Chapter 17

Regression-Based Forecasting

A popular forecasting tool is based on multiple linear regression models, using suitable predictors to capture trend and/or seasonality. In this chapter, we show how a linear regression model can be set up to capture a time series with a trend and/or seasonality. The model, which is estimated from the data, can then produce future forecasts by inserting the relevant predictor information into the estimated regression equation. We describe different types of common trends (linear, exponential, polynomial), as well as two types of seasonality (additive and multiplicative). Next, we show how a regression model can be used to quantify the correlation between neighboring values in a time series (called autocorrelation). This type of model, called an autoregressive model, is useful for improving forecast precision by making use of the information contained in the autocorrelation (beyond trend and seasonality). It is also useful for evaluating the predictability of a series, by evaluating whether the series is a "random walk." The various steps of fitting linear regression and autoregressive models, using them to generate forecasts, and assessing their predictive accuracy, are illustrated using the Amtrak ridership series.

17.1 A Model with Trend[1]

Linear Trend

To create a linear regression model that captures a time series with a global linear trend, the outcome variable (Y) is set as the time series values or some function of it, and the predictor (X) is set as a time index. Let us consider a simple

[1]This and subsequent sections in this chapter copyright © 2017 Datastats, LLC, and Galit Shmueli. Used by permission.

Data Mining for Business Analytics: Concepts, Techniques, and Applications in R, First Edition.
Galit Shmueli, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, and Kenneth C. Lichtendahl, Jr.
© 2018 John Wiley & Sons, Inc. Published 2018 by John Wiley & Sons, Inc.

code for creating Figure 17.1

```
library(forecast)
Amtrak.data <- read.csv("Amtrak.csv")

# create time series
ridership.ts <- ts(Amtrak.data$Ridership, start = c(1991,1),
                   end = c(2004,3), freq = 12)

# produce linear trend model
ridership.lm <- tslm(ridership.ts ~ trend)

# plot the series
plot(ridership.ts, xlab = "Time", ylab = "Ridership", ylim = c(1300,2300),
     bty = "l")
lines(ridership.lm$fitted, lwd = 2)
```

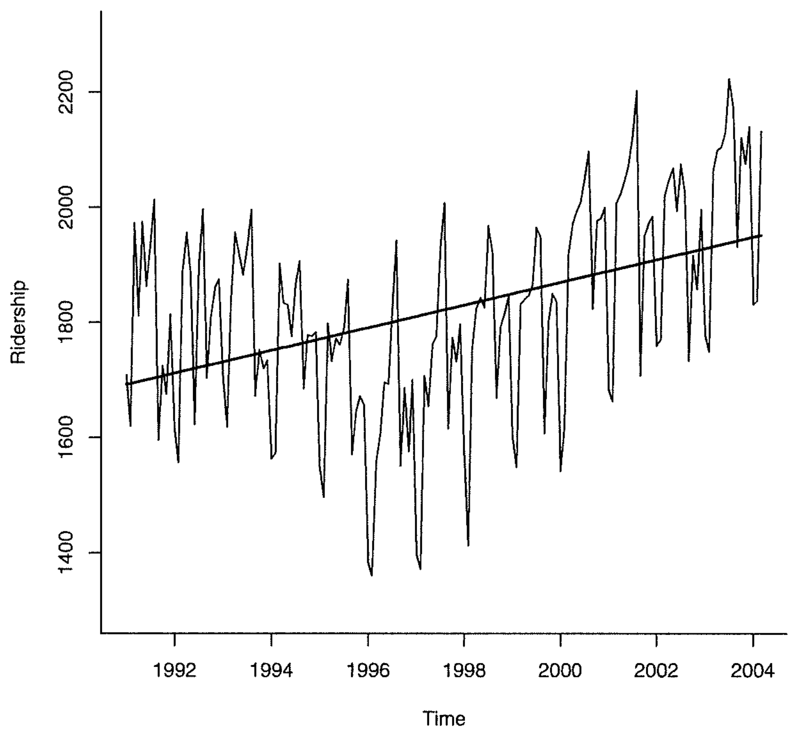

FIGURE 17.1 A LINEAR TREND FIT TO AMTRAK RIDERSHIP

example: fitting a linear trend to the Amtrak ridership data. This type of trend is shown in Figure 17.1.

From the time plot, it is obvious that the global trend is not linear. However, we use this example to illustrate how a linear trend is fitted, and later we consider more appropriate models for this series.

To obtain a linear relationship between Ridership and Time, we set the output variable Y as the Amtrak Ridership and create a new variable that is a time index $t = 1, 2, 3, \ldots$ This time index is then used as a single predictor in the regression model:

$$Y_t = \beta_0 + \beta_1 t + \epsilon,$$

where Y_t is the Ridership at period t and ϵ is the standard noise term in a linear regression. Thus, we are modeling three of the four time series components: level (β_0), trend (β_1), and noise (ϵ). Seasonality is not modeled. A snapshot of the two corresponding columns (Y and t) are shown in Table 17.1.

TABLE 17.1 OUTCOME VARIABLE (MIDDLE) AND PREDICTOR VARIABLE (RIGHT) USED TO FIT A LINEAR TREND

Month	Ridership (Y_t)	t
Jan 91	1709	1
Feb 91	1621	2
Mar 91	1973	3
Apr 91	1812	4
May 91	1975	5
Jun 91	1862	6
Jul 91	1940	7
Aug 91	2013	8
Sep 91	1596	9
Oct 91	1725	10
Nov 91	1676	11
Dec 91	1814	12
Jan 92	1615	13
Feb 92	1557	14

After partitioning the data into training and validation sets, the next step is to fit a linear regression model to the training set, with t as the single predictor (function *tslm()* relies on *ts()* which automatically creates t and calls it *trend*). Applying this to the Amtrak ridership data (with a validation set consisting of the last 12 months) results in the estimated model shown in Figure 17.2. The actual and fitted values and the residuals (or forecast errors) are shown in the two time plots.

Table 17.2 contains a report of the estimated coefficients. Note that examining only the estimated coefficients and their statistical significance can be misleading! In this example, they would indicate that the linear fit is reasonable, although it is obvious from the time plots that the trend is not linear. An inadequate trend shape is easiest to detect by examining the time plot of the residuals.

code for creating Figure 17.2

```
# fit linear trend model to training set and create forecasts
train.lm <- tslm(train.ts ~ trend)
train.lm.pred <- forecast(train.lm, h = nValid, level = 0)

par(mfrow = c(2, 1))
plot(train.lm.pred, ylim = c(1300, 2600),  ylab = "Ridership", xlab = "Time",
     bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "", flty = 2)
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(train.lm.pred$fitted, lwd = 2, col = "blue")
lines(valid.ts)
plot(train.lm.pred$residuals, ylim = c(-420, 500),  ylab = "Forecast Errors",
    xlab = "Time", bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "")
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(valid.ts - train.lm.pred$mean, lwd = 1)
```

Code for data partition is given in Figure 16.4

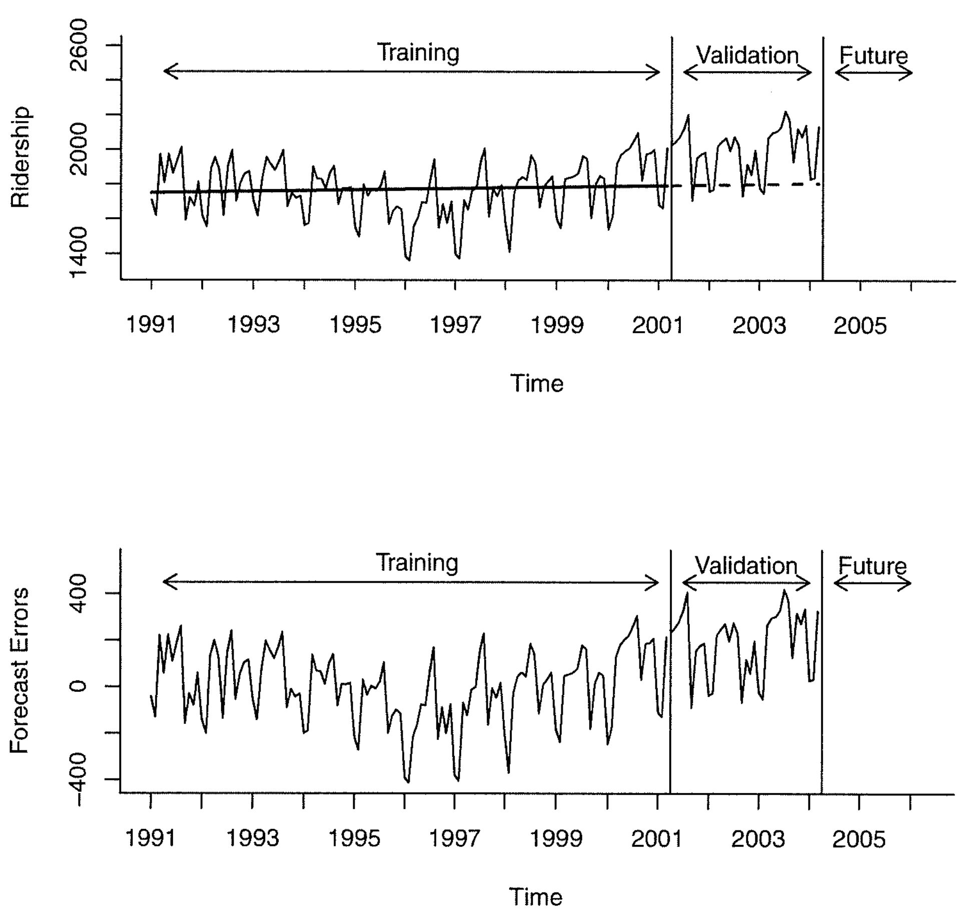

FIGURE 17.2 A LINEAR TREND FIT TO AMTRAK RIDERSHIP IN THE TRAINING PERIOD AND FORECASTED IN THE VALIDATION PERIOD

TABLE 17.2 SUMMARY OF OUTPUT FROM A LINEAR REGRESSION MODEL APPLIED TO THE AMTRAK RIDERSHIP DATA IN THE TRAINING PERIOD

```
> summary(train.lm)

Call: lm(formula = formula, data = "train.ts", na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-411.29 -114.02   16.06  129.28  306.35

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1750.3595    29.0729  60.206   <2e-16 ***
trend          0.3514     0.4069   0.864     0.39
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 160.2 on 121 degrees of freedom
Multiple R-squared:  0.006125,       Adjusted R-squared:  -0.002089
F-statistic: 0.7456 on 1 and 121 DF,  p-value: 0.3896
```

LINEAR REGRESSION FOR TIME SERIES IN R

Fitting a regression model to time series (and using it to generate forecasts) can be done in R using the function *tslm()* in the `forecast` package.

Exponential Trend

Several alternative trend shapes are useful and easy to fit via a linear regression model. One such shape is an exponential trend. An exponential trend implies a multiplicative increase/decrease of the series over time ($Y_t = ce^{\beta_1 t+\epsilon}$). To fit an exponential trend, simply replace the outcome variable Y with $\log Y$ (where log is the natural logarithm), and fit a linear regression ($\log Y_t = \beta_0 + \beta_1 t + \epsilon$). In the Amtrak example, for instance, we would fit a linear regression of log(Ridership) on the index variable t. Exponential trends are popular in sales data, where they reflect percentage growth. In R, fitting exponential trend is done by setting argument *lambda = 0* in function *tslm()*.[2]

Note: As in the general case of linear regression, when comparing the predictive accuracy of models that have a different output variable, such as a linear

[2]Argument *lambda* is used to apply the Box-Cox transformation to the values of the time series: $(y^\lambda - 1)/\lambda$ if $\lambda \neq 0$. When $\lambda = 0$, the transformation is defined as $\log(y)$. When $\lambda = 1$, the series is not transformed (except for the subtraction of 1 from each value), so the model has a linear trend.

trend model (with Y) and an exponential trend model (with $\log Y$), it is essential to compare forecasts or forecast errors on the same scale. In R, when using an exponential trend model, the forecasts of $\log Y$ are made and then automatically converted back to the original scale. An example is shown in Figure 17.3, where an exponential trend is fit to the Amtrak ridership data.

R code for creating Figure 17.3

```
# fit exponential trend using tslm() with argument lambda = 0
train.lm.expo.trend <- tslm(train.ts ~ trend, lambda = 0)
train.lm.expo.trend.pred <- forecast(train.lm.expo.trend, h = nValid, level = 0)

# fit linear trend using tslm() with argument lambda = 1 (no transform of y)
train.lm.linear.trend <- tslm(train.ts ~ trend, lambda = 1)
train.lm.linear.trend.pred <- forecast(train.lm.linear.trend, h = nValid, level = 0)

plot(train.lm.expo.trend.pred, ylim = c(1300, 2600),  ylab = "Ridership",
 xlab = "Time", bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "", flty = 2)
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(train.lm.expo.trend.pred$fitted, lwd = 2, col = "blue")  # Added in 6-5
lines(train.lm.linear.trend.pred$fitted, lwd = 2, col = "black", lty = 3)
lines(train.lm.linear.trend.pred$mean, lwd = 2, col = "black", lty = 3)
lines(valid.ts)
```

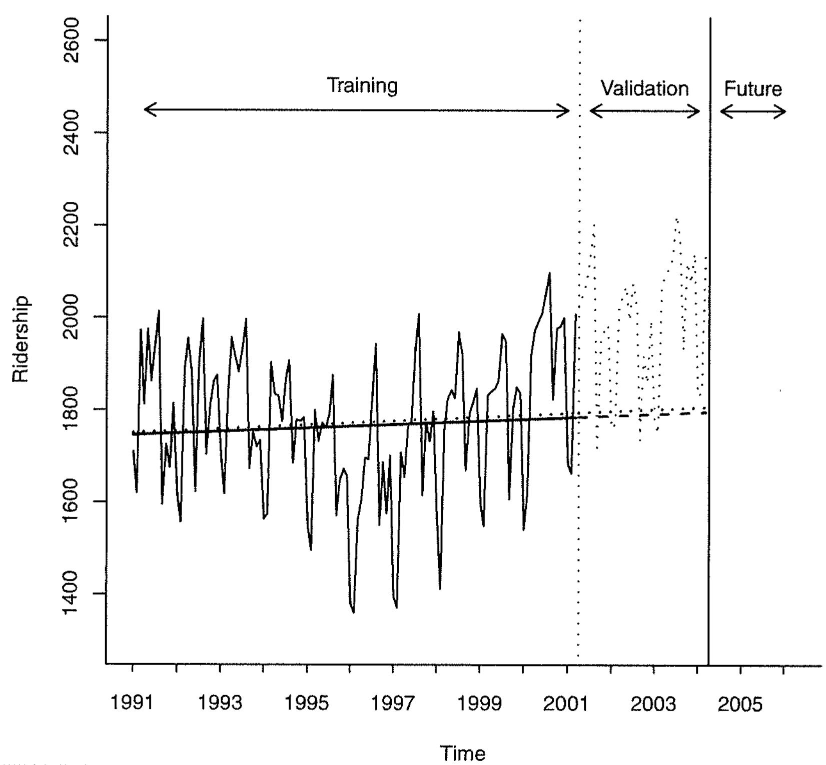

FIGURE 17.3 EXPONENTIAL (AND LINEAR) TREND USED TO FORECAST AMTRAK RIDERSHIP

Polynomial Trend

Another non-linear trend shape that is easy to fit via linear regression is a polynomial trend, and in particular, a quadratic relationship of the form $Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon$. This is done by creating an additional predictor t^2 (the square of t), and fitting a multiple linear regression with the two predictors t and t^2. In R, we fit a quadratic trend using function *I()*, which treats an object "as is" (Figure 17.4). For the Amtrak ridership data, we have already seen a U-shaped trend in the data. We therefore fit a quadratic model. The fitted and residual charts are shown in Figure 17.4. We conclude from these plots that this shape adequately captures the trend. The forecast errors are now devoid of trend and exhibit only seasonality.

In general, any type of trend shape can be fit as long as it has a mathematical representation. However, the underlying assumption is that this shape is applicable throughout the period of data that we have and also during the period that we are going to forecast. Do not choose an overly complex shape. Although it will fit the training data well, it will in fact be overfitting them. To avoid overfitting, always examine the validation performance and refrain from choosing overly complex trend patterns.

17.2 A Model with Seasonality

A seasonal pattern in a time series means that observations that fall in some seasons have consistently higher or lower values than those that fall in other seasons. Examples are day-of-week patterns, monthly patterns, and quarterly patterns. The Amtrak ridership monthly time series, as can be seen in the time plot, exhibits strong monthly seasonality (with highest traffic during summer months).

Seasonality is captured in a regression model by creating a new categorical variable that denotes the season for each value. This categorical variable is then turned into dummies, which in turn are included as predictors in the regression model. To illustrate this, we created a new "Season" column for the Amtrak ridership data, as shown in Table 17.3. Then, to include the Season categorical variable as a predictor in a regression model for Y (Ridership), we turn it into dummies (for $m = 12$ seasons we create 11 dummies, which are binary variables that take on the value 1 if the record falls in that particular season, and 0 otherwise[3]).

In R, function *tslm()* uses *ts()* which automatically creates the categorical Season column (called *season*) and converts it into dummy variables.

[3]We use only m-1 dummies because information about the $m-1$ seasons is sufficient. If all $m-1$ variables are zero, then the season must be the mth season. Including the mth dummy causes redundant information and multicollinearity.

R code for creating Figure 17.4

```
# fit quadratic trend using function I(), which treats an object "as is".
train.lm.poly.trend <- tslm(train.ts ~ trend + I(trend^2))
summary(train.lm.poly.trend)
train.lm.poly.trend.pred <- forecast(train.lm.poly.trend, h = nValid, level = 0)

par(mfrow = c(2,1))
plot(train.lm.poly.trend.pred, ylim = c(1300, 2600),  ylab = "Ridership",
 xlab = "Time", bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "", flty = 2)
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(train.lm.poly.trend$fitted, lwd = 2)
lines(valid.ts)

plot(train.lm.poly.trend$residuals, ylim = c(-400, 550),  ylab = "Forecast Errors",
 xlab = "Time", bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "")
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(valid.ts - train.lm.poly.trend.pred$mean, lwd = 1)
```

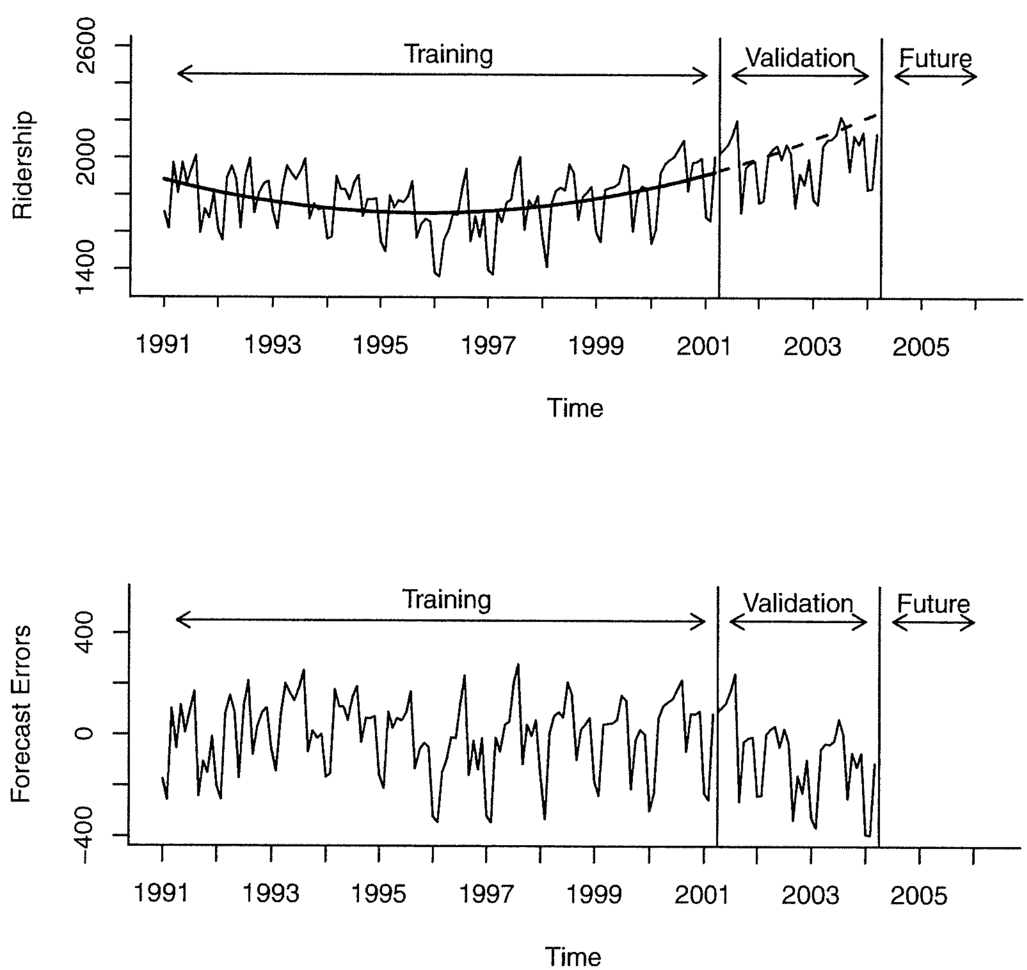

FIGURE 17.4 QUADRATIC TREND MODEL USED TO FORECAST AMTRAK RIDERSHIP. PLOTS OF FITTED, FORECASTED, AND ACTUAL VALUES (TOP) AND FORECAST ERRORS (BOTTOM)

TABLE 17.3 NEW CATEGORICAL VARIABLE (RIGHT) TO BE USED (VIA DUMMIES) AS PREDICTOR(S) IN A LINEAR REGRESSION MODEL

Month	Ridership	Season
Jan 91	1709	Jan
Feb 91	1621	Feb
Mar 91	1973	Mar
Apr 91	1812	Apr
May 91	1975	May
Jun 91	1862	Jun
Jul 91	1940	Jul
Aug 91	2013	Aug
Sep 91	1596	Sep
Oct 91	1725	Oct
Nov 91	1676	Nov
Dec 91	1814	Dec
Jan 92	1615	Jan
Feb 92	1557	Feb
Mar 92	1891	Mar
Apr 92	1956	Apr
May 92	1885	May

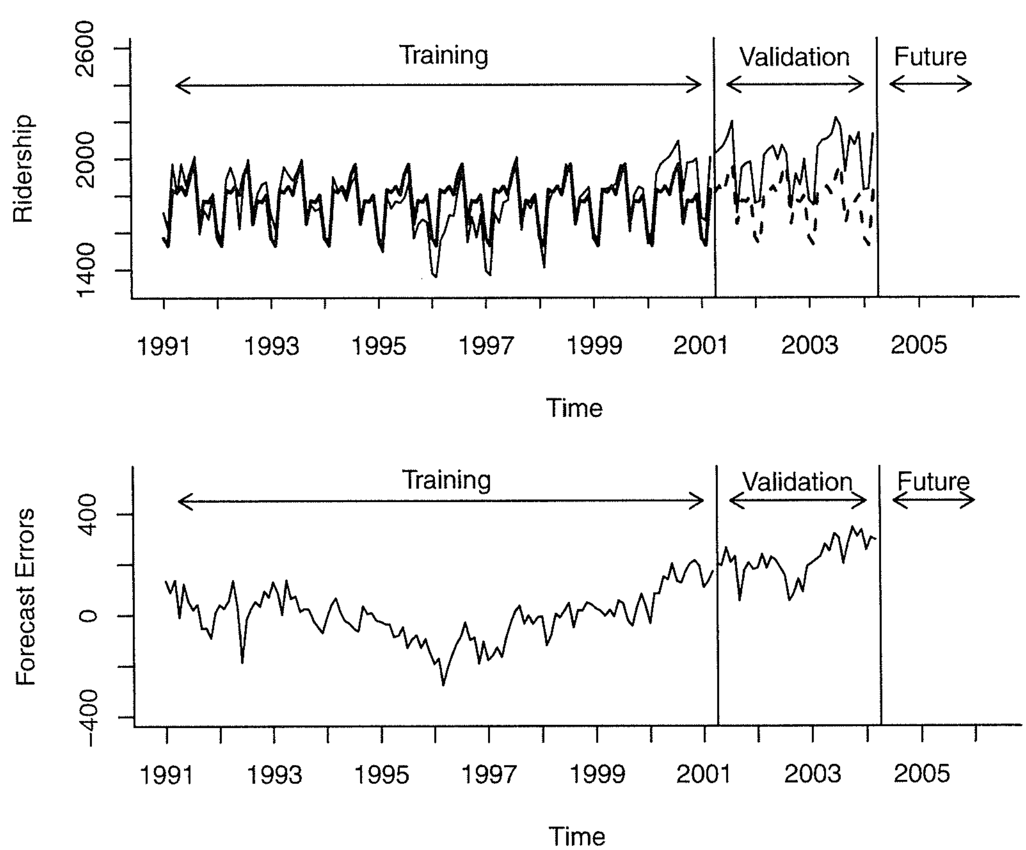

FIGURE 17.5 REGRESSION MODEL WITH SEASONALITY APPLIED TO THE AMTRAK RIDERSHIP (TOP) AND ITS FORECAST ERRORS (BOTTOM)

After partitioning the data into training and validation sets (see Section 16.5), we fit the regression model to the training data. The fitted series and the residuals from this model are shown in Figure 17.5. The model appears to capture the seasonality in the data. However, since we have not included a trend component in the model (as shown in Section 17.1), the fitted values do not capture the existing trend. Therefore, the residuals, which are the difference between the actual and the fitted values, clearly display the remaining U-shaped trend.

When seasonality is added as described above (create categorical seasonal variable, then create dummies from it, then regress on Y), it captures *additive seasonality*. This means that the average value of Y in a certain season is a fixed amount more or less than that in another season. Table 17.4 shows the output of a linear regression fit to Ridership (Y) with seasonality. For example, in the

TABLE 17.4 SUMMARY OF OUTPUT FROM FITTING ADDITIVE SEASONALITY TO THE AMTRAK RIDERSHIP DATA IN THE TRAINING PERIOD

```
> # include season as a predictor in tslm(). Here it creates 11 dummies,
> # one for each month except for the first season, January.
> train.lm.season <- tslm(train.ts ~ season)
> summary(train.lm.season)

Call:
lm(formula = formula, data = "train.ts", na.action = na.exclude)

Residuals:
     Min       1Q   Median       3Q      Max
-276.165  -52.934    5.868   54.544  215.081

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1573.97      30.58  51.475  < 2e-16 ***
season2       -42.93      43.24  -0.993   0.3230
season3       260.77      43.24   6.030 2.19e-08 ***
season4       245.09      44.31   5.531 2.14e-07 ***
season5       278.22      44.31   6.279 6.81e-09 ***
season6       233.46      44.31   5.269 6.82e-07 ***
season7       345.33      44.31   7.793 3.79e-12 ***
season8       396.66      44.31   8.952 9.19e-15 ***
season9        75.76      44.31   1.710   0.0901 .
season10      200.61      44.31   4.527 1.51e-05 ***
season11      192.36      44.31   4.341 3.14e-05 ***
season12      230.42      44.31   5.200 9.18e-07 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 101.4 on 111 degrees of freedom
Multiple R-squared:  0.6348,        Adjusted R-squared:  0.5986
F-statistic: 17.54 on 11 and 111 DF,  p-value: < 2.2e-16
```

Amtrak ridership, the coefficient for season8 (396.66) indicates that the average number of passengers in August is higher by 396.66 thousand passengers than the average in January (the reference category). Using regression models, we can also capture *multiplicative seasonality*, where values in a certain season are on average, higher or lower by a percentage amount compared to another season. To fit multiplicative seasonality, we use the same model as above, except that we use $\log(Y)$ as the outcome variable. In R, this is achieved by setting *lambda=0* in the *tslm()* function.

17.3 A Model with Trend and Seasonality

Finally, we can create models that capture both trend and seasonality by including predictors of both types. For example, from our exploration of the Amtrak Ridership data, it appears that a quadratic trend and monthly seasonality are both warranted. We therefore fit a model to the training data with 13 predictors: 11 dummies for month, and t and t^2 for trend. The fit and output from this final model are shown in Figure 17.6 and Table 17.5. If we are satisfied with

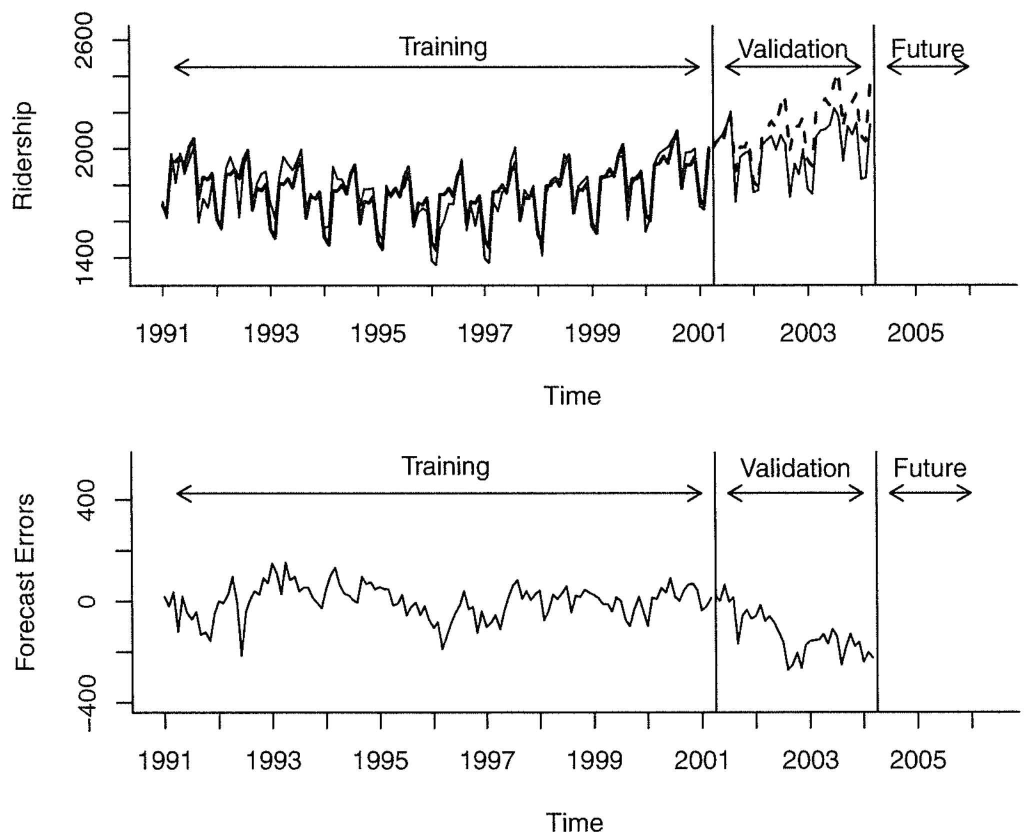

FIGURE 17.6 REGRESSION MODEL WITH TREND AND SEASONALITY APPLIED TO AMTRAK RIDERSHIP (TOP) AND ITS FORECAST ERRORS (BOTTOM)

TABLE 17.5 SUMMARY OF OUTPUT FROM FITTING TREND AND SEASONALITY TO AMTRAK RIDERSHIP IN THE TRAINING PERIOD

```
> train.lm.trend.season <- tslm(train.ts ~ trend + I(trend^2) + season)
> summary(train.lm.trend.season)

Call:
tslm(formula = train.ts ~ trend + I(trend^2) + season)

Residuals:
    Min      1Q  Median      3Q     Max
-213.77  -39.36    9.71   42.42  152.19

Coefficients:
             Estimate Std. Error t value           Pr(>|t|)
(Intercept) 1696.9794    27.6752   61.32 < 0.0000000000000002 ***
trend         -7.1559     0.7293   -9.81 < 0.0000000000000002 ***
I(trend^2)     0.0607     0.0057   10.66 < 0.0000000000000002 ***
season2      -43.2458    30.2407   -1.43               0.1556
season3      260.0149    30.2423    8.60   0.00000000000006604 ***
season4      260.6175    31.0210    8.40   0.00000000000018264 ***
season5      293.7966    31.0202    9.47   0.00000000000000069 ***
season6      248.9615    31.0199    8.03   0.00000000000126033 ***
season7      360.6340    31.0202   11.63 < 0.0000000000000002 ***
season8      411.6513    31.0209   13.27 < 0.0000000000000002 ***
season9       90.3162    31.0223    2.91               0.0044 **
season10     214.6037    31.0241    6.92   0.00000000032920793 ***
season11     205.6711    31.0265    6.63   0.00000000133918009 ***
season12     242.9294    31.0295    7.83   0.00000000000344281 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 70.9 on 109 degrees of freedom
Multiple R-squared:  0.825,       Adjusted R-squared:  0.804
F-statistic: 39.4 on 13 and 109 DF,  p-value: <0.0000000000000002
```

this model after evaluating its predictive performance on the validation data and comparing it against alternatives, we would re-fit it to the entire un-partitioned series. This re-fitted model can then be used to generate k-step-ahead forecasts (denoted by F_{t+k}) by plugging in the appropriate month and index terms.

17.4 AUTOCORRELATION AND ARIMA MODELS

When we use linear regression for time series forecasting, we are able to account for patterns such as trend and seasonality. However, ordinary regression models do not account for dependence between values in different periods, which in cross-sectional data is assumed to be absent. Yet, in the time series context, values in neighboring periods tend to be correlated. Such correlation, called *autocorrelation*, is informative and can help in improving forecasts. If we know that a high value tends to be followed by high values (positive autocorrelation), then we can use that to adjust forecasts. We will now discuss how to compute the autocorrelation of a series, and how best to utilize the information for improving forecasts.

Computing Autocorrelation

Correlation between values of a time series in neighboring periods is called *autocorrelation*, because it describes a relationship between the series and itself. To compute autocorrelation, we compute the correlation between the series and a lagged version of the series. A *lagged series* is a "copy" of the original series which is moved forward one or more time periods. A lagged series with lag-1 is the original series moved forward one time period; a lagged series with lag-2 is the original series moved forward two time periods, etc. Table 17.6 shows the first 24 months of the Amtrak ridership series, the lag-1 series and the lag-2 series.

Next, to compute the lag-1 autocorrelation, which measures the linear relationship between values in consecutive time periods, we compute the correlation

TABLE 17.6 FIRST 24 MONTHS OF AMTRAK RIDERSHIP SERIES WITH LAG-1 AND LAG-2 SERIES

Month	Ridership	Lag-1 Series	Lag-2 Series
Jan 91	1709		
Feb 91	1621	1709	
Mar 91	1973	1621	1709
Apr 91	1812	1973	1621
May 91	1975	1812	1973
Jun 91	1862	1975	1812
Jul 91	1940	1862	1975
Aug 91	2013	1940	1862
Sep 91	1596	2013	1940
Oct 91	1725	1596	2013
Nov 91	1676	1725	1596
Dec 91	1814	1676	1725
Jan 92	1615	1814	1676
Feb 92	1557	1615	1814
Mar 92	1891	1557	1615
Apr 92	1956	1891	1557
May 92	1885	1956	1891
Jun 92	1623	1885	1956
Jul 92	1903	1623	1885
Aug 92	1997	1903	1623
Sep 92	1704	1997	1903
Oct 92	1810	1704	1997
Nov 92	1862	1810	1704
Dec 92	1875	1862	1810

between the original series and the lag-1 series (e.g., via the function *cor()*) to be 0.08. Note that although the original series in Table 17.6 has 24 time periods, the lag-1 autocorrelation will only be based on 23 pairs (because the lag-1 series does not have a value for January 1991). Similarly, the lag-2 autocorrelation, measuring the relationship between values that are two time periods apart, is the correlation between the original series and the lag-2 series (yielding −0.15).

We can use R's *Acf()* function in the `forecast` package to directly compute and plot the autocorrelation of a series at different lags. For example, the output for the 24-month ridership is shown in Figure 17.7.

code for creating Figure 17.7

```
ridership.24.ts <- window(train.ts, start = c(1991, 1), end = c(1991, 24))
Acf(ridership.24.ts, lag.max = 12, main = "")
```

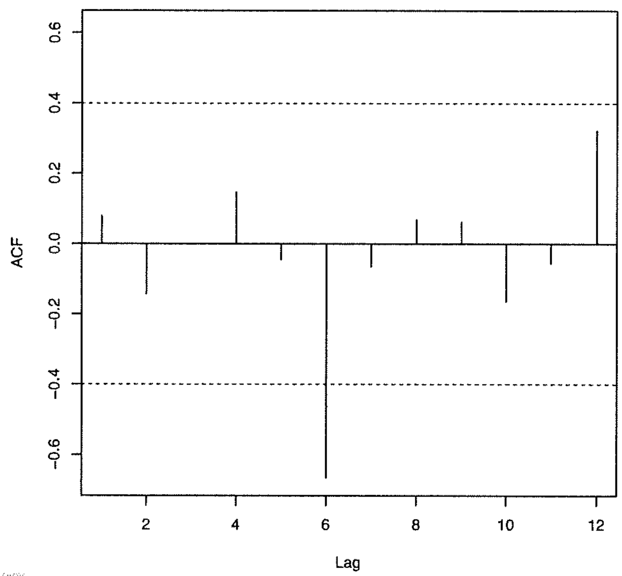

FIGURE 17.7 AUTOCORRELATION PLOT FOR LAGS 1–12 (FOR FIRST 24 MONTHS OF AMTRAK RIDERSHIP)

A few typical autocorrelation behaviors that are useful to explore are:

Strong autocorrelation (positive or negative) at a lag k larger than 1 and its multiples $(2k, 3k, \ldots)$ typically reflects a cyclical pattern. For example, strong positive lag-12 autocorrelation in monthly data will reflect an annual seasonality (where values during a given month each year are positively correlated).

Positive lag-1 autocorrelation (called "stickiness") describes a series where consecutive values move generally in the same direction. In the presence of a strong linear trend, we would expect to see a strong and positive lag-1 autocorrelation.

Negative lag-1 autocorrelation reflects swings in the series, where high values are immediately followed by low values and vice versa.

Examining the autocorrelation of a series can therefore help to detect seasonality patterns. In Figure 17.7, for example, we see that the strongest autocorrelation is at lag 6 and is negative. This indicates a bi-annual pattern in ridership, with 6-month switches from high to low ridership. A look at the time plot confirms the high-summer low-winter pattern.

In addition to looking at the autocorrelation of the raw series, it is very useful to look at the autocorrelation of the *residual series*. For example, after fitting a regression model (or using any other forecasting method), we can examine the autocorrelation of the series of residuals. If we have adequately modeled the seasonal pattern, then the residual series should show no autocorrelation at the season's lag. Figure 17.8 displays the autocorrelations for the residuals from the regression model with seasonality and quadratic trend shown in Figure 17.6. It is clear that the 6-month (and 12-month) cyclical behavior no longer dominates the series of residuals, indicating that the regression model captured them adequately. However, we can also see a strong positive autocorrelation from lag 1 on, indicating a positive relationship between neighboring residuals. This is valuable information, which can be used to improve forecasts.

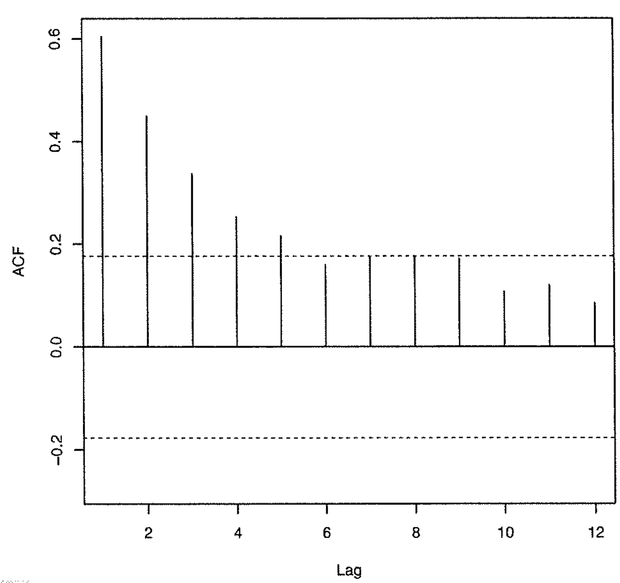

FIGURE 17.8 AUTOCORRELATION PLOT OF FORECAST ERRORS SERIES FROM FIGURE 17.6

Improving Forecasts by Integrating Autocorrelation Information

In general, there are two approaches to taking advantage of autocorrelation. One is by directly building the autocorrelation into the regression model, and the other is by constructing a second-level forecasting model on the residual series.

Among regression-type models that directly account for autocorrelation are *autoregressive* (AR) models, or the more general class of models called ARIMA (Autoregressive Integrated Moving Average) models. AR models are similar to linear regression models, except that the predictors are the past values of the series. For example, an autoregressive model of order 2, denoted AR(2), can be written as

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \epsilon \tag{17.1}$$

Estimating such models is roughly equivalent to fitting a linear regression model with the series as the outcome variable, and the two lagged series (at lag 1 and 2 in this example) as the predictors. However, it is better to use designated ARIMA estimation methods (e.g., those available in R's `forecast` package) over ordinary linear regression estimation, to produce more accurate results.[4] Moving from AR to ARIMA models creates a larger set of more flexible forecasting models, but also requires much more statistical expertise. Even with the simpler AR models, fitting them to raw time series that contain patterns such as trends and seasonality requires the user to perform several initial data transformations and to choose the order of the model. These are not straightforward tasks. Because ARIMA modeling is less robust and requires more experience and statistical expertise than other methods, the use of such models for forecasting raw series is generally less popular in practical forecasting. We therefore direct the interested reader to classic time series textbooks [e.g., see Chapter 4 in Chatfield (2003)].

However, we do discuss one particular use of AR models that is straightforward to apply in the context of forecasting, which can provide a significant improvement to short-term forecasts. This relates to the second approach for utilizing autocorrelation, which requires constructing a second-level forecasting model for the residuals, as follows:

1. Generate a k-step-ahead forecast of the series (F_{t+k}), using any forecasting method
2. Generate a k-step-ahead forecast of the forecast error (residual) (E_{t+k}), using an AR (or other) model

[4] ARIMA model estimation differs from ordinary regression estimation by accounting for the dependence between records.

3. Improve the initial k-step-ahead forecast of the series by adjusting it according to its forecasted error: *Improved* $F^*_{t+k} = F_{t+k} + E_{t+k}$.

In particular, we can fit low-order AR models to series of residuals (or *forecast errors*) which can then be used to forecast future forecast errors. By fitting the series of residuals, rather than the raw series, we avoid the need for initial data transformations (because the residual series is not expected to contain any trends or cyclical behavior besides autocorrelation).

To fit an AR model to the series of residuals, we first examine the autocorrelations of the residual series. We then choose the order of the AR model according to the lags in which autocorrelation appears. Often, when autocorrelation exists at lag 1 and higher, it is sufficient to fit an AR(1) model of the form

$$E_t = \beta_0 + \beta_1 E_{t-1} + \epsilon \tag{17.2}$$

where E_t denotes the residual (or *forecast error*) at time t. For example, although the autocorrelations in Figure 17.8 appear large from lags 1 to 10 or so, it is likely that an AR(1) would capture all of these relationships. The reason is that if immediate neighboring values are correlated, then the relationship propagates to values that are two periods away, then three periods away, etc.[5]

The result of fitting an AR(1) model to the Amtrak ridership residual series is shown in Table 17.7. The AR(1) coefficient (0.5998) is close to the lag-1 autocorrelation (0.6041) that we found earlier (Figure 17.8). The forecasted residual for April 2001 is computed by plugging in the most recent residual from March 2001 (equal to 12.108) into the AR(1) model[6]:

$$0.3728491 + (0.5997814)(12.108 - 0.3728491) = 7.411.$$

You can obtain this number directly by using the *forecast()* function (see output in Table 17.7). The positive value tells us that the regression model will produce a ridership forecast for April 2001 that is too low and that we should adjust it up by adding 7411 riders. In this particular example, the regression model (with quadratic trend and seasonality) produced a forecast of 2,004,271 riders, and the improved two-stage model [regression + AR(1) correction] corrected it by increasing it to 2,011,906 riders. The actual value for April 2001 turned out to be 2,023,792 riders—much closer to the improved forecast.

[5] *Partial autocorrelations* (use function *Pacf()*) measure the contribution of each lag series *over and above* smaller lags. For example, the lag-2 partial autocorrelation is the contribution of lag-2 beyond that of lag-1.

[6] The intercept in the Coefficients table resulting from function *Arima()* is not exactly an intercept—it is the estimated mean of the series. Hence, to get a forecast, we must subtract this coefficient from our value. In this case, we have F_{t+1} = intercept + slope (y_t − intercept).

TABLE 17.7 OUTPUT FOR AR(1) MODEL ON RIDERSHIP RESIDUALS

R code for running AR(1) model on residuals

```
# fit linear regression with quadratic trend and seasonality to Ridership
train.lm.trend.season <- tslm(train.ts ~ trend + I(trend^2) + season)

# fit AR(1) model to training residuals
# use Arima() in the forecast package to fit an ARIMA model
# (that includes AR models); order = c(1,0,0) gives an AR(1).
train.res.arima <- Arima(train.lm.trend.season$residuals, order = c(1,0,0))
valid.res.arima.pred <- forecast(train.res.arima, h = 1)
```

Output

```
> summary(train.res.arima)
Series: train.lm.trend.season$residuals
ARIMA(1,0,0) with non-zero mean

Coefficients:
            ar1   intercept
      0.5997814   0.3728491
s.e.  0.0712246  11.8408218

sigma^2 estimated as 2829:  log likelihood=-663.54
AIC=1333.08   AICc=1333.29   BIC=1341.52

> valid.res.arima.pred
         Point Forecast        Lo 80       Hi 80        Lo 95       Hi 95
Apr 2001   7.4111097650 -61.31747817 76.13969770  -97.70019491 112.5224144
```

From the plot of the actual vs. forecasted residual series (Figure 17.9), we can see that the AR(1) model fits the residual series quite well. Note, however, that the plot is based on the training data (until March 2001). To evaluate predictive performance of the two-level model [regression + AR(1)], we would have to examine performance (e.g., via MAPE or RMSE metrics) on the validation data, in a fashion similar to the calculation that we performed for April 2001.

Finally, to examine whether we have indeed accounted for the autocorrelation in the series and that no more information remains in the series, we examine the autocorrelations of the series of residuals-of-residuals (the residuals obtained after the AR(1), which was applied to the regression residuals). This is seen in Figure 17.10. It is clear that no more autocorrelation remains, and that the addition of the AR(1) model has captured the autocorrelation information adequately.

We mentioned earlier that improving forecasts via an additional AR layer is useful for short-term forecasting. The reason is that an AR model of order k will usually only provide useful forecasts for the next k periods, and after that forecasts will rely on earlier forecasts rather than on actual data. For example, to

code for creating Figure 17.9

```
plot(train.lm.trend.season$residuals, ylim = c(-250, 250),  ylab = "Residuals",
    xlab = "Time", bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "")
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(train.res.arima.pred$fitted, lwd = 2, col = "blue")
```

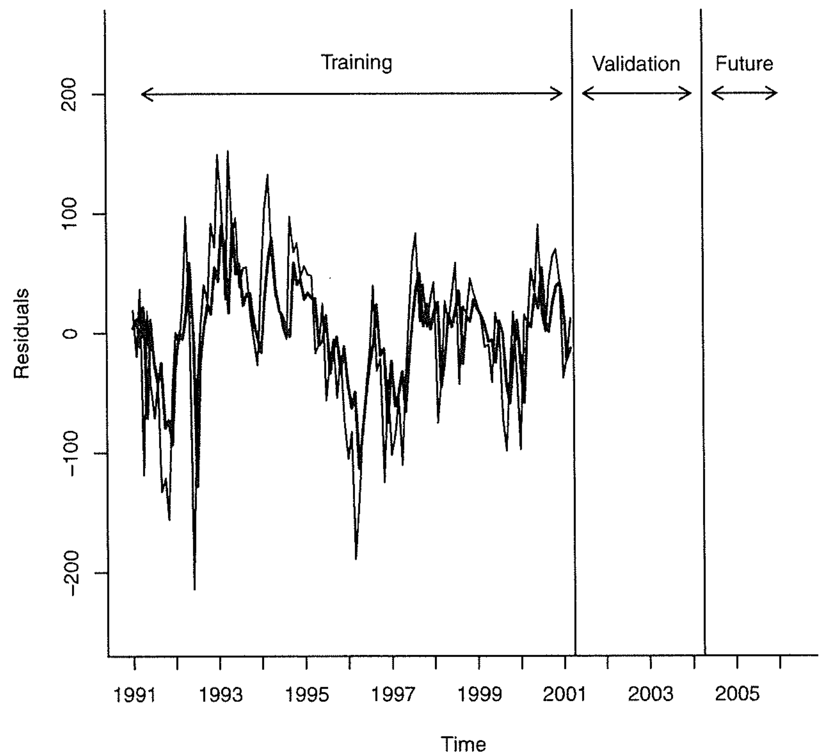

FIGURE 17.9 FITTING AN AR(1) MODEL TO THE RESIDUAL SERIES FROM FIGURE 17.6

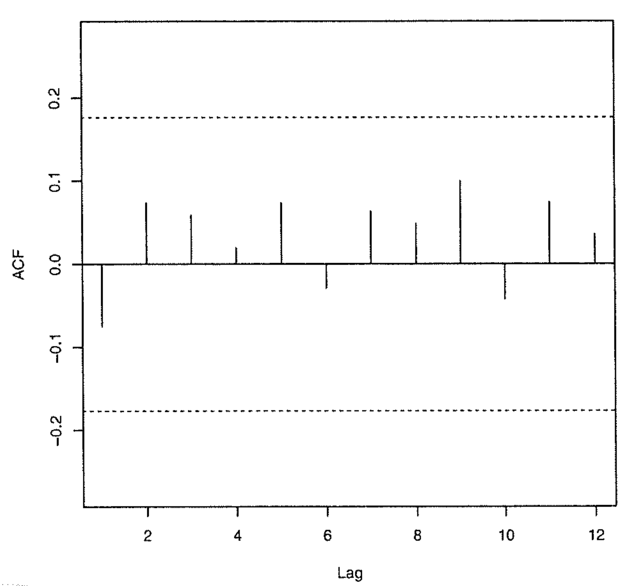

FIGURE 17.10 AUTOCORRELATIONS OF RESIDUALS-OF-RESIDUALS SERIES

forecast the residual of May 2001 when the time of prediction is March 2001, we would need the residual for April 2001. However, because that value is not available, it would be replaced by its forecast. Hence, the forecast for May 2001 would be based on the forecast for April 2001.

Evaluating Predictability

Before attempting to forecast a time series, it is important to determine whether it is predictable, in the sense that its past can be used to predict its future beyond the naive forecast. One useful way to assess predictability is to test whether the series is a *random walk*. A random walk is a series in which changes from one time period to the next are random. According to the efficient market hypothesis in economics, asset prices are random walks and therefore predicting stock prices is a game of chance.[7]

A random walk is a special case of an AR(1) model, where the slope coefficient is equal to 1:

$$Y_t = \beta_0 + Y_{t-1} + \epsilon_t. \tag{17.3}$$

We can also write this as

$$Y_t - Y_{t-1} = \beta_0 + \epsilon_t. \tag{17.4}$$

We see from the last equation that the difference between the values at periods $t-1$ and t is random, hence the term "random walk." Forecasts from such a model are basically equal to the most recent observed value (the naive forecast), reflecting the lack of any other information.

To test whether a series is a random walk, we fit an AR(1) model and test the hypothesis that the slope coefficient is equal to 1 ($H_0 : \beta_1 = 1$ vs. $H_1 : \beta_1 \neq 1$). If the null hypothesis is rejected (reflected by a small p-value), then the series is not a random walk and we can attempt to predict it.

As an example, consider the AR(1) model shown in Table 17.7. The slope coefficient (0.5998) is more than 5 standard errors away from 1, indicating that this is not a random walk. In contrast, consider the AR(1) model fitted to the series of S&P500 monthly closing prices between May 1995 and August 2003 (in *SP500.csv*, shown in Table 17.8). Here the slope coefficient is 0.9833, with a standard error of 0.0145. The coefficient is sufficiently close to 1 (around one standard error away), indicating that this is a random walk. Forecasting this series using any of the methods described earlier (aside from the naive forecast) is therefore futile.

[7] There is some controversy surrounding the efficient market hypothesis, with claims that there is slight autocorrelation in asset prices, which does make them predictable to some extent. However, transaction costs and bid-ask spreads tend to offset any prediction benefits.

TABLE 17.8 OUTPUT FOR AR(1) MODEL ON S&P500 MONTHLY CLOSING PRICES

```
> sp500.df <- read.csv("SP500.csv")

> sp500.ts <- ts(sp500.df$Close, start = c(1995, 5), end = c(2003, 8), freq = 12)
> sp500.arima <- Arima(sp500.ts, order = c(1,0,0))
> sp500.arima
Series: sp500.ts
ARIMA(1,0,0) with non-zero mean

Coefficients:
            ar1    intercept
      0.9833410  890.0706502
s.e.  0.0145083  221.0280170

sigma^2 estimated as 2833.446:  log likelihood=-540.05
AIC=1086.1   AICc=1086.35   BIC=1093.92
```

PROBLEMS

17.1 **Impact of September 11 on Air Travel in the United States.** The Research and Innovative Technology Administration's Bureau of Transportation Statistics conducted a study to evaluate the impact of the September 11, 2001 terrorist attack on US transportation. The 2006 study report and the data can be found at http://goo.gl/w2lJPV. The goal of the study was stated as follows:

> The purpose of this study is to provide a greater understanding of the passenger travel behavior patterns of persons making long distance trips before and after 9/11.

The report analyzes monthly passenger movement data between January 1990 and May 2004. Data on three monthly time series are given in file *Sept11Travel.csv* for this period: (1) Actual airline revenue passenger miles (Air), (2) rail passenger miles (Rail), and (3) vehicle miles traveled (Car).

In order to assess the impact of September 11, BTS took the following approach: using data before September 11, they forecasted future data (under the assumption of no terrorist attack). Then, they compared the forecasted series with the actual data to assess the impact of the event. Our first step, therefore, is to split each of the time series into two parts: pre- and post September 11. We now concentrate only on the earlier time series.

a. Plot the pre-event AIR time series. What time series components appear?

b. Figure 17.11 shows a time plot of the **seasonally adjusted** pre-September-11 AIR series. Which of the following methods would be adequate for forecasting the series shown in the figure?

- Linear regression model seasonality
- Linear regression model with trend
- Linear regression model with trend and seasonality

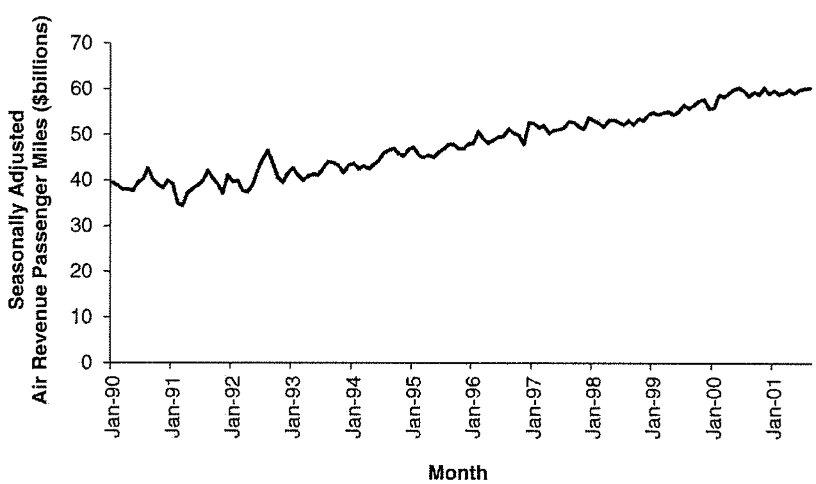

FIGURE 17.11 SEASONALLY ADJUSTED PRE-SEPTEMBER-11 AIR SERIES

c. Specify a linear regression model for the AIR series that would produce a seasonally adjusted series similar to the one shown in Figure 17.11, with multiplicative seasonality. What is the outcome variable? What are the predictors?

d. Run the regression model from (c). Remember to use only pre-event data.

i. What can we learn from the statistical insignificance of the coefficients for October and September?

ii. The actual value of AIR (air revenue passenger miles) in January 1990 was 35.153577 billion. What is the residual for this month, using the regression model? Report the residual in terms of air revenue passenger miles.

e. Create an ACF (autocorrelation) plot of the regression residuals.

i. What does the ACF plot tell us about the regression model's forecasts?

ii. How can this information be used to improve the model?

f. Fit linear regression models to Air, Rail, and to Auto with additive seasonality and an appropriate trend. For Air and Rail, fit a linear trend. For Rail, use a quadratic trend. Remember to use only pre-event data. Once the models are estimated, use them to forecast each of the three post-event series.

i. For each series (Air, Rail, Auto), plot the complete pre-event and post-event actual series overlayed with the predicted series.

ii. What can be said about the effect of the September 11 terrorist attack on the three modes of transportation? Discuss the magnitude of the effect, its time span, and any other relevant aspects.

17.2 **Analysis of Canadian Manufacturing Workers Workhours.** The time plot in Figure 17.12 describes the average annual number of weekly hours spent by Canadian manufacturing workers (data are available in *CanadianWorkHours.csv*, data courtesy of Ken Black).

a. Which of the following regression models would fit the series best? (Choose one.)

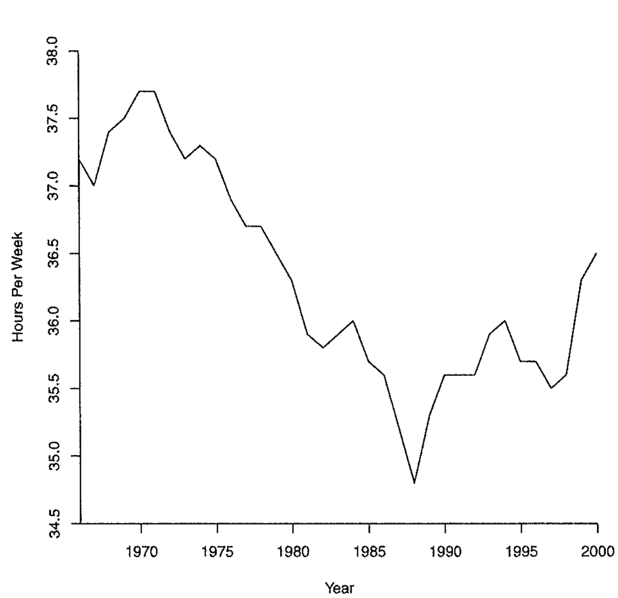

FIGURE 17.12 AVERAGE ANNUAL WEEKLY HOURS SPENT BY CANADIAN MANUFACTURING WORKERS

- Linear trend model
- Linear trend model with seasonality
- Quadratic trend model
- Quadratic trend model with seasonality

b. If we computed the autocorrelation of this series, would the lag-1 autocorrelation exhibit negative, positive, or no autocorrelation? How can you see this from the plot?

c. Compute the autocorrelation of the series and produce an ACF plot. Verify your answer to the previous question.

17.3 **Toys "R" Us Revenues.** Figure 17.13 is a time plot of the quarterly revenues of Toys "R" Us between 1992 and 1995 (thanks to Chris Albright for suggesting the use of these data, which are available in *ToysRUsRevenues.csv*).

a. Fit a regression model with a linear trend and additive seasonality. Use the entire series (excluding the last two quarters) as the training set.

b. A partial output of the regression model is shown in Table 17.9 (where *season2* is the Quarter 2 dummy). Use this output to answer the following questions:

i. Which two statistics (and their values) measure how well this model fits the training data?

ii. Which two statistics (and their values) measure the predictive accuracy of this model?

iii. After adjusting for trend, what is the average difference between sales in Q3 and sales in Q1?

iv. After adjusting for seasonality, which quarter (Q_1, Q_2, Q_3, or Q_4) has the highest average sales?

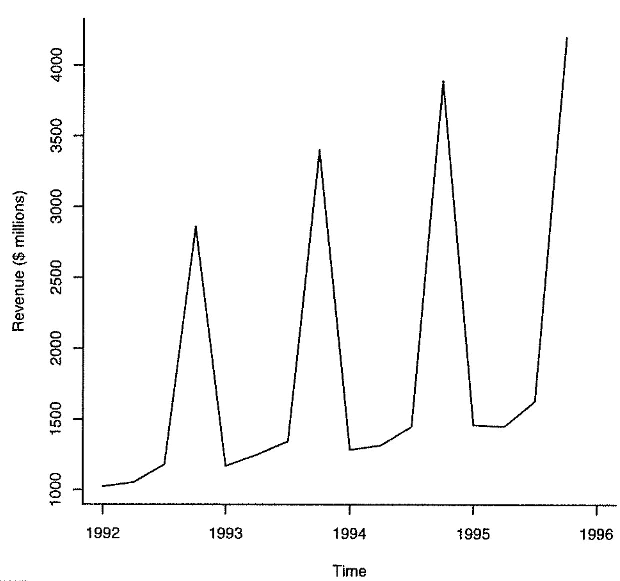

FIGURE 17.13 QUARTERLY REVENUES OF TOYS "R" US, 1992–1995

TABLE 17.9 REGRESSION MODEL FITTED TO TOYS "R" US TIME SERIES AND ITS PREDICTIVE PERFORMANCE IN TRAINING AND VALIDATION PERIODS

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   906.75     115.35   7.861 2.55e-05 ***
trend          47.11      11.26   4.185  0.00236 **
season2       -15.11     119.66  -0.126  0.90231
season3        89.17     128.67   0.693  0.50582
season4      2101.73     129.17  16.272 5.55e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 .

                   ME     RMSE       MAE       MPE     MAPE      MASE
Training set   0.0000 135.0795  92.53061 0.1614994 5.006914 0.4342122
Test set     183.1429 313.6820 254.66667 3.0193814 7.404655 1.1950571
```

17.4 Walmart Stock. Figure 17.14 shows the series of Walmart daily closing prices between February 2001 and February 2002 (Thanks to Chris Albright for suggesting the use of these data, which are publicly available, for example, at http://finance.yahoo.com and are in the file *WalMartStock.csv*).

a. Fit an AR(1) model to the close price series. Report the coefficient table.

b. Which of the following is/are relevant for testing whether this stock is a random walk?

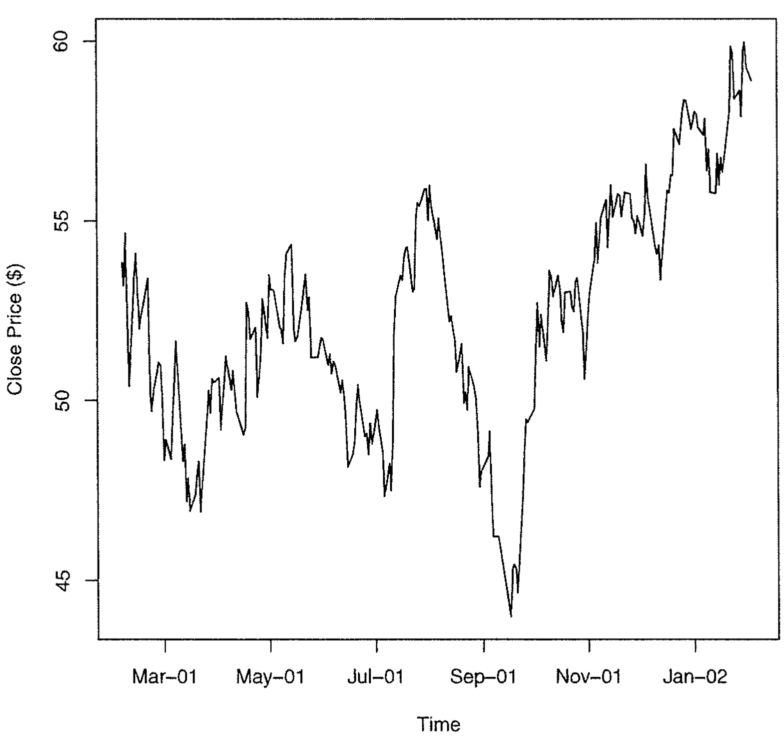

FIGURE 17.14 DAILY CLOSE PRICE OF WALMART STOCK, FEBRUARY 2001–2002

- The autocorrelations of the close prices series
- The AR(1) slope coefficient
- The AR(1) constant coefficient

c. Does the AR model indicate that this is a random walk? Explain how you reached your conclusion.

d. What are the implications of finding that a time-series is a random walk? Choose the correct statement(s) below.

- It is impossible to obtain forecasts that are more accurate than naive forecasts for the series
- The series is random
- The changes in the series from one period to the next are random

17.5 Department Store Sales. The time plot in Figure 17.15 describes actual quarterly sales for a department store over a 6-year period (data are available in *DepartmentStoreSales.csv*, data courtesy of Chris Albright).

a. The forecaster decided that there is an exponential trend in the series. In order to fit a regression-based model that accounts for this trend, which of the following operations must be performed?

- Take log of quarter index
- Take log of sales
- Take an exponent of sales
- Take an exponent of quarter index

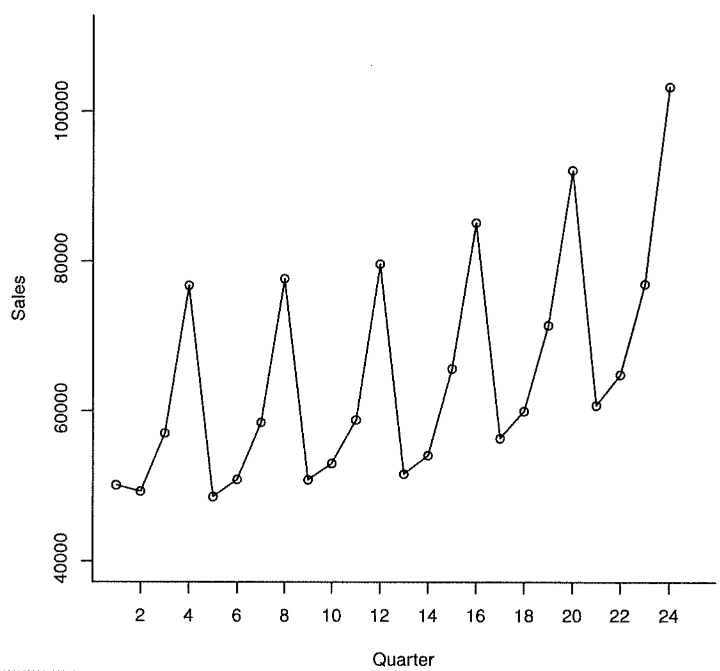

FIGURE 17.15 DEPARTMENT STORE QUARTERLY SALES SERIES

b. Fit a regression model with an exponential trend and seasonality, using the first 20 quarters as the training data (remember to first partition the series into training and validation series).

c. A partial output is shown in Table 17.10. From the output, after adjusting for trend, are Q2 average sales higher, lower, or approximately equal to the average Q1 sales?

TABLE 17.10 OUTPUT FROM REGRESSION MODEL FIT TO DEPARTMENT STORE SALES IN THE TRAINING PERIOD

```
> summary(tslm(sales.ts ~ trend + season, lambda = 0))

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.748945   0.018725 574.057  < 2e-16 ***
trend        0.011088   0.001295   8.561 3.70e-07 ***
season2      0.024956   0.020764   1.202    0.248
season3      0.165343   0.020884   7.917 9.79e-07 ***
season4      0.433746   0.021084  20.572 2.10e-12 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 .
```

d. Use this model to forecast sales in quarters 21 and 22.

e. The plots in Figure 17.16 describe the fit (top) and forecast errors (bottom) from this regression model.

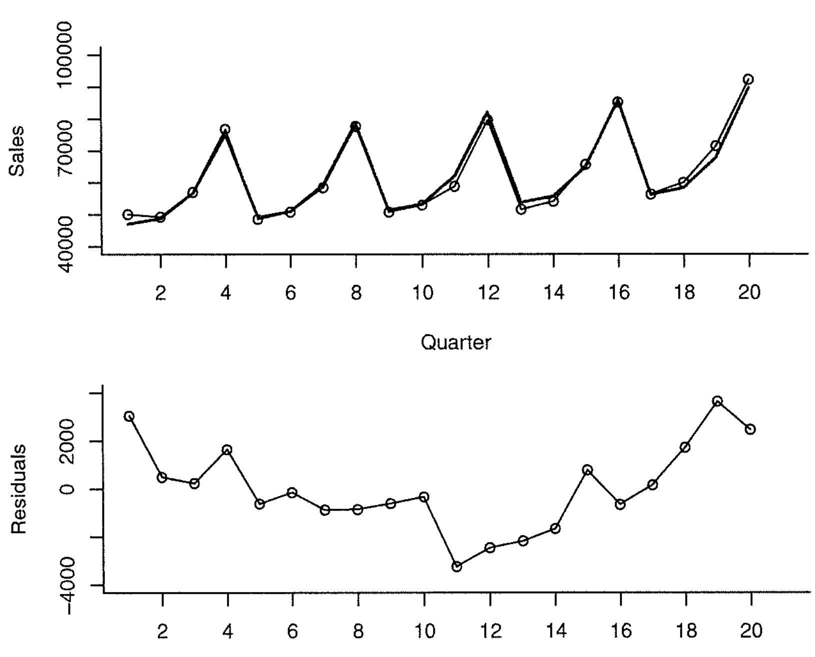

FIGURE 17.16 FIT OF REGRESSION MODEL FOR DEPARTMENT STORE SALES

i. Recreate these plots.

ii. Based on these plots, what can you say about your forecasts for quarters 21 and 22? Are they likely to over-forecast, under-forecast, or be reasonably close to the real sales values?

f. From the forecast errors plot, which of the following statements appear true?

- Seasonality is not captured well
- The regression model fits the data well
- The trend in the data is not captured well by the model

g. Which of the following solutions is adequate *and* a parsimonious solution for improving model fit?

- Fit a quadratic trend model to the residuals (with Quarter and Quarter^2)
- Fit an AR model to the residuals
- Fit a quadratic trend model to Sales (with Quarter and Quarter^2)

17.6 **Souvenir Sales.** Figure 17.17 shows a time plot of monthly sales for a souvenir shop at a beach resort town in Queensland, Australia, between 1995 and 2001 (Data are available in *SouvenirSales.csv*, source: Hyndman, R.J., Time Series Data Library, http://data.is/TSDLdemo. Accessed on 07/25/15.). The series is presented twice, in Australian dollars and in log-scale. Back in 2001, the store wanted to use the data to forecast sales for the next 12 months (year 2002). They hired an analyst to generate forecasts. The analyst first partitioned the data into training and validation sets, with the validation set containing the last 12 months of data (year 2001). She then fit a regression model to sales, using the training set.

a. Based on the two time plots, which predictors should be included in the regression model? What is the total number of predictors in the model?

b. Run a regression model with Sales (in Australian dollars) as the outcome variable, and with a linear trend and monthly seasonality. Remember to fit only the training data. Call this model A.

i. Examine the estimated coefficients: which month tends to have the highest average sales during the year? Why is this reasonable?

ii. The estimated trend coefficient in model A is 245.36. What does this mean?

c. Run a regression model with an exponential trend and multiplicative seasonality. Remember to fit only the training data. Call this model B.

i. Fitting a model to log(Sales) with a linear trend is equivalent to fitting a model to Sales (in dollars) with what type of trend?

ii. The estimated trend coefficient in model B is 0.02. What does this mean?

iii. Use this model to forecast the sales in February 2002.

d. Compare the two regression models (A and B) in terms of forecast performance. Which model is preferable for forecasting? Mention at least two reasons based on the information in the outputs.

e. Continuing with model B, create an ACF plot until lag 15 for the forecast errors. Now fit an AR model with lag 2 [ARIMA(2,0,0)] to the forecast errors.

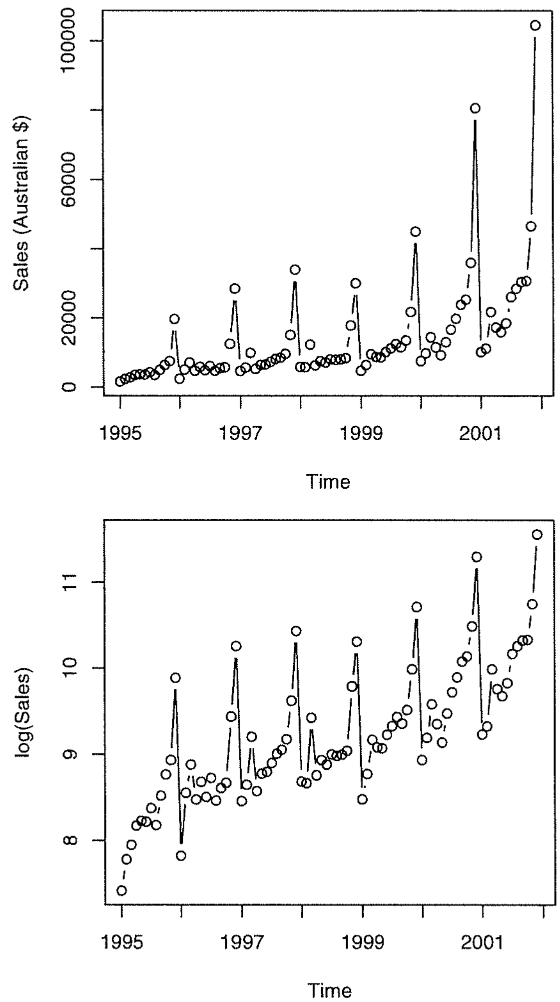

FIGURE 17.17 MONTHLY SALES AT AUSTRALIAN SOUVENIR SHOP IN DOLLARS (TOP) AND IN LOG-SCALE (BOTTOM)

   i. Examining the ACF plot and the estimated coefficients of the AR(2) model (and their statistical significance), what can we learn about the forecasts that result from model B?

   ii. Use the autocorrelation information to compute an improved forecast for January 2002, using model B and the AR(2) model above.

f. How would you model these data differently if the goal was to understand the different components of sales in the souvenir shop between 1995–2001? Mention two differences.

17.7 **Shipments of Household Appliances.** The time plot in Figure 17.18 shows the series of quarterly shipments (in million dollars) of US household appliances between 1985-1989 (data are available in *ApplianceShipments.csv*, data courtesy of Ken Black). If we compute the autocorrelation of the series, which lag (> 0) is most likely to have the largest coefficient (in absolute value)? Create an ACF plot and compare with your answer.

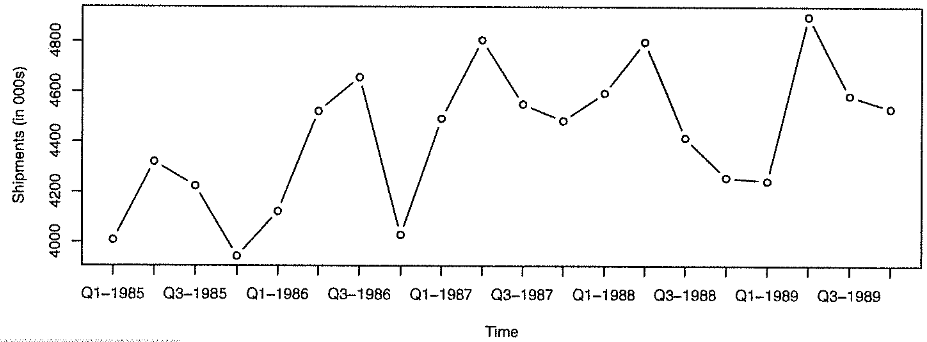

FIGURE 17.18 QUARTERLY SHIPMENTS OF US HOUSEHOLD APPLIANCES OVER 5 YEARS

17.8 **Australian Wine Sales.** Figure 17.19 shows time plots of monthly sales of six types of Australian wines (red, rose, sweet white, dry white, sparkling, and fortified) for 1980–1994 (Data are available in *AustralianWines.csv*, source: Hyndman, R.J., Time Series Data Library, http://data.is/TSDLdemo. Accessed on 07/25/15.). The units are thousands of litres. You are hired to obtain short term forecasts (2–3 months ahead) for each of the six series, and this task will be repeated every month.

a. Which forecasting method would you choose if you had to choose the same method for all series? Why?

b. Fortified wine has the largest market share of the above six types of wine. You are asked to focus on fortified wine sales alone, and produce as accurate as possible forecasts for the next 2 months.

- Start by partitioning the data: use the period until December 1993 as the training set.
- Fit a regression model to sales with a linear trend and additive seasonality.

i. Create the "actual vs. forecast" plot. What can you say about the model fit?

ii. Use the regression model to forecast sales in January and February 1994.

c. Create an ACF plot for the residuals from the above model until lag 12. Examining this plot (only), which of the following statements are reasonable conclusions?

- Decembers (month 12) are not captured well by the model.
- There is a strong correlation between sales on the same calendar month.
- The model does not capture the seasonality well.
- We should try to fit an autoregressive model with lag 12 to the residuals.

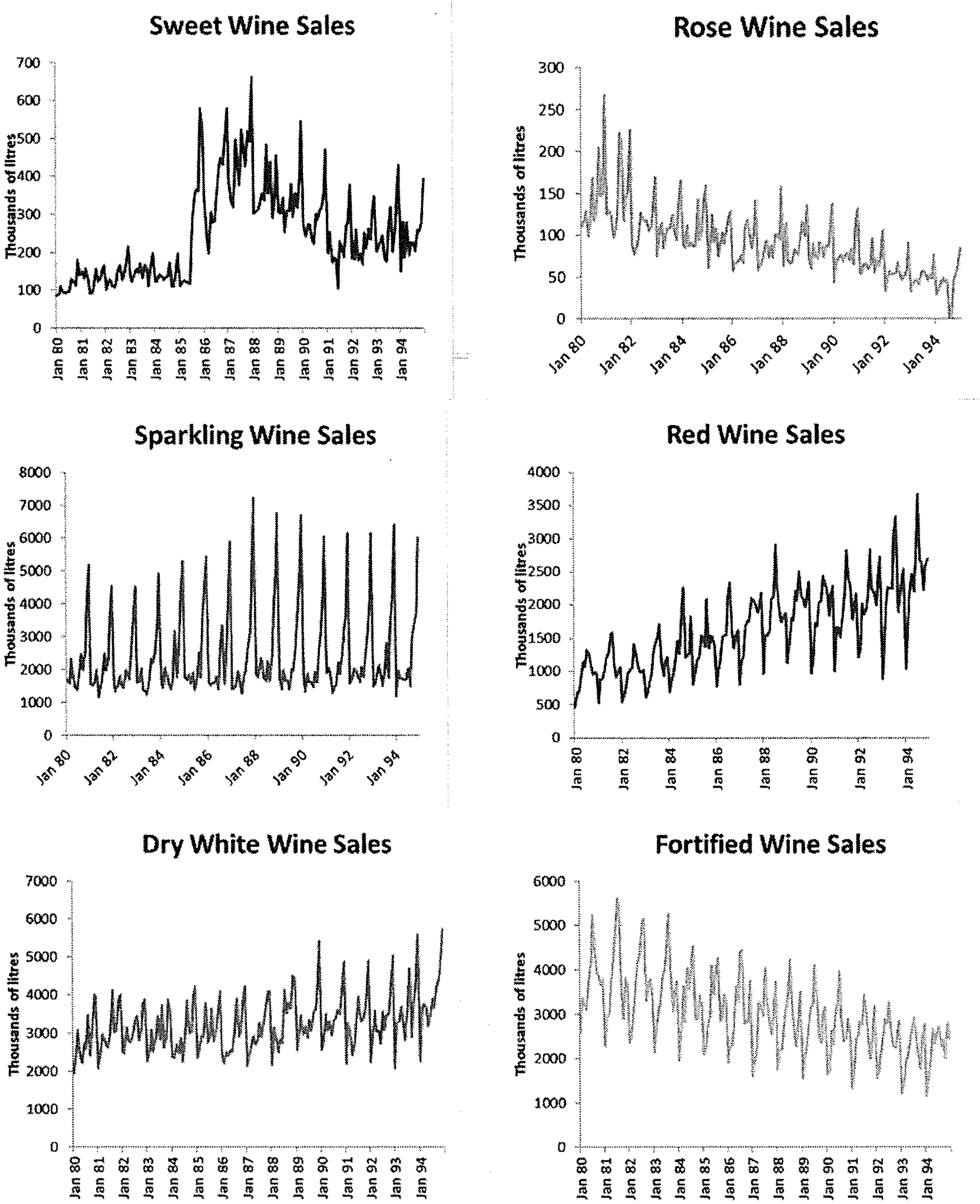

FIGURE 17.19 MONTHLY SALES OF SIX TYPES OF AUSTRALIAN WINES BETWEEN 1980–1994

CHAPTER 18

Smoothing Methods

In this chapter, we describe a set of popular and flexible methods for forecasting time series that rely on *smoothing*. Smoothing is based on averaging over multiple periods in order to reduce the noise. We start with two simple smoothers, the *moving average* and *simple exponential smoother*, which are suitable for forecasting series that contain no trend or seasonality. In both cases, forecasts are averages of previous values of the series (the length of the series history considered and the weights used in the averaging differ between the methods). We also show how a moving average can be used, with a slight adaptation, for data visualization. We then proceed to describe smoothing methods suitable for forecasting series with a trend and/or seasonality. Smoothing methods are data-driven, and are able to adapt to changes in the series over time. Although highly automated, the user must specify *smoothing constants* that determine how fast the method adapts to new data. We discuss the choice of such constants, and their meaning. The different methods are illustrated using the Amtrak ridership series.

18.1 INTRODUCTION[1]

A second class of methods for time series forecasting is *smoothing methods*. Unlike regression models, which rely on an underlying theoretical model for the components of a time series (e.g., linear trend or multiplicative seasonality), smoothing methods are data-driven, in the sense that they estimate time series components directly from the data without assuming a predetermined structure. Data-driven methods are especially useful in series where patterns change over

[1]This and subsequent sections in this chapter copyright © 2017 Datastats, LLC, and Galit Shmueli. Used by permission.

Data Mining for Business Analytics: Concepts, Techniques, and Applications in R, First Edition.
Galit Shmueli, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, and Kenneth C. Lichtendahl, Jr.

time. Smoothing methods "smooth" out the noise in a series in an attempt to uncover the patterns. Smoothing is done by averaging the series over multiple periods, where different smoothers differ by the number of periods averaged, how the average is computed, how many times averaging is performed, and so on. We now describe two types of smoothing methods that are popular in business applications due to their simplicity and adaptability. These are the moving average method and exponential smoothing.

18.2 Moving Average

The moving average is a simple smoother: it consists of averaging values across a window of consecutive periods, thereby generating a series of averages. A moving average with window width w means averaging across each set of w consecutive values, where w is determined by the user.

In general, there are two types of moving averages: a *centered moving average* and a *trailing moving average*. Centered moving averages are powerful for visualizing trends, because the averaging operation can suppress seasonality and noise, thereby making the trend more visible. In contrast, trailing moving averages are useful for forecasting. The difference between the two is in terms of the window's location on the time series.

Centered Moving Average for Visualization

In a centered moving average, the value of the moving average at time t (MA_t) is computed by centering the window around time t and averaging across the w values within the window:

$$\text{MA}_t = \left(Y_{t-(w-1)/2} + \cdots + Y_{t-1} + Y_t + Y_{t+1} + \cdots + Y_{t+(w-1)/2}\right)/w. \tag{18.1}$$

For example, with a window of width $w = 5$, the moving average at time point $t = 3$ means averaging the values of the series at time points $1, 2, 3, 4, 5$; at time point $t = 4$, the moving average is the average of the series at time points $2, 3, 4, 5, 6$, and so on.[2] This is illustrated in the top panel of Figure 18.1.

Choosing the window width in a seasonal series is straightforward: because the goal is to suppress seasonality for better visualizing the trend, the default choice should be the length of a seasonal cycle. Returning to the Amtrak ridership data, the annual seasonality indicates a choice of $w = 12$. Figure 18.2 (smooth black line) shows a centered moving average line overlaid on the original series. We can see a global U-shape, but unlike the regression model that fits

[2]For an even window width, for example, $w = 4$, obtaining the moving average at time point $t = 3$ requires averaging across two windows: across time points $1, 2, 3, 4$; across time points $2, 3, 4, 5$; and finally the average of the two averages is the final moving average.

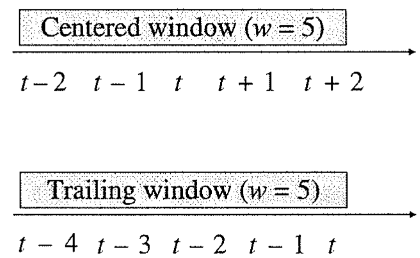

FIGURE 18.1 SCHEMATIC OF CENTERED MOVING AVERAGE (TOP) AND TRAILING MOVING AVERAGE (BOTTOM), BOTH WITH WINDOW WIDTH $W = 5$

a strict U-shape, the moving average shows some deviation, such as the slight dip during the last year.

Trailing Moving Average for Forecasting

Centered moving averages are computed by averaging across data in the past and the future of a given time point. In that sense, they cannot be used for forecasting because at the time of forecasting, the future is typically unknown. Hence, for purposes of forecasting, we use *trailing moving averages*, where the window of width w is set on the most recent available w values of the series. The k-step ahead forecast F_{t+k} ($k = 1, 2, 3, \ldots$) is then the average of these w values (see also bottom plot in Figure 18.1):

$$F_{t+k} = \left(Y_t + Y_{t-1} + \cdots + Y_{t-w+1}\right)/w$$

For example, in the Amtrak ridership series, to forecast ridership in February 1992 or later months, given information until January 1992 and using a moving average with window width $w = 12$, we would take the average ridership during the most recent 12 months (February 1991 to January 1992). Figure 18.2 (broken black line) shows a trailing moving average line overlaid on the original series.

Next, we illustrate a 12-month moving average forecaster for the Amtrak ridership. We partition the Amtrak ridership time series, leaving the last 36 months as the validation period. Applying a moving average forecaster with window $w = 12$, we obtained the output shown in Figure 18.3. Note that for the first 12 records of the training period, there is no forecast (because there are less than 12 past values to average). Also, note that the forecasts for all months in the validation period are identical (1938.481) because the method assumes that information is known only until March 2001.

In this example, it is clear that the moving average forecaster is inadequate for generating monthly forecasts because it does not capture the seasonality in the data. Seasons with high ridership are under-forecasted, and seasons with low ridership are over-forecasted. A similar issue arises when forecasting a series with a trend: the moving average "lags behind," thereby under-forecasting in

code for creating Figure 18.2

```
library(zoo)

# centered moving average with window order = 12
ma.centered <- ma(ridership.ts, order = 12)

# trailing moving average with window k = 12
# in rollmean(), use argument align = right to calculate a trailing moving average.
ma.trailing <- rollmean(ridership.ts, k = 12, align = "right")

# generate a plot
plot(ridership.ts, ylim = c(1300, 2200),  ylab = "Ridership",
    xlab = "Time", bty = "l", xaxt = "n",
    xlim = c(1991,2004.25), main = "")
axis(1, at = seq(1991, 2004.25, 1), labels = format(seq(1991, 2004.25, 1)))
lines(ma.centered, lwd = 2)
lines(ma.trailing, lwd = 2, lty = 2)
legend(1994,2200, c("Ridership","Centered Moving Average", "Trailing Moving Average"),
    lty=c(1,1,2), lwd=c(1,2,2), bty = "n")
```

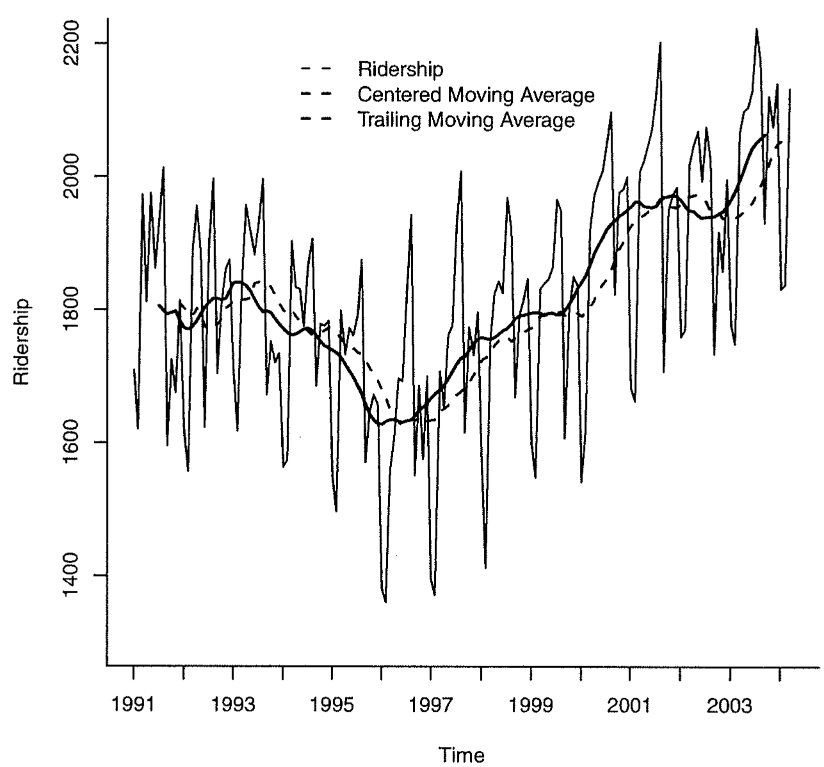

FIGURE 18.2 CENTERED MOVING AVERAGE (SMOOTH BLACK LINE) AND TRAILING MOVING AVERAGE (BROKEN BLACK LINE) WITH WINDOW $W = 12$, OVERLAID ON AMTRAK RIDERSHIP SERIES

R code for creating Figure 18.3

```
# partition the data
nValid <- 36
nTrain <- length(ridership.ts) - nValid
train.ts <- window(ridership.ts, start = c(1991, 1), end = c(1991, nTrain))
valid.ts <- window(ridership.ts, start = c(1991, nTrain + 1),
    end = c(1991, nTrain + nValid))

# moving average on training
ma.trailing <- rollmean(train.ts, k = 12, align = "right")

# obtain the last moving average in the training period
last.ma <- tail(ma.trailing, 1)

# create forecast based on last MA
ma.trailing.pred <- ts(rep(last.ma, nValid), start = c(1991, nTrain + 1),
    end = c(1991, nTrain + nValid), freq = 12)

# plot the series
plot(train.ts, ylim = c(1300, 2600),  ylab = "Ridership", xlab = "Time", bty = "l",
    xaxt = "n", xlim = c(1991,2006.25), main = "")
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(ma.trailing, lwd = 2, col = "blue")
lines(ma.trailing.pred, lwd = 2, col = "blue", lty = 2)
lines(valid.ts)
```

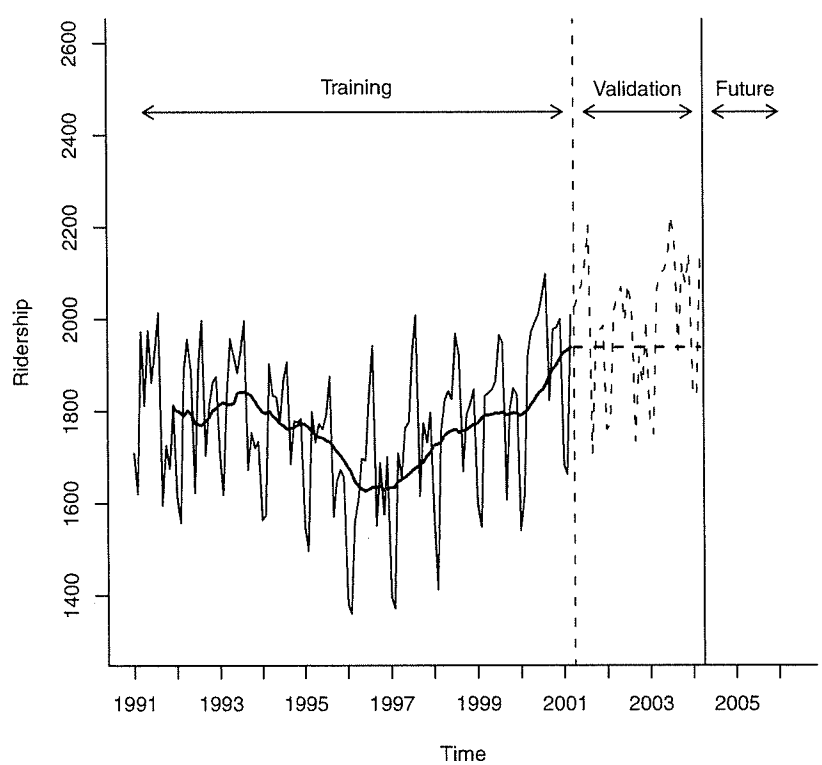

FIGURE 18.3 TRAILING MOVING AVERAGE FORECASTER WITH $W = 12$ APPLIED TO AMTRAK RIDERSHIP SERIES

the presence of an increasing trend and over-forecasting in the presence of a decreasing trend. This "lagging behind" of the trailing moving average can also be seen in Figure 18.2.

In general, the moving average should be used for forecasting *only in series that lack seasonality and trend*. Such a limitation might seem impractical. However, there are a few popular methods for removing trends (de-trending) and removing seasonality (de-seasonalizing) from a series, such as regression models. The moving average can then be used to forecast such de-trended and de-seasonalized series, and then the trend and seasonality can be added back to the forecast. For example, consider the regression model shown in Figure 17.6 in Chapter 17, which yields residuals devoid of seasonality and trend (see bottom chart). We can apply a moving average forecaster to that series of residuals (also called forecast errors), thereby creating a forecast for the next *forecast error*. For example, to forecast ridership in April 2001 (the first period in the validation set), assuming that we have information until March 2001, we use the regression model in Table 17.5 to generate a forecast for April 2001 (which yields 2004.271 thousand riders). We then use a 12-month moving average (using the period April 2000 to March 2001) to forecast the *forecast error* for April 2001, which yields 30.78068 (manually, or using R, as shown in Table 18.1). The positive value implies that the regression model's forecast for April 2001 is too low, and therefore we should adjust it by adding approximately 31 thousand riders to the regression model's forecast of 2004.271 thousand riders.

TABLE 18.1 APPLYING MA TO THE RESIDUALS FROM THE REGRESSION MODEL (WHICH LACK TREND AND SEASONALITY), TO FORECAST THE APRIL 2001 RESIDUAL

R code for applying moving average to residuals

```
# fit regression model with trend and seasonality
train.lm.trend.season <- tslm(train.ts ~ trend + I(trend^2) + season)

# create single-point forecast
train.lm.trend.season.pred <- forecast(train.lm.trend.season, h = 1, level = 0)

# apply MA to residuals
ma.trailing <- rollmean(train.lm.trend.season$residuals, k = 12, align = "right")
last.ma <- tail(ma.trailing, 1)
```

Output

```
> train.lm.trend.season.pred
         Point Forecast     Lo 0     Hi 0
Apr 2001       2004.271 2004.271 2004.271
> last.ma
[1] 30.78068
```

Choosing Window Width (w)

With moving average forecasting or visualization, the only choice that the user must make is the width of the window (w). As with other methods such as k-nearest neighbors, the choice of the smoothing parameter is a balance between under-smoothing and over-smoothing. For visualization (using a centered window), wider windows will expose more global trends, while narrow windows will reveal local trends. Hence, examining several window widths is useful for exploring trends of differing local/global nature. For forecasting (using a trailing window), the choice should incorporate domain knowledge in terms of relevance of past values and how fast the series changes. Empirical predictive evaluation can also be done by experimenting with different values of w and comparing performance. However, care should be taken not to overfit!

18.3 Simple Exponential Smoothing

A popular forecasting method in business is exponential smoothing. Its popularity derives from its flexibility, ease of automation, cheap computation, and good performance. Simple exponential smoothing is similar to forecasting with a moving average, except that instead of taking a simple average over the w most recent values, we take a *weighted average* of *all* past values, such that the weights decrease exponentially into the past. The idea is to give more weight to recent information, yet not to completely ignore older information.

Like the moving average, simple exponential smoothing should only be used for forecasting *series that have no trend or seasonality*. As mentioned earlier, such series can be obtained by removing trend and/or seasonality from raw series, and then applying exponential smoothing to the series of residuals (which are assumed to contain no trend or seasonality).

The exponential smoother generates a forecast at time $t+1$ (F_{t+1}) as follows:

$$F_{t+1} = \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2} + \dots, \tag{18.2}$$

where α is a constant between 0 and 1 called the *smoothing parameter.* The above formulation displays the exponential smoother as a weighted average of all past observations, with exponentially decaying weights.

It turns out that we can write the exponential forecaster in another way, which is very useful in practice:

$$F_{t+1} = F_t + \alpha E_t, \tag{18.3}$$

where E_t is the forecast error at time t. This formulation presents the exponential forecaster as an "active learner": It looks at the previous forecast (F_t) and how far it was from the actual value (E_t), and then corrects the next forecast based on that information. If in one period the forecast was too high, the next period is adjusted down. The amount of correction depends on the value of the smoothing parameter α. The formulation in (18.3) is also advantageous in terms of data storage and computation time: it means that we need to store and use only the forecast and forecast error from the most recent period, rather than the entire series. In applications where real-time forecasting is done, or many series are being forecasted in parallel and continuously, such savings are critical.

Note that forecasting further into the future yields the same forecast as a one-step-ahead forecast. Because the series is assumed to lack trend and seasonality, forecasts into the future rely only on information until the time of prediction. Hence, the k-step ahead forecast is equal to

$$F_{t+k} = F_{t+1}.$$

Choosing Smoothing Parameter α

The smoothing parameter α, which is set by the user, determines the rate of learning. A value close to 1 indicates fast learning (that is, only the most recent values have influence on forecasts) whereas a value close to 0 indicates slow learning (past values have a large influence on forecasts). This can be seen by plugging 0 or 1 into equation (18.2) or (18.3). Hence, the choice of α depends on the required amount of smoothing, and on how relevant the history is for generating forecasts. Default values that have been shown to work well are around 0.1–0.2. Some trial and error can also help in the choice of α: examine the time plot of the actual and predicted series, as well as the predictive accuracy (e.g., MAPE or RMSE of the validation set). Finding the α value that optimizes predictive accuracy on the validation set can be used to determine the degree of local vs. global nature of the trend. However, beware of choosing the "“best α” for forecasting purposes, as this will most likely lead to model overfitting and low predictive accuracy on future data.

To illustrate forecasting with simple exponential smoothing, we return to the residuals from the regression model, which are assumed to contain no trend or seasonality. To forecast the residual on April 2001, we apply exponential smoothing to the entire period until March 2001, and use the default $\alpha = 0.2$ value. The forecasts of this model are shown in Figure 18.4. The forecast for the residual (the horizontal broken line) is 14.143 (in thousands of riders), implying that we should adjust the regression's forecast by adding 14,143 riders from that forecast.

Relation Between Moving Average and Simple Exponential Smoothing

In both smoothing methods, the user must specify a single parameter: In moving averages, the window width (w) must be set; in exponential smoothing, the

R code for creating Figure 18.4

```
# get residuals
residuals.ts <- train.lm.trend.season$residuals

# run simple exponential smoothing
# use ets() with model = "ANN" (additive error (A), no trend (N), no seasonality (N))
# and alpha = 0.2 to fit simple exponential smoothing.
ses <- ets(residuals.ts, model = "ANN", alpha = 0.2)
ses.pred <- forecast(ses, h = nValid, level = 0)

plot(ses.pred, ylim = c(-250, 300),  ylab = "Ridership", xlab = "Time",
     bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "", flty = 2)
lines(train.lm.trend.season.pred$fitted, lwd = 2, col = "blue")
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(ses.pred$fitted, lwd = 2, col = "blue")
lines(valid.ts)
```

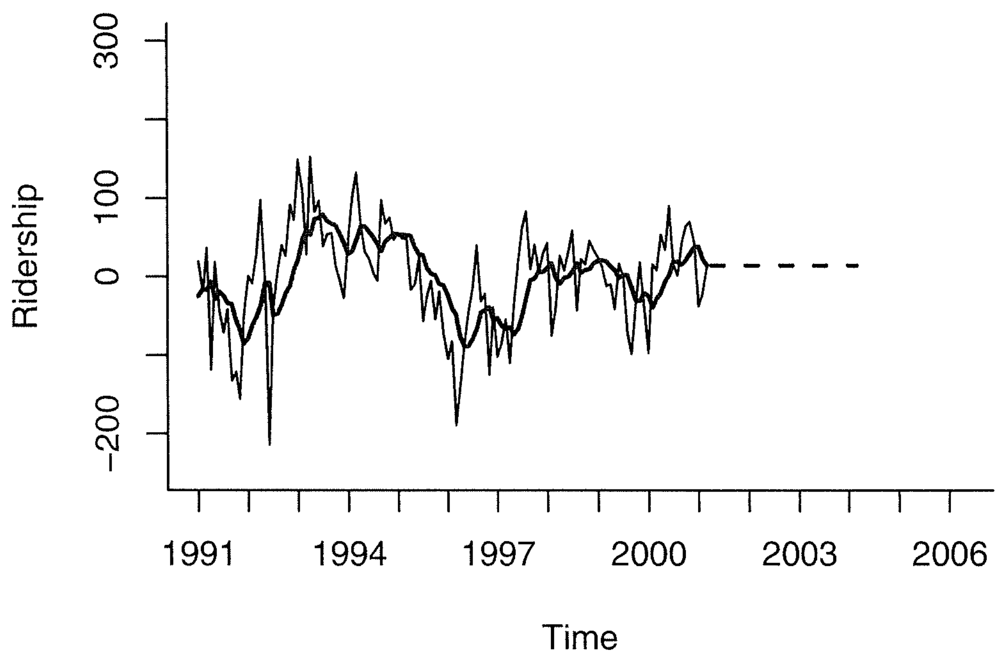

FIGURE 18.4 OUTPUT FOR SIMPLE EXPONENTIAL SMOOTHING FORECASTER WITH $\alpha = 0.2$, APPLIED TO THE SERIES OF RESIDUALS FROM THE REGRESSION MODEL (WHICH LACK TREND AND SEASONALITY). THE FORECAST VALUE IS 14.143.

smoothing parameter (α) must be set. In both cases, the parameter determines the importance of fresh information over older information. In fact, the two smoothers are approximately equal if the window width of the moving average is equal to $w = 2/\alpha - 1$.

18.4 ADVANCED EXPONENTIAL SMOOTHING

As mentioned earlier, both the moving average and simple exponential smoothing should only be used for forecasting series with no trend or seasonality; series that have only a level and noise. One solution for forecasting series with trend and/or seasonality is first to remove those components (e.g., via regression models). Another solution is to use a more sophisticated version of exponential smoothing, which can capture trend and/or seasonality.

Series with a Trend

For series that contain a trend, we can use "double exponential smoothing." Unlike in regression models, the trend shape is not assumed to be global, but rather, it can change over time. In double exponential smoothing, the local trend is estimated from the data and is updated as more data arrive. Similar to simple exponential smoothing, the level of the series is also estimated from the data, and is updated as more data arrive. The k-step-ahead forecast is given by combining the level estimate at time t (L_t) and the trend estimate at time t (T_t):

$$F_{t+k} = L_t + kT_t \tag{18.4}$$

Note that in the presence of a trend, one-, two-, three-step-ahead (etc.), forecasts are no longer identical. The level and trend are updated through a pair of updating equations:

$$L_t = \alpha Y_t + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{18.5}$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}. \tag{18.6}$$

The first equation means that the level at time t is a weighted average of the actual value at time t and the level in the previous period, adjusted for trend (in the presence of a trend, moving from one period to the next requires factoring in the trend). The second equation means that the trend at time t is a weighted average of the trend in the previous period and the more recent information on the change in level.[3] Here there are two smoothing parameters, α and β, which determine the rate of learning. As in simple exponential smoothing, they are both constants in the range [0,1], set by the user, with higher values leading to faster learning (more weight to most recent information).

[3]There are various ways to estimate the initial values L_1 and T_1, but the differences among these ways usually disappear after a few periods.

Series with a Trend and Seasonality

For series that contain both trend and seasonality, the "Holt–Winter's Exponential Smoothing" method can be used. This is a further extension of double exponential smoothing, where the k-step-ahead forecast also takes into account the seasonality at period $t + k$. Assuming seasonality with M seasons (e.g., for weekly seasonality $M = 7$), the forecast is given by

$$F_{t+k} = (L_t + kT_t)\, S_{t+k-M} \tag{18.7}$$

(Note that by the time of forecasting t, the series must have included at least one full cycle of seasons in order to produce forecasts using this formula, that is, $t > M$.)

Being an adaptive method, Holt–Winter's exponential smoothing allows the level, trend, and seasonality patterns to change over time. These three components are estimated and updated as more information arrives. The three updating equations are given by

$$L_t = \alpha Y_t / S_{t-M} + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{18.8}$$

$$T_t = \beta\,(L_t - L_{t-1}) + (1-\beta) T_{t-1} \tag{18.9}$$

$$S_t = \gamma Y_t / L_t + (1-\gamma) S_{t-M}. \tag{18.10}$$

The first equation is similar to that in double exponential smoothing, except that it uses the seasonally-adjusted value at time t rather than the raw value. This is done by dividing Y_t by its seasonal index, as estimated in the last cycle. The second equation is identical to double exponential smoothing. The third equation means that the seasonal index is updated by taking a weighted average of the seasonal index from the previous cycle and the current trend-adjusted value. Note that this formulation describes a multiplicative seasonal relationship, where values on different seasons differ by percentage amounts. There is also an additive seasonality version of Holt–Winter's exponential smoothing, where seasons differ by a constant amount (for more detail, see Shmueli and Lichtendahl, 2016).

To illustrate forecasting a series with the Holt–Winter's method, consider the raw Amtrak ridership data. As we observed earlier, the data contain both a trend and monthly seasonality. Figure 18.5 depicts the fitted and forecasted values. Table 18.2 presents a summary of the model.

Series with Seasonality (No Trend)

Finally, for series that contain seasonality but no trend, we can use a Holt–Winter's exponential smoothing formulation that lacks a trend term, by deleting the trend term in the forecasting equation and updating equations.

R code for creating Figure 18.5

```
# run Holt-Winters exponential smoothing
# use ets() with option model = "MAA" to fit Holt-Winter's exponential smoothing
# with multiplicative error, additive trend, and additive seasonality.
hwin <- ets(train.ts, model = "MAA")

# create predictions
hwin.pred <- forecast(hwin, h = nValid, level = 0)

# plot the series
plot(hwin.pred, ylim = c(1300, 2600),  ylab = "Ridership", xlab = "Time",
    bty = "l", xaxt = "n", xlim = c(1991,2006.25), main = "", flty = 2)
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(hwin.pred$fitted, lwd = 2, col = "blue")
lines(valid.ts)
```

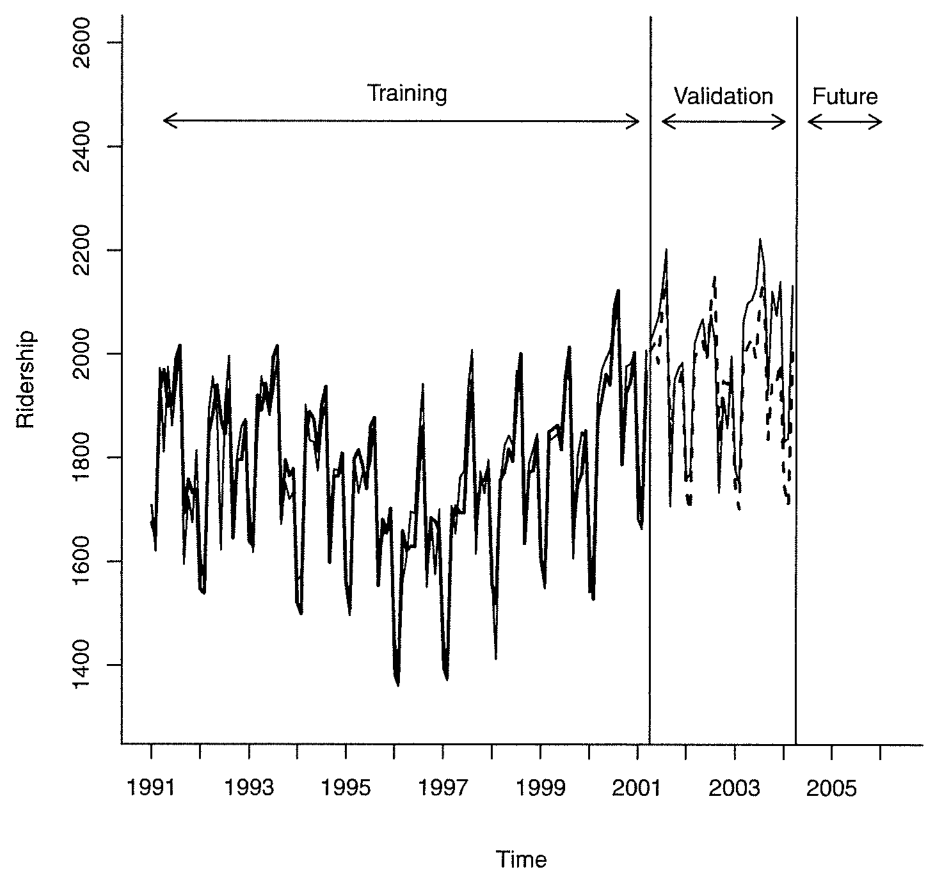

FIGURE 18.5 OUTPUT FOR HOLT-WINTERS, EXPONENTIAL SMOOTHING APPLIED TO AMTRAK RIDERSHIP SERIES

TABLE 18.2 SUMMARY OF A HOLT–WINTER'S EXPONENTIAL SMOOTHING MODEL APPLIED TO THE AMTRAK RIDERSHIP DATA. INCLUDED ARE THE INITIAL AND FINAL STATES

```
> hwin
ETS(M,A,A)
Call:
 ets(y = train.ts, model = "MAA")
  Smoothing parameters:
    alpha = 0.5483
    beta  = 1e-04
    gamma = 1e-04
  Initial states:
    l = 1881.6423
    b = 0.4164
    s=27.1143 -10.6847 -2.9465 -121.1763 201.1625 147.3359
           37.6688 75.8711 60.4021 44.4779 -252.047 -207.1783
  sigma:  0.0317
     AIC     AICc      BIC
1614.219 1619.351 1659.214
```

EXPONENTIAL SMOOTHING USING ets() IN R

In R, forecasting using exponential smoothing can be done via the *ets()* function in the `forecast` package. The three letters in *ets* stand for *error*, *trend*, and *seasonality*. Applying this function to a time series will yield forecasts and residuals for both the training and validation periods. You can use the default values for the smoothing parameters, set them to other values, or choose to find the optimal values (which optimize AIC—see Chapter 5). We also choose the type of trend, seasonality, and error. The three choices are made in the *method =* argument in the form of a three-letter combination (e.g., "MAA"). The first letter denotes the error type (A, M, or Z); the second letter denotes the trend type (N, A, M, or Z); and the third letter denotes the season type (N, A, M, or Z). In all cases, N = none, A = additive, M = multiplicative, and Z = automatically selected. For example, *method = "MAA"* indicates a multiplicative error, additive trend, and additive seasonality.

PROBLEMS

18.1 **Impact of September 11 on Air Travel in the United States.** The Research and Innovative Technology Administration's Bureau of Transportation Statistics conducted a study to evaluate the impact of the September 11, 2001 terrorist attack on US transportation. The 2006 study report and the data can be found at http://goo.gl/w2lJPV. The goal of the study was stated as follows:

> The purpose of this study is to provide a greater understanding of the passenger travel behavior patterns of persons making long distance trips before and after 9/11.

The report analyzes monthly passenger movement data between January 1990 and May 2004. Data on three monthly time series are given in file *Sept11Travel.csv* for this period: (1) Actual airline revenue passenger miles (Air), (2) Rail passenger miles (Rail), and (3) Vehicle miles traveled (Car).

In order to assess the impact of September 11, BTS took the following approach: using data before September 11, they forecasted future data (under the assumption of no terrorist attack). Then, they compared the forecasted series with the actual data to assess the impact of the event. Our first step, therefore, is to split each of the time series into two parts: pre- and post-September 11. We now concentrate only on the earlier time series.

a. Create a time plot for the pre-event AIR time series. What time series components appear from the plot?

b. Figure 18.6 shows a time plot of the **seasonally adjusted** pre-September-11 AIR series. Which of the following smoothing methods would be adequate for forecasting this series?

- Moving average (with what window width?)
- Simple exponential smoothing
- Holt exponential smoothing
- Holt–Winter's exponential smoothing

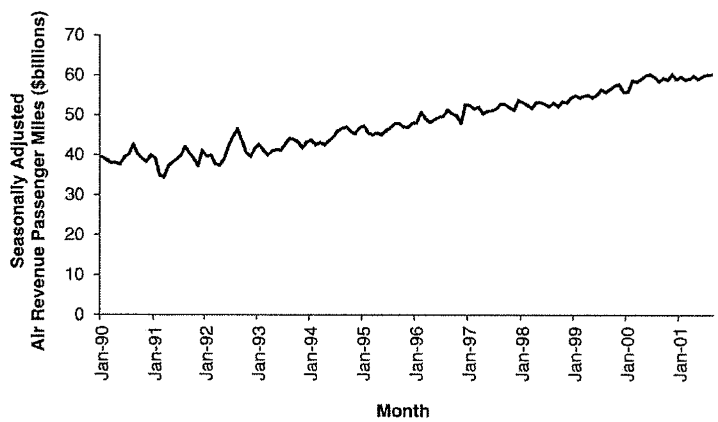

FIGURE 18.6 SEASONALLY ADJUSTED PRE-SEPTEMBER-11 AIR SERIES

18.2 Relation Between Moving Average and Exponential Smoothing. Assume that we apply a moving average to a series, using a very short window span. If we wanted to achieve an equivalent result using simple exponential smoothing, what value should the smoothing coefficient take?

18.3 Forecasting with a Moving Average. For a given time series of sales, the training set consists of 50 months. The first 5 months' data are shown below:

Month	Sales
Sept 98	27
Oct 98	31
Nov 98	58
Dec 98	63
Jan 99	59

a. Compute the sales forecast for January 1999 based on a moving average with $w = 4$.

b. Compute the forecast error for the above forecast.

18.4 Optimizing Holt–Winter's Exponential Smoothing. The table below shows the optimal smoothing constants from applying exponential smoothing to data, using automated model selection:

Level	1.000
Trend	0.000
Seasonality	0.246

a. The value of zero that is obtained for the trend smoothing constant means that (choose one of the following):

- There is no trend.
- The trend is estimated only from the first two periods.
- The trend is updated throughout the data.
- The trend is statistically insignificant.

b. What is the danger of using the optimal smoothing constant values?

18.5 Department Store Sales. The time plot in Figure 18.7 describes actual quarterly sales for a department store over a 6-year period (data are available in *DepartmentStoreSales.csv*, data courtesy of Chris Albright).

a. Which of the following methods would **not** be suitable for forecasting this series?

- Moving average of raw series
- Moving average of deseasonalized series
- Simple exponential smoothing of the raw series
- Double exponential smoothing of the raw series

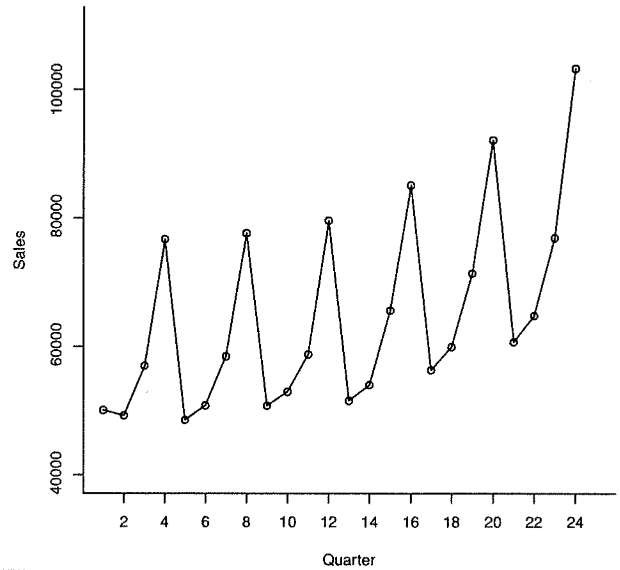

FIGURE 18.7 DEPARTMENT STORE QUARTERLY SALES SERIES

- Holt–Winter's exponential smoothing of the raw series
- Regression model fit to the raw series
- Random walk model fit to the raw series

b. The forecaster was tasked to generate forecasts for 4 quarters ahead. He therefore partitioned the data such that the last 4 quarters were designated as the validation period. The forecaster approached the forecasting task by using multiplicative Holt–Winter's exponential smoothing. The smoothing parameters used were $\alpha = 0.2, \beta = 0.15, \gamma = 0.05$.

i. Run this method on the data.

ii. The forecasts for the validation set are given in Table 18.3. Compute the MAPE values for the forecasts of quarters 21 and 22.

TABLE 18.3 FORECASTS FOR VALIDATION SERIES USING EXPONENTIAL SMOOTHING

Quarter	Actual	Forecast	Error
21	60,800	59,384.56586	1415.434145
22	64,900	61,656.49426	3243.505741
23	76,997	71,853.01442	5143.985579
24	103,337	95,074.69842	8262.301585

c. The fit and residuals from the exponential smoothing are shown in Figure 18.8. Using all the information thus far, is this model suitable for forecasting quarters 21 and 22?

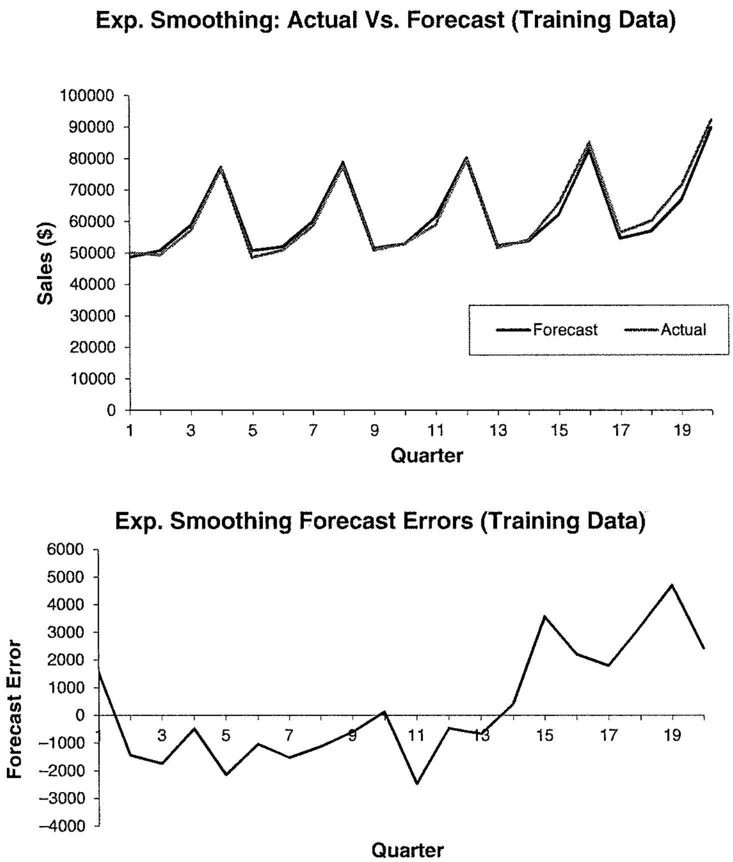

FIGURE 18.8 FORECASTS AND ACTUALS (TOP) AND FORECAST ERRORS (BOTTOM) USING EXPONENTIAL SMOOTHING

18.6 Shipments of Household Appliances. The time plot in Figure 18.9 shows the series of quarterly shipments (in million dollars) of US household appliances between 1985–1989 (data are available in *ApplianceShipments.csv*, data courtesy of Ken Black).

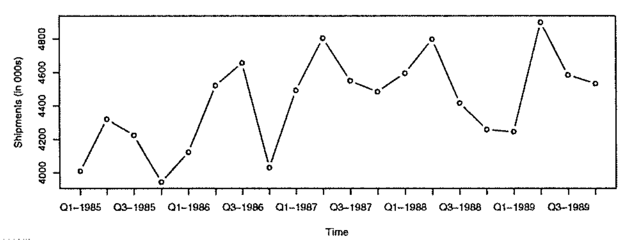

FIGURE 18.9 QUARTERLY SHIPMENTS OF US HOUSEHOLD APPLIANCES OVER 5 YEARS

a. Which of the following methods would be suitable for forecasting this series if applied to the raw data?

- Moving average
- Simple exponential smoothing
- Double exponential smoothing
- Holt–Winter's exponential smoothing

b. Apply a moving average with window span $w = 4$ to the data. Use all but the last year as the training set. Create a time plot of the moving average series.

i. What does the MA(4) chart reveal?

ii. Use the MA(4) model to forecast appliance sales in Q1-1990.

iii. Use the MA(4) model to forecast appliance sales in Q1-1991.

iv. Is the forecast for Q1-1990 most likely to under-estimate, over-estimate or accurately estimate the actual sales on Q1-1990? Explain.

v. Management feels most comfortable with moving averages. The analyst therefore plans to use this method for forecasting future quarters. What else should be considered before using the MA(4) to forecast future quarterly shipments of household appliances?

c. We now focus on forecasting beyond 1989. In the following, continue to use all but the last year as the training set, and the last four quarters as the validation set. First, fit a regression model to sales with a linear trend and quarterly seasonality to the training data. Next, apply Holt–Winter's exponential smoothing (with the default smoothing values) to the training data. Choose an adequate "season length."

i. Compute the MAPE for the validation data using the regression model.

ii. Compute the MAPE for the validation data using Holt–Winter's exponential smoothing.

iii. Which model would you prefer to use for forecasting Q1-1990? Give three reasons.

iv. If we optimize the smoothing parameters in the Holt–Winter's method, is it likely to get values that are close to zero? Why or why not?

18.7 Shampoo Sales. The time plot in Figure 18.10 describes monthly sales of a certain shampoo over a 3-year period (Data are available in *ShampooSales.csv*, source: Hyndman, R.J., Time Series Data Library, http://data.is/TSDLdemo. Accessed on 07/25/15.).

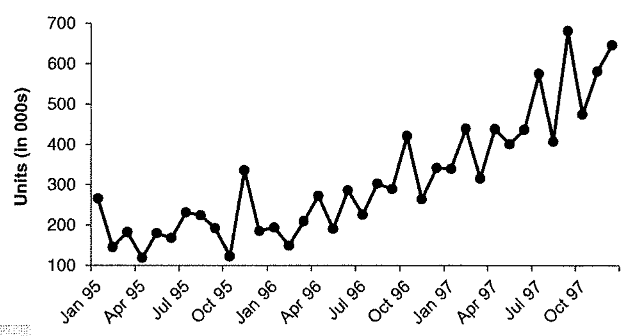

FIGURE 18.10 MONTHLY SALES OF A CERTAIN SHAMPOO

Which of the following methods would be suitable for forecasting this series if applied to the raw data?

- Moving average
- Simple exponential smoothing
- Double exponential smoothing
- Holt–Winter's exponential smoothing

18.8 **Natural Gas Sales.** Figure 18.11 is a time plot of quarterly natural gas sales (in billions of BTU) of a certain company, over a period of 4 years (data courtesy of George McCabe). The company's analyst is asked to use a moving average to forecast sales in Winter 2005.

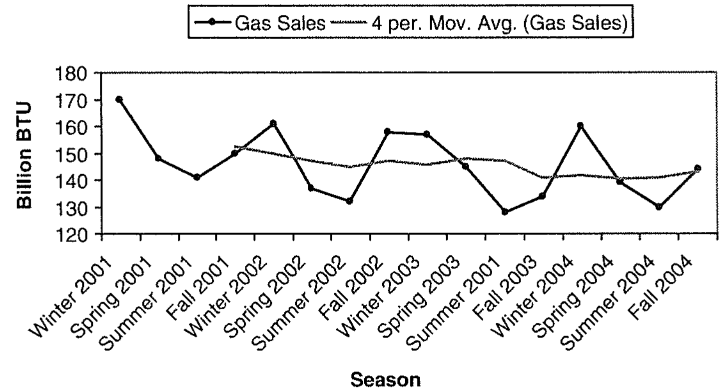

FIGURE 18.11 QUARTERLY SALES OF NATURAL GAS OVER 4 YEARS

a. Reproduce the time plot with the overlaying MA(4) line.

b. What can we learn about the series from the MA line?

c. Run a moving average forecaster with adequate season length. Are forecasts generated by this method expected to over-forecast, under-forecast, or accurately forecast actual sales? Why?

18.9 **Australian Wine Sales.** Figure 18.12 shows time plots of monthly sales of six types of Australian wines (red, rose, sweet white, dry white, sparkling, and fortified) for 1980–1994 (Data are available in *AustralianWines.csv*, source: Hyndman, R.J., Time Series Data Library, http://data.is/TSDLdemo. Accessed on 07/25/15.). The units are thousands of litres. You are hired to obtain short term forecasts (2–3 months ahead) for each of the six series, and this task will be repeated every month.

a. Which forecasting method would you choose if you had to choose the same method for all series? Why?

b. Fortified wine has the largest market share of the above six types of wine. You are asked to focus on fortified wine sales alone, and produce as accurate as possible forecasts for the next 2 months.

- Start by partitioning the data using the period until December 1993 as the training set.
- Apply Holt–Winter's exponential smoothing to sales with an appropriate season length (use the default values for the smoothing constants).

c. Create an ACF plot for the residuals from the Holt–Winter's exponential smoothing until lag 12.

i. Examining this plot, which of the following statements are reasonable conclusions?

- Decembers (month 12) are not captured well by the model.
- There is a strong correlation between sales on the same calendar month.

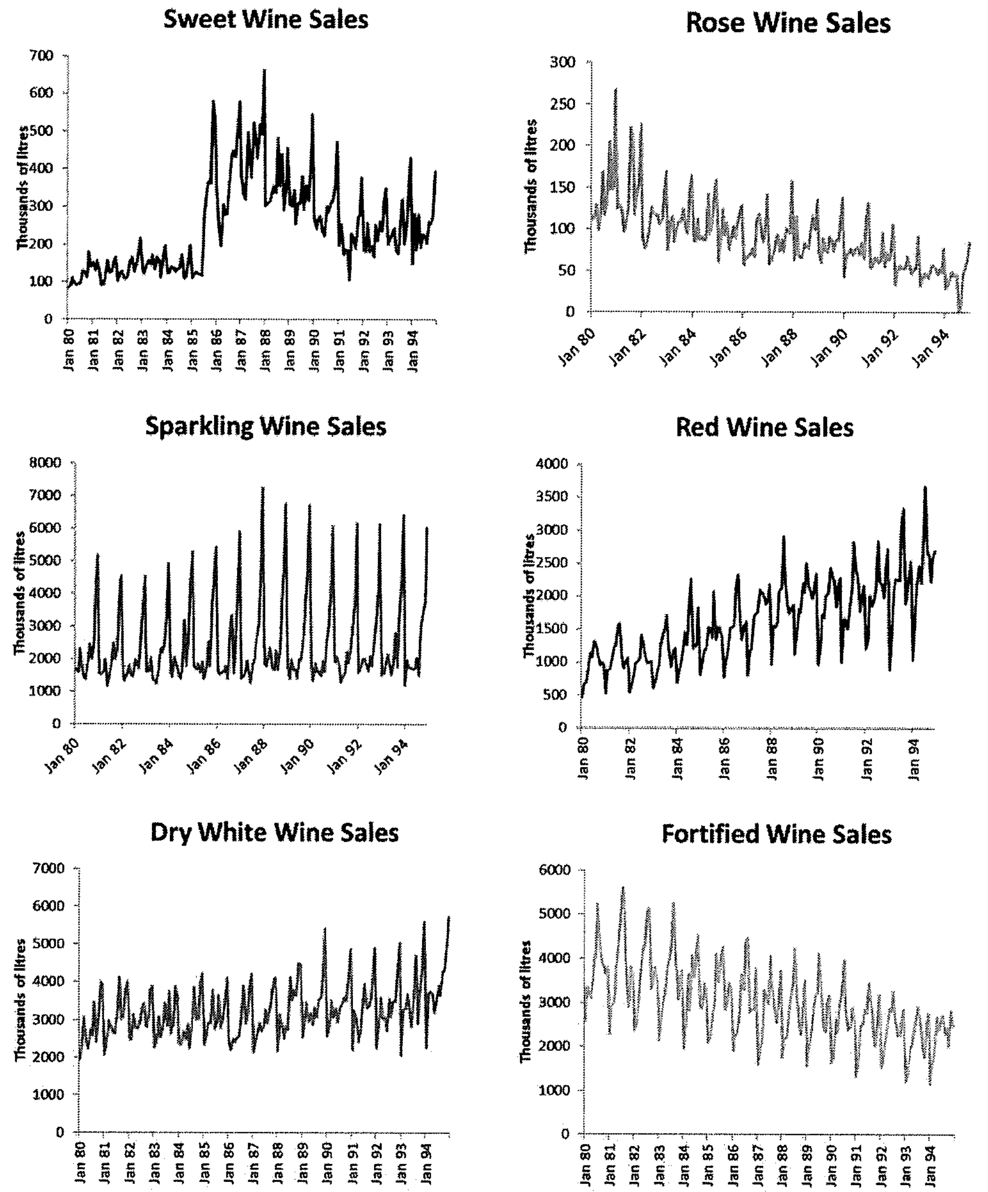

FIGURE 18.12 MONTHLY SALES OF SIX TYPES OF AUSTRALIAN WINES BETWEEN 1980 AND 1994

- The model does not capture the seasonality well.
- We should try to fit an autoregressive model with lag 12 to the residuals.
- We should first deseasonalize the data and then apply Holt–Winter's exponential smoothing.

ii. How can you handle the above effect without adding another layer to your model?

18.8 Natural Gas Sales. Figure 18.11 is a time plot of quarterly natural gas sales (in billions of BTU) of a certain company, over a period of 4 years (data courtesy of George McCabe). The company's analyst is asked to use a moving average to forecast sales in Winter 2005.

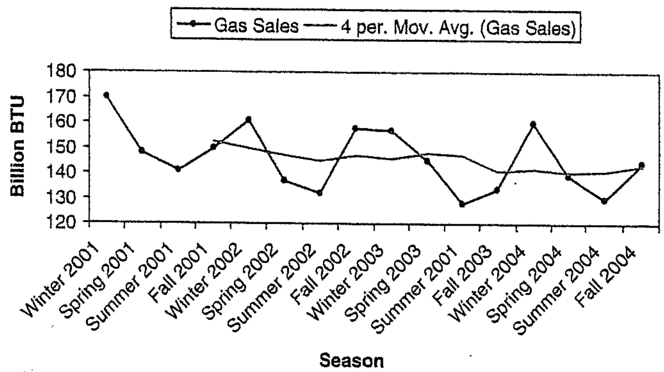

FIGURE 18.11 QUARTERLY SALES OF NATURAL GAS OVER 4 YEARS

a. Reproduce the time plot with the overlaying MA(4) line.

b. What can we learn about the series from the MA line?

c. Run a moving average forecaster with adequate season length. Are forecasts generated by this method expected to over-forecast, under-forecast, or accurately forecast actual sales? Why?

18.9 Australian Wine Sales. Figure 18.12 shows time plots of monthly sales of six types of Australian wines (red, rose, sweet white, dry white, sparkling, and fortified) for 1980–1994 (Data are available in *AustralianWines.csv*, source: Hyndman, R.J., Time Series Data Library, http://data.is/TSDLdemo. Accessed on 07/25/15.). The units are thousands of litres. You are hired to obtain short term forecasts (2–3 months ahead) for each of the six series, and this task will be repeated every month.

a. Which forecasting method would you choose if you had to choose the same method for all series? Why?

b. Fortified wine has the largest market share of the above six types of wine. You are asked to focus on fortified wine sales alone, and produce as accurate as possible forecasts for the next 2 months.

- Start by partitioning the data using the period until December 1993 as the training set.
- Apply Holt–Winter's exponential smoothing to sales with an appropriate season length (use the default values for the smoothing constants).

c. Create an ACF plot for the residuals from the Holt–Winter's exponential smoothing until lag 12.

i. Examining this plot, which of the following statements are reasonable conclusions?

- Decembers (month 12) are not captured well by the model.
- There is a strong correlation between sales on the same calendar month.

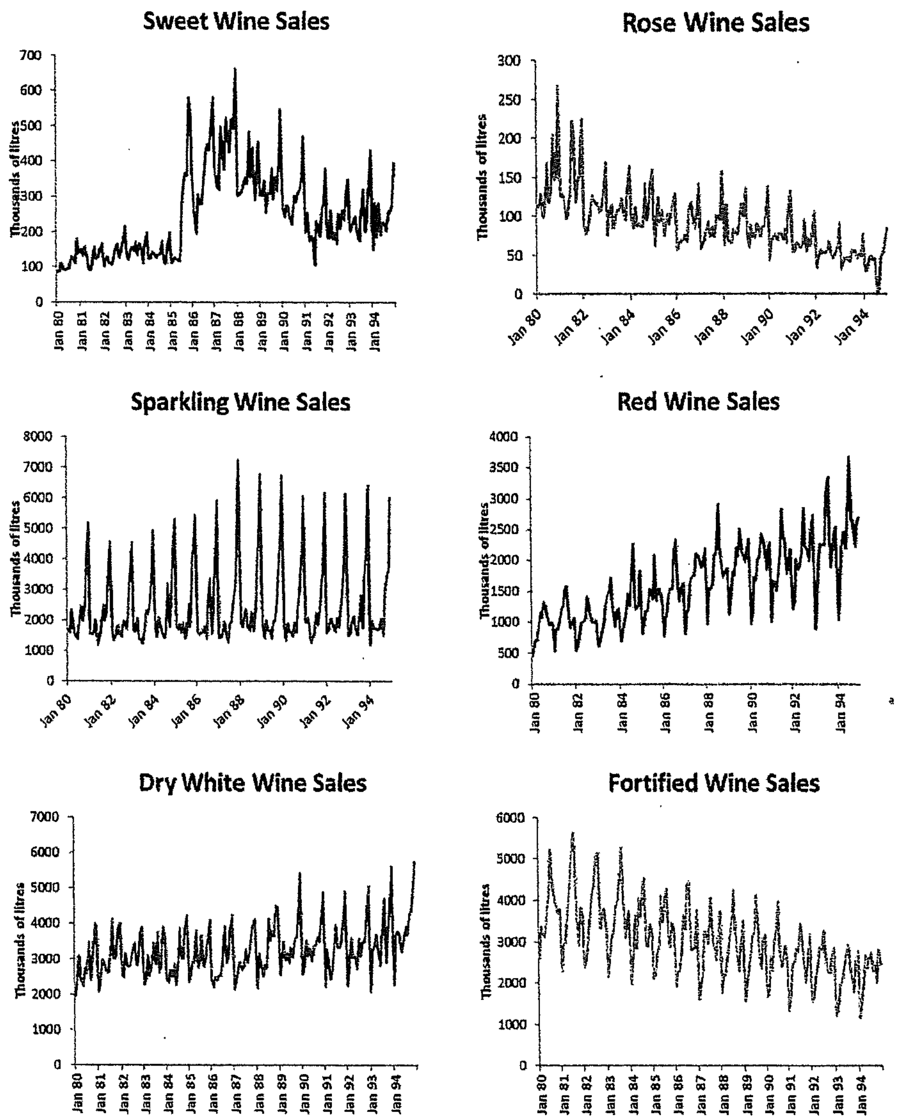

FIGURE 18.12. MONTHLY SALES OF SIX TYPES OF AUSTRALIAN WINES BETWEEN 1980 AND 1994

- The model does not capture the seasonality well.
- We should try to fit an autoregressive model with lag 12 to the residuals.
- We should first deseasonalize the data and then apply Holt–Winter's exponential smoothing.

ii. How can you handle the above effect without adding another layer to your model?