 1]) ||
      (zhat[i - 1] > zhat[i] && zhat[i] < zhat[i + 1])) T = T + 1
}
s = (T - m) / sqrt(v)
2 * (1 - pnorm(abs(s)))
```

The tail probability is doubled because the alternative hypothesis is two-tailed for the turning point test. The test statistic s is -0.83 and the p -value is $p = 0.41$. We again fail to reject the null hypothesis in this case. The adequacy of the fit of the AR(1) model is not rejected by this test.

- (e) The residuals are standardized by dividing by their sample standard deviation. The following additional R statements plot a histogram of the standardized residuals using the `hist` function and a QQ plot to assess normality using the `qqnorm` function.

```
par(mfrow = c(1, 2))
hist(zhat / sd(zhat))
qqnorm(zhat / sd(zhat))
```

The plots are shown in Figure 9.10. The histogram shows that all standardized residuals fall between -3 and 3 and exhibit a bell-shaped probability distribution. The horizontal axis on the histogram is the standardized residual and the vertical axis is the frequency. The QQ plot is approximately linear, indicating a reasonable approximation to normality based on the $n - 1 = 61$ residuals plotted. The horizontal axis on the QQ plot is the standardized theoretical quantile and the vertical axis is the associated normal data quantile. Although a formal statistical goodness-of-fit test should be conducted, it appears that the assumption of Gaussian white noise is appropriate for the AR(1) time series model based on these two plots.

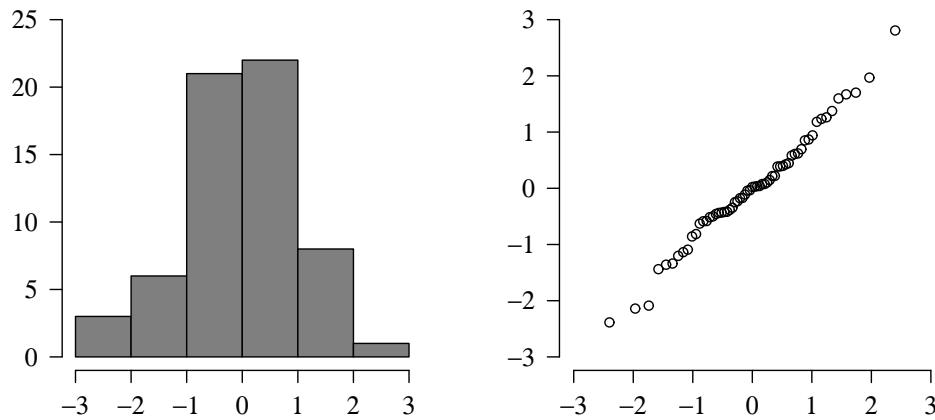


Figure 9.10: Histogram (left) and QQ plot (right) of the fitted AR(1) standardized residuals.

We have seen a number of indicators that the AR(1) time series model is an adequate model for the active beaver temperatures. But how do we know that there is not a better model with more terms lurking below the surface that might provide a better fit? The next subsection considers the process of model selection.

Model Selection

One way of eliminating the possibility of a better time series model is to overfit the tentative AR(1) time series model with ARMA(p, q) models of higher order. We have not yet surveyed the techniques for estimating the parameters in these models with additional terms, so for now we will let the `arma` function in R estimate their parameters and compare them via their AIC (Akaike's Information Criterion) statistics. The AIC statistic was introduced in Section 8.2.4.

Example 9.9 For the $n = 62$ temperatures of an active beaver given in Example 9.3, find the ARMA(p, q) model that minimizes the AIC.

The R code below creates a 4×4 matrix `a` that will be populated with the AIC statistics for the ARMA(p, q) time series models, for $p = 0, 1, 2, 3$ and $q = 0, 1, 2, 3$ using nested `for` loops. The `arma` function is used to fit the models via maximum likelihood estimation, whose AIC values are placed in the matrix `a`.

```
a = matrix(0, 4, 4)
x = beaver2$temp[beaver2$activ == 1]
for (p in 0:3)
  for (q in 0:3)
    a[p + 1, q + 1] = arima(x, order = c(p, 0, q), method = "ML")$aic
```

The results of this code are given in Table 9.4. The smallest AIC value (barely!) is set in boldface type and corresponds to the AR(1) model. This provides further evidence that the AR(1) model is adequate, and a more complex model is probably not warranted. Although the AR(2) and ARMA(1, 1) models have nearly identical AIC values, the additional parameter did not overcome the penalty inflicted by AIC for its inclusion in the time series model.

	$q = 0$	$q = 1$	$q = 2$	$q = 3$
$p = 0$	-10.9	-51.1	-60.9	-64.4
$p = 1$	-69.7	-69.6	-67.6	-67.0
$p = 2$	-69.6	-67.6	-65.6	-66.4
$p = 3$	-67.7	-65.7	-65.1	-64.7

Table 9.4: AIC statistics for ARMA(p, q) models for the $n = 62$ beaver temperatures.

The `$` extractor with the `aic` argument was used to extract the AIC statistics from the results of the call to `arima`. If the `coef` and `sigma2` components are extracted from the list returned by the call to `arima`, our final model is the AR(1) model with maximum likelihood estimates for the parameters given by

$$\hat{\mu} = 37.9, \quad \hat{\phi} = 0.787, \quad \hat{\sigma}_Z^2 = 0.017.$$

The final model is therefore

$$X_t - 37.9 = 0.787(X_{t-1} - 37.9) + Z_t,$$

where Z_t is a time series of Gaussian white noise values with $\sigma_Z^2 = 0.017$, as established by the histogram and QQ plot in Example 9.8.

In some applications, just describing the time series model for the beaver temperatures in the active state with the fitted AR(1) model is adequate. In other applications, simulating the values in the fitted AR(1) model is the goal. But in many application areas, particularly economics, there is often an interest in forecasting future values of a time series from a realization. In our setting, we might be interested in this particular beaver's future temperature based on the $n = 62$ temperature values collected. The next subsection considers forecasting for the AR(1) model.

Forecasting

We now pivot to the development of a procedure to forecast future values of a time series that is governed by an AR(1) model. To review the notation for forecasting, the observed time series values are $x_1, x_2, \dots, x_n$. The forecast is being made at time $t = n$. The random future value of the time series that is h time units in the future is denoted by X_{n+h} . The associated forecasted value is denoted by $\hat{X}_{n+h}$, and is the conditional expected value

$$\hat{X}_{n+h} = E[X_{n+h} | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n].$$

We would like to calculate this forecasted value and an associated prediction interval for the AR(1) model. As in Section 8.2.2, we assume that all parameters are known in the derivations that follow.

Recall from Example 8.12 that the forecasted value for one time unit into the future for a shifted AR(1) model is

$$\hat{X}_{n+1} = \mu + \phi(x_n - \mu).$$

We would like to generalize this so as to find the forecasted value h time units into the future. In other words, we want to find $\hat{X}_{n+h}$. The shifted AR(1) model is

$$X_t - \mu = \phi(X_{t-1} - \mu) + Z_t.$$

Replacing t by $n+h$, which is the time value of interest, gives

$$X_{n+h} - \mu = \phi (X_{n+h-1} - \mu) + Z_{n+h}.$$

Taking the conditional expected value of each side of this equation results in

$$\hat{X}_{n+h} = \mu + \phi (\hat{X}_{n+h-1} - \mu).$$

Iterating on this equation for time values that are sequentially one time unit closer to the present time $t = n$ yields

$$\begin{aligned} \hat{X}_{n+h} &= \mu + \phi (\hat{X}_{n+h-1} - \mu) \\ &= \mu + \phi [\mu + \phi (\hat{X}_{n+h-2} - \mu) - \mu] \\ &= \mu + \phi^2 (\hat{X}_{n+h-2} - \mu) \\ &\vdots \\ &= \mu + \phi^{h-1} (\hat{X}_{n+1} - \mu) \\ &= \mu + \phi^{h-1} [\mu + \phi (x_n - \mu) - \mu] \\ &= \mu + \phi^h (x_n - \mu). \end{aligned}$$

Notice that the forecasted value is a function of x_n , but not a function of $x_1, x_2, \dots, x_{n-1}$. This is a sensible forecast in the sense that for a long time horizon h into the future and a stationary shifted AR(1) model with $-1 < \phi < 1$,

$$\lim_{h \rightarrow \infty} \hat{X}_{n+h} = \mu.$$

If you were asked to forecast your temperature one year from now, you would probably say 98.6° Fahrenheit (or whatever your average temperature might be), regardless of whether you are healthy or have a fever right now. Long-term forecasts for stationary time series models always tend to the population mean.

As is typically the case in statistics, we would like to pair our point estimator $\hat{X}_{n+h}$ with an interval estimator, which is a prediction interval in this setting. The prediction interval gives us an indication of the precision of the forecast. In order to derive an exact two-sided $100(1 - \alpha)\%$ prediction interval for X_{n+h} , it is helpful to write the shifted AR(1) model as a shifted MA(∞) model. Using successive substitutions, each one time unit prior to the previous substitution,

$$\begin{aligned} X_t - \mu &= \phi (X_{t-1} - \mu) + Z_t \\ &= \phi [\phi (X_{t-2} - \mu) + Z_{t-1}] + Z_t \\ &= \phi^2 (X_{t-2} - \mu) + Z_t + \phi Z_{t-1} \\ &= \phi^2 [(\phi X_{t-3} - \mu) + Z_{t-2}] + Z_t + \phi Z_{t-1} \\ &= \phi^3 (X_{t-3} - \mu) + Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2} \\ &\vdots \end{aligned}$$

For $-1 < \phi < 1$ corresponding to a stationary shifted AR(1) model, the limiting expression for X_t is

$$X_t = \mu + Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2} + \dots,$$

which is a shifted MA(∞) model. Replacing t with $n+h$ results in

$$X_{n+h} = \mu + Z_{n+h} + \phi Z_{n+h-1} + \phi^2 Z_{n+h-2} + \cdots,$$

Taking the conditional variance of both sides of this equation yields

$$\begin{aligned} V[X_{n+h} \mid X_1 = x_1, X_2 = x_2, \dots, X_n = x_n] \\ &= V[\mu + Z_{n+h} + \phi Z_{n+h-1} + \phi^2 Z_{n+h-2} + \cdots \mid X_1 = x_1, X_2 = x_2, \dots, X_n = x_n] \\ &= \sigma_Z^2 + \phi^2 \sigma_Z^2 + \phi^4 \sigma_Z^2 + \cdots + \phi^{2h-2} \sigma_Z^2 \\ &= (1 + \phi^2 + \phi^4 + \cdots + \phi^{2h-2}) \sigma_Z^2 \\ &= \frac{1 - \phi^{2h}}{1 - \phi^2} \sigma_Z^2 \end{aligned}$$

because the error terms at time n and prior are observed and can therefore be treated as constants. Assuming Gaussian white noise terms, an *exact* two-sided $100(1 - \alpha)\%$ prediction interval for X_{n+h} is

$$\hat{X}_{n+h} - z_{\alpha/2} \sqrt{\frac{1 - \phi^{2h}}{1 - \phi^2}} \sigma_Z < X_{n+h} < \hat{X}_{n+h} + z_{\alpha/2} \sqrt{\frac{1 - \phi^{2h}}{1 - \phi^2}} \sigma_Z.$$

In most practical problems, the parameters in this prediction interval will be estimated from data, which results in the following *approximate* two-sided $100(1 - \alpha)\%$ prediction interval.

Theorem 9.8 For a stationary shifted AR(1) time series model, the forecasted value of X_{n+h} is

$$\hat{X}_{n+h} = \hat{\mu} + \hat{\phi}^h (x_n - \hat{\mu}).$$

and an approximate two-sided $100(1 - \alpha)\%$ prediction interval for X_{n+h} is

$$\hat{X}_{n+h} - z_{\alpha/2} \sqrt{\frac{1 - \hat{\phi}^{2h}}{1 - \hat{\phi}^2}} \hat{\sigma}_Z < X_{n+h} < \hat{X}_{n+h} + z_{\alpha/2} \sqrt{\frac{1 - \hat{\phi}^{2h}}{1 - \hat{\phi}^2}} \hat{\sigma}_Z.$$

Example 9.10 For the beaver temperature time series values $x_1, x_2, \dots, x_{62}$ from Example 9.3, forecast the next six values in the time series and give approximate 95% prediction intervals for the forecasted values, assuming that the time series arises from a shifted AR(1) model fitted by maximum likelihood estimation.

The R code below calculates the forecasted values and associated approximate 95% prediction interval limits. The forecasts are stored in the vector `pred`, the lower and upper prediction limits are stored in the vectors `lo` and `hi`, respectively.

```
x      = beaver2$temp[beaver2$activ == 1]
model  = ar(x, order.max = 1, aic = FALSE, method = "mle")
phihat = model$ar
muhat  = model$x.mean
sighat = sqrt(model$var.pred)
n      = length(x)
hmax   = 6
```

```

alpha = 0.05
crit = qnorm(1 - alpha / 2)
pred = numeric(hmax)
lo = numeric(hmax)
hi = numeric(hmax)
for (h in 1:hmax) {
  pred[h] = muhat + phihat ^ h * (x[n] - muhat)
  stderr = sqrt((1 - phihat ^ (2 * h)) / (1 - phihat ^ 2)) * sighat
  lo[h] = pred[h] - crit * stderr
  hi[h] = pred[h] + crit * stderr
}
cat(lo, "\n", pred, "\n", hi, "\n")

```

Some keystrokes can be saved by using the R built-in generic `predict` function to compute the forecasts and the associated standard errors.

```

x = beaver2$temp[beaver2$activ == 1]
model = ar(x, order.max = 1, aic = FALSE, method = "mle")
predict(model, n.ahead = 6)

```

The two code segments produce identical results, which are summarized in Table 9.5. Notice that the forecasts trend monotonically toward $\bar{x} = 37.90$ and the standard errors increase as the time horizon h increases. The increasing standard error is consistent with having less precision in the forecast as the time horizon h increases.

Time	$t = 63$	$t = 64$	$t = 65$	$t = 66$	$t = 67$	$t = 68$
Forecast	38.04	38.01	37.99	37.97	37.96	37.95
Standard error	0.130	0.166	0.184	0.195	0.201	0.205
Lower prediction bound	37.78	37.69	37.63	37.59	37.57	37.55
Upper prediction bound	38.29	38.34	38.35	38.36	38.36	38.35

Table 9.5: Forecasts and 95% prediction intervals for the beaver temperatures.

Figure 9.11 shows (a) the original time series $x_1, x_2, \dots, x_{62}$ as points ($\bullet$) connected by lines, (b) the first 12 forecasted temperatures $\hat{X}_{63}, \hat{X}_{64}, \dots, \hat{X}_{74}$ as open circles ($\circ$), and (c) the 95% prediction intervals as a shaded region. There are three key observations concerning this figure.

- Even though the last five observations in the time series $x_{58}, x_{59}, \dots, x_{62}$ show an increasing trend, the forecasts, which are a function only of $x_n = x_{62} = 38.07$, monotonically approach $\hat{\mu} = \bar{x} = 37.90$.
- The widths of the prediction intervals increase as the time horizon h increases. These widths do not increase indefinitely, but rather approach a limit as $h \rightarrow \infty$.
- The random sampling variability which is evident in the observed time series values $x_1, x_2, \dots, x_{62}$ is not apparent in the forecasted values $\hat{X}_{63}, \hat{X}_{64}, \dots, \hat{X}_{74}$. Observed time series values tend to exhibit the typical random sampling variability; forecasted values for a shifted AR(1) model with $0 < \phi < 1$ tend to be smooth.

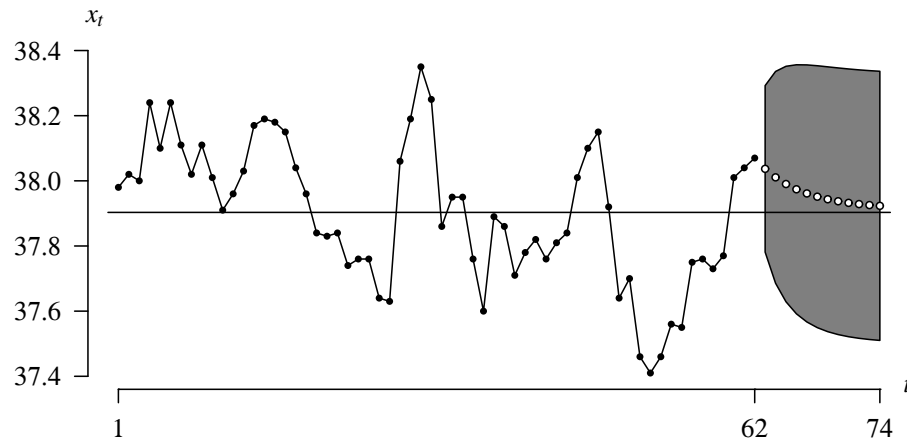


Figure 9.11: Wisconsin beaver forecasted temperatures and 95% prediction intervals.

This subsection has introduced the AR(1) time series model. The key results for an AR(1) model are listed below.

- The standard AR(1) model can be written algebraically and with the backshift operator B as

$$X_t = \phi X_{t-1} + Z_t \quad \text{and} \quad (1 - \phi B)X_t = Z_t,$$

where $Z_t \sim WN(0, \sigma_Z^2)$ and $\sigma_Z^2 > 0$.

- The shifted AR(1) model can be written algebraically and with the backshift operator B as

$$X_t - \mu = \phi(X_{t-1} - \mu) + Z_t \quad \text{and} \quad (1 - \phi B)(X_t - \mu) = Z_t.$$

- The AR(1) model is always invertible; the AR(1) model is stationary for $-1 < \phi < 1$.
- The stationary shifted AR(1) model can be written as an MA(∞) model for $-1 < \phi < 1$ as

$$X_t = \mu + Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2} + \cdots$$

- The AR(1) population autocorrelation function is $\rho(k) = \phi^k$ for $-1 < \phi < 1$ and $k = 1, 2, \dots$
- The AR(1) population partial autocorrelation function at lag one is $\rho^*(1) = \phi$ for $-1 < \phi < 1$ and $\rho^*(k) = 0$ for $k = 2, 3, \dots$
- The three parameters in the shifted AR(1) model, μ , ϕ , and σ_Z^2 , can be estimated from a realization of a time series $x_1, x_2, \dots, x_n$ by the method of moments, least squares, and maximum likelihood. The point estimators for μ , ϕ , and σ_Z^2 are denoted by $\hat{\mu}$, $\hat{\phi}$, and $\hat{\sigma}_Z^2$, and are typically paired with asymptotically exact two-sided $100(1 - \alpha)\%$ confidence intervals.
- The forecast value $\hat{X}_{n+h} = \hat{\mu} + \hat{\phi}^h(x_n - \hat{\mu})$ for an AR(1) model approaches $\hat{\mu} = \bar{x}$ as $h \rightarrow \infty$. The associated prediction intervals have widths that increase as h increases and approach a limit as the time horizon $h \rightarrow \infty$.

If the time series of interest is the daily high temperatures in July in Tuscaloosa, then an AR(1) model would be appropriate if tomorrow's daily high temperature (X_t) can be modeled as a linear function of

- today's high temperature (X_{t-1}), and
- a random shock (Z_t).

But what if weather had more of a memory than just one day? What if tomorrow's daily high temperature (X_t) is better modeled as a linear function of

- today's high temperature (X_{t-1}),
- yesterday's high temperature (X_{t-2}), and
- a random shock (Z_t).

This is an example of the thinking that lies behind the AR(2) model, which is introduced in the next section.

9.1.2 The AR(2) Model

The second-order autoregressive model, denoted by AR(2), can be used for modeling a stationary time series in instances in which the current value of the time series is a linear combination of the two previous values plus a random shock. The mathematics associated with the AR(2) model is somewhat more difficult than that associated with the AR(1) model.

Definition 9.2 A second-order autoregressive time series model, denoted by AR(2), for the time series $\{X_t\}$ is defined by

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t,$$

where ϕ_1 and ϕ_2 are real-valued parameters and $\{Z_t\}$ is a time series of white noise:

$$Z_t \sim WN(0, \sigma_Z^2).$$

There are three parameters that define an AR(2) model: the real-valued coefficients ϕ_1 and ϕ_2 , and the population variance of the white noise σ_Z^2 . The AR(2) model can be written more compactly in terms of the backshift operator B as

$$\phi(B)X_t = Z_t,$$

where $\phi(B)$ is the second-order polynomial

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

The AR(2) model has the form of a multiple linear regression model with two independent variables and no intercept term. The current value X_t is modeled as a linear combination of the two previous values of the time series, X_{t-1} and X_{t-2} , plus a white noise term. The parameters ϕ_1 and ϕ_2 control the inclination of the regression plane in three-dimensional space. The parameter σ_Z^2 reflects the magnitude of the dispersion of the time series values from the regression plane.

To illustrate the thinking behind the AR(2) model in a specific context, let X_t represent the annual return of a particular stock market index in year t . The AR(2) model indicates that the annual return in year t equals ϕ_1 multiplied by the previous year's annual return ($\phi_1 X_{t-1}$), plus ϕ_2 multiplied by the annual return two years prior ($\phi_2 X_{t-2}$), plus the year t random white noise term Z_t .

Stationarity

Theorem 8.3 indicates that all AR(2) models are invertible, but are stationary when the roots of

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

lie outside of the unit circle in the complex plane. Let B_1 and B_2 denote these two roots. Using the quadratic equation, the two roots are

$$B_1 = \frac{\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2} \quad \text{and} \quad B_2 = \frac{\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2}.$$

Since $\phi(B_1) = \phi(B_2) = 0$, the quadratic function $\phi(B)$ can also be written in factored form as

$$\phi(B) = (1 - B_1^{-1}B)(1 - B_2^{-1}B).$$

Equating the two versions of $\phi(B)$ above and matching coefficients results in

$$\phi_1 = B_1^{-1} + B_2^{-1} \quad \text{and} \quad \phi_2 = -(B_1 B_2)^{-1}.$$

These two equations define the mapping from the complex plane, which contains the roots B_1 and B_2 , to the plane that contains the AR(2) parameters ϕ_1 and ϕ_2 . To find the stationary region, we must find the mapping of the part of the complex plane outside of the unit circle to the (ϕ_1, ϕ_2) plane. The mapping yields a triangular-shaped stationary region, as specified in the following result.

Theorem 9.9 The AR(2) time series model is stationary when ϕ_1 and ϕ_2 satisfy

$$\phi_1 + \phi_2 < 1, \quad \phi_2 - \phi_1 < 1, \quad \text{and} \quad \phi_2 > -1.$$

Proof The three cases considered below are based on whether the discriminant in the roots of $\phi(B) = 0$ is zero (two identical real roots), positive (two distinct real roots), or negative (two complex roots).

Case 1: Two identical real roots ($\phi_1^2 + 4\phi_2 = 0$). This is a concave-down parabola through the origin in the (ϕ_1, ϕ_2) plane. The single real root is

$$B_1 = B_2 = -\frac{\phi_1}{2\phi_2} = \frac{\phi_1}{\phi_1^2/2} = \frac{2}{\phi_1}.$$

Since this single real root must lie outside of the unit circle for a stationary model,

$$\left| \frac{2}{\phi_1} \right| > 1 \quad \Rightarrow \quad -2 < \phi_1 < 2.$$

This portion of the parabola is the leftmost graph in Figure 9.12.

Case 2: Two distinct real roots ($\phi_1^2 + 4\phi_2 > 0$). This is the region *above* the parabola $\phi_1^2 + 4\phi_2 = 0$ in the (ϕ_1, ϕ_2) plane. Since B_1 and B_2 are real-valued, the conditions for stationarity $|B_1| > 1$ and $|B_2| > 1$ that correspond to having both B_1 and B_2 falling outside of the unit circle are equivalent to

$$-1 < \frac{1}{B_1} < 1 \quad \text{and} \quad -1 < \frac{1}{B_2} < 1.$$

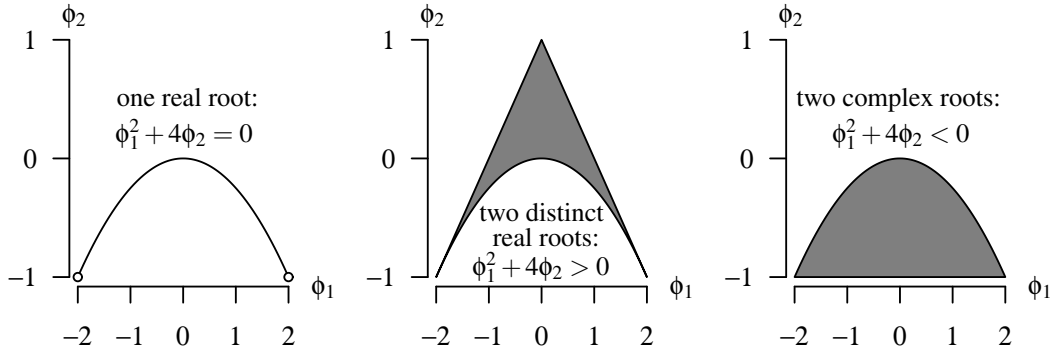


Figure 9.12: Partitioning the stationary region for the AR(2) time series model.

The reciprocals of the roots of $\phi(B) = 0$ are

$$\frac{1}{B_1} = \left[\frac{-2\phi_2}{\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}} \right] \cdot \left[\frac{\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}}{\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}} \right] = \frac{\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}}{2}$$

and

$$\frac{1}{B_2} = \left[\frac{-2\phi_2}{\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}} \right] \cdot \left[\frac{\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}}{\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}} \right] = \frac{\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}}{2}.$$

Since B_1 and B_2 are real and distinct, stationarity is achieved when

$$-1 < \frac{1}{B_2} < \frac{1}{B_1} < 1$$

or

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

Writing the leftmost inequality as $\sqrt{\phi_1^2 + 4\phi_2} < \phi_1 + 2$, squaring both sides of this inequality, and simplifying gives

$$\phi_2 - \phi_1 < 1.$$

Applying a similar approach to the rightmost inequality gives

$$\phi_1 + \phi_2 < 1.$$

So for the AR(2) model in the case of $\phi(B)$ having two distinct real roots, the model is stationary when the three inequalities

$$\phi_1^2 + 4\phi_2 > 0, \quad \phi_2 - \phi_1 < 1, \quad \text{and} \quad \phi_1 + \phi_2 < 1$$

are satisfied. This region is shaded in the middle graph in Figure 9.12.

Case 3: Two complex conjugate roots ($\phi_1^2 + 4\phi_2 < 0$). This is the region *below* the parabola $\phi_1^2 + 4\phi_2 = 0$ in the (ϕ_1, ϕ_2) plane. For the model to be stationary, B_1 and B_2 must lie outside of the unit circle in the complex plane. Since the roots are complex conjugates, $|B_1| = |B_2|$, which is calculated as

$$|B_1| = \sqrt{\left(\frac{\phi_1}{-2\phi_2}\right)^2 + \left(\frac{\sqrt{-\phi_1^2 - 4\phi_2}}{-2\phi_2}\right)^2} = \sqrt{\frac{-4\phi_2}{4\phi_2^2}} = \sqrt{-\frac{1}{\phi_2}},$$

which is greater than 1 for $-1 < \phi_2 < 0$. (The imaginary part of the discriminant was negated to avoid taking the square root of a negative number because the discriminant is negative in Case 3.) The region associated with the inequalities

$$\phi_1^2 + 4\phi_2 < 0 \quad \text{and} \quad \phi_2 > -1$$

is the shaded region in the rightmost graph in Figure 9.12.

The union of these three regions is the interior of the triangular region described by the three inequalities in Theorem 9.9, which proves the result. $\square$

Population Autocorrelation Function

Now that the stationary region for an AR(2) time series model has been established, we turn to the derivation of the population autocorrelation function. Assuming that the parameters ϕ_1 and ϕ_2 fall in the stationary region, the AR(2) model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$$

can be multiplied by X_{t-k} to give

$$X_t X_{t-k} = \phi_1 X_{t-1} X_{t-k} + \phi_2 X_{t-2} X_{t-k} + Z_t X_{t-k}.$$

Taking the expected value of both sides of this equation results in the recursive equation

$$\gamma(k) = \phi_1 \gamma(k-1) + \phi_2 \gamma(k-2)$$

for $k = 1, 2, \dots$ because Z_t has expected value zero and is independent of X_{t-k} . Dividing both sides of this equation by $\gamma(0) = V[X_t]$ gives the recursive equation

$$\rho(k) = \phi_1 \rho(k-1) + \phi_2 \rho(k-2)$$

for $k = 1, 2, \dots$. These linear equations, whether written in terms of $\gamma(k)$ or $\rho(k)$, are known in time series analysis as the *Yule–Walker equations* after British statisticians George Udny Yule and Sir Gilbert Walker. Once the first two values of $\gamma(k)$ or $\rho(k)$ are known, these recursive equations can be used to calculate subsequent values. The next two paragraphs focus on determining the first two values of $\gamma(k)$ and $\rho(k)$, respectively.

For a stationary AR(2) time series model, we derive expressions for $\gamma(0)$ and $\gamma(1)$. The AR(2) model is

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t.$$

Squaring both sides of this equation and taking the expected value of both sides gives

$$\gamma(0) = \phi_1^2 \gamma(0) + \phi_2^2 \gamma(0) + \sigma_Z^2 + 2\phi_1 \phi_2 \gamma(1).$$

Using the symmetry of the population autocovariance function, the Yule–Walker equation with $k = 1$ is

$$\gamma(1) = \phi_1\gamma(0) + \phi_2\gamma(1) \quad \Rightarrow \quad \gamma(1) = \frac{\phi_1\gamma(0)}{1 - \phi_2}.$$

Replacing this expression for $\gamma(1)$ in the previous equation gives

$$\gamma(0) = \phi_1^2\gamma(0) + \phi_2^2\gamma(0) + \sigma_Z^2 + 2\phi_1\phi_2\frac{\phi_1\gamma(0)}{1 - \phi_2}.$$

Moving all terms involving $\gamma(0)$ to the left-hand side of this equation gives

$$\gamma(0) \left[1 - \phi_1^2 - \phi_2^2 - 2\phi_1\phi_2\frac{1}{1 - \phi_2} \right] = \sigma_Z^2.$$

Solving this equation for $\gamma(0)$,

$$\begin{aligned} \gamma(0) &= \frac{(1 - \phi_2)\sigma_Z^2}{(1 - \phi_2) - (1 - \phi_2)\phi_1^2 - (1 - \phi_2)\phi_2^2 - 2\phi_1^2\phi_2} \\ &= \frac{(1 - \phi_2)\sigma_Z^2}{1 - \phi_2 - \phi_1^2 + \phi_1^2\phi_2 - \phi_2^2 + \phi_2^3 - 2\phi_1^2\phi_2} \\ &= \frac{(1 - \phi_2)\sigma_Z^2}{(1 + \phi_2)(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)}. \end{aligned}$$

An expression for $\gamma(1)$ is

$$\gamma(1) = \frac{\phi_1\gamma(0)}{1 - \phi_2} = \frac{\phi_1\sigma_Z^2}{(1 + \phi_2)(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)}.$$

These two values can be used as arguments in the Yule–Walker equations to obtain subsequent values for $\gamma(k)$.

We now turn to the problem of finding $\rho(1)$ and $\rho(2)$. The first two Yule–Walker equations in terms of $\rho(k)$ are

$$\begin{aligned} \rho(1) &= \phi_1\rho(0) + \phi_2\rho(-1) \\ \rho(2) &= \phi_1\rho(1) + \phi_2\rho(0). \end{aligned}$$

Since $\rho(0) = 1$ and $\rho(-k) = \rho(k)$ via Theorem 7.1, these equations reduce to

$$\begin{aligned} \rho(1) &= \phi_1 + \phi_2\rho(1) \\ \rho(2) &= \phi_1\rho(1) + \phi_2, \end{aligned}$$

which are easily solved for $\rho(1)$ and $\rho(2)$:

$$\rho(1) = \frac{\phi_1}{1 - \phi_2} \quad \text{and} \quad \rho(2) = \frac{\phi_1^2}{1 - \phi_2} + \phi_2.$$

A general formula for $\rho(k)$ exists, but it can involve complex numbers and is unwieldy. An exercise concerning its calculation is given at the end of the chapter. These results are summarized in the following theorem.

Theorem 9.10 The population autocovariance function for a stationary AR(2) time series model is calculated by

$$\gamma(k) = \phi_1 \gamma(k-1) + \phi_2 \gamma(k-2)$$

for $k = 1, 2, \dots$, where

$$\gamma(0) = \frac{(1 - \phi_2)\sigma_Z^2}{(1 + \phi_2)(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)} \quad \text{and} \quad \gamma(1) = \frac{\phi_1 \sigma_Z^2}{(1 + \phi_2)(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)}.$$

The population autocorrelation function for a stationary AR(2) time series model is calculated by

$$\rho(k) = \phi_1 \rho(k-1) + \phi_2 \rho(k-2)$$

for $k = 1, 2, \dots$, where

$$\rho(1) = \frac{\phi_1}{1 - \phi_2} \quad \text{and} \quad \rho(2) = \frac{\phi_1^2}{1 - \phi_2} + \phi_2.$$

We now focus in on the values of $\rho(1)$ and $\rho(2)$. We can solve for ϕ_1 and ϕ_2 in terms of $\rho(1)$ and $\rho(2)$ as

$$\phi_1 = \frac{\rho(1)(1 - \rho(2))}{1 - \rho(1)^2} \quad \text{and} \quad \phi_2 = \frac{\rho(2) - \rho(1)^2}{1 - \rho(1)^2}.$$

These equations can be helpful in the three settings described below.

1. These equations bear some practical use in that the first two *sample* autocorrelation function values, r_1 and r_2 , can be calculated from a time series and used as approximations for $\rho(1)$ and $\rho(2)$, yielding estimates for ϕ_1 and ϕ_2 . These can in turn be used as initial estimates for finding point estimates for ϕ_1 and ϕ_2 by, for example, least squares or maximum likelihood estimation, should numerical methods be required.
2. Level surfaces (that is, contours) in the triangular-shaped stationary region from Theorem 9.9 can be determined by fixing values of $\rho(1)$ and $\rho(2)$. As an illustration, consider $\rho(1) = 0$. In this case, $\phi_1 = 0$ and $\phi_2 = \rho(2)$, which is a line segment in the stationary region. Continuing in this fashion for several fixed values of $\rho(1)$ [with varying values of $\rho(2)$] and then for several fixed values of $\rho(2)$ [with varying values of $\rho(1)$] results in the graph of the stationary region with the level surfaces included shown in Figure 9.13. The level surfaces associated with fixed values of $\rho(1)$ are lines; the level surfaces associated with fixed values of $\rho(2)$ are curves.
3. Since $\rho(1)$ and $\rho(2)$ are population correlations, the obvious constraints on their values for an AR(2) time series model are $-1 < \rho(1) < 1$ and $-1 < \rho(2) < 1$. Additionally, since $\phi_2 > -1$ in order to fall into the triangular-shaped stationary region defined in Theorem 9.9 for the AR(2) time series model,

$$\phi_2 > -1 \quad \Rightarrow \quad \frac{\rho(2) - \rho(1)^2}{1 - \rho(1)^2} > -1 \quad \Rightarrow \quad \rho(2) > 2\rho(1)^2 - 1.$$

The boundary of this third constraint is a parabola in the $(\rho(1), \rho(2))$ plane. The shaded region in Figure 9.14 shows the $\rho(1)$ and $\rho(2)$ values that are associated with stationary AR(2) time series models. Unlike the AR(1) population autocorrelation function, it is possible to

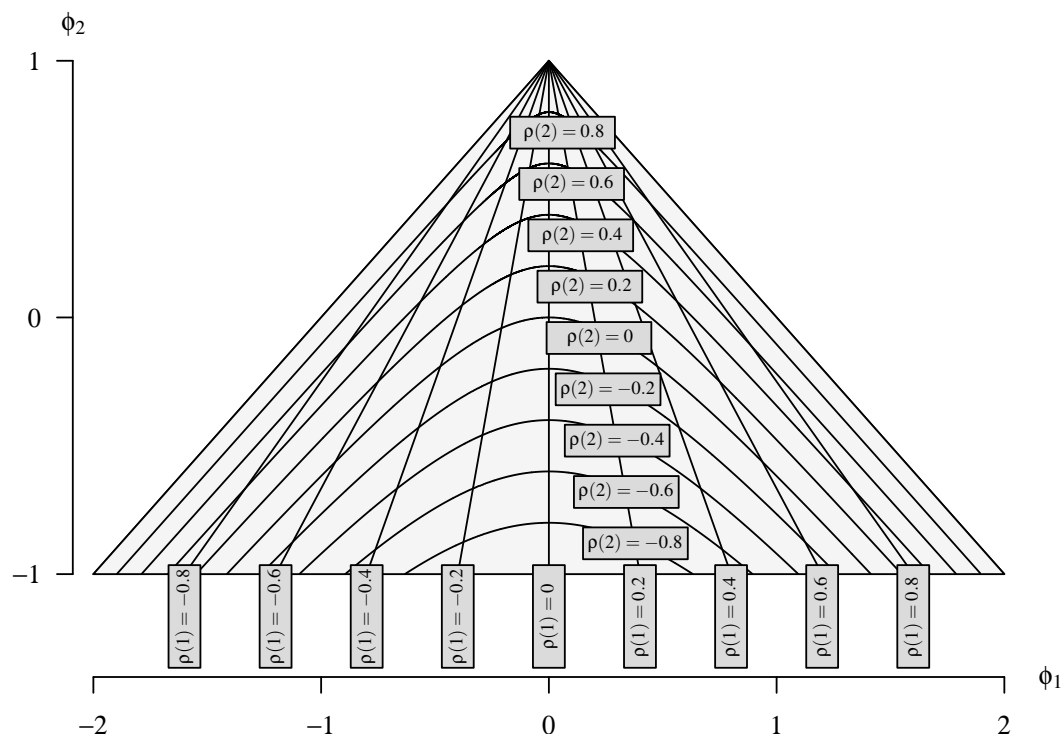


Figure 9.13: Stationary region for an AR(2) time series model with level surfaces.

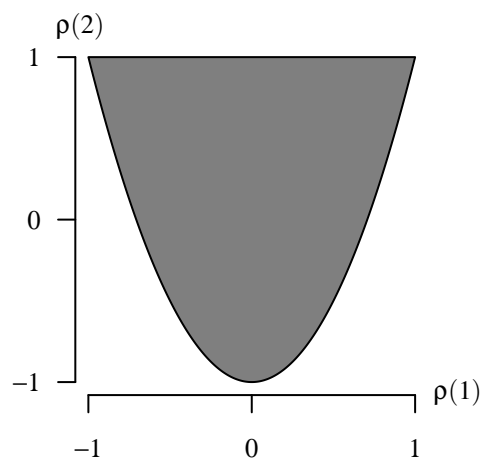


Figure 9.14: Values of $\rho(1)$ and $\rho(2)$ associated with a stationary AR(2) model.

achieve a stationary model with $|\rho(2)| > |\rho(1)|$. The AR(2) population autocorrelation function values are not necessarily monotonically decreasing in magnitude as they were in the AR(1) model.

Population Partial Autocorrelation Function

We now determine the population partial autocorrelation function for an AR(2) model. Using Definition 7.4, the population lag 0 partial autocorrelation is $\rho^*(0) = 1$. The population lag 1 partial autocorrelation is $\rho^*(1) = \rho(1) = \phi_1/(1 - \phi_2)$. After evaluating the determinants and simplifying, the population lag 2 partial autocorrelation is

$$\rho^*(2) = \frac{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & \rho(2) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & 1 \end{vmatrix}} = \frac{\begin{vmatrix} 1 & \phi_1/(1 - \phi_2) \\ \phi_1/(1 - \phi_2) & (\phi_1^2 - \phi_2^2 + \phi_2)/(1 - \phi_2) \end{vmatrix}}{\begin{vmatrix} 1 & \phi_1/(1 - \phi_2) \\ \phi_1/(1 - \phi_2) & 1 \end{vmatrix}} = \phi_2.$$

Appealing to the Yule–Walker equations from Theorem 9.10 to define the third column of the determinant of the numerator, the population lag 3 partial autocorrelation is

$$\rho^*(3) = \frac{\begin{vmatrix} 1 & \rho(1) & \rho(1) \\ \rho(1) & 1 & \rho(2) \\ \rho(2) & \rho(1) & \rho(3) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) & \rho(2) \\ \rho(1) & 1 & \rho(1) \\ \rho(2) & \rho(1) & 1 \end{vmatrix}} = \frac{\begin{vmatrix} 1 & \rho(1) & \phi_1 + \phi_2\rho(1) \\ \rho(1) & 1 & \phi_1\rho(1) + \phi_2 \\ \rho(2) & \rho(1) & \phi_1\rho(2) + \phi_2\rho(1) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) & \rho(2) \\ \rho(1) & 1 & \rho(1) \\ \rho(2) & \rho(1) & 1 \end{vmatrix}} = 0.$$

The determinant in the numerator is zero because the third column is a linear combination of the first two columns. This pattern continues for the higher lags. When computing $\rho^*(k)$ for $k = 3, 4, \dots$, the first, second, and last columns of the matrix in the numerator are

$$\begin{bmatrix} 1 \\ \rho(1) \\ \rho(2) \\ \vdots \\ \rho(k-1) \end{bmatrix}, \begin{bmatrix} \rho(1) \\ 1 \\ \rho(1) \\ \vdots \\ \rho(k-2) \end{bmatrix}, \text{ and } \begin{bmatrix} \phi_1 + \phi_2\rho(1) \\ \phi_1\rho(1) + \phi_2 \\ \phi_1\rho(2) + \phi_2\rho(1) \\ \vdots \\ \phi_1\rho(k-1) + \phi_2\rho(k-2) \end{bmatrix}.$$

The last column of the matrix in the numerator is a linear combination of the first two columns. The matrix in the numerator of the calculation of $\rho^*(k)$ is singular, which means that its determinant is zero. This constitutes a proof of the following result.

Theorem 9.11 The population partial autocorrelation function for a stationary AR(2) time series model is

$$\rho^*(k) = \begin{cases} 1 & k = 0 \\ \phi_1/(1 - \phi_2) & k = 1 \\ \phi_2 & k = 2 \\ 0 & k = 3, 4, \dots \end{cases}$$

The population partial autocorrelation function for the AR(2) model cuts off after lag 2. A graph of the sample partial autocorrelation function (that is, a graph of r_k^* for the first few values

of k), should also cut off after lag 2 if the AR(2) model is appropriate. This sample partial autocorrelation function shape is easier to recognize than the associated sample autocorrelation function shape because cutting off is easier to recognize than tailing off in the presence of random sampling variability.

A careful inspection of Theorem 9.11 reveals that the signs of ϕ_1 and ϕ_2 match the signs of $\rho^*(1)$ and $\rho^*(2)$, respectively:

$$\text{sgn}(\phi_1) = \text{sgn}(\rho^*(1)) \quad \text{and} \quad \text{sgn}(\phi_2) = \text{sgn}(\rho^*(2))$$

for ϕ_1 and ϕ_2 falling in the triangular-shaped stationary region. Figure 9.15 shows the stationary region from Theorem 9.9, along with plots of the representative population autocorrelation function and population partial autocorrelation function. There are four points, one in each quadrant, that are plotted. The population autocorrelation function and the population partial autocorrelation function associated with those four points are plotted in each of the quadrants. As expected, the signs of the values of ϕ_1 and $\rho^*(1)$ match and the signs of the values of ϕ_2 and $\rho^*(2)$ match. The *quadrant* in the stationary region determines the signs of $\rho^*(1)$ and $\rho^*(2)$, as illustrated by the four representative population partial autocorrelation functions graphed in Figure 9.15. As you can see by inspecting the shapes of $\rho(k)$ and $\rho^*(k)$ from Figure 9.15, the addition of the parameter ϕ_2 in the transition from the AR(1) model to the AR(2) model results in significant additional modeling capability. The following observations can be gleaned from Figure 9.15.

- As expected, all population partial autocorrelation functions cut off after lag two.
- When $\phi(B)$ has real roots, the population autocorrelation function consists of mixtures of damped exponentials.

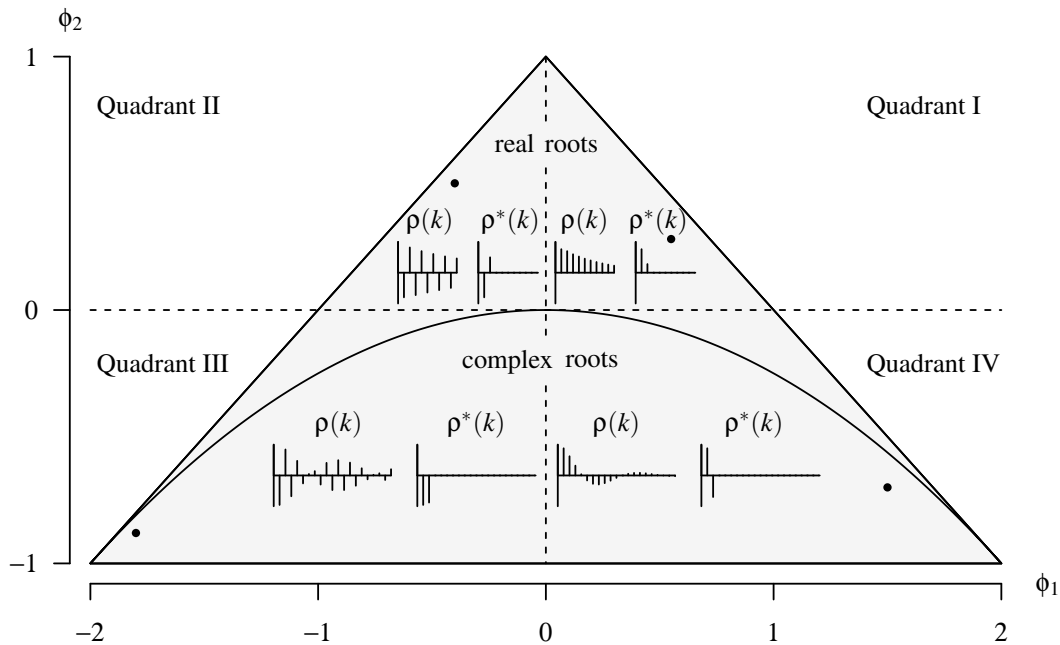


Figure 9.15: Stationary AR(2) time series model signature $\rho(k)$ and $\rho^*(k)$ shapes.

- When $\phi(B)$ has complex roots, the population autocorrelation function has a damped sinusoidal shape.

The population autocorrelations on the tiny inset plots of $\rho(k)$ and $\rho^*(k)$ in Figure 9.15 can be calculated using the recursive relationships from Theorem 9.10 [for $\rho(k)$] and Theorem 9.11 [for $\rho^*(k)$]. They can also be calculated using the R `ARMAacf` function. Consider the two inset plots in the fourth quadrant of the graph in Figure 9.15, for example, that correspond to $\phi_1 = 1.5$ and $\phi_2 = -0.7$. The graph of the first 20 lags of $\rho(k)$ can be plotted with the R command

```
plot(ARMAacf(ar = c(1.5, -0.7), ma = 0, lag.max = 20), type = "h")
```

Likewise, the graph of the first 20 lags of $\rho^*(k)$ can be plotted with

```
plot(ARMAacf(ar = c(1.5, -0.7), ma = 0, lag.max = 20, pacf = TRUE), type = "h")
```

The `ar` argument defines the ϕ_1 and ϕ_2 parameters of the AR(2) model, the `ma` argument is set to zero to indicate that there are no moving average terms in the AR(2) model, the `lag.max` argument is set to 20 to return the first 20 autocorrelations, and the `type` argument in the call to `plot` is set to "h" in order to graph the autocorrelations as spikes rather than points.

As was the case with the AR(1) time series model, the AR(2) time series can be written as an MA(∞) time series model. This alternative representation can be useful for deriving certain quantities associated with the AR(2) model, in particular, standard errors of forecasted values. The first form of a general linear model, which is equivalent to an MA(∞) model, is

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots$$

Our goal is to determine the values of $\theta_1, \theta_2, \dots$ that correspond to fixed parameters ϕ_1 and ϕ_2 . Since the MA(∞) model is valid at time t , it is also valid at times $t-1$ and $t-2$:

$$X_{t-1} = Z_{t-1} + \theta_1 Z_{t-2} + \theta_2 Z_{t-3} + \cdots$$

and

$$X_{t-2} = Z_{t-2} + \theta_1 Z_{t-3} + \theta_2 Z_{t-4} + \cdots$$

So the AR(2) time series model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$$

as established in Definition 9.2, can be rewritten as

$$Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots = \phi_1 (Z_{t-1} + \theta_1 Z_{t-2} + \theta_2 Z_{t-3} + \cdots) + \phi_2 (Z_{t-2} + \theta_1 Z_{t-3} + \theta_2 Z_{t-4} + \cdots) + Z_t.$$

Equating the coefficients of Z_{t-1} gives

$$\theta_1 = \phi_1.$$

Equating the coefficients of Z_{t-2} gives

$$\theta_2 = \phi_1 \theta_1 + \phi_2 = \phi_1^2 + \phi_2.$$

Equating the coefficients of Z_{t-k} gives the recursive equation

$$\theta_k = \phi_1 \theta_{k-1} + \phi_2 \theta_{k-2}$$

for $k = 3, 4, \dots$

Theorem 9.12 A stationary AR(2) model with parameters ϕ_1 and ϕ_2 can be written as an MA(∞) model

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots,$$

where $\theta_1 = \phi_1$, $\theta_2 = \phi_1^2 + \phi_2$, and

$$\theta_k = \phi_1 \theta_{k-1} + \phi_2 \theta_{k-2}$$

for $k = 3, 4, \dots$.

An exercise at the end of the chapter highlights other methods for calculating the coefficients $\theta_1, \theta_2, \dots$ in the MA(∞) model which is equivalent to the stationary AR(2) model.

Example 9.11 Calculate the first six coefficients of an MA(∞) model associated with the stationary AR(2) model with $\phi_1 = 1$ and $\phi_2 = -1/2$.

The AR(2) model is stationary because the point $(\phi_1, \phi_2) = (1, -1/2)$ falls in the triangular-shaped stationary region defined by the inequalities in Theorem 9.9. Using Theorem 9.12, the first six coefficients of the MA(∞) model are

$$\theta_1 = 1, \quad \theta_2 = \frac{1}{2}, \quad \theta_3 = 0, \quad \theta_4 = -\frac{1}{4}, \quad \theta_5 = -\frac{1}{4}, \quad \theta_6 = -\frac{1}{8}.$$

These coefficients can also be calculated in R with the `ARMAtoMA` function. The statement

```
ARMAtoMA(ar = c(1, -1 / 2), ma = 0, lag.max = 6)
```

returns the same coefficients calculated above. The `ar` argument defines the ϕ_1 and ϕ_2 parameters of the AR(2) model, the `ma` argument is set to zero to indicate that there are no moving average terms in the AR(2) model, and the `lag.max` argument is set to 6 in order to calculate the first six coefficients in the MA(∞) model and return these values in a vector in R.

The Shifted AR(2) Model

For a stationary AR(2) model expressed as an MA(∞) model, it is clear that $E[X_t] = 0$. This model is not of much use in practice because most real-world time series are not centered around zero. Adding a shift parameter μ overcomes this shortcoming. Since population variance and covariance are unaffected by a shift, the associated population autocorrelation and partial autocorrelation functions remain the same as those given in Theorems 9.10 and 9.11.

Theorem 9.13 A shifted second-order autoregressive model for the time series $\{X_t\}$ is defined by

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + Z_t,$$

where ϕ_1, ϕ_2, μ , and $\sigma_Z^2 > 0$ are real-valued parameters and $\{Z_t\}$ is a time series of white noise. This model is stationary when ϕ_1 and ϕ_2 satisfy the three inequalities given in Theorem 9.9. The expected value of X_t is $E[X_t] = \mu$. The population autocorrelation function can be calculated using the recursive equations in Theorem 9.10. The population partial autocorrelation function is given in Theorem 9.11.

The shifted AR(2) model can be written in terms of the backshift operator B as

$$\phi(B)(X_t - \mu) = Z_t,$$

where $\phi(B) = 1 - \phi_1 B - \phi_2 B^2$. The practical problem of fitting a shifted AR(2) model to an observed time series of n values $x_1, x_2, \dots, x_n$ will be illustrated later in this subsection.

Simulation

An AR(2) time series can be simulated by appealing to the defining formula for the AR(2) model. Iteratively applying the defining formula for a standard AR(2) model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$$

from Definition 9.2 results in the simulated values $X_1, X_2, \dots, X_n$. The primary difficult aspect of devising a simulation algorithm is generating the first two values, X_1 and X_2 . For simplicity, assume that the white noise terms are Gaussian white noise terms. There are two approaches to overcome this initialization problem. The first approach generates X_1 and X_2 from a bivariate normal distribution with population mean vector $\mathbf{0} = (0, 0)'$ and variance–covariance matrix

$$\Sigma = \begin{bmatrix} \gamma(0) & \gamma(1) \\ \gamma(1) & \gamma(0) \end{bmatrix} = \frac{\sigma_Z^2}{(1 + \phi_2)(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)} \begin{bmatrix} 1 - \phi_2 & \phi_1 \\ \phi_1 & 1 - \phi_2 \end{bmatrix}$$

via Theorem 9.10. Notice that in the special case of $\phi_1 = \phi_2 = 0$ this matrix reduces to the variance–covariance matrix for Gaussian white noise, which is $I\sigma_Z^2$. The algorithm given as pseudocode below generates initial time series observations X_1 and X_2 as indicated above, and then uses an additional $n - 2$ Gaussian white noise terms $Z_3, Z_4, \dots, Z_n$ to generate the remaining time series values $X_3, X_4, \dots, X_n$ using the AR(2) defining formula from Definition 9.2. Indentation denotes nesting in the algorithm.

```

generate  $(X_1, X_2) \sim BVN(\mathbf{0}, \Sigma)$ 
 $t \leftarrow 2$ 
while  $(t < n)$ 
   $t \leftarrow t + 1$ 
  generate  $Z_t \sim N(0, \sigma_Z^2)$ 
   $X_t \leftarrow \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$ 

```

The four-parameter shifted AR(2) time series model which includes a population mean parameter μ can be simulated by simply adding μ to each time series observation generated by this algorithm. The next example implements this algorithm in R.

Example 9.12 Generate a realization of $n = 100$ observations from an AR(2) time series model with $\phi_1 = 1.5$, $\phi_2 = -0.7$, and Gaussian white noise error terms with $\sigma_Z^2 = 16$.

Since $(\phi_1, \phi_2) = (1.5, -0.7)$ lies in the triangular-shaped stationary region defined in Theorem 9.9, the simulated values will be generated from a stationary time series model. The population autocorrelation function $\rho(k)$ and the population partial autocorrelation function $\rho^*(k)$ are displayed in the fourth quadrant of Figure 9.15, and we expect similar shaped functions r_k and r_k^* from our simulated values. The optional first statement in the R code below uses the `set.seed` function to establish the random number seed. The second and third statements set the AR(2) coefficients to $\phi_1 = 1.5$ and

$\phi_2 = -0.7$. The fourth statement sets the standard deviation of the Gaussian white noise to $\sigma_Z = 4$. The fifth statement places the variance–covariance matrix of X_1 and X_2 in the 2×2 matrix `sigma`. The sixth statement sets the number of simulated values to $n = 100$. The seventh statement defines the vector `x` of length $n = 100$ to hold the simulated time series values. The eighth statement uses the `mvrnorm` function from the MASS package to generate the first two simulated time series observations X_1 and X_2 from the appropriate bivariate normal distribution. Finally, the `for` loop iterates through the defining formula for the AR(2) model generating the remaining observations $X_3, X_4, \dots, X_{100}$.

```
set.seed(3)
phi1 = 1.5
phi2 = -0.7
sigz = 4
sigma = matrix(c(1 - phi2, phi1, phi1, 1 - phi2), 2, 2) * sigz /
  ((1 + phi2) * (1 + phi1 - phi2) * (1 - phi1 - phi2))
n = 100
x = numeric(n)
x[1:2] = MASS::mvrnorm(1, mu = c(0, 0), Sigma = sigma)
for (t in 3:n) x[t] = phi1 * x[t - 1] + phi2 * x[t - 2] + rnorm(1, 0, sigz)
```

Using the `plot.ts` function to make a plot of the time series contained in `x`, the `acf` function to plot the associated correlogram, the `pacf` function to plot the associated sample partial autocorrelation function, and the `layout` function to arrange the graphs as in Example 7.24, the resulting trio of graphs are displayed in Figure 9.16. The sample partial autocorrelation function has statistically significant spikes at lags 1 and 2 with $r_1^* = 0.8036$ and $r_2^* = -0.6229$, and then cuts off after lag 2 as expected from the population counterparts in Figure 9.15. The approximate 95% confidence intervals indicated by the dashed lines show that the values of the sample partial autocorrelation function do not significantly differ from zero at lags beyond lag 2. The sample autocorrelation function displays a damped sinusoidal shape as expected. The time series plot shows that observations tend to linger on one side of the population mean (indicated by a horizontal line), which is consistent with the two initial statistically significant positive spikes in the sample autocorrelation function. However, the time that the observations linger on one side of the mean is inhibited by the statistically significant negative spikes at lags 4, 5, 6, and 7 in the sample autocorrelation function. There is thus some tug exerted by the time series model to linger on one side of the mean for only a limited amount of time.

We recommend running the simulation code from the previous example several dozen times in a loop and viewing the associated plots of x_t , r_k , and r_k^* in search of patterns. This will allow you to see how various realizations of a simulated AR(2) time series model vary from one realization to the next. So when you then view a *single* realization of a real-life time series, you will have a sense of how far these plots might deviate from their expected patterns.

There is a second way to overcome the initialization problem in simulating observations from an AR(2) time series. This second technique starts the time series with two initial arbitrary values, and then allows the time series to “warm up” or “burn in” for several time periods before producing the first observation X_1 . Reasonable initial arbitrary values for the standard AR(2) model are 0; reasonable initial arbitrary values for the shifted AR(2) model are μ . This is the approach taken by the built-in R function named `arima.sim`, which simulates a realization of a time series. Using the

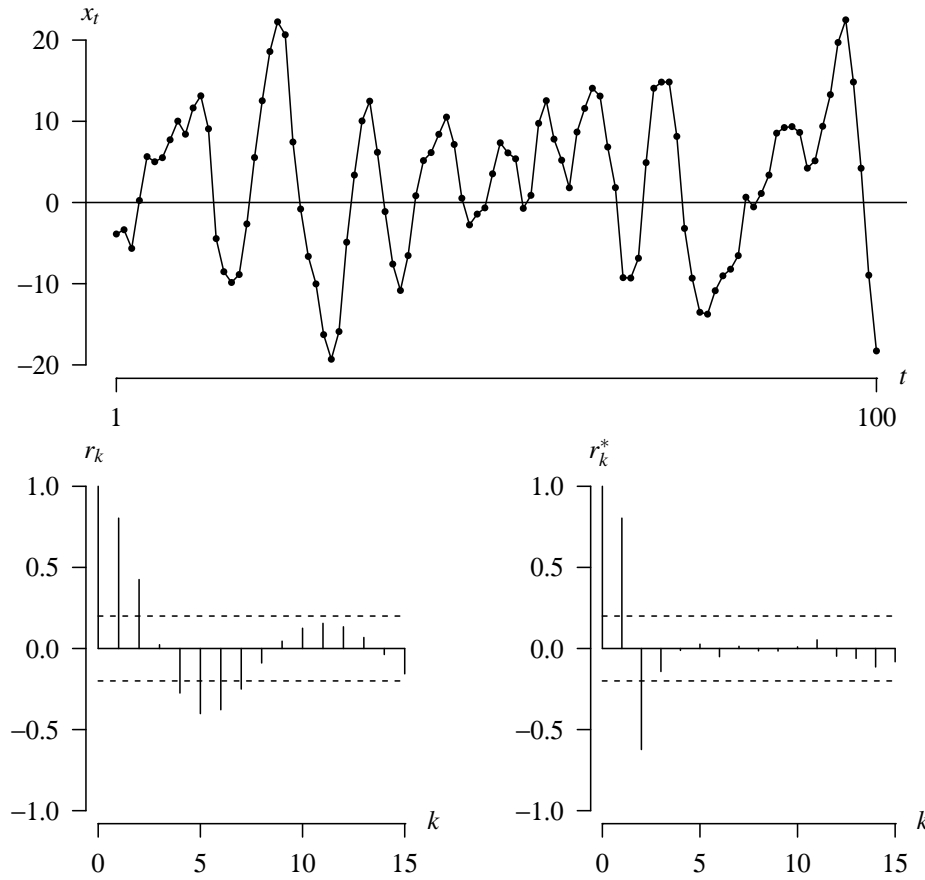


Figure 9.16: Time series plot, r_k , and r_k^* for $n = 100$ simulated values from an AR(2) model.

`arima.sim` function saves a few keystrokes over the approach taken in the previous example, as illustrated next.

Example 9.13 Generate a realization of $n = 100$ observations from a shifted AR(2) time series model with coefficients $\phi_1 = -1.8$ and $\phi_2 = -0.88$, population mean value $\mu = 10$, and Gaussian white noise error terms with $\sigma_Z^2 = 16$.

Since there is now a nonzero population mean value, the shifted AR(2) model is

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + Z_t,$$

where $\mu = 10$, $\phi_1 = -1.8$, $\phi_2 = -0.88$, and $\sigma_Z = 4$. Since $(\phi_1, \phi_2) = (-1.8, -0.88)$ lies in the triangular-shaped stationary region defined in Theorem 9.9, this is a stationary AR(2) time series model. The $\rho(k)$ and $\rho^*(k)$ values for this model are plotted in the third quadrant of Figure 9.15. The `model` argument in the `arima.sim` function is a list containing the values of the coefficients ϕ_1 and ϕ_2 . The second argument to `arima.sim` is `n`, the number of time series observations to be generated. The `sd` argument defines

the standard deviation of the white noise σ_Z . The `n.start` argument gives the number of observations in the burn-in period, which we specify here as 50. The R code to generate $n = 100$ values from the shifted AR(2) model is given below.

```
set.seed(9)
x = 10 + arima.sim(model = list(ar = c(-1.8, -0.88)),
  n = 100, sd = 4, n.start = 50)
```

Figure 9.17 shows the three plots associated with the simulated values in the vector `x` using the `plot.ts`, `acf`, and `pacf` functions. The time series plot shows a radically different pattern than the time series in the previous example in several aspects. First, this simulated time series is centered around $\mu = 10$ (indicated by a horizontal line) rather than $\mu = 0$. Second, the time series jumps from one side of the population mean to the other from one observation to the next. This is consistent with the highly statistically significant negative lag 1 sample autocorrelation $r_1 = -0.9546$. The signs of the initial sample autocorrelation function values alternate, their magnitudes decrease, and sample

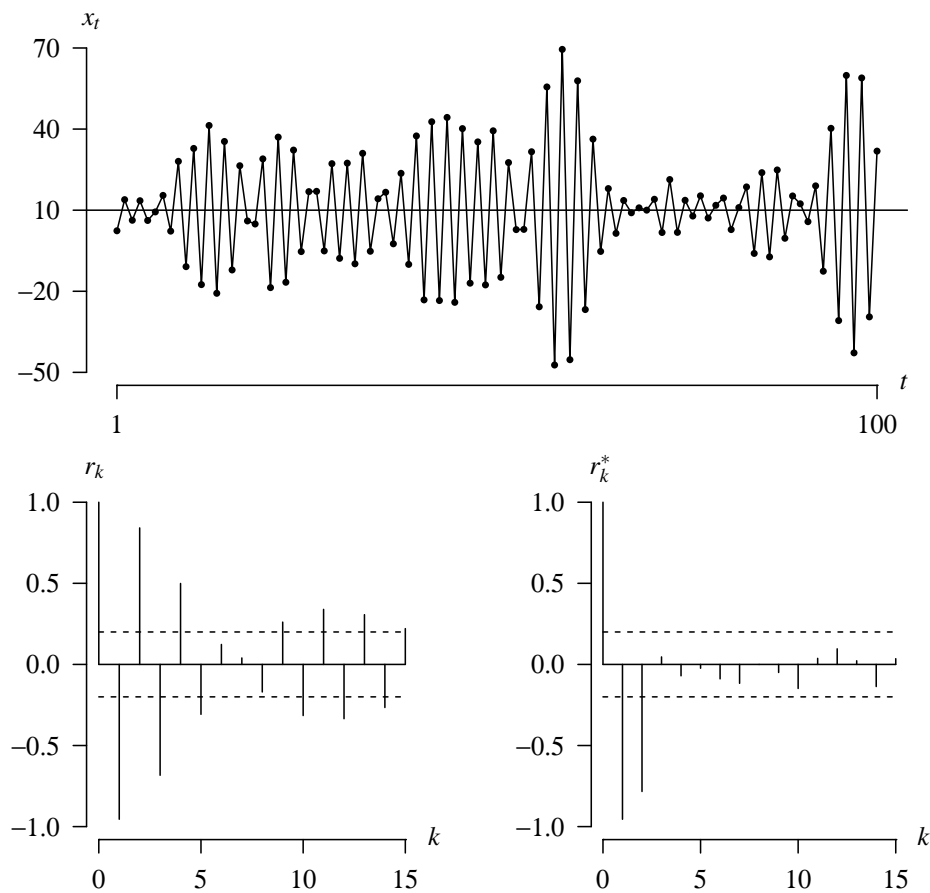


Figure 9.17: Time series plot, r_k , and r_k^* for $n = 100$ simulated values from an AR(2) model.

autocorrelations at subsequent lags follow a damped sinusoidal pattern. In addition, the lag 1 sample autocorrelation is so close to -1 that adjacent observations often tend to be about the same distance away from μ . Third, the two statistically significant spikes in the partial autocorrelation function, $r_1^* = -0.9546$ and $r_2^* = -0.7834$, have the same signs as ϕ_1 and ϕ_2 . As expected, the partial autocorrelation function cuts off after lag 2.

The remaining topics associated with the AR(2) time series model are statistical in nature: parameter estimation, model assessment, model selection, and forecasting. A sample time series that will be revisited throughout these topics is introduced next.

Example 9.14 The five Great Lakes that lie along the U.S.–Canada border are Huron, Ontario, Michigan, Erie, and Superior. Their names are easily remembered with the acronym HOMES. The built-in R time series `LakeHuron` consists of $n = 98$ monthly mean levels (in feet) of the lake level of Lake Huron taken at the Harbor Beach, Michigan water level gauge every July from 1875–1972. The measurements are essentially the number of feet above sea level of Lake Huron over time. Plot the time series, sample autocorrelation function, and sample partial autocorrelation function, and suggest a tentative time series model.

For simplicity, we define time $t = 1$ to be the year 1875 and $t = 98$ to be the year 1972. The time series of levels, the sample autocorrelation function, and the sample partial autocorrelation function can be graphed with the R statements

```
x = LakeHuron
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(x)
acf(x)
pacf(x)
```

The trio of graphs is displayed in Figure 9.18. A horizontal line has been added to the time series plot at $\bar{x} = 579$ feet. A visual assessment of the $n = 98$ observations from the time series reveals that the population mean level might be systematically decreasing relative to sea level over the time period. The tied observations in the years 1925–1926 ($t = 51$ and $t = 52$), the local minimum in year 1934 ($t = 60$), and the global minimum in year 1964 ($t = 90$), along with nearby observations, provide a downward tug on the mean value of the time series as time advances. Alternatively, the initial 15 or so levels might be drawn from a non-representative population. The population variance of the observations in the time series seems to be stable over time.

Now we turn to the sample autocorrelation function and sample partial autocorrelation function. The sample autocorrelation function appears to be tailing out. The initial positive spikes in the sample autocorrelation function are consistent with nearby observations in the time series lingering above and below the sample mean. The sample partial autocorrelation function has a statistically significant positive spike at lag 1, and a marginally significant negative spike at lag 2. The sample partial autocorrelation function does not have any statistically significant spikes after lag 2. These two graphs indicate that a shifted AR(2) model might be a reasonable tentative time series model.

The question concerning the possible downward trend of the level of Lake Huron in the time series remains a stumbling block to an enthusiastic recommendation of the shifted

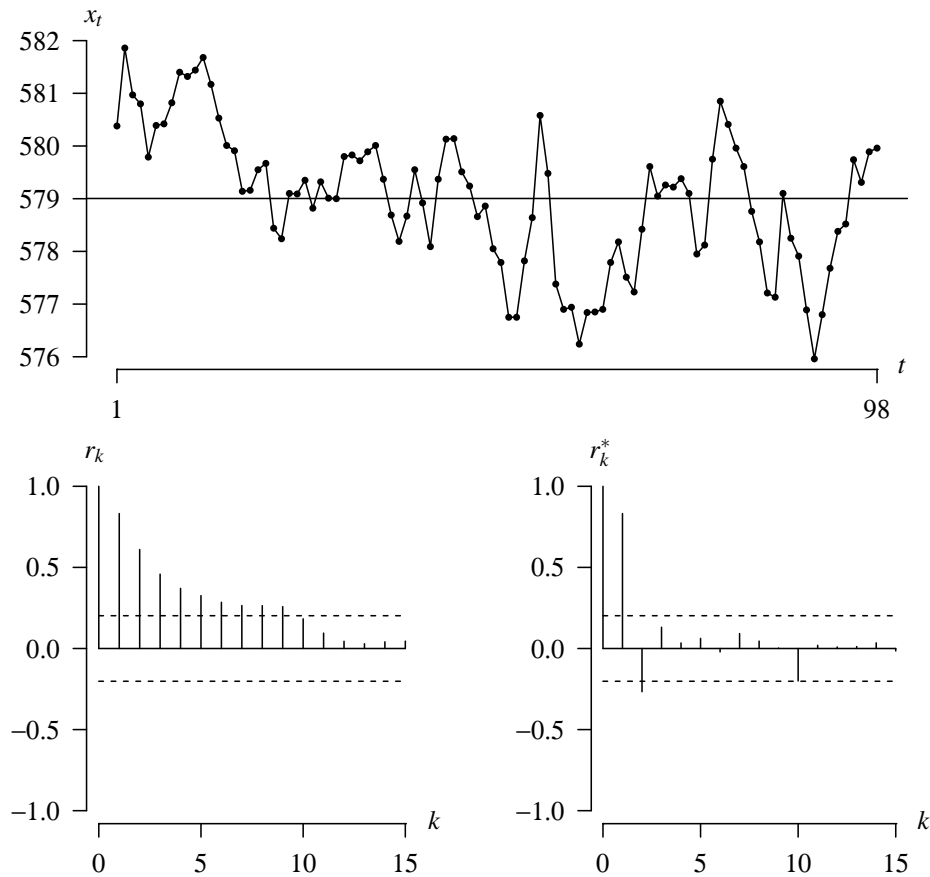


Figure 9.18: Time series plot, r_k , and r_k^* for $n = 98$ levels of Lake Huron (1875–1972).

AR(2) model. Perhaps a nonstationary model is appropriate. The list below provides four ways to proceed.

- Eliminate the first 15 or so observations from the time series if we can find an *assignable cause* that made the initial observations higher than the others. Here are some examples of potential assignable causes. Was the measuring equipment, location, procedure, or personnel changed at some point in the time series? There were various bridge, power, and flow control projects conducted during the early part of this time series. Some projects increased flow; others decreased flow. Could these projects account for the increased early observations in the time series? The primary driver of the year-to-year variability in water levels is the regional climate, which is influenced by global oceanic and atmospheric patterns. Lake Huron thermodynamics can be influenced by above-average lake evaporation rates. Might any of these factors account for the increased early observations in the time series? Did the episodic dredging of Lake Huron's outlet, the St. Clair River, result in a lowering of the level of Lake Huron? In the case of the beaver

temperature time series from Example 9.3, it was easy to find an assignable cause, because the beaver's temperature was clearly lower when in the lodge than when outside of the lodge. Such an assignable cause might be more difficult to identify in the case of the time series of Lake Huron levels.

- If an assignable cause cannot be found, fit a simple linear regression model to the original time series and consider the time series consisting of the original time series values minus the fitted values to be a stationary time series. Figure 9.19 shows the fitted regression line using the model

$$Y = \beta_0 + \beta_1 X + \epsilon,$$

where X is time (which is measured without error), Y is the random and continuous lake level, β_1 is the slope of the regression line, β_0 is the intercept of the regression line, and ϵ is an error term. The hypothesis test

$$H_0 : \beta_1 = 0$$

versus

$$H_1 : \beta_1 \neq 0$$

results in a tiny p -value ($p = 4 \cdot 10^{-8}$), which confirms our visual assessment. There does indeed appear to be a decrease in the level of the water in Lake Huron over time. The estimated regression coefficients and p -values can be calculated in R with the statement

```
summary(lm(LakeHuron ~ seq(1, 98)))
```

The estimated slope $\hat{\beta}_1 = -0.024$ indicates that the level of Lake Huron is decreasing by an average of about a quarter of an inch annually over this time horizon. Extrapolation of the simple linear regression model outside of the years 1875–1972 is probably not warranted in this setting. The usual regression assumption of independence is clearly violated in this setting because the observations in the time series are autocorrelated.

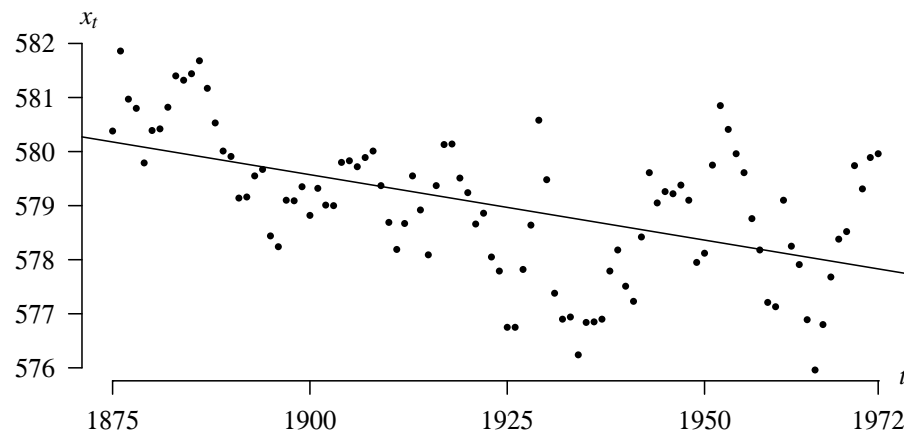


Figure 9.19: Lake Huron levels (1875–1972) with regression line.

- Again assuming that an assignable cause cannot be identified in order to remove the initial observations, the time series can be differenced (via $x_{i+1} - x_i$ for $i = 1, 2, \dots, n-1$) in order to remove the possible linear trend. The $n-1 = 97$ observations in this differenced series can then be evaluated as a stationary time series. This approach is analogous to the simple linear regression approach.
- Leave the time series alone, and assume that the early observations being larger than the rest of the observations is due to random sampling variability.

This might seem like a lot of fuss to establish a stationary time series model, but this crucial early detective work is common when trying to formulate a tentative time series model. We take the fourth approach from the list above for now and fit a tentative shifted AR(2) model to the $n = 98$ original observations in the time series. Fitting this tentative model will illustrate all of the steps involved with fitting, evaluating, and applying an AR(2) model: estimating the model parameters by the three techniques (method of moments, least squares, and maximum likelihood estimation), constructing confidence intervals for these point estimators, assessing the validity of the fitted model, performing model selection procedures for the AR(2) model, and forecasting the level of Lake Huron into the future. Later in the chapter, we will re-analyze this time series using a nonstationary time series model.

In conclusion, a preliminary graphical analysis of the $n = 98$ Lake Huron levels suggests a tentative AR(2) time series model should be on the short list. It is worthwhile investigating the possibility of an assignable cause which artificially elevates the initial 15 observations. There is significant concern about nonstationarity, which will be addressed later in the chapter. The next step is to estimate the parameters in the model.

Parameter Estimation

There are four parameters, μ , ϕ_1 , ϕ_2 , and σ_Z^2 , to estimate in the shifted AR(2) model

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + Z_t.$$

The three parameter estimation techniques outlined in Section 8.2.1, method of moments, least squares, and maximum likelihood estimation, are applied to the shifted AR(2) time series model next.

Approach 1: Method of moments. In the case of estimating the four parameters in the shifted AR(2) model by the method of moments, we match the population and sample (a) first-order moments, (b) second-order moments, (c) lag 1 autocorrelation, and (d) lag 2 autocorrelation. Placing the population moments on the left-hand side of the equation and the associated sample moments on the right-hand side of the equation results in four equations in four unknowns:

$$\begin{aligned} E[X_t] &= \frac{1}{n} \sum_{t=1}^n X_t \\ E[X_t^2] &= \frac{1}{n} \sum_{t=1}^n X_t^2 \\ \rho(1) &= r_1 \\ \rho(2) &= r_2. \end{aligned}$$

The expected value of X_t is μ , the expected value of X_t^2 can be found by using the shortcut formula for the population variance and by using the value of $\gamma(0) = V[X_t]$ from Theorem 9.10, and the values of $\rho(1)$ and $\rho(2)$ are also obtained from Theorem 9.10. So the four equations become

$$\begin{aligned}\mu &= \frac{1}{n} \sum_{t=1}^n X_t \\ \frac{(1-\phi_2)\sigma_Z^2}{(1+\phi_2)(1+\phi_1-\phi_2)(1-\phi_1-\phi_2)} + \mu^2 &= \frac{1}{n} \sum_{t=1}^n X_t^2 \\ \frac{\phi_1}{1-\phi_2} &= r_1 \\ \frac{\phi_1^2}{1-\phi_2} + \phi_2 &= r_2.\end{aligned}$$

Solving these equations for the four unknown parameters μ , ϕ_1 , ϕ_2 and σ_Z^2 yields closed-form solutions for the method of moments estimators

$$\begin{aligned}\hat{\mu} &= \bar{X} \\ \hat{\phi}_1 &= \frac{r_1(1-r_2)}{1-r_1^2} \\ \hat{\phi}_2 &= \frac{r_2-r_1^2}{1-r_1^2} \\ \hat{\sigma}_Z^2 &= \left[\frac{1}{n} \sum_{t=1}^n X_t^2 - \hat{\mu}^2 \right] \frac{(1+\hat{\phi}_2)(1+\hat{\phi}_1-\hat{\phi}_2)(1-\hat{\phi}_1-\hat{\phi}_2)}{1-\hat{\phi}_2}.\end{aligned}$$

This constitutes a proof of the following result.

Theorem 9.14 The method of moments estimators of the parameters in a shifted AR(2) model are

$$\begin{aligned}\hat{\mu} &= \bar{X}, & \hat{\phi}_1 &= \frac{r_1(1-r_2)}{1-r_1^2}, & \hat{\phi}_2 &= \frac{r_2-r_1^2}{1-r_1^2}, \\ \hat{\sigma}_Z^2 &= \left[\frac{1}{n} \sum_{t=1}^n X_t^2 - \hat{\mu}^2 \right] \frac{(1+\hat{\phi}_2)(1+\hat{\phi}_1-\hat{\phi}_2)(1-\hat{\phi}_1-\hat{\phi}_2)}{1-\hat{\phi}_2}.\end{aligned}$$

These estimators are random variables and have been written as a function of the random time series values $X_1, X_2, \dots, X_n$. For observed time series values $x_1, x_2, \dots, x_n$, the lowercase versions of the formulas will be used. These estimators are often known as the *Yule–Walker estimators* because their derivation involved the Yule–Walker equations from Theorem 9.10.

Example 9.15 For the time series of $n = 98$ Lake Huron levels from Example 9.14, find the method of moments estimators of μ , ϕ_1 , ϕ_2 , and σ_Z^2 for the AR(2) model.

The R code below calculates and prints the point estimates of the μ , ϕ_1 , ϕ_2 , and σ_Z^2 parameters using the method of moments estimators given in Theorem 9.14.

```
x = LakeHuron
```

```

r1      = acf(x, plot = FALSE)$acf[2]
r2      = acf(x, plot = FALSE)$acf[3]
muhat   = mean(x)
philhat = r1 * (1 - r2) / (1 - r1 ^ 2)
phi2hat = (r2 - r1 ^ 2) / (1 - r1 ^ 2)
sig2hat = (mean(x ^ 2) - muhat ^ 2) * (1 + phi2hat) *
          (1 + philhat - phi2hat) * (1 - philhat - phi2hat) /
          (1 - phi2hat)
print(c(muhat, philhat, phi2hat, sig2hat))

```

The point estimates for the unknown parameters computed by this code are

$$\hat{\mu} = 579.00 \quad \hat{\phi}_1 = 1.0538 \quad \hat{\phi}_2 = -0.26675 \quad \hat{\sigma}_Z^2 = 0.49199.$$

These point estimates are reported to five digits because the data values were given to five-digit accuracy. The positive value for $\hat{\phi}_1$ and the negative value for $\hat{\phi}_2$ are consistent with the sample partial autocorrelation function in Figure 9.18. Figure 9.20 is analogous to Figure 9.13 but contains just two of the level surfaces associated with the method of moments match on the population and sample autocorrelations at lags 1 and 2: the line associated with $\rho(1) = r_1 = 0.83191$ and the curve associated with $\rho(2) = r_2 = 0.60994$. These two level surfaces intersect at the point $(\hat{\phi}_1, \hat{\phi}_2) = (1.0538, -0.26675)$.

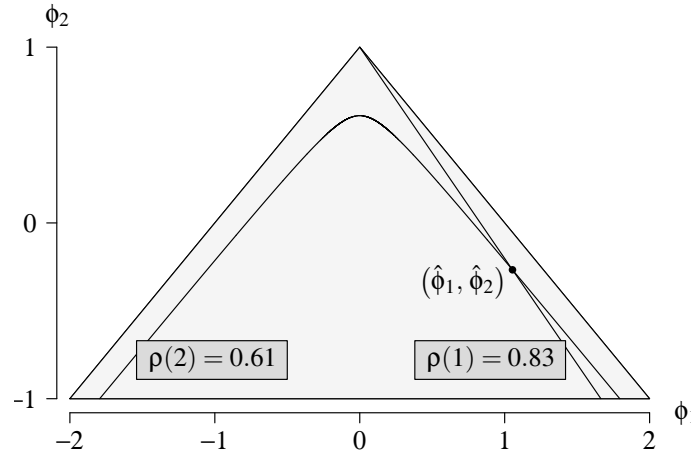


Figure 9.20: Level surfaces for the Lake Huron time series in the stationary region.

Approach 2: Least squares. Consider the shifted stationary AR(2) model

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + Z_t.$$

For least squares estimation, we first establish the sum of squares S as a function of the parameters μ , ϕ_1 , and ϕ_2 . This time, however, we forgo the calculus and leave the optimization to the R `optim` function in order to find the least squares estimators of μ , ϕ_1 , and ϕ_2 . Once these least squares estimators have been determined, the population variance of the white noise σ_Z^2 will be estimated.

The sum of squared errors is

$$S = \sum_{t=3}^n Z_t^2 = \sum_{t=3}^n [X_t - \mu - \phi_1 (X_{t-1} - \mu) - \phi_2 (X_{t-2} - \mu)]^2.$$

If this derivation were being done by hand, we would now calculate the partial derivatives of S with respect to the unknown parameters μ , ϕ_1 , and ϕ_2 , equate them to zero and solve. As was the case with the AR(1) model, there is no closed-form solution, so numerical methods are required to calculate the parameter estimates. In the example that follows, we will use the `optim` function in R to determine the least squares parameter estimates that minimize S .

The last parameter to estimate is σ_Z^2 . Since

$$\gamma(0) = \frac{(1 - \phi_2)\sigma_Z^2}{(1 + \phi_2)(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)}$$

from Theorem 9.10 for an AR(2) time series model, the population variance of the white noise can be expressed as

$$\sigma_Z^2 = \frac{(1 + \phi_2)(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)\gamma(0)}{(1 - \phi_2)}.$$

Replacing ϕ_1 and ϕ_2 by their least squares estimators $\hat{\phi}_1$ and $\hat{\phi}_2$, respectively, and replacing the lag 0 autocovariance $\gamma(0) = V[X_t]$ by its estimator $c_0 = \frac{1}{n} \sum_{t=1}^n (X_t - \bar{X})^2$ gives the estimator

$$\hat{\sigma}_Z^2 = \frac{(1 + \hat{\phi}_2)(1 + \hat{\phi}_1 - \hat{\phi}_2)(1 - \hat{\phi}_1 - \hat{\phi}_2)c_0}{(1 - \hat{\phi}_2)}.$$

This derivation constitutes a proof of the following result.

Theorem 9.15 The least squares estimators of the parameters in a shifted AR(2) model are the $\hat{\mu}$, $\hat{\phi}_1$, and $\hat{\phi}_2$ values that minimize

$$S = \sum_{t=3}^n Z_t^2 = \sum_{t=3}^n [X_t - \mu - \phi_1 (X_{t-1} - \mu) - \phi_2 (X_{t-2} - \mu)]^2$$

and the population variance of the white noise is estimated by

$$\hat{\sigma}_Z^2 = \frac{(1 + \hat{\phi}_2)(1 + \hat{\phi}_1 - \hat{\phi}_2)(1 - \hat{\phi}_1 - \hat{\phi}_2)c_0}{(1 - \hat{\phi}_2)}.$$

We now use numerical methods to find the least squares estimates for the unknown parameters in the AR(2) time series model for the Lake Huron time series from Example 9.14.

Example 9.16 Find the least squares estimators of μ , ϕ_1 , ϕ_2 , and σ_Z^2 for the AR(2) time series model associated with the $n = 98$ lake level observations from Example 9.14.

The R code that follows contains a function `s` which calculates the sum of squares, and then uses the R `optim` function to minimize the sum of squares using the method of moments estimates as initial estimates. The `optim` function minimizes the objective function by default.

```

x = LakeHuron
n = length(x)
s = function(parameters) {
  mu = parameters[1]
  phi1 = parameters[2]
  phi2 = parameters[3]
  sum((x[3:n] - mu - phi1 * (x[2:(n - 1)] - mu)
      - phi2 * (x[1:(n - 2)] - mu)) ^ 2)
}
fit = optim(c(579, 1.0538, -0.26675), s)$par
muhat = fit[1]
philhat = fit[2]
phi2hat = fit[3]
sig2hat = (1 + phi2hat) * (1 + philhat - phi2hat) *
  (1 - philhat - phi2hat) * mean((x - mean(x)) ^ 2) /
  (1 - phi2hat)

```

The point estimates for the unknown parameters computed by this code are

$$\hat{\mu} = 578.89 \quad \hat{\phi}_1 = 1.0217 \quad \hat{\phi}_2 = -0.23760 \quad \hat{\sigma}_Z^2 = 0.51680,$$

which corresponds to a sum of squares $S = 43.58$. The optimal sum of squares can be extracted with the additional R command `s(fit)`. These least squares point estimates of the unknown parameters in the AR(2) time series model are close to the associated method of moments point estimates. The left-hand graph in Figure 9.21 shows the sum of squares as a function of ϕ_1 for fixed values of the parameters $\hat{\mu} = 578.89$ and $\hat{\phi}_2 = -0.23760$. The sum of squares is minimized at $\hat{\phi}_1 = 1.0217$. The right-hand graph in Figure 9.21 shows the sum of squares as a function of ϕ_2 for fixed values of the parameters $\hat{\mu} = 578.89$ and $\hat{\phi}_1 = 1.0217$. The sum of squares is minimized at $\hat{\phi}_2 = -0.23760$.

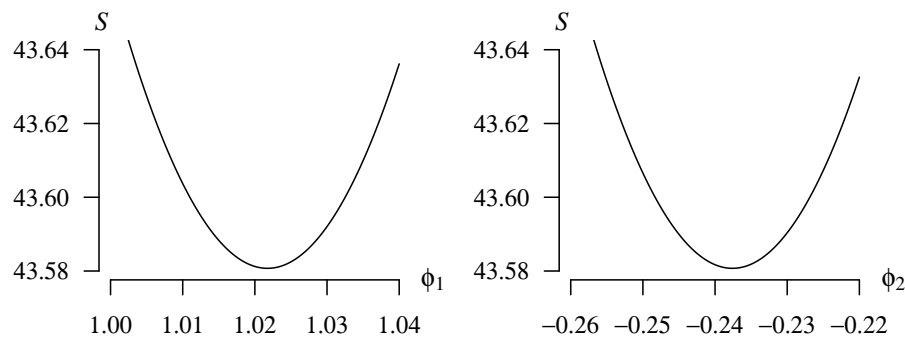


Figure 9.21: Sum of squares as a function of ϕ_1 and ϕ_2 for an AR(2) model.

Approach 3: Maximum likelihood estimation. The procedure for determining the maximum likelihood estimators for the unknown parameters in an AR(2) time series model follows along the same lines as in the AR(1) time series model from the previous section. Once again, to use maximum

likelihood estimation, we must assume that the random shocks from the white noise are Gaussian white noise, with associated probability density function

$$f_{Z_t}(z_t) = \frac{1}{\sqrt{2\pi\sigma_Z^2}} e^{-z_t^2/(2\sigma_Z^2)} \quad -\infty < z_t < \infty,$$

for $t = 1, 2, \dots, n$. Determining the likelihood function, which is the joint probability density function of the observed values in the time series $X_1, X_2, \dots, X_n$, involves finding

$$L(\mu, \phi_1, \phi_2, \sigma_Z^2) = f(x_1, x_2, \dots, x_n),$$

where the $x_1, x_2, \dots, x_n$ arguments on L and the μ, ϕ_1, ϕ_2 , and σ_Z^2 arguments on f have been dropped for brevity. As before, it is not possible to simply multiply the marginal probability density functions because the values in the AR(2) time series model are correlated. As in the case of an AR(1) model, we use the transformation technique to find the conditional joint probability density function of $X_3, X_4, \dots, X_n$ conditioned on $X_1 = x_1$ and $X_2 = x_2$, which is denoted by

$$f_{X_3, X_4, \dots, X_n | X_1, X_2}(x_3, x_4, \dots, x_n | X_1 = x_1, X_2 = x_2)$$

for $(x_3, x_4, \dots, x_n) \in \mathcal{R}^{n-2}$. This conditional joint probability density function is multiplied by the marginal joint probability density function of X_1 and X_2 (which has the bivariate normal distribution) resulting in a joint probability density function of $X_1, X_2, \dots, X_n$:

$$f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = f_{X_3, X_4, \dots, X_n | X_1, X_2}(x_3, x_4, \dots, x_n | X_1 = x_1, X_2 = x_2) f_{X_1, X_2}(x_1, x_2)$$

for $(x_1, x_2, \dots, x_n) \in \mathcal{R}^n$. This function serves as the likelihood function, which should be maximized with respect to the unknown parameters μ, ϕ_1, ϕ_2 , and σ_Z^2 . One can easily imagine how complicated this expression is, based on the values of $\gamma(0)$ and $\gamma(1)$ from Theorem 9.10. So we forgo the tedious mathematics and leave the calculations to the `ar` function in R when determining the maximum likelihood estimates for the parameters in fitting the Lake Huron time series to the shifted AR(2) time series model.

Example 9.17 Find the maximum likelihood estimators of μ, ϕ_1, ϕ_2 , and σ_Z^2 for the time series of $n = 98$ observations of the level of Lake Huron from Example 9.14 for a shifted AR(2) time series model

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + Z_t,$$

where ϕ_1, ϕ_2, μ , and $\sigma_Z^2 > 0$ are real-valued parameters and $\{Z_t\}$ is a time series of Gaussian white noise.

The point estimates for the unknown parameters in the shifted AR(2) time series model are computed by the single R command

```
ar(LakeHuron, order.max = 2, aic = FALSE, method = "mle")
```

Unlike the procedure described above, the `ar` function first subtracts the sample mean $\bar{x} = 579$ from each observation and then proceeds to fit the remaining parameters to the standard AR(2) time series model. This function call returns the maximum likelihood estimates for the parameters as

$$\hat{\mu} = 579.00 \quad \hat{\phi}_1 = 1.0437 \quad \hat{\phi}_2 = -0.2496 \quad \hat{\sigma}_Z^2 = 0.4788.$$

These parameter estimates are near the associated method of moments and least squares estimates from the previous two examples. The fitted shifted AR(2) model by maximum likelihood estimation is

$$X_t - 579.00 = 1.0437(X_{t-1} - 579.00) - 0.2496(X_{t-2} - 579.00) + Z_t,$$

where $Z_t \sim N(0, 0.4788)$.

Table 9.6 summarizes the point estimators for the AR(2) model for the Lake Huron time series calculated by the R commands

```
ar(LakeHuron, order.max = 2, aic = FALSE, method = "yule-walker")
ar(LakeHuron, order.max = 2, aic = FALSE, method = "ols")
ar(LakeHuron, order.max = 2, aic = FALSE, method = "mle")
```

The point estimators associated with the three methods are quite close for this particular time series. The R function `ar` fits autoregressive models. There are tiny differences between some of the entries in Table 9.6 and those from Examples 9.15 and 9.16 which might be due to slightly different approximations and/or roundoff in the optimization routines.

Method	$\hat{\mu}$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\sigma}_Z^2$
Method of moments (Yule–Walker)	579.0	1.0538	−0.2668	0.5075
Ordinary least squares	579.0	1.0217	−0.2376	0.4540
Maximum likelihood estimation	579.0	1.0437	−0.2496	0.4788

Table 9.6: AR(2) point estimators for the $n = 98$ Lake Huron levels via the `ar` function.

The focus on estimation thus far has been on point estimation techniques. We also want to report some indication of the *precision* associated with these point estimators. The sampling distributions of $\hat{\mu}$, $\hat{\phi}_1$, $\hat{\phi}_2$, and $\hat{\sigma}_Z^2$ in the AR(2) model are too complicated to derive analytically. As an illustration of how to construct an approximate confidence interval for ϕ_1 or ϕ_2 , we use the asymptotic normality of the maximum likelihood estimator of ϕ_1 and ϕ_2 in the following result. The asymptotic variance–covariance matrix associated with the parameters ϕ_1 and ϕ_2 is

$$\frac{1}{n} \begin{bmatrix} 1 - \phi_2^2 & -\phi_1(1 + \phi_2) \\ -\phi_1(1 + \phi_2) & 1 - \phi_2^2 \end{bmatrix}.$$

Using just the diagonal elements of this matrix results in the following asymptotically exact two-sided $100(1 - \alpha)\%$ confidence interval for ϕ_1 and ϕ_2 .

Theorem 9.16 For a stationary AR(2) time series model, an asymptotically exact two-sided $100(1 - \alpha)\%$ confidence interval for ϕ_i is given by

$$\hat{\phi}_i - z_{\alpha/2} \sqrt{\frac{1 - \hat{\phi}_2^2}{n}} < \phi_i < \hat{\phi}_i + z_{\alpha/2} \sqrt{\frac{1 - \hat{\phi}_2^2}{n}}$$

for $i = 1, 2$, where $\hat{\phi}_i$ is the maximum likelihood estimator of ϕ_i and $z_{\alpha/2}$ is the $1 - \alpha/2$ fractile of the standard normal distribution.

These asymptotically exact confidence intervals for ϕ_1 and ϕ_2 will now be illustrated for the lake levels from the Lake Huron time series from the previous four examples.

Example 9.18 Find an approximate 95% confidence interval for ϕ_1 for the AR(2) time series model associated with the $n = 98$ Lake Huron time series values from Example 9.14 and assess its actual coverage.

Recall from Table 9.6 that the maximum likelihood estimators of ϕ_1 and ϕ_2 returned by the `ar` function are $\hat{\phi}_1 = 1.0437$ and $\hat{\phi}_2 = -0.2496$. We seek an asymptotically exact two-sided 95% confidence interval for ϕ_1 , which is given by

$$1.0437 - 1.96\sqrt{\frac{1 - (-0.2496)^2}{98}} < \phi_1 < 1.0437 + 1.96\sqrt{\frac{1 - (-0.2496)^2}{98}}$$

or

$$0.8519 < \phi_1 < 1.2354.$$

This confidence interval does not contain $\phi_1 = 0$, which leads us to conclude that ϕ_1 is a statistically significant parameter in the AR(2) model. A similar procedure could be used to find a confidence interval for ϕ_2 .

To assess the actual coverage of this 95% confidence interval for ϕ_1 requires a Monte Carlo simulation experiment. The code below uses population parameters that are near the parameter estimates for the Lake Huron time series.

```
n      = 98
mu     = 579
phi1   = 1
phi2   = -1 / 4
sigz   = sqrt(1 / 2)
crit   = qnorm(0.975)
nrep   = 40000
count  = 0
for (i in 1:nrep) {
  x      = mu + arima.sim(model = list(ar = c(phi1, phi2), sd = sigz),
                           n = n, n.start = 50)
  fit    = arima(x, order = c(2, 0, 0), method = "ML")
  phi1hat = fit$coef[1]
  phi2hat = fit$coef[2]
  std    = sqrt((1 - phi2hat ^ 2) / n)
  lo     = phi1hat - crit * std
  hi     = phi1hat + crit * std
  if (lo < phi1 && hi > phi1) count = count + 1
}
print(count / nrep)
```

After a call to `set.seed(3)` to establish the random number stream, five runs of this simulation yield:

0.9376 0.9390 0.9366 0.9385 0.9395.

The conclusion that can be drawn from these simulations is that the actual coverage of the approximate 95% confidence interval is about 93.8%. When this code is executed for larger values of n , the anticipated asymptotic results are achieved, as displayed in Figure 9.22. Keep in mind that these actual coverages are not for an AR(2) model in general, but rather an AR(2) model with these particular parameter settings.

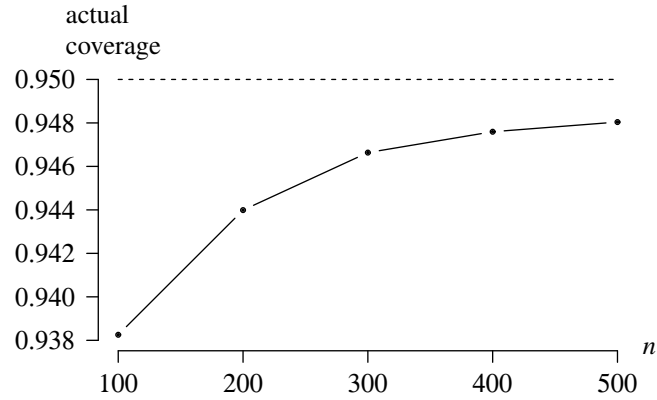


Figure 9.22: Asymptotic 95% confidence interval for ϕ_1 actual coverage.

Model Assessment

Now that techniques for point and interval estimates for the parameters in the AR(2) model have been established, we are interested in assessing the adequacy of the AR(2) time series model. This will involve an analysis of the residuals. Recall from Section 8.2.3 that the residuals are defined by

$$[\text{residual}] = [\text{observed value}] - [\text{predicted value}]$$

or

$$\hat{Z}_t = X_t - \hat{X}_t.$$

Since $\hat{X}_t$ is the one-step-ahead forecast from the time origin $t - 1$, this is more clearly written as

$$\hat{Z}_t = X_t - \hat{X}_{t-1}(1).$$

From Theorem 9.13, the shifted AR(2) model is

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + Z_t$$

or

$$X_t = \mu + \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + Z_t.$$

Taking the conditional expected value of both sides of this equation gives

$$E[X_t | X_1 = x_1, X_2 = x_2, \dots, X_{t-1} = x_{t-1}] = \mu + \phi_1 (x_{t-1} - \mu) + \phi_2 (x_{t-2} - \mu).$$

Replacing the parameters by their point estimators, the one-step-ahead forecast from the time origin $t - 1$ is

$$\hat{X}_{t-1}(1) = \hat{\mu} + \hat{\phi}_1 (x_{t-1} - \hat{\mu}) + \hat{\phi}_2 (x_{t-2} - \hat{\mu}).$$

Therefore, for the time series $x_1, x_2, \dots, x_n$ and the fitted AR(2) model with parameter estimates $\hat{\mu}$, $\hat{\phi}_1$, and $\hat{\phi}_2$, the residual at time t is

$$\hat{Z}_t = x_t - [\hat{\mu} + \hat{\phi}_1 (x_{t-1} - \hat{\mu}) + \hat{\phi}_2 (x_{t-2} - \hat{\mu})]$$

for $t = 3, 4, \dots, n$. The next example shows the steps associated with assessing the adequacy of the AR(2) model for the Lake Huron lake level time series.

Example 9.19 Fit the AR(2) model to the Lake Huron levels from Example 9.14 using the sample mean to estimate μ and the maximum likelihood estimators for ϕ_1 , ϕ_2 , and σ_Z^2 .

- (a) Calculate and plot the residuals, their sample autocorrelation function, and their sample partial autocorrelation function.
 - (b) Conduct a test of independence on the residuals using the number of sample autocorrelation function values for the first $m = 40$ lags which fall outside of $\pm 1.96/\sqrt{n}$.
 - (c) Conduct the Box–Pierce and Ljung–Box tests for independence of the residuals.
 - (d) Conduct the turning point test for independence of the residuals.
 - (e) Plot a histogram and a QQ plot of the standardized residuals in order to assess the normality of the residuals.
- (a) The following R commands calculate the $n - 2 = 96$ residuals and plot them as a time series, along with the associated sample autocorrelation function and sample partial autocorrelation function.

```
x      = LakeHuron
n      = length(x)
m      = 40
muhat  = mean(x)
fit     = ar(x, order.max = 2, aic = FALSE, method = "mle")
phi1hat = fit$ar[1]
phi2hat = fit$ar[2]
zhat    = x[3:n] - (muhat + phi1hat * (x[2:(n - 1)] - muhat) +
                  phi2hat * (x[1:(n - 2)] - muhat))
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(zhat)
acf(zhat, lag.max = m)
pacf(zhat, lag.max = m)
```

The results are displayed in Figure 9.23. The residuals do not appear to have any cyclic variation, trend, or serial correlation.

- (b) There are no sample autocorrelation function values that fall outside of the limits $\pm 1.96/\sqrt{n}$ in the plot in Figure 9.23 of the first 40 sample autocorrelation function values associated with the residuals. Since we expect $40 \cdot 0.05 = 2$ values to fall outside of these limits in the case of a good fit, we fail to reject H_0 in this case. The fit of the AR(2) model is not rejected by this test.
- (c) The additional R code below calculates the Box–Pierce test statistic and the Ljung–Box test statistic and the associated p -values using the built-in `Box.test` function.

```
Box.test(zhat, lag = 40, type = "Box-Pierce", fitdf = 3)
Box.test(zhat, lag = 40, type = "Ljung-Box", fitdf = 3)
```

The Box–Pierce test statistic is 18.7 and the associated p -value is $p = 0.995$. The Ljung–Box test statistic is 24.9 and the associated p -value is $p = 0.935$. We fail to reject H_0 in both tests based on the chi-square critical value with $40 - 3 = 37$ degrees of freedom. The fit of the AR(2) model is not rejected by these tests.

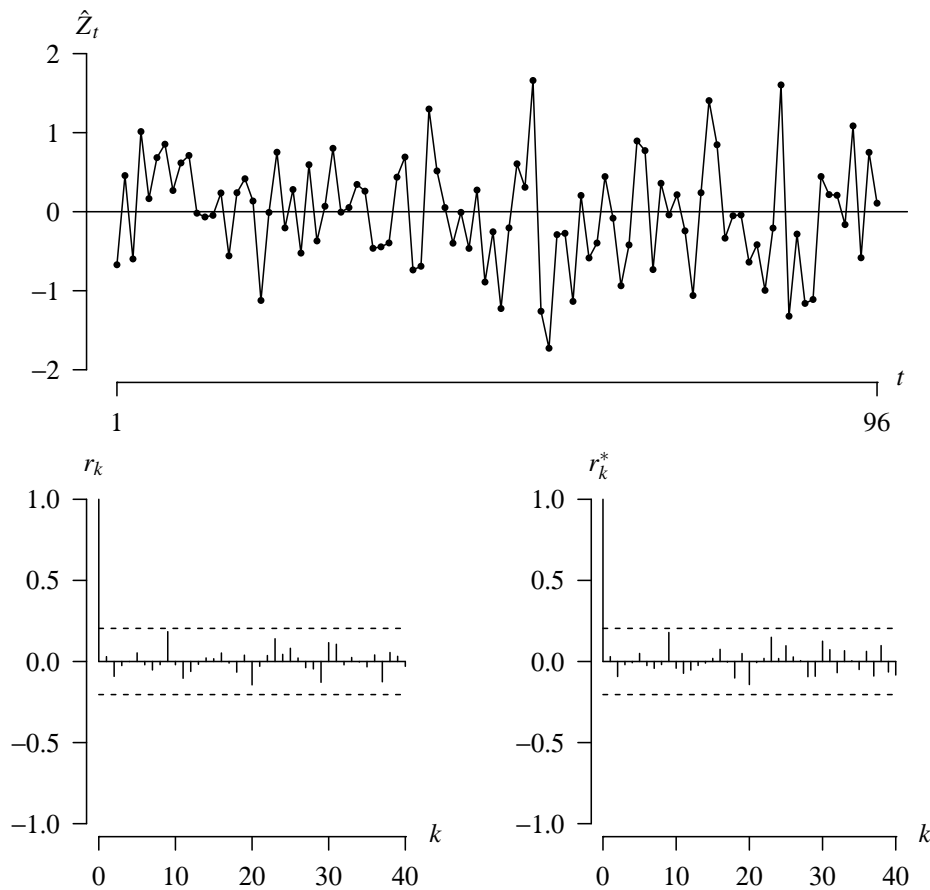


Figure 9.23: Time series plot, r_k , and r_k^* for $n - 2 = 96$ residuals from AR(2) fitted model.

- (d) The following additional R code calculates the test statistic and the p -value for the turning point test applied to the time series consisting of the $n - 2 = 96$ residual values for the AR(2) fit to the Lake Huron time series.

```
n = n - 2
m = (2 / 3) * (n - 2)
v = (16 * n - 29) / 90
T = 0
for (i in 2:(n - 1)) {
  if ((zhat[i - 1] < zhat[i] && zhat[i] > zhat[i + 1]) ||
      (zhat[i - 1] > zhat[i] && zhat[i] < zhat[i + 1])) T = T + 1
}
s = (T - m) / sqrt(v)
2 * (1 - pnorm(abs(s)))
```

The tail probability is doubled because the alternative hypothesis is two-tailed for the turning point test. The test statistic s is 0.0815 and the p -value is $p = 0.94$. The

turning point test found that there were $T = 63$ turning points in the time series of the residuals, and that is about the number that we expect to have if the residuals from the fitted AR(2) model were mutually independent random variables. We again fail to reject the null hypothesis in this case. The fit of the AR(2) model is not rejected by this test.

- (e) The residuals are standardized by dividing by their sample standard deviation. The following additional R statements plot a histogram of the standardized residuals using the `hist` function and a QQ plot to assess normality using the `qqnorm` function.

```
hist(zhat / sd(zhat))
qqnorm(zhat / sd(zhat))
```

The plots are shown in Figure 9.24. The histogram shows that all standardized residuals fall between -3 and 3 and exhibit a roughly bell-shaped probability distribution. The horizontal axis on the histogram is the standardized residual and the vertical axis is the frequency. The QQ plot is approximately linear, indicating a reasonable approximation to normality based on the $n - 2 = 96$ residuals plotted. The horizontal axis on the QQ plot is the standardized theoretical quantile and the vertical axis is the associated normal data quantile. Although a formal statistical goodness-of-fit test (such as the Shapiro–Wilk or the Kolmogorov–Smirnov test) should be conducted, it appears that the assumption of Gaussian white noise is appropriate for the AR(2) time series model based on these two plots.

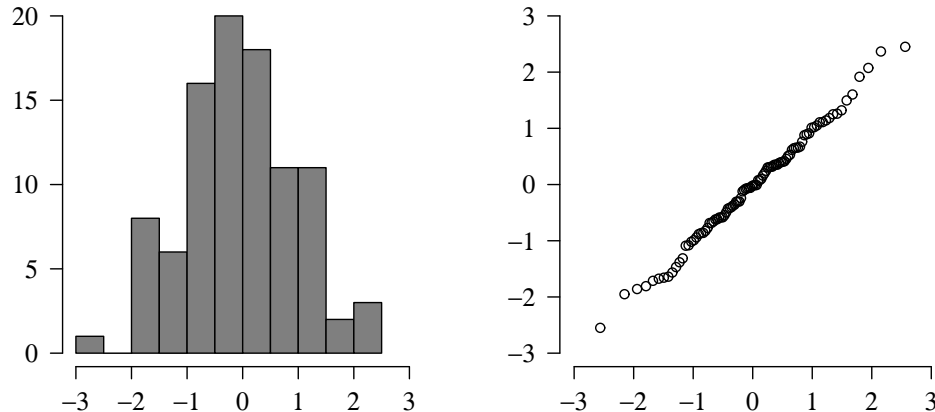


Figure 9.24: Histogram (left) and QQ plot (right) of the fitted AR(2) standardized residuals.

Model Selection

We have seen a number of indicators that the AR(2) time series model seems to be an adequate model for the Lake Huron lake level time series, with the exception of a linear trend apparent by viewing the time series in Figure 9.18. The model has not been rejected by any of the model adequacy tests. We now overfit the tentative AR(2) time series model with ARMA(p, q) models of higher order. We have not yet surveyed the techniques for estimating the parameters in these models with additional terms, so for now we will let the `arima` function in R estimate their parameters

and compare them via their AIC (Akaike's Information Criterion) statistics. The AIC statistic was introduced in Section 8.2.4

Example 9.20 For the $n = 98$ levels of Lake Huron from Example 9.14, determine the $\text{ARMA}(p, q)$ model that minimizes the AIC.

The R code below creates a 4×4 matrix `a` which will be populated with the AIC statistics for the $\text{ARMA}(p, q)$ time series models, for $p = 0, 1, 2, 3$ and $q = 0, 1, 2, 3$ using nested for loops. The `arima` function is used to fit the models via maximum likelihood estimation, and the AIC values are placed in the matrix `a`.

```
a = matrix(0, 4, 4)
x = LakeHuron
for (p in 0:3)
  for (q in 0:3)
    a[p + 1, q + 1] = arima(x, order = c(p, 0, q), method = "ML")$aic
```

The results of this code are given in Table 9.7. The two smallest AIC values are set in boldface type; they correspond to the AR(2) and ARMA(1, 1) models. These two models seem to be close competitors for providing a probabilistic model for the time series.

	$q = 0$	$q = 1$	$q = 2$	$q = 3$
$p = 0$	335	255	231	222
$p = 1$	219	214	216	218
$p = 2$	215	216	218	220
$p = 3$	216	218	220	220

Table 9.7: AIC statistics for $\text{ARMA}(p, q)$ models for the $n = 98$ lake water levels.

The `$` extractor with the `aic` argument was used to extract the AIC statistics from the list returned by the call to `arima`. If the `coef` and `sigma2` components are extracted from the list returned by the call to `arima`, our final model is the AR(2) model with maximum likelihood estimates for the parameters given by

$$\hat{\mu} = 579.05 \quad \hat{\phi}_1 = 1.0436 \quad \hat{\phi}_2 = -0.24949 \quad \hat{\sigma}_Z^2 = 0.47882,$$

which corresponds to the fitted AR(2) model

$$X_t - 579.05 = 1.0436(X_{t-1} - 579.05) - 0.24949(X_{t-2} - 579.05) + Z_t,$$

where Z_t is a time series of Gaussian white noise values with $\sigma_Z^2 = 0.47822$, as established by the histogram and QQ plot in Example 9.19.

The analysis here suggests that this tentative fitted shifted AR(2) time series model should be compared with (a) a shifted ARMA(1, 1) model because of the lower value for its AIC for an identical number of parameters, and (b) a time series model based on removing the possible downward trend in the time series by using regression or differencing as described in Example 9.14.

Forecasting

We now consider forecasting future values of a time series that is governed by a shifted AR(2) time series model. In the case of the Lake Huron time series, this corresponds to the one-step-ahead forecast for 1973, the two-steps-ahead forecast for 1974, the three-steps-ahead forecast for 1975, etc. To review forecasting notation, the observed time series values are $x_1, x_2, \dots, x_n$. The forecast is being made at time $t = n$. The random future value of the time series that is h time units in the future is denoted by X_{n+h} . The associated forecasted value is denoted by $\hat{X}_{n+h}$, and is the conditional expected value

$$\hat{X}_{n+h} = E[X_{n+h} | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n].$$

We would like to find this forecasted value and an associated prediction interval for a shifted AR(2) model. As in Section 8.2.2, we assume that all parameters are known in the derivations that follow. We also assume that the parameters ϕ_1 and ϕ_2 correspond to a stationary shifted AR(2) time series model.

The shifted AR(2) model is

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + Z_t.$$

Replacing t by $n+1$ and solving for X_{n+1} , this becomes

$$X_{n+1} = \mu + \phi_1 (X_n - \mu) + \phi_2 (X_{n-1} - \mu) + Z_{n+1}.$$

Taking the conditional expected value of each side of this equation results in the one-step-ahead forecast

$$\hat{X}_{n+1} = \mu + \phi_1 (x_n - \mu) + \phi_2 (x_{n-1} - \mu)$$

because x_{n-1} and x_n have already been observed in the time series $x_1, x_2, \dots, x_n$. The forecasted value at time $n+1$ is a function of the last two values in the time series. Applying this same process to the predicted value at time $n+2$ results in the time series model

$$X_{n+2} = \mu + \phi_1 (X_{n+1} - \mu) + \phi_2 (X_n - \mu) + Z_{n+2}.$$

This time, the value of X_{n+1} has not been observed, so we replace it by its forecasted value when taking the conditional expected value of both sides of the equation

$$\hat{X}_{n+2} = \mu + \phi_1 (\hat{X}_{n+1} - \mu) + \phi_2 (x_n - \mu),$$

because x_n has already been observed. Continuing in this fashion, a recursive formula for the forecasted value of X_{n+h} is

$$\hat{X}_{n+h} = \mu + \phi_1 (\hat{X}_{n+h-1} - \mu) + \phi_2 (\hat{X}_{n+h-2} - \mu).$$

Although we would prefer an explicit formula, the recursive formula is easy to implement for an observed time series $x_1, x_2, \dots, x_n$. As in the case of the AR(1) model, long-term forecasts for a stationary AR(2) time series model tend to μ as the time horizon $h \rightarrow \infty$.

We would like to pair our point estimator $\hat{X}_{n+h}$ with an interval estimator, which is a prediction interval in this setting. The prediction interval gives us an indication of the precision of the forecast. In order to derive an exact two-sided $100(1 - \alpha)\%$ prediction interval for X_{n+h} , it is helpful to write the shifted AR(2) model as a shifted MA(∞) model. The coefficients $\theta_1, \theta_2, \dots$ of a stationary shifted AR(2) model written as an MA(∞) model

$$X_t = \mu + Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \dots$$

are given in terms of ϕ_1 and ϕ_2 in Theorem 9.12. Consider this model at time $t = n + 1$. Since the error terms $Z_n, Z_{n-1}, Z_{n-2}, \dots$ are unknown but fixed because they are associated with the observed time series $x_1, x_2, \dots, x_n$, the conditional population variance of X_{n+1} is

$$V[X_{n+1}] = V[Z_{n+1}] = \sigma_Z^2$$

because the population variance of μ is zero and Z_{n+1} is the only random term in the model. The error terms at time n and prior are observed even though unknown and can therefore be treated as constants. Likewise, considering the MA(∞) model at time $t = n + 2$, the conditional population variance of X_{n+2} is

$$V[X_{n+2}] = V[Z_{n+2} + \theta_1 Z_{n+1}] = (1 + \theta_1^2) \sigma_Z^2.$$

Similarly, the conditional population variance of X_{n+3} is

$$V[X_{n+3}] = V[Z_{n+3} + \theta_1 Z_{n+2} + \theta_2 Z_{n+1}] = (1 + \theta_1^2 + \theta_2^2) \sigma_Z^2.$$

Continuing in this fashion, the conditional population variance of X_{n+h} is

$$V[X_{n+h}] = (1 + \theta_1^2 + \theta_2^2 + \dots + \theta_{h-1}^2) \sigma_Z^2.$$

If we assume that the white noise terms in the MA(∞) representation of the AR(2) time series model are Gaussian white noise terms, then X_{n+h} is also normally distributed because a linear combination of mutually independent normal random variables is also normally distributed. So an exact two-sided $100(1 - \alpha)\%$ prediction interval for X_{n+h} is

$$\hat{X}_{n+h} - z_{\alpha/2} \sqrt{1 + \theta_1^2 + \theta_2^2 + \dots + \theta_{h-1}^2} \sigma_Z < X_{n+h} < \hat{X}_{n+h} + z_{\alpha/2} \sqrt{1 + \theta_1^2 + \theta_2^2 + \dots + \theta_{h-1}^2} \sigma_Z.$$

In most practical problems, the parameters in this prediction interval will be estimated from data, which results in the following *approximate* two-sided $100(1 - \alpha)\%$ prediction interval.

Theorem 9.17 For a stationary shifted AR(2) time series model, a forecasted value of X_{n+h} can be found by the recursive equation

$$\hat{X}_{n+h} = \hat{\mu} + \hat{\phi}_1 (\hat{X}_{n+h-1} - \hat{\mu}) + \hat{\phi}_2 (\hat{X}_{n+h-2} - \hat{\mu}),$$

where $\hat{X}_{n+1} = \hat{\mu} + \hat{\phi}_1 (x_n - \hat{\mu}) + \hat{\phi}_2 (x_{n-1} - \hat{\mu})$. An approximate two-sided $100(1 - \alpha)\%$ prediction interval for X_{n+h} is

$$\hat{X}_{n+h} - z_{\alpha/2} \sqrt{1 + \hat{\theta}_1^2 + \hat{\theta}_2^2 + \dots + \hat{\theta}_{h-1}^2} \hat{\sigma}_Z < X_{n+h} < \hat{X}_{n+h} + z_{\alpha/2} \sqrt{1 + \hat{\theta}_1^2 + \hat{\theta}_2^2 + \dots + \hat{\theta}_{h-1}^2} \hat{\sigma}_Z,$$

where $\hat{\theta}_1, \hat{\theta}_2, \dots$ are the estimated coefficients in the MA(∞) model associated with the estimated AR(2) model.

Example 9.21 For the time series of Lake Huron levels $x_1, x_2, \dots, x_{98}$ from Example 9.14, forecast the next five values (for years 1973–1977) in the time series and give approximate 95% prediction intervals for the forecasted values assuming that the time series arises from a shifted AR(2) model with parameters estimated by maximum likelihood.

The R code below uses the `ar` function to estimate the parameters in the shifted AR(2) model via maximum likelihood estimation. The `predict` function implements Theorem 9.17 to calculate the forecasted values and associated standard errors. These standard errors can be used to calculate approximate 95% prediction interval limits.

```
model = ar(LakeHuron, order.max = 2, aic = FALSE, method = "mle")
predict(model, n.ahead = 5)
```

The results are summarized in Table 9.8. Notice that the forecasts trend monotonically toward $\bar{x} = 579$ and the standard errors increase as the time horizon h increases. The increasing standard error is consistent with having less precision in the forecast as the time horizon h increases.

Time	$t = 99$	$t = 100$	$t = 101$	$t = 102$	$t = 103$
Year	1973	1974	1975	1976	1977
Forecast	579.79	579.59	579.43	579.31	579.23
Standard error	0.692	1.000	1.157	1.233	1.269
Lower prediction bound	578.43	577.63	577.16	576.89	576.74
Upper prediction bound	581.15	581.55	581.70	581.73	581.71

Table 9.8: Forecasts and 95% prediction intervals for the Lake Huron time series.

Figure 9.25 shows (a) the original time series $x_1, x_2, \dots, x_{98}$ as points ($\bullet$) connected by lines, (b) the first 10 forecasted lake levels $\hat{X}_{99}, \hat{X}_{100}, \dots, \hat{X}_{108}$ as open circles ($\circ$), (c) the 95% prediction intervals as a shaded region, and (d) the next 10 actual average lake level values in July for the years 1973–1982 taken from the NOAA Great Lakes Experimental Research Laboratory website,

580.98, 581.04, 580.49, 580.52, 578.57, 578.96, 579.94, 579.77, 579.44, 578.97,

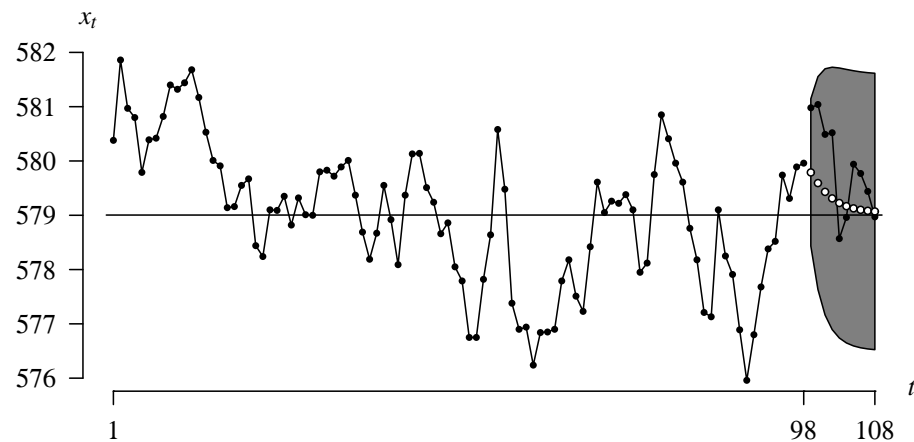


Figure 9.25: Lake Huron forecasts and 95% prediction intervals.

as points (•) connected by lines. There are four key observations concerning Figure 9.25.

- Even though the last three observations in the Lake Huron water level time series, $x_{96} = 579.31$, $x_{97} = 579.89$, and $x_{98} = 579.96$, show an increasing trend, the forecasts, which are a function only of $x_{n-1} = x_{97}$ and $x_n = x_{98}$, monotonically approach $\hat{\mu} = \bar{x} = 579$. The reason that the forecasts approach $\hat{\mu} = \bar{x} = 579$ in a damped exponential fashion is that the maximum likelihood estimators $\hat{\phi}_1$ and $\hat{\phi}_2$ satisfy $\hat{\phi}_1^2 + 4\hat{\phi}_2 > 0$, which indicates that the characteristic equation has two real roots which fall outside of the unit circle in the complex plane (see the proof of Theorem 9.9). Had the two roots been complex conjugates, the forecasted values would likewise approach $\hat{\mu} = \bar{x} = 579$, but in a damped sinusoidal fashion.
- The widths of the prediction intervals increase as the time horizon h increases. These widths do not increase indefinitely, but rather approach a limit as $h \rightarrow \infty$.
- The random sampling variability which is evident in the observed time series values $x_1, x_2, \dots, x_{98}$ is not apparent in the forecasted values $\hat{X}_{99}, \hat{X}_{100}, \dots, \hat{X}_{108}$. Observed time series values tend to exhibit the typical random sampling variability; forecasted values for a stationary shifted AR(2) time series model tend to be smooth.
- The first actual value in the forecast region, $x_{99} = 580.98$ for the year 1973, nearly falls outside of the associated 95% prediction interval. Even if the AR(2) model is a good fit for this time series, there is still a probability of approximately 0.05 that a future observation will fall outside of the associated 95% prediction interval. One value out of ten falling outside of the prediction intervals would not be shocking to see, assuming that a reasonable time series model has been formulated.

This section has introduced the AR(2) time series model. The important results for an AR(2) model are listed below.

- The standard AR(2) model can be written algebraically and with the backshift operator B as

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t \quad \text{and} \quad \phi(B)X_t = Z_t,$$

where $\phi(B) = 1 - \phi_1 B - \phi_2 B^2$ is the characteristic polynomial and $Z_t \sim WN(0, \sigma_Z^2)$ (Definition 9.2).

- The shifted AR(2) model can be written algebraically and with the backshift operator B as (Theorem 9.13)

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + Z_t \quad \text{and} \quad \phi(B)(X_t - \mu) = Z_t.$$

- The AR(2) model is always invertible; the AR(2) model is stationary when ϕ_1 and ϕ_2 fall in a triangular-shaped region in the (ϕ_1, ϕ_2) plane defined by the constraints $\phi_1 + \phi_2 < 1$, $\phi_2 - \phi_1 < 1$, and $\phi_2 > -1$ (Theorem 9.9).
- The AR(2) population autocorrelation function is a mixture of damped exponential functions, when $\phi(B)$ has real roots, or a damped sinusoidal function, when $\phi(B)$ has complex roots (Theorem 9.10).

- The AR(2) population partial autocorrelation function cuts off after lag 2 (Theorem 9.11), making its shape easier to recognize than the population autocorrelation function for the statistical counterparts associated with a realization of a time series.
- The stationary shifted AR(2) model can be written as a shifted MA(∞) model (Theorem 9.12).
- The four parameters in the shifted AR(2) model, μ , ϕ_1 , ϕ_2 and σ_Z^2 , can be estimated from a realization of a time series $x_1, x_2, \dots, x_n$ by the method of moments (Theorem 9.14), least squares (Theorem 9.15), and maximum likelihood using at least $n = 60$ or $n = 70$ observations. The point estimators for μ , ϕ_1 , ϕ_2 , and σ_Z^2 are denoted by $\hat{\mu}$, $\hat{\phi}_1$, $\hat{\phi}_2$, and $\hat{\sigma}_Z^2$, and are typically paired with asymptotically exact two-sided $100(1 - \alpha)\%$ confidence intervals (Theorem 9.16).
- The forecasted value $\hat{X}_{n+h}$ in an AR(2) model is a function of x_{n-1} and x_n and can be calculated by a recursive formula. It approaches $\hat{\mu} = \bar{x}$ as the time horizon $h \rightarrow \infty$. The associated prediction intervals have widths that increase as h increases and approach a limit as the time horizon $h \rightarrow \infty$ (Theorem 9.17).

The AR(1) time series model expresses the current value in the time series X_t as a constant times the previous value in the time series plus a random shock. The AR(2) time series model expresses the current value in the time series X_t as a linear combination of the previous two values in the time series plus a random shock. There is conceptually no difficulty extending this thinking to the AR(p) time series model in which the current value in the time series X_t is expressed as a linear combination of the previous p values in the time series plus a random shock. The AR(p) time series model is the subject of the next section.

9.1.3 The AR(p) Model

The order p autoregressive model, denoted by AR(p), is a straightforward generalization of the AR(2) model. The use of matrices in the derivations will be novel, along with the inability to easily visualize the stationary region as a function of the parameters. The AR(p) model is appropriate in instances in which the current value of the time series is a linear combination of the p previous values in the time series plus a random shock.

Definition 9.3 An order p autoregressive time series model, denoted by AR(p), for the time series $\{X_t\}$ is defined by

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + Z_t,$$

where $\phi_1, \phi_2, \dots, \phi_p$ are real-valued parameters and $\{Z_t\}$ is a time series of white noise:

$$Z_t \sim WN(0, \sigma_Z^2).$$

The $p + 1$ parameters that define an AR(p) model are the real-valued coefficients $\phi_1, \phi_2, \dots, \phi_p$, and the population variance of the white noise σ_Z^2 . The final coefficient, ϕ_p , must be nonzero. The AR(p) model can be written more compactly in terms of the backshift operator B as

$$\phi(B)X_t = Z_t,$$

where $\phi(B)$ is the order p characteristic polynomial

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p.$$

The $AR(p)$ model has the form of a multiple linear regression model with p independent variables and no intercept term. The current value X_t is being modeled as a linear combination of the p previous values of the time series, $X_{t-1}, X_{t-2}, \dots, X_{t-p}$, plus a white noise term Z_t that provides a random shock to the model. The parameters $\phi_1, \phi_2, \dots, \phi_p$ control the inclination of the regression line ($p = 1$), plane ($p = 2$), or hyperplane ($p > 2$). The σ_Z^2 parameter reflects the magnitude of the dispersion of the time series values about the regression plane.

Stationarity

Theorem 8.3 indicates that all $AR(p)$ models are invertible, but are stationary when all of the roots of $\phi(B)$ lie outside of the unit circle in the complex plane. Let $B_1, B_2, \dots, B_p$ denote the p solutions of $\phi(B) = 0$. For a stationary model, all of these roots will be real-valued or complex conjugate pairs that lie outside of the unit circle in the complex plane. Since $\phi(B_1) = \phi(B_2) = \dots = \phi(B_p) = 0$, the order p characteristic polynomial $\phi(B)$ can also be written in factored form as

$$\phi(B) = (1 - B_1^{-1}B)(1 - B_2^{-1}B) \dots (1 - B_p^{-1}B).$$

Unfortunately, except for the cases of $p = 1$ and $p = 2$, the region in the space of $(\phi_1, \phi_2, \dots, \phi_p)$ corresponding to a stationary model cannot be expressed in a simple mathematical form. The following example illustrates how to determine whether an $AR(4)$ model is stationary. This $AR(4)$ model will be used in the next five examples.

Example 9.22 Determine whether the $AR(4)$ model with characteristic polynomial

$$\phi(B) = 1 - \frac{21}{20}B - \frac{1}{20}B^2 + \frac{23}{40}B^3 - \frac{3}{10}B^4$$

is stationary.

The $AR(4)$ model is stationary if all of the roots of $\phi(B)$ lie outside the unit circle in the complex plane. The characteristic polynomial can be factored as

$$\phi(B) = -\frac{1}{40}(4B - 5)(3B + 4)(B^2 - 2B + 2).$$

Using the quadratic formula, the solutions of $\phi(B) = 0$ are

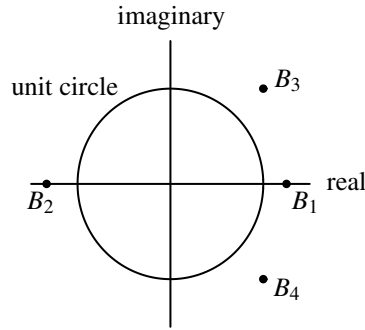
$$B_1 = \frac{5}{4} \quad B_2 = -\frac{4}{3} \quad B_3 = 1 + i \quad B_4 = 1 - i.$$

The first two roots are real-valued, and the other two roots are complex-valued conjugates. The four roots are plotted in Figure 9.26. Since all four roots lie outside of the unit circle in the complex plane, this $AR(4)$ model is stationary.

Duality

As was the case with the $AR(1)$ and $AR(2)$ time series models, a stationary $AR(p)$ time series model can be written as an $MA(\infty)$ time series model. This alternative representation can be useful for estimating standard errors of forecasted values. One way to frame the problem of writing an $AR(p)$ time series model as an $MA(\infty)$ time series model is to write the compact form of the $AR(p)$ model as

$$\phi(B)X_t = Z_t$$

Figure 9.26: Unit circle in the complex plane and the solutions of $\phi(B) = 0$.

and divide both sides by $\phi(B)$, which results in

$$X_t = \frac{Z_t}{\phi(B)}.$$

Therefore, the conversion from the $AR(p)$ form of the model to the $MA(\infty)$ form involves finding the coefficients $\theta_1, \theta_2, \dots$ such that

$$X_t = \frac{Z_t}{\phi(B)} = (1 + \theta_1 B + \theta_2 B^2 + \dots) Z_t.$$

The coefficients $\theta_1, \theta_2, \dots$ essentially correspond to finding the inverse of the $\phi(B)$ characteristic polynomial. Taking the expected value of both sides of this equation leads to the important result: $E[X_t] = 0$ for all values of t . As was the case of the $AR(2)$ time series model, the coefficients for the $MA(\infty)$ time series model are found by equating coefficients. This process will be illustrated in the next example for the $AR(4)$ model. Generalization to the $AR(p)$ model is straightforward.

Example 9.23 Calculate the first six coefficients of the $MA(\infty)$ model associated with the stationary $AR(4)$ model from Example 9.22 with characteristic polynomial

$$\phi(B) = 1 - \frac{21}{20}B - \frac{1}{20}B^2 + \frac{23}{40}B^3 - \frac{3}{10}B^4.$$

The $MA(\infty)$ model has the form

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \theta_3 Z_{t-3} + \theta_4 Z_{t-4} + \dots$$

The $AR(4)$ model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \phi_3 X_{t-3} + \phi_4 X_{t-4} + Z_t$$

can be written in terms of $\theta_1, \theta_2, \dots$ as

$$\begin{aligned} Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \theta_3 Z_{t-3} + \theta_4 Z_{t-4} + \dots = \\ \phi_1 (Z_{t-1} + \theta_1 Z_{t-2} + \theta_2 Z_{t-3} + \theta_3 Z_{t-4} + \dots) + \\ \phi_2 (Z_{t-2} + \theta_1 Z_{t-3} + \theta_2 Z_{t-4} + \dots) + \\ \phi_3 (Z_{t-3} + \theta_1 Z_{t-4} + \dots) + \\ \phi_4 (Z_{t-4} + \dots) + Z_t. \end{aligned}$$

Equating the coefficients of Z_{t-1} gives

$$\theta_1 = \phi_1.$$

Equating the coefficients of Z_{t-2} gives

$$\theta_2 = \phi_1 \theta_1 + \phi_2 = \phi_1^2 + \phi_2.$$

Equating the coefficients of Z_{t-3} and simplifying gives

$$\theta_3 = \phi_1^3 + 2\phi_1 \phi_2 + \phi_3.$$

Equating the coefficients of Z_{t-4} and simplifying gives

$$\theta_4 = \phi_1^4 + 3\phi_1^2 \phi_2 + \phi_2^2 + 2\phi_1 \phi_3 + \phi_4.$$

Equating the coefficients of Z_{t-k} gives the recursive equation

$$\theta_k = \phi_1 \theta_{k-1} + \phi_2 \theta_{k-2} + \phi_3 \theta_{k-3} + \phi_4 \theta_{k-4}$$

for $k = 5, 6, \dots$. The coefficients of the AR(4) model of interest are

$$\phi_1 = \frac{21}{20}, \quad \phi_2 = \frac{1}{20}, \quad \phi_3 = -\frac{23}{40}, \quad \phi_4 = \frac{3}{10}.$$

Using the equations derived here, the first six coefficients of the associated MA(∞) model as exact fractions are

$$\theta_1 = \frac{21}{20}, \theta_2 = \frac{461}{400}, \theta_3 = \frac{5501}{8000}, \theta_4 = \frac{76141}{160000}, \theta_5 = \frac{596381}{3200000}, \theta_6 = \frac{10870221}{64000000}.$$

The `ARMAtoMA` function in R can also compute these coefficients as follows.

```
ARMAtoMA(ar = c(21 / 20, 1 / 20, -23 / 40, 3 / 10), ma = 0, lag.max = 6)
```

This R command returns the decimal approximations of the exact fractions:

$$\theta_1 = 1.05, \theta_2 = 1.1525, \theta_3 = 0.6876, \theta_4 = 0.4759, \theta_5 = 0.1864, \theta_6 = 0.1698.$$

Population Autocorrelation Function

We now pivot to the derivation of the population autocovariance and autocorrelation functions. Assuming that the parameters $\phi_1, \phi_2, \dots, \phi_p$ are associated with a stationary model, the AR(p) model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + Z_t$$

can be multiplied by X_{t-k} to give

$$X_t X_{t-k} = \phi_1 X_{t-1} X_{t-k} + \phi_2 X_{t-2} X_{t-k} + \cdots + \phi_p X_{t-p} X_{t-k} + Z_t X_{t-k}.$$

Taking the expected value of both sides of this equation for $k = 0$ results in

$$\gamma(0) = \phi_1 \gamma(1) + \phi_2 \gamma(2) + \cdots + \phi_p \gamma(p) + \sigma_Z^2$$

and the recursive equation

$$\gamma(k) = \phi_1 \gamma(k-1) + \phi_2 \gamma(k-2) + \cdots + \phi_p \gamma(k-p)$$

for $k = 1, 2, \dots$ because Z_t has expected value zero and is independent of X_{t-k} . For $k = 1, 2, \dots, p$, the recursive equation can be written as the system of linear equations

$$\begin{aligned} \gamma(1) &= \phi_1 \gamma(0) + \phi_2 \gamma(1) + \phi_3 \gamma(2) + \cdots + \phi_p \gamma(p-1) \\ \gamma(2) &= \phi_1 \gamma(1) + \phi_2 \gamma(0) + \phi_3 \gamma(1) + \cdots + \phi_p \gamma(p-2) \\ \gamma(3) &= \phi_1 \gamma(2) + \phi_2 \gamma(1) + \phi_3 \gamma(0) + \cdots + \phi_p \gamma(p-3) \\ &\vdots \\ \gamma(p) &= \phi_1 \gamma(p-1) + \phi_2 \gamma(p-2) + \phi_3 \gamma(p-3) + \cdots + \phi_p \gamma(0), \end{aligned}$$

which relies on the symmetry of the population autocovariance function: $\gamma(-k) = \gamma(k)$. This linear system of p equations in $p+1$ unknowns can be written in matrix form as

$$\gamma = \Gamma \phi,$$

where

$$\gamma = \begin{bmatrix} \gamma(1) \\ \gamma(2) \\ \gamma(3) \\ \vdots \\ \gamma(p) \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \gamma(0) & \gamma(1) & \gamma(2) & \cdots & \gamma(p-1) \\ \gamma(1) & \gamma(0) & \gamma(1) & \cdots & \gamma(p-2) \\ \gamma(2) & \gamma(1) & \gamma(0) & \cdots & \gamma(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma(p-1) & \gamma(p-2) & \gamma(p-3) & \cdots & \gamma(0) \end{bmatrix}, \quad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_p \end{bmatrix}.$$

Given the values of the parameters $\phi_1, \phi_2, \dots, \phi_p$, and σ_Z^2 , this set of linear equations and

$$\gamma(0) = \phi_1 \gamma(1) + \phi_2 \gamma(2) + \cdots + \phi_p \gamma(p) + \sigma_Z^2,$$

one can compute the first $p+1$ population autocovariances $\gamma(0), \gamma(1), \dots, \gamma(p)$ by solving these linear equations. The recursion relationship can be used to compute subsequent autocovariances.

Example 9.24 Calculate the initial values of the population autocovariance function $\gamma(0), \gamma(1), \dots, \gamma(6)$ associated with the stationary AR(4) model with characteristic polynomial

$$\phi(B) = 1 - \frac{21}{20}B - \frac{1}{20}B^2 + \frac{23}{40}B^3 - \frac{3}{10}B^4$$

and white noise variance $\sigma_Z^2 = 1$.

The coefficients of the AR(4) model of interest are

$$\phi_1 = \frac{21}{20} \quad \phi_2 = \frac{1}{20} \quad \phi_3 = -\frac{23}{40} \quad \phi_4 = \frac{3}{10}.$$

To find the initial population autocovariances, solve the 5×5 set of linear equations

$$\begin{aligned}\gamma(0) &= \phi_1\gamma(1) + \phi_2\gamma(2) + \phi_3\gamma(3) + \phi_4\gamma(4) + \sigma_Z^2 \\ \gamma(1) &= \phi_1\gamma(0) + \phi_2\gamma(1) + \phi_3\gamma(2) + \phi_4\gamma(3) \\ \gamma(2) &= \phi_1\gamma(1) + \phi_2\gamma(0) + \phi_3\gamma(1) + \phi_4\gamma(2) \\ \gamma(3) &= \phi_1\gamma(2) + \phi_2\gamma(1) + \phi_3\gamma(0) + \phi_4\gamma(1) \\ \gamma(4) &= \phi_1\gamma(3) + \phi_2\gamma(2) + \phi_3\gamma(1) + \phi_4\gamma(0)\end{aligned}$$

for $\gamma(0), \gamma(1), \gamma(2), \gamma(3), \gamma(4)$. The recursive equation

$$\gamma(k) = \phi_1\gamma(k-1) + \phi_2\gamma(k-2) + \phi_3\gamma(k-3) + \phi_4\gamma(k-4)$$

can be used to calculate $\gamma(k)$ values for $k = 5, 6, \dots$. The initial population autocovariance values are

$$\begin{aligned}\gamma(0) &= \frac{3520}{819} \cong 4.298, \gamma(1) = \frac{2960}{819} \cong 3.614, \gamma(2) = \frac{2260}{819} \cong 2.759, \gamma(3) = \frac{1385}{819} \cong 1.691, \\ \gamma(4) &= \frac{3685}{3276} \cong 1.125, \gamma(5) = \frac{10001}{13104} \cong 0.763, \gamma(6) = \frac{186881}{262080} \cong 0.713.\end{aligned}$$

These population autocovariances can be used to calculate the associated population autocorrelations by dividing each of them by $\gamma(0)$.

Dividing both sides of the recursive equation for calculating population autocovariance by $\gamma(0) = V[X_t]$ gives the recursive equation

$$\rho(k) = \phi_1\rho(k-1) + \phi_2\rho(k-2) + \dots + \phi_p\rho(k-p)$$

for $k = 1, 2, \dots$. Exploiting the symmetry of the $\rho(k)$ function, the first p of these equations are

$$\begin{aligned}\rho(1) &= \phi_1\rho(0) + \phi_2\rho(1) + \phi_3\rho(2) + \dots + \phi_p\rho(p-1) \\ \rho(2) &= \phi_1\rho(1) + \phi_2\rho(0) + \phi_3\rho(1) + \dots + \phi_p\rho(p-2) \\ \rho(3) &= \phi_1\rho(2) + \phi_2\rho(1) + \phi_3\rho(0) + \dots + \phi_p\rho(p-3) \\ &\vdots \\ \rho(p) &= \phi_1\rho(p-1) + \phi_2\rho(p-2) + \phi_3\rho(p-3) + \dots + \phi_p\rho(0).\end{aligned}$$

Since $\rho(0) = 1$, this linear system of p equations in the p unknowns $\rho(1), \rho(2), \dots, \rho(p)$ can be written in matrix form as

$$\rho = P\phi,$$

where

$$\rho = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \rho(3) \\ \vdots \\ \rho(p) \end{bmatrix}, \quad P = \begin{bmatrix} 1 & \rho(1) & \rho(2) & \dots & \rho(p-1) \\ \rho(1) & 1 & \rho(1) & \dots & \rho(p-2) \\ \rho(2) & \rho(1) & 1 & \dots & \rho(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \rho(p-3) & \dots & 1 \end{bmatrix}, \quad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_p \end{bmatrix}.$$

Given the values of the parameters $\phi_1, \phi_2, \dots, \phi_p$, these linear equations can be solved for the initial p population autocorrelation function values $\rho(1), \rho(2), \dots, \rho(p)$, and the recursive function can be used to calculate subsequent values of the population autocorrelation values.

As was the case with the AR(2) time series model, (a) the real roots of $\phi(B)$ correspond to contributions to the population autocorrelation function which are mixtures of damped exponential terms, and (b) the complex conjugate roots of $\phi(B)$ correspond to contributions to the population autocorrelation function which are damped sinusoidal terms.

These equations bear some practical use in that the first p sample autocorrelation function values, $r_1, r_2, \dots, r_p$, can be calculated from an observed time series and used as approximations for $\rho(1), \rho(2), \dots, \rho(p)$, yielding estimators for $\phi_1, \phi_2, \dots, \phi_p$. These estimates are known as the *Yule–Walker estimators*. These can in turn be used as initial estimates for finding point estimates for $\phi_1, \phi_2, \dots, \phi_p$ by, for example, least squares or maximum likelihood estimation, should numerical methods be required.

The results concerning the calculation of the population autocovariance function $\gamma(k)$ and the population autocorrelation function $\rho(k)$ are summarized below.

Theorem 9.18 The population autocovariance function for a stationary AR(p) time series model is calculated by

$$\gamma(k) = \phi_1 \gamma(k-1) + \phi_2 \gamma(k-2) + \dots + \phi_p \gamma(k-p)$$

for $k = 1, 2, \dots$. The first p of these equations can be written in matrix form as

$$\gamma = \Gamma \phi.$$

The population variance of X_t is

$$V[X_t] = \gamma(0) = \phi_1 \gamma(1) + \phi_2 \gamma(2) + \dots + \phi_p \gamma(p) + \sigma_Z^2.$$

The population autocorrelation function for a stationary AR(p) time series model is calculated by

$$\rho(k) = \phi_1 \rho(k-1) + \phi_2 \rho(k-2) + \dots + \phi_p \rho(k-p)$$

for $k = 1, 2, \dots$. The first p of these equations can be written in matrix form as

$$\rho = P\phi.$$

The system of linear equations in Theorem 9.18, whether written in terms of $\gamma(k)$ or $\rho(k)$ as $\gamma = \Gamma\phi$ or $\rho = P\phi$, is known in time series analysis as the *Yule–Walker equations*.

Population Partial Autocorrelation Function

We now determine the population partial autocorrelation function for an AR(p) model. Using Definition 7.4, the initial population partial autocorrelation values are

$$\rho^*(0) = 1, \quad \rho^*(1) = \rho(1), \quad \rho^*(2) = \frac{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & \rho(2) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & 1 \end{vmatrix}}, \quad \rho^*(3) = \frac{\begin{vmatrix} 1 & \rho(1) & \rho(2) \\ \rho(1) & 1 & \rho(3) \\ \rho(2) & \rho(1) & \rho(3) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) & \rho(2) \\ \rho(1) & 1 & \rho(1) \\ \rho(2) & \rho(1) & 1 \end{vmatrix}},$$

etc. One distinctive characteristic of the $AR(p)$ population partial autocorrelation function is that it cuts off after lag p . To see why this is the case, consider the first p columns of the matrix in the numerator of $\rho^*(k)$ for $k > p$:

$$\begin{bmatrix} 1 \\ \rho(1) \\ \rho(2) \\ \vdots \\ \rho(k-1) \end{bmatrix}, \begin{bmatrix} \rho(1) \\ 1 \\ \rho(1) \\ \vdots \\ \rho(k-2) \end{bmatrix}, \dots, \begin{bmatrix} \rho(p-1) \\ \rho(p-2) \\ \rho(p-3) \\ \vdots \\ \rho(k-p) \end{bmatrix}.$$

Using Theorem 9.18, the last column of the matrix in the numerator of $\rho^*(k)$ is

$$\begin{bmatrix} \phi_1 + \phi_2\rho(1) + \phi_3\rho(2) + \dots + \phi_p\rho(p-1) \\ \phi_1\rho(1) + \phi_2 + \phi_3\rho(1) + \dots + \phi_p\rho(p-2) \\ \phi_1\rho(2) + \phi_2\rho(1) + \phi_3 + \dots + \phi_p\rho(p-3) \\ \vdots \\ \phi_1\rho(k-1) + \phi_2\rho(k-2) + \phi_3\rho(k-3) + \dots + \phi_p\rho(k-p) \end{bmatrix}.$$

The last column of the matrix in the numerator of $\rho^*(k)$ is a linear combination of the first p columns with coefficients $\phi_1, \phi_2, \dots, \phi_p$. Thus, the matrix in the numerator of the calculation of $\rho^*(k)$ is singular, which means that its determinant is zero. So $\rho^*(k) = 0$ for $k = p+1, p+2, \dots$ for an $AR(p)$ time series model. This constitutes a proof of the following result.

Theorem 9.19 The population partial autocorrelation function for a stationary $AR(p)$ time series model cuts off after lag p .

A graph of the *sample* partial autocorrelation function r_k^* for the first few values of k , should also cut off after lag p if the $AR(p)$ model is appropriate. This sample partial autocorrelation function shape is easier to recognize than the associated sample autocorrelation function shape because cutting off is typically easier to recognize than tailing off in the presence of random sampling variability.

There is a second interpretation of the partial autocorrelation function that ties it more closely to determining the order of the autoregressive portion of the model. The partial autocorrelation at lag k is the value of the final coefficient ϕ_k in an autoregressive model of order k . This coefficient measures the excess correlation at lag k which is not accounted for by an autoregressive model of order $k-1$. It is for this reason that many authors use the notation ϕ_{kk} for the population lag k partial autocorrelation.

The population autocorrelation function and the population partial autocorrelation functions can be calculated using the formulas given here, but can also be calculated using the R `ARMAacf` function, as illustrated in the next example.

Example 9.25 Calculate and plot the values of the population autocorrelation function and the population partial autocorrelation function associated with the $AR(4)$ model with characteristic polynomial

$$\phi(B) = 1 - \frac{21}{20}B - \frac{1}{20}B^2 + \frac{23}{40}B^3 - \frac{3}{10}B^4.$$

The coefficients of the $AR(4)$ model of interest are

$$\phi_1 = \frac{21}{20} \quad \phi_2 = \frac{1}{20} \quad \phi_3 = -\frac{23}{40} \quad \phi_4 = \frac{3}{10}.$$

The matrix equation $\rho = P\phi$ from Theorem 9.18 could be solved for the initial values of $\rho(k)$. Alternatively, the values of $\gamma(k)$ calculated in the previous example could be divided by $\gamma(0)$ to arrive at the population autocorrelation function values. The first two such autocorrelation function values, for example, are

$$\rho(1) = \frac{\gamma(1)}{\gamma(0)} = \frac{2960/819}{3520/819} = \frac{2960}{3520} \cong 0.8409,$$

$$\rho(2) = \frac{\gamma(2)}{\gamma(0)} = \frac{2260/819}{3520/819} = \frac{2260}{3520} \cong 0.6420,$$

The R function `ARMAacf` can also be used to calculate the population autocorrelation function values for the first 15 lags, as illustrated below. The `ar` argument is a vector containing the coefficients ϕ_1, ϕ_2, ϕ_3 , and ϕ_4 , and the `ma` argument is set to zero because there are no moving average terms.

```
ARMAacf(ar = c(21 / 20, 1 / 20, -23 / 40, 3 / 10), ma = 0, 15)
```

The population partial autocorrelation function values can be computed by taking the ratios of the determinants from Definition 7.4. Alternatively, the `pacf` argument to the `ARMAacf` function can be set to `TRUE` to compute the values of $\rho^*(k)$ for the first 15 lags.

```
ARMAacf(ar = c(21 / 20, 1 / 20, -23 / 40, 3 / 10), ma = 0, 15, pacf = TRUE)
```

Table 9.9 contains the numeric values of the first seven values of $\rho(k)$ and $\rho^*(k)$. Figure 9.27 contains a plot of $\rho(k)$ and $\rho^*(k)$ for the first 15 lags. The population autocorrelation function includes the effects of mixtures of damped exponential terms (associated with the two real roots $B_1 = 5/4$ and $B_2 = -4/3$ of $\phi(B) = 0$ computed in Example 9.22) and damped sinusoidal terms (associated with the two complex roots $B_3 = 1 + i$ and $B_4 = 1 - i$ of $\phi(B) = 0$ computed in Example 9.22). As expected, the population partial autocorrelation function cuts off after lag 4.

k	1	2	3	4	5	6	7
$\rho(k)$	0.8409	0.6420	0.3935	0.2617	0.1776	0.1659	0.1506
$\rho^*(k)$	0.8409	-0.2222	-0.2857	0.3000	0	0	0

Table 9.9: The first seven values of $\rho(k)$ and $\rho^*(k)$ for an AR(4) time series model.

The Shifted AR(p) Model

The standard AR(p) model from Definition 9.3 is not of much practical use because most real-world time series are not centered around zero. Adding a shift parameter μ overcomes this shortcoming. Since population variance and covariance are unaffected by a shift, the associated population autocorrelation and partial autocorrelation functions remain the same as those given in Theorems 9.18 and 9.19.

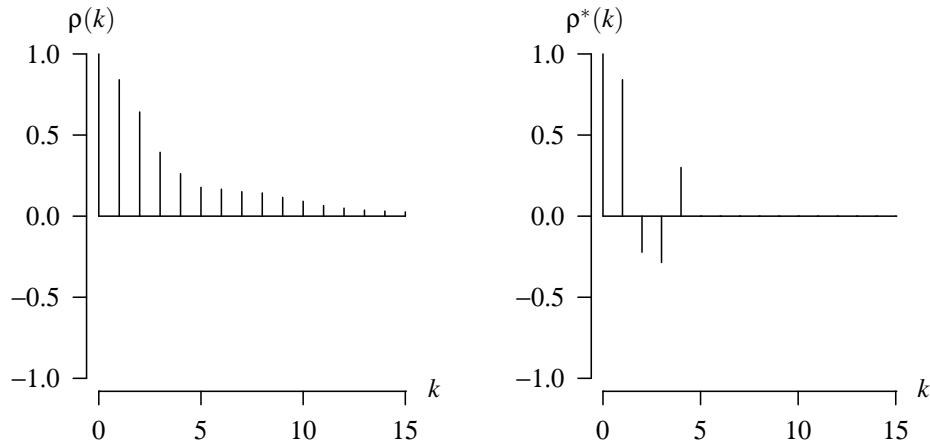


Figure 9.27: The first 15 values of $\rho(k)$ and $\rho^*(k)$ for an AR(4) time series model.

Theorem 9.20 A shifted order p autoregressive model for the time series $\{X_t\}$ is defined by

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + \cdots + \phi_p (X_{t-p} - \mu) + Z_t,$$

where $\phi_1, \phi_2, \dots, \phi_p, \mu$, and $\sigma_Z^2 > 0$ are real-valued parameters, and $\{Z_t\}$ is a time series of white noise. This model is stationary when all of the roots of the characteristic equation $\phi(B) = 0$ fall outside of the unit circle in the complex plane. The expected value of X_t is $E[X_t] = \mu$. The population autocorrelation function can be calculated using the recursive equations in Theorem 9.18. The population partial autocorrelation function can be calculated using the defining formulas in Definition 7.4.

The shifted AR(p) model can be written in terms of the backshift operator B as

$$\phi(B)(X_t - \mu) = Z_t,$$

where $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$. The practical problem of fitting a shifted AR(p) model to an observed time series of n values $x_1, x_2, \dots, x_n$ will be illustrated later in this subsection.

Simulation

An AR(p) time series can be simulated by appealing to the defining formula for the AR(p) model. Iteratively applying the defining formula for a standard AR(p) model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + Z_t$$

from Definition 9.3 results in the simulated values $X_1, X_2, \dots, X_n$. The difficult aspect of devising a simulation algorithm is generating the first p simulated values, $X_1, X_2, \dots, X_p$. For simplicity, assume that the white noise terms are Gaussian white noise terms. There are two approaches to overcome this initialization problem. The first approach generates $X_1, X_2, \dots, X_p$ from a multivariate normal distribution with population mean p -vector $\mathbf{0} = (0, 0, \dots, 0)'$ and $p \times p$ variance-covariance

matrix

$$\Gamma = \begin{bmatrix} \gamma(0) & \gamma(1) & \gamma(2) & \cdots & \gamma(p-1) \\ \gamma(1) & \gamma(0) & \gamma(1) & \cdots & \gamma(p-2) \\ \gamma(2) & \gamma(1) & \gamma(0) & \cdots & \gamma(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma(p-1) & \gamma(p-2) & \gamma(p-3) & \cdots & \gamma(0) \end{bmatrix},$$

which was defined in Theorem 9.18. The algorithm given below generates initial time series observations $X_1, X_2, \dots, X_p$ as indicated above, and then uses an additional $n - p$ Gaussian white noise terms $Z_{p+1}, Z_{p+2}, \dots, Z_n$ to generate the remaining time series values $X_{p+1}, X_{p+2}, \dots, X_n$ using the AR(p) defining formula from Definition 9.3. Indentation denotes nesting in the algorithm.

```

generate  $(X_1, X_2, \dots, X_p) \sim N(\mathbf{0}, \Gamma)$ 
 $t \leftarrow p$ 
while  $(t < n)$ 
   $t \leftarrow t + 1$ 
  generate  $Z_t \sim N(0, \sigma_Z^2)$ 
   $X_t \leftarrow \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + Z_t$ 

```

The $(p+2)$ -parameter shifted AR(p) time series model which includes a population mean parameter μ can be simulated by simply adding μ to each time series observation generated by this algorithm. The next example implements this algorithm in R.

Example 9.26 Generate a realization of $n = 100$ observations from the stationary AR(4) time series model with

$$\phi_1 = \frac{21}{20} \quad \phi_2 = \frac{1}{20} \quad \phi_3 = -\frac{23}{40} \quad \phi_4 = \frac{3}{10}$$

and Gaussian white noise error terms with $\sigma_Z^2 = 1$.

This model is stationary (see Example 9.22). The population autocorrelation function $\rho(k)$ and the population partial autocorrelation function $\rho^*(k)$ are displayed in Figure 9.27; we expect similar shaped functions r_k and r_k^* from our simulated values. The first statement in the R code below uses the `set.seed` function to establish the random number seed. The second statement sets $p = 4$, corresponding to an AR(4) model. The third statement sets the vector `phi` to the AR(4) coefficients $\phi_1 = 21/20$, $\phi_2 = 1/20$, $\phi_3 = -23/40$, and $\phi_4 = 3/10$. The fourth statement places the initial population autocovariance values from Example 9.24, namely $\gamma(0) = 3520/819$, $\gamma(1) = 2960/819$, $\gamma(2) = 2260/819$, and $\gamma(3) = 1385/819$, into the vector `gam`. The subsequent nested `for` loops place these population autocovariance values in the 4×4 variance-covariance matrix `GAMMA`. The next statement sets the standard deviation of the Gaussian white noise to $\sigma_Z = 1$. The next statement sets the number of simulated values to $n = 100$. The next statement defines the vector `x` of length $n = 100$ to hold the simulated time series values. The next statement uses the `mvrnorm` function from the `MASS` package to generate the first four simulated time series observations X_1, X_2, X_3, X_4 from the appropriate multivariate normal distribution. Finally, the `for` loop iterates through the defining formula for the AR(4) model generating the remaining observations $X_5, X_6, \dots, X_{100}$.

```

set.seed(9)
p      = 4

```

```

phi    = c(21 / 20, 1 / 20, -23 / 40, 3 / 10)
gam    = c(3520 / 819, 2960 / 819, 2260 / 819, 1385 / 819)
GAMMA  = matrix(0, p, p)
for (i in 1:p) for (j in 1:p) GAMMA[i, j] = gam[abs(i - j) + 1]
sigz   = 1
n      = 100
x      = numeric(n)
x[1:p] = MASS::mvrnorm(1, mu = rep(0, p), Sigma = GAMMA)
for (t in (p + 1):n) x[t] = sum(phi * x[(t - 1):(t - p)]) +
                           rnorm(1, 0, sigz)

```

Using the `plot.ts` function to make a plot of the time series contained in `x`, the `acf` function to plot the associated correlogram, the `pacf` function to plot the associated sample partial autocorrelation function, and the `layout` function to arrange the graphs as in Example 7.24, the resulting trio of graphs are displayed in Figure 9.28. The sample partial autocorrelation function has four statisti-

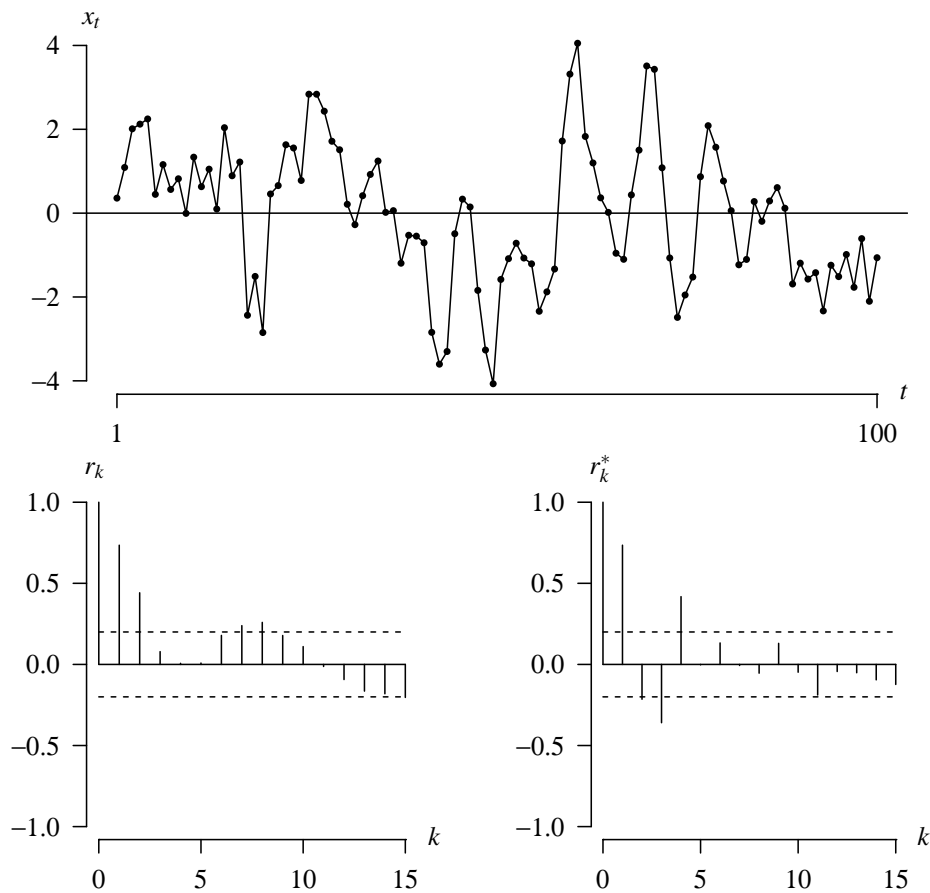


Figure 9.28: Time series plot, r_k , and r_k^* for $n = 100$ simulated values from an AR(4) model.

cally significant spikes at lags 1, 2, 3, and 4 which is consistent with an AR(4) model. The spikes cut off after lag 4 as expected from the population counterparts in Figure 9.27. The approximate 95% confidence intervals indicated by the dashed lines show that the values of the sample partial autocorrelation function do not significantly differ from zero at lags beyond lag 4. The sample autocorrelation function displays a mixture of damped exponential terms damped sinusoidal terms as expected, with statistically significant autocorrelations at the first two lags: $r_1^* = 0.7351$ and $r_2^* = 0.4417$. The time series plot shows that observations tend to linger on one side of the population mean (indicated by a horizontal line at $\mu = 0$), which is consistent with the two initial statistically significant positive spikes in the sample autocorrelation function.

We recommend running the simulation code from the previous example several dozen times in a loop and viewing the associated plots of x_t , r_k , and r_k^* in search of patterns. This will allow you to see how various realizations of this simulated AR(4) time series model vary from one realization to the next. So when you then view a *single* realization of a real-life time series, you will have a sense of how far these plots might deviate from their expected patterns.

There is a second way to overcome the initialization problem in simulating observations from an AR(p) time series. This second technique starts the time series with p initial arbitrary values, and then allows the time series to “warm up” or “burn in” for several time periods before producing the first observation X_1 . Reasonable p initial arbitrary values for the standard AR(p) model are 0; reasonable p initial arbitrary values for the shifted AR(p) model are μ . This approach can be implemented in R with the `filter` function with “recursive” as the `method` argument. The code below generates $n = 100$ values in the AR(4) time series model from the previous example using a warm-up period of 50 observations.

```
phi = c(21 / 20, 1 / 20, -23 / 40, 3 / 10)
z   = rnorm(150)
x   = filter(z, filter = phi, method = "recursive")
x   = x[51:150]
```

This is also the approach taken by the built-in R function named `arima.sim`, which simulates a realization of a time series. Using the `arima.sim` function means that $n = 100$ observations from the AR(4) time series model from the previous example can be simulated using a single command, using a warm-up period of 50 observations.

```
x = arima.sim(model = list(ar = c(21 / 20, 1 / 20, -23 / 40, 3 / 10)),
              n = 100, sd = 1, n.start = 50)
```

The remaining topics associated with the AR(p) time series model are statistical in nature: parameter estimation, model assessment, model selection, and forecasting. We begin with parameter estimation.

Parameter Estimation

The $p + 2$ parameters to estimate in a shifted AR(p) time series model are $\phi_1, \phi_2, \dots, \phi_p, \mu, \sigma_Z^2$. There are three techniques for estimating these parameters considered here: method of moments, least squares, and maximum likelihood estimation. These techniques were introduced in Section 8.2.1. These three techniques are outlined in the following paragraphs.

Approach 1: Method of moments. In the case of estimating the $p + 2$ parameters in the shifted AR(p) time series model by the method of moments, we match the population and sample first-order moments, second-order moments, lag 1 autocorrelation, lag 2 autocorrelation, $\dots$, lag p autocorrelation. Placing the population moments on the left-hand side of the equation and the associated sample

moments on the right-hand side of the equation results in $(p+2)$ equations in $(p+2)$ unknowns:

$$\begin{aligned} E[X_t] &= \frac{1}{n} \sum_{t=1}^n X_t \\ E[X_t^2] &= \frac{1}{n} \sum_{t=1}^n X_t^2 \\ \rho(1) &= r_1 \\ \rho(2) &= r_2 \\ &\vdots = \vdots \\ \rho(p) &= r_p. \end{aligned}$$

Since $E[X_t] = \mu$ for a stationary shifted AR(p) time series model, the first equation gives the method of moments estimator $\hat{\mu} = \bar{X}$. Recall from Theorem 9.18 that the relationship between $\phi_1, \phi_2, \dots, \phi_p$ and $\rho(1), \rho(2), \dots, \rho(p)$ is given by the matrix equation

$$\rho = P\phi,$$

where

$$\rho = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \rho(3) \\ \vdots \\ \rho(p) \end{bmatrix}, \quad P = \begin{bmatrix} 1 & \rho(1) & \rho(2) & \cdots & \rho(p-1) \\ \rho(1) & 1 & \rho(1) & \cdots & \rho(p-2) \\ \rho(2) & \rho(1) & 1 & \cdots & \rho(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \rho(p-3) & \cdots & 1 \end{bmatrix}, \quad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_p \end{bmatrix}.$$

Satisfying the method of moments criteria, the lag k population autocorrelation $\rho(k)$ can be replaced with its statistical analog r_k , for $k = 1, 2, \dots, p$. The resulting matrix equation is

$$r = R\phi,$$

where

$$r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_p \end{bmatrix}, \quad R = \begin{bmatrix} 1 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & 1 & r_1 & \cdots & r_{p-2} \\ r_2 & r_1 & 1 & \cdots & r_{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & 1 \end{bmatrix}, \quad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_p \end{bmatrix}.$$

This matrix equation can be solved for the method of moments estimators as

$$\hat{\phi} = R^{-1}r.$$

These are known as the *Yule–Walker estimators* because of their relationship to the Yule–Walker equations. Finally, the remaining parameter to estimate is the population variance of the white noise σ_Z^2 . From Theorem 9.18,

$$\sigma_Z^2 = \gamma(0) - \phi_1\gamma(1) - \phi_2\gamma(2) - \cdots - \phi_p\gamma(p).$$

Multiplying and dividing the right-hand side of this equation by $\gamma(0)$ gives

$$\sigma_Z^2 = \gamma(0) [1 - \phi_1 \rho(1) - \phi_2 \rho(2) - \cdots - \phi_p \rho(p)].$$

Replacing these elements by their method of moments estimators gives

$$\hat{\sigma}_Z^2 = c_0 [1 - \hat{\phi}_1 r_1 - \hat{\phi}_2 r_2 - \cdots - \hat{\phi}_p r_p],$$

which can be expressed in matrix form as

$$\hat{\sigma}_Z^2 = c_0 (1 - r' \hat{\phi}).$$

Since the formula for these estimators does not require any iterative methods, the method of moments estimators are often used as initial parameter estimates for the least squares estimators and the maximum likelihood estimators, which do require iterative methods. These point estimators for the parameters in a shifted AR(p) model are summarized below.

Theorem 9.21 The method of moments estimators of the parameters in a shifted AR(p) model are

$$\hat{\mu} = \bar{X} \qquad \hat{\phi} = R^{-1} r \qquad \hat{\sigma}_Z^2 = c_0 (1 - r' \hat{\phi}).$$

Example 9.27 We now revisit the modeling of the built-in R time series `LakeHuron` from Example 9.14 consisting of $n = 98$ monthly mean levels (in feet) of the lake level of Lake Huron from 1875–1972. An AR(2) time series model was fit to this time series using the method of moments in Example 9.15. The fitted AR(2) model was deemed to be a reasonable fit via the goodness-of-fit tests in Example 9.19. Calculate the method of moments parameter estimates for the overfitted AR(3) model.

Since estimating the parameters involves just a matrix inverse and a matrix multiplication, these estimators are easily computed in an R function. The user-written `YuleWalker` function given below has the time series observations in the vector `x` and the order of the AR(p) time series model `p` as arguments. It uses the built-in `acf` function to compute $r_1, r_2, \dots, r_p$ and the `solve` function to compute the inverse of the R matrix. The R code below calculates and prints the point estimates of the parameters $\mu, \phi_1, \phi_2, \phi_3$, and σ_Z^2 parameters for the AR(3) time series model using the method of moments estimators given in Theorem 9.21.

```
YuleWalker = function(x, p) {
  muhat = mean(x)
  r = acf(x, plot = FALSE, lag.max = p)$acf
  R = matrix(1, p, p)
  for (i in 1:p) for (j in 1:p) R[i, j] = r[abs(i - j) + 1]
  r = r[2:(p + 1)]
  phihat = solve(R) %*% r
  sig2hat = mean((x - muhat) ^ 2) * (1 - sum(r * phihat))
  c(muhat, phihat, sig2hat)
}
YuleWalker(LakeHuron, 3)
```

The method of moments point estimates for the unknown parameters computed by this code are

$$\hat{\mu} = 579.00, \quad \hat{\phi}_1 = 1.0887, \quad \hat{\phi}_2 = -0.40454, \quad \hat{\phi}_3 = 0.13075, \quad \hat{\sigma}_Z^2 = 0.48358.$$

Alternatively, some keystrokes can be saved by using the built-in `ar` function to estimate the parameters in the AR(3) time series model, as shown below. The results are identical except the estimate of the population variance of the white noise differs slightly because of differing assumptions made within the `ar` function.

```
fit = ar(LakeHuron, order.max = 3, aic = FALSE, method = "yw")
fit$x.mean
fit$ar
fit$var.pred
```

Approach 2: Least squares. Consider the shifted stationary AR(p) model

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + \cdots + \phi_p (X_{t-p} - \mu) + Z_t.$$

For least squares estimation, we first establish the sum of squares S as a function of the parameters $\mu, \phi_1, \phi_2, \dots, \phi_p$. We leave the optimization to the R `ar` function in order to calculate the least squares estimators of $\mu, \phi_1, \phi_2, \dots, \phi_p$. Once these least squares estimators have been determined, the population variance of the white noise σ_Z^2 will be estimated.

The sum of squared errors is

$$S = \sum_{t=p+1}^n Z_t^2 = \sum_{t=p+1}^n [X_t - \mu - \phi_1 (X_{t-1} - \mu) - \phi_2 (X_{t-2} - \mu) - \cdots - \phi_p (X_{t-p} - \mu)]^2.$$

If this derivation were being done by hand, we would now calculate the partial derivatives of S with respect to the unknown parameters $\mu, \phi_1, \phi_2, \dots, \phi_p$, equate them to zero and solve. As was the case with the AR(1) and AR(2) models, there is no closed-form solution, so numerical methods are required to calculate the parameter estimates. In the example that follows, we will use the `ar` function in R to determine the least squares parameter estimates that minimize S .

The last parameter to estimate is the population variance of the white noise σ_Z^2 . The same estimator as the method of moments will be used:

$$\hat{\sigma}_Z^2 = c_0 (1 - r' \hat{\phi}).$$

Least squares estimation for a shifted AR(p) time series model is summarized below.

Theorem 9.22 The least squares estimators of the parameters in a shifted AR(p) time series model are the $\hat{\mu}, \hat{\phi}_1, \hat{\phi}_2, \dots, \hat{\phi}_p$ values that minimize

$$S = \sum_{t=p+1}^n Z_t^2 = \sum_{t=p+1}^n [X_t - \mu - \phi_1 (X_{t-1} - \mu) - \phi_2 (X_{t-2} - \mu) - \cdots - \phi_p (X_{t-p} - \mu)]^2$$

and the population variance of the white noise is estimated by

$$\hat{\sigma}_Z^2 = c_0 (1 - r' \hat{\phi}).$$

We now use numerical methods to find the least squares estimates for the unknown parameters in the AR(p) time series model for the Lake Huron time series from Example 9.14.

Example 9.28 Find the least squares estimates of μ , ϕ_1 , ϕ_2 , $\dots$, ϕ_p , and σ_Z^2 from the $AR(p)$ time series model for the time series of $n = 98$ Lake Huron lake level observations from Example 9.14, for $p = 1, 2, 3, 4$. Plot the sum of squares associated with the least squares estimates as a function of p .

The R code below uses a `for` loop to iterate over the various values of p . It uses the `ar` function with the `method` argument set to "ols" (for ordinary least squares) to calculate the least squares estimates of the unknown parameters. It uses a nested `for` loop to calculate the sum of squares at the values of the point estimates.

```
x = LakeHuron
n = length(x)
for (p in 1:4) {
  fit      = ar(x, order.max = p, aic = FALSE, method = "ols")
  muhat    = fit$x.mean
  phihat   = fit$ar
  sig2hat  = fit$var.pred
  S        = 0
  for (t in (p + 1):n) {
    S = S + (x[t] - muhat - sum(phihat * (x[(t - 1):(t - p)] - muhat))) ^ 2
  }
  print(c(muhat, phihat, sig2hat, S))
}
```

The point estimates for the unknown parameters and the sums of squares at the point estimates that are computed by this code are given in Table 9.10. Notice that the least squares point estimators for the $AR(3)$ model are close to the method of moments point estimators for the $AR(3)$ model calculated in Example 9.27. The graph in Figure 9.29 shows the sum of squares as a function of the order of the autoregressive model p . The sum of squares shows a “law of diminishing returns” as p increases. There is a large decrease in the sum of squares on the transition from $p = 1$ term to $p = 2$ terms. Beyond $p = 2$, however, the decreases are substantially smaller. This pattern is consistent with the $q = 0$ column from Table 9.7 in Example 9.20, which indicated that the AIC statistic was minimized for $p = 2$, which corresponds to a shifted $AR(2)$ time series model.

p	$\hat{\mu}$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\phi}_3$	$\hat{\phi}_4$	$\hat{\sigma}_Z^2$	S
1	579.00	0.8364				0.5090	49.38
2	579.00	1.0217	-0.2376			0.4540	43.64
3	579.00	1.0719	-0.3653	0.1088		0.4488	42.66
4	579.00	1.0738	-0.3739	0.0569	0.0625	0.4475	42.12

Table 9.10: Least squares parameter estimates and sums of squares for $AR(p)$ models.

Approach 3: Maximum likelihood estimation. The procedure for determining the maximum likelihood estimators for the unknown parameters in a shifted $AR(p)$ time series model follows along the same lines as in the $AR(1)$ and $AR(2)$ time series models from the previous subsections. Once again, to use maximum likelihood estimation, we must assume that the random shocks from the

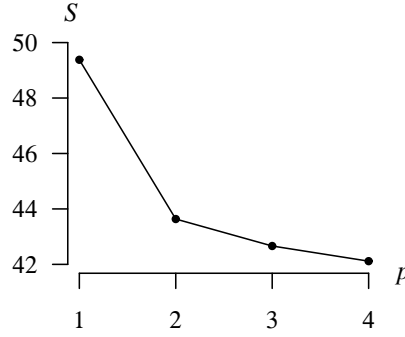


Figure 9.29: Sum of squares as a function of p for $AR(p)$ models.

white noise are Gaussian white noise, with associated probability density function

$$f_{Z_t}(z_t) = \frac{1}{\sqrt{2\pi\sigma_Z^2}} e^{-z_t^2/(2\sigma_Z^2)} \quad -\infty < z_t < \infty,$$

for $t = 1, 2, \dots, n$. Determining the likelihood function, which is the joint probability density function of the observed values in the time series $X_1, X_2, \dots, X_n$, involves finding

$$L(\mu, \phi_1, \phi_2, \dots, \phi_p, \sigma_Z^2) = f(x_1, x_2, \dots, x_n),$$

where the $x_1, x_2, \dots, x_n$ arguments on L and the $\mu, \phi_1, \phi_2, \dots, \phi_p$, and σ_Z^2 arguments on f have been dropped for brevity and $n > p$. As before, it is not possible to simply multiply the marginal probability density functions because the values in the $AR(p)$ time series model are correlated. As in the case of the $AR(1)$ and $AR(2)$ models, we use the transformation technique to find the conditional joint probability density function of $X_{p+1}, X_{p+2}, \dots, X_n$ conditioned on $X_1 = x_1, X_2 = x_2, \dots, X_p = x_p$, which is denoted by

$$f_{X_{p+1}, X_{p+2}, \dots, X_n | X_1, X_2, \dots, X_p}(x_{p+1}, x_{p+2}, \dots, x_n | X_1 = x_1, X_2 = x_2, \dots, X_p = x_p)$$

for $(x_{p+1}, x_{p+2}, \dots, x_n) \in \mathcal{R}^{n-p}$. This conditional joint probability density function is multiplied by the marginal joint probability density function of $X_1, X_2, \dots, X_p$ (which has the p -dimensional multivariate normal distribution) resulting in a joint probability density function of $X_1, X_2, \dots, X_n$:

$$\begin{aligned} f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = \\ f_{X_{p+1}, X_{p+2}, \dots, X_n | X_1, X_2, \dots, X_p}(x_{p+1}, x_{p+2}, \dots, x_n | X_1 = x_1, X_2 = x_2, \dots, X_p = x_p) \times \\ f_{X_1, X_2, \dots, X_p}(x_1, x_2, \dots, x_p) \end{aligned}$$

for $(x_1, x_2, \dots, x_n) \in \mathcal{R}^n$. This function serves as the likelihood function, which should be maximized with respect to the unknown parameters $\mu, \phi_1, \phi_2, \dots, \phi_p$, and σ_Z^2 . We leave the maximization to the `ar` and `arima` functions in R when determining the maximum likelihood estimates for the parameters for a particular time series to be fitted to the shifted $AR(p)$ time series model.

In addition to point estimators for the parameters, we are also interested in confidence intervals that capture the precision of the point estimators. The population variance of the vector of parameter estimators $\hat{\phi} = (\hat{\phi}_1, \hat{\phi}_2, \dots, \hat{\phi}_p)'$ is given by the variance–covariance matrix

$$V[\hat{\phi}] = \frac{1}{n} (1 - \rho' \phi) P^{-1}.$$

Since the maximum likelihood estimators for $\phi_1, \phi_2, \dots, \phi_p$ are asymptotically unbiased and normally distributed under certain regularity conditions,

$$\hat{\phi} \xrightarrow{D} N\left(\phi, \frac{1}{n} (1 - \rho' \phi) P^{-1}\right).$$

For $p = 1$, this reduces to

$$\hat{\phi}_1 \xrightarrow{D} N\left(\phi_1, \frac{1 - \phi_1^2}{n}\right).$$

For $p = 2$, this reduces to

$$\begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \end{bmatrix} \xrightarrow{D} N\left(\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}, \frac{1}{n} \begin{bmatrix} 1 - \phi_2^2 & -\phi_1(1 + \phi_2) \\ -\phi_1(1 + \phi_2) & 1 - \phi_2^2 \end{bmatrix}\right).$$

These asymptotic results for $p = 1$ and $p = 2$ were used in the confidence intervals given in Theorems 9.7 and 9.16. When the quantities in this expression are replaced by their statistical counterparts, the estimated variance–covariance matrix of the vector $\hat{\phi}$ is

$$\hat{V}[\hat{\phi}] = \frac{1}{n} (1 - r' \hat{\phi}) R^{-1}.$$

Using the diagonal elements of this matrix and the asymptotic normality of maximum likelihood estimators, an asymptotically exact $100(1 - \alpha)\%$ confidence interval for ϕ_i is easily constructed.

Theorem 9.23 For a stationary $AR(p)$ time series model, an asymptotically exact two-sided $100(1 - \alpha)\%$ confidence interval for ϕ_i is given by

$$\hat{\phi}_i - z_{\alpha/2} \sqrt{\left[\frac{1}{n} (1 - r' \hat{\phi}) R^{-1}\right]_{i,i}} < \phi_i < \hat{\phi}_i + z_{\alpha/2} \sqrt{\left[\frac{1}{n} (1 - r' \hat{\phi}) R^{-1}\right]_{i,i}}$$

for $i = 1, 2, \dots, p$, where $\hat{\phi}_i$ is the maximum likelihood estimator of ϕ_i and $z_{\alpha/2}$ is the $1 - \alpha/2$ fractile of the standard normal distribution.

The maximum likelihood estimates and associated confidence intervals will be illustrated for an economic time series in the next example.

Example 9.29 Table 9.11 contains the annual lynx (*Lynx canadensis*) pelt sales, read row-wise, at the Hudson's Bay Company in Canada from 1857 to 1911. Suggest a time series model for the annual pelt sales.

23362	31642	33757	23226	15178	7272	4448	4926	5437	16498
35971	76556	68392	37447	45686	7942	5123	7106	11250	18774
30508	42834	27345	17834	15386	9443	7599	8061	27187	51511
74050	78773	33899	18886	11520	8352	8660	12902	20331	36853
56407	39437	26761	15185	4473	5781	9117	19267	36116	58850
61478	36300	9704	3410	3774					

Table 9.11: Annual lynx pelt sales at the Hudson's Bay Company, 1857–1911.

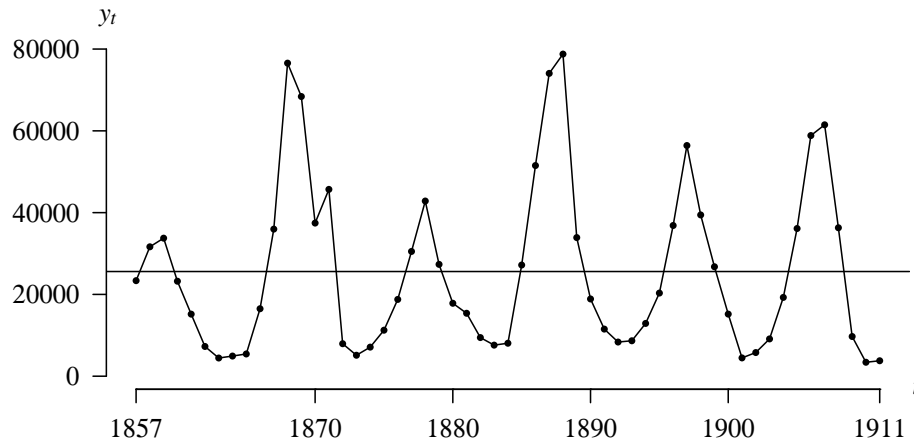


Figure 9.30: Time series plot for $n = 55$ annual lynx pelt sales (1857–1911).

The time series is plotted in Figure 9.30. The annual sales figures vary widely, from a minimum of 3410 pelts sold in 1910 to a maximum of 78,773 pelts sold in 1888. A horizontal line is drawn at the average sales over this time horizon at 25,600 pelts. The time series appears to have a periodic component that seems to cycle about every ten years or so, although nothing about the sales of lynx pelts would seem to account for an approximately decade-long periodicity. There are local maximums in the time series associated with the years 1859, 1868, 1878, 1888, 1897, and 1907. Is consumer behavior driving this periodicity? Are the prices of the pelts driving this periodic behavior? Is the availability of the pelts driving this periodic behavior?

Some further analysis of the time series indicates that ecology can answer some of the questions. Lynxes depend on the snowshoe rabbit (*Lepus americanus*) for food, and lynxes starve when the rabbits near extinction periodically. This is an example of one time series depending on another time series. We ignore the dependence on the snowshoe rabbit in our analysis because multivariate time series analysis is a topic for a more advanced time series course. We consider an $AR(p)$ model here and consider a time series model with a periodic component subsequently.

Based on Figure 9.30, is a stationary time series model appropriate? There does not appear to be any trend in the time series, but the population variance does not appear to be stable. The first local maximum (in 1859) and the third local maximum (in 1878) are not as pronounced as the others. One remedy to this nonconstant population variance is to transform the time series. Taking the logarithm of the time series values, $x_t = \ln y_t$, reduces the impact of the nonconstant variance. Figure 9.31 contains a time series plot of the logarithm of the sales figures, along with the associated sample autocorrelation and partial autocorrelation functions. For simplicity, we define time $t = 1$ to be the year 1857 and $t = 55$ to be the year 1911. The transformation has proven to be effective. As expected, the first and third peaks are still the smallest local maximums of the group, but they are less pronounced than those in the raw data. The original time series values are denoted by y_t , and the transformed time series values are denoted by $x_t = \ln y_t$. The sample autocorrelation function appears to have a damped sinusoidal shape, and the

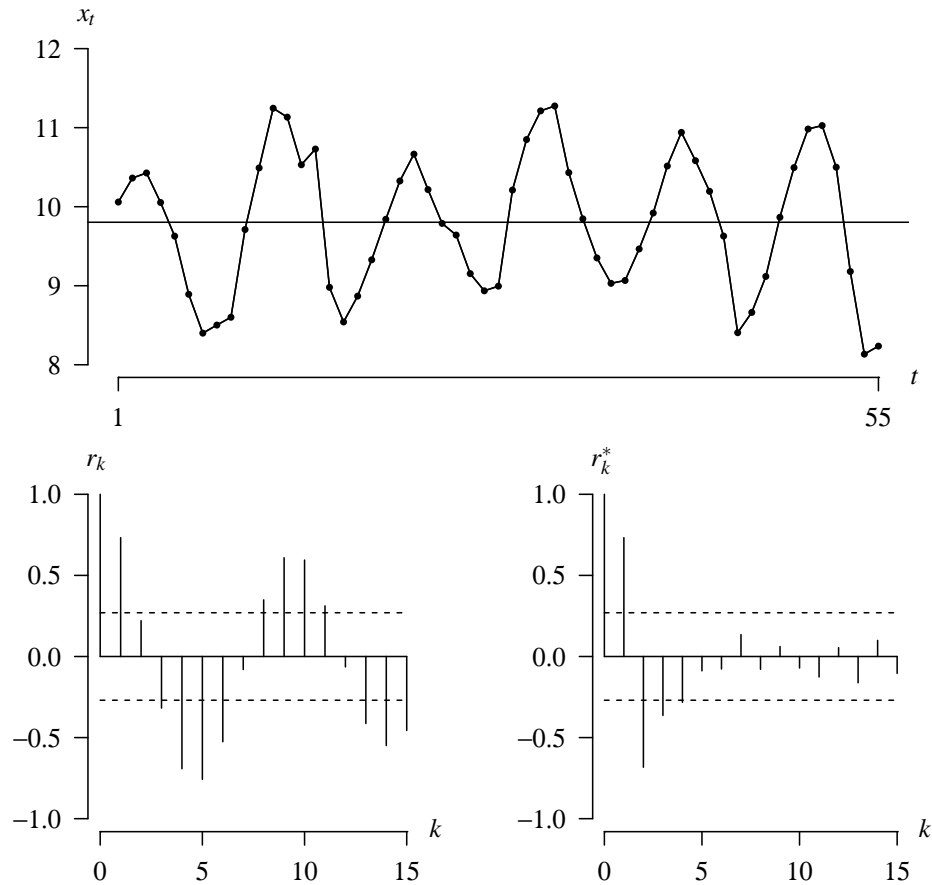


Figure 9.31: Time series plot, r_k , and r_k^* for $n = 55$ log annual lynx sales (1857–1911).

sample partial autocorrelation function cuts off after lag 4, although the spike at lag 4 is only marginally significant. The fact that the sample partial autocorrelation function cuts off leads us to consider an autoregressive time series model. Based on these graphs, we will attempt to fit tentative AR(3) and AR(4) models to the transformed time series $x_1, x_2, \dots, x_{55}$. The R code to produce the graphs in Figure 9.31 is given below.

```
y = c(23362, 31642, 33757, 23226, 15178, 7272, 4448, 4926, 5437,
      16498, 35971, 76556, 68392, 37447, 45686, 7942, 5123, 7106,
      11250, 18774, 30508, 42834, 27345, 17834, 15386, 9443, 7599,
      8061, 27187, 51511, 74050, 78773, 33899, 18886, 11520, 8352,
      8660, 12902, 20331, 36853, 56407, 39437, 26761, 15185, 4473,
      5781, 9117, 19267, 36116, 58850, 61478, 36300, 9704, 3410, 3774)
x = log(y)
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(x)
abline(h = mean(x))
```

$$X_t - 9.807 = \underset{(0.051)}{0.774} (X_{t-1} - 9.807) - \underset{(0.125)}{0.151} (X_{t-2} - 9.807) - \underset{(0.165)}{0.120} (X_{t-3} - 9.807) - \underset{(0.163)}{0.378} (X_{t-4} - 9.807) + Z_t, \underset{(0.127)}{}$$

where Z_t is white noise with estimated population variance $\hat{\sigma}_Z^2 = 0.102$. The associated approximate 95% confidence intervals are

$$\begin{aligned} 0.529 &< \phi_1 < 1.018, \\ -0.474 &< \phi_2 < 0.172, \\ -0.439 &< \phi_3 < 0.200, \\ -0.626 &< \phi_4 < -0.130, \\ 9.708 &< \mu < 9.907. \end{aligned}$$

Again, the confidence intervals for ϕ_2 and ϕ_3 containing zero should not deter us from considering the AR(4) model because the confidence interval for ϕ_4 has bounds which do not include zero.

So should the AR(3) or AR(4) model be considered the preferred stationary model? A check of the solutions of $\hat{\phi}(B) = 0$ indicates that all of the solutions lie outside of the unit circle in the complex plane for both the AR(3) and AR(4) models. Another way to select between the two models is to calculate the AIC statistic for these models. The additional R statement

```
for (p in 0:5) print(arima(x, order = c(p, 0, 0), method = "ML")$aic)
```

calculates the AIC statistic associated with the fitted AR(p) models, for $p = 0, 1, \dots, 5$. The results are shown in Table 9.12.

$p = 0$	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
145.3	101.5	63.4	52.3	46.2	48.2

Table 9.12: AIC values for AR(p) models for the transformed annual lynx pelt sales.

The AIC statistic is minimized for $p = 4$, indicating that the AR(4) model is selected over the AR(3) model. So to summarize, the tentative AR(p) time series model based on (a) the time series plot, (b) the sample autocorrelation function, (c) the sample partial autocorrelation function, and (d) the AIC statistic, is the fitted AR(4) model

$$\begin{aligned} \ln Y_t - 9.807 &= 0.774 (\ln Y_{t-1} - 9.807) - 0.151 (\ln Y_{t-2} - 9.807) - \\ &\quad 0.120 (\ln Y_{t-3} - 9.807) - 0.378 (\ln Y_{t-4} - 9.807) + Z_t, \end{aligned}$$

where Y_t corresponds to the original time series consisting of annual lynx pelt sales at the Hudson's Bay Company, and Z_t is white noise with estimated variance $\hat{\sigma}_Z^2 = 0.102$.

Model Assessment

Now that techniques for point and interval estimates for the parameters in the AR(p) model have been established, we are interested in assessing the adequacy of the fitted AR(p) time series model. This will involve an analysis of the residuals. Recall from Section 8.2.3 that the residuals are defined by

$$[\text{residual}] = [\text{observed}] - [\text{predicted}]$$

or

$$\hat{Z}_t = X_t - \hat{X}_t.$$

Since $\hat{X}_t$ is the one-step-ahead forecast from the time origin $t - 1$, this is more clearly written as

$$\hat{Z}_t = X_t - \hat{X}_{t-1}(1).$$

From Theorem 9.20, the shifted AR(p) model is

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + \cdots + \phi_p (X_{t-p} - \mu) + Z_t$$

or

$$X_t = \mu + \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + \cdots + \phi_p (X_{t-p} - \mu) + Z_t.$$

Taking the conditional expected value of both sides of this equation gives

$$E[X_t | X_1 = x_1, X_2 = x_2, \dots, X_{t-1} = x_{t-1}] = \mu + \phi_1 (x_{t-1} - \mu) + \phi_2 (x_{t-2} - \mu) + \cdots + \phi_p (x_{t-p} - \mu).$$

Replacing the parameters by their point estimators, the one-step-ahead forecast from the time origin $t - 1$ is

$$\hat{X}_{t-1}(1) = \hat{\mu} + \hat{\phi}_1 (x_{t-1} - \hat{\mu}) + \hat{\phi}_2 (x_{t-2} - \hat{\mu}) + \cdots + \hat{\phi}_p (x_{t-p} - \hat{\mu}).$$

Therefore, for the time series $x_1, x_2, \dots, x_n$ and the fitted AR(p) model with parameter estimates $\hat{\mu}$, $\hat{\phi}_1, \hat{\phi}_2, \dots, \hat{\phi}_p$, the residual at time t is

$$\hat{Z}_t = x_t - [\hat{\mu} + \hat{\phi}_1 (x_{t-1} - \hat{\mu}) + \hat{\phi}_2 (x_{t-2} - \hat{\mu}) + \cdots + \hat{\phi}_p (x_{t-p} - \hat{\mu})]$$

for $t = p + 1, p + 2, \dots, n$. The next example shows the steps associated with assessing the adequacy of the AR(4) model for the time series of annual lynx pelt sales.

Example 9.30 Fit the AR(4) time series model to the transformed annual lynx sales from Example 9.29 via maximum likelihood estimation.

- (a) Calculate and plot the residuals, their sample autocorrelation function, and their sample partial autocorrelation function.
 - (b) Conduct a test of independence on the residuals using the number of sample autocorrelation function values for the first $m = 40$ lags which fall outside of $\pm 1.96/\sqrt{n}$.
 - (c) Conduct the Box–Pierce and Ljung–Box tests for independence of the residuals.
 - (d) Conduct the turning point test for independence of the residuals.
 - (e) Plot a histogram and a QQ plot of the standardized residuals in order to assess the normality of the residuals.
- (a) The following R commands calculate the $n - 4 = 51$ residuals of the transformed time series and plot them as a time series, along with the associated sample autocorrelation function and sample partial autocorrelation function.

```

y = c(23362, 31642, 33757, 23226, 15178, 7272, 4448, 4926, 5437,
      16498, 35971, 76556, 68392, 37447, 45686, 7942, 5123, 7106,
      11250, 18774, 30508, 42834, 27345, 17834, 15386, 9443, 7599,
      8061, 27187, 51511, 74050, 78773, 33899, 18886, 11520, 8352,
      8660, 12902, 20331, 36853, 56407, 39437, 26761, 15185, 4473,
      5781, 9117, 19267, 36116, 58850, 61478, 36300, 9704, 3410, 3774)
x = log(y)
n = length(x)
p = 4
m = 40
fit = arima(x, order = c(p, 0, 0), include.mean = TRUE, method = "ML")
phi1hat = fit$coef[1]
phi2hat = fit$coef[2]
phi3hat = fit$coef[3]
phi4hat = fit$coef[4]
muhat = fit$coef[5]
zhat = x[(p + 1):n] - (muhat + phi1hat * (x[4:(n - 1)] - muhat) +
                      phi2hat * (x[3:(n - 2)] - muhat) +
                      phi3hat * (x[2:(n - 3)] - muhat) +
                      phi4hat * (x[1:(n - 4)] - muhat))
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(zhat)
acf(zhat, lag.max = m)
pacf(zhat, lag.max = m)

```

The results are displayed in Figure 9.32. The residuals do not appear to have any cyclic variation, trend, or serial correlation.

- (b) There are no sample autocorrelation function values that fall outside of the limits $\pm 1.96/\sqrt{n}$ in the plot in Figure 9.32 of the first 40 sample autocorrelation function values associated with the residuals. Since we expect $40 \cdot 0.05 = 2$ values to fall outside of these limits in the case of a good fit, we fail to reject H_0 in this case. The fit of the AR(4) model is not rejected by this test.
- (c) The additional R statements below calculate the Box–Pierce test statistic and the Ljung–Box test statistic and the associated p -values using the built-in `Box.test` function.

```

Box.test(zhat, lag = 40, type = "Box-Pierce", fitdf = 5)
Box.test(zhat, lag = 40, type = "Ljung-Box", fitdf = 5)

```

The Box–Pierce test statistic is 19.6 and the associated p -value is $p = 0.984$. The Ljung–Box test statistic is 36.1 and the associated p -value is $p = 0.418$. We fail to reject H_0 in both tests based on the chi-square critical value with $40 - 5 = 35$ degrees of freedom. Since both p -values exceed 0.05, the fit of the AR(4) model is not rejected by these tests.

- (d) The following additional R statements calculate the test statistic and the p -value for the turning point test applied to the time series consisting of the $n - 4 = 51$ residual values for the AR(4) fit to the transformed annual lynx pelt sales time series.

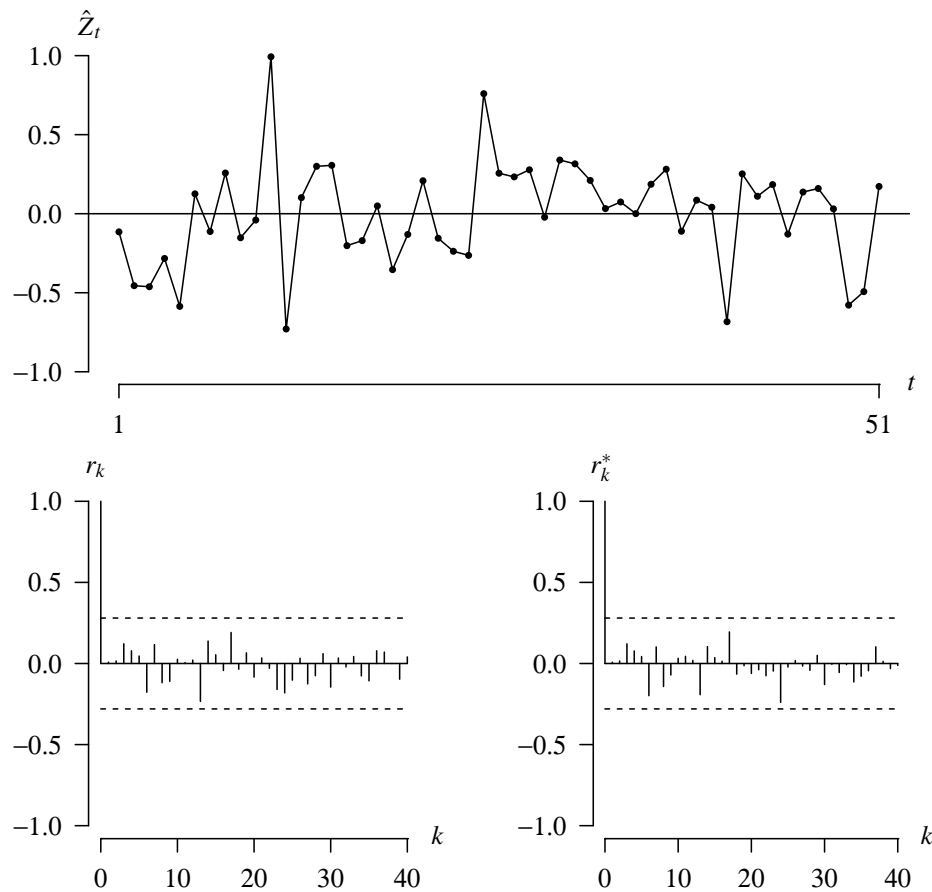


Figure 9.32: Time series plot, r_k , and r_k^* for $n - 4 = 51$ residuals from AR(4) fitted model.

```

n = n - 4
m = (2 / 3) * (n - 2)
v = (16 * n - 29) / 90
T = 0
for (i in 2:(n - 1)) {
  if ((zhat[i - 1] < zhat[i] && zhat[i] > zhat[i + 1]) ||
      (zhat[i - 1] > zhat[i] && zhat[i] < zhat[i + 1])) T = T + 1
}
s = (T - m) / sqrt(v)
2 * (1 - pnorm(abs(s)))

```

The tail probability is doubled because the alternative hypothesis is two-tailed for the turning point test. The test statistic is 0.1127 and the p -value is $p = 0.91$. The turning point test detected 33 turning points in the time series of the 51 residuals, and that is about the number that we expect to have if the residuals from the fitted AR(4) time series model of the transformed annual lynx pelt sales were mutually

independent random variables. We again fail to reject the null hypothesis in this case. The fit of the AR(4) model is not rejected by this test.

- (e) The residuals are standardized by dividing by their sample standard deviation. The following additional R statements plot a histogram of the standardized residuals using the `hist` function and a QQ plot to assess normality using the `qqnorm` function.

```
hist(zhat / sd(zhat))
qqnorm(zhat / sd(zhat))
```

The plots are shown in Figure 9.33. The histogram shows that all standardized residuals fall between -3 and 3 , but deviate significantly from a bell-shaped probability distribution, particularly in the right-hand tail. The horizontal axis on the histogram is the standardized residual and the vertical axis is the frequency. The QQ plot shows considerable nonlinearity, indicating a possible departure from normality based on the $n - 4 = 51$ residuals plotted. The horizontal axis on the QQ plot is the standardized theoretical quantile and the vertical axis is the associated normal data quantile. These plots indicate that a formal statistical goodness-of-fit test for normality should be conducted in order to assess whether Gaussian white noise is appropriate for the residuals of the fitted AR(4) time series model based on these two plots.

In summary, the model adequacy tests applied to the residuals on the AR(4) time series model of the transformed observations of the annual lynx pelt sales have revealed that the mutual independence of the residuals cannot be rejected by four statistical tests. The histogram and QQ plot of the residuals appear to not support the assumption of normally distributed residuals. We conclude that the AR(4) time series model is an adequate model for the transformed annual lynx pelt sales at the Hudson's Bay Company time series, with the exception of non-Gaussian error terms apparent in Figure 9.33. Another way to visually assess the adequacy of the time series model is to inspect time series plots of simulations of the fitted model, which is left as an exercise.

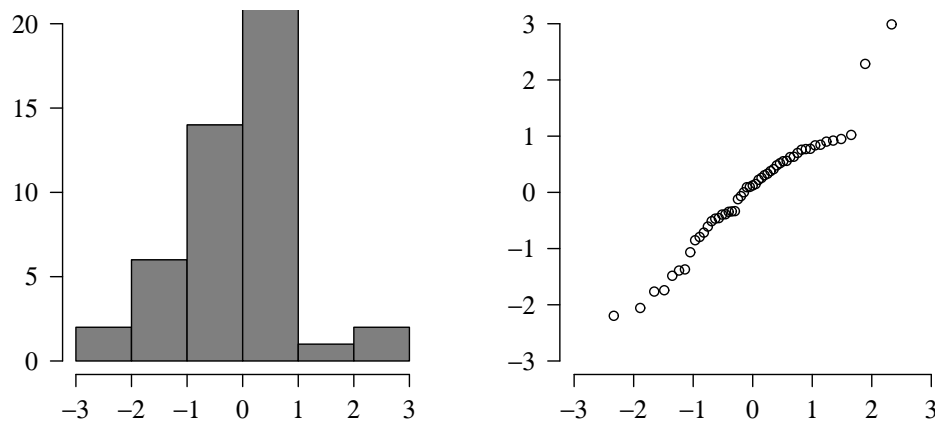


Figure 9.33: Histogram (left) and QQ plot (right) of the fitted AR(4) standardized residuals.

Forecasting

We now consider the question of forecasting future values of a time series that is governed by a shifted AR(p) time series model. In the case of the annual lynx pelt sales time series, this corresponds to the one-step-ahead forecast for 1912, the two-steps-ahead forecast for 1913, the three-steps-ahead forecast for 1914, etc. To review forecasting notation, the observed time series values are $x_1, x_2, \dots, x_n$. The forecast is being made at time $t = n$. The random future value of the time series that is h time units in the future is denoted by X_{n+h} . The associated forecasted value is denoted by $\hat{X}_{n+h}$, and is the conditional expected value

$$\hat{X}_{n+h} = E[X_{n+h} | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n].$$

We would like to find this forecasted value and an associated prediction interval for a shifted AR(p) model. As in Section 8.2.2, we assume that all parameters are known in the derivations that follow. We also assume that the parameters $\phi_1, \phi_2, \dots, \phi_p$ correspond to a stationary shifted AR(p) time series model and $p < n$.

The shifted AR(p) model is

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + \dots + \phi_p (X_{t-p} - \mu) + Z_t.$$

Replacing t by $n+1$ and solving for X_{n+1} , this becomes

$$X_{n+1} = \mu + \phi_1 (X_n - \mu) + \phi_2 (X_{n-1} - \mu) + \dots + \phi_p (X_{n-p+1} - \mu) + Z_{n+1}.$$

Taking the conditional expected value of each side of this equation results in the one-step-ahead forecast

$$\hat{X}_{n+1} = \mu + \phi_1 (x_n - \mu) + \phi_2 (x_{n-1} - \mu) + \dots + \phi_p (x_{n-p+1} - \mu)$$

because the final p observations $x_{n-p+1}, x_{n-p+2}, \dots, x_n$ in the time series $x_1, x_2, \dots, x_n$ have already been observed. The forecasted value at time $n+1$ is a function of the final p values in the time series. Applying this same process to the predicted value at time $n+2$ results in the time series model

$$X_{n+2} = \mu + \phi_1 (X_{n+1} - \mu) + \phi_2 (X_n - \mu) + \dots + \phi_p (X_{n-p+2} - \mu) + Z_{n+2}.$$

This time, the value of X_{n+1} has not been observed, so we replace it by its forecasted value when taking the conditional expected value of both sides of the equation

$$\hat{X}_{n+2} = \mu + \phi_1 (\hat{X}_{n+1} - \mu) + \phi_2 (x_n - \mu) + \dots + \phi_p (x_{n-p+2} - \mu),$$

because $x_{n-p+2}, x_{n-p+3}, \dots, x_n$ have already been observed. Continuing in this fashion, a recursive formula for the forecasted value of X_{n+h} is

$$\hat{X}_{n+h} = \mu + \phi_1 (\hat{X}_{n+h-1} - \mu) + \phi_2 (\hat{X}_{n+h-2} - \mu) + \dots + \phi_p (\hat{X}_{n+h-p} - \mu).$$

Although we would prefer an explicit formula, the recursive formula is easy to implement for an observed time series $x_1, x_2, \dots, x_n$. As in the case of the AR(1) and AR(2) models, long-term forecasts for a stationary AR(p) time series model tend to μ as the time horizon $h \rightarrow \infty$.

We would like to pair the point estimator $\hat{X}_{n+h}$ with an interval estimator, which is a prediction interval in this setting. The prediction interval gives us an indication of the precision of the forecast. In order to derive an exact two-sided $100(1 - \alpha)\%$ prediction interval for X_{n+h} , it is helpful to write the shifted AR(p) model as a shifted MA(∞) model. The coefficients $\theta_1, \theta_2, \dots$ of a stationary shifted AR(p) model written as an MA(∞) model

$$X_t = \mu + Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \dots$$

are given in terms of $\phi_1, \phi_2, \dots, \phi_p$ as was illustrated for $p = 4$ in Example 9.23. Consider this model at time $t = n + 1$. Since the error terms $Z_n, Z_{n-1}, Z_{n-2}, \dots$ are unknown but fixed because they are associated with the observed time series $x_1, x_2, \dots, x_n$, the conditional population variance of X_{n+1} is

$$V[X_{n+1}] = V[Z_{n+1}] = \sigma_Z^2$$

because the population variance of μ is zero and Z_{n+1} is the only random term in the model. The error terms at time n and prior are observed and can therefore be treated as constants. Likewise, considering the MA(∞) model at time $t = n + 2$, the conditional population variance of X_{n+2} is

$$V[X_{n+2}] = V[Z_{n+2} + \theta_1 Z_{n+1}] = (1 + \theta_1^2) \sigma_Z^2.$$

Similarly, the conditional population variance of X_{n+3} is

$$V[X_{n+3}] = V[Z_{n+3} + \theta_1 Z_{n+2} + \theta_2 Z_{n+1}] = (1 + \theta_1^2 + \theta_2^2) \sigma_Z^2.$$

Continuing in this fashion, the conditional population variance of X_{n+h} is

$$V[X_{n+h}] = (1 + \theta_1^2 + \theta_2^2 + \dots + \theta_{h-1}^2) \sigma_Z^2.$$

If we assume that the white noise terms in the MA(∞) representation of the AR(p) time series model are Gaussian white noise terms, then X_{n+h} is also normally distributed because a linear combination of mutually independent normal random variables is also normally distributed. So an exact two-sided $100(1 - \alpha)\%$ prediction interval for $\hat{X}_{n+h}$ is

$$\hat{X}_{n+h} - z_{\alpha/2} \sqrt{1 + \theta_1^2 + \theta_2^2 + \dots + \theta_{h-1}^2} \sigma_Z < X_{n+h} < \hat{X}_{n+h} + z_{\alpha/2} \sqrt{1 + \theta_1^2 + \theta_2^2 + \dots + \theta_{h-1}^2} \sigma_Z.$$

In most practical problems, the parameters in this prediction interval will be estimated from data, which results in the following *approximate* two-sided $100(1 - \alpha)\%$ prediction interval provided next.

Theorem 9.24 For a stationary shifted AR(p) time series model, a forecasted value of X_{n+h} can be calculated by the recursive equation

$$\hat{X}_{n+h} = \hat{\mu} + \hat{\phi}_1 (\hat{X}_{n+h-1} - \hat{\mu}) + \hat{\phi}_2 (\hat{X}_{n+h-2} - \hat{\mu}) + \dots + \hat{\phi}_p (\hat{X}_{n+h-p} - \hat{\mu}),$$

where $\hat{X}_{n+1} = \hat{\mu} + \hat{\phi}_1 (x_n - \hat{\mu}) + \hat{\phi}_2 (x_{n-1} - \hat{\mu}) + \dots + \hat{\phi}_p (x_{n-p+1} - \hat{\mu})$. An approximate two-sided $100(1 - \alpha)\%$ prediction interval for X_{n+h} is

$$\hat{X}_{n+h} - z_{\alpha/2} \sqrt{1 + \hat{\theta}_1^2 + \hat{\theta}_2^2 + \dots + \hat{\theta}_{h-1}^2} \hat{\sigma}_Z < X_{n+h} < \hat{X}_{n+h} + z_{\alpha/2} \sqrt{1 + \hat{\theta}_1^2 + \hat{\theta}_2^2 + \dots + \hat{\theta}_{h-1}^2} \hat{\sigma}_Z,$$

where $\hat{\theta}_1, \hat{\theta}_2, \dots$ are the estimated coefficients in the MA(∞) model associated with the estimated AR(p) model.

Example 9.31 For the time series of annual lynx pelt sales $x_1, x_2, \dots, x_{55}$ from Example 9.29, forecast the next five values (for years 1912–1916) in the time series and give approximate 95% prediction intervals for the forecasted values assuming that the transformed values in the time series arise from a shifted AR(4) time series model with parameters estimated by maximum likelihood.

The R code below uses the `ar` function to estimate the parameters in the shifted AR(4) time series model to the natural logarithm of the time series values via maximum likelihood estimation. The `predict` function implements Theorem 9.24 to calculate the forecasted values and associated standard errors for the fitted AR(4) model. These standard errors can be used to calculate approximate 95% prediction interval limits. The R `exp` function is used to convert the forecasted values $\hat{X}_{t+h}$, whose units are the natural logarithm of the annual number of lynx pelts sold, back to the original time series, whose units are the annual number of lynx pelts sold.

```
y = c(23362, 31642, 33757, 23226, 15178, 7272, 4448, 4926, 5437,
      16498, 35971, 76556, 68392, 37447, 45686, 7942, 5123, 7106,
      11250, 18774, 30508, 42834, 27345, 17834, 15386, 9443, 7599,
      8061, 27187, 51511, 74050, 78773, 33899, 18886, 11520, 8352,
      8660, 12902, 20331, 36853, 56407, 39437, 26761, 15185, 4473,
      5781, 9117, 19267, 36116, 58850, 61478, 36300, 9704, 3410, 3774)
x      = log(y)
z      = qnorm(0.975)
model  = ar(x, order.max = 4, aic = FALSE, method = "mle")
forecast = predict(model, n.ahead = 40)
xhat   = exp(forecast$pred)
pred.lo = exp(forecast$pred - z * forecast$se)
pred.hi = exp(forecast$pred + z * forecast$se)
```

The results for the first five forecasted values are summarized in Table 9.13.

Time	$t = 56$	$t = 57$	$t = 58$	$t = 59$	$t = 60$
Year	1912	1913	1914	1915	1916
Forecast	5750	14,639	41,540	74,000	75,380
Lower prediction bound	3078	6642	17,962	31,907	31,031
Upper prediction bound	10,742	32,263	96,067	171,620	183,116

Table 9.13: Forecasts and 95% prediction intervals for the annual lynx pelt sales.

For the first time, we have encountered prediction interval bounds which are not symmetric about the point estimate because of the exponentiation of the forecasted values and their prediction intervals. The widths of the prediction intervals in Table 9.13 tend to increase with the predicted value. The first forecasted value has 95% prediction interval

$$3078 < Y_{56} < 10,742$$

and the fifth forecasted value has the considerably wider 95% prediction interval

$$31,031 < Y_{60} < 183,116.$$

Figure 9.34 shows (a) the original time series $y_1, y_2, \dots, y_{55}$ as points ($\bullet$) connected by lines, (b) the first 40 forecasted annual lynx pelt sales $\hat{Y}_{56}, \hat{Y}_{57}, \dots, \hat{Y}_{95}$ as open circles ($\circ$), and (c) the 95% prediction intervals as a shaded region. There are six key observations concerning Figure 9.34.

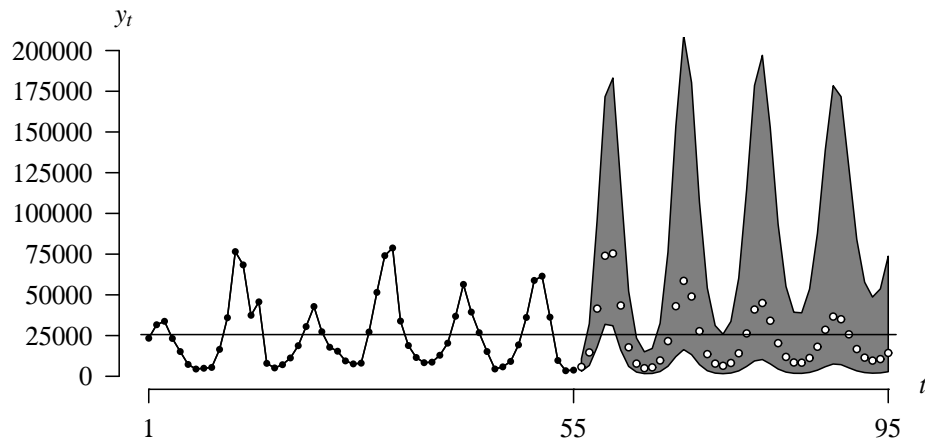


Figure 9.34: Annual lynx pelt sales forecasts and 95% prediction intervals.

- The forecasted values exhibit a similar periodicity to that of the original time series.
- The forecasted values seem to be reasonable estimates of the future values of the time series for the first cycle or two.
- The widths of the 95% prediction intervals associated with $h = 7, 8, \dots, 12$ are considerably narrower than the width of the 95% prediction interval at $h = 5$. So unlike the previous two time series (the active beaver temperatures fit to the AR(1) model in Example 9.10 and the Lake Huron levels fit to the AR(2) model in Example 9.21), the prediction interval widths do not increase monotonically in h .
- The amplitude of the cycles of the forecasted values decreases with time. As the time horizon h increases, it becomes less certain where the time series is in a cycle, so the forecasts converge to $\hat{\mu} = \bar{y} = 25,600$ pelts sold annually. Depending on the application, this might not be a welcome aspect of the forecasted values. Using a time series model that explicitly contains a cyclic component might be more appropriate for forecasting in this setting.
- The random sampling variability which is evident in the observed time series values $y_1, y_2, \dots, x_{55}$ is less evident in the forecasted values $\hat{Y}_{56}, \hat{Y}_{57}, \dots, \hat{Y}_{95}$. Observed time series values tend to exhibit the typical random sampling variability; forecasted values for a stationary shifted AR(4) time series model of the transformed time series tend to be smooth.

This subsection has introduced the AR(p) time series model. The important results for an AR(p) model are listed below.

- The standard AR(p) model can be written algebraically and with the backshift operator B as

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + Z_t \quad \text{and} \quad \phi(B)X_t = Z_t,$$

where $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$ is the characteristic polynomial and $Z_t \sim WN(0, \sigma_Z^2)$ (Definition 9.3).

- The shifted $AR(p)$ model can be written algebraically and with the backshift operator B as (Theorem 9.20)

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \phi_2 (X_{t-2} - \mu) + \cdots + \phi_p (X_{t-p} - \mu) + Z_t \quad \text{and} \quad \phi(B) (X_t - \mu) = Z_t.$$

- The $AR(p)$ model is always invertible; the $AR(p)$ model is stationary when the solutions of $\phi(B) = 0$ all lie outside of the unit circle in the complex plane (Theorem 8.3).
- The $AR(p)$ population autocorrelation function is a mixture of damped exponential functions, associated with real roots of $\phi(B)$, and damped sinusoidal functions, associated with complex roots of $\phi(B)$ (Theorem 9.18).
- The $AR(p)$ population partial autocorrelation function cuts off after lag p (Theorem 9.19), making its shape easier to recognize than the population autocorrelation function for the statistical counterparts associated with a realization of a time series.
- The stationary shifted $AR(p)$ model can be written as a shifted $MA(\infty)$ model (as illustrated in Example 9.23).
- The $p + 2$ parameters in the shifted $AR(p)$ model, $\mu, \phi_1, \phi_2, \dots, \phi_p$, and σ_Z^2 , can be estimated from a realization of a time series $x_1, x_2, \dots, x_n$ by the method of moments (Theorem 9.21), least squares (Theorem 9.22), and maximum likelihood. The point estimators for $\mu, \phi_1, \phi_2, \dots, \phi_p$, and σ_Z^2 are denoted by $\hat{\mu}, \hat{\phi}_1, \hat{\phi}_2, \dots, \hat{\phi}_p$, and $\hat{\sigma}_Z^2$, and are typically paired with asymptotically exact two-sided $100(1 - \alpha)\%$ confidence intervals (Theorem 9.23).
- The forecasted value $\hat{X}_{n+h}$ in a shifted $AR(p)$ model is a function of the last p values in an observed time series $x_1, x_2, \dots, x_n$ and can be calculated by a recursive formula. The forecast approaches $\hat{\mu} = \bar{x}$ as the time horizon $h \rightarrow \infty$. The associated prediction interval has width that increases as h increases and approaches a limit as the time horizon $h \rightarrow \infty$ (Theorem 9.24).

9.1.4 Computing

The R time series functions used in this section are summarized here. The `ARMAacf` function computes the population autocorrelation function or the population partial autocorrelation function for an $ARMA(p, q)$ time series model. The generic version of the function is

```
ARMAacf(ar = numeric(), ma = numeric(), lag.max = r, pacf = FALSE)
```

where `ar` is a vector containing the autoregressive coefficients $\phi_1, \phi_2, \dots, \phi_p$, `ma` is a vector containing the moving average coefficients $\theta_1, \theta_2, \dots, \theta_q$, `lag.max` contains the number of lags required, and `pacf` is a logical object. The function returns $\rho(0), \rho(1), \dots, \rho(\text{lag.max})$ when `pacf` is `FALSE`, or $\rho^*(1), \rho^*(2), \dots, \rho^*(\text{lag.max})$ when `pacf` is `TRUE`. The `ARMAacf` function is illustrated in Example 9.25.

The `arima.sim` function generates a simulation of a time series. The generic version of the function is

```
arima.sim(model, n, rand.gen = rnorm, innov = rand.gen(n, ...),
          n.start = NA, start.innov = rand.gen(n.start, ...),
          ...)
```

where `model` is a list with components `ar` containing the autoregressive coefficients $\phi_1, \phi_2, \dots, \phi_p$, and `ma` containing the moving average coefficients $\theta_1, \theta_2, \dots, \theta_q$, `n` is the length of the simulated time series to be generated, `rand.gen` is a function to generate the white noise terms, `n.start` is the length of the warm-up period, and `start.innov` is a time series of white noise terms used in the warm-up period. The returned value is a vector containing the n simulated time series values $x_1, x_2, \dots, x_n$. The `arima.sim` function is illustrated in Examples 9.2, 9.13, and 9.18. The warm-up period associated with the `arima.sim` function can be avoided by generating initial values from the appropriate multivariate distribution. For an AR(1) model with Gaussian white noise error terms, the `rnorm` function, whose generic syntax is

```
rnorm(n, mean = 0, sd = 1)
```

where `n` is the number of random variates to generate, `mean` is the population mean, and `sd` is the population standard deviation, can be used to seed the simulated time series. The `rnorm` function is illustrated in Example 9.1. For an AR(p) model, with $p > 1$, with Gaussian white noise error terms, the `mvnorm` function from the MASS package, whose generic syntax is

```
mvnorm(n = 1, mu, Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
```

where `n` is the number of random vectors to generate, `mu` is the population mean vector, and `Sigma` is the population variance–covariance matrix, can be used to seed the simulated time series. The `mvnorm` function is illustrated in Examples 9.12 and 9.26.

When determining parameter estimates that cannot be expressed in closed form, the `optim` function provides general-purpose optimization capability that can be applied to minimizing the sum of squares to find the least squares estimates or maximizing the log likelihood function to find the maximum likelihood estimators. The generic syntax for `optim` is

```
optim(par, fn, gr = NULL, ...,
      method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN", "Brent"),
      lower = -Inf, upper = Inf, control = list(), hessian = FALSE)
```

where `par` is a vector containing initial parameter estimates and `fn` is the function to be minimized (by default). The `optim` function is illustrated in Examples 9.5, 9.6, and 9.16. A parameter estimation function that is exclusively for autoregressive time series models is `ar`. The generic format for `ar` is

```
ar(x, aic = TRUE, order.max = NULL,
   method = c("yule-walker", "burg", "ols", "mle", "yw"),
   na.action, series, ...)
```

where `x` is a vector containing the observed time series values, `aic` is a logical variable (TRUE means that the Akaike Information Criterion is used to choose the order of the model and FALSE means that an autoregressive model of order `order.max` is fitted), `order.max` is maximum order of the autoregressive model to fit, `method` is the estimation method ("yule-walker" or "yw" for Yule–Walker, "burg" for Burg’s algorithm, "ols" for least squares, "mle" for maximum likelihood), and `na.action` indicates how to handle missing values in the time series. The `ar` function is illustrated in Examples 9.7, 9.8, 9.10, 9.17, 9.19, 9.21, 9.27, 9.28, and 9.31. The `arima` function also estimates parameters from an observed time series. The generic format for `arima` is

```
arima(x, order = c(0L, 0L, 0L),
      seasonal = list(order = c(0L, 0L, 0L), period = NA),
```

```
xreg = NULL, include.mean = TRUE, transform.pars = TRUE,
fixed = NULL, init = NULL, method = c("CSS-ML", "ML", "CSS"), n.cond,
SSinit = c("Gardner1980", "Rossignol2011"), optim.method = "BFGS",
optim.control = list(), kappa = 1e6)
```

where $\mathbf{x}$ is a vector containing the observed time series values, order is a vector containing the values of p , d , and q , `include.mean` is a logical variable (TRUE includes estimation of a population mean term μ and FALSE estimates just the parameters in the standard model), and `method` is CSS (conditional sum of squares) or ML (maximum likelihood). The `arima` function is illustrated in Examples 9.9, 9.18, 9.20, and 9.29.

Three functions were introduced in this section for assessing model adequacy. The `Box.test` function computes the Box–Pierce or Ljung–Box test statistic and associated p -value. The generic syntax is

```
Box.test(x, lag = 1, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)
```

where $\mathbf{x}$ is a vector containing the observed time series values, `lag` is the number of sample autocorrelation function values to be used in the test, `type` is either "Box-Pierce" or "Ljung-Box", and `fitdf` is the number of degrees of freedom to be subtracted in the case of $\mathbf{x}$ being a time series of residuals. The `Box.test` function is illustrated in Examples 9.8, 9.19, and 9.30, along with the `hist` and `qqnorm` functions, which are helpful in visually assessing the normality of the residuals.

Forecasting can be performed automatically using the generic `predict` function, which calculates predicted values of a time series from a fitted function. The `predict` function is illustrated in Examples 9.10, 9.21, and 9.31.

More details on the R functions used in this section can be found using the `help` function. Sample invocations of the functions are displayed using the `example` function.

This concludes the introduction to the autoregressive time series model, with subsections devoted to the $\text{AR}(1)$, $\text{AR}(2)$, and $\text{AR}(p)$ models. An analogous treatment of moving average models is contained in the next section.

9.2 Moving Average Models

Moving average models for a time series will be introduced in this section. A moving average model of order q is a special case of an $\text{ARMA}(p, q)$ model with no autoregressive terms (that is, $p = 0$) and q moving average terms, specified as

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots + \theta_q Z_{t-q},$$

where $\theta_1, \theta_2, \dots, \theta_q$ are real-valued parameters and $\{Z_t\}$ is a time series of white noise. Rather than diving right into an $\text{MA}(q)$ model, we first have separate subsections for the $\text{MA}(1)$ and $\text{MA}(2)$ models because the mathematics are somewhat easier than the general case and some important geometry and intuition can be developed with these restricted models. In the subsection on the $\text{MA}(1)$ model that follows, we will

- define the time series model for $\{X_t\}$,
- determine the values of the parameters associated with an invertible model,
- derive the population autocorrelation and partial autocorrelation functions,

- develop algorithms for simulating observations from the time series,
- inspect simulated realizations to establish patterns, and
- estimate parameters from a time series realization $\{x_t\}$.

The important steps of model assessment, model selection, and forecasting future values of the times series are left as exercises because they follow along the same lines as those steps for the autoregressive models covered in the previous section.

The purpose of deriving the population autocorrelation and partial autocorrelation functions is to build an inventory of shapes and patterns for these functions that can be used to identify tentative time series models from their sample counterparts by making a visual comparison between population and sample versions. This inventory of shapes and patterns plays an analogous role to knowing the shapes of various probability density functions (for example, the bell-shaped normal probability density function or the rectangular-shaped uniform distribution) in the analysis of univariate data in which the shape of the histogram is visually compared to the inventory of probability density function shapes.

In the MA(1) subsection that follows, a single example of a time series will be carried through the various statistical procedures given in the list above. Stationarity plays a critical role in time series analysis because we are not able to forecast future values of the time series without knowing that the probability model is stable over time. This is why the visual assessment of a plot of the time series is always a critical first step in the analysis of a time series. Fortunately, all MA(q) time series models are stationary.

9.2.1 The MA(1) Model

The moving average model with one term is the simplest of the ARMA family of time series models in terms of the ability to derive probabilistic properties.

Definition 9.4 A first-order moving average time series model, denoted by MA(1), for the time series $\{X_t\}$ is defined by

$$X_t = Z_t + \theta Z_{t-1},$$

where θ is a real-valued parameter and $\{Z_t\}$ is a time series of white noise:

$$Z_t \sim WN(0, \sigma_Z^2).$$

An observed value in the time series, X_t , is given by the current white noise term, plus the parameter θ multiplied by the white noise term from one time period ago. No subscript is necessary on the θ parameter because there is only one θ parameter in the MA(1) model. So there are two parameters that define an MA(1) model: the coefficient θ and the population variance of the white noise σ_Z^2 .

Some authors prefer to parameterize the MA(1) model as

$$X_t = \theta_0 Z_t + \theta_1 Z_{t-1},$$

where θ_0 and θ_1 are real-valued parameters. We avoid this parameterization because the θ_0 parameter is redundant in the sense that the population variance of the white noise σ_Z^2 is absorbed into the θ_0 parameter. Also, some authors use a $-$ rather than a $+$ between two terms on the right-hand side of the model.

To illustrate the thinking behind the MA(1) model in a specific context, let X_t represent the monthly unemployment, as a percentage, in month t . The MA(1) model indicates that this month's unemployment, denoted by X_t , equals θ multiplied by last month's random white noise term, θZ_{t-1} , plus this month's random white noise term Z_t .

MA(1) models are used less often than autoregressive models, and this is partly due to more limited potential shapes for the population autocorrelation function, as will be seen next.

Stationarity and the Population Autocorrelation Function

One initial important question concerning the MA(1) model is whether or not the model is stationary. Rather than appealing to Theorem 8.4, we show this below using first principles. Recall from Definition 7.6 that a time series model is stationary if (a) the expected value of X_t is constant for all t , and (b) the population covariance between X_s and X_t depends only on the lag $|t - s|$. The expected value of X_t is

$$E[X_t] = E[Z_t + \theta Z_{t-1}] = E[Z_t] + \theta E[Z_{t-1}] = 0$$

for all values of the parameters θ and σ_Z^2 , and all values of t . Using the defining formula for population covariance, the population autocovariance function is

$$\begin{aligned} \gamma(s, t) &= \text{Cov}(X_s, X_t) \\ &= E[(X_s - E[X_s])(X_t - E[X_t])] \\ &= E[X_s X_t] \\ &= E[(Z_s + \theta Z_{s-1})(Z_t + \theta Z_{t-1})] \\ &= E[Z_s Z_t] + \theta E[Z_{s-1} Z_t] + \theta E[Z_s Z_{t-1}] + \theta^2 E[Z_{s-1} Z_{t-1}] \\ &= \begin{cases} V[Z_t] + \theta^2 V[Z_{t-1}] & |t - s| = 0 \\ \theta V[Z_t] & |t - s| = 1 \\ 0 & |t - s| = 2, 3, \dots \end{cases} \\ &= \begin{cases} (1 + \theta^2) \sigma_Z^2 & |t - s| = 0 \\ \theta \sigma_Z^2 & |t - s| = 1 \\ 0 & |t - s| = 2, 3, \dots \end{cases} \end{aligned}$$

Since $E[X_t] = 0$ for all values of t and the population autocovariance function depends only on the lag $|t - s|$, we conclude that the MA(1) time series model is stationary. Furthermore, the population autocovariance function can be expressed in terms of the lag k as

$$\gamma(k) = \begin{cases} (1 + \theta^2) \sigma_Z^2 & k = 0 \\ \theta \sigma_Z^2 & k = 1 \\ 0 & k = 2, 3, \dots \end{cases}$$

Dividing by the population autocovariance function by $\gamma(0) = V[X_t] = (1 + \theta^2) \sigma_Z^2$ gives the population autocorrelation function

$$\rho(k) = \begin{cases} 1 & k = 0 \\ \theta / (1 + \theta^2) & k = 1 \\ 0 & k = 2, 3, \dots \end{cases}$$

This derivation constitutes a proof of the following result.

Theorem 9.25 The MA(1) time series $\{X_t\}$ is stationary for all values of the parameters θ and σ_Z^2 with population autocorrelation function

$$\rho(k) = \begin{cases} 1 & k = 0 \\ \theta / (1 + \theta^2) & k = 1 \\ 0 & k = 2, 3, \dots \end{cases}$$

So the population autocorrelation function consists of a single nonzero value at lag 1 for a nonzero parameter θ and zero values thereafter. Six important observations concerning this population autocorrelation function are given below.

- The sign of $\rho(1)$ is the same as the sign of θ .
- The population autocorrelation function cuts off after lag 1 for an MA(1) time series model. The time series model has a “memory” of just one time period. Figure 9.35 illustrates the relationship between the white noise values $\{Z_t\}$ and the MA(1) time series observations $\{X_t\}$. Observations of the time series that are two or more time periods apart, such as X_2 and X_4 , have no white noise terms in common, so the lag 2 population autocorrelation, $\rho(2)$, is zero. The third observation in the time series X_3 , for example, shares the white noise term Z_2 with X_2 and the white noise term Z_3 with X_4 , but is not affected by any white noise terms before Z_2 or after Z_3 .

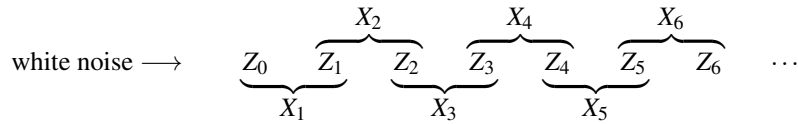


Figure 9.35: Relationship between white noise $\{Z_t\}$ and $\{X_t\}$ for an MA(1) model.

- The lag 1 population autocorrelation $\rho(1) = \theta / (1 + \theta^2)$ can be written as a quadratic equation in θ as

$$\rho(1)\theta^2 - \theta + \rho(1) = 0.$$

For nonzero values of θ , the two roots of this quadratic equation are both positive or both negative. Furthermore, a little algebra reveals that the product of the two roots of this quadratic equation equals 1. Figure 9.36 shows the parabolas associated with this quadratic equation for $\rho(1) = 2/5$ (with associated roots $\theta = 1/2$ and $\theta = 2$) and $\rho(1) = -2/5$ (with associated roots $\theta = -1/2$ and $\theta = -2$).

- The value $\rho(1)$ must lie in the interval $-1/2 \leq \rho(1) \leq 1/2$. This can be seen in the plot of $\rho(1) = \theta / (1 + \theta^2)$ versus θ given by the solid curve in Figure 9.37, which indicates that $\rho(1)$ is minimized at $\rho(1) = -1/2$ when $\theta = -1$ and maximized at $\rho(1) = 1/2$ when $\theta = 1$. This constraint means that the MA(1) model is more limited in application than the autoregressive models from the previous chapter. In order to fit an MA(1) model to observed time series values $x_1, x_2, \dots, x_n$, it must be the case that (a) the length of the time series n is large enough (about $n = 50$ or $n = 60$) to use an ARMA model, (b) the sample autocorrelation function has a single statistically significant spike at lag 1, and (c) the statistically significant spike at lag 1 satisfies $-1/2 \leq r_1 \leq 1/2$ to be compatible with the constraint $-1/2 \leq \rho(1) \leq 1/2$.

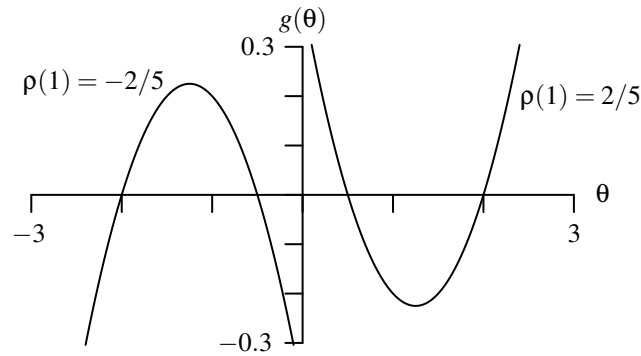


Figure 9.36: The parabola $g(\theta) = \rho(1)\theta^2 - \theta + \rho(1)$ for $\rho(1) = 2/5$ and $\rho(1) = -2/5$.

- Figure 9.37 also reveals a more subtle aspect of the population lag 1 autocorrelation. Notice that for $\theta = 1/2$, the population lag 1 autocorrelation is $\rho(1) = 2/5$. But for $\theta = 2$, the population lag 1 autocorrelation is also $\rho(1) = 2/5$. The geometry associated with these two values of θ resulting in the same value for $\rho(1)$ is indicated by the dashed lines in Figure 9.37. This problem is not just limited to $\theta = 1/2$ and $\theta = 2$; there are an infinite number of pairs of θ values that will result in the same population lag 1 autocorrelation function value. More generally, the MA(1) model

$$X_t = Z_t + \theta Z_{t-1}$$

and the MA(1) model

$$X_t = Z_t + \frac{1}{\theta} Z_{t-1}$$

have identical population autocorrelation functions. This means that there is not a one-to-one

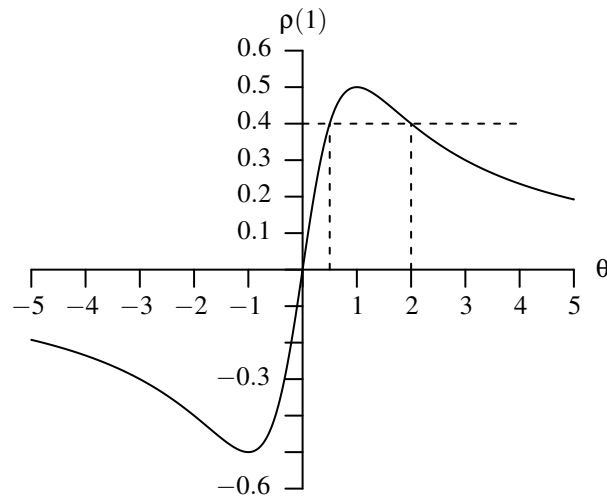


Figure 9.37: Graph of $\rho(1) = \theta / (1 + \theta^2)$ versus θ for an MA(1) time series model.

correspondence between a particular value of θ and the associated value of $\rho(1)$. This brings up the notion of *invertibility*, which was defined in Definition 8.3. An invertible time series model has a unique value of θ in the MA(1) model corresponding to a particular population autocorrelation function.

Invertibility

All MA(1) models are stationary per Theorem 8.3 because there are a finite number of moving average terms in Definition 9.4. Recall from Definition 8.3 that an ARMA(p, q) time series model for $\{X_t\}$ is *invertible* if the white noise term at time t can be expressed as

$$Z_t = \sum_{j=0}^{\infty} \pi_j X_{t-j},$$

where the coefficients π_j satisfy

$$\sum_{j=0}^{\infty} \pi_j^2 < \infty.$$

There are no restrictions on θ necessary to ensure stationarity for an MA(1) model. However, it can be advantageous to restrict the values of θ in order to achieve invertibility. Returning to Figure 9.37, we can use the definition of invertibility to determine whether we use $|\theta| < 1$ or $|\theta| > 1$ for the invertibility region for an MA(1) model.

Just as we were able to write an AR(1) time series model as an MA(∞) time series model in Section 9.1.1, we now perform the algebraic steps necessary to write an MA(1) time series model as an AR(∞) time series model. We want to write Z_t in terms of current and previous values of X_t as shown in Definition 8.3. To begin, recall that the MA(1) model given by

$$X_t = Z_t + \theta Z_{t-1}$$

can be shifted in time and is equally valid for other t values, for example,

$$\begin{aligned} X_{t-1} &= Z_{t-1} + \theta Z_{t-2} \\ X_{t-2} &= Z_{t-2} + \theta Z_{t-3} \\ \vdots &= \vdots \end{aligned}$$

These formulas can be solved for $Z_{t-1}, Z_{t-2}, \dots$ as

$$\begin{aligned} Z_{t-1} &= X_{t-1} - \theta Z_{t-2} \\ Z_{t-2} &= X_{t-2} - \theta Z_{t-3} \\ \vdots &= \vdots \end{aligned}$$

Making successive substitutions into the MA(1) model results in

$$\begin{aligned} X_t &= Z_t + \theta Z_{t-1} \\ &= Z_t + \theta (X_{t-1} - \theta Z_{t-2}) \\ &= Z_t + \theta X_{t-1} - \theta^2 Z_{t-2} \\ &= Z_t + \theta X_{t-1} - \theta^2 (X_{t-2} - \theta Z_{t-3}) \end{aligned}$$

$$\begin{aligned}
&= Z_t + \theta X_{t-1} - \theta^2 X_{t-2} + \theta^3 X_{t-3} \\
&\quad \vdots \\
&= Z_t + \theta X_{t-1} - \theta^2 X_{t-2} + \theta^3 X_{t-3} - \theta^4 X_{t-4} + \cdots .
\end{aligned}$$

This can be recognized as an $AR(\infty)$ time series model.

Theorem 9.26 An MA(1) model with parameter θ can be written as the $AR(\infty)$ model

$$X_t = Z_t + \theta X_{t-1} - \theta^2 X_{t-2} + \theta^3 X_{t-3} - \theta^4 X_{t-4} + \cdots .$$

Representing an MA(1) model as an $AR(\infty)$ model is known as *duality*. Solving this equation for Z_t gives

$$Z_t = X_t - \theta X_{t-1} + \theta^2 X_{t-2} - \theta^3 X_{t-3} + \theta^4 X_{t-4} - \cdots ,$$

which is the form required for Definition 8.3. So the coefficients $\pi_0, \pi_1, \pi_2, \dots$ for the MA(1) model from Definition 8.3 are

$$\pi_0 = 1, \pi_1 = -\theta, \pi_2 = \theta^2, \pi_3 = -\theta^3, \pi_4 = \theta^4, \dots ,$$

or in general, $\pi_j = (-\theta)^j$, for $j = 0, 1, 2, \dots$. Definition 8.3 requires that

$$\sum_{j=0}^{\infty} |\pi_j| = \sum_{j=0}^{\infty} |(-\theta)^j| = 1 + \theta + \theta^2 + \theta^3 + \cdots < \infty$$

to achieve stationarity. This summation is a geometric series that converges when $|\theta| < 1$, so this is the invertibility region for an MA(1) model.

Theorem 9.27 The MA(1) time series model is invertible when $-1 < \theta < 1$.

The invertibility criterion $-1 < \theta < 1$ ensures that each value of θ in the interval corresponds to a unique MA(1) time series model. Stated in another fashion, invertibility implies that there is a one-to-one correspondence between the value of the θ parameter and the population autocorrelation function.

The MA(1) time series model can be written in terms of the backshift operator B as

$$X_t = (1 + \theta B)Z_t = Z_t + \theta Z_{t-1}.$$

Doubling up the use of θ as a function name, the expression

$$\theta(B) = 1 + \theta B$$

is the *characteristic polynomial* for the MA(1) model. Notice that the MA(1) model is invertible when $|\theta| < 1$, which corresponds to the root of $\theta(B) = 0$ falling *outside* of the interval $[-1, 1]$. Solving $\theta(B) = 1 + \theta B = 0$ results in $B = -1/\theta$. This notion will be generalized in the next two subsections for higher-order MA models as the roots of $\theta(B) = 0$ falling outside of the unit circle in the complex plane to establish invertibility.

Now that stationarity for all MA(1) time series models has been established, the condition for invertibility has been established, and the population autocorrelation function has been derived, we turn to determining the partial autocorrelation function.

Population Partial Autocorrelation Function

The population partial autocorrelation function can be determined by using the defining formula in Definition 7.4. The lag zero population partial autocorrelation is $\rho^*(0) = 1$. The lag one population partial autocorrelation is $\rho^*(1) = \rho(1) = \theta / (1 + \theta^2)$. After a little algebra, the lag two population partial autocorrelation is

$$\rho^*(2) = \frac{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & \rho(2) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) \\ \rho(1) & 1 \end{vmatrix}} = \frac{\rho(2) - [\rho(1)]^2}{1 - [\rho(1)]^2} = -\frac{[\rho(1)]^2}{1 - [\rho(1)]^2} = -\frac{\theta^2 (1 - \theta^2)}{1 - \theta^6}.$$

The lag three population partial autocorrelation is

$$\rho^*(3) = \frac{\begin{vmatrix} 1 & \rho(1) & \rho(1) \\ \rho(1) & 1 & \rho(2) \\ \rho(2) & \rho(1) & \rho(3) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) & \rho(2) \\ \rho(1) & 1 & \rho(1) \\ \rho(2) & \rho(1) & 1 \end{vmatrix}} = \frac{[\rho(1)]^3}{1 - 2[\rho(1)]^2} = \frac{\theta^3 (1 - \theta^2)}{1 - \theta^8}.$$

This pattern generalizes to the lag k population partial autocorrelation

$$\rho^*(k) = \frac{(-1)^{k+1} \theta^k (1 - \theta^2)}{1 - \theta^{2(k+1)}}$$

for $k = 1, 2, \dots$, which can also be written as

$$\rho^*(k) = \frac{(-1)^{k+1} \theta^k}{1 + \theta^2 + \theta^4 + \dots + \theta^{2k}}$$

for $k = 1, 2, \dots$. This constitutes a proof of the following result.

Theorem 9.28 The invertible MA(1) time series model for $\{X_t\}$ with $-1 < \theta < 1$ has population partial autocorrelation function

$$\rho^*(k) = \frac{(-1)^{k+1} \theta^k (1 - \theta^2)}{1 - \theta^{2(k+1)}}$$

for $k = 1, 2, \dots$.

When $\theta = 0$, both the population autocorrelation function and the partial autocorrelation function have just a single spike at $\rho(0) = \rho^*(0) = 1$; the MA(1) model reduces to just white noise in this case. When $0 < \theta < 1$, $\rho(1) > 0$ and $\rho^*(k)$ tails out and alternates in sign. When $-1 < \theta < 0$, $\rho(1) < 0$ and $\rho^*(k)$ tails out and is negative for $k = 1, 2, \dots$.

Example 9.32 Plot the first eight lags of the population autocorrelation function $\rho(k)$ and the population partial autocorrelation function $\rho^*(k)$ for an MA(1) model with $\theta = 0.9$.

The values of the population autocorrelation function and the population partial autocorrelation function can be calculated using the formulas in Theorems 9.25 and 9.28, respectively, or by calling the built-in R `ARMAacf` function as shown below.

```
ARMAacf(ar = 0, ma = 9 / 10, 8)
ARMAacf(ar = 0, ma = 9 / 10, 8, pacf = TRUE)
```

The plots of these functions are displayed in Figure 9.38. As expected for an MA(1) model, the population autocorrelation function cuts off after lag 1 and the population partial autocorrelation alternates in sign and tails out.

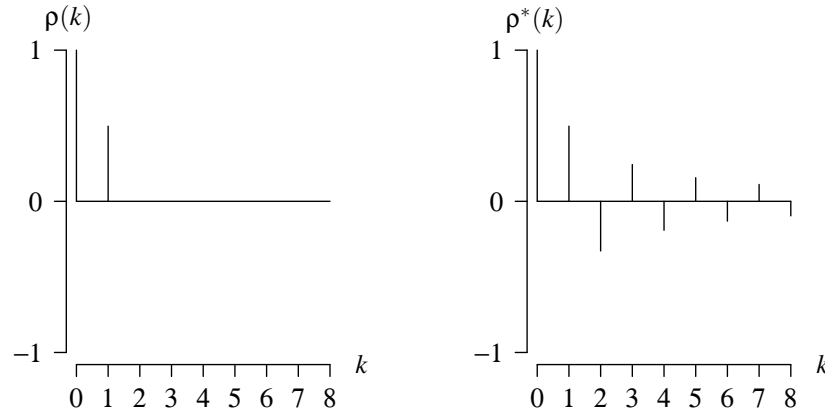


Figure 9.38: Graphs of $\rho(k)$ (left) and $\rho^*(k)$ (right) for an MA(1) model with $\theta = 9/10$.

The Shifted MA(1) Model

The population mean function for the MA(1) model is $E[X_t] = 0$, which is not of much use in practice because most real-world time series are not centered around zero. Adding a third parameter μ to overcome this shortcoming results in the enhanced MA(1) model

$$X_t = \mu + Z_t + \theta Z_{t-1},$$

which has population mean function $E[X_t] = \mu$ and population autocorrelation function and population partial autocorrelation function given in Theorems 9.25 and 9.28 because population variance and covariance are unaffected by a shift in the time series model. There are now three parameters for the time series model: μ , θ , and σ_Z^2 .

Theorem 9.29 A shifted first-order moving average model for the time series $\{X_t\}$ is defined by

$$X_t = \mu + Z_t + \theta Z_{t-1},$$

where μ , θ , and $\sigma_Z^2 > 0$ are real-valued parameters and $\{Z_t\}$ is a time series of white noise. This model is invertible when $-1 < \theta < 1$. This model is stationary for all values of the parameters μ , θ , and σ_Z^2 . The expected value of X_t is $E[X_t] = \mu$. The population autocorrelation function and partial autocorrelation function are given in Theorems 9.25 and 9.28.

Simulation

An MA(1) time series can be simulated by appealing to the defining formula for the MA(1) model from Definition 9.4:

$$X_t = Z_t + \theta Z_{t-1}.$$

The algorithm given below generates an initial white noise value Z_0 , and then uses an additional n white noise terms $Z_1, Z_2, \dots, Z_n$ to generate the time series values $X_1, X_2, \dots, X_n$ using the MA(1) defining formula. Indentation denotes nesting in the algorithm.

```

t ← 0
generate  $Z_t \sim WN(0, \sigma_Z^2)$ 
while ( $t < n$ )
  t ← t + 1
  generate  $Z_t \sim WN(0, \sigma_Z^2)$ 
   $X_t \leftarrow Z_t + \theta Z_{t-1}$ 

```

The three-parameter shifted MA(1) time series model that includes a population mean parameter μ can be simulated by simply adding μ to each time series observation generated by this algorithm. So to generate a realization of an MA(1) time series model in R, we must define (a) the value of θ , (b) the distribution of the white noise, (c) the value of σ_Z^2 , and, if this is a shifted MA(1) model, (d) the value of the shift parameter μ .

Example 9.33 Generate a realization of $n = 100$ observations from an MA(1) time series model with $\theta = 9/10$, Gaussian white noise terms, and $\sigma_Z^2 = 1$.

The parameter $\theta = 9/10$ just barely falls in the invertibility region $-1 < \theta < 1$. This choice of θ results in a population lag 1 autocorrelation that is very close to its largest possible value. The population lag 1 autocorrelation function value associated with this model is

$$\rho(1) = \frac{\theta}{1 + \theta^2} = \frac{9/10}{1 + (9/10)^2} = \frac{90}{181} \cong 0.4972,$$

so we expect a nearby value for r_1 from the simulated time series values. The R code below generates $n = 100$ simulated time series values and places them in the vector named **x**.

```

set.seed(37)
n      = 100
z      = rnorm(n + 1)
x      = numeric(n)
theta  = 0.9
for (t in 1:n) x[t] = z[t + 1] + theta * z[t]

```

Use the `plot.ts` function to plot the time series contained in **x**, the `acf` function to plot the associated correlogram, and the `pacf` function plot the associated sample partial autocorrelation function.

```

layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(x)
acf(x)
pacf(x)

```

The time series plot of the realization, the associated correlogram, and the associated sample partial autocorrelation function for $\theta = 0.9$ and standard normal noise terms are given in Figure 9.39. A horizontal line is drawn on the time series plot at $E[X_t] = 0$.

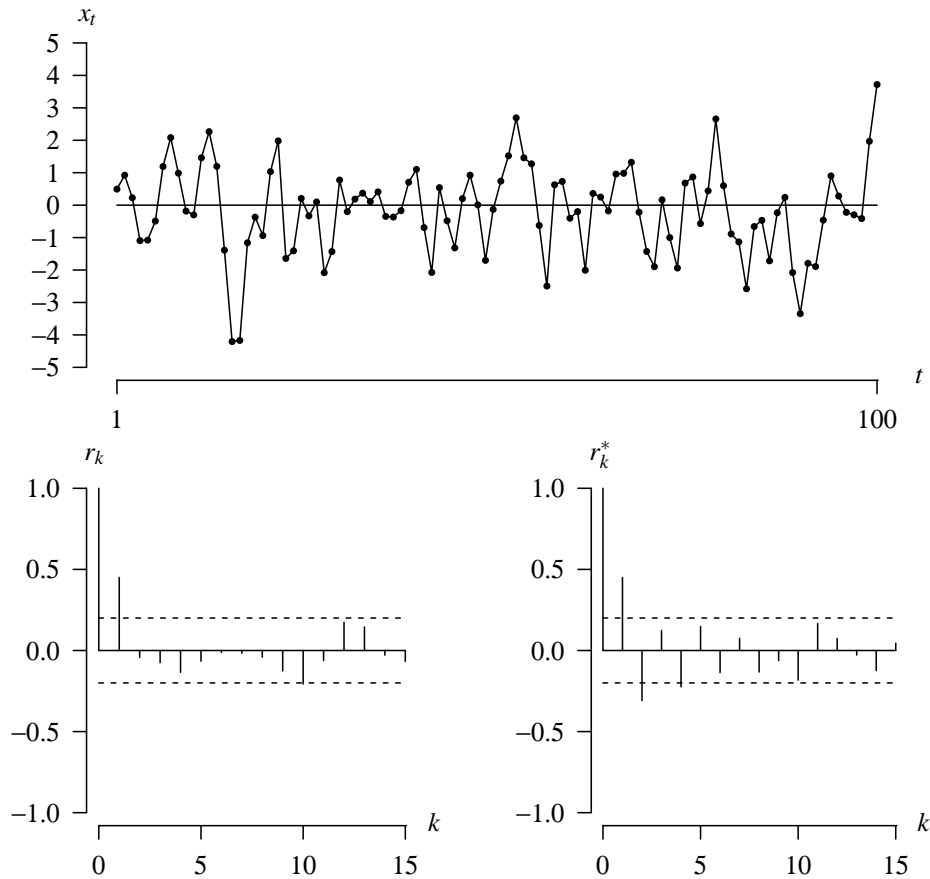


Figure 9.39: Time series plot, r_k , and r_k^* for $n = 100$ simulated values from an MA(1) model.

The time series contains *short runs* above and below the population mean, which are consistent with the statistically significant lag 1 sample autocorrelation function value $r_1 = 0.450$. This value is slightly smaller than the associated population value. The sample autocorrelation function values at lags 2 through 15 do not differ from zero by a statistically significant amount, except for the sample lag 10 autocorrelation function value $r_{10} = -0.207$. This value falls just slightly outside of the 95% confidence bounds

$$\pm \frac{z_{\alpha/2}}{\sqrt{n}} = \pm \frac{z_{0.025}}{\sqrt{100}} = \pm \frac{1.96}{10} = \pm 0.196.$$

This spike in the correlogram is not considered statistically significant because (a) there is nothing about the time series that would indicate that a lag of 10 is a special lag, (b) the spike in the correlogram at lag 10 falls just slightly outside of the confidence bounds, and (c) we expect 1 in 20 of the correlogram spikes to fall outside of the 95% confidence bounds because of random sampling variability, even if the MA(1) model were perfect (as it is in this case). The graphs of $\rho(k)$ and $\rho^*(k)$ mirror their population

counterparts in Figure 9.38. You are encouraged to place these R statements in a `for` loop (of course, with the `set.seed` call outside of the loop) that generates multiple realizations of the MA(1) time series with a call to `Sys.sleep` to provide a short time delay for you to inspect the trio of plots. This will give you a feel for how this MA(1) time series, its correlogram, and its sample partial autocorrelation function vary from one realization to the next.

Another way to think about a realization of an MA(1) model is to make scatterplots of adjacent observations and observations that are two time units apart. The left-hand plot in Figure 9.40 illustrates the positive sample correlation between x_{t-1} and x_t for the realization, which is consistent with the positive lag 1 population autocorrelation. The $n - 1 = 99$ pairs of points plotted are the adjacent values in the realization of the time series, (x_{t-1}, x_t) . The population autocorrelation function cuts off after lag 1 for an MA(1) time series model. This is supported by the right-hand plot in Figure 9.40, which shows the $n - 2 = 98$ pairs (x_{t-2}, x_t) for the realization, which appear to be independent. The regression lines have been added to each plot. The additional R statements below indicate that the p -values for the statistical significance of the slopes associated with the two plots are $p = 9.3 \cdot 10^{-7}$ and $p = 0.65$, respectively.

```
summary(lm(x[2:n] ~ x[1:(n - 1)]))
summary(lm(x[3:n] ~ x[1:(n - 2)]))
```

These p -values indicate that the slope of the line in the left-hand plot differs significantly from zero, while the slope of the line in the right-hand plot does not differ significantly from zero.

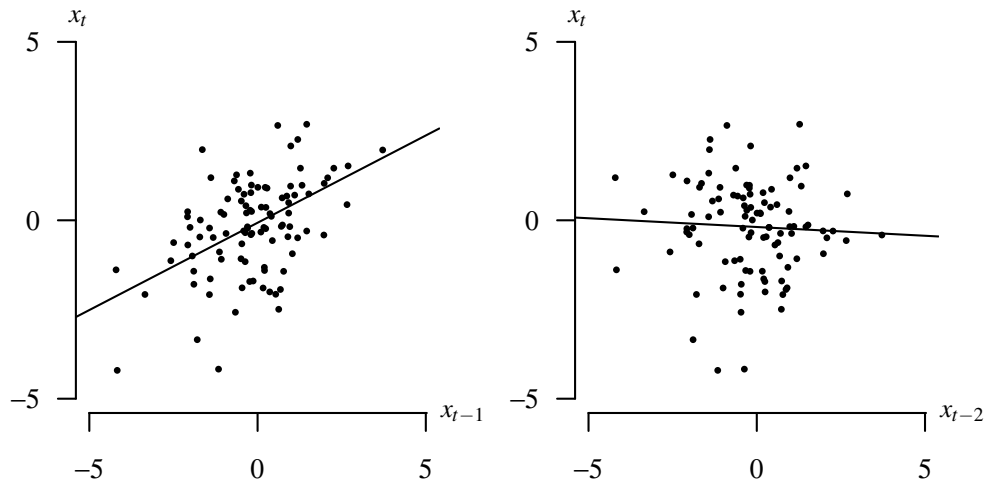


Figure 9.40: Scatterplots of pairs of simulated MA(1) observations.

Example 9.34 Generate a realization of $n = 100$ observations from a shifted MA(1) time series model with $\mu = 20$, $\theta = -9/10$, Gaussian white noise terms, and $\sigma_Z^2 = 1$.

This time series corresponds to the opposite extreme case of the MA(1) because the coefficient $\theta = -0.9$ also (barely) falls in the invertibility region $-1 < \theta < 1$. We again

assume that the Gaussian white noise has population variance $\sigma_Z^2 = 1$, but now include a shift parameter $\mu = 20$. The population lag 1 autocorrelation function value associated with this model is

$$\rho(1) = \frac{\theta}{1 + \theta^2} = \frac{-9/10}{1 + (-9/10)^2} = -\frac{90}{181} \cong -0.4972.$$

So this choice of θ results in a population lag 1 autocorrelation that is very close to its smallest possible value. The R code below simulates $n = 100$ observations from this shifted MA(1) model. This code differs from the previous code in that it avoids the use of a `for` loop, which is a more efficient way to generate observations in R. The time series plot of the realization of $n = 100$ observations and the associated correlogram for $\theta = -0.9$ is given in Figure 9.41.

```
set.seed(37)
n      = 100
z      = rnorm(n + 1)
```

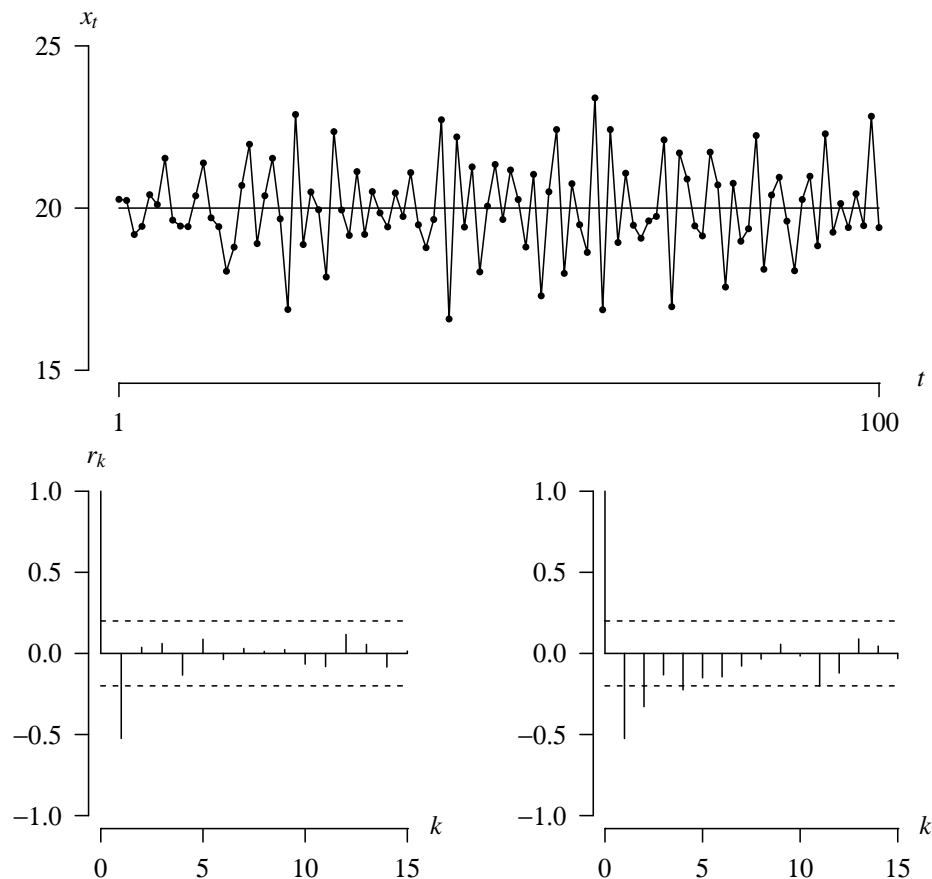


Figure 9.41: Time series plot, r_k , and r_k^* for $n = 100$ simulated values from an MA(1) model.

```

theta = -0.9
x      = z[2:(n + 1)] + theta * z[1:n]
x      = x + 20

```

The time series plot reveals that adjacent observations in the time series tend to be on opposite sides of the population mean, which is consistent with the sample lag 1 autocorrelation $r_1 = -0.524$. The sample autocorrelation function values for lags 2 through 15 do not differ from zero by a statistically significant amount. So the sample autocorrelation cuts off after lag 1 as expected. Furthermore, the sample partial autocorrelation tails out as expected. The `arima.sim` function can also be used to generate a realization of an MA(1) time series model using fewer keystrokes. The R single statement

```
x = 20 + arima.sim(list(ma = -0.9), n = 100)
```

generates 100 values from a shifted MA(1) model with $\theta = -0.9$, $\mu = 20$ and $\sigma_Z^2 = 1$. The default probability distribution for the white noise terms is Gaussian white noise.

Having established the probabilistic properties of the MA(1) model, we now turn to statistical topics, beginning with the estimation of the model parameters.

Parameter Estimation

There are several techniques for estimating the parameters in an MA(1) model; as was the case for the autoregressive models, we look at the method of moments, least squares, and maximum likelihood estimation techniques separately. Parameter estimation is more difficult for moving average models, as numerical methods are typically required to calculate the parameter estimates.

Approach 1: Method of moments. We begin with the shifted MA(1) model from Definition 9.4:

$$X_t = \mu + Z_t + \theta Z_{t-1}.$$

We want to estimate the three unknown parameters μ , θ , and σ_Z^2 from an observed time series $\{x_t\}$. In the case of the shifted MA(1) model, we match the population and sample (a) first-order moments, (b) second-order moments, and (c) lag 1 autocorrelation. These will be written with upper case values X_t although these will be replaced with numeric values x_t for a particular observed time series. Placing the population moments of the left-hand side of the equation and the associated sample moments on the right-hand side of the equation results in three equations in three unknowns:

$$\begin{aligned}
 E[X_t] &= \frac{1}{n} \sum_{t=1}^n X_t \\
 E[X_t^2] &= \frac{1}{n} \sum_{t=1}^n X_t^2 \\
 \rho(1) &= r_1
 \end{aligned}$$

or

$$\begin{aligned}
 \mu &= \bar{X} \\
 V[X_t] + E[X_t]^2 &= \gamma(0) + \mu^2 = (1 + \theta^2) \sigma_Z^2 + \mu^2 = \frac{1}{n} \sum_{t=1}^n X_t^2 \\
 \frac{\theta}{1 + \theta^2} &= r_1.
 \end{aligned}$$

The third equation is a quadratic equation in θ :

$$r_1\theta^2 - \theta + r_1 = 0,$$

which corresponds to the parabolas in Figure 9.36, except that $\rho(1)$ is replaced by r_1 . Using the quadratic formula, the product of the two roots

$$\theta = \frac{1 \pm \sqrt{1 - 4r_1^2}}{2r_1}$$

equals 1, so the root that falls within the invertibility region $-1 < \hat{\theta} < 1$ should be chosen. Some algebra shows that this can be done by always selecting the minus in the $\pm$ portion of the formula. Once the point estimator $\hat{\theta}$ has been chosen, the first two equations can be solved as

$$\hat{\mu} = \bar{X} \quad \text{and} \quad \hat{\sigma}_Z^2 = \frac{(1/n) \sum_{t=1}^n X_t^2 - \hat{\mu}^2}{1 + \hat{\theta}^2}.$$

It appears that we have closed-form solutions to the method of moments estimators, but there is a subtle wrinkle in this derivation. Because of random sampling variability there is a chance that the lag 1 sample autocorrelation r_1 might be greater than $1/2$ or less than $-1/2$, even if the population time series model truly is a shifted MA(1) model satisfying the invertibility criterion $-1 < \theta < 1$. In this case the quadratic formula yields complex roots. So the method of moments parameter estimation approach is recommended for initial parameter estimates only if the constraint $|r_1| < 1/2$ stated in the result that follows is met.

Theorem 9.30 The method of moments estimators for a shifted MA(1) model from a time series $x_1, x_2, \dots, x_n$ with $|r_1| < 1/2$ are

$$\hat{\mu} = \bar{X}, \quad \hat{\theta} = \frac{1 - \sqrt{1 - 4r_1^2}}{2r_1}, \quad \hat{\sigma}_Z^2 = \frac{(1/n) \sum_{t=1}^n X_t^2 - \hat{\mu}^2}{1 + \hat{\theta}^2}.$$

Thus, the method of moments point estimators in Theorem 9.30 should only be used for determining *initial* estimators of μ , θ , and σ_Z^2 from $x_1, x_2, \dots, x_n$ when the following criteria are met:

- the number of observations in the time series is greater than about $n = 60$ or $n = 70$,
- the time series appears to be stationary,
- the sample autocorrelation function has a single statistically significant spike at lag 1,
- the sample partial autocorrelation function tails out, and
- $-1/2 < r_1 < 1/2$.

The method of moments estimators are generally used to find initial point estimates for the parameters in an MA(1) model, which are subsequently used in an iterative scheme to find the least squares or maximum likelihood estimators. This will be illustrated next on a time series consisting of chemical yields.

Example 9.35 Consider the time series consisting of the production record of $n = 210$ consecutive yield values in a chemical production process in Table 9.14. The entries are read row wise. Determine an appropriate model for this time series and find the method of moments initial estimates for the parameters.

The first step is to plot the time series, sample autocorrelation function, and sample partial autocorrelation function. The following R code uses the `plot.ts`, `acf`, and `pacf` functions to produce the graphs of the time series and correlogram given in Figure 9.42. The raw time series values are stored in a file named `yields.dat`.

```
x = scan("yields.dat")
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(x)
acf(x)
pacf(x)
```

The time series appears to be stationary. The time series frequently jumps from one side of the sample mean $\bar{x} = 84.1$ to the other, indicating a negative sample correlation between adjacent values in the time series. The sample lag 1 autocorrelation $r_1 = -0.289$ falls outside of the 95% confidence bounds, so we can conclude that there is a single statistically significant spike at lag 1. There are marginally statistically significant spikes at lags 2 and 6, which we will attribute to random sampling variability. The sample partial autocorrelation function has statistically significant spikes at lags 1, 2, and 3 which are negative and decrease in magnitude. Since $-1/2 < r_1 < 1/2$, the time series plot and the shapes of r_k and r_k^* indicate that the shifted MA(1) model

$$X_t = \mu + Z_t + \theta Z_{t-1}$$

with a negative value of θ might be appropriate. The R statements that follow are used to estimate the model parameters using Theorem 9.30.

85.5	81.7	80.6	84.7	88.2	84.9	81.8	84.9	85.2	81.9	89.4	79.0	81.4	84.8
85.9	88.0	80.3	82.6	83.5	80.2	85.2	87.2	83.5	84.3	82.9	84.7	82.9	81.5
83.4	87.7	81.8	79.6	85.8	77.9	89.7	85.4	86.3	80.7	83.8	90.5	84.5	82.4
86.7	83.0	81.8	89.3	79.3	82.7	88.0	79.6	87.8	83.6	79.5	83.3	88.4	86.6
84.6	79.7	86.0	84.2	83.0	84.8	83.6	81.8	85.9	88.2	83.5	87.2	83.7	87.3
83.0	90.5	80.7	83.1	86.5	90.0	77.5	84.7	84.6	87.2	80.5	86.1	82.6	85.4
84.7	82.8	81.9	83.6	86.8	84.0	84.2	82.8	83.0	82.0	84.7	84.4	88.9	82.4
83.0	85.0	82.2	81.6	86.2	85.4	82.1	81.4	85.0	85.8	84.2	83.5	86.5	85.0
80.4	85.7	86.7	86.7	82.3	86.4	82.5	82.0	79.5	86.7	80.5	91.7	81.6	83.9
85.6	84.8	78.4	89.9	85.0	86.2	83.0	85.4	84.4	84.5	86.2	85.6	83.2	85.7
83.5	80.1	82.2	88.6	82.0	85.0	85.2	85.3	84.3	82.3	89.7	84.8	83.1	80.6
87.4	86.8	83.5	86.2	84.1	82.3	84.8	86.6	83.5	78.1	88.8	81.9	83.3	80.0
87.2	83.3	86.6	79.5	84.1	82.2	90.8	86.5	79.7	81.0	87.2	81.6	84.4	84.4
82.2	88.9	80.9	85.1	87.1	84.0	76.5	82.7	85.1	83.3	90.4	81.0	80.3	79.8
89.0	83.7	80.9	87.3	81.1	85.6	86.6	80.0	86.6	83.3	83.1	82.3	86.7	80.2

Table 9.14: Production record of $n = 210$ consecutive chemical yields.

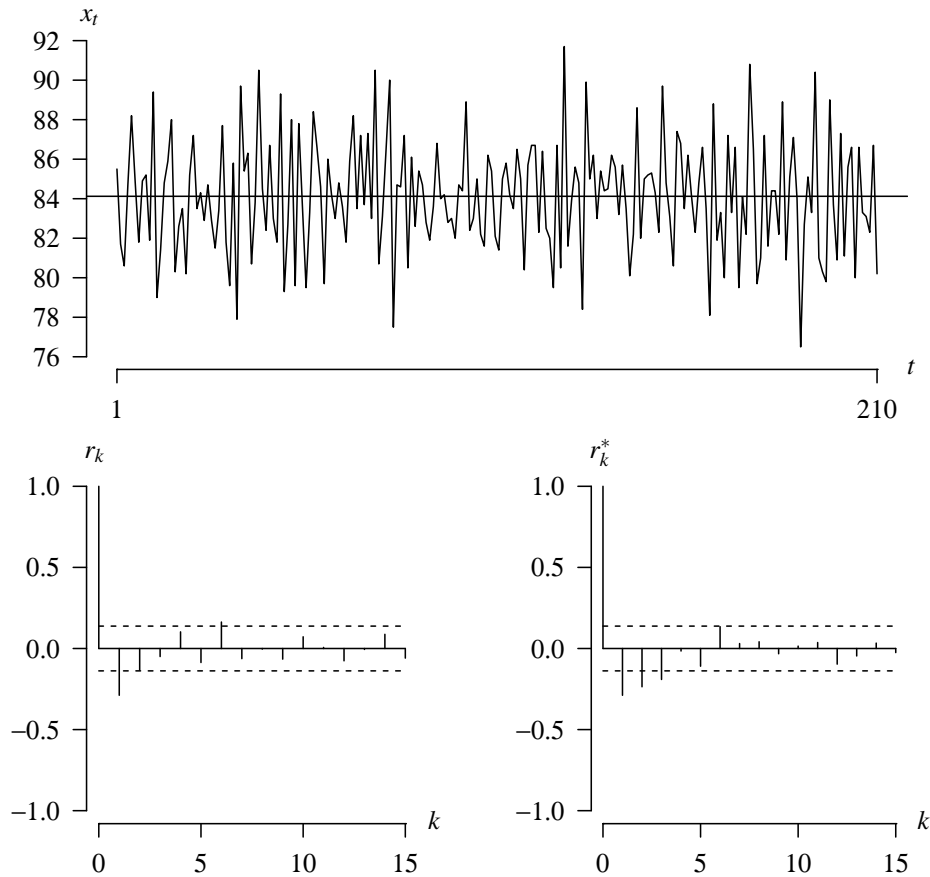


Figure 9.42: Time series plot, r_k , and r_k^* for $n = 210$ chemical yields.

```
x      = scan("yields.dat")
r1     = acf(x, plot = FALSE)$acf[2]
muhat  = mean(x)
thetahat = (1 - sqrt(1 - 4 * r1 ^ 2)) / (2 * r1)
sigma2hat = (mean(x ^ 2) - muhat ^ 2) / (1 + thetahat ^ 2)
```

This code yields the following method of moments point estimators for the three parameters:

$$\hat{\mu} = 84.1 \quad \hat{\theta} = -0.318 \quad \hat{\sigma}^2 = 7.50.$$

This value of $\hat{\theta}$ falls within the invertibility region $-1 < \theta < 1$ for the shifted MA(1) time series model.

Approach 2: Least squares. Consider the shifted stationary MA(1) model

$$X_t = \mu + Z_t + \theta Z_{t-1}$$

from Theorem 9.29. For least squares estimation, we first establish the sum of squares S as a function of the parameters μ and θ . Numerical methods are required to determine the least squares estimators of μ and θ . Once these least squares estimators have been determined, the population variance of the white noise σ_Z^2 will be estimated.

Solving the shifted MA(1) model defining formula for Z_t results in

$$Z_t = X_t - \mu - \theta Z_{t-1}.$$

Seeding this recursive formula with $Z_0 = 0$ gives the residuals

$$\begin{aligned} Z_1 &= X_1 - \mu \\ Z_2 &= X_2 - \mu - \theta Z_1 \\ Z_3 &= X_3 - \mu - \theta Z_2 \\ &\vdots \\ Z_n &= X_n - \mu - \theta Z_{n-1}. \end{aligned}$$

Thus, the sum of squared errors is

$$S = \sum_{t=1}^n Z_t^2 = (X_1 - \mu)^2 + \sum_{t=2}^n (X_t - \mu - \theta Z_{t-1})^2.$$

Numerical methods are required to find the parameter estimates. This will be illustrated next for the time series of chemical yields.

Example 9.36 Find the least squares estimators for the time series of chemical yields given in Example 9.35.

The R code below uses the `optim` function to perform a search to find the least squares parameter estimates for μ and θ . The first statement reads the time series values into the vector `x`. The second statement uses the `length` function to calculate the number of observations in the time series. The third statement defines the function `s`, which calculates the sum of squares. The last statement calls the `optim` function with the method of moments estimators as initial parameter estimates as its first argument. The `optim` function *minimizes* the function in its second argument by default.

```
x = scan("yields.dat")
n = length(x)
s = function(parameters) {
  mu = parameters[1]
  theta = parameters[2]
  z = numeric(n)
  z[1] = x[1] - mu
  for (t in 2:n) z[t] = x[t] - mu - theta * z[t - 1]
  sum(z ^ 2)
}
optim(c(84.1, -0.318), s, method = "L-BFGS-B")
```

The `optim` function is being called in this case to perform a two-dimensional search without any derivative information being supplied. In order to visualize what the `optim`

function is up against, picture yourself standing blindfolded on the side a mountain. The position where you are standing corresponds to the initial estimates for the parameters. In order to find the *least* squares parameter estimates, you want to take steps that lead you to the bottom of the valley, where the height corresponds to the sum of squares of the residuals. For *maximum* likelihood estimation, you want to take steps that lead you to the peak of the mountain, where the height corresponds to the likelihood function. But you are not given any gradient information from the function about the best direction to proceed for your next step. Regardless of the argument selected in the `method` argument, the internal algorithm in the `optim` function converges to roughly the same parameter estimates, as shown in Table 9.15. All five of the methods round to $\hat{\mu} = 84.13$ and $\hat{\theta} = -0.483$. The columns in Table 9.15 give the method, the number of calls to the function, the number of calls to evaluate the gradient, the least squares parameter estimates of μ , the least squares parameter estimates of θ , and the associated sums of squares S .

Method	Function	Gradient	$\hat{\mu}$	$\hat{\theta}$	S
Nelder-Mead	67	NA	84.12932	-0.48259	1484.989
BFGS	22	6	84.12942	-0.48261	1484.989
CG	180	23	84.12942	-0.48261	1484.989
L-BFGS-B	17	17	84.12942	-0.48261	1484.989
SANN	10000	NA	84.13053	-0.48269	1484.990

Table 9.15: MA(1) least squares parameter estimates for the chemical yields.

There is a large difference between the method of moments estimator of θ , which was $\hat{\theta} = -0.318$ from Example 9.35, and the least squares estimator of θ , which is $\hat{\theta} = -0.483$ from Table 9.15. This is an instance of why the method of moments estimators are only used for initial estimates for iterative schemes for finding least squares estimates or maximum likelihood estimates. The population variance of the white noise can be estimated using the method of moments formula as

$$\hat{\sigma}_Z^2 = \frac{(1/n) \sum_{t=1}^n X_t^2 - \hat{\mu}^2}{1 + \hat{\theta}^2} = 5.61.$$

Approach 3: Maximum likelihood estimation. We use the `arima` function to do the heavy lifting with respect to the estimation of the parameters in the MA(1) time series model via maximum likelihood. In addition to the point estimates, confidence intervals are based on the asymptotic distribution of the maximum likelihood estimator $\hat{\theta}$, which for large values of n is

$$V[\hat{\theta}] \cong \frac{1 - \theta^2}{n}.$$

So when the parameter θ is estimated by maximum likelihood from a time series, an asymptotically exact $100(1 - \alpha)\%$ confidence interval for θ is given in the result below. It is based on the consistency and the asymptotic normality of maximum likelihood estimators, which in this case implies that

$$\hat{\theta} \stackrel{a}{\sim} N\left(\theta, \frac{1 - \theta^2}{n}\right).$$

Replacing θ by its maximum likelihood estimator in the variance yields the following result.

Theorem 9.31 For an invertible MA(1) time series model, an asymptotically exact two-sided $100(1 - \alpha)\%$ confidence interval for θ is given by

$$\hat{\theta} - z_{\alpha/2} \sqrt{\frac{1 - \hat{\theta}^2}{n}} < \theta < \hat{\theta} + z_{\alpha/2} \sqrt{\frac{1 - \hat{\theta}^2}{n}},$$

where $\hat{\theta}$ is the maximum likelihood estimator and $z_{\alpha/2}$ is the $1 - \alpha/2$ fractile of the standard normal distribution.

The formula for the confidence interval from Theorem 9.31 will be illustrated for the chemical yield data from the previous two examples.

Example 9.37 Find the maximum likelihood estimators for μ , θ , and σ_Z^2 for fitting a shifted MA(1) time series model to the time series of chemical yields from Example 9.35. Give a 95% confidence interval for θ and μ .

The following R code uses the `arima` function to calculate the maximum likelihood estimators for μ , θ , and σ_Z^2 , estimate the variance–covariance matrix of the standard errors, and display the standard errors of the estimators.

```
x = scan("yields.dat")
fit = arima(x, order = c(0, 0, 1), include.mean = TRUE, method = "ML")
fit$coef
fit$var.coef
fit$sigma2
sqrt(fit$var.coef[1, 1])
sqrt(fit$var.coef[2, 2])
```

The resulting fitted MA(1) time series model to significant digits via maximum likelihood estimation is

$$X_t = \hat{\mu} + Z_t + \hat{\theta}Z_{t-1}$$

or

$$X_t = 84.13 + Z_t - 0.480Z_{t-1},$$

(0.0958) (0.0667)

where Z_t is white noise with estimated population variance $\hat{\sigma}_Z^2 = 7.071$. The numbers in parentheses just below the parameter estimates are the estimated standard errors of the associated parameter estimates. Using the standard errors from `arima`, the associated approximate 95% confidence intervals are

$$\begin{aligned} -0.611 &< \theta < -0.349 \\ 83.94 &< \mu < 84.32. \end{aligned}$$

The 95% confidence interval for θ using Theorem 9.31 is slightly narrower than that produced by the `arima` function: $-0.599 < \theta < -0.362$.

The parameter estimates using the method of moments, least squares, and maximum likelihood estimation from the previous three examples are summarized in Table 9.16. Notice that the least

Method	$\hat{\mu}$	$\hat{\theta}$	$\hat{\sigma}_Z^2$
Method of moments	84.1	-0.318	7.50
Ordinary least squares	84.1	-0.483	5.61
Maximum likelihood estimation	84.1	-0.480	7.07

Table 9.16: Point estimators for the MA(1) parameters for the $n = 210$ chemical yields.

squares and maximum likelihood estimates of θ differ significantly from the associated method of moments estimator of θ .

In the interest of brevity, we leave the model assessment, model selection, and forecasting steps of the process for the chemical yields time series as an exercise. The derivations for these procedures follow along the same lines as those for the autoregressive models from the previous section.

This subsection has introduced the MA(1) time series model. The key results for an MA(1) model are listed below.

- The standard MA(1) model can be written algebraically and with the backshift operator B as

$$X_t = Z_t + \theta Z_{t-1} \quad \text{and} \quad X_t = \theta(B)Z_t,$$

where $Z_t \sim WN(0, \sigma_Z^2)$, $\sigma_Z^2 > 0$, and $\theta(B) = 1 + \theta B$ (Definition 9.4).

- The shifted MA(1) model can be written algebraically and with the backshift operator B as (Theorem 9.29)

$$X_t = \mu + Z_t + \theta Z_{t-1} \quad \text{and} \quad X_t = \mu + \theta(B)Z_t.$$

- The MA(1) model is stationary for all finite real-valued parameters θ and σ_Z^2 (Theorem 9.25).
- The MA(1) model is invertible when $-1 < \theta < 1$ (Theorem 9.27).
- The MA(1) model can be written as an AR(∞) model when $-1 < \theta < 1$ as (Theorem 9.26)

$$X_t = Z_t + \theta X_{t-1} - \theta^2 X_{t-2} + \theta^3 X_{t-3} - \cdots$$

- The MA(1) model lag 1 population autocorrelation is $\rho(1) = \theta / (1 + \theta^2)$, and $\rho(k) = 0$ for $k = 2, 3, \dots$ (Theorem 9.25). The lag 1 population autocorrelation satisfies the inequality $-1/2 \leq \rho(1) \leq 1/2$ (Figure 9.37).
- The MA(1) lag k population partial autocorrelation for $-1 < \theta < 1$ is

$$\rho^*(k) = \frac{(-1)^{k+1} \theta^k (1 - \theta^2)}{1 - \theta^{2(k+1)}}$$

for $k = 1, 2, \dots$ (Theorem 9.28).

- A time series of $n + 1$ white noise values $Z_0, Z_1, Z_2, \dots, Z_n$ can be converted to n simulated observations $X_1, X_2, \dots, X_n$ by using the MA(1) defining formula $X_t = Z_t + \theta Z_{t-1}$.
- The parameters in the MA(1) model can be estimated via the method of moments, least squares estimation, and maximum likelihood estimation.

9.2.2 The MA(2) Model

The additional term in the MA(2) model gives it increased flexibility over the associated MA(1) model.

Definition 9.5 A second-order moving average time series model, denoted by MA(2), for the time series $\{X_t\}$ is defined by

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2},$$

where θ_1 and θ_2 are real-valued parameters and $\{Z_t\}$ is a time series of white noise:

$$Z_t \sim WN(0, \sigma_Z^2).$$

An observed value in the time series, X_t , is given by the current white noise term, plus the parameter θ_1 multiplied by the white noise term from one time period ago, plus the parameter θ_2 multiplied by the white noise term from two time periods ago. So there are three parameters that define an MA(2) model: the coefficients θ_1 and θ_2 , and the population variance of the white noise σ_Z^2 . As was the case of the MA(1) model, some authors use a $-$ rather than a $+$ between three terms on the right-hand side of the model.

The probabilistic properties and statistical methods associated with an MA(2) model are straightforward generalizations of those properties and methods for the MA(1) model. Rather than deriving these results from first principles, we simply state several of these results without proof and then conduct a Monte Carlo simulation experiment which highlights issues that arise in model selection.

- The standard MA(2) model can be written algebraically and with the backshift operator B as

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} \quad \text{and} \quad X_t = \theta(B)Z_t,$$

where $Z_t \sim WN(0, \sigma_Z^2)$, $\sigma_Z^2 > 0$, and $\theta(B) = 1 + \theta_1 B + \theta_2 B^2$.

- The shifted MA(2) model can be written algebraically and with the backshift operator B as

$$X_t = \mu + Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} \quad \text{and} \quad X_t = \mu + \theta(B)Z_t.$$

- MA(2) models are stationary for all finite, real-valued parameters μ , θ_1 , θ_2 , and σ_Z^2 .
- Just as the stationarity region for the AR(2) model has a triangular shape, the invertibility region for the MA(2) model also has a triangular shape defined by the three constraints

$$\theta_1 + \theta_2 > -1, \quad \theta_2 - \theta_1 > -1, \quad \theta_2 < 1.$$

This region is an upside-down version of the region for an AR(2) time series model depicted in Figure 9.12. In other words, the triangles are equivalent when reflected vertically about the origin.

- The population autocovariance function is

$$\gamma(k) = \begin{cases} (1 + \theta_1^2 + \theta_2^2) \sigma_Z^2 & k = 0 \\ (\theta_1 + \theta_1 \theta_2) \sigma_Z^2 & k = 1 \\ \theta_2 \sigma_Z^2 & k = 2 \\ 0 & k = 3, 4, \dots \end{cases}$$

- The population autocorrelation function is

$$\rho(k) = \begin{cases} 1 & k = 0 \\ (\theta_1 + \theta_1\theta_2) / (1 + \theta_1^2 + \theta_2^2) & k = 1 \\ \theta_2 / (1 + \theta_1^2 + \theta_2^2) & k = 2 \\ 0 & k = 3, 4, \dots \end{cases}$$

- The population partial autocorrelation function of an MA(2) model can be determined by using the defining formula from Definition 7.8.
- A simulated realization $X_1, X_2, \dots, X_n$ of a time series from an MA(2) model is generated by the following algorithm.

```

t ← -1
generate  $Z_t \sim WN(0, \sigma_Z^2)$ 
t ← t + 1
generate  $Z_t \sim WN(0, \sigma_Z^2)$ 
while (t < n)
  t ← t + 1
  generate  $Z_t \sim WN(0, \sigma_Z^2)$ 
   $X_t \leftarrow Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2}$ 

```

- The parameters of an MA(2) time series model can be estimated by the method of moments, least squares, and maximum likelihood estimation. As shown in the next example, the `arma` function can be used in R to calculate these parameter estimates.

The previous subsections have analyzed n observed values of a time series in order to determine an AR(p) or MA(q) model which adequately describes the probabilistic mechanism governing the observed time series. Instead of following this same pattern, we instead conduct a Monte Carlo simulation experiment that highlights weaknesses in the model selection process.

Example 9.38 Consider a standard (unshifted) MA(2) model with parameters $\theta_1 = 0.4$, $\theta_2 = 0.6$, and $\sigma_Z^2 = 1$. For a realization of $n = 60$ observations from this time series fitted by maximum likelihood estimation, use Monte Carlo simulation to estimate the probability that the correct model is identified if the AIC criterion is used to determine the correct model.

This Monte Carlo simulation experiment answers an important question in time series analysis. If we have just a single realization of a time series (this is often the case in practice) which is governed by an approximately ARMA model, what is the probability that we correctly identify the p and q values associated with the ARMA(p, q) time series model which generated the observations?

The MA(2) model is the population time series model in this particular simulation experiment. The choice of parameters $\theta_1 = 0.4$ and $\theta_2 = 0.6$ falls in the invertibility region, so this particular population MA(2) model is both stationary and invertible. Furthermore, the parameters $\theta_1 = 0.4$ and $\theta_2 = 0.6$ have been chosen so that the population autocorrelation function is

$$\rho(k) = \begin{cases} 1 & k = 0 \\ (0.4 + 0.4 \cdot 0.6) / (1 + 0.4^2 + 0.6^2) & k = 1 \\ 0.6 / (1 + 0.4^2 + 0.6^2) & k = 2 \\ 0 & k = 3, 4, \dots \end{cases}$$

or

$$\rho(k) = \begin{cases} 1 & k = 0 \\ 0.42 & k = 1 \\ 0.39 & k = 2 \\ 0 & k = 3, 4, \dots \end{cases}$$

As anticipated, the population autocorrelation cuts off after lag 2. So we expect that the first two values in the sample autocorrelation function computed from a realization of this time series model, r_1 and r_2 , will be statistically significant, and the others will fall between the confidence bounds $\pm 1.96/\sqrt{60} = \pm 0.25$. The coefficients θ_1 and θ_2 have been chosen so that the first two values in the population autocorrelation function, $\rho(1) = 0.42$ and $\rho(2) = 0.39$, both fall outside of the confidence bounds $\pm 1.96/\sqrt{60} = \pm 0.25$ associated with the sample autocorrelation function. This choice of parameters has been made to give the ARMA modeling procedure a good chance of correctly identifying the underlying population MA(2) time series model.

The R code below generates realizations of 1000 time series of length $n = 60$ from an MA(2) model with $\theta_1 = 0.4$, $\theta_2 = 0.6$, and $\sigma_Z^2 = 1$. The `arma.sim` function is used to generate each realization, and the simulated values are placed in the vector `x`. The two inner nested `for` loops fit all ARMA(p, q) models to the simulated values, for $p = 0, 1, 2, \dots, 5$ and $q = 0, 1, 2, \dots, 5$ using the `arma` function. The AIC for each of the fitted models are stored in the `a` matrix. Finally, the `which.min` function is used to determine which of the $6 \cdot 6 = 36$ models has the lowest AIC value.

```
set.seed(3)
nrep = 1000
a = matrix(0, 6, 6)
r = matrix(0, 6, 6)
for (i in 1:nrep) {
  x = arma.sim(list(ma = c(0.4, 0.6)), n = 60)
  for (p in 0:5)
    for (q in 0:5)
      a[p + 1, q + 1] = arma(x, order = c(p, 0, q), method = "ML")$aic
  j = which.min(a)
  r[j] = r[j] + 1
  print(a)
  print(j)
  print(r)
}
100 * r / nrep
```

Table 9.17 contains the estimated probabilities of the selection of the various models expressed as percents. The good news is that the MA(2) model is the one that is chosen most often of the 36 models based on the AIC criterion. The associated bad news is that the MA(2) model is chosen less than half of the time. So in the practical case of analyzing a single time series of $n = 60$ values, there is a better than 0.5 probability that this procedure will identify the *wrong* population time series model. This illustrates the unwelcome effect of random sampling variability in model selection.

Bear in mind that the estimated probabilities given in Table 9.17 only apply to the use of the AIC criterion and the parameter settings $\theta_1 = 0.4$, $\theta_2 = 0.6$, $\sigma_Z^2 = 1$, and $n = 60$.

	$q = 0$	$q = 1$	$q = 2$	$q = 3$	$q = 4$	$q = 5$
$p = 0$	0.2%	0.0%	40.1%	4.9%	2.4%	1.9%
$p = 1$	0.7%	0.0%	3.1%	6.0%	1.4%	1.6%
$p = 2$	0.7%	0.9%	3.7%	1.3%	5.3%	1.5%
$p = 3$	3.2%	1.6%	2.6%	0.7%	1.9%	2.2%
$p = 4$	0.9%	0.9%	2.0%	1.4%	1.5%	1.1%
$p = 5$	0.6%	0.6%	0.8%	0.7%	1.0%	0.6%

Table 9.17: Estimated probabilities of selection based on AIC criterion.

Changing any one of these parameters will alter the probabilities. The purpose of this example is to highlight the pitfalls associated with fitting a time series model to a single realization of n time series values. The probability of an incorrect selection is high, and this is an argument for collecting a longer time series when possible. In addition, if a time series is collected periodically (for example, n values collected annually), then the fits to various realizations should be compared.

To summarize the models considered so far in this chapter, the AR(1), AR(2), MA(1), and MA(2) models are parsimonious in the sense that they have significant explanatory power with few parameters. By deriving the population autocorrelation function and partial autocorrelation function for these models, we now possess an inventory of possible shapes that guide us toward one particular time series model or another. Figure 9.43 gives examples of these shapes for various values of the parameters.

9.2.3 The MA(q) Model

The MA(1) and MA(2) models introduced in the previous two subsections generalize to the MA(q) model defined in this section.

Definition 9.6 A moving average time series model with q terms, denoted by MA(q), for the time series $\{X_t\}$ is defined by

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots + \theta_q Z_{t-q},$$

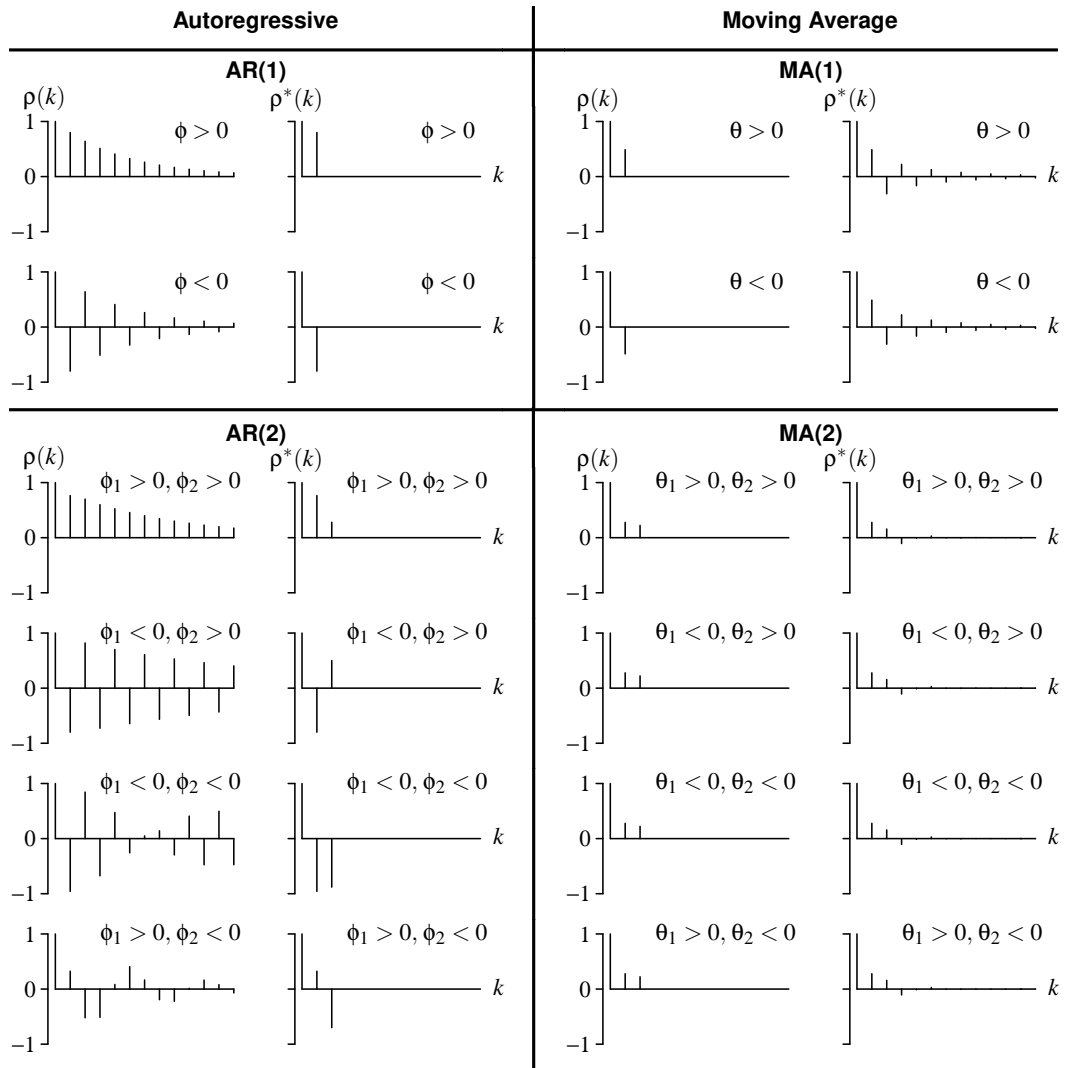
where $\theta_1, \theta_2, \dots, \theta_q$ are real-valued parameters and $\{Z_t\}$ is a time series of white noise:

$$Z_t \sim WN(0, \sigma_Z^2).$$

An observed value in the time series, X_t , is given by the current white noise term plus a linear combination of the q previous white noise terms. So there are $q + 1$ parameters that define an MA(q) model: the coefficients $\theta_1, \theta_2, \dots, \theta_q$, and the population variance of the white noise σ_Z^2 . As was the case of the MA(1) and MA(2) models, some authors use a $-$ rather than a $+$ between terms on the right-hand side of the model.

The probabilistic properties and statistical methods associated with an MA(q) model are determined in the usual fashion. Here are several of these results stated without proof.

- The population mean and variance of X_t are easily calculated by taking the expected value and

Figure 9.43: Characteristic shapes of $\rho(k)$ and $\rho^*(k)$ for AR(1), AR(2), MA(1), and MA(2) models.

the population variance of both sides of the equation given in Definition 9.6:

$$E[X_t] = E[Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots + \theta_q Z_{t-q}] = 0$$

and

$$V[X_t] = V[Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots + \theta_q Z_{t-q}] = (1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2) \sigma_Z^2.$$

- The standard MA(q) model can be written algebraically and with the backshift operator B as

$$X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots + \theta_q Z_{t-q} \quad \text{and} \quad X_t = \theta(B)Z_t,$$

where $Z_t \sim WN(0, \sigma_Z^2)$, $\sigma_Z^2 > 0$, and $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$.

- The shifted MA(q) model can be written algebraically and with the backshift operator B as

$$X_t = \mu + Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots + \theta_q Z_{t-q} \quad \text{and} \quad X_t = \mu + \theta(B)Z_t.$$

- MA(q) models are stationary for all finite, real-valued parameters $\mu, \theta_1, \theta_2, \dots, \theta_q$, and σ_Z^2 .
- MA(q) models are invertible when the q roots of the characteristic equation

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q = 0$$

all lie outside of the unit circle in the complex plane.

- The population autocovariance function is

$$\gamma(k) = \begin{cases} (1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2) \sigma_Z^2 & k = 0 \\ (\theta_1 + \theta_1 \theta_2 + \theta_2 \theta_3 + \cdots + \theta_{q-1} \theta_q) \sigma_Z^2 & k = 1 \\ (\theta_2 + \theta_1 \theta_3 + \theta_2 \theta_4 + \cdots + \theta_{q-2} \theta_q) \sigma_Z^2 & k = 2 \\ \vdots & \vdots \\ \theta_q \sigma_Z^2 & k = q \\ 0 & k = q, q+1, \dots \end{cases}$$

This can be written more compactly as

$$\gamma(k) = \begin{cases} (\theta_k + \theta_1 \theta_{k+1} + \theta_2 \theta_{k+2} + \cdots + \theta_{q-k} \theta_q) \sigma_Z^2 & k = 0, 1, 2, \dots, q \\ 0 & k = q, q+1, \dots \end{cases}$$

where $\theta_0 = 1$.

- The population autocorrelation function is

$$\rho(k) = \begin{cases} (\theta_k + \theta_1 \theta_{k+1} + \theta_2 \theta_{k+2} + \cdots + \theta_{q-k} \theta_q) / (1 + \theta_1^2 + \cdots + \theta_q^2) & k = 0, 1, \dots, q \\ 0 & k = q, q+1, \dots \end{cases}$$

As expected, the population autocorrelation function cuts off after lag q .

- The population partial autocorrelation function of an MA(q) model can be determined by using the defining formula from Definition 7.8.
- A simulated realization $X_1, X_2, \dots, X_n$ of a time series from an MA(q) model is generated by the following algorithm.

```
generate  $Z_{-(q-1)}, Z_{-(q-2)}, \dots, Z_0 \sim WN(0, \sigma_Z^2)$ 
 $t \leftarrow 0$ 
while ( $t < n$ )
   $t \leftarrow t + 1$ 
  generate  $Z_t \sim WN(0, \sigma_Z^2)$ 
   $X_t \leftarrow Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots + \theta_q Z_{t-q}$ 
```

- The parameters of an MA(q) time series model can be estimated by the method of moments, least squares, and maximum likelihood estimation. The `arima` function can be used in R to calculate these parameter estimates for particular values of a time series $x_1, x_2, \dots, x_n$.

Table 9.18 shows some of the symmetry between autoregressive and moving average models. When one aspect of the time series model is easier to derive for one of the models, it is often more difficult to derive for the analogous time series model. The population autocorrelation function for an MA(q) model is closed form, for example, but the population autocorrelation function for an AR(p) model requires solving the Yule–Walker equations. As a second example on the statistical side, the least squares estimators for the AR(1) model are closed form, but the least squares estimators for the MA(1) model require numerical methods.

	Autoregressive: AR(p)	Moving Average: MA(q)
Model definition	$\phi(B)X_t = Z_t$	$X_t = \theta(B)Z_t$
Characteristic polynomial	$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$	$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$
Stationarity condition	$\phi(B) = 0$ roots outside of unit circle	always stationary
Invertibility condition	always invertible	$\theta(B) = 0$ roots outside of unit circle
Equivalent model	MA(∞) when stationary	AR(∞) when invertible
General linear model π weights	finite series	infinite series
General linear model ψ weights	infinite series	finite series
Shape of $\rho(k)$	tails out	cuts off after lag q
Shape of $\rho^*(k)$	cuts off after lag p	tails out
Simulating a realization	warm up period needed	no warm up period needed

Table 9.18: AR(p) versus MA(q) models.

9.3 ARMA(p, q) Models

The autoregressive and moving average models outlined in the previous two sections often prove to be inadequate time series models in a particular application. Occasions arise in which the best model for a time series involves both autoregressive and moving average terms. Recall from Definition 8.4 that an ARMA(p, q) time series model with p autoregressive terms and q moving average terms is

$$X_t = \overbrace{\phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p}}^{\text{autoregressive portion}} + \underbrace{Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots + \theta_q Z_{t-q}}_{\text{moving average portion}},$$

where $\{X_t\}$ is the time series of interest, $\{Z_t\}$ is a time series of white noise, $\phi_1, \phi_2, \dots, \phi_p$ are real-valued parameters associated with the AR portion of the model, and $\theta_1, \theta_2, \dots, \theta_q$ are real-valued

parameters associated with the MA portion of the model. The ARMA(p, q) model can be written more compactly as

$$\phi(B)X_t = \theta(B)Z_t,$$

where $\phi(B)$ and $\theta(B)$ are the characteristic polynomials defined by

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$$

and

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

This model on its own is of little practical use because most real-world time series are not centered around $E[X_t] = 0$. Using the compact notation for the ARMA(p, q) time series model, a shift parameter μ is easily added:

$$\phi(B)(X_t - \mu) = \theta(B)Z_t.$$

So there are $p + q + 2$ parameters that define a shifted ARMA(p, q) time series model: the p autoregressive coefficients $\phi_1, \phi_2, \dots, \phi_p$, the q moving average coefficients $\theta_1, \theta_2, \dots, \theta_q$, the shift parameter μ , and the population variance of the white noise σ_Z^2 .

Recall from Table 9.7 in Example 9.20 that the ARMA(1, 1) model fitted by maximum likelihood estimation gave a slightly lower AIC than the associated AR(2) model when applied to the Lake Huron level time series. This section will consist of one long example that concerns the fitting and assessing this ARMA(1, 1) model to determine whether it is an adequate model for the Lake Huron levels. Rather than deriving all of the probabilistic properties and statistical methods for the ARMA(1, 1) model, the `arma` function in R will be used to perform the fitting, leaving the details to the reader. By default, the `arma` function (*a*) ignores external regressor variables, (*b*) ignores seasonal variation, (*c*) includes a shift parameter μ , (*d*) uses the same parameterization for the ARMA(p, q) process as that used in this text, (*e*) transforms the AR parameters $\phi_1, \phi_2, \dots, \phi_p$ if necessary so that they stay in the stationarity region, and (*f*) uses a conditional sum of squares method as initial parameter estimates, then returns the maximum likelihood estimators.

Example 9.39 Fit the ARMA(1, 1) model to the $n = 98$ annual Lake Huron levels from 1875–1972 described in Example 9.14. Assess the model adequacy of the fit and predict the level of Lake Huron for the next five years (1973–1977).

The first R statement below fits the ARMA(1, 1) model to the Lake Huron levels. The next four statements extract the estimated coefficients, estimated white noise variance, estimated variance–covariance matrix of the coefficients, and the residuals.

```
fit          = arima(LakeHuron, order = c(1, 0, 1))
coefficients = fit$coef
variance     = fit$sigma2
varianc cov  = fit$var.coef
residuals    = fit$residuals
```

The parameter estimates (to five-digit accuracy that is inherent in the time series values) are

$$\hat{\phi}_1 = 0.74490 \quad \hat{\theta}_1 = 0.32059 \quad \hat{\mu} = 579.06 \quad \hat{\sigma}_Z^2 = 0.47494.$$

The estimated variance–covariance matrix of $\hat{\phi}_1$, $\hat{\theta}_1$, and $\hat{\mu}$ is

$$\begin{bmatrix} 0.0060296 & -0.0046761 & 0.0017655 \\ -0.0046761 & 0.0128889 & -0.0020637 \\ 0.0017655 & -0.0020637 & 0.1225691 \end{bmatrix}.$$

Using the square roots of the diagonal elements of the variance–covariance matrix as standard error estimates, the following additional R commands give approximate two-sided 95% confidence intervals for the parameters.

```
coefficients[1] + c(-1, 1) * qnorm(0.975) * sqrt(varianc cov[1, 1])
coefficients[2] + c(-1, 1) * qnorm(0.975) * sqrt(varianc cov[2, 2])
coefficients[3] + c(-1, 1) * qnorm(0.975) * sqrt(varianc cov[3, 3])
```

The approximate 95% confidence intervals are

$$0.59271 < \phi_1 < 0.89709 \quad 0.09808 < \theta_1 < 0.54310 \quad 578.37 < \mu < 579.74.$$

Since none of these confidence intervals contains zero, we continue to entertain this tentative ARMA(1, 1) model and transition to an analysis of the residuals.

The following R commands plot the residuals as a time series, along with the associated sample autocorrelation function and sample partial autocorrelation function.

```
zhat = arima(LakeHuron, order = c(1, 0, 1))$residuals
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(zhat)
acf(zhat, lag.max = 40)
pacf(zhat, lag.max = 40)
```

The results are displayed in Figure 9.44. From the top graph, the residuals do not appear to have any cyclic variation, trend, or serial correlation. The sample autocorrelation function values for the residuals do not have any values that fall outside of the 95% confidence limits. Likewise for the sample partial autocorrelation function values.

Since there are no sample autocorrelation function values that fall outside of the 95% confidence limits $\pm 1.96/\sqrt{n}$ in the plot in Figure 9.44 of the first 40 sample autocorrelation function values associated with the residuals, and we expect $40 \cdot 0.05 = 2$ values to fall outside of these limits in the case of a good fit, we fail to reject H_0 in this case. The independence of the residuals is not rejected by this test. The tentative fitted ARMA(1, 1) model is not rejected by this test.

The R code below calculates the Box–Pierce test statistic and the Ljung–Box test statistic and the associated p -values using the built-in `Box.test` function.

```
zhat = arima(LakeHuron, order = c(1, 0, 1))$residuals
Box.test(zhat, lag = 40, type = "Box-Pierce", fitdf = 3)
Box.test(zhat, lag = 40, type = "Ljung-Box", fitdf = 3)
```

The Box–Pierce test statistic is 17.4 and the associated p -value is $p = 0.997$. The Ljung–Box test statistic is 23.0 and the associated p -value is $p = 0.966$. We fail to

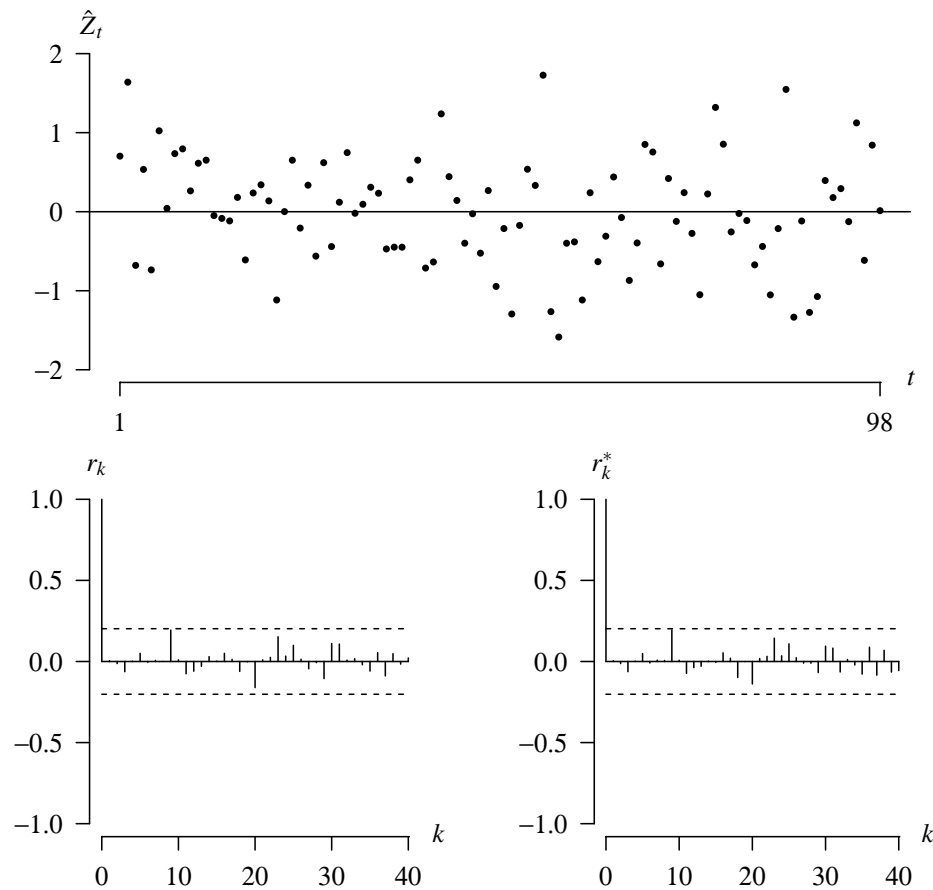


Figure 9.44: Time series plot, r_k , and r_k^* for the residuals from a fitted ARMA(1, 1) model.

reject H_0 in both tests based on the chi-square critical value with $40 - 3 = 37$ degrees of freedom. The independence of the residuals is not rejected by this test. The tentative fitted ARMA(1, 1) model is not rejected by these tests.

The following R code calculates the test statistic and the p -value for the turning point test applied to the time series consisting of the residual values for the ARMA(1, 1) fit to the Lake Huron time series.

```
zhat = arima(LakeHuron, order = c(1, 0, 1))$residuals
n = length(zhat)
m = (2 / 3) * (n - 2)
v = (16 * n - 29) / 90
T = 0
for (i in 2:(n - 1)) {
  if ((zhat[i - 1] < zhat[i] && zhat[i] > zhat[i + 1]) ||
      (zhat[i - 1] > zhat[i] && zhat[i] < zhat[i + 1])) T = T + 1
```

```

}
s = (T - m) / sqrt(v)
2 * (1 - pnorm(abs(s)))

```

The tail probability is doubled because the alternative hypothesis is two-tailed for the turning point test. The test statistic s is 1.21 and the p -value is $p = 0.23$. The turning point test found that there were $T = 69$ turning points in the time series of the residuals, and that is just slightly higher than the number that we expect to have if the residuals from the fitted ARMA(1, 1) model were mutually independent random variables. We again fail to reject the null hypothesis in this case. The independence of the residuals is not rejected by this test. The tentative fitted ARMA(1, 1) model is not rejected by this test.

The residuals are standardized by dividing by their sample standard deviation. The following R statements plot a histogram of the standardized residuals using the `hist` function and a QQ plot to assess normality using the `qqnorm` function.

```

zhat = arima(LakeHuron, order = c(1, 0, 1))$residuals
hist(zhat / sd(zhat))
qqnorm(zhat / sd(zhat))

```

The plots are shown in Figure 9.45. The histogram shows that all standardized residuals fall between -2.5 and 2.5 and exhibit a roughly bell-shaped probability distribution, with the exception of a deficit of residuals falling between -1.5 and -1.0 . The horizontal axis on the histogram is the standardized residual and the vertical axis is the frequency. The QQ plot is approximately linear, indicating a reasonable approximation to normality for the standardized residuals. The horizontal axis on the QQ plot is the standardized theoretical quantile and the vertical axis is the associated normal data quantile. Although a formal statistical goodness-of-fit test (such as the Shapiro–Wilk or the Kolmogorov–Smirnov test) should be conducted, it appears that the assumption of Gaussian white noise is appropriate for the ARMA(1, 1) time series model based on these two plots.

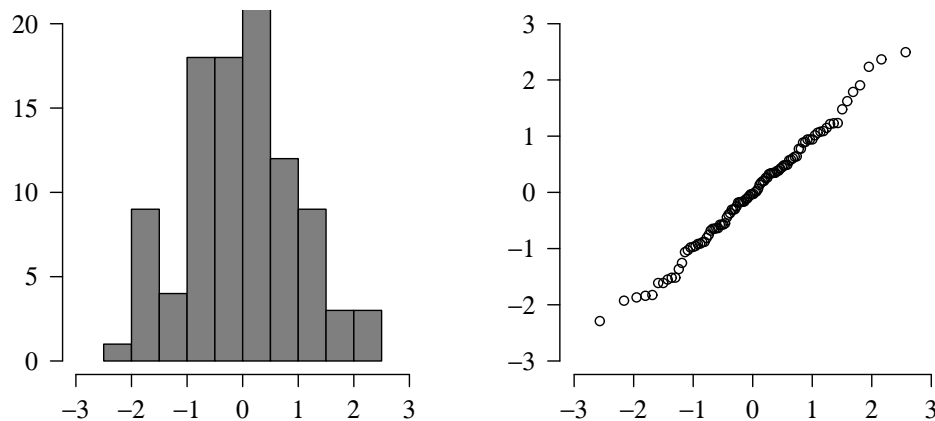


Figure 9.45: Histogram (left) and QQ plot (right) of the fitted ARMA(1, 1) standardized residuals.

We have seen a number of indicators that the ARMA(1, 1) time series model with Gaussian error terms seems to be an adequate model for the Lake Huron lake level time series, with the exception of a linear trend apparent by viewing the time series in Figure 9.19. The ARMA(1, 1) model has not been rejected by any of the model adequacy tests.

The final fitted shifted ARMA(1, 1) model with maximum likelihood estimates for the parameters is given by

$$X_t = 579.06 + 0.74490(X_{t-1} - 579.06) + 0.32059Z_{t-1} + Z_t,$$

where Z_t is a sequence of independent and identically distributed $N(0, 0.47494)$ error terms.

With the shifted ARMA(1, 1) model established, we now consider forecasting future values of a time series. In the case of the Lake Huron time series, this corresponds to the one-step-ahead forecast for 1973, the two-steps-ahead forecast for 1974, the three-steps-ahead forecast for 1975, etc. The code below uses the R `predict` function to generate the forecasted values and their standard errors.

```
fit      = arima(LakeHuron, order = c(1, 0, 1))
forecast = predict(fit, n.ahead = 5)
lower    = forecast$pred - qnorm(0.975) * forecast$se
upper    = forecast$pred + qnorm(0.975) * forecast$se
```

These standard errors can be used to calculate approximate two-sided 95% prediction interval limits on the forecasted values. The results are summarized in Table 9.19. Notice that the forecasts trend monotonically toward $\bar{x} = 579$ and the standard errors increase as the time horizon h increases. The increasing standard error is consistent with having less precision in the forecast as the time horizon h increases.

Time	$t = 99$	$t = 100$	$t = 101$	$t = 102$	$t = 103$
Year	1973	1974	1975	1976	1977
Forecast	579.73	579.56	579.43	579.34	579.26
Standard error	0.689	1.007	1.146	1.216	1.254
Lower prediction bound	578.38	577.59	577.19	576.95	576.81
Upper prediction bound	581.08	581.53	581.68	581.72	581.72

Table 9.19: Forecasts and 95% prediction intervals for the Lake Huron time series.

Figure 9.46 shows (a) the original time series $x_1, x_2, \dots, x_{98}$ as points (●) connected by lines, (b) the first 10 forecasted lake levels $\hat{X}_{99}, \hat{X}_{100}, \dots, \hat{X}_{108}$ as open circles (○), (c) the 95% prediction intervals as a shaded region, and (d) the next 10 actual average lake level values in July for the years 1973–1982 taken from the NOAA Great Lakes Experimental Research Laboratory website,

580.98, 581.04, 580.49, 580.52, 578.57, 578.96, 579.94, 579.77, 579.44, 578.97,

as points (●) connected by lines. The forecasted values as well as the prediction intervals given in Figure 9.46 associated with the fitted ARMA(1, 1) model are very similar to those in Figure 9.25 from Example 9.21. The two models are clearly close competitors for modeling the Lake Huron levels.

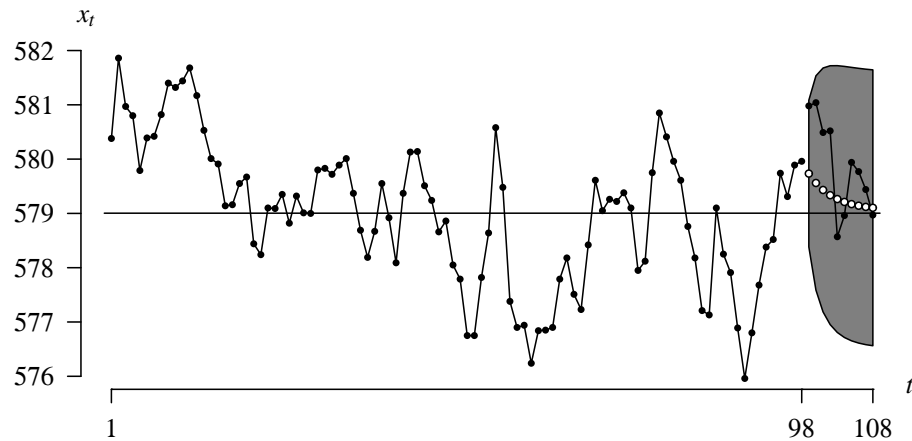


Figure 9.46: Lake Huron level forecasts and 95% prediction intervals from an ARMA(1, 1) model.

ARMA modeling can achieve population autocorrelation function and population partial autocorrelation function shapes that are not possible with just $AR(p)$ and $MA(q)$ models alone. For an $ARMA(p, q)$ model with $p > 0$ and $q > 0$, both the population autocorrelation function and the population partial autocorrelation function tail off; neither of the two cut off after a certain lag.

An inherent weakness of ARMA modeling is that it requires stationarity. Many time series which occur in practice are not stationary, and the next section gives techniques that can be used to overcome this weakness.

9.4 Nonstationary Models

There are two commonly-used strategies for converting a nonstationary time series to a stationary time series in order to use ARMA modeling (or some other model which requires stationarity) on the resultant stationary time series. The first strategy is known as *detrending*. In this case, the modeler estimates the trend, and then fits a stationary time series model to the difference between the raw time series data and the estimated trend. The second strategy is known as *differencing*. In this case the modeler differences the time series one or more times, resulting in a stationary time series. Differencing carries the added benefit that no parameters are required other than the number of differences to take. The following two subsections consider these two strategies.

9.4.1 Removing Trends Via Regression

Although regression is not the only way to detrend a time series, it provides an adequate roadmap on how to proceed with the detrending process that generalizes to other mechanisms. This subsection illustrates detrending with a single example. We return for a third time to the Lake Huron levels which were fit to an $AR(2)$ model in Section 9.1.2 and fit to an $ARMA(1, 1)$ model in Section 9.3.

Example 9.40 We again consider the construction of a time series model from the $n = 98$ annual observations of the level of Lake Huron (in feet) between 1875 and 1972

that was first encountered in Example 9.14. The observations are stored in a time series in R named `LakeHuron`. The scatterplot of the lake levels depicted in Figure 9.47 includes a regression line showing the downward trend in the lake levels over time. The p -value for the statistical test for significance of the slope of this regression line is $p = 4 \cdot 10^{-8}$, providing strong evidence of a downward trend over time, even though the usual assumptions associated with simple linear regression with normal error terms are not perfectly satisfied in this setting. Although the $AR(2)$ and $ARMA(1, 1)$ models have been successfully fitted to this time series treating it as stationary, this tiny p -value prevents us from fully embracing either of these models. The purpose of this example is to explicitly consider this downward trend by fitting the residuals from this simple linear regression model to an ARMA time series model.

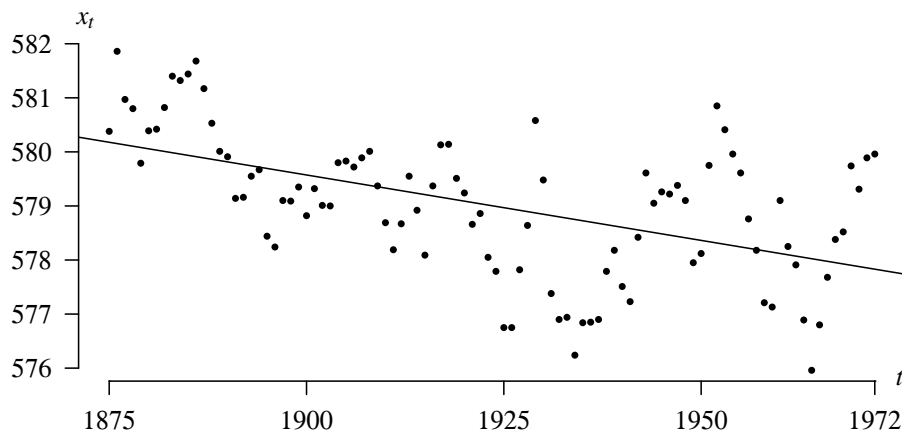
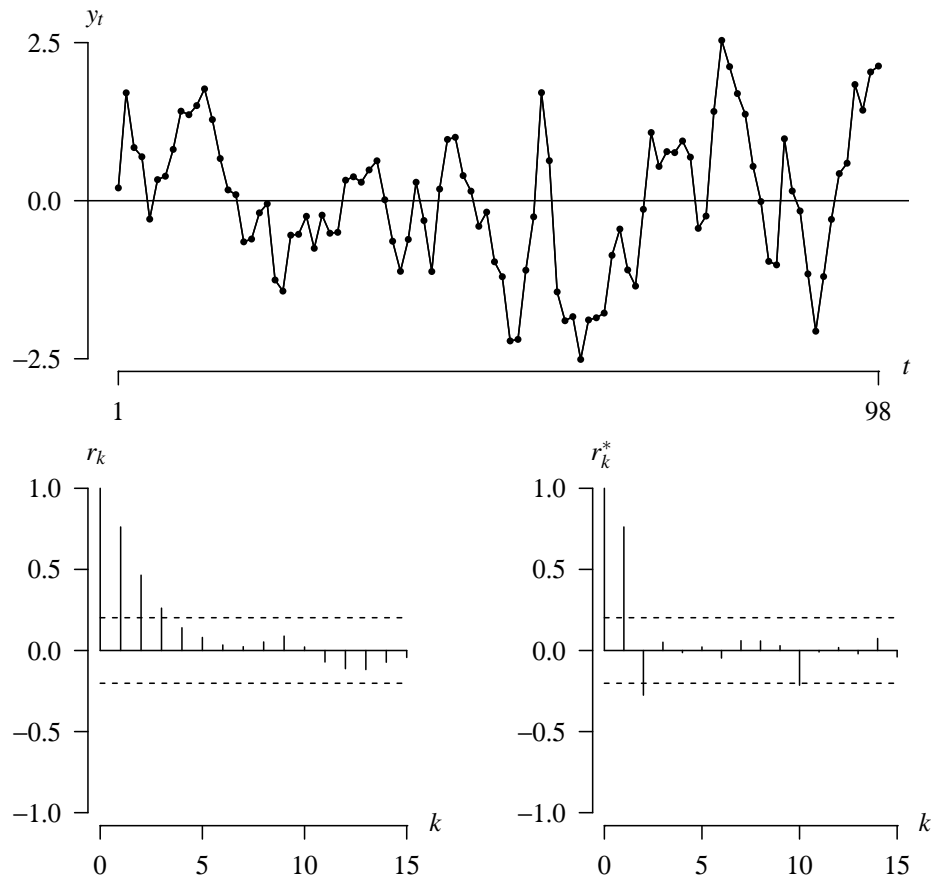


Figure 9.47: Lake Huron levels (1875–1972) with regression line.

The residuals from this simple linear regression form a new time series which will be denoted by $\{y_t\}$, where $y_t = x_t - \hat{x}_t$ and $\hat{x}_t$ is the fitted value in the simple linear regression. The R statements below generate plots of the time series, the sample autocorrelation function, and the sample partial autocorrelation function for the residuals of the simple linear regression.

```
y = lm(LakeHuron ~ seq(1:98))$resid
plot.ts(y)
abline(h = mean(y))
acf(y, lag.max = 40)
pacf(y, lag.max = 40)
```

These plots are displayed in Figure 9.48. The time series of the residuals appears to have no trend and also appears to be centered around zero. In fact, the time series is *exactly* centered around zero because the residuals of this regression must sum to zero via Theorem 1.6. This means that there is no need to include a shift parameter μ in the ARMA model that we develop for the residuals. The sample autocorrelation function of the residuals appears to be tailing out and the first two sample partial autocorrelation

Figure 9.48: Residuals plot, r_k , and r_k^* for Lake Huron lake levels.

function values are statistically significant. This is strong evidence that an AR(2) model is an appropriate tentative model for the residuals.

The following R statements fit the AR(2) model via maximum likelihood estimation to the time series of residuals.

```
y = lm(LakeHuron ~ seq(1, 98))$resid
model = ar(y, order.max = 2, method = "mle")
```

This model should be subjected to all of the model assessment tests that have been applied to all previous time series analyzed in previous examples. The residuals of the estimated AR(2) model to the simple linear regression residuals result in large p -values for the Box–Pierce test and the Box–Ljung test, along with a bell-shaped histogram and an almost perfectly linear QQ normal plot. This evidence confirms the evidence in the plots of r_k and r_k^* which pointed to an AR(2) model for the residuals of the simple linear regression. The `predict` function can then be used in the usual fashion to forecast the residuals into the future. The plot of the residuals, the next 10 forecasted residuals,

and the associated 95% prediction intervals is given in Figure 9.49. As expected, the forecasted values are smooth and converge to zero.

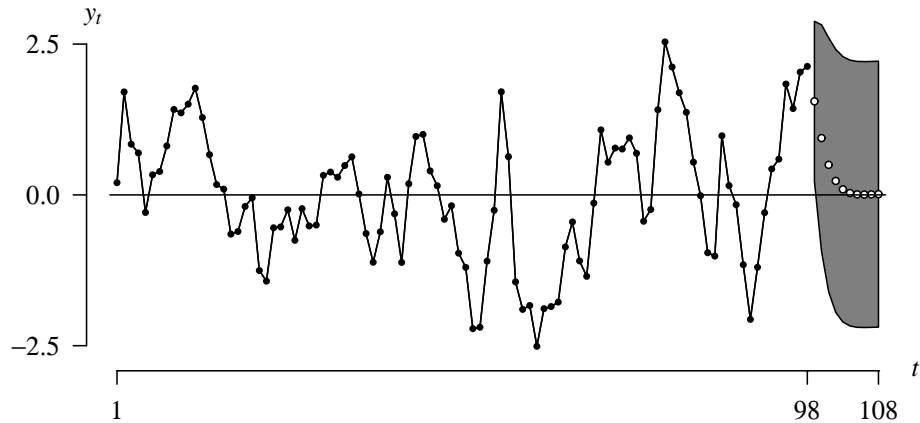


Figure 9.49: Lake Huron residuals, forecasts, and 95% prediction intervals.

Finally, the last step is to translate Figure 9.49 back to the raw time series observations. Figure 9.50 shows (a) the original time series $x_1, x_2, \dots, x_{98}$ as points ($\bullet$) connected by lines, (b) the regression line associated with the original time series, (c) the first 10 forecasted lake levels $\hat{X}_{99}, \hat{X}_{100}, \dots, \hat{X}_{108}$ as open circles ($\circ$), (d) the 95% prediction intervals as a shaded region, and (e) the next 10 actual average lake level values in July for the years 1973–1982 taken from the NOAA Great Lakes Experimental Research Laboratory website,

580.98, 581.04, 580.49, 580.52, 578.57, 578.96, 579.94, 579.77, 579.44, 578.97,

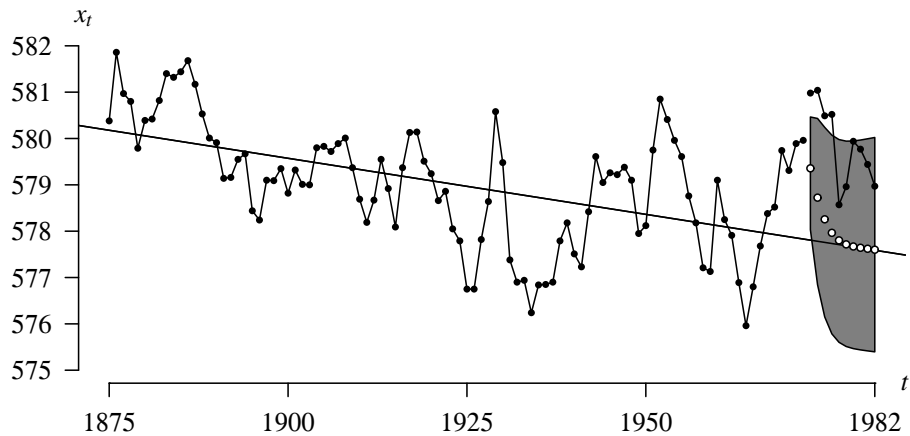


Figure 9.50: Lake Huron levels (1875–1972) with regression line and forecasts.

as points (●) connected by lines. Notice that the forecasted values converge to the regression line as anticipated. Notice also that the actual values all exceed the forecasted values and the first four forecasted values fall outside of the prediction limits. If we did not know of these actual values, we would be satisfied with this detrended AR(2) time series model. However, these actual values call into question whether the lake levels are truly decreasing over time.

So far, there have been three different approaches to constructing a time series model for the Lake Huron levels:

- the shifted AR(2) model from Examples 9.14, 9.15, 9.16, 9.17, 9.18, 9.19, 9.20, and 9.21,
- the shifted ARMA(1, 1) model from Example 9.39, and
- the AR(2) model applied to the residuals from a simple linear regression from Example 9.40.

Which approach is preferred? Although the shifted AR(2) and shifted ARMA(1, 1) models fitted to the raw time series are roughly comparable and give nearly-identical forecasts, the shifted ARMA(1, 1) model has a slight edge for the following two reasons. First, from Table 9.7, the AIC value is 215 for the shifted AR(2) model and the AIC value is 214 for the shifted ARMA(1, 1) model. A smaller value implies a better fit. Second, the sum of squared residuals for the shifted AR(2) model is 46.9 and the sum of squared residuals for the shifted ARMA(1, 1) model is 46.5. A smaller sum of squared residuals for two models with an equal number of parameters is preferred. Both models have four parameters. These two sums of squared residuals for the two models are computed with the R statements

```
sum(arima(LakeHuron, order = c(2, 0, 0))$residuals ^ 2)
sum(arima(LakeHuron, order = c(1, 0, 1))$residuals ^ 2)
```

Although the differences between the AIC values and the sums of squares is small, the shifted ARMA(1, 1) model holds a slight edge.

The detrended model from Example 9.40, on the other hand, is preferred over the two stationary models because it explicitly models the decreasing lake levels over time. However, the fact that all of the forecasted values in the detrended model are low relative to the actual values in the years 1973 to 1982 is troubling. Could it be the case that there was no downward trend after all? At this point, some serious detective work is in order to see if the early values in the raw time series were elevated by some external influence and should not be included as a part of the time series. A rigorous search should be conducted for any external cause which might elevate the early values in the time series: excess rainfall, elevated temperatures, dredging, bridge projects, flow control projects, etc. As a particular instance, if the first 20 values of the time series can be eliminated due to the identification of an assignable cause for the years 1875–1894, for example, the p -value from simple linear regression testing for the statistical significance of the slope increases from a highly significant $p = 4 \cdot 10^{-8}$ to a nonsignificant $p = 0.11$. The downward trend would now be slight and a stationary model could be fitted to the remaining values in the time series.

Detrending has proved to be an effective method for transforming a nonstationary time series to a stationary time series. The second technique involves differencing.

9.4.2 ARIMA(p, d, q) Models

George Box and Gwilym Jenkins devised a time series modeling methodology known as *ARIMA modeling*. The I between AR and MA stands for *integrated*. These models are sometimes referred to

as Box–Jenkins models. An $\text{ARIMA}(p, d, q)$ time series model is one in which the d th-differenced times series, $\nabla^d X_t$, is an $\text{ARMA}(p, q)$ time series. So ARIMA time series modeling uses repeated differencing of the raw time series in order to achieve a time series which appears to be stationary. ARMA modeling can be then applied to the resulting stationary time series.

Definition 9.7 An $\text{ARIMA}(p, d, q)$ time series model for $\{X_t\}$ is one in which the d th-differenced times series, $\nabla^d X_t$, is an $\text{ARMA}(p, q)$ time series for some nonnegative integer d . An $\text{ARIMA}(p, d, q)$ model can be written in compact form as

$$\phi(B)\nabla^d X_t = \theta(B)Z_t,$$

where $\phi(B)$ and $\theta(B)$ are the usual characteristic polynomials for an $\text{ARMA}(p, q)$ model and $Z_t \sim \text{WN}(0, \sigma_Z^2)$.

Three key parameters in an ARIMA model are p , d , and q , which are all nonnegative integers. The parameter p is the number of coefficient parameters in the autoregressive portion of the model. The parameter d is the number of differences that are applied to the original time series in order to achieve stationarity. The parameter q is the number of coefficient parameters in the moving average portion of the model. So the general format for specifying an ARIMA model is $\text{ARIMA}(p, d, q)$. In addition to the parameters p , d , and q , there are $p + q + 1$ parameters that define an $\text{ARIMA}(p, d, q)$ model: the p autoregressive parameters $\phi_1, \phi_2, \dots, \phi_p$, the q moving average parameters $\theta_1, \theta_2, \dots, \theta_q$, and the variance of the white noise σ_Z^2 . As in the case of ARMA models, a shift parameter μ can be included in the model. If one or more of these parameters is zero, they are omitted from the specification. An $\text{IMA}(2, 1)$ model, for example, has $p = 0$ autoregressive terms, $d = 2$ differences, and $q = 1$ moving average term. If a model only involves, for example, the autoregressive portion of the model with two terms (that is, no differencing and no moving average terms), then this model is specified as an $\text{AR}(2)$ model. An $\text{ARMA}(p, q)$ model is a special case of an $\text{ARIMA}(p, d, q)$ model when $d = 0$.

ARIMA modeling will be illustrated by a simulation example that will reveal what a realization of an ARIMA process looks like, along with the R code required to fit these simulated values to an ARIMA model.

Example 9.41 Simulate a realization of $n = 100$ observations from an $\text{ARI}(1, 1)$ time series model with $\phi = 0.8$ and $\sigma_Z^2 = 4$. Fit the resulting simulated values to an ARIMA model.

This problem gives one instance of what an ARIMA model with a nonzero value for d looks like. The R code below uses the `arima.sim` function to generate a realization of an $\text{ARIMA}(1, 1, 0)$ time series model, which is more commonly known as an $\text{ARI}(1, 1)$ model. Even though 99 observations are requested, a total of 100 will be generated because the differencing operator is being undone within `arima.sim`. The code also plots the sample autocorrelation function and the sample partial autocorrelation function of the simulated realization.

```
set.seed(1)
x = arima.sim(list(order = c(1, 1, 0), ar = 0.8), n = 99, sd = 2)
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(x)
acf(x, lag.max = 40)
pacf(x, lag.max = 40)
```

The results are shown in Figure 9.51. The realization is clearly generated from a nonstationary time series model with an overall meandering upward trend. This conclusion is supported by the graphs of r_k and r_k^* .

The augmented Dickey–Fuller test can be used to assess the stationarity of the simulated time series. It has been implemented in R in the `adf.test` function in the `tseries` package. There is no need to run this test for this particular realization of the time series; the time series plot clearly shows that this is a nonstationary time series. Now consider fitting this time series realization to an $\text{ARIMA}(p, d, q)$ model. Since the time series realization exhibits a meandering linear increase, it is possible that a single difference might be adequate for transforming this time series to achieve stationarity. Although it is in some sense cheating because we know that the realization was generated from an $\text{ARI}(1, 1)$ time series model, the R code that follows takes a single difference of the time series depicted in Figure 9.51 and plots the differenced series $y_t = \nabla x_t = x_t - x_{t-1}$, the associated sample autocorrelation function, and the sample partial autocorrelation function.

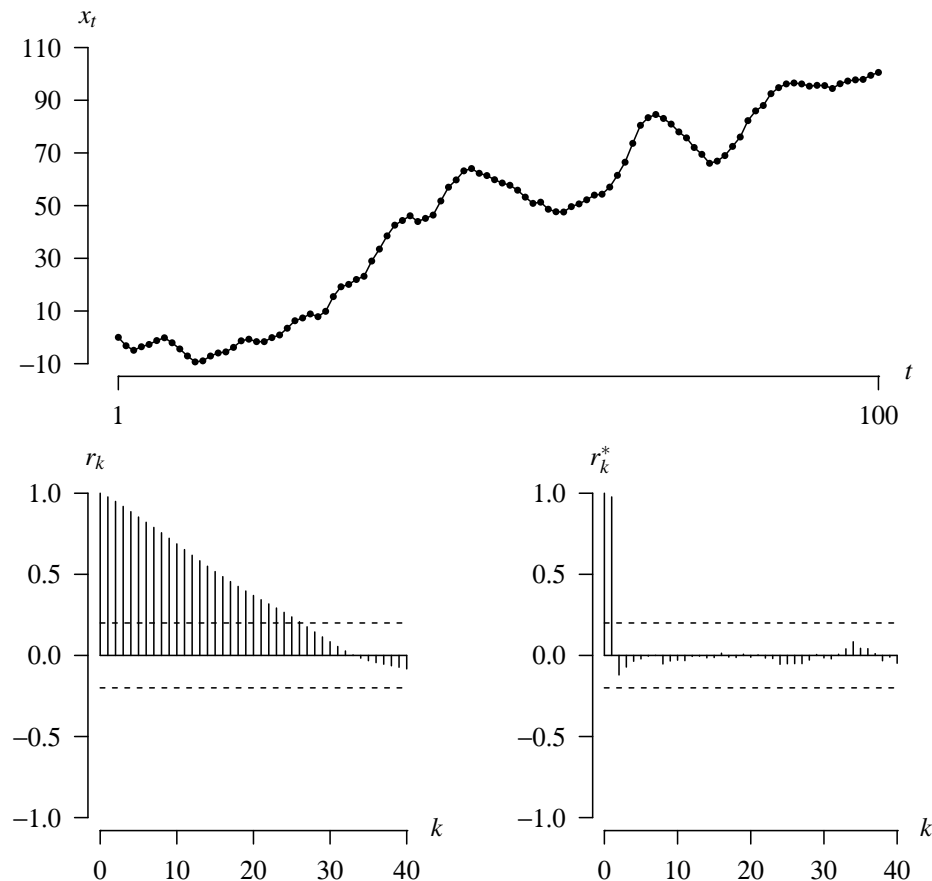


Figure 9.51: Time series plot, r_k , and r_k^* for a realization of a simulated $\text{ARI}(1, 1)$ model.

```

set.seed(1)
x = arima.sim(list(order = c(1, 1, 0), ar = 0.8), n = 100, sd = 2)
y = diff(x)
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(y)
acf(y, lag.max = 40)
pacf(y, lag.max = 40)

```

Figure 9.52 shows a graph of the differenced time series and the associated graphs of r_k and r_k^* . The differencing has achieved its goal; the differenced values appear to be stationary. Furthermore, the sample partial autocorrelation function has a single statistically significant value at lag 1 and then cuts off. (The statistically significant value at lag 18 is attributed to random sampling variability because we expect that 2 of the 40 r_k^* values will lie outside of the 95% bounds by chance.) The sample autocorrelation appears to be gradually tailing out. This is evidence that supports an AR(1) model for

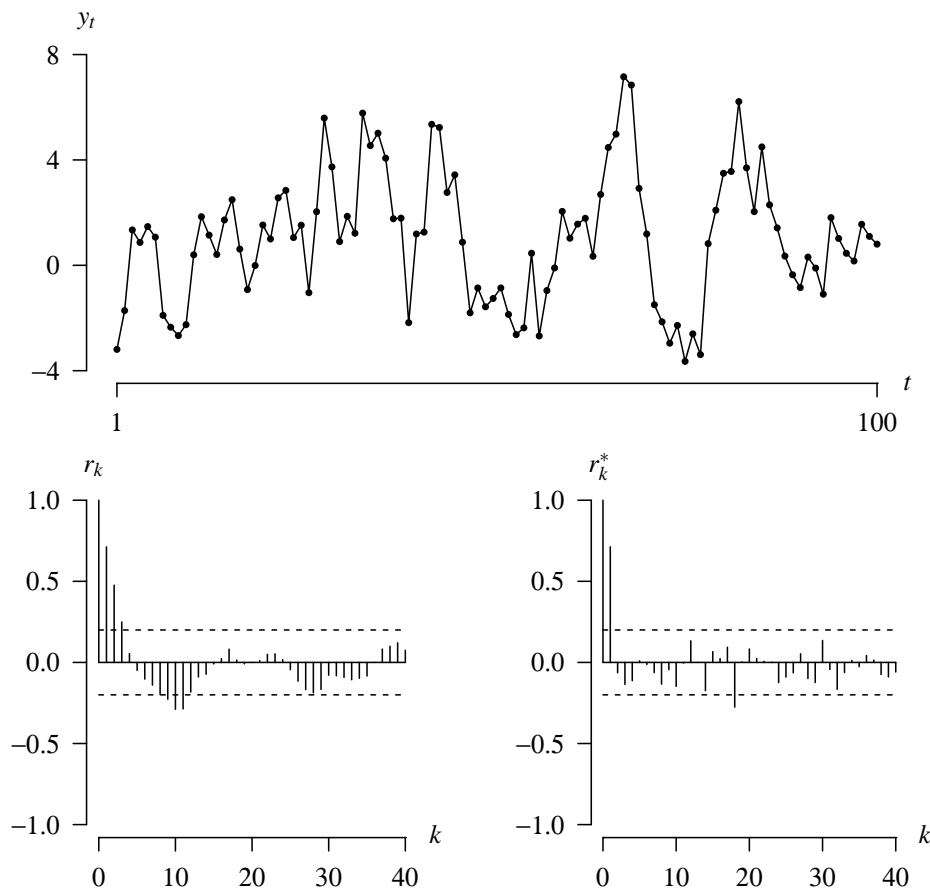


Figure 9.52: Time series plot, r_k , and r_k^* for differences of a realization of the simulated values.

the differenced values from the $\text{ARI}(1, 1)$ realization, just as we suspected would be the case.

The R statements

```
set.seed(1)
x = arima.sim(list(order = c(1, 1, 0), ar = 0.8), n = 100, sd = 2)
y = diff(x)
fit = arima(y, order = c(1, 0, 0), method = "ML")
fit$coef[1]
fit$sigma2
```

return an estimated coefficient $\hat{\phi} = 0.73$ (which is near the population value $\phi = 0.8$) and estimated white noise variance $\hat{\sigma}_Z^2 = 2.9$ (which is near the population value $\sigma_Z^2 = 4$). A Monte Carlo simulation could be conducted to see how far these estimated values stray from their population counterparts. Increasing the length of the time series will make these estimates closer to their associated population values on average.

The ARIMA modeling process is adequate for nonstationary models but is not well-suited to handling cyclic variation. The SARIMA (seasonal autoregressive integrated moving average) model has been formulated to overcome this weakness.

Definition 9.8 A seasonal ARIMA time series model for $\{X_t\}$, denoted by a SARIMA model of order $(p, d, q) \times (P, D, Q)_s$ with seasonal order s , is given in compact form by

$$\phi(B)\Phi(B^s)\nabla^d\nabla_s^D X_t = \theta(B)\Theta(B^s)Z_t,$$

where p, d, q, P, D, Q , and s are nonnegative integers,

- $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$,
- $\Phi(B) = 1 - \Phi_1 B - \Phi_2 B^2 - \dots - \Phi_P B^P$,
- $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q$,
- $\Theta(B) = 1 + \Theta_1 B + \Theta_2 B^2 + \dots + \Theta_Q B^Q$,

and $\{Z_t\} \sim WN(0, \sigma_Z^2)$.

An ARIMA model is a special case of a SARIMA model when $P = D = Q = 0$. The ∇^d term in the SARIMA model is associated with an ordinary difference; the ∇_s^D term is associated with a seasonal difference. Consider the inside portion of the SARIMA defining formula, $\nabla^d\nabla_s^D X_t$, in a modeling setting in which monthly data is being collected and the modeler believes that there is cyclic annual variation, so $s = 12$. In the case of $d = 1$ ordinary difference and $D = 1$ seasonal difference, this portion of the SARIMA defining formula becomes

$$\begin{aligned}\nabla\nabla_{12}X_t &= \nabla(\nabla_{12}X_t) \\ &= \nabla(X_t - X_{t-12}) \\ &= (X_t - X_{t-12}) - (X_{t-1} - X_{t-13}).\end{aligned}$$

The ∇ operator is being used to eliminate a linear trend and the ∇_{12} operator is being used to eliminate seasonality. The seasonal AR term $\Phi(B^s)$ and the seasonal MA term $\Theta(B^s)$ in Definition 9.8 provide autoregressive and moving average terms for observations that are s units distant in time.

Example 9.42 Forecast the next three years of international air travel based on the `AirPassengers` time series from Example 7.2.

The plot of the time series is given in Figure 9.53. As indicated in Example 7.32, the annual cycle associated with international air travel over this period does not appear to be sinusoidal in nature. The peak months for international travel are in July and August when school is not in session and the low month for international travel is November, as seen in Figure 7.30. This time series provides a challenging modeling exercise because it exhibits a nonconstant variance, a trend, and periodicity. These three modeling challenges will be addressed in that order, one by one.

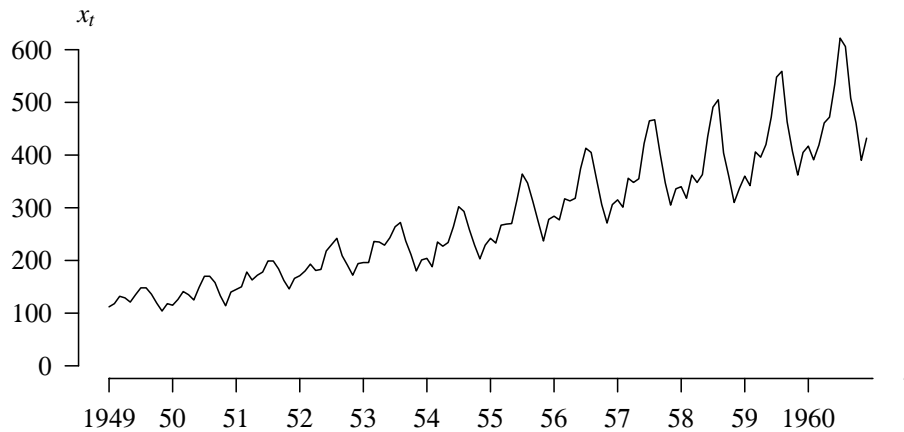


Figure 9.53: International airline passengers (in thousands) 1949–1960.

We begin by addressing the nonconstant variance. Since the variance appears to be increasing over time, a logarithmic transformation is a reasonable transformation to apply to the time series. Let $\{x_t\}$ denote the original time series and let $y_t = \ln x_t$. The R statement

```
ts.plot(log(AirPassengers))
```

plots the natural logarithm of the raw time series. The plot of the transformed time series is given in Figure 9.54.

The transformation appears to be effective. The variance of the logarithms of the raw time series observations is now close to constant over time. The next step is to address the trend. Since the trend of the transformed time series depicted in Figure 9.54 is approximately linear, a single difference ($d = 1$) is taken. The differenced time series is

$$w_t = \nabla y_t = \nabla \ln x_t = \ln x_t - \ln x_{t-1}.$$

(The resulting time series is not named z_t to avoid any conflict with the white noise terms.) This transformed and differenced time series, along with the associated sample autocorrelation function and sample partial autocorrelation function are graphed with the R statements

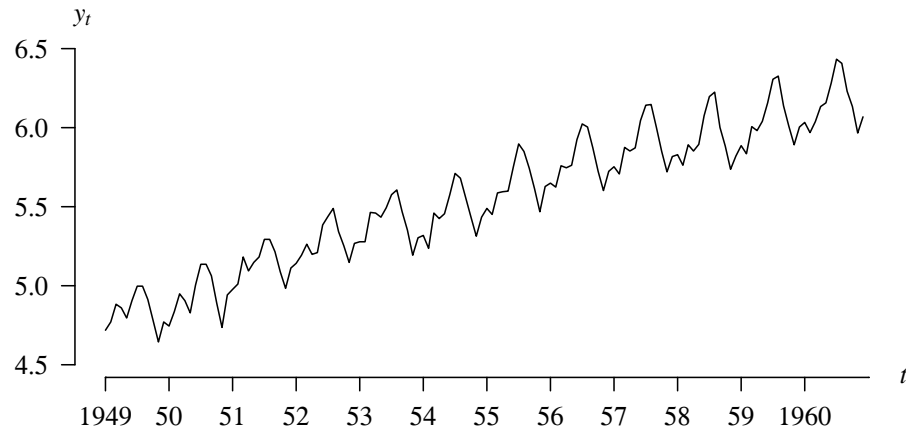


Figure 9.54: Logarithm of international airline passengers (in thousands) 1949–1960.

```
w = diff(log(AirPassengers))
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot.ts(w)
acf(w, lag.max = 40)
pacf(w, lag.max = 40)
```

The associated graphs are displayed in Figure 9.55. Instead of the usual $12 \cdot 12 = 144$ observations from the previous two figures, the differencing operation leaves only 143 observations, which is reflected in the labels on the horizontal axis of the plot of the differences of the logarithms of the original time series values. The differencing has proved to be successful. The time series plot of w_t appears to be stationary. The strongly statistically significant sample autocorrelation function values at lags 12, 24, and 36 are a reminder that even though the nonconstant variance and trend have been addressed, the cyclic variation has not been addressed. ARMA modeling is *not* appropriate at this point because r_k is neither tailing out nor cutting off. There is still an annual cyclic component present in $\{w_t\}$. A reasonable way to proceed is to employ a seasonal ARIMA model to account for the cyclic variation. Backing up one level, we would like to fit a SARIMA $(p, d, q) \times (P, D, Q)_{12}$ model to the natural logarithms of the raw passenger counts. The choice $s = 12$ for the seasonal order is to account for the monthly collection of the passenger counts which exhibit a clear annual cycle; the choice of $d = 1$ is appropriate based on the fact that the time series $\{w_t\}$ in Figure 9.55 appears to be stationary. But what about the other parameters (p , q , P , D , and Q)? An exhaustive search using the `arima` function in R to locate the smallest value of AIC results in the following settings:

$$d = P = Q = 1$$

and

$$p = q = D = 0.$$

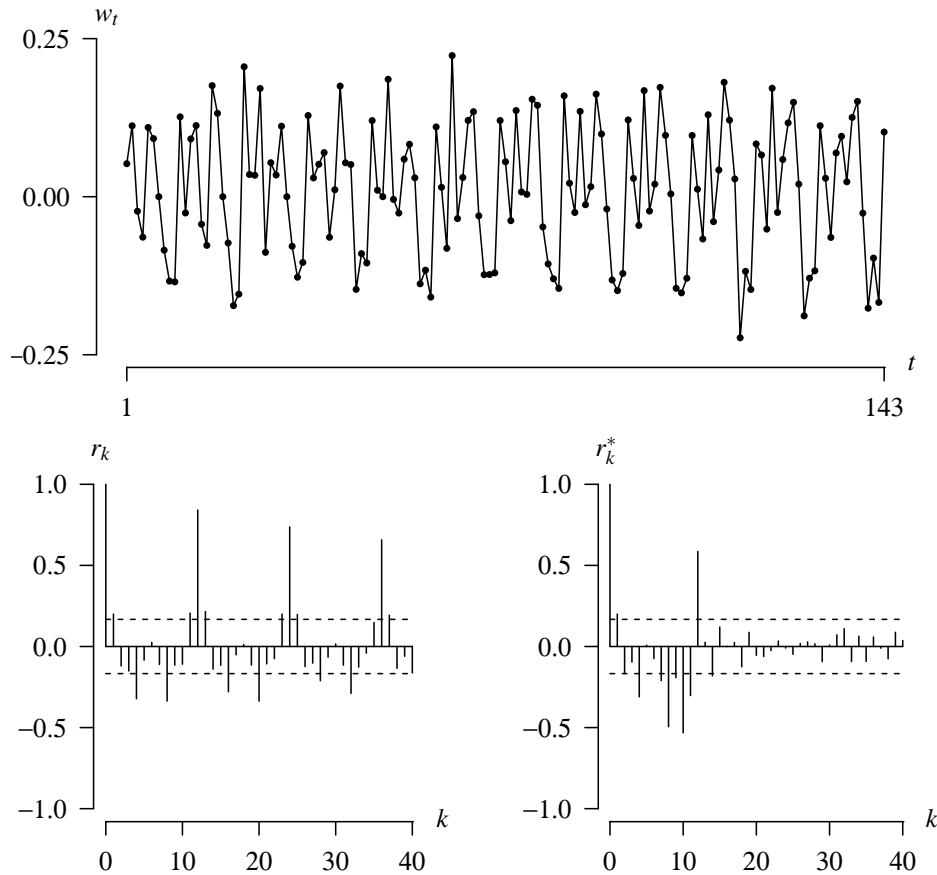


Figure 9.55: Logarithm of international airline passengers (in thousands) 1949–1960.

The single R statement below fits the SARIMA $(0, 1, 0) \times (1, 0, 1)_{12}$ model to the logarithms of the passenger counts in the `AirPassengers` time series.

```
fit = arima(log(AirPassengers),
            order = c(0, 1, 0),
            seasonal = list(order = c(1, 0, 1), period = 12))
```

The maximum likelihood estimates of the parameters are $\Phi = 0.9877$, $\Theta = -0.5935$ and $\hat{\sigma}_Z^2 = 0.001526$. So the fitted model is

$$(1 - 0.9877B^{12}) \nabla Y_t = (1 - 0.5935B^{12}) Z_t,$$

where $Y_t = \ln X_t$, and $Z_t \sim WN(0, 0.001526)$. This fitted SARIMA model achieves an AIC value of -486.9953 . There are several other competing SARIMA models with nearby AIC values.

The final step is to use the fitted SARIMA model to forecast international airline travel for the subsequent three years (36 months). The R code below fits the SARIMA model

with the `arima` function, uses the `predict` function to calculate the forecasted values and their standard errors, and then plots the original time series, the forecasted values, and the 95% prediction intervals.

```
fit      = arima(log(AirPassengers),
                  order = c(0, 1, 0),
                  seasonal = list(order = c(1, 0, 1), period = 12))
n        = length(AirPassengers)
h        = 36
forecast = predict(fit, n.ahead = h)
alpha    = 0.05
crit     = qnorm(1 - alpha / 2)
lo       = forecast$pred - crit * forecast$se
hi       = forecast$pred + crit * forecast$se
beginx   = time(AirPassengers)[n]
deltax   = deltat(AirPassengers)
xval1    = seq(beginx + deltax, beginx + h * deltax, length.out = h)
xval2    = c(xval1, rev(xval1))
yvals    = exp(c(lo, rev(hi)))
ts.plot(AirPassengers, exp(forecast$pred), ylim = c(0, max(yvals)))
polygon(xval2, yvals, col = "gray50")
points(xval1, exp(forecast$pred), pch = 16, cex = 0.7, col = "white")
points(xval1, exp(forecast$pred), pch = 1, cex = 0.7)
```

This graph of the original time series and the 36 forecasted values is given in Figure 9.56. The forecasts from the SARIMA model show that the nonconstant variance, trend, and cyclic variation have been adequately captured by the model. Since there are $144 + 36 = 180$ points squeezed so tightly together in the plot, a second graph of the forecasted values for just the last three cycles of the observed time series and the forecasted values is given in Figure 9.57.

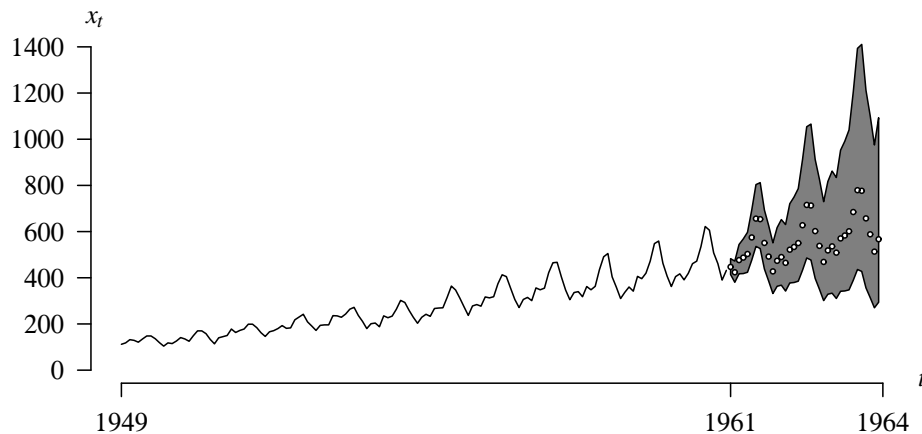


Figure 9.56: Forecasted international travel and 95% prediction intervals.

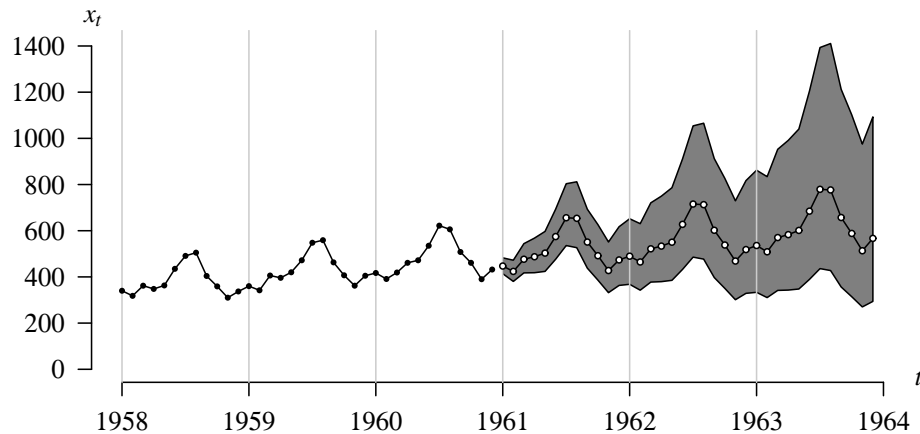


Figure 9.57: Forecasted international travel and 95% prediction intervals.

These forecasts are crucial for airports, airline manufacturers, and associated businesses as they can predict the impact of growth on supply chains, personnel requirements, and logistics necessary to support air travel.

To summarize this section, the modeling of a nonstationary time series involves the following steps.

- Plot the time series, noting any trends, seasonality, and nonconstant variance.
- Make the variance stable by applying appropriate transformations if necessary.
- Use detrending (possibly regression) or repeated differencing (to use an ARIMA model) to create a stationary time series.
- Plot the stationary time series along with its sample autocorrelation function and sample partial autocorrelation function.
- Hypothesize a tentative ARMA model for the stationary time series model. If there is a seasonal component, consider a SARIMA model on the transformed time series.
- Fit the tentative ARMA or SARIMA model. Perform the model assessment tests on the tentative time series model. If the fitted tentative ARMA or SARIMA model fails these tests, then hypothesize a new tentative model.
- Perform overfitting in the final model selection process to ensure that the best model has been selected.
- Apply the final time series model in the fashion dictated by the problem setting (this is often forecasting future values of the time series).

As illustrated in Example 9.42, time series modeling can be thought of as a step-by-step process of identifying and removing causes of variation in the time series (for example, trend, cycles, autocorrelation) until all that remains is white noise.

9.5 Spectral Analysis

In practice, many time series exhibit cyclic variation. The first two examples in Chapter 7 concerning monthly residential power consumption and monthly international airline travel both contain a cyclic component. The time series models derived from the general linear model do not explicitly consider cyclic variation; these models exist in what is known as the *time domain*. Spectral analysis considers modeling in the *frequency domain*. Spectral analysis decomposes a stationary time series into sinusoidal components (that is, sine and cosine functions) in order to identify frequencies associated with periodic components. Just as autoregressive models use regression on previous values of a time series in the time domain, spectral analysis uses regression on sine and cosine terms in the frequency domain.

Table 9.20 presents some new terminology that arises in spectral analysis and presents some analogies with known data analysis techniques. The column headings indicate that the three subsequent rows contain three application areas, three probability constructs, and their three statistical counterparts.

- The first row concerns the analysis of univariate data. In probability theory, several commonly-used probability distributions (for example, the exponential, normal, and binomial distribution) are investigated in order to build an inventory of potential probability distributions that might adequately describe a univariate data set. When an analyst encounters a univariate data set, one of the early steps in the analysis is to plot a histogram and compare its shape to the inventory of probability density functions associated with known probability distributions.
- The second row concerns time series analysis in the time domain. Shapes of the population autocorrelation function are derived for several commonly-used time series models (for example, the AR(2), MA(1), and ARMA(1, 1) models) in order to build an inventory of shapes such as those given in Figure 9.43 that might adequately describe the time series. When a time series analyst encounters time series observations, one of the early steps in the analysis is to plot the correlogram (a.k.a., the sample autocorrelation function) and compare its shape to the inventory of known population autocorrelation functions.
- The third row concerns time series analysis in the frequency domain. Shapes of the *spectral density function* are derived for several commonly-used time series models in order to build an inventory of shapes that might adequately describe the periodic nature of a time series. When a time series analyst encounters time series observations, one of the early steps is to plot the *periodogram* and compare its shape to the inventory of known spectral density functions.

The next two subsections will focus on the spectral density function and its statistical counterpart, the periodogram.

Application area	Probability construct	Statistical counterpart
univariate data analysis	probability density function	histogram
time series analysis: time domain	population autocorrelation function	correlogram
time series analysis: frequency domain	spectral density function	periodogram

Table 9.20: Population versus sample representations.

9.5.1 The Spectral Density Function

The emphasis in the spectral analysis of a time series is the identification of the frequencies associated with cycles. The frequencies will be denoted here by ω . Just as the population autocorrelation function is the natural tool for identifying and quantifying autocorrelation in the time domain, the spectral density function is the natural tool for identifying and quantifying the frequencies associated with cyclic variation in the frequency domain. As seen in the following definition, the spectral density function can be written in terms of the population autocovariance function.

Definition 9.9 Let $\{X_t\}$ be a stationary time series with population autocovariance function $\gamma(k)$. The *spectral density function* $f(\omega)$ is

$$f(\omega) = \frac{1}{\pi} \left[\gamma(0) + 2 \sum_{k=1}^{\infty} \gamma(k) \cos(\omega k) \right] \quad 0 < \omega < \pi.$$

The interpretation of the spectral density function is that $f(\omega)\Delta\omega$ reflects the contribution of frequencies in the interval $(\omega, \omega + \Delta\omega)$ to the variance of X_t for small values of $\Delta\omega$. When $f(\omega)$ is high, then frequencies near ω have a large impact on X_t . When $f(\omega)$ is low, then frequencies near ω have a small impact on X_t . The upper limit of the support of the spectral density function, π , is known as the *Nyquist frequency*. Frequencies that exceed π are not captured by the spectral density function. This is not a universal choice for the definition of the spectral density function or the upper limit of its support. There are many valid alternative choices. A common alternative choice for the upper limit of the support is $1/2$.

The first example illustrates the calculation of a spectral density function for one of the most basic time series models.

Example 9.43 Find the spectral density function for an ARMA(0, 0) time series model.

An ARMA(0, 0) model is simply white noise, so the population autocovariance function is

$$\gamma(k) = \begin{cases} \sigma_Z^2 & k = 0 \\ 0 & k = 1, 2, \dots \end{cases}$$

Using Definition 9.9, the spectral density function is

$$f(\omega) = \frac{1}{\pi} \left[\gamma(0) + 2 \sum_{k=1}^{\infty} \gamma(k) \cos(\omega k) \right] = \frac{\sigma_Z^2}{\pi} \quad 0 < \omega < \pi.$$

Figure 9.58 shows the spectral density function for the ARMA(0, 0) process. Since there is no cyclic variation whatsoever in the ARMA(0, 0) time series model, no frequency stands out over another, so the spectral density function is uniformly distributed of the frequencies between 0 and π . Each frequency on the interval $(0, \pi)$ contributes equally to the variance of X_t . Neither high frequencies nor low frequencies play a dominant role in the in terms of cyclic variation of this process. Notice that the area under $f(\omega)$ between 0 and π is $\sigma_X^2 = \sigma_Z^2$.

The next example calculates the spectral density function of an MA(1) model. This particular model was chosen because it has an autocovariance function that cuts off after lag 1, which means that the summation given in Definition 9.9 consists of just a single term.

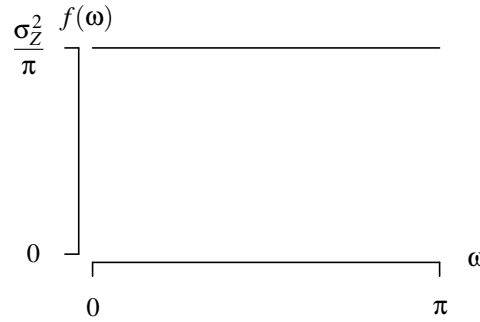


Figure 9.58: Spectral density function for an ARMA(0, 0) model.

Example 9.44 Find the spectral density function for an MA(1) time series model.

As derived in Section 9.2.1, the population autocovariance function of an MA(1) time series model is

$$\gamma(k) = \begin{cases} (1 + \theta^2) \sigma_Z^2 & k = 0 \\ \theta \sigma_Z^2 & k = 1 \\ 0 & k = 2, 3, \dots \end{cases}$$

Using Definition 9.9, the spectral density function is

$$f(\omega) = \frac{1}{\pi} \left[\gamma(0) + 2 \sum_{k=1}^{\infty} \gamma(k) \cos(\omega k) \right] = \frac{\sigma_Z^2}{\pi} [1 + \theta^2 + 2\theta \cos \omega] \quad 0 < \omega < \pi.$$

In order to develop some intuition about the spectral density function, consider two special cases of the MA(1) model: $\theta = 9/10$ and $\theta = -9/10$. These two values of θ correspond to stationary and invertible MA(1) time series models.

When $\theta = 9/10$, the spectral density function reduces to

$$f(\omega) = \frac{\sigma_Z^2}{\pi} \left[\frac{181}{100} + \frac{9}{5} \cos \omega \right] \quad 0 < \omega < \pi.$$

Figure 9.59 shows the spectral density function for an MA(1) model with $\theta = 9/10$. Since $\theta > 0$, the lag 1 population autocorrelation is positive, which means that a realization of this time series will linger above the mean for a few observations and then linger below the mean for a few observations. But the number of observations that the sequence lingers above or below the mean is random. This is the behavior that we saw in the simulated values in Example 9.33. In the simulated realization, sometimes the time series only lingers above or below the mean for just 2 or 3 simulated observations. In other cases, the time series lingers above or below the mean for 6 or 7 simulated observations. In other words, there is low-frequency variation in this time series, but it does not have a single consistent frequency. This pattern of lingering on one side of the mean corresponds to *low frequency* cycles, and those low frequency cycles correspond to smaller values of ω . This is reflected in the spectral density function in Figure 9.59, where the lower frequencies have larger values of $f(\omega)$ than the higher frequencies. Lower frequency variation dominates.

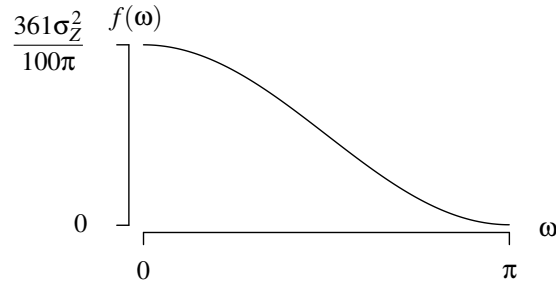


Figure 9.59: Spectral density function for an MA(1) model with $\theta = 9/10$.

When $\theta = -9/10$, the spectral density function reduces to

$$f(\omega) = \frac{\sigma_Z^2}{\pi} \left[\frac{181}{100} - \frac{9}{5} \cos \omega \right] \quad 0 < \omega < \pi.$$

Figure 9.60 shows the spectral density function for an MA(1) model with $\theta = -9/10$. Since $\theta < 0$, the lag 1 population autocorrelation is negative, which means that the observations in a realization of this time series will often jump from one side of the mean value to the other. This is the behavior that we saw with the simulated values in Example 9.34. In most cases, when one observation was on one side of the mean, the next observation was on the other side of the mean. Occasionally, however, the time series lingered for 2 or 3 observations on one side of the mean. Once again, this behavior is random and does not correspond to a single consistent frequency. This pattern of adjacent observations jumping from one side of the mean to the other corresponds to *high frequency* cycles, and those high frequency cycles correspond to larger values of ω . This is reflected in the spectral density function in Figure 9.60, where the higher frequencies have larger values of $f(\omega)$. Higher frequency variation dominates.

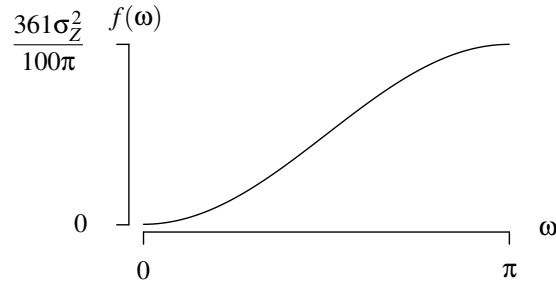


Figure 9.60: Spectral density function for an MA(1) model with $\theta = -9/10$.

One common element from the spectral density functions given in the two previous examples is that they both integrate to σ_X^2 . This is true in general. Some time series analysts prefer to divide the spectral density function by σ_X^2 so that it will integrate to 1, making it a true probability density function. The *normalized spectral density function* is given by

$$f^*(\omega) = \frac{f(\omega)}{\sigma_X^2} \quad 0 < \omega < \pi.$$

Dividing both sides of the equation in Definition 9.9 by $\gamma(0) = \sigma_X^2$ gives

$$f^*(\omega) = \frac{1}{\pi} \left[1 + 2 \sum_{k=1}^{\infty} \rho(k) \cos(\omega k) \right] \quad 0 < \omega < \pi.$$

The associated *normalized spectral cumulative distribution function* is defined on the support of ω in the usual fashion as

$$F^*(\omega) = \int_0^{\omega} f^*(w) dw \quad 0 < \omega < \pi.$$

One advantage of normalizing these two functions is that there is now a clean interpretation of $F^*(\omega)$. For frequencies ω_1 and ω_2 satisfying $0 < \omega_1 < \omega_2 < \pi$, the expression $F^*(\omega_2) - F^*(\omega_1)$ denotes the *proportion* of the variance in $\{X_t\}$ accounted for by frequencies on the interval (ω_1, ω_2) .

9.5.2 The Periodogram

The periodogram is the statistical counterpart to the spectral density function. The periodogram estimates the spectral density function for all frequencies between 0 and π . The shape of the periodogram reflects the frequencies that correspond to significant cyclic variation in a time series. Peaks in the periodogram reveal the dominant frequencies associated with cyclical components in an observed time series.

One topic that is crucial in time series analysis in the frequency domain is how often a time series should be sampled. Consider sampling the outdoor air temperature, for example, in Washington, DC. There are two significant cyclic components that should become apparent in such a time series. First, there is a daily temperature cycle. Temperatures are warmer during the day and cooler at night. This corresponds to high frequency variation. Second, there is an annual temperature cycle. Temperatures are warmer during the summer and cooler during the winter. This corresponds to low frequency variation. There is a factor of 365 (well, actually 365.24219) that separates the frequencies of these two cycles which should be accounted for in how often the time series is sampled. The following illustrations provide instances of sampling this time series too often, sampling this time series not often enough, and sampling this time series at about the right intervals to capture these two frequencies in a periodogram.

- Let's say you sample 1000 outdoor air temperatures at Reagan National Airport in Washington DC *every second* beginning at noon on July 20, 1969. This data collection will be over very soon because 1000 seconds is only about 17 minutes. But you have not covered a daily cycle or an annual cycle, so the frequencies for these two cycles cannot be detected from this sample. The sampling is too frequent.
- Let's say you sample 100 outdoor air temperatures at Reagan National Airport in Washington DC *annually* beginning at noon on July 20, 1969. This experiment will take you a long time to collect because the last value collected will be at noon on July 20, 2068. Even though you have collected the observations through 100 annual temperature cycles and tens of thousands of daily temperature cycles, neither the daily nor the annual cycle can be detected. All observations were made during the summer and during the day. The sampling was too infrequent.
- If you desire to detect both the daily and the annual outdoor air temperature cycles at Reagan National Airport, then a sampling interval between the two extremes (every second and every year from the previous two illustrations) must be used. So if you begin sampling *hourly* data at noon on July 20, 1969 and collect this data for three years, you will have collected outdoor

temperature observations over three annual cycles and about a thousand daily cycles. This requires $3 \cdot 365 \cdot 24 = 26,280$ outdoor air temperatures to be collected. This time series allows an analyst to detect both daily and annual cycles. The periodogram, which estimates the spectral density function will have a peak associated with the low frequency (annual) cycles and a second peak associated with the high frequency (daily) cycles.

The details associated with computing the periodogram are left for a full-semester class in time series analysis. Some of the fundamental ideas will be presented here in order to give a sense of the development of the estimator. As has been the case in regression and survival analysis, we begin with a model for a time series having cyclic behavior. One such model is

$$X_t = c \cdot \cos(\omega t + \phi),$$

where c is the amplitude of the cyclic variation, ω is the frequency of the cyclic variation, ϕ is a phase shift parameter, and the angle is measured in radians. (The ϕ used here has nothing to do with ϕ from the autoregressive time series models in the time domain.) Unfortunately, this model does not contain any random terms, and such a time series only occurs rarely in practice. So adding a time series of white noise $\{Z_t\}$ results in the much more practical model

$$X_t = c \cdot \cos(\omega t + \phi) + Z_t.$$

Since the phase shift parameter can be tedious in parameter estimation, it is common practice in spectral analysis to apply the trigonometric identity $\cos(x+y) = \cos x \cos y - \sin x \sin y$ to this model, which results in

$$X_t = a \cos(\omega t) + b \sin(\omega t) + Z_t,$$

where $a = c \cdot \cos(\phi)$ and $b = -c \cdot \sin(\phi)$. This result is symmetric in the two primary trigonometric functions sine and cosine. The derivation thus far has only involved a single frequency ω . As in the previous outdoor air temperature example, it is often the case that there are multiple frequencies of interest. The current time series model can be generalized by summing over the k frequencies $\omega_1, \omega_2, \dots, \omega_k$:

$$X_t = \sum_{j=1}^k (a_j \cos(\omega_j t) + b_j \sin(\omega_j t)) + Z_t,$$

where the amplitudes a_j and b_j reflect the contribution of frequency ω_j to the variability of X_t . For example, if $a_j = b_j = 0$ for one particular index j , then the associated frequency ω_j makes no contribution to the variability of X_t . The three remaining loose ends are (a) the number of frequencies k to consider, (b) which frequencies $\omega_1, \omega_2, \dots, \omega_k$ to consider, and (c) how to estimate the amplitudes $a_1, a_2, \dots, a_k$ and $b_1, b_2, \dots, b_k$. These loose ends are easier to navigate if the number of elements in the time series n happens to be even, which is assumed for now. If so, then the usual practice is to let $k = n/2$ and space the ω_j values uniformly between 0 and π as

$$\omega_m = 2\pi m/n \quad m = 1, 2, \dots, n/2.$$

The lowest frequency that can be detected by the periodogram is $\omega_1 = 2\pi/n$ and the highest frequency that can be detected by the periodogram is $\omega_{n/2} = \pi$, the Nyquist frequency. The periodogram can be calculated in R with the `spectrum` function, which is available in the base language. Periodograms often contain significant sampling variability and do not provide a consistent estimator of the spectral density function, so time series analysts often use various techniques to smooth the raw periodogram values.

Example 9.45 Conduct a Monte Carlo simulation experiment with 1000 replications that averages the periodograms associated with an MA(1) model with $\theta = -0.9$ and $\sigma_Z^2 = 1$ from a time series of $n = 100$ observed values.

This example uses the average of 1000 periodograms to estimate the spectral density function. The R code below uses the `arima.sim` function to generate 1000 time series of length $n = 100$ from an MA(1) time series model with $\theta = -0.9$ and $\sigma_Z^2 = 1$. The realizations of the time series are stored in the vector named `x`. The periodogram values are computed by the `spectrum` function and their cumulative values are stored in the vector named `s`. Setting the `plot` argument to `FALSE` suppresses the plots of the individual periodograms in the `for` loop. Setting the `spans` argument to a vector of odd integers smooths the periodogram values. The `spec` component of the list returned by the `spectrum` function returns the smoothed periodogram values. Finally, the `plot` function is used to plot the periodogram values. Since the `spectrum` function returns a support of $(0, 1/2)$, this is stretched to yield a support of $(0, \pi)$ in the final plot. The curve that is plotted is $k = n/2 = 100/2 = 50$ segments connecting the spectral density function estimates for the frequencies $\omega_1 = 2\pi/100$, $\omega_2 = 4\pi/100$, $\dots$, $\omega_{50} = \pi$, which are contained in the `freq` component of the list returned by the `spectrum` function.

```
set.seed(3)
s = numeric(50)
nrep = 1000
for (i in 1:nrep) {
  x = arima.sim(list(order = c(0, 0, 1), ma = -0.9), n = 100)
  s = s + spectrum(x, plot = FALSE, spans = c(3, 5))$spec
}
f = spectrum(x, plot = FALSE, spans = c(3, 5))$freq
plot(2 * pi * f, s / nrep, type = "l")
```

The average of the 1000 periodograms is plotted in Figure 9.61. As anticipated, the shape of the average of the periodograms is about the same as the shape of the spectral density function in Figure 9.60. The smoothness of the periodogram displayed in Figure 9.60 is deceiving. It is smooth because it is an average of 1000 periodograms. The individual periodograms generated within the `for` loop are very noisy, particularly when the smoothing parameters in the call to `spectrum` are eliminated. In a time series application, you will seldom work with the average of 1000 periodograms.

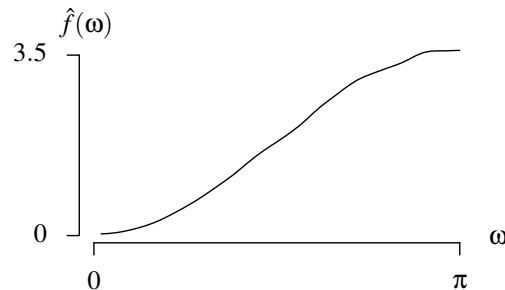


Figure 9.61: Periodogram averages for an MA(1) model with $\theta = -9/10$.

The previous example showed that the periodogram for the target MA(1) process, on average, appears to converge to the associated spectral density function. As illustrated in the final example, you will typically be working with just a single periodogram, which is typically quite noisy.

Example 9.46 Plot the periodogram associated with the annual lynx pelt sales at Hudson's Bay Company in Canada from 1857 to 1911. This data set was first encountered in Example 9.29.

The time series plot is given in Figure 9.62. There is a clear periodic component to the time series with a spike in sales every 9 or 10 years, which should be captured by the periodogram. The quickest way to generate a periodogram is with the R statements given below.

```
pelt = c(23362, 31642, 33757, 23226, 15178, 7272, 4448, 4926,
        5437, 16498, 35971, 76556, 68392, 37447, 45686, 7942,
        5123, 7106, 11250, 18774, 30508, 42834, 27345, 17834,
        15386, 9443, 7599, 8061, 27187, 51511, 74050, 78773,
        33899, 18886, 11520, 8352, 8660, 12902, 20331, 36853,
        56407, 39437, 26761, 15185, 4473, 5781, 9117, 19267,
        36116, 58850, 61478, 36300, 9704, 3410, 3774)
spectrum(pelt)
```

This code can be embellished a little to (a) avoid the special treatment of the Nyquist frequency, (b) extend the horizontal axis to π , (c) avoid the use of a logarithmic vertical axis, and (d) include some smoothing of the periodogram with the following additional R statements.

```
spec = spectrum(pelt, spans = c(3, 5), plot = FALSE)$spec
freq = spectrum(pelt, spans = c(3, 5), plot = FALSE)$freq
plot(2 * pi * freq, spec, type = "l")
```

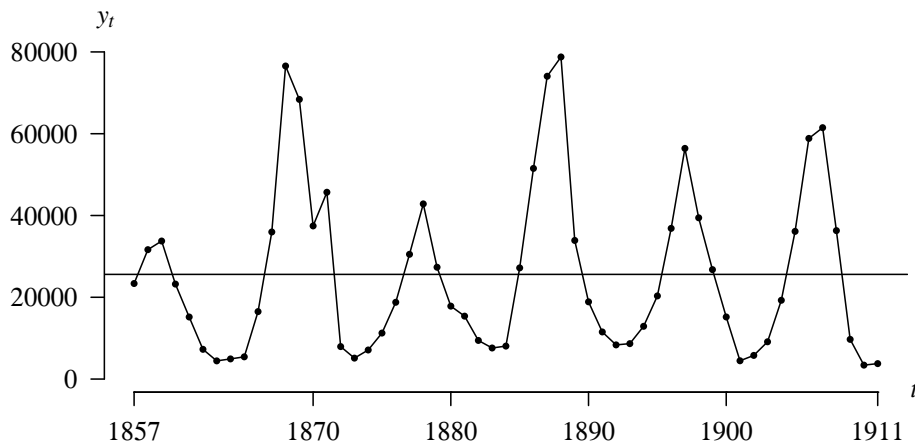


Figure 9.62: Time series plot for $n = 55$ annual lynx pelt sales (1857–1911).

The smoothed periodogram is given in Figure 9.63. There is a pronounced spike corresponding to a frequency of $\omega = 0.1 \cdot 2 \cdot \pi = 0.6283$. This corresponds to a period of

$$\frac{2 \cdot \pi}{0.1 \cdot 2 \cdot \pi} = 10,$$

which is consistent with the time series plot from Figure 9.62, which clearly displays a cycle of length 10.

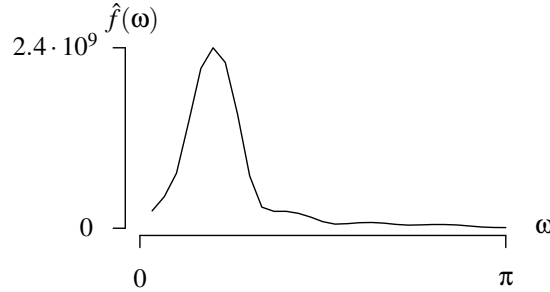


Figure 9.63: Periodogram for the lynx pelt sales.

9.6 Exercises

- 9.1 For a stationary AR(1) model, find $V[\bar{X}]$. Give an approximation for $V[\bar{X}]$ for large values of n .
- 9.2 Implement a Monte Carlo simulation that evaluates the method of moments, least squares, and maximum likelihood estimation techniques for an AR(1) model with $n = 100$ observed values and population parameters $\phi = -3/4$ and $\sigma_Z^2 = 1$ and identify the technique that has the smallest mean square error for estimating ϕ .
- 9.3 Consider a shifted AR(1) time series model with known parameter values μ , ϕ , and σ_Z^2 . One realization of the time series $x_1, x_2, \dots, x_{100}$ has been observed. Perform Monte Carlo simulation experiments that provide convincing numerical evidence that the exact two-sided 95% prediction intervals for X_{101} and X_{102} are indeed exact prediction intervals for parameter settings of your choice.
- 9.4 Consider a stationary shifted AR(1) model defined by

$$X_t = \mu + \phi X_{t-1} + Z_t,$$

where μ , $-1 < \phi < 1$, and $\sigma_Z^2 > 0$ are fixed known parameters and Z_t is Gaussian white noise. Find expressions for

- (a) $\lim_{h \rightarrow \infty} E[X_{n+h} | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n]$
 - (b) $\lim_{h \rightarrow \infty} V[X_{n+h} | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n]$.
- 9.5 Find the limiting half-width of a exact two-sided $100(1 - \alpha)\%$ prediction interval for $E[\hat{X}_{n+h}]$ as the time horizon $h \rightarrow \infty$ for an AR(1) time series model with all parameters known.

- 9.6** For a stationary shifted ARMA(p, q) time series model with population autocorrelation function $\rho(k)$, the population variance of the sample mean is

$$V[\bar{X}] = \frac{\sigma_X^2}{n} \left[1 + 2 \sum_{k=1}^{n-1} \left(1 - \frac{k}{n} \right) \rho(k) \right].$$

This result was proved in Section 8.2.1. Use this result to find an approximate 95% confidence interval for μ for the beaver data from Example 9.3 for a fitted shifted AR(1) time series model with Gaussian white noise error terms.

- 9.7** The built-in R time series `lh` consists of $n = 48$ observations of the luteinizing hormone in blood samples from a woman taken at 10 minute intervals.

- (a) Plot the time series, the sample autocorrelation function and the sample partial autocorrelation function.
- (b) Suggest an ARMA(p, q) model based on your plots.
- (c) Make a scatter plot of the data pairs (x_{t-1}, x_t) .
- (d) Compute the method of moments estimates of the parameters in the model suggested in part (b).
- (e) Compute the maximum likelihood estimates of the parameters in the model suggested in part (b).
- (f) Compute an approximate 95% confidence interval for ϕ .
- (g) Forecast the next three values in the time series and report 95% prediction intervals for the three forecasts.
- (h) Perform some research on the luteinizing hormone and indicate some scientific evidence that the time series model you suggested in part (b) is plausible.

- 9.8** Report the test statistic and p -value for the turning point test applied to the time series of beaver temperatures in their active state from Example 9.3. Comment on the sign of the test statistic and the magnitude of the p -value.

- 9.9** Consider the time series of $n = 70$ consecutive yields from a batch chemical process (from Box, G.E.P., and Jenkins, G.M. (1976), *Time Series Analysis: Forecasting and Control*, Revised Edition, Holden-Day, page 32) given in Example 7.20.

- (a) Plot the time series, the sample autocorrelation function and the sample partial autocorrelation function.
- (b) Suggest an ARMA(p, q) model based on your plots.
- (c) Make a scatter plot of the data pairs (x_{t-1}, x_t) .
- (d) Compute the method of moments estimates of the parameters in the model suggested in part (b).
- (e) Compute the maximum likelihood estimates of the parameters in the model suggested in part (b).
- (f) Compute an approximate 95% confidence interval for ϕ .
- (g) Forecast the next three values in the time series and report 95% prediction intervals for the three forecasts.

- 9.10** Consider an AR(1) model with population parameters $\phi = 0.8$ and $\sigma_Z^2 = 1$, and Gaussian white noise. Let r_1, r_2, r_3 denote the sample autocorrelation function values of the *residuals* of the fitted time series associated with $n = 100$ observations. Use Monte Carlo simulation to estimate the population mean vector and population variance–covariance matrix, to one-digit accuracy, of r_1, r_2, r_3 when maximum likelihood estimation is used to estimate the parameters.
- 9.11** Let B_1 and B_2 be the roots of the characteristic equation $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 = 0$ for an AR(2) time series model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t.$$

Let $G_1 = B_1^{-1}$ and $G_2 = B_2^{-1}$. A general solution for the lag k autocorrelation is (see Box, G.E.P., and Jenkins, G.M. (1976), *Time Series Analysis: Forecasting and Control*, Revised Edition, Holden–Day, page 59)

$$\rho(k) = \frac{(1 - G_2^2) G_1^{k+1} - (1 - G_1^2) G_2^{k+1}}{(G_1 - G_2)(1 + G_1 G_2)}$$

for $G_1 \neq G_2$. Show that calculating the population autocorrelation in this fashion is the same as using the recursive equation for the first five lags for an AR(2) process with parameters

- (a) $\phi_1 = 1/2, \phi_2 = 1/3$,
- (b) $\phi_1 = 1, \phi_2 = -1/2$.

- 9.12** Create a plot like the one in Figure 9.13 for an AR(2) model stationary region with $\rho(1) = -0.9, -0.8, \dots, 0.9$ and $\rho(2) = -0.9, -0.8, \dots, 0.9$. No labels are necessary on your plot.
- 9.13** A stationary AR(2) time series model can be written as an MA(∞) time series model. The coefficients $\theta_1, \theta_2, \dots$ in the MA(∞) model can be calculated in four fashions. First, they can be calculated using the recursive formulas in Theorem 9.12. Second, they can be written explicitly as (Cryer, J.D. and Chan, K–S, *Time Series Analysis: With Applications in R*, 2008, Springer, page 75):

$$\theta_i = \begin{cases} (i+1)G_1^i & \phi_1^2 + 4\phi_2 = 0 \\ (G_1^{i+1} - G_2^{i+1}) / (G_1 - G_2) & \phi_1^2 + 4\phi_2 > 0 \\ R^i \sin[(i+1)\Theta] / \sin \Theta & \phi_1^2 + 4\phi_2 < 0 \end{cases}$$

for $i = 1, 2, \dots$, where B_1 and B_2 are the roots of $\phi(B) = 1 - \phi_1 B - \phi_2 B^2$, $G_1 = B_1^{-1}$, $G_2 = B_2^{-1}$, $R = \sqrt{-\phi_2}$, and $\cos \Theta = \phi_1 / (2R)$. Third, the coefficients can be calculated by using the factored form of the characteristic polynomial, and writing the model in terms of X_t and equating coefficients. Fourth, the coefficients can be calculated by using the `ARMAtoMA` function in R. Calculate the first eight coefficients of the MA(∞) model, $\theta_1, \theta_2, \dots, \theta_8$, using these four methods for the following sets of AR(2) parameters:

- (a) $\phi_1 = 1, \phi_2 = -1/4$,
- (b) $\phi_1 = 1/2, \phi_2 = 1/9$,
- (c) $\phi_1 = 1, \phi_2 = -1/2$.

These three parameter combinations correspond to one real root with multiplicity two, two distinct real roots, and two complex roots of the characteristic equation $\phi(B) = 0$.

- 9.14** For an AR(2) time series model, the asymptotic variance–covariance matrix of the maximum likelihood estimates $\hat{\phi}_1$ and $\hat{\phi}_2$ is

$$\frac{1}{n} \begin{bmatrix} 1 - \phi_2^2 & -\phi_1(1 + \phi_2) \\ -\phi_1(1 + \phi_2) & 1 - \phi_2^2 \end{bmatrix}.$$

What is the asymptotic population correlation between $\hat{\phi}_1$ and $\hat{\phi}_2$?

- 9.15** Consider an AR(2) time series model with $\phi_1 = 1$, $\phi_2 = -1/2$, and $\sigma_Z^2 = 1$. For a realization of $n = 100$ observations $X_1, X_2, \dots, X_{100}$ from this AR(2) model, give convincing numerical evidence that the forecasted value for X_{103} is unbiased and that the 95% prediction interval for X_{103} is exact.
- 9.16** Implement Theorem 9.17 on the R built-in `LakeHuron` time series to calculate the first five forecasted values and associated prediction intervals. Do not just use the `predict` function.
- 9.17** Consider a standard AR(2) model for an observed time series of $n = 100$ values. The last two values in the time series are $x_{99} = 3$ and $x_{100} = 4$. The estimated coefficients in the AR(2) model are $\hat{\phi}_1 = 1$ and $\hat{\phi}_2 = -0.5$. Compute the next ten forecasted values $\hat{X}_{101}, \hat{X}_{102}, \dots, \hat{X}_{110}$ and comment on the shape of the forecasted values.
- 9.18** Consider a realization $x_1, x_2, \dots, x_n$ of a stationary shifted AR(2) time series model with fixed known parameters μ, ϕ_1, ϕ_2 , and σ_Z^2 . Write a formula for $\hat{X}_{n+3}$ in terms of x_{n-1} and x_n .
- 9.19** Consider the annual Lake Huron water level heights from 1875 to 1972 given in the R built-in data set `LakeHuron`, appended by the next ten observations,

580.98, 581.04, 580.49, 580.52, 578.57, 578.96, 579.94, 579.77, 579.44, 578.97,

for the years 1973 to 1982. Give the p -value associated with a test of the statistical significance of the slope of the simple linear regression line for the augmented time series.

- 9.20** Consider the AR(3) model with coefficients

$$\phi_1 = 3/2 \quad \phi_2 = -1 \quad \phi_3 = 1/4.$$

- (a) Is this model invertible?
- (b) Is this model stationary?
- (c) Calculate the first six coefficients of the associated MA(∞) model.

- 9.21** Two necessary, but not sufficient, conditions for stationarity of an AR(p) time series model are (Cryer, J.D. and Chan, K-S, *Time Series Analysis: With Applications in R*, 2008, Springer, page 76):

$$\phi_1 + \phi_2 + \dots + \phi_p < 1 \quad \text{and} \quad |\phi_p| < 1.$$

- (a) Show that these conditions hold for the stationary AR(4) time series model with

$$\phi_1 = \frac{21}{20}, \phi_2 = \frac{1}{20}, \phi_3 = -\frac{23}{40}, \phi_4 = \frac{3}{10}.$$

- (b) Graphically or algebraically, show that these conditions are necessary but not sufficient for falling in the triangular-shaped stationary region from Theorem 9.9 for an AR(2) time series model.

- 9.22** Consider the AR(4) time series model with characteristic polynomial

$$\phi(B) = 1 - \frac{21}{20}B - \frac{1}{20}B^2 + \frac{23}{40}B^3 - \frac{3}{10}B^4$$

and Gaussian white noise with population variance $\sigma_Z^2 = 1$. Conduct a Monte Carlo simulation experiment that provides convincing numerical evidence that $\gamma(0) = 3520/819$.

- 9.23** The R vector `phi` contains the parameters $\phi_1, \phi_2, \dots, \phi_p$ in an AR(p) model. Write an R function named `is.stationary` with a single parameter `phi` that returns TRUE if the AR(p) model is stationary and FALSE otherwise.
- 9.24** The R code below takes initial p autocovariances $\gamma(0), \gamma(1), \dots, \gamma(p-1)$ for an AR(p) model, which are stored in the vector `gam`, and places them in a variance-covariance matrix `GAMMA` (denoted by Γ in the text).

```
GAMMA = matrix(0, p, p)
for (i in 1:p) {
  for (j in 1:p) {
    GAMMA[i, j] = gam[abs(i - j) + 1]
  }
}
```

The code makes this conversion by using two nested `for` loops. Heather can do this calculation without using `for` loops. How does she do it?

- 9.25** Consider a time series that is governed by an AR(4) model with characteristic polynomial

$$\phi(B) = 1 - \frac{21}{20}B - \frac{1}{20}B^2 + \frac{23}{40}B^3 - \frac{3}{10}B^4$$

and Gaussian white noise with population variance $\sigma_Z^2 = 1$. Conduct a Monte Carlo simulation experiment that provides convincing numerical evidence that the 95% confidence interval for ϕ_3 based on the maximum likelihood estimators for an AR(4) time series model is asymptotically exact.

- 9.26** For logarithms of the $n = 55$ annual lynx pelt sales time series from Example 9.29, find the values of p and q associated with the ARMA(p, q) model that minimizes the AIC statistic. Assume that the models are fitted by maximum likelihood.
- 9.27** Fit the AR(4) model to the logarithms of the $n = 55$ annual lynx pelt sales time series from Example 9.29 by maximum likelihood. Simulate the fitted model to generate $n = 55$ random annual lynx pelt sales from the fitted model. View a dozen or so such realizations and comment on your faith in the fitted AR(4) time series model. Repeat the experiment for a fitted ARMA(2, 3) time series model and comment.

- 9.28** Show that the MA(1) model

$$X_t = Z_t + \theta Z_{t-1},$$

and the MA(1) model

$$X_t = Z_t + \frac{1}{\theta} Z_{t-1}$$

have the same population autocorrelation function.

- 9.29** Show that $-1/2 \leq \rho(1) \leq 1/2$ for an MA(1) model.
- 9.30** Derive the population autocorrelation function for the MA(1) model with arbitrary mean value μ given by

$$X_t = \mu + Z_t + \theta Z_{t-1},$$

in a similar fashion to the derivation for the standard MA(1) model.

- 9.31** Conduct the following Monte Carlo simulation experiment. Generate $n = 100$ observations from an MA(1) time series model with $\theta = 0.9$ and standard normal white noise terms. Estimate the expected value and standard deviation of r_1 and r_2 . Run enough replications to that you can report your estimates to two significant digits.
- 9.32** Consider an MA(1) model with $\theta = -0.9$ and Gaussian white noise with $\sigma_Z^2 = 1$. Generate a dozen realizations of this time series for $n = 100$ observations each. Plot the time series and the associated correlogram, using a call to `Sys.sleep` between each realization to view the graphs. Write a paragraph that describes what you observe in the dozen realizations.
- 9.33** Consider an MA(1) time series model

$$X_t = Z_t + \theta Z_{t-1},$$

where $\{Z_t\}$ denotes Gaussian white noise. Let $\hat{\theta}_{\text{MOM}}$ be the method of moments estimator of θ and let $\hat{\theta}_{\text{MLE}}$ be the maximum likelihood estimator of θ . One way to compare these two estimators is the asymptotic relative efficiency, defined as

$$\lim_{n \rightarrow \infty} \frac{V[\hat{\theta}_{\text{MOM}}]}{V[\hat{\theta}_{\text{MLE}}]}.$$

Brockwell and Davis (2016, page 129) give the population variance of $\hat{\theta}_{\text{MOM}}$ and $\hat{\theta}_{\text{MLE}}$ as approximately

$$V[\hat{\theta}_{\text{MOM}}] \cong \frac{1 + \theta^2 + 4\theta^4 + \theta^6 + \theta^8}{n(1 + \theta^2)^2} \quad \text{and} \quad V[\hat{\theta}_{\text{MLE}}] \cong \frac{1 - \theta^2}{n}.$$

Write a Monte Carlo simulation that confirms these two formulas for $n = 400$, $\theta = 1/2$, and $\sigma_Z^2 = 1$.

- 9.34** The $n = 45$ daily average number of defects per truck at the final inspection at a manufacturing facility (from Burr, 1976, *Statistical Quality Control Methods*, Marcel Dekker, New York), read row-wise, are given below.

1.20	1.50	1.54	2.70	1.95	2.40	3.44	2.83	1.76
2.00	2.09	1.89	1.80	1.25	1.58	2.25	2.50	2.05
1.46	1.54	1.42	1.57	1.40	1.51	1.08	1.27	1.18
1.39	1.42	2.08	1.85	1.82	2.07	2.32	1.23	2.91
1.77	1.61	1.25	1.15	1.37	1.79	1.68	1.78	1.84

Fit these data values to a shifted MA(1) time series model by the method of moments, least squares, and maximum likelihood estimation.

- 9.35** The formula for the population variance of the sample mean for a stationary time series model (which was proved in Section 8.2.1) is

$$V[\bar{X}] = \frac{\sigma_X^2}{n} \left[1 + 2 \sum_{k=1}^{n-1} \left(1 - \frac{k}{n} \right) \rho(k) \right]$$

Show that this is approximately

$$V[\bar{X}] \cong \frac{\sigma_X^2}{n} \left[1 + 2 \sum_{k=1}^{\infty} \rho(k) \right]$$

or, equivalently,

$$V[\bar{X}] \cong \frac{\sigma_X^2}{n} \left[\sum_{k=-\infty}^{\infty} \rho(k) \right]$$

for large values of n whenever the autocorrelation function values decay rapidly enough with increasing k such that

$$\sum_{k=1}^{\infty} |\rho(k)| < \infty.$$