Time series - graphics

Forecasting: principles and practice

book by Rob Hyndman and George Athanasopoulos
slides by Peter Fuleky and modified by Joseph ALBA

The basic steps in a forecasting task

1. Problem definition.
2. Gathering information:
    - *statistical data*
    - *expertise*
3. Preliminary (exploratory) analysis. **Graphs!**
4. Choosing and fitting models. 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.
5. Using and evaluating a forecasting model.

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 visualized 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 far 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.

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. As discussed in lecture 1, this information can be stored as a *ts* object in R.

We will use *ts* object in R and we will learn more of it in our discussions. For more information, please refer to *forecast package* reference manual available at NTULearn

To use the *forecast package* and the datasets in the *fpp 2nd edition* and the *fpp 1st edition*:

```
library(fpp2)
library(fpp)
```

The datasets in *fpp2* are described in the pdf file *fpp2* also available at NTULearn.

Note that *fpp* is the earlier version of *fpp2*. Some datasets are in *fpp* but not in *fpp2* and vice versa.

fpp.pdf, available at NTULearn, describes the datasets used in *fpp 1st edition*

Time series patterns

We will refer to three types of time series patterns.

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 changing direction when it might go from an increasing trend to a decreasing trend.

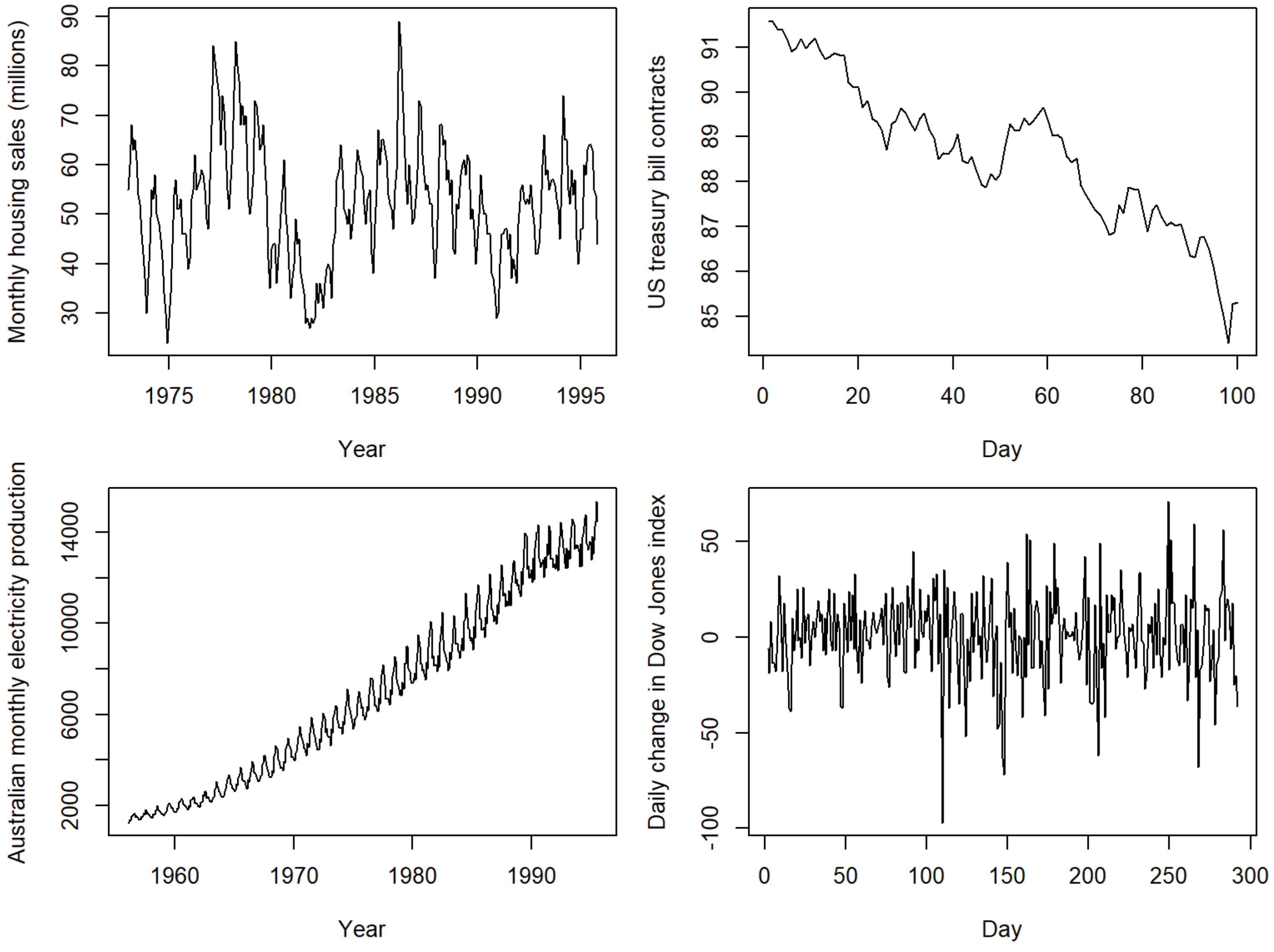
Seasonal

- A seasonal pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, or day of the week). Seasonality is always of a fixed and known period.

Cycle

- A cyclic pattern exists when data exhibit rises and falls that are not of fixed period. The duration of these fluctuations is usually of at least 2 years.

If the fluctuations are not of fixed period then they are cyclic; if the period is unchanging and associated with some aspect of the calendar, then the pattern is seasonal.


Monthly housing sales (millions)
90
80
70
60
50
40
30
1975
1980
1985
1990
1995
Year
US treasury bill contracts
91
90
89
88
87
86
85
0
20
40
60
80
100
Day
Australian monthly electricity production
14000
10000
6000
2000
1960
1970
1980
1990
Year
Daily change in Dow Jones index
50
0
-50
-100
0
50
100
150
200
250
300
Day

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.

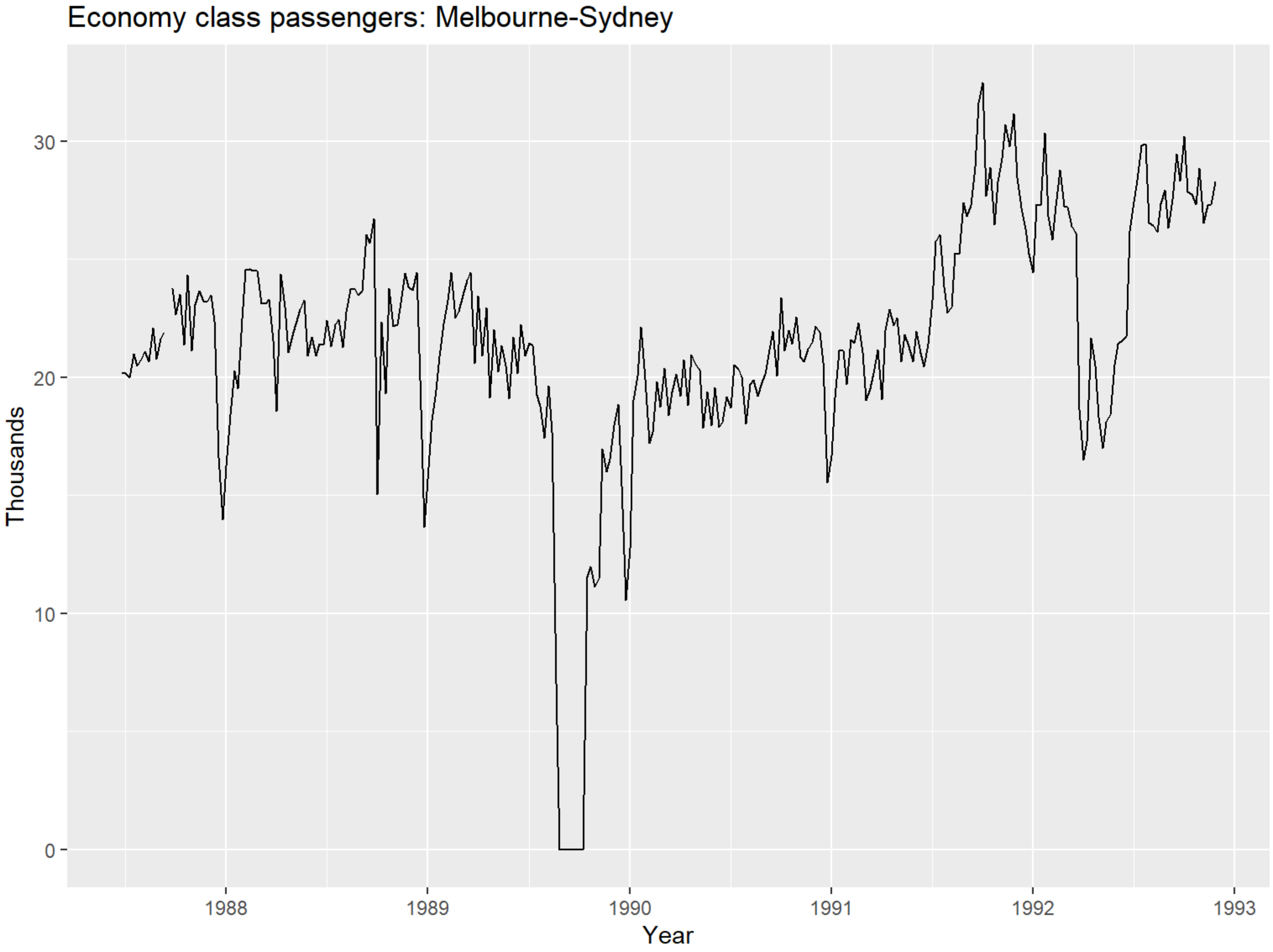
3. The Australian monthly electricity production (bottom left) shows a strong increasing trend, with strong seasonality. There is no evidence of any cyclic behaviour here.

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.

These time series patterns will be discussed in more detail in Chapter 6 entitled *time series decompositions*

Time series plots

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

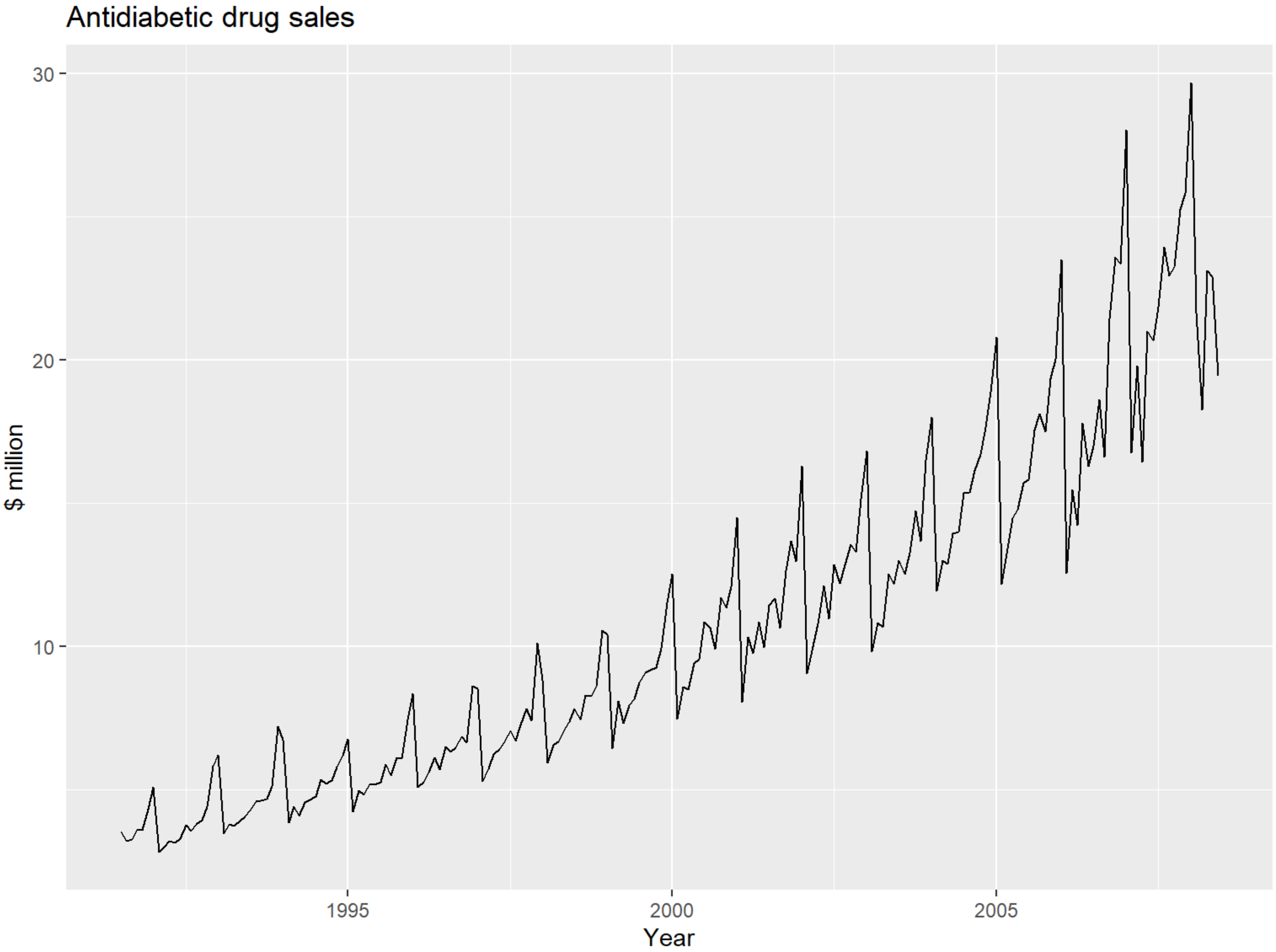

Economy class passengers: Melbourne-Sydney
Thousands
30
20
10
0
1988
1989
1990
1991
1992
1993
Year

The time plot immediately reveals some interesting features.

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

Another time series plot

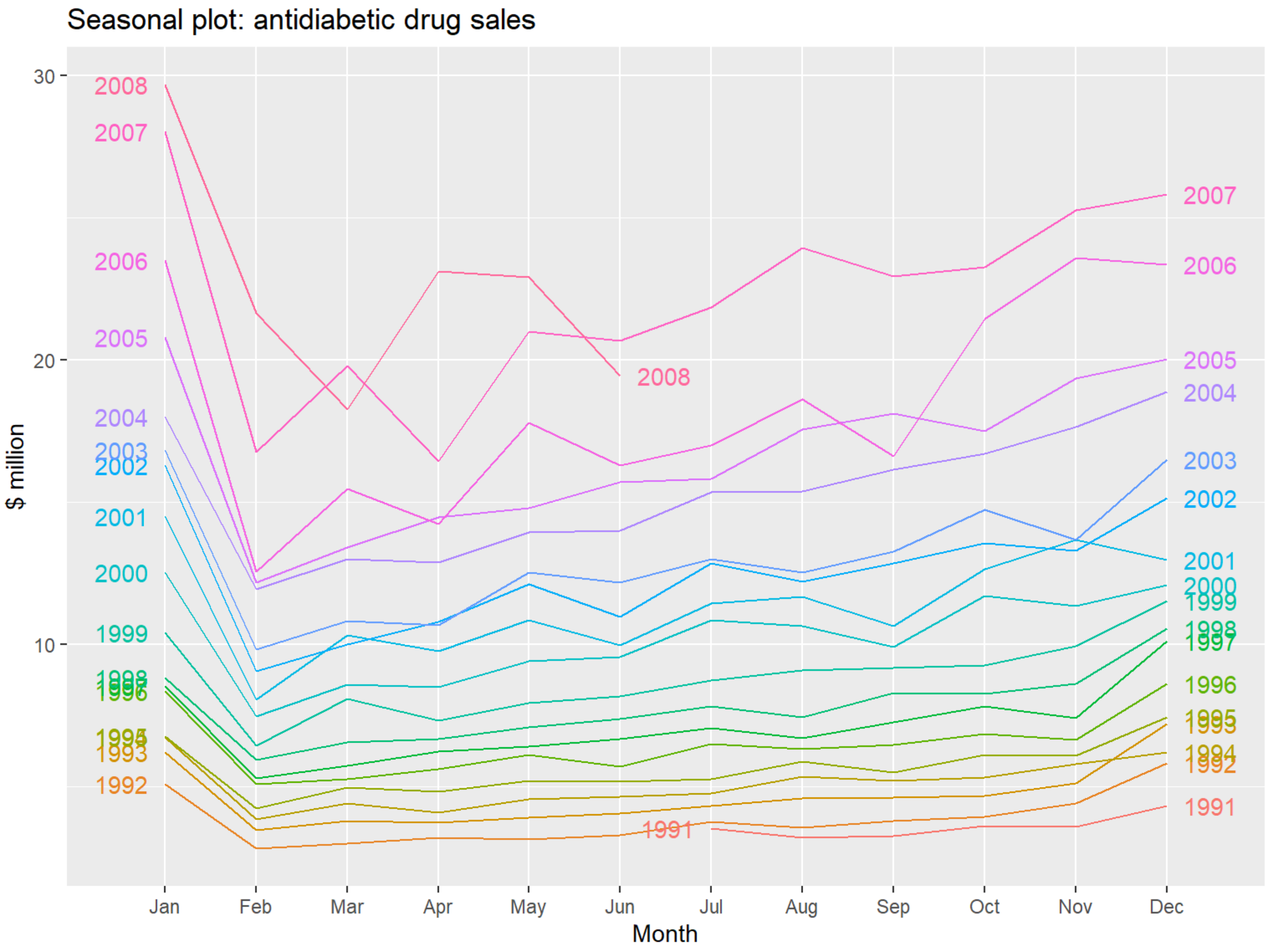
```
autoplot(a10) +
ggtitle("Antidiabetic drug sales") +
ylab("$ million") + xlab("Year")
```


Antidiabetic drug sales
$ million
30
20
10
1995
2000
2005
Year

- 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 end 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.)

Seasonal plots

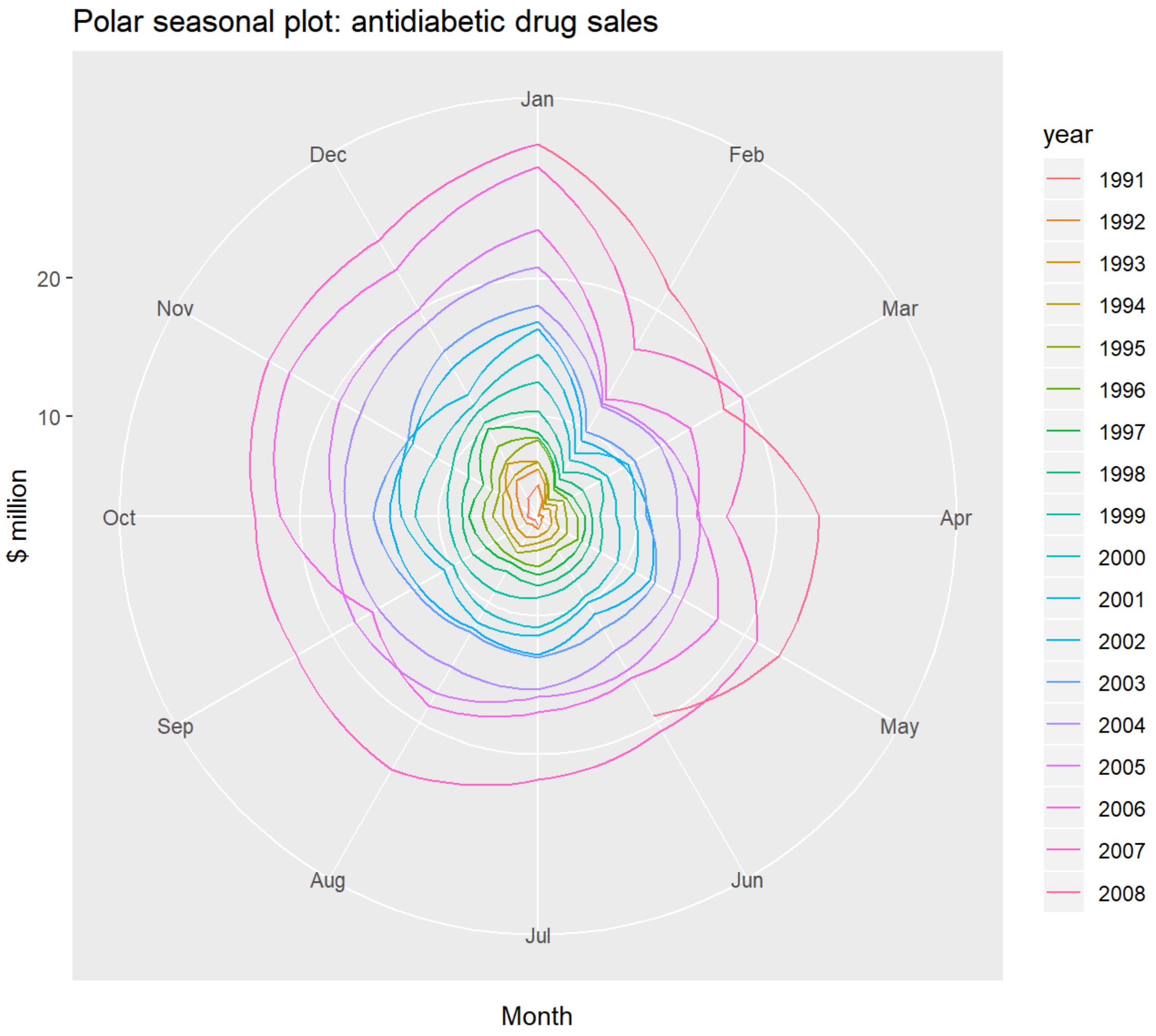
```r
ggseasonplot(a10, year.labels=TRUE, year.labels.left=TRUE) +
ylab("$ million") + ggtitle("Seasonal plot: antidiabetic drug sales")
```


Seasonal plot: antidiabetic drug sales
$ million
30
20
10
Month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
2008
2007
2006
2005
2004
2003
2002
2001
2000
1999
1998
1997
1996
1995
1994
1993
1992
1991

- 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 low 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.

Seasonal plots

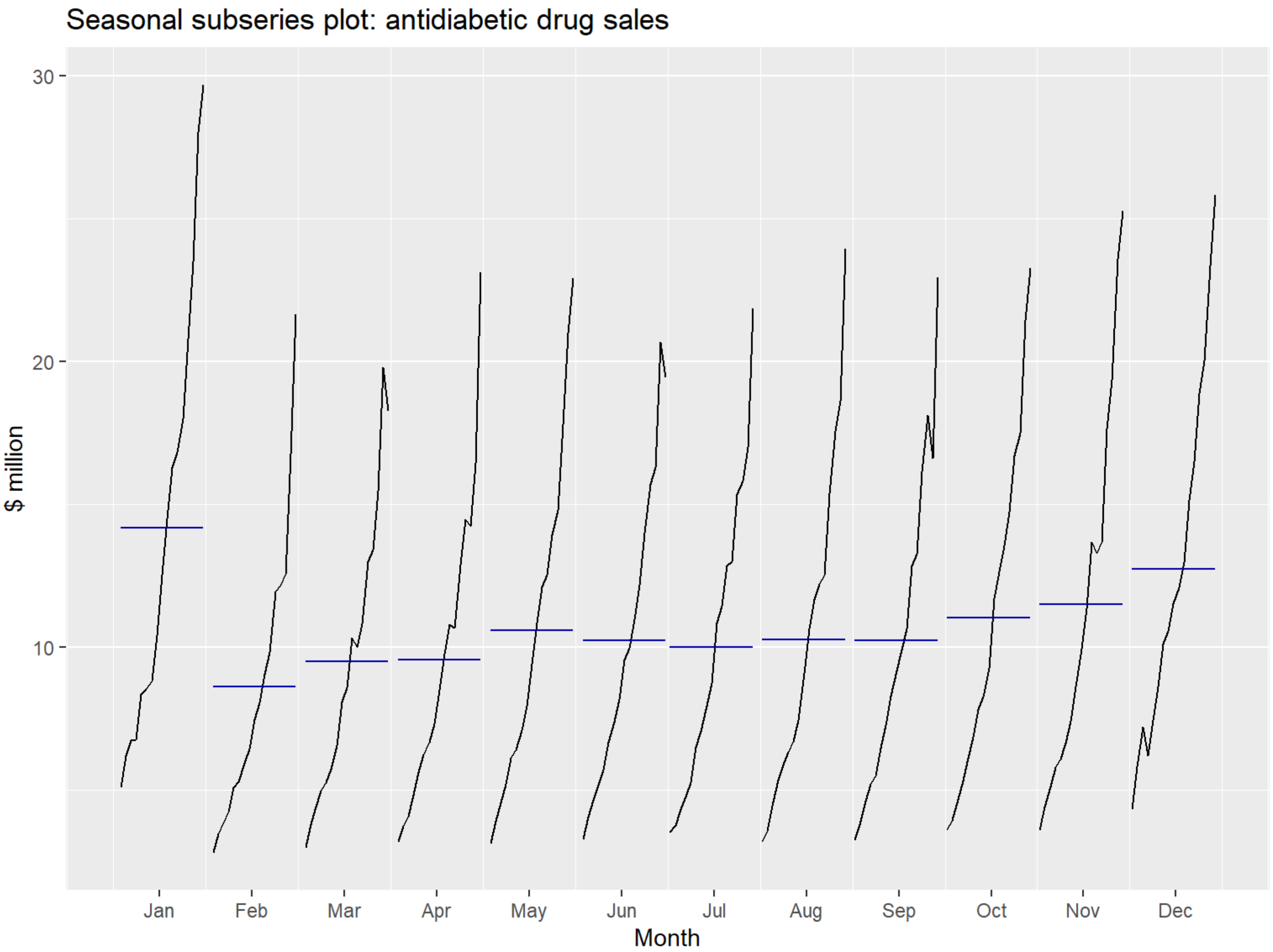
```
ggseasonplot(a10, polar=TRUE) +
ylab("$ million") + ggtitle("Polar seasonal plot: antidiabetic drug sales")
```


Polar seasonal plot: antidiabetic drug sales
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
20
10
$ million
Month
year
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008

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

Seasonal subseries plots

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

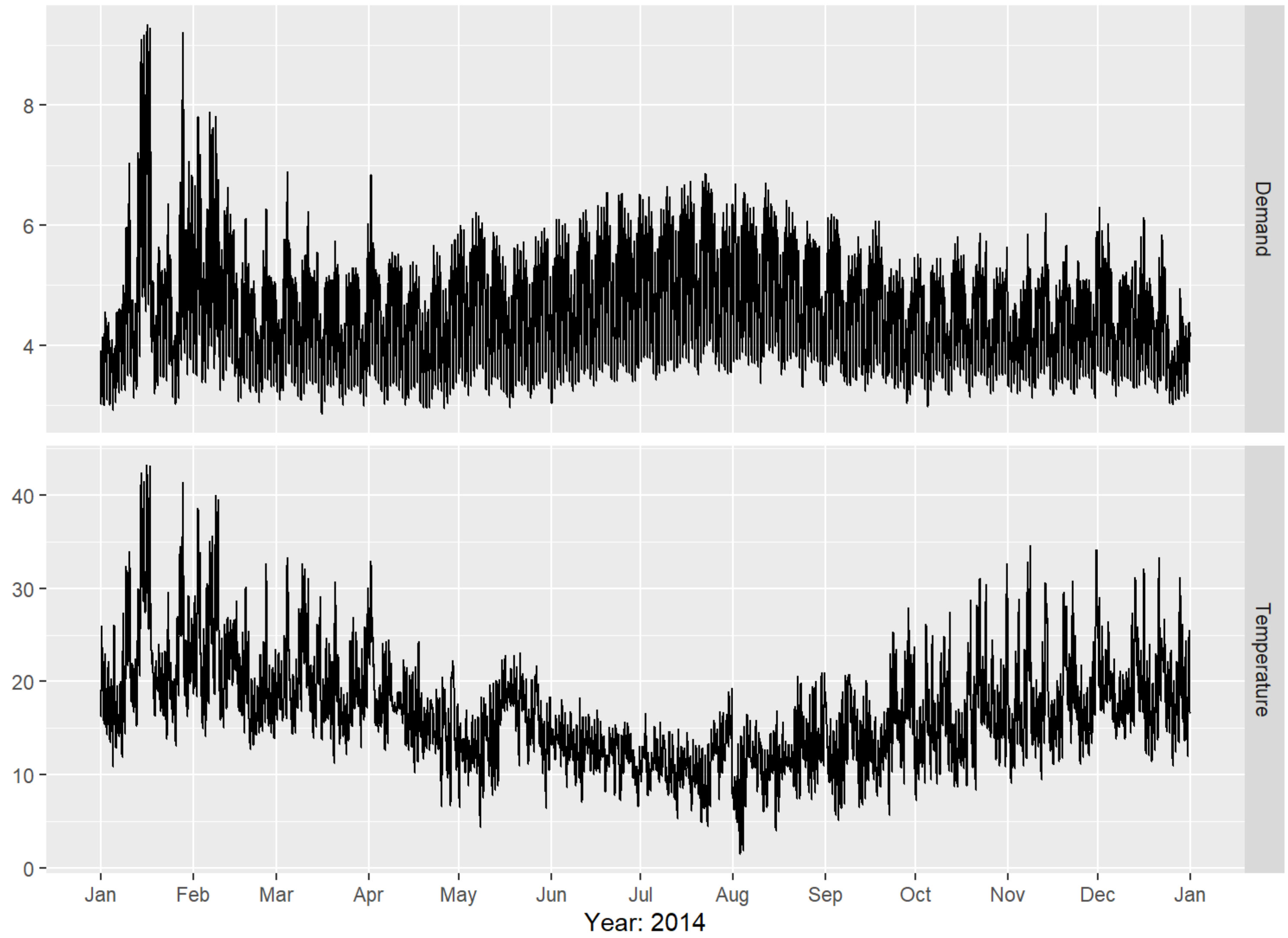

Seasonal subseries plot: antidiabetic drug sales
$ million
30
20
10
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month

Scatterplots

Two time series are plotted, the 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.

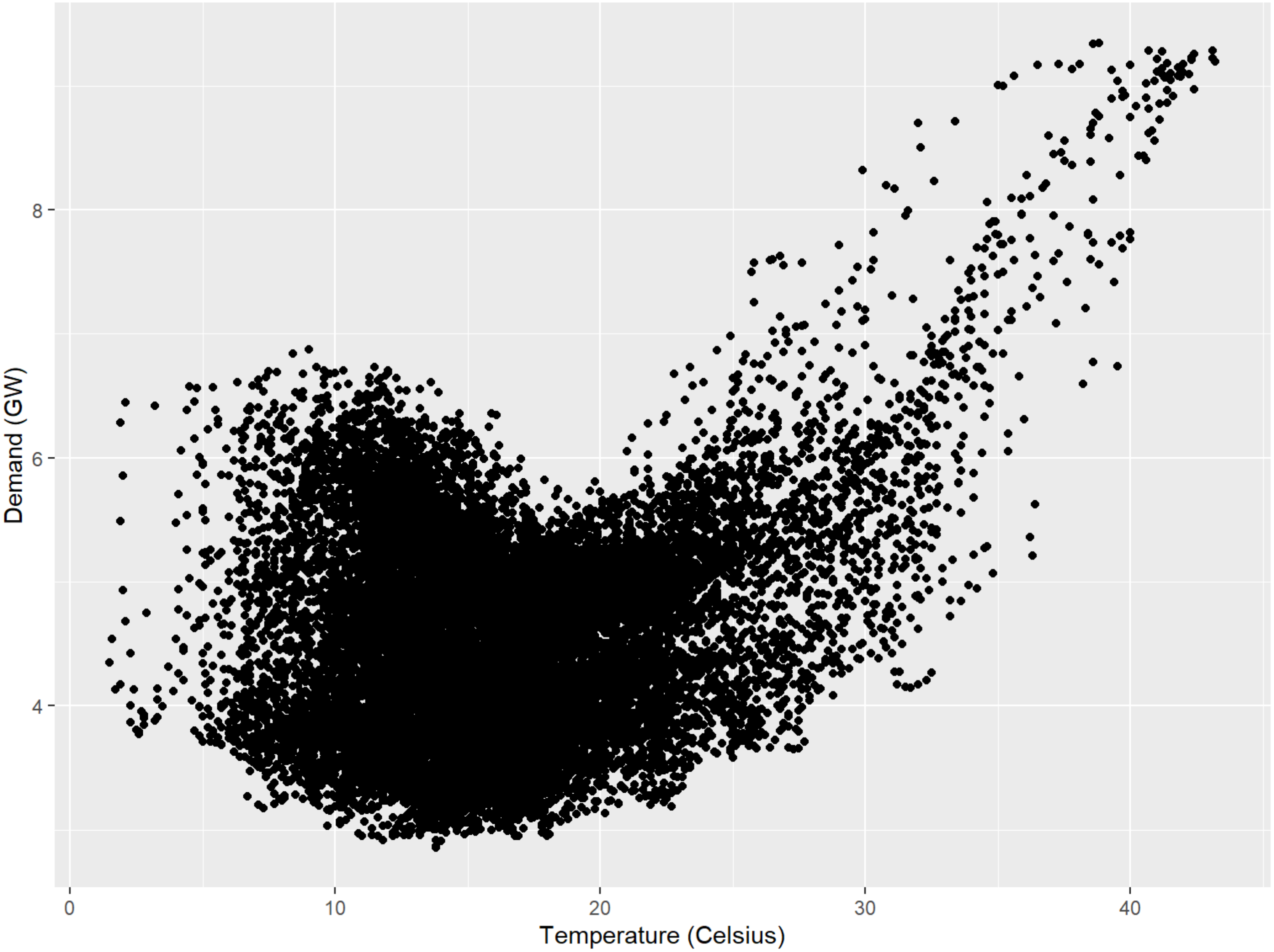
```
month.breaks <- cumsum(c(0,31,28,31,30,31,30,31,31,30,31,30,31)*48)
autoplot(elecdemand[,c(1,3)], facet=TRUE) +
xlab("Year: 2014") + ylab("") +
ggtitle("Half-hourly electricity demand: Victoria, Australia") +
scale_x_continuous(breaks=2014+month.breaks/max(month.breaks),
minor_breaks=NULL, labels=c(month.abb,month.abb[1]))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

Half-hourly electricity demand: Victoria, Australia



The relationship between demand and temperature could be studied by plotting one series against the other:

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


Demand (GW)
8
6
4
0
10
20
30
40
Temperature (Celsius)

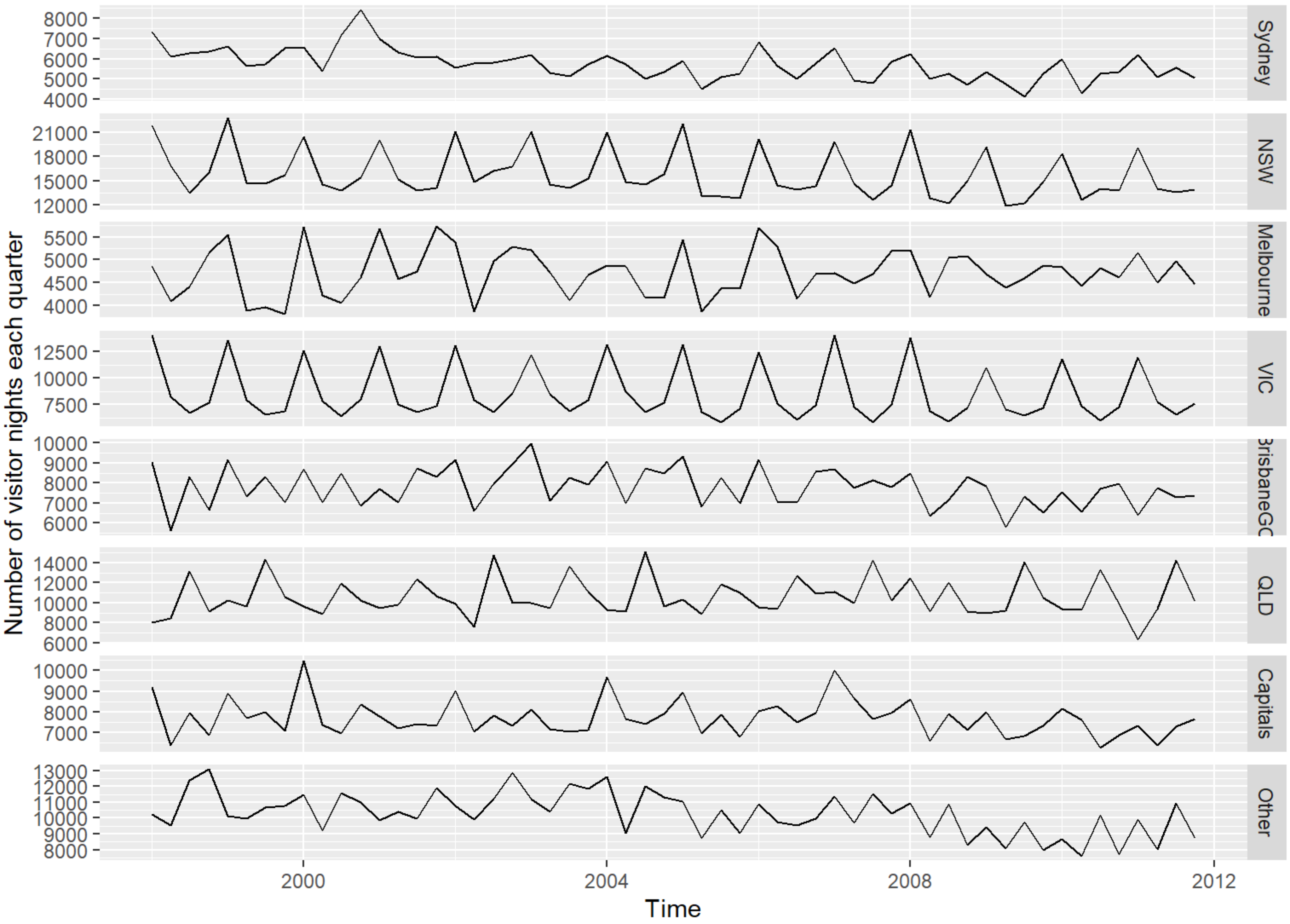
The plot shows high demand happens when temperatures are high due to the effect of air-conditioning. However, demand also increases with low temperatures due to the effect of heating.

Scatterplot matrices

With several potential predictor variables, each variable could be plotted against the each of the other variables

Each of the variables are plotted as follows:

```
autoplot(vn, facets=TRUE) +
ylab("Number of visitor nights each quarter")
```

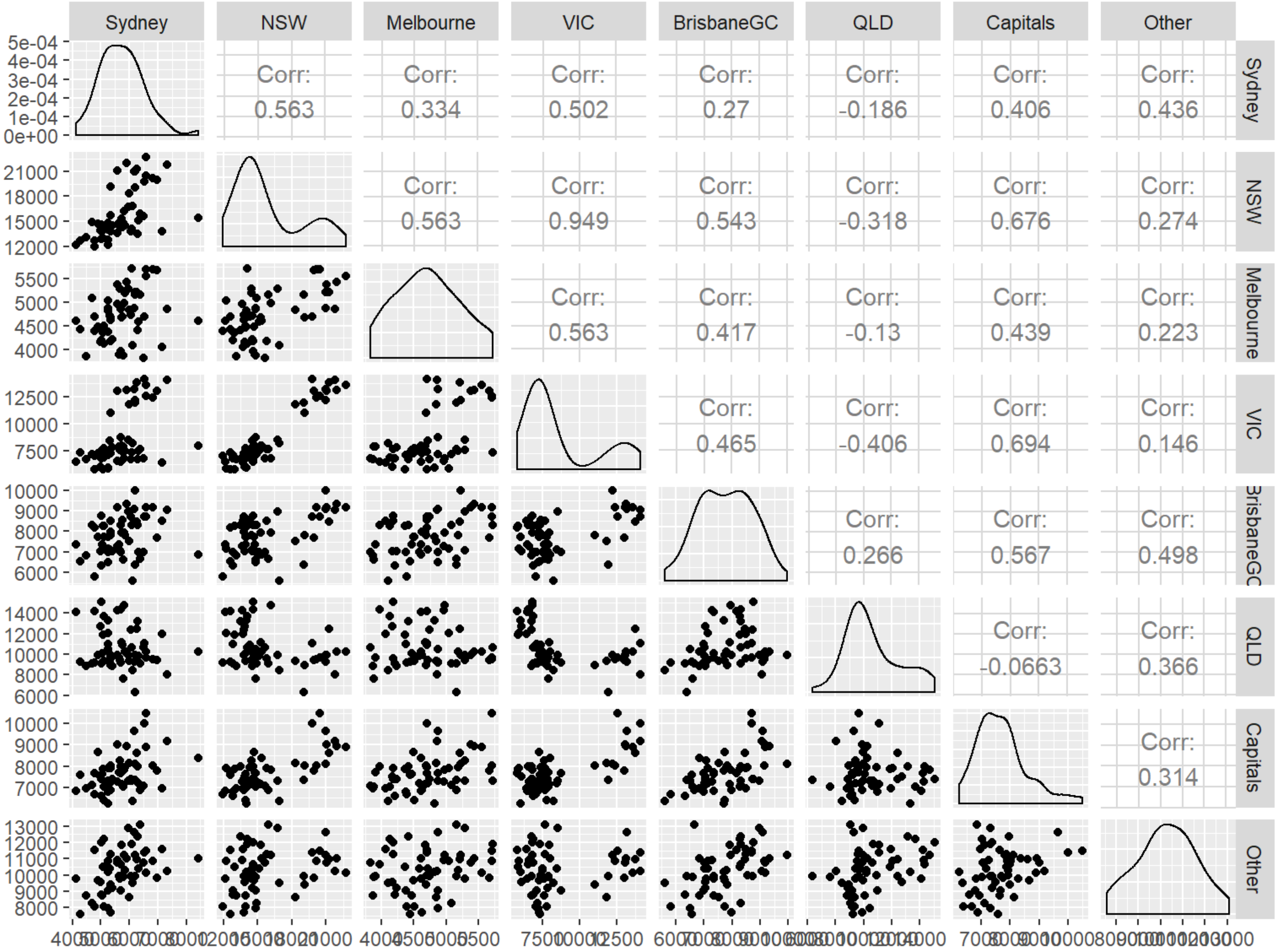

Number of visitor nights each quarter
Sydney
8000
7000
6000
5000
4000
NSW
21000
18000
15000
12000
Melbourne
5500
5000
4500
4000
VIC
12500
10000
7500
BrisbaneGC
10000
9000
8000
7000
6000
QLD
14000
12000
10000
8000
6000
Capitals
10000
9000
8000
7000
Other
13000
12000
11000
10000
9000
8000
2000
2004
2008
2012
Time

The relationships among the eight time series can be plotted each against the others as:

```
vn %>% as.data.frame() %>% GGally::ggpairs()
```

(Note1: GGally package may not be installed. Please use install button to install GGally making sure “install dependencies” is checked. If after running the above commands, an error message appears that a package is missing, please type in the name of the package in the install window.)

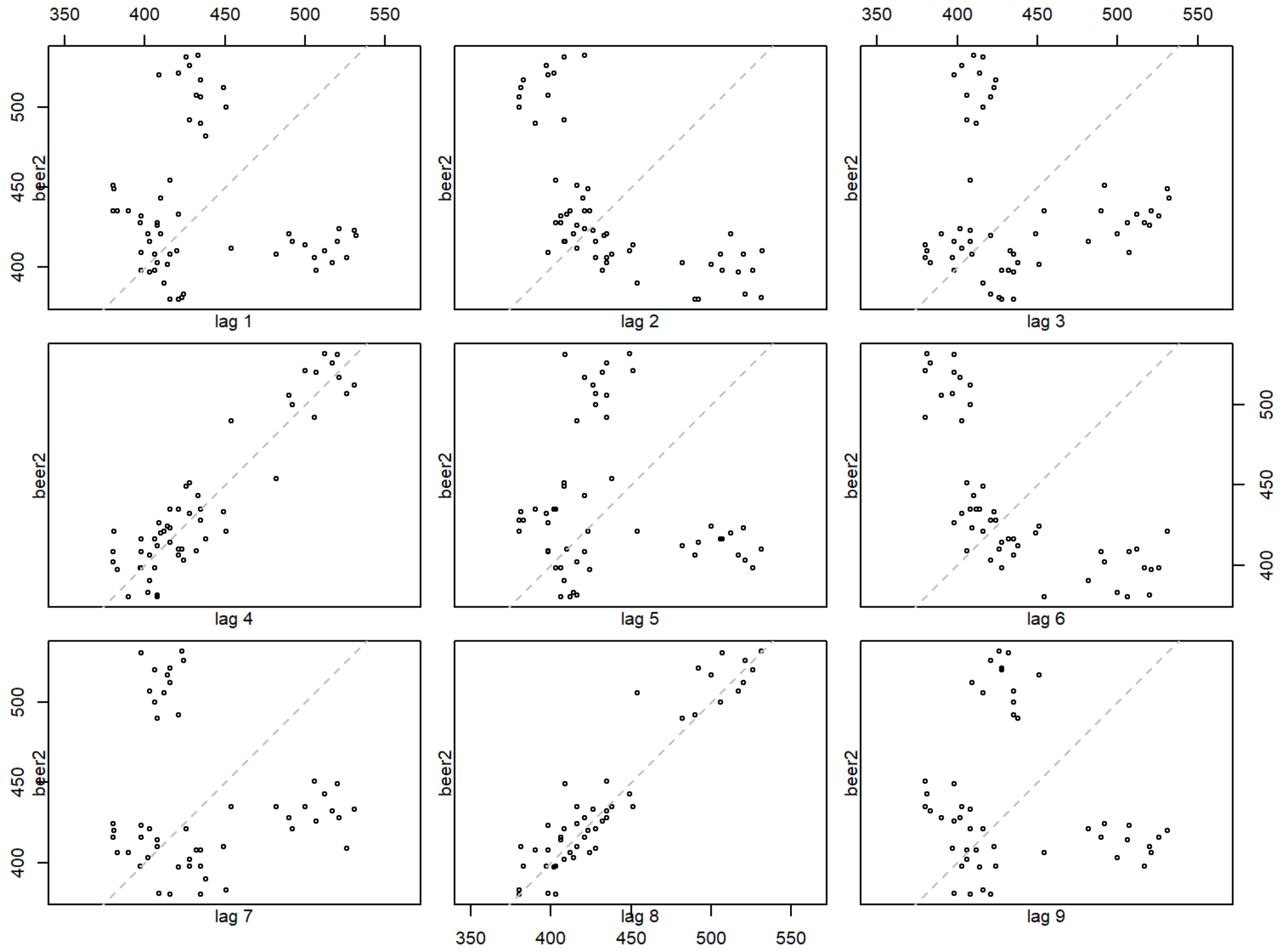
(Note2: %>% is the “pipe” operator which allows commands in R to be written from left to right. The left hand side of each pipe is passed as the first argument to the function on the right hand side.)


Sydney
NSW
Melbourne
VIC
BrisbaneGC
QLD
Capitals
Other
Corr: 0.563
Corr: 0.334
Corr: 0.502
Corr: 0.27
Corr: -0.186
Corr: 0.406
Corr: 0.436
Corr: 0.563
Corr: 0.949
Corr: 0.543
Corr: -0.318
Corr: 0.676
Corr: 0.274
Corr: 0.563
Corr: 0.417
Corr: -0.13
Corr: 0.439
Corr: 0.223
Corr: 0.465
Corr: -0.406
Corr: 0.694
Corr: 0.146
Corr: 0.266
Corr: 0.567
Corr: 0.498
Corr: -0.0663
Corr: 0.366
Corr: 0.314
5e-04
4e-04
3e-04
2e-04
1e-04
0e+00
21000
18000
15000
12000
5500
5000
4500
4000
12500
10000
7500
10000
9000
8000
7000
6000
14000
12000
10000
8000
6000
10000
9000
8000
7000
13000
12000
11000
10000
9000
8000
Sydney
NSW
Melbourne
VIC
BrisbaneGC
QLD
Capitals
Other
7500
10000
12500

- 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. 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.
- Outliers can also be seen. In this example, there is one unusually high quarter for Sydney, corresponding to the 2000 Sydney Olympics.

Lag plots

```
beer2 <- window(ausbeer, start=1992, end=2006)
lag.plot(beer2, lags=9, do.lines=FALSE)
```


350
400
450
500
550
beer2
lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
lag 9

- Graph displays scatterplots of quarterly Australian beer production, where the horizontal axis shows lagged values of the time series. Each graph shows y_t plotted against y_{t-k} for different values of k.
- The relationship is strongly positive at lags 4 and 8, reflecting the strong quarterly seasonality in the data.
- 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.

Numerical data summaries

- Numerical summaries of data sets are widely used to capture some essential features of the data with a few numbers.
- A summary number calculated from the data is called a statistic.
- For a single data set, the most widely used statistics are the *average* and *median*.
- The most commonly used bivariate statistic is the *correlation coefficient*.

Average

Suppose N denotes the total number of observations and x_i denotes the ith observation. Then the average can be written as

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = (x_1 + x_2 + x_3 + \cdots + x_N)/N\ .$$

The average is also called the sample mean.

Consider the carbon footprint from 20 vehicles:
$x = \{4.0, 4.4, 5.9, 5.9, 6.1, 6.1, 6.1, 6.3, 6.3, 6.3, 6.6, 6.6, 6.6,.$
$6.6, 6.6, 6.6, 6.6, 6.8, 6.8, 6.8\}$

In this example, $N = 20$ and x_i denotes the carbon footprint of vehicle i.

Then the average carbon footprint is

$$\bar{x} = \frac{1}{20}\sum_{i=1}^{20} x_i = (x_1 + x_2 + x_3 + \cdots + x_{20})/20$$
$$= (4.0 + 4.4 + 5.9 + \cdots + 6.8 + 6.8 + 6.8)/20$$
$$= 124/20 = 6.2 \text{ tons CO2.}$$

Median

The median is the middle observation when the data are placed in order.

In this case, there are 20 observations and so the median is the average of the 10th and 11th largest observations. That is

$$median = (6.3 + 6.6)/2 = 6.45\ .$$

Percentiles are useful for describing the distribution of data. For example, 90% of the data are no larger than the 90th percentile.

In the carbon footprint example, the 90th percentile is 6.8 because 90% of the data (18 observations) are less than or equal to 6.8. Similarly, the 75th percentile is 6.6 and the 25th percentile is 6.1.

The median is the 50th percentile.

Measures of spread

A useful measure of how spread out the data are is the *interquartile range* or IQR. This is simply the difference between the 75th and 25th percentiles. Thus it contains the middle 50% of the data. For the example,

$$IQR = (6.6 - 6.1) = 0.5\ .$$

An alternative and more common measure of spread is the *standard deviation*. This is given by the formula

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}\,.$$

In the example, the standard deviation is

$$s = \sqrt{\frac{1}{19}[(4.0 - 6.2)^2 + (4.4 - 6.2)^2 + \cdots + (6.8 - 6.2)^2]} = 0.74.$$

Correlation coefficient

measures the strength of the relationship between two variables and can be written as

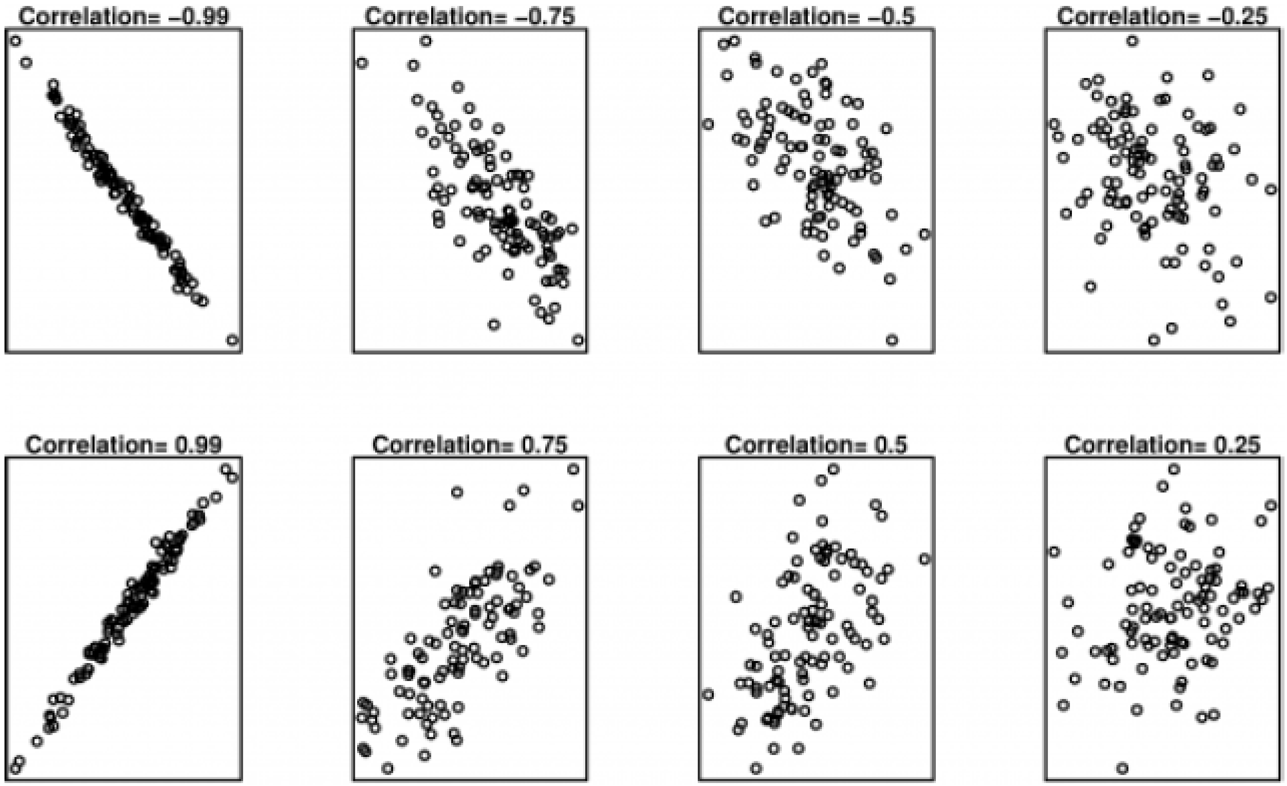
$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}} \ ,$$

where the first variable is denoted by x and the second variable by y.

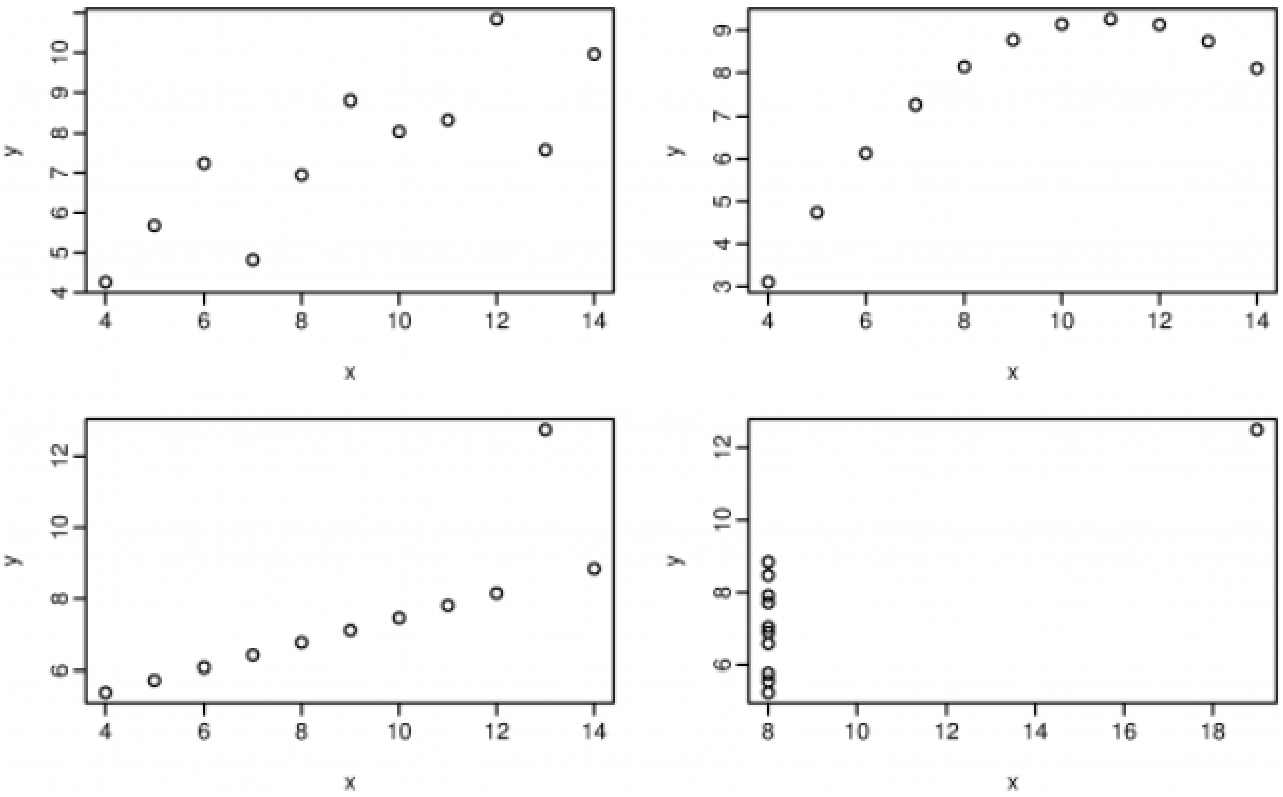
The correlation coefficient only measures the strength of the *linear* relationship; it is possible for two variables to have a strong non-linear relationship but low correlation coefficient.

The value of r always lies between -1 and 1 with negative values indicating a negative relationship and positive values indicating a postive relationship.

Correlation examples



Scatterplots of data sets with different levels of correlation.



Each of these plots has a correlation coefficient of 0.82. This shows how important it is not to rely only on correlation coefficients but also to look at the plots of the data.

Autocorrelation

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

There are several autocorrelation coefficients, depending on the lag length.

For example, r_1 measures the relationship between y_t and y_{t-1}, r_2 measures the relationship between y_t and y_{t-2}, and so on.

The value of r_k can be written as

$$r_k = \frac{\sum\limits_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum\limits_{t=1}^{T} (y_t - \bar{y})^2},$$

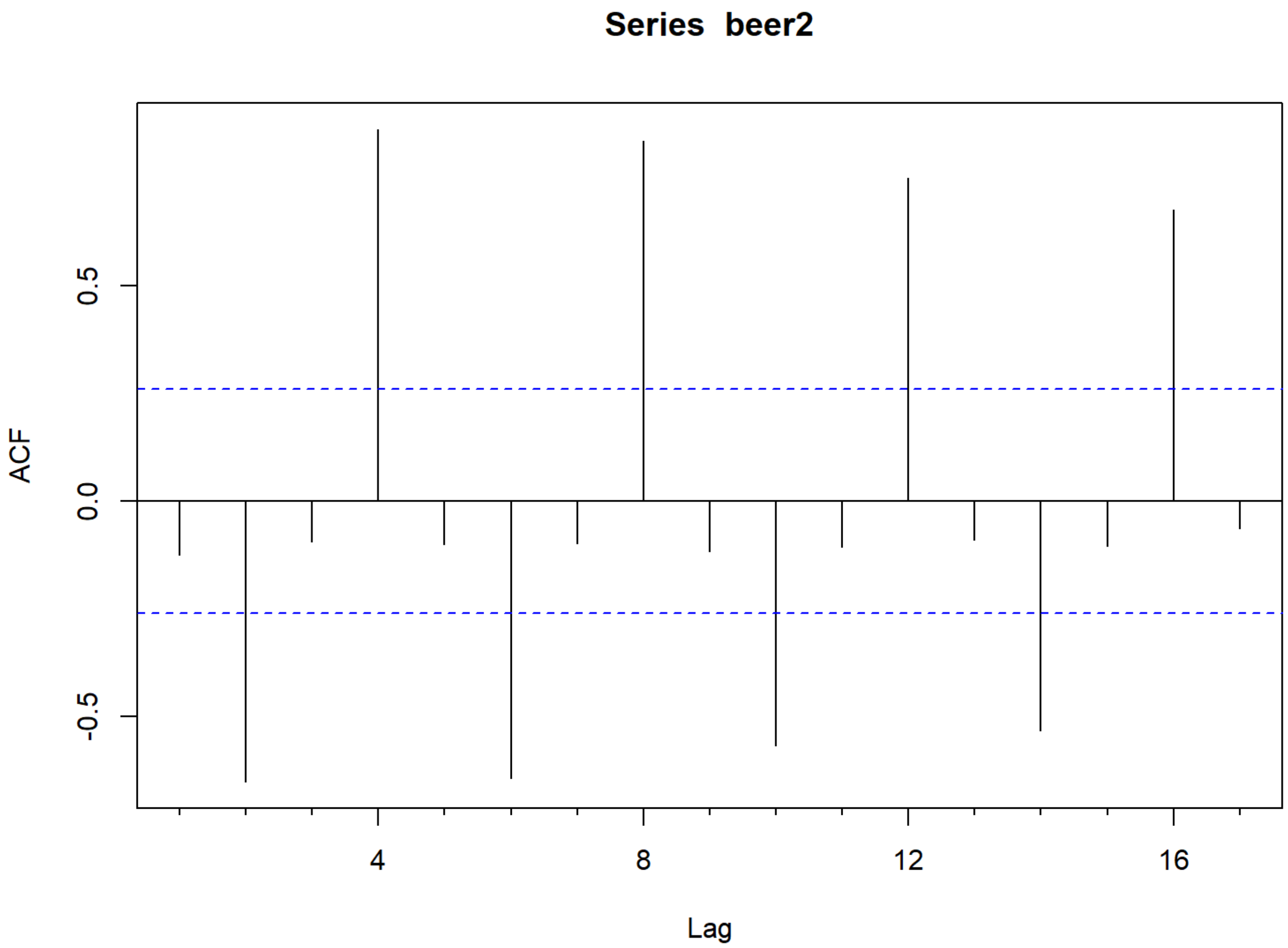
where T is the length of the time series.

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

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
-0.126	-0.650	-0.094	0.863	-0.099	-0.642	-0.098	0.834	-0.116

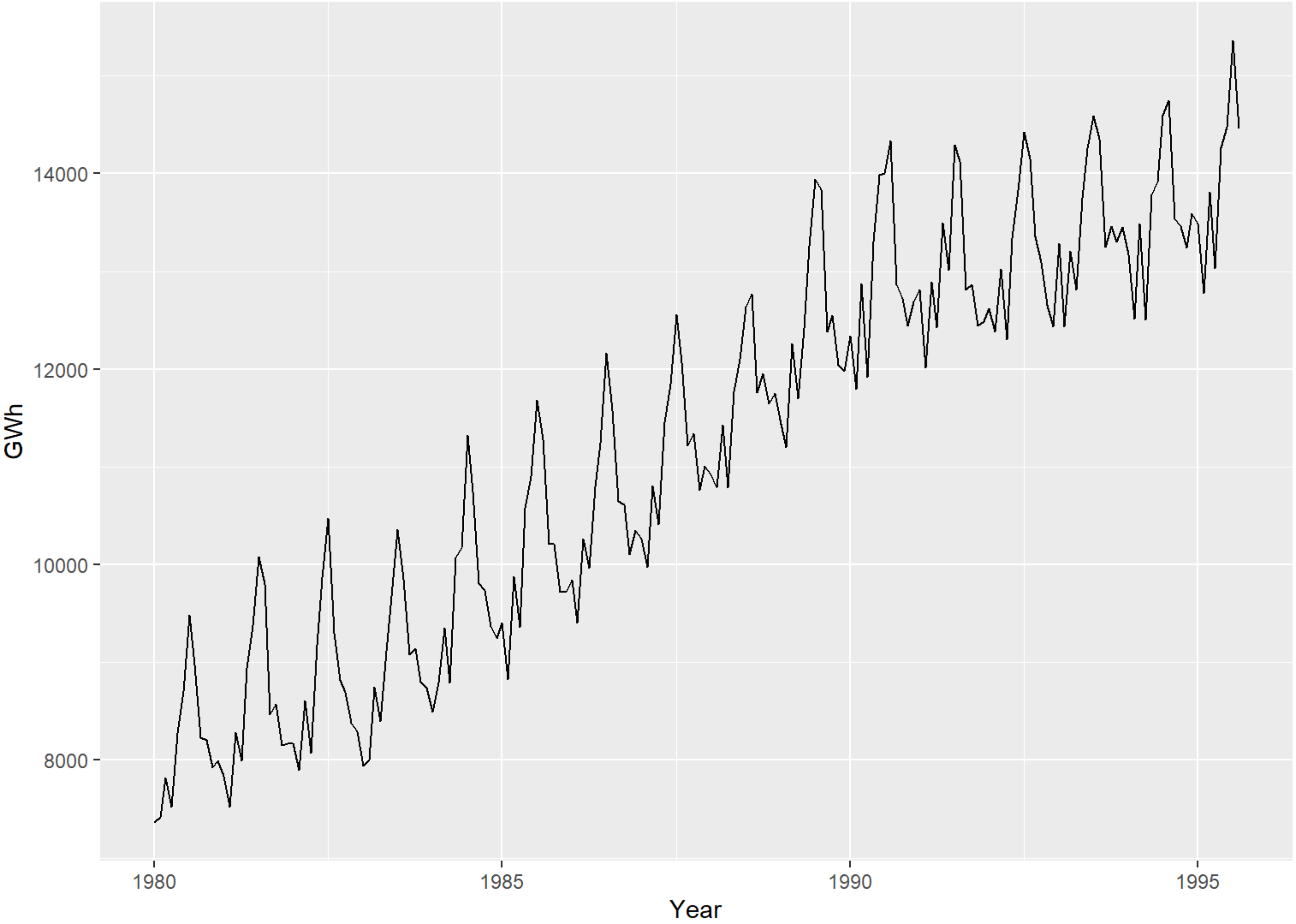
Autocorrelation function (ACF) or correlogram

```
Acf(beer2)
```


Series beer2
ACF
0.5
0.0
-0.5
4
8
12
16
Lag

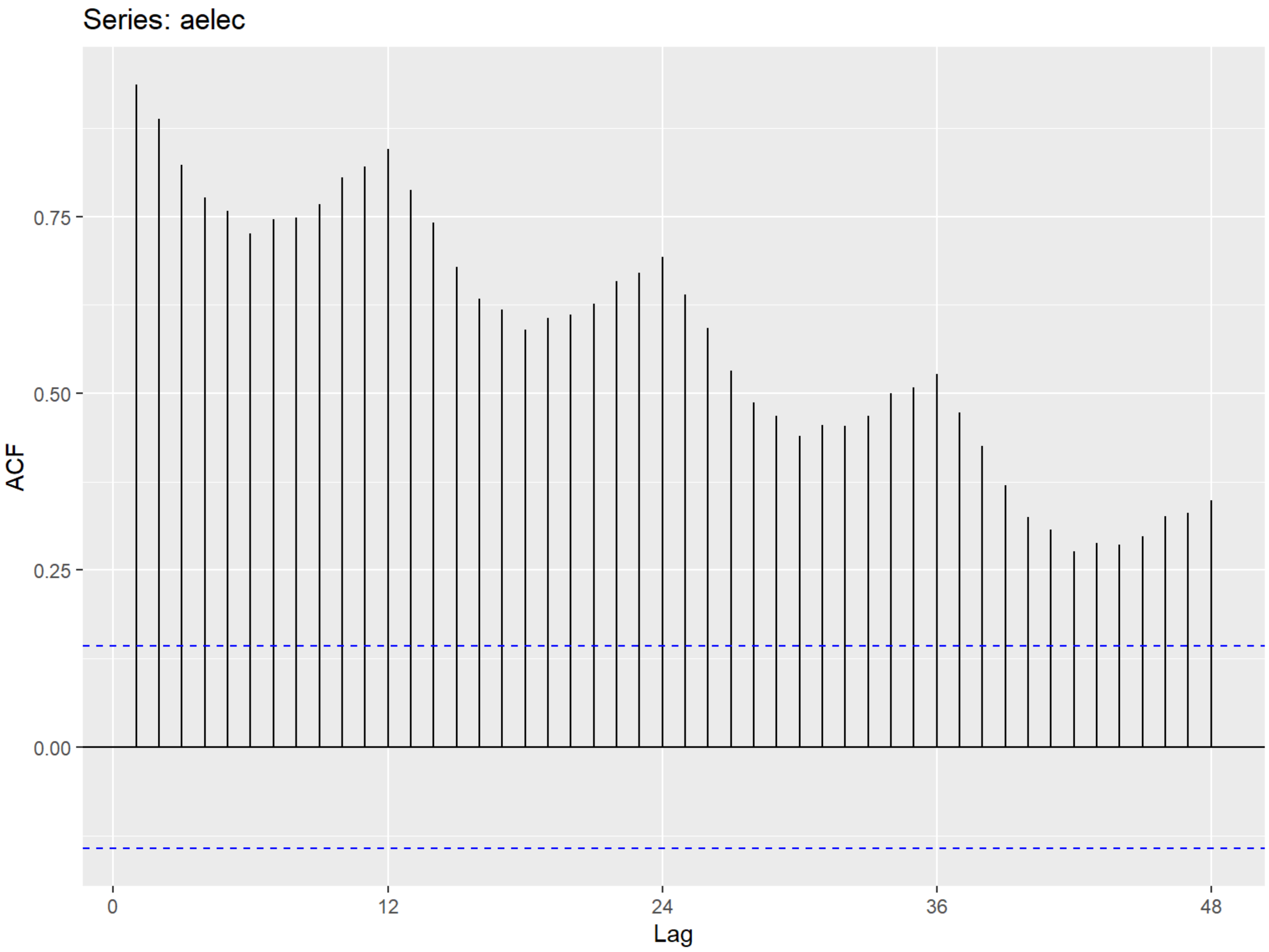
Trend and seasonality in ACF plots

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


GWh
14000
12000
10000
8000
1980
1985
1990
1995
Year

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

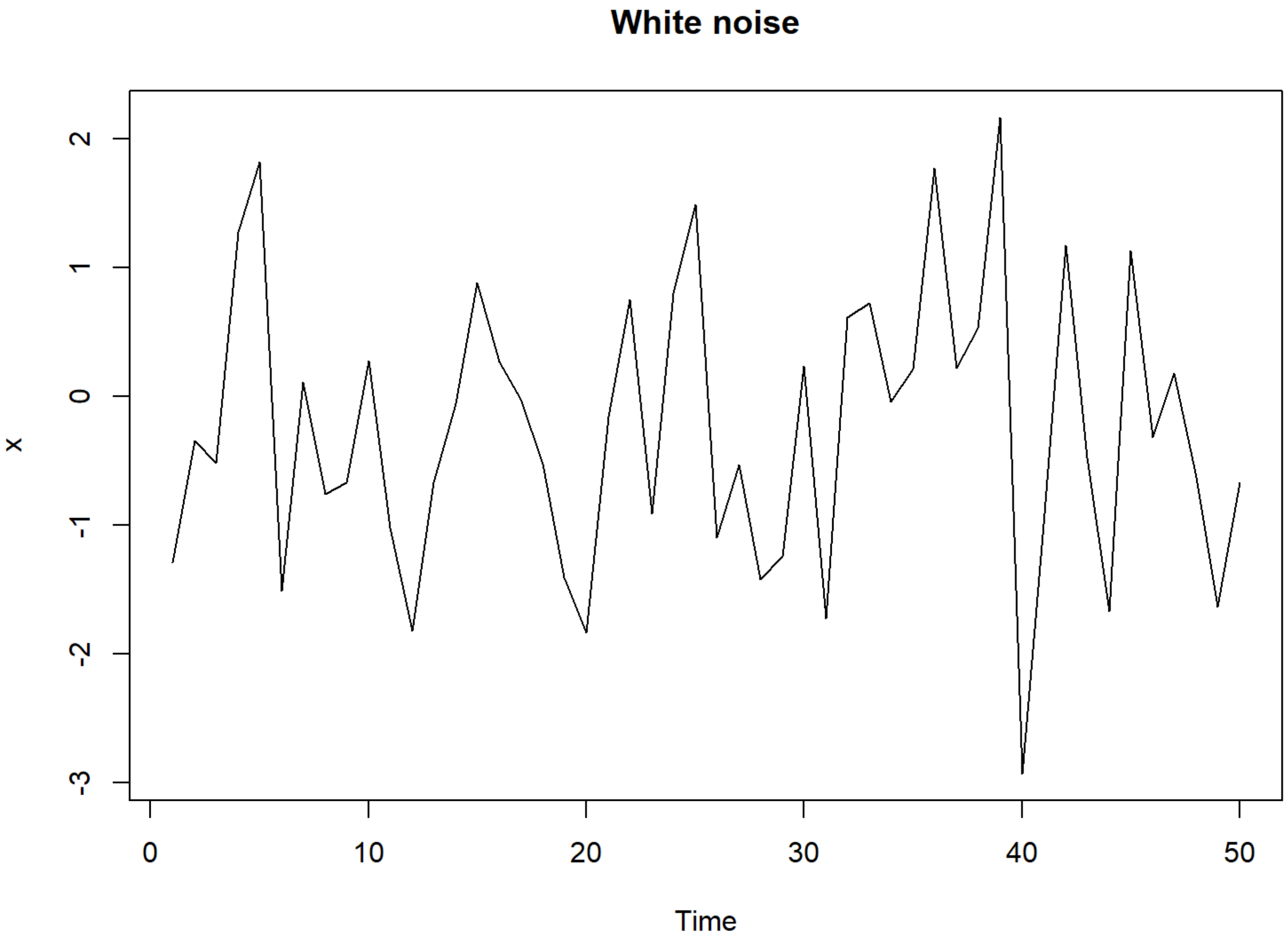
```
ggAcf(aelec, lag=48)
```


Series: aelec
ACF
0.75
0.50
0.25
0.00
Lag
0
12
24
36
48

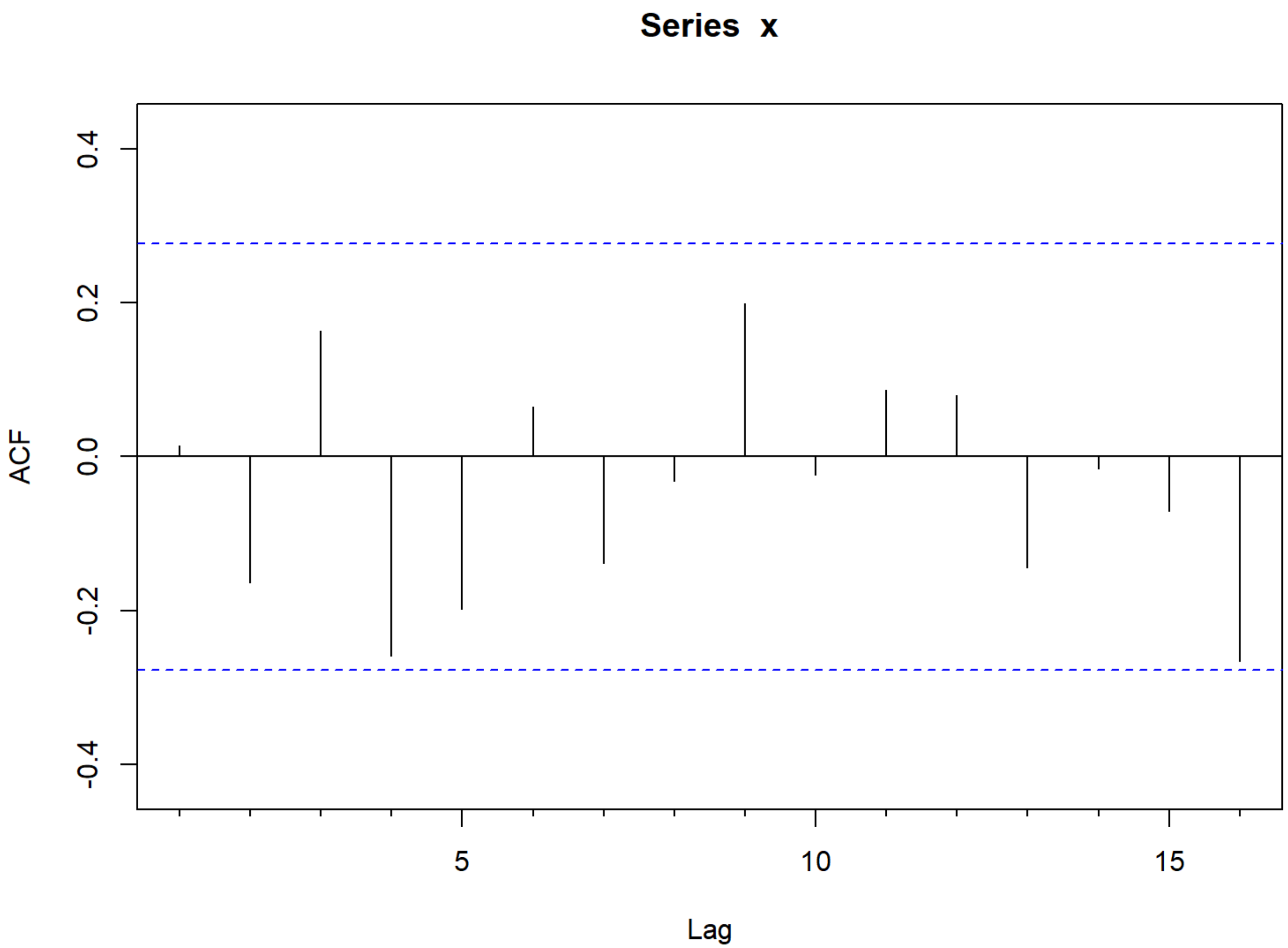
White noise

Series that show no autocorrelation are called “white noise”.

```r
set.seed(30); x <- ts(rnorm(50)); plot(x, main="White noise")
```


White noise
x
2
1
0
-1
-2
-3
0
10
20
30
40
50
Time

```
Acf(x)
```



For a white noise series, we expect 95% of the spikes in the ACF to lie within $\pm 2/\sqrt{T}$. Here $T = 50$: $\pm 2/\sqrt{50} = \pm 0.28$.

Homework 1: Section 2.10 (2^{nd} edition), Exercise 3. Due next Tuesday, 29 Jan.

If time permits, we will continue with Chapter 3.1 (Some Simple forecasting methods)

Chapter 3

Section 3.1: Some simple forecasting methods

Some forecasting methods are very simple and surprisingly effective. Here are four methods that we will use as benchmarks for other forecasting methods.

1. Average method
2. Naive method
3. Seasonal naive method
4. Drift method

Average method

Here, the forecasts of all future values are equal to the mean of the historical data:

$$\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T \, .$$

The notation $\hat{y}_{T+h|T}$ is a short-hand for the estimate of y_{T+h} based on the data $y_1, \ldots, y_T$.

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

This method can also be used for cross-sectional data (when we are predicting a value not included in the data set). Then the prediction for values not observed is the average of those values that have been observed.

Naive method

All forecasts are simply set to be the value of the last observation. That is, the forecasts of all future values are set to be y_T, where y_T is the last observed value.

```
naive(y, h)
rwf(y, h) # Alternative
```

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

Seasonal naive method

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).

- For example, with monthly data, the forecast for all future February values is equal to the last observed February value.
- With quarterly data, the forecast of all future Q2 values is equal to the last observed Q2 value (where Q2 means the second quarter).
- Similar rules apply for other months and quarters, and for other seasonal periods.

```
snaive(y, h)
```

Drift method

A variation on the naive 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.

So the forecast for time $T + h$ is given by

$$y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + h\left(\frac{y_T - y_1}{T-1}\right).$$

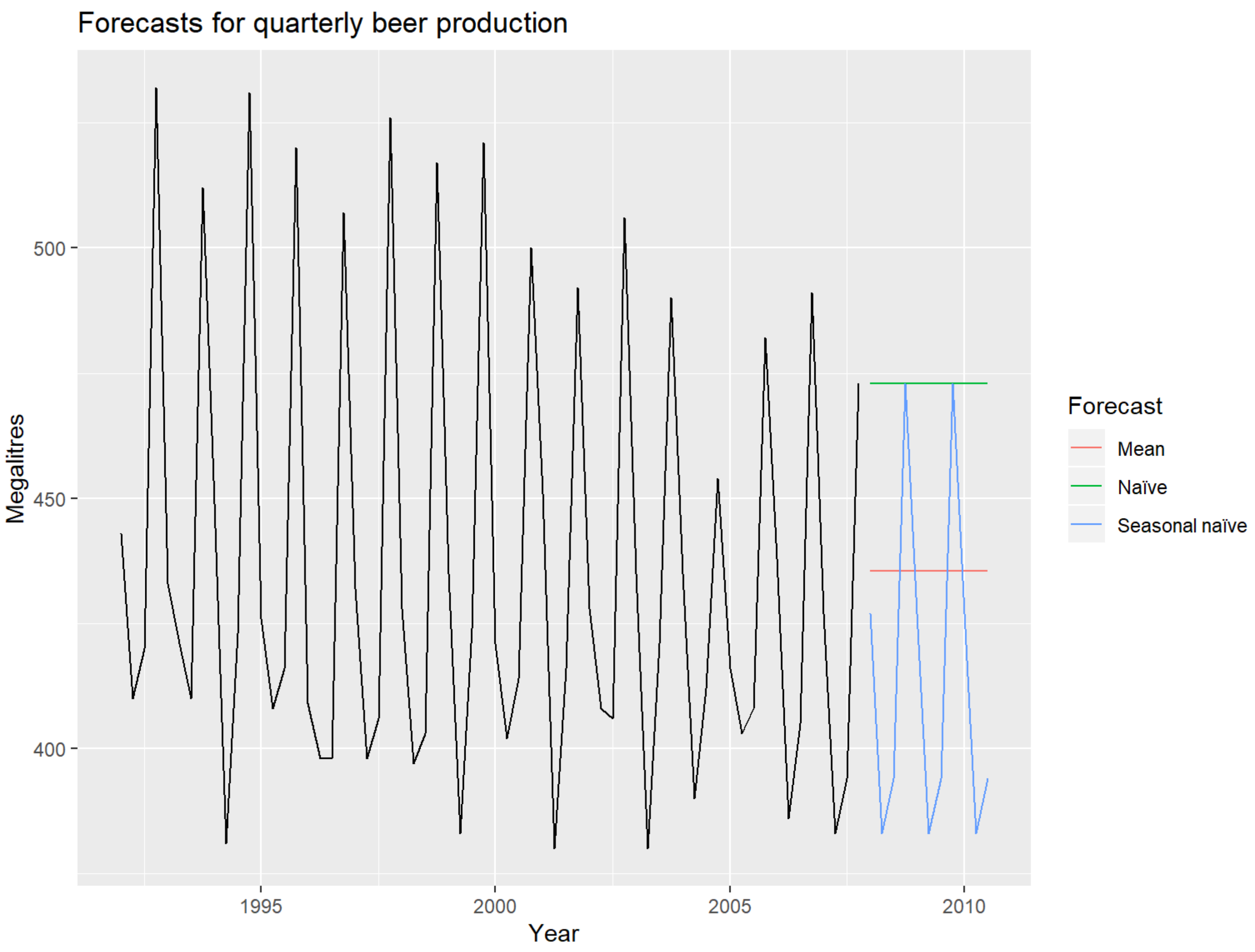
This is equivalent to drawing a line between the first and last observation, and extrapolating it into the future.

```
rwf(y, h, drift=TRUE)
```

Example I

The first three methods applied to the quarterly beer production data.

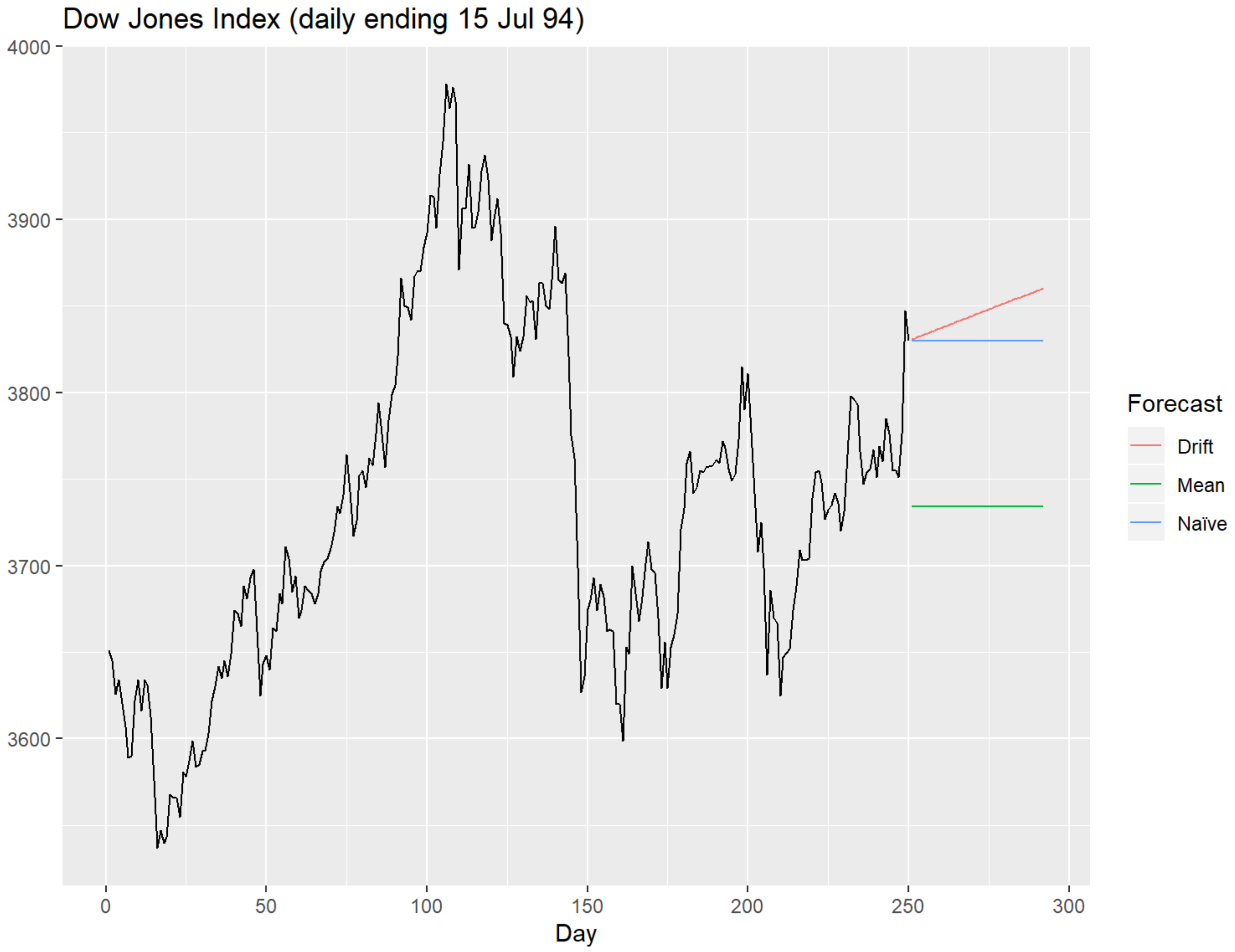
```
# Set training data from 1992-2007
beer2 <- window(ausbeer,start=1992,end=c(2007,4))
# Plot some forecasts
autoplot(beer2) +
forecast::autolayer(meanf(beer2, h=11)$mean, series="Mean") +
forecast::autolayer(naive(beer2, h=11)$mean, series="NaÃ¯ve") +
forecast::autolayer(snaive(beer2, h=11)$mean, series="Seasonal naÃ¯ve") +
ggtitle("Forecasts for quarterly beer production") +
xlab("Year") + ylab("Megalitres") +
guides(colour=guide_legend(title="Forecast"))
```


Forecasts for quarterly beer production
Megalitres
500
450
400
1995
2000
2005
2010
Year
Forecast
Mean
Naïve
Seasonal naïve

Example 2

Here the non-seasonal methods were applied to a series of 250 days of the Dow Jones Index.

```
# Set training data to first 250 days
dj2 <- window(dj,end=250)
# Plot some forecasts
autoplot(dj2) +
forecast::autolayer(meanf(dj2, h=42)$mean, series="Mean") +
forecast::autolayer(rwf(dj2, h=42)$mean, series="NaÃ¯ve") +
forecast::autolayer(rwf(dj2, drift=TRUE, h=42)$mean, series="Drift") +
ggtitle("Dow Jones Index (daily ending 15 Jul 94)") +
xlab("Day") + ylab("") +
guides(colour=guide_legend(title="Forecast"))
```


Dow Jones Index (daily ending 15 Jul 94)
4000
3900
3800
3700
3600
0
50
100
150
200
250
300
Day
Forecast
Drift
Mean
Naïve

- At times, these simple methods may be the best forecasting method available; but in many cases, these methods will serve as benchmarks rather than the method of choice.
- Any other new forecasting methods considered 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.

Thank you for your attention

We continue next week with Chapter 3