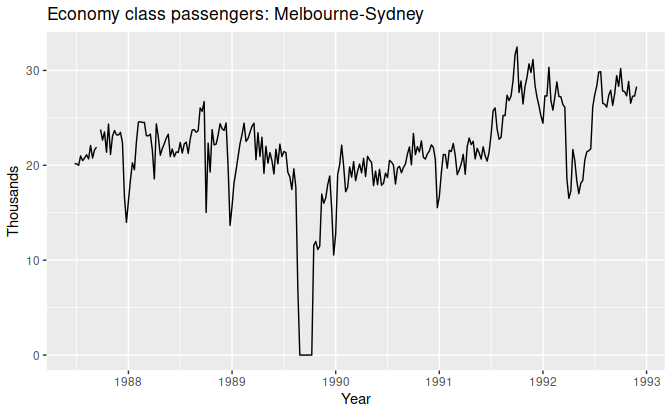
**Time plots**

For time series data, the obvious graph to start with is a time plot. That is, the observations are plotted against the time of observation, with consecutive observations joined by straight lines. Figure [2.1](https://otexts.com/fpp2/time-plots.html#fig:ansett) below shows the weekly economy passenger load on Ansett Airlines between Australia’s two largest cities.

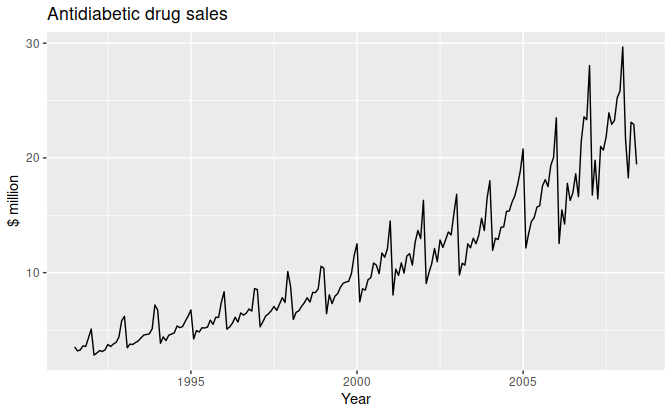


The time plot immediately reveals some interesting features.

* There was a period in 1989 when no passengers were carried — this was due to an industrial dispute.
* There was a period of reduced load in 1992. This was due to a trial in which some economy class seats were replaced by business class seats.
* A large increase in passenger load occurred in the second half of 1991.
* There are some large dips in load around the start of each year. These are due to holiday effects.
* There is a long-term fluctuation in the level of the series which increases during 1987, decreases in 1989, and increases again through 1990 and 1991.
* There are some periods of missing observations.

Any model will need to take all these features into account in order to effectively forecast the passenger load into the future.

A simpler time series is shown in Figure [2.2](https://otexts.com/fpp2/time-plots.html#fig:a10).



Here, there is a clear and increasing trend. There is also a strong seasonal pattern that increases in size as the level of the series increases. The sudden drop at the start of each year is caused by a government subsidisation scheme that makes it cost-effective for patients to stockpile drugs at the end of the calendar year. Any forecasts of this series would need to capture the seasonal pattern, and the fact that the trend is changing slowly.

**2.3 Time series patterns**

In describing these time series, we have used words such as “trend” and “seasonal” which need to be defined more carefully.

Trend

A *trend* exists when there is a long-term increase or decrease in the data. It does not have to be linear. Sometimes we will refer to a trend as “changing direction”, when it might go from an increasing trend to a decreasing trend. There is a trend in the antidiabetic drug sales data shown in Figure [2.2](https://otexts.com/fpp2/time-plots.html#fig:a10).

Seasonal

A *seasonal* pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week. Seasonality is always of a fixed and known frequency. The monthly sales of antidiabetic drugs above shows seasonality which is induced partly by the change in the cost of the drugs at the end of the calendar year.

Cyclic

A *cycle* occurs when the data exhibit rises and falls that are not of a fixed frequency. These fluctuations are usually due to economic conditions, and are often related to the “business cycle”. The duration of these fluctuations is usually at least 2 years.

Many people confuse cyclic behaviour with seasonal behaviour, but they are really quite different. If the fluctuations are not of a fixed frequency then they are cyclic; if the frequency is unchanging and associated with some aspect of the calendar, then the pattern is seasonal. In general, the average length of cycles is longer than the length of a seasonal pattern, and the magnitudes of cycles tend to be more variable than the magnitudes of seasonal patterns.

Many time series include trend, cycles and seasonality. When choosing a forecasting method, we will first need to identify the time series patterns in the data, and then choose a method that is able to capture the patterns properly.

The examples in Figure [2.3](https://otexts.com/fpp2/tspatterns.html#fig:fourexamples) show different combinations of the above components.

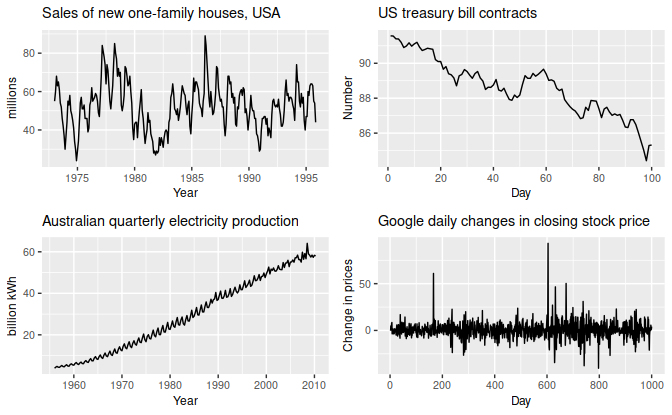


Figure 2.3: Four examples of time series showing different patterns.

1. The monthly housing sales (top left) show strong seasonality within each year, as well as some strong cyclic behaviour with a period of about 6–10 years. There is no apparent trend in the data over this period.
2. The US treasury bill contracts (top right) show results from the Chicago market for 100 consecutive trading days in 1981. Here there is no seasonality, but an obvious downward trend. Possibly, if we had a much longer series, we would see that this downward trend is actually part of a long cycle, but when viewed over only 100 days it appears to be a trend.
3. The Australian quarterly electricity production (bottom left) shows a strong increasing trend, with strong seasonality. There is no evidence of any cyclic behaviour here.
4. The daily change in the Google closing stock price (bottom right) has no trend, seasonality or cyclic behaviour. There are random fluctuations which do not appear to be very predictable, and no strong patterns that would help with developing a forecasting model.

## 2.4 Seasonal plots

A seasonal plot is similar to a time plot except that the data are plotted against the individual “seasons” in which the data were observed. An example is given below showing the antidiabetic drug sales.

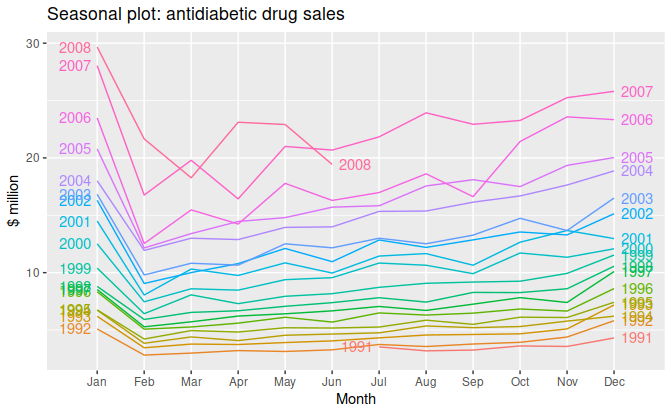


Figure 2.4: Seasonal plot of monthly antidiabetic drug sales in Australia.

These are exactly the same data as were shown earlier, but now the data from each season are overlapped. A seasonal plot allows the underlying seasonal pattern to be seen more clearly, and is especially useful in identifying years in which the pattern changes.

In this case, it is clear that there is a large jump in sales in January each year. Actually, these are probably sales in late December as customers stockpile before the end of the calendar year, but the sales are not registered with the government until a week or two later. The graph also shows that there was an unusually small number of sales in March 2008 (most other years show an increase between February and March). The small number of sales in June 2008 is probably due to incomplete counting of sales at the time the data were collected.

## 2.8 Autocorrelation

Just as correlation measures the extent of a linear relationship between two variables, autocorrelation measures the linear relationship between lagged values of a time series.

There are several autocorrelation coefficients, corresponding to each panel in the lag plot. For example, r1

measures the relationship between yt and yt−1, r2 measures the relationship between yt and yt−2

, and so on.

The value of rk

can be written as rk=T∑t=k+1(yt−¯y)(yt−k−¯y)T∑t=1(yt−¯y)2, where T

is the length of the time series.

The first nine autocorrelation coefficients for the beer production data are given in the following table.

| **r1** |
| --- |

|  | **r2** |
| --- | --- |

|  | **r3** |
| --- | --- |

|  | **r4** |
| --- | --- |

|  | **r5** |
| --- | --- |

|  | **r6** |
| --- | --- |

|  | **r7** |
| --- | --- |

|  | **r8** |
| --- | --- |

|  | **r9** |
| --- | --- |

|  |
| --- |
| -0.102 | -0.657 | -0.060 | 0.869 | -0.089 | -0.635 | -0.054 | 0.832 | -0.108 |

These correspond to the nine scatterplots in Figure [2.13](https://otexts.com/fpp2/lag-plots.html#fig:beerlagplot). The autocorrelation coefficients are plotted to show the autocorrelation function or ACF. The plot is also known as a correlogram.

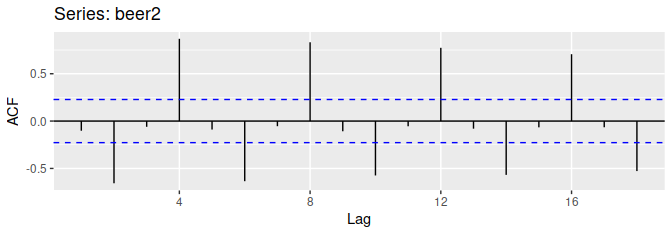


Figure 2.14: Autocorrelation function of quarterly beer production.

In this graph:

* r4

 is higher than for the other lags. This is due to the seasonal pattern in the data: the peaks tend to be four quarters apart and the troughs tend to be four quarters apart.

 r2

* is more negative than for the other lags because troughs tend to be two quarters behind peaks.
* The dashed blue lines indicate whether the correlations are significantly different from zero. These are explained in Section [2.9](https://otexts.com/fpp2/wn.html#wn).

### Trend and seasonality in ACF plots

When data have a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size. So the ACF of trended time series tend to have positive values that slowly decrease as the lags increase.

When data are seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal frequency) than for other lags.

When data are both trended and seasonal, you see a combination of these effects. The monthly Australian electricity demand series plotted in Figure [2.15](https://otexts.com/fpp2/autocorrelation.html#fig:aelec) shows both trend and seasonality. Its ACF is shown in Figure [2.16](https://otexts.com/fpp2/autocorrelation.html#fig:acfelec).

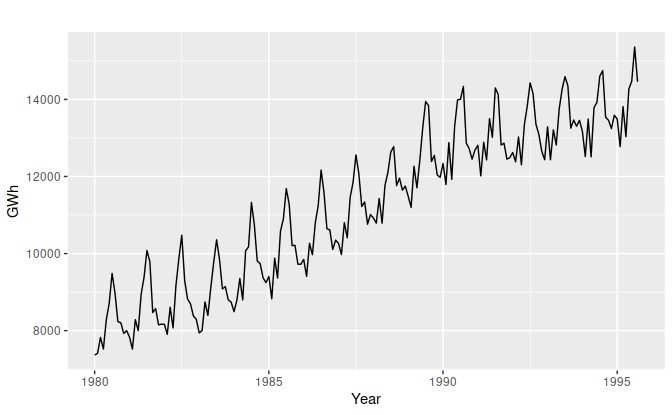


Figure 2.15: Monthly Australian electricity demand from 1980–1995.

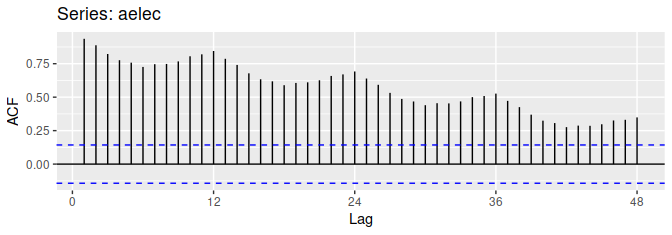


Figure 2.16: ACF of monthly Australian electricity demand.

The slow decrease in the ACF as the lags increase is due to the trend, while the “scalloped” shape is due the seasonality.

## 3.1 Some simple forecasting methods

Some forecasting methods are extremely simple and surprisingly effective. We will use the following four forecasting methods as benchmarks throughout this book.

### Average method

Here, the forecasts of all future values are equal to the average (or “mean”) of the historical data. If we let the historical data be denoted by y1,…,yT

, then we can write the forecasts as ^yT+h|T=¯y=(y1+⋯+yT)/T. The notation ^yT+h|T is a short-hand for the estimate of yT+h based on the data y1,…,yT.

### Naïve method

For naïve forecasts, we simply set all forecasts to be the value of the last observation. That is, ^yT+h|T=yT.

This method works remarkably well for many economic and financial time series.

Because a naïve forecast is optimal when data follow a random walk (see Section [8.1](https://otexts.com/fpp2/stationarity.html#stationarity)), these are also called **random walk forecasts**.

### Seasonal naïve method

A similar method is useful for highly seasonal data. In this case, we set each forecast to be equal to the last observed value from the same season of the year (e.g., the same month of the previous year). Formally, the forecast for time T+h

is written as ^yT+h|T=yT+h−m(k+1), where m= the seasonal period, and k is the integer part of (h−1)/m (i.e., the number of complete years in the forecast period prior to time T+h). This looks more complicated than it really is. For example, with monthly data, the forecast for all future February values is equal to the last observed February value. With quarterly data, the forecast of all future Q2 values is equal to the last observed Q2 value (where Q2 means the second quarter). Similar rules apply for other months and quarters, and for other seasonal periods.

### Drift method

A variation on the naïve method is to allow the forecasts to increase or decrease over time, where the amount of change over time (called the **drift**) is set to be the average change seen in the historical data. Thus the forecast for time T+h

is given by ^yT+h|T=yT+hT−1T∑t=2(yt−yt−1)=yT+h(yT−y1T−1). This is equivalent to drawing a line between the first and last observations, and extrapolating it into the future.

## 3.3 Residual diagnostics

### Fitted values

Each observation in a time series can be forecast using all previous observations. We call these **fitted values** and they are denoted by ^yt|t−1

, meaning the forecast of yt based on observations y1,…,yt−1 . We use these so often, we sometimes drop part of the subscript and just write ^yt instead of ^yt|t−1

. Fitted values always involve one-step forecasts.

Actually, fitted values are often not true forecasts because any parameters involved in the forecasting method are estimated using all available observations in the time series, including future observations. For example, if we use the average method, the fitted values are given by ^yt=^c

where ^c is the average computed over all available observations, including those at times after t. Similarly, for the drift method, the drift parameter is estimated using all available observations. In this case, the fitted values are given by ^yt=yt−1+^c where ^c=(yT−y1)/(T−1). In both cases, there is a parameter to be estimated from the data. The “hat” above the c reminds us that this is an estimate. When the estimate of c involves observations after time t

, the fitted values are not true forecasts. On the other hand, naïve or seasonal naïve forecasts do not involve any parameters, and so fitted values are true forecasts in such cases.

### Residuals

The “residuals” in a time series model are what is left over after fitting a model. For many (but not all) time series models, the residuals are equal to the difference between the observations and the corresponding fitted values: et=yt−^yt.

Residuals are useful in checking whether a model has adequately captured the information in the data. A good forecasting method will yield residuals with the following properties:

1. The residuals are uncorrelated. If there are correlations between residuals, then there is information left in the residuals which should be used in computing forecasts.
2. The residuals have zero mean. If the residuals have a mean other than zero, then the forecasts are biased.

Any forecasting method that does not satisfy these properties can be improved. However, that does not mean that forecasting methods that satisfy these properties cannot be improved. It is possible to have several different forecasting methods for the same data set, all of which satisfy these properties. Checking these properties is important in order to see whether a method is using all of the available information, but it is not a good way to select a forecasting method.

If either of these properties is not satisfied, then the forecasting method can be modified to give better forecasts. Adjusting for bias is easy: if the residuals have mean m

, then simply add m

to all forecasts and the bias problem is solved. Fixing the correlation problem is harder, and we will not address it until Chapter [9](https://otexts.com/fpp2/dynamic.html#dynamic).

In addition to these essential properties, it is useful (but not necessary) for the residuals to also have the following two properties.

1. The residuals have constant variance.
2. The residuals are normally distributed.

These two properties make the calculation of prediction intervals easier (see Section [3.5](https://otexts.com/fpp2/prediction-intervals.html#prediction-intervals) for an example). However, a forecasting method that does not satisfy these properties cannot necessarily be improved. Sometimes applying a Box-Cox transformation may assist with these properties, but otherwise there is usually little that you can do to ensure that your residuals have constant variance and a normal distribution. Instead, an alternative approach to obtaining prediction intervals is necessary. Again, we will not address how to do this until later in the book.

## 3.4 Evaluating forecast accuracy

### Training and test sets

It is important to evaluate forecast accuracy using genuine forecasts. Consequently, the size of the residuals is not a reliable indication of how large true forecast errors are likely to be. The accuracy of forecasts can only be determined by considering how well a model performs on new data that were not used when fitting the model.

When choosing models, it is common practice to separate the available data into two portions, **training** and **test** data, where the training data is used to estimate any parameters of a forecasting method and the test data is used to evaluate its accuracy. Because the test data is not used in determining the forecasts, it should provide a reliable indication of how well the model is likely to forecast on new data.



The size of the test set is typically about 20% of the total sample, although this value depends on how long the sample is and how far ahead you want to forecast. The test set should ideally be at least as large as the maximum forecast horizon required. The following points should be noted.

* A model which fits the training data well will not necessarily forecast well.
* A perfect fit can always be obtained by using a model with enough parameters.
* Over-fitting a model to data is just as bad as failing to identify a systematic pattern in the data.

Some references describe the test set as the “hold-out set” because these data are “held out” of the data used for fitting. Other references call the training set the “in-sample data” and the test set the “out-of-sample data”. We prefer to use “training data” and “test data” in this book.

### Forecast errors

A forecast “error” is the difference between an observed value and its forecast. Here “error” does not mean a mistake, it means the unpredictable part of an observation. It can be written as eT+h=yT+h−^yT+h|T,

where the training data is given by {y1,…,yT} and the test data is given by {yT+1,yT+2,…}

.

Note that forecast errors are different from residuals in two ways. First, residuals are calculated on the training set while forecast errors are calculated on the test set. Second, residuals are based on one-step forecasts while forecast errors can involve multi-step forecasts.

We can measure forecast accuracy by summarising the forecast errors in different ways.

### Scale-dependent errors

The forecast errors are on the same scale as the data. Accuracy measures that are based only on et

are therefore scale-dependent and cannot be used to make comparisons between series that involve different units.

The two most commonly used scale-dependent measures are based on the absolute errors or squared errors: Mean absolute error: MAE=mean(|et|),Root mean squared error: RMSE=√mean(e2t).

When comparing forecast methods applied to a single time series, or to several time series with the same units, the MAE is popular as it is easy to both understand and compute. A forecast method that minimises the MAE will lead to forecasts of the median, while minimising the RMSE will lead to forecasts of the mean. Consequently, the RMSE is also widely used, despite being more difficult to interpret.

### Percentage errors

The percentage error is given by pt=100et/yt

. Percentage errors have the advantage of being unit-free, and so are frequently used to compare forecast performances between data sets. The most commonly used measure is: Mean absolute percentage error: MAPE=mean(|pt|). Measures based on percentage errors have the disadvantage of being infinite or undefined if yt=0 for any t in the period of interest, and having extreme values if any yt

is close to zero. Another problem with percentage errors that is often overlooked is that they assume the unit of measurement has a meaningful zero.[2](https://otexts.com/fpp2/accuracy.html#fn2) For example, a percentage error makes no sense when measuring the accuracy of temperature forecasts on either the Fahrenheit or Celsius scales, because temperature has an arbitrary zero point.

They also have the disadvantage that they put a heavier penalty on negative errors than on positive errors. This observation led to the use of the so-called “symmetric” MAPE (sMAPE) proposed by Armstrong ([1978](https://otexts.com/fpp2/accuracy.html#ref-Armstrong85), p. 348), which was used in the M3 forecasting competition. It is defined by sMAPE=mean(200|yt−^yt|/(yt+^yt)).

However, if yt is close to zero, ^yt

is also likely to be close to zero. Thus, the measure still involves division by a number close to zero, making the calculation unstable. Also, the value of sMAPE can be negative, so it is not really a measure of “absolute percentage errors” at all.

Hyndman & Koehler ([2006](https://otexts.com/fpp2/accuracy.html#ref-HK06)) recommend that the sMAPE not be used. It is included here only because it is widely used, although we will not use it in this book.

### Scaled errors

Scaled errors were proposed by Hyndman & Koehler ([2006](https://otexts.com/fpp2/accuracy.html#ref-HK06)) as an alternative to using percentage errors when comparing forecast accuracy across series with different units. They proposed scaling the errors based on the training MAE from a simple forecast method.

For a non-seasonal time series, a useful way to define a scaled error uses naïve forecasts: qj=ej1T−1T∑t=2|yt−yt−1|.

Because the numerator and denominator both involve values on the scale of the original data, qj

is independent of the scale of the data. A scaled error is less than one if it arises from a better forecast than the average naïve forecast computed on the training data. Conversely, it is greater than one if the forecast is worse than the average naïve forecast computed on the training data.

For seasonal time series, a scaled error can be defined using seasonal naïve forecasts: qj=ej1T−mT∑t=m+1|yt−yt−m|.

The mean absolute scaled error is simply MASE=mean(|qj|).

## 5.1 The linear model

### Simple linear regression

In the simplest case, the regression model allows for a linear relationship between the forecast variable y

and a single predictor variable x: yt=β0+β1xt+εt. An artificial example of data from such a model is shown in Figure [5.1](https://otexts.com/fpp2/regression-intro.html#fig:SLRpop1). The coefficients β0 and β1 denote the intercept and the slope of the line respectively. The intercept β0 represents the predicted value of y when x=0. The slope β1 represents the average predicted change in y resulting from a one unit increase in x

### . Multiple linear regression

When there are two or more predictor variables, the model is called a **multiple regression model**. The general form of a multiple regression model is yt=β0+β1x1,t+β2x2,t+⋯+βkxk,t+εt,(5.1)

where y is the variable to be forecast and x1,…,xk are the k predictor variables. Each of the predictor variables must be numerical. The coefficients β1,…,βk measure the effect of each predictor after taking into account the effects of all the other predictors in the model. Thus, the coefficients measure the marginal effects of the predictor variables.

### Assumptions

When we use a linear regression model, we are implicitly making some assumptions about the variables in Equation [(5.1)](https://otexts.com/fpp2/regression-intro.html#eq:lm).

First, we assume that the model is a reasonable approximation to reality; that is, the relationship between the forecast variable and the predictor variables satisfies this linear equation.

Second, we make the following assumptions about the errors (ε1,…,εT)

:

* they have mean zero; otherwise the forecasts will be systematically biased.
* they are not autocorrelated; otherwise the forecasts will be inefficient, as there is more information in the data that can be exploited.
* they are unrelated to the predictor variables; otherwise there would be more information that should be included in the systematic part of the model.

It is also useful to have the errors being normally distributed with a constant variance σ2

in order to easily produce prediction intervals.

Another important assumption in the linear regression model is that each predictor x

is not a random variable. If we were performing a controlled experiment in a laboratory, we could control the values of each x (so they would not be random) and observe the resulting values of y. With observational data (including most data in business and economics), it is not possible to control the value of x, we simply observe it. Hence we make this an assumption.

## 5.2 Least squares estimation

In practice, of course, we have a collection of observations but we do not know the values of the coefficients β0,β1,…,βk

. These need to be estimated from the data.

The least squares principle provides a way of choosing the coefficients effectively by minimising the sum of the squared errors. That is, we choose the values of β0,β1,…,βk

that minimise T∑t=1ε2t=T∑t=1(yt−β0−β1x1,t−β2x2,t−⋯−βkxk,t)2.

This is called **least squares** estimation because it gives the least value for the sum of squared errors. Finding the best estimates of the coefficients is often called “fitting” the model to the data, or sometimes “learning” or “training” the model. The line shown in Figure [5.3](https://otexts.com/fpp2/regression-intro.html#fig:ConsInc2) was obtained in this way.

When we refer to the estimated coefficients, we will use the notation ^β0,…,^βk

. The equations for these will be given in Section [5.7](https://otexts.com/fpp2/regression-matrices.html#regression-matrices).

The tslm() function fits a linear regression model to time series data. It is similar to the lm() function which is widely used for linear models, but tslm() provides additional facilities for handling time series.

### Fitted values

Predictions of y

can be obtained by using the estimated coefficients in the regression equation and setting the error term to zero. In general we write, ^yt=^β0+^β1x1,t+^β2x2,t+⋯+^βkxk,t.(5.2) Plugging in the values of x1,t,…,xk,t for t=1,…,T returns predictions of yt within the training-sample, referred to as fitted values. Note that these are predictions of the data used to estimate the model, not genuine forecasts of future values of y.

### Goodness-of-fit

A common way to summarise how well a linear regression model fits the data is via the coefficient of determination, or R2

. This can be calculated as the square of the correlation between the observed y values and the predicted ^y values. Alternatively, it can also be calculated as, R2=∑(^yt−¯y)2∑(yt−¯y)2,

where the summations are over all observations. Thus, it reflects the proportion of variation in the forecast variable that is accounted for (or explained) by the regression model.

In simple linear regression, the value of R2

is also equal to the square of the correlation between y and x

(provided an intercept has been included).

If the predictions are close to the actual values, we would expect R2

to be close to 1. On the other hand, if the predictions are unrelated to the actual values, then R2=0 (again, assuming there is an intercept). In all cases, R2

lies between 0 and 1.

The R2

value is used frequently, though often incorrectly, in forecasting. The value of R2 will never decrease when adding an extra predictor to the model and this can lead to over-fitting. There are no set rules for what is a good R2 value, and typical values of R2 depend on the type of data used. Validating a model’s forecasting performance on the test data is much better than measuring the R2 value on the training data.

### Standard error of the regression

Another measure of how well the model has fitted the data is the standard deviation of the residuals, which is often known as the “residual standard error”. This is shown in the above output with the value 0.329.

It is calculated using ^σe= ⎷1T−k−1T∑t=1e2t,(5.3)

where k is the number of predictors in the model. Notice that we divide by T−k−1 because we have estimated k+1

parameters (the intercept and a coefficient for each predictor variable) in computing the residuals.

The standard error is related to the size of the average error that the model produces. We can compare this error to the sample mean of y

or with the standard deviation of y

to gain some perspective on the accuracy of the model.

The standard error will be used when generating prediction intervals, discussed in Section [5.6](https://otexts.com/fpp2/forecasting-regression.html#forecasting-regression).

## 5.3 Evaluating the regression model

The differences between the observed y

values and the corresponding fitted ^y values are the training-set errors or “residuals” defined as, et=yt−^yt=yt−^β0−^β1x1,t−^β2x2,t−⋯−^βkxk,t for t=1,…,T

. Each residual is the unpredictable component of the associated observation.

The residuals have some useful properties including the following two: T∑t=1et=0andT∑t=1xk,tet=0for all k.

As a result of these properties, it is clear that the average of the residuals is zero, and that the correlation between the residuals and the observations for the predictor variable is also zero. (This is not necessarily true when the intercept is omitted from the model.)

After selecting the regression variables and fitting a regression model, it is necessary to plot the residuals to check that the assumptions of the model have been satisfied. There are a series of plots that should be produced in order to check different aspects of the fitted model and the underlying assumptions. We will now discuss each of them in turn.

### ACF plot of residuals

With time series data, it is highly likely that the value of a variable observed in the current time period will be similar to its value in the previous period, or even the period before that, and so on. Therefore when fitting a regression model to time series data, it is common to find autocorrelation in the residuals. In this case, the estimated model violates the assumption of no autocorrelation in the errors, and our forecasts may be inefficient — there is some information left over which should be accounted for in the model in order to obtain better forecasts. The forecasts from a model with autocorrelated errors are still unbiased, and so are not “wrong”, but they will usually have larger prediction intervals than they need to. Therefore we should always look at an ACF plot of the residuals.

Another useful test of autocorrelation in the residuals designed to take account for the regression model is the **Breusch-Godfrey** test, also referred to as the LM (Lagrange Multiplier) test for serial correlation. It is used to test the joint hypothesis that there is no autocorrelation in the residuals up to a certain specified order. A small p-value indicates there is significant autocorrelation remaining in the residuals.

The Breusch-Godfrey test is similar to the Ljung-Box test, but it is specifically designed for use with regression models.

### Histogram of residuals

It is always a good idea to check whether the residuals are normally distributed. As we explained earlier, this is not essential for forecasting, but it does make the calculation of prediction intervals much easier.

### Outliers and influential observations

Observations that take extreme values compared to the majority of the data are called **outliers**. Observations that have a large influence on the estimated coefficients of a regression model are called **influential observations**. Usually, influential observations are also outliers that are extreme in the x

direction.

There are formal methods for detecting outliers and influential observations that are beyond the scope of this textbook. As we suggested at the beginning of Chapter [2](https://otexts.com/fpp2/graphics.html#graphics), becoming familiar with your data prior to performing any analysis is of vital importance. A scatter plot of y

against each x

is always a useful starting point in regression analysis, and often helps to identify unusual observations.

One source of outliers is incorrect data entry. Simple descriptive statistics of your data can identify minima and maxima that are not sensible. If such an observation is identified, and it has been recorded incorrectly, it should be corrected or removed from the sample immediately.

Outliers also occur when some observations are simply different. In this case it may not be wise for these observations to be removed. If an observation has been identified as a likely outlier, it is important to study it and analyse the possible reasons behind it. The decision to remove or retain an observation can be a challenging one (especially when outliers are influential observations). It is wise to report results both with and without the removal of such observations.

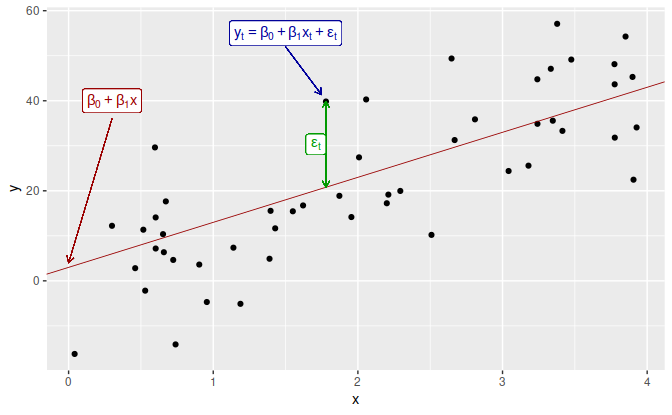


Figure 5.1: An example of data from a simple linear regression model.

Notice that the observations do not lie on the straight line but are scattered around it. We can think of each observation yt

as consisting of the systematic or explained part of the model, β0+β1xt, and the random “error”, εt. The “error” term does not imply a mistake, but a deviation from the underlying straight line model. It captures anything that may affect yt other than xt.

# Chapter 6 Time series decomposition

Time series data can exhibit a variety of patterns, and it is often helpful to split a time series into several components, each representing an underlying pattern category.

In Section [2.3](https://otexts.com/fpp2/tspatterns.html#tspatterns) we discussed three types of time series patterns: trend, seasonality and cycles. When we decompose a time series into components, we usually combine the trend and cycle into a single **trend-cycle** component (sometimes called the **trend** for simplicity). Thus we think of a time series as comprising three components: a trend-cycle component, a seasonal component, and a remainder component (containing anything else in the time series).

In this chapter, we consider some common methods for extracting these components from a time series. Often this is done to help improve understanding of the time series, but it can also be used to improve forecast accuracy.

## 6.1 Time series components

If we assume an additive decomposition, then we can write yt=St+Tt+Rt,

where yt is the data, St is the seasonal component, Tt is the trend-cycle component, and Rt is the remainder component, all at period t. Alternatively, a multiplicative decomposition would be written as yt=St×Tt×Rt.

The additive decomposition is the most appropriate if the magnitude of the seasonal fluctuations, or the variation around the trend-cycle, does not vary with the level of the time series. When the variation in the seasonal pattern, or the variation around the trend-cycle, appears to be proportional to the level of the time series, then a multiplicative decomposition is more appropriate. Multiplicative decompositions are common with economic time series.

An alternative to using a multiplicative decomposition is to first transform the data until the variation in the series appears to be stable over time, then use an additive decomposition. When a log transformation has been used, this is equivalent to using a multiplicative decomposition because yt=St×Tt×Rtis equivalent tologyt=logSt+logTt+logRt.

### Electrical equipment manufacturing

We will look at several methods for obtaining the components St

, Tt and Rt

later in this chapter, but first, it is helpful to see an example. We will decompose the new orders index for electrical equipment shown in Figure [6.1](https://otexts.com/fpp2/components.html#fig:elecequip-trend). The data show the number of new orders for electrical equipment (computer, electronic and optical products) in the Euro area (16 countries). The data have been adjusted by working days and normalised so that a value of 100 corresponds to 2005.

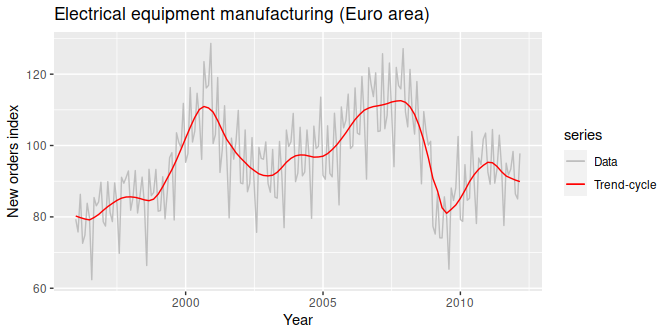


Figure 6.1: Electrical equipment orders: the trend-cycle component (red) and the raw data (grey).

Figure [6.1](https://otexts.com/fpp2/components.html#fig:elecequip-trend) shows the trend-cycle component, Tt

, in red and the original data, yt

, in grey. The trend-cycle shows the overall movement in the series, ignoring the seasonality and any small random fluctuations.

Figure [6.2](https://otexts.com/fpp2/components.html#fig:elecequip-stl) shows an additive decomposition of these data. The method used for estimating components in this example is STL, which is discussed in Section [6.6](https://otexts.com/fpp2/stl.html#stl).

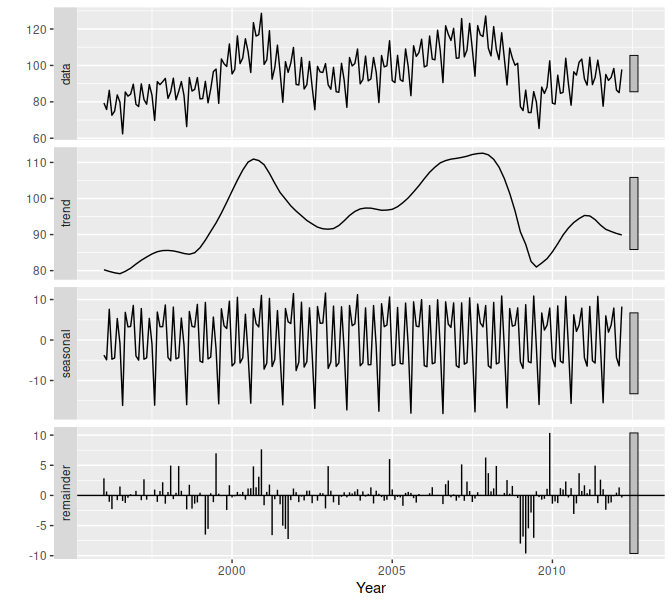


Figure 6.2: The electrical equipment orders (top) and its three additive components.

The three components are shown separately in the bottom three panels of Figure [6.2](https://otexts.com/fpp2/components.html#fig:elecequip-stl). These components can be added together to reconstruct the data shown in the top panel. Notice that the seasonal component changes slowly over time, so that any two consecutive years have similar patterns, but years far apart may have different seasonal patterns. The remainder component shown in the bottom panel is what is left over when the seasonal and trend-cycle components have been subtracted from the data.

The grey bars to the right of each panel show the relative scales of the components. Each grey bar represents the same length but because the plots are on different scales, the bars vary in size. The large grey bar in the bottom panel shows that the variation in the remainder component is small compared to the variation in the data, which has a bar about one quarter the size. If we shrunk the bottom three panels until their bars became the same size as that in the data panel, then all the panels would be on the same scale.

### Seasonally adjusted data

If the seasonal component is removed from the original data, the resulting values are the “seasonally adjusted” data. For an additive decomposition, the seasonally adjusted data are given by yt−St

, and for multiplicative data, the seasonally adjusted values are obtained using yt/St

.

Figure [6.3](https://otexts.com/fpp2/components.html#fig:elecequip-sa) shows the seasonally adjusted electrical equipment orders.

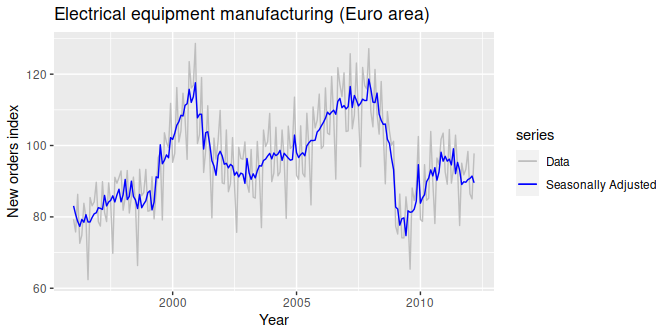


Figure 6.3: Seasonally adjusted electrical equipment orders (blue) and the original data (grey).

If the variation due to seasonality is not of primary interest, the seasonally adjusted series can be useful. For example, monthly unemployment data are usually seasonally adjusted in order to highlight variation due to the underlying state of the economy rather than the seasonal variation. An increase in unemployment due to school leavers seeking work is seasonal variation, while an increase in unemployment due to an economic recession is non-seasonal. Most economic analysts who study unemployment data are more interested in the non-seasonal variation. Consequently, employment data (and many other economic series) are usually seasonally adjusted.

Seasonally adjusted series contain the remainder component as well as the trend-cycle. Therefore, they are not “smooth”, and “downturns” or “upturns” can be misleading. If the purpose is to look for turning points in a series, and interpret any changes in direction, then it is better to use the trend-cycle component rather than the seasonally adjusted data.

## 6.2 Moving averages

The classical method of time series decomposition originated in the 1920s and was widely used until the 1950s. It still forms the basis of many time series decomposition methods, so it is important to understand how it works. The first step in a classical decomposition is to use a moving average method to estimate the trend-cycle, so we begin by discussing moving averages.

### Moving average smoothing

A moving average of order m

can be written as ^Tt=1mk∑j=−kyt+j,(6.1) where m=2k+1. That is, the estimate of the trend-cycle at time t is obtained by averaging values of the time series within k periods of t. Observations that are nearby in time are also likely to be close in value. Therefore, the average eliminates some of the randomness in the data, leaving a smooth trend-cycle component. We call this an **m-MA**, meaning a moving average of order m.

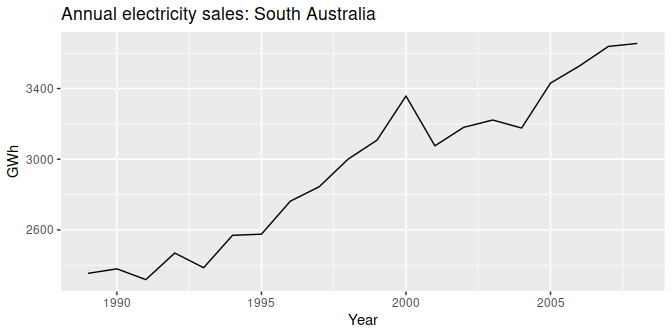


Figure 6.4: Residential electricity sales (excluding hot water) for South Australia: 1989–2008.

For example, consider Figure [6.4](https://otexts.com/fpp2/moving-averages.html#fig:ressales1) which shows the volume of electricity sold to residential customers in South Australia each year from 1989 to 2008 (hot water sales have been excluded). The data are also shown in Table [6.1](https://otexts.com/fpp2/moving-averages.html#tab:elecsales).

In the last column of this table, a moving average of order 5 is shown, providing an estimate of the trend-cycle. The first value in this column is the average of the first five observations (1989–1993); the second value in the 5-MA column is the average of the values for 1990–1994; and so on. Each value in the 5-MA column is the average of the observations in the five year window centred on the corresponding year. In the notation of Equation [(6.1)](https://otexts.com/fpp2/moving-averages.html#eq:ma), column 5-MA contains the values of ^Tt with k=2 and m=2k+1=5. This is easily computed using

There are no values for either the first two years or the last two years, because we do not have two observations on either side. Later we will use more sophisticated methods of trend-cycle estimation which do allow estimates near the endpoints.

### Estimating the trend-cycle with seasonal data

The most common use of centred moving averages is for estimating the trend-cycle from seasonal data. Consider the 2×4

-MA: ^Tt=18yt−2+14yt−1+14yt+14yt+1+18yt+2. When applied to quarterly data, each quarter of the year is given equal weight as the first and last terms apply to the same quarter in consecutive years. Consequently, the seasonal variation will be averaged out and the resulting values of ^Tt will have little or no seasonal variation remaining. A similar effect would be obtained using a 2×8-MA or a 2×12

-MA to quarterly data.

In general, a 2×m

-MA is equivalent to a weighted moving average of order m+1 where all observations take the weight 1/m, except for the first and last terms which take weights 1/(2m). So, if the seasonal period is even and of order m, we use a 2×m-MA to estimate the trend-cycle. If the seasonal period is odd and of order m, we use a m-MA to estimate the trend-cycle. For example, a 2×12

-MA can be used to estimate the trend-cycle of monthly data and a 7-MA can be used to estimate the trend-cycle of daily data with a weekly seasonality.

Other choices for the order of the MA will usually result in trend-cycle estimates being contaminated by the seasonality in the data.

### Weighted moving averages

Combinations of moving averages result in weighted moving averages. For example, the 2×4

-MA discussed above is equivalent to a weighted 5-MA with weights given by [18,14,14,14,18]. In general, a weighted m-MA can be written as ^Tt=k∑j=−kajyt+j, where k=(m−1)/2, and the weights are given by [a−k,…,ak]. It is important that the weights all sum to one and that they are symmetric so that aj=a−j. The simple m-MA is a special case where all of the weights are equal to 1/m

.

A major advantage of weighted moving averages is that they yield a smoother estimate of the trend-cycle. Instead of observations entering and leaving the calculation at full weight, their weights slowly increase and then slowly decrease, resulting in a smoother curve.

# Chapter 7 Exponential smoothing

Exponential smoothing was proposed in the late 1950s (Brown, [1959](https://otexts.com/fpp2/expsmooth.html#ref-Brown59); Holt, [1957](https://otexts.com/fpp2/expsmooth.html#ref-Holt57); Winters, [1960](https://otexts.com/fpp2/expsmooth.html#ref-Winters60)), and has motivated some of the most successful forecasting methods. Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older. In other words, the more recent the observation the higher the associated weight. This framework generates reliable forecasts quickly and for a wide range of time series, which is a great advantage and of major importance to applications in industry.

This chapter is divided into two parts. In the first part (Sections [7.1](https://otexts.com/fpp2/ses.html#ses)–[7.4](https://otexts.com/fpp2/taxonomy.html#taxonomy)) we present the mechanics of the most important exponential smoothing methods, and their application in forecasting time series with various characteristics. This helps us develop an intuition to how these methods work. In this setting, selecting and using a forecasting method may appear to be somewhat ad hoc. The selection of the method is generally based on recognising key components of the time series (trend and seasonal) and the way in which these enter the smoothing method (e.g., in an additive, damped or multiplicative manner).

In the second part of the chapter (Sections [7.5](https://otexts.com/fpp2/ets.html#ets)–[7.7](https://otexts.com/fpp2/ets-forecasting.html#ets-forecasting)) we present the statistical models that underlie exponential smoothing methods. These models generate identical point forecasts to the methods discussed in the first part of the chapter, but also generate prediction intervals. Furthermore, this statistical framework allows for genuine model selection between competing models.

## 7.1 Simple exponential smoothing

The simplest of the exponentially smoothing methods is naturally called **simple exponential smoothing** (SES)[13](https://otexts.com/fpp2/ses.html#fn13). This method is suitable for forecasting data with no clear trend or seasonal pattern. For example, the data in Figure [7.1](https://otexts.com/fpp2/ses.html#fig:7-oil) do not display any clear trending behaviour or any seasonality. (There is a rise in the last few years, which might suggest a trend. We will consider whether a trended method would be better for this series later in this chapter.) We have already considered the naïve and the average as possible methods for forecasting such data (Section [3.1](https://otexts.com/fpp2/simple-methods.html#simple-methods)).

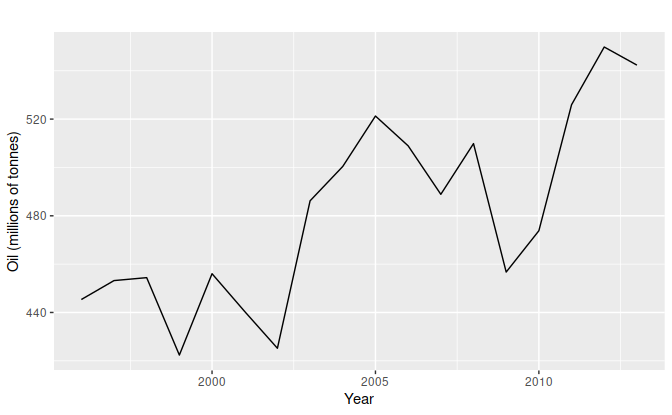


Figure 7.1: Oil production in Saudi Arabia from 1996 to 2013.

Using the naïve method, all forecasts for the future are equal to the last observed value of the series, ^yT+h|T=yT,

for h=1,2,…

. Hence, the naïve method assumes that the most recent observation is the only important one, and all previous observations provide no information for the future. This can be thought of as a weighted average where all of the weight is given to the last observation.

Using the average method, all future forecasts are equal to a simple average of the observed data, ^yT+h|T=1TT∑t=1yt,

for h=1,2,…

. Hence, the average method assumes that all observations are of equal importance, and gives them equal weights when generating forecasts.

We often want something between these two extremes. For example, it may be sensible to attach larger weights to more recent observations than to observations from the distant past. This is exactly the concept behind simple exponential smoothing. Forecasts are calculated using weighted averages, where the weights decrease exponentially as observations come from further in the past — the smallest weights are associated with the oldest observations: ^yT+1|T=αyT+α(1−α)yT−1+α(1−α)2yT−2+⋯,(7.1)

where 0≤α≤1 is the smoothing parameter. The one-step-ahead forecast for time T+1 is a weighted average of all of the observations in the series y1,…,yT. The rate at which the weights decrease is controlled by the parameter α.

### Weighted average form

The forecast at time T+1

is equal to a weighted average between the most recent observation yT and the previous forecast ^yT|T−1: ^yT+1|t=αyT+(1−α)^yT|T−1, where 0≤α≤1 is the smoothing parameter. Similarly, we can write the fitted values as ^yt+1|t=αyt+(1−α)^yt|t−1, for t=1,…,T

. (Recall that fitted values are simply one-step forecasts of the training data.)

The process has to start somewhere, so we let the first fitted value at time 1 be denoted by ℓ0

(which we will have to estimate). Then ^y2|1=αy1+(1−α)ℓ0^y3|2=αy2+(1−α)^y2|1^y4|3=αy3+(1−α)^y3|2⋮^yT|T−1=αyT−1+(1−α)^yT−1|T−2^yT+1|T=αyT+(1−α)^yT|T−1. Substituting each equation into the following equation, we obtain ^y3|2=αy2+(1−α)[αy1+(1−α)ℓ0]=αy2+α(1−α)y1+(1−α)2ℓ0^y4|3=αy3+(1−α)[αy2+α(1−α)y1+(1−α)2ℓ0]=αy3+α(1−α)y2+α(1−α)2y1+(1−α)3ℓ0  ⋮^yT+1|T=T−1∑j=0α(1−α)jyT−j+(1−α)Tℓ0. The last term becomes tiny for large T

. So, the weighted average form leads to the same forecast Equation [(7.1)](https://otexts.com/fpp2/ses.html#eq:7-ses).

### Component form

An alternative representation is the component form. For simple exponential smoothing, the only component included is the level, ℓt

. (Other methods which are considered later in this chapter may also include a trend bt and a seasonal component st.) Component form representations of exponential smoothing methods comprise a forecast equation and a smoothing equation for each of the components included in the method. The component form of simple exponential smoothing is given by: Forecast equation^yt+h|t=ℓtSmoothing equationℓt=αyt+(1−α)ℓt−1, where ℓt is the level (or the smoothed value) of the series at time t. Setting h=1 gives the fitted values, while setting t=T

gives the true forecasts beyond the training data.

The forecast equation shows that the forecast value at time t+1

is the estimated level at time t. The smoothing equation for the level (usually referred to as the level equation) gives the estimated level of the series at each period t

.

If we replace ℓt

with ^yt+1|t and ℓt−1 with ^yt|t−1

in the smoothing equation, we will recover the weighted average form of simple exponential smoothing.

The component form of simple exponential smoothing is not particularly useful, but it will be the easiest form to use when we start adding other components.

### Flat forecasts

Simple exponential smoothing has a “flat” forecast function: ^yT+h|T=^yT+1|T=ℓT,h=2,3,….

That is, all forecasts take the same value, equal to the last level component. Remember that these forecasts will only be suitable if the time series has no trend or seasonal component.

### Optimisation

The application of every exponential smoothing method requires the smoothing parameters and the initial values to be chosen. In particular, for simple exponential smoothing, we need to select the values of α

and ℓ0

. All forecasts can be computed from the data once we know those values. For the methods that follow there is usually more than one smoothing parameter and more than one initial component to be chosen.

In some cases, the smoothing parameters may be chosen in a subjective manner — the forecaster specifies the value of the smoothing parameters based on previous experience. However, a more reliable and objective way to obtain values for the unknown parameters is to estimate them from the observed data.

In Section [5.2](https://otexts.com/fpp2/least-squares.html#least-squares), we estimated the coefficients of a regression model by minimising the sum of the squared residuals (usually known as SSE or “sum of squared errors”). Similarly, the unknown parameters and the initial values for any exponential smoothing method can be estimated by minimising the SSE. The residuals are specified as et=yt−^yt|t−1

for t=1,…,T. Hence, we find the values of the unknown parameters and the initial values that minimise SSE=T∑t=1(yt−^yt|t−1)2=T∑t=1e2t.(7.2)

Unlike the regression case (where we have formulas which return the values of the regression coefficients that minimise the SSE), this involves a non-linear minimisation problem, and we need to use an optimisation tool to solve it.

## 7.2 Trend methods

### Holt’s linear trend method

Holt ([1957](https://otexts.com/fpp2/holt.html#ref-Holt57)) extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend): Forecast equation^yt+h|t=ℓt+hbtLevel equationℓt=αyt+(1−α)(ℓt−1+bt−1)Trend equationbt=β∗(ℓt−ℓt−1)+(1−β∗)bt−1,

where ℓt denotes an estimate of the level of the series at time t, bt denotes an estimate of the trend (slope) of the series at time t, α is the smoothing parameter for the level, 0≤α≤1, and β∗ is the smoothing parameter for the trend, 0≤β∗≤1. (We denote this as β∗ instead of β

for reasons that will be explained in Section [7.5](https://otexts.com/fpp2/ets.html#ets).)

As with simple exponential smoothing, the level equation here shows that ℓt

is a weighted average of observation yt and the one-step-ahead training forecast for time t, here given by ℓt−1+bt−1. The trend equation shows that bt is a weighted average of the estimated trend at time t based on ℓt−ℓt−1 and bt−1

, the previous estimate of the trend.

The forecast function is no longer flat but trending. The h

-step-ahead forecast is equal to the last estimated level plus h times the last estimated trend value. Hence the forecasts are a linear function of h.

### Damped trend methods

The forecasts generated by Holt’s linear method display a constant trend (increasing or decreasing) indefinitely into the future. Empirical evidence indicates that these methods tend to over-forecast, especially for longer forecast horizons. Motivated by this observation, Gardner & McKenzie ([1985](https://otexts.com/fpp2/holt.html#ref-GarMacK1985)) introduced a parameter that “dampens” the trend to a flat line some time in the future. Methods that include a damped trend have proven to be very successful, and are arguably the most popular individual methods when forecasts are required automatically for many series.

In conjunction with the smoothing parameters α

and β∗ (with values between 0 and 1 as in Holt’s method), this method also includes a damping parameter 0<ϕ<1: ^yt+h|t=ℓt+(ϕ+ϕ2+⋯+ϕh)btℓt=αyt+(1−α)(ℓt−1+ϕbt−1)bt=β∗(ℓt−ℓt−1)+(1−β∗)ϕbt−1. If ϕ=1, the method is identical to Holt’s linear method. For values between 0 and 1, ϕ dampens the trend so that it approaches a constant some time in the future. In fact, the forecasts converge to ℓT+ϕbT/(1−ϕ) as h→∞ for any value 0<ϕ<1

. This means that short-run forecasts are trended while long-run forecasts are constant.

In practice, ϕ

is rarely less than 0.8 as the damping has a very strong effect for smaller values. Values of ϕ close to 1 will mean that a damped model is not able to be distinguished from a non-damped model. For these reasons, we usually restrict ϕ to a minimum of 0.8 and a maximum of 0.98.

## 7.3 Holt-Winters’ seasonal method

Holt ([1957](https://otexts.com/fpp2/holt-winters.html#ref-Holt57)) and Winters ([1960](https://otexts.com/fpp2/holt-winters.html#ref-Winters60)) extended Holt’s method to capture seasonality. The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations — one for the level ℓt

, one for the trend bt, and one for the seasonal component st, with corresponding smoothing parameters α, β∗ and γ. We use m to denote the frequency of the seasonality, i.e., the number of seasons in a year. For example, for quarterly data m=4, and for monthly data m=12

.

There are two variations to this method that differ in the nature of the seasonal component. The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series. With the additive method, the seasonal component is expressed in absolute terms in the scale of the observed series, and in the level equation the series is seasonally adjusted by subtracting the seasonal component. Within each year, the seasonal component will add up to approximately zero. With the multiplicative method, the seasonal component is expressed in relative terms (percentages), and the series is seasonally adjusted by dividing through by the seasonal component. Within each year, the seasonal component will sum up to approximately m

.

### Holt-Winters’ additive method

The component form for the additive method is: ^yt+h|t=ℓt+hbt+st+h−m(k+1)ℓt=α(yt−st−m)+(1−α)(ℓt−1+bt−1)bt=β∗(ℓt−ℓt−1)+(1−β∗)bt−1st=γ(yt−ℓt−1−bt−1)+(1−γ)st−m,

where k is the integer part of (h−1)/m, which ensures that the estimates of the seasonal indices used for forecasting come from the final year of the sample. The level equation shows a weighted average between the seasonally adjusted observation (yt−st−m) and the non-seasonal forecast (ℓt−1+bt−1) for time t. The trend equation is identical to Holt’s linear method. The seasonal equation shows a weighted average between the current seasonal index, (yt−ℓt−1−bt−1), and the seasonal index of the same season last year (i.e., m

time periods ago).

The equation for the seasonal component is often expressed as st=γ∗(yt−ℓt)+(1−γ∗)st−m.

If we substitute ℓt from the smoothing equation for the level of the component form above, we get st=γ∗(1−α)(yt−ℓt−1−bt−1)+[1−γ∗(1−α)]st−m, which is identical to the smoothing equation for the seasonal component we specify here, with γ=γ∗(1−α). The usual parameter restriction is 0≤γ∗≤1, which translates to 0≤γ≤1−α

.

### Holt-Winters’ multiplicative method

The component form for the multiplicative method is: ^yt+h|t=(ℓt+hbt)st+h−m(k+1)ℓt=αytst−m+(1−α)(ℓt−1+bt−1)bt=β∗(ℓt−ℓt−1)+(1−β∗)bt−1st=γyt(ℓt−1+bt−1)+(1−γ)st−m

### Holt-Winters’ damped method

Damping is possible with both additive and multiplicative Holt-Winters’ methods. A method that often provides accurate and robust forecasts for seasonal data is the Holt-Winters method with a damped trend and multiplicative seasonality: ^yt+h|t=[ℓt+(ϕ+ϕ2+⋯+ϕh)bt]st+h−m(k+1).ℓt=α(yt/st−m)+(1−α)(ℓt−1+ϕbt−1)bt=β∗(ℓt−ℓt−1)+(1−β∗)ϕbt−1st=γyt(ℓt−1+ϕbt−1)+(1−γ)st−m.

# Chapter 8 ARIMA models

ARIMA models provide another approach to time series forecasting. Exponential smoothing and ARIMA models are the two most widely used approaches to time series forecasting, and provide complementary approaches to the problem. While exponential smoothing models are based on a description of the trend and seasonality in the data, ARIMA models aim to describe the autocorrelations in the data.

Before we introduce ARIMA models, we must first discuss the concept of stationarity and the technique of differencing time series.

## 8.1 Stationarity and differencing

A stationary time series is one whose properties do not depend on the time at which the series is observed.[14](https://otexts.com/fpp2/stationarity.html#fn14) Thus, time series with trends, or with seasonality, are not stationary — the trend and seasonality will affect the value of the time series at different times. On the other hand, a white noise series is stationary — it does not matter when you observe it, it should look much the same at any point in time.

Some cases can be confusing — a time series with cyclic behaviour (but with no trend or seasonality) is stationary. This is because the cycles are not of a fixed length, so before we observe the series we cannot be sure where the peaks and troughs of the cycles will be.

In general, a stationary time series will have no predictable patterns in the long-term. Time plots will show the series to be roughly horizontal (although some cyclic behaviour is possible), with constant variance.

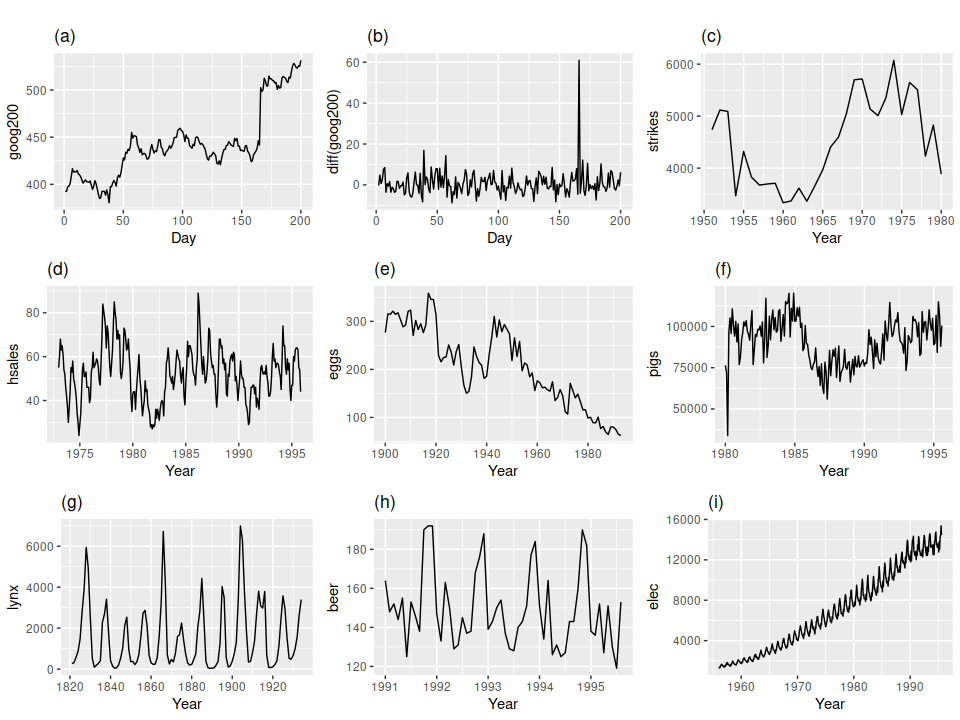


Figure 8.1: Which of these series are stationary? (a) Google stock price for 200 consecutive days; (b) Daily change in the Google stock price for 200 consecutive days; (c) Annual number of strikes in the US; (d) Monthly sales of new one-family houses sold in the US; (e) Annual price of a dozen eggs in the US (constant dollars); (f) Monthly total of pigs slaughtered in Victoria, Australia; (g) Annual total of lynx trapped in the McKenzie River district of north-west Canada; (h) Monthly Australian beer production; (i) Monthly Australian electricity production.

Consider the nine series plotted in Figure [8.1](https://otexts.com/fpp2/stationarity.html#fig:stationary). Which of these do you think are stationary?

Obvious seasonality rules out series (d), (h) and (i). Trends and changing levels rules out series (a), (c), (e), (f) and (i). Increasing variance also rules out (i). That leaves only (b) and (g) as stationary series.

At first glance, the strong cycles in series (g) might appear to make it non-stationary. But these cycles are aperiodic — they are caused when the lynx population becomes too large for the available feed, so that they stop breeding and the population falls to low numbers, then the regeneration of their food sources allows the population to grow again, and so on. In the long-term, the timing of these cycles is not predictable. Hence the series is stationary.

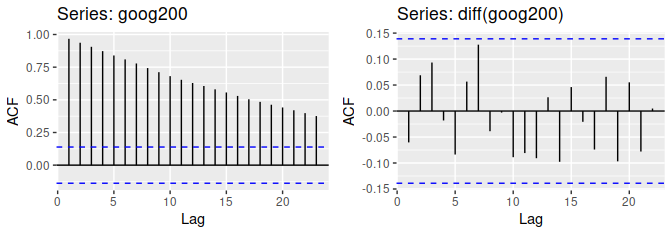
### Differencing

In Figure [8.1](https://otexts.com/fpp2/stationarity.html#fig:stationary), note that the Google stock price was non-stationary in panel (a), but the daily changes were stationary in panel (b). This shows one way to make a non-stationary time series stationary — compute the differences between consecutive observations. This is known as **differencing**.

Transformations such as logarithms can help to stabilise the variance of a time series. Differencing can help stabilise the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality.

As well as looking at the time plot of the data, the ACF plot is also useful for identifying non-stationary time series. For a stationary time series, the ACF will drop to zero relatively quickly, while the ACF of non-stationary data decreases slowly. Also, for non-stationary data, the value of r1

is often large and positive.



The ACF of the differenced Google stock price looks just like that of a white noise series. There are no autocorrelations lying outside the 95% limits, and the Ljung-Box Q∗ statistic has a p-value of 0.355 (for h=10

). This suggests that the daily change in the Google stock price is essentially a random amount which is uncorrelated with that of previous days.

### Random walk model

The differenced series is the change between consecutive observations in the original series, and can be written as y′t=yt−yt−1.

The differenced series will have only T−1 values, since it is not possible to calculate a difference y′1

for the first observation.

When the differenced series is white noise, the model for the original series can be written as yt−yt−1=εt,

where εt denotes white noise. Rearranging this leads to the “random walk” model yt=yt−1+εt.

Random walk models are widely used for non-stationary data, particularly financial and economic data. Random walks typically have:

* long periods of apparent trends up or down
* sudden and unpredictable changes in direction.

The forecasts from a random walk model are equal to the last observation, as future movements are unpredictable, and are equally likely to be up or down. Thus, the random walk model underpins naïve forecasts, first introduced in Section [3.1](https://otexts.com/fpp2/simple-methods.html#simple-methods).

A closely related model allows the differences to have a non-zero mean. Then yt−yt−1=c+εtoryt=c+yt−1+εt.

The value of c is the average of the changes between consecutive observations. If c is positive, then the average change is an increase in the value of yt. Thus, yt will tend to drift upwards. However, if c is negative, yt

will tend to drift downwards.

This is the model behind the drift method, also discussed in Section [3.1](https://otexts.com/fpp2/simple-methods.html#simple-methods).

### Second-order differencing

Occasionally the differenced data will not appear to be stationary and it may be necessary to difference the data a second time to obtain a stationary series: y′′t=y′t−y′t−1=(yt−yt−1)−(yt−1−yt−2)=yt−2yt−1+yt−2.

In this case, y′′t will have T−2

values. Then, we would model the “change in the changes” of the original data. In practice, it is almost never necessary to go beyond second-order differences.

### Seasonal differencing

A seasonal difference is the difference between an observation and the previous observation from the same season. So y′t=yt−yt−m,

where m= the number of seasons. These are also called “lag-m differences”, as we subtract the observation after a lag of m

periods.

If seasonally differenced data appear to be white noise, then an appropriate model for the original data is yt=yt−m+εt.

Forecasts from this model are equal to the last observation from the relevant season. That is, this model gives seasonal naïve forecasts, introduced in Section [3.1](https://otexts.com/fpp2/simple-methods.html#simple-methods).

The bottom panel in Figure [8.3](https://otexts.com/fpp2/stationarity.html#fig:a10diff) shows the seasonal differences of the logarithm of the monthly scripts for A10 (antidiabetic) drugs sold in Australia. The transformation and differencing have made the series look relatively stationary.

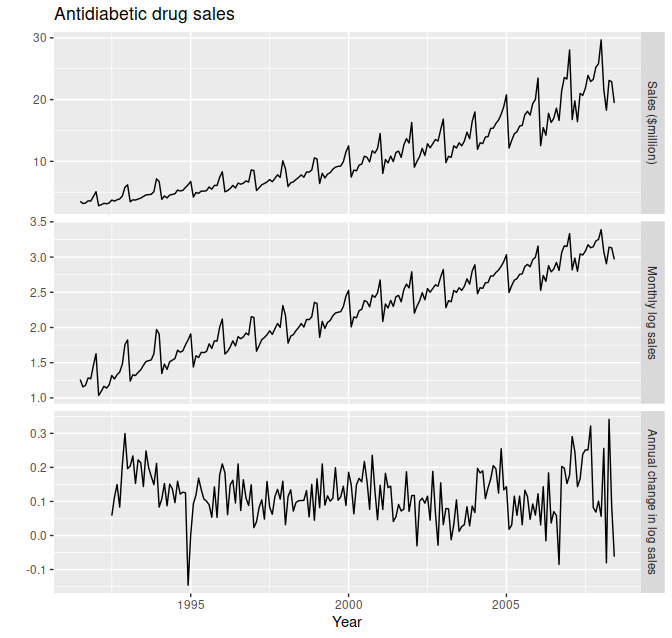


Figure 8.3: Logs and seasonal differences of the A10 (antidiabetic) sales data. The logarithms stabilise the variance, while the seasonal differences remove the seasonality and trend.

To distinguish seasonal differences from ordinary differences, we sometimes refer to ordinary differences as “first differences”, meaning differences at lag 1.

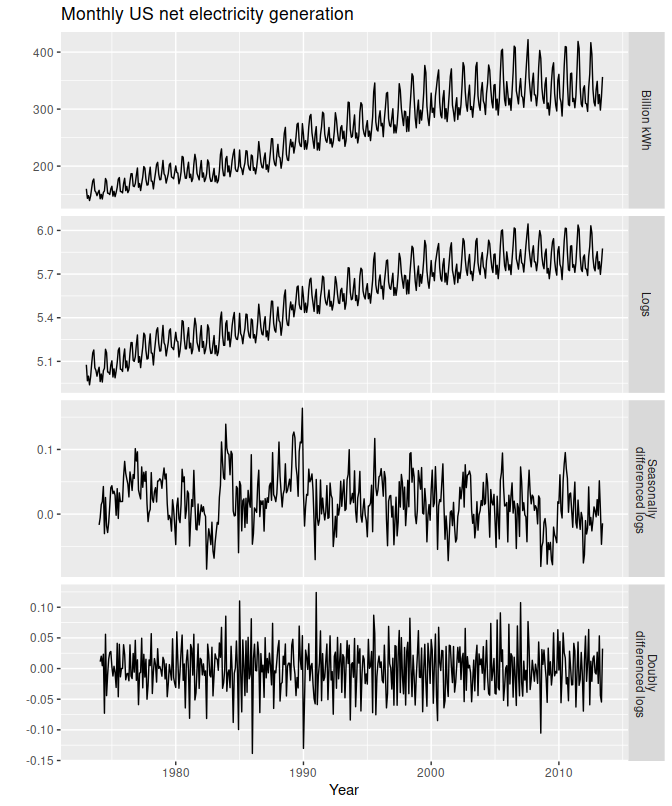


Figure 8.4: Top panel: US net electricity generation (billion kWh). Other panels show the same data after transforming and differencing.

There is a degree of subjectivity in selecting which differences to apply. The seasonally differenced data in Figure [8.3](https://otexts.com/fpp2/stationarity.html#fig:a10diff) do not show substantially different behaviour from the seasonally differenced data in Figure [8.4](https://otexts.com/fpp2/stationarity.html#fig:usmelec). In the latter case, we could have decided to stop with the seasonally differenced data, and not done an extra round of differencing. In the former case, we could have decided that the data were not sufficiently stationary and taken an extra round of differencing. Some formal tests for differencing are discussed below, but there are always some choices to be made in the modelling process, and different analysts may make different choices.

If y′t=yt−yt−m

denotes a seasonally differenced series, then the twice-differenced series is y′′t=y′t−y′t−1=(yt−yt−m)−(yt−1−yt−m−1)=yt−yt−1−yt−m+yt−m−1

When both seasonal and first differences are applied, it makes no difference which is done first—the result will be the same. However, if the data have a strong seasonal pattern, we recommend that seasonal differencing be done first, because the resulting series will sometimes be stationary and there will be no need for a further first difference. If first differencing is done first, there will still be seasonality present.

It is important that if differencing is used, the differences are interpretable. First differences are the change between one observation and the next. Seasonal differences are the change between one year to the next. Other lags are unlikely to make much interpretable sense and should be avoided.

## 8.2 Backshift notation

The backward shift operator B

is a useful notational device when working with time series lags: Byt=yt−1. (Some references use L for “lag” instead of B for “backshift”.) In other words, B, operating on yt, has the effect of shifting the data back one period. Two applications of B to yt shifts the data back two periods: B(Byt)=B2yt=yt−2. For monthly data, if we wish to consider “the same month last year,” the notation is B12yt = yt−12

.

The backward shift operator is convenient for describing the process of differencing. A first difference can be written as y′t=yt−yt−1=yt−Byt=(1−B)yt.

Note that a first difference is represented by (1−B). Similarly, if second-order differences have to be computed, then: y′′t=yt−2yt−1+yt−2=(1−2B+B2)yt=(1−B)2yt. In general, a dth-order difference can be written as (1−B)dyt.

Backshift notation is particularly useful when combining differences, as the operator can be treated using ordinary algebraic rules. In particular, terms involving B

can be multiplied together.

For example, a seasonal difference followed by a first difference can be written as (1−B)(1−Bm)yt=(1−B−Bm+Bm+1)yt=yt−yt−1−yt−m+yt−m−1,

the same result we obtained earlier.

**8.3 Autoregressive models**

In a multiple regression model, we forecast the variable of interest using a linear combination of predictors. In an autoregression model, we forecast the variable of interest using a linear combination of *past values of the variable*. The term *auto*regression indicates that it is a regression of the variable against itself.

Thus, an autoregressive model of order p

can be written as yt=c+ϕ1yt−1+ϕ2yt−2+⋯+ϕpyt−p+εt, where εt is white noise. This is like a multiple regression but with *lagged values* of yt as predictors. We refer to this as an **AR(p) model**, an autoregressive model of order p

.

Autoregressive models are remarkably flexible at handling a wide range of different time series patterns. The two series in Figure [8.5](https://otexts.com/fpp2/AR.html#fig:arp) show series from an AR(1) model and an AR(2) model. Changing the parameters ϕ1,…,ϕp

results in different time series patterns. The variance of the error term εt

will only change the scale of the series, not the patterns.

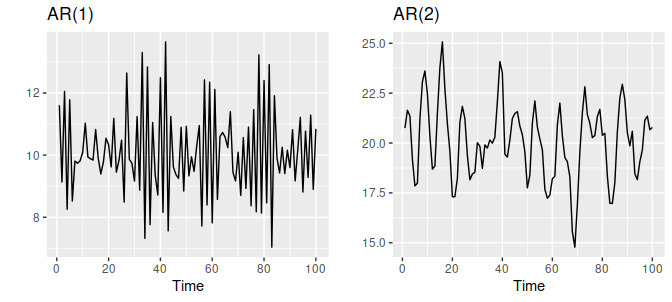


Figure 8.5: Two examples of data from autoregressive models with different parameters. Left: AR(1) with yt=18−0.8yt−1+εt

. Right: AR(2) with yt=8+1.3yt−1−0.7yt−2+εt. In both cases, εt

is normally distributed white noise with mean zero and variance one.

For an AR(1) model:

* when ϕ1=0

, yt

 is equivalent to white noise;

 when ϕ1=1 and c=0, yt

 is equivalent to a random walk;

 when ϕ1=1 and c≠0, yt

 is equivalent to a random walk with drift;

 when ϕ1<0, yt

* tends to oscillate around the mean.

We normally restrict autoregressive models to stationary data, in which case some constraints on the values of the parameters are required.

* For an AR(1) model: −1<ϕ1<1

 .

 For an AR(2) model: −1<ϕ2<1, ϕ1+ϕ2<1, ϕ2−ϕ1<1

* .

When p≥3

, the restrictions are much more complicated. R takes care of these restrictions when estimating a model.

**8.4 Moving average models**

Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model. yt=c+εt+θ1εt−1+θ2εt−2+⋯+θqεt−q,

where εt is white noise. We refer to this as an **MA(q) model**, a moving average model of order q. Of course, we do not *observe* the values of εt

, so it is not really a regression in the usual sense.

Notice that each value of yt

can be thought of as a weighted moving average of the past few forecast errors. However, moving average *models* should not be confused with the moving average *smoothing* we discussed in Chapter [6](https://otexts.com/fpp2/decomposition.html#decomposition). A moving average model is used for forecasting future values, while moving average smoothing is used for estimating the trend-cycle of past values.

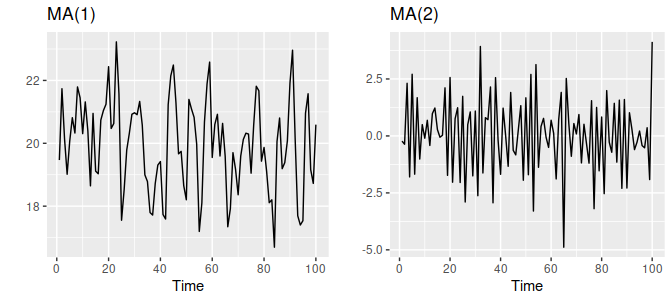


Figure 8.6: Two examples of data from moving average models with different parameters. Left: MA(1) with yt=20+εt+0.8εt−1

. Right: MA(2) with yt=εt−εt−1+0.8εt−2. In both cases, εt

is normally distributed white noise with mean zero and variance one.

Figure [8.6](https://otexts.com/fpp2/MA.html#fig:maq) shows some data from an MA(1) model and an MA(2) model. Changing the parameters θ1,…,θq

results in different time series patterns. As with autoregressive models, the variance of the error term εt

will only change the scale of the series, not the patterns.

It is possible to write any stationary AR(p

) model as an MA(∞) model. For example, using repeated substitution, we can demonstrate this for an AR(1) model: yt=ϕ1yt−1+εt=ϕ1(ϕ1yt−2+εt−1)+εt=ϕ21yt−2+ϕ1εt−1+εt=ϕ31yt−3+ϕ21εt−2+ϕ1εt−1+εtetc.

Provided −1<ϕ1<1

, the value of ϕk1 will get smaller as k gets larger. So eventually we obtain yt=εt+ϕ1εt−1+ϕ21εt−2+ϕ31εt−3+⋯, an MA(∞

) process.

The reverse result holds if we impose some constraints on the MA parameters. Then the MA model is called **invertible**. That is, we can write any invertible MA(q

) process as an AR(∞

) process. Invertible models are not simply introduced to enable us to convert from MA models to AR models. They also have some desirable mathematical properties.

For example, consider the MA(1) process, yt=εt+θ1εt−1

. In its AR(∞) representation, the most recent error can be written as a linear function of current and past observations: εt=∞∑j=0(−θ)jyt−j. When |θ|>1, the weights increase as lags increase, so the more distant the observations the greater their influence on the current error. When |θ|=1, the weights are constant in size, and the distant observations have the same influence as the recent observations. As neither of these situations make much sense, we require |θ|<1, so the most recent observations have higher weight than observations from the more distant past. Thus, the process is invertible when |θ|<1

.

The invertibility constraints for other models are similar to the stationarity constraints.

* For an MA(1) model: −1<θ1<1

 .

 For an MA(2) model: −1<θ2<1,  θ2+θ1>−1,  θ1−θ2<1

* .

More complicated conditions hold for q≥3

. Again, R will take care of these constraints when estimating the models.

## 8.5 Non-seasonal ARIMA models

If we combine differencing with autoregression and a moving average model, we obtain a non-seasonal ARIMA model. ARIMA is an acronym for AutoRegressive Integrated Moving Average (in this context, “integration” is the reverse of differencing). The full model can be written as y′t=c+ϕ1y′t−1+⋯+ϕpy′t−p+θ1εt−1+⋯+θqεt−q+εt,(8.1)

where y′t is the differenced series (it may have been differenced more than once). The “predictors” on the right hand side include both lagged values of yt and lagged errors. We call this an **ARIMA(p,d,q**

**) model**, where

|  |  |
| --- | --- |
| p= |  |

|  |  |
| --- | --- |
|  | order of the autoregressive part; |
| d= |  |

|  |  |
| --- | --- |
|  | degree of first differencing involved; |
| q= |  |

|  |  |
| --- | --- |
|  | order of the moving average part. |

The same stationarity and invertibility conditions that are used for autoregressive and moving average models also apply to an ARIMA model.

Many of the models we have already discussed are special cases of the ARIMA model, as shown in Table [8.1](https://otexts.com/fpp2/non-seasonal-arima.html#tab:arimaspecialcases).

|  |  |
| --- | --- |
| Table 8.1: Special cases of ARIMA models. | |
| White noise | ARIMA(0,0,0) |
| Random walk | ARIMA(0,1,0) with no constant |
| Random walk with drift | ARIMA(0,1,0) with a constant |
| Autoregression | ARIMA(p |

|  |
| --- |
| ,0,0) |
| Moving average | ARIMA(0,0,q |

|  |
| --- |
| ) |

Once we start combining components in this way to form more complicated models, it is much easier to work with the backshift notation. For example, Equation [(8.1)](https://otexts.com/fpp2/non-seasonal-arima.html#eq:8-arima) can be written in backshift notation as (1−ϕ1B−⋯−ϕpBp)(1−B)dyt=c+(1+θ1B+⋯+θqBq)εt↑↑↑AR(p)d differencesMA(q)(8.2)

R uses a slightly different parameterisation: (1−ϕ1B−⋯−ϕpBp)(y′t−μ)=(1+θ1B+⋯+θqBq)εt,(8.3)

where y′t=(1−B)dyt and μ is the mean of y′t. To convert to the form given by [(8.2)](https://otexts.com/fpp2/non-seasonal-arima.html#eq:arimaB), set c=μ(1−ϕ1−⋯−ϕp)

.

Selecting appropriate values for p

, d and q

can be difficult. However, the auto.arima() function in R will do it for you automatically. In Section [8.7](https://otexts.com/fpp2/arima-r.html#arima-r), we will learn how this function works, along with some methods for choosing these values yourself.

### Understanding ARIMA models

The auto.arima() function is useful, but anything automated can be a little dangerous, and it is worth understanding something of the behaviour of the models even when you rely on an automatic procedure to choose the model for you.

The constant c

has an important effect on the long-term forecasts obtained from these models.

* If c=0

and d=0

 , the long-term forecasts will go to zero.

 If c=0 and d=1

 , the long-term forecasts will go to a non-zero constant.

 If c=0 and d=2

 , the long-term forecasts will follow a straight line.

 If c≠0 and d=0

 , the long-term forecasts will go to the mean of the data.

 If c≠0 and d=1

 , the long-term forecasts will follow a straight line.

 If c≠0 and d=2

* , the long-term forecasts will follow a quadratic trend.

The value of d

also has an effect on the prediction intervals — the higher the value of d, the more rapidly the prediction intervals increase in size. For d=0

, the long-term forecast standard deviation will go to the standard deviation of the historical data, so the prediction intervals will all be essentially the same.

This behaviour is seen in Figure [8.8](https://otexts.com/fpp2/non-seasonal-arima.html#fig:usconsumptionf) where d=0

and c≠0

. In this figure, the prediction intervals are almost the same for the last few forecast horizons, and the point forecasts are equal to the mean of the data.

The value of p

is important if the data show cycles. To obtain cyclic forecasts, it is necessary to have p≥2, along with some additional conditions on the parameters. For an AR(2) model, cyclic behaviour occurs if ϕ21+4ϕ2<0. In that case, the average period of the cycles is[15](https://otexts.com/fpp2/non-seasonal-arima.html#fn15) 2πarc cos(−ϕ1(1−ϕ2)/(4ϕ2)).

### ACF and PACF plots

It is usually not possible to tell, simply from a time plot, what values of p

and q are appropriate for the data. However, it is sometimes possible to use the ACF plot, and the closely related PACF plot, to determine appropriate values for p and q

.

Recall that an ACF plot shows the autocorrelations which measure the relationship between yt

and yt−k for different values of k. Now if yt and yt−1 are correlated, then yt−1 and yt−2 must also be correlated. However, then yt and yt−2 might be correlated, simply because they are both connected to yt−1, rather than because of any new information contained in yt−2 that could be used in forecasting yt

.

To overcome this problem, we can use **partial autocorrelations**. These measure the relationship between yt

and yt−k after removing the effects of lags 1,2,3,…,k−1. So the first partial autocorrelation is identical to the first autocorrelation, because there is nothing between them to remove. Each partial autocorrelation can be estimated as the last coefficient in an autoregressive model. Specifically, αk, the kth partial autocorrelation coefficient, is equal to the estimate of ϕk in an AR(k) model. In practice, there are more efficient algorithms for computing αk

than fitting all of these autoregressions, but they give the same results.

Figures [8.9](https://otexts.com/fpp2/non-seasonal-arima.html#fig:usconsumptionacf) and [8.10](https://otexts.com/fpp2/non-seasonal-arima.html#fig:usconsumptionpacf) shows the ACF and PACF plots for the US consumption data shown in Figure [8.7](https://otexts.com/fpp2/non-seasonal-arima.html#fig:usconsumption). The partial autocorrelations have the same critical values of ±1.96/√T

as for ordinary autocorrelations, and these are typically shown on the plot as in Figure [8.9](https://otexts.com/fpp2/non-seasonal-arima.html#fig:usconsumptionacf).

ggAcf(uschange[,"Consumption"])

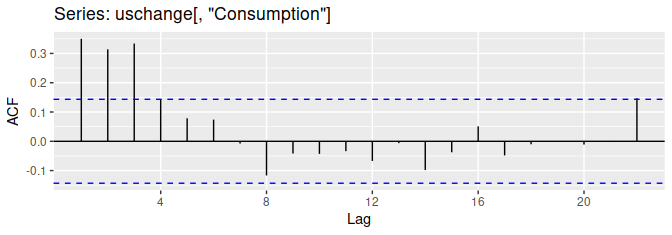


Figure 8.9: ACF of quarterly percentage change in US consumption.

ggPacf(uschange[,"Consumption"])

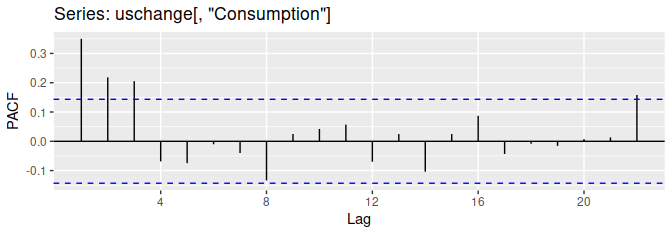


Figure 8.10: PACF of quarterly percentage change in US consumption.

If the data are from an ARIMA(p

,d,0) or ARIMA(0,d,q) model, then the ACF and PACF plots can be helpful in determining the value of p or q.[16](https://otexts.com/fpp2/non-seasonal-arima.html#fn16) If p and q are both positive, then the plots do not help in finding suitable values of p and q

.

The data may follow an ARIMA(p

,d

,0) model if the ACF and PACF plots of the differenced data show the following patterns:

* the ACF is exponentially decaying or sinusoidal;
* there is a significant spike at lag p

in the PACF, but none beyond lag p

* .

The data may follow an ARIMA(0,d

,q

) model if the ACF and PACF plots of the differenced data show the following patterns:

* the PACF is exponentially decaying or sinusoidal;
* there is a significant spike at lag q

in the ACF, but none beyond lag q

* .

In Figure [8.9](https://otexts.com/fpp2/non-seasonal-arima.html#fig:usconsumptionacf), we see that there are three spikes in the ACF, followed by an almost significant spike at lag 4. In the PACF, there are three significant spikes, and then no significant spikes thereafter (apart from one just outside the bounds at lag 22). We can ignore one significant spike in each plot if it is just outside the limits, and not in the first few lags. After all, the probability of a spike being significant by chance is about one in twenty, and we are plotting 22 spikes in each plot. The pattern in the first three spikes is what we would expect from an ARIMA(3,0,0), as the PACF tends to decrease. So in this case, the ACF and PACF lead us to think an ARIMA(3,0,0) model might be appropriate.

## 8.6 Estimation and order selection

### Maximum likelihood estimation

Once the model order has been identified (i.e., the values of p

, d and q), we need to estimate the parameters c, ϕ1,…,ϕp, θ1,…,θq. When R estimates the ARIMA model, it uses maximum likelihood estimation (MLE). This technique finds the values of the parameters which maximise the probability of obtaining the data that we have observed. For ARIMA models, MLE is similar to the least squares estimates that would be obtained by minimising T∑t=1ε2t.

(For the regression models considered in Chapter [5](https://otexts.com/fpp2/regression.html#regression), MLE gives exactly the same parameter estimates as least squares estimation.) Note that ARIMA models are much more complicated to estimate than regression models, and different software will give slightly different answers as they use different methods of estimation, and different optimisation algorithms.

In practice, R will report the value of the log likelihood of the data; that is, the logarithm of the probability of the observed data coming from the estimated model. For given values of p

, d and q

, R will try to maximise the log likelihood when finding parameter estimates.

### Information Criteria

Akaike’s Information Criterion (AIC), which was useful in selecting predictors for regression, is also useful for determining the order of an ARIMA model. It can be written as AIC=−2log(L)+2(p+q+k+1),

where L is the likelihood of the data, k=1 if c≠0 and k=0 if c=0. Note that the last term in parentheses is the number of parameters in the model (including σ2

, the variance of the residuals).

For ARIMA models, the corrected AIC can be written as AICc=AIC+2(p+q+k+1)(p+q+k+2)T−p−q−k−2,

and the Bayesian Information Criterion can be written as BIC=AIC+[log(T)−2](p+q+k+1).

Good models are obtained by minimising the AIC, AICc or BIC. Our preference is to use the AICc.

It is important to note that these information criteria tend not to be good guides to selecting the appropriate order of differencing (d

) of a model, but only for selecting the values of p and q. This is because the differencing changes the data on which the likelihood is computed, making the AIC values between models with different orders of differencing not comparable. So we need to use some other approach to choose d, and then we can use the AICc to select p and q.

## 8.9 Seasonal ARIMA models

So far, we have restricted our attention to non-seasonal data and non-seasonal ARIMA models. However, ARIMA models are also capable of modelling a wide range of seasonal data.

A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

|  |  |
| --- | --- |
| ARIMA | (p,d,q) |

|  |  |
| --- | --- |
|  | (P,D,Q)m |

|  |
| --- |
|  |
|  | ↑ |

|  |  |
| --- | --- |
|  | ↑ |

|  |
| --- |
|  |
|  | Non-seasonal part | Seasonal part of |
|  | of the model | of the model |

where m=

number of observations per year. We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period. For example, an ARIMA(1,1,1)(1,1,1)4

model (without a constant) is for quarterly data (m=4), and can be written as (1−ϕ1B) (1−Φ1B4)(1−B)(1−B4)yt=(1+θ1B) (1+Θ1B4)εt.

The additional seasonal terms are simply multiplied by the non-seasonal terms.

### ACF/PACF

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF. For example, an ARIMA(0,0,0)(0,0,1)12

model will show:

* a spike at lag 12 in the ACF but no other significant spikes;
* exponential decay in the seasonal lags of the PACF (i.e., at lags 12, 24, 36, …).

Similarly, an ARIMA(0,0,0)(1,0,0)12

model will show:

* exponential decay in the seasonal lags of the ACF;
* a single significant spike at lag 12 in the PACF.

In considering the appropriate seasonal orders for a seasonal ARIMA model, restrict attention to the seasonal lags.

The modelling procedure is almost the same as for non-seasonal data, except that we need to select seasonal AR and MA terms as well as the non-seasonal components of the model. The process is best illustrated via examples.