Forecasting Princples and Practice

Koji Mizumura

August 28, 2018(Beg) -

Table of Contents

# Resources

[Forecasting: princples and practice](https://otexts.org/fpp2/)  
[Slides](https://robjhyndman.com/seminars/uwa2017/)  
[Github](https://github.com/nealxun/ForecastingPrinciplePractices)

From the Github, we can install the package fpp2.

install.packages("fpp2",dependencies = TRUE)

# Chapter 1: Introduction

<https://github.com/nealxun/ForecastingPrinciplePractices/blob/master/01-intro.Rmd>

## 1.1 What can be forecast?

Forecasting is required in many situations: deciding whether to build another power generation plant in the next five years requires forecasts of future demand; scheduling staff in a call centre next week requires forecasts of call volumes; stocking an inventory requires forecasts of stock requirements. Forecasts can be required several years in advance (for the case of capital investments), or only a few minutes beforehand (for telecommunication routing). Whatever the circumstances or time horizons involved, forecasting is an important aid to effective and efficient planning.

Some things are easier to forecast than others. The time of the sunrise tomorrow morning can be forecast precisely. On the other hand, tomorrow’s lotto numbers cannot be forecast with any accuracy. The predictability of an event or a quantity depends on several factors including:

how well we understand the factors that contribute to it; how much data are available; whether the forecasts can affect the thing we are trying to forecast. For example, forecasts of electricity demand can be highly accurate because all three conditions are usually satisfied. We have a good idea of the contributing factors: electricity demand is driven largely by temperatures, with smaller effects for calendar variation such as holidays, and economic conditions. Provided there is a sufficient history of data on electricity demand and weather conditions, and we have the skills to develop a good model linking electricity demand and the key driver variables, the forecasts can be remarkably accurate.

On the other hand, when forecasting currency exchange rates, only one of the conditions is satisfied: there is plenty of available data. However, we have a limited understanding of the factors that affect exchange rates, and forecasts of the exchange rate have a direct effect on the rates themselves. If there are well-publicised forecasts that the exchange rate will increase, then people will immediately adjust the price they are willing to pay and so the forecasts are self-fulfilling. In a sense, the exchange rates become their own forecasts. This is an example of the “efficient market hypothesis”. Consequently, forecasting whether the exchange rate will rise or fall tomorrow is about as predictable as forecasting whether a tossed coin will come down as a head or a tail. In both situations, you will be correct about 50% of the time, whatever you forecast. In situations like this, forecasters need to be aware of their own limitations, and not claim more than is possible.

Often in forecasting, a key step is knowing when something can be forecast accurately, and when forecasts will be no better than tossing a coin. Good forecasts capture the genuine patterns and relationships which exist in the historical data, but do not replicate past events that will not occur again. In this book, we will learn how to tell the difference between a random fluctuation in the past data that should be ignored, and a genuine pattern that should be modelled and extrapolated.

Many people wrongly assume that forecasts are not possible in a changing environment. Every environment is changing, and a good forecasting model captures the way in which things are changing. Forecasts rarely assume that the environment is unchanging. What is normally assumed is that the way in which the environment is changing will continue into the future. That is, a highly volatile environment will continue to be highly volatile; a business with fluctuating sales will continue to have fluctuating sales; and an economy that has gone through booms and busts will continue to go through booms and busts. A forecasting model is intended to capture the way things move, not just where things are. As Abraham Lincoln said, “If we could first know where we are and whither we are tending, we could better judge what to do and how to do it”.

Forecasting situations vary widely in their time horizons, factors determining actual outcomes, types of data patterns, and many other aspects. Forecasting methods can be simple, such as using the most recent observation as a forecast (which is called the naïve method), or highly complex, such as neural nets and econometric systems of simultaneous equations. Sometimes, there will be no data available at all. For example, we may wish to forecast the sales of a new product in its first year, but there are obviously no data to work with. In situations like this, we use judgmental forecasting, discussed in Chapter 4. The choice of method depends on what data are available and the predictability of the quantity to be forecast.

## 1.2 Forecasting, planning and goals

Forecasting is a common statistical task in business, where it helps to inform decisions about the scheduling of production, transportation and personnel, and provides a guide to long-term strategic planning. However, business forecasting is often done poorly, and is frequently confused with planning and goals. They are three different things.

### *Forecasting*

is about predicting the future as accurately as possible, given all of the information available, including historical data and knowledge of any future events that might impact the forecasts.

### *Goals*

are what you would like to have happen. Goals should be linked to forecasts and plans, but this does not always occur. Too often, goals are set without any plan for how to achieve them, and no forecasts for whether they are realistic.

### *Planning*

is a response to forecasts and goals. Planning involves determining the appropriate actions that are required to make your forecasts match your goals.

Forecasting should be an integral part of the decision-making activities of management, as it can play an important role in many areas of a company. Modern organisations require short-term, medium-term and long-term forecasts, depending on the specific applicatio

### *Short-term forecasts*

are needed for the scheduling of personnel, production and transportation. As part of the scheduling process, forecasts of demand are often also required.

### *Medium-term forecasts*

are needed to determine future resource requirements, in order to purchase raw materials, hire personnel, or buy machinery and equipment.

### *Long-term forecasts*

are used in strategic planning. Such decisions must take account of market opportunities, environmental factors and internal resources.

An organisation needs to develop a forecasting system that involves several approaches to predicting uncertain events. Such forecasting systems require the development of expertise in identifying forecasting problems, applying a range of forecasting methods, selecting appropriate methods for each problem, and evaluating and refining forecasting methods over time. It is also important to have strong organisational support for the use of formal forecasting methods if they are to be used successfully.

## 1.3 Determining what to forecast

In the early stages of a forecasting project, decisions need to be made about what should be forecast. For example, if forecasts are required for items in a manufacturing environment, it is necessary to ask whether forecasts are needed for:

1. every product line, or for groups of products?
2. every sales outlet, or for outlets grouped by region, or only for total sales?
3. weekly data, monthly data or annual data?

It is also necessary to consider the forecasting horizon. Will forecasts be required for one month in advance, for 6 months, or for ten years? Different types of models will be necessary, depending on what forecast horizon is most important.

How frequently are forecasts required? Forecasts that need to be produced frequently are better done using an automated system than with methods that require careful manual work.

It is worth spending time talking to the people who will use the forecasts to ensure that you understand their needs, and how the forecasts are to be used, before embarking on extensive work in producing the forecasts.

Once it has been determined what forecasts are required, it is then necessary to find or collect the data on which the forecasts will be based. The data required for forecasting may already exist. These days, a lot of data are recorded, and the forecaster’s task is often to identify where and how the required data are stored. The data may include sales records of a company, the historical demand for a product, or the unemployment rate for a geographic region. A large part of a forecaster’s time can be spent in locating and collating the available data prior to developing suitable forecasting methods.

## 1.4 Forecasting data and methods

The appropriate forecasting methods depend largely on what data are available.

If there are no data available, or if the data available are not relevant to the forecasts, then **qualitative forecasting** methods must be used. These methods are not purely guesswork—there are well-developed structured approaches to obtaining good forecasts without using historical data. These methods are discussed in Chapter 4.

**Quantitative forecasting** can be applied when two conditions are satisfied:

1. numerical information about the past is available;
2. it is reasonable to assume that some aspects of the past patterns will continue into the future. There is a wide range of quantitative forecasting methods, often developed within specific disciplines for specific purposes. Each method has its own properties, accuracies, and costs that must be considered when choosing a specific method.

Most quantitative prediction problems use either time series data (collected at regular intervals over time) or cross-sectional data (collected at a single point in time). In this book we are concerned with forecasting future data, and we concentrate on the time series domain.

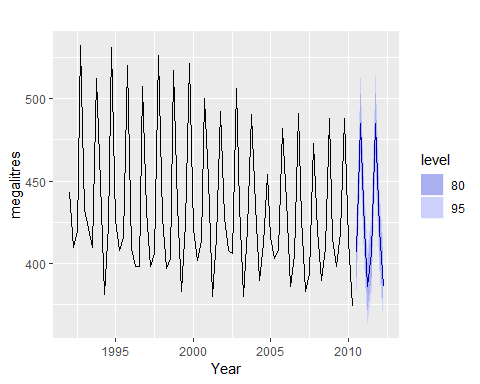
### 1) Time series forecasting

Examples of time series data include:

* Daily IBM stock prices
* Monthly rainfall
* Quarterly sales results for Amazon
* Annual Google profits

Anything that is observed sequentially over time is a time series. In this book, we will only consider time series that are observed at regular intervals of time (e.g., hourly, daily, weekly, monthly, quarterly, annually). Irregularly spaced time series can also occur, but are beyond the scope of this book.

When forecasting time series data, the aim is to estimate how the sequence of observations will continue into the future. Figure 1.1 shows the quarterly Australian beer production from 1992 to the second quarter of 2010.



Australian quarterly beer production:1992Q1-2010Q2,with two years of forecasts.

The blue lines show forecasts for the next two years. Notice how the forecasts have captured the seasonal pattern seen in the historical data and replicated it for the next two years. The dark shaded region shows 80% prediction intervals. That is, each future value is expected to lie in the dark shaded region with a probability of 80%. The light shaded region shows 95% prediction intervals. These prediction intervals are a useful way of displaying the uncertainty in forecasts. In this case the forecasts are expected to be accurate, and hence the prediction intervals are quite narrow.

The simplest time series forecasting methods use only information on the variable to be forecast, and make no attempt to discover the factors that affect its behaviour. Therefore they will extrapolate trend and seasonal patterns, but they ignore all other information such as marketing initiatives, competitor activity, changes in economic conditions, and so on.

Time series models used for forecasting include decomposition models, exponential smoothing models and ARIMA models. These models are discussed in Chapters 6, 7 and 8, respectively.

### 2) Predictor variables and time series forecastin

Predictor variables are often useful in time series forecasting. For example, suppose we wish to forecast the hourly electricity demand (ED) of a hot region during the summer period. A model with predictor variables might be of the form

The relationship is not exact — there will always be changes in electricity demand that cannot be accounted for by the predictor variables. The “error” term on the right allows for random variation and the effects of relevant variables that are not included in the model. We call this an **explanatory model** because it helps explain what causes the variation in electricity demand.

Because the electricity demand data form a time series, we could also use a time series model for forecasting. In this case, a suitable time series forecasting equation is of the form.

where, *t* is the present hour, *t-1* is the previous hour, *t-2* is two hours ago, and so on. Here, prediction of the future is based on past values of a variable, but not on external variables which may affect the system. Again, the “error” term on the right allows for random variation and the effects of relevant variables that are not included in the model.

There is also a third type of model which combines the features of the above two models. For example, it might be given by

These types of “mixed models” have been given various names in different disciplines. They are known as dynamic regression models, panel data models, longitudinal models, transfer function models, and linear system models (assuming that  
*f* is linear). These models are discussed in Chapter 9.

An explanatory model is useful because it incorporates information about other variables, rather than only historical values of the variable to be forecast. However, there are several reasons a forecaster might select a time series model rather than an explanatory or mixed model. First, the system may not be understood, and even if it was understood it may be extremely difficult to measure the relationships that are assumed to govern its behaviour. Second, it is necessary to know or forecast the future values of the various predictors in order to be able to forecast the variable of interest, and this may be too difficult. Third, the main concern may be only to predict what will happen, not to know why it happens. Finally, the time series model may give more accurate forecasts than an explanatory or mixed model.

The model to be used in forecasting depends on the resources and data available, the accuracy of the competing models, and the way in which the forecasting model is to be used.

## 1.6 The basic steps in a forecasting task

<https://otexts.org/fpp2/basic-steps.html>

A forecasting task usually involves five basic steps.

### Step1: Problem definition

Often this is the most difficult part of forecasting. Defining the problem carefully requires an understanding of the way the forecasts will be used, who requires the forecasts, and how the forecasting function fits within the organisation requiring the forecasts. A forecaster needs to spend time talking to everyone who will be involved in collecting data, maintaining databases, and using the forecasts for future planning.

### Step2: Gather information

There are always at least two kinds of information required: (a) statistical data, and (b) the accumulated expertise of the people who collect the data and use the forecasts. Often, it will be difficult to obtain enough historical data to be able to fit a good statistical model. In that case, the judgmental forecasting methods of Chapter 4 can be used. Occasionally, old data will be less useful due to structural changes in the system being forecast; then we may choose to use only the most recent data. However, remember that good statistical models will handle evolutionary changes in the system; don’t throw away good data unnecessarily.

### Step3: Preliminary (exploratory) analysis

Always start by graphing the data. Are there consistent patterns? Is there a significant trend? Is seasonality important? Is there evidence of the presence of business cycles? Are there any outliers in the data that need to be explained by those with expert knowledge? How strong are the relationships among the variables available for analysis? Various tools have been developed to help with this analysis. These are discussed in Chapters 2 and 6.

### Step4: Choosing and fitting models

The best model to use depends on the availability of historical data, the strength of relationships between the forecast variable and any explanatory variables, and the way in which the forecasts are to be used. It is common to compare two or three potential models. Each model is itself an artificial construct that is based on a set of assumptions (explicit and implicit) and usually involves one or more parameters which must be estimated using the known historical data. We will discuss regression models [Chapter 5](https://otexts.org/fpp2/regression.html#regression), exponential smoothing methods [Chapter 7](https://otexts.org/fpp2/expsmooth.html#expsmooth), Box-Jenkins ARIMA models [Chapter 8](https://otexts.org/fpp2/arima.html#arima),Dynamic regression models[Chapter 9](https://otexts.org/fpp2/dynamic.html#dynamic), Hierarchical forecasting [Chapter 10](https://otexts.org/fpp2/hierarchical.html#hierarchical), and several advanced methods including neural networks and vector autoregression in [Chapter 11](https://otexts.org/fpp2/advanced.html#advanced).

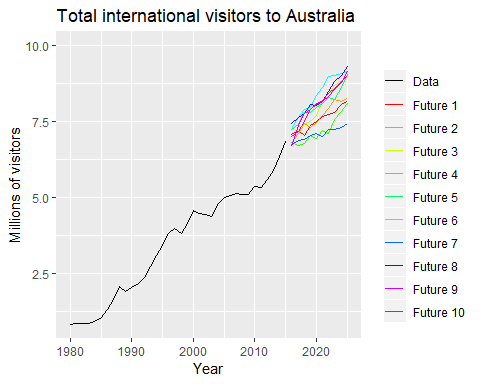
### Step5: Using an evaluating a forecasting model

Once a model has been selected and its parameters estimated, the model is used to make forecasts. The performance of the model can only be properly evaluated after the data for the forecast period have become available. A number of methods have been developed to help in assessing the accuracy of forecasts. There are also organisational issues in using and acting on the forecasts. A brief discussion of some of these issues is given in Chapter 3. When using a forecasting model in practice, numerous practical issues arise such as how to handle missing values and outliers, or how to deal with short time series. These are discussed in Chapter 12.

## 1.7 The statistical forecasting perspective

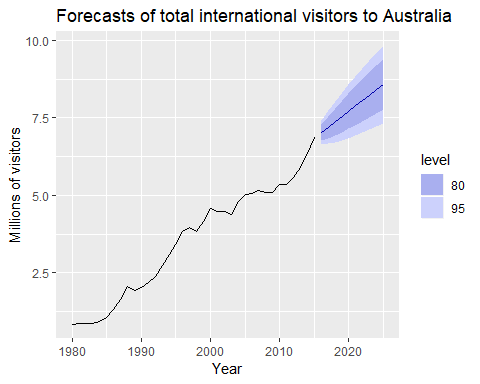
The thing we are trying to forecast is unknown (or we wouldn’t be forecasting it), and so we can think of it as a *random variable*. For example, the total sales for next month could take a range of possible values, and until we add up the actual sales at the end of the month, we don’t know what the value will be. So until we know the sales for next month, it is a random quantity.

Because next month is relatively close, we usually have a good idea what the likely sales values could be. On the other hand, if we are forecasting the sales for the same month next year, the possible values it could take are much more variable. In most forecasting situations, the variation associated with the thing we are forecasting will shrink as the event approaches. In other words, the further ahead we forecast, the more uncertain we are.

We can imagine many possible futures, each yielding a different value for the thing we wish to forecast. Plotted in black in Figure 1.2 are the total international visitors to Australia from 1980 to 2015. Also shown are ten possible futures from 2016–2025. 

When we obtain a forecast, we are estimating the *middle* of the range of possible values the random variable could take. Often, a forecast is accompanied by a **prediction interval** giving a range of values the random variable could take with relatively high probability. For example, a 95% prediction interval contains a range of values which should include the actual future value with probability 95%.

Instead of plotting individual possible futures as shown in Figure 1.2, we usually show these prediction intervals instead. The plot below shows 80% and 95% intervals for the future Australian international visitors. The blue line is the average of the possible future values, which we call the point forecasts.



Total international visitors to Australia (1980-2015) along with 10-year forecasts and 80% and 95% prediction intervanls.

We will use the subscript for time. For example, will denote the observation at time . Suppose we denote all the information we have observed as ${\cal I}$ and we want to forecast . We then write $y\_{t} | {\cal I}$ meaning “the random variable given what we know in ${\cal I}$”. The set of values that this random variable could take, along with their relative probabilities, is known as the “probability distribution” of $y\_{t} |{\cal I}$. In forecasting, we call this the “forecast distribution”.

When we talk about the “forecast”, we usually mean the average value of the forecast distribution, and we put a “hat” over to show this. Thus, we write the forecast of as , meaning the average of the possible values that could take given everything we know. Occasionally, we will use to refer to the *median* (or middle value) of the forecast distribution instead.

It is often useful to specify exactly what information we have used in calculating the forecast. Then we will write, for example, to mean the forecast of taking account of all previous observations . Similarly, means the forecast of taking account of (i.e., an -step forecast taking account of all observations up to time ).

# Chapter 2: Time series graphics

The first thing to do in any data analysis task is to plot the data. Graphs enable many features of the data to be visualised, including patterns, unusual observations, changes over time, and relationships between variables. The features that are seen in plots of the data must then be incorporated, as much as possible, into the forecasting methods to be used. Just as the type of data determines what forecasting method to use, it also determines what graphs are appropriate. But before we produce graphs, we need to set up our time series in R.

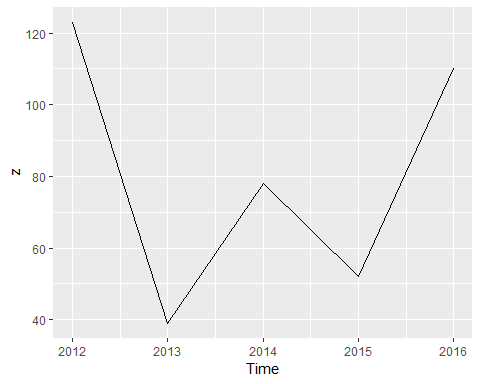
## 2.1 ts objects

A time series can be thought of as a list of numbers, along with some information about what times those numbers were recorded. This information can be stored as ts object in R.

Suppose you have annual observations for the last few years:

We turn this into a ts object using the ts() function:

## Time Series:  
## Start = 2012   
## End = 2016   
## Frequency = 1   
## [1] 123 39 78 52 110



If you have annual data, with one observation per year, you only need to provide the starting year (or the ending year).

For observations that are more frequent than once per year, you simply add a frequency argument. For example, if your monthly data is already stored as a numerical vector z, then it can be converted to a ts object like this:

y <- ts(z,start=2003,frequency=12)  
y

## Jan Feb Mar Apr May  
## 2003 123 39 78 52 110

Almost all of the data used in this book is already stored as ts objects. But if you want to work with your own data, you will need to use the ts() function before proceeding with the analysis.

### Frequency of a time series

The “frequency” is the number of observations before the seasonal pattern repeats.

This is the opposite of the definition of frequency in physics, or in Fourier analysis, where this would be called the “period”.

When using the ts() function in R, the following choices should be used.

|  |  |
| --- | --- |
| Data | frequency |
| Annual | 1 |
| Quartely | 4 |
| Monthly | 12 |
| Weekly | 52 |

Actually, there are not 52 weeks in a year, but on average, allowing for a leap year every fourth year. But most functions which use ts objects require integer frequency.

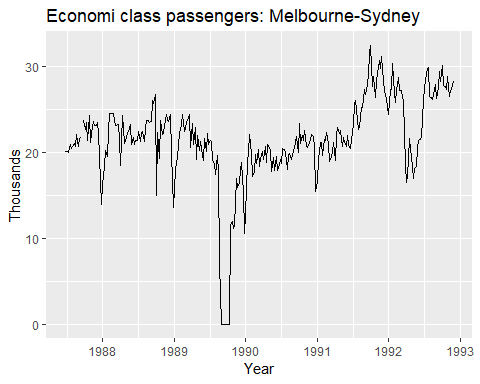
If the frequency of observations is greater than once per week, then there is usually more than one way of handling the frequency. For example, data with daily observations might have a weekly seasonality (frequency) or an annual seasonality (frequency). Similarly, data that are observed every minute might have an hourly seasonality (frequency ), a daily seasonality (frequency), a weekly seasonality (frequency) and an annual seasonality (frequency). If you want to use a ts object, then you need to decide which of these is the most important.

In [chapter 11](https://otexts.org/fpp2/advanced.html#advanced) we will look at handling these types of multiple seasonality, without having to choose just one of the frequencies.

## 2.2 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 observations, with consecutive observations joined by straight lines. Figure 2.1 below shows the weekly economiy passenger load on Ansett Airlines between Australia’s two largest cities.

autoplot(melsyd[,"Economy.Class"])+  
 ggtitle("Economi class passengers: Melbourne-Sydney")+  
 xlab("Year")+  
 ylab("Thousands")



Weekly economy passenger load on Ansett Airlines.

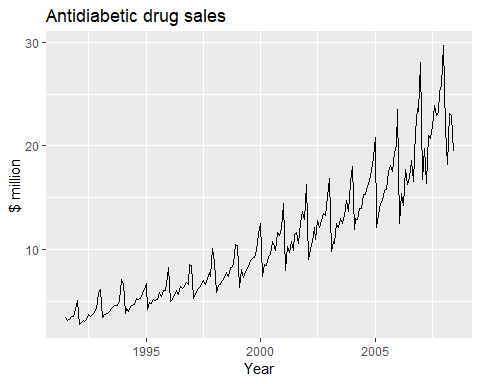
We will use the autopot() command frequently. It automatically produces an appropriate plot of whatever you pass to it in the first argument. In this case, it recognises melsyd[,"Economy.Class"] as a time series and produces a time plot

The time plot immediately reveals some intersting 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 @ref(fig:a10).

autoplot(a10)+  
 ggtitle("Antidiabetic drug sales")+  
 ylab("$ million")+  
 xlab("Year")



Monthly sales of antidiabetic drugs in Australia.

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.

### Seasonal

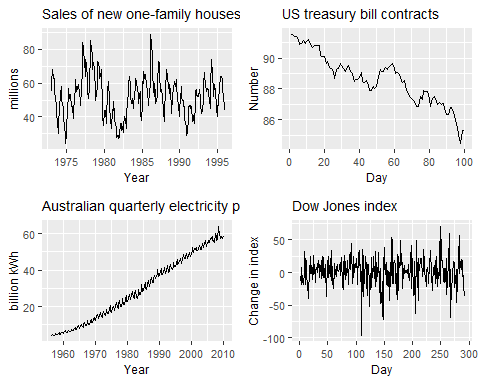
A seasonal pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week. Seasonality is always of a fixed and known frequency. The monthly sales of antidiabetic drugs above shows seasonality which is induced partly by the change in the cost of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

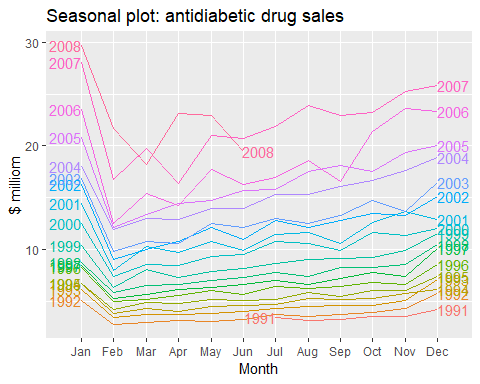
The examples in Figure 2.3 show different combinations of the above components. 

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 monthly 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 Dow Jones index (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.

ggseasonplot(a10,year.labels = T,year.labels.left = T)+  
 ylab("$ milliom")+  
 ggtitle("Seasonal plot: antidiabetic drug sales")



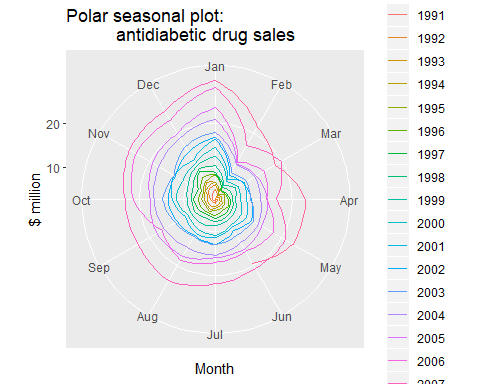
Seasonal plot of monthly antidiabetic drug sales in Australia.

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.

A useful variation on the seasonal plot uses polar coordinates. Setting polar=TRUE makes the time series axis circular rather than horizontal, as shown below.

ggseasonplot(a10,polar=T)+  
 ylab("$ million")+  
 ggtitle("Polar seasonal plot:  
 antidiabetic drug sales")

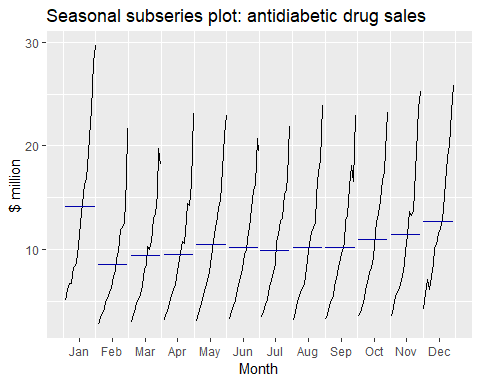


Polar seasonal plot of monthly antidiabetic drug sales in Australia.

## 2.5 Seasonal subseries plots

An alternative plot that emphasizes the seasonal patterns is where the data for each season are collected together in separate mini time plots.

ggsubseriesplot(a10)+  
 ylab("$ million")+  
 ggtitle("Seasonal subseries plot: antidiabetic drug sales")



Seasonal subseries plot of monthly antidiabetic drug sales in Australia.

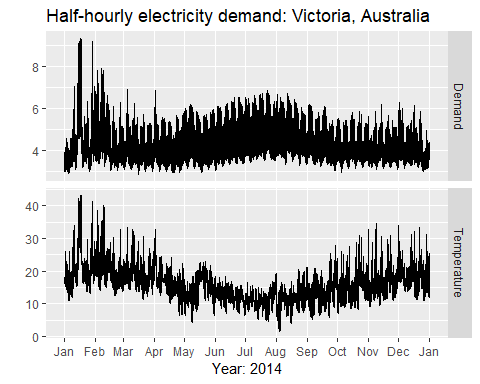
The horizontal lines indicate the means for each month. This form of plot enables the underlying seasonal pattern to be seen clearly, and also shows the changes in seasonality over time. It is especially useful in identifying changes within particular seasons. In this example, the plot is not particularly revealing; but in some cases, this is the most useful way of viewing seasonal changes over time.

## 2.6 Scatterplots

The graphs discussed so far are useful for visualising individual time series. It is also useful to explore relationships *between* time series.

Figure 2.7 shows two time series: half-hourly electricity demand (in Gigawatts) and temperature (in degrees Celsius), for 2014 in Victoria, Australia. The temperatures are for Melbourne, the largest city in Victoria, while the demand values are for the entire state.

month.breaks <- cumsum(c(0,31,28,31,30,31,30,31,31,30,31,30,31)\*48)  
autoplot(elecdemand[,c(1,3)], facet=TRUE) +  
 xlab("Year: 2014") + ylab("") +  
 ggtitle("Half-hourly electricity demand: Victoria, Australia")+  
 scale\_x\_continuous(breaks=2014+month.breaks/max(month.breaks),  
 minor\_breaks=NULL, labels=c(month.abb,month.abb[1]))

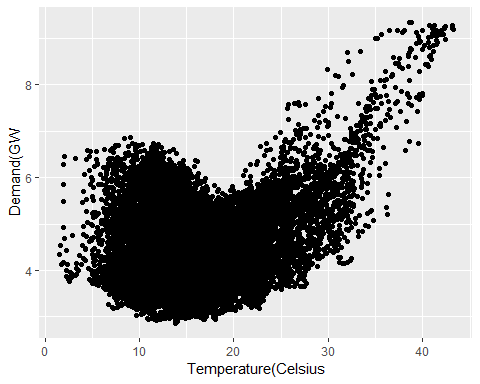


Half hourly electricity demand and temperatures in Victoria, Australia, for 2014.

(The actual code for this plot is little more complidated than what is shown in order to include the months on the x-axis.)

We can study the relationship between demand and temparature by plotting one series against the other.

qplot(Temperature,Demand,data=as.data.frame(elecdemand))+  
 ylab("Demand(GW")+xlab("Temperature(Celsius")

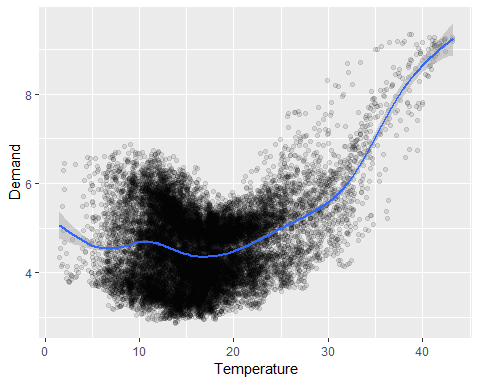


head(data.frame(elecdemand))

## Demand WorkDay Temperature  
## 1 3.914647 0 18.2  
## 2 3.672550 0 17.9  
## 3 3.497539 0 17.6  
## 4 3.339145 0 16.8  
## 5 3.204313 0 16.3  
## 6 3.100043 0 16.6

## ggplot()  
data.frame(elecdemand) %>%   
 ggplot(aes(x=Temperature,y=Demand))+  
 geom\_jitter(alpha=0.1)+  
 geom\_smooth()

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



This scatterplot helps us to visualise the relationship between the variables. It is clear that high demand occurs when temperatures are high due to the effect of air-conditioning. But there is also a heating effect, where demand increases for very low temperatures.

### Correlation

It is common to compute *correlation coefficient* to measure the strength of the relationship between two variables. The correlation between variable and is given by:

The value of always lies between and with negative values indicating a negative relationship and positive values indicating a positive relationship. The graphs in Figure 2.9 show examples of data sets with varying levels of correlation.

The correlation coefficient only measures the strength of the linear relationship, and can sometimes be misleading. For example, the correlation for the electricity demand and temperature data shown in Figure 2.8 is , but the non-linear relationship is stronger than that.

The plots in Figure 2.10 all have correlation coefficients of 0.82, but they have very different relationships. This shows how important it is look at the plots of the data and not simply rely on correlation values.

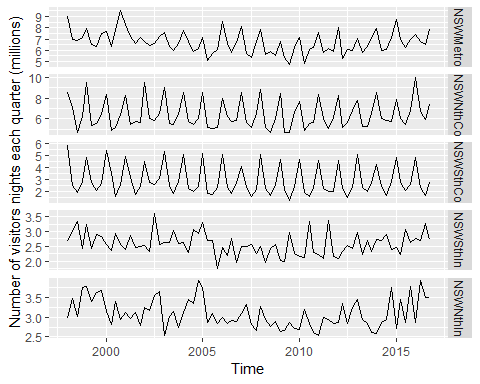
### Scatterplot matrices

When there are several potential predictor variables, it is useful to plot each variable against each other variable. Consider the five time series shown in Figure 2.11, showing quarterly visitor numbers for five regions of New South Wales, Australia.

# data overview  
data.frame(fpp2::visnights) %>%   
 head()

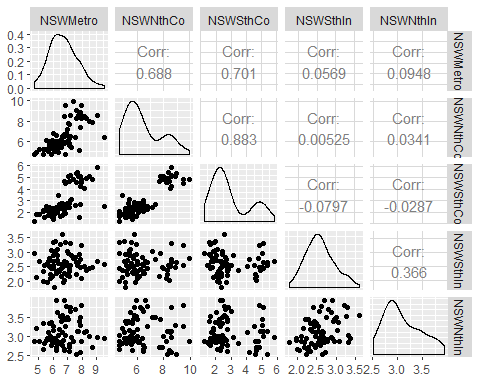
## NSWMetro NSWNthCo NSWSthCo NSWSthIn NSWNthIn QLDMetro QLDCntrl QLDNthCo  
## 1 9.047095 8.565678 5.818029 2.679538 2.977507 12.106052 2.748374 2.137234  
## 2 6.962126 7.124468 2.466437 3.010732 3.477703 7.786687 4.040915 2.269596  
## 3 6.871963 4.716893 1.928053 3.328869 3.014770 11.380024 5.343964 4.890227  
## 4 7.147293 6.269299 2.797556 2.417772 3.757972 9.311460 4.260419 2.621548  
## 5 7.956923 9.493901 4.853681 3.224285 3.790760 12.671942 4.186113 2.483203  
## 6 6.542243 5.401201 2.759843 2.428489 3.395284 9.582965 4.237806 3.377830  
## SAUMetro SAUCoast SAUInner VICMetro VICWstCo VICEstCo VICInner  
## 1 2.881372 2.591997 0.8948773 7.490382 2.4420048 3.381972 5.326655  
## 2 2.124736 1.375780 0.9792509 5.198178 0.9605047 1.827940 4.441119  
## 3 2.284870 1.079542 0.9803289 5.244217 0.7559744 1.351952 3.815645  
## 4 1.785889 1.497664 1.5094343 6.274246 1.2716040 1.493415 3.859567  
## 5 2.293873 2.247684 0.9635227 9.187422 2.3850583 2.896929 4.588755  
## 6 2.197418 1.672802 0.9968803 4.992303 1.3288638 1.547901 4.070401  
## WAUMetro WAUCoast WAUInner OTHMetro OTHNoMet  
## 1 3.075779 3.066555 0.6949954 3.437924 2.073469  
## 2 2.154929 3.334405 0.5576796 2.677081 1.787939  
## 3 2.787286 4.365844 1.0061844 3.793743 2.345021  
## 4 2.752910 4.521996 1.1725514 3.304231 1.943689  
## 5 3.519564 3.579347 0.3981829 3.510819 2.165838  
## 6 3.160430 3.408533 0.5960182 2.871867 1.803940

autoplot(visnights[,1:5],facets=T)+  
 ylab("Number of visitors nights each quarter (millions)")



To see the relationships between these five time series, we can plot each time series against the others. These plots can be arranged in a scatterplot matrix, as shown in Figure 2.12. (This plot requires the GGally package to be installed.)

GGally::ggpairs(as.data.frame(visnights[,1:5]))



For each panel, the variable on the vertical axis is given by the variable name in that row, and the variable on the horizontal axis is given by the variable name in that column. There are many options available to produce different plots within each panel. In the default version, the correlations are shown in the upper right half of the plot, while the scatterplots are shown in the lower half. On the diagonal are shown density plots.

The value of the scatterplot matrix is that it enables a quick view of the relationships between all pairs of variables. In this example, the second column of plots shows there is a strong positive relationship between visitors to the NSW north coast and visitors to the NSW south coast, but no detectable relationship between visitors to the NSW north coast and visitors to the NSW south inland. Outliers can also be seen. There is one unusually high quarter for the NSW Metropolitan region, corresponding to the 2000 Sydney Olympics. This is most easily seen in the first two plots in the left column of Figure 2.12, where the largest value for NSW Metro is separate from the main cloud of observations.

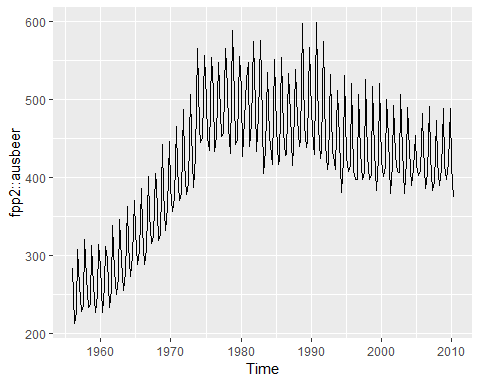
## 2.7 Lag plots

Figure 2.13 displays scatterplots of quarterly Australian beer production, where the horizontal axis shows lagged values of the time series. Each graph shows plotted against for different values of .

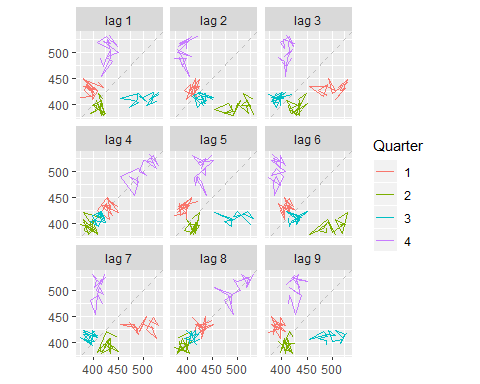
# aus beer overview  
fpp2::ausbeer %>%   
 as.data.frame() %>%   
 head()

## x  
## 1 284  
## 2 213  
## 3 227  
## 4 308  
## 5 262  
## 6 228

autoplot(fpp2::ausbeer)



beer2 <- window(ausbeer,start=1992)  
gglagplot(beer2)



Here the colours indicate the quarter of the variable on the vertical axis. The lines connect points in chronological order. The relationship is strongly positive at lags 4 and 8, reflecting the strong quarterly seasonality in the data. The negative relationship seen for lags 2 and 6 occurs because peaks (in Q4) are plotted against troughs (in Q2)

The window() function used here is very useful when extracting a portion of a time series. In this case, we have extracted the data from ausbeer, beginning in 1992.

## 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 sefveral autocorrelation coefficients, corresponding to each panel in the lag plot. For example, measures the relationship between and , measures the relationship between and and so on.

The value of can be written as

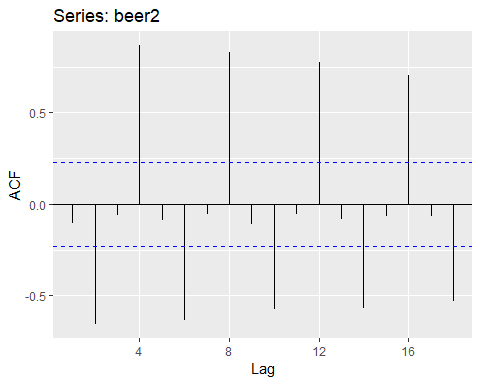
, where is the length of the time series.

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| -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. The autocorrelation coefficients are plotted to show the *autocorrelation function* of **ACF**. The plot is also known as a *correlogram*.

ggAcf(beer2)



In this graph: - 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 two quarters apart. - 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.

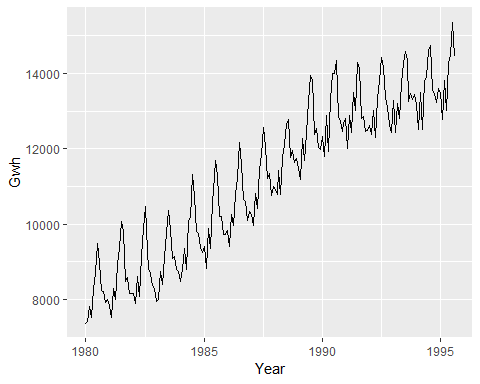
### Trend and seasonality in ACF plots

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

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

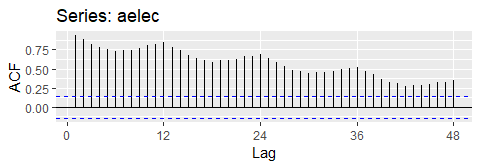
When data are both trended and seasonal, you see a combination of these effects. The monthly, Australian electricity demand series plotted in Figure 2.15 shows both trend and seasonality. Its ACF is shown in Figure 2.16.

aelec <- window(elec,start=1980)  
autoplot(aelec)+xlab("Year")+ylab("Gwh")



Monthly Australian electricity demand from 1980–1995.

ggAcf(aelec,lag=48)



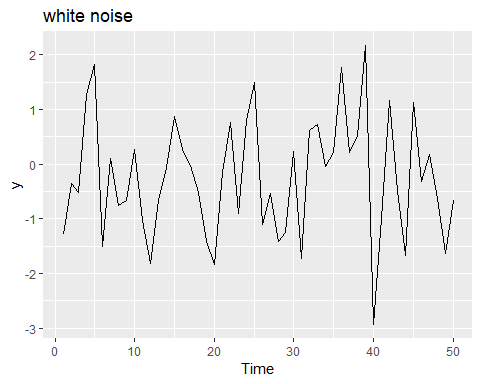
ACF of monthly Australian electricity demand.

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

## 2.9 White noise

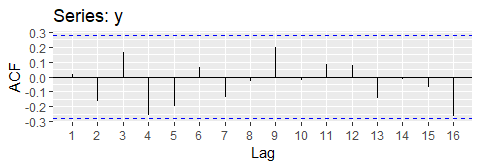
Time series that show no autocorrelation are called **white nose**. Figure 2.17 gives an example of a white noise series.

set.seed(30)  
y <- ts(rnorm(50))  
autoplot(y)+ggtitle("white noise")



A white noise time series.

ggAcf(y)



Autocorrelation function for the white noise series.

For white noise series, we expect each autocorrelation to be close to zero. Of course, they will not be exactly equal to zero as there is some random variation. For a white noise series, we expect 95% of the spikes in the ACF to lie within where is the length of the time series. It is common to plot these bounds on a graph of the ACF (the blue dashed lines above). IF one or more large spikes are outside these bounds, or if substantially more than 5% or spikes are outside these bounds, then the series is probably not white noise.

In this example, T=50 and the bounds are . All of the autocorrelation coefficients lie within these limits, confirming that the data are white noise.

## 2.10 Exercises

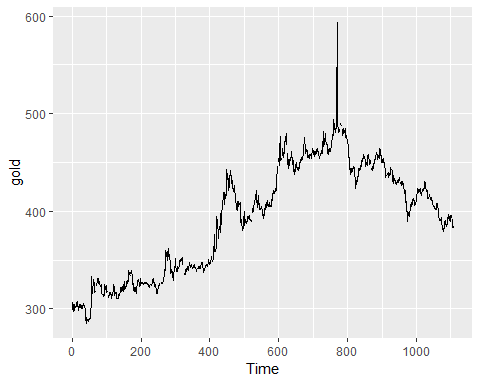
<https://github.com/mldataanalysis/Time-Series-Solutions>

1. Usethe help function to explore what the series gold, woolyrinq and gas represent.

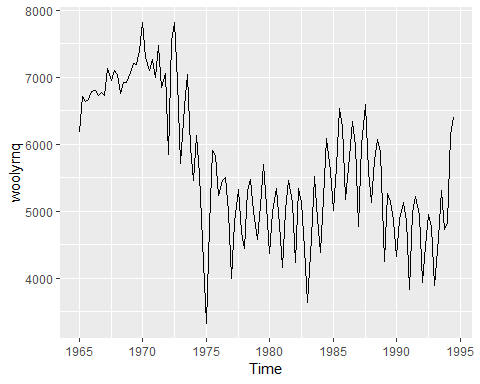
* *gold*: Daily morning gold prices in US dollars. 1 January 1985 – 31 March 1989.
* *woolynrnq*: Quarterly production of woollen yarn in Australia: tonnes. Mar 1965 – Sep 1994
* *gas*:Australian monthly gas production 1956-1995

1. Use autoplot() to plot each of these in separate plots.

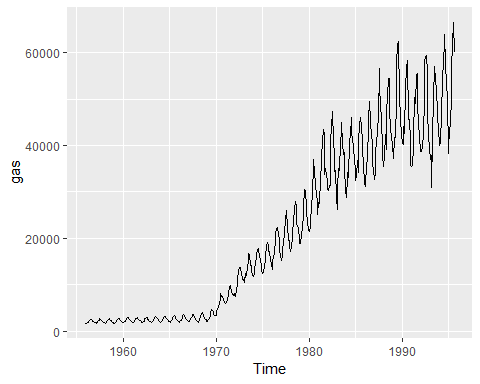
library(fpp2)  
autoplot(gold)



autoplot(woolyrnq)



autoplot(gas)



1. What is the frequency of each series. Hint: apply the frequency function.

frequency(gold)

## [1] 1

frequency(woolyrnq)

## [1] 4

frequency(gas)

## [1] 12

1. Download the file tute1.csv from the [the book website](http://otexts.org/fpp2/extrafiles/tute1.csv), open it in excel and review its contents. You should find four columns of information. Columns B through D each contains a quarterly series,labelled Sales, AdBudget and GDP. Sales contains the quarterly sales for a small company over the period 1981-2005. AdBudget is the advertising budget and GDP is the gross domestic product. All series have been adjusted for inflation.
2. Read the data into R with the following script:

tute1 <- read\_csv("C:/Users/kojikm.mizumura/Desktop/Data Science/Forecasting Princples and Practice (FPP)/tute1.csv")

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:  
## cols(  
## X1 = col\_character(),  
## Sales = col\_double(),  
## AdBudget = col\_double(),  
## GDP = col\_double()  
## )

head(tute1)

## # A tibble: 6 x 4  
## X1 Sales AdBudget GDP  
## <chr> <dbl> <dbl> <dbl>  
## 1 Mar-81 1020. 659. 252.  
## 2 Jun-81 889. 589 291.  
## 3 Sep-81 795 512. 291.  
## 4 Dec-81 1004. 614. 292.  
## 5 Mar-82 1058. 647. 279.  
## 6 Jun-82 944. 602 254

1. Convert the data to time series

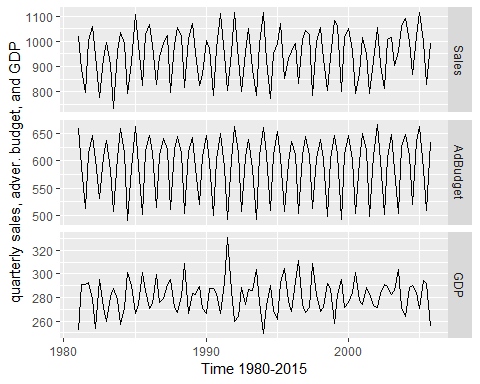
mytimeseries <- ts(tute1[,-1],start=1981,frequency = 4)  
mytimeseries %>% head()

## Sales AdBudget GDP  
## 1981 Q1 1020.2 659.2 251.8  
## 1981 Q2 889.2 589.0 290.9  
## 1981 Q3 795.0 512.5 290.8  
## 1981 Q4 1003.9 614.1 292.4  
## 1982 Q1 1057.7 647.2 279.1  
## 1982 Q2 944.4 602.0 254.0

Note that the [,-1] removes the first column which contains the quarters as we don’t need them now.

1. Construct time series plots of each of the three series

autoplot(mytimeseries,facet=TRUE)+  
 xlab("Time 1980-2015")+  
 ylab("quarterly sales, adver. budget, and GDP")



1. Download some monthly Australian retail data from the book website. These represent retail sales in various categories for different Australian states, and are stored in a MS-Excel file.
2. You can read the data into R with the following script

retaildata <- readxl::read\_excel("C:/Users/kojikm.mizumura/Desktop/Data Science/Forecasting Princples and Practice (FPP)/retail.xlsx",skip=1)  
head(retaildata)

## # A tibble: 6 x 190  
## `Series ID` A3349335T A3349627V A3349338X A3349398A A3349468W  
## <dttm> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1982-04-01 00:00:00 303. 41.7 63.9 409. 65.8  
## 2 1982-05-01 00:00:00 298. 43.1 64 405. 65.8  
## 3 1982-06-01 00:00:00 298 40.3 62.7 401 62.3  
## 4 1982-07-01 00:00:00 308. 40.9 65.6 414. 68.2  
## 5 1982-08-01 00:00:00 299. 42.1 62.6 404. 66   
## 6 1982-09-01 00:00:00 305. 42 64.4 412. 62.3  
## # ... with 184 more variables: A3349336V <dbl>, A3349337W <dbl>,  
## # A3349397X <dbl>, A3349399C <dbl>, A3349874C <dbl>, A3349871W <dbl>,  
## # A3349790V <dbl>, A3349556W <dbl>, A3349791W <dbl>, A3349401C <dbl>,  
## # A3349873A <dbl>, A3349872X <dbl>, A3349709X <dbl>, A3349792X <dbl>,  
## # A3349789K <dbl>, A3349555V <dbl>, A3349565X <dbl>, A3349414R <dbl>,  
## # A3349799R <dbl>, A3349642T <dbl>, A3349413L <dbl>, A3349564W <dbl>,  
## # A3349416V <dbl>, A3349643V <dbl>, A3349483V <dbl>, A3349722T <dbl>,  
## # A3349727C <dbl>, A3349641R <dbl>, A3349639C <dbl>, A3349415T <dbl>,  
## # A3349349F <dbl>, A3349563V <dbl>, A3349350R <dbl>, A3349640L <dbl>,  
## # A3349566A <dbl>, A3349417W <dbl>, A3349352V <dbl>, A3349882C <dbl>,  
## # A3349561R <dbl>, A3349883F <dbl>, A3349721R <dbl>, A3349478A <dbl>,  
## # A3349637X <dbl>, A3349479C <dbl>, A3349797K <dbl>, A3349477X <dbl>,  
## # A3349719C <dbl>, A3349884J <dbl>, A3349562T <dbl>, A3349348C <dbl>,  
## # A3349480L <dbl>, A3349476W <dbl>, A3349881A <dbl>, A3349410F <dbl>,  
## # A3349481R <dbl>, A3349718A <dbl>, A3349411J <dbl>, A3349638A <dbl>,  
## # A3349654A <dbl>, A3349499L <dbl>, A3349902A <dbl>, A3349432V <dbl>,  
## # A3349656F <dbl>, A3349361W <dbl>, A3349501L <dbl>, A3349503T <dbl>,  
## # A3349360V <dbl>, A3349903C <dbl>, A3349905J <dbl>, A3349658K <dbl>,  
## # A3349575C <dbl>, A3349428C <dbl>, A3349500K <dbl>, A3349577J <dbl>,  
## # A3349433W <dbl>, A3349576F <dbl>, A3349574A <dbl>, A3349816F <dbl>,  
## # A3349815C <dbl>, A3349744F <dbl>, A3349823C <dbl>, A3349508C <dbl>,  
## # A3349742A <dbl>, A3349661X <dbl>, A3349660W <dbl>, A3349909T <dbl>,  
## # A3349824F <dbl>, A3349507A <dbl>, A3349580W <dbl>, A3349825J <dbl>,  
## # A3349434X <dbl>, A3349822A <dbl>, A3349821X <dbl>, A3349581X <dbl>,  
## # A3349908R <dbl>, A3349743C <dbl>, A3349910A <dbl>, A3349435A <dbl>,  
## # A3349365F <dbl>, A3349746K <dbl>, ...

The second argument skip=1 is required because the Excel sheet has two header rows.

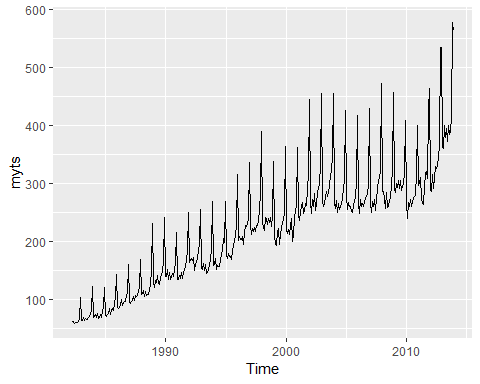
1. Select one of the time series as follows (but replace the column name with your own chosen column):

myts <- ts(retaildata[,"A3349873A"],  
 frequency=12,start=c(1982,4))  
head(myts)

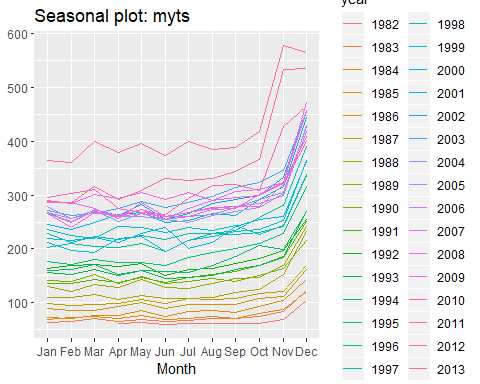
## Apr May Jun Jul Aug Sep  
## 1982 62.4 63.1 59.6 61.9 60.7 61.2

1. Explore your chosen retail time series using the following functions:

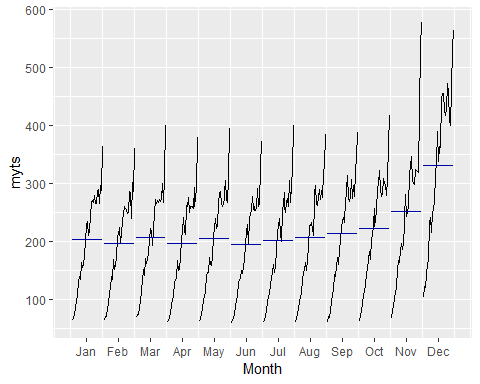
autoplot(myts)



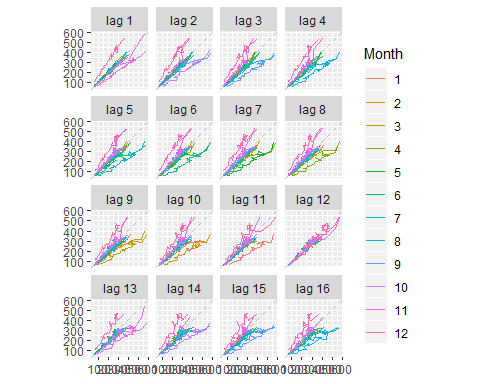
# year-based split  
ggseasonplot(myts)



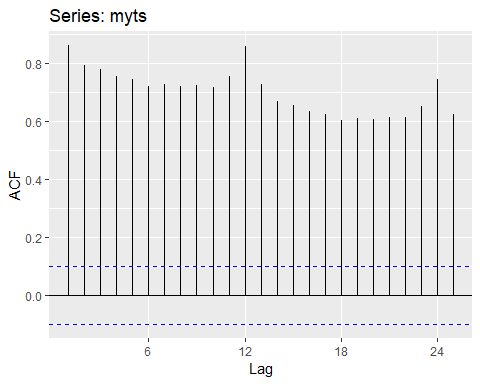
# month-based split  
ggsubseriesplot(myts)



# lag-plot  
# Each graph shows $y\_t$ plotted against $y\_{t−k}$ for different values of $k$.  
gglagplot(myts)



ggAcf(myts)



1. Create time plots of the following time series: bicoal, chicken, dole,usdeaths,lynx,goog,writing,fancy,a10,h02

* usehelp() to fundout about the data in each series.
* For the goog plot, modify the axis label and title.

goog: Closing stock prices of GOOG from the NASDAQ exchange, for 1000 consecutive trading days between 25 February 2013 and 13 February 2017. Adjusted for splits. goog200 contains the first 200 observations from goog.

# help(goog)  
# str(dole)  
glimpse(goog)

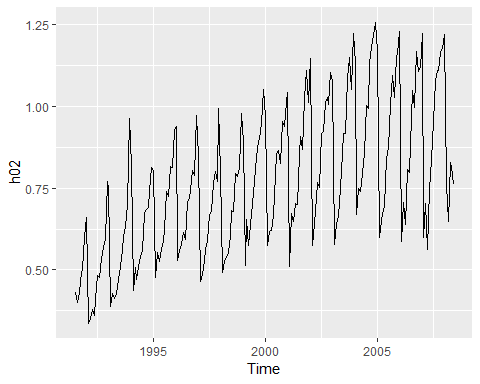
## Time-Series [1:1000] from 1 to 1000: 393 393 397 398 400 ...

head(goog)

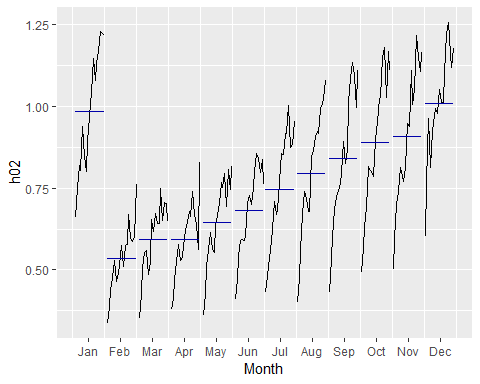
## Time Series:  
## Start = 1   
## End = 6   
## Frequency = 1   
## [1] 392.8300 392.5121 397.3059 398.0113 400.4902 408.0957

**5**. Use the ggseasonplot() and ggsubseriesplot() functions to explore the seasonal patterns in the following time series: writing, fancy, a10, h02. - What can you say about the seasonal patterns? - Can you identify any unusual years?

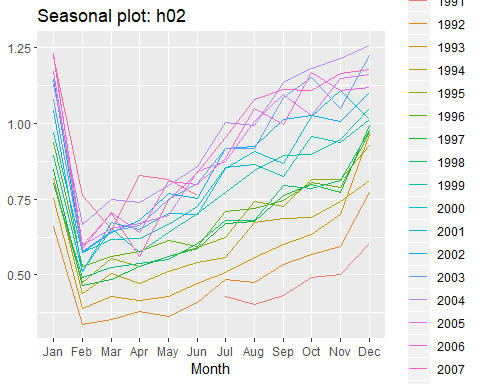
# h02 dataset  
# Monthly corticosteroid drug sales in Australia from July 1991 to June 2008.  
  
autoplot(h02)



ggsubseriesplot(h02)



ggseasonplot(h02)

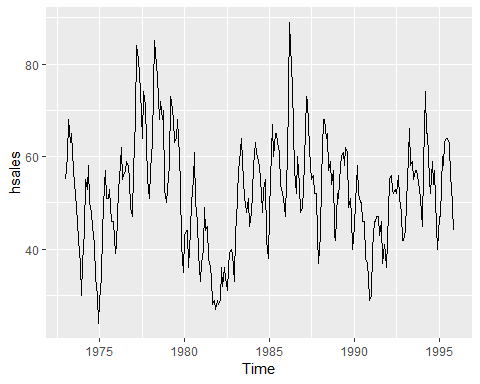


**6.** Use the following graphics functions: autoplot(), ggseasonplot(), ggsubseriesplot(), gglagplot(), ggAcf() and explore features from the following time series: hsales, usdeaths, bricksq, sunspotarea, gasoline. - Can you spot any seasonality, cyclicity and trend? - What do you learn about the series?

# Monthly sales of new one-family houses sold in the USA since 1973.  
head(hsales)

## Jan Feb Mar Apr May Jun  
## 1973 55 60 68 63 65 61

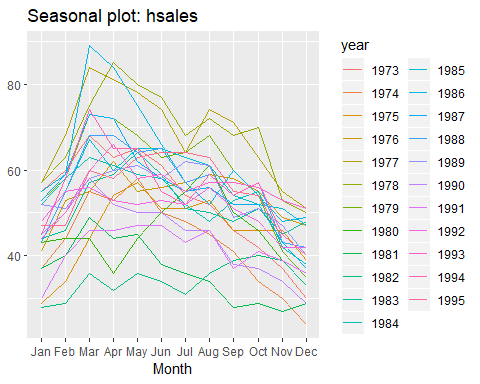
# dataset overview  
autoplot(hsales)



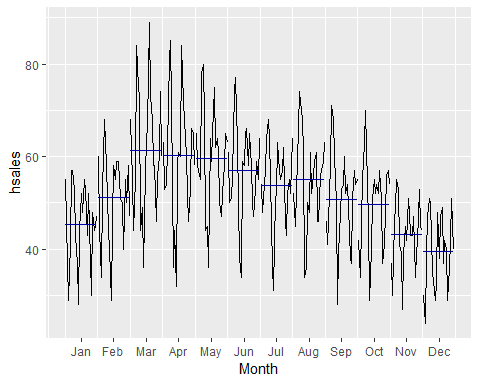
frequency(hsales) # monthly ts data

## [1] 12

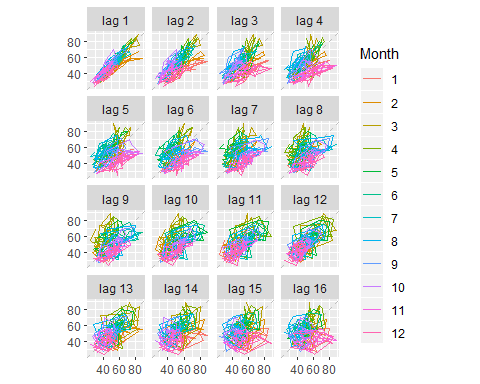
ggseasonplot(hsales)



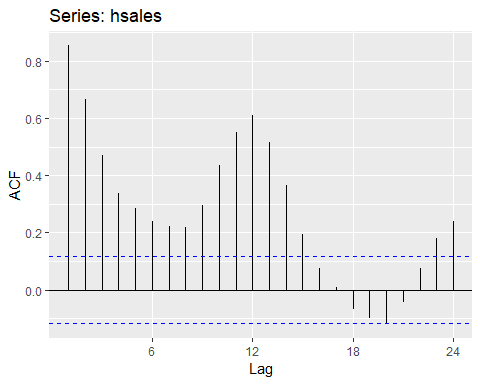
ggsubseriesplot(hsales)



gglagplot(hsales)

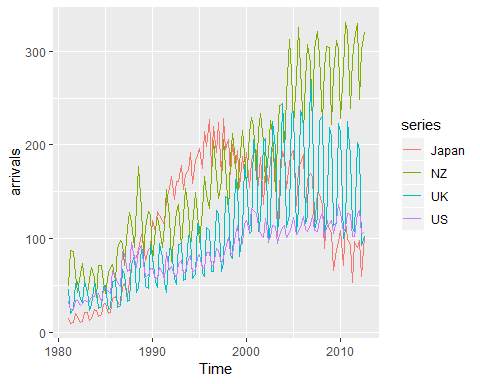


ggAcf(hsales)



1. The arrivals data set comprises quarterly international arrivals (in thousands) to Australia from Japan, New Zealand, UK and the US. Use autoplot(), ggseasonplot() and ggsubseriesplot() to compare the differences between the arrivals from these four countries. Can you identify any unusual observations?

# Quarterly international arrivals (in thousands) to Australia from Japan, New Zealand, UK and the US. 1981Q1 - 2012Q3.  
autoplot(arrivals)



# ggseasonplot(arrivals)  
# ggsubseriesplot(arrivals)

1. Skipped

**9**.The pigs data shows the monthly total number of pigs slaughtered in Victoria, Australia, from Jan 1980 to Aug 1995. Use mypigs <- window(pigs, start=1990) to select the data starting from 1990. Use autoplot and ggAcf for mypigs series and compare these to white noise plots from Figures 2.17 and 2.18.

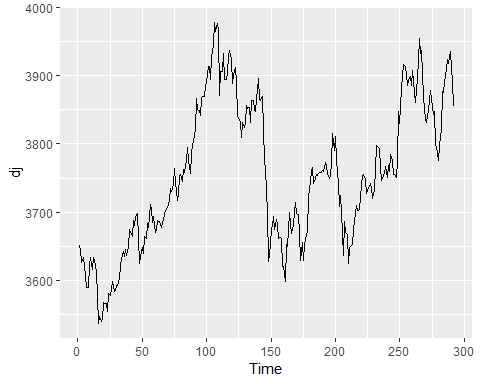
head(pigs)

## Jan Feb Mar Apr May Jun  
## 1980 76378 71947 33873 96428 105084 95741

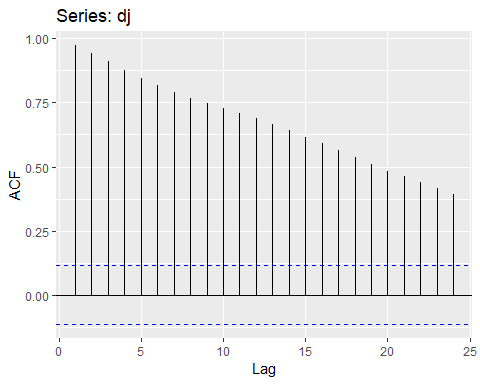
**10**. dj contains 292 consecutive traiding days of the Dow Jones Index. Use ddj<-diff(dj) to compute the daily changes in the index. Plot ddj and its ACF. Do the changes in the Dow Jones Index look like white noise?

dj: Dow-Jones index Dow-Jones index on 251 trading days ending 26 Aug 1994.

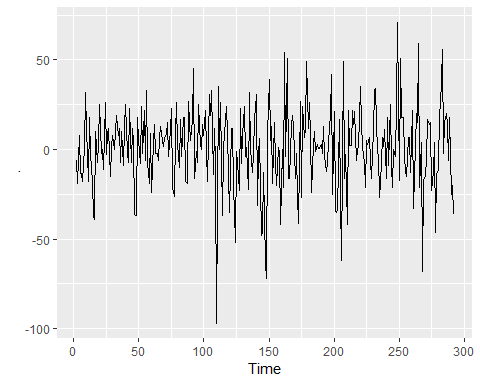
# data overview  
# help(dj)  
autoplot(dj)



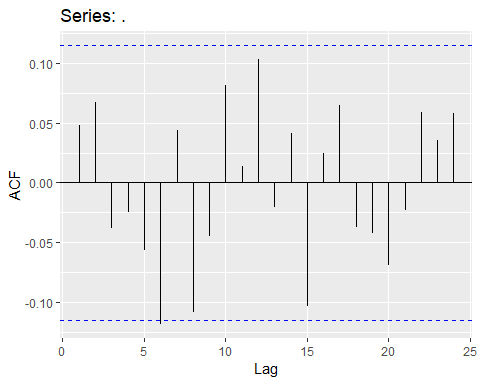
ggAcf(dj)



# DID   
diff(dj) %>%   
 autoplot()



diff(dj) %>%   
ggAcf()



## 2.11 Further reading

* Cleveland (1993) is a classic book on the principles of visualisation for data analysis. While it is more than 20 years old, the ideas are timeless.
* Unwin (2015) is a modern introduction to graphical data analysis using R. It does not have much information on time series graphics, but plenty of excellent general advice on using graphics for data analysis.

Cleveland, W. S. (1993). Visualizing data. Hobart Press. Unwin, A. (2015). Graphical data analysis with R. Chapman; Hall/CRC.

# Chapter 3: The forecaster’s toolbox

In this chapter, we discuss some general tools that are useful for many different forecasting situations. We will describe some benchmark forecasting methods, ways of making the forecasting task simpler using transformations and adjustments, methods for checking whether a forecasting method has adequately utilised the available information, and techniques for computing prediction intervals.

Each of the tools discussed in this chapter will be used repeatedly in subsequent chapters as we develop and explore a range of forecasting methods.

## 3.1 some simple forecasting methods

Some forecasing methods are extremely simple and surprisingly effective. We will use the following four forecasing 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 then we can write the forecasts as

The notation is a short-hand for the estimate of based on the data .

meanf(y,h)  
# y contains the time series  
# h is the forecast horizon

### Naive method

For naive forecasts, we simply set all forecasts to be the value of the last observation. That is,

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

naive(y,h)  
rwf(y,h)

Because a naive forecast is optimal when data follow a random walk (see Section 8.1), these are also called random walk forecast.

### Seasonal naive 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 of time is written as

where = the seasonal period, and is the integer part of (i.e., the number of compete years in the forecast period prior to time ). This looks more complidated than it really is.

For example ,with monthly data, the forecast for all future February values is equalt to the last observed February value. With quartely 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.

snaive(y,h)

### 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 is given by

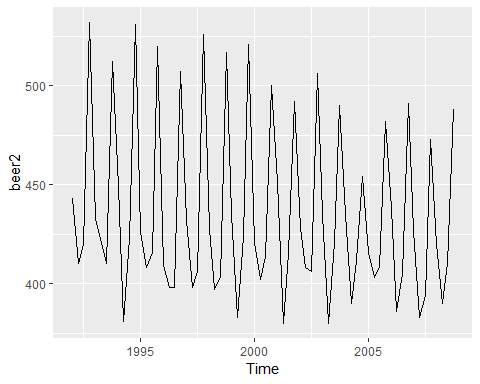
This is equivalent to drawing a line between the first and last observations, and extrapolating it into the future.

rwf(y,j,drift=TRUE)

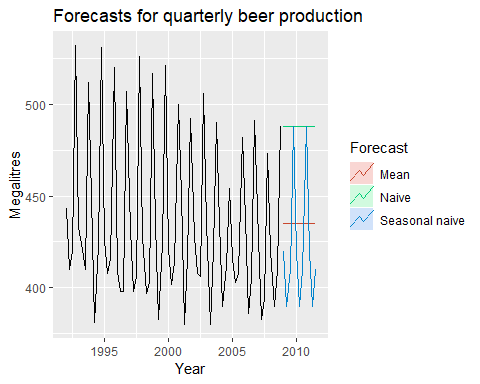
### Examples

Figure 3.1 shows the first three methods applied to the quarterly beer production data.

# Set training data from 1992 to 2007  
beer2 <- window(ausbeer,start=c(1992,1),end=c(2008,4))  
  
# plot some forecasts  
autoplot(beer2)

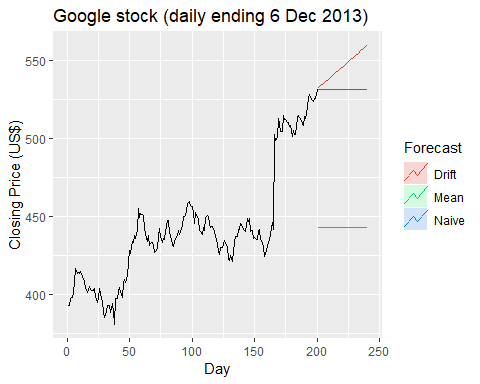


autoplot(beer2)+  
 autolayer(meanf(beer2,h=11),  
 series="Mean",PI=FALSE)+  
 autolayer(naive(beer2,h=11),  
 series="Naive",PI=FALSE)+  
 autolayer(snaive(beer2,h=11),  
 series="Seasonal naive",PI=FALSE)+  
 ggtitle("Forecasts for quarterly beer production")+  
 xlab("Year")+ylab("Megalitres")+  
 guides(colour=guide\_legend(title="Forecast"))



In Figure 3.2, the non-seasonal methods are applied to a series of 200 days of the Google daily closing stock price.

# set training data to first 250 days  
dj2 <- window(dj,end=250)  
  
autoplot(goog200) +  
 autolayer(meanf(goog200, h=40),  
 series="Mean", PI=FALSE) +  
 autolayer(rwf(goog200, h=40),  
 series="Naïve", PI=FALSE) +  
 autolayer(rwf(goog200, drift=TRUE, h=40),  
 series="Drift", PI=FALSE) +  
 ggtitle("Google stock (daily ending 6 Dec 2013)") +  
 xlab("Day") + ylab("Closing Price (US$)") +  
 guides(colour=guide\_legend(title="Forecast"))



Forecasts based on 250 days of the Dow Jones Index.

Sometimes one of these simple methods will be the best forecasting method available; but in many cases, these methods will serve as benchmarks rather than the method of choice. That is, any forecasting methods we develop will be compared to these simple methods to ensure that the new method is better than these simple alternatives. If not, the new method is not worth considering.

## 3.2 Transformations and adjustments

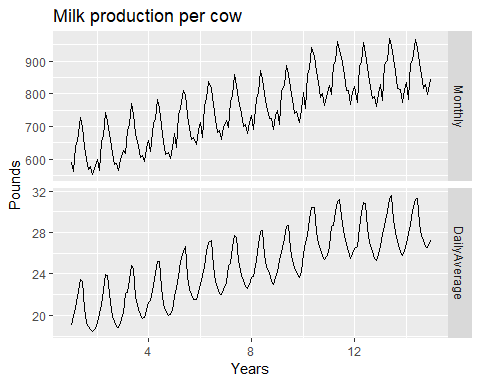
Adjusting the historical data can often lead to a simpler forecasting task. Here, we deal with four kinds of adjustments: calendar adjustments, population adjustments, inflation adjustments and mathematical transformations. The purpose of these adjustments and transformations is to simplify the patterns in the historical data by removing known sources of variation or by making the pattern more consistent across the whole data set. Simpler patterns usually lead to more accurate forecasts.

### Calendar adjustments

Some of the variation seen in seasonal data may be due to simple calender effects. In such cases, it is usually much easier to remove the variation before fitting a forecasting model. The monthdays() function will compute the number of days in each month or quarter.

For example, if you are studying the monthly milk production on a farm, there will be variation between the months simply because of the different numbers of days in each month, in addition to the seasonal variation across the year.

dframe <- cbind(Monthly = milk, DailyAverage=milk/monthdays(milk))   
 autoplot(dframe, facet=TRUE) + xlab("Years") + ylab("Pounds") +  
 ggtitle("Milk production per cow")



Monthly milk production per cow. Source: @Cryer2008.

Notice how much simpler the seasonal pattern is in the average daily production plot compared to the average monthly production plot. By looking at the average daily production instead of the average monthly production, we effectively remove the variation due to the different month lengths. Simpler patterns are usually easier to model and lead to more accurate forecasts.

A similar adjustment can be done for sales data when the number of trading days in each month varies. In this case, the sales per trading day can be modelled instead of the total sales for each month.

### Population adjustments

Any data that are affected by population changes can be adjusted to give per-capita data. That is, consider the data per person (or per thousand people, or per million people) rather than the total. For example, if you are studying the number of hospital beds in a particular region over time, the results are much easier to interpret if you remove the effects of population changes by considering the number of beds per thousand people. Then you can see whether there have been real increases in the number of beds, or whether the increases are due entirely to population increases. It is possible for the total number of beds to increase, but the number of beds per thousand people to decrease. This occurs when the population is increasing faster than the number of hospital beds. For most data that are affected by population changes, it is best to use per-capita data rather than the totals.

### Inflation adjustments

Data which are affected by the value of money are best adjusted before modelling. For example, the average cost of a new house will have increased over the last few decades due to inflation. A $200,000 house this year is not the same as a $200,000 house twenty years ago. For this reason, financial time series are usually adjusted so that all values are stated in dollar values from a particular year. For example, the house price data may be stated in year 2000 dollars.

To make these adjustments, a price index is used. If denotes the price index and denotes the original house price in year , then gives the adjusted house price at year 2000 dollar values. Price indexes are often constructed by government agencies. For consumer goods, a common price index is the Consumer Price Index (or CPI)

### Mathematical transformations

If the data show variation that increases or decreases with the level of the series, then a transformation can be useful. For example, a logarithmic transformation is often useful. If we denote the original observations as and the transformed observations as , then . Logarithms are useful because they are interpretable: changes in a log value are relative (or percentage) changes on the original scale. So if log base 10 is used, then an increase of 1 on the log scale corresponds to a multiplication of 10 on the original scale. Another useful feature of log transformations is that they constrain the forecasts to stay positive on the original scale.

Sometimes other transformations are also used (although they are not so interpretable). For example, square roots and cube roots can be used. These are called *power transformations* because they can be written in the form .

A useful family of transformations, that includes both logarithms and power transformations, is the family of “Box-Cox transformations”, which depend on the parameter and are defined as follows:

The logarithm in a Box-Cox transformation is always a natural logarithm (i.e., to base ). So if , natural logarithms are used, but if , a power transformation is used, followed by some simple scaling.

If , then , so the transformed data is shifted downwards but there is no change in the shape of the time series. But for all other values of , the time series will change shape.

{r BoxCoxShiny, child=childdoc,eval=FALSE}

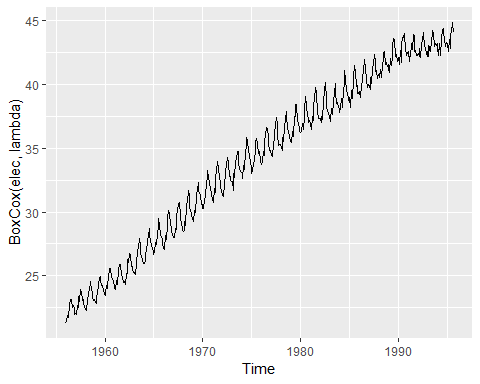
A good value of is one which makes the size of the seasonal variation about the same across the whole series, as that makes the forecasting model simpler. In this case, works quite well, although any value of between 0 and 0.5 would give similar results.

The BoxCox.Lambda function will choose a value of lambda for your.

(lambda <- BoxCox.lambda(elec))

## [1] 0.2654076

autoplot(BoxCox(elec,lambda))



Having chosen a transformation, we need to forecast the transformed data. Then, we need to reverse the transformation (or *back-transform*) to obtain forecasts on the original scale. The reverse Box-Cox transformation is given by

### Features of power transformations

* If some $y\_t≤0, no power transformation is possible unless all observations are adjusted by adding a constant to all values.
* Choose a simple value of . It makes explanations easier.
* The forecasting results are relatively insensitive to the value of .
* Often no transformation is needed.
* Transformations sometimes make little difference to the forecasts but have a large effect on prediction intervals.

### Bias adjustments

One issue with using mathematical transformations such as Box-Cox transformations is that the back-transformed forecast will not be the mean of the forecast distribution. In fact, it will usually be the median of the forecast distribution (assuming that the distribution on the transformed space is symmetric). For many purposes, this is acceptable, but occasionally the mean forecast is required. For example, you may wish to add up sales forecasts from various regions to form a forecast for the whole country. But medians do not add up, whereas means do.

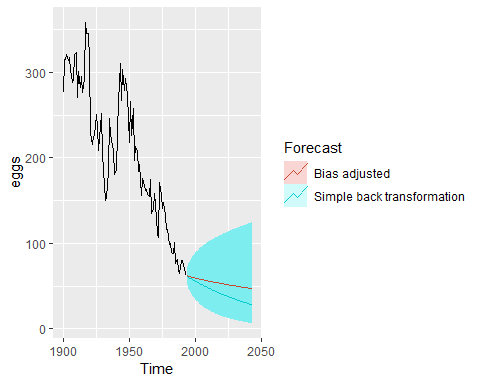
For a Box-Cox transformation, the back-transformed mean is given by

where, is the -step forecast variance. The larger the forecast variance, the bigger the difference between the mean and the median.

The difference between the simple back-transformed forecast given by (3.1) and the mean given by (3.2) is called the **bias**. When we use the mean, rather than the median, we say the point forecasts have been **bias-adjusted**.

To see how much difference this bias-adjustment makes, consider the following example, where we forecast average annual price of eggs using the drift method with a log transformation (). The log transformation is useful in this case to ensure the forecasts and the prediction intervals stay positive.

fc <- rwf(eggs,drift=TRUE,lambda=0,h=50,level=80)  
fc2 <- rwf(eggs,drift=TRUE,lambda=0,h=50,level=80,  
 biasad=TRUE)  
  
autoplot(eggs)+  
 autolayer(fc,series="Simple back transformation")+  
 autolayer(fc2,series="Bias adjusted",PI=FALSE)+  
 guides(colour=guide\_legend(title="Forecast"))



The blue line in Figure 3.4 shows the forecast medians while the red line shows the forecast means. Notice how the skewed forecast distribution pulls up the point forecast when we use the bias adjustment.

Bias adjustment is not done by default in the forecast package. If you want your forecasts to be means rather than medians, use the argument biasadj=TRUE when you select your Box-Cox transformation parameter.

## 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 , meaning the forecast of based on observations ,…,. We use these so often, we sometimes drop part of the subscript and just write  
 instead of . 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:

, where is the average computed over all available observations, including those at times *aftert*. Similarly, for the drift method, the drift parameter is estimated using all available observations. In this case, the fitted values are given by

, where . 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 observation after time *t*, the fitted values are not true forecasts. On the other hand, naive or seasonal naive 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 correspondending fitted values:

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 residual which should be used in computing forecasts.
2. The reisiduals 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.

In addition to these essential properties, it is useful (bot 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). However, a forecasting method that does not satisfy these properties annot 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.

### Example: Forecasting the Google daily closing stock price

For stock market prices and indexes, the best forecasting method is often the **naive method**. That is, each forecast is simply equal to the last observied value, . Hence, the residuals are simply equal to the difference between consecutive observations:

The following graph shows the Google daily closing stock price (GOOG). The large jump at day 166 corresponds to 18 October 2013 when the price jumped 12% due to unexpectedly strong third quarter results.

autoplot(goog200)+  
 xlab("Day")+  
 ylab("Closing Price (USD$)")+  
 ggtitle("Google Stock (daily ending 6 December 2013)")

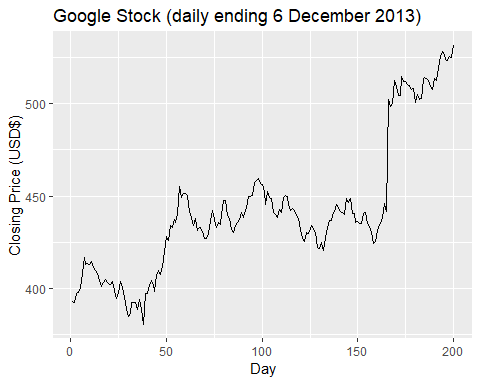


Figure 3.5: The daily Google stock price to 6 Dec 2013

goog200 %>%   
 naive() %>%   
 autoplot()

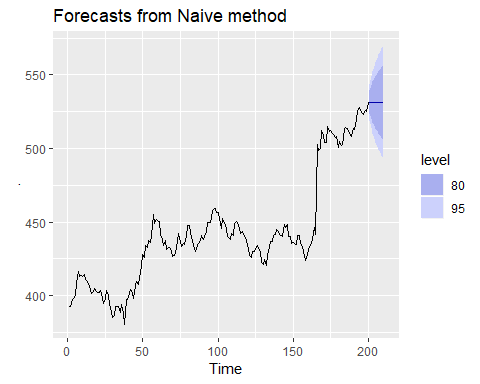


Figure 3.5: The daily Google stock price to 6 Dec 2013

res <- residuals(naive(goog200))  
autoplot(res)+  
 labs(  
 x="Day",  
 y=""  
 )+  
 ggtitle("Residuals from naive method")

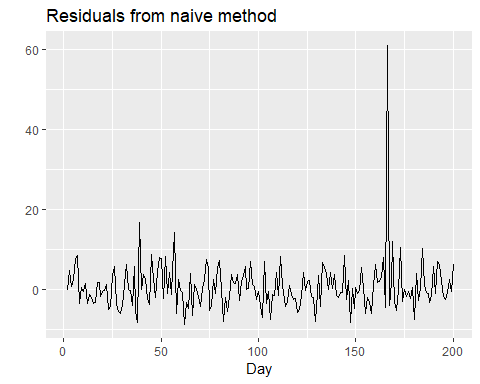
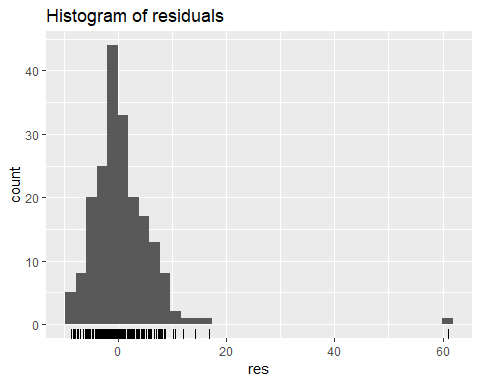


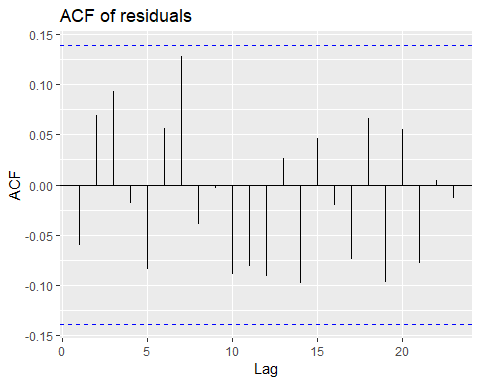
Figure 3.6: Residuals from forecasting the Google stock price using the naive method.

gghistogram(res) + ggtitle("Histogram of residuals")

## Warning: Removed 1 rows containing non-finite values (stat\_bin).



ggAcf(res)+  
 ggtitle("ACF of residuals")

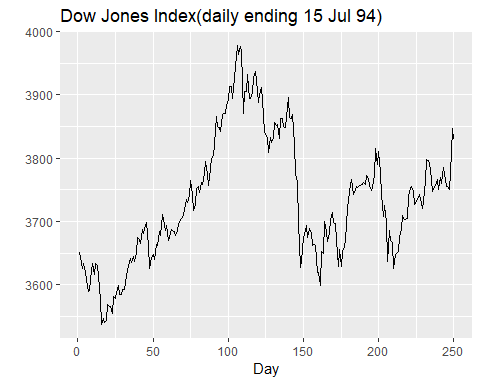


Old example from Dow Jones(dj) data.

dj2 <- window(dj,end=250)  
head(dj2,10)

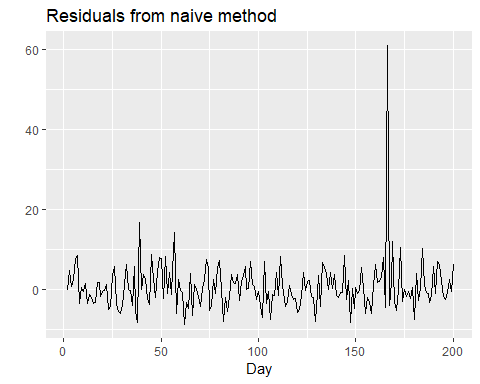
## Time Series:  
## Start = 1   
## End = 10   
## Frequency = 1   
## [1] 3651.0 3645.0 3626.0 3634.0 3620.5 3607.0 3589.0 3590.0 3622.0 3634.0

autoplot(dj2)+  
 labs(  
 x="Day",  
 y=""  
 )+  
 ggtitle("Dow Jones Index(daily ending 15 Jul 94)")



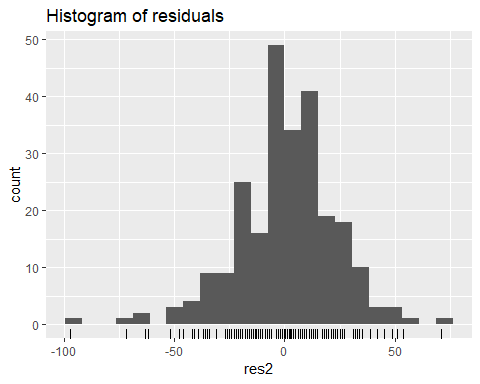
Residuals from forecasting the Dow Jones Index using the naive method.

res2 <- residuals(naive(dj2))  
autoplot(res)+  
 labs(  
 x="Day",  
 y=""  
 )+  
 ggtitle("Residuals from naive method")



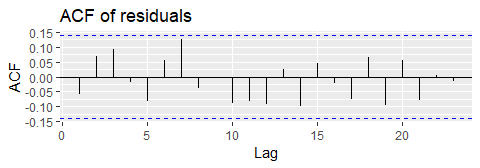
Residuals from forecasting the Dow Jones Index using the naive method.

gghistogram(res2) + ggtitle("Histogram of residuals")



Histogram of the residuals from the naive method applied to the Dow Jones Index. The left tail is a little too long for a normal distribution.

ggAcf(res) + ggtitle("ACF of residuals")



ACF of the residuals from the naive method applied to the Dow Jones Index. The lack of correlation suggesting the forecasts are good.

These graphs show that the naive method produces forecasts that appear to account for all available information. The mean of the residuals is close to zero, and there is no significant correlation in the residual series.

The time plot of the residuals shows that the variation can be treated as constant. This can be seen on the histogram of the residuals. The hisogram suggests that the residuals may not be normal — the right tail seems a little too long, even when we ignore the outlier. Consequently, forecasts from this method will probably be quite good, but prediction intervals that are computed assuming a normal distribution may be inaccurate.

### Portmanteau tests for autocorrelation

In addition to looking at the ACF plot, we can also do more formal test for autocorrelation by considering a whole set of values as a group, rather than treating each one separately.

Recal that is the autocorrelation for lag . When we look at the ACF plot to see whether each spike is within the required limits, we are implicitly carrying out multiple **hypothesis tests**, each one with a small probability of giving a false positive. When enough of these tests are done, it is likely that at least one will give a false positive, and so we may conclude that the residuals have some remaining autocorrelation, when in fact they do not.

In order to overcome this problem, we test whether the first autocorrelations are significantly different from what would be expected from a white noise process. A test for a group of autocorrelation is called a **portmanteau test** from a French word describing a suitcase containing number of items.

One such test is the **Box-Pierce test**, based on the following statistic

, where is the maximum lag being considered and is the number of observations. If each is close to zero, then will be small. IF some values are large, then will be large.

We suggest using for non-seasonal data and for seasonal data, where is the period of seasonality. However, the test is not good when is large, so if these values are larger than then use

A related (and more accurate) test is the **Ljung-Box test**, based on

Again, **larger** values of suggests that the autocorrelation do not come from a white noise.

How large is too large? If the autocorrelation did come form a white noise series, then both Q and Q\* would have a distribution with degrees of freedom, where is the number of parameters in the model. If they are calculated from raw data (rather than the residuals from a model), then set .

For the Google stock price example, the naive model has no parameters, so **K=0** in that case also.

# Lag=h and fitdf=K  
Box.test(res,lag=10,fitdf=0)

##   
## Box-Pierce test  
##   
## data: res  
## X-squared = 10.611, df = 10, p-value = 0.3886

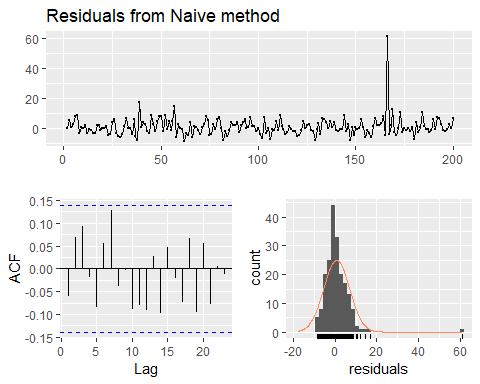
Box.test(res,lag=10,fitdf=0,type="Lj")

##   
## Box-Ljung test  
##   
## data: res  
## X-squared = 11.031, df = 10, p-value = 0.3551

For both and , the results are not significant(i.e.,the p-values are relatively large). Thus, we can conclude that the residuals are not ditinguishable from a white noise series.

All of these methods for checking residuals are conveniently packaged into one R function checkresiduals(), which will produce a time plot, ACF plot and histogram of the residuals (with an overlaid normal distribution for comparison), and do a Ljung=Box test with the correct degrees of freedom.

checkresiduals(naive(goog200))



##   
## Ljung-Box test  
##   
## data: Residuals from Naive method  
## Q\* = 11.031, df = 10, p-value = 0.3551  
##   
## Model df: 0. Total lags used: 10

## 3.4 Evaluation 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 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.

### Functions to subset a time series

The windows function introduced in Chapter2 is useful when extracting a portion of a time series, such as we need when creating training and test sets. In the window(), we specify the start and/or end of th portion of time series requred using time values. for example,

window(ausbeer,start=1995)

## Qtr1 Qtr2 Qtr3 Qtr4  
## 1995 426 408 416 520  
## 1996 409 398 398 507  
## 1997 432 398 406 526  
## 1998 428 397 403 517  
## 1999 435 383 424 521  
## 2000 421 402 414 500  
## 2001 451 380 416 492  
## 2002 428 408 406 506  
## 2003 435 380 421 490  
## 2004 435 390 412 454  
## 2005 416 403 408 482  
## 2006 438 386 405 491  
## 2007 427 383 394 473  
## 2008 420 390 410 488  
## 2009 415 398 419 488  
## 2010 414 374

extracts all data from 1995 onward.

Another useful function is subset() which allows for more types of subsetting. A great advantage of this function is that it allows the use of indices to choose a subset. For example,

subset(ausbeer,start=length(ausbeer)-4\*5)

## Qtr1 Qtr2 Qtr3 Qtr4  
## 2005 403 408 482  
## 2006 438 386 405 491  
## 2007 427 383 394 473  
## 2008 420 390 410 488  
## 2009 415 398 419 488  
## 2010 414 374

extracts the last 5 years of observation from ausbeer. It also allows extracting all values for a specific reason. For example,

subset(ausbeer,quarter=1)

## Time Series:  
## Start = 1956   
## End = 2010   
## Frequency = 1   
## [1] 284 262 272 261 286 295 279 294 313 331 335 353 393 383 387 410 419  
## [18] 458 465 500 510 486 515 503 513 548 493 475 453 464 459 481 474 467  
## [35] 485 464 443 433 449 426 409 432 428 435 421 451 428 435 435 416 438  
## [52] 427 420 415 414

Finally, head and tail are useful for extracting the first few or last few observations. For example, the last 5 years of ausbeer can also be obtained using

tail(ausbeer,4\*5)

## Qtr1 Qtr2 Qtr3 Qtr4  
## 2005 408 482  
## 2006 438 386 405 491  
## 2007 427 383 394 473  
## 2008 420 390 410 488  
## 2009 415 398 419 488  
## 2010 414 374

### 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

,where the training data is given by and the test data is given by .

Note that forecast errors are different from residuals in two ways. First, residuals are calculated on the while forecast errors are calculated on the . Second, residuals are based on 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 are therefore scale-dependent and cannot be used to make comparison between series that involve different units.

The two most commonly used scale-dependent measures are based on the absolute errors or square errors:

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 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:

Measures basedd on performance errors have the disadvantage of being infinite or underdefined if at any in the period of interest, , and having extreme values if any 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.

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 hanve 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, which was used in the forecasting competition. It is defined by

However, if is close to zero, 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 error” at all.

Hyndman & Koehler (2006) 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) 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 naive forecasts:

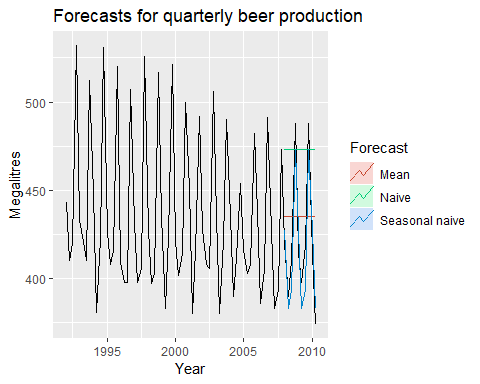
Because the numerator and denominator both involve values on the scale of the original data, is independent of the scale of the data. A scaled error is less than one if it arises from a better forecast than the average naïve forecast computed on the training data. Conversely, it is greater than one if the forecast is worse than the average naïve forecast computed on the training data.

For seasonal time series, a scaled error can be defued using seasonal naive forecasts:

The *mean absolute scaled error* si simply

#### Examples

beer2 <- window(ausbeer,start=1992,end=c(2007,4))  
beerfit1 <- meanf(beer2,h=10)  
beerfit2 <- rwf(beer2,h=10)  
beerfit3 <- snaive(beer2,h=10)  
autoplot(window(ausbeer, start=1992)) +  
 autolayer(beerfit1, series="Mean", PI=FALSE) +  
 autolayer(beerfit2, series="Naïve", PI=FALSE) +  
 autolayer(beerfit3, series="Seasonal naïve", PI=FALSE) +  
 xlab("Year") + ylab("Megalitres") +  
 ggtitle("Forecasts for quarterly beer production") +  
 guides(colour=guide\_legend(title="Forecast"))



Forecasts of Australian quarterly beer production using data up to the end of 2007.

Figure 3.9 shows three forecasting methods applied to the quartely Australian beer production using data only to the end of 2007. The actual values for the period 2008-2010 are also shown. We compute the forecast accuracy measures for this period.

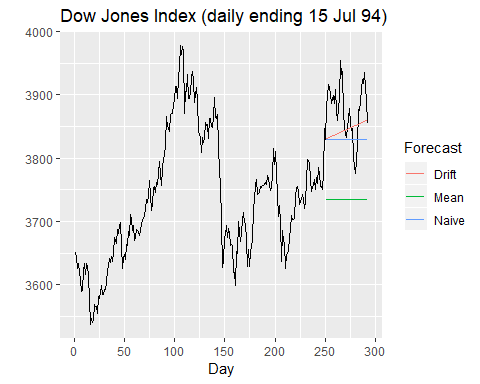
beer3 <- window(ausbeer, start=2008)  
accuracy(beerfit1, beer3)  
accuracy(beerfit2, beer3)  
accuracy(beerfit3, beer3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RMSE | MAE | MAPE | MASE |
| Mean method | 38.45 | 34.83 | 8.28 | 2.44 |
| Naive method | 62.69 | 57.40 | 14.18 | 4.01 |
| Seasonal naive method | 14.31 | 13.40 | 3.17 | 0.94 |

It is obvious from the graph that the seasonal naïve method is best for these data, although it can still be improved, as we will discover later. Sometimes, different accuracy measures will lead to different results as to which forecast method is best. However, in this case, all of the results point to the seasonal naïve method as the best of these three methods for this data set.

To take a non-seasonal example, consider the Google stock price. The following graph shows the 200 observations ending on 6 Dec 2013, along with forecasts of the next 40 days obtained from three different methods.

dj2 <- window(dj, end=250)  
djfc1 <- meanf(dj2, h=42)  
djfc2 <- rwf(dj2, h=42)  
djfc3 <- rwf(dj2, drift=TRUE, h=42)  
autoplot(dj) +  
 forecast::autolayer(djfc1$mean, series="Mean") +  
 forecast::autolayer(djfc2$mean, series="Naïve") +  
 forecast::autolayer(djfc3$mean, series="Drift") +  
 xlab("Day") + ylab("") +  
 ggtitle("Dow Jones Index (daily ending 15 Jul 94)") +  
 guides(colour=guide\_legend(title="Forecast"))



Forecasts of the Dow Jones Index from 16 July 1994.

dj3 <- window(dj, start=251)  
accuracy(djfc1, dj3)  
accuracy(djfc2, dj3)  
accuracy(djfc3, dj3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RMSE | MAE | MAPE | MASE |
| Mean method | 148.24 | 142.42 | 3.66 | 8.70 |
| Naive method | 62.03 | 54.44 | 1.40 | 3.32 |
| Drift method | 53.70 | 45.73 | 1.18 | 2.79 |

### Time series cross validation

A more sophisticated version of training/test sets is time series cross-validation. In this procedure, there are a series of test sets, each consisting of a single observation. The corresponding training set consists only of observations that occurred prior to the observation that forms the test set. Thus, no future observations can be used in constructing the forecast. Since it is not possible to obtain a reliable forecast based on a small training set, the earliest observations are not considered as test sets.

The following diagram illustrates the series of training and test sets, where the blue observations form the training sets, and the red observations form the test sets.

The forecast accuracy is computed by averaging over the test sets. This procedure is sometimes known as “evaluation on a rolling forecasting origin” because the “origin” at which the forecast is based rolls forward in time.

With time series forecasting, one-step forecasts may not be as relevant as multi-step forecasts. In this case, the cross-validation procedure based on a rolling forecasting origin can be modified to allow multi-step errors to be used. Suppose that we are interested in models that produce good -step-ahead forecasts. Then the corresponding diagram is shown below.

Time series cross-validation is implemented with the tsCV() function. In the following example, we compare the RMSE obtained via time series cross-validation with the residual RMSE.

## [1] 22.68249

## [1] 22.49681

As expected, the *RMSE* from the residuals is smaller, as the corresponding “**forecasts**” are based on a model fitted to the entire data set, rather than being true forecasts.

A good way to choose the best forecasting model is to find the model with the smallest RMSE computed using time series cross-validation.

### Pipe operator

The ugliness of the above R code makes this a good opportunity to introduce some alternative ways of stringing R functions together. In the above code, we are nesting functions within functions within functions, so you have to read the code from the inside out, making it difficult to understand what is being computed. Instead, we can use the pipe operator %>% as follows.

dj %>% tsCV(forecastfunction=rwf,drift=TRUE,h=1) ->e  
e^2 %>% mean(na.rm=TRUE) %>% sqrt()

## [1] 22.68249

dj %>% rwf(drift=TRUE) %>% residuals() -> res  
res^2 %>% mean(na.rm=TRUE) %>% sqrt()

## [1] 22.49681

The left hand side of each pipe is passed as the first argument to the function on the right hand side. This is consistent with the way we read from left to right in English. When using pipes, all other arguments must be named, which also helps readability. When using pipes, it is natural to use the right arrow assignment -> rather than the left arrow. For example, the third line above can be read as “Take the dj series, pass it to rwf() with drift=TRUE, compute the resulting residuals, and store them as res”.

We will use the pipe operator whenever it makes the code easier to read. In order to be consistent, we will always follow a function with parentheses to differentiate it from other objects, even if it has no arguments. See, for example, the use of sqrt() and residuals() in the code above.

#### Example: using tsCV()

The dj data, plotted in the above figure, includes daily closing stock price of Google Inc from the NASDAQ exchange for 200 consecutive trading days starting on 25 February 2013.

The code below evaluates the forecasting performance of 1- to 8-step-ahead naïve forecasts with tsCV(), using MSE as the forecast error measure. The plot shows that the forecast error increases as the forecast horizon increases, as we would expect.

e <- tsCV(goog200,forecastfunction = naive,h=8)  
# Comute the MSE values and remove missing values  
mse <- colSums(e^2,na.rm = TRUE)  
# Plot the MSE values against the forecast horizon  
data.frame(h=1:8,MSE=mse) %>%   
 ggplot(aes(x=h,y=MSE))+  
 geom\_point()+  
 geom\_line()



#### Bibliography

Armstrong, J. S. (1978). Long-range forecasting: From crystal ball to computer. John Wiley & Sons. [Amazon]

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22, 679–688. <https://robjhyndman.com/publications/automatic-forecasting/>

## 3.5 Prediction intervals

As discussed in Section 1.7, a prediction interval gives an interval within which we expect to lie with a specified probability.

For example, assuming the forecast errors are normally distributed, a 95% prediction interval for the step forecast is

, where is an estimate of the standard deviation of the -step forecast distribution.

More generally, a prediction interval can be written as $$ +-c

$$ here the multiplier depends on the coverage probability. In this book we usually calculate 80% intervals and 95% intervals, although any percentage may be used. The following table gives the value of for a range of coverage probabilities assuming normally distributed forecast errors.

Multipliers to be used for prediction intervals.

|  |  |
| --- | --- |
| Percentage | Multiplier |
| 50 | 0.67 |
| 55 | 0.76 |
| 60 | 0.84 |
| 65 | 0.93 |
| 70 | 1.04 |
| 75 | 1.15 |
| 80 | 1.28 |
| 85 | 1.44 |
| 90 | 1.64 |
| 95 | 1.96 |
| 96 | 2.05 |
| 97 | 2.17 |
| 98 | 2.33 |
| 99 | 2.58 |

The value of prediction intervals is that they express the uncertainty in the forecasts. If we only produce point forecasts, there is no way of telling how accurate the forecasts are. However, if we also produce prediction intervals, then it is clear how much uncertainty is associated with each forecast. For this reason, point forecasts can be of almost no value without the accompanying prediction intervals.

### One-step prediction intervals

When forecasting one step ahead, the standard deviation of the forecast distribution is almost the same as the standard deviation of the residuals (In fact, the two standard deviations are ientical if there are no parameters to be estimated, as is the case with the naive method. For forecasting methods involving parameters to be estimated, the standard deviation of the forecast distribution is slightly larger than the residual standard deviation, although this difference is often ignored.)

For example, consider a naïve forecast for the Google stock price data goog200 (shown in Figure 3.5). The last value of the observed series is 531.48, so the forecast of the next value of the GSP is 531.48. The standard deviation of the residuals from the naïve method is 6.21. Hence, a 95% prediction interval for the next value of the GSP is

Similarly, an 80% prediction interval is given by

The value of the multiplier (1.96 or 1.28) is taken from Table @ref(tab:pcmultipliers).

### Multi-step prediction intervals

A common feature of prediction intervals is that they increase in length as the forecast horizon increases. The further ahead we forecast, the more uncertainty is associated with the forecast, and thus the wider the prediction intervals. That is, usually increases with (although there are some non-linear forecasting methods that do not have this property).

To produce a prediction interval, it is necessary to have an estimate of . As already noted, for one-step forecasts (), the residual standard deviation provides a good estimate of the forecast standard deviation . For multi-step forecasts, a more complicated method of calculation is required. These calculations assume that the residuals are uncorrelated.

### Benchmark methods

For the four benchmark methods, it is possible to mathematically derive the forecast standard deviation under the assumption of uncorrelated residuals. If denotes the standard deviation of the -step forecast distribution, and is the residual standard deviation, then we can use the following expressions.

**Mean forecasts**:

**Naïve forecasts**:

**Seasonal naïve forecasts** .

**Drift forecasts**: .

Note that when and is large, these all give the same approximate value .

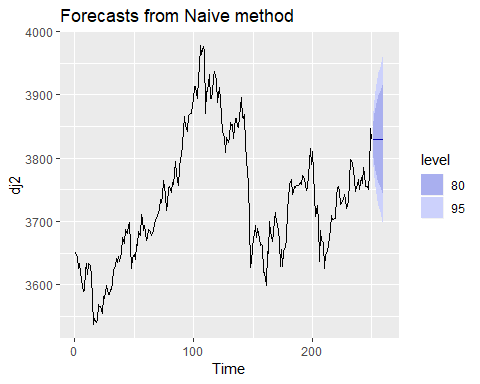
Prediction intervals will be computed for you when using any of the benchmark forecasting methods. For example, here is the output when using the naive method for the Dow-Jones index.

naive(dj2)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 251 3830 3801.804 3858.196 3786.878 3873.122  
## 252 3830 3790.125 3869.875 3769.016 3890.984  
## 253 3830 3781.163 3878.837 3755.311 3904.689  
## 254 3830 3773.608 3886.392 3743.756 3916.244  
## 255 3830 3766.952 3893.048 3733.577 3926.423  
## 256 3830 3760.935 3899.065 3724.373 3935.627  
## 257 3830 3755.401 3904.599 3715.910 3944.090  
## 258 3830 3750.250 3909.750 3708.033 3951.967  
## 259 3830 3745.412 3914.588 3700.634 3959.366  
## 260 3830 3740.837 3919.163 3693.637 3966.363

When plotted, the prediction intervals are shown as shaded region, with the strength of colour indicating the probability associated with the interval.

autoplot(naive(dj2))



### prediction intervals from bootstrapped residuals

When a normal distribution for the forecast errors is an unreasonable assumption, one alternative is to use bootstrapping, which only assumes that the forecast errors are uncorrelated.

A forecast error is defined as . We can re-write this as

So we can simulate the next observation of a time series using

where is the one-step forecast and is the unknown future error. Assuming future errors will be similar to past errors, we can replace by sampling from the collection of errors we have seen in the past (i.e., the residuals). Adding the new simulated observation to our data set, we can repeat the process to obtain

where is another draw from the collection of residuals. Continuing in this way, we can simulate an entire set of future values for our time series.

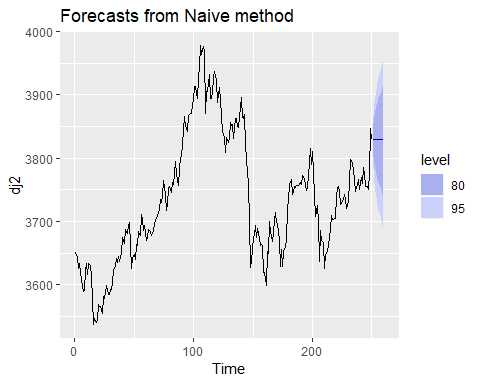
Doing this repeatedly, we obtain many possible futures. Then we can compute prediction intervals by calculating percentiles for each forecast horizon. The result is called a “bootstrapped” prediction interval. The name “bootstrap” is a reference to pulling ourselves up by our bootstraps, because the process allows us to measure future uncertainty by only using the historical data.

To generate such intervals, we can simply add the bootstrap argument to our forecasting functions. For example:

naive(dj2,bootstrap = T)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 251 3830 3803.269 3854.269 3783.269 3871.269  
## 252 3830 3790.539 3866.539 3762.539 3886.539  
## 253 3830 3781.808 3874.808 3747.808 3899.808  
## 254 3830 3774.078 3884.078 3739.078 3909.506  
## 255 3830 3767.347 3891.347 3730.689 3920.347  
## 256 3830 3761.300 3896.616 3718.787 3929.616  
## 257 3830 3756.386 3902.886 3710.227 3940.215  
## 258 3830 3752.155 3908.155 3704.826 3947.155  
## 259 3830 3746.608 3913.425 3693.595 3954.254  
## 260 3830 3742.694 3918.290 3686.694 3958.694

naive(dj2,bootstrap = T) %>%   
 autoplot()



In this case, they are very similar (but not identical) to the prediction intervals based on the normal distribution.

### Prediction intervals with transformations

If a transformation has been used, then the prediction interval should be computed on the transformed scale, and the end points back-transformed to give a prediction interval on the original scale. This approach preserves the probability coverage of the prediction interval, although it will no longer be symmetric around the point forecast.

The back-transformation of prediction intervals is done automatically using the functions in the forecast package in R, provided you have used the lambda argument when computing the forecasts.

## 3.6 The forecast package in R

This book uses the facilities in the **forecast** package in R (which is loaded automatically whenever you load the **fpp2** package). This appendix briefly summarises some of the features of the package. Please refer to the help files for individual functions to learn more, and to see some examples of their use.

### Functions that output a forecast object:

Many functions, including meanf, naive, snaive and rwf, produce output in the form of a forecast object (i.e., an object of class forecast). This allows other functions (such as autoplot) to work consistently across a range of forecasting models.

Objects of class forecast contain information about the forecasting method, the data used, the point forecasts obtained, prediction intervals, residuals and fitted values. There are several functions designed to work with these objects including autoplot, summary and print.

The following list shows all the functions that produce forecast objects.

* meanf()
* naive(), snaive()
* rwf()
* croston()
* stlf()
* ses()
* holt(), hw()
* splinef()
* thetaf()
* forecast()

### forecast() function

So far we have used functions which produce a forecast object directly. But a more common approach, which we will focus on in the rest of the book, will be to fit a model to the data, and then use the forecast() function to produce forecasts from that model.

The forecast() function works with many different types of input. It generally takes a time series or time series model as its main argument, and produces forecasts appropriately. It always returns objects of class forecast.

If the first argument is of class ts, it returns forecasts from the automatic ETS algorithm discussed in Chapter @ref(ch-expsmooth).

Here is a simple example, applying forecast() to the ausbeer data:

ausbeer %>%   
 as.data.frame() %>%   
 tail(5)

## x  
## 214 398  
## 215 419  
## 216 488  
## 217 414  
## 218 374

forecast(ausbeer)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2010 Q3 404.6025 385.8695 423.3355 375.9529 433.2521  
## 2010 Q4 480.3982 457.5283 503.2682 445.4216 515.3748  
## 2011 Q1 417.0367 396.5112 437.5622 385.6456 448.4277  
## 2011 Q2 383.0996 363.5063 402.6930 353.1341 413.0651  
## 2011 Q3 403.0006 380.1438 425.8574 368.0442 437.9570  
## 2011 Q4 478.4943 450.1973 506.7913 435.2177 521.7708  
## 2012 Q1 415.3821 389.6837 441.0806 376.0797 454.6845  
## 2012 Q2 381.5781 356.8145 406.3418 343.7053 419.4509

That works quite well if you have no idea what sort of model to use. But by the end of this book, you should not need to use forecast() in this “blind” fashion. Instead, you will fit a model appropriate to the data, and then use forecast() to produce forecasts from that model.

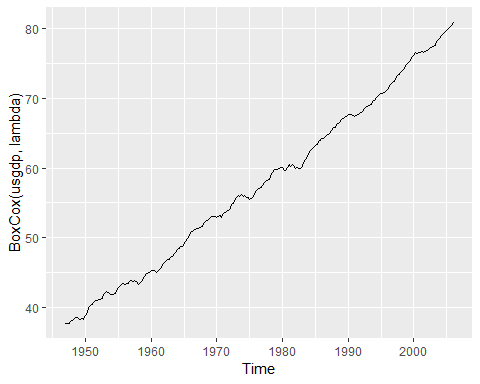
## Exercises

**1.** For the following series, find an appropriate Box-Cox transformation in order to stabilise the variance. *usenetelec* usgdp *mcopper* enplanements

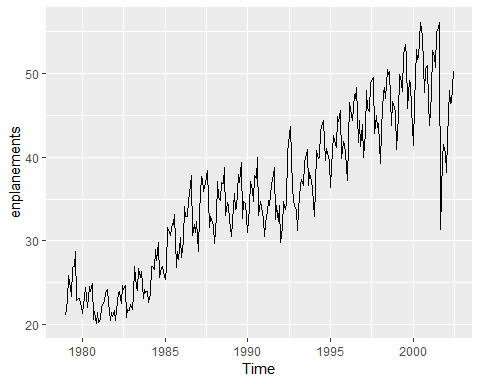
library(expsmooth)  
  
# Annual US net electricity generation  
expsmooth::usgdp %>%   
 head() %>%   
 autoplot()



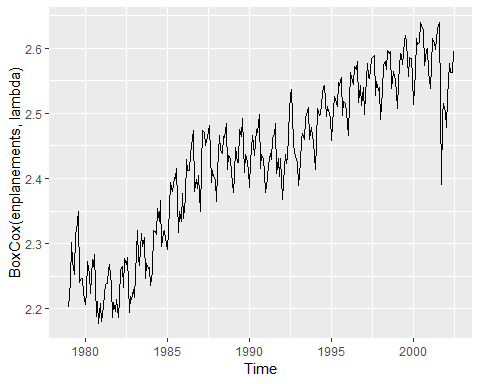
# Box-cox transaformation (Please refer 3.2)  
lambda <- BoxCox.lambda(usgdp)  
autoplot(BoxCox(usgdp,lambda))



# # enplanements  
# Monthly US domestic enplanements  
autoplot(enplanements)



lambda <- BoxCox.lambda(enplanements)  
autoplot(BoxCox(enplanements,lambda))

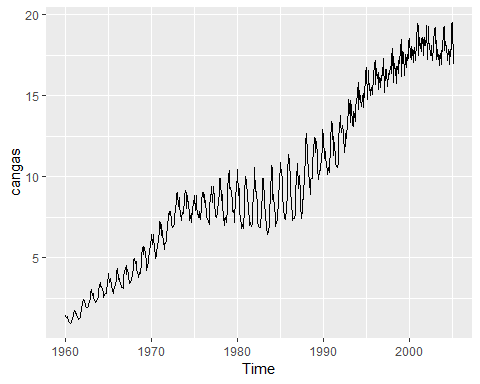


**2**. Why is a Box-Cox transformation for the cangas data?

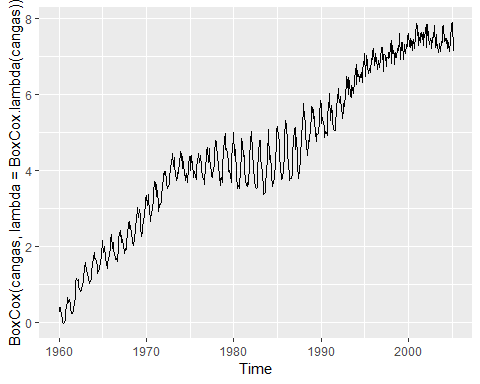
expsmooth::cangas %>%   
 head()

## Jan Feb Mar Apr May Jun  
## 1960 1.4306 1.3059 1.4022 1.1699 1.1161 1.0113

# Monthly Canadian gas production  
  
autoplot(cangas)



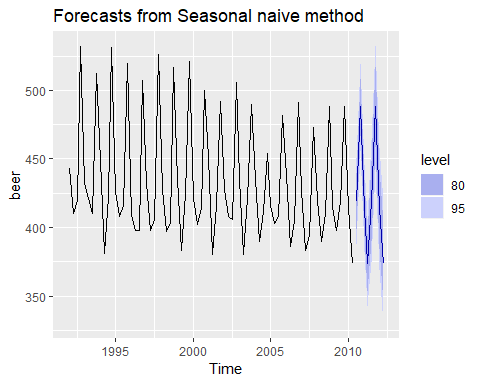
autoplot(BoxCox(cangas,lambda=BoxCox.lambda(cangas)))



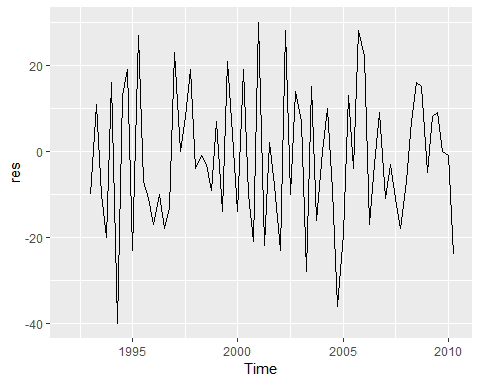
# We don't see a huge change in Box-Cox transformed data

1. What Box-Cox transformation would you select for your retail data (from Exercise 3 in Section @ref(ex-graphics))?
2. Calculate the residuals from a seasonal naïve forecast applied to the quarterly Australian beer production data from 1992. The following code will help.

beer <- window(ausbeer, start=1992)  
fc <- snaive(beer)  
autoplot(fc)

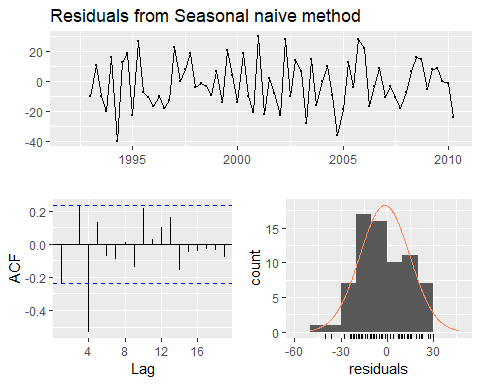


res <- residuals(fc)  
autoplot(res)



Test if the residuals are white noise and normally distributed.

checkresiduals(fc)

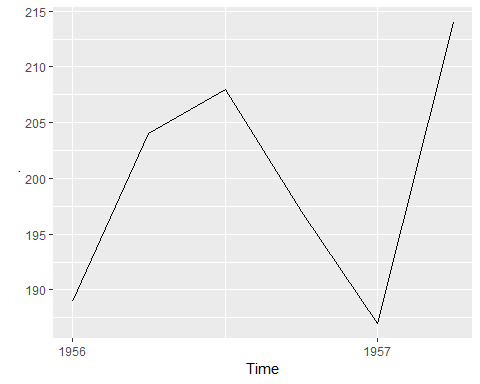


##   
## Ljung-Box test  
##   
## data: Residuals from Seasonal naive method  
## Q\* = 32.269, df = 8, p-value = 8.336e-05  
##   
## Model df: 0. Total lags used: 8

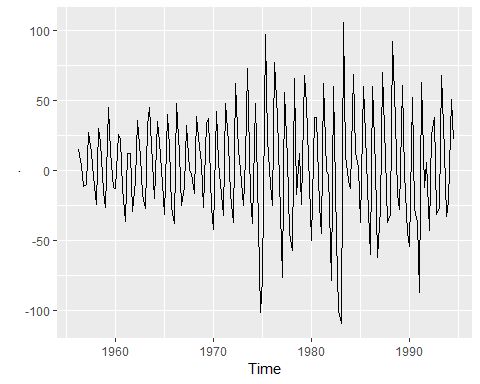
What do you conclude? From the ACF, there is a value exceeding the threshold.

1. Repeat the exercise for the WWWusage and bricksq data. Use whichever of naive or snaive is more appropriate in each case.

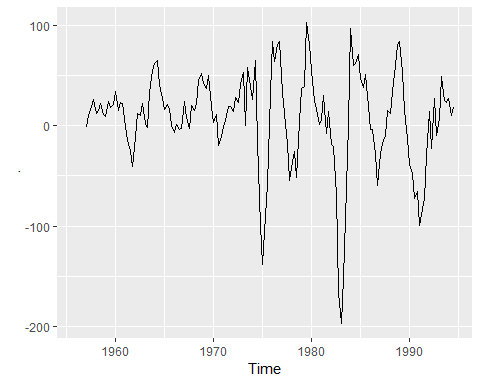
fma::bricksq %>%   
 head() %>%   
 autoplot()



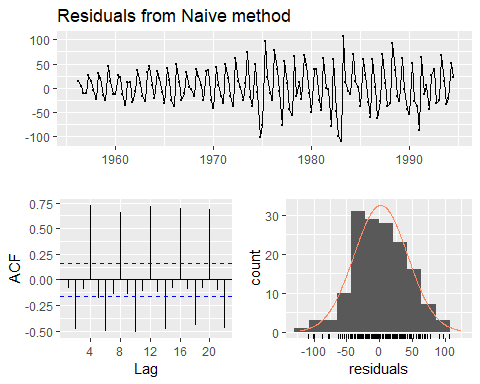
# Quarterly clay brick production  
  
fc1 <- naive(bricksq)  
fc2 <- snaive(bricksq)  
  
residuals(fc1) %>%   
 autoplot()



residuals(fc2) %>%   
 autoplot()

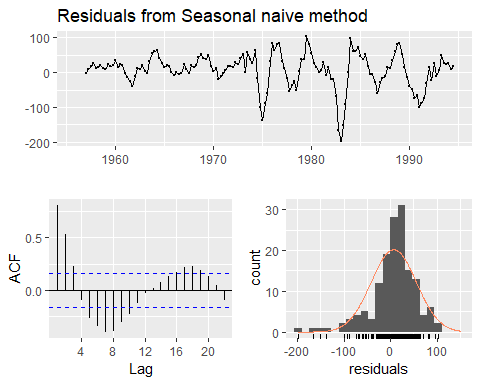


checkresiduals(fc1)



##   
## Ljung-Box test  
##   
## data: Residuals from Naive method  
## Q\* = 244.43, df = 8, p-value < 2.2e-16  
##   
## Model df: 0. Total lags used: 8

checkresiduals(fc2)

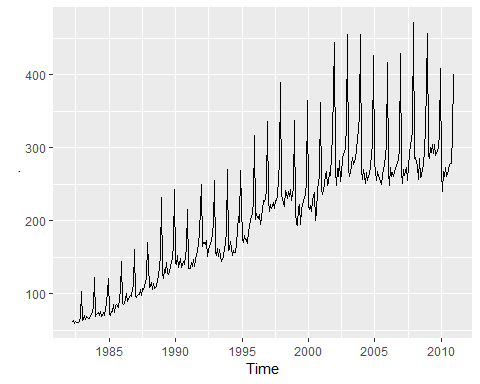


##   
## Ljung-Box test  
##   
## data: Residuals from Seasonal naive method  
## Q\* = 233.2, df = 8, p-value < 2.2e-16  
##   
## Model df: 0. Total lags used: 8

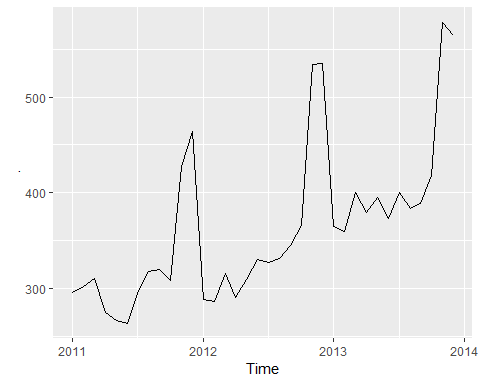
**7**. Are the following statements true or false? Explain your answer. a. Good forecast methods should have normally distributed residuals. b. A model with small residuals will give good forecasts. c. The best measure of forecast accuracy is MAPE. d. If your model doesn’t forecast well, you should make it more complicated. e. Always choose the model with the best forecast accuracy as measured on the test set.

**8**. For your retail time series a. Splot the data into two parts using

myts.train <- window(myts,end=c(2010,12))  
myts.train %>%   
 autoplot()

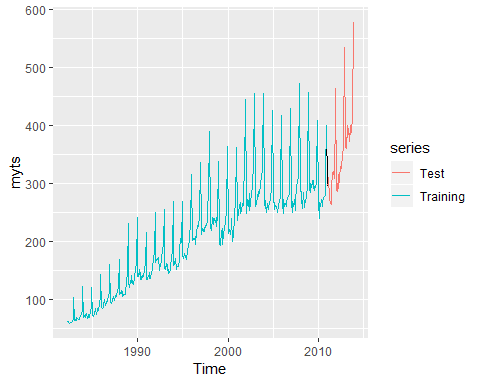


myts.test <- window(myts,start=2011)  
myts.test %>%   
 autoplot()



1. Check that your data have been split appropriately by producing the following plot.

autoplot(myts)+  
 autolayer(myts.train,series="Training")+  
 autolayer(myts.test,series = "Test")



1. Calculate forecast using snaive applied to myts.train

fc <- snaive(myts.train)

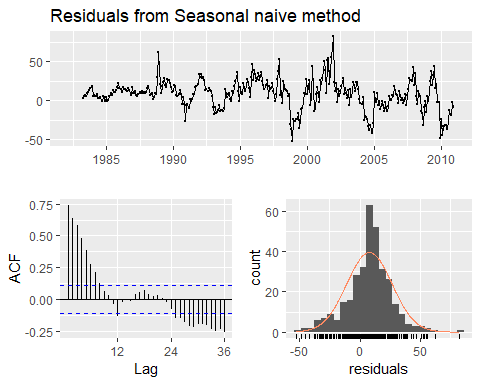
1. Compare the accuracy of your forecasts against the actual values stored in myts.test

accuracy(fc,myts.test)

## ME RMSE MAE MPE MAPE MASE  
## Training set 7.772973 20.24576 15.95676 4.702754 8.109777 1.000000  
## Test set 55.300000 71.44309 55.78333 14.900996 15.082019 3.495907  
## ACF1 Theil's U  
## Training set 0.7385090 NA  
## Test set 0.5315239 1.297866

1. Check the residuals

checkresiduals(fc)



##   
## Ljung-Box test  
##   
## data: Residuals from Seasonal naive method  
## Q\* = 624.45, df = 24, p-value < 2.2e-16  
##   
## Model df: 0. Total lags used: 24

**9**. visnights contains quarterly visitor nights (in millions) from 1998 to 2016 for twenty regions of Ausutralia.

1. Use window() to create three training sets for visnights[,"QLFMetro"],omitting the last 1,2, and 3 years; call these train1, train2 and train3, respectively. For example, train1<-window(visnights[,"QLDMetro],end=c(2015,4).

visnights %>%   
 glimpse() %>%   
 tail()

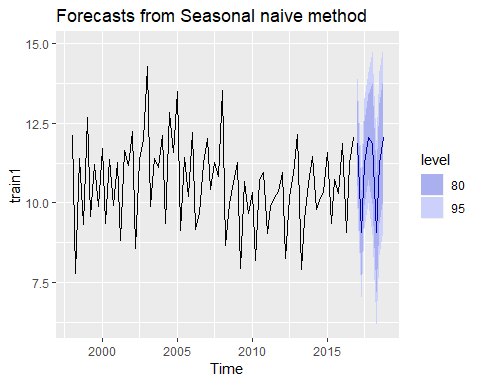
## Time-Series [1:76, 1:20] from 1998 to 2017: 9.05 6.96 6.87 7.15 7.96 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:20] "NSWMetro" "NSWNthCo" "NSWSthCo" "NSWSthIn" ...

## NSWMetro NSWNthCo NSWSthCo NSWSthIn NSWNthIn QLDMetro QLDCntrl  
## 2015 Q3 6.294490 5.486429 2.097544 3.079726 2.877699 10.740806 6.155201  
## 2015 Q4 6.945476 6.818165 2.561440 2.630874 3.776077 10.300925 4.449971  
## 2016 Q1 7.373757 9.945415 4.789194 2.767853 2.871376 11.875199 4.498251  
## 2016 Q2 6.792234 6.664099 2.320229 2.691365 3.928462 9.058698 4.867572  
## 2016 Q3 6.530568 5.978458 1.651091 3.269565 3.514431 11.236115 6.434020  
## 2016 Q4 7.878277 7.362045 2.713210 2.739392 3.490721 12.057719 6.572584  
## QLDNthCo SAUMetro SAUCoast SAUInner VICMetro VICWstCo VICEstCo  
## 2015 Q3 6.530297 1.911144 0.996588 1.0210521 6.545236 0.8112797 1.1358062  
## 2015 Q4 3.637882 2.383107 1.972753 1.2512836 7.724151 1.5260334 1.8311882  
## 2016 Q1 3.127780 2.874414 2.860267 1.1136515 9.972275 2.4951023 3.2038931  
## 2016 Q2 3.658800 2.075967 1.680254 1.4646183 7.184241 0.9304271 1.2493981  
## 2016 Q3 5.359851 2.121646 1.554085 1.3354876 6.909728 0.9411218 0.9678755  
## 2016 Q4 3.302716 2.407233 1.705153 0.9250705 8.012161 1.3520852 1.7666327  
## VICInner WAUMetro WAUCoast WAUInner OTHMetro OTHNoMet  
## 2015 Q3 4.569014 2.922297 6.347613 1.088081 4.419011 1.819066  
## 2015 Q4 4.584000 3.640983 6.174618 1.306545 3.809973 2.379368  
## 2016 Q1 5.762570 4.169990 6.738586 1.033123 4.288449 2.853101  
## 2016 Q2 4.382705 3.262924 5.937479 1.243278 3.278023 1.917090  
## 2016 Q3 4.285911 2.837998 6.065808 1.169328 4.290925 2.131902  
## 2016 Q4 4.955138 3.367659 5.983870 1.345384 4.233838 2.015250

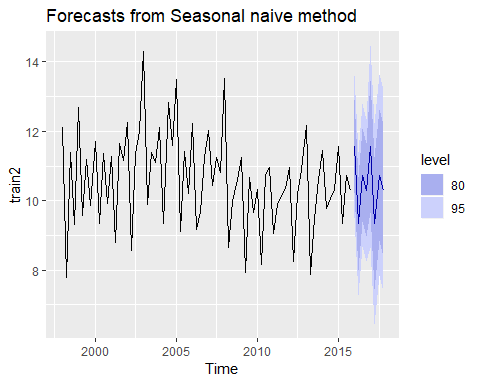
train1 <- window(visnights[,"QLDMetro"],end=c(2016,4))  
train2 <- window(visnights[,"QLDMetro"],end=c(2015,4))  
train3 <- window(visnights[,"QLDMetro"],end=c(2014,4))

1. Compute one year of forecasts for each training set using the snaive() method. Call these fc1, fc2, and fc3, respectively.

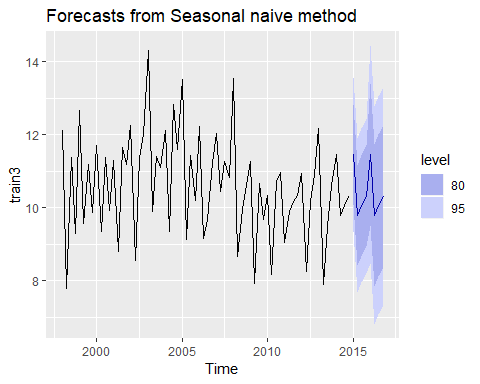
fc1 <- snaive(train1)  
fc2 <- snaive(train2)  
fc3 <- snaive(train3)  
  
autoplot(fc1)



autoplot(fc2)



autoplot(fc3)



1. Use accuracy() to compaqre the MAPE over the three test sets. Comment on these.

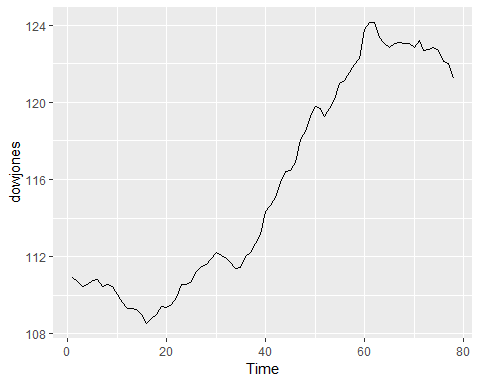
acc1 <- accuracy(fc1) %>% data.frame()  
acc2 <- accuracy(fc2) %>% data.frame()  
acc3 <- accuracy(fc3) %>% data.frame()  
  
bind\_rows(acc1,acc2,acc3) %>%   
 mutate(model=factor(c("train1","train2","train3"))) %>%   
 select(model,MAPE)

## model MAPE  
## 1 train1 7.875818  
## 2 train2 7.976760  
## 3 train3 8.284216

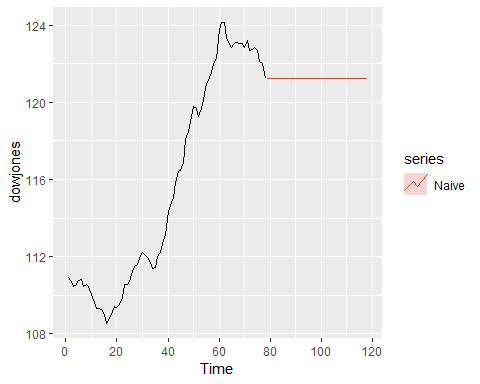
1. Use the Dow Jones index (dataset dowjones) to do the following:
2. Produce a timelot of the series.
3. Produce a forecasts using the drift method and plot them.
4. Show that the forecats are identical to extending the line drawn between the first and last observations.
5. Try using some of the other benchmark functions to forecasts the same dataset. Which do you think is best?

We will produce the forecasts using the drift method by autolayer.

autoplot(dowjones)



autoplot(dowjones) +  
 autolayer(rwf(dowjones, h=40),  
 series="Naïve", PI=FALSE)



Next, we would use other methods, such as mean and naive.

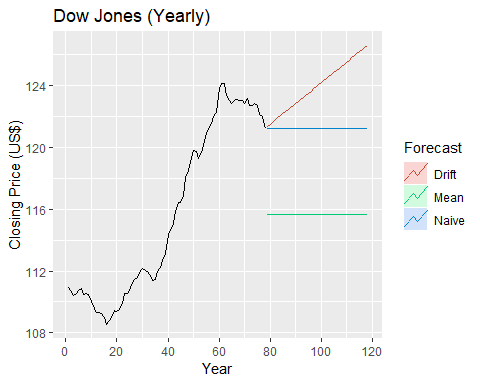
glimpse(dowjones)

## Time-Series [1:78] from 1 to 78: 111 111 110 111 111 ...

frequency(dowjones)

## [1] 1

autoplot(dowjones) +  
 autolayer(meanf(dowjones, h=40),  
 series="Mean", PI=FALSE) +  
 autolayer(rwf(dowjones, h=40),  
 series="Naïve", PI=FALSE) +  
 autolayer(rwf(dowjones, drift=TRUE, h=40),  
 series="Drift", PI=FALSE) +  
 ggtitle("Dow Jones (Yearly)") +  
 xlab("Year") + ylab("Closing Price (US$)") +  
 guides(colour=guide\_legend(title="Forecast"))



## 3.8 Further reading

* Ord et al. (2017) provides further discussion of simple benchmark forecasting methods.
* A review of forecast evaluation methods is given in Hyndman & Koehler (2006), looking at the strengths and weaknesses of different approaches. This is the paper that introduced the MASE as a general-purpose forecast accuracy measure.

Bibliography Ord, J. K., Fildes, R., & Kourentzes, N. (2017). Principles of business forecasting (2nd ed.). Wessex Press Publishing Co. [Amazon]

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22, 679–688. <https://robjhyndman.com/publications/automatic-forecasting/>

# Chapter 4: Judgemental forecasts

orecasting using judgement is common in practice. In many cases, judgmental forecasting is the only option, such as when there is a complete lack of historical data, or when a new product is being launched, or when a new competitor enters the market, or during completely new and unique market conditions. For example, in December 2012, the Australian government was the first in the world to pass legislation that banned the use of company logos on cigarette packets, and required all cigarette packets to be a dark green colour. Judgement must be applied in order to forecast the effect of such a policy, as there are no historical precedents.

There are also situations where the data are incomplete, or only become available after some delay. For example, central banks include judgement when forecasting the current level of economic activity, a procedure known as nowcasting, as GDP is only available on a quarterly basis.

Research in this area3 has shown that the accuracy of judgmental forecasting improves when the forecaster has 1). important domain knowledge, and 2). more timely, up-to-date information. A judgmental approach can be quick to adjust to such changes, information or events.

Over the years, the acceptance of judgmental forecasting as a science has increased, as has the recognition of its need. More importantly, the quality of judgmental forecasts has also improved, as a direct result of recognising that improvements in judgmental forecasting can be achieved by implementing well-structured and systematic approaches. It is important to recognise that judgmental forecasting is subjective and comes with limitations. However, implementing systematic and well-structured approaches can confine these limitations and markedly improve forecast accuracy.

There are three general settings in which judgmental forecasting is used:

1. there are no available data, so that statistical methods are not applicable and judgmental forecasting is the only feasible approach;
2. data are available, statistical forecasts are generated, and these are then adjusted using judgement; and
3. data are available and statistical and judgmental forecasts are generated independently and then combined. We should clarify that when data are available, applying statistical methods (such as those discussed in other chapters of this book), is preferable and should always be used as a starting point. Statistical forecasts are generally superior to generating forecasts using only judgement. For the majority of the chapter, we focus on the first setting where no data are available, and in the last section we discuss the judgmental adjustment of statistical forecasts. We discuss combining forecasts in Section 12.4.

## 4.1 Beware of limitations

Judgmental forecasts are subjective, and therefore do not come free of bias or limitations.

Judgmental forecasts can be inconsistent. Unlike statistical forecasts, which can be generated by the same mathematical formulas every time, judgmental forecasts depend heavily on human cognition, and are vulnerable to its limitations. For example, a limited memory may render recent events more important than they actually are and may ignore momentous events from the more distant past; or a limited attention span may result in important information being missed; or a misunderstanding of causal relationships may lead to erroneous inferences. Furthermore, human judgement can vary due to the effect of psychological factors. One can imagine a manager who is in a positive frame of mind one day, generating forecasts that may tend to be somewhat optimistic, and in a negative frame of mind another day, generating somewhat less optimistic forecasts.

Judgement can be clouded by personal or political agendas, where targets and forecasts (as defined in Chapter 1) are not segregated. For example, if a sales manager knows that the forecasts she generates will be used to set sales expectations (targets), she may tend to set these low in order to show a good performance (i.e., exceed the expected targets). Even in cases where targets and forecasts are well segregated, judgement may be plagued by optimism or wishful thinking. For example, it would be highly unlikely that a team working towards launching a new product would forecast its failure. As we will discuss later, this optimism can be accentuated in a group meeting setting. “Beware of the enthusiasm of your marketing and sales colleagues”.

Another undesirable property which is commonly seen in judgmental forecasting is the effect of anchoring. In this case, the subsequent forecasts tend to converge or be close to an initial familiar reference point. For example, it is common to take the last observed value as a reference point. The forecaster is influenced unduly by prior information, and therefore gives this more weight in the forecasting process. Anchoring may lead to conservatism and undervaluing new and more current information, and thereby create a systematic bias.

## 4.2 Key principles

Using a systematic and well structured approach in judgmental forecasting helps to reduce the adverse effects of the limitations of judgmental forecasting, some of which we listed in the previous section. Whether this approach involves one individual or many, the following principles should be followed.

### Set the forecasting task clearly and concisely

Care is needed when setting the forecasting challenges and expressing the forecasting tasks. It is important that everyone be clear about what the task is. All definitions should be clear and comprehensive, avoiding ambiguous and vague expressions. Also, it is important to avoid incorporating emotive terms and irrelevant information that may distract the forecaster. In the Delphi method that follows (see Section 4.3), it may sometimes be useful to conduct a preliminary round of information gathering before setting the forecasting task.

### Implement a systematic approach

Forecast accuracy and consistency can be improved by using a systematic approach to judgmental forecasting involving checklists of categories of information which are relevant to the forecasting task. For example, it is helpful to identify what information is important and how this information is to be weighted. When forecasting the demand for a new product, what factors should we account for and how should we account for them? Should it be the price, the quality and/or quantity of the competition, the economic environment at the time, the target population of the product? It is worthwhile to devote significant effort and resources to put together decision rules that will lead to the best possible systematic approach.

## Document and justify

Formalising and documenting the decision rules and assumptions implemented in the systematic approach can promote consistency, as the same rules can be implemented repeatedly. Also, requesting a forecaster to document and justify their forecasts leads to accountability, which can lead to reduced bias. Furthermore, formal documentation aids significantly in the systematic evaluation process that is suggested next.

## Systematically evaluate

Systematically monitoring the forecasting process can identify unforeseen irregularities. In particular, keep records of forecasts and use them to obtain feedback when the corresponding observations become available. Although you may do your best as a forecaster, the environment you operate in is dynamic. Changes occur, and you need to monitor these in order to evaluate the decision rules and assumptions. Feedback and evaluation help forecasters learn and improve their forecast accuracy.

## Segregate forecasters and users

Forecast accuracy may be impeded if the forecasting task is carried out by users of the forecasts, such as those responsible for implementing plans of action about which the forecast is concerned. We should clarify again here (as in Section 1.2), that forecasting is about predicting the future as accurately as possible, given all of the information available, including historical data and knowledge of any future events that may impact the forecasts. Forecasters and users should be clearly segregated. A classic case is that of a new product being launched. The forecast should be a reasonable estimate of the sales volume of a new product, which may differ considerably from what management expects or hopes the sales will be in order to meet company financial objectives. In this case, a forecaster may be delivering a reality check to the user.

It is important that forecasters communicate forecasts to potential users thoroughly. As we will see in Section 4.7, users may feel distant and disconnected from forecasts, and may not have full confidence in them. Explaining and clarifying the process and justifying the basic assumptions that led to the forecasts will provide some assurance to users.

The way in which forecasts may then be used and implemented will clearly depend on managerial decision making. For example, management may decide to adjust a forecast upwards (be over-optimistic), as the forecast may be used to guide purchasing and stock keeping levels. Such a decision may be taken after a cost-benefit analysis reveals that the cost of holding excess stock is much lower than that of lost sales. This type of adjustment should be part of setting goals or planning supply, rather than part of the forecasting process. In contrast, if forecasts are used as targets, they may be set low so that they can be exceeded more easily. Again, setting targets is different from producing forecasts, and the two should not be confused.

The example that follows comes from our experience in industry. It exemplifies two contrasting styles of judgmental forecasting — one that adheres to the principles we have just presented and one that does not.

## Example: Pharmaceutical benefits scheme (PBS)

The Australian government subsidises the cost of a wide range of prescription medicines as part of the PBS. Each subsidised medicine falls into one of four categories: concession copayments, concession safety net, general copayments, and general safety net. Each person with a concession card makes a concession copayment per PBS medicine (), until they reach a set threshold amount labelled the concession safety net (). For the rest of the financial year, all PBS-listed medicines are free. Each general patient makes a general copayment per PBS medicine () until the general safety net amount is reached (). For the rest of the financial year, they contribute a small amount per PBS-listed medicine (). The PBS forecasting process uses 84 groups of PBS-listed medicines, and produces forecasts of the medicine volume and the total expenditure for each group and for each of the four PBS categories, a total of 672 series. This forecasting process aids in setting the government budget allocated to the PBS, which is over $7 billion per year, or approximately 1% of GDP.

Figure 4.1 summarises the forecasting process. Judgmental forecasts are generated for new listings of medicines and for estimating the impact of new policies. These are shown by the green items. The pink items indicate the data used which were obtained from various government departments and associated authorities. The blue items show things that are calculated from the data provided. There were judgmental adjustments to the data to take account of new listings and new policies, and there were also judgmental adjustments to the forecasts. Because of the changing size of both the concession population and the total population, forecasts are produced on a per-capita basis, and then multiplied by the forecast population to obtain forecasts of total volume and expenditure per month.

One of us (Hyndman) was asked to evaluate the **forecasting process** a few years ago. We found that using judgement for new listings and new policy impacts gave better forecasts than using a statistical model alone. However, we also found that the forecasting accuracy and consistency could be improved through a more structured and systematic process, especially for policy impacts.

*Forecasting new listings*: Companies who apply for their medicine to be added to the PBS are asked to submit detailed forecasts for various aspects of the medicine, such as projected patient numbers, market share of the new medicine, substitution effects, etc. The Pharmaceutical Benefits Advisory Committee provides guidelines describing a highly structured and systematic approach for generating these forecasts, and requires careful documentation for each step of the process. This structured process helps to reduce the likelihood and effects of deliberate self-serving biases. Two detailed evaluation rounds of the company forecasts are implemented by a sub-committee, one before the medicine is added to the PBS and one after it is added. Finally, comparisons of observations versus forecasts for some selected new listings are performed, 12 months and 24 months after the listings, and the results are sent back to the companies for comment.

*Policy impact forecasts*: In contrast to the highly structured process used for new listings, there were no systematic procedures for policy impact forecasts. On many occasions, forecasts of policy impacts were calculated by a small team, and were often heavily reliant on the work of one person. The forecasts were not usually subject to a formal review process. There were no guidelines for how to construct judgmental forecasts for policy impacts, and there was often a lack of adequate documentation about how these forecasts were obtained, the assumptions underlying them, etc.

Consequently, we recommended several changes: - that guidelines for forecasting new policy impacts be developed, to encourage a more systematic and structured forecasting approach; that the forecast methodology be documented in each case, including all assumptions made in forming the forecasts; - that new policy forecasts be made by at least two people from different areas of the organisation; - that a review of forecasts be conducted one year after the implementation of each new policy by a review committee, especially for new policies that have a significant annual projected cost or saving. The review committee should include those involved in generating the forecasts, but also others.

## 4.3 The Delphi method

The Delphi method was invented by Olaf Helmer and Norman Dalkey of the Rand Corporation in the 1950s for the purpose of addressing a specific military problem. The method relies on the key assumption that forecasts from a group are generally more accurate than those from individuals. The aim of the Delphi method is to construct consensus forecasts from a group of experts in a structured iterative manner. A facilitator is appointed in order to implement and manage the process. The Delphi method generally involves the following stages:

1. A panel of experts is assembled
2. Forecasting tasks/challenges are set and distributed to the experts
3. Experts return initial forecasts and justifications. Theese are compiled and summarised in order to provide feedback
4. Feedback is provided to the experts, who now review their forecasts in light of the feedback. This step may be iterated until a satisfactory level of consensus is reached.
5. Final forecasts are constructed by aggregating the experts’ forecasts.

Each stage of the Delphi method comes with its own challenges. In what folows, we provide some suggestions and discussions about each of these.

### Experts and anonimity

The first challenge of the facilitator is to identify a group of experts who can contribute to the forecasting task. The usual suggestion is somewhere between 5 and 20 experts with diverse expertise. Experts submit forecasts and also provide detailed qualitative justifications for these.

A key feature of the **Delphi method** is that the participating experts remain anonymous at all times. This means that the experts cannot be influenced by political and social pressures in their forecasts. Furthermore, all experts are given an equal say and all are held accountable for their forecasts. This avoids the situation where a group meeting is held and some members do not contribute, while others dominate. It also prevents members exerting undue influence based on seniority or personality. There have been suggestions that even something as simple as the seating arrangements in a group setting can influence the group dynamics. Furthermore, there is ample evidence that a group meeting setting promotes enthusiasm and influences individual judgement, leading to optimism and overconfidence.7

A by-product of anonymity is that the experts do not need to meet as a group in a physical location. An important advantage of this is that it increases the likelihood of gathering experts with diverse skills and expertise from varying locations. Furthermore, it makes the process cost-effective by eliminating the expense and inconvenience of travel, and it makes it flexible, as the experts only have to meet a common deadline for submitting forecasts, rather than having to set a common meeting time.

### Setting the forecasting task in a Delphi

In a Delphi setting, it may be useful to conduct a preliminary round of information gathering from the experts before setting the forecasting tasks. Alternatively, as experts submit their initial forecasts and justifications, valuable information which is not shared between all experts can be identified by the facilitator when compiling the feedback.

### Feedback

Feedback to the experts should include summary statistics of the forecasts and outlines of qualitative justifications. Numerical data summaries and graphical representations can be used to summarise the experts’ forecasts.

As the feedback is controlled by the facilitator, there may be scope to direct attention and information from the experts to areas where it is most required. For example, the facilitator may direct the experts’ attention to responses that fall outside the interquartile range, and the qualitative justification for such forecasts.

### Iteration

The process of the experts submitting forecasts, receiving feedback, and reviewing their forecasts in light of the feedback, is repeated until a satisfactory level of consensus between the experts is reached. Satisfactory consensus does not mean complete convergence in the forecast value; it simply means that the variability of the responses has decreased to a satisfactory level. Usually two or three rounds are sufficient. Experts are more likely to drop out as the number of iterations increases, so too many rounds should be avoided.

### Final forecasts

The final forecasts are usually constructed by giving equal weight to all of the experts’ forecasts. However, the facilitator should keep in mind the possibility of extreme values which can distort the final forecast.

### Limitations and variations

Applying the Delphi method can be time consuming. In a group meeting, final forecasts can possibly be reached in hours or even minutes — something which is almost impossible to do in a Delphi setting. If it is taking a long time to reach a consensus in a Delphi setting, the panel may lose interest and cohesiveness.

In a group setting, personal interactions can lead to quicker and better clarifications of qualitative justifications. A variation of the Delphi method which is often applied is the “estimate-talk-estimate” method, where the experts can interact between iterations, although the forecast submissions can still remain anonymous. A disadvantage of this variation is the possibility of the loudest person exerting undue influence.

### The facilitator

The role of the facilitator is of the utmost importance. The facilitator is largely responsible for the design and administration of the Delphi process. The facilitator is also responsible for providing feedback to the experts and generating the final forecasts. In this role, the facilitator needs to be experienced enough to recognise areas that may need more attention, and to direct the experts’ attention to these. Also, as there is no face-to-face interaction between the experts, the facilitator is responsible for disseminating important information. The efficiency and effectiveness of the facilitator can dramatically increase the probability of a successful Delphi method in a judgmental forecasting setting.

## 4.4 Forecasting by analogy

A useful judgmental approach which is often implemented in practice is forecasting by analogy. A common example is the pricing of a house through an appraisal process. An appraiser estimates the market value of a house by comparing it to similar properties that have sold in the area. The degree of similarity depends on the attributes considered. With house appraisals, attributes such as land size, dwelling size, numbers of bedrooms and bathrooms, and garage space are usually considered.

Even thinking and discussing analogous products or situations can generate useful (and sometimes crucial) information. We illustrate this point with the following example.

### Example: Designing a high school curriculum

A small group of academics and teachers were assigned the task of developing a curriculum for teaching judgement and decision making under uncertainty for high schools in Israel. Each group member was asked to forecast how long it would take for the curriculum to be completed. Responses ranged between 18 and 30 months. One of the group members who was an expert in curriculum design was asked to consider analogous curricula developments around the world. He concluded that 40% of analogous groups he considered never completed the task. The rest took between 7 to 10 years. The Israel project was completed in 8 years.

Obviously, forecasting by analogy comes with challenges. We should aspire to base forecasts on multiple analogies rather than a single analogy, which may create biases. However, these may be challenging to identify. Similarly, we should aspire to consider multiple attributes. Identifying or even comparing these may not always be straightforward. As always, we suggest performing these comparisons and the forecasting process using a systematic approach. Developing a detailed scoring mechanism to rank attributes and record the process of ranking will always be useful.

### A structured analogy

Alternatively, a structured approach comprising a panel of experts can be implemented, as was proposed by Green & Armstrong (2007). The concept is similar to that of a Delphi; however, the forecasting task is completed by considering analogies. First, a facilitator is appointed. Then the structured approach involves the following steps.

1. A panel of experts who are likely to have experience with analogous situations is assembled.
2. Tasks/challenges are set and distributed to the experts.
3. Experts identify and describe as many analogies as they can, and generate forecasts based on each analogy.
4. Experts list similarities and differences of each analogy to the target situation, then rate the similarity of each analogy to the target situation on a scale.
5. Forecasts are derived by the facilitator using a set rule. This can be a weighted average, where the weights can be guided by the ranking scores of each analogy by the experts.

As with the Delphi approach, anonymity of the experts may be an advantage in not suppressing creativity, but could hinder collaboration. Green and Armstrong found no gain in collaboration between the experts in their results. A key finding was that experts with multiple analogies (more than two), and who had direct experience with the analogies, generated the most accurate forecasts.

### Bibligraphy

Green, K. C., & Armstrong, J. S. (2007). Structured analogies for forecasting. International Journal of Forecasting, 23(3), 365–376. <https://doi.org/10.1016/j.ijforecast.2007.05.005>

## 4.5 Scenario forecasting

A fundamentally different approach to judgmental forecasting is **scenario-based forecasting**. The aim of this approach is to generate forecasts based on plausible scenarios. In contrast to the two previous approaches (Delphi and forecasting by analogy) where the resulting forecast is intended to be a likely outcome, each scenario-based forecast may have a low probability of occurrence. The scenarios are generated by considering all possible factors or drivers, their relative impacts, the interactions between them, and the targets to be forecast.

Building forecasts based on scenarios allows a wide range of possible forecasts to be generated and some extremes to be identified. For example it is usual for “best”, “middle” and “worst” case scenarios to be presented, although many other scenarios will be generated. Thinking about and documenting these contrasting extremes can lead to early contingency planning.

With scenario forecasting, decision makers often participate in the generation of scenarios. While this may lead to some biases, it can ease the communication of the scenario-based forecasts, and lead to a better understanding of the results.

## 4.6 New product forecasting

The definition of a new product can vary. It may be an entirely new product which has been launched, a variation of an existing product (“new and improved”), a change in the pricing scheme of an existing product, or even an existing product entering a new market.

Judgmental forecasting is usually the only available method for new product forecasting, as historical data are unavailable. The approaches we have already outlined (Delphi, forecasting by analogy and scenario forecasting) are all applicable when forecasting the demand for a new product.

Other methods which are more specific to the situation are also available. We briefly describe three such methods which are commonly applied in practice. These methods are less structured than those already discussed, and are likely to lead to more biased forecasts as a result.

### Sales force composite

In this approach, forecasts for each outlet/branch/store of a company are generated by salespeople, and are then aggregated. This usually involves sales managers forecasting the demand for the outlet they manage. Salespeople are usually closest to the interaction between customers and products, and often develop an intuition about customer purchasing intentions. They bring this valuable experience and expertise to the forecast.

However, having salespeople generate forecasts violates the key principle of segregating forecasters and users, which can create biases in many directions. It is common for the performance of a salesperson to be evaluated against the sales forecasts or expectations set beforehand. In this case, the salesperson acting as a forecaster may introduce some self-serving bias by generating low forecasts. On the other hand, one can imagine an enthusiastic salesperson, full of optimism, generating high forecasts.

Moreover a successful salesperson is not necessarily a successful nor well-informed forecaster. A large proportion of salespeople will have no or limited formal training in forecasting. Finally, salespeople will feel customer displeasure at first hand if, for example, the product runs out or is not introduced in their store. Such interactions will cloud their judgement.

### Executive opinion

In contrast to the sales force composite, this approach involves staff at the top of the managerial structure generating aggregate forecasts. Such forecasts are usually generated in a group meeting, where executives contribute information from their own area of the company. Having executives from different functional areas of the company promotes great skill and knowledge diversity in the group.

This process carries all of the advantages and disadvantages of a group meeting setting which we discussed earlier. In this setting, it is important to justify and document the forecasting process. That is, executives need to be held accountable in order to reduce the biases generated by the group meeting setting. There may also be scope to apply variations to a Delphi approach in this setting; for example, the estimate-talk-estimate process described earlier.

### Cusomer intentions

Customer intentions can be used to forecast the demand for a new product or for a variation on an existing product. Questionnaires are filled in by customers on their intentions to buy the product. A structured questionnaire is used, asking customers to rate the likelihood of them purchasing the product on a scale; for example, highly likely, likely, possible, unlikely, highly unlikely.

Survey design challenges, such as collecting a representative sample, applying a time- and cost-effective method, and dealing with non-responses, need to be addressed.9

Furthermore, in this survey setting we must keep in mind the relationship between purchase intention and purchase behaviour. Customers do not always do what they say they will. Many studies have found a positive correlation between purchase intentions and purchase behaviour; however, the strength of these correlations varies substantially. The factors driving this variation include the timings of data collection and product launch, the definition of “new” for the product, and the type of industry. Behavioural theory tells us that intentions predict behaviour if the intentions are measured just before the behaviour.10 The time between intention and behaviour will vary depending on whether it is a completely new product or a variation on an existing product. Also, the correlation between intention and behaviour is found to be stronger for variations on existing and familiar products than for completely new products.

Whichever method of new product forecasting is used, it is important to thoroughly document the forecasts made, and the reasoning behind them, in order to be able to evaluate them when data become available.

## 4.7 Judgemental adjustments

n this final section, we consider the situation where historical data are available and are used to generate statistical forecasts. It is common for practitioners to then apply judgmental adjustments to these forecasts. These adjustments can potentially provide all of the advantages of judgmental forecasting which have been discussed earlier in this chapter. For example, they provide an avenue for incorporating factors that may not be accounted for in the statistical model, such as promotions, large sporting events, holidays, or recent events that are not yet reflected in the data. However, these advantages come to fruition only when the right conditions are present. Judgmental adjustments, like judgmental forecasts, come with biases and limitations, and we must implement methodical strategies in order to minimise them.

### Use adjustments sparingly

Practitioners adjust much more often than they should, and many times for the wrong reasons. By adjusting statistical forecasts, users of forecasts create a feeling of ownership and credibility. Users often do not understand or appreciate the mechanisms that generate the statistical forecasts (as they will usually have no training in this area). By implementing judgmental adjustments, users feel that they have contributed to and completed the forecasts, and they can now relate their own intuition and interpretations to these. The forecasts have become their own.

Judgmental adjustments should not aim to correct for a systematic pattern in the data that is thought to have been missed by the statistical model. This has been proven to be ineffective, as forecasters tend to read non-existent patterns in noisy series. Statistical models are much better at taking account of data patterns, and judgmental adjustments only hinder accuracy.

Judgmental adjustments are most effective when there is significant additional information at hand or strong evidence of the need for an adjustment. We should only adjust when we have important extra information which is not incorporated in the statistical model. Hence, adjustments seem to be most accurate when they are large in size. Small adjustments (especially in the positive direction promoting the illusion of optimism) have been found to hinder accuracy, and should be avoided.

### Apply a structured approach

Using a structured and systematic approach will improve the accuracy of judgmental adjustments. Following the key principles outlined in Section 4.2 is vital. In particular, having to document and justify adjustments will make it more challenging to override the statistical forecasts, and will guard against unnecessary adjustments.

It is common for adjustments to be implemented by a panel (see the example that follows). Using a Delphi setting carries great advantages. However, if adjustments are implemented in a group meeting, it is wise to consider the forecasts of key markets or products first, as panel members will get tired during this process. Fewer adjustments tend to be made as the meeting goes on through the day.

### Example: Tourism Forecasting Committee (TFC)

Tourism Australia publishes forecasts for all aspects of Australian tourism twice a year. The published forecasts are generated by the TFC, an independent body which comprises experts from various government and industry sectors; for example, the Australian Commonwealth Treasury, airline companies, consulting firms, banking sector companies, and tourism bodies.

The forecasting methodology applied is an iterative process. First, model-based statistical forecasts are generated by the forecasting unit within Tourism Australia, then judgmental adjustments are made to these in two rounds. In the first round, the TFC Technical Committee11 (comprising senior researchers, economists and independent advisers) adjusts the model-based forecasts. In the second and final round, the TFC (comprising industry and government experts) makes final adjustments. In both rounds, adjustments are made by consensus.

In 2008, we12 analysed forecasts for Australian domestic tourism. We concluded that the published TFC forecasts were optimistic, especially for the long-run, and we proposed alternative model-based forecasts. We now have access to observed data up to and including 2011. In Figure 4.2, we plot the published forecasts against the actual data. We can see that the published TFC forecasts have continued to be optimistic.

What can we learn from this example? Although the TFC clearly states in its methodology that it produces ‘forecasts’ rather than ‘targets’, could this be a case where these have been confused? Are the forecasters and users sufficiently well-segregated in this process? Could the iterative process itself be improved? Could the adjustment process in the meetings be improved? Could it be that the group meetings have promoted optimism? Could it be that domestic tourism should have been considered earlier in the day?

## 4.8 Further reading

Many forecasting textbooks ignore judgmental forecasting altogether. Here are three which do cover it in some detail.

* Chapter 11 of Ord et al. (2017) provides an excellent review of some of the same topics as this chapter, but also includes using judgement to assessing forecast uncertainty, and forecasting using prediction markets.
* Goodwin & Wright (2009) is a book-length treatment of the use of judgement in decision marking by two of the leading researchers in the field.
* Kahn (2006) covers techniques for new product forecasting, where judgmental methods play an important role.

There have been some helpful survey papers on judgmental forecasting published in the last 20 years. We have found these three particularly helpful.

* Fildes & Goodwin (2007b)
* Fildes & Goodwin (2007a)
* Harvey (2001)

Some helpful papers on individual judgmental forecasting methods are listed in the table below.

### Books

* @Ord2012
* @GW04
* @Kahn2006

### General papers

* @Fildes2007a
* @Fildes2007
* @Harvey2001

### Delphi

* @RW99
* @Rowe2007

### Adjustments

* @Sanders2005
* @Eroglu2010
* @Franses2013
* @Goodwin00

### Analogy

* @Green2007

### Scenarios

* @Onkal2012

### Customer intentions

* @Morwitz2007