MONASH University

MONASH BUSINESS SCHOOL

Time Series in R: Forecasting and Visualisation

Time series in R

29 May 2017

Outline

Time series

- Time series consist of sequences of observations collected over time.
- We will assume the time periods are equally spaced.

Time series examples

- Daily IBM stock prices
- Monthly rainfall
- Annual Google profits
- Quarterly Australian beer production

ts objects and ts function

A time series is stored in a ts object in R:

- a list of numbers
- information about times those numbers were recorded.

Example

Year	Observation
2012	123
2013	39
2014	78
2015	52
2016	110

```
y <- ts(c(123,39,78,52,110), start=2012)
```

ts objects and ts function

For observations that are more frequent than once per year, add a frequency argument.
E.g., monthly data stored as a numerical vector z:

```
y <- ts(z, frequency=12, start=c(2003, 1))
```

`ts` objects and `ts` function

```
ts(data, frequency, start)
```

Type of data	frequency	start example
Annual	1	1995
Quarterly	4	c(1995,2)
Monthly	12	c(1995,9)
Daily	7 *or* 365.25	1 *or* c(1995,234)
Weekly	52.18	c(1995,23)
Hourly	24 *or* 168 *or* 8,766	1
Half-hourly	48 *or* 336 *or* 17,532	1

ts objects

- Class: “ts”
- Print and plotting methods available.

```
ausgdp
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 1971           4612 4651
## 1972 4645 4615 4645 4722
## 1973 4780 4830 4887 4933
## 1974 4921 4875 4867 4905
## 1975 4938 4934 4942 4979
## 1976 5028 5079 5112 5127
```

ts objects

```
start(ausgdp)
```

```
## [1] 1971    3
```

```
end(ausgdp)
```

```
## [1] 1998    1
```

```
frequency(ausgdp)
```

```
## [1] 4
```

ts objects

Residential electricity sales

```
elecsales
```

```
## Time Series:
## Start = 1989
## End = 2008
## Frequency = 1
##  [1] 2354 2380 2319 2469 2386 2569 2576 2763 2844
## [10] 3001 3108 3358 3076 3181 3222 3176 3431 3527
## [19] 3638 3655
```

ts objects

```r
start(elecsales)
```

```
## [1] 1989    1
```

```r
end(elecsales)
```

```
## [1] 2008    1
```

```r
frequency(elecsales)
```

```
## [1] 1
```

fpp2

Main package used in this course

```
> library(fpp2)
```

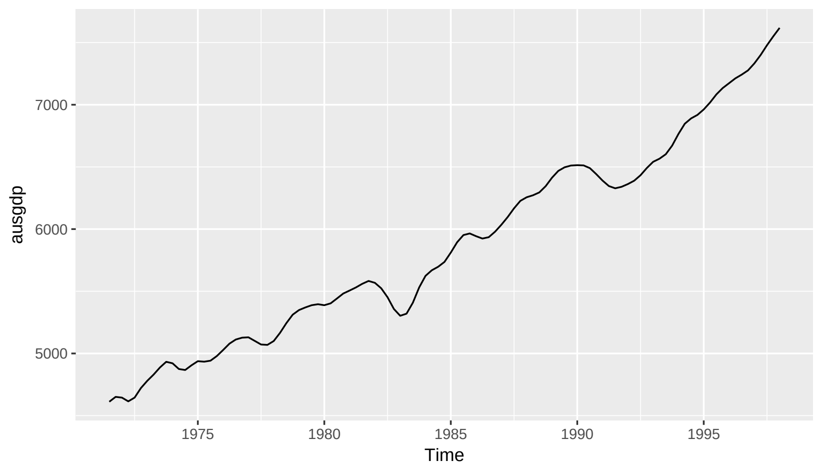
This loads:

- some data for use in examples and exercises
- **forecast** package (for forecasting functions)
- **ggplot2** package (for graphics)
- **fma** package (for lots of time series data)
- **expsmooth** package (for more time series data)

Outline

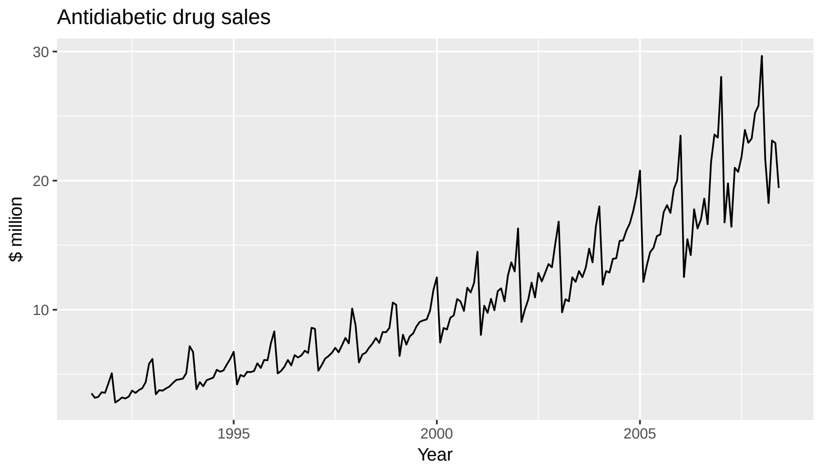
ts objects

```r
autoplot(ausgdp)
```



Time plots

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



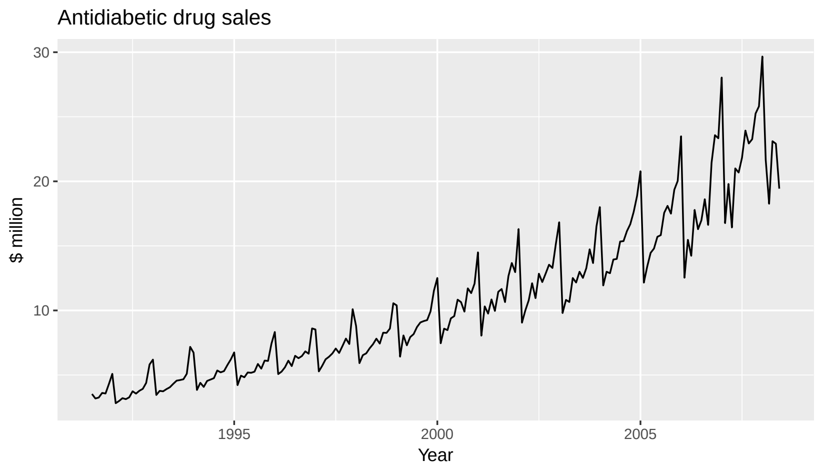
Outline

Lab Session 1

Outline

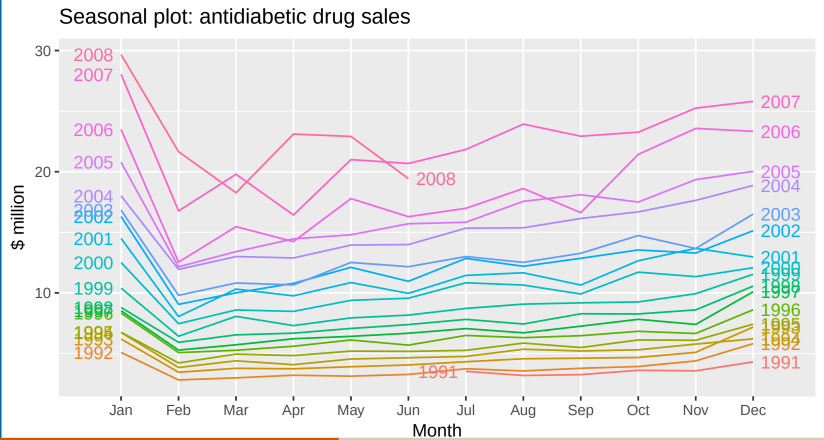
Time plot

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



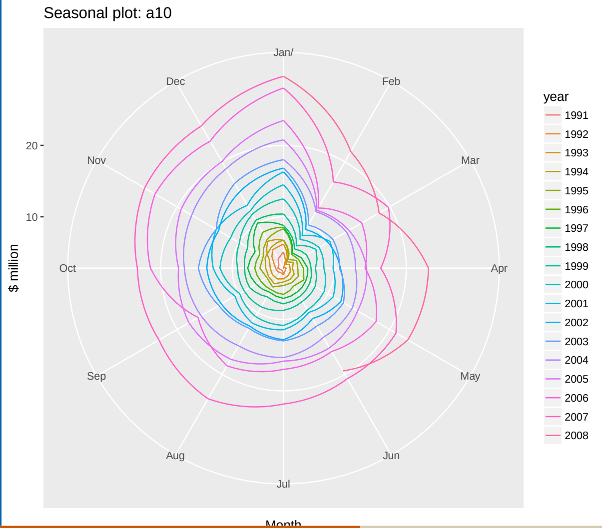
Seasonal plot

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



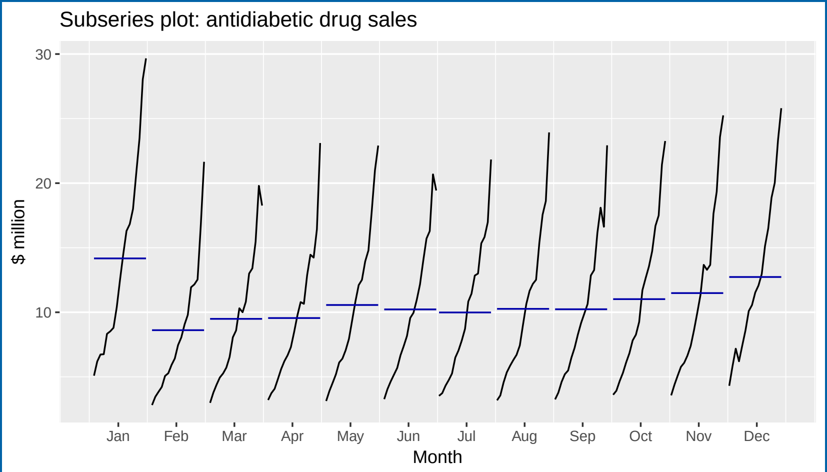
Seasonal polar plots

```r
ggseasonplot(a10, polar=TRUE) + ylab("$ million")
```



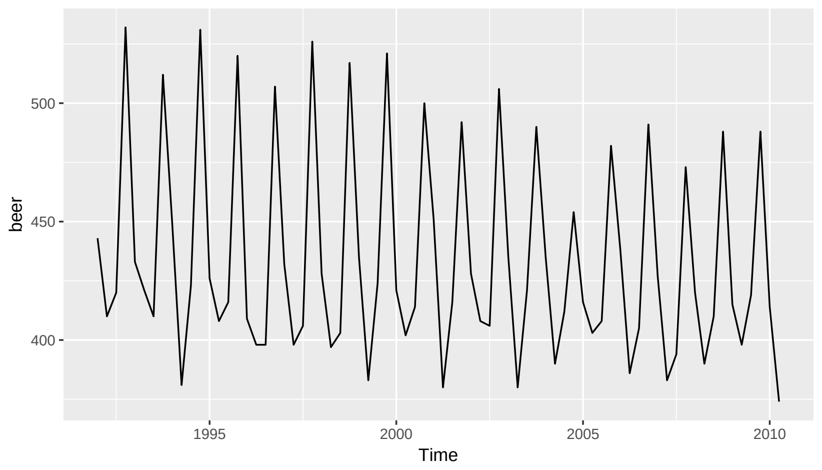
Seasonal subseries plots

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



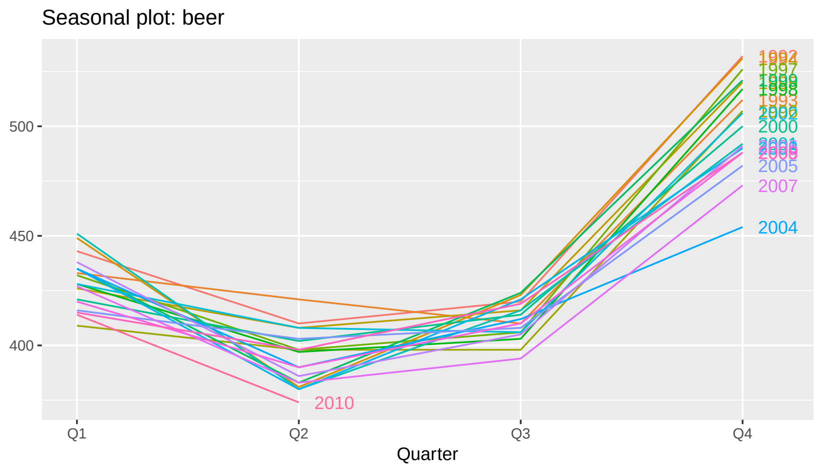
Quarterly Australian Beer Production

```r
beer <- window(ausbeer,start=1992)
autoplot(beer)
```



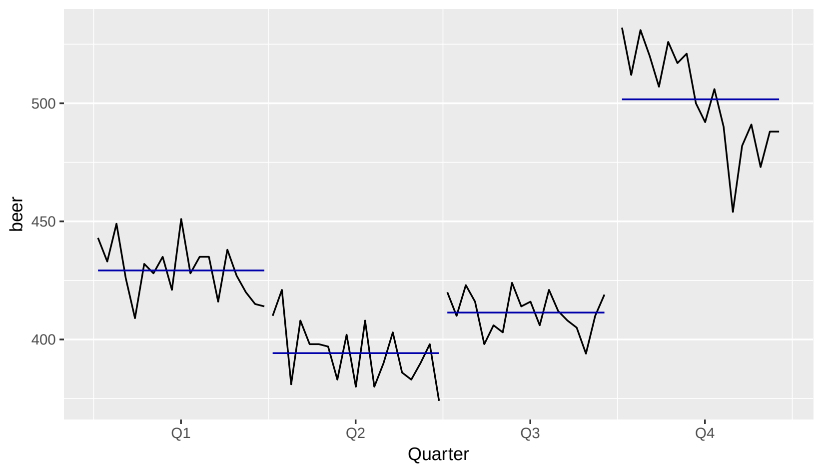
Quarterly Australian Beer Production

```r
ggseasonplot(beer,year.labels=TRUE)
```



Quarterly Australian Beer Production

```
ggsubseriesplot(beer)
```



Outline

Time series patterns

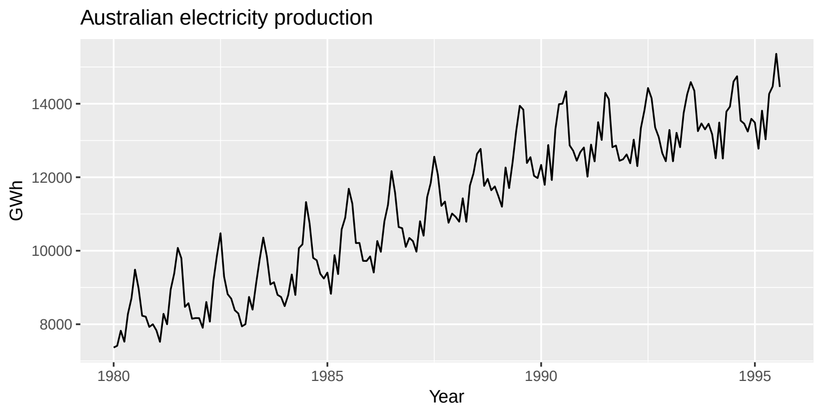
Trend pattern exists when there is a long-term increase or decrease in the data.

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).

Cyclic pattern exists when data exhibit rises and falls that are *not of fixed period* (duration usually of at least 2 years).

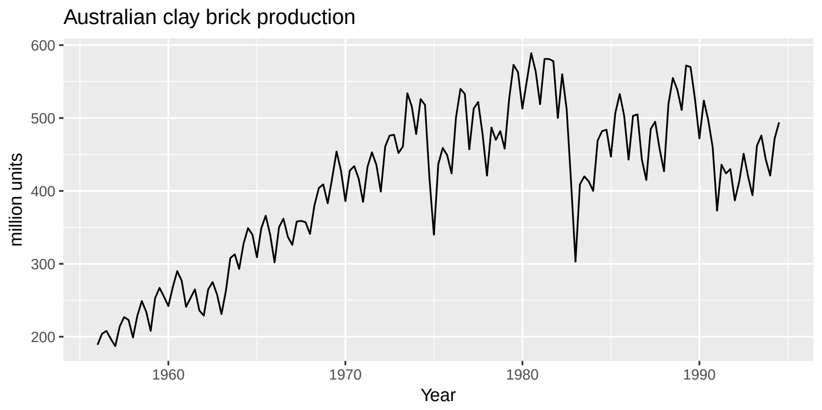
Time series patterns

```
autoplot(window(elec, start=1980)) +
  ggtitle("Australian electricity production") +
  xlab("Year") + ylab("GWh")
```



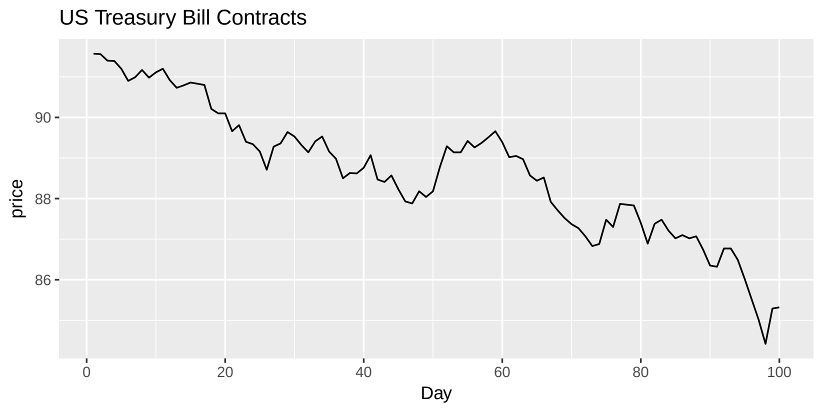
Time series patterns

```
autoplot(bricksq) +
  ggtitle("Australian clay brick production") +
  xlab("Year") + ylab("million units")
```



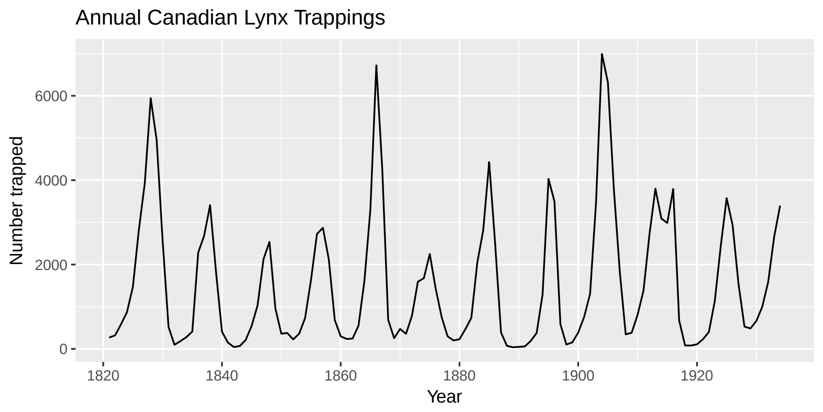
Time series patterns

```r
autoplot(ustreas) +
  ggtitle("US Treasury Bill Contracts") +
  xlab("Day") + ylab("price")
```



Time series patterns

```r
autoplot(lynx) +
  ggtitle("Annual Canadian Lynx Trappings") +
  xlab("Year") + ylab("Number trapped")
```



Seasonal or cyclic?

Differences between seasonal and cyclic patterns:

- seasonal pattern constant length; cyclic pattern variable length
- average length of cycle longer than length of seasonal pattern
- magnitude of cycle more variable than magnitude of seasonal pattern

Seasonal or cyclic?

Differences between seasonal and cyclic patterns:

- seasonal pattern constant length; cyclic pattern variable length
- average length of cycle longer than length of seasonal pattern
- magnitude of cycle more variable than magnitude of seasonal pattern

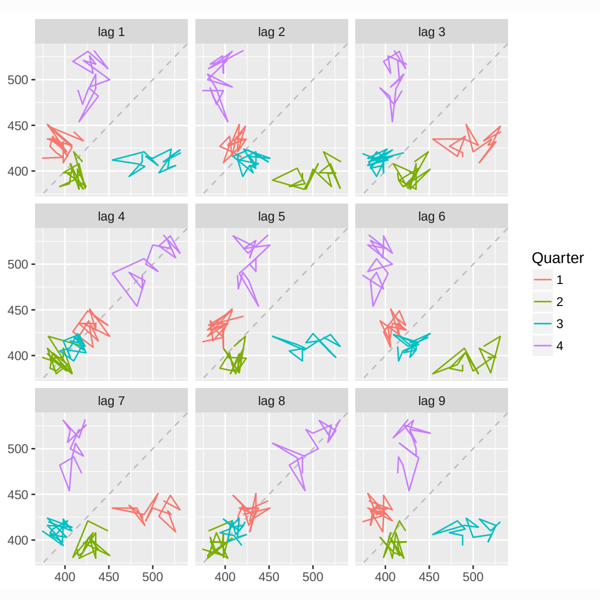
The timing of peaks and troughs is predictable with seasonal data, but unpredictable in the long term with cyclic data.

Outline

Example: Beer production

```
beer <- window(ausbeer, start=1992)
gglagplot(beer)
```

Example: Beer production



Lagged scatterplots

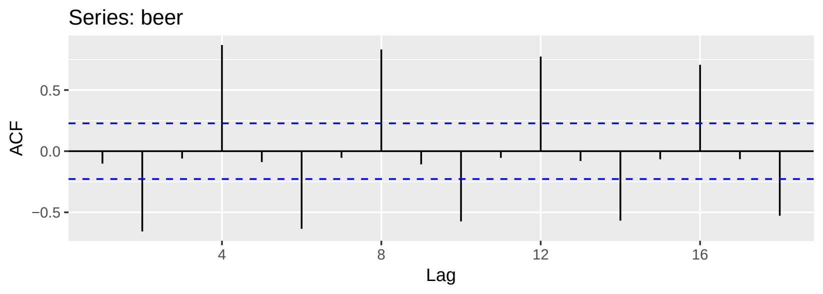
- Each graph shows y_t plotted against y_{t-k} for different values of k.
- The autocorrelations are the correlations associated with these scatterplots.

Autocorrelation

Results for first 9 lags for beer data:

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
-0.102	-0.657	-0.060	0.869	-0.089	-0.635	-0.054	0.832	-0.108

```
ggAcf(beer)
```



Autocorrelation

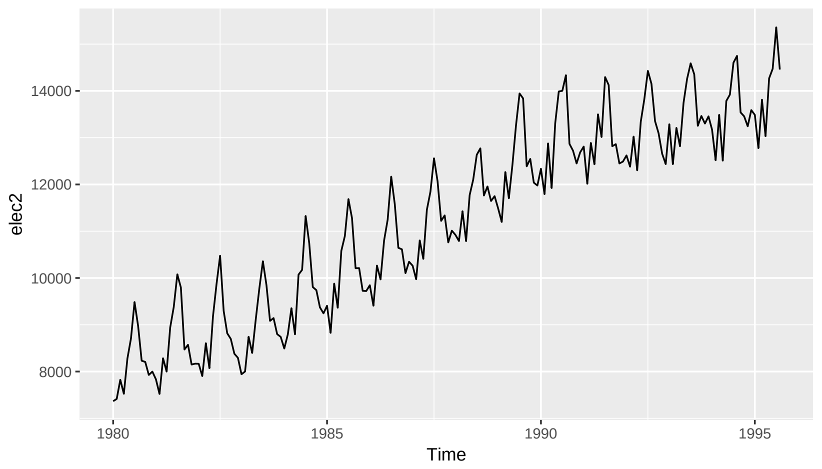
- r_4 higher than for the other lags. This is due to **the seasonal pattern in the data**: the peaks tend to be **4 quarters** apart and the troughs tend to be **2 quarters** apart.
- r_2 is more negative than for the other lags because troughs tend to be 2 quarters behind peaks.
- Together, the autocorrelations at lags 1, 2, $\ldots$, make up the *autocorrelation* or ACF.
- The plot is known as a **correlogram**

Trend and seasonality in ACF plots

- When data have a trend, the autocorrelations for small lags tend to be large and positive.
- When data are seasonal, the autocorrelations will be larger at the seasonal lags (i.e., at multiples of the seasonal frequency)
- When data are trended and seasonal, you see a combination of these effects.

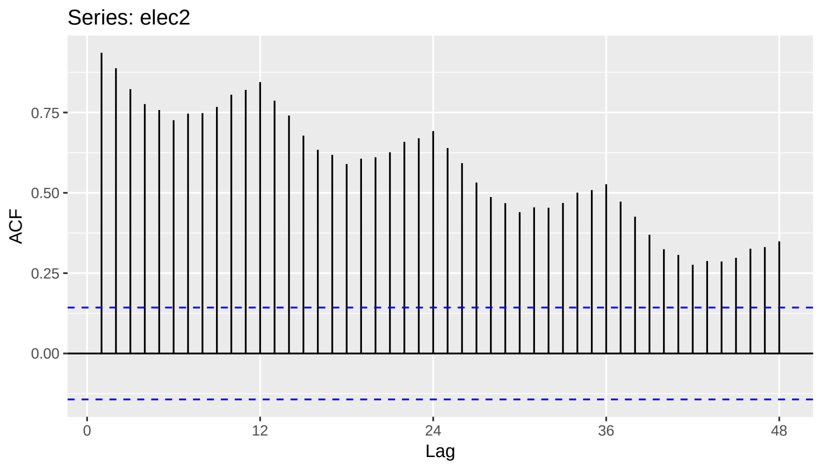
Aus monthly electricity production

```r
elec2 <- window(elec, start=1980)
autoplot(elec2)
```



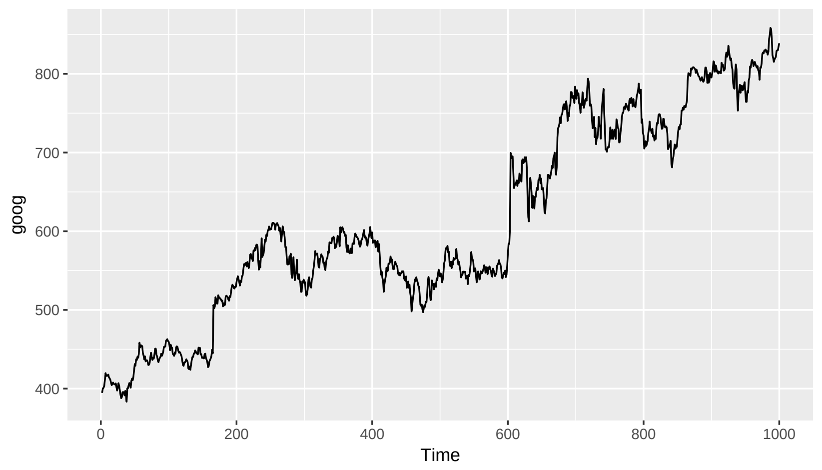
Aus monthly electricity production

```
ggAcf(elec2, lag.max=48)
```



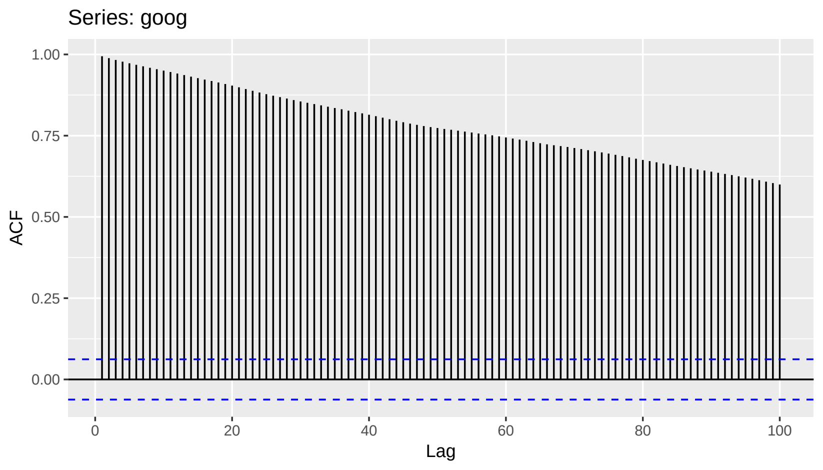
Google stock price

```
autoplot(goog)
```

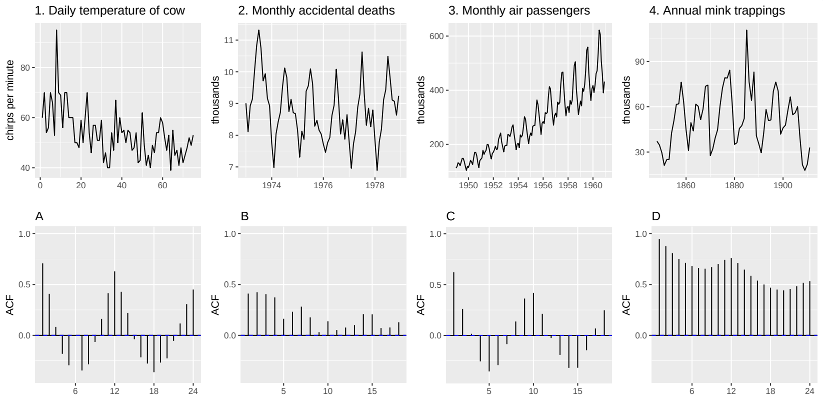


Google stock price

```r
ggAcf(goog, lag.max=100)
```



Which is which?



Outline

Lab Session 2