ARIMA and Seasonal Models 7

Chapters 5 and 6 focused on ARMA models (along with their special cases, AR and MA models) and how these models are used for modeling and forecasting *stationary time series data*. The assertion was made that many (or most) time series do not satisfy the conditions of stationarity given in Section 3.3, and consequently there is a need for time series models that focus on modeling nonstationary behavior in time series. There are several ways that a time series can be nonstationary. Among these violations of stationarity are:

1. *The mean changes with time:* This often occurs when there is a deterministic component such as a linear trend. This is the topic of Section 8.1.
2. *The variance changes with time:* The Air Passenger data first shown in Figure 1.5 are an example of a time series for which the variance seems to be increasing in time.
3. *The correlation structure depends on time*: An example of this is the bat echolocation data and the seismic data in Figure 3.15(a) and (b), respectively.[1]
4. *The AR-characteristic equation has one or more roots on the unit circle:* Recall that Theorem 5.7 states that an ARMA(p,q) process is stationary if and only if all the roots of the AR-characteristic equation fall outside the unit circle.

In this chapter we will consider models which are nonstationary due to "Violation 4" with some discussion of "Violation 2". Chapter 8: Time Series Regression will discuss models for time series whose mean value changes with +time.

7.1 ARIMA(*p, d, q*) MODELS

The ARIMA(p,d,q) models have proven to be very useful in practice for modeling data that have a wandering behavior but do not seem to have an attraction to a process mean. This model is very popular in the modeling of economic time series, but the applications of these models are extensive. In Section 7.1.1 we define the ARIMA(p,d,q) model and discuss its properties while in Section 7.1.2 we describe how to fit an ARIMA model to a set of time series data.

1 A broad class of time series which have autocorrelations that depend on time are the G-stationary time series discussed in Chapter 13 of Woodward et al. (2017). These models will not be discussed in this text, but readers who want more information on G-stationary processes should refer to Woodward et al. (2017) and research articles referenced therein.

DOI: 10.1201/9781003089070-7

7.1.1 Properties of the ARIMA(*p,d,q*) Model

The process, X_t, is an ARIMA(p,d,q) process if $Y_t = (1-B)^d X_t$ satisfies a stationary ARMA(p,q) model. An ARIMA(p,d,q) model is referred to as an *autoregressive integrated moving average process of orders* $p, d,$ *and* q.The operator $(1-B)^d$ is defined in Definition 7.1.

Definition 7.1: $(\mathbf{1}-\boldsymbol{B})^{\boldsymbol{d}}$ The operator $(1-B)^d$ is the dth difference operator where d is a non-negative integer.

(a) If $d = 1$ then $(1-B)^d = (1-B)$ and $(1-B)X_t = X_t - X_{t-1}$.
(b) If $d = 2$ then $(1-B)^d = (1-B)^2 = (1-B)(1-B) = 1 - 2B + B^2$ so that
$$(1-B)^2 X_t = (1 - 2B + B^2)X_t = X_t - 2X_{t-1} + X_{t-2}.$$
(c) If $d = 3, 4, \ldots,$ then $(1-B)^d$ is defined analogously.
(d) If $d = 0$, then $(1-B)^0 X_t = X_t$.

Definition 7.2: The ARIMA(p,d,q) model is defined by

$$\phi(B)(1-B)^d X_t = \theta(B)a_t, \tag{7.1}$$

where $\phi(B)$ and $\theta(B)$ are "stationary and invertible" operators for which all roots of the characteristic equations $\phi(z) = 0$ and $\theta(z) = 0$ lie outside the unit circle. Note that the "autoregressive part" of this model is $\phi(B)(1-B)^d$.

7.1.1.1 Some ARIMA(p,d,q) Models

The following are examples of ARIMA(p,d,q) models which we will discuss in more detail in the following and will be referred to as Models (a), (b), and (c), respectively.

(a) $(1-B)X_t = a_t$: This is an ARIMA(0,1,0), the simplest of all nonstationary ARIMA(p,d,q) models.
(b) $(1 - 1.3B + .65B^2)(1-B)X_t = a_t$: An ARIMA(2,1,0) model with $\phi(B) = 1 - 1.3B + .65B^2$.
(c) $(1 - 1.3B + .65B^2)(1-B)^2 X_t = a_t$: An ARIMA(2,2,0) model with $\phi(B) = 1 - 1.3B + .65B^2$.

Key Points:

1. ARIMA(p,d,q) processes with $d > 0$ have the property that one or more roots of the autoregressive characteristic equation are ***on the unit circle***. We say that these models have ***unit roots.***
2 Because of the unit root(s), these models do not satisfy the property of stationary ARMA models which require all roots of the autoregressive characteristic equation to be outside the unit circle. However, these processes are on the "border" of the stationary region.

7.1.1.2 Characteristic Equations for Models (a)–(c)

(a) The ARIMA(0,1,0) model $(1-B)X_t = a_t$ has the associated characteristic equation $1-z=0$, which has a single root $r=1$. Because this root is *on* the unit circle, the ARIMA(0,1,0) model is not a stationary AR(1) model.

(b) The autoregressive part of the model in (b) is $(1-1.3B+.65B^2)(1-B)$ so the autoregressive characteristic equation is $(1-1.3z+.65z^2)(1-z)=0$. Based on this factorization of the characteristic equation we see that $z = 1, 1+0.73i$, and $1–0.73i$ are solutions, so the process is nonstationary with a root of $r=1$, which is on the unit circle and a complex conjugate pair that are outside the unit circle.

(c) The autoregressive component in (c) is $(1-1.3B+.65B^2)(1-B)^2$ so the autoregressive characteristic equation is $(1-1.3z+.65z^2)(1-z)^2=0$ which has two roots of +1 on the unit circle.

(1) Model (a) ARIMA(0,1,0): $(1-B)X_t = a_t$

We now consider the simple ARIMA$(0,1,0)$ model

$$(1-B)X_t = a_t. \tag{7.2}$$

This has the appearance of an AR(1) model with $\phi_1 = 1$. However, because the characteristic equation $1-z=0$ has a root $r=1$, which is *on* the unit circle, the ARIMA(0,1,0) model is nonstationary as discussed above. However, every AR(1) model for which ϕ_1 is less than 1 in absolute value corresponds to a stationary model, no matter how close ϕ_1 is to 1. For example, $(1-.999B)X_t = a_t$ is stationary while $(1-B)X_t = a_t$ is not (see Problem 7.2).

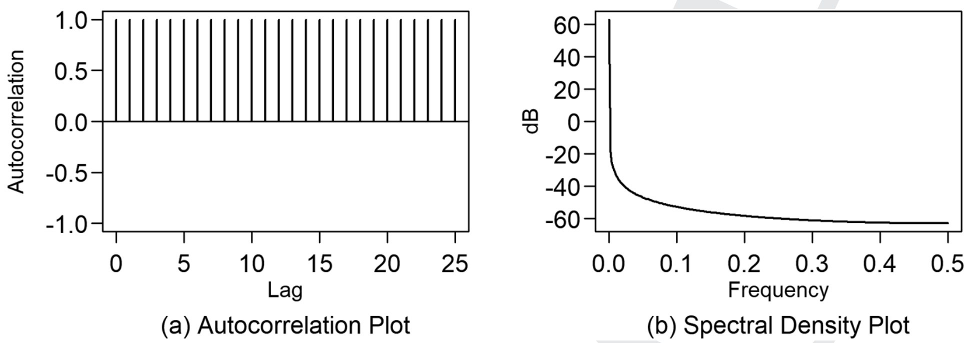
7.1.1.3 Limiting Autocorrelations

From Equation 5.6 we know that for the stationary AR(1) process $(1-\phi_1 B)(X_t-\mu) = a_t$, with $|\phi_1|<1$, the variance $\sigma_X^2\ (=\gamma_0)$ is given by $\gamma_0 = \frac{\sigma_a^2}{1-\phi_1^2}$. Thus, it follows that the process variance $\sigma_X^2 \to \infty$ as $\phi_1 \to 1$. The autocovariance, γ_k for $k>0$,of a stationary AR(1) process is given by $\gamma_k = \gamma_0\phi_1^k$ which also approaches ∞ as $\phi_1 \to 1$. Because the autocorrelation of an AR(1) is $\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{\gamma_0\phi_1^k}{\gamma_0} = \frac{\infty}{\infty}$ if $\phi_1^k = 1$, we clearly need another definition for the autocorrelation function when $\phi_1 = 1$. Note that if $|\phi_1|<1$ then $\gamma_0 = \sigma_X^2$ is finite. So, $\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{\gamma_0\phi_1^k}{\gamma_0} = \phi_1^k$ for each $|\phi_1|<1$ as $\phi_1 \to 1$. Consequently, it is reasonable to define the autocorrelation function for the limiting case, that is, $\phi_1 = 1$, as the limit of $\rho_k = \phi_1^k$ as $\phi_1 \to 1$. That is, the "autocorrelation" for $\phi_1 = 1$ is defined to be $1^k = 1$ for each k.

Key Point: In general, for ARIMA and seasonal models, *the autocorrelation is defined as a limit.* These limiting autocorrelations are also referred to as "extended autocorrelations" and are denoted by ρ_k^* in Chapter 5 of Woodward et al., 2017. In the current book we will continue to simply refer to them as autocorrelations and denote them by ρ_k.

Also, note that the spectral density of the ARIMA(0,1,0) has a peak at zero that is infinite. Figure 7.1(a) shows the (limiting) autocorrelations which are $\rho_k = 1$ for all integer k. Figure 7.1(b) is a plot of the spectral density for this model, for which the peak in the spectral density is technically infinite (which we cannot plot). The ***tswge*** package doesn't have a routine for plotting the autocorrelations and spectral densities for nonstationary processes due to the unbounded behavior associated with them. Consequently, the plots in Figure 7.1 were obtained by using a very nearly nonstationary model.[2]

```
plotts.true.wge(phi=.99999,plot.data=FALSE)
```



QR 7.1
Explosive
Behavior

FIGURE 7.1 Model based (a) autocorrelations and (b) spectral density for $(1-B)X_t = a_t$.

Figure 7.2(a) is a plot of a realization of length $n = 200$ from the ARIMA(0,1,0) model along with the sample autocorrelations and Parzen spectral density estimate. The realization has the wandering behavior typical for an AR(1) model with ϕ_1 close to one. Recall that Figure 5.6(a) is a plot of a realization from the nonstationary model in (7.1). The realizations in Figure 5.6(a) and Figure 7.2(a) are similar in that the data seem to be wandering. The data in Figure 5.6(a) show an overall "upward" wandering while Figure 7.2(a) shows a downward trending behavior. In Figure 7.2(b) we show the sample autocorrelations, and note that they are not identically equal to one for all lags. However, the sample autocorrelations for small lags are quite close to one and the damping is relatively slow. The Parzen spectral density estimate has a sharp peak at $f = 0$, which is finite. The realization in Figure 7.2(a) was generated using the ***tswge*** command **`gen.arima.wge`** and the plots below were obtained using the commands[3]

```
x010=gen.arima.wge(n=100,d=1,sn=409)
plotts.sample.wge(x010)
```

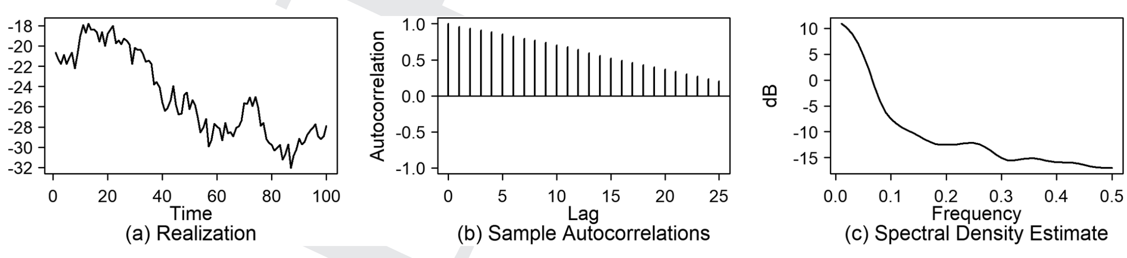


FIGURE 7.2 (a) Realization, (b) sample autocorrelations, and (c) Parzen spectral density estimate for a realization of length $n = 100$ from an ARIMA(0,1,0) model.

2 Note that for the stationary model $(1-.9999B)X_t = a_t$ it follows that $\rho_{25} = \phi_1^{25} = .9999^{25} = .9975$. So, although the true autocorrelations for an ARIMA(0,1,0) model are all equal to one, the autocorrelations above are visually indistinguishable from one. Similarly, the spectral density (dB) at $f = 0$ is quite large but not infinite.

3 The ***tswge*** function **`gen.arima.wge`** generates a realization from an ARIMA model. It is described in Appendix 7A.

Key Point: A realization with wandering behavior, slowly damping sample autocorrelations, and a peak in the Parzen spectral density estimate at $f = 0$ "suggests" that a $(1-B)$ factor may be present in the data. Note the quotes around "suggests". We will return to this issue in Section 7.1.2 when we discuss model identification.

7.1.1.4 Lack of Attraction to a Mean

Recall that in the discussion of stationary ARMA models, it was noted that one of their features was that realizations tend to show an "attraction to the mean". That is, if $\mu = 50$ is the theoretical mean of a stationary ARMA model, then realizations from that model that drift very far from the mean (50) will eventually "be attracted back toward the mean". In fact, for sufficiently long realizations from a stationary model, the sample mean, $\bar{X} = \frac{1}{n}\sum_{t=1}^{n} X_t$, will tend to be a good estimator of the true mean.

You may have noticed that the ARIMA(p,d,q) processes in (7.1) and (7.2) did not contain a mean term which are in the models for stationary models. To understand why, recall that the AR(1) model, $(1-\phi_1 B)(X_t - \mu) = a_t$, is sometimes written as $X_t - \phi_1 X_{t-1} = (1-\phi_1)\mu + a_t$ where $(1-\phi_1)\mu$ is referred to as the moving average constant. Consider the ARIMA(0,1,0) model written to include a mean, that is $(1-B)(X_t - \mu) = a_t$. The moving average constant is $(1-\phi_1)\mu = (1-1)\mu = 0$, regardless of μ. Consequently, the model $(1-B)(X_t - \mu) = a_t$ doesn't actually depend on μ. Consequently, the models $(1-B)(X_t - 50) = a_t, s(1-B)(X_t + 600) = a_t$, and $(1-B)X_t = a_t$ are all the same. Thus, it is common to write the ARIMA model without the mean term as we have done in (7.1) and (7.2).

Figure 7.3(a) shows a realization of length $n = 10{,}000$from the stationary (but near nonstationary) model $(1-.999B)(X_t - 50) = a_t$, generated using the command

```
x9=gen.arma.wge(n=10000,phi=.999,mu=50,sn=509)
```

Note that whenever values of the realization tend to stray very far from the mean of 50 (shown on the graph as a horizontal line), there will eventually be a tendency (especially visible in longer realizations) for the values to return toward 50. Figure 7.3(b) is a realization of length $n = 10{,}000$ from the nonstationary model $(1-B)(X_t - 50) = a_t$, generated using the command

```
x1=gen.arima.wge(n=10000,d=1,mu=50,sn=340)
```

where of course 50 is the "mean in name only", and it is reasonable to question the meaning of **`mu`** in the call statement. Function **`gen.arma.wge`** generates realizations by setting the first p data values equal to **`mu`**. Similarly, in **`gen.arima.wge,`** the first $p+d$ observations are set equal to the value of **`mu`**.The functions then generate 2000 data values using the recursive relationship associated with the model. The final series of length n returned from the functions are the data values $2001, 2002, \ldots, 2000+n$. Using this procedure, the stationary realization in Figure 7.3(a) did indeed tend to wander around 50 as previously mentioned, but the nonstationary realization Figure 7.3(b) shows no attraction to the "mean" (50) and in fact 50 is not even on the vertical scale.

QR 7.2
Nonstationary
Wandering

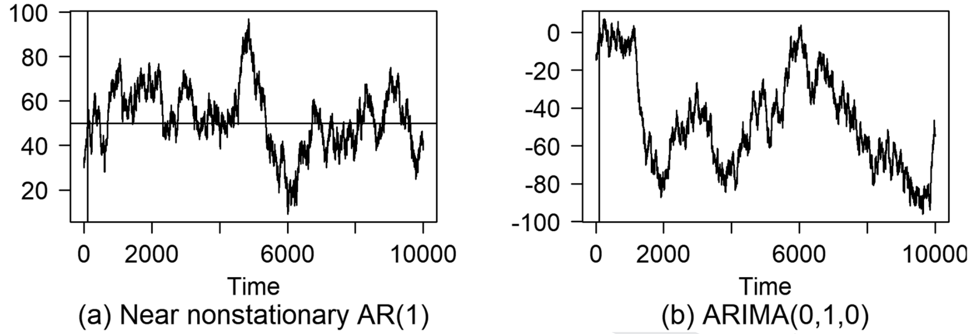
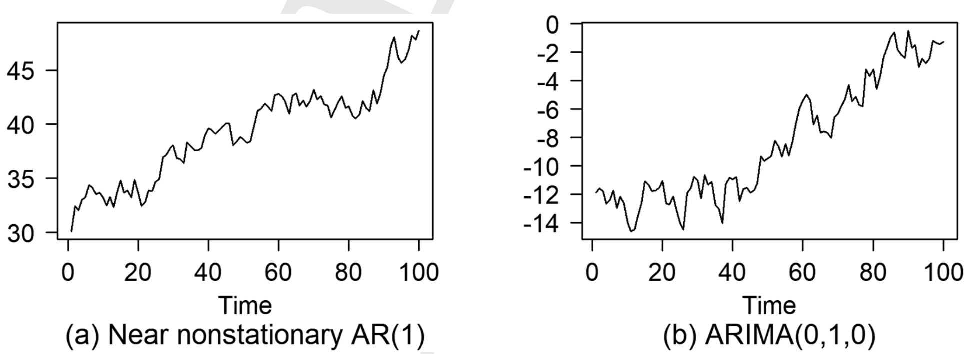


FIGURE 7.3 Realizations of length $n = 10{,}000$ from (a) $(1 - .999B)(X_t - 50) = a_t$ and (b) $(1 - B)(X_t - 50) = a_t$.

7.1.1.5 Random Trends

Figures 7.4(a) and (b) show the first 100 observations from the corresponding plots in Figure 7.3. Because of the length of the realizations plotted in Figure 7.3, it is difficult to "get a good look" at the first 100 observations. The plots in Figure 7.4 are the observations to the left of the vertical lines at $t = 100$ in the plots in Figure 7.3 (trust us!). The point is that ARIMA(0,1,0) models and nearly nonstationary AR(1) models produce *random trends*. The longer realizations in Figure 7.3 show that these random trends may result in significant time intervals in which the data are "linear trending up" or "linear trending down". There also seems to be no "warning" that a turn in direction is imminent. The longer realizations in Figures 7.3(a) and (b) make it clear that it would have been a mistake to predict the trends visible in the first one hundred time values to continue. In fact, if the realizations in Figures 7.4(a) and (b) were all that were available to the analyst, it would be difficult (or maybe impossible) to determine whether the visible trend is due to an actual deterministic trend component in the data (that we would be tempted to predict would continue into the future). The decision whether to predict an observable trend to continue into the future must often be made by the analyst based on domain knowledge. We will discuss this further in the context of forecasting in Section 7.1.3. The "partial realizations" in Figure 7.4 were obtained using the commands

```
plotts.wge(x9[1:100])
plotts.wge(x1[1:100])
```



QR 7.3
Random
Trends

FIGURE 7.4 Realizations of length $n = 100$ from (a) $(1 - .999B)(X_t - 50) = a_t$ and (b) $(1 - B)(X_t - 50) = a_t$.

Key Points:

1. There is no attraction to a mean for realizations from the ARIMA(0,1,0) model in (7.2). This is a common characteristic of all ARIMA(p,d,q) models.
2. "Shorter" realization lengths from an ARIMA(0,1,0) (and an AR(1) with ϕ_1 close to one) can show trending behavior, but these are "random trends" and should not be predicted to continue.
3. Figure 7.3(b) shows that due to the lack of an attraction to the mean, the random trending behavior in an ARIMA(0,1,0) model can be quite lengthy. Specifically, the realization has a long downward trend starting at about $t = 6{,}000$. However, this trend reverses dramatically at about $t = 9{,}900$.

Try This: One of the best ways to get a feel for the idea of random trends (as the case has been throughout the book) is to try it out! The following code could be used to generate Figures 7.4(a) and (b):

```
x9.100=gen.arma.wge(n=100,phi=.999,mu=50,sn=509)
x1.100=gen.arima.wge(n=100,d=1,mu=50,sn=340)
```

Take a few minutes and run these commands using different **sn** values (or removing the **sn=** definition to produce randomly selected realizations). Examine the realizations that the models produce, and decide whether many of these realizations (that only have random trends) would have led you to believe there was an underlying linear (deterministic) component in the generating model.

7.1.1.6 Differencing an ARIMA(0,1,0) Model

Suppose we "difference" a realization from the ARIMA(0,1,0) model $(1-B)X_t = a_t$. That is, we calculate $Y_t = X_t - X_{t-1}$. Note that the ARIMA(0,1,0) can be written as $(1-B)X_t = X_t - X_{t-1} = a_t$, and recalling that $Y_t = X_t - X_{t-1}$, it follows that $Y_t = a_t$. That is, if you difference data that are properly modeled as an ARIMA(0,1,0) model, the differenced data should have the appearance of white noise.

Note that for a realization $x_1, x_2, \ldots, x_n$, the "differenced" realization $y_t = x_t - x_{t-1}$ has only $n-1$ observations because $y_1 = x_1 - x_0$ cannot be computed (x_0 is not an observed data value). We often "violate" notation by referring to the differenced data as $y_1 = x_2 - x_1,\ y_2 = x_3 - x_2, \ldots, y_{n-1} = x_n - x_{n-1}$.[4] The command **`artrans.wge`** in ***tswge*** performs "autoregressive transformations". These are transformations of the form

$$Y_t = X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p.}$$

As an example, in order to apply the transformation $Y_t = X_t + .2X_{t-1} - .48X_{t-2}$ to a dataset **x** in R, the command would be

```
y=artrans.wge(x,phi.tr=c(-.2,.48))
```

This command takes the data in realization **x**, applies the transformation $Y_t = X_t + .2X_{t-1} - .48X_{t-2}$ to the data and saves the transformed data in **y**. For the simple difference transformation, $Y_t = X_t - X_{t-1}$, it follows that **`phi.tr=1.`**

4 Actually, by definition, $y_1 = x_1 - x_0$, $y_2 = x_2 - x_1, \ldots, y_n = x_n - x_{n-1}$. The first value, $y_1 = x_1 - x_0$, cannot be calculated, leaving us with the realization, $y_2, y_3, \ldots, y_n$. Our "violation" of notation is, for notational convenience, to refer to the y_t realization as $y_1, y_2, \ldots, y_{n-1}$.

The differences of the data in Figure 7.2(a) are obtained using the command

```
x010=gen.arima.wge(n=100,d=1,sn=409)
d1=artrans.wge(x010,phi.tr=1)
```

Figure 7.5 shows a plot of the differences (**d1**) of the ARIMA(0,1,0) data in Figure 7.5(a) along with the sample autocorrelations and Parzen spectral density estimate of the differenced data. The data appear uncorrelated and the sample autocorrelation plot in Figure 7.5(b) includes the 95% limit lines as a reference. All of the sample autocorrelations are within the 95% limit lines although $\hat{\rho}_1$ and $\hat{\rho}_{13}$ are close to the boundaries. The Parzen spectral density estimate in Figure 7.5(c) is fairly flat. The plots in Figure 7.5 are consistent with white noise. Figure 7.5 is obtained using the code

```
x010=gen.arima.wge(n=100,d=1,sn=409,plot=FALSE)
d1=artrans.wge(x010,phi.tr=1)
plotts.sample.wge(d1)
```

> **Key Point:** If you difference a realization of length n from an ARIMA(0,1,0) model, the differenced data should appear to be a white noise realization of length $n-1$.

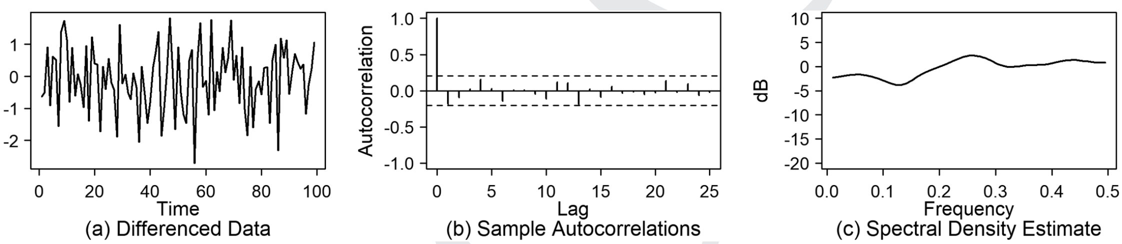


FIGURE 7.5 (a) Data in Figure 7.2(a) differenced, (b) sample autocorrelations, and (c) Parzen spectral density estimate for the differenced data in (a).

7.1.1.7 ARIMA Models with Stationary and Nonstationary Components

To this point we have considered the ARIMA(0,1,0) model in some detail. In this section we will consider ARIMA(p,d,q) models that have both stationary and nonstationary components. We note that Models (b) and (c) are both ARIMA($2,d,0$) models containing the stationary AR component $1-1.3B+.65B^2$. Before examining Models (b) and (c) we first consider the stationary AR(2) model

$$\left(1-1.3B+.65B^2\right)X_t = a_t.$$

7.1.1.8 The Stationary AR(2) Model: $(1\text{-}1.4B+.65B^2)X_t = a_t$

The factor table in Table 7.1 shows that this AR(2) model is associated with a system frequency of about $f=.10$, and that the roots of the characteristic equation are outside, but not very close to, the unit circle. Specifically, $|r|^{-1}=.81$.

TABLE 7.1 Factor Table for $\left(1-1.3B+.65B^2\right)X_t = a_t$

AR-FACTOR	*ROOTS*	$\lvert r\rvert^{-1}$	f_0
$1-1.3B+.65B^2$	$1.0\pm.73i$	.81	.10

Figure 7.6 shows a realization of length $n = 200$ along with the associated sample autocorrelations and Parzen spectral density estimate. The cyclic realization of length $n = 200$ with about 20 periods, the sample autocorrelations, and especially the Parzen spectral density estimate all indicate that the system frequency is about $f = .10$. We will refer back to Figure 7.6 and Table 7.1 as we analyze Models (b) and (c) and a seasonal model in Section 7.2. The plots below were obtained using the code

```
ar2=gen.arma.wge(n=200,phi=c(1.35,-.65),sn=637,plot=FALSE)
plotts.sample.wge(ar2)
```

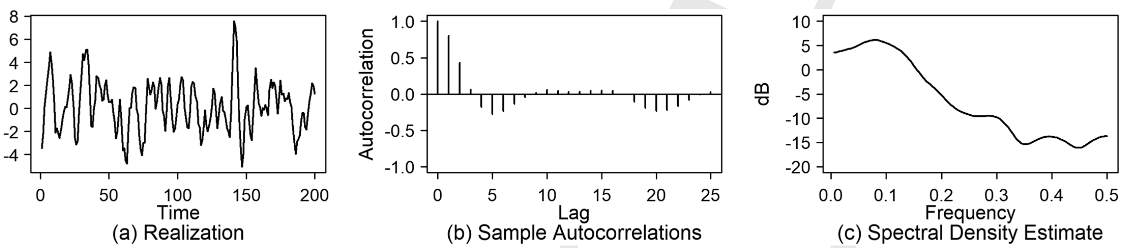


FIGURE 7.6 (a) Realization of length n = 200, sample autocorrelations, and Parzen spectral density estimate for the stationary AR(2) model $\left(1-1.3B+.65B^2\right)X_t = a_t$.

(1) Model (b) ARIMA(2,1,0): $\left(1-1.3B+.65B^2\right)\left(1-B\right)X_t = a_t$
We next consider the ARIMA(2,1,0) model

$$\left(1-1.3B+.65B^2\right)\left(1-B\right)X_t = a_t,$$

which we have referred to as Model (b). Table 7.2 shows the factor table for the nonstationary and stationary autoregressive components of this model.

TABLE 7.2 Factor Table for Model (b): $\left(1-1.3B+.65B^2\right)\left(1-B\right)X_t = a_t$

AR-FACTOR	*ROOTS*	$\lvert r\rvert^{-1}$	f_0
$1-B$	1	1	0
$1-1.3B+.65B^2$	$1.0 \pm .73i$	.81	.10

The factor table for the stationary AR(2) component of the model is shown in Table 7.1. Table 7.2 shows the additional factor of $1-B$ which is on the unit circle and has a system frequency of $f = 0$.

Figure 7.7(a) shows a realization of length $n = 200$ from Model (b), and it is clear that the nonstationary part of the model, $(1-B)$, dominates the behavior of the ARIMA(2,1,0) realization, and obscures the properties of the stationary AR(2) components of the model, sometimes to the point that they are not visible at all. Figures 7.7(b) and 7.7(c) show the sample autocorrelations and Parzen spectral density estimate, respectively, of the realization in Figure 7.7(a). The basic behaviors of the realization, sample autocorrelations, and spectral density are very similar to those in Figure 7.2, which were associated with a realization from an ARIMA(0,1,0). That is, the sample autocorrelations are slowly damping, and the spectral density has a strong peak at zero, which are the behaviors associated with the $\left(1-B\right)$ component in the model. The sample autocorrelations and Parzen spectral density estimate fail to indicate the presence of the AR(2) component.

We next difference the data in Figure 7.7(a). Recall that in the ARIMA(0,1,0) case, the differenced data were white noise. We issue the following commands to generate the data in Figure 7.7(a) and difference the data using **`artrans.wge`**.

```
x210=gen.arima.wge(n=200,phi= c(1.3,-.65),d=1,sn=4855)
d1=artrans.wge(x210,phi.tr=1)
# plot the data (first row) and the differenced data (second row)
plotts.sample.wge(x210)
plotts.sample.wge(d1)
```

QR 7.4
Differencing

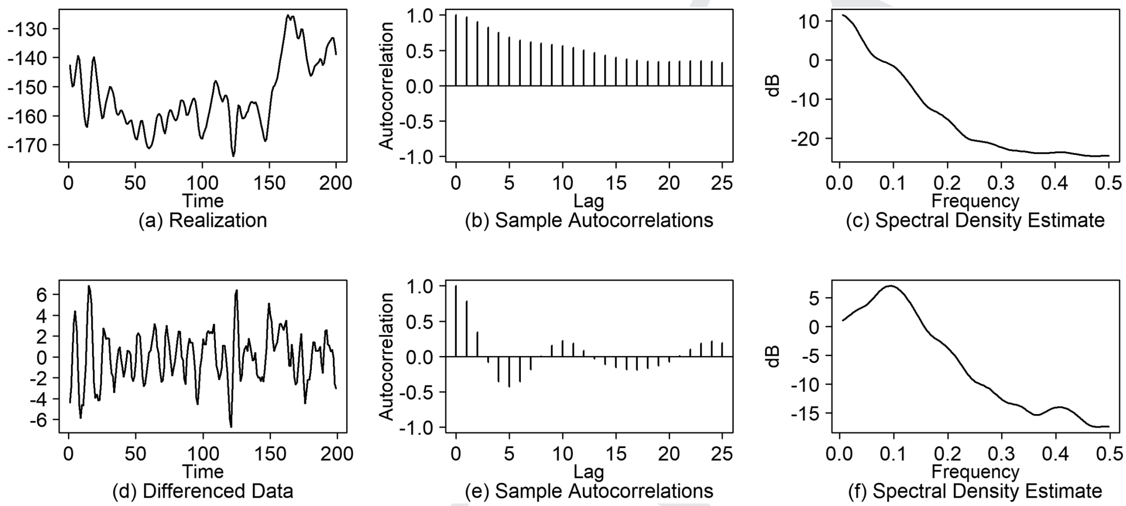


FIGURE 7.7 (a) Realization, (b) sample autocorrelations, and (c) Parzen spectral density estimate for a realization of length n = 200 from ARMA(2,1,0) model (7.3), (d) realization in (a) after differencing, (e) sample autocorrelation, and (f) Parzen spectrum of differenced data in (d).

The differenced dataset, **`d1`**, is a realization from the differenced process $Y_t = (1-B)X_t$. Because $(1-1.3B+.65B^2)(1-B)X_t = a_t$, it follows that Y_t satisfies the stationary model, $(1-1.3B+.65B^2)Y_t = a_t$. Figure 7.7(d) shows a plot of the differenced dataset, **`d1`**, along with the sample autocorrelations in Figure 7.7(e), and spectral density estimate in Figure 7.7(f). It is clear that the data are not white noise, the realization seems to be cyclic, the sample autocorrelations have a damped sinusoidal behavior, and the spectral density has peak at about $f = .10$. This behavior is consistent with the factor table in Table 7.1 for the stationary AR(2) component of the model, and the plots in Figures 7.7(d)–(f) are quite similar to those in Figure 7.6 for a realization from the corresponding stationary AR(2) model. That is, differencing the data has "revealed" the AR(2) behavior in the data that was not visible before the difference was taken.

Key Points:

1. Differencing the data automatically adds a $1-B$ factor in the final model.
2. Sample autocorrelations that have a slowly damping exponential appearance are an indication that a factor, $(1-B)$, should be considered in the model. Further discussion of ARIMA model identification will be given in Section 7.1.2.
3. The stationary and invertible AR and MA components of an ARIMA$(p,1,q)$ model may not be apparent until the data are differenced, which removes the dominating nonstationary factor, $(1-B)$.
4. In the end, the final model will include all the factors (both stationary and nonstationary) that were revealed/discovered.

(2) Model (c): ARIMA(2,2,0) $(1-1.3B+.65B^2)(1-B)^2 X_t = a_t$
Consider the ARIMA(2,2,0) model

$$(1-1.3B+.65B^2)(1-B)^2 X_t = a_t,$$

which we refer to as Model (c). This model differs from the ARIMA(2,1,0) Model (a) in that there are now two factors of $(1-B)$.That is, the model has two unit roots. Because the single unit root in the ARIMA(2,1,0) model in Model (b) dominated the behavior of the realization and its characteristics, it is reasonable to expect that a similar model, but with two unit roots, will *really* be dominated by the unit roots. If this is what you expected, you would be correct! Figure 7.8(a–c) shows a realization from this ARIMA(2,2,0) model along with sample autocorrelations and Parzen spectral density estimate.

The realization in Figure 7.8(a) has the appearance of a smooth curve with sections of near linear behavior and a large variance as can be seen by values of the realization. The $(1-B)^2$ behavior *totally* dominates the realization and there is absolutely no indication of the stationary AR(2) components of the model. The sample autocorrelations in Figure 7.8(b) damp very slowly, which is consistent with the behavior of the data.[5] As expected, the Parzen spectral density estimate in Figure 7.8(c) has a peak at $f = 0$. The data in Figure 7.8(a) were generated using the ***tswge*** command

```
x220=gen.arima.wge(n=200,phi=c(1.3,-.65),d=2,sn=450)
```

Based on our strategy for "stationarizing" the ARIMA(2,1,0) model, it seems reasonable that the data will need to be differenced two times to "uncover" the stationary components of the model. Because $(1-B)^2 = (1-2B+B^2)$ there are two ways to difference the data "two times". We could use the commands

```
d1=artrans.wge(x220,phi.tr=1)
d2=artrans.wge(d1,phi.tr=1)
```

or

```
d2=artrans.wge(x220,phi.tr=c(2,-1))
```

It is instructive to use the first approach and apply the 1st-order differences sequentially. Figures 7.8(d–f) show the differenced data **(d1)**, the sample autocorrelations of the differenced data, and the Parzen spectral density estimate, respectively. The differenced data in Figure 7.8(d) do not have the extremely smooth appearance and high variance behavior of the data in Figure 7.8(a) (note the difference in the vertical axis values). However, the data do not appear to be attracted to a mean and the sample autocorrelations have much the same behavior as those in Figure 7.8(b). The twice differenced data in Figure 7.6(g), that is, $Y_t = (1-B)(1-B)X_t = (1-B)^2 X_t$, follow the model for $Y_t = (1-B)^2 X_t$ which is $(1-1.3B+.65B^2)Y_t = a_t$. Twice differencing the data in Figure 7.8(g) results in $Y_t = (1-B)(1-B)X_t = (1-B)^2 X_t$, which removes the influence of the dominant nonstationary components and reveals any stationary behavior that may remain (which in this case we know satisfies $(1-1.3B+B^2)Y_t = a_t$). Remarkably, but not surprisingly, the realization, sample autocorrelations, and spectral density in Figures 7.8(g)–(i) are similar to those shown in Figure 7.6. Note the similarities which include the once again visible peak in the spectral density at $f = .10$ in Figure 7.8(i).

5 The true/model based autocorrelations and spectral density from any ARMA(p,d,q) model with $d \geq 1$ are the same as those shown in Figure 7.1 for the ARIMA(0,1,0) model (see Woodward, et al., 2017).

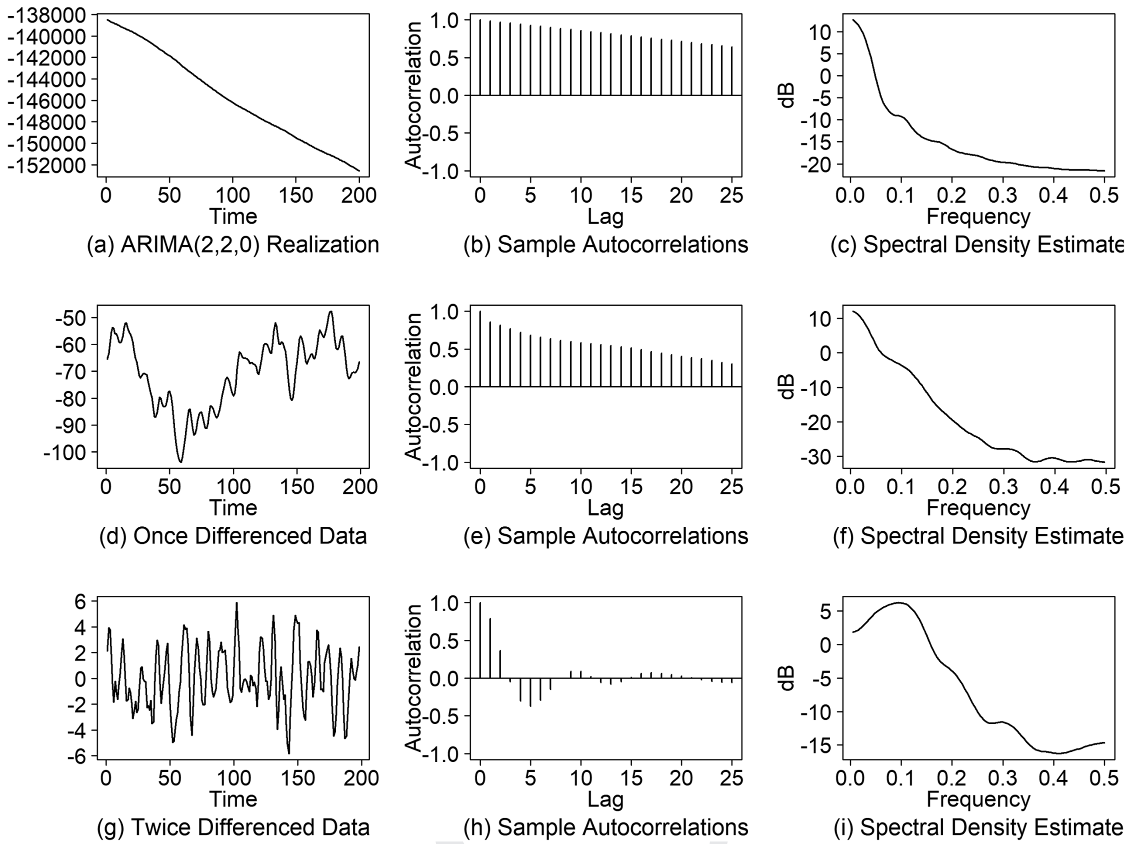


FIGURE 7.8 (a) Realization, (b) sample autocorrelations, and (c) Parzen spectral density estimate for ARMA(2,2,0) model in (7.4); (d) data in (a) differenced, (e) sample autocorrelation, and (f) Parzen spectrum of differenced data in (d); (g) data in (d) differenced, (h) sample autocorrelations, and (i) Parzen spectrum of data in (g).

The plots in Figure 7.8 are created using the commands

```
x220=gen.arima.wge(n=200,phi=c(1.3,-.65),d=2,sn=450)
d1=artrans.wge(x220,phi.tr=1,plot=FALSE)
d2=artrans.wge(d1,phi.tr=1,plot=FALSE)
plotts.sample.wge(x220)
plotts.sample.wge(d1)
plotts.sample.wge(d2)
```

Key Points:

1. Data with strong positive correlations resulting in “smooth” realizations such as seen in Figure 7.8(a) may require two differences to recover the underlying subtle stationary components of the data.
2. In the case of $d = 2$, notice that there is absolutely no indication of the stationary components in the original data.

7.1.2 Model Identification and Parameter Estimation of ARIMA(*p,d,q*) Models

In this section we discuss the issues involved with fitting the ARIMA(p,d,q) model in (7.1) to a time series realization. As noted in the examination of ARIMA models (a)–(c) in Section 7.1, the behavior of the data from an ARIMA(p,d,q) model with $d \geq 1$ is dominated by the $(1-B)^d$ factor. Realizations from the ARIMA models tend to wander aimlessly and are not attracted to an overall mean level while sample autocorrelations have the appearance of a slowly damping exponential. The stationary components in realizations, autocorrelations, and spectral densities are typically only visible after the data have been differenced d times.

7.1.2.1 Deciding Whether to Include One or More 1–B Factors (That Is, Unit Roots) in the Model

Consider the realization, sample autocorrelations, and Parzen spectral density estimate in Figure 7.2. These data are aimlessly wandering and the sample autocorrelations have a slowly damping exponential behavior. The question at hand is the following:

Should we fit

(a) an ARIMA model with $d \geq 1$ (that is, to include a unit root)
or
(b) a stationary ARMA model that has a root of the AR characteristic equation close to one?

In the following we discuss possible approaches to answering this question.

(a) A naive approach: An obvious approach is to fit a model to the dataset (say using ML estimates) and check the factor table for one or more factors of $1-B$.

First we simplify things by assuming a 1st-order model. For example, consider realizations from the ARIMA(0,1,0) model $(1-B)X_t = a_t$. It would be ideal if we could simply use ML estimates to fit a 1st-order AR model, that is, $(1-\phi_1 B)X_t = a_t$, to the data and check to see whether

(a) $\hat{\phi}_1 = 1$ (in which case we identify the underlying model as an ARIMA(0,1,0))
or
(b) $|\hat{\phi}_1| < 1$ (that is, we would identify the model as a stationary AR(1)).

Oh, if it were only that simple!

To check the viability of this procedure, we generated 100 realizations of length $n = 100$ from the ARIMA(0,1,0) model. For each realization we found the ML estimate, $\hat{\phi}_1$, associated with a 1st-order model. The average of the ML estimates was obtained along with the maximum and minimum from the 100 replications. The results are shown in column one of Table 7.3 where it can be seen that while the estimate did equal the "hoped for" $\hat{\phi}_1 = 1$, this actually occurred (rounded to 3 decimal places) in *only four realizations out of 100*. In fact, the average $\hat{\phi}_1$ is .958, and in one realization, the ML estimate was as small as $\hat{\phi}_1 = .797$. Consequently, the above strategy ***absolutely does not work***!

The simulation procedure was repeated using realizations from the stationary (but nearly nonstationary) models $(1-.99B)X_t = a_t$ and $(1-.975B)X_t = a_t$. The results for these models, shown in the second and third columns, respectively, of Table 7.3 are very similar to those for the actual ARIMA(0,1,0) model. Also, some of the realizations from the stationary models yielded $\hat{\phi}_1 = 1$ to three decimal places. Specifically, the results in Table 7.3 illustrate the difficulty of the following decision.

TABLE 7.3 Simulation Results Comparing 100 Realizations of Length $n = 100$ from the Models $(1-B)X_t = a_t$, $(1-.99B)X_t = a_t$, and $(1-.975)X_t = a_t$

	$(1-B)X_t = a_t$	$(1-.99B)X_t = a_t$	$(1-.975B)X_t = a_t$
Average $\hat{\phi}_1$	.958	.950	.928
Maximum $\hat{\phi}_1$	1	1	1
Minimum $\hat{\phi}_1$	.797	.850	.786

In fact, any real dataset is almost certainly neither perfectly fit by an ARMA nor an ARIMA model. Recall the opening of Chapter 5 in which we mentioned that real data are almost never perfectly fit by an ARMA model. This is true of ARIMA models and essentially any mathematical model fit to a set of data. Such models are used because they are useful in the analysis of data. To this end, we cannot overstate George Box's famous quote

George E. P. Box:
"*All models are wrong , but some are useful***."**

This is the spirit in which the model identification and parameter estimation in Chapter 6 and the remainder of this book should be viewed. That is, in Chapter 6, given a set of data which seemed to satisfy the basic properties of stationarity, AIC-type measures were used to find an ARMA model that was helpful in describing the behavior of the data and provide a method for forecasting future values. In the current section we introduce another level of complication in the attempt to find an imperfect but useful model. Specifically, we need to decide whether a model should include a unit root, that is, a factor, $\mathbf{1-B}$, and if so, how many, *before* we use AIC-type measures to select model orders and ultimately estimate the parameters of the stationary components of the model.

Key Points:

1. Fitting a model to data from an ARIMA model using ML estimates will nearly always produce a stationary model.
2. Consequently, unit roots cannot be identified by simply fitting a model to the data and checking the factor table for unit roots.
3. Points 1 and 2 and the simulations reported in Table 7.3 illustrate the fact that the decision whether to include a unit root in the model is a difficult one.
4. As will be discussed in the following, the decision whether to include a unit root in the model is one the investigator will need to make, based to some extent on physical considerations/domain knowledge.

(1) Unit Root Tests

Unit root tests are more sophisticated procedures (than the naïve approach in (a)) that give the investigator a "yes or no" answer concerning whether to include a unit root. The unit root tests are appealing because they "make the decision for you". We will not discuss the tests in detail but will show how to apply the tests to data and discuss their performance. The tests we will consider are:

(a) ADF (Augmented Dickey-Fuller) tests
(b) KPSS (Kwiatkowski-Phillips-Schmidt-Shin) tests

(a) The Augmented Dickey-Fuller Test
This test has been in use for many years to test the hypotheses:

H_0 : the model contains a unit root
H_a : the model does not contain a unit root (and is thus stationary)

Test statistic: τ
Rejection region: Reject H_0 if $\tau < d_\alpha$ where d_α is the α-level critical value.

Dickey (1976) finds the (complicated) limiting distribution of the test statistic. If $\tau \geq d_{.05}$ then H_0 is not rejected and the Dickey-Fuller test "detects" a unit root. Note that rejection of the null hypothesis leads to a conclusion that the process is stationary. Thus, the conclusion of a unit root is based on *not rejecting the null hypothesis*. This is in a sense "backward" because as statisticians/data scientists we know that failure to reject the null hypothesis does not imply belief that the null hypothesis is true, but simply that there was not sufficient evidence to reject it. To conduct unit root tests, we will use the function **ur.df** in the R package ***urca***. There are a variety of options, and we use the following command that includes a constant but not a trend in the model and uses AIC to select the number of lags. See Dickey and Fuller (1979) or Fuller (1996). The following command uses the Dickey-Fuller test for a unit root given a realization **x**.

```
h.df=ur.df(x,type='drift',selectlags='AIC')
```

The data in Figure 7.2(a) are a realization from an ARIMA(0,1,0) model (the model has a *unit root*). The data are generated using the command

```
x010=gen.arima.wge(n=100,d=1,sn=409)
```

After running the **ur.df** command using the **x010** data, the output, **h**, contains several items. For our purposes the test statistic, τ, is found in **h.df@teststat[1]** and $\tau = -1.076$. The 5% critical value, $d_{.05}$, is found in the output value **d05=h.df@cval[1,2],** where we see that $d_{.05} = -2.89$. The null hypothesis of a unit root is rejected at the $\alpha = .05$ level if $\tau < d_{.05}$. Because $-1.076 \geq -2.89$ we *do not reject* the null hypothesis at the $\alpha = .05$ level, and correctly *conclude that the model contains a unit root*.

(b) The KPSS Test
The KPSS test (see Kwiatkowski, Phillips, Schmidt, and Shin (1992)) has the desirable feature that it is designed to test the hypotheses

H_0 : the model is stationary
H_a : the model does not contain a unit root (and is thus stationary)

Test statistic: *kpss*
Reject region: Reject H_0 if $kpss > k_\alpha$ where k_α is the α-level critical value.

This is a more desirable setting for "concluding" a model has a unit root because the probability of rejecting the null hypothesis (concluding a unit root) when the null hypothesis is true (the model is stationary) can be made small (usually we require it to be less than .05). The KPSS test can be implemented using the function **ur.kpss** which is also in the ***urca*** package. The command below states that a trend term is not included in the model (**type='mu'**). Also, **lags='short'** specifies the formula to use for calculating the number of lags. (See ***urca*** documentation.) To apply the KPSS test to the data in **x010,** we use the command

```
h.kpss=ur.kpss(x010,type='mu',lags='short')
```

The output, **`h.kpss,`** contains several elements. The KPSS test statistic, *kpss*, is found in **`h.kpss@teststat,`** and in this case $kpss = 1.795$. The critical value, $k_{.05}$, is found in **`ko5=h.kpss@cval[2].`** In this case, $k_{.05} = .463$, and because $1.795 > .463$, the null hypothesis of stationarity is correctly rejected at the .05 level. The conclusion is that the model contains a unit root.

(2) Further Analysis of the Performance of Unit Root tests
We have seen that the ADF and KPSS tests gave the "correct" answers on the ARIMA(0,1,0) data in Figure 7.2(a). However, before recommending the tests for deciding whether a model has unit roots, we further investigate their performance using simulations.

(i) Ability of the tests to correctly detect unit roots when they are in the model (that is, when the model is nonstationary)

A useful test for unit roots should detect unit roots when the model in fact contains unit roots. To examine the ADF and KPSS tests in this regard, we generated 100 realizations from the ARIMA(0,1,0) model. We then calculate the percentage of times that the test found a unit root. Realization lengths $n = 100, 200, 500,$ and 4000 were used. These results are given in Table 7.4, and note that for a good test, the percentages of "detection" in the table should be "high". The results in the table show that both tests detect a unit root a high percentage of the time, and that the KPSS test seems to perform better with increasing realization length while the ADF test does not.

TABLE 7.4 Simulation Results Showing Percentage of 100 Realizations from an ARIMA(0,1,0) Model for which the Test *Correctly* Detected a Unit Root

	$n = 100$	$n = 200$	$n = 500$	$n = 10000$
ADF Test Average $\hat{\phi}_1$	91%	95%	96%	95%
KPSS Test	90%	97%	100%	100%

QR 7.5 ADF and KPSS Performance

(ii) Ability of the tests to correctly not detect unit roots when models are stationary

Table 7.4 shows that if a model has a unit root, the ADF and KPSS tests are "good" about detecting it. If a test is designed to tell us whether a model contains a unit root, then *when there is not a unit root in the model, it should not detect one*. We state the obvious: All *stationary* AR(1) models do not have a unit root. But of course, there are many "stationary" AR(1) models. Table 7.5 shows the extent to which these tests can distinguish between ARIMA(0,1,0) models with unit roots and stationary AR(1) models with roots "close to" the unit circle. The results in Table 7.4 for $n = 200$ were consistent with those for larger n. Table 7.5 is based on realizations of length $n = 200$ from the stationary AR(1) models with $\phi_1 = .9, .95, .99,$ and .999. The percentage of times that the test (incorrectly) detected a unit root are shown in Table 7.5. Because these models do not have a unit root, we would hope that the percentages are small.

TABLE 7.5 Simulation Results Showing Percentage of 100 Realizations of length $n = 200$ from a Stationary AR(1) Model for which the Test *Incorrectly* Detected a Unit Root

	$\phi_1 = .999$	$\phi_1 = .99$	$\phi_1 = .95$	$\phi_1 = .9$
ADF Test Average $\hat{\phi}_1$	95%	95%	72%	17%
KPSS Test	89%	90%	76%	43%

QR 7.6 ADF and KPSS Type II Error

The results of Table 7.5 show that for stationary AR(1) models with $\phi_1 \geq .95$, the ADF and KPSS tests are very likely to *incorrectly* detect a unit root. Even for $\phi_1 = .9$ the tests detect a unit root substantially more than 5% of the time, yet for $\phi_1 = .8$ the ADF test *never* detected a unit root, but the KPSS detected one 28% of the time. That is, if these tests detect a unit root, the investigator cannot feel confident that the model actually contains a unit root!

Key Point:

1. Whenever a unit root test "detects" a unit root, the correct interpretation is that a model with unit roots is "plausible".
2. We do not recommend using the ADF, KPSS, or other similar tests as the definitive basis of a yes-or-no decision to include a unit root.
3. We emphasize again that the final decision should include physical considerations/domain knowledge when it exists. See Example 7.1(a).

The Key Points above are disappointing, because it would have been ideal if the unit root tests would provide investigators with a decisive yes-or-no conclusion concerning whether to include a unit root in your model. ***However, they do not!***

7.1.2.2 General Procedure for Fitting an ARIMA(p,d,q) Model to a Set of Time Series Data

Once the decision to include a unit root is made, the general procedure for fitting an ARIMA(p,d,q) model is as follows:

1. *Difference the data:* Given that the decision has been made to include a unit root, difference the data. Suppose the data are in **x**. Then difference the data using the command

   ```
   d1=artrans.wge(x,phi.tr=1) # data are plotted in Figure 7.6(a)
   ```

2. *Examine the differenced data:*
   If the model for the differenced data should include a unit root
   - difference the data again

   If the differenced data appear to be stationary,
   - then set $d = 1$ and include the factor $1 - B$ in the model
   - go to step 4
3. *Continue to difference the data enough times to produce stationarity:*
   - the number of differences required to "stationarize" the data is the estimated value of d.
   - the factor $(1 - B)^d$ should be included in the final model.
4. *Fit an ARMA(p,q) model to the stationary data.*

Key Points:

1. Don't "overdo" the differencing.
2. These authors have never encountered real data for which $d = 3$ or more was appropriate.

(1) Obtaining a Final ARIMA(p,d,q*) Model*
In this section we use the above general procedure for obtaining final ARIMA models for the datasets from Models (a), (b), and (c) in Section 7.1.1. In all three cases the issue of deciding whether borderline situations lead to stationary or nonstationary models will not be addressed. Nonstationary ARIMA models will be used in cases where they are suggested as *plausible* using such tools as the unit root tests and the damping behavior of the sample autocorrelations. Because these are simulated datasets, there are no physical or domain knowledge factors to consider. In Example 7.1(a) we consider these factors in our modeling of Dow Jones data.

(2) Model (a): Final Model Fit to ARIMA(0,1,0) Data
Recall that the data in Figure 7.2(a) were generated using the command

```
x010=gen.arima.wge(n=100,d=1,sn=409)
```

The following analysis proceeds as if the true model is unknown. As previously noted, the data are wandering and the sample autocorrelations have a slowly damping exponential behavior. The dataset **x010** is the dataset we previously used to illustrate the ADF and KPSS tests. In both cases, the tests found a unit root. Assuming that the decision has been made to include a unit root in the model, the procedure is as follows:

1. *Difference the data:*
2. **d1=artrans.wge(x010,phi.tr=1)**
3. *Examine the differenced data:* The differenced data, sample autocorrelations, and Parzen spectral density estimate were shown in Figure 7.5 (a)−(c). It was previously noted that the differenced data appear to be stationary (and in fact looked like white noise). Therefore, we select $\boldsymbol{d}=1$, and thus the model contains a factor $1-\boldsymbol{B}$. Go to Step 4.
4. *Fit an ARMA(p,q) model to the differenced data:* As noted in Step 2, the differenced data appear uncorrelated, the sample autocorrelations are very close to zero and tend to stay within the 95% limit lines,and the Parzen spectral density estimate is fairly flat. These are characteristics of white noise data and indicate that no further analysis is needed (that is, we fit an ARMA(0,0) model to the data). The white noise variance is estimated by the variance of the differenced data using the base R command **var(d1)** from which we find that the variance of **d1** is 1.035. Consequently, because $d=1$, we include a first difference, $1-B$, term to the model, and the final model is

$$(1-B)X_t = a_t, \text{ with } \hat{\sigma}_a^2 = 1.035.$$

(3) Model (b): Final Model Fit to ARIMA(2,1,0) Data
The data in Figure 7.7(a) were generated using the command

```
x210=gen.arima.wge(n=200,phi=c(1.3,-.65),d=1,sn=25)
```

The data are again analyzed as if the true model is unknown. As noted previously, the data and sample autocorrelations suggest the possibility of a unit root. Applying the ADF test yields $\tau=-2.56$ and $d_{.05}=-2.88$, so because $-2.56\geq-2.88$,the ADF test does not reject the null hypothesis of a unit root. For the KPSS test, $kpss=3.69$and $k_{.05}=.463$. So, because $3.69>.463$ the KPSS test detects a unit root. Assuming that we have decided to include a unit root in the model based on the available knowledge, we proceed as follows.

1. *Difference the data:*

   ```
   d1=artrans.wge(x210,phi.tr=1)
   ```

2. *Examine the differenced data:* The differenced data, shown in Figure 7.7(d), appear to be stationary (but not white noise). (Also, the ADF and KPSS tests fail to detect a unit root.) Therefore, we select $d = 1$.Thus, the model contains a factor $1 - B$ Go to Step 4.
4. *Fit an ARMA(p,q) model to the differenced data:* The following command uses AIC to select ARMA model orders for the differenced data (**d1**) letting p range from 0 to 10 and q from 0 to 2.

```
aic.wge(d1,p=0:10,q=0:2)
```

AIC selects $p = 2$ and $q = 0$, so we estimate the parameters of the AR(2) model fit to the stationary differenced data using the command

```
est.ar.wge(d1,p=2)
```

The ML estimates for the differenced data are $\hat{\phi}_1 = 1.25$, $\hat{\phi}_2 = -.64$, and $\hat{\sigma}_a^2 = 1.03$. The final model is

$$(1 - 1.25B + .64B^2B^2)(1 - B)X_t = a_t, \tag{7.3}$$

with $\hat{\sigma}_a^2 = 1.03$. This model is "similar" to the true model, and the factor table shows that the fitted model has system frequency $f = .10$which is consistent with the features of the true AR(2) factor (see Table 7.1). Notice that because of the factor $1 - B$ in the model, we do not estimate the mean.

(4) Model (c): Final Model Fit to ARIMA(2,2,0) Data

The data in Figure 7.8(a) were generated using the following command, and again the data are analyzed as if the true model is unknown.

```
x220=gen.arima.wge(n=200,phi=c(1.3,-.65),d=2,sn=450)
```

The data in Figure 7.8(a) have a dramatic wandering behavior along with sample autocorrelations that damp slowly. This strongly suggests a unit root. Also, $\tau = -1.167$ and $d_{.05} = -2.88$, and since $-1.167 \geq -2.88$, the ADF test does not reject a unit root. Also, $kpss = 4.097$ and $k_{.05} = .463$, and because $4.097 > .463$,the KPSS test detects a unit root. Evidence suggests a unit root, so we proceed as follows.

1. *Difference the data:*

```
d1=artrans.wge(x220,phi.tr=1)
```

2. *Examine the differenced data:* The differenced data, shown in Figure 7.8(d), have a wandering appearance and associated sample autocorrelations that are slowly damping. For the differenced data, $\boldsymbol{\tau} = -\mathbf{3.878}$ is less than $\boldsymbol{d}_{.05} = -\mathbf{2.88}$**,** so the ADF test rejects the null hypothesis of a unit root. However, $\boldsymbol{kpss} = \mathbf{1.060}$ is greater than $\boldsymbol{k}_{.05} = \mathbf{.463}$ so the KPSS test rejects the null hypothesis of stationarity and concludes that the differenced data has a unit root. Despite the conflicting results, we decide to difference the data again.
3. *Continue to difference the data enough times to produce stationarity:* We difference the data again using the command

```
d2=artrans.wge(d1,phi.tr=1)
```

The twice differenced data are plotted in Figure 7.8(g) along with sample autocorrelations and spectral density in Figures 7.8 (h) and (i), respectively. These data appear to be stationary but not white noise. (The ADF and KPSS tests do not detect another unit root.) We conclude that $\boldsymbol{d} = 2$, and thus that the model contains a factor $(1 - B)^2$.

4. *Fit an ARMA(p,q) model to the differenced data:* The following command uses AIC to select ARMA model orders for the differenced data (in **d1**) letting p range from 0 to 10 and q from 0 to 2.

```
aic.wge(d2,p=0:10,q=0:2)
```

AIC selects $p = 2$ and $q = 0$, so we use the following command to estimate the parameters of the AR(2) model fit to the stationary differenced data.

```
est.ar.wge(d1,p=2)
```

The ML estimates for the differenced data are $\hat{\phi}_1 = 1.33$, $\hat{\phi}_2 = -.69$, and $\hat{\sigma}_a^2 = 1.06$. The final model is

$$(1-1.33B++.69B^2)(1-B)^2 X_t = a_t, \tag{7.4}$$

with $\hat{\sigma}_a^2 = 1.06$. This model is very similar to the actual ARMA(2,2,0) model, and again, because the factor $(1-B)^2$ is in the model, we do not estimate the mean.

Key Points:

1. If the original data appear to be stationary, then we do not difference the data and fit an ARMA(p,q) model to the data.
2. Each time we difference the data, we automatically add a $1-B$ factor to the model.
3. For the data from Models (a)−(c), we "know" that the true models are ARIMA. Had these been real datasets for which we did not know a true model, then the most we could say is that ARIMA models with $d = 1,1,$ and 2 are *plausible* models for datasets **`x010`**, **`x210`**, and **`x220`**, respectively.
4. Running the code in these examples and obtaining the results builds confidence and momentum.
   - ***Run the code!***

QR 7.7 Building an ARIMA Model

Example 7.1(a) DOW Jones Data

In Example 5.1 we considered dataset **`dow1985`** in ***tswge*** which is a record of the daily DOW Jones closing averages by month from March 1985 through December 2020. (Source: Yahoo Finance). The **`dow1985`** data, sample autocorrelations, and Parzen spectral density estimate are shown in Figure 7.9(a), (b), and (c), respectively. The data have an upward wandering behavior and the sample autocorrelations are slowly damping exponential in nature. The ADF and KPSS tests both suggest a unit root. The plots in Figure 7.9 can be obtained using the ***tswge*** commands

```
data(dow1985)
plotts.sample.wge(dow1985)
ddow=artrans.wge(dow1985,phi.tr=1,plot=FALSE)
plotts.sample.wge(ddow,arlimits=TRUE)
```

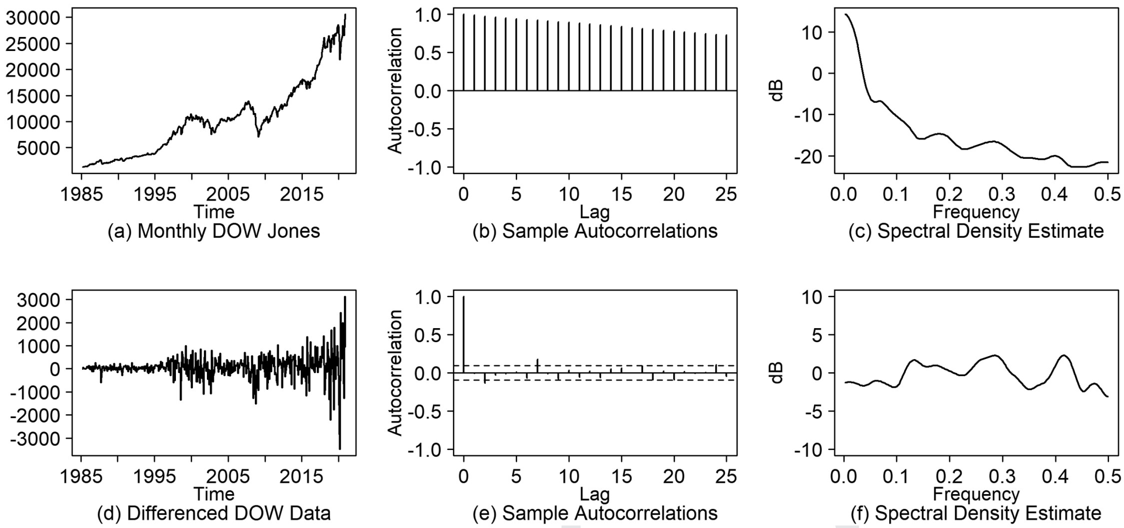


FIGURE 7.9 (a) DOW Jones closing average by month from March 1985 through December 2020 with (b) sample autocorrelations and (c) Parzen spectral density estimate for the data in (a), (d) DOW data in (a) differenced along with (e) sample autocorrelations and (f) Parzen spectral density estimate of the differenced data in (a).

(5) Should We Use a Stationary or Nonstationary Model for DOW Data?

The Dow Jones data provide a good example concerning the decision whether to use a nonstationary or stationary model. The mean of the DOW data from 1985 through 2020 is \$10,352. If we decided to fit a stationary model by simply estimating the parameters of an AR(1) fit to the data, the model would be $(1-.999B)(X_t-10352)=a_t$. Notice that the mean term was included in the model because it is stationary. Use of the stationary model implies that the DOW will have some attraction for a mean value which is estimated by $\bar{X}$=\$10,352.

Example of Domain Knowledge:

Although there are dips, for example, around 2009, historically the stock market increases in the long term. Figure 7.10 is a plot of ***tswge*** dataset **`dow.annual`** which shows the historical annual closing averages from 1915–2020. There certainly seems to be an upward trend with no tendency to be attracted back to an overall mean. Consequently, for the 1985–2020 data modeled in this example, we select the nonstationary model, believing that there is no expectation for the future Dow Jones averages to be attracted back to earlier mean levels.[6]

6 The authors make this claim even though as we write this footnote on Friday, February 28, 2020, the DOW Jones average has dropped by about 3,500 points for the week over concerns about the coronavirus. We will keep this footnote unchanged, and by the time you read the footnote you will know whether the DOW continued to wander (mostly upward) or whether it turned downward and tended to be attracted to the mean (which is estimated in this dataset by $\bar{X}=\$10,352$). As investors, we hope the former is the case!

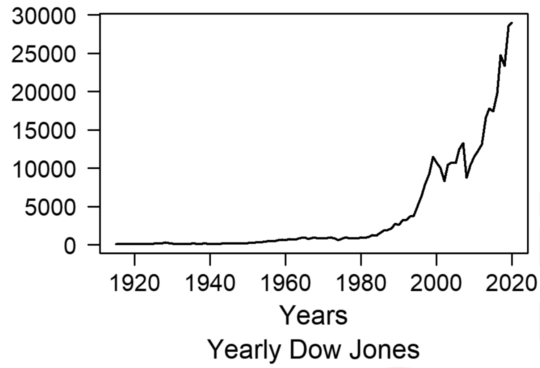


FIGURE 7.10 DOW Jones year-end averages for 1915–2020.

Back to the analysis of the **dow1985** data, we follow the steps used before:

1. *Difference the data:*

   d1=artrans.wge(dow1985,phi.tr=1)

2. *Examine the differenced data:* The differenced data (**d1**) appear to be random but with an increasing variance in time. The sample autocorrelations in Figure 7.9(e) are all small (95% limit lines are added as a reference), and the spectral density in Figure 7.9(f) is fairly "flat". These suggest white noise except for the increased variability in time. At this point we will model the differenced data as white noise and further analysis is not needed. We conclude that $d = 1$ so that the model contains a factor of $1 - B$.
3. *Fit an ARMA(p,q) model to the differenced data:* The white noise variance is estimated by the variance of the differenced data. Using the base R command **var(d1)** we find that the variance of the differenced data is 294,793. Consequently, because the first difference added a $1 - B$ term to the model, the final model is

$$(1-B)X_t = a_t, \text{with } \hat{\sigma}_a^2 = 294,793$$

Note: The increased variability is concerning. Consequently, it might be beneficial to take the log of the DOW data before analysis.

Example 7.1(b) US Population Data

Figure 7.11(a) shows the estimated US population by year from 1900 through 2020 (shown here in millions of people). (Source: usafacts.org). These data are in ***tswge*** file **uspop** and have the appearance of nonstationary data, similar to the ARIMA(2,2,0) data in Figure 7.8(a) except for the fact that the population is increasing in Figure 7.11(a) while the data in Figure 7.8(a) are decreasing.

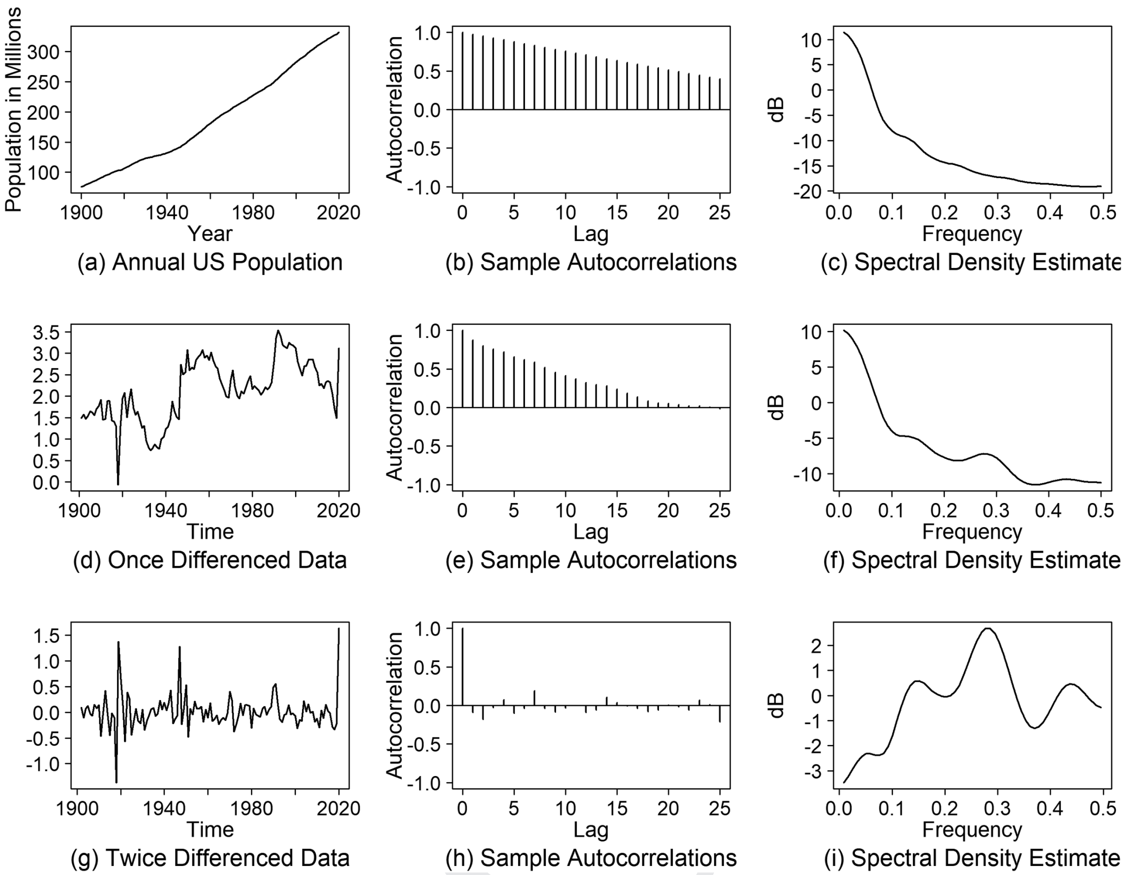


FIGURE 7.11 (a) US annual population data in millions, (b) sample autocorrelations and (c) Parzen spectral density estimate of data in (a); (d) data in (a) differenced, (e) sample autocorrelation, and (f) Parzen spectrum of differenced data in (d); (g) data in (d) differenced, (h) sample autocorrelations, and (i) Parzen spectrum of data in (g).

The slowly damping sample autocorrelations suggest the need to difference the data. Also, $\tau = 1.956$ and $d_{.05} = -2.88$, and since $1.956 \ge -2.88$, the ADF test does not reject a unit root. Also, $k_{05} = 2.496$ and $k_{.05} = .463$, and because $2.496 > .463$, the KPSS test detects a unit root. Evidence suggests a unit root, so we proceed as follows.

1. *Difference the data:*

```
d1.pop=artrans.wge(uspop,phi.tr=1)
```

2. *Examine the differenced data:* The differenced data are shown in Figure 7.11(d) along with sample autocorrelations and spectral density in Figures 7.11 (e) and (f), respectively. The differenced data have a wandering appearance and associated sample autocorrelations that are slowly damping. For the differenced data, $\boldsymbol{\tau} = \mathbf{-2.324}$ and $\boldsymbol{d}_{.05} = \mathbf{-2.88}$, so because $\mathbf{-2.324} \ge \mathbf{-2.88}$ the ADF test fails to reject the null hypothesis of a unit root. Also, ***kpss*** = **1.2759** is greater than $\boldsymbol{k}_{.05} = \mathbf{.463}$, so the KPSS test rejects the null hypothesis of stationarity and concludes that the differenced data has a unit root. Consequently, we decide to difference the data again.
3. *Continue to difference the data enough times to produce stationarity:* We difference the data again using the command

```
d2.pop=artrans.wge(d1.pop,phi.tr=1)
```

The twice differenced data are plotted in Figure 7.11(g) along with sample autocorrelations and spectral density in Figures 7.11 (h) and (i), respectively. These data appear to be stationary and possibly white noise. We conclude that $\boldsymbol{d} = 2$, and thus that the model contains a factor $(1-B)^2$.

4. *Fit an ARMA(p,q) model to the differenced data:* Using the commands

```
aic.wge(d2,p=0:10,q=0:4)
aic.wge(d2,p=0:10,q=0:4,type='bic')
```

AIC selects $p = 0$ and $q = 2$ and BIC selects white noise ($p = 0, q = 0$). The Ljung-Box test on the **d2.pop** data has p-values of .705 and .806 for $k = 24$ and 48, respectively. Based on this evidence we decide to consider **d2.pop** to be white noise, and the final model is

$$(1-B)^2 X_t = a_t,$$

with $\hat{\sigma}_a^2 = .113$. This model is very similar to the actual ARMA(2,2,0) model, and again, because the factor $(1-B)^2$ is in the model, we do not estimate the mean.

Note: The `d2.pop` data, while passing the white noise tests, seems to have more variability early in the 20th century. The "rate of change" was greater with the smaller population. This is not unusual for "real" datasets for which we do not know the true model, but hopefully we have found a useful one. See Problem 7.4 for an alternative approach to analyzing the `uspop` data.

Before moving on to the topic of forecasting, we close this section with the following Key Points.

Key Points:

1. If the original data appear to be stationary, then we do not difference the data and fit an ARMA(p,q) model to the data.
2. Each time we difference the data, we automatically add a $1-B$ factor to the model.
3. For the data from Models (a)−(c), we "know" that the true models are ARIMA. Had these been real datasets for which we did not know a true model, then the most we could say is that ARIMA models with $d = 1,1,$ and 2 are *plausible* models for datasets `x010,x210,` and `x220,` respectively.

7.1.3 Forecasting with ARIMA Models

Because ARIMA models are basically limiting versions of ARMA models where roots of the characteristic equation are allowed to be on the unit circle, we quickly review forecasts using ARMA models.

7.1.3.1 ARMA Forecast Formula

The basic formula for forecasts from an ARMA(p,q) model is

$$\hat{X}_{t_0}(\ell) = \sum_{j=1}^{p} \hat{\phi}_j \hat{X}_{t_0}(\ell - j) - \sum_{j=\ell}^{q} \hat{\theta}_j \hat{a}_{t_0+\ell-j} + \bar{x}\left[1 - \sum_{j=1}^{p} \hat{\phi}_j\right].$$

(1) ARIMA Forecast Formula

For a fitted ARIMA(p,d,q)model, $\hat{\phi}(B)(1-B)^d X_t = \hat{\theta}(B)a_t$, the "AR" part is the $p+d$ order operator $\hat{\phi}^*(B)=\hat{\phi}(B)(1-B)^d$ or in expanded form $\hat{\phi}^*(B)=1-\sum_{j=1}^{p+d}\hat{\phi}_j^* B^j$,
where $\hat{\phi}_j^*$ denotes the jth coefficient. For example, consider the ARIMA(2,1,0) model fit to the data in Figure 7.7(a). The model, given in (7.3) is $(1-1.25B+.64B^2)(1-B)X_t=a_t$. In this case, $\hat{\phi}^*(B)=1-1.25B+.64B^2)(1-B)$, by multiplying operators it follows that $\hat{\phi}^*(B)=(1-2.25B+1.89B^2-.64B^3)$, and $\hat{\phi}_1^*=2.25$, $\hat{\phi}_2^*=-1.89$, and $\hat{\phi}_3^*=.64$.

For the ARIMA(p,d,q) case, the forecast formula analogous to the ARMA formula above is

$$\hat{X}_{t_0}(\ell)=\sum_{j=1}^{p+d}\hat{\phi}_j^*\hat{X}_{t_0}(\ell-j)-\sum_{j=\ell}^{q}\hat{\theta}_j\hat{a}_{t_0+\ell-j}+\bar{x}\left[1-\sum_{j=1}^{p+d}\hat{\phi}_j^*\right] \tag{7.5}$$

Key Points:

1. The characteristic polynomial associated the $\boldsymbol{p+d}$ order operator can be written as:
   (a) $\hat{\phi}^*(z)=\hat{\phi}(z)(1-z)^d$
   (b) $\hat{\phi}^*(z)=1-\sum_{j=1}^{p+d}\hat{\phi}_j^* z^j$
2. Form (a) shows that $\hat{\phi}^*(1)=\hat{\phi}(z)(1-1)^d=0$.
3. From (2) it follows that $\hat{\phi}^*(1)=1-\sum_{j=1}^{p+d}\hat{\phi}_j^*=0$.
4. From (3) it follows that the last term in (7.5) is always zero if $d>0$.

Based on Key Point (4) it follows that the last tern in (7.5) is always zero if $d>0$. So, the ARIMA(p,d,q) forecast formula in (7.5) can be simplified to

$$\hat{X}_{t_0}(\ell)=\sum_{j=1}^{p+d}\hat{\phi}_j^*\hat{X}_{t_0}(\ell-j)-\sum_{j=\ell}^{q}\hat{\theta}_j\hat{a}_{t_0+\ell-j}. \tag{7.6}$$

For the model in (7.3) the forecasts become

$$\boldsymbol{\hat{X}_{t_0}(\ell)=2.25\hat{X}_{t_0}(\ell-1)-1.89\hat{X}_{t_0}(\ell-2)+.64\hat{X}_{t_0}(\ell-3)}$$

Key Points:

1. From (7.6) it follows that the sample mean does not play a part, not even in the forecasts.
2. This is consistent with our expression of ARIMA models (which do not include a mean).

We begin our discussion of ARIMA forecasts with the ARIMA(0,1,0) model.

(2) Forecasts for an ARIMA(0,1,0) Model

Figure 7.2(a) showed a realization of length $n = 100$ from an ARIMA(0,1,0) model with $\sigma_a^2 = 1$. The data in Figure 7.2(a) seem to be linearly decreasing, and because of this, a time series regression fit to the data would probably be assessed to have a negative, statistically significant slope. Based on examination of the data, our heuristic forecasts might be for the future values to continue to decrease. For the ARIMA(0,1,0) model, $p = 0$, $d = 1$, and $\hat{\phi}^*(B) = \hat{\phi}(B)(1-B)^d = 1 - B$. That is, $\hat{\phi}_1^* = 1$, $\hat{\phi}_j^* = 0$ for $j > 1$, and from (7.8) it follows that $\hat{X}_{t_0}(\ell) = \hat{X}_{t_0}(\ell - 1)$. Using this expression, $\hat{X}_{t_0}(1) = \hat{X}_{t_0}(0) = X_{t_0}$, $\hat{X}_{t_0}(2) = \hat{X}_{t_0}(1) = X_{t_0}$, and so forth.

The data from Figure 7.2(a) (and shown again in Figure 7.12(a)) and the forecasts shown in Figure 7.12(b) are obtained using the commands

```
x010=gen.arima.wge(n=150,d=1,sn=409)
fore.010=fore.arima.wge(x010[1:100],d=1,n.ahead=50,limits=FALSE)
```

Figure 7.12(b) forecasts show no tendency to move downward or toward some mean level. Instead, the best forecast of future values is the last value observed. These are very "boring" and "disappointing" forecasts, and they seem to not be "as good as we could do" without using a model. However, because the model $(1-B)X_t = a_t$ can be written as $X_t = X_{t-1} + a_t$, where a_t is zero mean white noise, it follows that X_{t_0+1} is as likely to be greater than X_{t_0} as it is to be less than X_{t_0}. Consequently, there is no reason to predict that, for example, X_{t_0+1} should be above or below X_{t_0}. That is, $\hat{X}_{t_0}(1) = X_{t_0}$ is a reasonable predictor. Figure 7.12(b) also shows the 95% prediction limits obtained using the formula

$$\hat{X}_{t_0}(\ell) \pm 1.96\hat{\sigma}_a \left\{ \sum_{k=0}^{\ell-1} \hat{\psi}_k^2 \right\}^{\frac{1}{2}},$$

where the ψ-weights are based on the coefficients from the "full" AR part. For the ARIMA(0,1,0) model, $\hat{\phi}^*(B) = 1 - B$, so the ψ-weights can be found using the command

```
psi.weights.wge(phi=1,lag.max=50)
```

The resulting ψ-weights are $\hat{\psi}_j = 1$ for each $j > 1$. Consequently, the series $\sum_{k=0}^{\ell-1} \hat{\psi}_k^2 = \sum_{k=0}^{\ell-1} 1 = \ell$, does not converge as $\ell \to \infty$, so the limits increase without bound. The 95% prediction limits in Figure 7.12(b) indicate that there is substantial uncertainty regarding future values.

The first 100 data values in **x010** were used to obtain the model, and to further examine the forecasts in Figure 7.12(b) we obtain forecasts for X_{101} through X_{150}. The actual data values x_{101}through x_{150} are available in the full **x010** dataset. Figure 7.12(c) is a plot of Figure 7.12(b) along with the points **x010[101:150]**, and we see that in fact, the trend did not continue, and the forecasts were about as good as you could expect. The wide prediction limits indicate that forecasts very far into the future may not be very good. The trend seen in Figure 7.12(a) is a random trend, and cannot be "trusted" to continue into the future.

The plots in Figure 7.12 were obtained using the ***tswge*** commands

```
par(mfrow=c(1,3))
t=1:150
plot(t[1:100],x010[1:100],xlim=c(0,150), type='l')
fore.arima.wge(x010[1:100],d=1,n.ahead=50)
fore.arima.wge(x010[1:100],d=1,n.ahead=50)
points(t[101:150],x010[101:150], type='l')
```

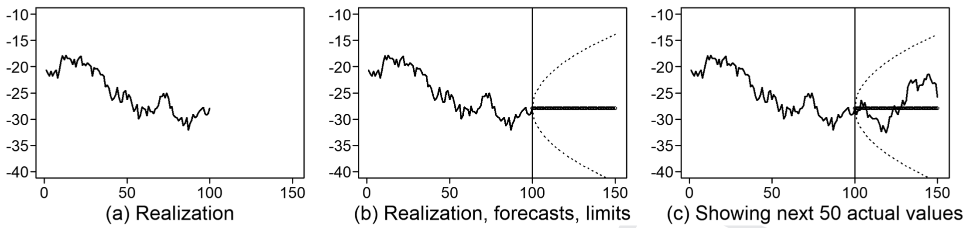


FIGURE 7.12 (a) Figure 7.2(a) showing realization of length $n = 100$ from an ARIMA(0,1,0) model, (b) realization in (a) along with first 50 step ahead forecasts and 95% limits, and (c) same as (b) showing the next 50 actual data values in the series `x010`.

(3) Forecasts from an AR(1) Model Fit to `x010`

Although the **`x010`** dataset is a realization from the nonstationary ARIMA(0,1,0) model, in practice we will not "know" this. Referring back to the discussion about whether to include a unit root, suppose the decision was made to fit a stationary AR(1) model to the **`x010`** data shown again in Figure 7.12(a). Using the command

```
est.ar.wge(x010,p=1)
```

the estimated stationary model is

$$(1-.965B)(X_t+24.93)=a_t, \tag{7.7}$$

with $\hat{\sigma}_a^2 = 1.02$. Because this is a stationary model, forecasts trend "upward" toward the sample mean, $\bar{x} = -24.93$.This tendency is visible in Figure 7.13(b). Figure 7.13(b) also shows the 95% prediction limits which are "tighter" than those in Figure 7.12(b). That is, if we believe the model is stationary, we are not as concerned that data will trend far away from the mean. Figure 7.13(c) adds the actual values as shown in Figure 7.12(c). Both sets of forecasts (those in 7.12 and 7.13) were "good". One measure of the forecast quality is that in both cases, all forecasts stayed within the 95% limits.

The plots in Figure 7.13 were obtained using the following code:

```
t=1:150
plot(t[1:100],x010[1:100],xlim=c(0,150), type='l')
fore.arma.wge(x010[1:100],phi=.965,n.ahead=50)
```

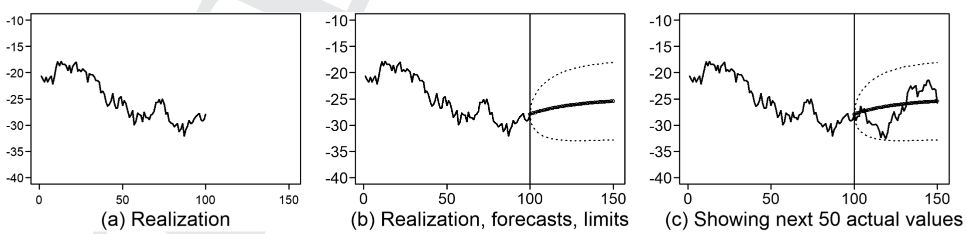


FIGURE 7.13 (a) Figure 7.12(a) showing realization of length $n = 100$ from an ARIMA(0,1,0) model, (b) realization in (a) along with first 50 step ahead forecasts and 95% limits based on fitting the stationary model in (7.9) to the data, and (c) same as (b) showing the next 50 actual data values in the series `x010`.

Comparing Forecasts from an AR(1) model fit to **`x010 to the ARIMA(0,1,0)`**

The AR(1) model and the ARIMA(0,1,0) model discussed previously can be compared using the rolling window RMSE method discussed in Chapter 6. Below, the rolling window RMSE is calculated for both models again with a horizon of 50:

```
roll.win.rmse.wge(x010,horizon = 50,d = 1) #ARIMA(0,1,0)
```

```
"Please Hold For a Moment, TSWGE is processing the Rolling Window RMSE with 97
windows."
"The Summary Statistics for the Rolling Window RMSE Are:"
Min. 1st Qu. Median  Mean 3rd Qu.  Max.
1.414  2.329  3.446  3.844  5.743  7.632
"The Rolling Window RMSE is: 3.844"
```

```
roll.win.rmse.wge(x010,horizon = 50, phi = .965) #AR(1)
```

```
"Please Hold For a Moment, TSWGE is processing the Rolling Window RMSE with 99
windows."
"The Summary Statistics for the Rolling Window RMSE Are:"
Min. 1st Qu. Median  Mean 3rd Qu.  Max.
2.156  2.577  2.873  2.961  3.287  4.328
"The Rolling Window RMSE is: 2.961"
```

Interestingly, although the ARIMA(0,1,0) is the "right" model, the rolling window RMSE is lower for the AR(1) is 2.961 as compared to 3.844 for the ARIMA(0,1,0) model.

Key Points:

1. Trends seen in ARIMA realizations with $d = 1$ are random trends and a given realization may show time intervals with upward trending behavior along with other intervals of downward trending behavior.
   – Realizations can "turn on a dime."
2. If an ARIMA model with $d = 1$ is an appropriate model for a set of data, any observable trending behavior should not be "boldly" predicted to continue.
3. The forecasts depend ***heavily*** on the type of model selected.
   – Fitting an ARIMA model with $d = 1$ produces flat forecasts suggesting the trending might or might not continue.
   – Fitting a stationary ARMA model produces forecasts that move toward the sample mean.
   – Fitting a line-plus-noise model (Chapter 8) produces forecasts that that predict the existing trend to continue.

(4) Forecasts for an ARIMA(2,1,0) Model

Figure 7.7(a) showed a realization of length $n = 200$ from the ARIMA(2,1,0) model with $\sigma_a^2 = 1$, referred to as Model (b) in Section 7.1. The data in Figure 7.7(a) from the ARMA(2,1,0) Model (b) (shown again in Figure 7.14(a)) and the forecasts in Figure 7.14(b) (based on the fitted model in (7.3)) were found using the commands

```
x210=gen.arima.wge(n=250,phi= c(1.3,-.65),d=1,sn=4855)
fore.210=nfore.arima.wge(x210,phi=c(1.25,-.64),d=1,n.ahead=50,limits=FALSE)
```

In Section 7.1.2 we fit the ARIMA(2,1,0) model $(1-1.25B+.64B^2)(1-B)X_t = a_t$, with $\hat{\sigma}_a^2 = 1.03$ to this dataset. Figure 7.14(b) shows the data in Figure 7.14(a) along with forecasts for the next 50 steps ahead. Close examination shows that the stationary AR(2) part of the model influences the first few forecasts, but that the forecasts quickly level out, in this case to –89.3. The last data value (that is, at $t = 200$) is –88.9. Consequently, the forecasts level off at a value that is neither the last value nor the sample mean of the first 200 values, $\bar{x} = -103.3$. Figure 7.14(c) shows the 50 forecasts along with forecast limits, and the next 50 actual data values in the simulated series. In this case the actual values decline rapidly while the forecasts are all close to the value at $t = 200$. The forecasts are not "good", but the data values stay within the 95% prediction limits.

Key Points:

1. Forecasts from am ARIMA(0,1,0) are simply a repetition of the last value
2. Short lead-time forecasts from an ARIMA(p,1,q) model with either $p > 0$ or $q > 0$ adjust for the underlying ARMA(p,q) behavior but eventually level out to a "horizontal" forecast line that tends to be different from the last observed data value.

The plots in Figure 7.14 were produced with the following code

```
x210=gen.arima.wge(n=250,phi=c(1.3,-.65),d=1,sn=4855,plot=FALSE)
t=1:250
plot(t[1:200],x210[1:200],xlim=c(0,250),type='l',xlab='Time')
fore.210=fore.arima.wge(x210[1:200],phi=c(1.25,-.64),d=1,n.ahead=50,limits=
FALSE)
fore.210=fore.arima.wge(x210[1:200],phi=c(1.25,-.64),d=1,n.ahead=50)
points(t[201:250],x210[201:250],type='l')
```

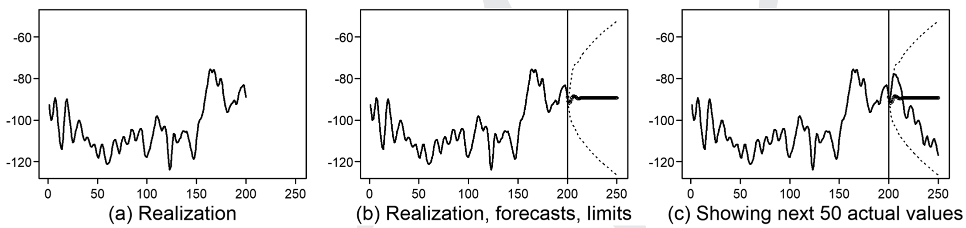


FIGURE 7.14 (a) Realization of length n = 200 from ARIMA(2,1,0) Model (b), (b) close-up of last 25 data values in (a) along with first 50 forecasts, and (c) realization in (a) along with 50 step ahead forecasts, 95% prediction limits and actual data values for t = 201– 250.

(5) Forecasts for ARIMA$(\mathrm{p,d,q})$ *Models with* d = 2.

We now return to the ARIMA(2,2,0) realization in Figure 7.8(a). These data are generated using the command

```
x220=gen.arima.wge(n=200,phi=c(1.3,-.65),d=2,sn=450)
```

The data follow a smooth curve which in the case of Figure 7.8(a) is almost linear. In contrast to the ARIMA models with $d = 1$, forecasts from an ARIMA model with $d = 2$ *do predict a current trend to continue.* It can be shown that the forecasts from an ARIMA(0,2,0) model follow a straight line that passes through the last two points in the realization. (See Woodward et al. (2017.)

The model (shown in Equation 7.4) fit to the data in Figure 7.8(a) is $(1-1.33B+.69B^2)(1-B)^2 X_t = a_t$, with $\hat{\sigma}_a^2 = 1.06$. The data and forecasts up to 50 steps ahead are shown in Figure 7.15(b) along with 95% prediction limits. In Figure 7.15(c) we show the realization in Figure 7.15(b) along with the actual data values for times t=201−250. As can be seen, the forecasts are quite good.

To further illustrate the nature of realizations from the ARIMA(2,2,0) model, Figure 7.15(d) shows a realization of length $n = 2{,}000$. The data values to the left of the vertical line at $n = 200$ are those plotted in Figure 7.15(a). As can be seen from Figure 7.15(d), realizations from ARIMA models with $d = 2$ tend to wander, but "change course" much more slowly than in the case of $d = 1$. Figure 7.15(d) shows that for fairly short-term forecasts, the decision to predict the existing trend to continue is reasonable. Plots in Figure 7.15 are produced using the following ***tswge*** code

```
par(mfrow=c(2,2))
x220=gen.arima.wge(n=2000,phi=c(1.3,-.65),d=2,sn=450,plot=FALSE)
t=1:2000
plot(t[1:200],x220[1:200],xlim=c(0,250),type='l',xlab='Time')
fore.220=fore.arima.wge(x220[1:200],phi=c(1.33,-.69),d=2,n.ahead=50)
abline(v=200)
fore.220=fore.arima.wge(x220[1:200],phi=c(1.33,-.69),d=2,n.ahead=50)
points(t[201:250],x220[201:250],type= 'l',lwd=2,col='black')
abline(v=200)
plot(t,x220, type='l')
abline(v=200)
```

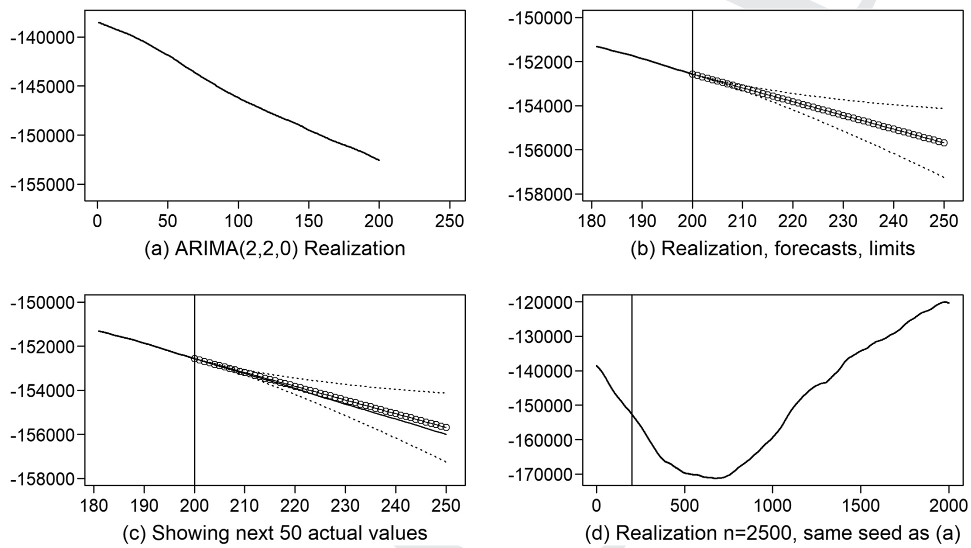


FIGURE 7.15 (a) Figure 7.8(a) showing realization of length n = 200 from ARIMA(2,2,0) Model (c); (b) last 20 data values in (a) along with forecasts and forecast limits for times t = 201-250; (c) Same as (b) including true values (solid line) slightly below the forecasts; and (d) Realization of length n = 2,000 for which the first 200 points are plotted in Figure 7.15(a)

7.2 SEASONAL MODELS

Seasonal models are nonstationary models that are useful for modeling data such as monthly or quarterly data which exhibit repetitive behavior. For example, each year December sales may tend to be high while sales in certain other months are traditionally low. In this section we will discuss seasonal ARIMA models which are defined in Definition 7.3.

Definition 7.3: A seasonal ARIMA model is a model of the form

$$\phi(B)\left(1-B^s\right)X_t = \theta(B)a_t, \tag{7.8}$$

where s is a non-negative integer, and where $\phi(B)$ and $\theta(B)$ are stationary and invertible operators of orders p and q, respectively (for which all roots of the characteristic equations $\phi(z)=0$ and $\theta(z)=0$ lie outside the unit circle).

Section 7.2.1 discusses the properties of the seasonal models, Section 7.2.2 illustrates how to fit these models to data, and Section 7.2.3 discusses forecasting using seasonal models.

7.2.1 Properties of Seasonal Models

The variable s is the length of the season. For example, for quarterly sales data which exhibit seasonal behavior, $s = 4$, while seasonal monthly data would be associated with $s = 12$. For the model in (7.8) to be a seasonal model, we require that $s > 1$. For example, if $s = 0$ or $s = 1$, then the model in (7.8) simplifies to an ARMA(p,q) and an ARIMA$(p,1,q)$ model, respectively, neither of which is a seasonal model.

7.2.1.1 Some Seasonal Models

In the following examples we will examine the properties of the seasonal models below which will be referred to as Models (A) and (B).

(A) $(1-B^s)X_t = a_t$: we consider the cases $s = 4$ and $s = 12$.
(B) $(1-1.3B+.65B^2)(1-B^4)X_t = a_t$

(1) Seasonal Model (A): $(1-B^s)X_t = a_t$
Model (A) is a "purely" seasonal model because it has no other autoregressive or moving average components. In this example the focus will be on $s = 4$ and $s = 12$.

Models with $s = 4$ will be referred to as models for quarterly data. For such data, quarters one through four correspond to year one, quarters five through eight are year two, and so forth.[7] The model $(1-B^4)X_t = a_t$ can be written as $X_t - X_{t-4} = a_t$, or in the more instructive form

$$X_t = X_{t-4} + a_t. \tag{7.9}$$

Equation 7.9 shows that the value of the process at time t is equal to the value at time $t-4$ plus a random noise component. A realization of length $n = 72$ from (7.9) was generated using the ***tswge*** command

```
xA4=gen.arima.wge(n=72,s=4,mu=50,sn=52)
```

The first 24 data values in this realization are shown in Figure 7.16(a). The dotted vertical lines separate the data into years. For convenience of discussion we will refer to the data as some measure of sales in thousands of dollars. Notice that in the first year, the first quarter ($t = 1$) sales tend to be about 18 (\$18k) with an increase in quarter two to about \$42k, followed by slight drops in the third and fourth quarters to sales in the range of \$36k each. The fifth quarter (that is, the first quarter in year two) drops to about \$18k as in the first year, and the general behavior is repeated for the following five years of data in the plot.

Figure 7.16(b) shows the first two years of a different set of monthly sales data using

```
xA12=gen.arima.wge(n=72,s=12,mu=50,sn=100)
```

7 While quarterly data based on "3-month quarters" is a very common application for seasonal data with $s = 4$, a "quarter" need not have this interpretation and could be appropriate for data recorded at any time interval.

From the plot it is seen that monthly sales tend to be highest in January, February, June, and November with the lowest sales in April, May, and July. (Strange sales pattern!!) Figure 7.16 plots were obtained using the commands

```
xA4=gen.arima.wge(n=72,s=4,mu=50,sn=52)
xA12=gen.arima.wge(n=72,s=12,mu=50,sn=100)
plot(xA4[1:24],type='l',xlab='Time')
abline(v=4.5,lty=2);abline(v=8.5,lty=2);abline(v=12.5,lty=2)
abline(v=16.5,lty=2);abline(v=20.5,lty=2)
plot(xA12[1:24],type='l',xlab='Time')
abline(v=12.5,lty=2)
```

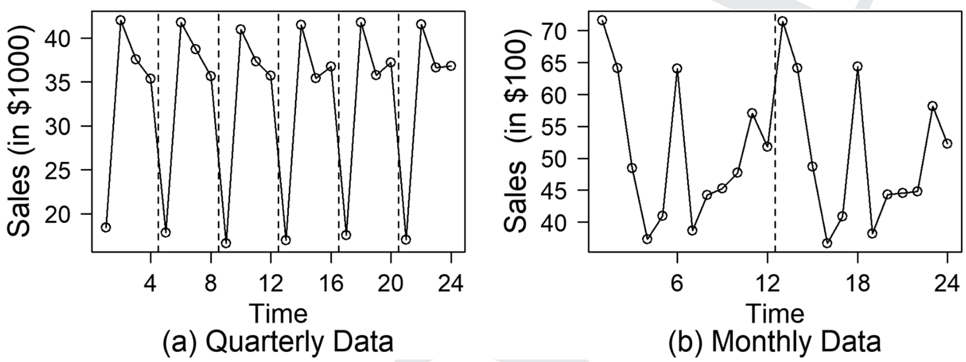


FIGURE 7.16 Realizations of length 24 from the seasonal Model (A) (a) with $s = 4$, and (b) with $s = 12$.

Key Point: For both $s = 4$ and $s = 12$, the seasonal nature within a year is not sinusoidal nor does it seem to follow any such mathematical pattern. However, the sales within a year do follow a particular, possibly irregular pattern, and the "seasonal" aspect of the data is the fact that this pattern is approximately repeated from "year to year".

(a) Factor Tables for Purely Seasonal Models with $\mathbf{s = 4}$ *and* $\mathbf{s = 12}$

In this section we show the factor tables for the purely seasonal models of order $\boldsymbol{s = 4}$ and $\boldsymbol{s = 12}$. Before showing the factor table for $\boldsymbol{s = 4}$ we note that the associated characteristic polynomial is $\mathbf{1 - z^4}$. From algebra, recall that $\mathbf{1 - z^4}$ can be factored as $\left(\mathbf{1 - z^2}\right)\left(\mathbf{1 - z^2}\right) = \left(\mathbf{1 - z}\right)\left(\mathbf{1 - z}\right)\left(\mathbf{1 - z^2}\right)$. Consequently, $\left(\mathbf{1 - B^4}\right)\boldsymbol{X_t} = \boldsymbol{a_t}$ can be factored as

$$\left(1 - B^4\right)X_t = \left(1 - B\right)\left(1 + B\right)\left(1 + B^2\right)X_t = a_t. \tag{7.10}$$

The factors in (7.10) are the three factors in the associated factor table. To find the factor table, we issue the ***tswge*** command

```
factor.wge(phi=c(0,0,0,1))
```

or

```
factor.wge(phi=c(rep(0,3),1))
```

because the polynomial $1 - z^4$ has zero coefficients for z, z^2, and z^3 and the coefficient on z^4 is +1. (Remember the treatment of signs for autoregressive and moving average coefficients.) The resulting factor table is given in Table 7.6, where it can be seen that $\left(1-B^4\right)X_t = a_t$ is associated with system frequencies of 0, .25, and .5 and also that all of the roots are on the unit circle.

TABLE 7.6 Factor Table for $\left(1-B^4\right)X_t = a_t$

AR-FACTOR	*ROOTS*	$\lvert r\rvert^{-1}$	f_0
$1-B$	1	1	0
$1+B^2$	$\pm i$	1	.25
$1+B$	–1	1	.50

Factoring the characteristic polynomial $1-z^{12}$ is more complicated and we use the **`factor.wge`** command for this purpose. The factor table for the purely seasonal model $\left(1-B^{12}\right)X_t = a_t$ is obtained using the command

```
factor.wge(phi=c(0,0,0,0,0,0,0,0,0,0,0,1))
```

or

```
factor.wge(phi=c(rep(0,11),1))
```

and is given in Table 7.7.

QR 7.8 Factor Table and Spectral Density of Model with $1-B^s$

TABLE 7.7 Factor Table for $\left(1-B^{12}\right)X_t = a_t$

AR-FACTOR	*ROOTS*	$\lvert r\rvert^{-1}$	f_0
$1-B$	1	1	.000
$1-1.732B+B^2$	$.866\pm.5i$	1	.083
$1-B+B^2$	$.5\pm.866i$	1	.167
$1+B^2$	$\pm i$	1	.250
$1+B+B^2$	$-.5\pm.866i$	1	.333
$1+1.732B+B^2$	$-.866\pm.5i$	1	.417
$1+B$	–1	1	.500

These two factor tables will be very useful as we model data and decide whether to include a seasonal factor in the model.

Key Points:

1. The factor tables for $\left(1-B^4\right)X_t = a_t$ and $\left(1-B^{12}\right)X_t = a_t$ show that all roots of the associated characteristic equations are on the unit circle. Consequently, these models are nonstationary. This is true for $\left(1-B^s\right)X_t = a_t$ for any positive integer s.
2. Quarterly data is not the same as $s=4$ and monthly data does not necessarily indicate that the use of a model with $s=12$ is warranted. See (3) below.
3. Factor Tables 7.6 and 7.7 show the factors associated with the seasonal models $\left(1-B^4\right)X_t = a_t$ and $\left(1-B^{12}\right)X_t = a_t$, respectively. If we have quarterly or monthly data, then the factor table associated with a model fit to the data *should have factors similar to those in the above corresponding factor tables in order to be justified to include the seasonal factor* $1-B^4$ *or* $1-B^{12}$ *in the model*. We will return to this in Section 7.2.2 where we discuss model identification.(4)

(b) Autocorrelations and Spectral Density for $\left(1-\mathrm{B}^{4}\right)\mathrm{X}_{\mathrm{t}}=\mathrm{a}_{\mathrm{t}}$ *and* $\left(1-\mathrm{B}^{12}\right)\mathrm{X}_{\mathrm{t}}=\mathrm{a}_{\mathrm{t}}$

Consider the quarterly seasonal model $\left(1-B^{4}\right)X_t=a_t$ and note again that the realization in Figure 7.16(a) shows a very strong correlation between data values that are four quarters apart. That is, $\hat{\rho}_4$ should be strongly positive. Also, sales for the same quarter two years apart should also be positively correlated. That is, we should also expect $\hat{\rho}_8$ to be strongly positive. In general, $\hat{\rho}_{4m}$ will tend to be positive for integers *m*. Also note that when the seasonal model $\left(1-\mathrm{B}^{4}\right)X_t=a_t$ is written in the form $X_t=X_{t-4}+a_t$, it is clear that the model does not "say anything" about the relationship between, for example, X_t and X_{t+1}. Consequently, the sample autocorrelations, $\hat{\rho}_k$, for integers k that are not multiples of four would be expected to be smaller in magnitude than those for which k is a multiple of four. Using the same logic, for the monthly seasonal model $\left(1-B^{12}\right)X_t=a_t$, sample autocorrelations $\hat{\rho}_k$ where k is an integer multiple of 12 will tend to be positive and larger in magnitude than the other sample autocorrelations.[8]

Figure 7.17(a) is a plot of a realization of length $n=72$ from a purely seasonal model with $s=4$. Note that the data in Figure 7.16(a) are simply the first 24 data values of the realization in Figure 7.17(a). The sample autocorrelations and Parzen spectral density estimate for the data in Figure 7.17(a) are shown in Figures 7.17(b) and (c), respectively. The quarterly behavior can be seen in Figure 7.17(a), but it is not as obvious as it was in the first 24 values shown in Figure 7.16(a). The sample autocorrelations at lags 4, 8, 12, … are "large" and positive which is consistent with the above discussion. Also, the spectral density has peaks at the system frequencies (see Table 7.6), $f=0$, .25, and .5.

Figure 7.17(d) is a realization of length $n=72$ from a purely seasonal model with $s=12$,and for which the first 24 values are shown in Figure 7.16(b). As would be expected, the sample autocorrelations in Figure 7.17(e) at lags $k=12,24$, and so forth are large and positive while the peaks in the Parzen spectral density estimate in Figure 7.17(f) correspond to the system frequencies in Table 7.7 (although the peaks at $f=0$ and .5 are weak). The plots in Figure 7.16 can be obtained using the commands

```
xA4=gen.arima.wge(n=72,s=4,mu=50,sn=52)
xA12=gen.arima.wge(n=72,s=12,mu=50,sn=100)
plotts.sample.wge(xA4)
plotts.sample.wge(xA12,trunc=32)
```

Key Point: In practice, the main feature of sample autocorrelations for seasonal data is that autocorrelations at multiples of 4 for $s=4$ (or of 12 for $s=12$) tend to be positive and larger in magnitude than sample autocorrelations at lags that are not multiples of *s*.

8 The seasonal model $\left(1-B^{s}\right)X_t=a_t$ is a nonstationary model, and as in the ARIMA case, the autocorrelations are defined as "limiting autocorrelations". It can be shown that the limiting autocorrelations for $\left(1-B^{s}\right)X_t=a_t$ are $\rho_0=\rho_s=\rho_{2s}=\cdots=1$; that is, $\rho_{ks}=1$ for all integer *k*, positive or negative. The model-based autocorrelations for lags that are not integer multiples of s are equal to zero.

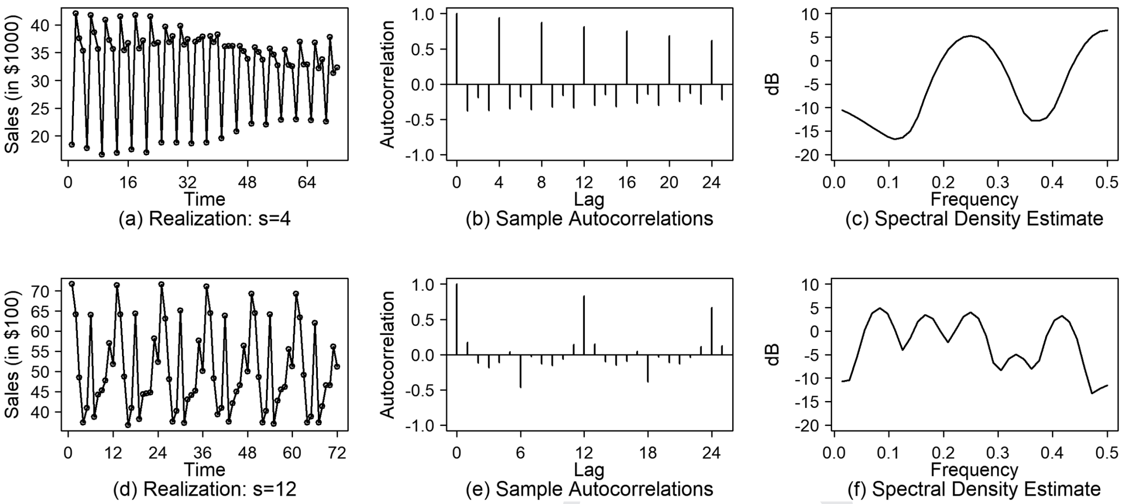


FIGURE 7.17 (a) Realization of length n = 72 from the model $\left(1-B^4\right)X_t = a_t$; (b) sample autocorrelations; (c) Parzen spectral density estimate; (d) Realization of length n=72 from the model $\left(1-B^{12}\right)X_t = a_t$; (e) sample autocorrelations; (f) Parzen spectral density estimate with M=30.

(2) Seasonal Model (B): $\left(1-1.3B+.65B^2\right)\left(1-B^4\right)X_t = a_t$

Seasonal Model (B) not only has the seasonal factor $\left(1-B^4\right)$ but also contains the stationary autoregressive factor, $\left(1-1.3B+.65B^2\right)$,which is the same AR(2) component used in the previous ARIMA examples. Recall that in the ARIMA case, the factors $(1-B)^d$ dominated the behavior associated with the stationary factors. In the case of seasonal Model (B), the nonstationary factor is $\left(1-B^4\right)$. The command

```
xB=gen.arima.wge(n=100,s=4,phi= c(1.3,-.65),sn=290)
```

generates a realization from Model (B). Figures 7.18(a−c) display the resulting realization, sample autocorrelations, and Parzen spectral density estimate. Again, for convenience of discussion, we assume the simulated data are sales (in $1,000). The realization in Figure 7.18(a) is of length $n = 100$ (that is, 25 years). The quarterly behavior of the data is not very obvious, although throughout the realization the first quarter sales tend to be much lower than those in the other three quarters. The main thing to notice about the sample autocorrelations is that they are strongly positive at lags that are multiples of four. It will often be the case that sample autocorrelations at lags say, $k = 2,6,\ldots$ may not be very close to zero but will tend to be smaller in magnitude than those at multiples of four.

Recall that in order to reveal the effect of the stationary factors in the ARIMA data we needed to difference the data. In the seasonal case, we apply a *seasonal difference* $Y_t = X_t - X_{t-s}$, in order to remove the dominating nonstationary behavior. In the $s = 4$ case, seasonally differencing the data in Figure 7.18(a) yields $Y_t = X_t - X_{t-4} = \left(1-B^4\right)X_t$. Notice that because $\left(1-1.3B+.65B^2\right)\left(1-B^4\right)X_t = a_t$, then $Y_t = \left(1-B^4\right)X_t$ satisfies the stationary AR(2) model $\left(1-1.3B+.65B^2\right)Y_t = a_t$. The seasonal transformation is accomplished using the ***tswge*** command

```
sB=artrans.wge(xB,phi.tr=c(0,0,0,1))
```

The seasonally transformed data are shown in Figure 7.18(d) along with the associated sample autocorrelations and Parzen spectral density estimates in Figures 7.18 (e) and (f), respectively. The cyclic behavior has a cycle length of about 10 time units (quarters in our case). The transformed data and their characteristics appear similar to those shown in Figure 7.6 for the associated stationary AR(2) model, but these features were not apparent in the original dataset, **`xB`.**

The plots in Figure 7.17 were obtained using the following code

```
xB=gen.arima.wge(n=100,s=4,phi= c(1.3,-.65),sn=290)
sB=artrans.wge(xB,phi.tr=c(0,0,0,1),plot=FALSE)
plotts.sample.wge(xB)
plotts.sample.wge(sB)
```

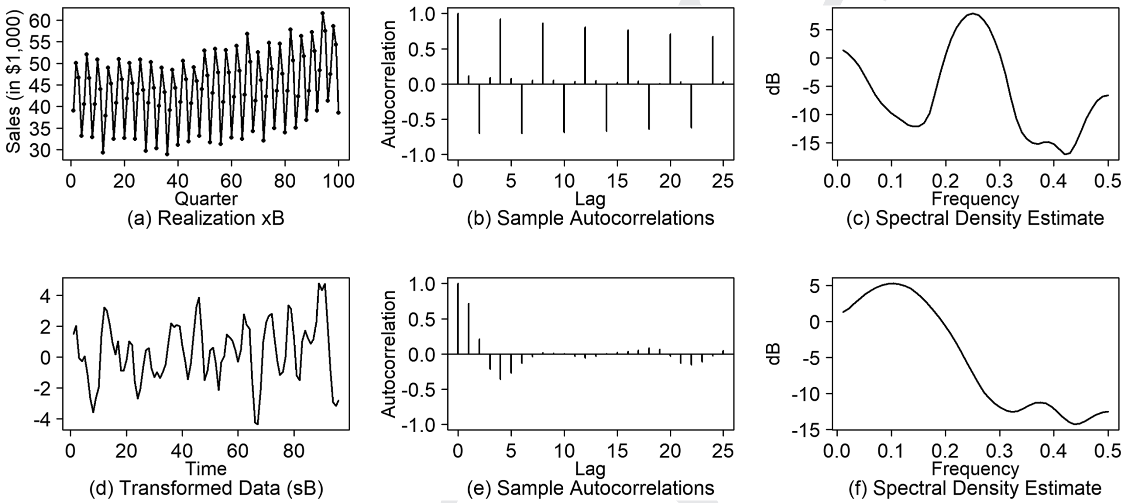


FIGURE 7.18 (a) Realization (`xB`) of length n=100, (b) sample autocorrelations, and (c) Parzen spectral density estimate from Model B: $(1-1.3B+.65B^2)(1-B^4)X_t = a_t$, (d) data in (a) transformed by $(1-B^4)$, (e) sample autocorrelations, and (f) Parzen spectral density estimate of the data in (d).

(a) The Airline Model

We return to the Base R **`AirPassengers`** dataset which was discussed at length in Chapters 1 and 2. In Chapter 2 we produced Holt-Winters forecasts for the **`AirPassengers`** data and for the logarithms of the **`AirPassengers`** data. Recall that the two datasets differ in that the within-year variability increased in the raw **`AirPassengers`** data while this variability was fairly constant in the log data. See Figure 2.7. The log data are shown in Figure 7.19(a) along with the associated sample autocorrelations in Figure 7.19(b) and the Parzen spectral density estimate in Figure 7.19(c). The plots in Figure 7.19 are obtained using the commands

```
data(AirPassengers)
logAP=log(AirPassengers)
plotts.sample.wge(logAP)
```

Key Points:

1. In the introduction to this chapter it was noted that a violation of stationarity occurs if the process variance changes with time.
2. The logarithm of the **`AirPassengers`** data is taken so that the variance doesn't change with time.

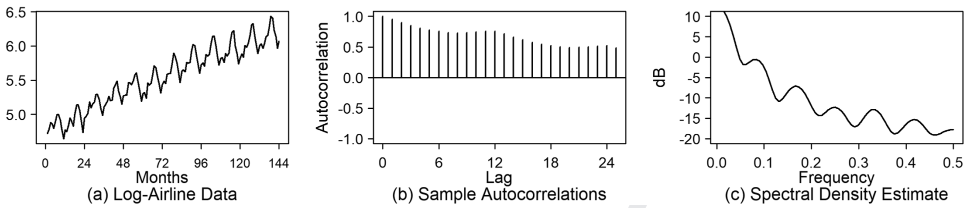


FIGURE 7.19 (a) Log Air Passengers data, (b) sample autocorrelations of the data in (a), and (c) Parzen spectral density estimate.

Notice that the **AirPassengers** data are like the seasonal dataset in Figure 7.17(d) but have the additional feature that there is a strong trending behavior. Also, the sample autocorrelations have slight "bumps" at lags 12 and 24, but they are dominated by the appearance of a slowly damping exponential behavior. The Parzen spectral density estimate has peaks at the system frequencies in Table 7.7 but the peaks at and near $f = 0$ are accentuated. Many analysts have fit models to the log **AirPassengers** data of the form:

$$\phi(B)(1-B)\left(1-B^{12}\right)X_t = \theta(B)a_t. \tag{7.11}$$

In fact, models containing $(1-B)\left(1-B^{12}\right)$ are sometimes referred to as "Airline Models". Note that the system frequencies for $1-B^{12}$ shown in Table 7.7 contain a factor, $1-B$, so that model (7.11) has two factors of $1-B$.Consequently, the behavior of data from the Airline Model would be expected to have seasonal behavior along with trending behavior such as was seen in the realization in Figure 7.8(a) from a model containing $(1-B)^2$. Models containing the factors $(1-B)\left(1-B^{12}\right)$ will be discussed in more detail when we discuss forecasting using nonstationary models.

7.2.2 Fitting Seasonal Models to Data

In this section we discuss the issues involved with fitting a seasonal model to a time series realization. As mentioned previously, in seasonal datasets the nonstationary component is the seasonal factor, $1-B^s$. The procedure for finding models for seasonal data is a generalization of the technique used for ARIMA models:

(1) *Select the value of s* :That is, determine whether a seasonal factor is appropriate for the data, and if so select the value of s.

(2) *Seasonally difference the data:* For example, if $s = 4$, then seasonally difference the data to obtain $Y_t = \left(1-B^4\right)X_t = X_t - X_{t-4}$. Letting $\mathbf{x}$ denote the original data, then the seasonally differenced data can be obtained using the command

```
y=artrans.wge(x,phi.tr=c(0,0,0,1))
```

(3) *Fit an ARMA(p,q) model to the seasonally differenced data:* Note that for data satisfying the Airline Model, the seasonally differenced data, Y_t, may show evidence of a unit root, in which case the Y_t (the seasonally differenced data) would need to be differenced.

Step 1, that is, determining whether a seasonal factor is required and if so, identifying the seasonal order, s, is the most difficult step in this procedure. We highly recommend the overfitting procedure described below for this purpose.

7.2.2.1 Overfitting

As mentioned previously, simply because data are collected monthly does not imply that a seasonal factor, $1-B^{12}$, should be included in the model. Recall that the factor tables in Tables 7.6 and 7.7 show the underlying 1st- and 2nd-order factors of the polynomials $1-B^4$ and $1-B^{12}$, respectively. Although $s=4$ and $s=12$ are the most common seasonal factors, similar factor tables could be obtained for any positive integer value of s.

The strategy of overfitting is based on a result due to Tiao and Tsay (1983) that loosely says that if a high order AR(p) model is fit to a realization from a nonstationary model with roots on the unit circle, then factors associated with "nonstationary factors" will "show up" in the AR factor tables for autoregressive fits to the data for a wide range of autoregressive orders. Specifically, overfitting involves fitting several "high" order AR models to a set of data to identify possible nonstationarities. For a seasonal model with $s=4$ there are three factors associated with roots on the unit circle: $1-B, 1+B$, and $1+B^2$. The Tiao-Tsay result says that high order AR fits to data from the model $\phi(B)\left(1-B^4\right)X_t=\theta(B)a_t$ should consistently tend to show factors close to these three factors. Woodward et al. (2017) recommend fitting "high order AR models" using Burg estimates. Recall that Burg estimates always produce a stationary model, so we will be looking for factors "close" to the three seasonal factors in the $s=4$ case. Our strategy will be to use Burg estimates for overfitting.[9] The overfitting procedure will be illustrated as we fit seasonal models below.

(1) Model (A): Final Model for Data in Figure 7.17(d)
Consider again the data in Figure 7.17(d) which were generated from the seasonal model

$$\left(1-B^{12}\right)X_t=a_t. \tag{7.12}$$

where $\sigma_a^2=1$. These data were generated using the command

```
xA12=gen.arima.wge(n=72,s=12,mu=50,sn=100)
```

(1) *Select the value for s:* The overfitting procedure involves fitting a few high order *autoregressive models* to the data and examining the factor tables. Table 7.8 shows the factor table for an AR(16) fit to the data. Note that if the data are monthly, and because we suspect that a 12th-order seasonal factor is appropriate, then the AR order of the fit to the data should be larger than 12. The command for obtaining the overfit factor table in Table 7.8 is[10]

```
est.ar.wge(xA12,p=16,type='burg')
```

9 ML estimates could also be used, but there are cases in which the ML estimation procedure does not converge.

10 Note that the factor table is returned in the output from `est.ar.wge` so we did not need to use `factor.wge`.

TABLE 7.8 $p = 16$ Overfit Factor Tables for Realization from $\left(1 - B^{12}\right)X_t = a_t$ AR(16) Overfit

AR-FACTOR	*ROOTS*	$\lvert r \rvert^{-1}$	f_0
$1-1.736B+.9996B^2$	**$.868 \pm .497i$**	**.9998**	**.083**
$1+1.723B+.9992B^2$	**$-.862 \pm .507i$**	**.9996**	**.415**
$1-.012B+.999B^2$	**$.006 \pm 1.001i$**	**.9995**	**.249**
$1+.976B+.993B^2$	**$-.491 \pm .875i$**	**.997**	**.331**
$1-.964B+.993B^2$	**$.485 \pm .878i$**	**.997**	**.167**
$1-.991B$	**-1.009**	**.991**	**.500**
$1-.981B$	**1.019**	**.981**	**.000**
$1-.906B+.736B^2$	$.616 \pm .990i$	.858	.161
$1+.816B+.333B^2$	$-1.225 \pm 1.225i$	.577	.375

Take a few minutes to compare the first seven rows of Table 7.8 (shown in bold type) with Table 7.7. Note that Table 7.7 shows the actual factored version of $1-B^{12}$ while Table 7.8 is the factor table associated with an AR(16) fit to the data in Figure 7.17(d). Notice how the first seven lines of Table 7.8 closely approximate Table 7.7 (the factors are in different orders in the two tables). Notice also that the other two factors in Table 7.8 are much further from the unit circle than are the "seasonal" factors in the first seven lines of the table. If you have actual monthly data, and if the associated overfit factor table is similar to the one in Table 7.8, then a seasonal model with $s = 12$ is a reasonable model. Overfitting in this setting usually involves looking at factor tables for a variety of values for $p > 12$. For example, factor tables for $p = 14, 15$, and 17 (not shown – but you should check them out) exhibit similar patterns.[11]

(2) *Seasonally difference the data:* Based on the previous discussion, we select $s = 12$ and seasonally difference the data which adds a factor of $1-B^{12}$ to the model. The command to seasonally difference the data in **`xA12`** is

```
sA12=artrans.wge(xA12,phi.tr=c(rep(0,11),1))
```

(3) *Fit an ARMA(p,q) model to the seasonally differenced data:* For convenience, Figure 7.17(d–f) is repeated as Figure 7.20(a–c). These show the data in **`xA12`**, along with the corresponding sample autocorrelations and Parzen spectral density estimate. The seasonally differenced data, **`sA12`**, are shown in Figure 7.20(d) along with the sample autocorrelations and Parzen spectral density estimate in Figures 7.20(e) and (f), respectively. The seasonally differenced data appear uncorrelated, the sample autocorrelations are very close to zero and tend to stay within the 95% limit lines,and the Parzen spectral density estimate is fairly flat. These are characteristics of white noise data which indicate that no further analysis is needed. This is consistent with the fact that the true model is $\left(1-B^{12}\right)X_t = a_t$, in which case the seasonally differenced data, $Y_t = \left(1-B^{12}\right)X_t$ follows the model $Y_t = a_t$. That is, the seasonally differenced data are from a white noise model. The white noise variance is estimated by the variance of the differenced data using the base R command **`var(sA12)`** from which we find that the sample variance of **`sA12`** is .97. Consequently, because the seasonal difference added a $1-B^{12}$ term to the model, the final model is

$$\left(1-B^{12}\right)X_t = a_t,$$

with $\hat{\sigma}_a^2 = .97$, which is very similar to the true model. The plots in Figure 7.20 can be obtained using the code

11 Factor tables of excessively high order (say 25 or more) will begin to bring several roots closer to the unit circle.

```
xA12=gen.arima.wge(n=72,s=12,mu=50,sn=100)
sA12=artrans.wge(xA12,phi.tr=c(rep(0,11),1),plot=FALSE)
plotts.sample.wge(xA12)
plotts.sample.wge(sA12)
```

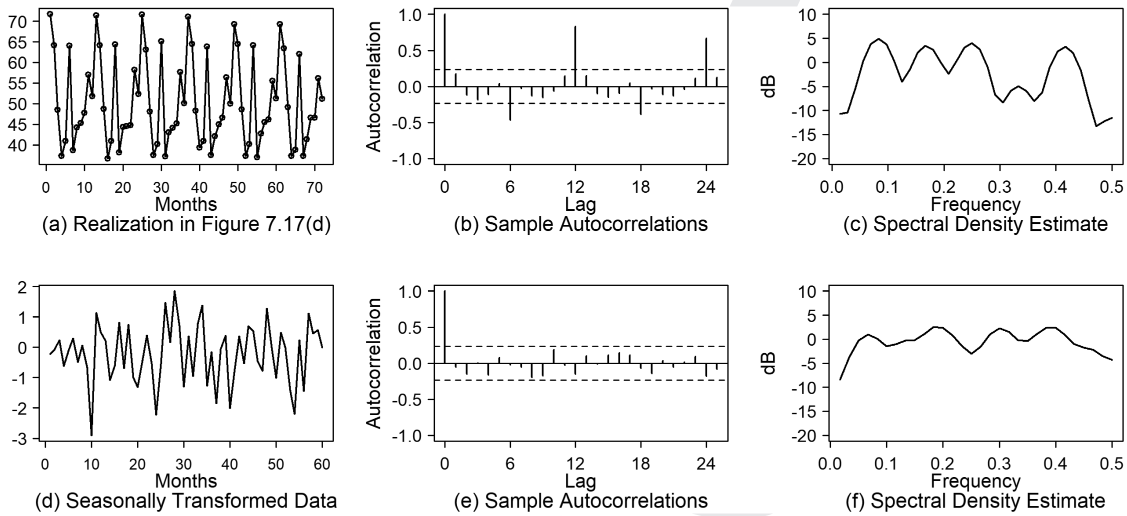


FIGURE 7.20 (a) Data in Figure 7.17(d), (b) sample autocorrelations of data in (a), (c) Parzen spectral density of data in (a) with M=30, (d) data in (a) seasonally transformed with $s = 12$, autocorrelations of the data in (d), and (c) Parzen spectral density estimate of data in (d) with M=30.

(2) Model (B): Final Model for Data in Figure 7.18(a)
Consider again the data in Figure 7.18(a) which were generated from the seasonal model

$$\left(1-1.3B+.65B^2\right)\left(1-B^4\right)X_t = a_t.$$

where $\sigma_a^2 = 1$.The realization in Figure 7.18(a) was generated using the command

```
xB=gen.arima.wge(n=100,phi=c(1.3,-.65),s=4,sn=290)
```

(1) *Select the value for s:* We again use the overfitting procedure. Because we are assuming that this is quarterly data we use the overfit procedure with values of p such as $p = 8,10$, and 12 (that is, we do not have to assure that the overfit order is greater than 12 as we did when we assumed a 12th-order seasonality). However, overfitting with $p = 14$ gives similar results to those used below. The overfit factor table is shown for $p = 10$ in Table 7.9. We examined other values of p and obtained similar results. The command for obtaining the overfit factor table in Table 7.9 is

```
est.ar.wge(xB, p=10,type='burg')
```

Comparing the first 3 lines of Table 7.9 (shown in bold) with Table 7.6 we see that the factors in Table 7.9 with roots closest to the unit circle are very similar to the factors of $1 - B^4$, that is, $1 - B, 1 + B^2$, and $1 + B$. The other factors in Table 7.9 are associated with roots much further from the unit circle. Note that $1 - .974B$ is not as close to the unit circle as the other two. However, its presence in the factor table along with the other two factors that are very close to $1 + B^2$ and

$1+B$ suggests that a seasonal factor of $1-B^4$ is a reasonable choice, especially if the data have a natural 4th-order seasonal behavior, such as quarterly sales data. Therefore, we decide to fit a seasonal model with $s = 4$.

TABLE 7.9 $p = 10$ Overfit Factor Tables for Realization from $(1-1.3B+.65B^2)(1-B^4)X_t = a_t$ AR(10) Overfit

AR-FACTOR	*ROOTS*	$\lvert r\rvert^{-1}$	f_0
$1-.0013B+.9990B^2$	**$.000\pm1.0005i$**	**.9995**	**.250**
$1+.995B$	**-1.005**	**.995**	**.500**
$1+.974B$	**1.027**	**.974**	**.000**
$1-.826B+.633B^2$	$.653\pm1.075i$	.795	.163
$1-1.140B+.612B^2$	$1.139\pm.579i$	.783	.075
$1+B+.366B^2$	$-1.365\pm.931i$	.605	.405

(2) *Seasonally difference the data:* Based on the previous discussion, we select $s = 4$, and seasonally difference the data using

```
sB4=artrans.wge(xB,phi.tr=c(0,0,0,1))
```

The seasonal 4th-order seasonal difference adds a factor of $1-B^4$ to the model.

(3) *Fit an ARMA(p,q) model to the seasonally differenced data:* The seasonally differenced data in **`sB4,`** are shown in Figure 7.18(d) along with the sample autocorrelations and Parzen spectral density estimate in Figures 7.18(e) and (f). Again, we see cyclic behavior and quickly damping sample autocorrelations only after seasonally differencing the data. Specifically, the data appear to be stationary but not white noise. The following command uses AIC to select ARMA model orders for the seasonally differenced data (in **`sB4`**) letting p range from 0 to 10 and q from 0 to 2.

```
aic.wge(sB4,p=0:10,q=0:2)
```

AIC selects $p = 2$ and $q = 0$, so the following command is used to estimate the parameters of the AR(2) model fit to the stationary, seasonally differenced data.

```
est.ar.wge(sB4,p=2)
```

The ML estimates for the differenced data are $\hat{\phi}_1 = 1.19$, $\hat{\phi}_2 = -.64$, and $\hat{\sigma}_a^2 = 1.04$, so the final model is

$$(1-1.19B+.64B^2)(1-B^4)X_t = a_t, \tag{7.13}$$

with $\hat{\sigma}_a^2 = 1.04$. This model is "similar" to the true Model (B), and the factor table (check it out!) shows that the fitted model has a system frequency close to $f = .10$ which is consistent with the features of the true AR(2) model (see Table 7.1). Recall that because the factor $1-B$ is a factor of $1-B^4$, we do not estimate the mean.

Example 7.2 Air Passengers Data

In Figure 7.19(a) we showed the log of the monthly number of airline passengers (in thousands) for the 12 years, 1949–1960. We noted that the **`AirPassengers`** data appear similar to the seasonal datasets in Figure 7.17(d) with the additional feature that there is a strong trending behavior. We also noted that the sample autocorrelations (Figure 7.19(b)) have slight "bumps" at lags 12 and 24 but are dominated by the appearance of a slowly damping exponential behavior. The spectral density (Figure 7.19(c)) has peaks at the system frequencies in Table 7.7 but the peaks at and near $f = 0$ are accentuated. Figures 7.19(a–c) are obtained using the commands

```
data(AirPassengers)
logAP=log(AirPassengers)
plotts.sample.wge(logAP)
```

(1) *Select the value for s:* Because these are monthly data, we use the overfit procedure to determine whether the factors associated with $1-B^{12}$ are in the model. For this purpose we fit an AR(15) using Burg estimates

```
est.ar.wge(airlog,p=15,type='burg')
```

The resulting factor table is shown in Table 7.10, and in Table 7.11 we compare the factors of the AR(15) overfit (shown in bold) with those of $1-B^{12}$.

TABLE 7.10 $p = 15$ Overfit Factor Table with $p = 15$ for Log `AirPassengers` Data AR(15) Overfit

AR-FACTOR	*ROOTS*	$\lvert r\rvert^{-1}$	f_0
$1-1.723B+.995B^2$	**$.867\pm.504i$**	**.998**	**.084**
$1-.986B+.995B^2$	**$.496\pm.872i$**	**.998**	**.168**
$1-.987B+.985B^2$	**$-.501\pm874i$**	**.993**	**.333**
$1+.038B+.979B^2$	**$0.19\pm.1.011i$**	**.989**	**.247**
$1-1.9697B+.9704B^2$	**$1.015\pm.022i$**	**.985**	**.003**
$1-1.689B+.950B^2$	**$-.889\pm.512i$**	**.975**	**.417**
$1-.854B$	**-1.171**	**.854**	**.500**
$1+.600B$	-1.667	.600	.500
$1-.111B$	9.007	.111	.000

TABLE 7.11 Comparison of the Factors of $1-B^{12}$ with Those Shown in Table 7.10

FACTOR IN AR(15) OVERFIT	*ASSOCIATED FACTOR OF* $1-B^{12}$
$1-1.723B+.995B^2$	$1-1.732B+B^2$
$1-.986B+.995B^2$	$1-B+B^2$
$1+.987B+.985B^2$	$1+B+B^2$
$1-.038B+.979B^2$	$1+B^2$
$1-1.9697B+.9704B^2$	See discussion below
$1-1.689B+.950B^2$	$1-1.732B+B^2$
$1+.854B$	$1+B$

From Table 7.11 we see that the factor $1-B$ does not seem to be represented in the AR(15) factor table. However, the factor $1-1.9697B+.9704B^2$ is close to the factor $1-2B+B^2=(1-B)(1-B)$, so that the AR(15) factor table suggests two factors of $1-B$. This is consistent with the Airline Model in which $(1-B)(1-B^{12})$ contains two $1-B$ factors. Notice also that the factor $1+.854B$ is not very close to the unit circle. However, because there is such a close approximation to the other factors of $(1-B)(1-B^{12})$, we believe that there is sufficient evidence to suggest fitting a model with these two "Airline Model" factors.[12]

(2) *Transform the data using* $(1-B)(1-B^{12})$: To perform this transformation on **`airlog`**, we first apply a seasonal transformation and then difference the transformed data. We use the commands

```
s12=artrans.wge(logAP,phi.tr=c(rep(0,11),1))
s12d1=artrans.wge(s12,phi.tr=1)
```

(3) *Fit an ARMA(p,q) model to the transformed data:* The log **`AirPassengers`** data along with the sample autocorrelations and Parzen spectral density estimate are shown in the first row of Figure 7.21. The log **`AirPassengers`** data transformed by $(1-B)(1-B^{12})$ are shown in Figure 7.21(d) along with sample autocorrelations and spectral density in Figures 7.21(e)–(f).

The sample autocorrelations at lags $k=1$ and $k=12$, shown in Figure 7.21(e), are outside the limit lines leading us to believe that there may still be some 12$^{\text{th}}$-order seasonal behavior to be modeled. For this reason, we issue the following command to let AIC select the top five ARMA model orders for the transformed data (in **`s12d1`**) letting p range from 0 to 15 and q from 0 to 2. (We let p extend beyond 12 to capture 12$^{\text{th}}$-order seasonal behavior that might still be present.)

```
aic5.wge(s12d1,p=0:15,q=0:2)
```

AIC selects an ARMA(12,1) followed by an AR(13) as the top two models. We fit the data with an AR(13) model and use Burg estimates to be consistent with the model selected by Gray and Woodward (1981). The command

```
est.ar.wge(s12d1,p=13,type='burg')
```

produces the model

$$\phi_{13}(B)(1-B)(1-B^{12})X_t = a_t, \tag{7.14}$$

where

$$\begin{aligned}\phi_{13}(B) = 1 + .41B + .06B^2 + .13B^3 + .09B^4 - .05B^5 - .09B^6 + .01B^7 - .03B^8 \\ - .15B^9 - .01B^{10} + .11B^{11} - .44B^{12} - .14B^{13},\end{aligned} \tag{7.15}$$

and $\hat{\sigma}_a^2 = 0.0013$. These commands produce the plots in Figure 7.21.

```
data(AirPassengers)
airlog=log(AirPassengers)
s12=artrans.wge(airlog,phi.tr=c(rep(0,11),1),plot=FALSE)
s12d1=artrans.wge(s12,phi.tr=1,plot=FALSE)
plotts.sample.wge(airlog)
plotts.sample.wge(s12d1)
plotts.sample.wge(s12,arlimits=TRUE)
plotts.sample.wge(s12d1,arlimits=TRUE)
```

QR 7.9 Air Passenger Example

12 Note that the factor immediately following $1+.854B$ in the AR(15) overfit is $1+.600B$ which "enhances" the effect of the factor associated with $f=.5$.

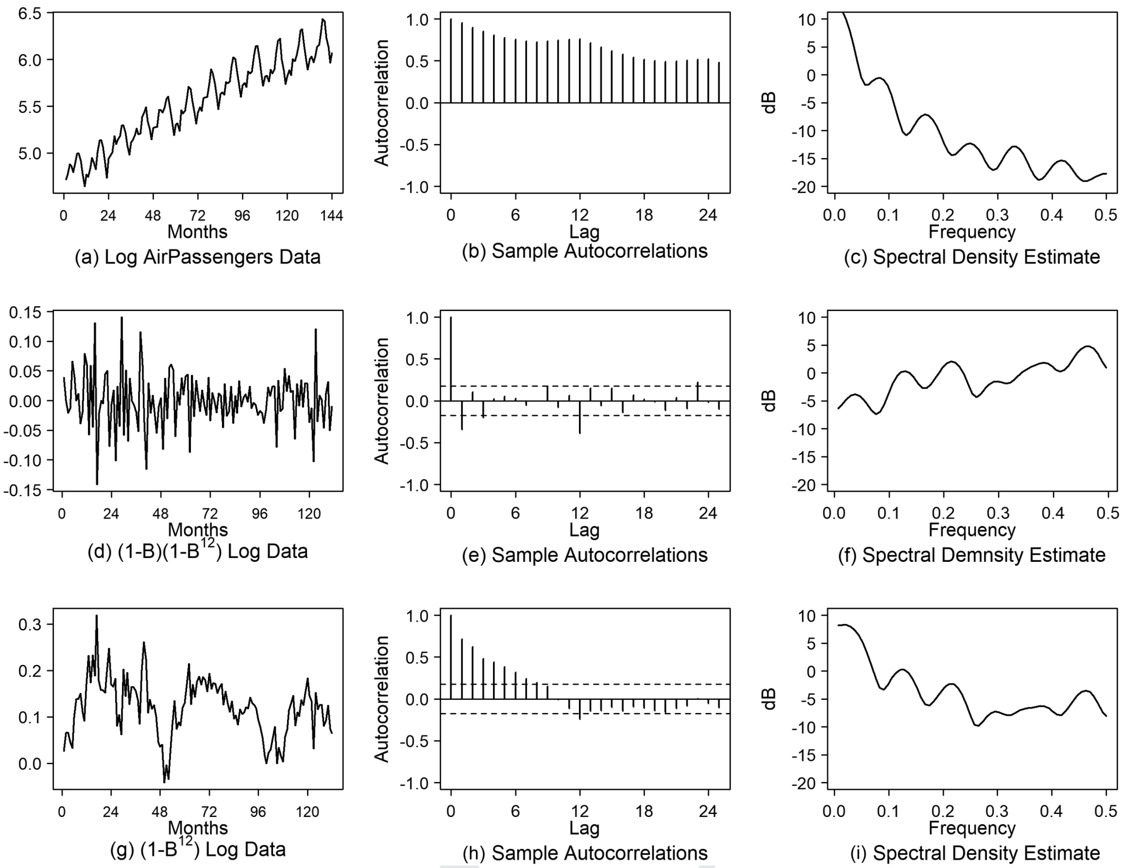


FIGURE 7.21 (a) Log **`AirPassengers`** data in Figure 7.19, (b) sample autocorrelations, and (c) Parzen spectral density estimate, (d) Log **`AirPassengers`** data transformed by $(1-B)(1-B^{12})$, (e) sample autocorrelations of the data in (d) showing limit lines, and (f) Parzen spectral density estimate, (g) Log **`AirPassengers`** data transformed by $(1-B^{12})$, (e) sample autocorrelations of the data in (d) showing limit lines, and (f) Parzen spectral density estimate.

(3) An Alternative Model for the Log `AirPassengers` *Data*

One strategy might be to examine the factors in Table 7.10 and decide to transform the data by $1-B^{12}$ (instead of $(1-B)(1-B^{12})$) and then model the transformed data. To this end, Figures 7.21(g)–(i) show the log **`AirPassengers`** data transformed by $1-B^{12}$ along with the associated sample autocorrelations and spectral density. The data are wandering, the sample autocorrelations damp (fairly quickly), and the spectral density has a peak at $f=0$. Table 7.12 is the overfit factor table using $p=15$ for log **`AirPAssengers`** data transformed only by $1-B^{12}$.

TABLE 7.12 $p=15$ Overfit Factor Table for the Log `AirPassengers` Data Transformed by $1-B^{12}$

AR-FACTOR	*ROOTS*	$\lvert r\rvert^{-1}$	f_0
$1-1.323B+.912B^2$	$.726\pm.755i$	.955	.128
$1-.453B+.910B^2$	$.249\pm1.018i$	.954	.212
$1+1.821B+.885B^2$	$-1.029\pm.268i$	.941	.460
$1-1.828B+.876B^2$	$1.044\pm.230i$	.936	.035
$1+1.289B+.820B^2$	$-.786\pm.776i$	.906	.376

TABLE 7.12 Continued

AR-FACTOR	*ROOTS*	$\lvert r\rvert^{-1}$	f_0
$1+.545B+.753B^2$	$-.362\pm1.094i$	.868	.301
$1-.837B$	1.195	.837	.000
$1+.274B+.095B^2$	$-1.445\pm2.907i$	.308	.323

We would have expected to see a factor close to $1-B$ to show up in the factor table after transforming only by $1-B^{12}$. However, the overfit table is not suggestive of a nearly nonstationary factor close to $1-B$. The related factor is $1-.837B$ which is the next to last factor in Table 7.12. This suggests modeling the $1-B^{12}$ transformed data with a stationary model. Using the strategy in obtaining the model (7.16), we use AIC to find the "best" AR model using Burg estimates for the seasonally differenced data in **s12.** Because we are dealing with data having 12th -order seasonality, we let p range from 0 to 15 using the commands

```
s12=artrans.wge(airlog,phi.tr=c(rep(0,11),1))
aic5.wge(s12,p=0:15,q=0:0)
# AIC selects an AR(13)
est.ar.wge(s12,p=13)
```

AIC selects an AR(13), and the final model is

$$\phi_{s13}(B)\left(1-B^{12}\right)X_t = a_t, \tag{7.16}$$

where

$$\begin{aligned}\phi_{s13}(B)=\phi_{13}(B)=1&-.54B-.28B^2+.08B^3-.02B^4-.14B^5-.05B^6+.10B^7\\&-.04B^8-.14B^9+.14B^{10}+.14B^{11}+.32B^{12}-.30B^{13}.\end{aligned} \tag{7.17}$$

and $\hat{\sigma}_a^2 = 0.0013$. We will return to models (7.14) and (7.16) for the log **`AirPassengers`** data when we compare their forecast performances.

Example 7.3 DFW Temperature Data

Dataset **`DFW.month`**, which is a ***tswge ts*** file, consists of monthly average temperatures in degrees Fahrenheit for the Dallas-Ft.Worth area from January 1900 through December 2020. This dataset was plotted in Figure 1.19 where the data are so dense that it is difficult to detect seasonal patterns. Figure 1.4 shows a subset of the DFW data from 1979 through 1986 and focuses on the seasonal pattern. It is noted there that the data have a smooth progression from summer to winter and again from winter to summer with the average temperatures in the Fall and Spring being similar. The resulting overall pattern is "sort of" sinusoidal, or pseudo-sinusoidal.

In this example we will use the temperature data from January 2000 through December 2020. The following commands create the desired dataset and plot Figures 7.22(a)–(c).

```
data(DFW.month)
DFW.2000=window(DFW.month,start=c(2000,1))
```

The seasonal pattern can be seen in the data, the sample autocorrelations have a slowly damping sinusoidal behavior with cycle length 12, and the dominant feature of the spectral density estimate is a peak at about $f=.08=1/12$.

In order to verify that seasonal factors approximating those in Table 7.7 for monthly seasonal datasets are present in overfit tables, we fit AR(15), AR(16), and AR(17) models using Burg estimates for the data in

DFW.2000. The tables are similar to each other, and the factor table for the AR(16) fit is shown in Table 7.13. Examination of Table 7.13 shows that the overfit factor table is not what was expected, and differs dramatically from those shown in Table 7.7 for monthly seasonal datasets. In the case of dataset **DFW.2000**, the factor $1-1.730B+.998B^2$ totally dominates the model fit and approximates the factor $1-1.732B+B^2$ very closely. Note that some other factors in Table 7.13 approximate factors of $1-B^{12}$. For example, the second factor, $1-.973B+.896B^2$, is an approximation of $1-B+B^2$. Recall, however, that in Tables 7.10 and 7.11 the estimates of the factors of $1-B^{12}$ in the AR overfit table for the log **AirPassengers** data are each about the same distance from the unit circle.[13] The factor $1-1.730B+.998B^2$ is associated with a frequency of $f=.084$, that is, a period of $1/.084=12$ months. Consequently, the data are not "seasonal" in the sense of Table 7.7 but instead have a cyclic nature with a period of 12. This brings to mind the comment earlier that the data have a pseudo-sinusoidal behavior. This contrasts with the **AirPassengers** data in which the "seasonal" pattern within a year was not sinusoidal but followed a fairly irregular pattern that persisted from year to year. Examination of Figures 7.22(b) and (c) show that the data behave like that of an AR(2) model with system frequency about $f=1/12$ and roots very close to the unit circle

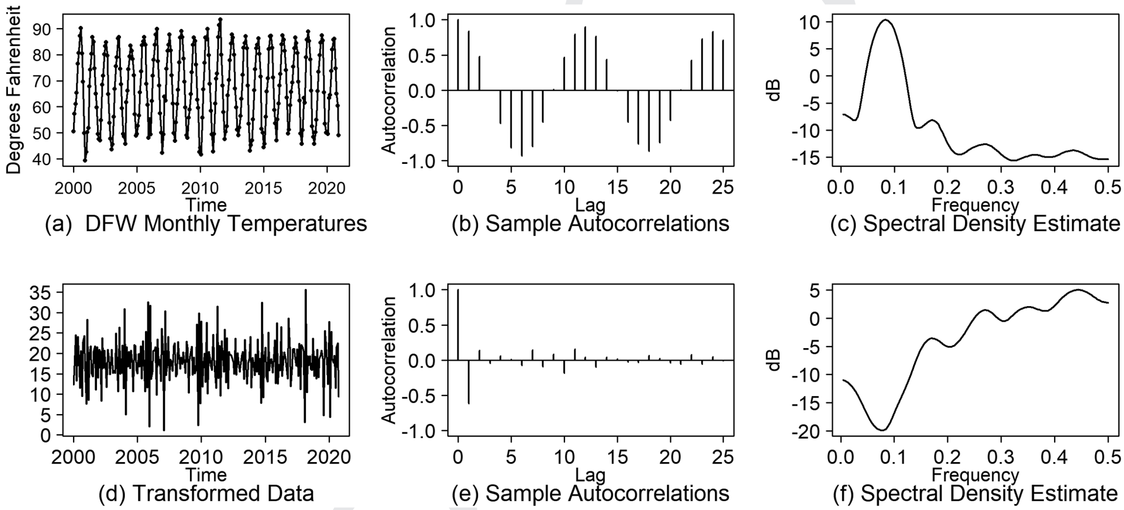


FIGURE 7.22 (a) DFW average monthly temperatures 2000–2020; (b) sample autocorrelations; (c) Parzen spectral density estimatefor data in (a), (d) data in (a) transformed by $Y_t=\left(1-1.732+B^2\right)X_t$; (e) sample autocorrelations; (f) Parzen spectral density estimate of the data in (d).

The above discussion suggests a model containing the nonstationary factor $1-1.732B+B^2$, or a nearly nonstationary factor close to it. We know from previous experience that nonstationary factors (in ARIMA and seasonal models) obscure underlying stationary behavior. Consistent with our analysis of ARIMA and stationary models, we transform the data by $Y_t=\left(1-1.732B+B^2\right)X_t$ to reveal the stationary aspects of the data.

The data are transformed and the transformed data, sample autocorrelations, and Parzen spectral density estimate are plotted in Figures 7.22(d)–(f), respectively, using the commands

```
DFW.tr12=artrans.wge(DFW.2000,phi.tr=c(1.732,-1),plot=FALSE)
plotts.sample.wge(DFW.tr12)
```

13 In Table 7.10 the factor $1+.854B$ was further from the unit circle than the other factors. We used the factoring $(1-B)(1-B^{12})$ because the other factors were consistently close to factors of $(1-B)(1-B^{12})$.

TABLE 7.13 $p = 15$ Overfit Factor Table with for DFW Monthly Temperature Data in `DFW.2000`
AR(15) Overfit

AR-FACTOR	*ROOTS*	$\lvert r\rvert^{-1}$	f_0
$1-1.728B+.998B^2$	$.866\pm.502i$	.999	.084
$1-.958B+.910B^2$	$.527\pm.907i$	.954	.166
$1-.914B$	1.094	.914	.000
$1+1.648B+.804B^2$	$-1.025\pm.440i$	.897	.436
$1-.138B+.778B^2$	$-.089\pm1.130i$	.882	.262
$1+1.063B+.760B^2$	$-.699\pm.909i$	.872	.354
$1+.816B$	−1.226	.816	.500
$1+.163B+.449B^2$	$-.181\pm1.482i$	.670	.269
$1-.660B$	1.514	.660	.000

Although the transformed data appear to be white noise, the large negative autocorrelation at lag 1 and the dip in the spectral density at about $f = .08$ suggest that modeling should continue. Using the command

```
aic5.wge(DFW.tr12,p=0:15,q=0:2)
```

AICC also selects an ARMA(11,1) while BIC selects an AR(10). Because of the consistency of choices, fit an ARMA(11,1) model for Y_t. This model is $\hat{\phi}_{11}(B)Y_t = (1-.95B)a_t$, where

$$\hat{\phi}_{11}(B) = 1+.38B-.02B^2-.30B^3-.52B^4-.56B^5-.46B^6-.36B^7+.01B^8+.30B^9+.51B^{10}+.23B^{11}.$$

Because $Y_t = (1-1.732B+B^2)X_t$ and $\bar{x} = 67.36$, it follows that the final model for the temperature data is

$$\hat{\phi}_{11}(B)(1-1.732B+B^2)(X_t-67.36) = (1-.95B)a_t, \tag{7.18}$$

and $\hat{\sigma}_a^2 = 9.41$.[14]

Key Points:

1. ***Do not*** include $1 - B^{12}$, for example, in the model simply because there is a 12th-order nonstationarity, or because the data show a period of 12, or just because you have monthly data.
   – ***Check factor tables***!
2. The factor $1-1.732B+B^2$ produces sinusoidal-type behavior of period 12. For the **`DFW.2000`** data one might also consider a cosine+noise model discussed in Section 8.2.

7.2.3 Forecasting Using Seasonal Models

In this section we discuss forecasting using the seasonal model $\boldsymbol{\phi}(B)(1-B^s)X_t = \boldsymbol{\theta}(B)a_t$. For this seasonal model, the "AR part" is the $\boldsymbol{p+s}$ order factor $\boldsymbol{\phi}(B)(1-B^s)$, and, analogous to the ARIMA case, we use the notation $\hat{\boldsymbol{\phi}}^*(B) = \hat{\boldsymbol{\phi}}(B)(1-B^s)$ to denote the AR part of the fitted seasonal model. Forecasts for seasonal models use the following formula which is analogous to (7.6) for ARIMA models:

14 The mean, 67.36, is included in model (7.19) because $1-1.732+1\neq0$, and so the mean plays a role in the model.

$$\hat{X}_{t_0}(\ell) = \sum_{j=1}^{p+s} \hat{\phi}_j^* \hat{X}_{t_0}(\ell - j) - \sum_{j=\ell}^{q} \hat{\theta}_j \hat{a}_{t_0+\ell-j}. \quad (7.19)$$

The constant term is zero in seasonal models because $1-B^s$ has a factor of $1-B$ for all $s>0$.[15] We begin our discussion of forecasting from seasonal models by considering forecasts from the model $(1-B^4)X_t = a_t$.

(1) Model (A): Forecasts for the Purely Seasonal Model $(1-B^4)X_t = a_t$

Figure 7.16(a) showed the first 24 data values from a realization of length $n=72$ from the purely seasonal model with $s=4$. The full dataset was generated using the command

```
xA4=gen.arima.wge(n=72,s=4,trmean=50,sn=52)
```

Figure 7.23(a) is a plot of the data in Figure 7.16(a) along with forecasts for the next 12 quarters (three years). By carefully observing forecasts for $t = 25, 26, 27$, and 28 it can be seen that these are exact replicas of the data values for the last four quarters observed, that is, $t = 21, 22, 23$ and 24. Notice that Figure 7.23(a) shows forecasts for the next three years, and for each of these years, the forecasts are the values observed for $t = 21, 22, 23$, and 24. This is analogous to the ARIMA(0,1,0) model for which forecasts are simply replications of the last data value.

In order to understand these forecasts, we write $(1-B^4)X_t = a_t$ as $X_t = X_{t-4} + a_t$. Using this expression, it can be seen that, for example, the value for $t=25$ is the value for $t=21$ plus a zero-mean white noise value that is as likely to be positive as it is to be negative. Consequently, it makes intuitive sense that $\hat{X}_{24}(1) = x_{21}$, that is, x_{21} is the "best" forecast for X_{25}.

Figure 7.23(b) shows the purely seasonal (monthly) data with $s=12$, shown previously in Figure 7.16(b) along with forecasts for the next year (that is, 12 months). Close examination shows that these forecasts are replicas of the sales for the last year observed.

```
xA4=gen.arima.wge(n=72,s=4,mu=50,sn=52)
xA12=gen.arima.wge(n=72,s=12,mu=50,sn=100)
par(mfrow=c(1,2))
fore.arima.wge(xA4[1:24],s=4,n.ahead=12,limits=FALSE)
abline(v=24.5,lwd=2)
fore.arima.wge(xA12[1:24],s=12,n.ahead=12,limits=FALSE)
abline(v=24.5,lwd=2)
```

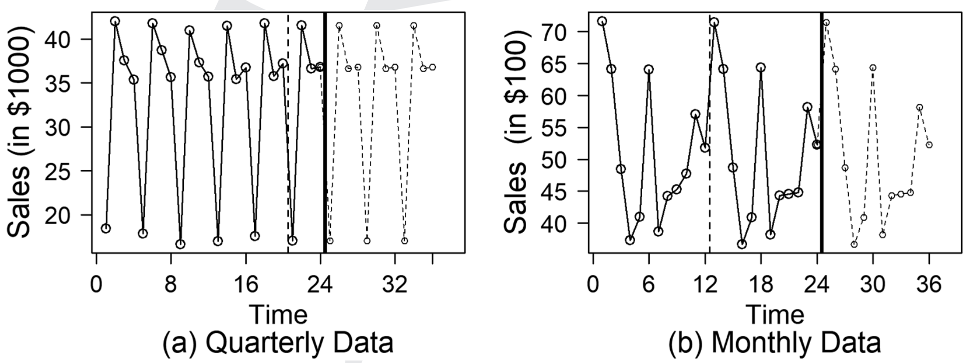


FIGURE 7.23 (a) Realization of length 24 from a seasonal model with $s=4$ along with forecasts for the next 12 quarters and (b) realization of length 24 from a seasonal model with $s=12$ along with forecasts for the next 12 months.

15 The characteristic equation $1-z^s=0$, that is $z^s=1$, always has a solution $z=1$. That is, $1-z$ is a factor of $1-z^s$.

(2) Model (B): Forecasts Using Data from Seasonal Model $\left(1-1.3\mathrm{B}+.65\mathrm{B}^2\right)\left(1-\mathrm{B}^4\right)\mathrm{X}_t=\mathrm{a}_t$

Figure 7.18(a) showed a realization of length $n=100$ from Model (B), $\left(1-1.3B+.65B^2\right)\left(1-B^4\right)X_t=a_t$, with $\sigma_a^2=1$. The data in Figure 7.18(a) were generated using the command

```
xB=gen.arima.wge(n=100,phi=c(1.3,-.65),s=4,sn=290)
```

In Section 7.2.2 the model $\left(1-1.19B+.64B^2\right)\left(1-B^4\right)X_t=a_t$,with $\hat{\sigma}_a^2=1.04$ was fit to the data. The forecasts for the next 12 quarters (three years) are calculated using the command

```
fore.xB=fore.arima.we(xB,phi=c(1.19,-.64),s=4,n.ahead=12)
```

Figure 7.24(a) shows the realization previously plotted in Figure 7.18(a). Figure 7.24(b) shows a close-up showing the last 12 data values, the first 12 forecasts, and the 95% limits. The forecasts look like exact replicas of the last four data values. In this case, however, the last four data values are 58.5, 74.6, 21.8, and 17.7, while the first four forecasts are 55.1, 73.4, 20.1, and 16.1, respectively. Thus, the AR component has modified the forecasts slightly. Also, note that the prediction limits are very "tight". Figure 7.24(c) shows Figure 7.24(b) along with the next 12 data values in the simulated series. The true values are indicated by solid squares. Notice that it is difficult to see the forecasts in this plot because they are very close to the true values. Additionally, the true values fall within the prediction limits. That is, the forecasts are quite good. The plots in Figure 7.24 were generated using the code

```
xB=gen.arima.wge(n=112,phi=c(1.3,-.65),s=4,sn=290,plot=FALSE)
t=1:112
plot(t[1:100],xB[1:100],type='l',xlim=c(0,120))
fore.xB=fore.arima.wge(xB[89:100],phi=c(1.19,-.64),s=4,n.ahead=12)
fore.xB=fore.arima.wge(xB[89:100],phi=c(1.19,-.64),s=4,n.ahead=12,limits=
FALSE)
points(t[13:24],xB[101:112],type='o')
```

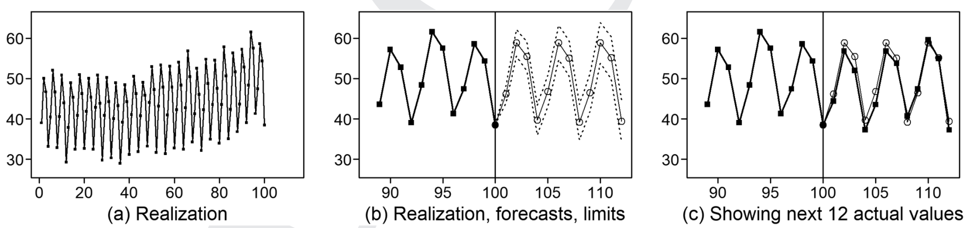


FIGURE 7.24 (a) Realization of length $n=100$ from Model (B), (b) close-up of last 12 data values in (a) along with first 12 forecasts and 95% limit lines, and (c) realization in (a) along with actual data values for $t=101-112$.

Example 7.4 Forecasting the AirPassengers Data

(a) Using the "Airline Model" in (7.16): In Example 7.2 the "Airline Model" $\phi_{13}(B)(1-B)\left(1-B^{12}\right)X_t=a_t$, where $\phi_{13}(B)$ as defined in (7.15) was fit to the log **AirPassengers** data. To simplify notation, the data will be numbered in months, 1:144. In this example forecasts will be compared with actual values of the log **AirPassengers** data using several forecast origins:

(a) Forecast origin $t_0=132$, forecast last 12 months
(b) Forecast origin $t_0=120$, forecast last 24 months
(c) Forecast origin $t_0=108$, forecast last 36 months
(d) Forecast origin $t_0=96$, forecast last 48 months

Since we fit an AR(13) model to $Y_t = (1-B)(1-B^{12})X_t$ when analyzing the entire dataset in Example 7.2, we will fit an AR(13) using data up to and including the forecast origin for each set of forecasts. The forecasts are shown in Figure 7.25, where it can be seen that for all forecast origins, the forecasts are quite good. Specifically, the seasonal behavior is forecast well and the trend in the data is forecast to continue. This is due to the fact that the Airline Model operator, $(1-B)(1-B^{12})$, has two factors of $1-B$. Thus, the trend is forecast to continue analogously to the situation for ARIMA models with $d = 2$.

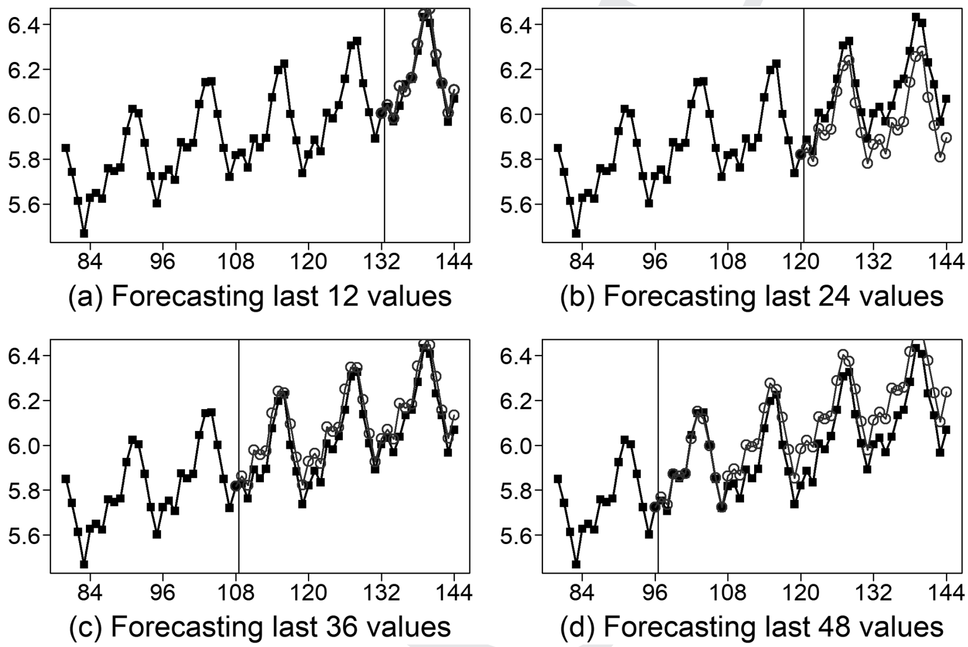


FIGURE 7.25 Comparing forecasts (open circles) with actual values (solid circles) for the last 12, 24, 36, and 48 months, respectively, for the log **AirPassengers** data using the Airline Model in (7.14).

(b) Using the Alternative Model in (7.18) for AirPassengers data:

We next consider model (7.16), which has a seasonal factor but not a second factor of $1-B$. In Figure 7.26 we compare forecasts of the last 12, 24, 36, and 48 months with actual values using the procedure described above for forecasts shown in Figure 7.25. Clearly, the upward trend is not forecast to continue, and in Figures 7.26(b)–(d) the forecasts are quite poor.

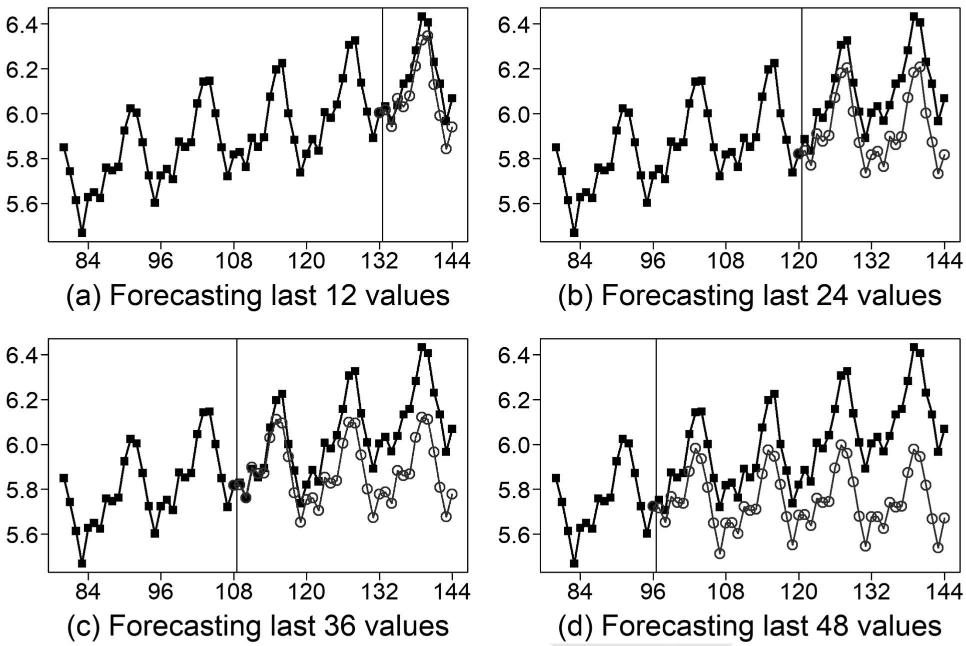


FIGURE 7.26 Comparing forecasts (open circles) with actual values (solid circles) for forecasting the last 12, 24, 36, and 48 months, respectively, for the Log `AirPassengers` data using model (7.18).

(c) Comparing the "Airline Model" to the ARIMA(13,1,0) in (7.16)

In order to compare these two models with a horizon 12, the rolling window RMSE was calculated for each model below:

```
data(AirPassengers)
airlog=log(AirPassengers)
s12=artrans.wge(airlog,phi.tr=c(rep(0,11),1),plot=FALSE)
s12d1=artrans.wge(s12,phi.tr=1,plot=FALSE)
estAIRLINE = est.ar.wge(s12d1,p=13,type='burg')

roll.win.rmse.wge(airlog, s = 12, d = 1, phi = estAIRLINE$phi, horizon = 12)

"Please Hold For a Moment, TSWGE is processing the Rolling Window RMSE with 105
windows."
"The Summary Statistics for the Rolling Window RMSE Are:"
Min. 1st Qu. Median  Mean 3rd Qu.  Max.
0.01750 0.03663 0.04530 0.05466 0.07067 0.14778
"The Rolling Window RMSE is: 0.055"

aic5.wge(s12,p=0:15,q=0:0)
# AIC selects an AR(13)
estALT = est.ar.wge(s12,p=13)

roll.win.rmse.wge(airlog, s = 12, d = 0, phi = estALT$phi,
horizon = 12)

"Please Hold For a Moment, TSWGE is processing the Rolling Window
RMSE with 106 windows."
```

QR 7.10
Rolling Window

```
"The Summary Statistics for the Rolling Window RMSE Are:"
Min. 1st Qu. Median  Mean 3rd Qu.  Max.
0.0381 0.0984 0.1165 0.1147 0.1429 0.1844
"The Rolling Window RMSE is: 0.115"
```

The "Airline Model" achieves the lower rolling window RMSE (.055) for a horizon of 12 when compared to the alternative model (.115). Table 7.14 below shows that the "Airline Model" achieves the lower rolling window RMSE for horizons 24, 26, and 48 as well thus clearly providing evidence that it is the more useful model.

TABLE 7.14 Rolling Window RMSEs for "Airline Model" and its Alternative

HORIZON	*"AIRLINE MODEL"*	*ALTERNATIVE MODEL*
12	.055	.155
24	.085	.189
36	.113	.253
48	.134	.328

Key Point:

1. Forecasts from an "Airline Model" such as (7.14) containing the factors $(1-B)$ and $(1-B^{12})$ forecast any existing trend to continue because the model has two unit roots.
2. Model (7.16) does not forecast the trend in the **`AirPassengers`** data to continue because the model contains only one unit root.

Example 7.5 Forecasting the DFW Monthly Temperature data
DFW.2000 forecasts using nonstationary model (7.18)

In Example 7.3 the nonstationary "seasonal" model

$$\hat{\phi}_{11}(B)\left(1-1.732B+B^2\right)\left(X_t-67.36\right)=\left(1-.95B\right)a_t, \tag{7.20}$$

was fit to the DFW.2000 data, where $\hat{\phi}_{11}(B)$ is given in Example 7.3 and $\hat{\sigma}_a^2=9.41$. Seasonal is in quotes because in this case, even though the data are monthly, the "seasonal" factor is $1-1.732B+B^2$ instead of $1-B^{12}$. Recall that we transformed the **`DFW.2000`** data to obtain $Y_t=\left(1-1.732B+B^2\right)X_t$ using the command

```
DFW.tr2=artrans.wge(DFW.2000,phi.tr=c(1.732,-1),plot=FALSE)
```

and then used AIC to select the model for the transformed data in **`DFW.tr2`**. AIC chose an ARMA(11,1) whose parameters are found using the command

```
est.tr2=est.arma.wge(DFW.tr2,p=11,q=1)
```

The vector **`est.tr2$phi`** contains the 11 AR coefficients of the ARMA(11,1) fit. In order to forecast using **`fore.arma.wge`** we must obtain the 13th-order AR operator obtained by multiplying the AR coefficients in **`est.tr2`** with **`seas.2=c(1.732,-1)`** using the commands

```
seas.2=c(1.732,-1)
DFW.13=mult.wge(fac1=est.tr2$phi,fac2=seas.2)
```

The resulting AR coefficients are in the vector **`DFW.13$model.coef`** while the MA coefficient is in **`est.tr2$theta`**. Figure 7.27 shows forecasts of the last 12, 36, 84, and 132 months.[16] These forecasts were obtained using the code

```
DFW.full=phi.full$model.coef
f12=fore.arma.wge(DFW.2000,phi=DFW.13$model.coef,theta=est.tr2$theta,
  lastn=TRUE,n.ahead=12,limits=FALSE)
f36=fore.arma.wge(DFW.2000,phi=DFW.13$model.coef,theta=est.tr2$theta,
  lastn=TRUE,n.ahead=36,limits=FALSE)
f84=fore.arma.wge(DFW.2000,phi=DFW.13$model.coef,theta=est.tr2$theta,
  lastn=TRUE,n.ahead=84,limits=FALSE)
f132=fore.arma.wge(DFW.2000,phi=DFW.13$model.coef,theta=est.tr2$theta,
  lastn=TRUE,n.ahead=132,limits=FALSE)
```

Examination of Figure 7.27 shows that the forecasts from forecast origins considered are very good, staying in sync with the cyclic behavior and predicting peaks and troughs quite well.

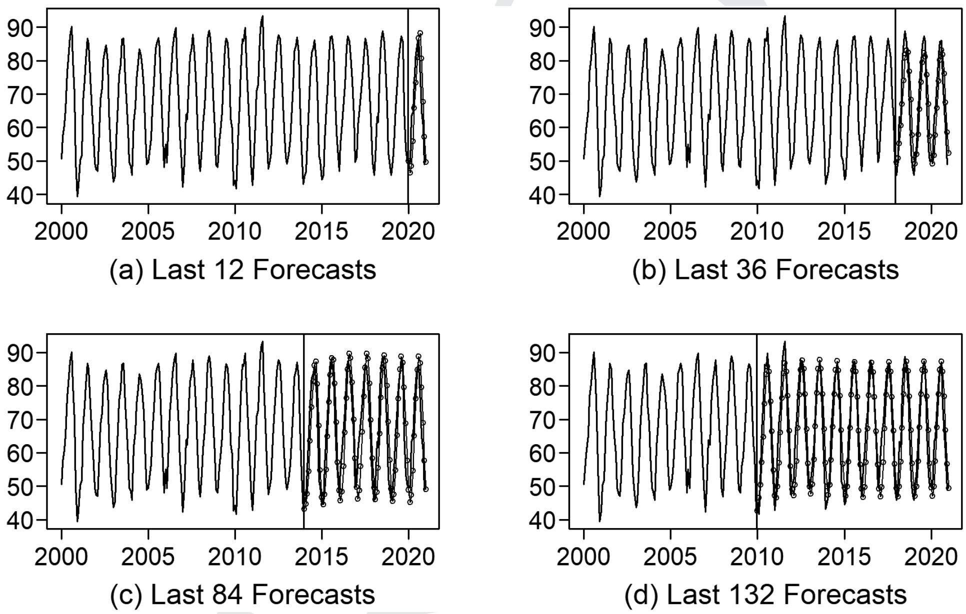


FIGURE 7.27 DFW monthly temperature with forecasts for last 12, 36, 84, and 132 months.

7.3 ARCH AND GARCH MODELS

Figure 7.28 is a plot of the Dow Jones daily rates of return from 1970 through 2020.[17] This is an interesting set of data. The rates of return clearly look more variable (volatile) during some time periods than they do in others. Isolated extremes on Black Monday, October 19, 1987 and the early stages of COVID pandemic awareness in early March, 2020 are visible. Also, the Great Recession from 2007–2009 was a period of continuing volatility.

16 Notice that **`DFW13$model.coef`** has a complex pair of roots on the unit circle. Even though this is not a stationary model, the function **`fore.arma.wge`** produces the correct forecasts.

17 We calculated rate of return as 100*(current value-previous value)/ *p* Previous value.

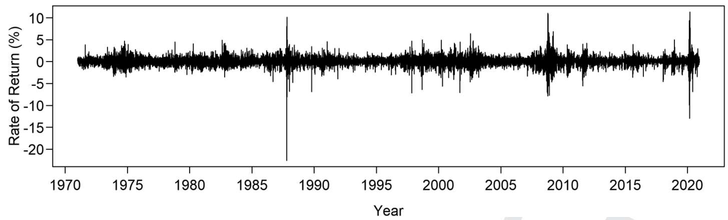


FIGURE 7.28 Daily Dow Jones rates of return from 1971 through 2020.

A key feature of the data is that the variance changes over time. The variance doesn't change gradually in time but rather experiences sudden periods of increased and decreased variability (or volatility). This type of behavior is called *conditional heteroskedasticity*. It is important to note that because the variance changes with time, this type of data should be modeled as a nonstationary process.

Taking a "closer look" at the data, in Figure 7.29(a) we plot the daily rates of return for the first 1,000 trading days beginning in July 1999. A unique feature of these data is that the sample autocorrelations in Figure 7.29(b) behave like sample autocorrelations of white noise. So, this is a type of "noise" that is uncorrelated yet nonstationary (the variance changes with time). Economists have encountered data such as these over the years, yet standard models such as ARMA and ARIMA are not designed to model this type of behavior. Figure 7.29(c) is a plot of the squared rates of return, and this plot really accentuates the spikes. Interestingly, the sample autocorrelations of the squared data in Figure 7.29(d) behave somewhat like the autocorrelations of an AR process. That is, the squared data are *correlated*. In a seminal paper, Engle (1982) introduced the *autoregressive conditional heteroskedastic* (ARCH) model to explain this behavior.[18] Based on data such as those in Figure 7.28, Engle's idea was to model the squared data as an AR process.

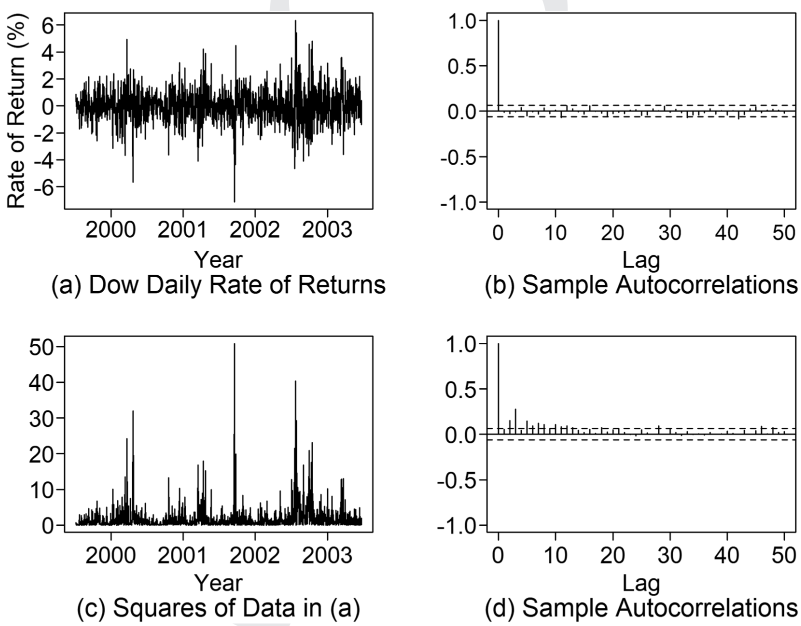


FIGURE 7.29 (a) Dow daily rates of return for first 1,000 days following July 1, 1999, (b) sample autocorrelations of the data in (a), (c) Data in (a) squared, and (d) sample autocorrelations of the squared data in (c).

18 The ARCH model idea was such a breakthrough that Robert Engle and his research partner Clive Granger were awarded the Nobel Prize in Economics in 2003.

7.3.1 ARCH(1) Model

The simplest ARCH model is an ARCH(1) model. Specifically, a time series u_t is said to satisfy a 1st-order ARCH model if

$$u_t = \varepsilon_t \sqrt{\alpha_0 + \alpha_1 u_{t-1}^2}, \tag{7.21}$$

where α_0 and α_1 are constants, ε_t is zero-mean normally distributed white noise with unit variance, and ε_t is uncorrelated with $u_{t-j}^2,\ \ j > 0$.

Note that because ε_t has mean zero, so does u_t.Squaring both sides of (7.21) gives

$$\begin{aligned} u_t^2 &= \varepsilon_t^2\left(\alpha_0 + \alpha_1 u_{t-1}^2\right) \\ &= \alpha_0 + \alpha_1 u_{t-1}^2 + w_t, \end{aligned} \tag{7.22}$$

where w_t is a zero-mean uncorrelated, zero-mean noise process.[19] See Appendix 7B. Thus, (7.22) shows that u_t^2 satisfies an AR(1) type model.

Notes:

(1) The behavior of the rates of return data is that there are dramatic changes in variability of the data, u_t, in different time periods. In Figure 7.29(d) we see the autocorrelations of the squared residuals that Engle modeled using an AR model. However, the AR model in (7.22) is an AR model for u_t^2, not for the *variance*. The two concepts are intimately related. Note that the mean of $u_t = 0$,and consequently

$$\begin{aligned} \operatorname{Var}\left(u_t\right) &= E\left[\left(u_t - \mu_{u_t}\right)^2\right] \text{ (definition of variance)} \\ &= E\left[u_t^2\right]. \end{aligned}$$

That is, the *variance of u_t is the expected value of u_t^2*.

(2) In the ARCH setting, the quantity under the radical in (7.21), $\alpha_0 + \alpha_1 u_{t-1}^2$, is the conditional variance of u_t given u_{t-1}.That is, standing at time $t-1$, and having already observed u_{t-1}, the conditional variance of u_t at time t is given by

$$\sigma_{t|t-1}^2 = \alpha_0 + \alpha_1 u_{t-1}^2. \tag{7.23}$$

(3) *Interpretation of (7.23) using the rates of return data as an example:* Regions of high volatility are periods in which the behavior of the rates of return from day to day is less predictable (more uncertain) than it is in low volatility periods. In the ARCH(1) model, $\alpha_1 \geq 0$, so if the value u_{t-1} is "large", then $\sigma_{t|t-1}^2 = \alpha_0 + \alpha_1 u_{t-1}^2$ is "large". This implies that, at time $t-1$, there is a "high" level of uncertainty about the rates of return on day t.

19 Note that we haven't used the notation a_t in these definitions. For example, ε_t is restricted to have unit variance and to be normally distributed. Neither of these conditions are imposed on the white noise in this book that we have denoted by a_t. Also, the variance of w_t depends on time, but a_t has constant variance. See Definition 3.14.

Key Points:

1. If u_{t-1} is "larger than usual" then there is increased *uncertainty* about the upcoming value of u_t.
2. Point (1) does not indicate that u_t itself will be large. It may be large, small, or "in the middle".
3. Equation (7.23) says that the *uncertainty* about the upcoming value, u_t, is large.[20]

Generating realizations from an ARCH(1) process. The ***tswge*** function **`gen.arch.wge`** *generates* realizations from an ARCH model. The following commands generate the ARCH(1) realizations in Figure 7.30. The sample autocorrelations can be found using standard techniques.

```
set.seed(11)
par(mfrow=c(4,1))
u1=gen.arch.wge(500,alpha0= 1,alpha=0)
u2=gen.arch.wge(500,alpha0=.6,alpha=.4)
u3=gen.arch.wge(500,alpha0=.3,alpha=.7)
u4=gen.arch.wge(500,alpha0=.1,alpha=.9)
```

When $\alpha_1 = 0$, u_t^2 in (7.22) is a "regular" white noise process. This is consistent with Figure 7.30(a) which is a realization in which $\alpha_1 = 0$. Note that the volatility increases as α_1 goes from .4 to .7 to .9 in Figures 7.30(c), (e), and (g), respectively. Note also that the sample autocorrelations in all cases behave like sample autocorrelations of white noise.

20 Suppose we have two random variables, X and Y, both with mean $\mu = 50$, but $\sigma_X^2 = 1$ and $\sigma_Y^2 = 100$. Suppose for this example we assume that they are both normally distributed and that we are about to observe a new observation, x and y, from each of the random variables. We know there is about a 95% chance that the new observation in each case will be within two standard deviations of the mean. That is, for random variable X, we are about 95% confident that the new observation will be between 48 and 52. However, for random variable Y, we are 95% confident that the new observation will be between $50 \pm 2(10)$, that is, between 30 and 70. *Obvious lesson:* There is more uncertainty about the *yet to be observed value* of y than there is about the upcoming value of x. So, a large value of u_{t-1} suggests that, at time $t-1$, there is a lot of uncertainty concerning the value of u_t.

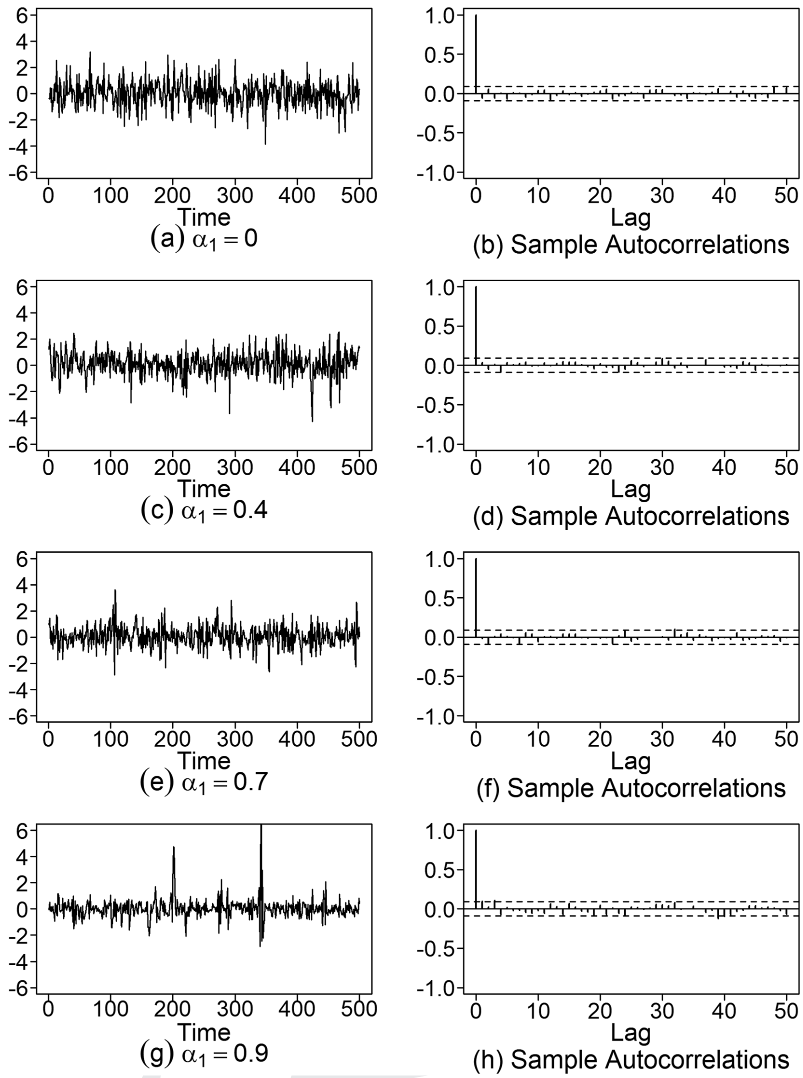
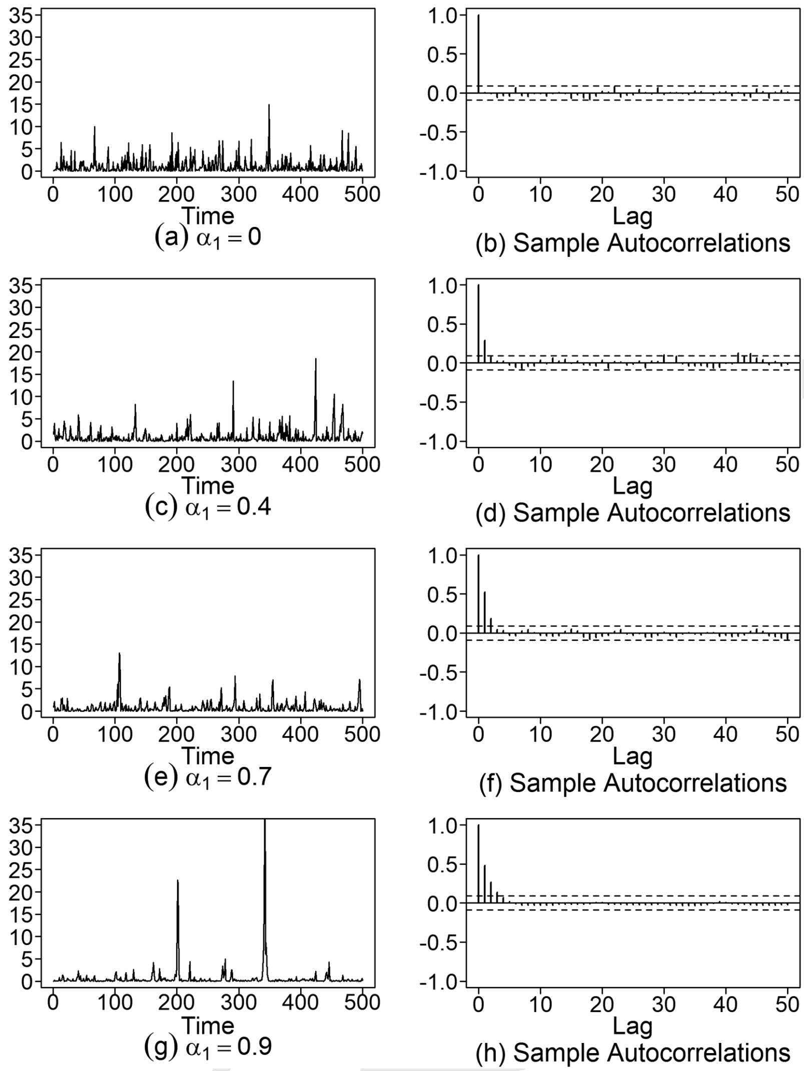


FIGURE 7.30 Realizations and associated sample autocorrelations from ARCH(1) models for various values of α_1.

Figures 7.31 (a), (c), (e), and (g) show the squares of the values in Figures 7.30 (a), (c), (e), and (g), respectively. In these plots we again see that the volatile behavior increases as α_1 increases. Also note in Figure 7.31(b) that the sample autocorrelations of the squared data for the $\alpha_1 = 0$ case (true white noise) are uncorrelated. However, as α_1 increases, the sample autocorrelations exhibit AR(1)-like autocorrelation behavior. The ARCH realizations have the appearance and the behavior associated with the DOW rates of return data implying that the ARCH model seems to be appropriate for modeling these types of data.



QR 7.11
ARCH(1)

FIGURE 7.31 Squared values of the realizations in Figure 7.30 and associated sample autocorrelations.

7.3.2 The ARCH(p) and GARCH(p,q) Processes

Notice that the periods of high volatility in the ARCH(1) realizations tended to be “spiked” in nature. The ARCH(1) model says how the size of the data value on one day predicts the volatility the next day. However, the Great Recession and the early 2000s (Gulf War early stages?) were periods of prolonged increased volatility. The ARCH(p) model is a generalization of the ARCH(1) that models longer stretches of volatility and is defined by

$$u_t = \varepsilon_t \sqrt{\alpha_0 + \alpha_1 u_{t-1}^2 + \cdots + \alpha_p u_{t-p}^2}, \tag{7.24}$$

where α_0, $\alpha_1, \ldots, \alpha_p$ are constants, ε_t is zero-mean normally distributed white noise with unit variance, and ε_t is uncorrelated with u_{t-j}^2, $j > 0$. Using the strategy for the ARCH(1) model, it can be shown that

$$u_t^2 = \alpha_0 + \alpha_1 u_{t-1}^2 + \cdots + \alpha_p u_{t-p}^2 + w_t, \tag{7.25}$$

where w_t is a zero-mean white noise process, and thus u_t^2 has an AR-type behavior. Analogous to (7.23) the conditional variance of u_t given data at times up to and including $t-1$, is given by[21]

$$\sigma_{t|t-1}^2 = \alpha_0 + \alpha_1 u_{t-1}^2 + \cdots + \alpha_p u_{t-p}^2. \tag{7.26}$$

In this case, "large" values of $\alpha_0 + \alpha_1 u_{t-1}^2 + \cdots + \alpha_p u_{t-p}^2$ imply that "standing at time $t-1$ there is "high" uncertainty about the value of u_t at time t.

An important generalization (analogous to the ARMA generalization of an AR model) is the GARCH(p,q) process (see Bollerslev, 1986) which is defined by

$$u_t = \varepsilon_t \sqrt{\alpha_0 + \alpha_1 u_{t-1}^2 + \cdots + \alpha_p u_{t-p}^2 + \beta_1 \sigma_{t-1|t-2}^2 + \cdots + \beta_q \sigma_{t-q|t-q-1}^2}. \tag{7.27}$$

Analogous to (7.26), it follows that

$$\sigma_{t|t-1}^2 = \alpha_0 + \alpha_1 u_{t-1}^2 + \cdots + \alpha_p u_{t-p}^2 + \beta_1 \sigma_{t-1|t-2}^2 + \cdots + \beta_q \sigma_{t-q|t-q-1}^2, \tag{7.28}$$

where ε_t is a zero-mean normal white noise process uncorrelated with u_{t-j}^2 for $j > 0$. That is, $\sigma_{t|t-1}^2$ is a linear combination of previous observed values u_t^2 and previous conditional variances. As with the ARMA model, a GARCH model can often be obtained that has fewer parameters than an appropriately fitting ARCH model.

Key Points:

1. It is possible for the residuals from an ARMA or ARIMA fit to appear to be white and pass the standard diagnostic tests, but for the squared residuals to not be white.
2. It is good practice to check the squared residuals, especially if conditional heteroskedasticity is suspected.

(1) Generating Realizations from ARCH/GARCH Models

Both the ARCH(p) with $p > 1$ and the GARCH(p,q) can model volatility that continues over a period of time. To better understand the ARCH(p), $p > 1$ and GARCH(p,q)behaviors, in Figure 7.32(a–c) we show realizations from ARCH(2), ARCH(3), and GARCH(1,1) models, respectively. The models are:

21 For the case of an ARCH(1), the interpretation we gave of $\sigma_{t|t-1}^2$ was that it was the conditional variance of u_t given data at time $t-1$. We continue to use the same notation in the ARCH(p) case, but note that it is more accurately defined as the conditional variance *given data up to and including* $t-1$.

(a) ARCH(2) with $\alpha_0 = .1, \alpha_1 = .5, \alpha_2 = .4$
(b) ARCH(3) with $\alpha_0 = .1$ and $\alpha_1, \alpha_2, \alpha_3$ given by .4,.3, and .2, respectively.
(c) GARCH(1,1) with $\alpha_0 = .1, \alpha_1 = .4$, and $\beta_1 = .5$

In each case (especially for the ARCH(3) and GARCH(1,1) cases) the realizations show periods of more sustained volatility than did the ARCH(1) realizations in Figure 7.30. The realizations in Figure 7.32 were generated using the commands

```
u1=gen.arch.wge(1000,alpha0=.1,alpha=c(.5,.4),sn=17)
u2=gen.arch.wge(1000,alpha0=.1,alpha=c(.4,.3,.2),sn=32)
u3=gen.garch.wge(1000,alpha0=.1,alpha=.4,beta=.5,sn=141)
```

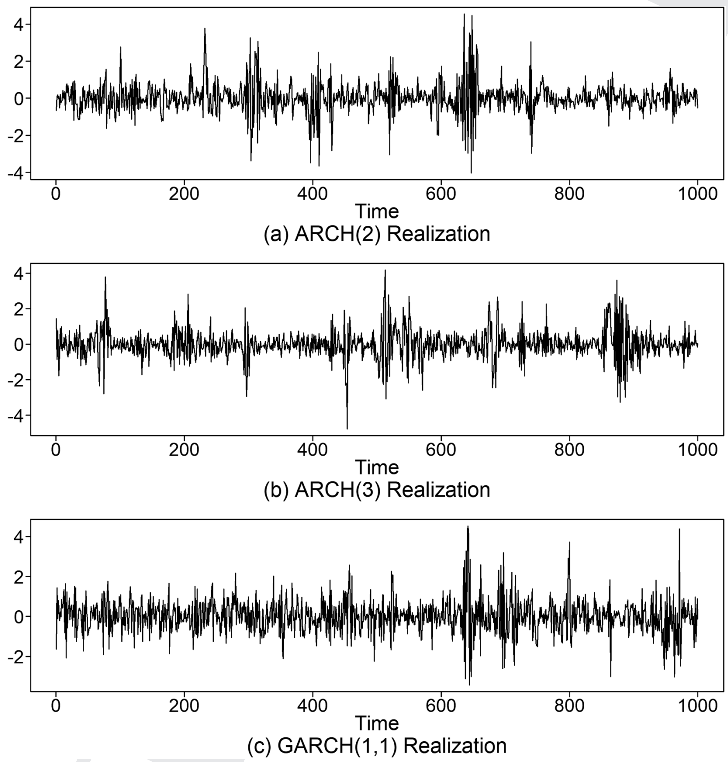


FIGURE 7.32 Realizations from an ARCH(2), ARCH(3), and GARCH(1,1) model, respectively.

7.3.3 Assessing the Appropriateness of an ARCH/GARCH Fit to a Set of Data

The question arises concerning how to evaluate the suitability of an ARCH/GARCH fit to a set of data. The most common way is to examine the residuals. What are the residuals? The answer to this question is not as obvious as it is for, say, an ARMA model. In the following we discuss the calculation and assessment of residuals from a GARCH fit to a set of data. Appropriate simplifications are in order if the model is ARCH.

Consider the defining relation (7.27) for the GARCH model. After a GARCH model has been fit, then

$$\hat{\varepsilon}_t \sqrt{\alpha_0 + \hat{\alpha}_1 u_{t-1}^2 + \cdots + \hat{\alpha}_p u_{t-p}^2 + \hat{\beta}_1 \sigma_{t-1|t-2}^2 + \cdots + \hat{\beta}_q \sigma_{t-q|t-q-1}^2}$$

is the model-based estimate of u_t. Consequently, the residual associated with the fit at time t is

$$\hat{\varepsilon}_t = u_t / \sqrt{\alpha_0 + \hat{\alpha}_1 u_{t-1}^2 + \cdots + \hat{\alpha}_p u_{t-p}^2 + \hat{\beta}_1 \sigma_{t-1|t-2}^2 + \cdots + \hat{\beta}_q \sigma_{t-q|t-q-1}^2}.$$

Under the assumptions of the model, ε_t is zero-mean normally distributed white noise with unit variance. Thus, a check for model appropriateness is to check the residuals, $\hat{\varepsilon}_t$, to see if they appear to be white noise and to check for normality. This can be done by checking the plot and sample autocorrelations of the residuals versus their 95% limit lines.

The assumption of approximate normality can be assessed using measures such as the Shapiro-Wilk test (see Shapiro and Wilk, 1965) which tests the null hypothesis that the data are normally distributed. We will return to the use of this test in Chapter 9. Note that if the residuals, $\hat{\varepsilon}_t$, are approximately uncorrelated normal random variables then the squared residuals should also be uncorrelated.[22] We will illustrate the concepts discussed above in the following sections.

7.3.4 Fitting ARCH/GARCH Models to Simulated Data

The CRAN package ***tseries*** has functions designed to analyze data using ARCH and GARCH models. Specifically, the ***tseries*** function **`garch`** can be used to fit an ARCH/GARCH model to a set of data. In the following we use the **`garch`** command to estimate the coefficients of the realizations **`u1, u2,`** and **`u3`** above.

(a) **`u1:`** ARCH(2) model in which the parameters α_0, α_1, and α_2 are .1, .5, and .4, respectively.

tseries commands:[23]

```
u1.out=garch(u1,order=(0,2))
summary(u1.out)
```

Output from **`garch`**:

```
Coefficient(s):
    Estimate  Std. Error  t value  Pr(>|t|)
a0  0.11932   0.01655     7.208    5.66e-13  ***
a1  0.47259   0.08721     5.419    6.00e-08  ***
a2  0.30559   0.07296     4.189    2.81e-05  ***

Diagnostic Tests:
       Jarque Bera Test
data: Residuals
X-squared = 0.053895, df = 2, p-value = 0.9734

       Box-Ljung test
data: Squared.Residuals
X-squared = 0.38239, df = 1, p-value = 0.5363
```

QR 7.12 ARCH(p) and GARCH(p,q)

The output shows that the estimates of all parameters were highly significant and close to the true values. The Jarque-Bera test (J. Bera, 1980, 1987) is a test to check the residuals for skewness and kurtosis consistent with normality and in this case failed to reject normality. Additionally, the

22 Three results from mathematical statistics are being used here: (1) If random variables are independent, then they are uncorrelated. (2) If normal random variables are uncorrelated, then they are independent. (3) Functions of independent random variables are independent (and thus uncorrelated). See Wackerly, Mendenhall, and Scheaffer (2008).

23 Using the notation adopted in this book, the **`order`** parameter in the **`garch`** function is the ordered pair (q, p). Be careful!

Ljung-Box test (or sometimes called the Box-Ljung test) is designed to test the null hypothesis of white noise. This test will be introduced in Section 9.1. In the output above, the Ljung-Box test on the squared residuals has a *p*-value of .536 thus not rejecting white noise for the squared residuals. This is consistent with normal residuals in which case if normally distributed residuals are uncorrelated then so are the squared residuals. (See Footnote 22.)

We additionally ran the Base R **`shapiro.test`** function on the residuals and obtained a *p*-value of .957. The residuals are contained in the vector **`u1.out$res`** but care must be taken because the first two value are `NA`.

(b) **`u2:`** ARCH(3) model in which the parameters $\alpha_0, \alpha_1, \alpha_2$, and α_3 are .1, .4, .3, and .2, respectively.

tseries commands:

```
u2.out=garch(u2,order=c(0,3))
u2.out
```

Output from **`garch:`**

```
Coefficient(s):
    Estimate  Std. Error  t value  Pr(>|t|)
a0  0.08632   0.01085     7.955    1.78e-15  ***
a1  0.41018   0.05893     6.960    3.40e-12  ***
a2  0.35192   0.05996     5.869    4.37e-09  ***
a3  0.19469   0.04684     4.157    3.23e-05  ***

Diagnostic Tests:
        Jarque Bera Test
data: Residuals
X-squared = 0.70133, df = 2, p-value = 0.7042

        Box-Ljung test
data: Squared.Residuals
X-squared = 0.021352, df = 1, p-value = 0.8838
```

The output shows that the estimates of all parameters were highly significant and close to the true values. The Jarque-Bera test failed to reject normality of the residuals. The Ljung-Box test on the squared residuals does not reject white noise. The Shapiro-Wilk test has a *p*-value of .587 and thus does not reject white noise.

(c) **`u3:`** GARCH(1,1) model in which the parameters α_0, α_1, and β_1 are .1, .4, and .5, respectively.

tseries commands:

```
u3.out=garch(u3,order=c(1,1))
summary(u3.out)
```

Output:

```
Coefficient(s):
    Estimate  Std.Error  t value   Pr(>|t|)
a0  0.13799   0.02431    5.676     1.38e-08  ***
a1  0.43508   0.05764    7.548     4.42e-14  ***
b1  0.39874   0.05949    6.703     2.04e-11  ***

Diagnostic Tests:
        Jarque Bera Test
data: Residuals
```

```
X-squared = 0.50614, df = 2, p-value = 0.7764

        Box-Ljung test
data: Squared.Residuals
X-squared = 0.0038326, df = 1, p-value = 0.9506
```

The output shows that the estimates of all parameters were highly significant and close to the true values. The Jarque Bera test fails to reject normality of the residuals. The Ljung-Box test on the squared residuals does not reject white noise. The Shapiro-Wilk *p*-value was .941 which does not reject normality. Again, the fit seems to be good.

Summary:
The model fits all seem to be good with diagnostics "checking out". However, in these cases we did know the true model orders and the ε_ts were normally distributed in the simulations. We close out the discussion of ARCH/GARCH by fitting a GARCH model on the Dow daily rates of return data in Figure 7.29.

7.3.5 Modeling Daily Rates of Return Data

The Dow daily rates of return data in Figure 7.28 shows the volatile behavior for which the ARCH or GARCH models were designed to model. The ***tswge*** dataset **`rate`** contains daily rates of return from 1971 through 2020. The data in Figure 7.29 were for the first 1,000 trading days beginning in July 1999. The commands below retrieve and analyze the rate data for these 1,000 days. We examined several models and the two that whitened the residuals and squared residuals the best were GARCH((1,1) and GARCH(2,1) with a slight edge given to GARCH(2,1). The following are the results for the GARCH(2,1) fit.

```
data(rate)
rate.2000= rate[7200:8199]
rate.2000.out=garch(rate.2000,order=c(1,2))
summary(rate.2000.out)
```

Output:

```
Coefficient(s):

      Estimate   Std. Error  t value  Pr(>|t|)
a0    0.08490    0.02628     3.230    0.001237   **
a1    0.01565    0.02617     0.598    0.549917
a2    0.09371    0.02666     3.516    0.000439   ***
b1    0.84668    0.02457     34.461   < 2e-16    ***

Diagnostic Tests:
       Jarque Bera Test
data: Residuals
X-squared = 58.251, df = 2, p-value = 2.244e-13

       Box-Ljung test
data: Squared.Residuals
X-squared = 0.0015587, df = 1, p-value = 0.9685
```

An interesting feature of the GARCH(2,1) fit is that the data value two days ahead makes the most difference concerning the rate on day *t*.However, the main factor predicting volatility is the conditional variance $\sigma^2_{t-1|t-2}$. The Jarque-Bera test "doesn't like" the skewness and kurtosis of the residuals *at all* and strongly rejects normality. The residuals and their sample autocorrelations are shown in Figures 7.33(a) and (b). The Ljung-Box test on the squared residuals did not reject white noise (which is an indicator of normally distributed noise). However, the Shapiro-Wilk test strongly rejected normality with a *p*-value less than .0001. All things considered, the GARCH(2,1) model seems to be our best (but not perfect) choice. Recall Box's quote about helpful models.

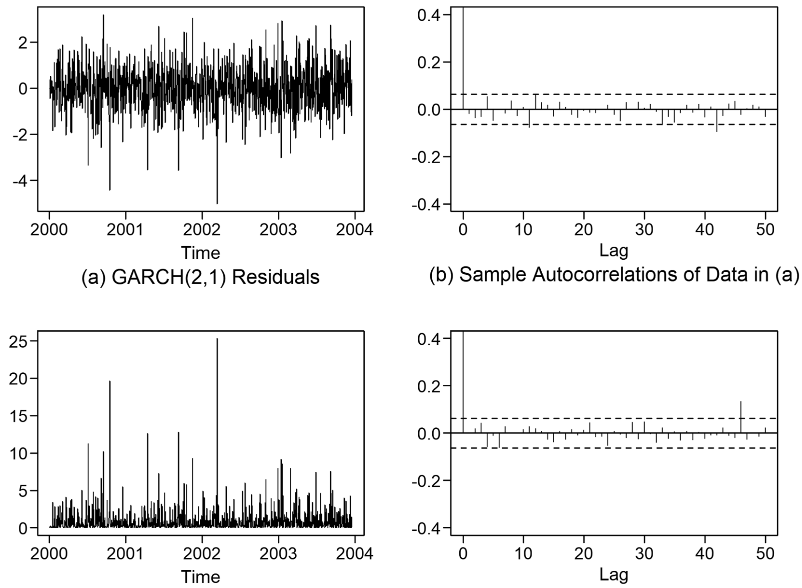


FIGURE 7.33 (a) Residuals from the GARCH(1,1) fit to the Dow rates of return data for 1,000 trading days beginning in July 1999 and (b) sample autocorrelations of the data in (a).

7.4 CONCLUDING REMARKS

In Chapter 6, we proposed techniques for analyzing stationary time series data. However, not all data satisfy the conditions of stationarity. In this chapter we analyzed data for which the roots of the fitted autoregressive characteristic equation are on the unit circle. These data violate stationarity (see Violation 4 in the chapter introduction). We introduced the ARIMA and seasonal ARIMA models to analyze such data. In Section 7.3 we discussed the ARCH/GARCH models for analyzing a class of time series data for which the variance changes erratically over time (another violation of stationarity).

Many economic datasets have a type of "wandering" behavior not consistent with stationarity. We introduced the autoregressive integrated moving average (ARIMA(p,d,q)) model for analyzing this type of nonstationary data. If $d = 0$ then an ARIMA(p,d,q) model simplifies to a stationary ARMA(p,q) model. If $d > 0$, then the ARIMA(p,d,q) model is said to contain one or more "unit roots". We discussed the assessment of whether a model containing a unit root should be used. We showed how to fit ARIMA(p,d,q) to data and how to use these models for forecasting future values. Model identification, model estimation, and forecasting techniques were illustrated using examples.

Seasonal models were introduced for the modeling of nonstationary data that contain repetitive seasonal (for example, monthly or quarterly) patterns. We introduced the seasonal ARIMA models and presented the method of overfitting. Overfitting is a method of identifying seasonal patterns by fitting "high-order" AR models to the data and examining the resulting factor tables. We discussed methods for estimating the parameters of a seasonal ARIMA model and forecasting future values. These techniques were illustrated using examples.

Finally, ARCH/GARCH models were presented as effective models for data which exhibit suddenly erratic change in variance behavior, as is frequently seen in important economic and financial data. We discussed the properties of these models and presented techniques for fitting ARCH/GARCH model to this type of data. Again, these techniques were illustrated using examples.

In the next chapter, we will discuss nonstationary models that contain a deterministic component.

APPENDIX 7A

TSWGE FUNCTIONS

(a) **`artrans.wge(x,phi.tr,lag.max,plottr)`** performs a transformation $y_t = \phi(B)x_t$ where the coefficients of $\phi(B)$ are given in phi.tr.

`x` = original realization.
`phi.tr` = vector of transforming coefficients
`lag.max` = maximum lag for calculated sample autocorrelations (for original and a transformed data) (default = 25)
`plottr` = logical variable.
If **`plottr = TRUE`** (default) then the function plots the original and transformed realizations and associated sample autocorrelations.

(b) **`fore.arima.wge(x,phi,theta,d,s,n.ahead,lastn,plot,alpha,limits)`** forecasts **`n.ahead`** steps ahead for stationary ARIMA and seasonal models. The forecasts and forecast limits can optionally be plotted. Forecasts can be calculated and plotted **`n.ahead`** steps beyond the end of the series, or you can forecast the last **`n.ahead`** data values (if **`lastn=TRUE`**). Note: The components of phi and/or theta may be obtained using **`mult.wge`** if the models are given in factored form.

Note: The ARIMA model can be a given/known model, but the more common usage will be to fit an ARMA(p,q) model to the data and base the forecasts on the fitted model

`x` is a vector containing the time series realization $x_t, t = 1,\ldots,n$
`phi` is a vector of autoregressive coefficients (default=0)
`theta` is a vector of moving average coefficients (default=0)
`d` is the order of the difference operator, (1 ? B)d, in the model (default = 0)
s is the seasonal order (default = 0)
`n.ahead` specifies the number of steps ahead you want to forecast (default=2)
`lastn` is a logical variable (default=**`FALSE`**)
If **`lastn=FALSE`** (default) then the forecasts are for $x(n+1), x(n+2),\ldots, x(n+n.ahead)$ where n is the length of the realization.
If **`lastn=TRUE`**, then the program forecasts the last **`n.ahead`** values in the realization.
`plot` is a logical variable. (default=**`TRUE`**)
If **`plot=TRUE`** then the forecasts will be plotted along with the realization.
If **`plot=FALSE`** then no plots are output.
`alpha` specifies the significance levels of the prediction limits. (default=.05)
`limits` is a logical variable. (default= **`TRUE`**)
If **`limits=TRUE`** then the (1-**`alpha`**)×100% prediction limits will be plotted along with the forecasts.

(d) **gen.arima.wge(n,phi,theta,d,s,lambda,vara,sn)** is a function that generates a realization of length **n** from a given AR, MA, ARMA, ARIMA, or seasonal model.
n is the realization length
phi is a vector of AR parameters (of the stationary part of the model) (default = 0)
theta is a vector of MA parameters (using signs as in this text) (default = 0)
d specifies the number of factors of $(1-B)$ in the model.
s specifies the order of the seasonal difference, $1-B^s$.
vara is the white noise variance (default = 1)
plot = TRUE (default) produces a plot of the realization generated
sn determines the seed used in the simulation (see Note (4) below).

Notes:

1. This function uses a call to the base R function **arima.sim** which uses the same signs as this text for the AR parameters but opposite signs for MA parameters. The appropriate adjustments are made within **gen.arima.sim** so that input vectors **phi** and **theta** for **gen.arima.wge** contain parameters using the signs as in this text. However, if you use **arima.sim** directly (which has options not employed in **gen.arima.sim**), then you must remember that the signs needed for the MA parameters are opposite to those in **gen.arima.wge**.
2. **sn = 0** (default) produces results based on a randomly selected seed. If you want to reproduce the same realization on subsequent runs of **gen.arima.wge** then set **sn** to the same positive integer value on each run.

Example: The command

```
x = gen.arima.wge(n = 200,phi = c(1.6,-.9),theta = .9,d = 2, vara = 1)
```

generates and plots a realization from the model $\left(1-1.6B+0.9B^2\right)\left(1-B\right)^2 Xt = \left(1-0.9B\right)a_t$ and stores it in the vector **x**.

APPENDIX 7B

The steps to obtain the equality in (7.22) are as follows:

$$
\begin{aligned}
u_t^2 &= \varepsilon_t^2\left(\alpha_0+\alpha_1 u_{t-1}^2\right)\\
&= \varepsilon_t^2\left(\alpha_0+\alpha_1 u_{t-1}^2\right)+\left(\alpha_0+\alpha_1 u_{t-1}^2\right)-\left(\alpha_0+\alpha_1 u_{t-1}^2\right)\\
&= \alpha_0+\alpha_1 u_{t-1}^2+\left(\varepsilon_t^2-1\right)\left(\alpha_0+\alpha_1 u_{t-1}^2\right)\\
&= \alpha_0+\alpha_1 u_{t-1}^2+w_t
\end{aligned}
$$

Recall that $E\left(\varepsilon_t\right)=\mu_{\varepsilon_t}=0$ and $Var\left(\varepsilon_t\right)=1.$ Thus we have

$$
\begin{aligned}
Var\left(\varepsilon_t\right) &= E\left[\left(\varepsilon_t-\mu_{\varepsilon_t}\right)^2\right]\\
&= E\left[\varepsilon_t^2\right]\\
&= 1,
\end{aligned}
$$

so, $E\left[\left(\varepsilon_t^2 - 1\right)\right] = 0$.Also, by assumption, ε_t and u_{t-1} are uncorrelated, so

$$\begin{aligned} E\left[w_t\right] &= E\left[\left(\varepsilon_t^2 - 1\right)\left(\alpha_0 + \alpha_1 \mathrm{u}_{t-1}^2\right)\right] \\ &= E\left[\left(\varepsilon_t^2 - 1\right)\right] E\left(\alpha_0 + \alpha_1 u_{t-1}^2\right)] \\ &= 0. \end{aligned}$$

So, $u_t^2 = \alpha_0 + \alpha_1 u_{t-1}^2 + w_t$,
where w_t, is a zero-mean process.

PROBLEMS

7.1 Generate a realization of length $n = 1000$ from each of the following models:
(a) $\left(1 - B\right)(X_t - 300) = a_t$.
(b) $\left(1 - .999B\right)\left(X_t - 300\right) = a_t$.
Explain the role of $\mu = 300$ in each realization.

7.2 Generate realizations of length $n = 200$ from each of the following models:
(a) $\left(1 + 1.2B + .9B^2\right)\left(1 - B\right)X_t = a_t$.
(b) $\left(1 + 1.2B + .9B^2\right)\left(1 - B\right)^2 X_t = a_t$.
(c) $\left(1 + 1.2B + .9B^2\right)\left(1 - B^2\right)X_t = a_t$.
(d) $\left(1 + 1.2B + .9B^2\right)\left(1 - B^{12}\right)X_t = a_t$.
Use the techniques described in this chapter to fit models to the datasets and specify your final models.

7.3 Using the monthly **`cement`** sales data in **`tswge`**, overfit a 13th, 15th, and 17th-order AR using Burg estimates.
(a) Explain whether you believe a $1 - B^{12}$ factor should be included in your model.
(b) Should $\left(1 - B\right)\left(1 - B^{12}\right)$ be included in the model? Explain.
(c) Whether you believe the model should include the seasonal factors $1 - B^{12}$ or $\left(1 - B\right)\left(1 - B^{12}\right)$, go ahead and model the data using the seasonal factor you like best.

7.4 (a) Repeat the analysis in Example 7.1(b) using the logarithm of the US population data in ***`tswge ts`*** object **`uspop`**. How do the residuals of the "log" model compare with those in Example 7.1(b).
(b) As in (a) analyze the logarithms of the **`dow1985`** data.

7.5 In this problem you will analyze the average monthly temperature in degrees Fahrenheit for Pennsylvania from January 1990 through December 2004. The data are in ***`tswge`*** file **`patemp`**.
(a) plot the data, sample autocorrelations, and Parzen spectral density estimate
(b) Use Burg estimates to overfit using value of $p = 14, 16,$ *and* 18.
(c) Do you see evidence of a $1 - B^{12}$? Explain.
(d) Fit a final model for these data and explain your findings.
(e) Use this model to forecast the temperatures for the next 5 years.
(f) Assess the accuracy of forecasts by finding the RMSE of your forecasts for the last 5 years of data.

7.6 (a) The Tesla data in **`tswge`** dataset **`tesla`** contains daily data from January 1, 2020 through April 30, 2021.
(a) Plot the data, sample autocorrelations, and spectral density.
(b) Fit a model to the data. Does your model include a unit root? Explain why or why not.

- (c) Forecast the Tesla stock data through the end of 2021.
- (d) Find the actual data values for this time frame.
- (e) Compare your forecasts with the actual values.
- (d) Find the RMSE of your forecasts and discuss your results.

7.7 Using the dow1985 dataset:
- (a) Difference the data using **`artrans.wge`** and plot the differenced data.
- (b) Compute the monthly rates of return for the dow1985 data and plot these rates of return.
- (c) Discuss the relationship between the plots in (a) and (b).

7.8 The ***tswge*** dataset **`rates`** contain the DOW daily rates of return for the year's daily rate of returns for the years 1971 through 2020. In Section 7.3.5 we fit a GARCH model to the first 1,000 days beginning in July 1999. Repeat the analysis used in that section to analyze:
- (a) the first 1,000 days beginning in 1971.
- (b) the first 500 days beginning in 1990.
- (c) the 1500 days beginning in 1980.

7.9 In Example 7.4 an "Airline Model" (ARIMA(13,1,0) s = 12) was compared to an Alternative model (ARIMA(13,0,0) s= 12). The "Airline Model" was fit by taking the first difference and then taking a seasonal difference with s = 12. At that point, we used **`aic5.wge`** to select the model orders for the stationary portion of the model. AIC selected $p = 13$.
- (a) Fit a third model using **`aic5.wge`** to select the model orders for the stationary portion of the model and then using the p and q that have the second smallest AIC.
- (b) Evaluate this model against the other two using the rolling window RMSE for a horizon of 12.
- (c) Evaluate this model using the rolling window RMSE for horizons of 24, 36, and 48 as well.