



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# STU33010: Forecasting

Ch1. Getting started

# Outline

1 What can we forecast?

2 Time series data

3 Some case studies

# What can we forecast?



# What can we forecast?



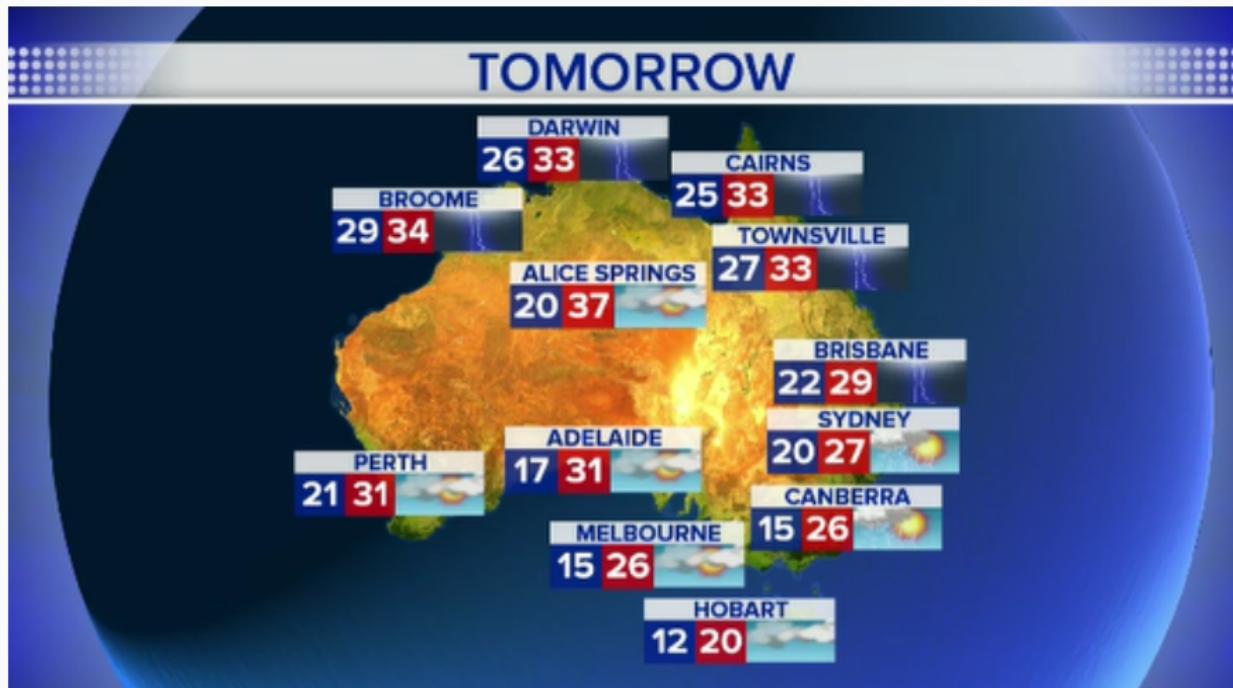
# What can we forecast?



# What can we forecast?



# What can we forecast?



# What can we forecast?



# What can we forecast?



# Which is easiest to forecast?

- 1 daily electricity demand in 3 days time
- 2 timing of next Halley's comet appearance
- 3 time of sunrise this day next year
- 4 Google stock price tomorrow
- 5 Google stock price in 6 months time
- 6 maximum temperature tomorrow
- 7 exchange rate of \$US/AUS next week
- 8 total sales of drugs in Australian pharmacies next month

# Which is easiest to forecast?

- 1 daily electricity demand in 3 days time
- 2 timing of next Halley's comet appearance
- 3 time of sunrise this day next year
- 4 Google stock price tomorrow
- 5 Google stock price in 6 months time
- 6 maximum temperature tomorrow
- 7 exchange rate of \$US/AUS next week
- 8 total sales of drugs in Australian pharmacies next month
- how do we measure “easiest”?
- what makes something easy/difficult to forecast?

# Factors affecting forecastability

Something is easier to forecast if:

- we have a good understanding of the factors that contribute to it
- there is lots of data available;
- the forecasts cannot affect the thing we are trying to forecast.
- there is relatively low natural/unexplainable random variation.
- the future is somewhat similar to the past

# Outline

**1** What can we forecast?

**2** Time series data

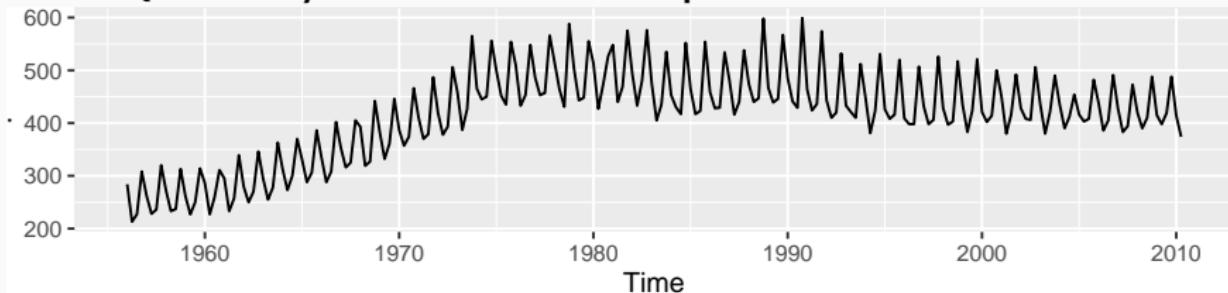
**3** Some case studies

# Time series data

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

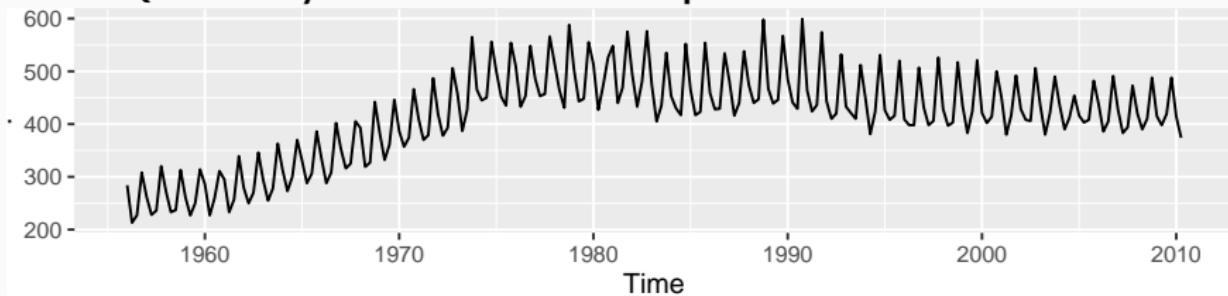
# Time series data

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



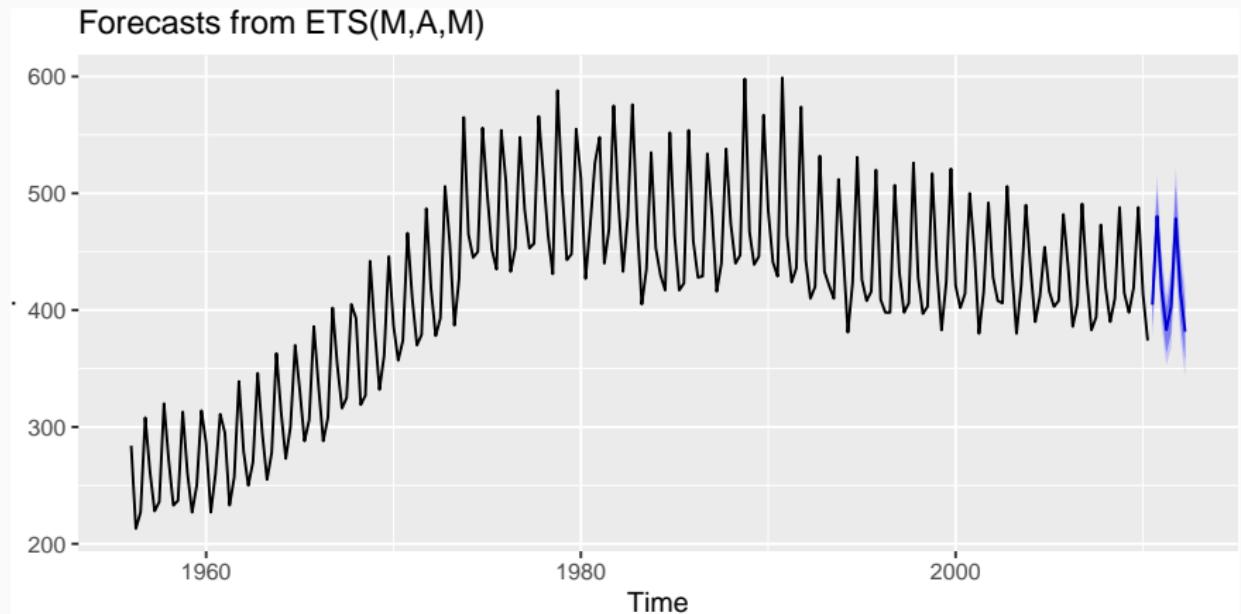
# Time series data

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



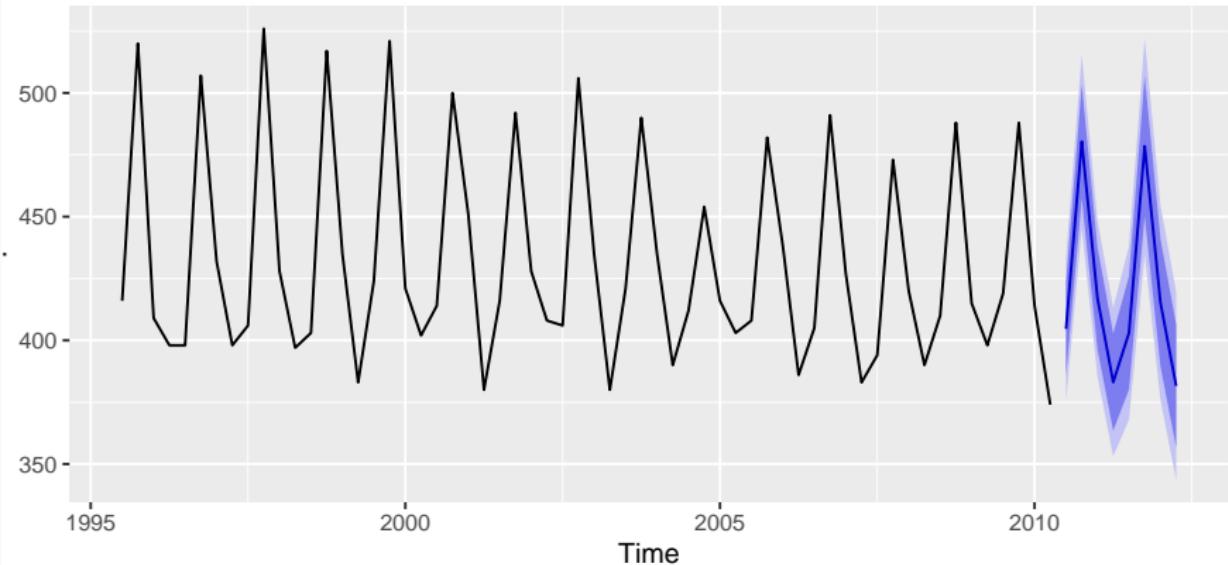
**Forecasting is estimating how the sequence of observations will continue into the future.**

# Australian beer production



# Australian beer production

Forecasts from ETS(M,A,M)



# Assignment 1: forecast the following series

- 1 Google closing stock price on 12 October 2020 based on data up to 30 September 2020
- 2 Google closing stock price on 20 April 2020 based on data up to 30 September 2020
- 3 Maximum temperature at Dublin airport on 12 October 2020 based on data up to 30 September 2020.

For each of these, give a point forecast and an 80% prediction interval.

# Assignment 1: forecast the following series

- 1 Google closing stock price on 12 October 2020 based on data up to 30 September 2020
- 2 Google closing stock price on 20 April 2020 based on data up to 30 September 2020
- 3 Maximum temperature at Dublin airport on 12 October 2020 based on data up to 30 September 2020.

For each of these, give a point forecast and an 80% prediction interval.

# Assignment 1: scoring

$Y$  = actual,  $F$  = point forecast,  $[L, U]$  = prediction interval

## Point forecasts:

$$\text{Absolute Error} = |Y - F|$$

## Prediction intervals:

$$\text{Interval Score} = (U - L) + 10(L - Y)_+ + 10(Y - U)_+$$

# Outline

1 What can we forecast?

2 Time series data

3 Some case studies

# CASE STUDY 1: Paperware company

**Problem:** Want forecasts of each of hundreds of items. Series can be stationary, trended or seasonal. They currently have a large forecasting program written in-house but it doesn't seem to produce sensible forecasts. They want me to tell them what is wrong and fix it.

## Additional information

- Their programmer has little experience in numerical computing.
- They employ no statisticians and want the program to produce forecasts automatically.



# CASE STUDY 1: Paperware company

## Methods currently used

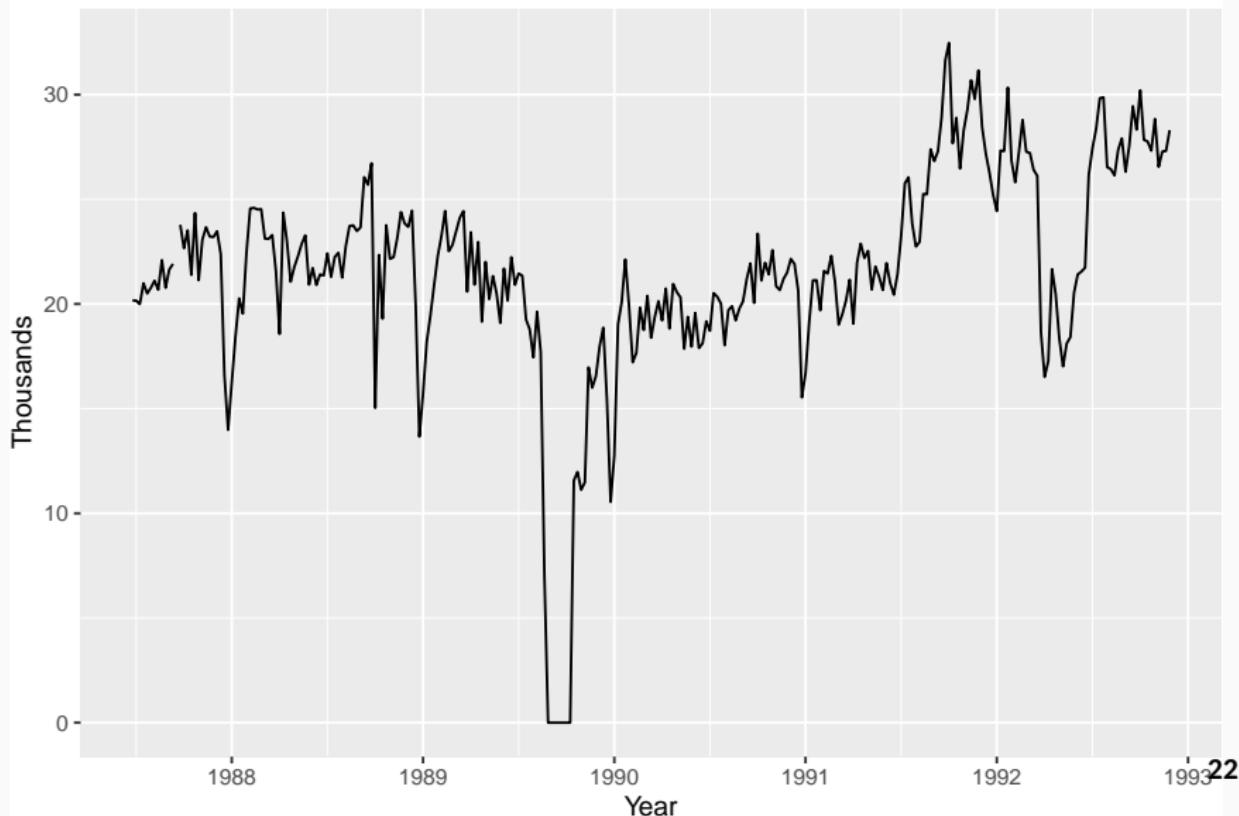
- A 12 month average
- C 6 month average
- E straight line regression over last 12 months
- G straight line regression over last 6 months
- H average slope between last year's and this year's values. (Equivalent to differencing at lag 12 and taking mean.)
- I Same as H except over 6 months.
- K I couldn't understand the explanation.

## CASE STUDY 2: Airline



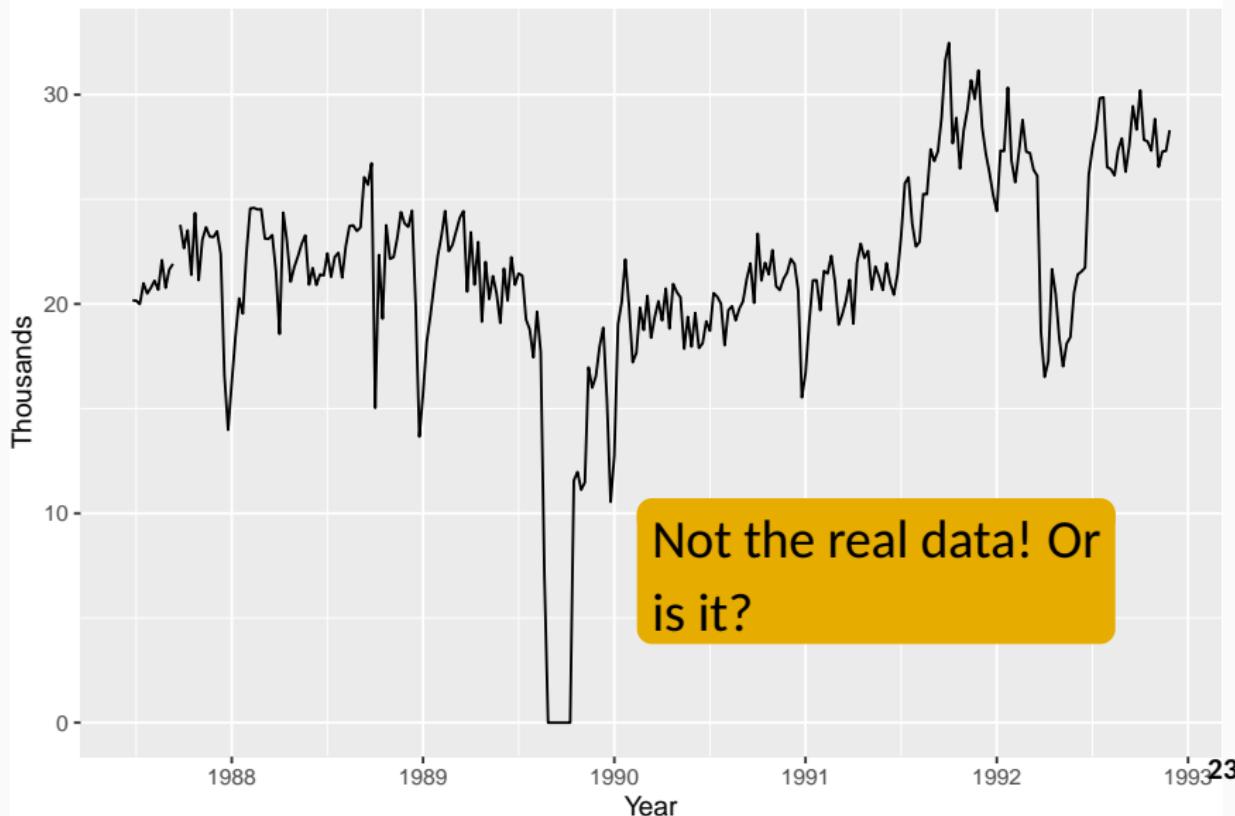
# CASE STUDY 2: Airline

Economy class passengers: Melbourne–Sydney



# CASE STUDY 2: Airline

Economy class passengers: Melbourne–Sydney



## CASE STUDY 2: Airline

**Problem:** how to forecast passenger traffic on major routes?

### Additional information

- They can provide a large amount of data on previous routes.
- Traffic is affected by school holidays, special events such as the Grand Prix, advertising campaigns, competition behaviour, etc.
- They have a highly capable team of people who are able to do most of the computing.

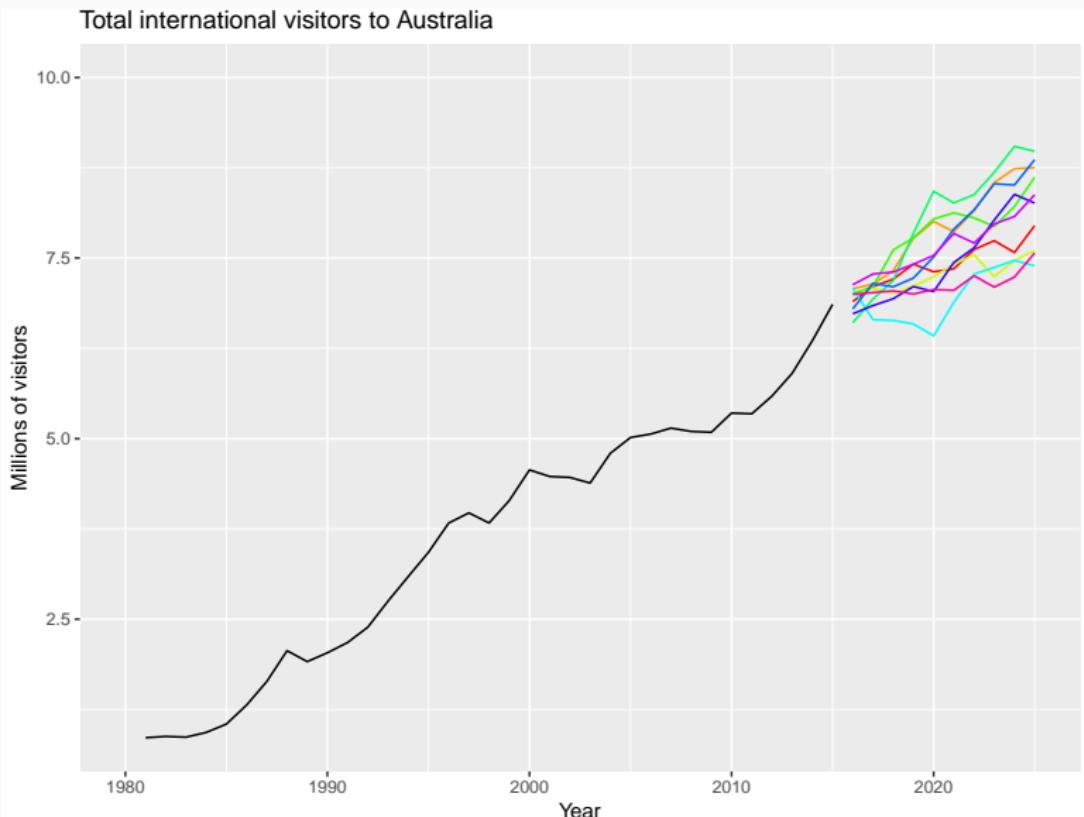
## Time plot details

- 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.

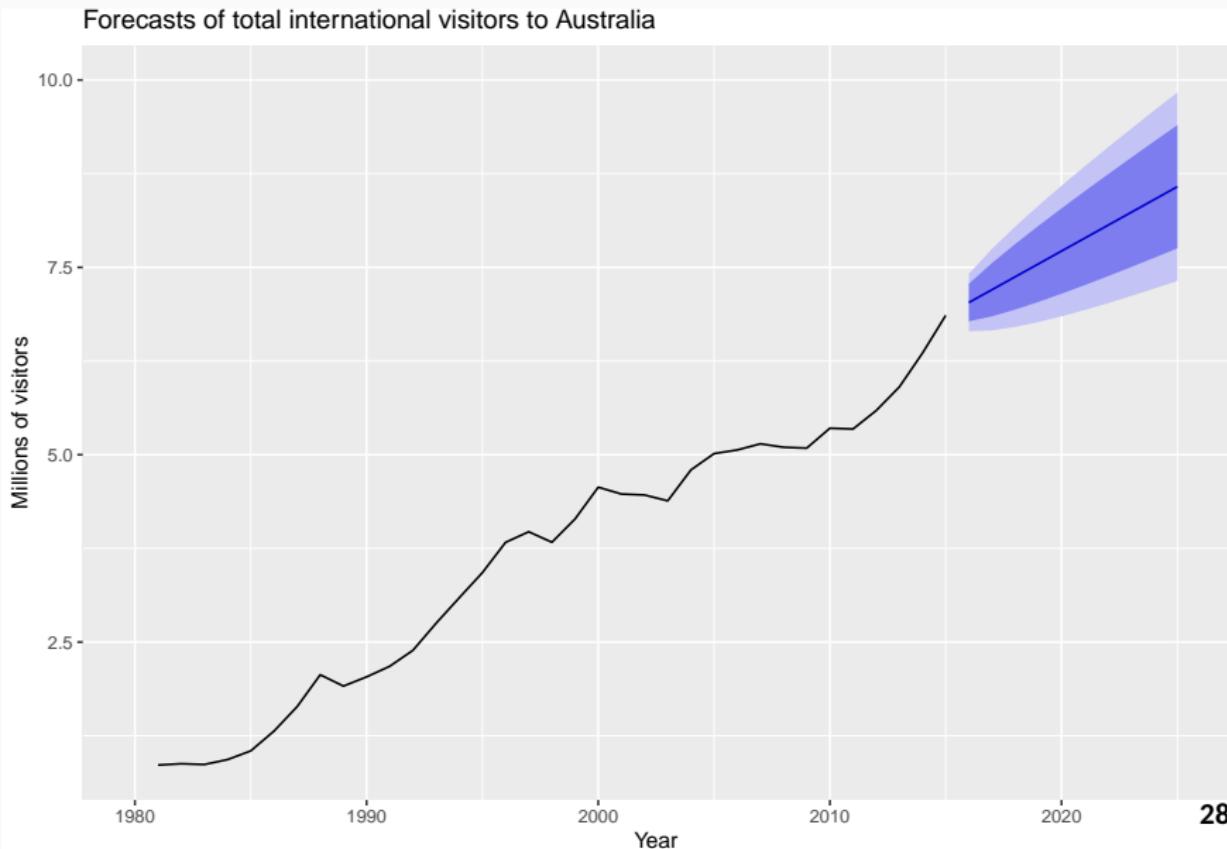
## Time plot details

- 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.  
# The statistical forecasting perspective

# Sample futures



# Forecast intervals



# Statistical forecasting

- Thing to be forecast: a random variable,  $y_t$ .
- Forecast distribution: If  $\mathcal{I}$  is all observations, then  $y_t|\mathcal{I}$  means “the random variable  $y_t$  given what we know in  $\mathcal{I}$ ”.
- The “point forecast” is the mean (or median) of  $y_t|\mathcal{I}$
- The “forecast variance” is  $\text{var}[y_t|\mathcal{I}]$
- A prediction interval or “interval forecast” is a range of values of  $y_t$  with high probability.
- With time series,  $y_{t|t-1} = y_t|\{y_1, y_2, \dots, y_{t-1}\}$ .
- $\hat{y}_{T+h|T} = E[y_{T+h}|y_1, \dots, y_T]$  (an  $h$ -step forecast taking account of all observations up to time  $T$ ).



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# STU33010: Forecasting

Ch2. Time series graphics

Alessio Benavoli

# Outline

- 1 Time series in R
- 2 Time plots
- 3 Seasonal plots
- 4 Seasonal or cyclic?
- 5 Lag plots and autocorrelation
- 6 White noise

# 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

Quarterly

Monthly

Daily

Weekly

Hourly

Half-hourly

# ts objects and ts function

**ts(data, frequency, start)**

Type of data	frequency	start example
Annual	1	
Quarterly	1	
Monthly	1	
Daily	1	
Weekly	1	
Hourly	1	
Half-hourly	1	

# ts objects and ts function

**ts(data, frequency, start)**

Type of data	frequency	start	example
Annual	1	1995	
Quarterly			
Monthly			
Daily			
Weekly			
Hourly			
Half-hourly			

# ts objects and ts function

**ts(data, frequency, start)**

Type of data	frequency	start example
Annual	1	1995
Quarterly	4	
Monthly		
Daily		
Weekly		
Hourly		
Half-hourly		

# ts objects and ts function

**ts(data, frequency, start)**

Type of data	frequency	start example
Annual	1	1995
Quarterly	4	c(1995,2)
Monthly		
Daily		
Weekly		
Hourly		
Half-hourly		

# 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	
Daily		
Weekly		
Hourly		
Half-hourly		

# 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		
Weekly		
Hourly		
Half-hourly		

# 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	
Weekly		
Hourly		
Half-hourly		

# 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		
Hourly		
Half-hourly		

# 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	
Hourly		
Half-hourly		

# 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		
Half-hourly		

# 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	
Half-hourly		

# 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		

# 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	

# 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

# Australian GDP

```
ausgdp <- ts(x, frequency=4, start=c(1971,3))
```

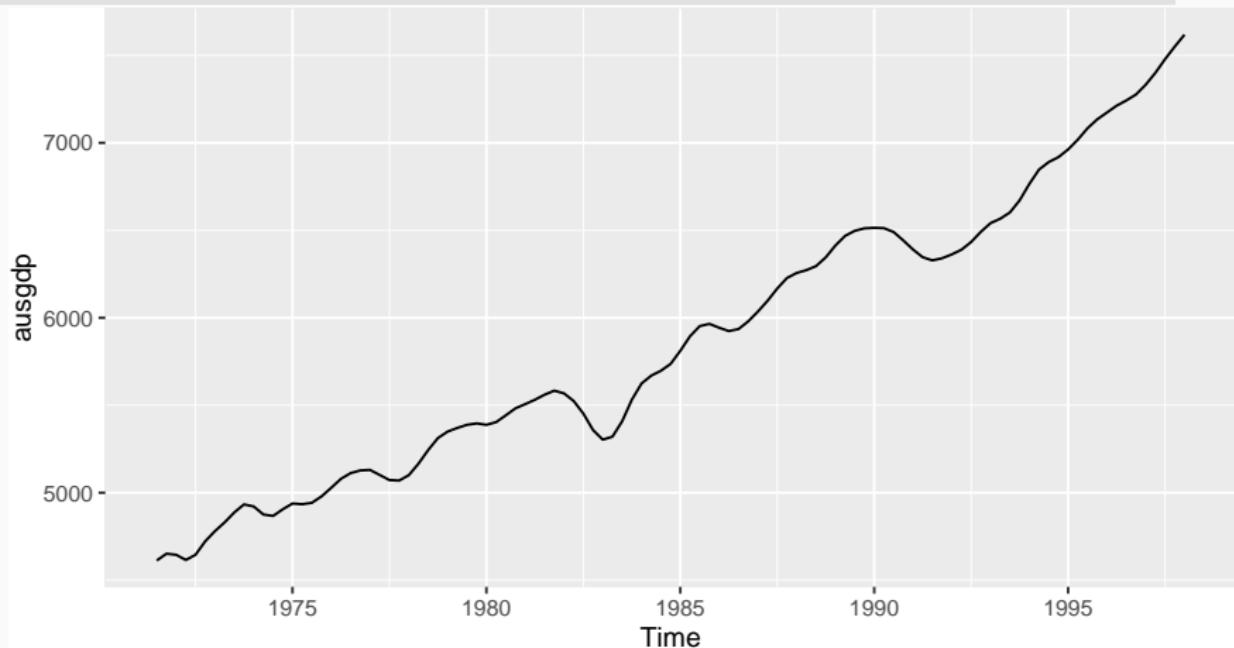
- Class: “ts”
- Print and plotting methods available.

```
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
## 1977 5130 5101 5072 5069
## 1978 5100 5166 5244 5312
## 1979 5210 5270 5350 5430
```

# Australian GDP

```
autoplots(ausgdp)
```



# Residential electricity sales

```
elecsales
```

```
## Time Series:  
## Start = 1989  
## End = 2008  
## Frequency = 1  
## [1] 2354.34 2379.71 2318.52 2468.99 2386.09  
## [6] 2569.47 2575.72 2762.72 2844.50 3000.70  
## [11] 3108.10 3357.50 3075.70 3180.60 3221.60  
## [16] 3176.20 3430.60 3527.48 3637.89 3655.00
```

# Class package

```
> library(fpp2)
```

# Class package

```
> library(fpp2)
```

This loads:

- some data for use in examples and exercises

# Class package

```
> library(fpp2)
```

This loads:

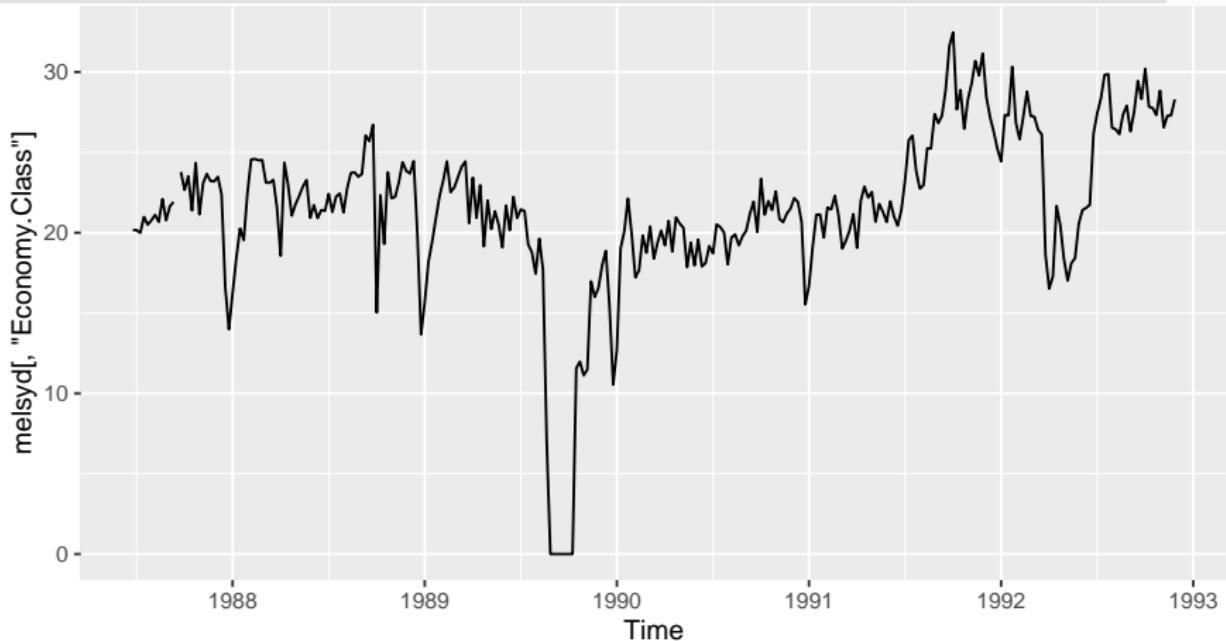
- some data for use in examples and exercises
- **forecast** package (for forecasting functions)
- **ggplot2** package (for graphics functions)
- **fma** package (for lots of time series data)
- **expsmooth** package (for more time series data)

# Outline

- 1 Time series in R
- 2 Time plots
- 3 Seasonal plots
- 4 Seasonal or cyclic?
- 5 Lag plots and autocorrelation
- 6 White noise

# Time plots

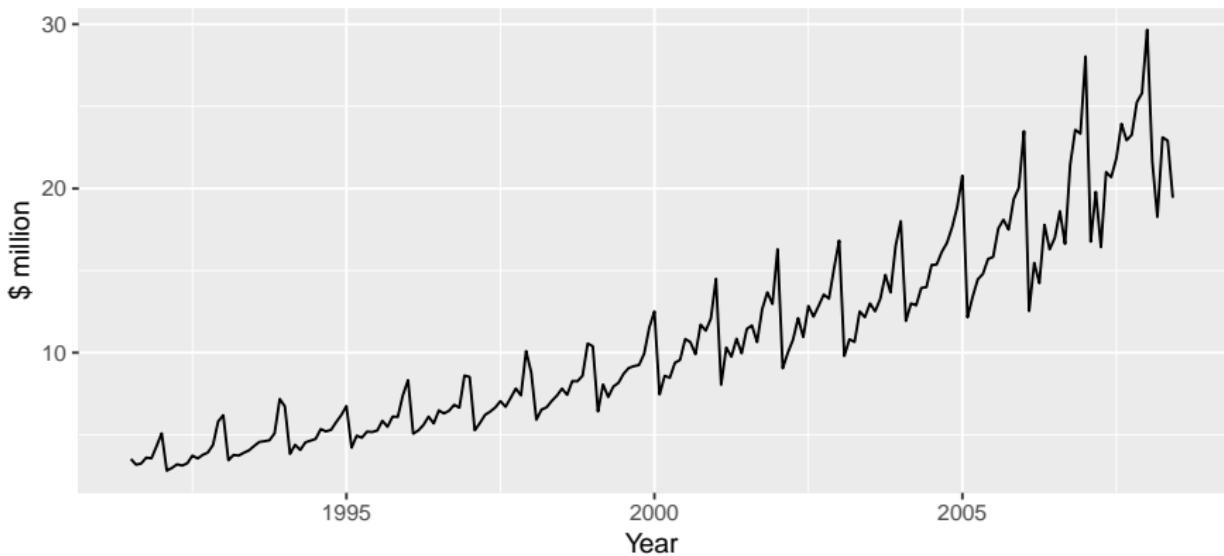
```
autoplot(melsyd[, "Economy.Class"])
```



# Time plots

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

Antidiabetic drug sales

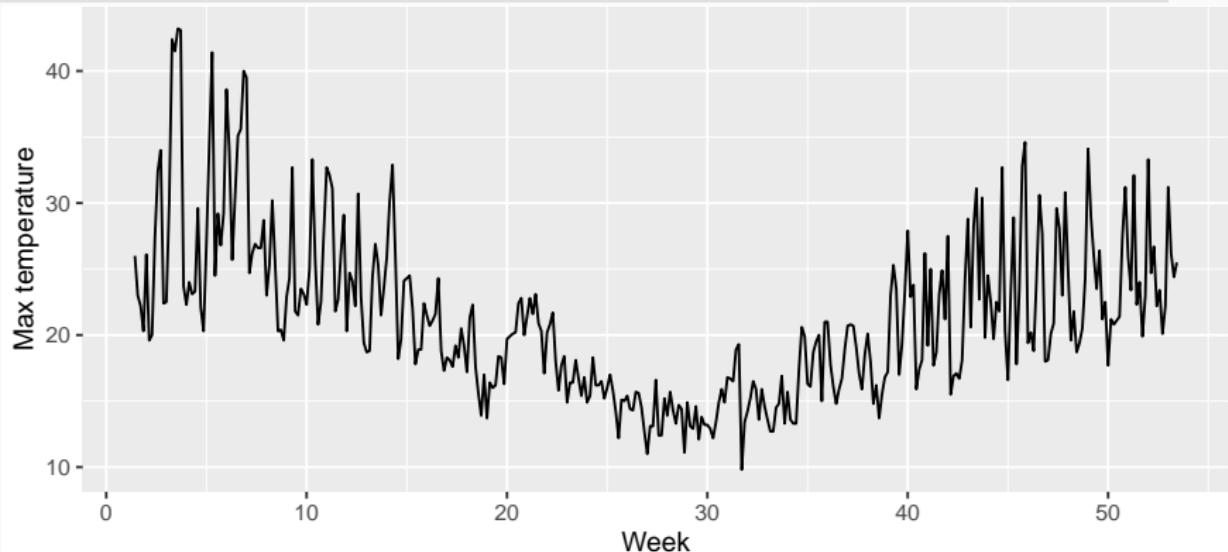


## Your turn

- Create plots of the following time series: dole, bricksq, lynx, goog
- Use `help()` to find out about the data in each series.
- For the last plot, modify the axis labels and title.

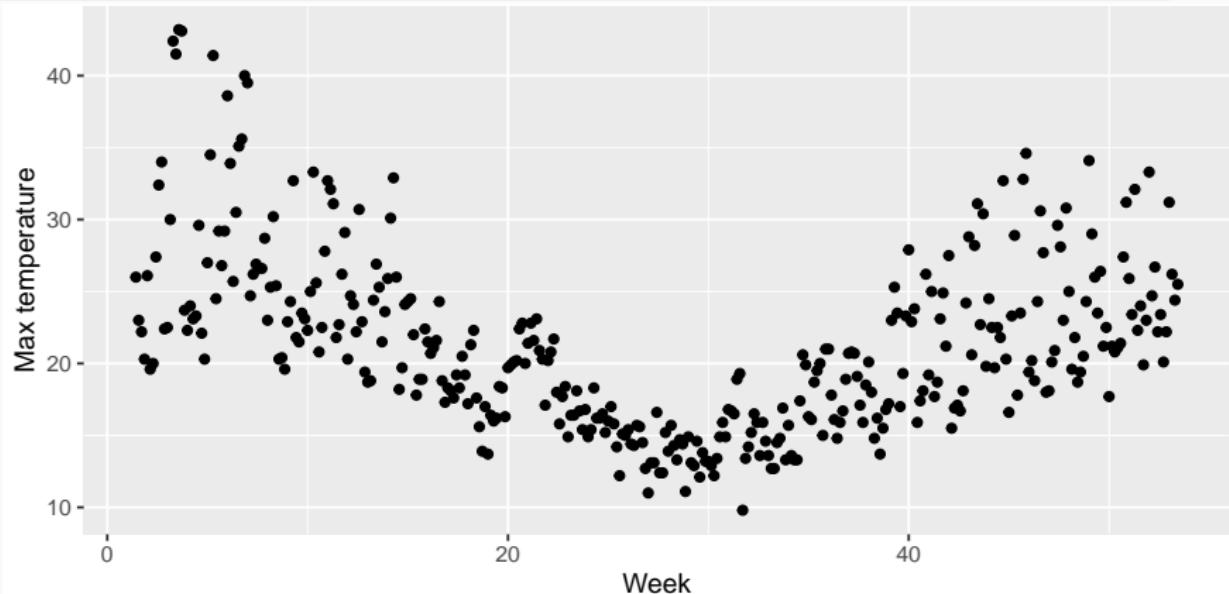
# Are time plots best?

```
autoplot(elecdaily[, "Temperature"]) +  
  xlab("Week") + ylab("Max temperature")
```

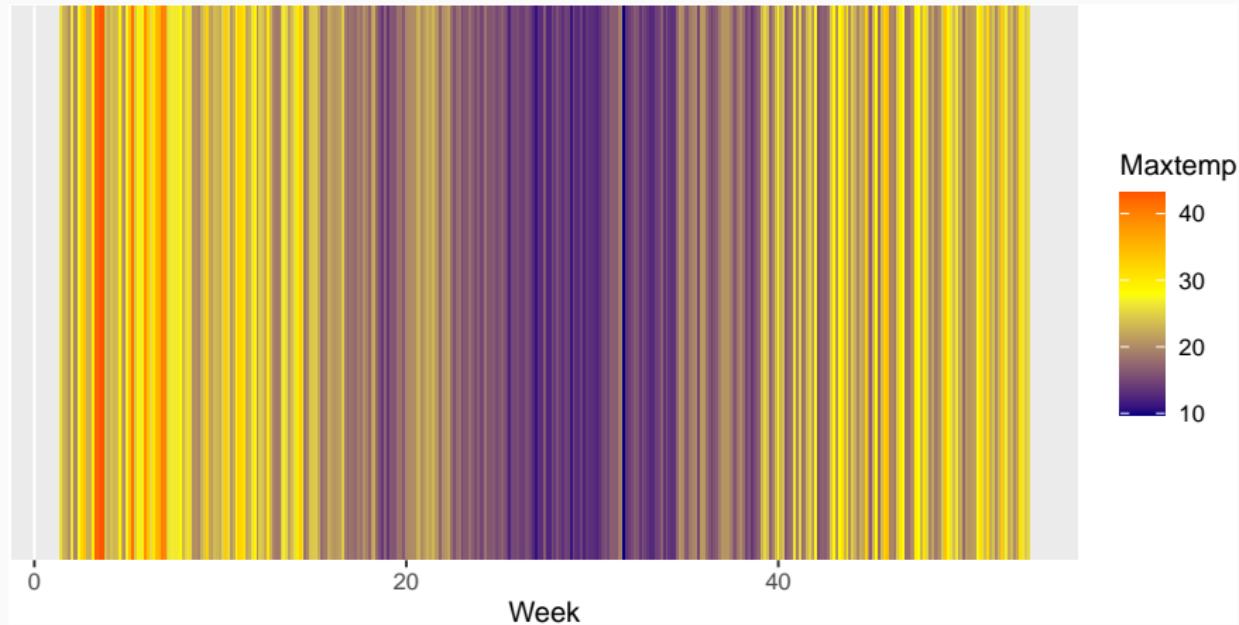


# Are time plots best?

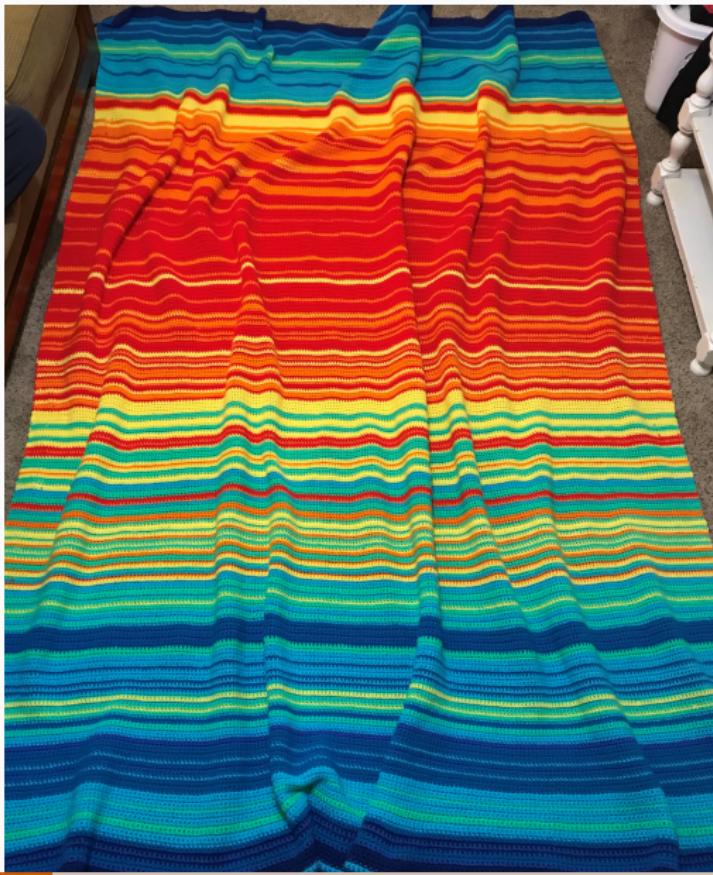
```
qplot(time(elecdaily), elecdaily[, "Temperature"]) +  
  xlab("Week") + ylab("Max temperature")
```



# Are time plots best?



# Are time plots best?

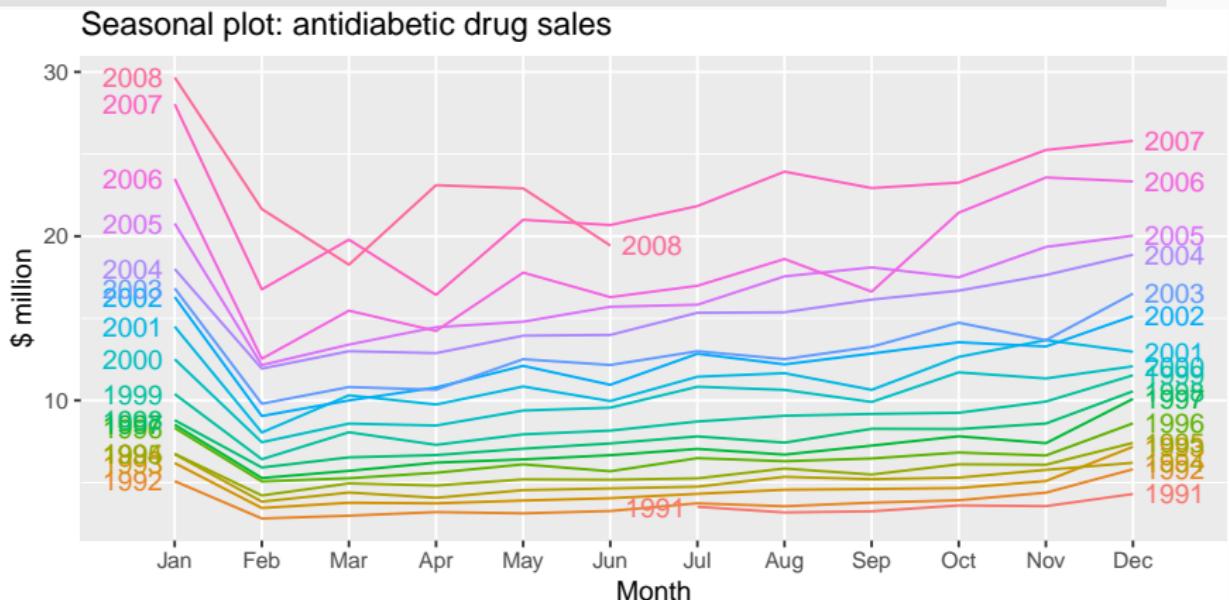


# Outline

- 1 Time series in R
- 2 Time plots
- 3 Seasonal plots
- 4 Seasonal or cyclic?
- 5 Lag plots and autocorrelation
- 6 White noise

# Seasonal plots

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

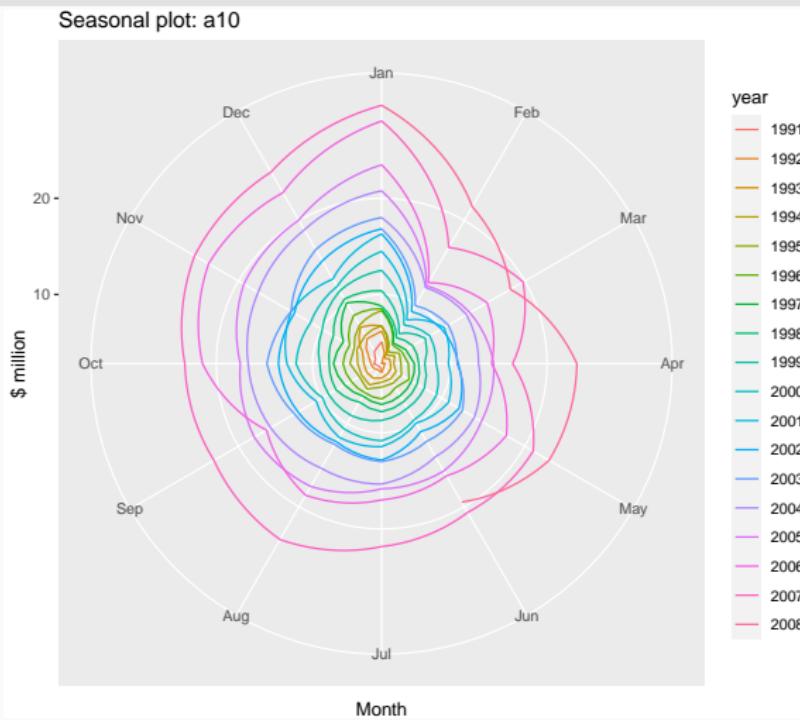


## Seasonal plots

- Data plotted against the individual “seasons” in which the data were observed. (In this case a “season” is a month.)
- Something like a time plot except that the data from each season are overlapped.
- Enables the underlying seasonal pattern to be seen more clearly, and also allows any substantial departures from the seasonal pattern to be easily identified.
- In R: `ggseasonplot()`

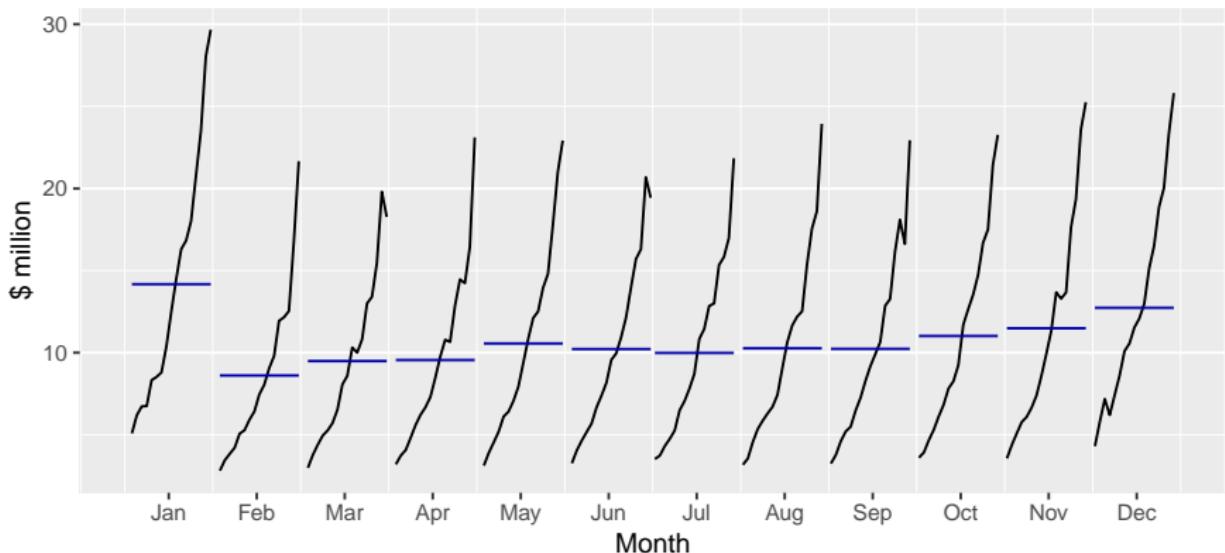
# Seasonal polar plots

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



# Seasonal subseries plots

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

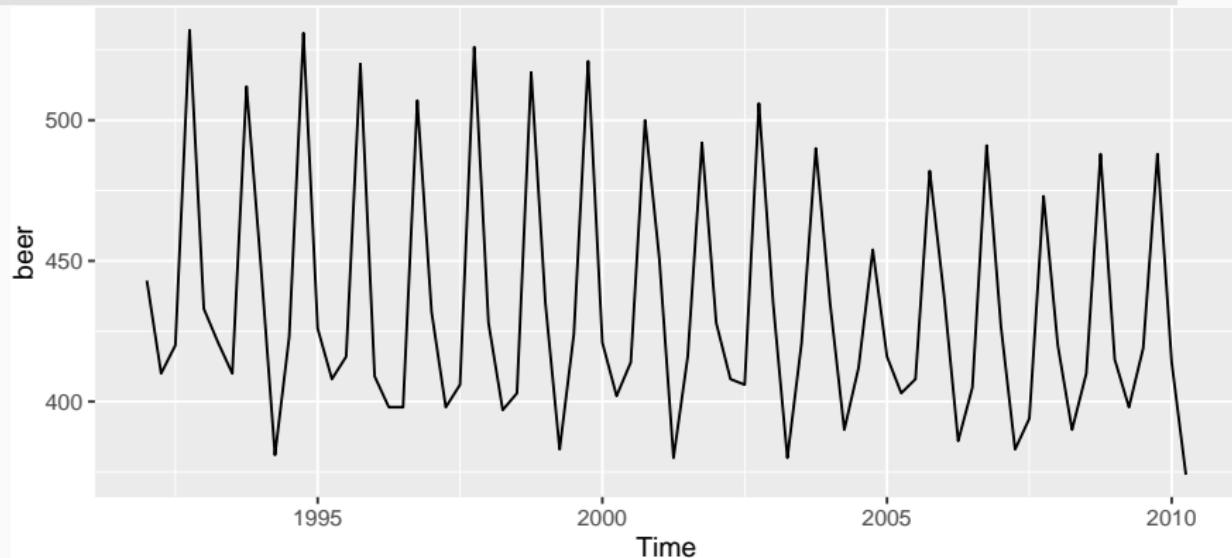


## Seasonal subseries plots

- Data for each season collected together in time plot as separate time series.
- Enables the underlying seasonal pattern to be seen clearly, and changes in seasonality over time to be visualized.
- In R: `ggsu`bsseriesplot()

# Quarterly Australian Beer Production

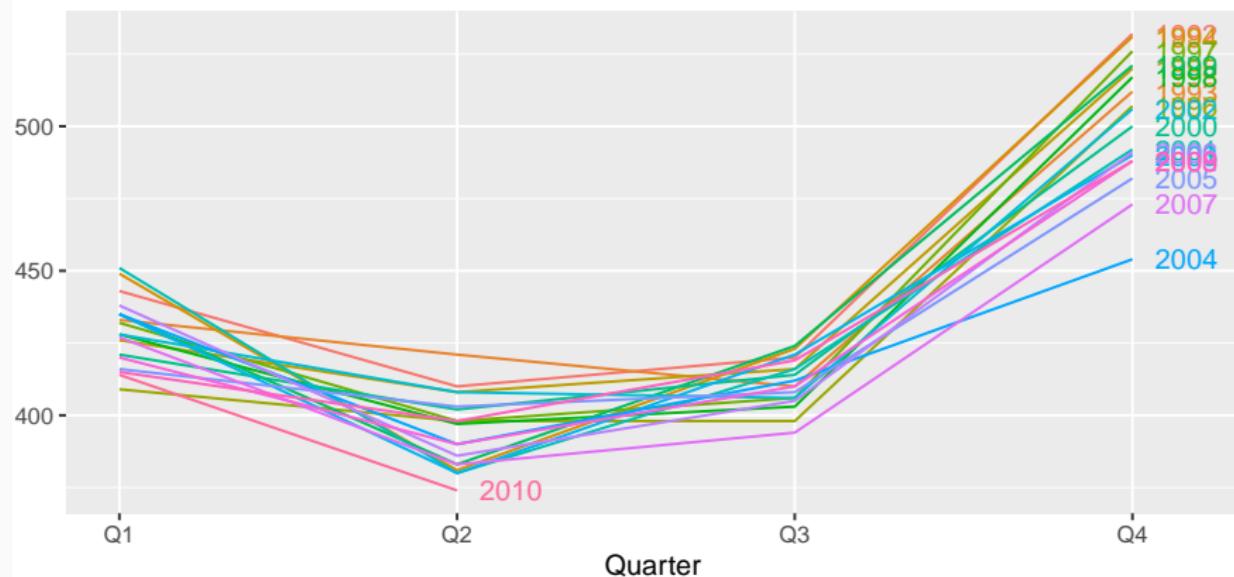
```
beer <- window(ausbeer,start=1992)  
autoplot(beer)
```



# Quarterly Australian Beer Production

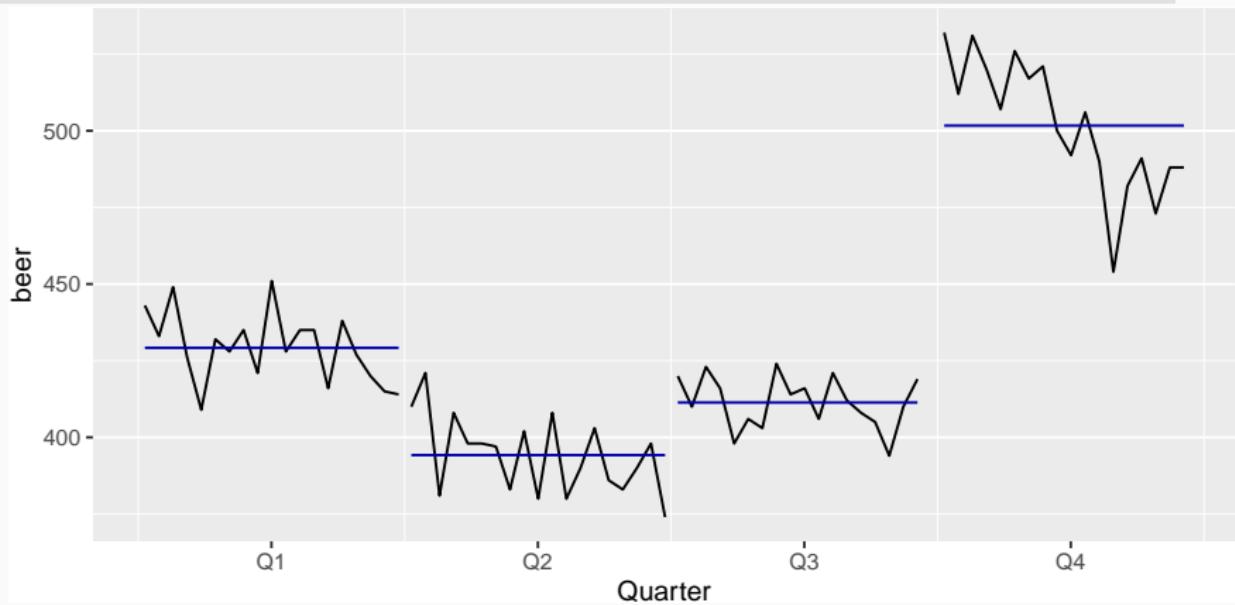
```
ggseasonplot(beer, year.labels=TRUE)
```

Seasonal plot: beer



# Quarterly Australian Beer Production

`ggsubseriesplot(beer)`



## Your turn

The arrivals data set comprises quarterly international arrivals (in thousands) to Australia from Japan, New Zealand, UK and the US.

- Use autoplot() and ggseasonplot() to compare the differences between the arrivals from these four countries.
- Can you identify any unusual observations?

# Outline

- 1 Time series in R
- 2 Time plots
- 3 Seasonal plots
- 4 Seasonal or cyclic?
- 5 Lag plots and autocorrelation
- 6 White noise

# 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 components

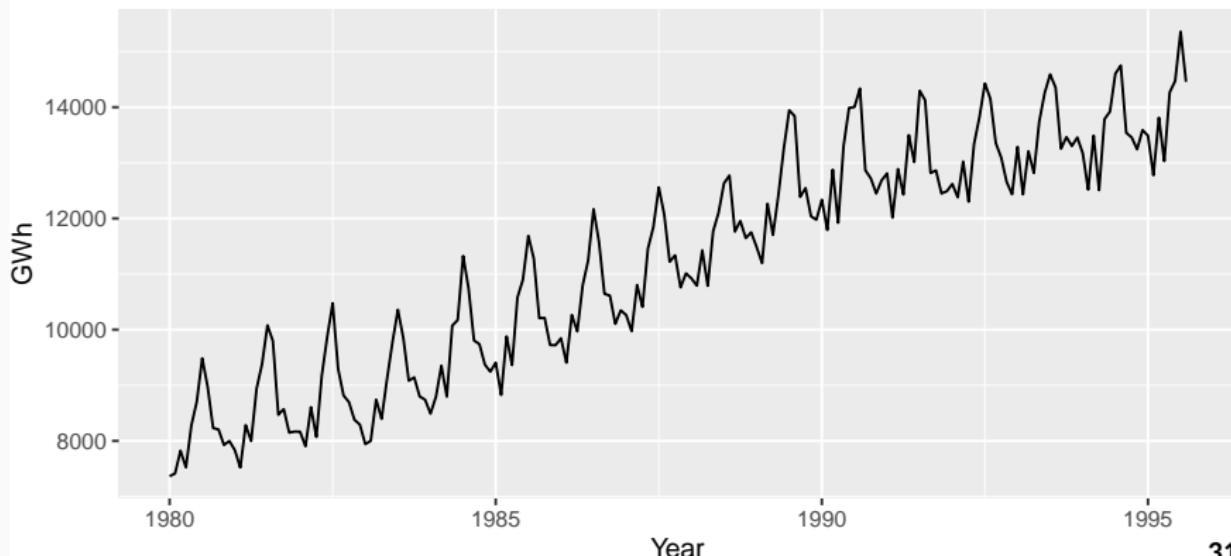
## Differences between seasonal and cyclic patterns:

- 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

# Time series patterns

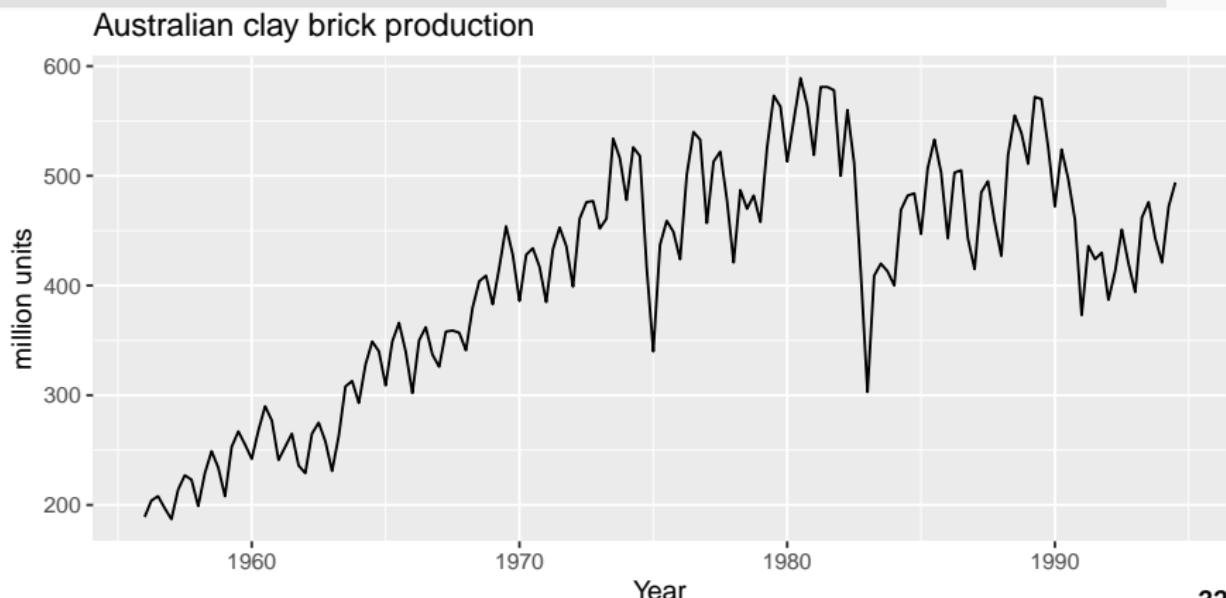
```
autoplot(window(elec, start=1980)) +  
  ggtitle("Australian electricity production") +  
  xlab("Year") + ylab("GWh")
```

Australian electricity production



# Time series patterns

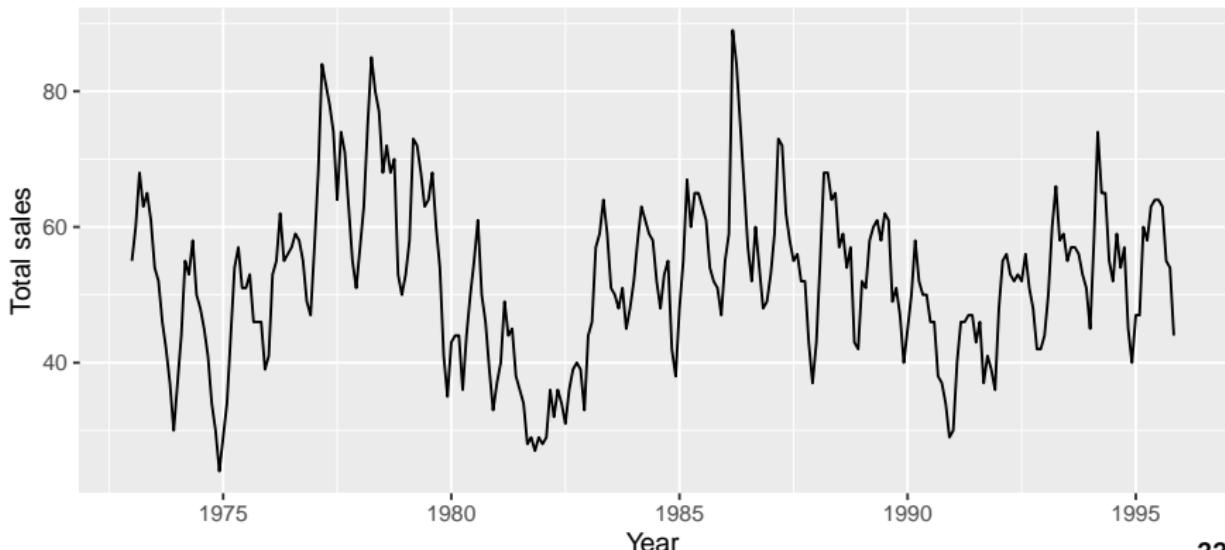
```
autoplot(bricksq) +  
  ggtitle("Australian clay brick production") +  
  xlab("Year") + ylab("million units")
```



# Time series patterns

```
autoplot(hsales) +  
  ggtitle("Sales of new one-family houses, USA") +  
  xlab("Year") + ylab("Total sales")
```

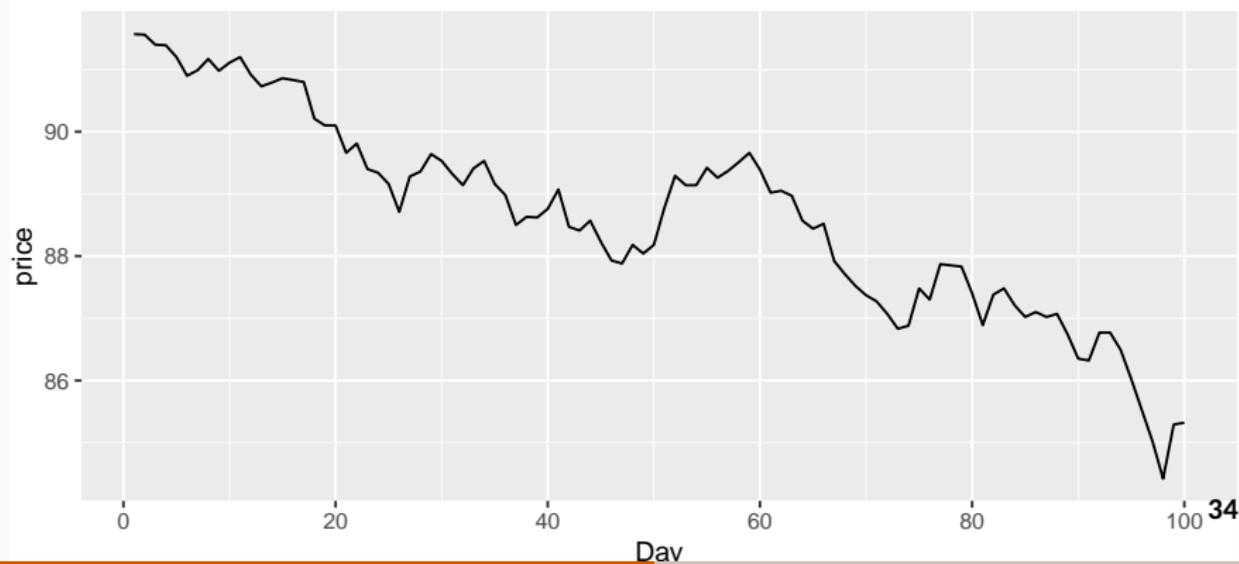
Sales of new one-family houses, USA



# Time series patterns

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

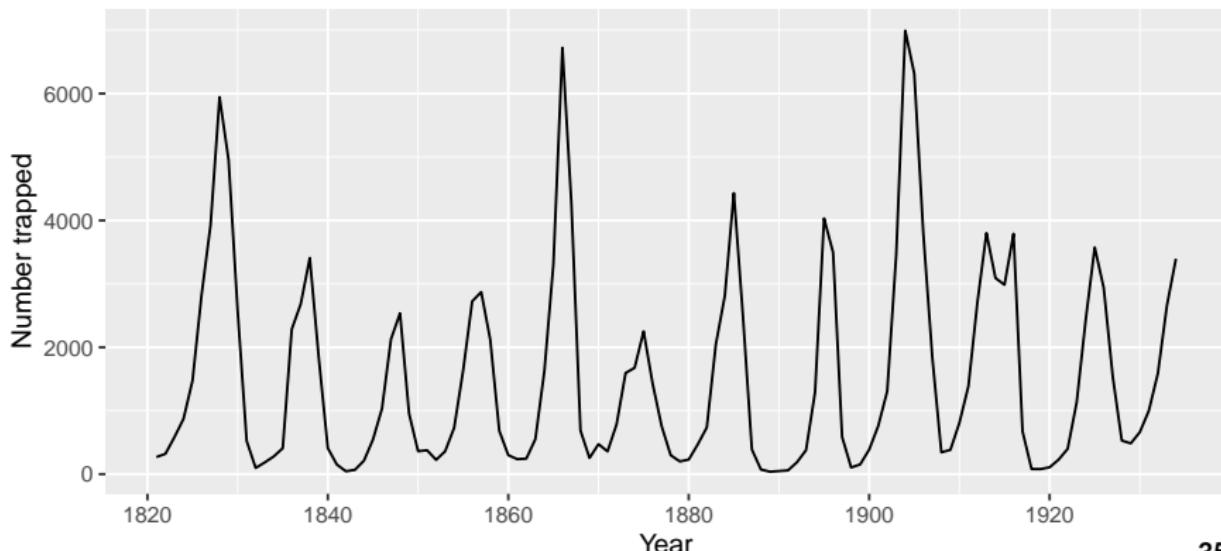
US Treasury Bill Contracts



# Time series patterns

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

Annual Canadian Lynx Trappings



# 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.

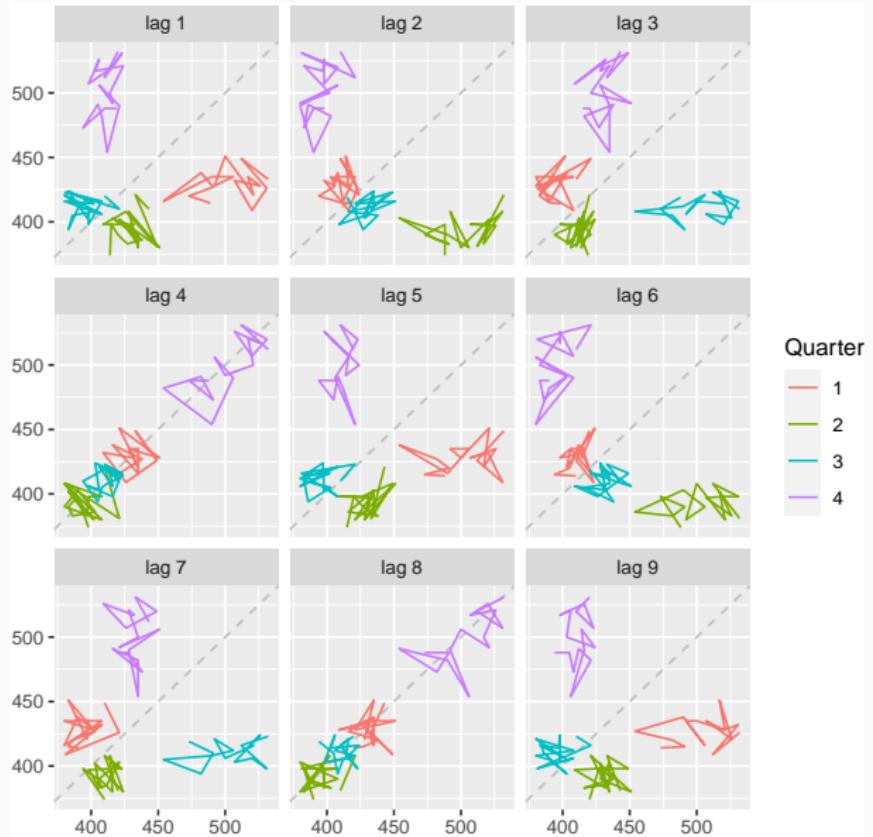
# Outline

- 1 Time series in R
- 2 Time plots
- 3 Seasonal plots
- 4 Seasonal or cyclic?
- 5 Lag plots and autocorrelation
- 6 White noise

## 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

**Covariance and correlation:** measure extent of  
**linear relationship** between two variables ( $y$  and  $X$ ).

# Autocorrelation

**Covariance and correlation:** measure extent of **linear relationship** between two variables ( $y$  and  $X$ ).  
**Autocovariance and autocorrelation:** measure linear relationship between **lagged values** of a time series  $y$ .

# Autocorrelation

**Covariance and correlation:** measure extent of linear relationship between two variables ( $y$  and  $X$ ).  
**Autocovariance and autocorrelation:** measure linear relationship between lagged values of a time series  $y$ .  
We measure the relationship between:

- $y_t$  and  $y_{t-1}$
- $y_t$  and  $y_{t-2}$
- $y_t$  and  $y_{t-3}$
- etc.

# Autocorrelation

We denote the sample autocovariance at lag  $k$  by  $c_k$  and the sample autocorrelation at lag  $k$  by  $r_k$ . Then define

$$c_k = \frac{1}{T} \sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})$$

and  $r_k = c_k/c_0$

# Autocorrelation

We denote the sample autocovariance at lag  $k$  by  $c_k$  and the sample autocorrelation at lag  $k$  by  $r_k$ . Then define

$$c_k = \frac{1}{T} \sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})$$

and  $r_k = c_k/c_0$

- $r_1$  indicates how successive values of  $y$  relate to each other
- $r_2$  indicates how  $y$  values two periods apart relate to each other
- $r_k$  is *almost* the same as the sample correlation between  $y_t$  and  $y_{t-k}$ .

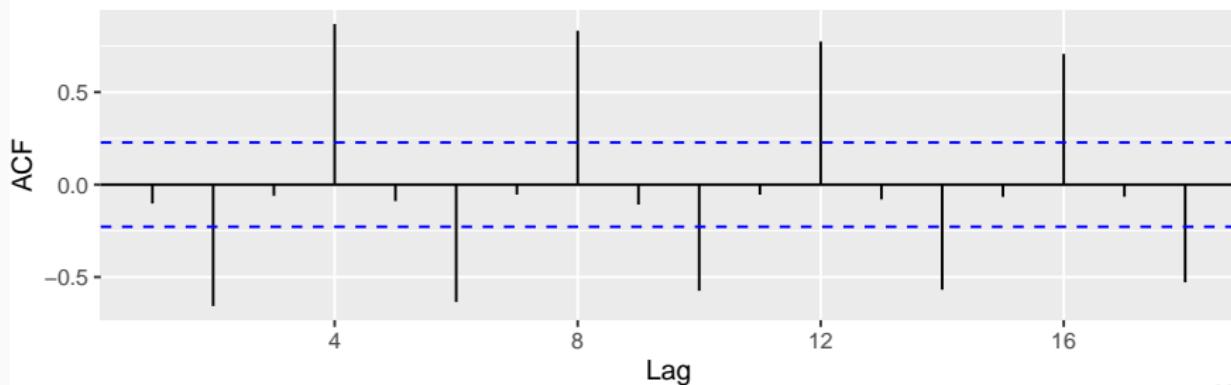
# Autocorrelation

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)

Series: beer



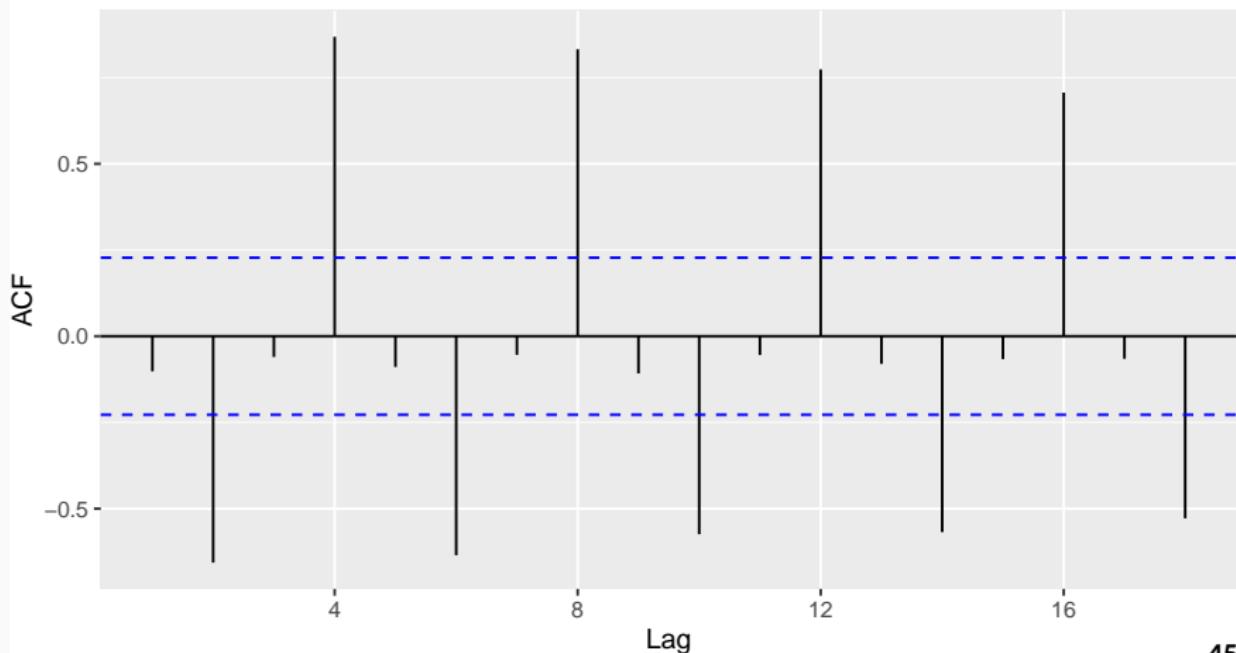
# Autocorrelation

- $r_4$  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, ..., make up the *autocorrelation* or ACF.
- The plot is known as a **correlogram**

# ACF

**ggAcf(beer)**

Series: beer

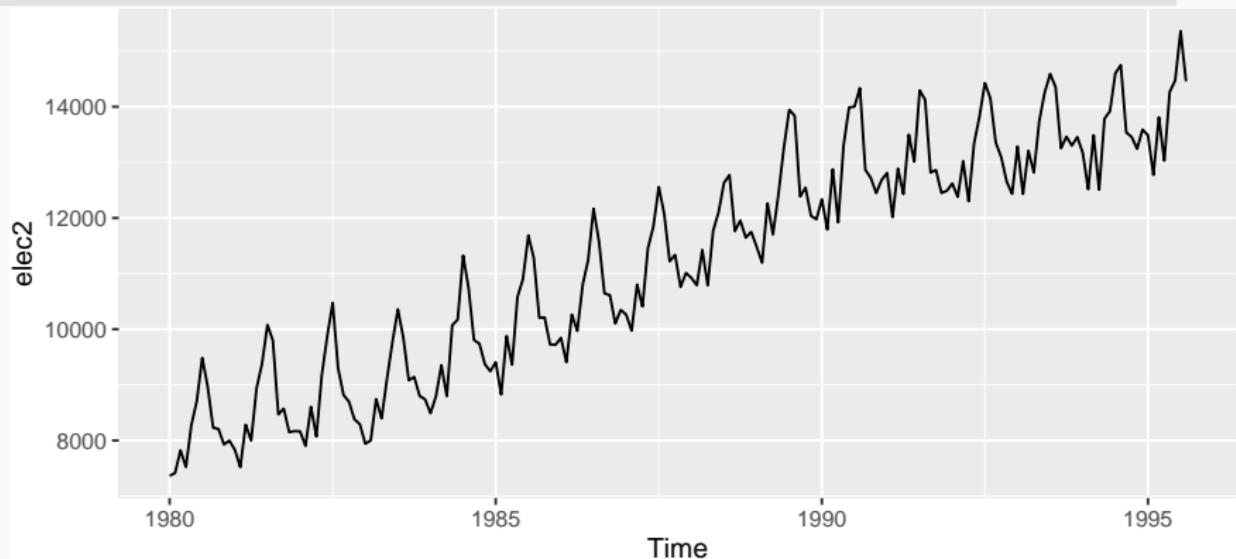


# Trend and seasonality in ACF plots

- 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

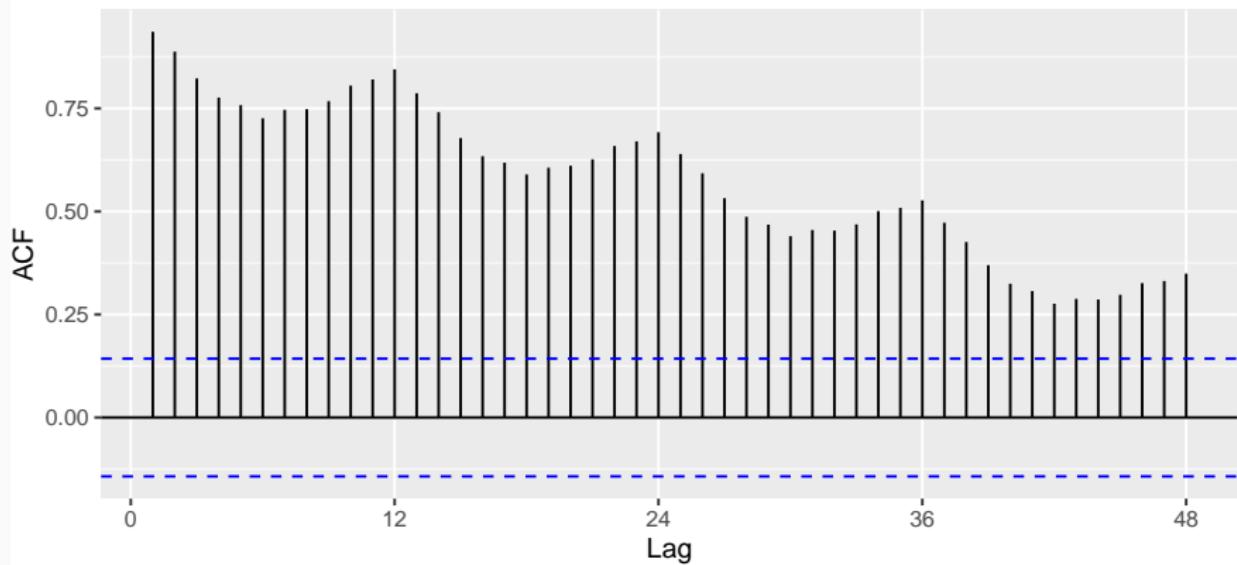
```
elec2 <- window(elec, start=1980)  
autoplot(elec2)
```



# Aus monthly electricity production

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

Series: elec2



# Aus monthly electricity production

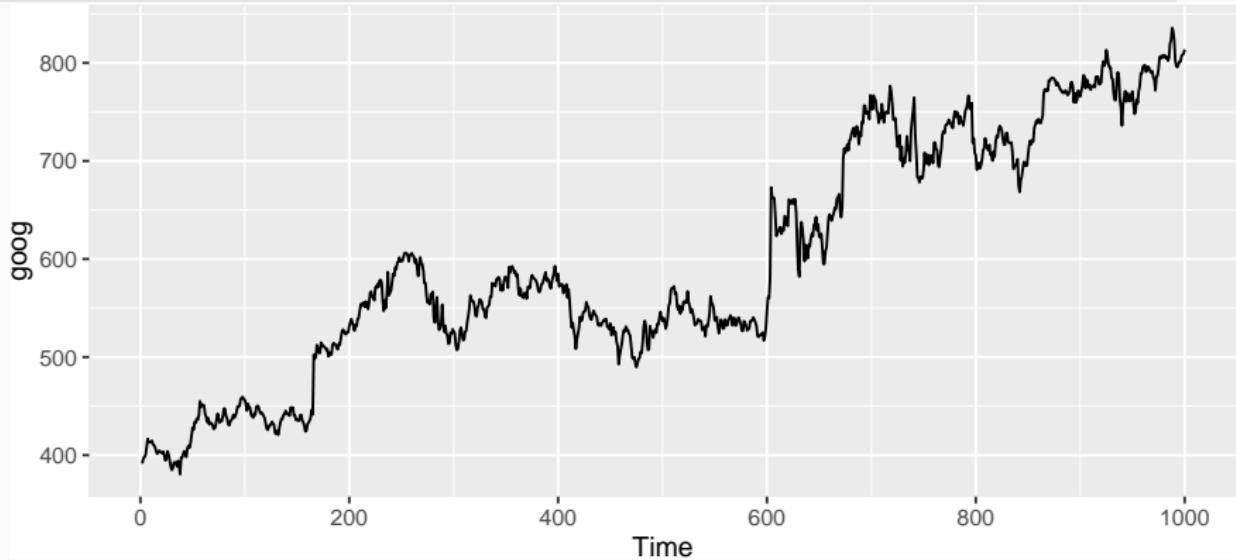
Time plot shows clear trend and seasonality.

The same features are reflected in the ACF.

- The slowly decaying ACF indicates trend.
- The ACF peaks at lags 12, 24, 36, ..., indicate seasonality of length 12.

# Google stock price

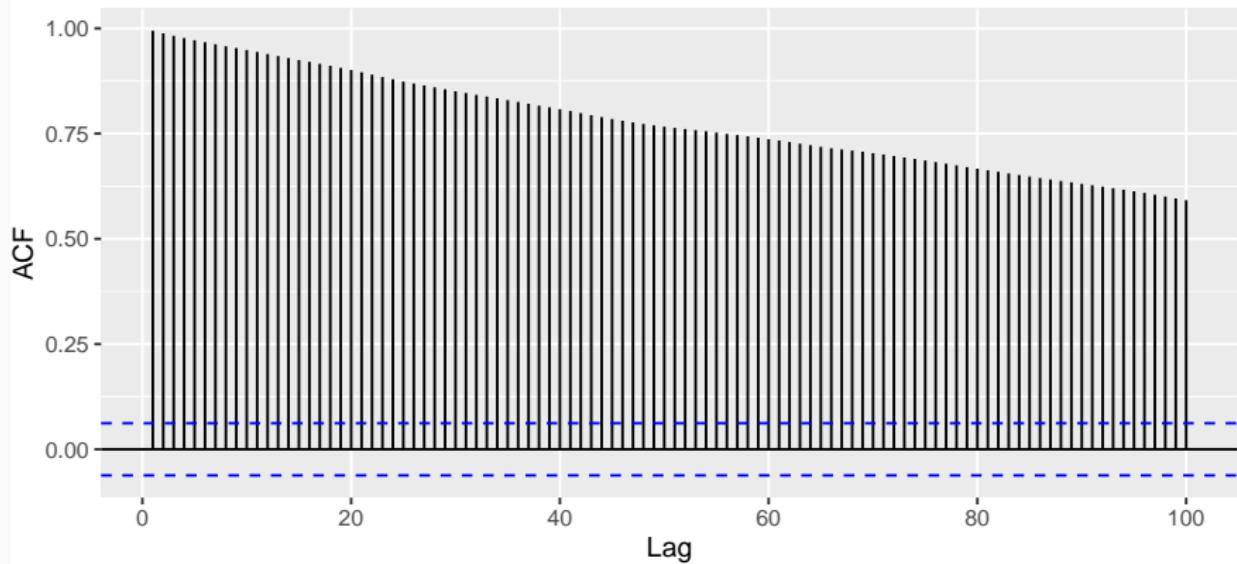
`autoplot(goog)`



# Google stock price

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

Series: goog



## Your turn

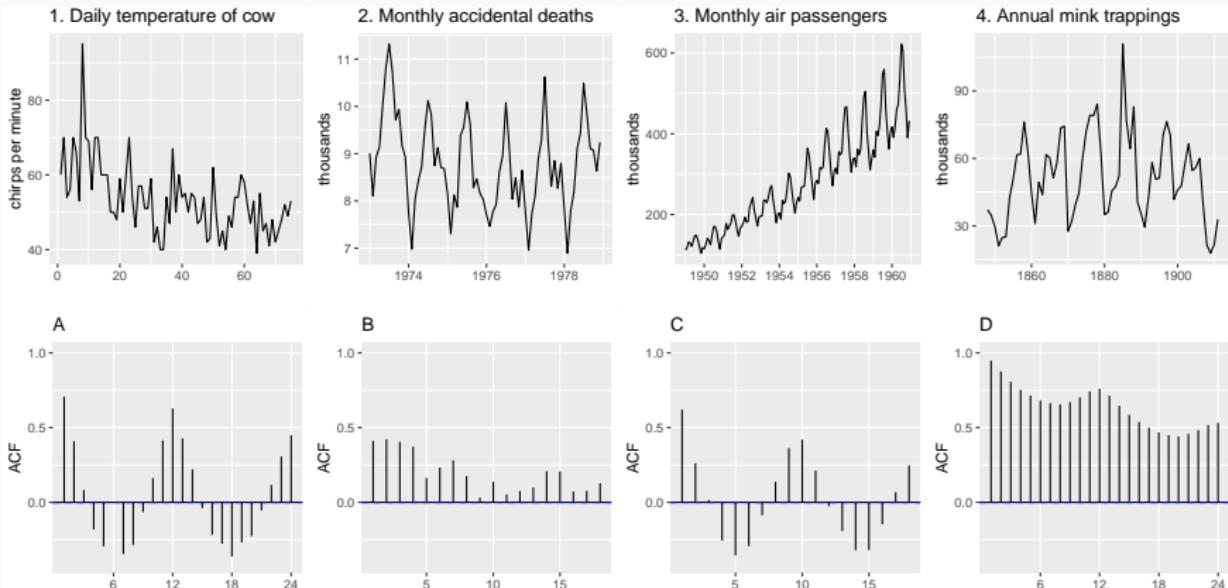
We have introduced the following graphics functions:

- gglagplot
- ggAcf

Explore the following time series using these functions. Can you spot any seasonality, cyclicity and trend? What do you learn about the series?

- hsales
- usdeaths
- bricksq
- sunspotarea
- gasoline

# Which is which?

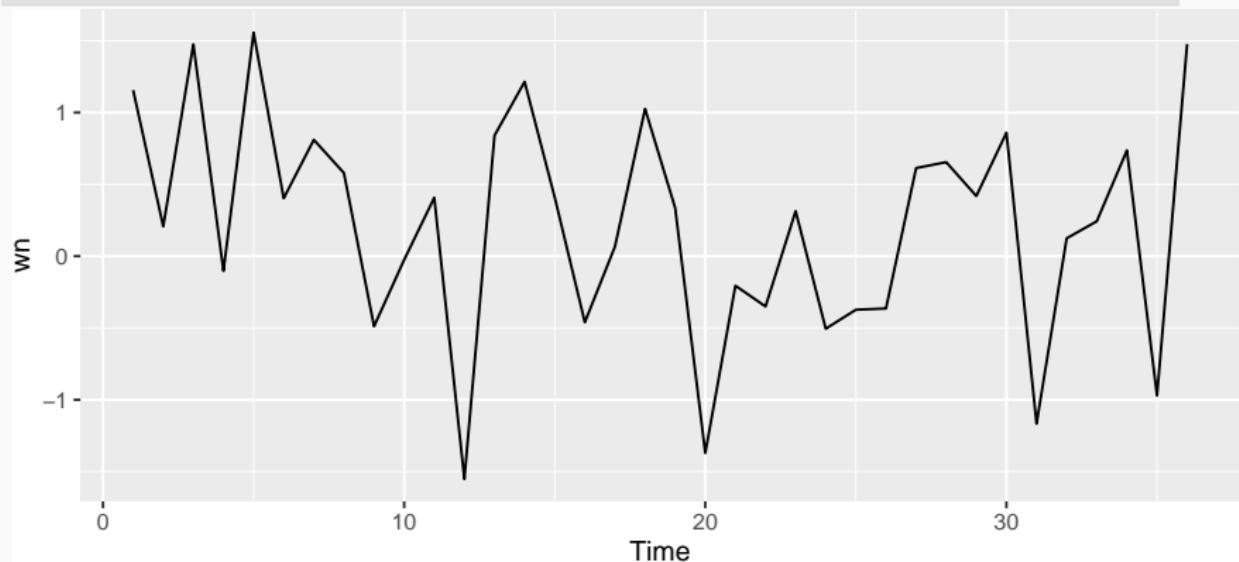


# Outline

- 1 Time series in R**
- 2 Time plots**
- 3 Seasonal plots**
- 4 Seasonal or cyclic?**
- 5 Lag plots and autocorrelation**
- 6 White noise**

# Example: White noise

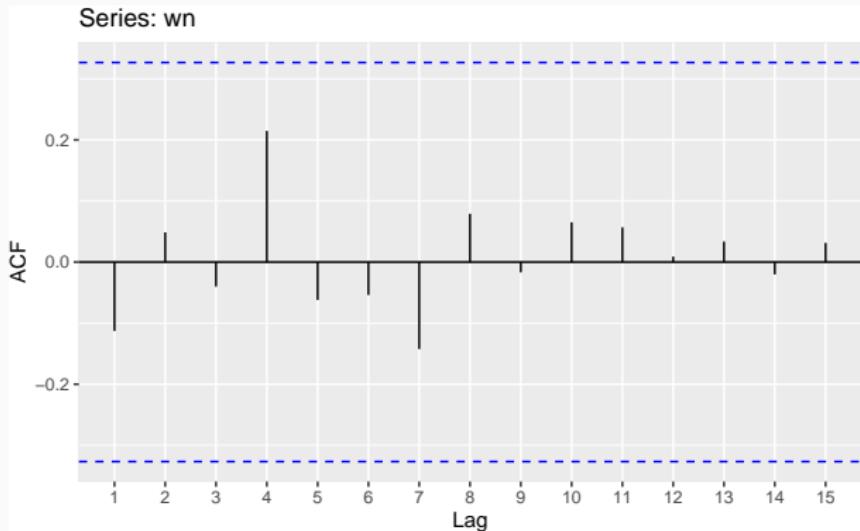
```
wn <- ts(rnorm(36))  
autoplot(wn)
```



# Example: White noise

---

$r_1$	-0.11
$r_2$	0.05
$r_3$	-0.04
$r_4$	0.21
$r_5$	-0.06
$r_6$	-0.05
$r_7$	-0.14
$r_8$	0.08
$r_9$	-0.02
$r_{10}$	0.07



Sample autocorrelations for white noise series.

We expect each autocorrelation to be close to zero.

## Sampling distribution of autocorrelations

Sampling distribution of  $r_k$  for white noise data is asymptotically  $N(0, 1/T)$ .

# Sampling distribution of autocorrelations

Sampling distribution of  $r_k$  for white noise data is asymptotically  $N(0, 1/T)$ .

- 95% of all  $r_k$  for white noise must lie within  $\pm 1.96/\sqrt{T}$ .
- If this is not the case, the series is probably not WN.
- Common to plot lines at  $\pm 1.96/\sqrt{T}$  when plotting ACF. These are the *critical values*.

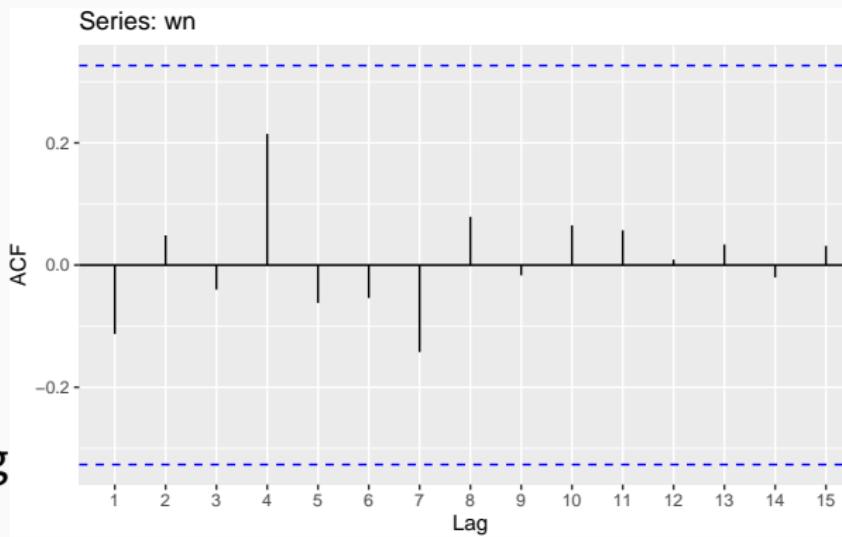
# Autocorrelation

## Example:

$T = 36$  and so critical values at

$$\pm 1.96/\sqrt{36} = \pm 0.327.$$

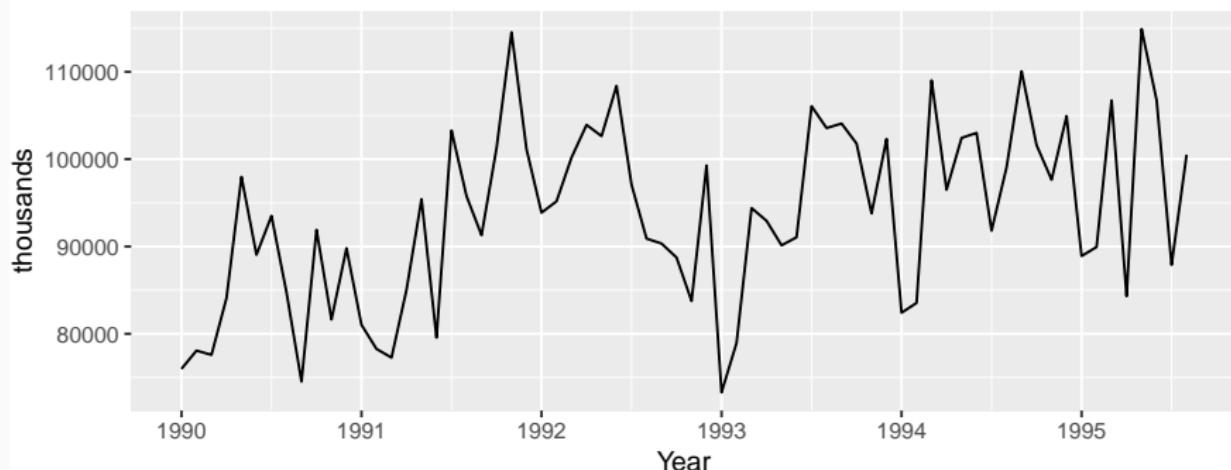
All autocorrelation coefficients lie within these limits, confirming that the data are white noise. (More precisely, the data cannot be distinguished from white noise.)



# Example: Pigs slaughtered

```
pigs2 <- window(pigs, start=1990)
autoplot(pigs2) +
  xlab("Year") + ylab("thousands") +
  ggtitle("Number of pigs slaughtered in Victoria")
```

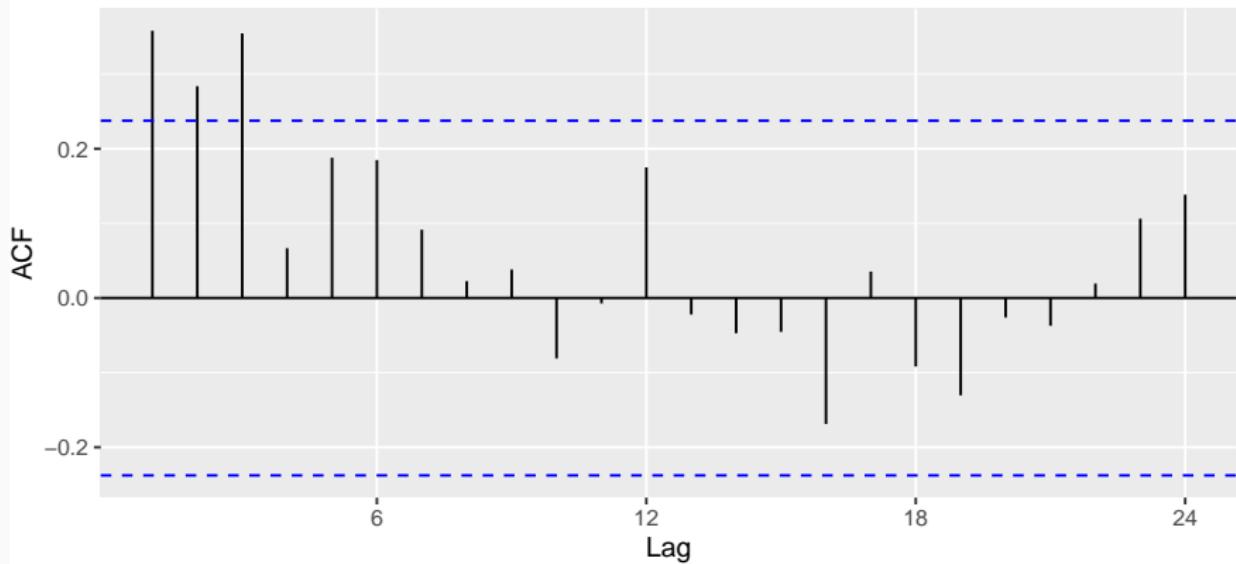
Number of pigs slaughtered in Victoria



# Example: Pigs slaughtered

`ggAcf(pigs2)`

Series: pigs2



## Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 1990 through August 1995. (Source: Australian Bureau of Statistics.)

## Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 1990 through August 1995. (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows some significant autocorrelation at lags 1, 2, and 3.
- $r_{12}$  relatively large although not significant. This may indicate some slight seasonality.

## Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 1990 through August 1995. (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows some significant autocorrelation at lags 1, 2, and 3.
- $r_{12}$  relatively large although not significant. This may indicate some slight seasonality.

These show the series is **not a white noise series**.

## Your turn

You can compute the daily changes in the Google stock price using

```
dgoog <- diff(goog)
```

Does dgoog look like white noise?



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# STU33010: Forecasting

Ch3. The forecasters' toolbox

# Outline

## 1 Some simple forecasting methods

## 2 Box-Cox transformations

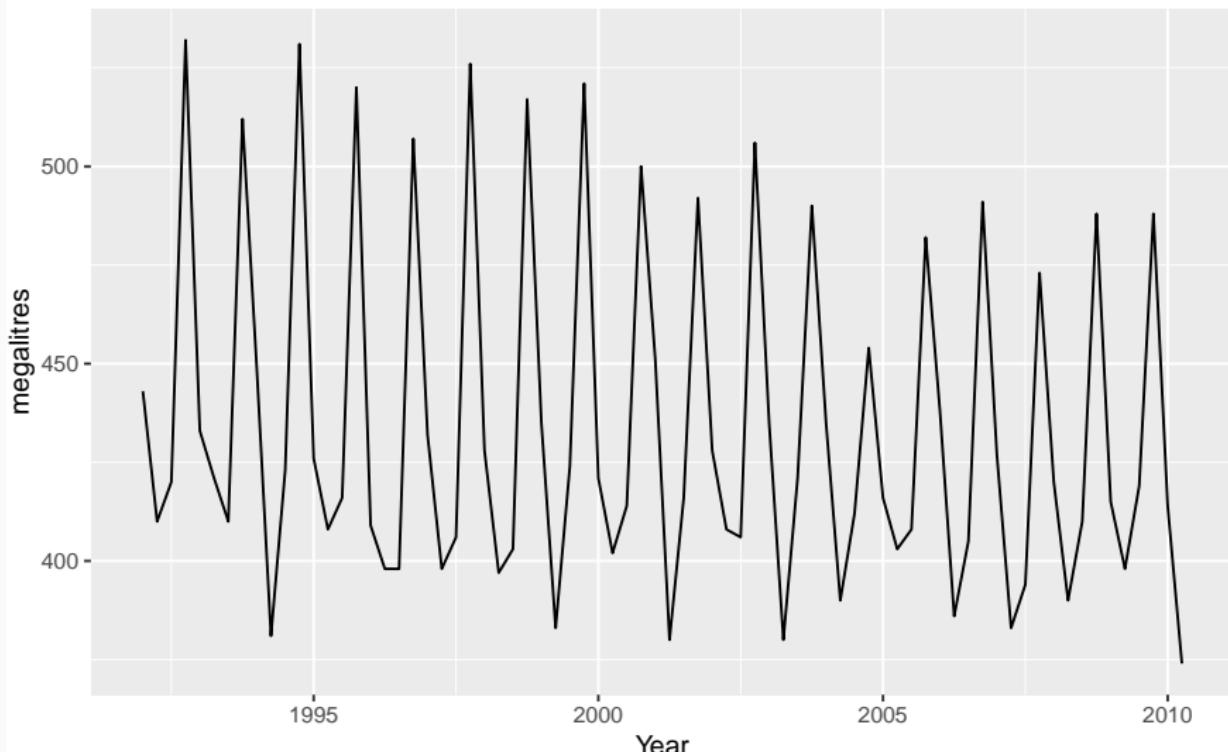
## 3 Residual diagnostics

## 4 Evaluating forecast accuracy

## 5 Prediction intervals

# Some simple forecasting methods

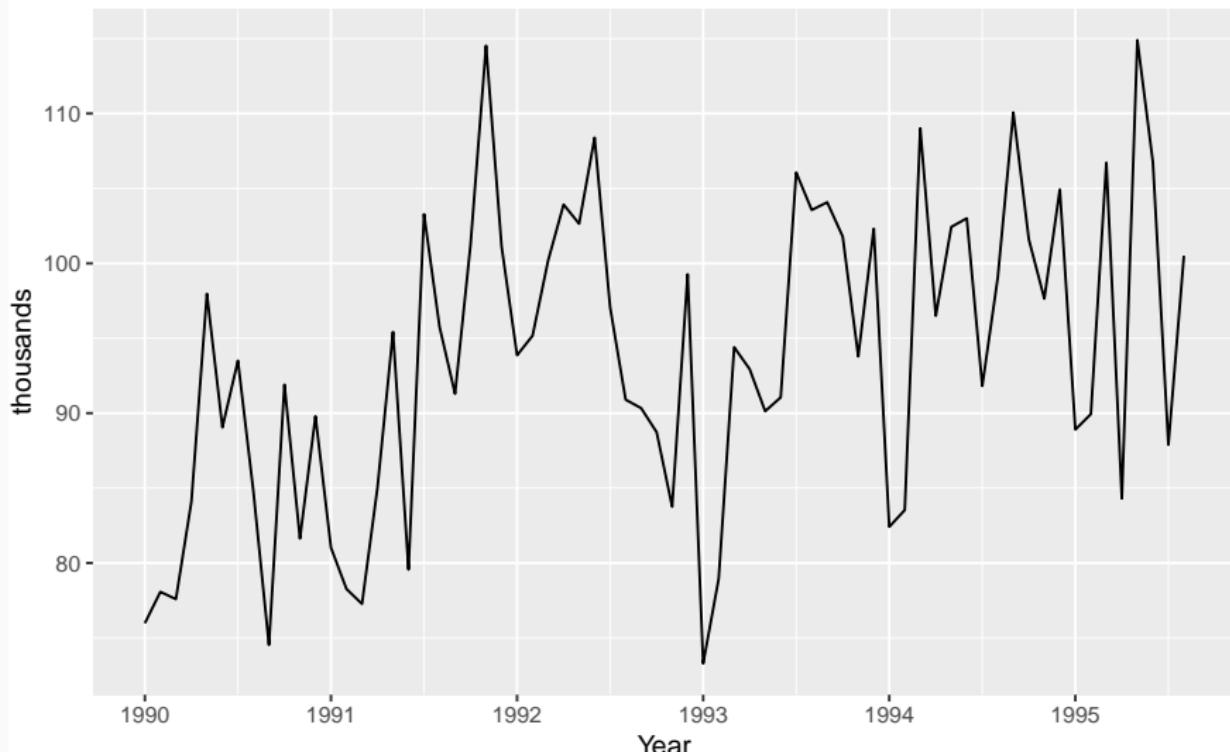
Australian quarterly beer production



How would you forecast these data?

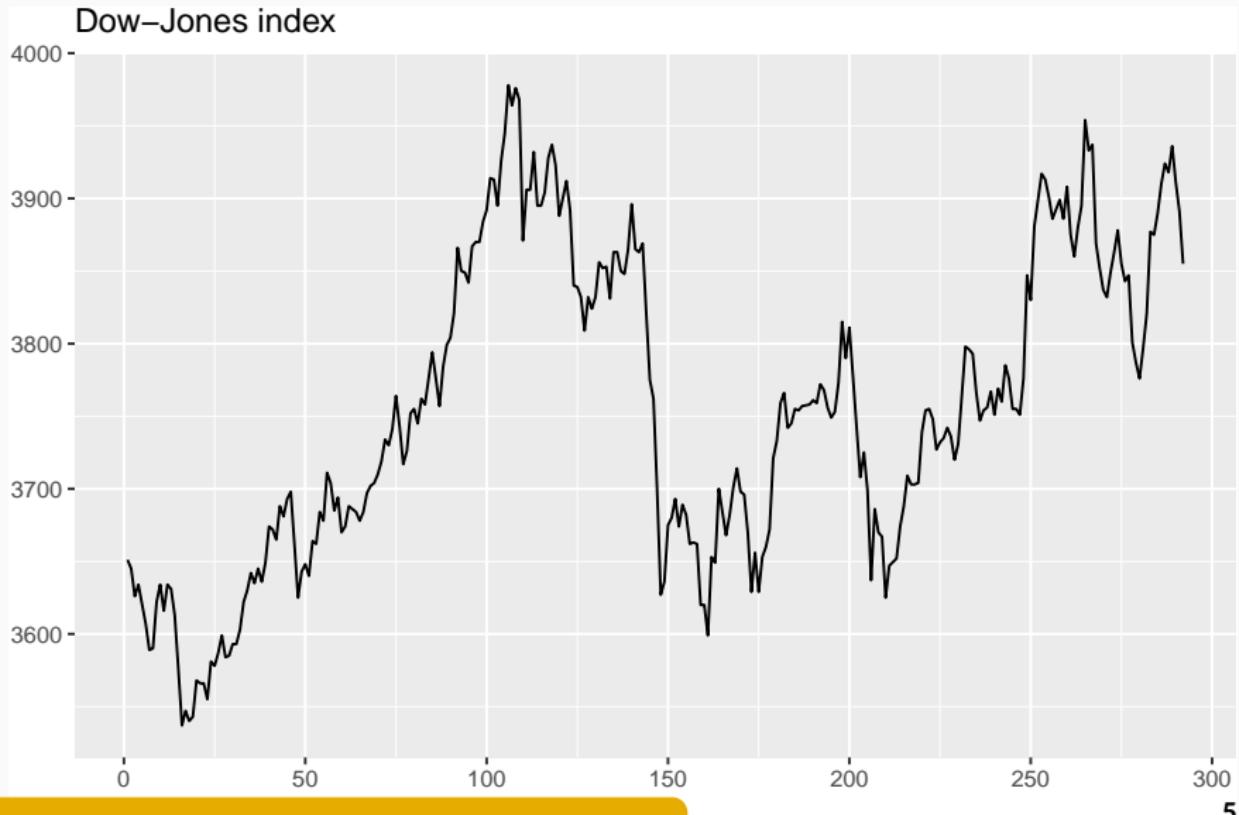
# Some simple forecasting methods

Number of pigs slaughtered in Victoria



How would you forecast these data?

# Some simple forecasting methods



How would you forecast these data?

# Some simple forecasting methods

## Average method

- Forecast of all future values is equal to mean of historical data  $\{y_1, \dots, y_T\}$ .
- Forecasts:  $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \dots + y_T)/T$

# Some simple forecasting methods

## Average method

- Forecast of all future values is equal to mean of historical data  $\{y_1, \dots, y_T\}$ .
- Forecasts:  $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \dots + y_T)/T$

## Naïve method

- Forecasts equal to last observed value.
- Forecasts:  $\hat{y}_{T+h|T} = y_T$ .
- Consequence of efficient market hypothesis.

# Some simple forecasting methods

## Average method

- Forecast of all future values is equal to mean of historical data  $\{y_1, \dots, y_T\}$ .
- Forecasts:  $\hat{y}_{T+h|T} = \bar{y} = (y_1 + \dots + y_T)/T$

## Naïve method

- Forecasts equal to last observed value.
- Forecasts:  $\hat{y}_{T+h|T} = y_T$ .
- Consequence of efficient market hypothesis.

## Seasonal naïve method

- Forecasts equal to last value from same season.
- Forecasts:  $\hat{y}_{T+h|T} = y_{T+h-m(k+1)}$ , where  $m$  = seasonal period and  $k$  is the integer part of  $(h - 1)/m$ .

# Some simple forecasting methods

## Drift method

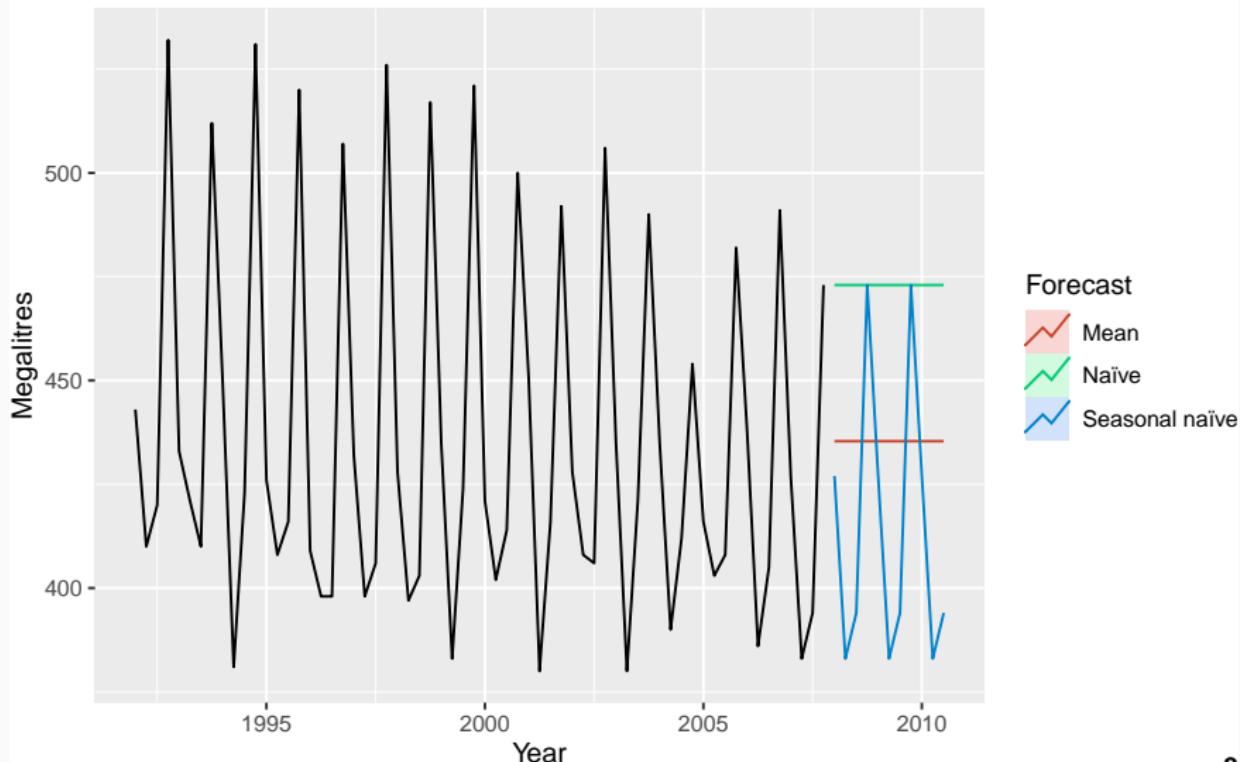
- Forecasts equal to last value plus average change.
- Forecasts:

$$\begin{aligned}\hat{y}_{T+h|T} &= y_T + \frac{h}{T-1} \sum_{t=2}^T (y_t - y_{t-1}) \\ &= y_T + \frac{h}{T-1} (y_T - y_1).\end{aligned}$$

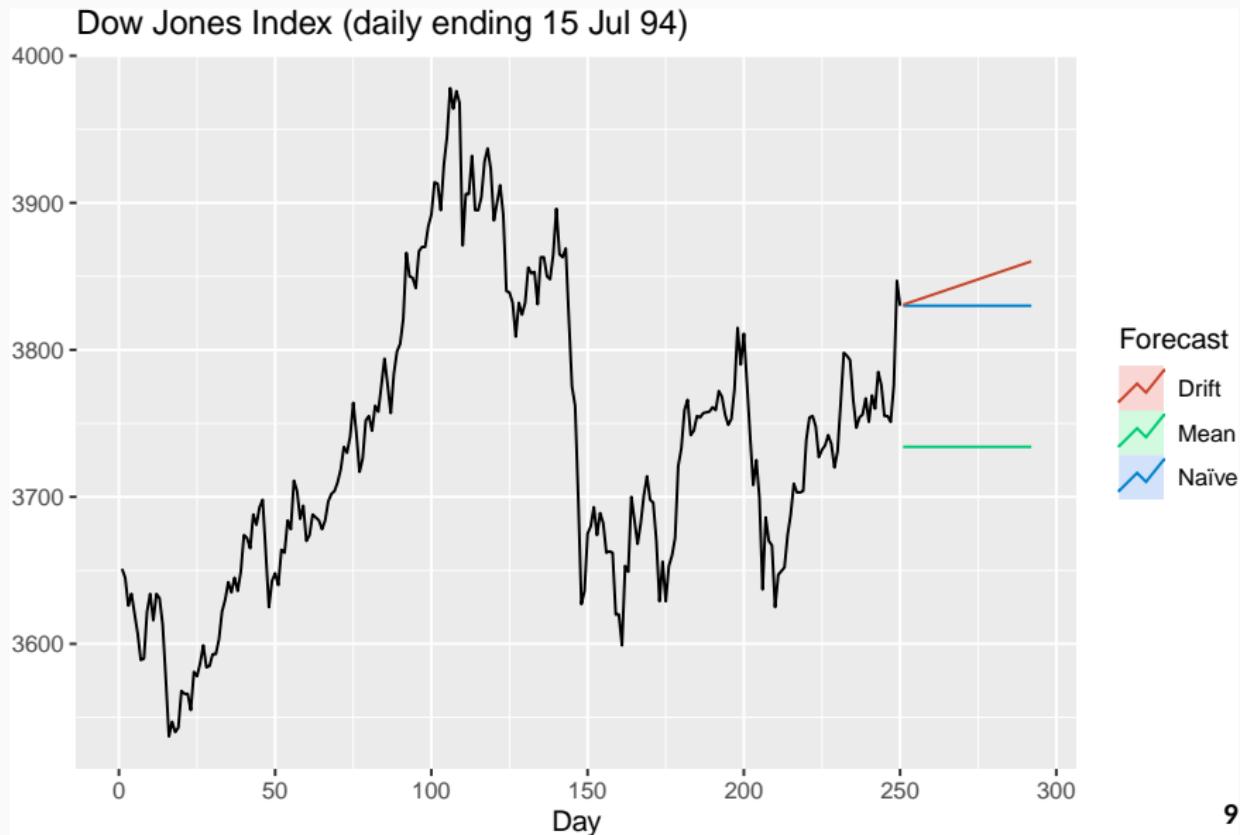
- Equivalent to extrapolating a line drawn between first and last observations.

# Some simple forecasting methods

Forecasts for quarterly beer production



# Some simple forecasting methods



# Some simple forecasting methods

- Mean: `meanf(y, h=20)`
- Naïve: `naive(y, h=20)`
- Seasonal naïve: `snaive(y, h=20)`
- Drift: `rwf(y, drift=TRUE, h=20)`

# Some simple forecasting methods

- Mean: `meanf(y, h=20)`
- Naïve: `naive(y, h=20)`
- Seasonal naïve: `snaive(y, h=20)`
- Drift: `rwf(y, drift=TRUE, h=20)`

## Your turn

- Use these four functions to produce forecasts for `goog` and `auscafe`.
- Plot the results using `autoplot()`.

# Outline

1 Some simple forecasting methods

2 Box-Cox transformations

3 Residual diagnostics

4 Evaluating forecast accuracy

5 Prediction intervals

## Variance stabilization

If the data show different variation at different levels of the series, then a transformation can be useful.

## Variance stabilization

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as  $y_1, \dots, y_n$  and transformed observations as  $w_1, \dots, w_n$ .

# Variance stabilization

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as  $y_1, \dots, y_n$  and transformed observations as  $w_1, \dots, w_n$ .

## Mathematical transformations for stabilizing variation

Square root     $w_t = \sqrt{y_t}$                 ↓

Cube root     $w_t = \sqrt[3]{y_t}$     Increasing

Logarithm     $w_t = \log(y_t)$     strength

# Variance stabilization

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as  $y_1, \dots, y_n$  and transformed observations as  $w_1, \dots, w_n$ .

## Mathematical transformations for stabilizing variation

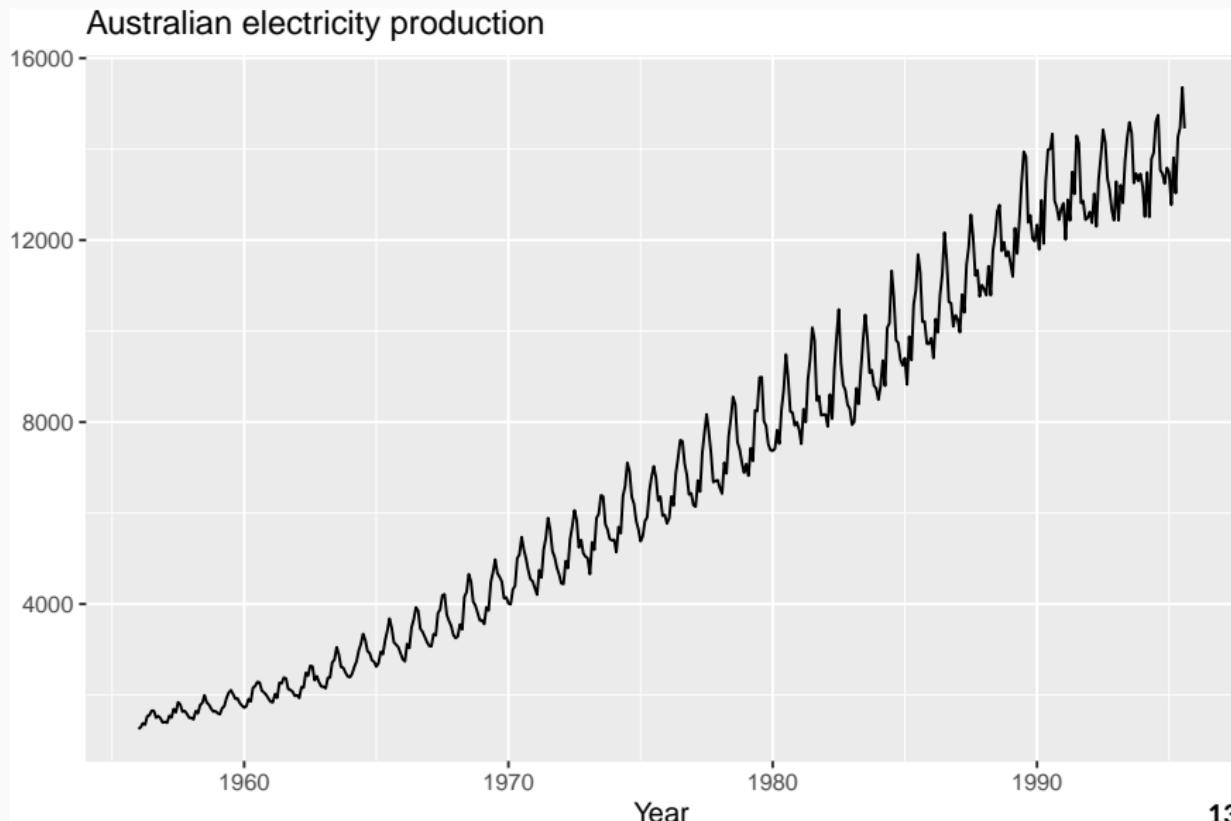
Square root     $w_t = \sqrt{y_t}$                       ↓

Cube root     $w_t = \sqrt[3]{y_t}$     Increasing

Logarithm     $w_t = \log(y_t)$     strength

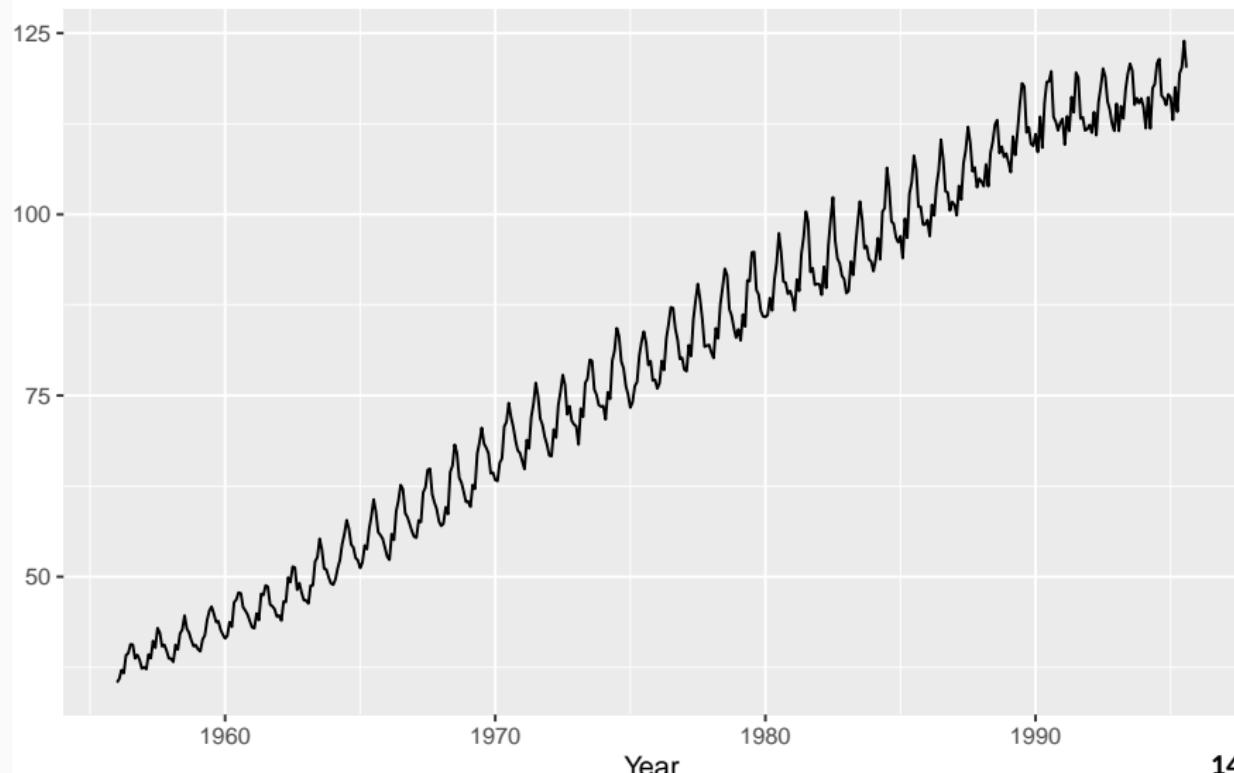
Logarithms, in particular, are useful because they are more interpretable: changes in a log value are **relative (percent) changes on the original scale**.

# Variance stabilization



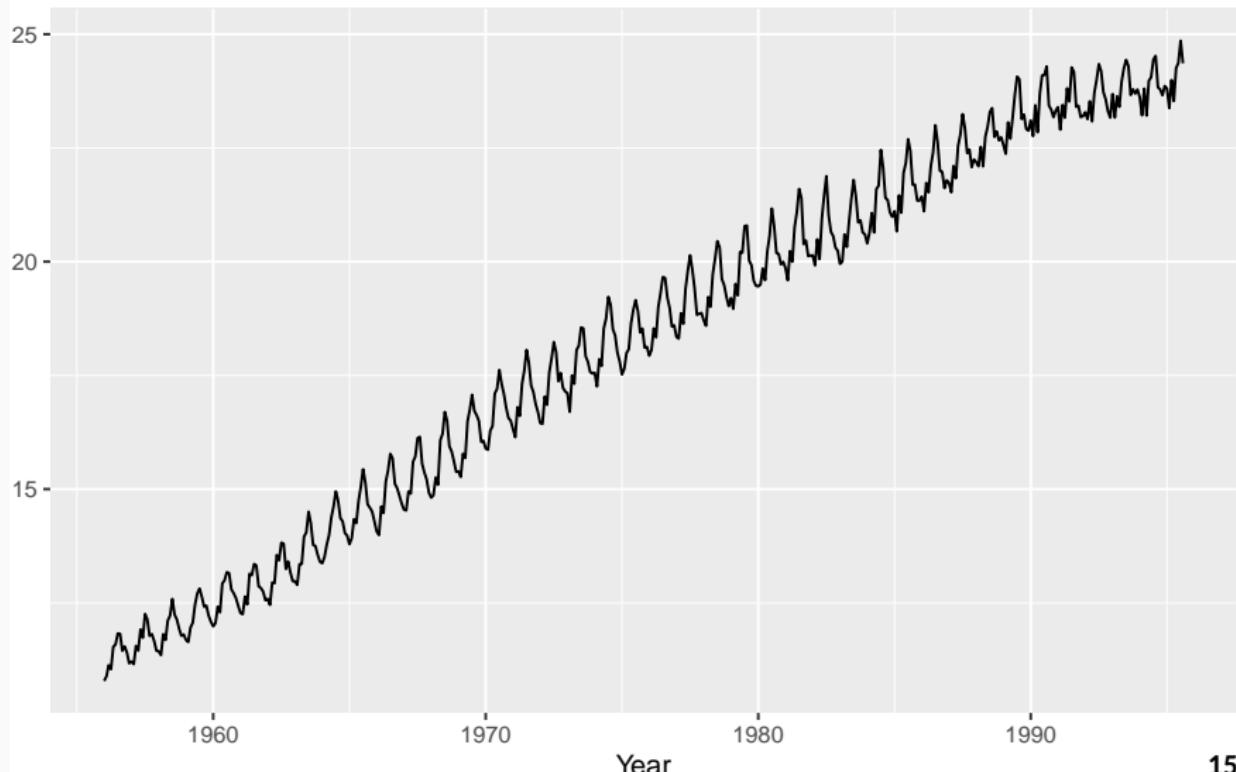
# Variance stabilization

Square root electricity production



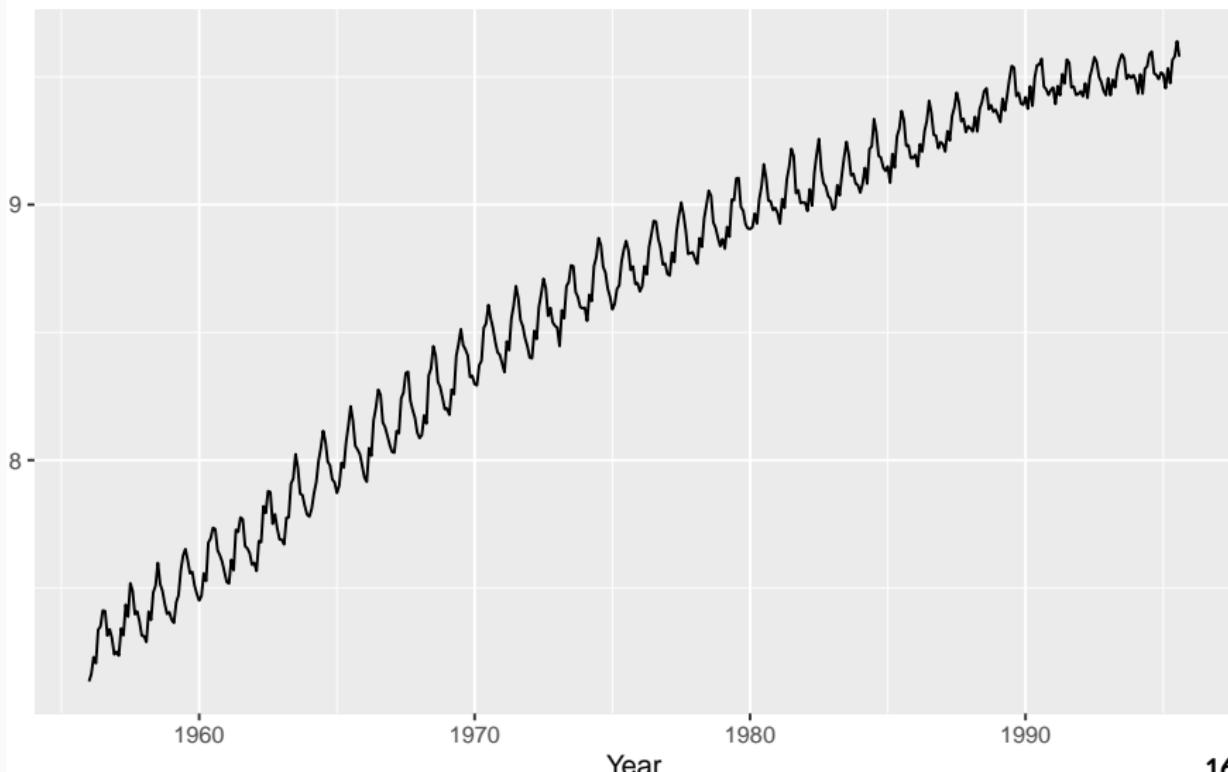
# Variance stabilization

Cube root electricity production



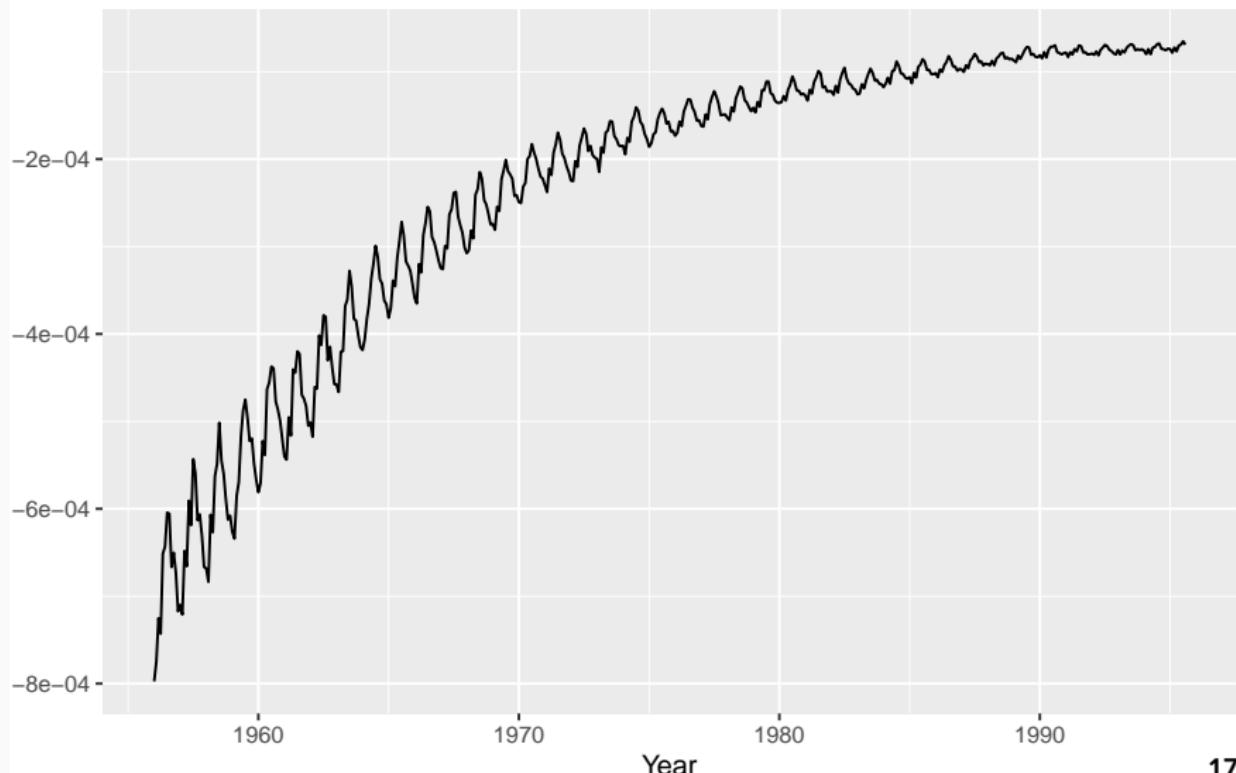
# Variance stabilization

Log electricity production



# Variance stabilization

Inverse electricity production



# Box-Cox transformations

Each of these transformations is close to a member of the family of **Box-Cox transformations**:

$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (y_t^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

# Box-Cox transformations

Each of these transformations is close to a member of the family of **Box-Cox transformations**:

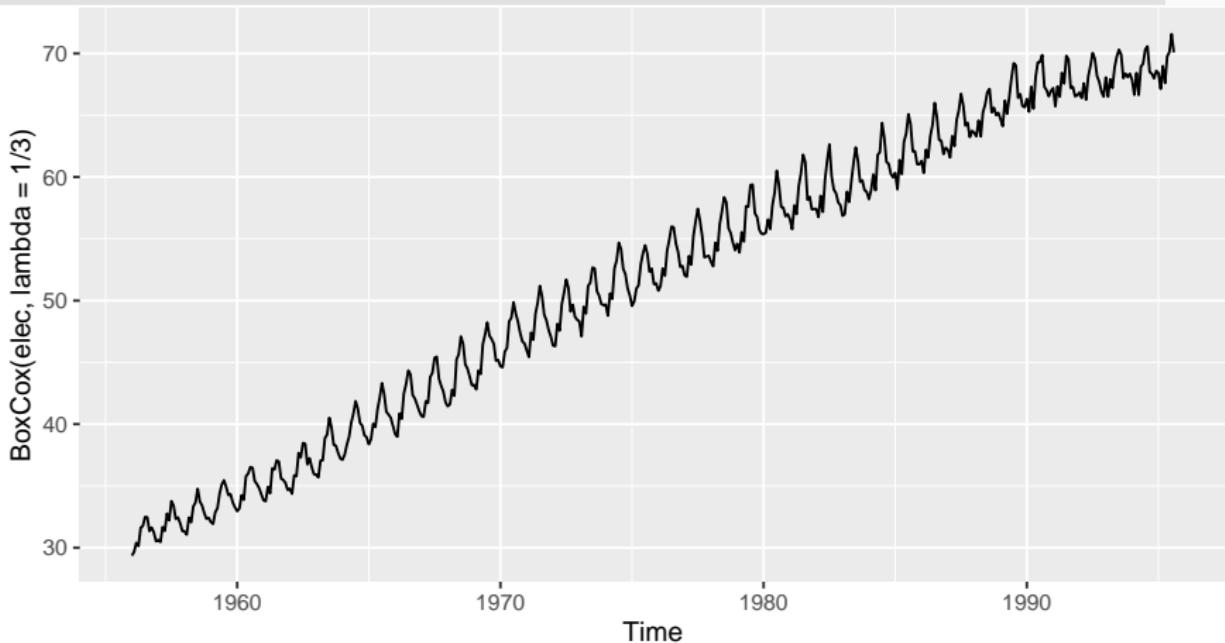
$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (y_t^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

- $\lambda = 1$ : (No substantive transformation)
- $\lambda = \frac{1}{2}$ : (Square root plus linear transformation)
- $\lambda = 0$ : (Natural logarithm)
- $\lambda = -1$ : (Inverse plus 1)

# Box-Cox transformations

# Box-Cox transformations

```
autoplot(BoxCox(elec, lambda=1/3))
```



# Box-Cox transformations

- $y_t^\lambda$  for  $\lambda$  close to zero behaves like logs.
- If some  $y_t = 0$ , then must have  $\lambda > 0$
- if some  $y_t < 0$ , no power transformation is possible unless all  $y_t$  adjusted by **adding a constant to all values.**
- Simple values of  $\lambda$  are easier to explain.
- Results are relatively insensitive to  $\lambda$ .
- Often no transformation ( $\lambda = 1$ ) needed.
- Transformation can have very large effect on PI.
- Choosing  $\lambda = 0$  is a simple way to force forecasts to be positive

# Automated Box-Cox transformations

```
(BoxCox.lambda(elec))
```

```
## [1] 0.2654076
```

# Automated Box-Cox transformations

```
(BoxCox.lambda(elec))
```

```
## [1] 0.2654076
```

- This attempts to balance the seasonal fluctuations and random variation across the series.
- Always check the results.
- A low value of  $\lambda$  can give extremely large prediction intervals.

# Back-transformation

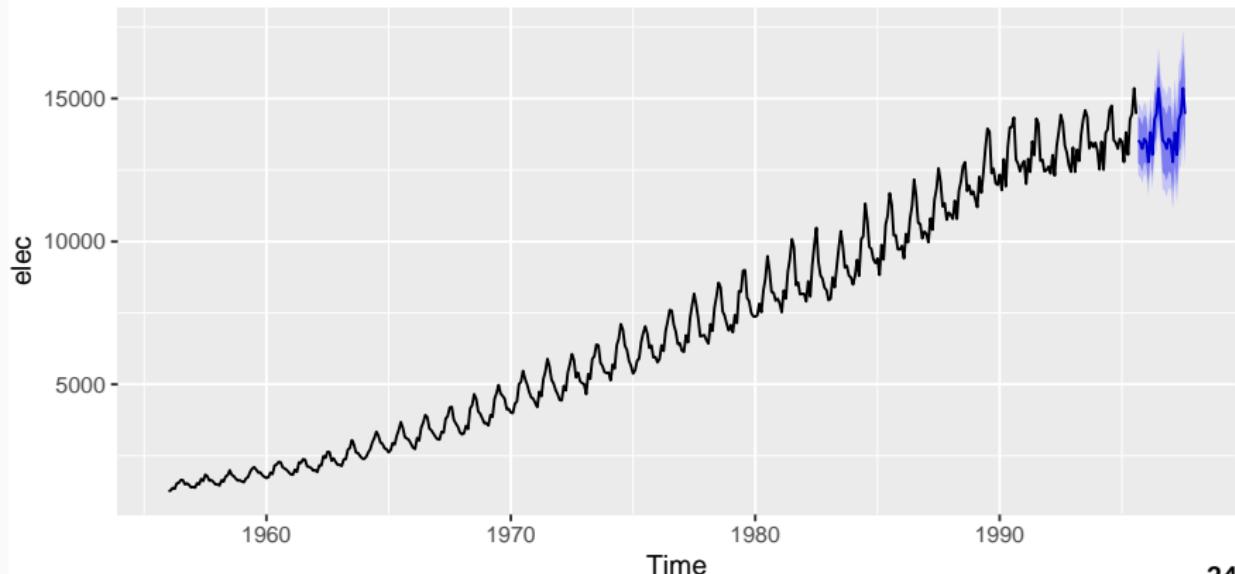
We must reverse the transformation (or *back-transform*) to obtain forecasts on the original scale. The reverse Box-Cox transformations are given by

$$y_t = \begin{cases} \exp(w_t), & \lambda = 0; \\ (\lambda w_t + 1)^{1/\lambda}, & \lambda \neq 0. \end{cases}$$

# Back-transformation

```
fit <- snaive(elec, lambda=1/3)  
autoplot(fit)
```

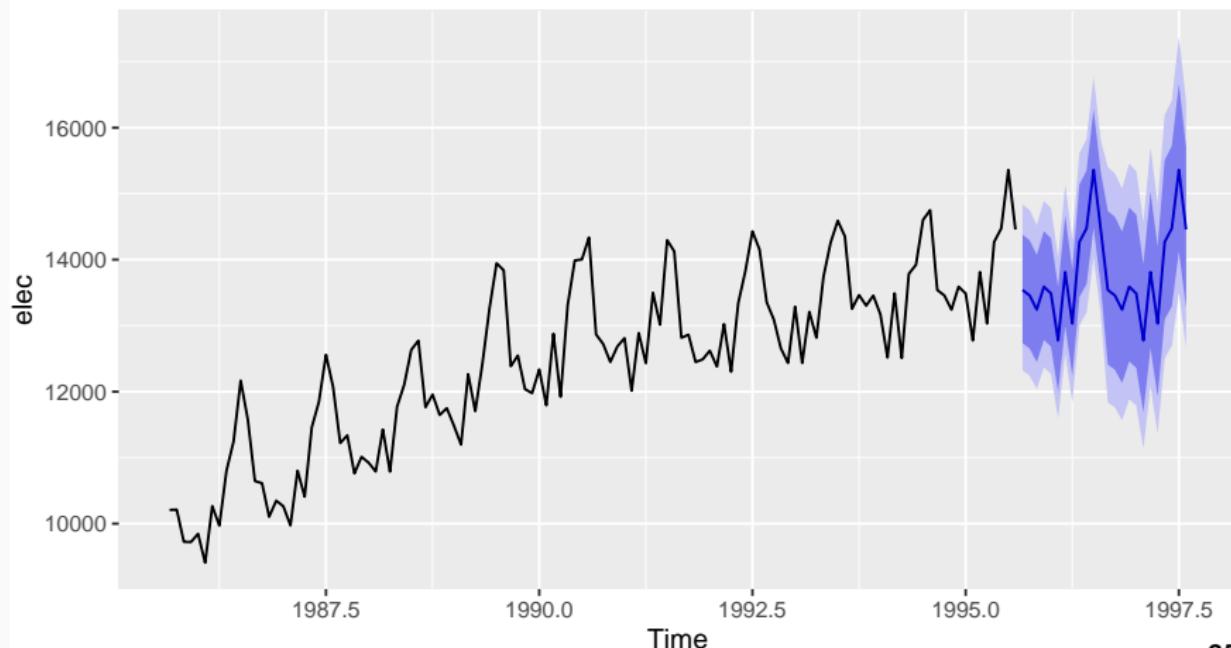
Forecasts from Seasonal naive method



# Back-transformation

```
autoplot(fit, include=120)
```

Forecasts from Seasonal naive method



## Your turn

Find a Box-Cox transformation that works for the gas data.

## Bias adjustment

- Back-transformed point forecasts are medians.
- Back-transformed PI have the correct coverage.

# Bias adjustment

- Back-transformed point forecasts are medians.
- Back-transformed PI have the correct coverage.

## Back-transformed means

Let  $X$  be have mean  $\mu$  and variance  $\sigma^2$ .

Let  $f(x)$  be back-transformation function, and  $Y = f(X)$ .

Taylor series expansion about  $\mu$ :

$$f(X) = f(\mu) + (X - \mu)f'(\mu) + \frac{1}{2}(X - \mu)^2f''(\mu).$$

# Bias adjustment

- Back-transformed point forecasts are medians.
- Back-transformed PI have the correct coverage.

## Back-transformed means

Let  $X$  be have mean  $\mu$  and variance  $\sigma^2$ .

Let  $f(x)$  be back-transformation function, and  $Y = f(X)$ .

Taylor series expansion about  $\mu$ :

$$f(X) = f(\mu) + (X - \mu)f'(\mu) + \frac{1}{2}(X - \mu)^2f''(\mu).$$

$$E[Y] = E[f(X)] = f(\mu) + \frac{1}{2}\sigma^2f''(\mu)$$

# Bias adjustment

Box-Cox back-transformation:

$$y_t = \begin{cases} \exp(w_t) & \lambda = 0; \\ (\lambda W_t + 1)^{1/\lambda} & \lambda \neq 0. \end{cases}$$

$$f(x) = \begin{cases} e^x & \lambda = 0; \\ (\lambda x + 1)^{1/\lambda} & \lambda \neq 0. \end{cases}$$

$$f''(x) = \begin{cases} e^x & \lambda = 0; \\ (1 - \lambda)(\lambda x + 1)^{1/\lambda - 2} & \lambda \neq 0. \end{cases}$$

# Bias adjustment

Box-Cox back-transformation:

$$y_t = \begin{cases} \exp(w_t) & \lambda = 0; \\ (\lambda W_t + 1)^{1/\lambda} & \lambda \neq 0. \end{cases}$$

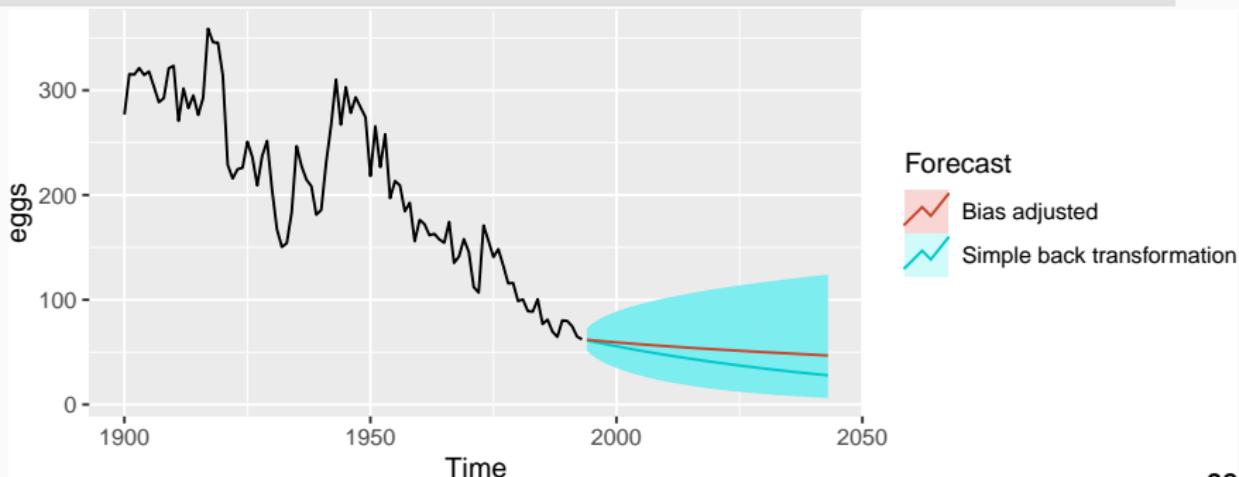
$$f(x) = \begin{cases} e^x & \lambda = 0; \\ (\lambda x + 1)^{1/\lambda} & \lambda \neq 0. \end{cases}$$

$$f''(x) = \begin{cases} e^x & \lambda = 0; \\ (1 - \lambda)(\lambda x + 1)^{1/\lambda - 2} & \lambda \neq 0. \end{cases}$$

$$E[Y] = \begin{cases} e^\mu \left[ 1 + \frac{\sigma^2}{2} \right] & \lambda = 0; \\ (\lambda \mu + 1)^{1/\lambda} \left[ 1 + \frac{\sigma^2(1-\lambda)}{2(\lambda \mu + 1)^2} \right] & \lambda \neq 0. \end{cases}$$

# Bias adjustment

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



# Outline

**1 Some simple forecasting methods**

**2 Box-Cox transformations**

**3 Residual diagnostics**

**4 Evaluating forecast accuracy**

**5 Prediction intervals**

# Fitted values

- $\hat{y}_{t|t-1}$  is the forecast of  $y_t$  based on observations  $y_1, \dots, y_{t-1}$ .
- We call these “fitted values”.
- Sometimes drop the subscript:  $\hat{y}_t \equiv \hat{y}_{t|t-1}$ .
- Often not true forecasts since parameters are estimated on all data.

## For example:

- $\hat{y}_t = \bar{y}$  for average method.
- $\hat{y}_t = y_{t-1} + (y_T - y_1)/(T - 1)$  for drift method.

# Forecasting residuals

**Residuals in forecasting:** difference between observed value and its fitted value:  $e_t = y_t - \hat{y}_{t|t-1}$ .

# Forecasting residuals

**Residuals in forecasting:** difference between observed value and its fitted value:  $e_t = y_t - \hat{y}_{t|t-1}$ .

## Assumptions

- 1  $\{e_t\}$  uncorrelated. If they aren't, then information left in residuals that should be used in computing forecasts.
- 2  $\{e_t\}$  have mean zero. If they don't, then forecasts are biased.

# Forecasting residuals

**Residuals in forecasting:** difference between observed value and its fitted value:  $e_t = y_t - \hat{y}_{t|t-1}$ .

## Assumptions

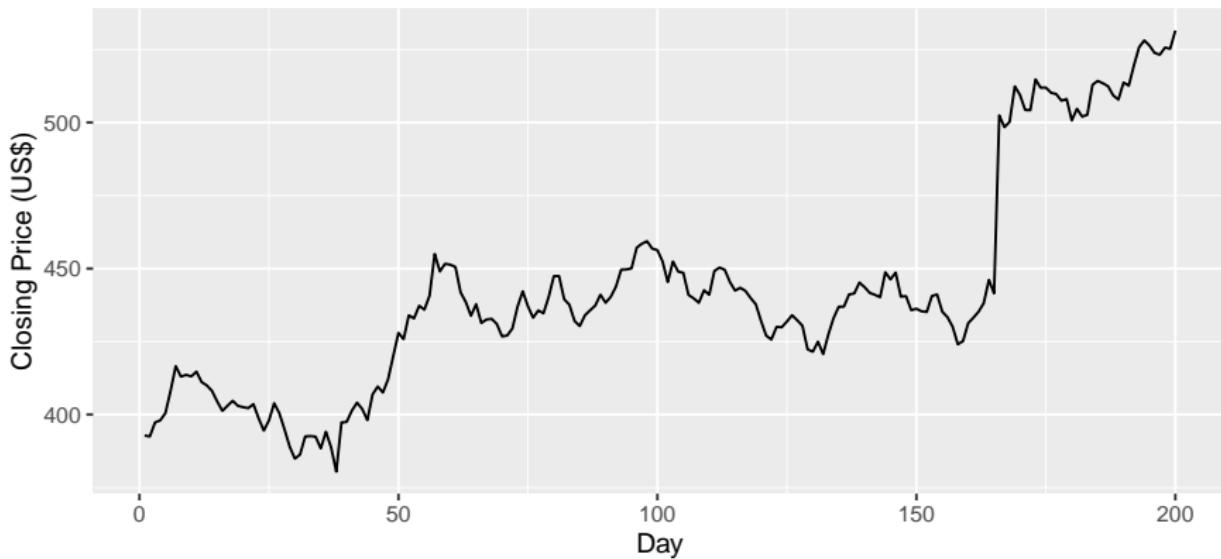
- 1  $\{e_t\}$  uncorrelated. If they aren't, then information left in residuals that should be used in computing forecasts.
- 2  $\{e_t\}$  have mean zero. If they don't, then forecasts are biased.

## Useful properties (for prediction intervals)

- 3  $\{e_t\}$  have constant variance.
- 4  $\{e_t\}$  are normally distributed.

# Example: Google stock price

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



# Example: Google stock price

Naïve forecast:

$$\hat{y}_{t|t-1} = y_{t-1}$$

# Example: Google stock price

Naïve forecast:

$$\hat{y}_{t|t-1} = y_{t-1}$$

$$e_t = y_t - y_{t-1}$$

# Example: Google stock price

Naïve forecast:

$$\hat{y}_{t|t-1} = y_{t-1}$$

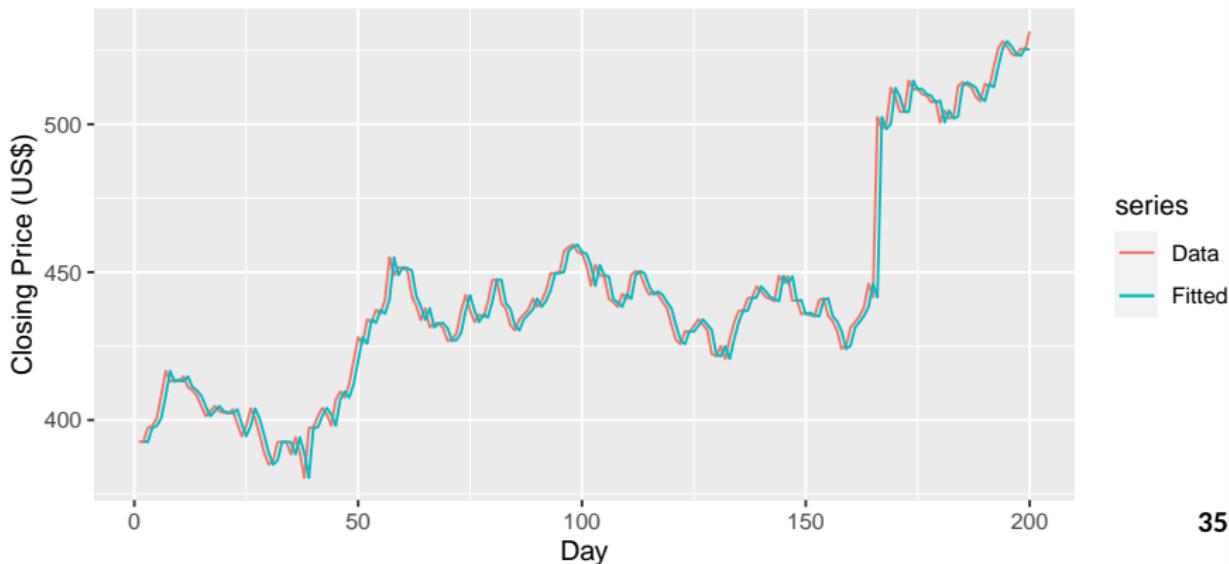
$$e_t = y_t - y_{t-1}$$

Note:  $e_t$  are one-step-forecast residuals

# Example: Google stock price

```
fits <- fitted(naive(goog200))
autoplot(goog200, series="Data") +
  autolayer(fits, series="Fitted") +
  xlab("Day") + ylab("Closing Price (US$)") +
  ggtitle("Google Stock (daily ending 6 December 2013)")
```

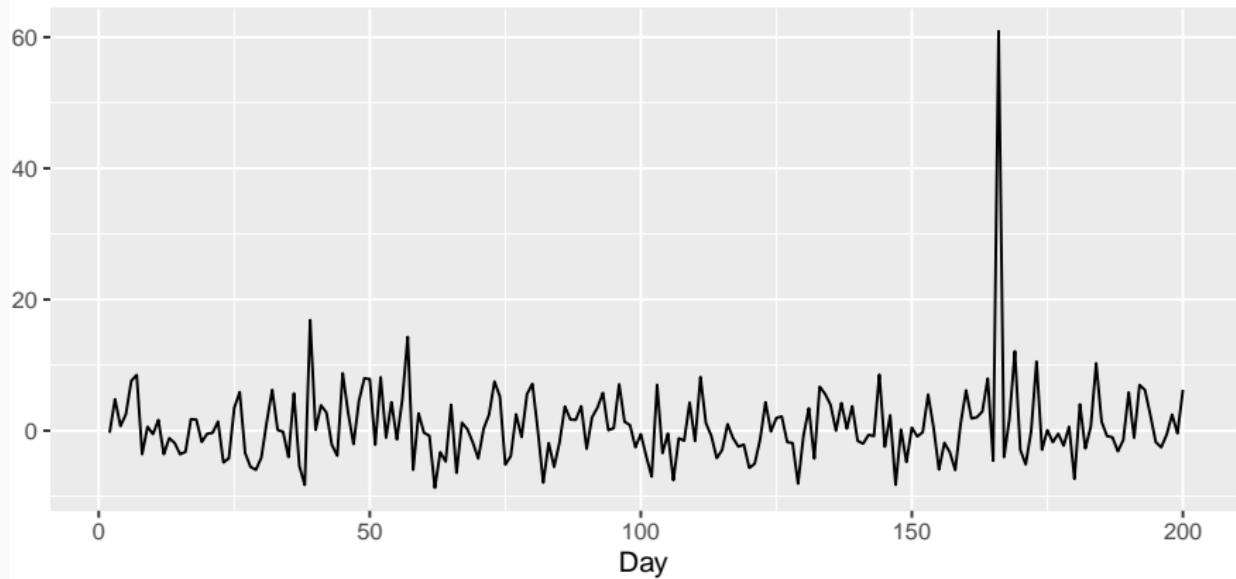
Google Stock (daily ending 6 December 2013)



# Example: Google stock price

```
res <- residuals(naive(goog200))
autoplot(res) + xlab("Day") + ylab("") +
ggtitle("Residuals from naïve method")
```

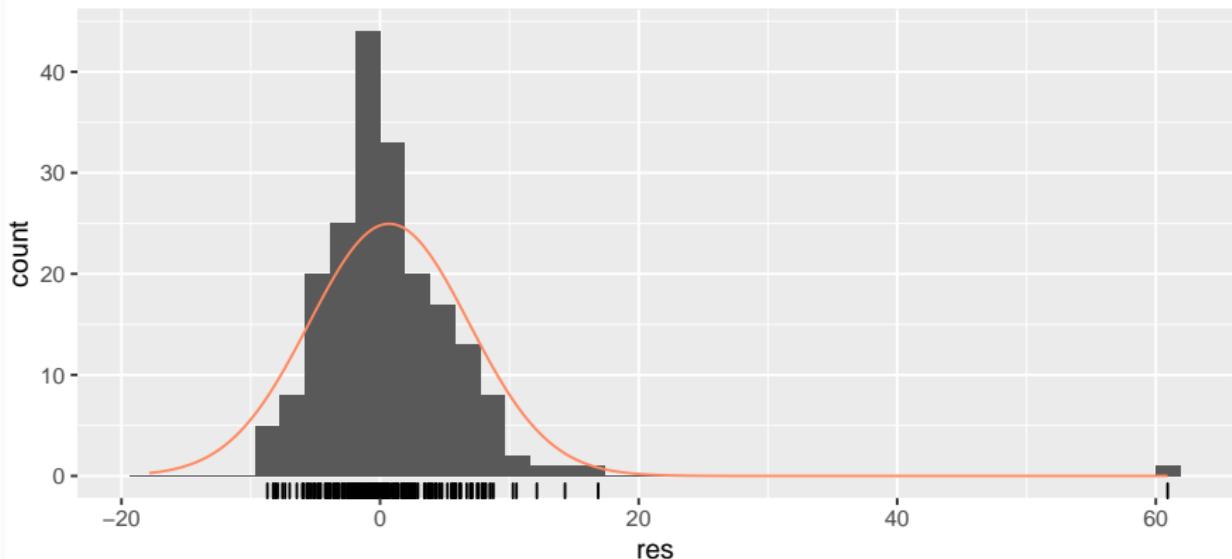
Residuals from naïve method



# Example: Google stock price

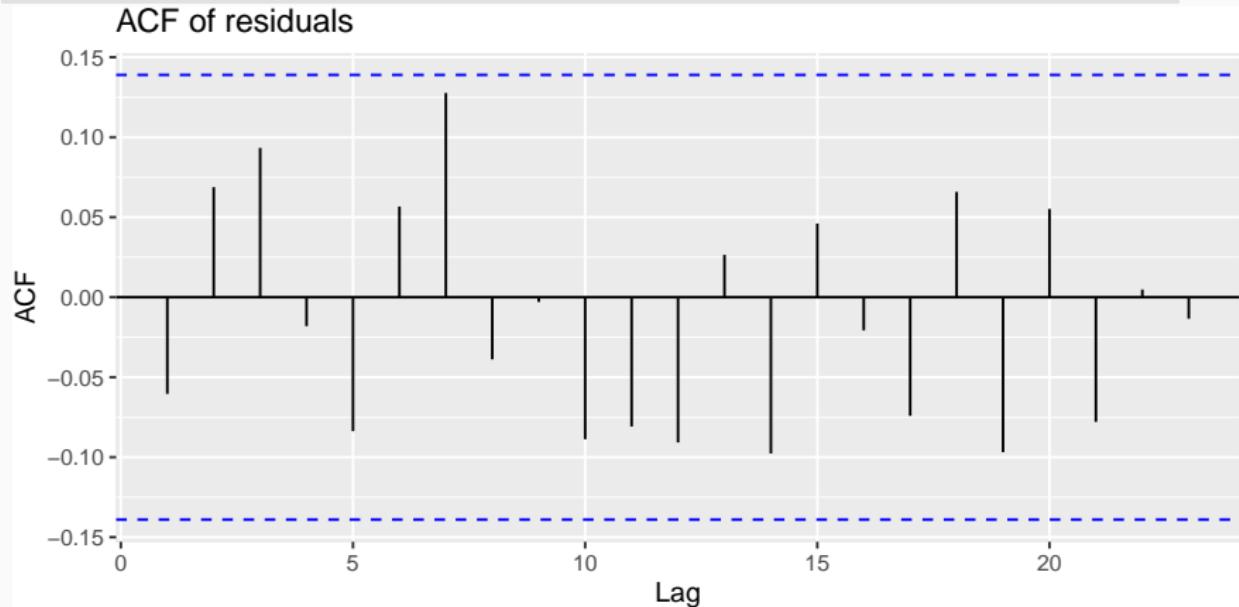
```
gghistogram(res, add.normal=TRUE) +  
  ggtitle("Histogram of residuals")
```

Histogram of residuals



# Example: Google stock price

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



## ACF of residuals

- We assume that the residuals are white noise (uncorrelated, mean zero, constant variance). If they aren't, then there is information left in the residuals that should be used in computing forecasts.
- So a standard residual diagnostic is to check the ACF of the residuals of a forecasting method.
- We *expect* these to look like white noise.

## Portmanteau tests

Consider a *whole set* of  $r_k$  values, and develop a test to see whether the set is significantly different from a zero set.

# Portmanteau tests

Consider a *whole set* of  $r_k$  values, and develop a test to see whether the set is significantly different from a zero set.

## Box-Pierce test

$$Q = T \sum_{k=1}^h r_k^2$$

where  $h$  is max lag being considered and  $T$  is number of observations.

- If each  $r_k$  close to zero,  $Q$  will be **small**.
- If some  $r_k$  values large (positive or negative),  $Q$  will be **large**.

# Portmanteau tests

Consider a *whole set* of  $r_k$  values, and develop a test to see whether the set is significantly different from a zero set.

## Ljung-Box test

$$Q^* = T(T + 2) \sum_{k=1}^h (T - k)^{-1} r_k^2$$

where  $h$  is max lag being considered and  $T$  is number of observations.

- My preferences:  $h = 10$  for non-seasonal data,  
 $h = 2m$  for seasonal data.
- Better performance, especially in small samples.

# Portmanteau tests

- If data are WN,  $Q^*$  has  $\chi^2$  distribution with  $(h - K)$  degrees of freedom where  $K$  = no. parameters in model.
- When applied to raw data, set  $K = 0$ .
- For the Google example:

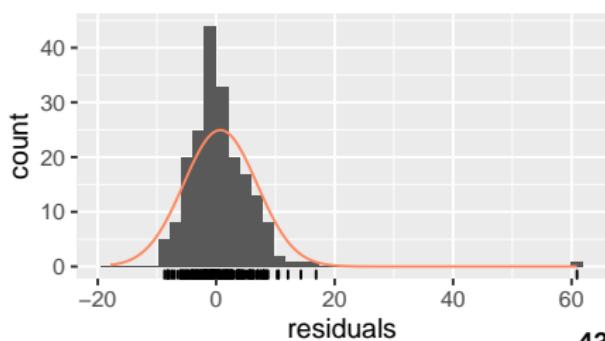
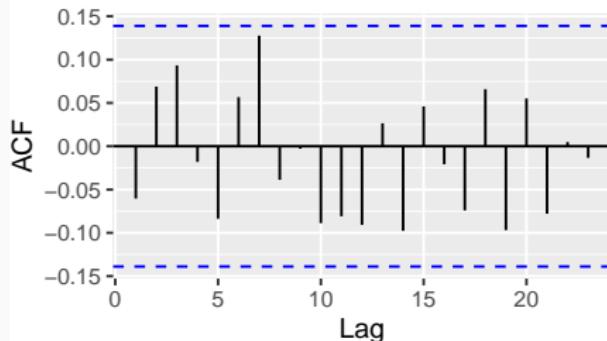
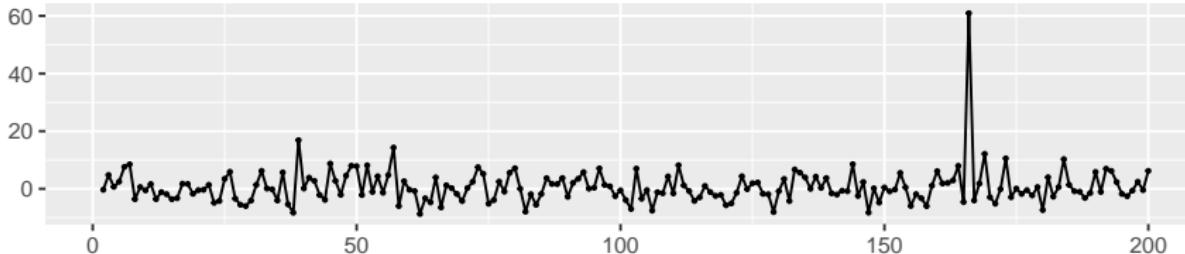
```
# lag=h and fitdf=K
Box.test(res, lag=10, fitdf=0, type="Lj")

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

# checkresiduals function

checkresiduals(naive(goog200))

Residuals from Naive method



## checkresiduals function

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

## Your turn

Compute seasonal naïve forecasts for quarterly Australian beer production from 1992.

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

Test if the residuals are white noise.

```
checkresiduals(fc)
```

What do you conclude?

# Outline

1 Some simple forecasting methods

2 Box-Cox transformations

3 Residual diagnostics

4 Evaluating forecast accuracy

5 Prediction intervals

# Training and test sets



- 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.
- The test set must not be used for *any* aspect of model development or calculation of forecasts.
- Forecast accuracy is based only on the test set.

## Forecast errors

Forecast “error”: the difference between an observed value and its forecast.

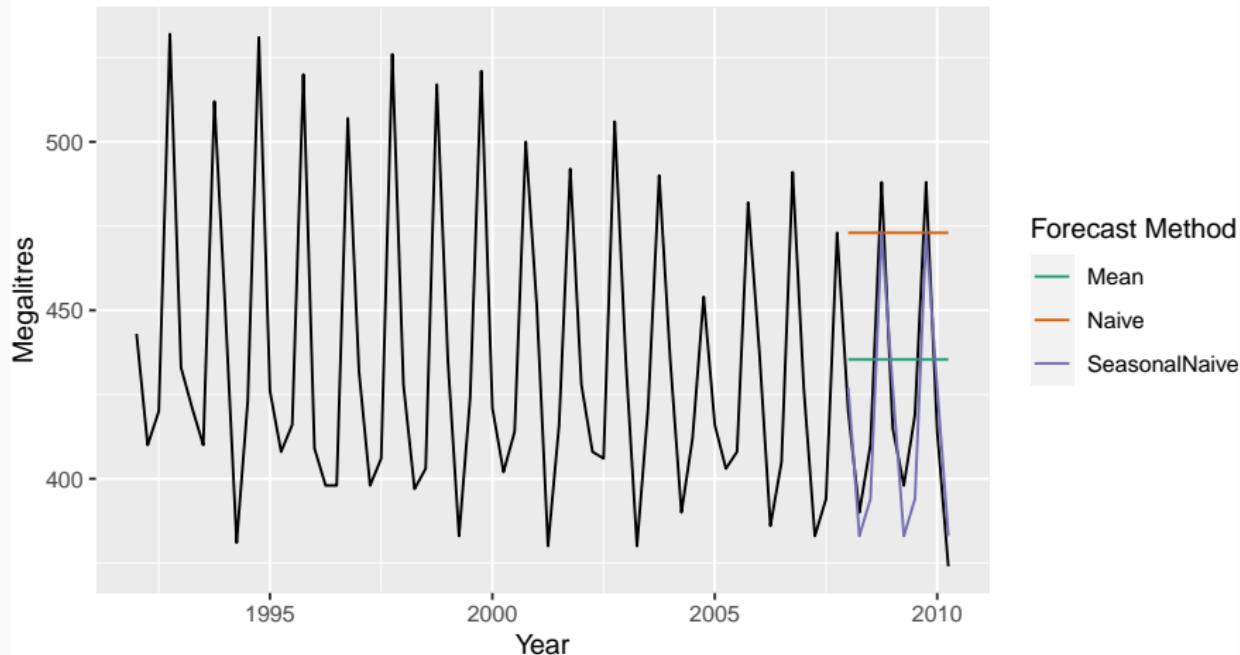
$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

where the training data is given by  $\{y_1, \dots, y_T\}$

- Unlike residuals, forecast errors on the test set involve multi-step forecasts.
- These are *true* forecast errors as the test data is not used in computing  $\hat{y}_{T+h|T}$ .

# Measures of forecast accuracy

Forecasts for quarterly beer production



## Measures of forecast accuracy

$y_{T+h}$  =  $(T + h)$ th observation,  $h = 1, \dots, H$

$\hat{y}_{T+h|T}$  = its forecast based on data up to time  $T$ .

$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T}$

MAE =  $\text{mean}(|e_{T+h}|)$

MSE =  $\text{mean}(e_{T+h}^2)$

RMSE =  $\sqrt{\text{mean}(e_{T+h}^2)}$

MAPE =  $100\text{mean}(|e_{T+h}|/|y_{T+h}|)$

## Measures of forecast accuracy

$y_{T+h}$  =  $(T + h)$ th observation,  $h = 1, \dots, H$

$\hat{y}_{T+h|T}$  = its forecast based on data up to time  $T$ .

$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T}$

MAE =  $\text{mean}(|e_{T+h}|)$

MSE =  $\text{mean}(e_{T+h}^2)$

RMSE =  $\sqrt{\text{mean}(e_{T+h}^2)}$

MAPE =  $100\text{mean}(|e_{T+h}|/|y_{T+h}|)$

- MAE, MSE, RMSE are all scale dependent.
- MAPE is scale independent but is only sensible if  $y_t \gg 0$  for all  $t$ , and  $y$  has a natural zero.

# Measures of forecast accuracy

## Mean Absolute Scaled Error

$$\text{MASE} = \text{mean}(|e_{T+h}| / Q)$$

where  $Q$  is a stable measure of the scale of the time series  $\{y_t\}$ .

Proposed by Hyndman and Koehler (IJF, 2006).

For non-seasonal time series,

$$Q = (T - 1)^{-1} \sum_{t=2}^T |y_t - y_{t-1}|$$

works well. Then MASE is equivalent to MAE relative to a naïve method.

# Measures of forecast accuracy

## Mean Absolute Scaled Error

$$\text{MASE} = \text{mean}(|e_{T+h}| / Q)$$

where  $Q$  is a stable measure of the scale of the time series  $\{y_t\}$ .

Proposed by Hyndman and Koehler (IJF, 2006).

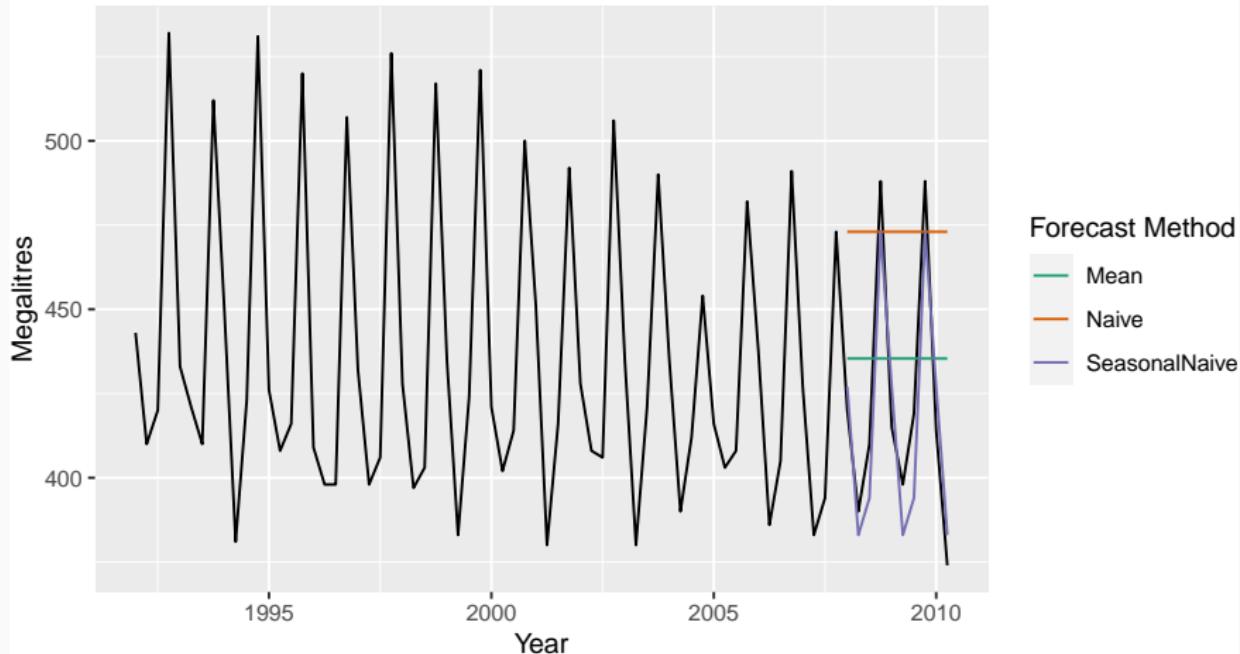
For seasonal time series,

$$Q = (T - m)^{-1} \sum_{t=m+1}^T |y_t - y_{t-m}|$$

works well. Then MASE is equivalent to MAE relative to a seasonal naïve method.

# Measures of forecast accuracy

Forecasts for quarterly beer production



# Measures of forecast accuracy

```
beer2 <- window(ausbeer, start=1992, end=c(2007,4))
beer3 <- window(ausbeer, start=2008)
beerfit1 <- meanf(beer2, h=10)
beerfit2 <- rwf(beer2, h=10)
beerfit3 <- snaive(beer2, h=10)
accuracy(beerfit1, beer3)
accuracy(beerfit2, beer3)
accuracy(beerfit3, beer3)
```

	RMSE	MAE	MAPE	MASE
Mean method	38.45	34.83	8.28	2.44
Naïve method	62.69	57.40	14.18	4.01
Seasonal naïve method	14.31	13.40	3.17	0.94

## Poll: true or false?

- 1 Good forecast methods should have normally distributed residuals.
- 2 A model with small residuals will give good forecasts.
- 3 The best measure of forecast accuracy is MAPE.
- 4 If your model doesn't forecast well, you should make it more complicated.
- 5 Always choose the model with the best forecast accuracy as measured on the test set.

# Time series cross-validation

## Traditional evaluation

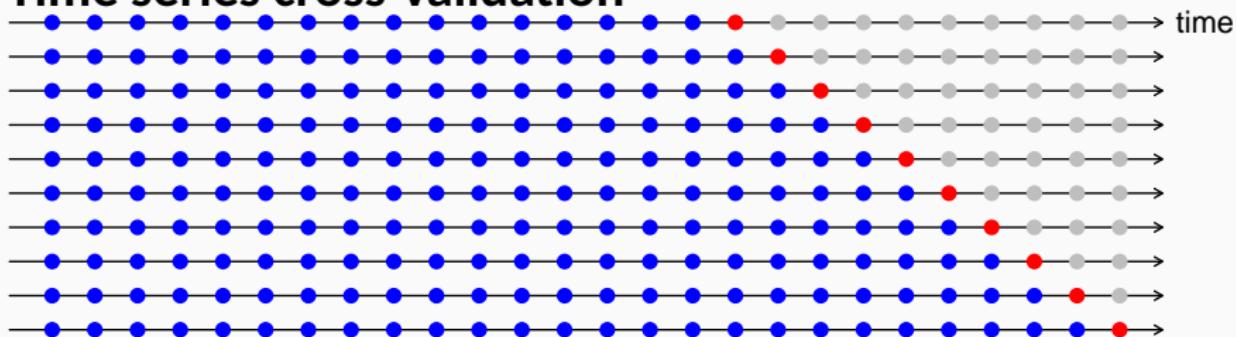


# Time series cross-validation

## Traditional evaluation



## Time series cross-validation

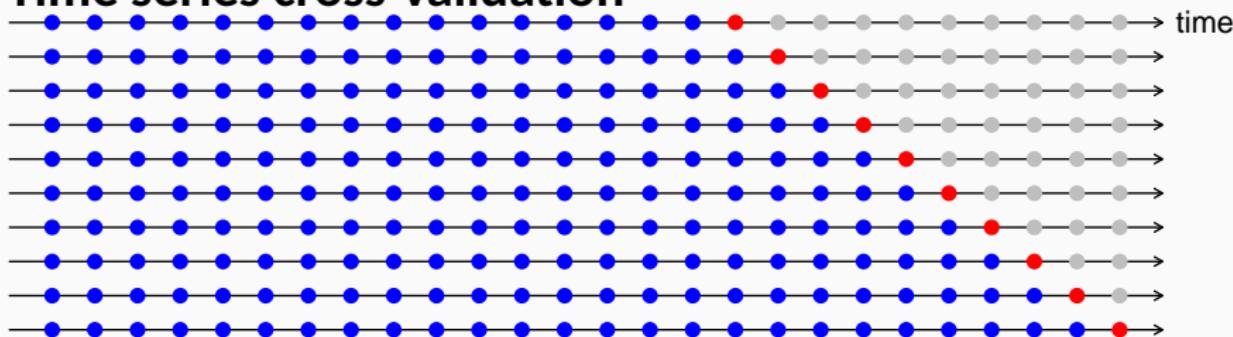


# Time series cross-validation

## Traditional evaluation



## Time series cross-validation



- Forecast accuracy averaged over test sets.
- Also known as “evaluation on a rolling forecasting origin”

## tsCV function:

```
e <- tsCV(goog200, rwf, drift=TRUE, h=1)
sqrt(mean(e^2, na.rm=TRUE))

## [1] 6.233245

sqrt(mean(residuals(rwf(goog200, drift=TRUE))^2,
na.rm=TRUE))
```

```
## [1] 6.168928
```

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 function

Ugly code:

```
e <- tsCV(goog200, rwf, drift=TRUE, h=1)
sqrt(mean(e^2, na.rm=TRUE))
sqrt(mean(residuals(rwf(goog200, drift=TRUE))^2,
          na.rm=TRUE))
```

Better with a pipe:

```
goog200 %>%
  tsCV(forecastfunction=rwf, drift=TRUE, h=1) -> e
e^2 %>% mean(na.rm=TRUE) %>% sqrt
goog200 %>% rwf(drift=TRUE) %>% residuals -> res
res^2 %>% mean(na.rm=TRUE) %>% sqrt
```

# Outline

**1 Some simple forecasting methods**

**2 Box-Cox transformations**

**3 Residual diagnostics**

**4 Evaluating forecast accuracy**

**5 Prediction intervals**

# Prediction intervals

- A forecast  $\hat{y}_{T+h|T}$  is (usually) the mean of the conditional distribution  $y_{T+h} \mid y_1, \dots, y_T$ .
- A prediction interval gives a region within which we expect  $y_{T+h}$  to lie with a specified probability.
- Assuming forecast errors are normally distributed, then a 95% PI is

$$\hat{y}_{T+h|T} \pm 1.96 \hat{\sigma}_h$$

where  $\hat{\sigma}_h$  is the st dev of the  $h$ -step distribution.

- When  $h = 1$ ,  $\hat{\sigma}_h$  can be estimated from the residuals.

# Prediction intervals

Naive forecast with prediction interval:

```
res_sd <- sqrt(mean(res^2, na.rm=TRUE))  
c(tail(goog200,1)) + 1.96 * res_sd * c(-1,1)
```

```
## [1] 519.3103 543.6462
```

```
naive(goog200, level=95)
```

	Point Forecast	Lo 95	Hi 95
## 201	531.4783	519.3105	543.6460
## 202	531.4783	514.2705	548.6861
## 203	531.4783	510.4031	552.5534
## 204	531.4783	507.1428	555.8138
## 205	531.4783	504.2704	558.6862
## 206	531.4783	501.6735	561.2830
## 207	531.4783	499.2854	563.6711

# Prediction intervals

- Point forecasts are often useless without prediction intervals.
- Prediction intervals require a stochastic model (with random errors, etc).
- Multi-step forecasts for time series require a more sophisticated approach (with PI getting wider as the forecast horizon increases).

# Prediction intervals

Assume residuals are normal, uncorrelated,  $\text{sd} = \hat{\sigma}$ :

**Mean forecasts:**  $\hat{\sigma}_h = \hat{\sigma}\sqrt{1 + 1/T}$

**Naïve forecasts:**  $\hat{\sigma}_h = \hat{\sigma}\sqrt{h}$

**Seasonal naïve forecasts**  $\hat{\sigma}_h = \hat{\sigma}\sqrt{k + 1}$

**Drift forecasts:**  $\hat{\sigma}_h = \hat{\sigma}\sqrt{h(1 + h/T)}$ .

where  $k$  is the integer part of  $(h - 1)/m$ .

Note that when  $h = 1$  and  $T$  is large, these all give the same approximate value  $\hat{\sigma}$ .

# Prediction intervals

- Computed automatically using: `naive()`,  
`snaive()`, `rwf()`, `meanf()`, etc.
- Use `level` argument to control coverage.
- Check residual assumptions before believing them.
- Usually too narrow due to unaccounted uncertainty.



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# STU33010: Forecasting

Ch5. Regression models

[OTexts.org/fpp2/](https://OTexts.org/fpp2/)

# Outline

- 1 The linear model with time series**
- 2 Residual diagnostics**
- 3 Some useful predictors for linear models**
- 4 Selecting predictors and forecast evaluation**
- 5 Forecasting with regression**
- 6 Matrix formulation**
- 7 Correlation, causation and forecasting**

# Multiple regression and forecasting

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t.$$

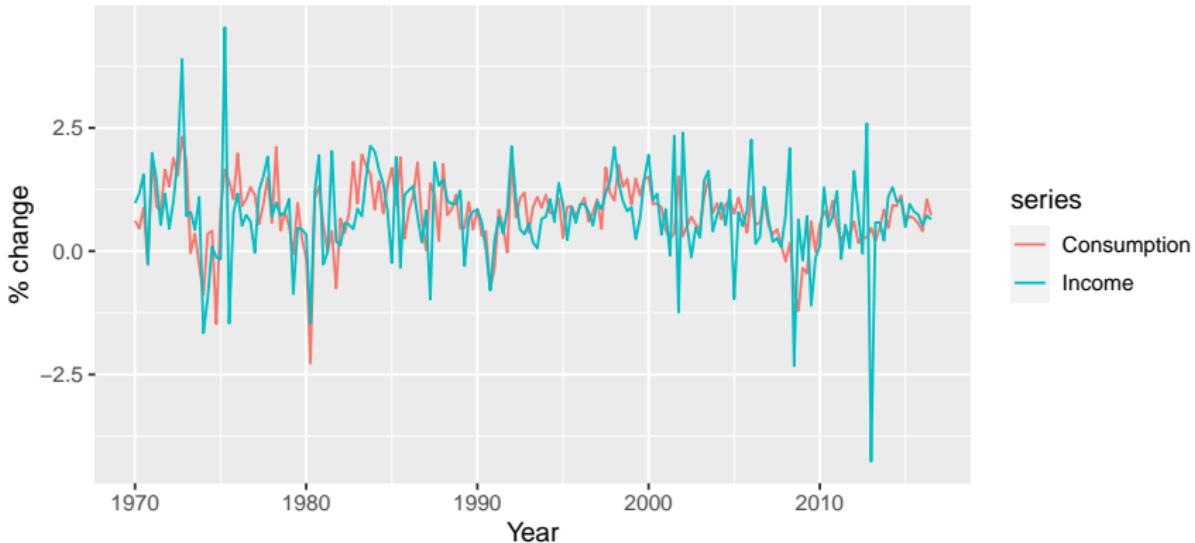
- $y_t$  is the variable we want to predict: the “response” variable
- Each  $x_{j,t}$  is numerical and is called a “predictor”. They are usually assumed to be known for all past and future times.
- The coefficients  $\beta_1, \dots, \beta_k$  measure the effect of each predictor after taking account of the effect of all other predictors in the model.

That is, the coefficients measure the marginal effects.

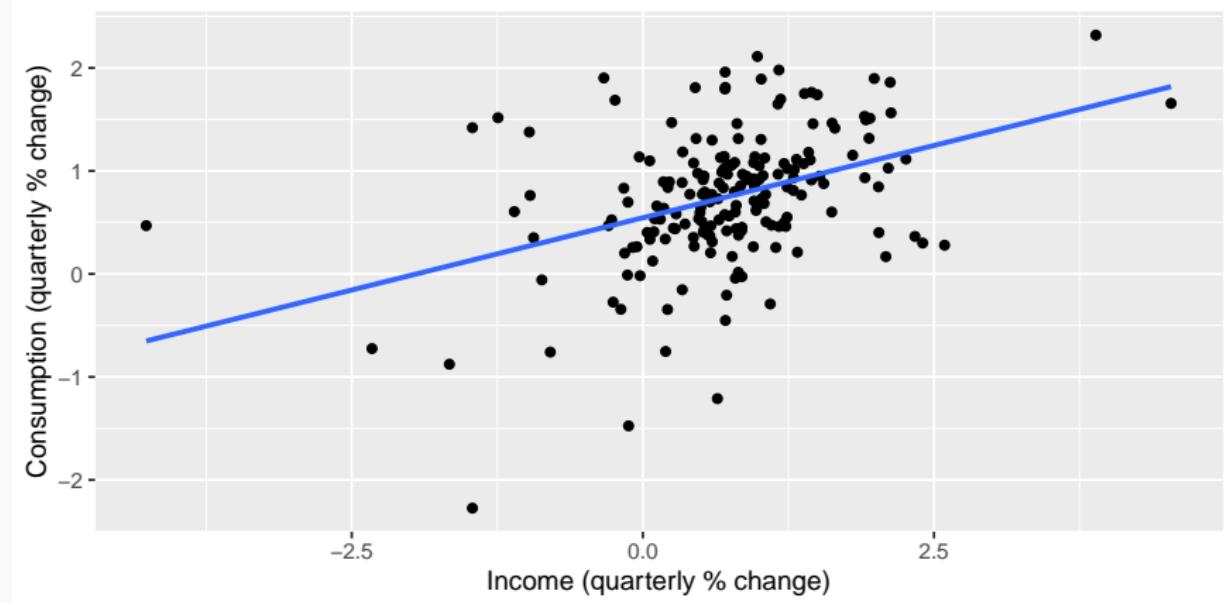
- $\varepsilon_t$  is a white noise error term

# Example: US consumption expenditure

```
autoplot(uschange[,c("Consumption","Income")]) +  
  ylab("% change") + xlab("Year")
```



# Example: US consumption expenditure

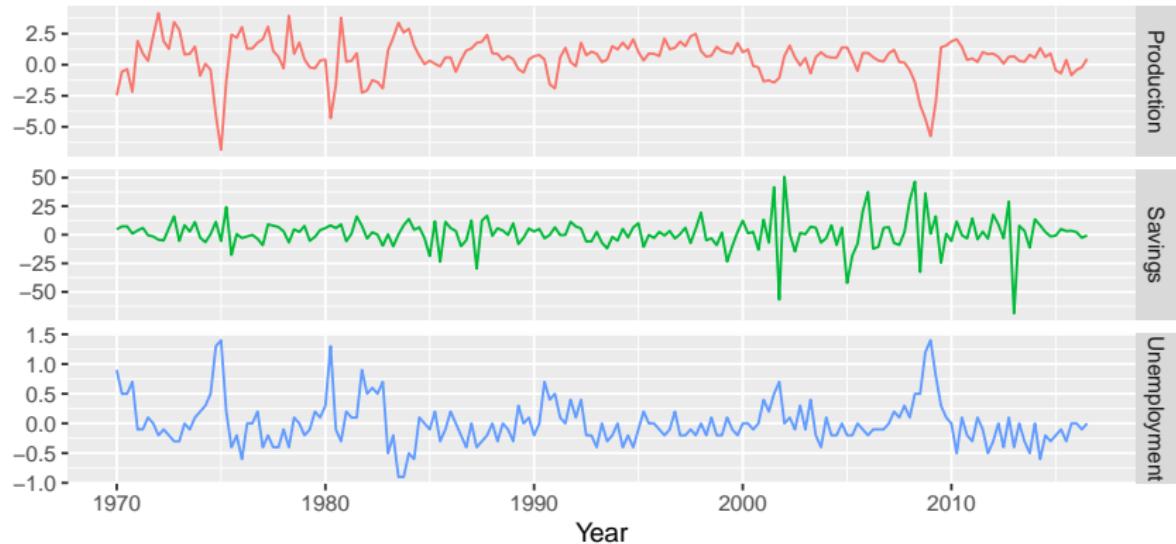


# Example: US consumption expenditure

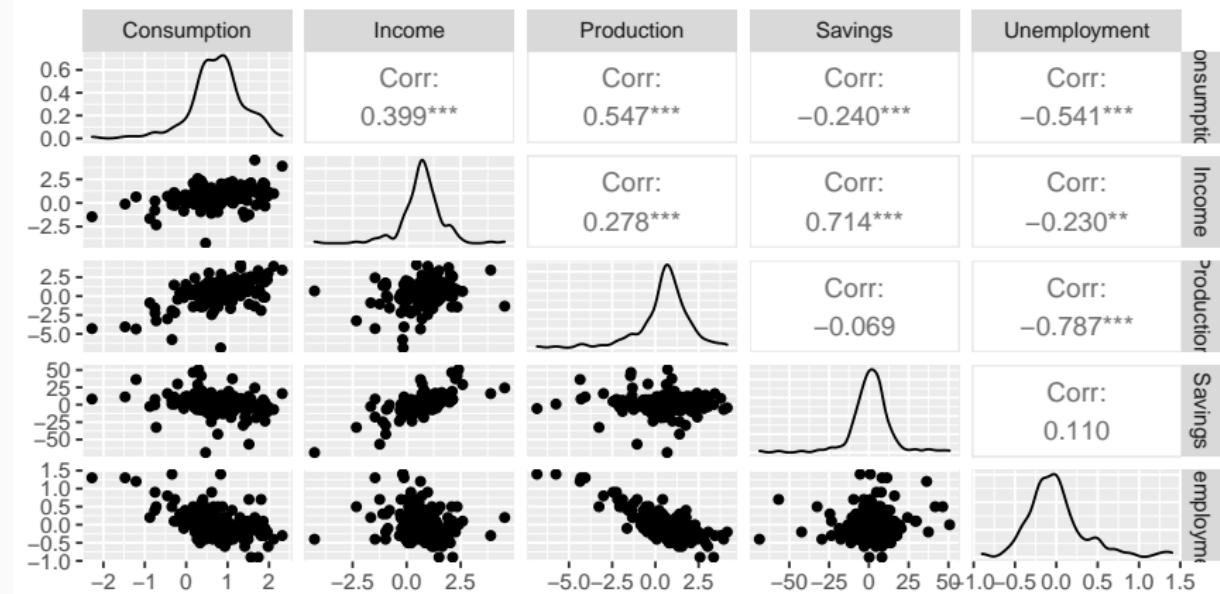
```
tslm(Consumption ~ Income, data=uschange) %>% summary

## 
## Call:
## tslm(formula = Consumption ~ Income, data = uschange)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.40845 -0.31816  0.02558  0.29978  1.45157 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.54510   0.05569   9.789 < 2e-16 ***
## Income       0.28060   0.04744   5.915 1.58e-08 ***
## ---    
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.6026 on 185 degrees of freedom
## Multiple R-squared:  0.159, Adjusted R-squared:  0.1545 
## F-statistic: 34.98 on 1 and 185 DF,  p-value: 1.577e-08
```

# Example: US consumption expenditure



# Example: US consumption expenditure

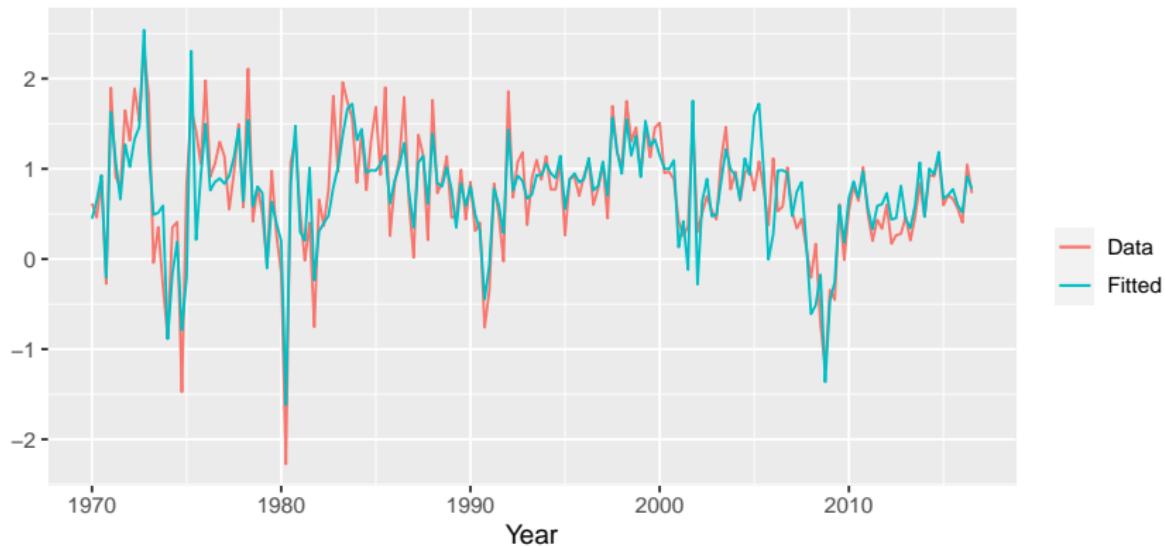


# Example: US consumption expenditure

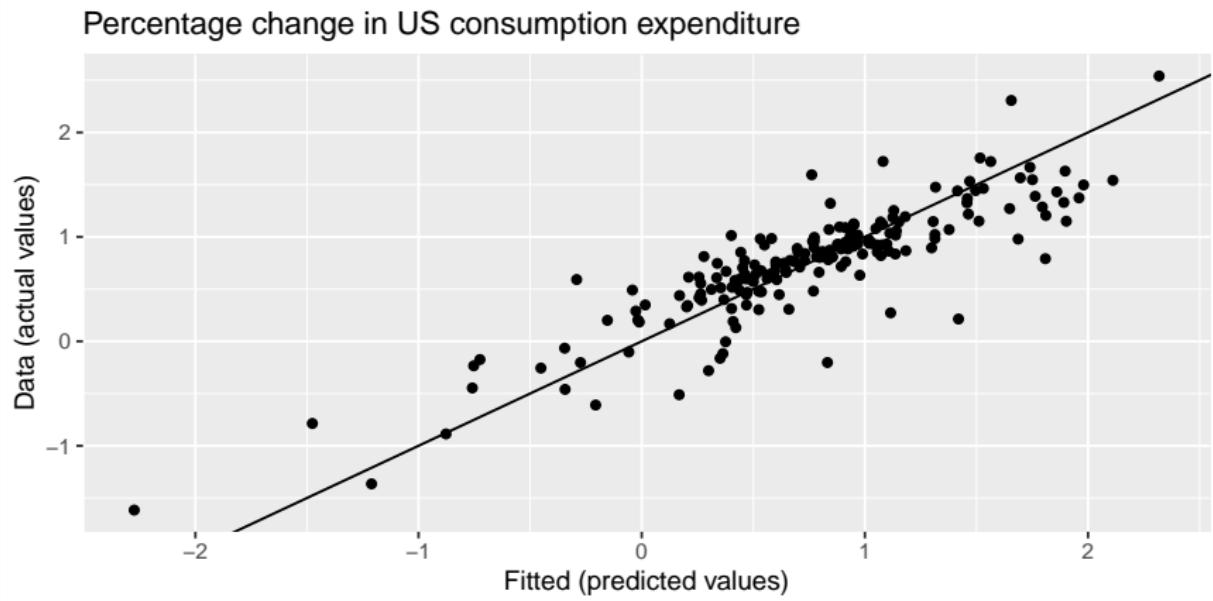
```
fit.consMR <- tslm(  
  Consumption ~ Income + Production + Unemployment + Savings,  
  data=uschange)  
summary(fit.consMR)  
  
##  
## Call:  
## tslm(formula = Consumption ~ Income + Production + Unemployment +  
##       Savings, data = uschange)  
##  
## Residuals:  
##      Min      1Q   Median      3Q     Max  
## -0.88296 -0.17638 -0.03679  0.15251  1.20553  
##  
##  
## Coefficients:  
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  0.26729   0.03721   7.184 1.68e-11 ***  
## Income       0.71449   0.04219  16.934  < 2e-16 ***  
## Production   0.04589   0.02588   1.773   0.0778 .  
## Unemployment -0.20477   0.10550  -1.941   0.0538 .  
## Savings      -0.04527   0.00278 -16.287  < 2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.3286 on 182 degrees of freedom  
## Multiple R-squared:  0.754, Adjusted R-squared:  0.7486  
## F-statistic: 139.5 on 4 and 182 DF, p-value: < 2.2e-16
```

# Example: US consumption expenditure

Percentage change in US consumption expenditure

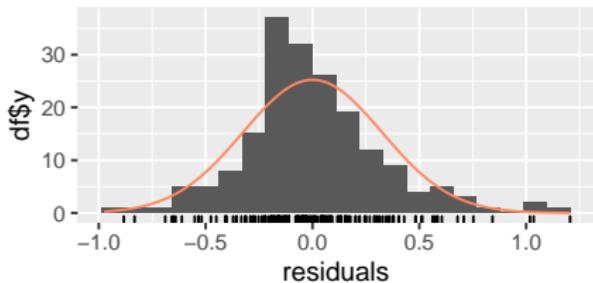
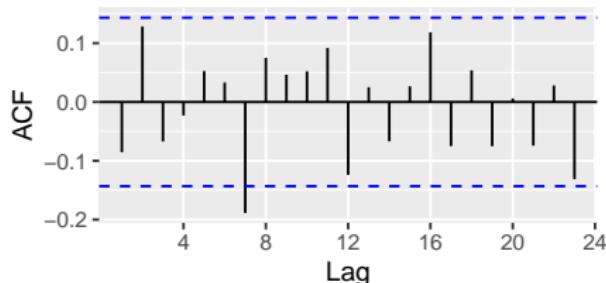
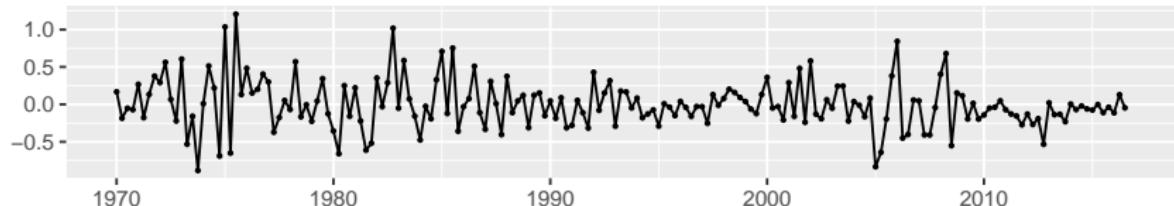


# Example: US consumption expenditure



# Example: US consumption expenditure

Residuals from Linear regression model



# Outline

- 1 The linear model with time series**
- 2 Residual diagnostics**
- 3 Some useful predictors for linear models**
- 4 Selecting predictors and forecast evaluation**
- 5 Forecasting with regression**
- 6 Matrix formulation**
- 7 Correlation, causation and forecasting**

# Multiple regression and forecasting

For forecasting purposes, we require the following assumptions:

- $\varepsilon_t$  are uncorrelated and zero mean
- $\varepsilon_t$  are uncorrelated with each  $x_{j,t}$ .

# Multiple regression and forecasting

For forecasting purposes, we require the following assumptions:

- $\varepsilon_t$  are uncorrelated and zero mean
- $\varepsilon_t$  are uncorrelated with each  $x_{j,t}$ .

It is **useful** to also have  $\varepsilon_t \sim N(0, \sigma^2)$  when producing prediction intervals or doing statistical tests.

# Residual plots

Useful for spotting outliers and whether the linear model was appropriate.

- Scatterplot of residuals  $\varepsilon_t$  against each predictor  $X_{j,t}$ .
- Scatterplot residuals against the fitted values  $\hat{y}_t$
- Expect to see scatterplots resembling a horizontal band with no values too far from the band and no patterns such as curvature or increasing spread.

# Residual patterns

- If a plot of the residuals vs any predictor in the model shows a pattern, then the relationship is nonlinear.
- If a plot of the residuals vs any predictor not in the model shows a pattern, then the predictor should be added to the model.
- If a plot of the residuals vs fitted values shows a pattern, then there is heteroscedasticity in the errors. (Could try a transformation.)

# Breusch-Godfrey test

OLS regression:

$$y_t = \beta_0 + \beta_1 x_{t,1} + \cdots + \beta_k x_{t,k} + u_t$$

Auxiliary regression:

$$\hat{u}_t = \beta_0 + \beta_1 x_{t,1} + \cdots + \beta_k x_{t,k} + \rho_1 \hat{u}_{t-1} + \cdots + \rho_p \hat{u}_{t-p} + \varepsilon_t$$

If  $R^2$  statistic is calculated for this model, then

$$(T - p)R^2 \sim \chi_p^2,$$

when there is no serial correlation up to lag  $p$ , and  $T$  = length of series.

- Breusch-Godfrey test better than Ljung-Box for regression models.

# US consumption again

```
##  
## Breusch-Godfrey test for serial correlation of order up to 8  
##  
## data: Residuals from Linear regression model  
## LM test = 14.874, df = 8, p-value = 0.06163
```

## If the model fails the Breusch-Godfrey test ...

- The forecasts are **not wrong**, but have higher variance than they need to.
- There is information in the residuals that we should exploit.
- This is done with a regression model with ARMA errors.

# Outline

- 1 The linear model with time series**
- 2 Residual diagnostics**
- 3 Some useful predictors for linear models**
- 4 Selecting predictors and forecast evaluation**
- 5 Forecasting with regression**
- 6 Matrix formulation**
- 7 Correlation, causation and forecasting**

# Trend

## Linear trend

$$x_t = t$$

- $t = 1, 2, \dots, T$
- Strong assumption that trend will continue.

# Dummy variables

If a categorical variable takes only two values (e.g., 'Yes' or 'No'), then an equivalent numerical variable can be constructed taking value 1 if yes and 0 if no. This is called a **dummy variable**.

	A	B
1	Yes	1
2	Yes	1
3	No	0
4	Yes	1
5	No	0
6	No	0
7	Yes	1
8	Yes	1
9	No	0
10	No	0
11	No	0
12	No	0
13	Yes	1
14	No	0
15		

# Dummy variables

If there are more than two categories, then the variable can be coded using several dummy variables (one fewer than the total number of categories).

	A	B	C	D	E
1	Monday	1	0	0	0
2	Tuesday	0	1	0	0
3	Wednesday	0	0	1	0
4	Thursday	0	0	0	1
5	Friday	0	0	0	0
6	Monday	1	0	0	0
7	Tuesday	0	1	0	0
8	Wednesday	0	0	1	0
9	Thursday	0	0	0	1
10	Friday	0	0	0	0
11	Monday	1	0	0	0
12	Tuesday	0	1	0	0
13	Wednesday	0	0	1	0
14	Thursday	0	0	0	1
15	Friday	0	0	0	0

# Beware of the dummy variable trap!

- Using one dummy for each category gives too many dummy variables!
- The regression will then be singular and inestimable.
- Either omit the constant, or omit the dummy for one category.
- The coefficients of the dummies are relative to the omitted category.

# Uses of dummy variables

## Seasonal dummies

- For quarterly data: use 3 dummies
- For monthly data: use 11 dummies
- For daily data: use 6 dummies
- What to do with weekly data?

# Uses of dummy variables

## Seasonal dummies

- For quarterly data: use 3 dummies
- For monthly data: use 11 dummies
- For daily data: use 6 dummies
- What to do with weekly data?

## Outliers

- If there is an outlier, you can use a dummy variable (taking value 1 for that observation and 0 elsewhere) to remove its effect.

# Uses of dummy variables

## Seasonal dummies

- For quarterly data: use 3 dummies
- For monthly data: use 11 dummies
- For daily data: use 6 dummies
- What to do with weekly data?

## Outliers

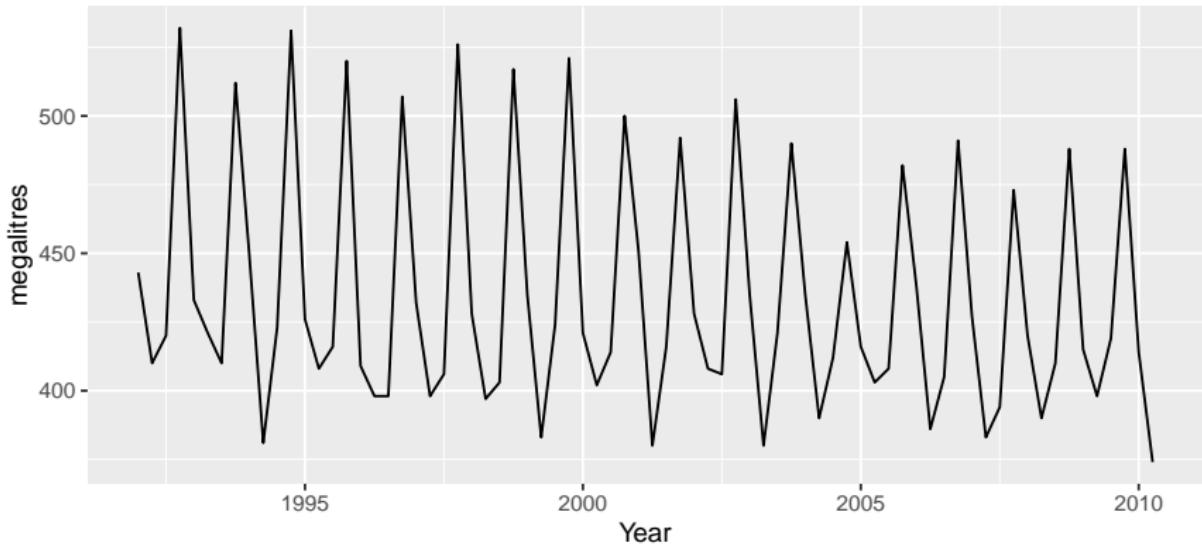
- If there is an outlier, you can use a dummy variable (taking value 1 for that observation and 0 elsewhere) to remove its effect.

## Public holidays

- For daily data: if it is a public holiday, dummy=1, otherwise dummy=0.

# Beer production revisited

Australian quarterly beer production



# Beer production revisited

## Regression model

$$y_t = \beta_0 + \beta_1 t + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + \varepsilon_t$$

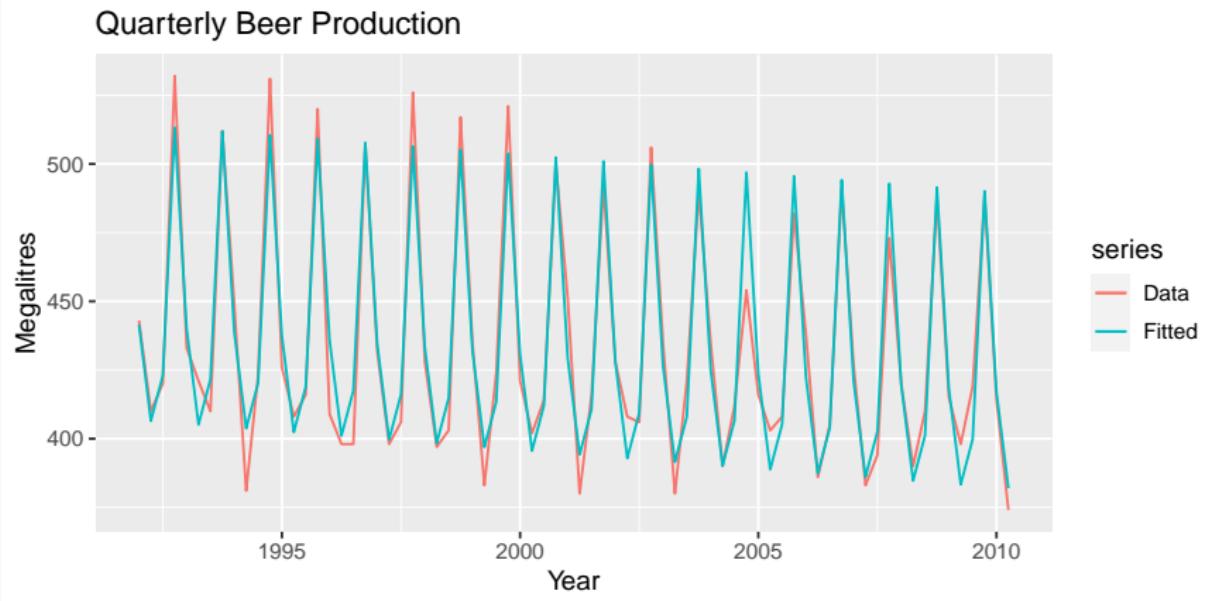
- $d_{i,t} = 1$  if  $t$  is quarter  $i$  and 0 otherwise.

# Beer production revisited

```
fit.beer <- tslm(beer ~ trend + season)
summary(fit.beer)

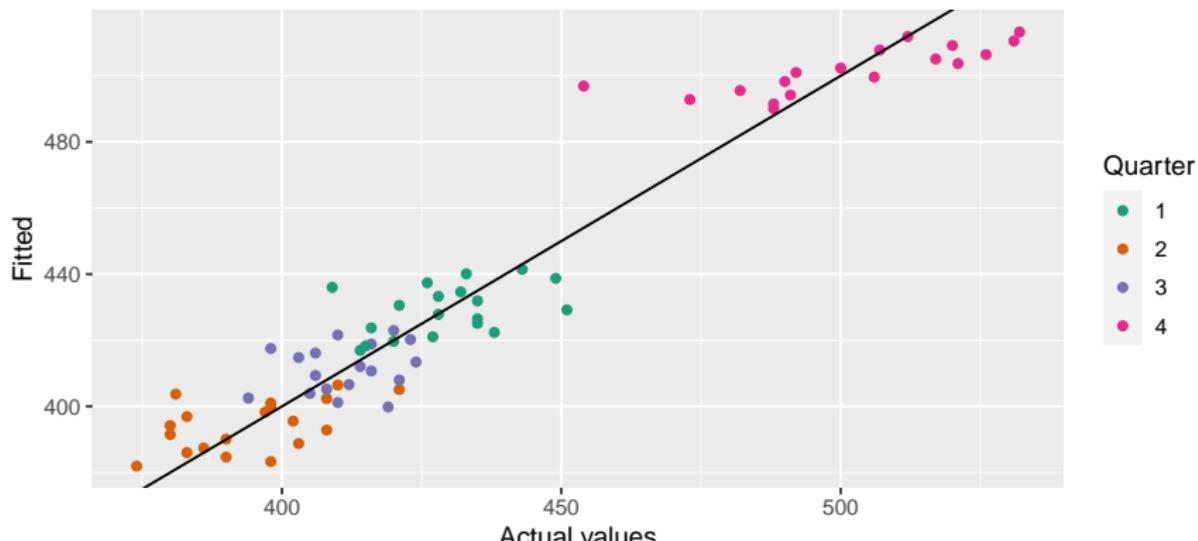
##
## Call:
## tslm(formula = beer ~ trend + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -42.903  -7.599  -0.459   7.991  21.789
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 441.80044    3.73353 118.333 < 2e-16 ***
## trend       -0.34027    0.06657 -5.111 2.73e-06 ***
## season2     -34.65973    3.96832 -8.734 9.10e-13 ***
## season3     -17.82164    4.02249 -4.430 3.45e-05 ***
## season4      72.79641    4.02305 18.095 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.23 on 69 degrees of freedom
## Multiple R-squared:  0.9243, Adjusted R-squared:  0.9199
## F-statistic: 210.7 on 4 and 69 DF,  p-value: < 2.2e-16
```

# Beer production revisited



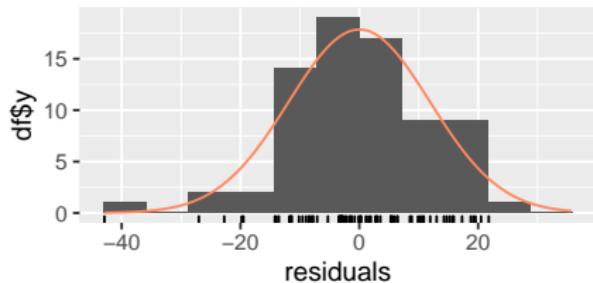
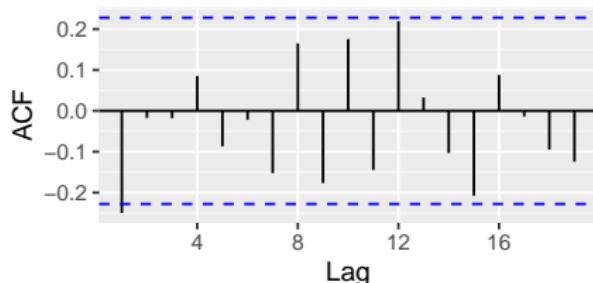
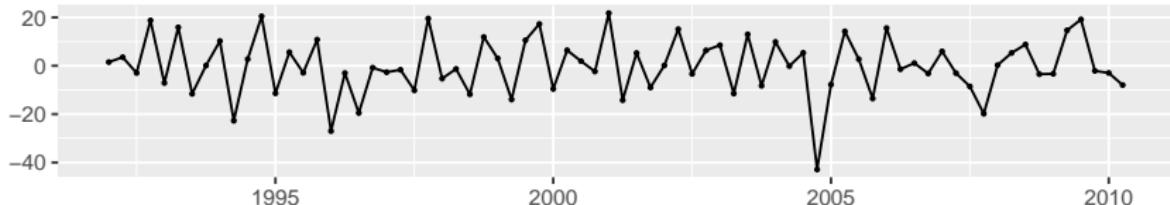
# Beer production revisited

Quarterly beer production



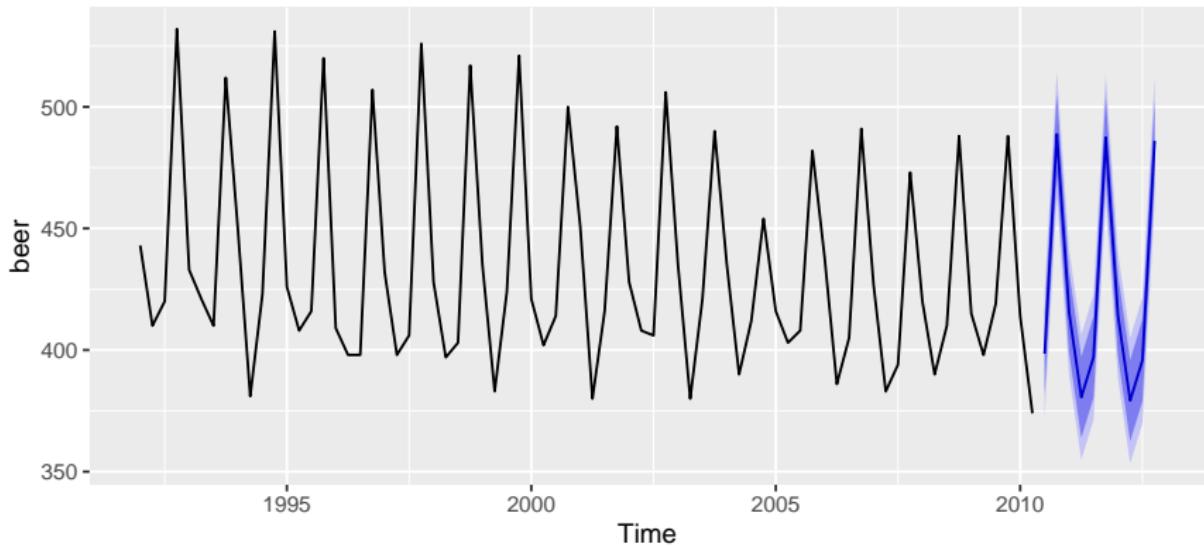
# Beer production revisited

Residuals from Linear regression model



# Beer production revisited

Forecasts from Linear regression model



# Fourier series

Periodic seasonality can be handled using pairs of Fourier terms:

$$s_k(t) = \sin\left(\frac{2\pi kt}{m}\right) \quad c_k(t) = \cos\left(\frac{2\pi kt}{m}\right)$$

$$y_t = a + bt + \sum_{k=1}^K [\alpha_k s_k(t) + \beta_k c_k(t)] + \varepsilon_t$$

- Every periodic function can be approximated by sums of sin and cos terms for large enough  $K$ .
- Choose  $K$  by minimizing AICc.
- Called “harmonic regression”

```
fit <- tslm(y ~ trend + fourier(y, K))
```

# Harmonic regression: beer production

```
fourier.beer <- tslm(beer ~ trend + fourier(beer, K=2))
summary(fourier.beer)

## 
## Call:
## tslm(formula = beer ~ trend + fourier(beer, K = 2))
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -42.903  -7.599  -0.459   7.991  21.789
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            446.87920   2.87321 155.533 < 2e-16 ***
## trend                  -0.34027   0.06657  -5.111 2.73e-06 ***
## fourier(beer, K = 2)S1-4  8.91082   2.01125   4.430 3.45e-05 ***
## fourier(beer, K = 2)C1-4  53.72807   2.01125  26.714 < 2e-16 ***
## fourier(beer, K = 2)C2-4 13.98958   1.42256   9.834 9.26e-15 ***
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 12.23 on 69 degrees of freedom
## Multiple R-squared:  0.9243, Adjusted R-squared:  0.9199 
## F-statistic: 210.7 on 4 and 69 DF,  p-value: < 2.2e-16
```

# Intervention variables

## Spikes

- Equivalent to a dummy variable for handling an outlier.

# Intervention variables

## Spikes

- Equivalent to a dummy variable for handling an outlier.

## Steps

- Variable takes value 0 before the intervention and 1 afterwards.

# Intervention variables

## Spikes

- Equivalent to a dummy variable for handling an outlier.

## Steps

- Variable takes value 0 before the intervention and 1 afterwards.

## Change of slope

- Variables take values 0 before the intervention and values  $\{1, 2, 3, \dots\}$  afterwards.

# Holidays

## For monthly data

- Christmas: always in December so part of monthly seasonal effect
- Easter: use a dummy variable  $v_t = 1$  if any part of Easter is in that month,  $v_t = 0$  otherwise.
- Ramadan and Chinese new year similar.

# Trading days

With monthly data, if the observations vary depending on how many different types of days in the month, then trading day predictors can be useful.

$z_1 = \# \text{ Mondays in month};$

$z_2 = \# \text{ Tuesdays in month};$

$\vdots$

$z_7 = \# \text{ Sundays in month}.$

## Distributed lags

Lagged values of a predictor.

Example:  $x$  is advertising which has a delayed effect

$x_1$  = advertising for previous month;

$x_2$  = advertising for two months previously;

$\vdots$

$x_m$  = advertising for  $m$  months previously.

# Nonlinear trend

Piecewise linear trend with bend at  $\tau$

$$x_{1,t} = t$$

$$x_{2,t} = \begin{cases} 0 & t < \tau \\ (t - \tau) & t \geq \tau \end{cases}$$

# Nonlinear trend

Piecewise linear trend with bend at  $\tau$

$$x_{1,t} = t$$

$$x_{2,t} = \begin{cases} 0 & t < \tau \\ (t - \tau) & t \geq \tau \end{cases}$$

Quadratic or higher order trend

$$x_{1,t} = t, \quad x_{2,t} = t^2, \quad \dots$$

# Nonlinear trend

Piecewise linear trend with bend at  $\tau$

$$x_{1,t} = t$$

$$x_{2,t} = \begin{cases} 0 & t < \tau \\ (t - \tau) & t \geq \tau \end{cases}$$

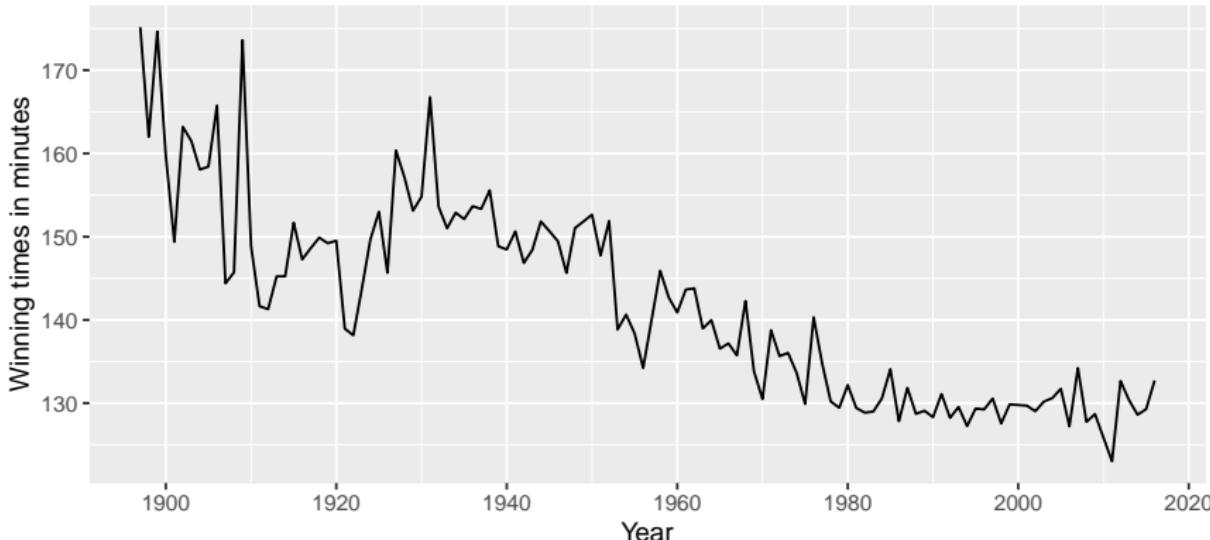
Quadratic or higher order trend

$$x_{1,t} = t, \quad x_{2,t} = t^2, \quad \dots$$

NOT RECOMMENDED!

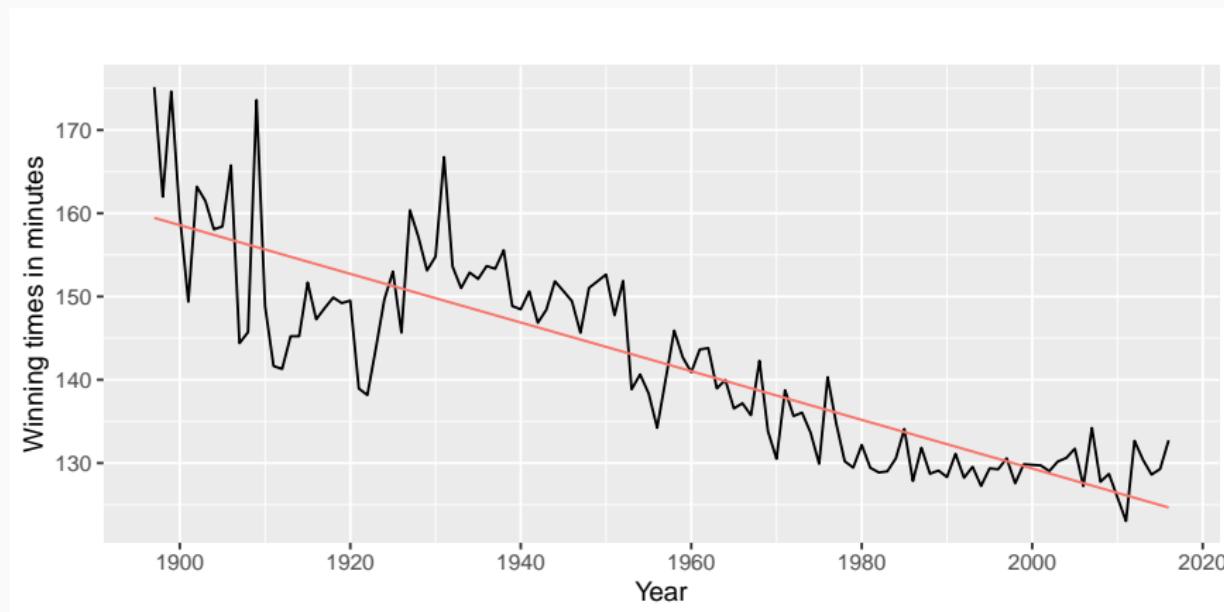
# Example: Boston marathon winning times

```
autoplot(marathon) +  
  xlab("Year") + ylab("Winning times in minutes")
```



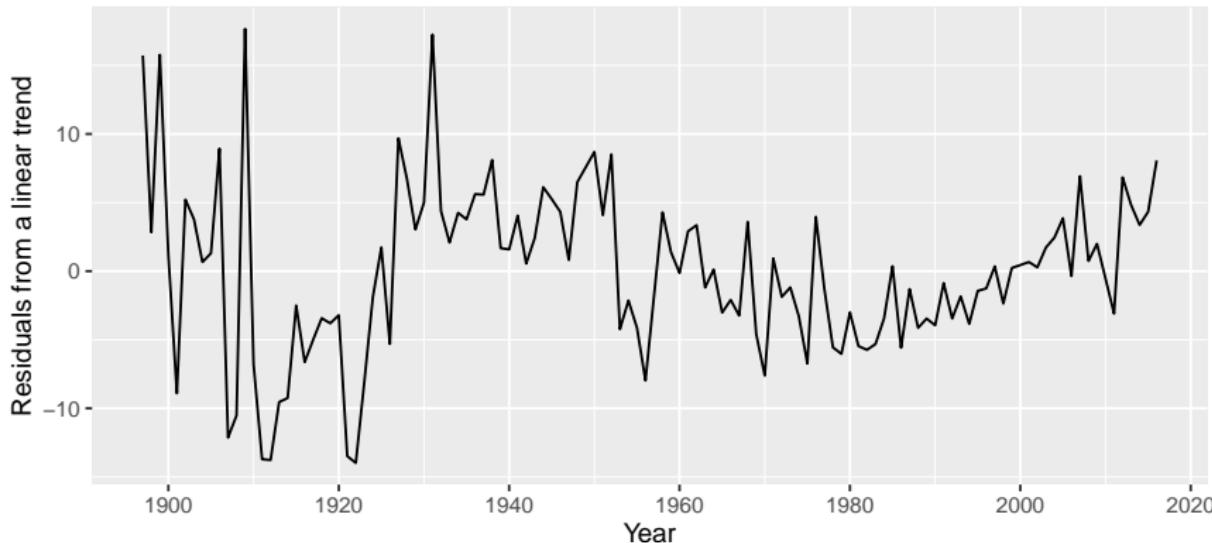
```
fit.lin <- tslm(marathon ~ trend)
```

# Example: Boston marathon winning times



# Example: Boston marathon winning times

```
autoplot(residuals(fit.lin)) +  
  xlab("Year") + ylab("Residuals from a linear trend")
```



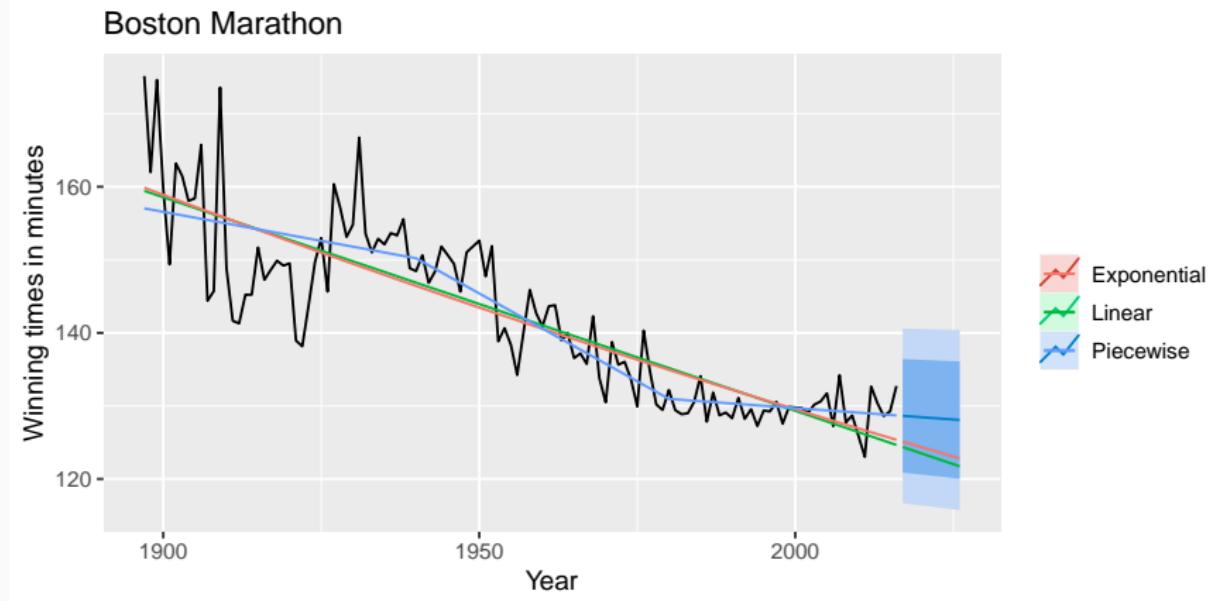
# Example: Boston marathon winning times

```
# Linear trend
fit.lin <- tslm(marathon ~ trend)
fcasts.lin <- forecast(fit.lin, h=10)

# Exponential trend
fit.exp <- tslm(marathon ~ trend, lambda = 0)
fcasts.exp <- forecast(fit.exp, h=10)

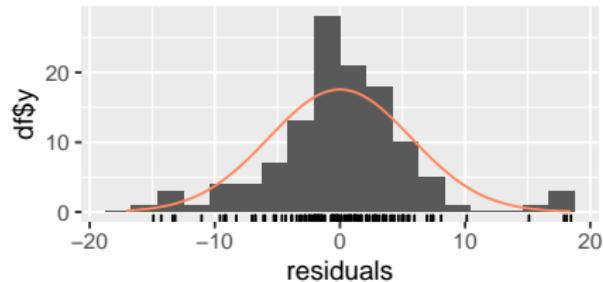
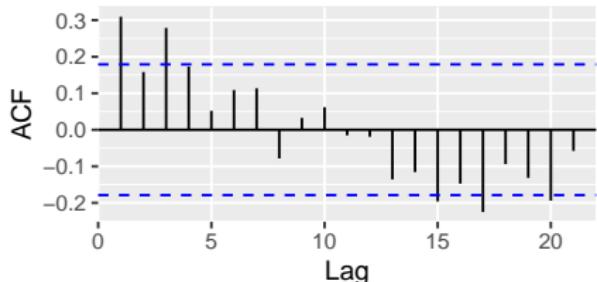
# Piecewise linear trend
t.break1 <- 1940
t.break2 <- 1980
t <- time(marathon)
t1 <- ts(pmax(0, t-t.break1), start=1897)
t2 <- ts(pmax(0, t-t.break2), start=1897)
fit.pw <- tslm(marathon ~ t + t1 + t2)
t.new <- t[length(t)] + seq(10)
t1.new <- t1[length(t1)] + seq(10)
t2.new <- t2[length(t2)] + seq(10)
newdata <- cbind(t=t.new, t1=t1.new, t2=t2.new) %>%
  as.data.frame
fcasts.pw <- forecast(fit.pw, newdata = newdata)
```

# Example: Boston marathon winning times

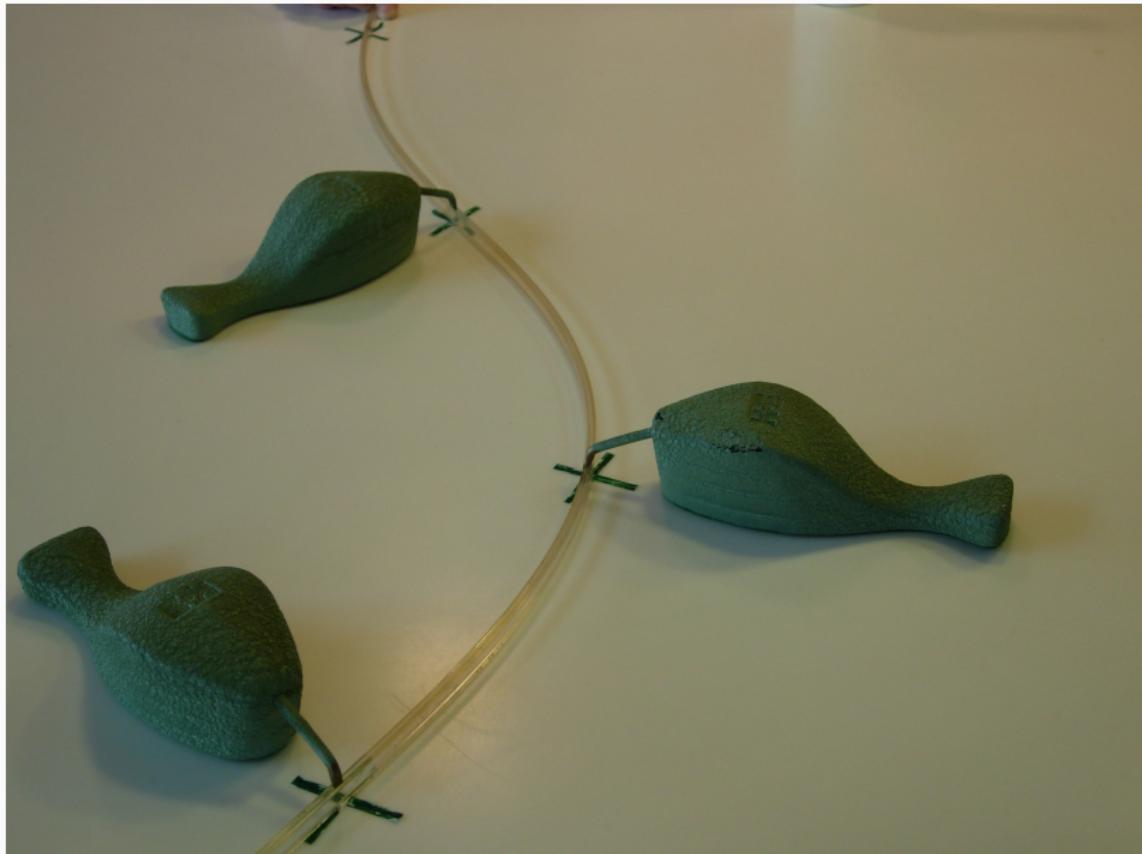


# Example: Boston marathon winning times

Residuals from Piecewise linear regression model



# Interpolating splines



# Interpolating splines



# Interpolating splines



# Interpolating splines

A spline is a continuous function  $f(x)$  interpolating all points  $(\kappa_j, y_j)$  for  $j = 1, \dots, K$  and consisting of polynomials between each consecutive pair of ‘knots’  $\kappa_j$  and  $\kappa_{j+1}$ .

# Interpolating splines

A spline is a continuous function  $f(x)$  interpolating all points  $(\kappa_j, y_j)$  for  $j = 1, \dots, K$  and consisting of polynomials between each consecutive pair of ‘knots’  $\kappa_j$  and  $\kappa_{j+1}$ .

- Parameters **constrained** so that  $f(x)$  is continuous.
- Further constraints imposed to give continuous derivatives.

# General linear regression splines

- Let  $\kappa_1 < \kappa_2 < \dots < \kappa_K$  be “knots” in interval  $(a, b)$ .
- Let  $x_1 = x$ ,  $x_j = (x - \kappa_{j-1})_+$  for  $j = 2, \dots, K + 1$ .
- Then the regression is piecewise linear with bends at the knots.

# General cubic splines

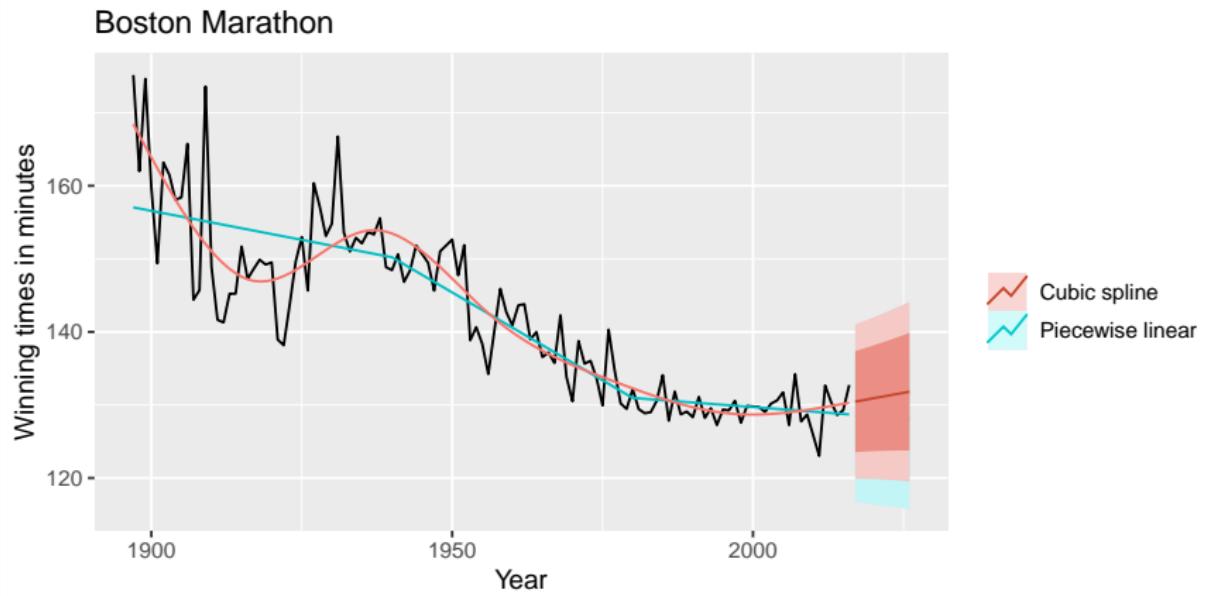
- Let  $x_1 = x$ ,  $x_2 = x^2$ ,  $x_3 = x^3$ ,  $x_j = (x - \kappa_{j-3})_+^3$  for  $j = 4, \dots, K + 3$ .
- Then the regression is piecewise cubic, but smooth at the knots.
- Choice of knots can be difficult and arbitrary.
- Automatic knot selection algorithms very slow.

# Example: Boston marathon winning times

```
# Spline trend
library(splines)
t <- time(marathon)
fit.splines <- lm(marathon ~ ns(t, df=6))
summary(fit.splines)

##
## Call:
## lm(formula = marathon ~ ns(t, df = 6))
##
## Residuals:
##       Min     1Q   Median     3Q    Max 
## -13.0028 -2.5722  0.0122  2.1242 21.5681 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 168.447    2.086   80.743 < 2e-16 ***
## ns(t, df = 6)1 -6.948    2.688   -2.584    0.011 *  
## ns(t, df = 6)2 -28.856    3.416   -8.448  1.16e-13 ***
## ns(t, df = 6)3 -35.081    3.045  -11.522 < 2e-16 *** 
## ns(t, df = 6)4 -32.563    2.652  -12.279 < 2e-16 *** 
## ns(t, df = 6)5 -64.847    5.322  -12.184 < 2e-16 *** 
## ns(t, df = 6)6 -21.002    2.403  -8.741  2.46e-14 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 4.834 on 113 degrees of freedom
## Multiple R-squared:  0.8418, Adjusted R-squared:  0.8334 
## F-statistic: 100.2 on 6 and 113 DF  p-value: < 2.2e-16
```

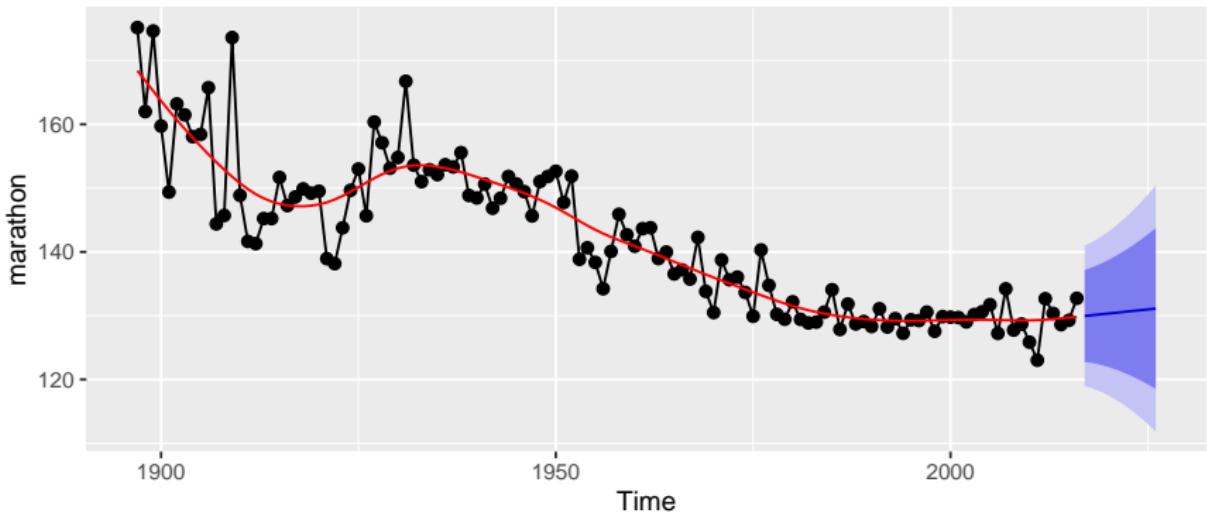
# Example: Boston marathon winning times



# splinef

A slightly different type of spline is provided by `splinef`

```
fc <- splinef(marathon)  
autoplot(fc)
```



## splinef

- Cubic **smoothing** splines (rather than cubic regression splines).
- Still piecewise cubic, but with many more knots (one at each observation).
- Coefficients constrained to prevent the curve becoming too “wiggly”.
- Degrees of freedom selected automatically.
- Equivalent to ARIMA(0,2,2) and Holt’s method.

# Outline

- 1 The linear model with time series**
- 2 Residual diagnostics**
- 3 Some useful predictors for linear models**
- 4 Selecting predictors and forecast evaluation**
- 5 Forecasting with regression**
- 6 Matrix formulation**
- 7 Correlation, causation and forecasting**

# Selecting predictors

- When there are many predictors, how should we choose which ones to use?
- We need a way of comparing two competing models.

# Selecting predictors

- When there are many predictors, how should we choose which ones to use?
- We need a way of comparing two competing models.

## What not to do!

- Plot  $y$  against a particular predictor ( $x_j$ ) and if it shows no noticeable relationship, drop it.
- Do a multiple linear regression on all the predictors and disregard all variables whose  $p$  values are greater than 0.05.
- Maximize  $R^2$  or minimize MSE

# Comparing regression models

Computer output for regression will always give the  $R^2$  value. This is a useful summary of the model.

- It is equal to the square of the correlation between  $y$  and  $\hat{y}$ .
- It is often called the “coefficient of determination”.
- It can also be calculated as follows:

$$R^2 = \frac{\sum(\hat{y}_t - \bar{y})^2}{\sum(y_t - \bar{y})^2}$$

- It is the proportion of variance accounted for (explained) by the predictors.

# Comparing regression models

However ...

- $R^2$  does not allow for “degrees of freedom”.
- Adding *any* variable tends to increase the value of  $R^2$ , even if that variable is irrelevant.

# Comparing regression models

However ...

- $R^2$  does not allow for “degrees of freedom”.
- Adding *any* variable tends to increase the value of  $R^2$ , even if that variable is irrelevant.

To overcome this problem, we can use *adjusted R<sup>2</sup>*:

$$\bar{R}^2 = 1 - (1 - R^2) \frac{T - 1}{T - k - 1}$$

where  $k$  = no. predictors and  $T$  = no. observations.

# Comparing regression models

However ...

- $R^2$  does not allow for “degrees of freedom”.
- Adding *any* variable tends to increase the value of  $R^2$ , even if that variable is irrelevant.

To overcome this problem, we can use *adjusted R<sup>2</sup>*:

$$\bar{R}^2 = 1 - (1 - R^2) \frac{T - 1}{T - k - 1}$$

where  $k$  = no. predictors and  $T$  = no. observations.

**Maximizing  $\bar{R}^2$  is equivalent to minimizing  $\hat{\sigma}^2$ .**

$$\hat{\sigma}^2 = \frac{1}{T - k - 1} \sum_{t=1}^T \varepsilon_t^2$$

# Cross-validation

## Cross-validation for regression

(Assuming future predictors are known)

- Select one observation for test set, and use *remaining* observations in training set. Compute error on test observation.
- Repeat using each possible observation as the test set.
- Compute accuracy measure over all errors.

# Cross-validation

## Traditional evaluation

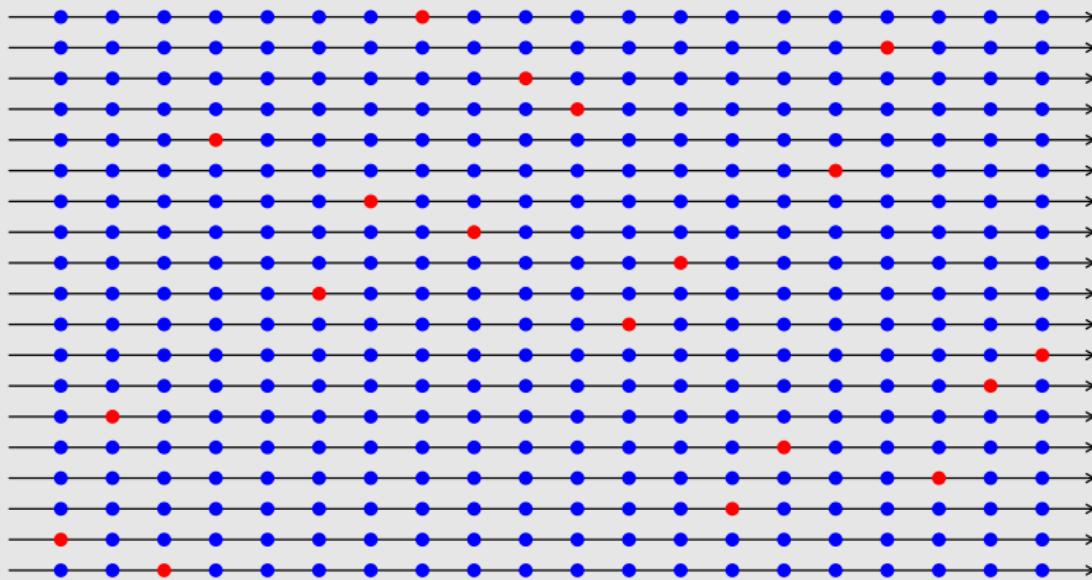


# Cross-validation

## Traditional evaluation



## Leave-one-out cross-validation



# Cross-validation

Leave-one-out cross-validation for regression can be carried out using the following steps.

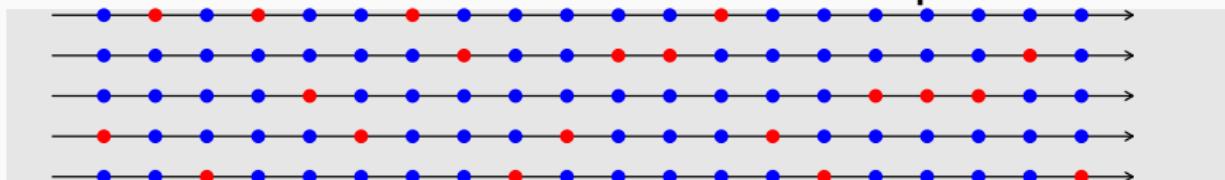
- Remove observation  $t$  from the data set, and fit the model using the remaining data. Then compute the error ( $e_t^* = y_t - \hat{y}_t$ ) for the omitted observation.
- Repeat step 1 for  $t = 1, \dots, T$ .
- Compute the MSE from  $\{e_1^*, \dots, e_T^*\}$ . We shall call this the CV.

The best model is the one with minimum CV.

# Cross-validation

## Five-fold cross-validation

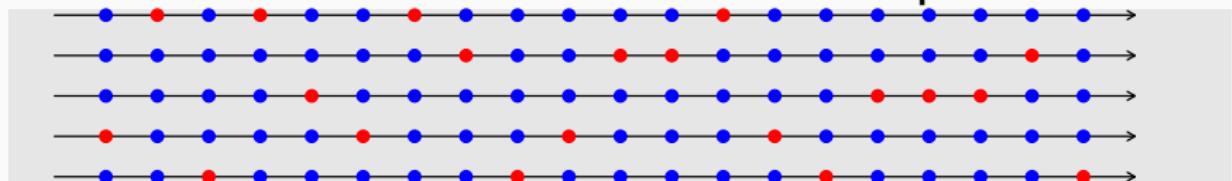
- 20 observations. 4 test observations per fold



# Cross-validation

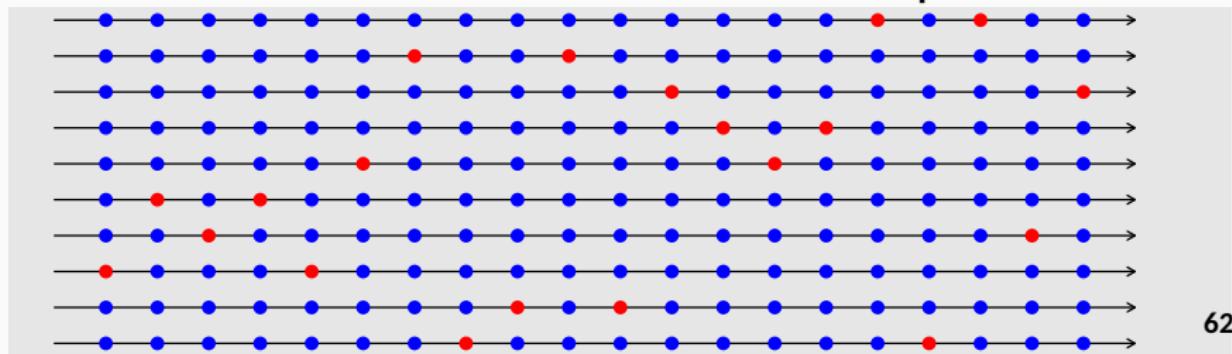
## Five-fold cross-validation

- 20 observations. 4 test observations per fold



## Ten-fold cross-validation

- 20 observations. 2 test observations per fold



# Cross-validation

## Ten-fold cross-validation

- Randomly split data into 10 parts.
- Select one part for test set, and use remaining parts as training set. Compute accuracy measures on test observations.
- Repeat for each of 10 parts
- Average over all measures.

## Akaike's Information Criterion

$$AIC = -2 \log(L) + 2(k + 2)$$

where  $L$  is the likelihood and  $k$  is the number of predictors in the model.

# Akaike's Information Criterion

$$AIC = -2 \log(L) + 2(k + 2)$$

where  $L$  is the likelihood and  $k$  is the number of predictors in the model.

- This is a penalized likelihood approach.
- Minimizing the AIC gives the best model for prediction.
- AIC penalizes terms more heavily than  $\bar{R}^2$ .
- Minimizing the AIC is asymptotically equivalent to minimizing MSE via leave-one-out cross-validation.

## Corrected AIC

For small values of  $T$ , the AIC tends to select too many predictors, and so a bias-corrected version of the AIC has been developed.

$$AIC_C = AIC + \frac{2(k + 2)(k + 3)}{T - k - 3}$$

As with the AIC, the  $AIC_C$  should be minimized.

# Bayesian Information Criterion

$$\text{BIC} = -2 \log(L) + (k + 2) \log(T)$$

where  $L$  is the likelihood and  $k$  is the number of predictors in the model.

# Bayesian Information Criterion

$$\text{BIC} = -2 \log(L) + (k + 2) \log(T)$$

where  $L$  is the likelihood and  $k$  is the number of predictors in the model.

- BIC penalizes terms more heavily than AIC
- Also called SBIC and SC.
- Minimizing BIC is asymptotically equivalent to leave- $v$ -out cross-validation when  $v = T[1 - 1/(\log(T) - 1)]$ .

# Choosing regression variables

## Best subsets regression

- Fit all possible regression models using one or more of the predictors.
- Choose the best model based on one of the measures of predictive ability (CV, AIC, AICc).

# Choosing regression variables

## Best subsets regression

- Fit all possible regression models using one or more of the predictors.
- Choose the best model based on one of the measures of predictive ability (CV, AIC, AICc).

## Warning!

- If there are a large number of predictors, this is not possible.
- For example, 44 predictors leads to 18 trillion possible models!

# Choosing regression variables

## Backwards stepwise regression

- Start with a model containing all variables.
- Try subtracting one variable at a time. Keep the model if it has lower CV or AICc.
- Iterate until no further improvement.

# Choosing regression variables

## Backwards stepwise regression

- Start with a model containing all variables.
- Try subtracting one variable at a time. Keep the model if it has lower CV or AICc.
- Iterate until no further improvement.

## Notes

- Stepwise regression is not guaranteed to lead to the best possible model.
- Inference on coefficients of final model will be wrong.

# Cross-validation

```
tslm(Consumption ~ Income + Production + Unemployment + Savings,  
  data=uschange) %>% CV()  
  
##          CV        AIC       AICc       BIC       AdjR2  
## 0.1163477 -409.2980298 -408.8313631 -389.9113781 0.7485856  
  
tslm(Consumption ~ Income + Production + Unemployment,  
  data=uschange) %>% CV()  
  
##          CV        AIC       AICc       BIC       AdjR2  
## 0.2776928 -243.1635677 -242.8320760 -227.0080246 0.3855438  
  
tslm(Consumption ~ Income + Production + Savings,  
  data=uschange) %>% CV()  
  
##          CV        AIC       AICc       BIC       AdjR2  
## 0.1178681 -407.4669279 -407.1354362 -391.3113848 0.7447840  
  
tslm(Consumption ~ Income + Unemployment + Savings,  
  data=uschange) %>% CV()  
  
##          CV        AIC       AICc       BIC       AdjR2  
## 0.1160223 -408.0941325 -407.7626408 -391.9385894 0.7456386  
  
tslm(Consumption ~ Production + Unemployment + Savings,  
  data=uschange) %>% CV()  
  
##          CV        AIC       AICc       BIC       AdjR2  
## 0.2927095 -234.3734580 -234.0419663 -218.2179149 0.3559711
```

# Outline

- 1 The linear model with time series
- 2 Residual diagnostics
- 3 Some useful predictors for linear models
- 4 Selecting predictors and forecast evaluation
- 5 Forecasting with regression
- 6 Matrix formulation
- 7 Correlation, causation and forecasting

# Ex-ante versus ex-post forecasts

- *Ex ante* forecasts are made using only information available in advance.
  - require forecasts of predictors
- *Ex post* forecasts are made using later information on the predictors.
  - useful for studying behaviour of forecasting models.
- trend, seasonal and calendar variables are all known in advance, so these don't need to be forecast.

# Scenario based forecasting

- Assumes possible scenarios for the predictor variables
- Prediction intervals for scenario based forecasts do not include the uncertainty associated with the future values of the predictor variables.

# Building a predictive regression model

- If getting forecasts of predictors is difficult, you can use lagged predictors instead.

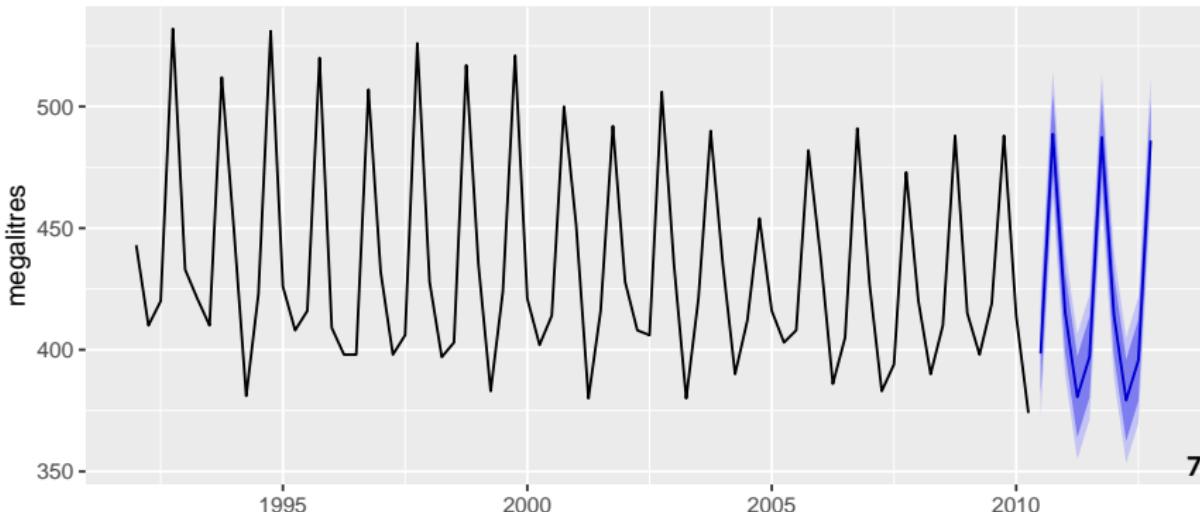
$$y_t = \beta_0 + \beta_1 x_{1,t-h} + \cdots + \beta_k x_{k,t-h} + \varepsilon_t$$

- A different model for each forecast horizon  $h$ .

# Beer production

```
beer2 <- window(ausbeer, start=1992)
fit.beer <- tslm(beer2 ~ trend + season)
fcast <- forecast(fit.beer)
autoplot(fcast) +
  ggtitle("Forecasts of beer production using regression") +
  xlab("Year") + ylab("megalitres")
```

Forecasts of beer production using regression

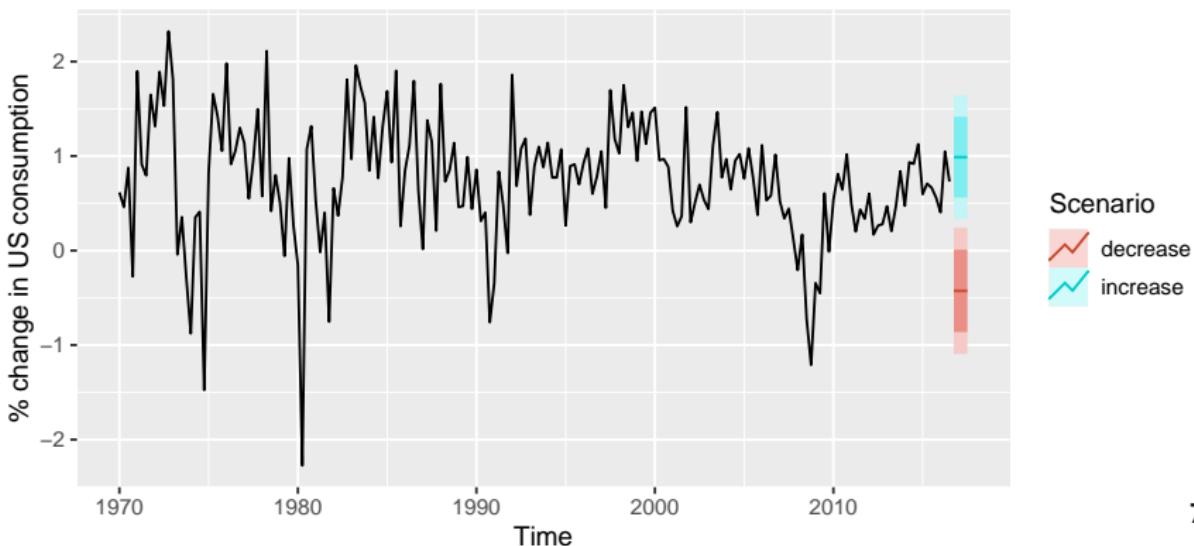


# US Consumption

```
fit.consBest <- tslm(  
  Consumption ~ Income + Savings + Unemployment,  
  data = uschange)  
h <- 4  
newdata <- data.frame(  
  Income = c(1, 1, 1, 1),  
  Savings = c(0.5, 0.5, 0.5, 0.5),  
  Unemployment = c(0, 0, 0, 0))  
fcast.up <- forecast(fit.consBest, newdata = newdata)  
newdata <- data.frame(  
  Income = rep(-1, h),  
  Savings = rep(-0.5, h),  
  Unemployment = rep(0, h))  
fcast.down <- forecast(fit.consBest, newdata = newdata)
```

# US Consumption

```
autoplot(uschange[, 1]) +  
  ylab("% change in US consumption") +  
  autolayer(fcast.up, PI = TRUE, series = "increase") +  
  autolayer(fcast.down, PI = TRUE, series = "decrease") +  
  guides(colour = guide_legend(title = "Scenario"))
```



# Outline

- 1 The linear model with time series**
- 2 Residual diagnostics**
- 3 Some useful predictors for linear models**
- 4 Selecting predictors and forecast evaluation**
- 5 Forecasting with regression**
- 6 Matrix formulation**
- 7 Correlation, causation and forecasting**

# Matrix formulation

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t.$$

# Matrix formulation

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t.$$

Let  $\mathbf{y} = (y_1, \dots, y_T)'$ ,  $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_T)'$ ,

$\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_k)'$  and

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{2,1} & \dots & x_{k,1} \\ 1 & x_{1,2} & x_{2,2} & \dots & x_{k,2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1,T} & x_{2,T} & \dots & x_{k,T} \end{bmatrix}.$$

# Matrix formulation

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t.$$

Let  $\mathbf{y} = (y_1, \dots, y_T)'$ ,  $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_T)'$ ,

$\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_k)'$  and

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{2,1} & \dots & x_{k,1} \\ 1 & x_{1,2} & x_{2,2} & \dots & x_{k,2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1,T} & x_{2,T} & \dots & x_{k,T} \end{bmatrix}.$$

Then

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

# Matrix formulation

## Least squares estimation

Minimize:  $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$

# Matrix formulation

## Least squares estimation

Minimize:  $(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$

Differentiate wrt  $\beta$  gives

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

# Matrix formulation

## Least squares estimation

Minimize:  $(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$

Differentiate wrt  $\beta$  gives

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

(The “normal equation”.)

# Matrix formulation

## Least squares estimation

Minimize:  $(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$

Differentiate wrt  $\beta$  gives

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

(The “normal equation”.)

$$\hat{\sigma}^2 = \frac{1}{T - k - 1}(\mathbf{y} - \mathbf{X}\hat{\beta})'(\mathbf{y} - \mathbf{X}\hat{\beta})$$

**Note:** If you fall for the dummy variable trap,  $(\mathbf{X}'\mathbf{X})$  is a singular matrix.

# Likelihood

If the errors are iid and normally distributed, then

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}).$$

# Likelihood

If the errors are iid and normally distributed, then

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}).$$

So the likelihood is

$$L = \frac{1}{\sigma^T (2\pi)^{T/2}} \exp \left( -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right)$$

# Likelihood

If the errors are iid and normally distributed, then

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}).$$

So the likelihood is

$$L = \frac{1}{\sigma^T (2\pi)^{T/2}} \exp \left( -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right)$$

which is maximized when  $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$  is minimized.

# Likelihood

If the errors are iid and normally distributed, then

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}).$$

So the likelihood is

$$L = \frac{1}{\sigma^T (2\pi)^{T/2}} \exp \left( -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right)$$

which is maximized when  $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$  is minimized.

So MLE = OLS.

# Multiple regression forecasts

## Optimal forecasts

$$\hat{y}^* = E(y^* | \mathbf{y}, \mathbf{X}, \mathbf{x}^*) = \mathbf{x}^* \hat{\boldsymbol{\beta}} = \mathbf{x}^* (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{y}$$

where  $\mathbf{x}^*$  is a row vector containing the values of the predictors for the forecasts (in the same format as  $\mathbf{X}$ ).

# Multiple regression forecasts

## Optimal forecasts

$$\hat{y}^* = E(y^* | \mathbf{y}, \mathbf{X}, \mathbf{x}^*) = \mathbf{x}^* \hat{\boldsymbol{\beta}} = \mathbf{x}^* (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{y}$$

where  $\mathbf{x}^*$  is a row vector containing the values of the predictors for the forecasts (in the same format as  $\mathbf{X}$ ).

## Forecast variance

$$\text{Var}(y^* | \mathbf{X}, \mathbf{x}^*) = \sigma^2 [1 + \mathbf{x}^* (\mathbf{X}' \mathbf{X})^{-1} (\mathbf{x}^*)']$$

# Multiple regression forecasts

## Optimal forecasts

$$\hat{y}^* = E(y^* | \mathbf{y}, \mathbf{X}, \mathbf{x}^*) = \mathbf{x}^* \hat{\boldsymbol{\beta}} = \mathbf{x}^* (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{y}$$

where  $\mathbf{x}^*$  is a row vector containing the values of the predictors for the forecasts (in the same format as  $\mathbf{X}$ ).

## Forecast variance

$$\text{Var}(y^* | \mathbf{X}, \mathbf{x}^*) = \sigma^2 [1 + \mathbf{x}^* (\mathbf{X}' \mathbf{X})^{-1} (\mathbf{x}^*)']$$

- This ignores any errors in  $\mathbf{x}^*$ .
- 95% prediction intervals assuming normal errors:

$$\hat{y}^* \pm 1.96 \sqrt{\text{Var}(y^* | \mathbf{X}, \mathbf{x}^*)}.$$

# Multiple regression forecasts

## Fitted values

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}\mathbf{y}$$

where  $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$  is the “hat matrix”.

# Multiple regression forecasts

## Fitted values

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}\mathbf{y}$$

where  $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$  is the “hat matrix”.

## Leave-one-out residuals

Let  $h_1, \dots, h_T$  be the diagonal values of  $\mathbf{H}$ , then the cross-validation statistic is

$$CV = \frac{1}{T} \sum_{t=1}^T [e_t / (1 - h_t)]^2,$$

where  $e_t$  is the residual obtained from fitting the model to all  $T$  observations.

# Outline

- 1 The linear model with time series**
- 2 Residual diagnostics**
- 3 Some useful predictors for linear models**
- 4 Selecting predictors and forecast evaluation**
- 5 Forecasting with regression**
- 6 Matrix formulation**
- 7 Correlation, causation and forecasting**

# Correlation is not causation

- When  $x$  is useful for predicting  $y$ , it is not necessarily causing  $y$ .
- e.g., predict number of drownings  $y$  using number of ice-creams sold  $x$ .
- Correlations are useful for forecasting, even when there is no causality.
- Better models usually involve causal relationships (e.g., temperature  $x$  and people  $z$  to predict drownings  $y$ ).

# Multicollinearity

In regression analysis, multicollinearity occurs when:

- Two predictors are highly correlated (i.e., the correlation between them is close to  $\pm 1$ ).
- A linear combination of some of the predictors is highly correlated with another predictor.
- A linear combination of one subset of predictors is highly correlated with a linear combination of another subset of predictors.

# Multicollinearity

If multicollinearity exists...

- the numerical estimates of coefficients may be wrong (worse in Excel than in a statistics package)
- don't rely on the  $p$ -values to determine significance.
- there is no problem with model *predictions* provided the predictors used for forecasting are within the range used for fitting.
- omitting variables can help.
- combining variables can help.

# Outliers and influential observations

## Things to watch for

- *Outliers*: observations that produce large residuals.
- *Influential observations*: removing them would markedly change the coefficients. (Often outliers in the  $x$  variable).
- *Lurking variable*: a predictor not included in the regression but which has an important effect on the response.
- Points should not normally be removed without a good explanation of why they are different.



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# STU33010: Forecasting

Ch6. Time series decomposition

[OTexts.org/fpp2/](https://OTexts.org/fpp2/)

# Outline

- 1 Time series components**
- 2 Seasonal adjustment**
- 3 X-11 decomposition**
- 4 SEATS decomposition**
- 5 STL decomposition**
- 6 Forecasting and decomposition**

# Time series patterns

## Recall

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

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

**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).

# Time series decomposition

$$y_t = f(S_t, T_t, R_t)$$

where  $y_t$  = data at period  $t$

$T_t$  = trend-cycle component at period  $t$

$S_t$  = seasonal component at period  $t$

$R_t$  = remainder component at period  $t$

# Time series decomposition

$$y_t = f(S_t, T_t, R_t)$$

where  $y_t$  = data at period  $t$

$T_t$  = trend-cycle component at period  $t$

$S_t$  = seasonal component at period  $t$

$R_t$  = remainder component at period  $t$

**Additive decomposition:**  $y_t = S_t + T_t + R_t.$

**Multiplicative decomposition:**  $y_t = S_t \times T_t \times R_t.$

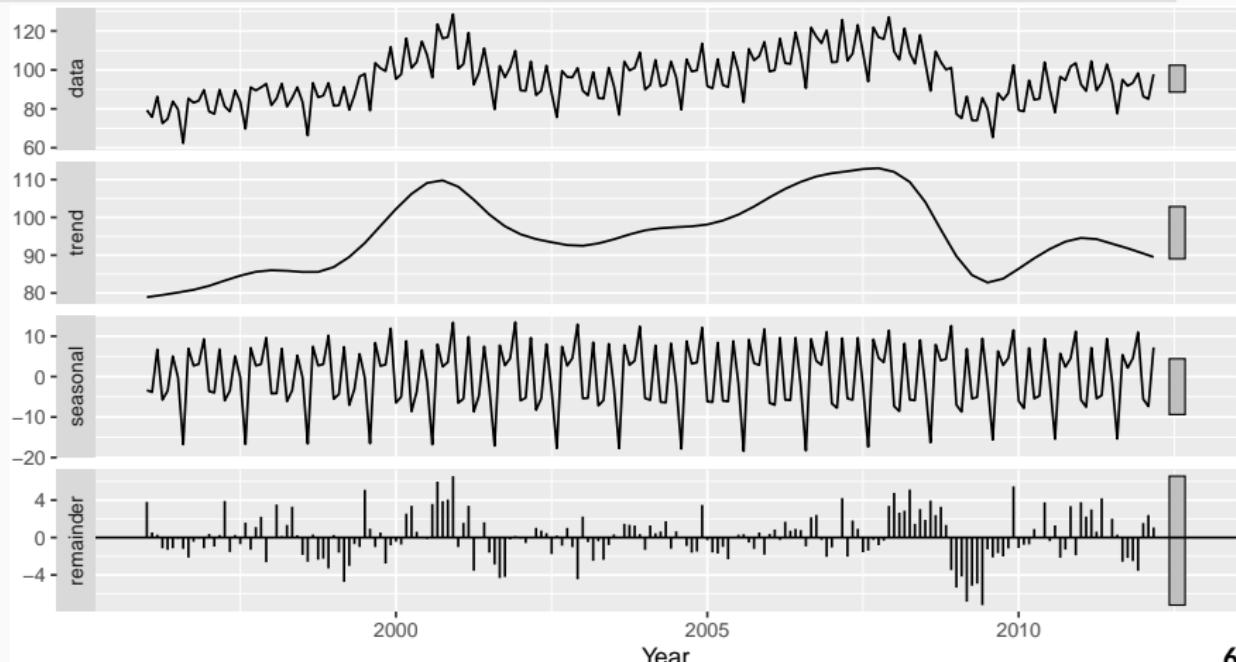
# Time series decomposition

- Additive model appropriate if magnitude of seasonal fluctuations does not vary with level.
- If seasonal are proportional to level of series, then multiplicative model appropriate.
- Multiplicative decomposition more prevalent with economic series
- Alternative: use a Box-Cox transformation, and then use additive decomposition.
- Logs turn multiplicative relationship into an additive relationship:

$$y_t = S_t \times T_t \times E_t \quad \Rightarrow \quad \log y_t = \log S_t + \log T_t + \log E_t.$$

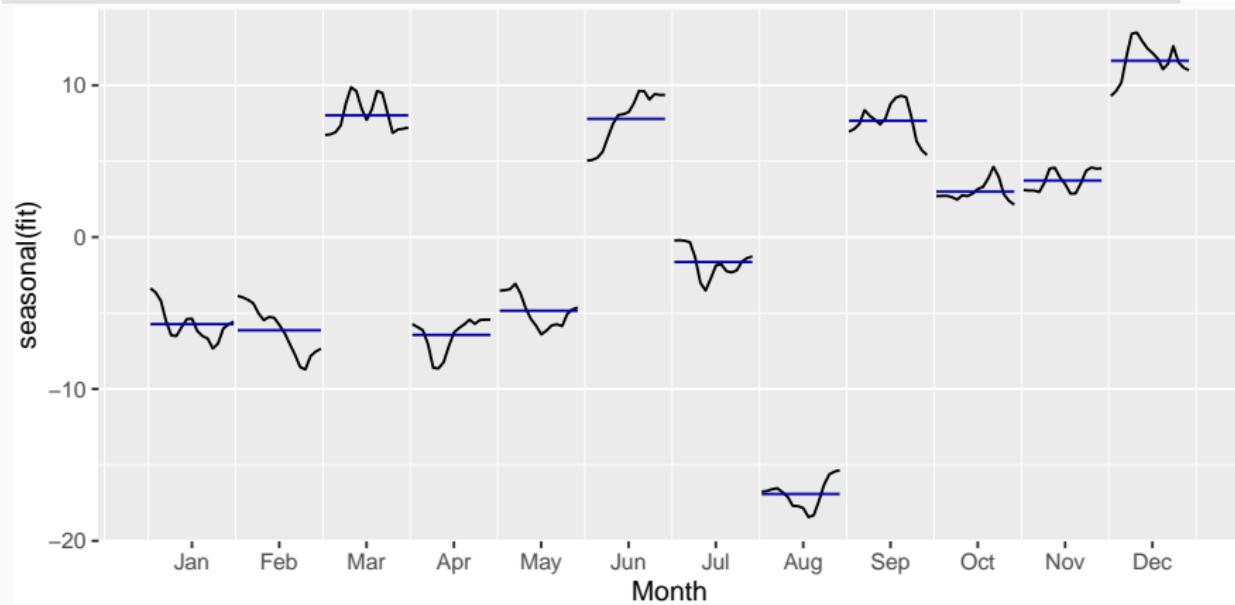
# Euro electrical equipment

```
fit <- stl(elecequip, s.window=7)  
autoplot(fit) + xlab("Year")
```



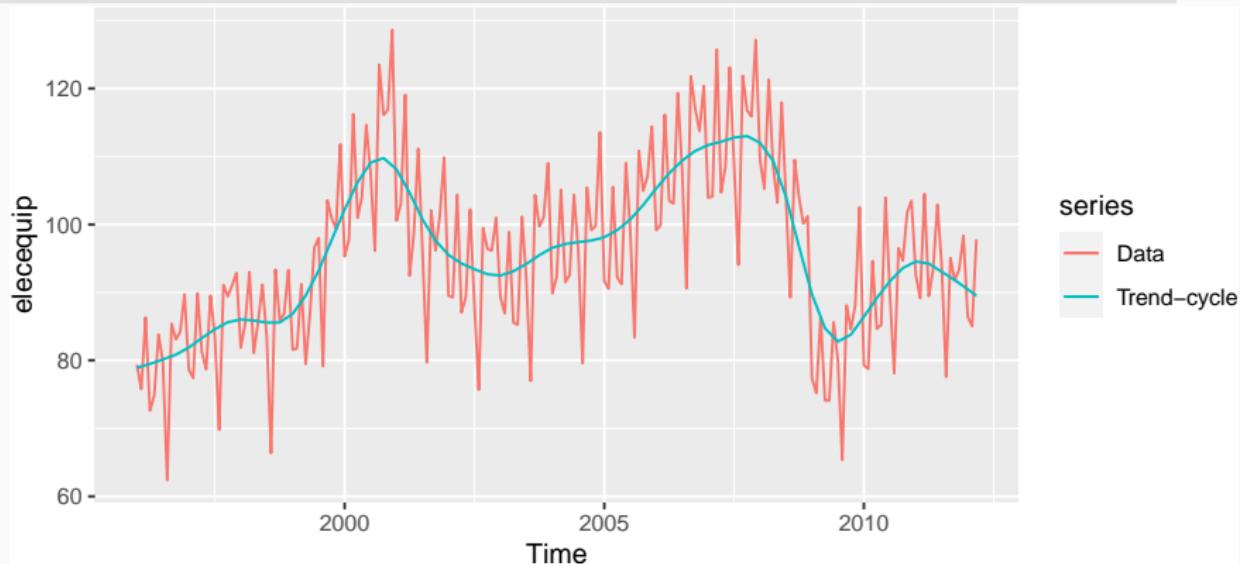
# Euro electrical equipment

```
ggsubseriesplot(seasonal(fit))
```



# Euro electrical equipment

```
autoplot(elecequip, series="Data") +  
  autolayer(trendcycle(fit), series="Trend-cycle")
```



# Helper functions

- `seasonal()` extracts the seasonal component
- `trendcycle()` extracts the trend-cycle component
- `remainder()` extracts the remainder component.
- `seasadj()` returns the seasonally adjusted series.

## Your turn

Repeat the decomposition using

```
elecequip %>%  
  stl(s.window=7, t.window=11) %>%  
  autoplot()
```

What happens as you change s.window and t.window?

# Outline

- 1 Time series components**
- 2 Seasonal adjustment**
- 3 X-11 decomposition**
- 4 SEATS decomposition**
- 5 STL decomposition**
- 6 Forecasting and decomposition**

# Seasonal adjustment

- Useful by-product of decomposition: an easy way to calculate seasonally adjusted data.
- Additive decomposition: seasonally adjusted data given by

$$y_t - S_t = T_t + R_t$$

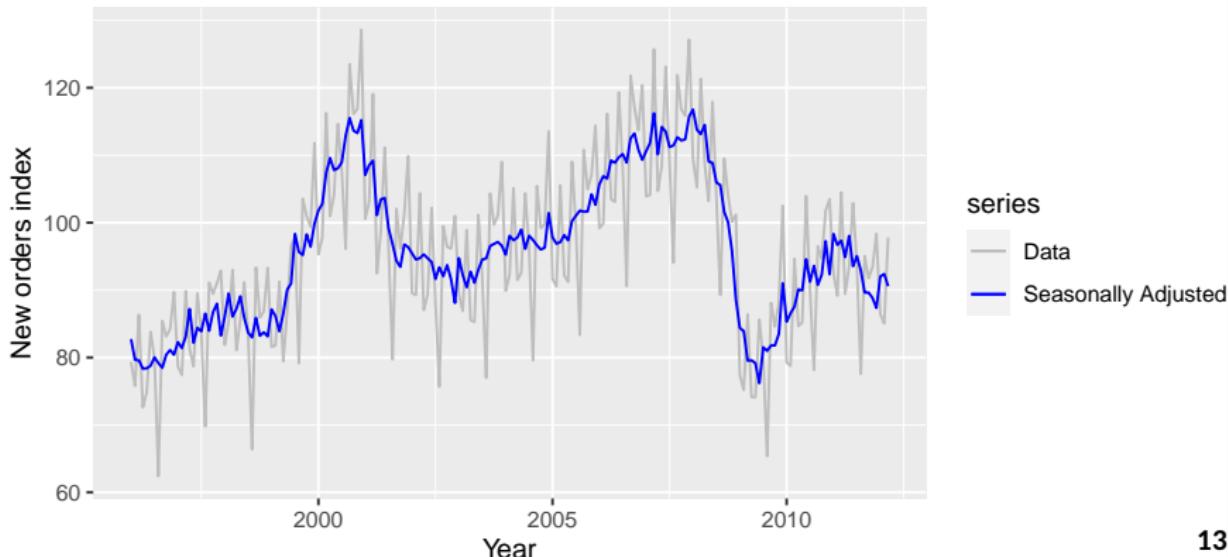
- Multiplicative decomposition: seasonally adjusted data given by

$$y_t/S_t = T_t \times R_t$$

# Euro electrical equipment

```
fit <- stl(elecequip, s.window=7)  
autoplot(elecequip, series="Data") +  
  autolayer(seasadj(fit), series="Seasonally Adjusted")
```

Electrical equipment manufacturing (Euro area)



# Seasonal adjustment

- We use estimates of  $S$  based on past values to seasonally adjust a current value.
- Seasonally adjusted series reflect **remainders** as well as **trend**. Therefore they are not “smooth”" and “downturns”" or “upturns” can be misleading.
- It is better to use the trend-cycle component to look for turning points.

# The ABS stuff-up

NEWS 

LOCATION:  
Clayton, Vic [Change](#)



Just In Australia World Business Sport Analysis & Opinion Fact Check Programs

**BREAKING NEWS** Police arrest man in connection with stabbing death of 17-year-old Masa Vukotic in M

 Print  Email  Facebook  Twitter  More

## Treasurer Joe Hockey calls for answers over Australian Bureau of Statistics jobs data

By Michael Vincent and Simon Frazer

Updated 9 Oct 2014, 12:17pm

**Federal Treasurer Joe Hockey says he wants answers to the problems the Australian Bureau of Statistics (ABS) has had with unemployment figures.**

Mr Hockey, who is in the US to discuss Australia's G20 agenda, said last month's unemployment figures were "extraordinary".

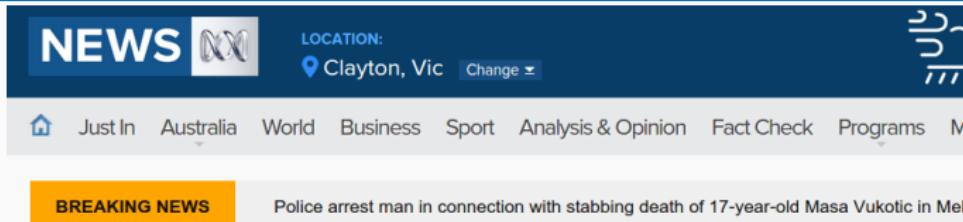
The rate was 6.1 per cent after jumping to a 12-year high of 6.4 per cent the previous month.

The ABS has now taken the rare step of abandoning seasonal adjustment for its latest employment data



**PHOTO:** Joe Hockey says he is unhappy with the volatility of ABS unemployment figures. (AAP: Alan Porritt)

# The ABS stuff-up



NEWS 

LOCATION: Clayton, Vic [Change](#)

Just In Australia World Business Sport Analysis & Opinion Fact Check Programs M

**BREAKING NEWS** Police arrest man in connection with stabbing death of 17-year-old Masa Vukotic in Mel

## ABS abandons seasonal adjustment for latest jobs data

By business reporter Michael Janda

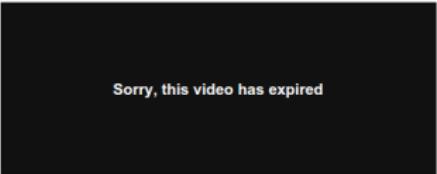
Updated 8 Oct 2014, 4:19pm

**The Australian Bureau of Statistics is taking the rare step of abandoning seasonal adjustment for its latest employment data.**

The ABS uses seasonal adjustment, based on historical experience, to account for the normal variation between hiring and firing patterns between different months.

However, after a winter where the seasonally adjusted unemployment rate swung wildly from 6.1 to 6.4 and back to 6.1 per cent, [the bureau released a statement](#) saying it will not adjust the original figure for September for seasonal factors.

It will also reset the seasonal adjustment for July and August to one, meaning that these months will



Sorry, this video has expired

**VIDEO:** Westpac chief economist Bill Evans discusses the ABS jobs data changes (ABC News)

**RELATED STORY:** Doubt the record breaking jobs figures? So does the ABS

**RELATED STORY:** Jobs increase record sees unemployment slashed

**RELATED STORY:** Unemployment surges to 12-year high at 6.4 pc

**MAP:** Australia

# The ABS stuff-up

## ABS jobs and unemployment figures – key questions answered by an expert

A professor of statistics at Monash University explains exactly what is seasonal adjustment, why it matters and what went wrong in the July and August figures

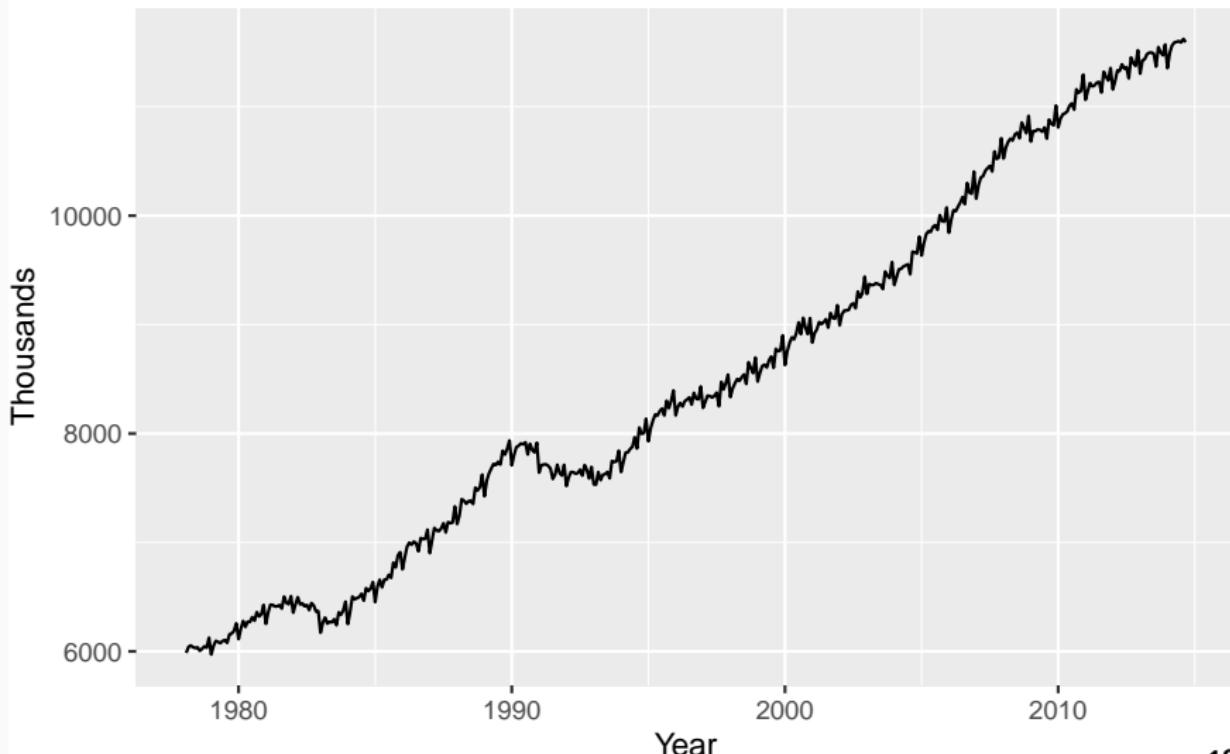


School leavers come on to the jobs market at the same time, causing a seasonal fluctuation. Photograph: Brian Snyder/Reuters

The Australian Bureau of Statistics has [retracted its seasonally adjusted employment data for July and August](#), which recorded huge swings in the jobless rate. The ABS is also planning to review the methods it uses for seasonal adjustment to ensure its figures are as accurate as possible. Rob Hyndman, a professor of statistics at Monash University and member of the bureau's

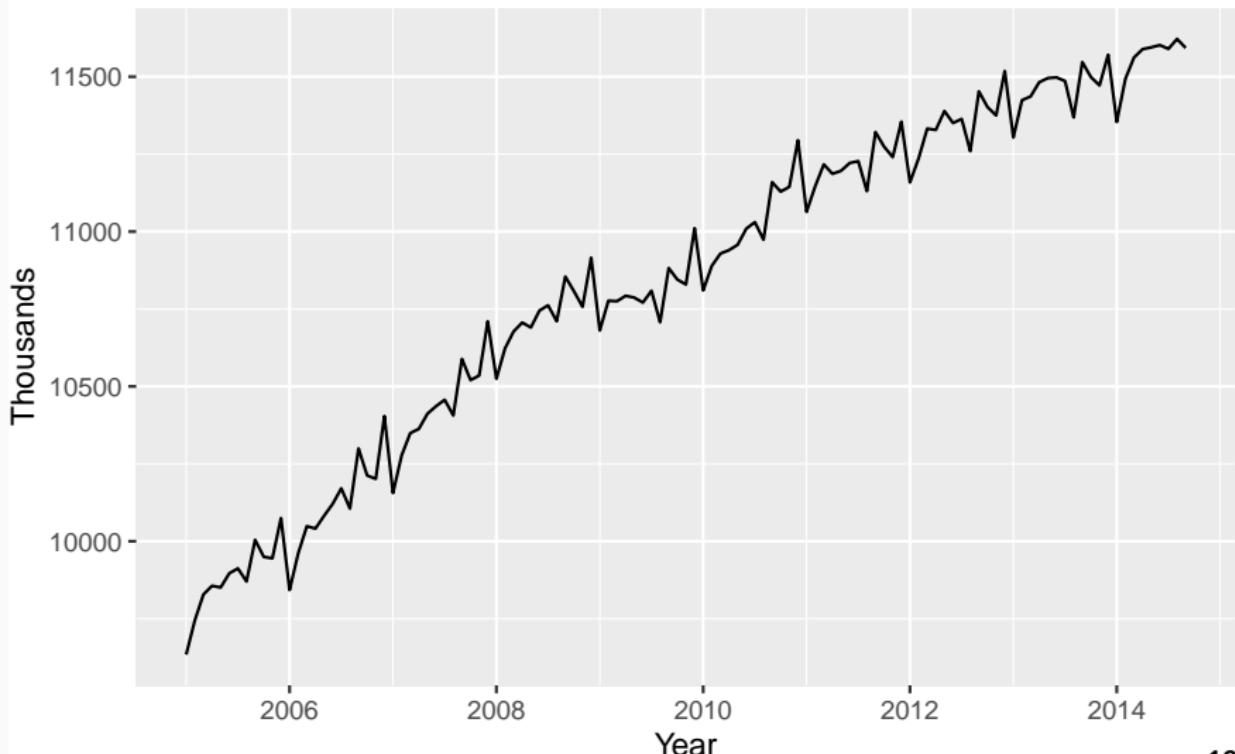
# The ABS stuff-up

Total employed



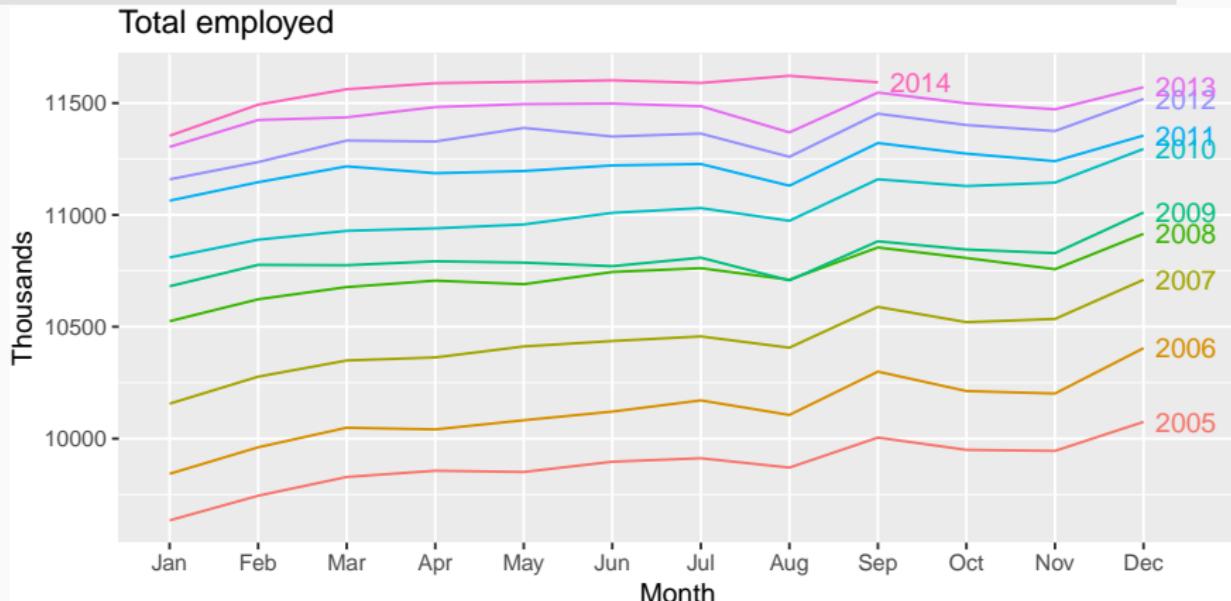
# The ABS stuff-up

Total employed



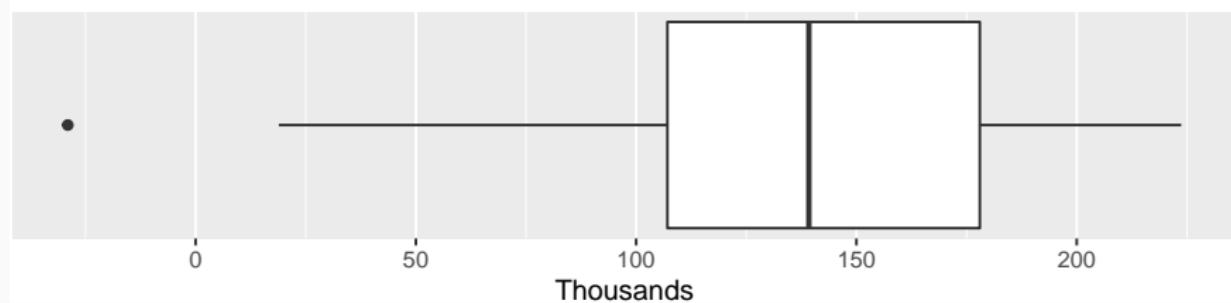
# The ABS stuff-up

```
ggseasonplot(window(x,start=c(2005,1)), year.labels=TRUE) +  
  ggtitle("Total employed") + ylab("Thousands")
```



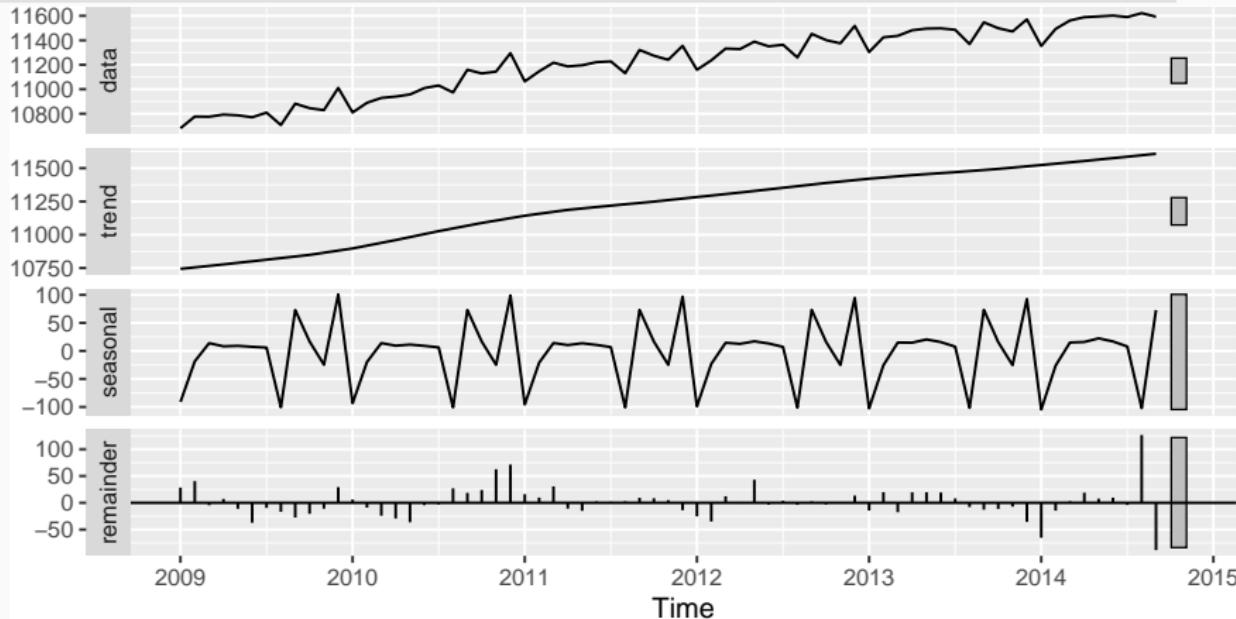
# The ABS stuff-up

Sep – Aug: total employed

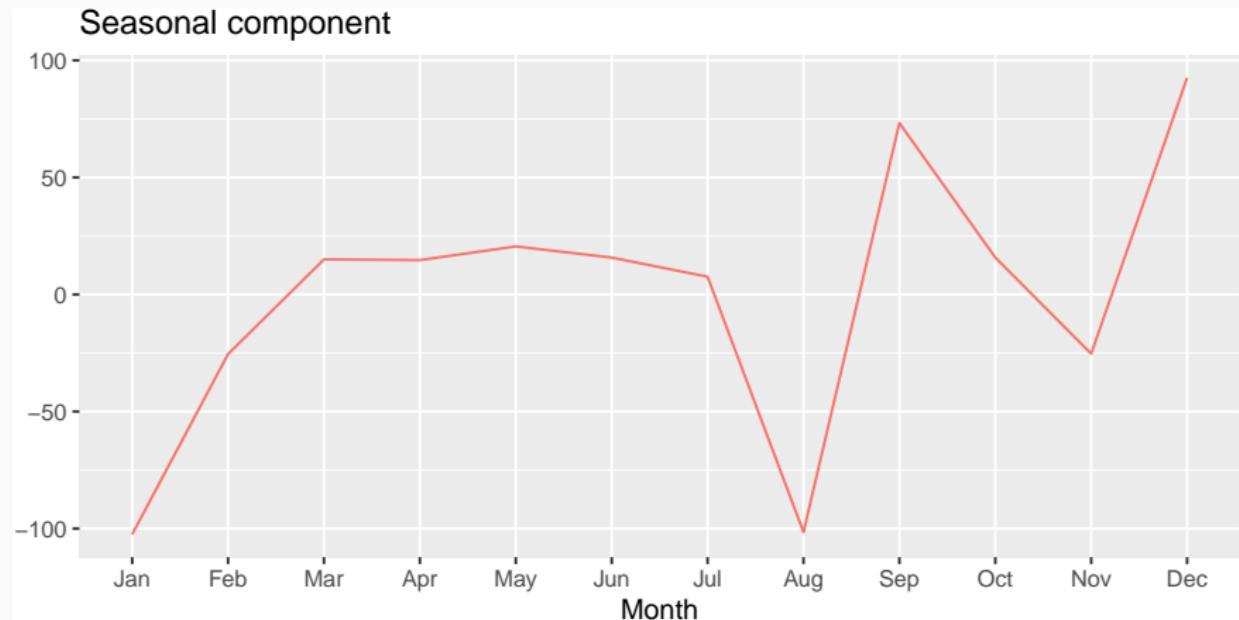


# The ABS stuff-up

```
x %>% window(start=2009) %>%  
  stl(s.window=11, robust=TRUE) -> fit  
  autoplot(fit)
```

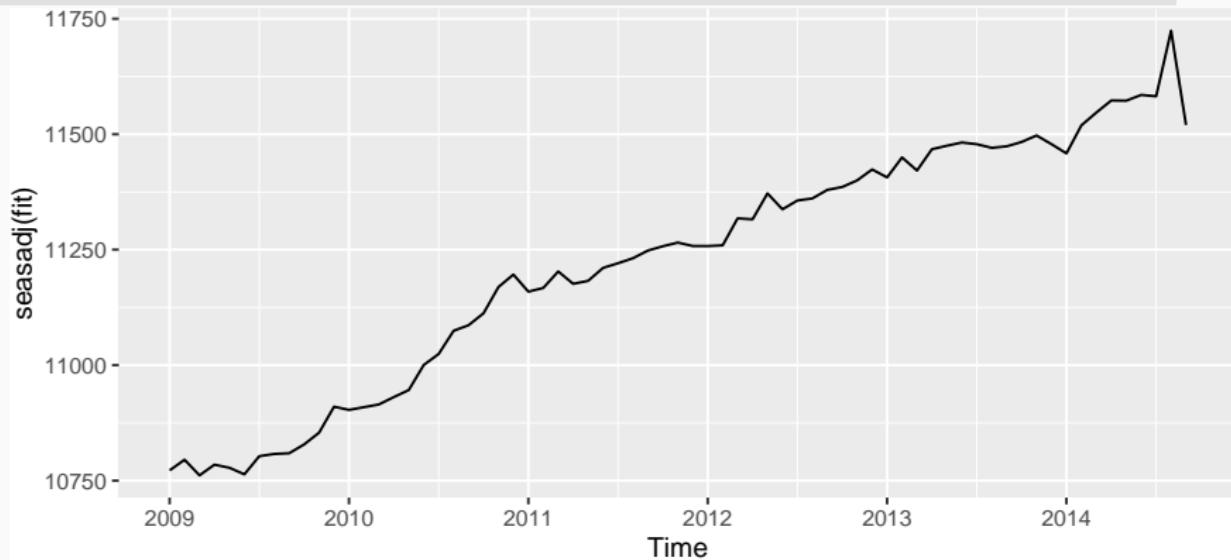


# The ABS stuff-up



# The ABS stuff-up

`autoplot(seasadj(fit))`



# The ABS stuff-up

- August 2014 employment numbers higher than expected.
- Supplementary survey usually conducted in August for employed people.
- Most likely, some employed people were claiming to be unemployed in August to avoid supplementary questions.
- Supplementary survey not run in 2014, so no motivation to lie about employment.
- In previous years, seasonal adjustment fixed the problem.
- The ABS has now adopted a new method to avoid the bias.

# History of time series decomposition

- Classical method originated in 1920s.
- Census II method introduced in 1957. Basis for X-11 method and variants (including X-12-ARIMA, X-13-ARIMA)
- STL method introduced in 1983
- TRAMO/SEATS introduced in 1990s.

# History of time series decomposition

- Classical method originated in 1920s.
- Census II method introduced in 1957. Basis for X-11 method and variants (including X-12-ARIMA, X-13-ARIMA)
- STL method introduced in 1983
- TRAMO/SEATS introduced in 1990s.

## National Statistics Offices

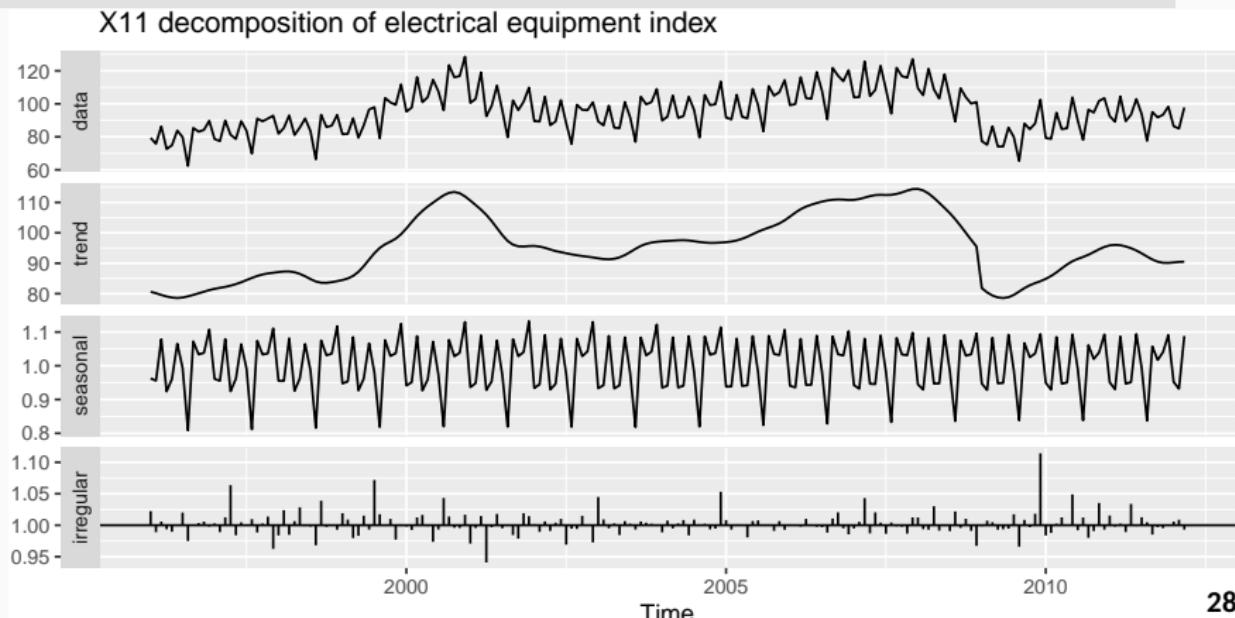
- ABS uses X-12-ARIMA
- US Census Bureau uses X-13-ARIMA-SEATS
- Statistics Canada uses X-12-ARIMA
- ONS (UK) uses X-12-ARIMA
- EuroStat use X-13-ARIMA-SEATS

# Outline

- 1 Time series components
- 2 Seasonal adjustment
- 3 X-11 decomposition
- 4 SEATS decomposition
- 5 STL decomposition
- 6 Forecasting and decomposition

# X-11 decomposition

```
library(seasonal)  
fit <- seas(elecequip, x11="")  
autoplot(fit)
```



# (Dis)advantages of X-11

## Advantages

- Relatively robust to outliers
- Completely automated choices for trend and seasonal changes
- Very widely tested on economic data over a long period of time.

# (Dis)advantages of X-11

## Advantages

- Relatively robust to outliers
- Completely automated choices for trend and seasonal changes
- Very widely tested on economic data over a long period of time.

## Disadvantages

- No prediction/confidence intervals
- Ad hoc method with no underlying model
- Only developed for quarterly and monthly data

## Extensions: X-12-ARIMA and X-13-ARIMA

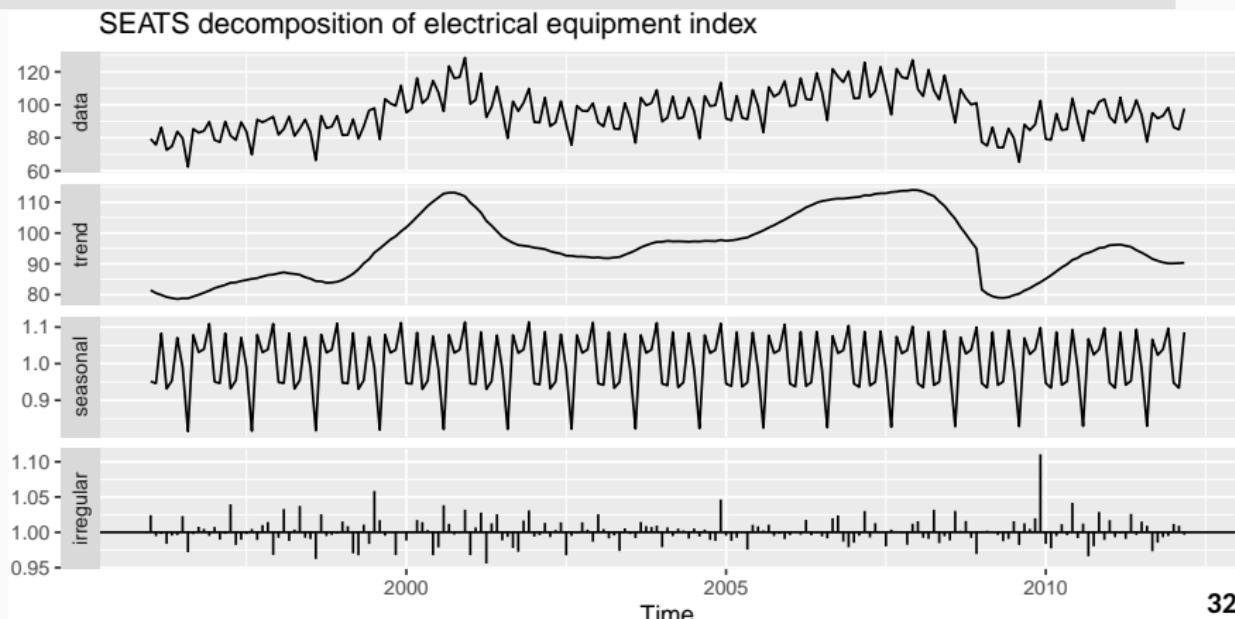
- The X-11, X-12-ARIMA and X-13-ARIMA methods are based on Census II decomposition.
- These allow adjustments for trading days and other explanatory variables.
- Known outliers can be omitted.
- Level shifts and ramp effects can be modelled.
- Missing values estimated and replaced.
- Holiday factors (e.g., Easter, Labour Day) can be estimated.

# Outline

- 1 Time series components**
- 2 Seasonal adjustment**
- 3 X-11 decomposition**
- 4 SEATS decomposition**
- 5 STL decomposition**
- 6 Forecasting and decomposition**

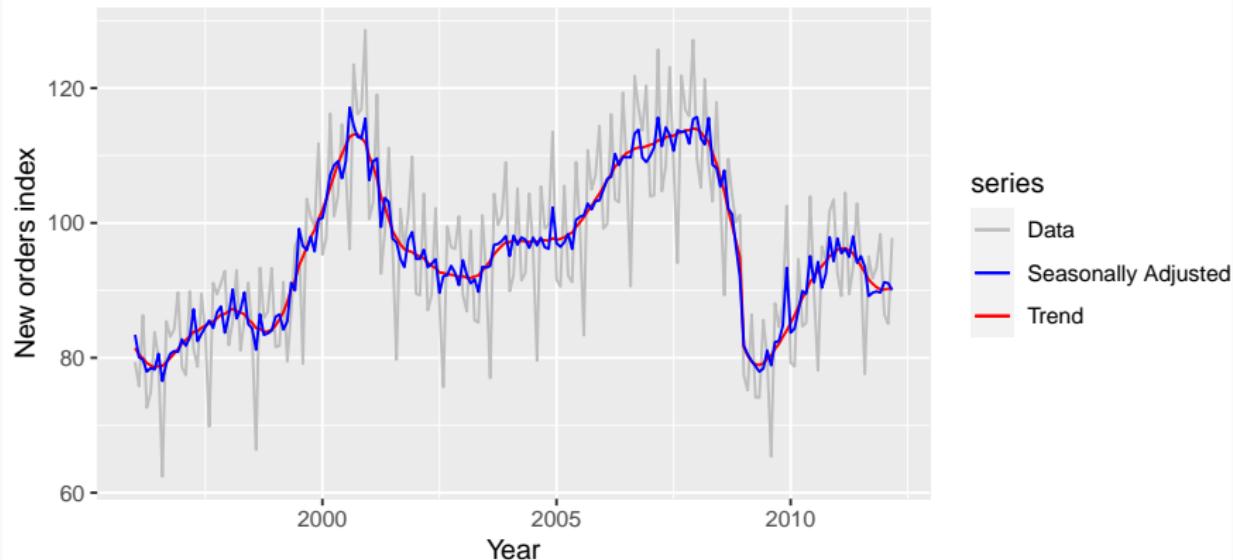
# SEATS decomposition

```
library(seas)
fit <- seas(elecequip)
autoplot(fit)
```



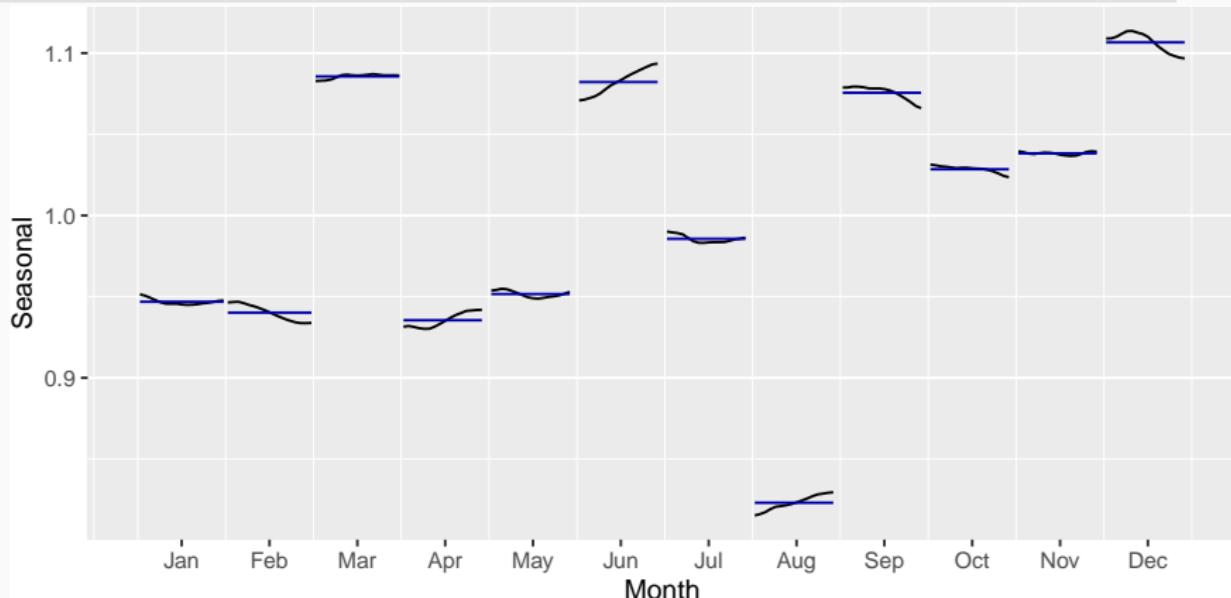
# SEATS decomposition

Electrical equipment manufacturing (Euro area)



# SEATS decomposition

```
ggsubseriesplot(seasonal(fit)) + ylab("Seasonal")
```



# (Dis)advantages of SEATS

## Advantages

- Model-based
- Smooth trend estimate
- Allows estimates at end points
- Allows changing seasonality
- Developed for economic data

# (Dis)advantages of SEATS

## Advantages

- Model-based
- Smooth trend estimate
- Allows estimates at end points
- Allows changing seasonality
- Developed for economic data

## Disadvantages

- Only developed for quarterly and monthly data

# Outline

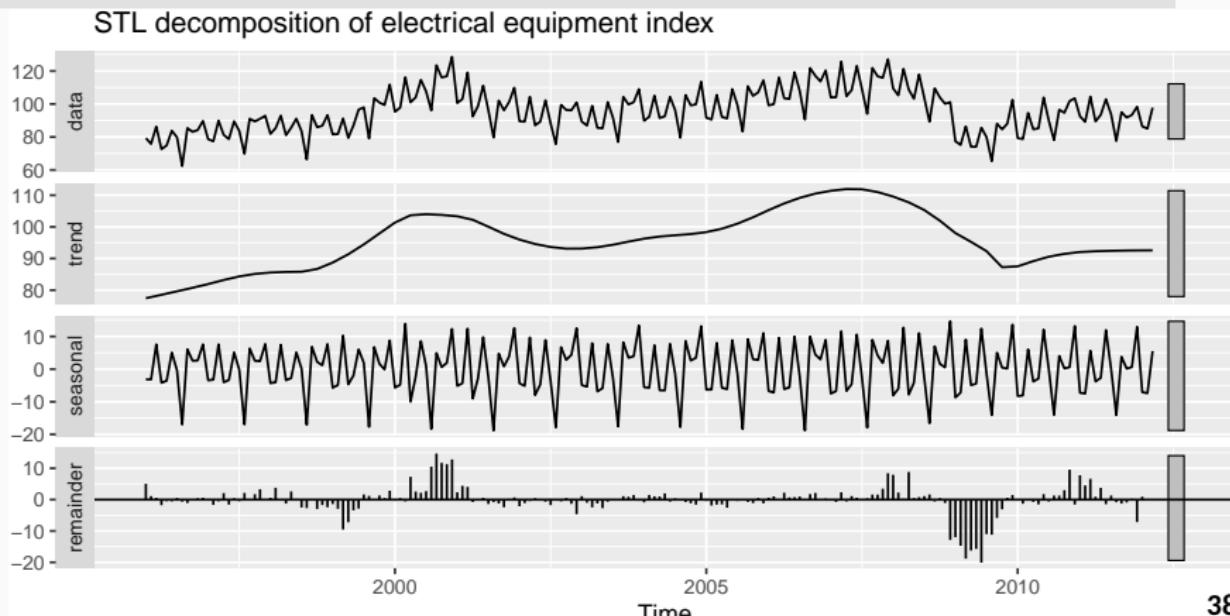
- 1 Time series components**
- 2 Seasonal adjustment**
- 3 X-11 decomposition**
- 4 SEATS decomposition**
- 5 STL decomposition**
- 6 Forecasting and decomposition**

# STL decomposition

- STL: “Seasonal and Trend decomposition using Loess”
- Very versatile and robust.
- Unlike X-12-ARIMA, STL will handle any type of seasonality.
- Seasonal component allowed to change over time, and rate of change controlled by user.
- Smoothness of trend-cycle also controlled by user.
- Robust to outliers
- Not trading day or calendar adjustments.
- Only additive.
- Take logs to get multiplicative decomposition.
- Use Box-Cox transformations to get other decompositions.

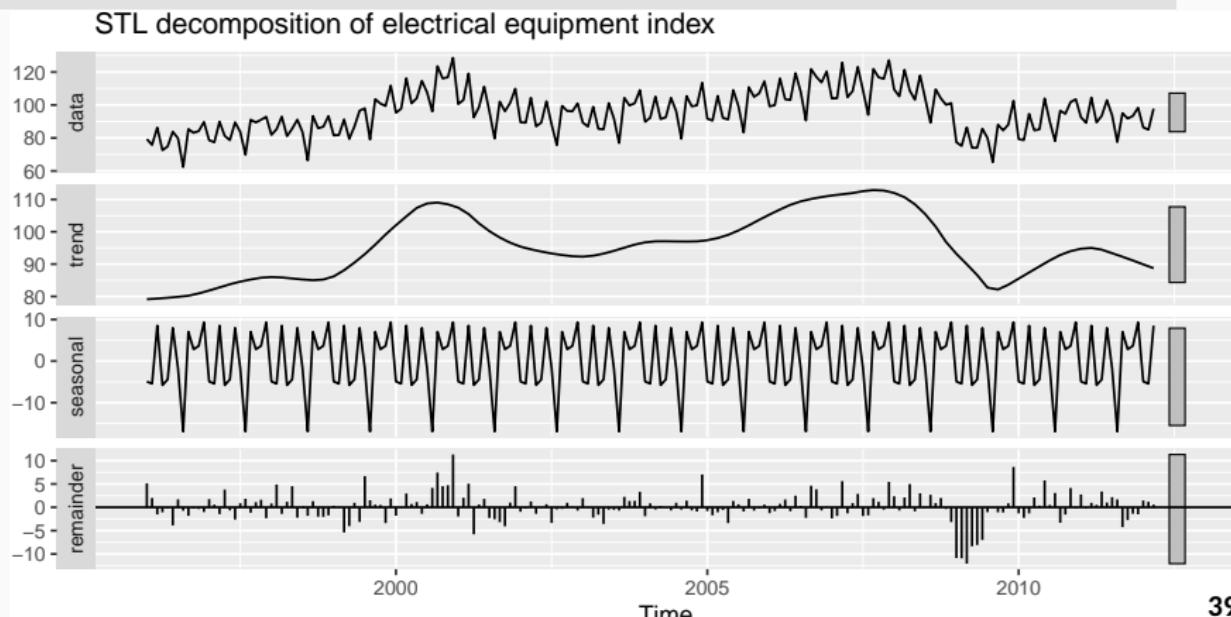
# STL decomposition

```
fit <- stl(elecequip, s.window=5, robust=TRUE)  
autoplot(fit) +  
  ggtitle("STL decomposition of electrical equipment index")
```



# STL decomposition

```
fit <- stl(elecequip, s.window="periodic", robust=TRUE)  
autoplot(fit) +  
  ggtitle("STL decomposition of electrical equipment index")
```



# STL decomposition

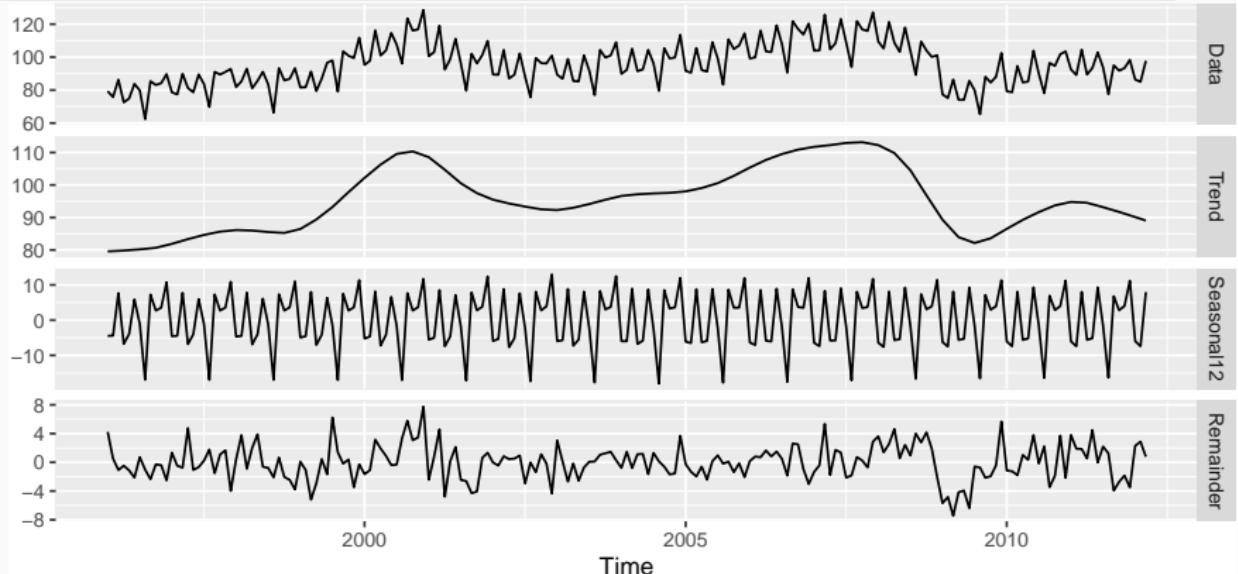
```
stl(elecequip, s.window=5)
```

```
stl(elecequip, t.window=15,  
    s.window="periodic", robust=TRUE)
```

- `t.window` controls wiggliness of trend component.
- `s.window` controls variation on seasonal component.

# STL decomposition

```
elecequip %>% mstl() %>% autoplot()
```



- `mstl()` chooses `s.window=13`
- Can include a `lambda` argument.

# Outline

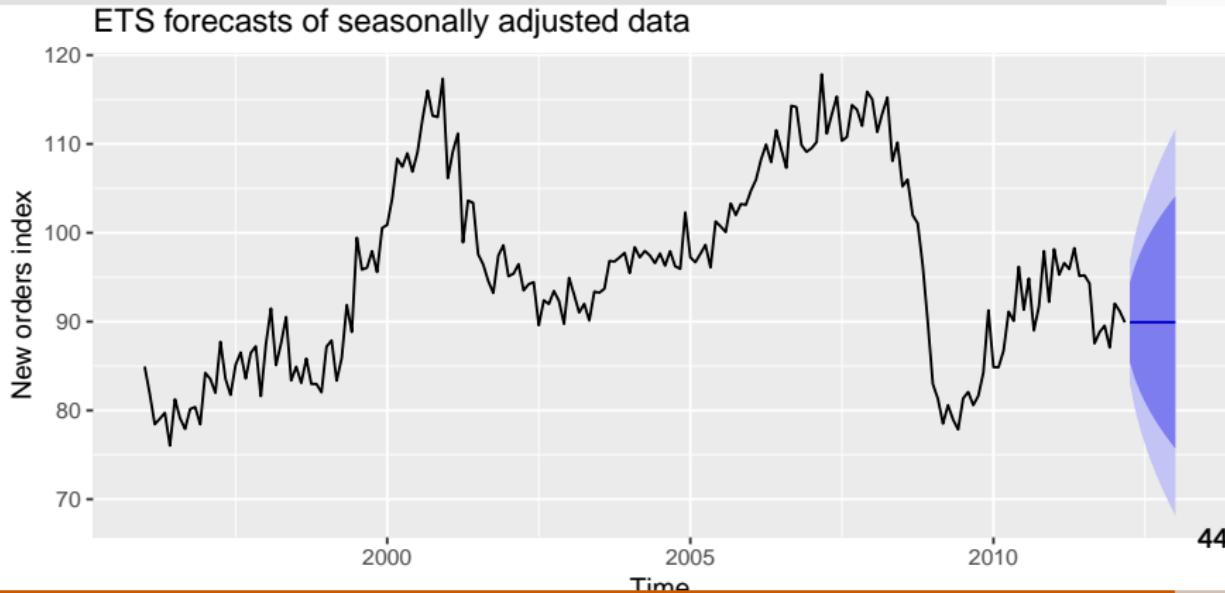
- 1 Time series components**
- 2 Seasonal adjustment**
- 3 X-11 decomposition**
- 4 SEATS decomposition**
- 5 STL decomposition**
- 6 Forecasting and decomposition**

# Forecasting and decomposition

- Forecast seasonal component by repeating the last year
- Forecast seasonally adjusted data using non-seasonal time series method.
- Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.
- Sometimes a decomposition is useful just for understanding the data before building a separate forecasting model.

# Electrical equipment

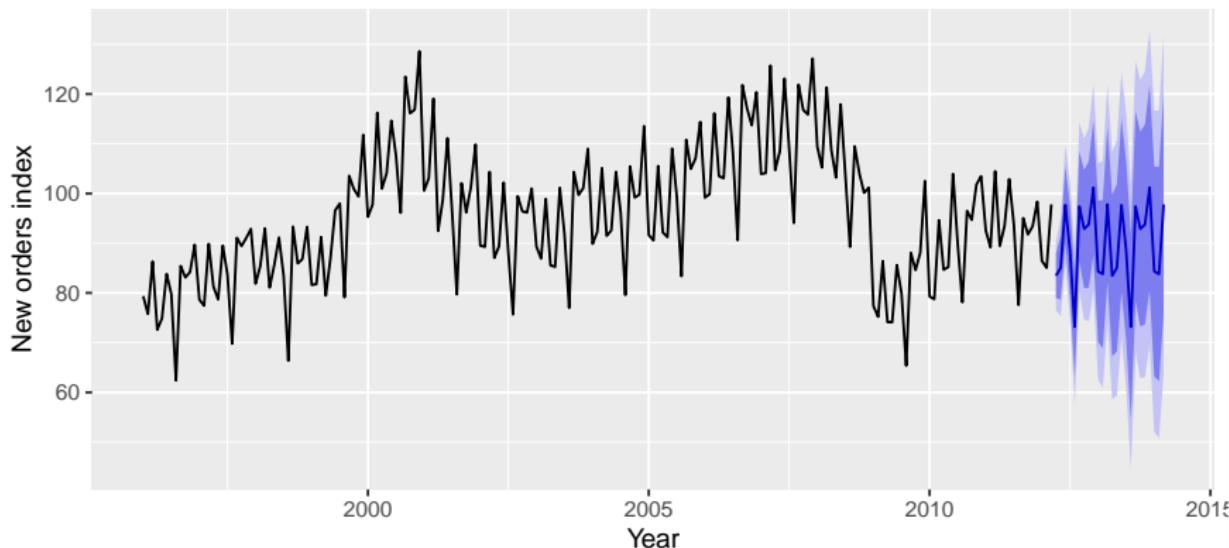
```
fit <- stl(elecequip, t.window=13, s.window="periodic")
fit %>% seasadj() %>% naive() %>%
  autoplot() + ylab("New orders index") +
  ggtitle("ETS forecasts of seasonally adjusted data")
```



# Electrical equipment

```
fit %>% forecast(method='naive') %>%  
  autoplot() + ylab("New orders index") + xlab("Year")
```

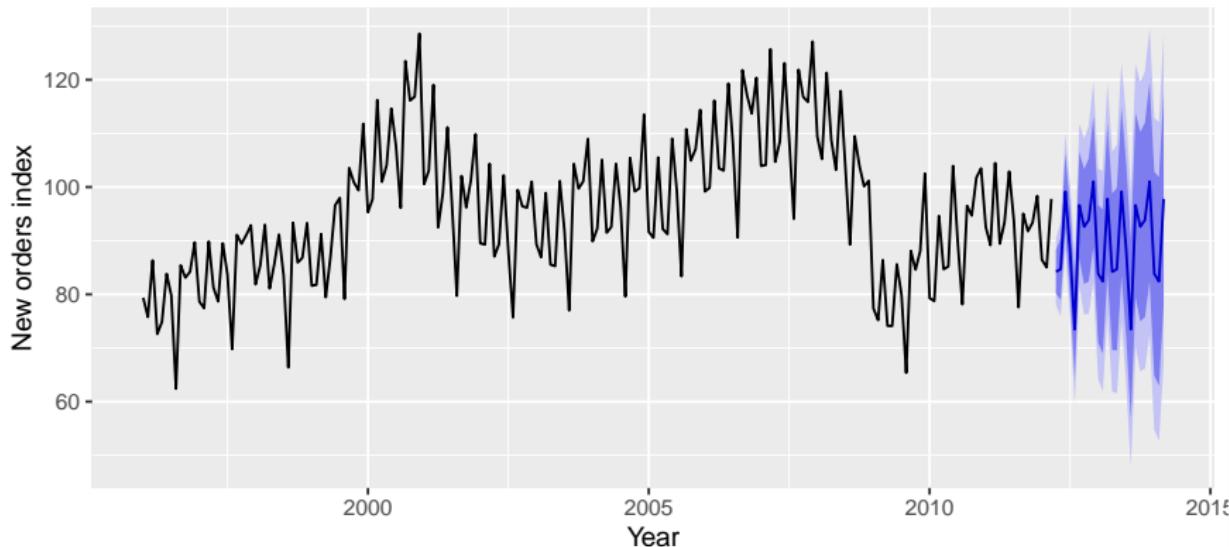
Forecasts from STL + Random walk



# Forecasting and decomposition

```
elecequip %>% stlf(method='naive') %>%  
  autoplot() + ylab("New orders index") + xlab("Year")
```

Forecasts from STL + Random walk



## Decomposition and prediction intervals

- It is common to take the prediction intervals from the seasonally adjusted forecasts and modify them with the seasonal component.
- This ignores the uncertainty in the seasonal component estimate.
- It also ignores the uncertainty in the future seasonal pattern.



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# STU33010: Forecasting

Ch7. Exponential smoothing

[OTexts.org/fpp2/](http://OTexts.org/fpp2/)

# Outline

- 1 Simple exponential smoothing
- 2 Trend methods
- 3 Seasonal methods
- 4 Taxonomy of exponential smoothing methods
- 5 Innovations state space models
- 6 ETS in R

# Simple methods

Time series  $y_1, y_2, \dots, y_T$ .

## Naive forecasts

$$\hat{y}_{T+h|T} = y_T$$

# Simple methods

Time series  $y_1, y_2, \dots, y_T$ .

## Naive forecasts

$$\hat{y}_{T+h|T} = y_T$$

## Average forecasts

$$\hat{y}_{T+h|T} = \frac{1}{T} \sum_{t=1}^T y_t$$

# Simple methods

Time series  $y_1, y_2, \dots, y_T$ .

## Naive forecasts

$$\hat{y}_{T+h|T} = y_T$$

## Average forecasts

$$\hat{y}_{T+h|T} = \frac{1}{T} \sum_{t=1}^T y_t$$

- Want something in between that **weights most recent data more highly.**
- Simple exponential smoothing uses a **weighted moving average** with weights that decrease exponentially.

# Simple Exponential Smoothing

## Forecast equation

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1 - \alpha)y_{T-1} + \alpha(1 - \alpha)^2 y_{T-2} + \dots$$

where  $0 \leq \alpha \leq 1$ .

# Simple Exponential Smoothing

## Forecast equation

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1 - \alpha)y_{T-1} + \alpha(1 - \alpha)^2y_{T-2} + \dots$$

where  $0 \leq \alpha \leq 1$ .

Weights assigned to observations for:

Observation	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
$y_T$	0.2	0.4	0.6	0.8
$y_{T-1}$	0.16	0.24	0.24	0.16
$y_{T-2}$	0.128	0.144	0.096	0.032
$y_{T-3}$	0.1024	0.0864	0.0384	0.0064
$y_{T-4}$	$(0.2)(0.8)^4$	$(0.4)(0.6)^4$	$(0.6)(0.4)^4$	$(0.8)(0.2)^4$
$y_{T-5}$	$(0.2)(0.8)^5$	$(0.4)(0.6)^5$	$(0.6)(0.4)^5$	$(0.8)(0.2)^5$

# Simple Exponential Smoothing

## Component form

Forecast equation  $\hat{y}_{t+h|t} = \ell_t$

Smoothing equation  $\ell_t = \alpha y_t + (1 - \alpha) \ell_{t-1}$

- $\ell_t$  is the level (or the smoothed value) of the series at time t.
- $\hat{y}_{t+1|t} = \alpha y_t + (1 - \alpha) \hat{y}_{t|t-1}$   
Iterate to get exponentially weighted moving average form.

## Weighted average form

$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} \alpha(1 - \alpha)^j y_{T-j} + (1 - \alpha)^T \ell_0$$

# Optimisation

- Need to choose value for  $\alpha$  and  $\ell_0$
- Similarly to regression — we choose  $\alpha$  and  $\ell_0$  by minimising

SSE:

$$\text{SSE} = \sum_{t=1}^T (y_t - \hat{y}_{t|t-1})^2.$$

- Unlike regression there is no closed form solution — use numerical optimization.

## Example: Oil production

```
oildata <- window(oil, start=1996)
# Estimate parameters
fc <- ses(oildata, h=5)
summary(fc[["model"]])
```

```
## Simple exponential smoothing
##
## Call:
##   ses(y = oildata, h = 5)
##
##   Smoothing parameters:
##     alpha = 0.8339
##
##   Initial states:
##     l = 446.5868
##
##   sigma: 29.83
##
##   AIC  AICC  BIC
```

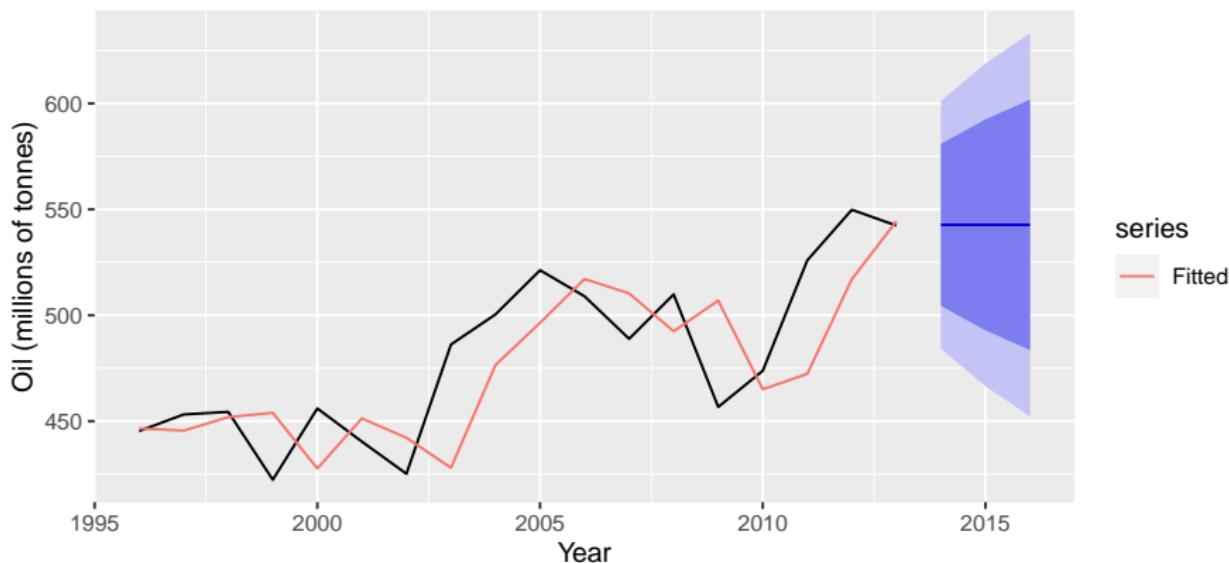
## Example: Oil production

```
##   Year    Time  Observation Level      Forecast
## 1   ""     "$t$"  "$y_t$"      "$\\ell_t$"  "$\\hat{y}_{t+1|t}$"
## 2 "1995" " 0"   ""           "446.59"    ""
## 3 "1996" " 1"   "445.36"    "445.57"    "446.59"
## 4 "1997" " 2"   "453.20"    "451.93"    "445.57"
## 5 "1998" " 3"   "454.41"    "454.00"    "451.93"
## 6 "1999" " 4"   "422.38"    "427.63"    "454.00"
## 7 "2000" " 5"   "456.04"    "451.32"    "427.63"
## 8 "2001" " 6"   "440.39"    "442.20"    "451.32"
## 9 "2002" " 7"   "425.19"    "428.02"    "442.20"
## 10 "2003" " 8"   "486.21"    "476.54"    "428.02"
## 11 "2004" " 9"   "500.43"    "496.46"    "476.54"
## 12 "2005" "10"   "521.28"    "517.15"    "496.46"
## 13 "2006" "11"   "508.95"    "510.31"    "517.15"
## 14 "2007" "12"   "488.89"    "492.45"    "510.31"
## 15 "2008" "13"   "509.87"    "506.98"    "492.45"
## 16 "2009" "14"   "456.72"    "465.07"    "506.98"
## 17 "2010" "15"   "473.82"    "472.36"    "465.07"
## 18 "2011" "16"   "525.95"    "517.05"    "472.36"
## 19 "2012" "17"   "549.83"    "544.39"    "517.05"
## 20 "2013" "18"   "542.34"    "542.68"    "544.39"
## 21   ""     "$h$"   ""          ""          "$\\hat{y}_{T+h|T}$"
## 22 "2014" " 1"   ""          ""          "542.68"
## 23 "2015" " 2"   ""          ""          "542.68"
## 24 "2016" " 3"   ""          ""          "542.68"
```

## Example: Oil production

```
autoplot(fc) +  
autolayer(fitted(fc), series="Fitted") +  
ylab("Oil (millions of tonnes)") + xlab("Year")
```

Forecasts from Simple exponential smoothing



# Outline

- 1 Simple exponential smoothing**
- 2 Trend methods**
- 3 Seasonal methods**
- 4 Taxonomy of exponential smoothing methods**
- 5 Innovations state space models**
- 6 ETS in R**

# Holt's linear trend

## Component form

Forecast       $\hat{y}_{t+h|t} = \ell_t + hb_t$

Level       $\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1})$

Trend       $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1},$

# Holt's linear trend

## Component form

Forecast       $\hat{y}_{t+h|t} = \ell_t + hb_t$

Level       $\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1})$

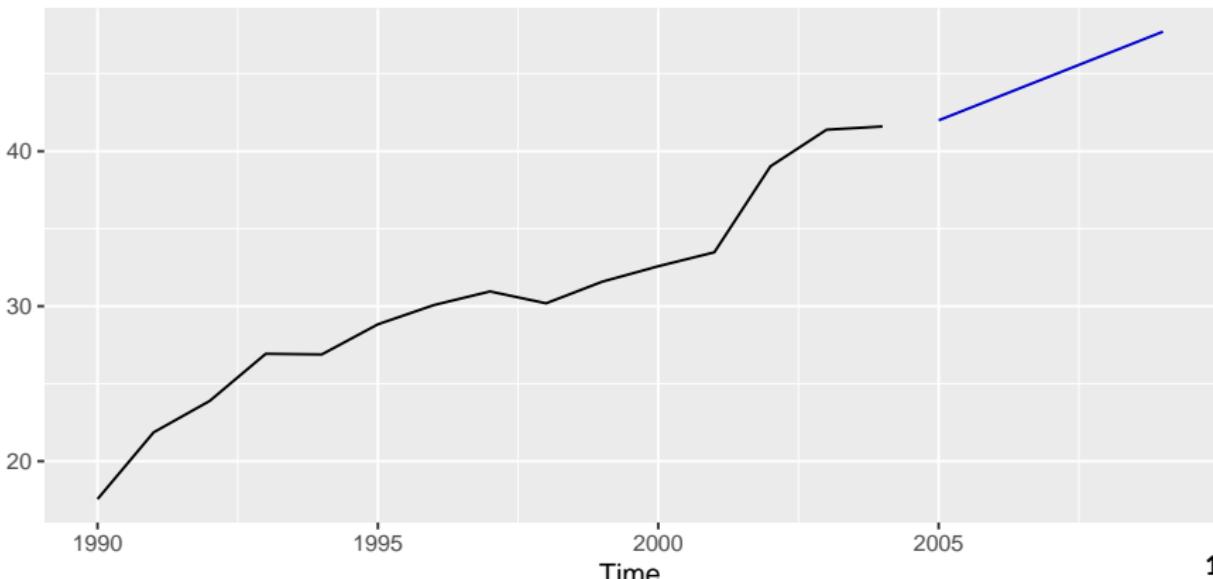
Trend       $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1},$

- Two smoothing parameters  $\alpha$  and  $\beta^*$  ( $0 \leq \alpha, \beta^* \leq 1$ ).
- $\ell_t$  level: weighted average between  $y_t$  and one-step ahead forecast for time  $t$ ,  $(\ell_{t-1} + b_{t-1} = \hat{y}_{t|t-1})$
- $b_t$  slope: weighted average of  $(\ell_t - \ell_{t-1})$  and  $b_{t-1}$ , current and previous estimate of slope.
- Choose  $\alpha, \beta^*, \ell_0, b_0$  to minimise SSE.

# Holt's method in R

```
window(ausair, start=1990, end=2004) %>%  
  holt(h=5, PI=FALSE) %>%  
  autoplot()
```

Forecasts from Holt's method



# Damped trend method

## Component form

$$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \dots + \phi^h)b_t$$

$$\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}.$$

# Damped trend method

## Component form

$$\hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \dots + \phi^h)b_t$$

$$\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$$

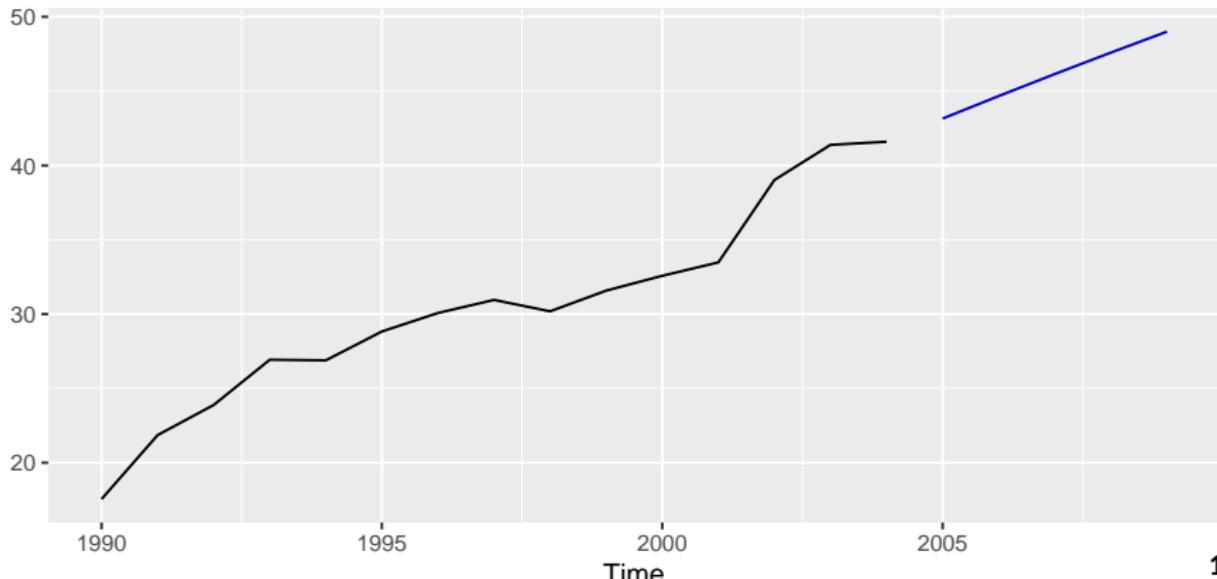
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$$

- Damping parameter  $0 < \phi < 1$ .
- If  $\phi = 1$ , identical to Holt's linear trend.
- As  $h \rightarrow \infty$ ,  $\hat{y}_{T+h|T} \rightarrow \ell_T + \phi b_T / (1 - \phi)$ .
- Short-run forecasts trended, long-run forecasts constant.

## Example: Air passengers

```
window(ausair, start=1990, end=2004) %>%  
  holt(damped=TRUE, h=5, PI=FALSE) %>%  
  autoplot()
```

Forecasts from Damped Holt's method



## Example: Sheep in Asia

We use a subset for training

```
livestock2 <- window(livestock, start=1970,  
                      end=2000)  
  
fit1 <- ses(livestock2)  
fit2 <- holt(livestock2)  
fit3 <- holt(livestock2, damped = TRUE)
```

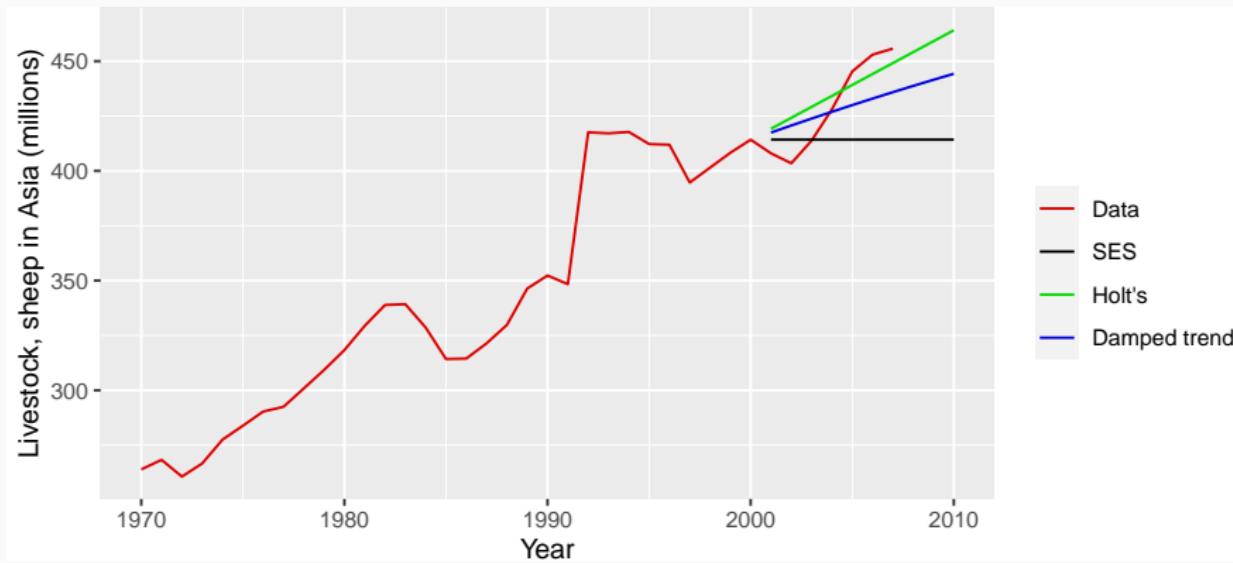
We compute accuracy on the whole dataset

```
accuracy(fit1, livestock)  
accuracy(fit2, livestock)  
accuracy(fit3, livestock)
```

## Example: Sheep in Asia

	SES	Linear trend	Damped trend
## $\alpha$	"1.00"	"0.98"	"0.97"
## $\beta^*$	""	"0.00"	"0.00"
## $\phi$	""	""	"0.98"
## $\ell_0$	"263.90"	"251.46"	"251.89"
## $b_0$	""	"4.99"	"6.29"
## Training RMSE	"14.77"	"13.98"	"14.00"
## Test RMSE	"25.46"	"11.88"	"14.73"
## Test MAE	"20.38"	"10.71"	"13.30"
## Test MAPE	"4.60"	"2.54"	"3.07"
## Test MASE	"2.26"	"1.19"	"1.48"

## Example: Sheep in Asia



## Your turn

eggs contains the price of a dozen eggs in the United States from 1900–1993

- 1 Use SES and Holt's method (with and without damping) to forecast "future" data.  
[Hint: use  $h=100$  so you can clearly see the differences between the options when plotting the forecasts.]
- 2 Which method gives the best training RMSE?
- 3 Are these RMSE values comparable?
- 4 Do the residuals from the best fitting method look like white noise?

# Outline

- 1 Simple exponential smoothing
- 2 Trend methods
- 3 Seasonal methods
- 4 Taxonomy of exponential smoothing methods
- 5 Innovations state space models
- 6 ETS in R

# Holt-Winters additive method

Holt and Winters extended Holt's method to capture seasonality.

## Component form

$$\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$$

$$\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1})$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$$

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m},$$

- $k = \text{integer part of } (h - 1)/m$ . Ensures estimates from the final year are used for forecasting.
- Parameters:  $0 \leq \alpha \leq 1$ ,  $0 \leq \beta^* \leq 1$ ,  $0 \leq \gamma \leq 1 - \alpha$  and  $m = \text{period of seasonality}$  (e.g.  $m = 4$  for quarterly data).

## Holt-Winters additive method

- Seasonal component is usually expressed as

$$s_t = \gamma^*(y_t - \ell_t) + (1 - \gamma^*)s_{t-m}.$$

- Substitute in for  $\ell_t$ :

$$s_t = \gamma^*(1 - \alpha)(y_t - \ell_{t-1} - b_{t-1}) + [1 - \gamma^*(1 - \alpha)]s_{t-m}$$

- We set  $\gamma = \gamma^*(1 - \alpha)$ .

- The usual parameter restriction is  $0 \leq \gamma^* \leq 1$ , which translates to  $0 \leq \gamma \leq (1 - \alpha)$ .

# Holt-Winters multiplicative method

For when seasonal variations are changing proportional to the level of the series.

## Component form

$$\hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t+h-m(k+1)}.$$

$$\ell_t = \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(\ell_{t-1} + b_{t-1})$$

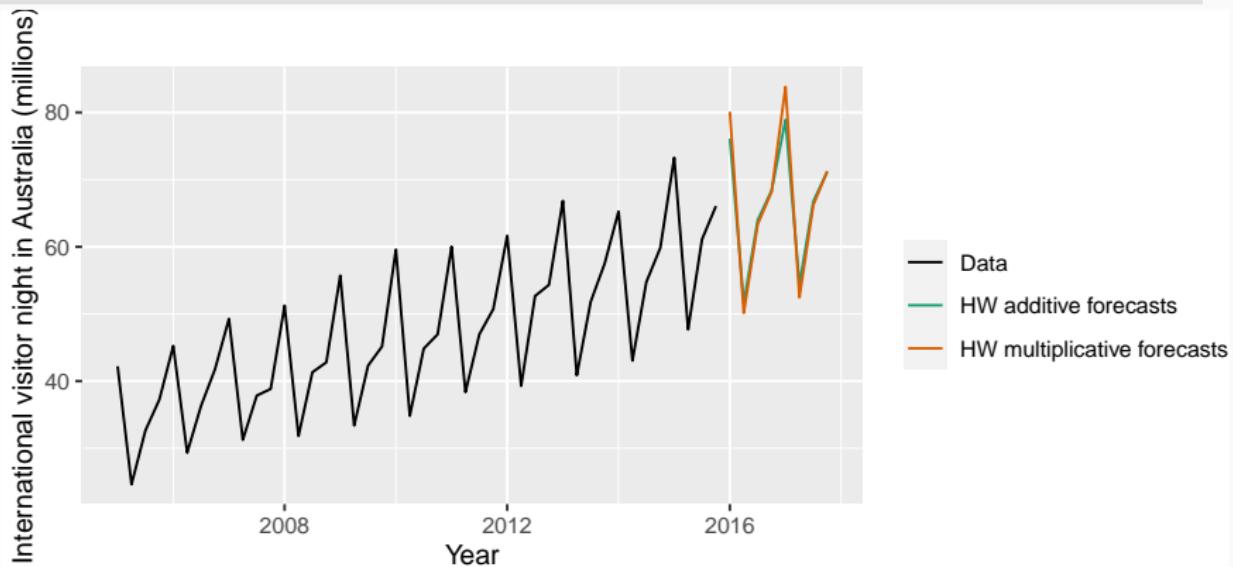
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$$

$$s_t = \gamma \frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1 - \gamma)s_{t-m}$$

- $k$  is integer part of  $(h - 1)/m$ .
- With additive method  $s_t$  is in absolute terms:  
within each year  $\sum_i s_i \approx 0$ .
- With multiplicative method  $s_t$  is in relative terms:  
within each year  $\sum_i s_i \approx m$ .

## Example: Visitor Nights

```
aust <- window(austourists,start=2005)  
fit1 <- hw(aust,seasonal="additive")  
fit2 <- hw(aust,seasonal="multiplicative")
```

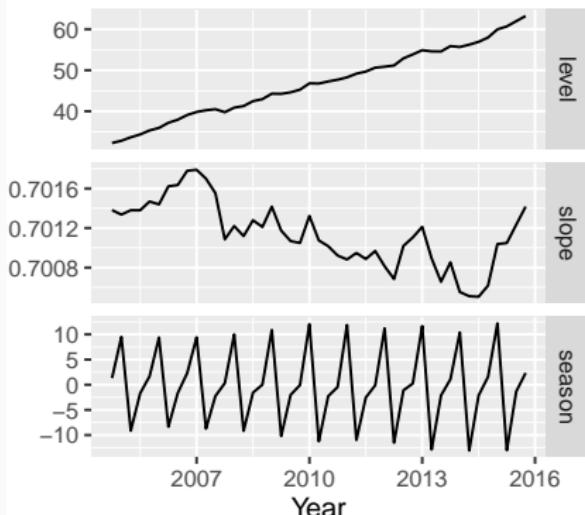


# Estimated components

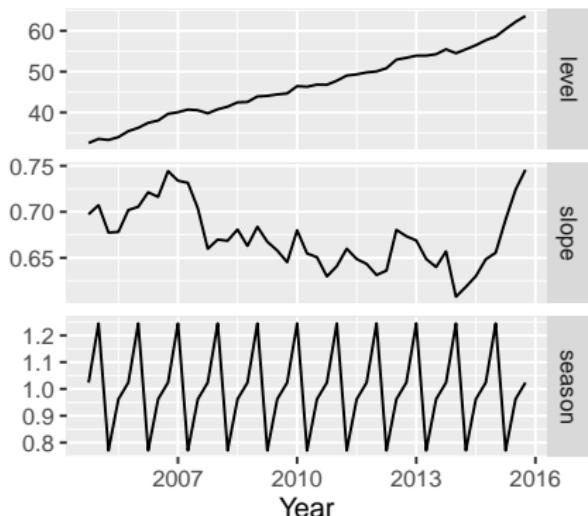
```
addstates <- fit1$model$states[,1:3]
```

```
multstates <- fit2$model$states[,1:3]
```

Additive states



Multiplicative states



## Holt-Winters damped method

Often the single most accurate forecasting method for seasonal data:

$$\hat{y}_{t+h|t} = [\ell_t + (\phi + \phi^2 + \dots + \phi^h)b_t]s_{t+h-m(k+1)}$$

$$\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$$

$$s_t = \gamma \frac{y_t}{(\ell_{t-1} + \phi b_{t-1})} + (1 - \gamma)s_{t-m}$$

## Your turn

Apply Holt-Winters' multiplicative method to the gas data.

- 1 Why is multiplicative seasonality necessary here?
- 2 Experiment with making the trend damped.
- 3 Check that the residuals from the best method look like white noise.

# Outline

- 1 Simple exponential smoothing
- 2 Trend methods
- 3 Seasonal methods
- 4 Taxonomy of exponential smoothing methods
- 5 Innovations state space models
- 6 ETS in R

# Exponential smoothing methods

		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	(N,N)	(N,A)	(N,M)
A	(Additive)	(A,N)	(A,A)	(A,M)
A <sub>d</sub>	(Additive damped)	(A <sub>d</sub> ,N)	(A <sub>d</sub> ,A)	(A <sub>d</sub> ,M)

(N,N): Simple exponential smoothing

(A,N): Holt's linear method

(A<sub>d</sub>,N): Additive damped trend method

(A,A): Additive Holt-Winters' method

(A,M): Multiplicative Holt-Winters' method

(A<sub>d</sub>,M): Damped multiplicative Holt-Winters' method

# Recursive formulae

Trend		Seasonal	
	N	A	M
	$\hat{y}_{t+h t} = \ell_t$	$\hat{y}_{t+h t} = \ell_t + s_{t+h-m(k+1)}$	$\hat{y}_{t+h t} = \ell_t s_{t+h-m(k+1)}$
N	$\ell_t = \alpha y_t + (1 - \alpha)\ell_{t-1}$	$\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1 - \gamma)s_{t-m}$	$\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1 - \gamma)s_{t-m}$
	$\hat{y}_{t+h t} = \ell_t + hb_t$	$\hat{y}_{t+h t} = \ell_t + hb_t + s_{t+h-m(k+1)}$	$\hat{y}_{t+h t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$
A	$\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$	$\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}$	$\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1 - \gamma)s_{t-m}$
	$\hat{y}_{t+h t} = \ell_t + \phi_h b_t$	$\hat{y}_{t+h t} = \ell_t + \phi_h b_t + s_{t+h-m(k+1)}$	$\hat{y}_{t+h t} = (\ell_t + \phi_h b_t)s_{t+h-m(k+1)}$
A <sub>d</sub>	$\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$	$\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1 - \gamma)s_{t-m}$	$\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1 - \gamma)s_{t-m}$

# R functions

- Simple exponential smoothing: no trend.

`ses(y)`

- Holt's method: linear trend.

`holt(y)`

- Damped trend method.

`holt(y, damped=TRUE)`

- Holt-Winters methods

`hw(y, damped=TRUE, seasonal="additive")`

`hw(y, damped=FALSE, seasonal="additive")`

`hw(y, damped=TRUE, seasonal="multiplicative")`

`hw(y, damped=FALSE, seasonal="multiplicative")`

- Combination of no trend with seasonality **not possible** using these functions.

# Outline

- 1 Simple exponential smoothing**
- 2 Trend methods**
- 3 Seasonal methods**
- 4 Taxonomy of exponential smoothing methods**
- 5 Innovations state space models**
- 6 ETS in R**

## Exponential smoothing methods

- Algorithms that return point forecasts.

# Methods v Models

## Exponential smoothing methods

- Algorithms that return point forecasts.

## Innovations state space models

- Generate same point forecasts but can also generate forecast intervals.
- A stochastic (or random) data generating process that can generate an entire forecast distribution.
- Allow for “proper” model selection.

## ETS models

- Each model has an *observation* equation and *transition* equations, one for each state (level, trend, seasonal), i.e., state space models.
- Two models for each method: one with additive and one with multiplicative errors, i.e., in total **18 models**.
- **ETS(Error,Trend,Seasonal):**
  - Error = {A,M}
  - Trend = {N,A,A<sub>d</sub>}
  - Seasonal = {N,A,M}.

# Exponential smoothing methods

		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N (None)		N,N	N,A	N,M
A (Additive)		A,N	A,A	A,M
A <sub>d</sub> (Additive damped)		A <sub>d</sub> ,N	A <sub>d</sub> ,A	A <sub>d</sub> ,M

# Exponential smoothing methods

		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N (None)		N,N	N,A	N,M
A (Additive)		A,N	A,A	A,M
A <sub>d</sub> (Additive damped)		A <sub>d</sub> ,N	A <sub>d</sub> ,A	A <sub>d</sub> ,M

General notation

ETS : ExponenTial Smoothing

Error Trend Seasonal

# Exponential smoothing methods

		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A <sub>d</sub>	(Additive damped)	A <sub>d</sub> ,N	A <sub>d</sub> ,A	A <sub>d</sub> ,M

General notation

ETS : ExponenTial Smoothing

Examples: Error Trend Seasonal

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

# Exponential smoothing methods

		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A <sub>d</sub>	(Additive damped)	A <sub>d</sub> ,N	A <sub>d</sub> ,A	A <sub>d</sub> ,M

General notation

ETS : ExponenTial Smoothing

Examples: Error Trend Seasonal

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

There are 18 separate models in the ETS framework

# A model for SES

## Component form

Forecast equation

$$\hat{y}_{t+h|t} = \ell_t$$

Smoothing equation

$$\ell_t = \alpha y_t + (1 - \alpha) \ell_{t-1}$$

# A model for SES

## Component form

Forecast equation  $\hat{y}_{t+h|t} = \ell_t$

Smoothing equation  $\ell_t = \alpha y_t + (1 - \alpha) \ell_{t-1}$

Forecast error:  $e_t = y_t - \hat{y}_{t|t-1} = y_t - \ell_{t-1}$ .

# A model for SES

## Component form

Forecast equation  $\hat{y}_{t+h|t} = \ell_t$

Smoothing equation  $\ell_t = \alpha y_t + (1 - \alpha) \ell_{t-1}$

Forecast error:  $e_t = y_t - \hat{y}_{t|t-1} = y_t - \ell_{t-1}$ .

## Error correction form

$$y_t = \ell_{t-1} + e_t$$

$$\ell_t = \ell_{t-1} + \alpha(y_t - \ell_{t-1})$$

$$= \ell_{t-1} + \alpha e_t$$

# A model for SES

## Component form

Forecast equation  $\hat{y}_{t+h|t} = \ell_t$

Smoothing equation  $\ell_t = \alpha y_t + (1 - \alpha) \ell_{t-1}$

Forecast error:  $e_t = y_t - \hat{y}_{t|t-1} = y_t - \ell_{t-1}$ .

## Error correction form

$$y_t = \ell_{t-1} + e_t$$

$$\ell_t = \ell_{t-1} + \alpha(y_t - \ell_{t-1})$$

$$= \ell_{t-1} + \alpha e_t$$

Specify probability distribution for  $e_t$ , we assume

$$e_t = \varepsilon_t \sim \text{NID}(0, \sigma^2).$$

Measurement equation

$$y_t = \ell_{t-1} + \varepsilon_t$$

State equation

$$\ell_t = \ell_{t-1} + \alpha \varepsilon_t$$

where  $\varepsilon_t \sim \text{NID}(0, \sigma^2)$ .

- “innovations” or “single source of error” because same error process,  $\varepsilon_t$ .
- Measurement equation: relationship between observations and states.
- Transition equation(s): evolution of the state(s) through time.

Holt's linear method with additive errors.

- Assume  $\varepsilon_t = y_t - \ell_{t-1} - b_{t-1} \sim \text{NID}(0, \sigma^2)$ .
- Substituting into the error correction equations for Holt's linear method

$$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$$

$$\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$$

$$b_t = b_{t-1} + \alpha \beta^* \varepsilon_t$$

- For simplicity, set  $\beta = \alpha \beta^*$ .

## Your turn

- Write down the model for ETS(A,Ad,N)

## ETS(A,A,A)

Holt-Winters additive method with additive errors.

Forecast equation       $\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$

Observation equation       $y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$

State equations       $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$

$$b_t = b_{t-1} + \beta \varepsilon_t$$

$$s_t = s_{t-m} + \gamma \varepsilon_t$$

■ Forecast errors:  $\varepsilon_t = y_t - \hat{y}_{t|t-1}$

■  $k$  is integer part of  $(h - 1)/m$ .

## Your turn

- Write down the model for ETS(A,N,A)

SES with multiplicative errors.

- Specify relative errors  $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting  $\hat{y}_{t|t-1} = \ell_{t-1}$  gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

SES with multiplicative errors.

- Specify relative errors  $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting  $\hat{y}_{t|t-1} = \ell_{t-1}$  gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Measurement equation

$$y_t = \ell_{t-1}(1 + \varepsilon_t)$$

State equation

$$\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$$

SES with multiplicative errors.

- Specify relative errors  $\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \sim \text{NID}(0, \sigma^2)$
- Substituting  $\hat{y}_{t|t-1} = \ell_{t-1}$  gives:
  - $y_t = \ell_{t-1} + \ell_{t-1}\varepsilon_t$
  - $e_t = y_t - \hat{y}_{t|t-1} = \ell_{t-1}\varepsilon_t$

Measurement equation

$$y_t = \ell_{t-1}(1 + \varepsilon_t)$$

State equation

$$\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$$

- Models with additive and multiplicative errors with the same parameters generate the same point forecasts but different prediction intervals.

Holt's linear method with multiplicative errors.

- Assume  $\varepsilon_t = \frac{y_t - (\ell_{t-1} + b_{t-1})}{(\ell_{t-1} + b_{t-1})}$
- Following a similar approach as above, the innovations state space model underlying Holt's linear method with multiplicative errors is specified as

$$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$$

$$\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$$

$$b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$$

where again  $\beta = \alpha\beta^*$  and  $\varepsilon_t \sim \text{NID}(0, \sigma^2)$ .

# Additive error models

Trend	Seasonal		
	N	A	M
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = \ell_{t-1} s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / \ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$ $b_t = b_{t-1} + \beta \varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$ $b_t = b_{t-1} + \beta \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1}) s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $b_t = b_{t-1} + \beta \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / (\ell_{t-1} + b_{t-1})$
A <sub>d</sub>	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t$ $b_t = \phi b_{t-1} + \beta \varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t$ $b_t = \phi b_{t-1} + \beta \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1}) s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $b_t = \phi b_{t-1} + \beta \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / (\ell_{t-1} + \phi b_{t-1})$

# Multiplicative error models

Trend		Seasonal		
	N	A		M
N	$y_t = \ell_{t-1}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\varepsilon_t$		$y_t = \ell_{t-1}s_{t-m}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A	$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$		$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
<b>A<sub>d</sub></b>	$y_t = (\ell_{t-1} + \phi b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$		$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$

## Estimating ETS models

- Smoothing parameters  $\alpha, \beta, \gamma$  and  $\phi$ , and the initial states  $\ell_0, b_0, s_0, s_{-1}, \dots, s_{-m+1}$  are estimated by maximising the “likelihood” = the probability of the data arising from the specified model.
- For models with additive errors equivalent to minimising SSE.
- For models with multiplicative errors, **not** equivalent to minimising SSE.
- We will estimate models with the `ets()` function in the `forecast` package.

# Innovations state space models

Let  $\mathbf{x}_t = (\ell_t, b_t, s_t, s_{t-1}, \dots, s_{t-m+1})$  and  $\varepsilon_t \stackrel{\text{iid}}{\sim} N(0, \sigma^2)$ .

$$y_t = \underbrace{h(\mathbf{x}_{t-1})}_{\mu_t} + \underbrace{k(\mathbf{x}_{t-1})\varepsilon_t}_{e_t}$$

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + g(\mathbf{x}_{t-1})\varepsilon_t$$

## Additive errors

$$k(x) = 1. \quad y_t = \mu_t + \varepsilon_t.$$

## Multiplicative errors

$$k(\mathbf{x}_{t-1}) = \mu_t. \quad y_t = \mu_t(1 + \varepsilon_t).$$

$\varepsilon_t = (y_t - \mu_t)/\mu_t$  is relative error.

# Innovations state space models

## Estimation

$$\begin{aligned} L^*(\theta, \mathbf{x}_0) &= n \log \left( \sum_{t=1}^n \varepsilon_t^2 / k^2(\mathbf{x}_{t-1}) \right) + 2 \sum_{t=1}^n \log |k(\mathbf{x}_{t-1})| \\ &= -2 \log(\text{Likelihood}) + \text{constant} \end{aligned}$$

- Estimate parameters  $\theta = (\alpha, \beta, \gamma, \phi)$  and initial states  $\mathbf{x}_0 = (\ell_0, b_0, s_0, s_{-1}, \dots, s_{-m+1})$  by minimizing  $L^*$ .

# Parameter restrictions

## *Usual region*

- Traditional restrictions in the methods  $0 < \alpha, \beta^*, \gamma^*, \phi < 1$  (equations interpreted as weighted averages).
- In models we set  $\beta = \alpha\beta^*$  and  $\gamma = (1 - \alpha)\gamma^*$ .
- Therefore  $0 < \alpha < 1$ ,  $0 < \beta < \alpha$  and  $0 < \gamma < 1 - \alpha$ .
- $0.8 < \phi < 0.98$  — to prevent numerical difficulties.

# Parameter restrictions

## Usual region

- Traditional restrictions in the methods  $0 < \alpha, \beta^*, \gamma^*, \phi < 1$  (equations interpreted as weighted averages).
- In models we set  $\beta = \alpha\beta^*$  and  $\gamma = (1 - \alpha)\gamma^*$ .
- Therefore  $0 < \alpha < 1$ ,  $0 < \beta < \alpha$  and  $0 < \gamma < 1 - \alpha$ .
- $0.8 < \phi < 0.98$  — to prevent numerical difficulties.

## Admissible region

- To prevent observations in the distant past having a continuing effect on current forecasts.
- Usually (but not always) less restrictive than the *traditional* region.
- For example for ETS(A,N,N):  
*traditional*  $0 < \alpha < 1$  — *admissible* is  $0 < \alpha < 2$ .

# Model selection

## Akaike's Information Criterion

$$AIC = -2 \log(L) + 2k$$

where  $L$  is the likelihood and  $k$  is the number of parameters initial states estimated in the model.

# Model selection

## Akaike's Information Criterion

$$AIC = -2 \log(L) + 2k$$

where  $L$  is the likelihood and  $k$  is the number of parameters initial states estimated in the model.

## Corrected AIC

$$AIC_c = AIC + \frac{2(k+1)(k+2)}{T-k}$$

which is the AIC corrected (for small sample bias).

# Model selection

## Akaike's Information Criterion

$$AIC = -2 \log(L) + 2k$$

where  $L$  is the likelihood and  $k$  is the number of parameters initial states estimated in the model.

## Corrected AIC

$$AIC_c = AIC + \frac{2(k+1)(k+2)}{T-k}$$

which is the AIC corrected (for small sample bias).

## Bayesian Information Criterion

$$BIC = AIC + k(\log(T) - 2).$$

# Automatic forecasting

From Hyndman et al. (IJF, 2002):

- Apply each model that is appropriate to the data. Optimize parameters and initial values using MLE (or some other criterion).
- Select best method using AICc:
- Produce forecasts using best method.
- Obtain forecast intervals using underlying state space model.

Method performed very well in M3 competition.

## Some unstable models

- Some of the combinations of (Error, Trend, Seasonal) can lead to numerical difficulties; see equations with division by a state.
- These are: ETS(A,N,M), ETS(A,A,M), ETS(A,A<sub>d</sub>,M).
- Models with multiplicative errors are useful for strictly positive data, but are not numerically stable with data containing zeros or negative values. In that case only the six fully additive models will be applied.

# Exponential smoothing models

Additive Error		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	A,N,N	A,N,A	<del>A,N,M</del>
A	(Additive)	A,A,N	A,A,A	<del>A,A,M</del>
A <sub>d</sub>	(Additive damped)	A,A <sub>d</sub> ,N	A,A <sub>d</sub> ,A	<del>A,A<sub>d</sub>,M</del>

Multiplicative Error		Seasonal Component		
Trend Component		N (None)	A (Additive)	M (Multiplicative)
N	(None)	M,N,N	M,N,A	M,N,M
A	(Additive)	M,A,N	M,A,A	M,A,M
A <sub>d</sub>	(Additive damped)	M,A <sub>d</sub> ,N	M,A <sub>d</sub> ,A	M,A <sub>d</sub> ,M

## Example: International tourists

```
aust <- window(austourists, start=2005)
```

```
fit <- ets(aust)
```

```
summary(fit)
```

```
## ETS(M,A,M)
```

```
##
```

```
## Call:
```

```
## ets(y = aust)
```

```
##
```

```
## Smoothing parameters:
```

```
##     alpha = 0.1908
```

```
##     beta  = 0.0392
```

```
##     gamma = 2e-04
```

```
##
```

```
## Initial states:
```

```
##     l = 32.3679
```

```
##     b = 0.9281
```

```
##     s = 1.022 0.9628 0.7683 1.247
```

```
##
```

```
## sigma: 0.0383
```

```
##
```

```
## AIC  AICc    BIC
```

```
## 224.9 230.2 240.9
```

```
##
```

## Example: International tourists

Model selected: ETS(M,A,M)

$$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}(1 + \varepsilon_t)$$

$$\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$$

$$b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$$

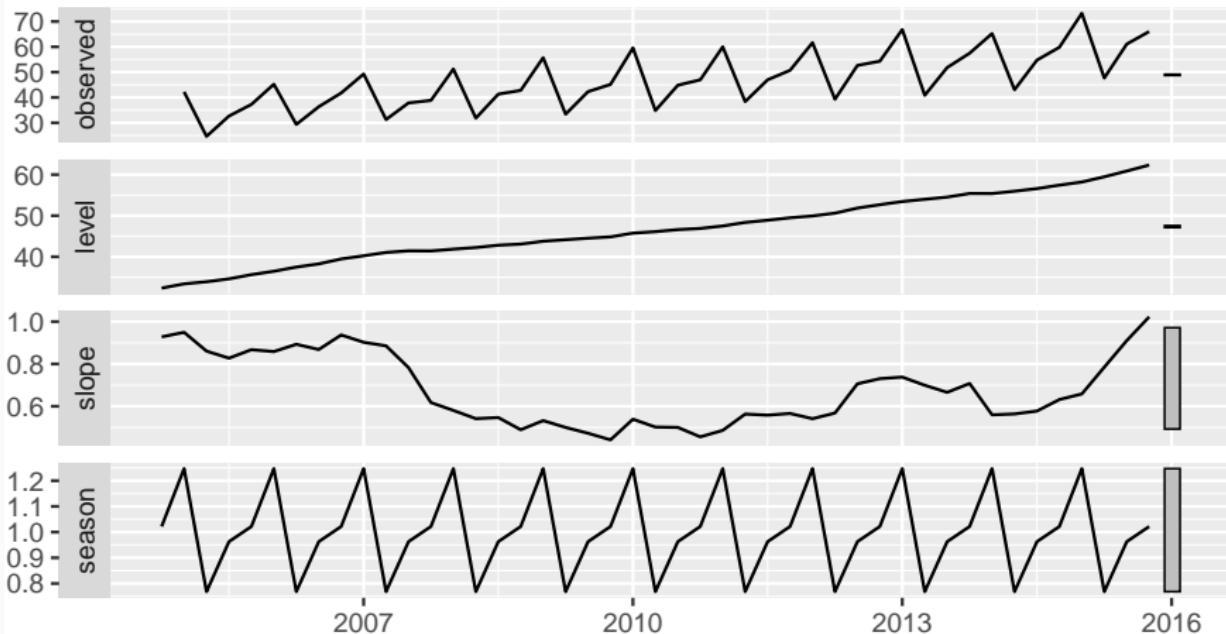
$$s_t = s_{t-m}(1 + \gamma\varepsilon_t).$$

$\hat{\alpha} = 0.1908$ ,  $\hat{\beta} = 0.0392$ , and  $\hat{\gamma} = 0.00019$ .

## Example: International tourists

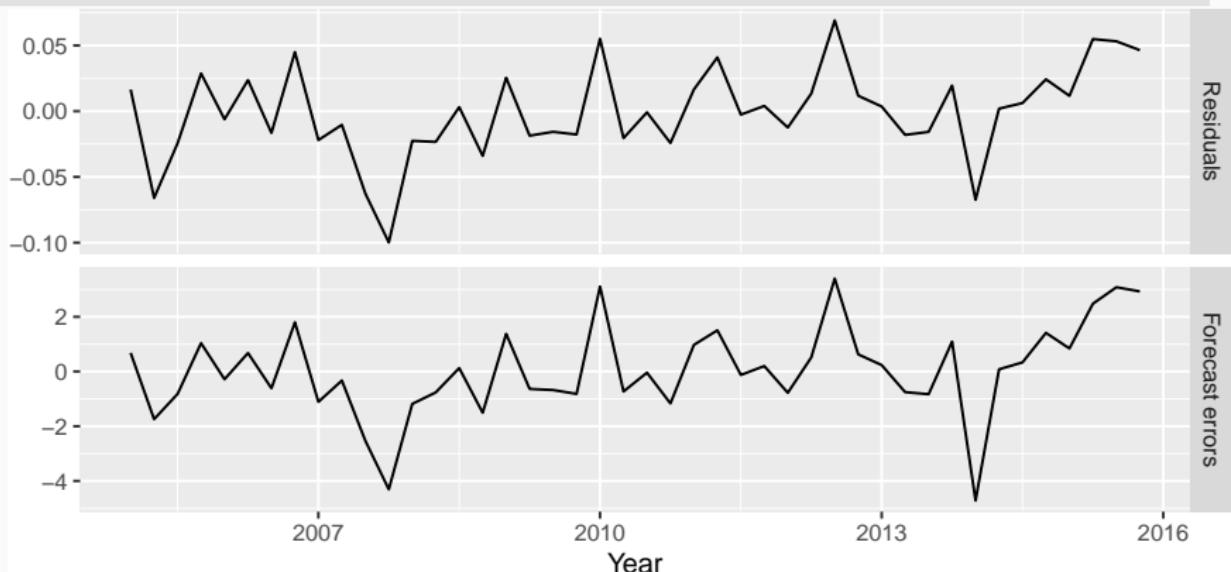
```
autoplot(fit)
```

Components of ETS(M,A,M) method



## Example: International tourists

```
cbind('Residuals' = residuals(fit),  
      'Forecast errors' = residuals(fit, type='response')) %>%  
  autoplot(facet=TRUE) + xlab("Year") + ylab("")
```



# Residuals

## Response residuals

$$\hat{e}_t = y_t - \hat{y}_{t|t-1}$$

## Innovation residuals

Additive error model:

$$\hat{\varepsilon}_t = y_t - \hat{y}_{t|t-1}$$

Multiplicative error model:

$$\hat{\varepsilon}_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}}$$

## Forecasting with ETS models

**Point forecasts:** iterate the equations for  $t = T + 1, T + 2, \dots, T + h$  and set all  $\varepsilon_t = 0$  for  $t > T$ .

## Forecasting with ETS models

**Point forecasts:** iterate the equations for  $t = T + 1, T + 2, \dots, T + h$  and set all  $\varepsilon_t = 0$  for  $t > T$ .

- Not the same as  $E(y_{t+h} | \mathbf{x}_t)$  unless trend and seasonality are both additive.
- Point forecasts for  $\text{ETS}(A,x,y)$  are identical to  $\text{ETS}(M,x,y)$  if the parameters are the same.

## Example: ETS(A,A,N)

$$y_{T+1} = \ell_T + b_T + \varepsilon_{T+1}$$

$$\hat{y}_{T+1|T} = \ell_T + b_T$$

$$y_{T+2} = \ell_{T+1} + b_{T+1} + \varepsilon_{T+2}$$

$$= (\ell_T + b_T + \alpha \varepsilon_{T+1}) + (b_T + \beta \varepsilon_{T+1}) + \varepsilon_{T+2}$$

$$\hat{y}_{T+2|T} = \ell_T + 2b_T$$

etc.

## Example: ETS(M,A,N)

$$y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$$

$$\hat{y}_{T+1|T} = \ell_T + b_T.$$

$$y_{T+2} = (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+2})$$

$$= \{(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + [b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}]\} (1 + \varepsilon_{T+2})$$

$$\hat{y}_{T+2|T} = \ell_T + 2b_T$$

etc.

## Forecasting with ETS models

**Prediction intervals:** cannot be generated using the methods, only the models.

- The prediction intervals will differ between models with additive and multiplicative errors.
- Exact formulae for some models.
- More general to simulate future sample paths, conditional on the last estimate of the states, and to obtain prediction intervals from the percentiles of these simulated future paths.
- Options are available in R using the `forecast` function in the `forecast` package.

## Prediction intervals

PI for most ETS models:  $\hat{y}_{T+h|T} \pm c\sigma_h$ , where  $c$  depends on coverage probability and  $\sigma_h$  is forecast standard deviation.

$$(A,N,N) \quad \sigma_h = \sigma^2 [1 + \alpha^2(h - 1)]$$

$$(A,A,N) \quad \sigma_h = \sigma^2 \left[ 1 + (h - 1) \{ \alpha^2 + \alpha\beta h + \frac{1}{6}\beta^2 h(2h - 1) \} \right]$$

$$\begin{aligned} (A,A_d,N) \quad \sigma_h = \sigma^2 & \left[ 1 + \alpha^2(h - 1) + \frac{\beta\phi h}{(1-\phi)^2} \{ 2\alpha(1 - \phi) + \beta\phi \} \right. \\ & \left. - \frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)} \{ 2\alpha(1 - \phi^2) + \beta\phi(1 + 2\phi - \phi^h) \} \right] \end{aligned}$$

$$(A,N,A) \quad \sigma_h = \sigma^2 \left[ 1 + \alpha^2(h - 1) + \gamma k(2\alpha + \gamma) \right]$$

$$(A,A,A) \quad \sigma_h = \sigma^2 \left[ 1 + (h - 1) \{ \alpha^2 + \alpha\beta h + \frac{1}{6}\beta^2 h(2h - 1) \} + \gamma k \{ 2\alpha + \gamma + \beta m(k + 1) \} \right]$$

$$\begin{aligned} (A,A_d,A) \quad \sigma_h = \sigma^2 & \left[ 1 + \alpha^2(h - 1) + \frac{\beta\phi h}{(1-\phi)^2} \{ 2\alpha(1 - \phi) + \beta\phi \} \right. \\ & \left. - \frac{\beta\phi(1-\phi^h)}{(1-\phi)^2(1-\phi^2)} \{ 2\alpha(1 - \phi^2) + \beta\phi(1 + 2\phi - \phi^h) \} \right. \\ & \left. + \gamma k(2\alpha + \gamma) + \frac{2\beta\gamma\phi}{(1-\phi)(1-\phi^m)} \{ k(1 - \phi^m) - \phi^m(1 - \phi^{mk}) \} \right] \end{aligned}$$

# Outline

- 1 Simple exponential smoothing
- 2 Trend methods
- 3 Seasonal methods
- 4 Taxonomy of exponential smoothing methods
- 5 Innovations state space models
- 6 ETS in R

## Example: drug sales

```
ets(h02)
## ETS(M,Ad,M)
##
## Call:
##   ets(y = h02)
##
##   Smoothing parameters:
##     alpha = 0.1953
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9798
##
##   Initial states:
##     l = 0.3945
##     b = 0.0085
##     s = 0.874 0.8197 0.7644 0.7693 0.6941 1.284
##           1.326 1.177 1.162 1.095 1.042 0.9924
##
##   sigma:  0.0676
##
##     AIC      AICc      BIC
## -122.91 -119.21  -63.18
```

## Example: drug sales

```
ets(h02, model="AAA", damped=FALSE)

## ETS(A,A,A)
##
## Call:
##   ets(y = h02, model = "AAA", damped = FALSE)
##
## Smoothing parameters:
##   alpha = 0.1672
##   beta  = 0.0084
##   gamma = 1e-04
##
## Initial states:
##   l = 0.3895
##   b = 0.0116
##   s = -0.1058 -0.1359 -0.1875 -0.1803 -0.2414 0.2097
##           0.2493 0.1426 0.1411 0.0823 0.0293 -0.0033
##
## sigma:  0.0642
##
##      AIC    AICc     BIC
## -18.26 -14.97  38.14
```

## The `ets()` function

- Automatically chooses a model by default using the AIC, AICc or BIC.
- Can handle any combination of trend, seasonality and damping
- Ensures the parameters are admissible (equivalent to invertible)
- Produces an object of class “ets”.

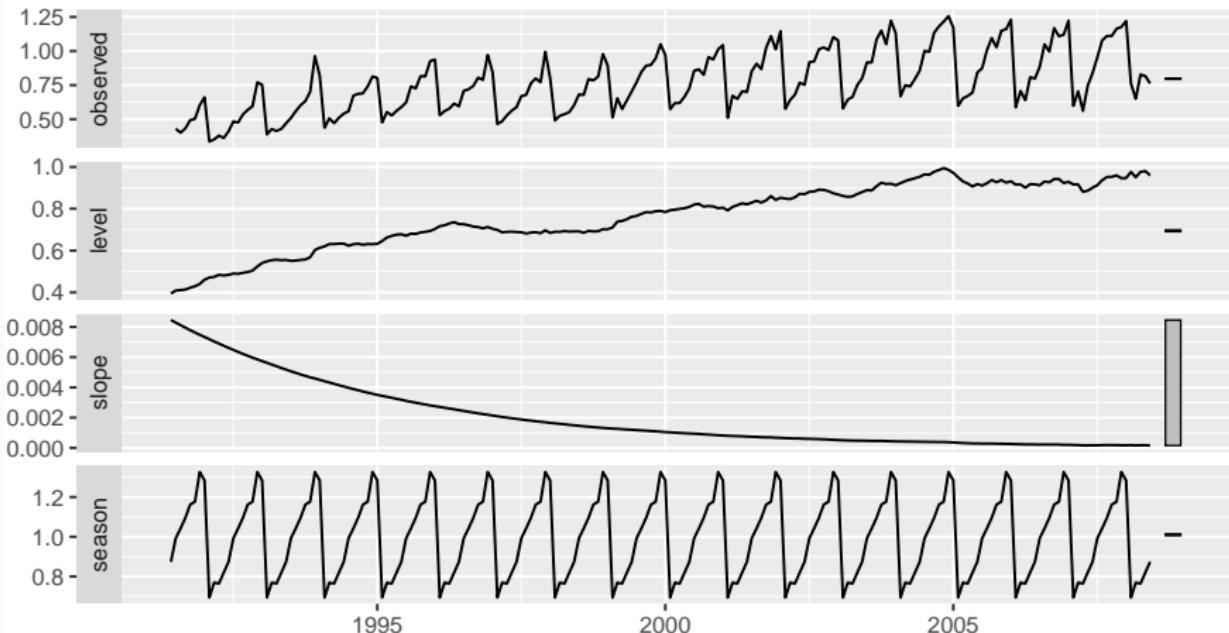
## ets objects

- **Methods:** `coef()`, `autoplot()`, `plot()`, `summary()`,  
`residuals()`, `fitted()`, `simulate()` and `forecast()`
- `autoplot()` shows time plots of the original time series  
along with the extracted components (level, growth and  
seasonal).

## Example: drug sales

```
h02 %>% ets() %>% autoplot()
```

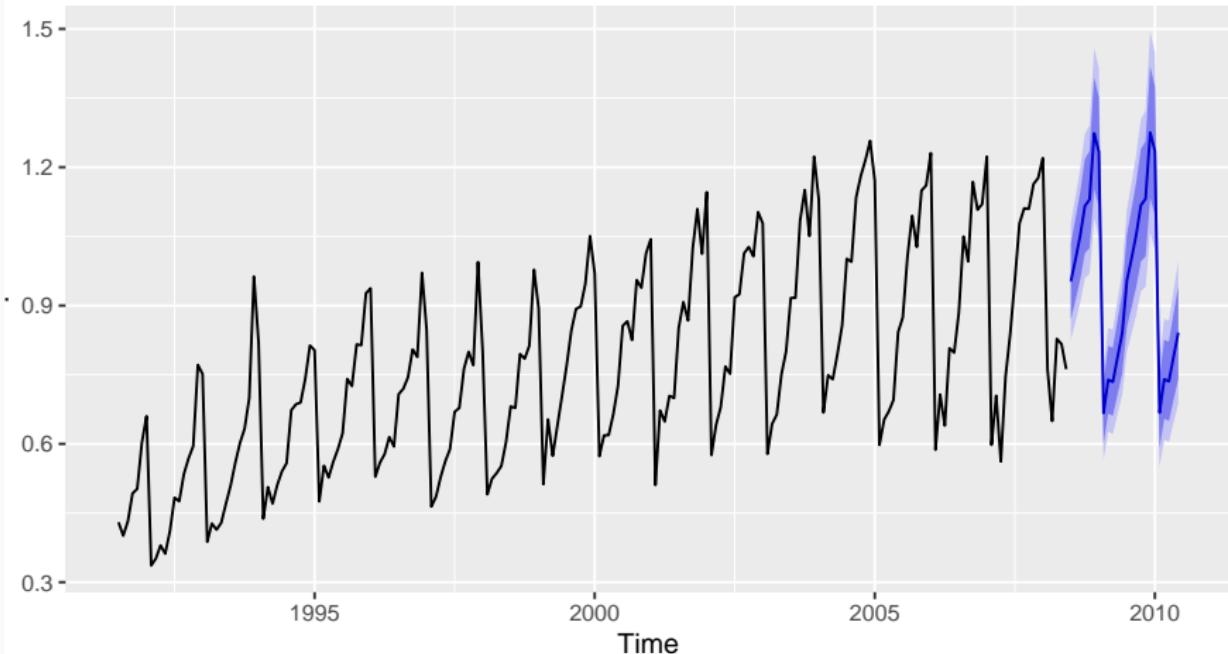
Components of ETS(M,Ad,M) method



## Example: drug sales

```
h02 %>% ets() %>% forecast() %>% autoplot()
```

Forecasts from ETS(M,Ad,M)



## Example: drug sales

```
h02 %>% ets() %>% accuracy()  
##  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.003873 0.05097 0.03904 0.1125 5.046 0.644 0  
  
h02 %>% ets(model="AAA", damped=FALSE) %>% accuracy()  
##  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.006447 0.0616 0.04949 -1.258 7.142 0.8164
```

# The ets() function

ets() function also allows refitting model to new data set.

```
train <- window(h02, end=c(2004,12))
test <- window(h02, start=2005)
fit1 <- ets(train)
fit2 <- ets(test, model = fit1)
```

## Model is being refit with current smoothing parameters but initial states are being estimated.

## Set 'use.initial.values=TRUE' if you want to re-use existing initial values.

```
accuracy(fit2)
```

##	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
----	----	------	-----	-----	------	------	------

## Training set 0.00144 0.05406 0.04314 -0.4332 5.218 0.6785 -0.4121

```
accuracy(forecast(fit1,10), test)
```

##	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
----	----	------	-----	-----	------	------	------	-----------

## Training set 0.003427 0.04453 0.03290 0.1589 4.364 0.558 0.02236 NA

## Test set -0.077245 0.09158 0.07955 -10.0413 10.252 1.349 -0.04361 0.6333

## The `ets()` function in R

```
ets(y, model = "ZZZ", damped = NULL,  
    additive.only = FALSE,  
    lambda = NULL, biasadj = FALSE,  
    lower = c(rep(1e-04, 3), 0.8),  
    upper = c(rep(0.9999, 3), 0.98),  
    opt.crit = c("lik", "amse", "mse", "sigma", "mae"),  
    nmse = 3,  
    bounds = c("both", "usual", "admissible"),  
    ic = c("aicc", "aic", "bic"),  
    restrict = TRUE,  
    allow.multiplicative.trend = FALSE, ...)
```

## The `ets()` function in R

- `y`  
The time series to be forecast.
- `model`  
use the ETS classification and notation: “N” for none, “A” for additive, “M” for multiplicative, or “Z” for automatic selection. Default zzz all components are selected using the information criterion.
- `damped`
  - If `damped=TRUE`, then a damped trend will be used (either  $A_d$  or  $M_d$ ).
  - `damped=FALSE`, then a non-damped trend will be used.
  - If `damped=NULL` (default), then either a damped or a non-damped trend will be selected according to the information criterion chosen.

## The `ets()` function in R

- `additive.only`

Only models with additive components will be considered if `additive.only=TRUE`. Otherwise all models will be considered.

- `lambda`

Box-Cox transformation parameter. It will be ignored if `lambda=NULL` (default). Otherwise, the time series will be transformed before the model is estimated. When `lambda` is not `NULL`, `additive.only` is set to `TRUE`.

- `biadadj`

Uses bias-adjustment when undoing Box-Cox transformation for fitted values.

## The `ets()` function in R

- lower,upper bounds for the parameter estimates of  $\alpha$ ,  $\beta^*$ ,  $\gamma^*$  and  $\phi$ .
- `opt.crit=lik` (default) optimisation criterion used for estimation.
- bounds Constraints on the parameters.
  - usual region – "bounds=usual";
  - admissible region – "bounds=admissible";
  - "bounds=both" (default) requires the parameters to satisfy both sets of constraints.
- `ic=aicc` (default) information criterion to be used in selecting models.
- `restrict=TRUE` (default) models that cause numerical problems not considered in model selection.
- `allow.multiplicative.trend` allows models with a multiplicative trend.

## The `forecast()` function in R

```
forecast(object,
  h=ifelse(object$m>1, 2*object$m, 10),
  level=c(80,95), fan=FALSE,
  simulate=FALSE, bootstrap=FALSE,
  npaths=5000, PI=TRUE,
  lambda=object$lambda, biasadj=FALSE,...)
```

- object: the object returned by the `ets()` function.
- h: the number of periods to be forecast.
- level: the confidence level for the prediction intervals.
- fan: if `fan=TRUE`, suitable for fan plots.

## The `forecast()` function in R

- `simulate`: If `TRUE`, prediction intervals generated via simulation rather than analytic formulae. Even if `FALSE` simulation will be used if no algebraic formulae exist.
- `bootstrap`: If `bootstrap=TRUE` and `simulate=TRUE`, then simulated prediction intervals use re-sampled errors rather than normally distributed errors.
- `npaths`: The number of sample paths used in computing simulated prediction intervals.
- `PI`: If `PI=TRUE`, then prediction intervals are produced; otherwise only point forecasts are calculated. If `PI=FALSE`, then `level`, `fan`, `simulate`, `bootstrap` and `npaths` are all ignored.

## The `forecast()` function in R

- `lambda`: The Box-Cox transformation parameter. Ignored if `lambda=NULL`. Otherwise, forecasts are back-transformed via inverse Box-Cox transformation.
- `biasadj`: Apply bias adjustment after Box-Cox?

## Your turn

- Use `ets()` on some of these series:  
*bicoal, chicken, dole, usdeaths, bricksq, lynx,  
ibmclose, eggs, bricksq, ausbeer*
- Does it always give good forecasts?
- Find an example where it does not work well. Can you figure out why?



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# STU33010: Forecasting

Ch8. ARIMA

[OTexts.org/fpp2/](https://OTexts.org/fpp2/)

# Outline

- 1 Stationarity and differencing**
- 2 Non-seasonal ARIMA models**
- 3 Estimation and order selection**
- 4 ARIMA modelling in R**
- 5 Forecasting**
- 6 Seasonal ARIMA models**
- 7 ARIMA vs ETS**

# Stationarity

## Definition

If  $\{y_t\}$  is a stationary time series, then for all  $s$ , the distribution of  $(y_t, \dots, y_{t+s})$  does not depend on  $t$ .

# Stationarity

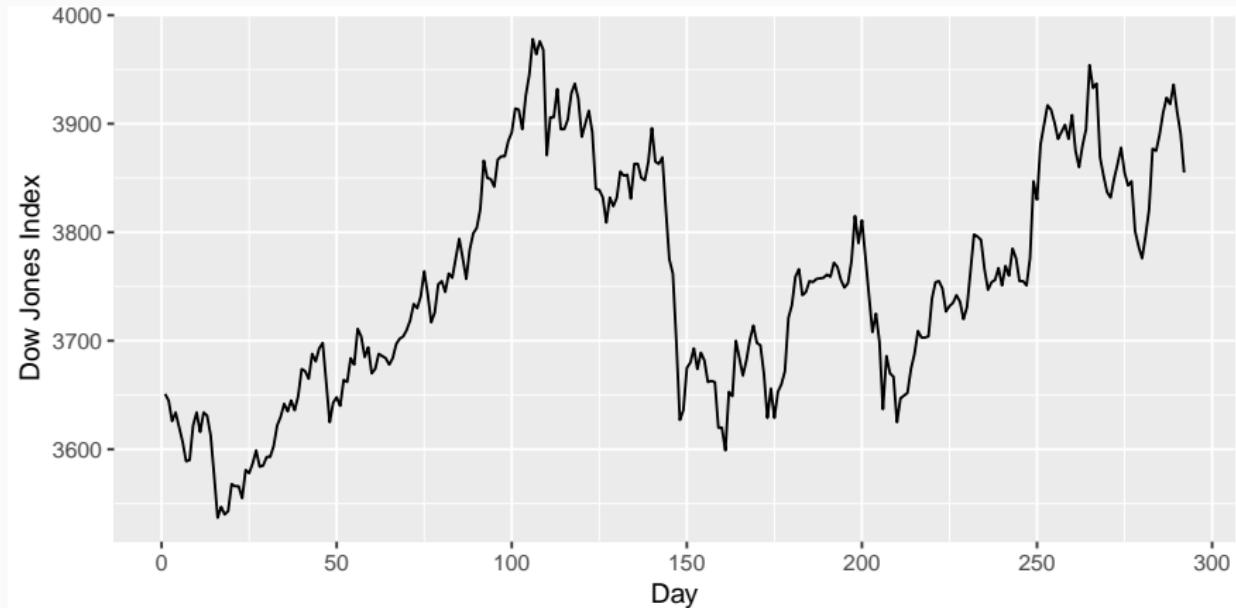
## Definition

If  $\{y_t\}$  is a stationary time series, then for all  $s$ , the distribution of  $(y_t, \dots, y_{t+s})$  does not depend on  $t$ .

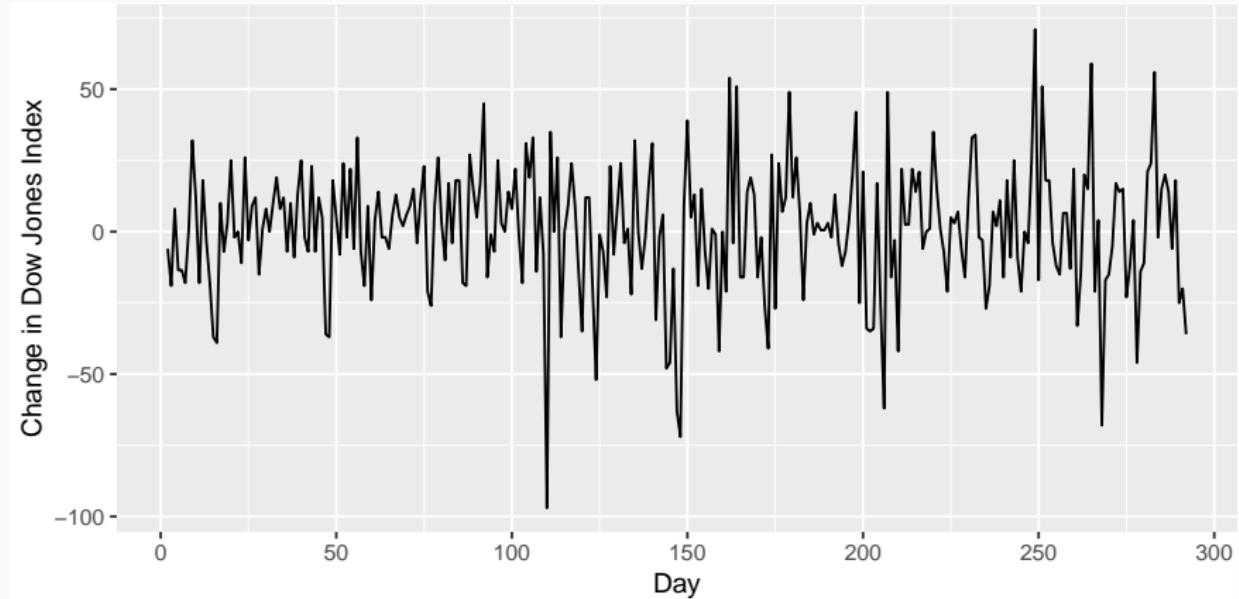
A **stationary series** is:

- roughly horizontal
- constant variance
- no patterns predictable in the long-term

# Stationary?



# Stationary?

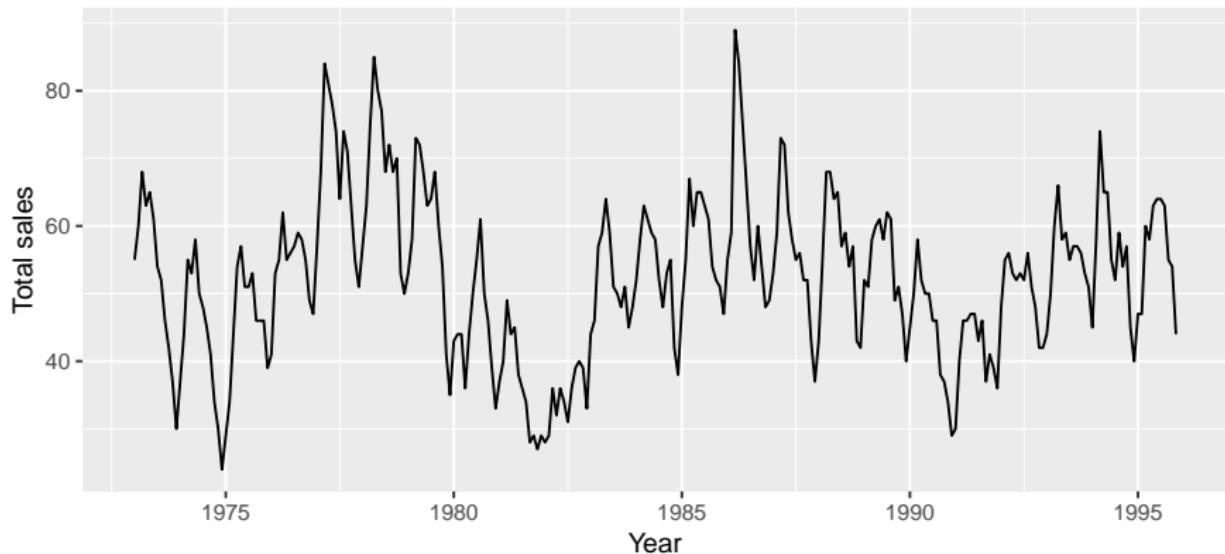


# Stationary?



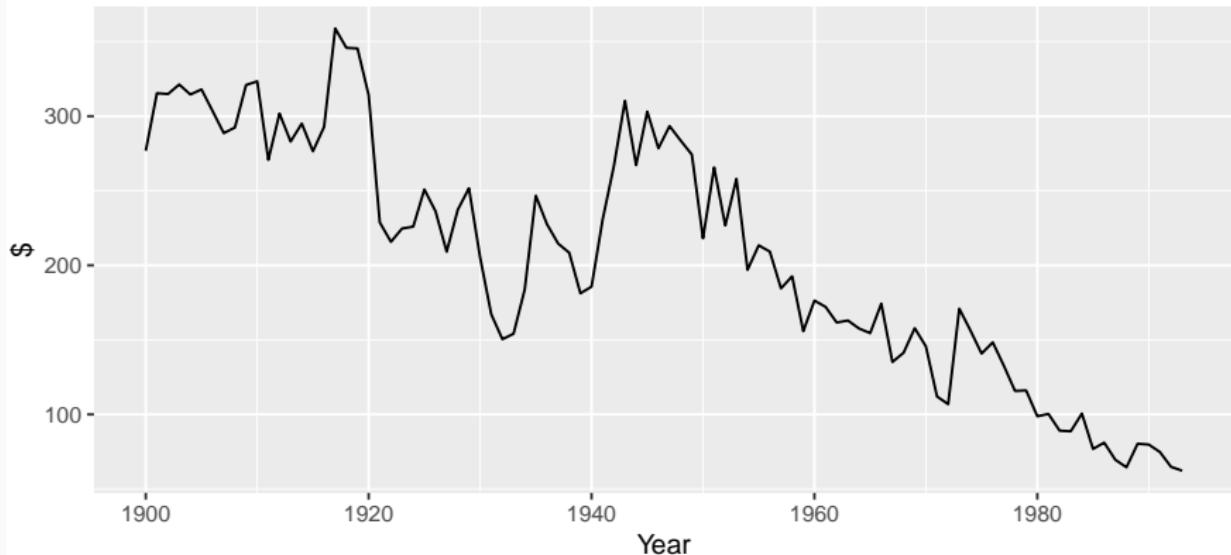
# Stationary?

Sales of new one-family houses, USA



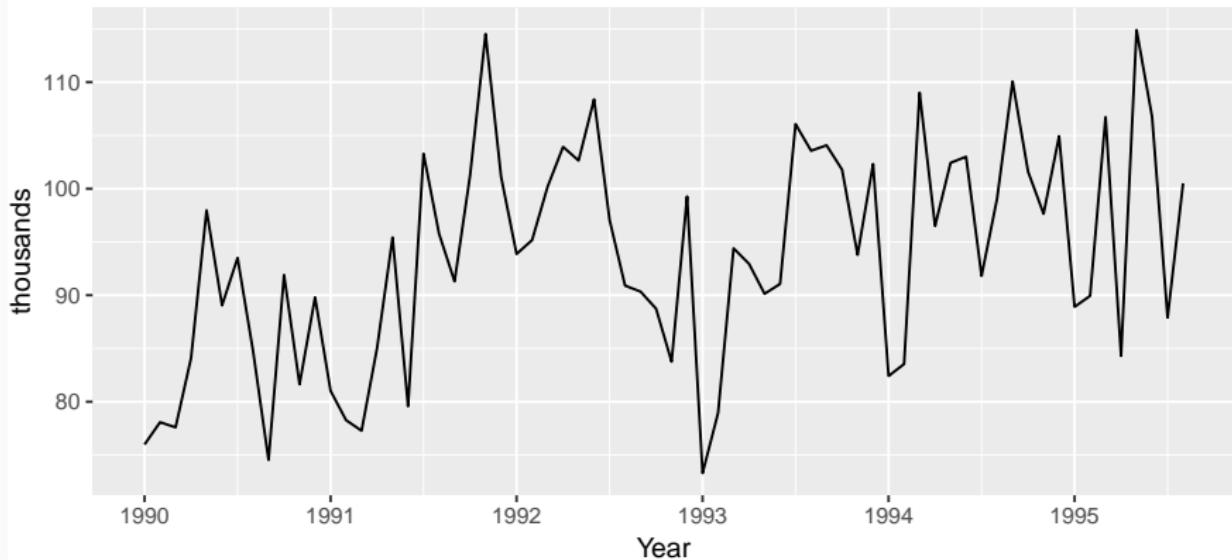
# Stationary?

Price of a dozen eggs in 1993 dollars



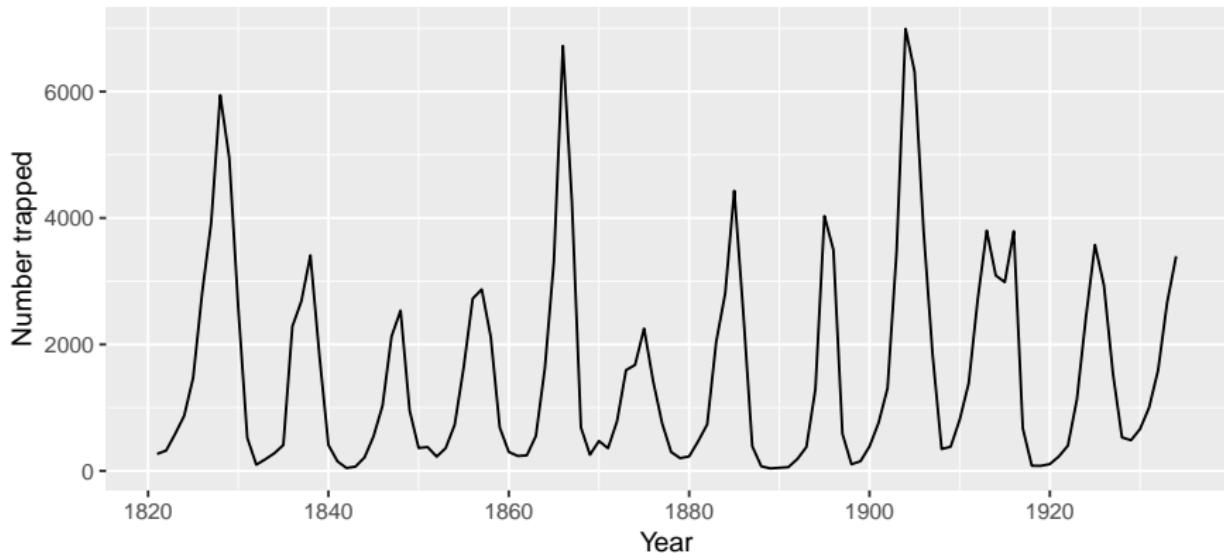
# Stationary?

Number of pigs slaughtered in Victoria



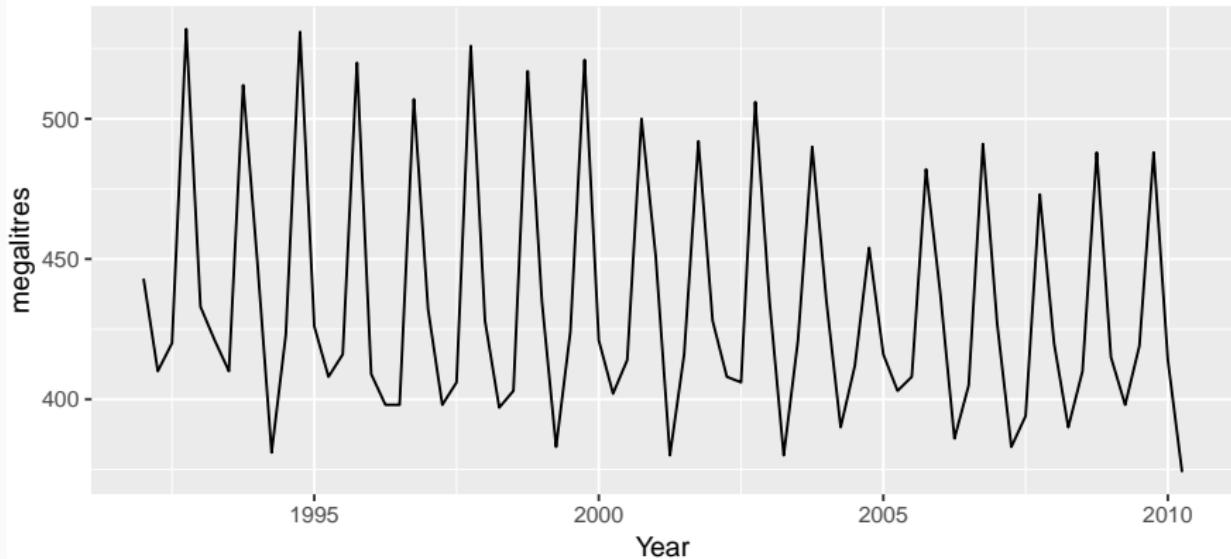
# Stationary?

Annual Canadian Lynx Trappings



# Stationary?

Australian quarterly beer production



# Stationarity

## Definition

If  $\{y_t\}$  is a stationary time series, then for all  $s$ , the distribution of  $(y_t, \dots, y_{t+s})$  does not depend on  $t$ .

# Stationarity

## Definition

If  $\{y_t\}$  is a stationary time series, then for all  $s$ , the distribution of  $(y_t, \dots, y_{t+s})$  does not depend on  $t$ .

Transformations help to **stabilize the variance**.

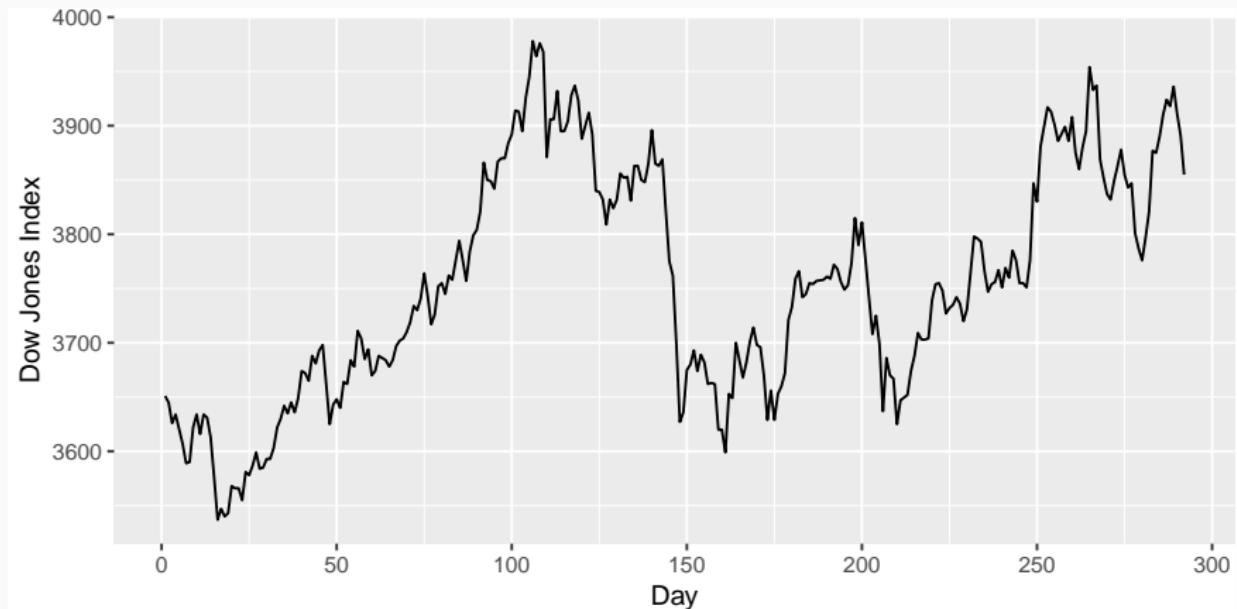
For ARIMA modelling, we also need to **stabilize the mean**.

# Non-stationarity in the mean

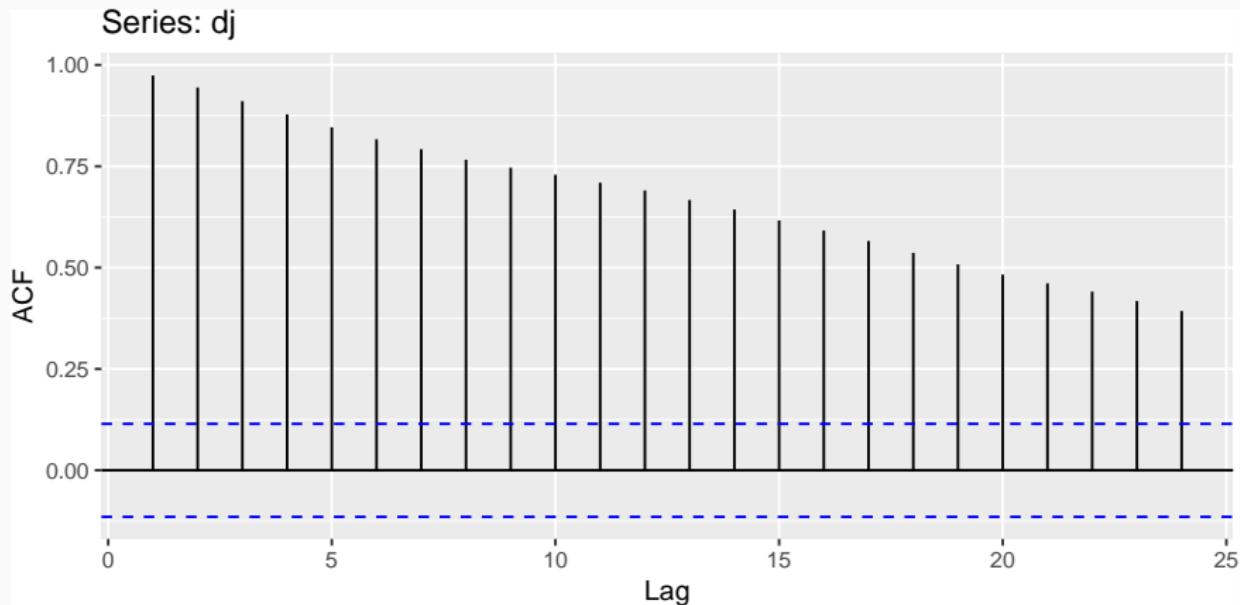
## Identifying non-stationary series

- time plot.
- The ACF of stationary data drops to zero relatively quickly
- The ACF of non-stationary data decreases slowly.
- For non-stationary data, the value of  $r_1$  is often large and positive.

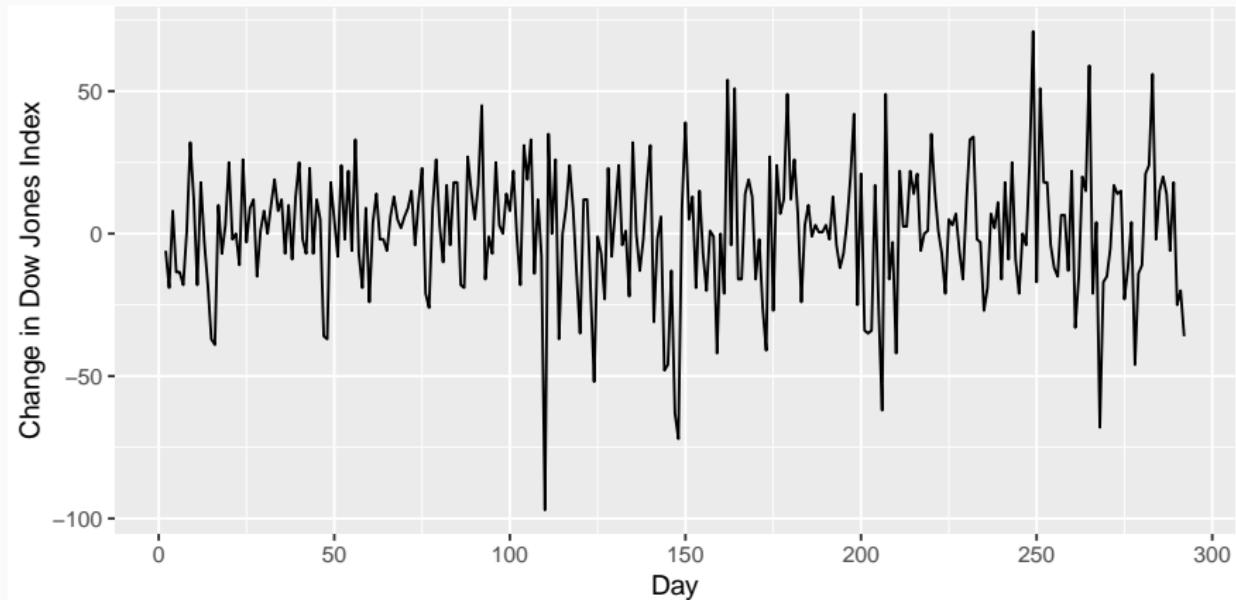
# Example: Dow-Jones index



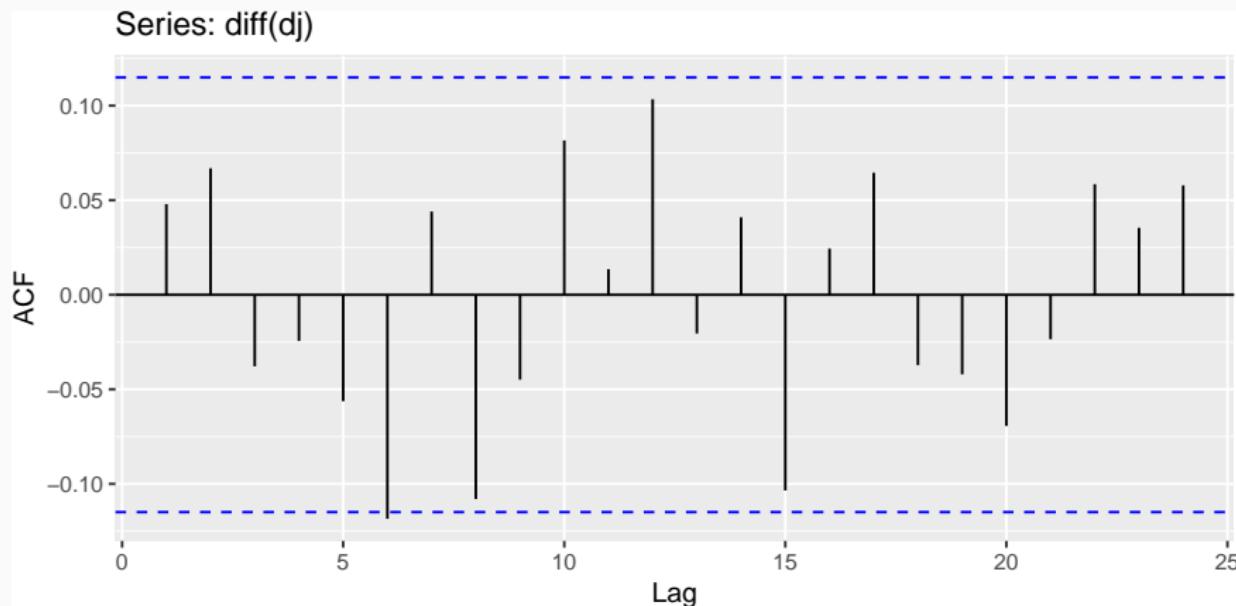
# Example: Dow-Jones index



# Example: Dow-Jones index



# Example: Dow-Jones index



# Differencing

- Differencing helps to **stabilize the mean**.
- The differenced series is the *change* between each observation in the original series:

$$y'_t = y_t - y_{t-1}.$$

- The differenced series will have only  $T - 1$  values since it is not possible to calculate a difference  $y'_1$  for the first observation.

## Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

## Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

$$\begin{aligned}y_t'' &= y_t' - y_{t-1}' \\&= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\&= y_t - 2y_{t-1} + y_{t-2}.\end{aligned}$$

## Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

$$\begin{aligned}y_t'' &= y'_t - y'_{t-1} \\&= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\&= y_t - 2y_{t-1} + y_{t-2}.\end{aligned}$$

- $y_t''$  will have  $T - 2$  values.
- In practice, it is almost never necessary to go beyond second-order differences.

## Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

## Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

$$y'_t = y_t - y_{t-m}$$

where  $m$  = number of seasons.

## Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

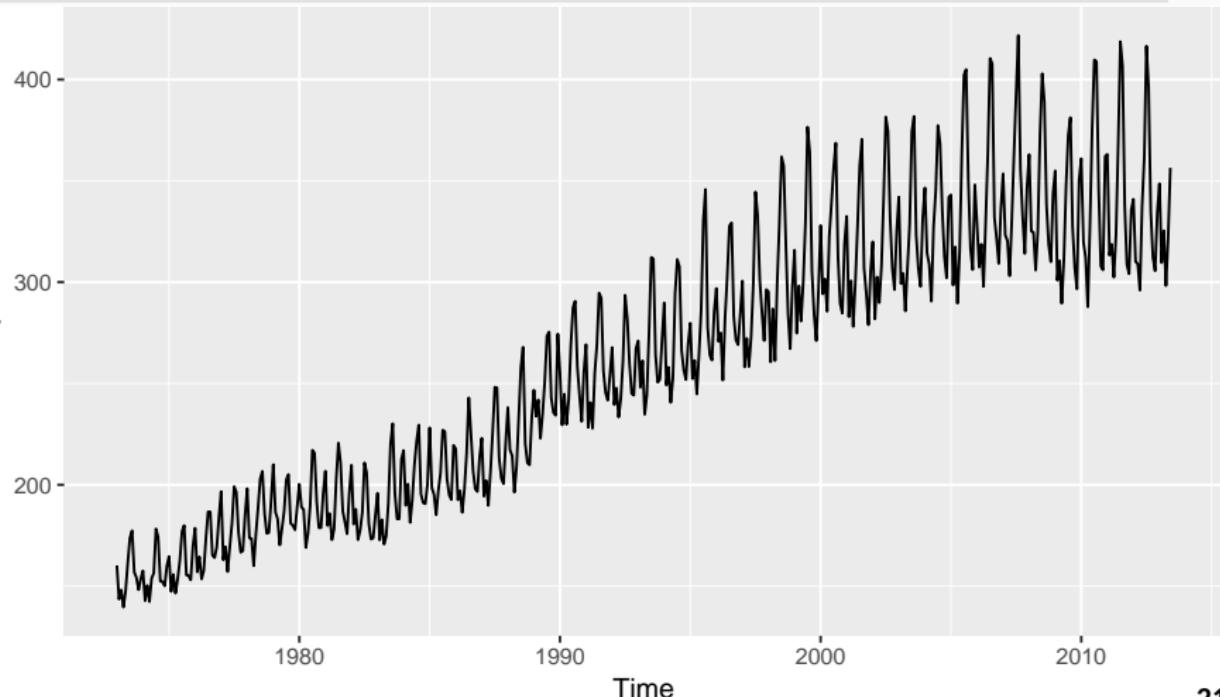
$$y'_t = y_t - y_{t-m}$$

where  $m$  = number of seasons.

- For monthly data  $m = 12$ .
- For quarterly data  $m = 4$ .

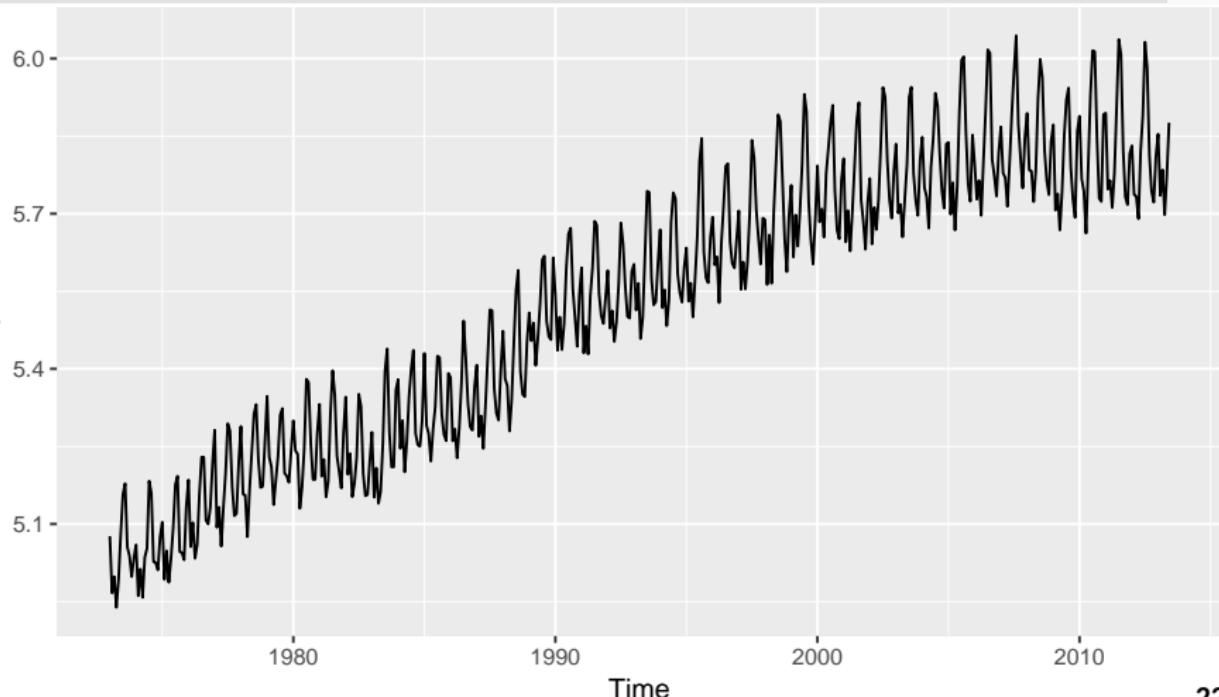
# Electricity production

```
usmlelec %>% autoplot()
```



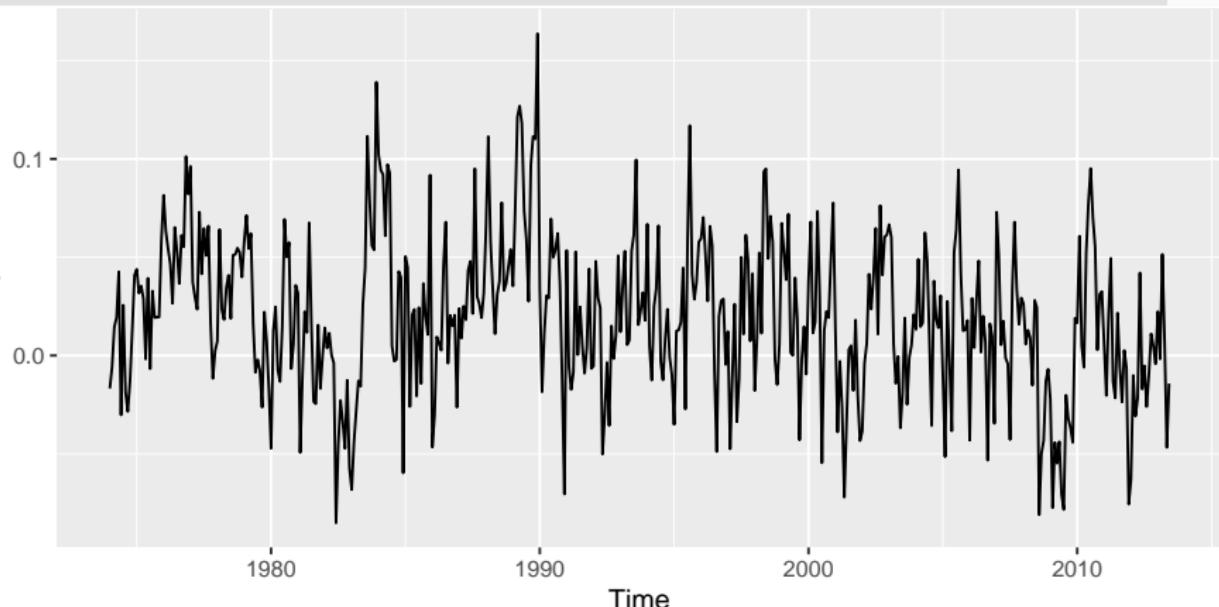
# Electricity production

```
usmlelec %>% log() %>% autoplot()
```



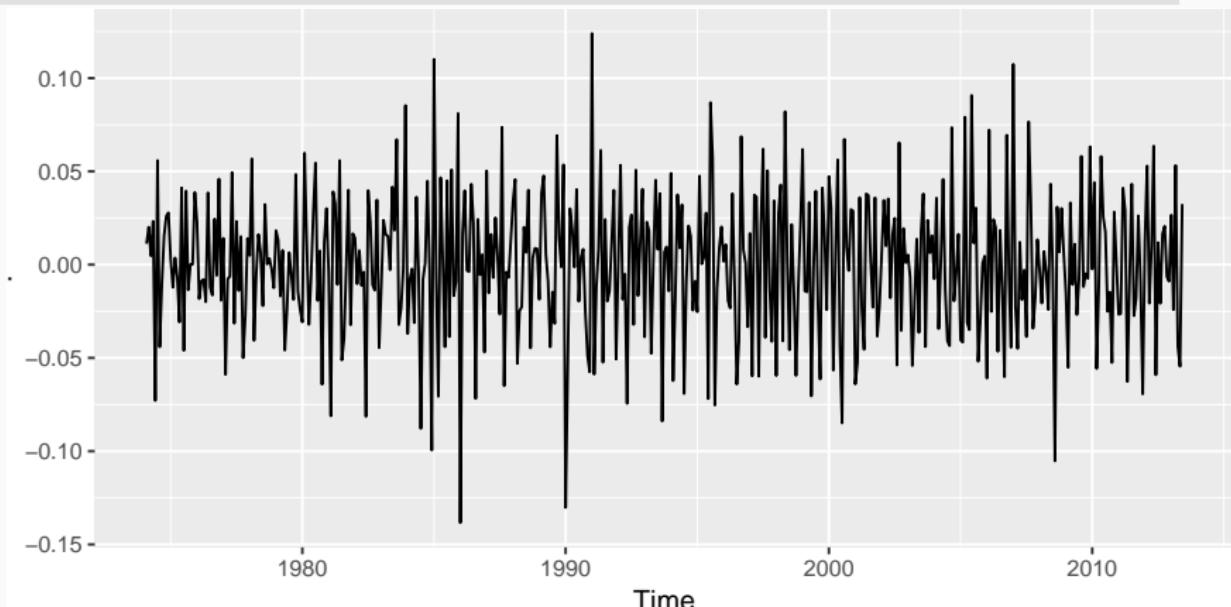
# Electricity production

```
usmlelec %>% log() %>% diff(lag=12) %>%  
  autoplot()
```



# Electricity production

```
usmlelec %>% log() %>% diff(lag=12) %>%  
  diff(lag=1) %>% autoplot()
```



# Electricity production

- Seasonally differenced series is closer to being stationary.
- Remaining non-stationarity can be removed with further first difference.

If  $y'_t = y_t - y_{t-12}$  denotes seasonally differenced series,  
then twice-differenced series is

$$\begin{aligned}y_t^* &= y'_t - y'_{t-1} \\&= (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) \\&= y_t - y_{t-1} - y_{t-12} + y_{t-13}.\end{aligned}$$

## Seasonal differencing

When both seasonal and first differences are applied...

# Seasonal differencing

When both seasonal and first differences are applied...

- it makes no difference which is done first—the result will be the same.
- If seasonality is strong, we recommend that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for further first difference.

## Seasonal differencing

When both seasonal and first differences are applied...

- it makes no difference which is done first—the result will be the same.
- If seasonality is strong, we recommend that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for further first difference.

It is important that if differencing is used, the differences are interpretable.

# Interpretation of differencing

- first differences are the change between **one observation and the next**;
- seasonal differences are the change between **one year to the next**.

# Interpretation of differencing

- first differences are the change between **one observation and the next**;
- seasonal differences are the change between **one year to the next**.

But taking lag 3 differences for yearly data, for example, results in a model which cannot be sensibly interpreted.

# Unit root tests

**Statistical tests to determine the required order of differencing.**

- 1 Augmented Dickey Fuller test: null hypothesis is that the data are non-stationary and non-seasonal.
- 2 Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test: null hypothesis is that the data are stationary and non-seasonal.
- 3 Other tests available for seasonal data.

# KPSS test

```
library(urca)
summary(ur.kpss(goog))

##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 10.7223
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

# KPSS test

```
library(urca)
summary(ur.kpss(goog))

##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 10.7223
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347  0.463  0.574  0.739

ndiffs(goog)

## [1] 1
```

## Automatically selecting differences

STL decomposition:  $y_t = T_t + S_t + R_t$

Seasonal strength  $F_s = \max \left( 0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(S_t+R_t)} \right)$

If  $F_s > 0.64$ , do one seasonal difference.

# Automatically selecting differences

STL decomposition:  $y_t = T_t + S_t + R_t$

Seasonal strength  $F_s = \max \left( 0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(S_t+R_t)} \right)$

If  $F_s > 0.64$ , do one seasonal difference.

```
usmelec %>% log() %>% nsdiffs()
```

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

```
usmelec %>% log() %>% diff(lag=12) %>% ndiffs()
```

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

## Your turn

For the visitors series, find an appropriate differencing (after transformation if necessary) to obtain stationary data.

## Backshift notation

A very useful notational device is the backward shift operator,  $B$ , which is used as follows:

$$By_t = y_{t-1} .$$

## Backshift notation

A very useful notational device is the backward shift operator,  $B$ , which is used as follows:

$$By_t = y_{t-1}.$$

In other words,  $B$ , operating on  $y_t$ , has the effect of **shifting the data back one period**.

## Backshift notation

A very useful notational device is the backward shift operator,  $B$ , which is used as follows:

$$By_t = y_{t-1}.$$

In other words,  $B$ , operating on  $y_t$ , has the effect of **shifting the data back one period**. Two applications of  $B$  to  $y_t$  **shifts the data back two periods**:

$$B(By_t) = B^2y_t = y_{t-2}.$$

## Backshift notation

A very useful notational device is the backward shift operator,  $B$ , which is used as follows:

$$By_t = y_{t-1}.$$

In other words,  $B$ , operating on  $y_t$ , has the effect of **shifting the data back one period**. Two applications of  $B$  to  $y_t$  **shifts the data back two periods**:

$$B(By_t) = B^2y_t = y_{t-2}.$$

For monthly data, if we wish to shift attention to “the same month last year,” then  $B^{12}$  is used, and the notation is  $B^{12}y_t = y_{t-12}$ .

## Backshift notation

The backward shift operator is convenient for describing the process of *differencing*.

## Backshift notation

The backward shift operator is convenient for describing the process of *differencing*. A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t .$$

## Backshift notation

The backward shift operator is convenient for describing the process of *differencing*. A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t .$$

Note that a first difference is represented by  $(1 - B)$ .

## Backshift notation

The backward shift operator is convenient for describing the process of *differencing*. A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - B y_t = (1 - B) y_t .$$

Note that a first difference is represented by  $(1 - B)$ . Similarly, if second-order differences (i.e., first differences of first differences) have to be computed, then:

$$y''_t = y_t - 2y_{t-1} + y_{t-2} = (1 - B)^2 y_t .$$

## Backshift notation

- Second-order difference is denoted  $(1 - B)^2$ .
- Second-order difference is not the same as a second difference, which would be denoted  $1 - B^2$ ;
- In general, a  $d$ th-order difference can be written as

$$(1 - B)^d y_t.$$

- A seasonal difference followed by a first difference can be written as

$$(1 - B)(1 - B^m)y_t .$$

## Backshift notation

The “backshift” notation is convenient because the terms can be multiplied together to see the combined effect.

$$\begin{aligned}(1 - B)(1 - B^m)y_t &= (1 - B - B^m + B^{m+1})y_t \\&= y_t - y_{t-1} - y_{t-m} + y_{t-m-1}.\end{aligned}$$

## Backshift notation

The “backshift” notation is convenient because the terms can be multiplied together to see the combined effect.

$$\begin{aligned}(1 - B)(1 - B^m)y_t &= (1 - B - B^m + B^{m+1})y_t \\&= y_t - y_{t-1} - y_{t-m} + y_{t-m-1}.\end{aligned}$$

For monthly data,  $m = 12$  and we obtain the same result as earlier.

# Outline

- 1 Stationarity and differencing**
- 2 Non-seasonal ARIMA models**
- 3 Estimation and order selection**
- 4 ARIMA modelling in R**
- 5 Forecasting**
- 6 Seasonal ARIMA models**
- 7 ARIMA vs ETS**

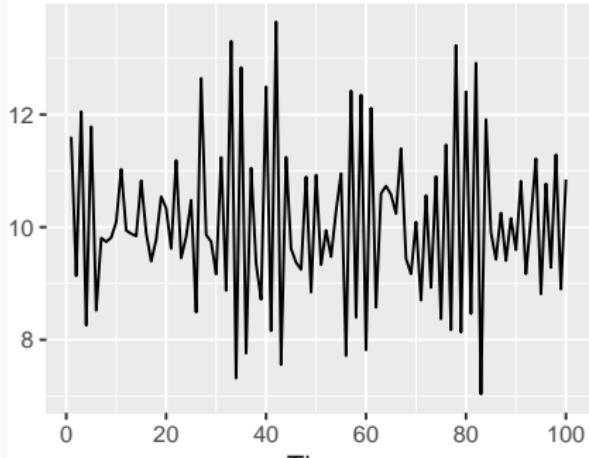
# Autoregressive models

## Autoregressive (AR) models:

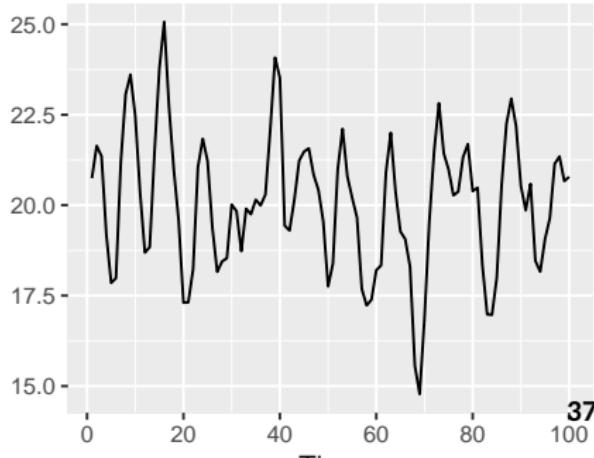
$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

where  $\varepsilon_t$  is white noise. This is a multiple regression with lagged values of  $y_t$  as predictors.

AR(1)



AR(2)

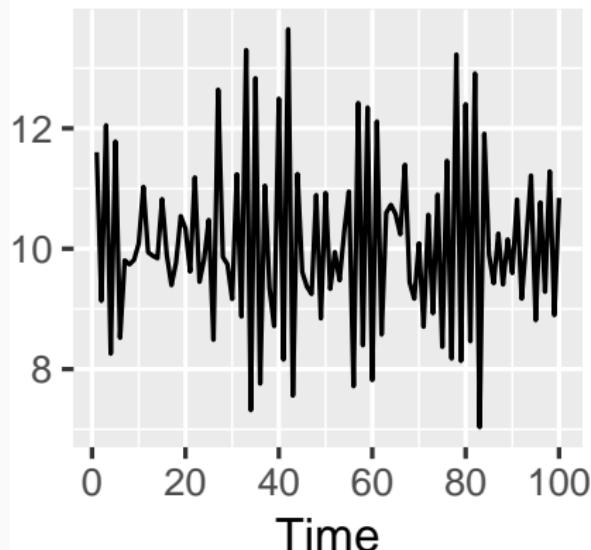


# AR(1) model

$$y_t = 2 - 0.8y_{t-1} + \varepsilon_t$$

$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$

AR(1)



# AR(1) model

$$y_t = c + \phi_1 y_{t-1} + \varepsilon_t$$

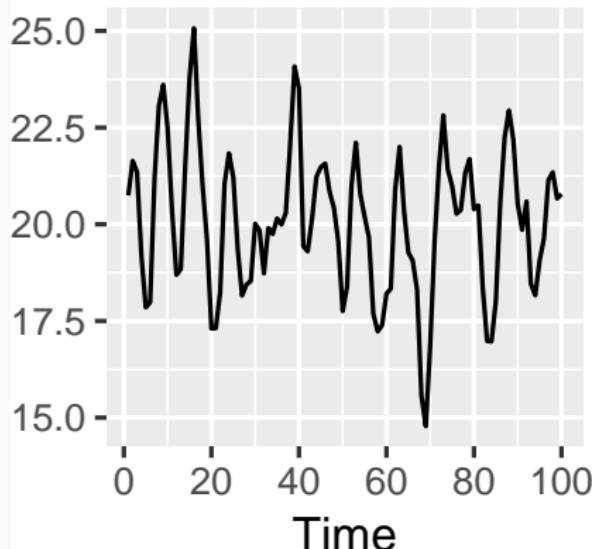
- When  $\phi_1 = 0$ ,  $y_t$  is **equivalent to WN**
- When  $\phi_1 = 1$  and  $c = 0$ ,  $y_t$  is **equivalent to a RW**
- When  $\phi_1 = 1$  and  $c \neq 0$ ,  $y_t$  is **equivalent to a RW with drift**
- When  $\phi_1 < 0$ ,  $y_t$  tends to **oscillate between positive and negative values.**

## AR(2) model

$$y_t = 8 + 1.3y_{t-1} - 0.7y_{t-2} + \varepsilon_t$$

$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$

AR(2)



# Stationarity conditions

We normally restrict autoregressive models to stationary data, and then some constraints on the values of the parameters are required.

## General condition for stationarity

Complex roots of  $1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p$  lie outside the unit circle on the complex plane.

# Stationarity conditions

We normally restrict autoregressive models to stationary data, and then some constraints on the values of the parameters are required.

## General condition for stationarity

Complex roots of  $1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p$  lie outside the unit circle on the complex plane.

- For  $p = 1$ :  $-1 < \phi_1 < 1$ .
- For  $p = 2$ :
$$-1 < \phi_2 < 1 \quad \phi_2 + \phi_1 < 1 \quad \phi_2 - \phi_1 < 1.$$
- More complicated conditions hold for  $p \geq 3$ .
- Estimation software takes care of this.

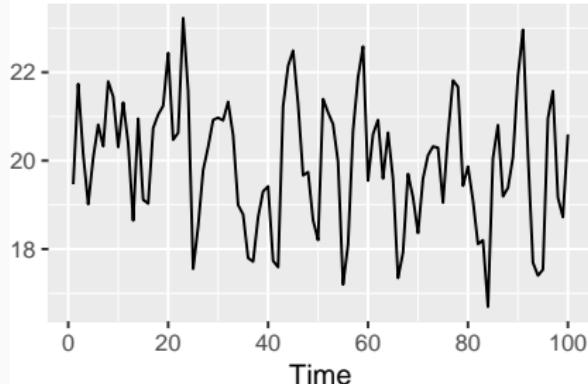
# Moving Average (MA) models

## Moving Average (MA) models:

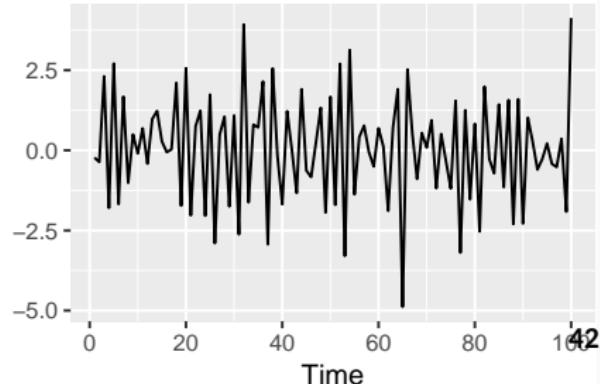
$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q},$$

where  $\varepsilon_t$  is white noise. This is a multiple regression with **past errors** as predictors. *Don't confuse this with moving average smoothing!*

MA(1)



MA(2)

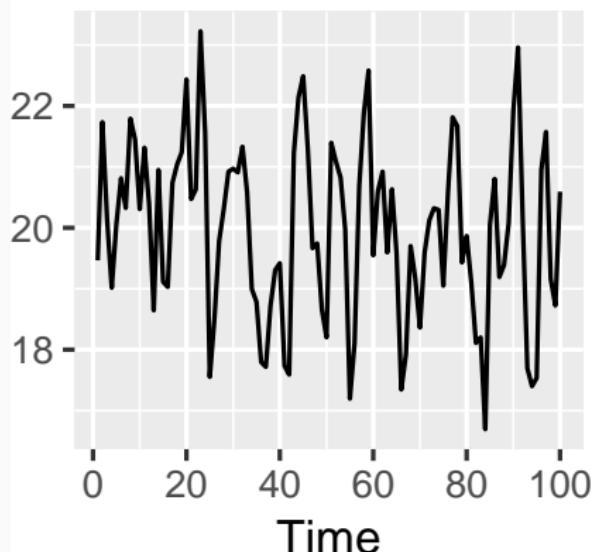


# MA(1) model

$$y_t = 20 + \varepsilon_t + 0.8\varepsilon_{t-1}$$

$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$

MA(1)

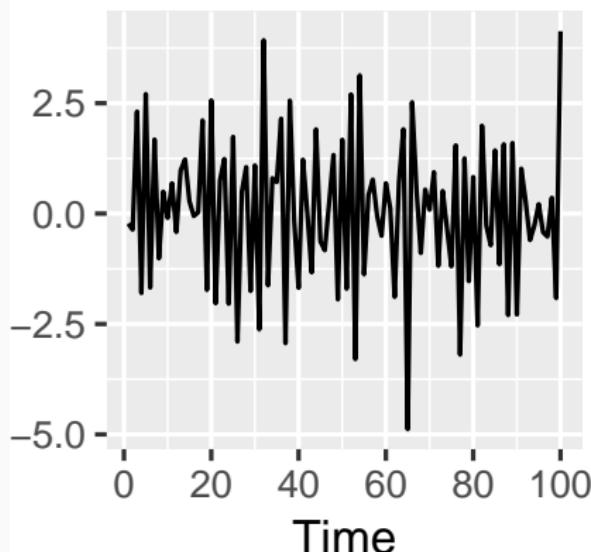


## MA(2) model

$$y_t = \varepsilon_t - \varepsilon_{t-1} + 0.8\varepsilon_{t-2}$$

$$\varepsilon_t \sim N(0, 1), \quad T = 100.$$

MA(2)



# MA( $\infty$ ) models

It is possible to write any stationary AR( $p$ ) process as an MA( $\infty$ ) process.

## Example: AR(1)

$$\begin{aligned}y_t &= \phi_1 y_{t-1} + \varepsilon_t \\&= \phi_1(\phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\&= \phi_1^2 y_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\&= \phi_1^3 y_{t-3} + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\&\dots\end{aligned}$$

## MA( $\infty$ ) models

It is possible to write any stationary AR( $p$ ) process as an MA( $\infty$ ) process.

### Example: AR(1)

$$\begin{aligned}y_t &= \phi_1 y_{t-1} + \varepsilon_t \\&= \phi_1(\phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\&= \phi_1^2 y_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\&= \phi_1^3 y_{t-3} + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\&\dots\end{aligned}$$

Provided  $-1 < \phi_1 < 1$ :

$$y_t = \varepsilon_t + \phi_1 \varepsilon_{t-1} + \phi_1^2 \varepsilon_{t-2} + \phi_1^3 \varepsilon_{t-3} + \dots$$

# Invertibility

- Any MA( $q$ ) process can be written as an AR( $\infty$ ) process if we impose some constraints on the MA parameters.
- Then the MA model is called “invertible”.
- Invertible models have some mathematical properties that make them easier to use in practice.
- Invertibility of an ARIMA model is equivalent to forecastability of an ETS model.

# Invertibility

## General condition for invertibility

Complex roots of  $1 + \theta_1 z + \theta_2 z^2 + \dots + \theta_q z^q$  lie outside the unit circle on the complex plane.

# Invertibility

## General condition for invertibility

Complex roots of  $1 + \theta_1 z + \theta_2 z^2 + \dots + \theta_q z^q$  lie outside the unit circle on the complex plane.

- For  $q = 1$ :  $-1 < \theta_1 < 1$ .
- For  $q = 2$ :
$$-1 < \theta_2 < 1 \quad \theta_2 + \theta_1 > -1 \quad \theta_1 - \theta_2 < 1.$$
- More complicated conditions hold for  $q \geq 3$ .
- Estimation software takes care of this.

# ARIMA models

## Autoregressive Moving Average models:

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} \\ + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t.$$

# ARIMA models

## Autoregressive Moving Average models:

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} \\ + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t.$$

- Predictors include both **lagged values of  $y_t$  and lagged errors.**
- Conditions on coefficients ensure stationarity.
- Conditions on coefficients ensure invertibility.

# ARIMA models

## Autoregressive Moving Average models:

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} \\ + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t.$$

- Predictors include both **lagged values of  $y_t$  and lagged errors.**
- Conditions on coefficients ensure stationarity.
- Conditions on coefficients ensure invertibility.

## Autoregressive Integrated Moving Average models

- Combine ARMA model with **differencing**.
- $(1 - B)^d y_t$  follows an ARMA model.

# ARIMA models

## Autoregressive Integrated Moving Average models

### ARIMA( $p, d, q$ ) model

AR:  $p$  = order of the autoregressive part

I:  $d$  = degree of first differencing involved

MA:  $q$  = order of the moving average part.

- White noise model: ARIMA(0,0,0)
- Random walk: ARIMA(0,1,0) with no constant
- Random walk with drift: ARIMA(0,1,0) with const.
- AR( $p$ ): ARIMA( $p, 0, 0$ )
- MA( $q$ ): ARIMA( $0, 0, q$ )

# Backshift notation for ARIMA

## ■ ARMA model:

$$y_t = c + \phi_1 B y_t + \cdots + \phi_p B^p y_t + \varepsilon_t + \theta_1 B \varepsilon_t + \cdots + \theta_q B^q \varepsilon_t$$

or  $(1 - \phi_1 B - \cdots - \phi_p B^p) y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q) \varepsilon_t$

## ■ ARIMA(1,1,1) model:

$$(1 - \phi_1 B) (1 - B) y_t = c + (1 + \theta_1 B) \varepsilon_t$$

↑                    ↑                    ↑  
AR(1)              First                MA(1)  
difference

# Backshift notation for ARIMA

## ■ ARMA model:

$$y_t = c + \phi_1 B y_t + \cdots + \phi_p B^p y_t + \varepsilon_t + \theta_1 B \varepsilon_t + \cdots + \theta_q B^q \varepsilon_t$$

$$\text{or } (1 - \phi_1 B - \cdots - \phi_p B^p) y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q) \varepsilon_t$$

## ■ ARIMA(1,1,1) model:

$$(1 - \phi_1 B) (1 - B) y_t = c + (1 + \theta_1 B) \varepsilon_t$$

↑                      ↑                      ↑  
AR(1)              First              MA(1)  
difference

Written out:

$$y_t = c + y_{t-1} + \phi_1 y_{t-1} - \phi_1 y_{t-2} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

# R model

## Intercept form

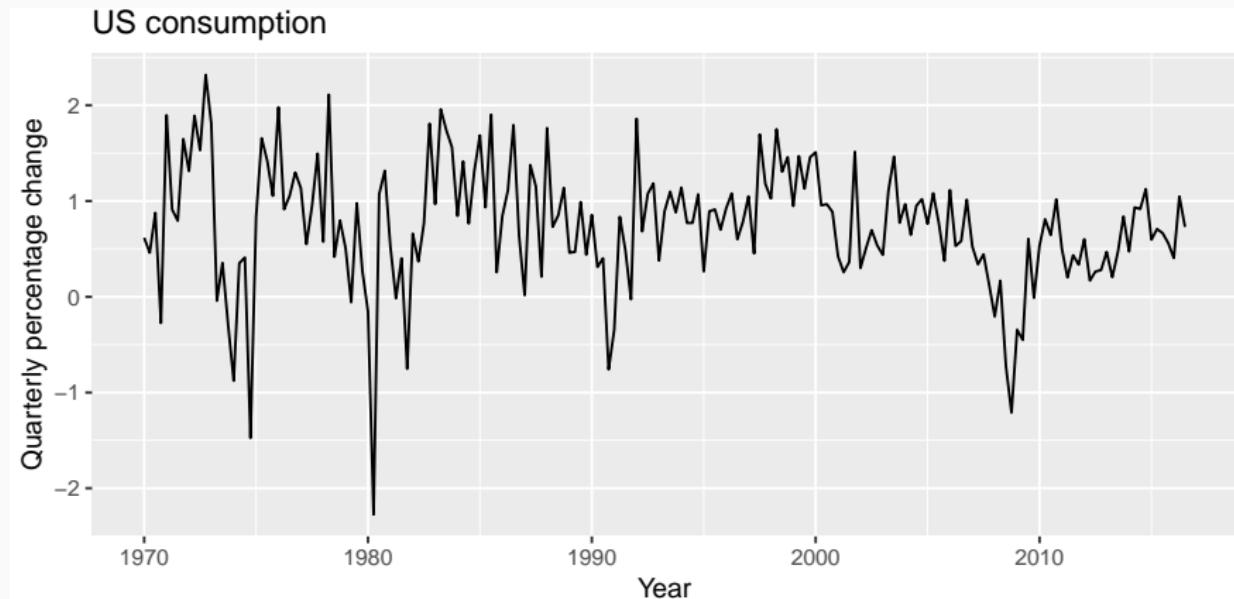
$$(1 - \phi_1 B - \cdots - \phi_p B^p) y'_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q) \varepsilon_t$$

## Mean form

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(y'_t - \mu) = (1 + \theta_1 B + \cdots + \theta_q B^q) \varepsilon_t$$

- $y'_t = (1 - B)^d y_t$
- $\mu$  is the mean of  $y'_t$ .
- $c = \mu(1 - \phi_1 - \cdots - \phi_p)$ .

# US personal consumption



# US personal consumption

```
(fit <- auto.arima(uschange[, "Consumption"]))

## Series: uschange[, "Consumption"]
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##             ar1      ar2      ma1      ma2      mean
##             1.3908 -0.5813 -1.1800  0.5584  0.7463
## s.e.    0.2553  0.2078  0.2381  0.1403  0.0845
##
## sigma^2 estimated as 0.3511: log likelihood=-165.14
## AIC=342.28   AICc=342.75   BIC=361.67
```

# US personal consumption

```
(fit <- auto.arima(uschange[, "Consumption"]))

## Series: uschange[, "Consumption"]
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##             ar1      ar2      ma1      ma2      mean
##             1.3908 -0.5813 -1.1800  0.5584  0.7463
## s.e.    0.2553  0.2078  0.2381  0.1403  0.0845
##
## sigma^2 estimated as 0.3511:  log likelihood=-165.14
## AIC=342.28    AICc=342.75    BIC=361.67
##
```

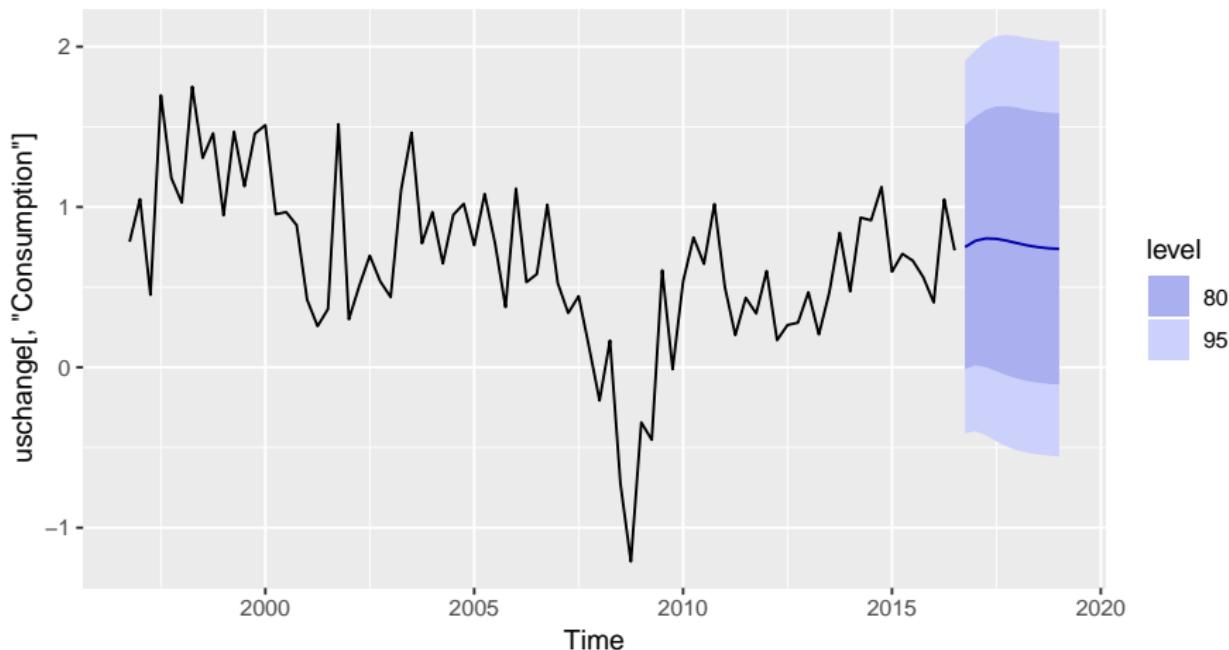
## ARIMA(2,0,2) model:

$y_t = c + 1.391y_{t-1} - 0.581y_{t-2} - 1.180\varepsilon_{t-1} + 0.558\varepsilon_{t-2} + \varepsilon_t$ ,  
where  $c = 0.746 \times (1 - 1.391 + 0.581) = 0.142$  and  $\varepsilon_t$  is white noise with a standard deviation of  $0.593 = \sqrt{0.351}$ .

# US personal consumption

```
fit %>% forecast(h=10) %>% autoplot(include=80)
```

Forecasts from ARIMA(2,0,2) with non-zero mean



# Understanding ARIMA models

- If  $c = 0$  and  $d = 0$ , the long-term forecasts will go to zero.
- If  $c = 0$  and  $d = 1$ , the long-term forecasts will go to a non-zero constant.
- If  $c = 0$  and  $d = 2$ , the long-term forecasts will follow a straight line.
- If  $c \neq 0$  and  $d = 0$ , the long-term forecasts will go to the mean of the data.
- If  $c \neq 0$  and  $d = 1$ , the long-term forecasts will follow a straight line.
- If  $c \neq 0$  and  $d = 2$ , the long-term forecasts will follow a quadratic trend.

# Understanding ARIMA models

## Forecast variance and $d$

- The higher the value of  $d$ , the more rapidly the prediction intervals increase in size.
- For  $d = 0$ , the long-term forecast standard deviation will go to the standard deviation of the historical data.

## Cyclic behaviour

- For cyclic forecasts,  $p \geq 2$  and some restrictions on coefficients are required.
- If  $p = 2$ , we need  $\phi_1^2 + 4\phi_2 < 0$ . Then average cycle of length

$$(2\pi) / [\text{arc cos}(-\phi_1(1 - \phi_2)/(4\phi_2))].$$

# Outline

- 1 Stationarity and differencing**
- 2 Non-seasonal ARIMA models**
- 3 Estimation and order selection**
- 4 ARIMA modelling in R**
- 5 Forecasting**
- 6 Seasonal ARIMA models**
- 7 ARIMA vs ETS**

## Maximum likelihood estimation

Having identified the model order, we need to estimate the parameters  $c, \phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q$ .

# Maximum likelihood estimation

Having identified the model order, we need to estimate the parameters  $c, \phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q$ .

- MLE is very similar to least squares estimation obtained by minimizing

$$\sum_{t=1}^T e_t^2.$$

- The Arima() command allows CLS or MLE estimation.
- Non-linear optimization must be used in either case.
- Different software will give different estimates.

# Partial autocorrelations

Partial autocorrelations measure relationship between  $y_t$  and  $y_{t-k}$ , when the effects of other time lags — 1, 2, 3, . . . ,  $k - 1$  — are removed.

# Partial autocorrelations

Partial autocorrelations measure relationship between  $y_t$  and  $y_{t-k}$ , when the effects of other time lags — 1, 2, 3, . . . ,  $k - 1$  — are removed.

$\alpha_k$  =  $k$ th partial autocorrelation coefficient  
= equal to the estimate of  $b_k$  in regression:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_k y_{t-k}.$$

# Partial autocorrelations

Partial autocorrelations measure relationship between  $y_t$  and  $y_{t-k}$ , when the effects of other time lags — 1, 2, 3, . . . ,  $k - 1$  — are removed.

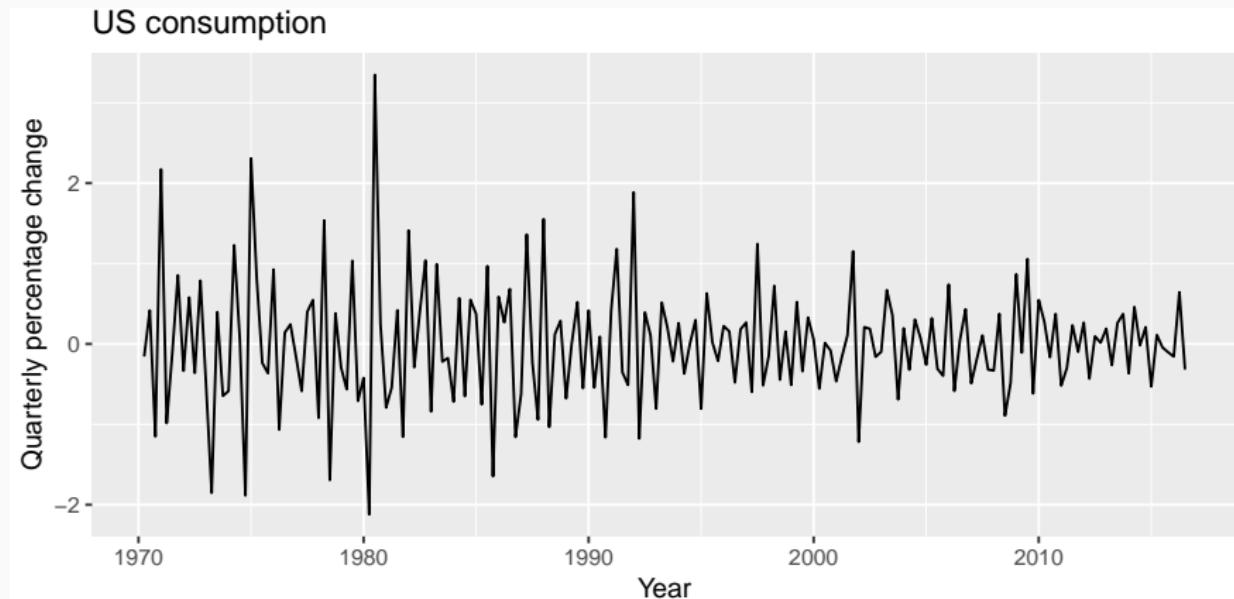
$\alpha_k$  =  $k$ th partial autocorrelation coefficient

= equal to the estimate of  $b_k$  in regression:

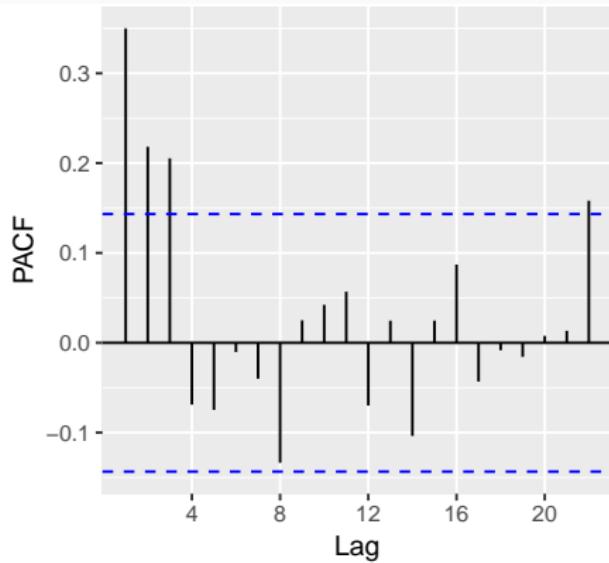
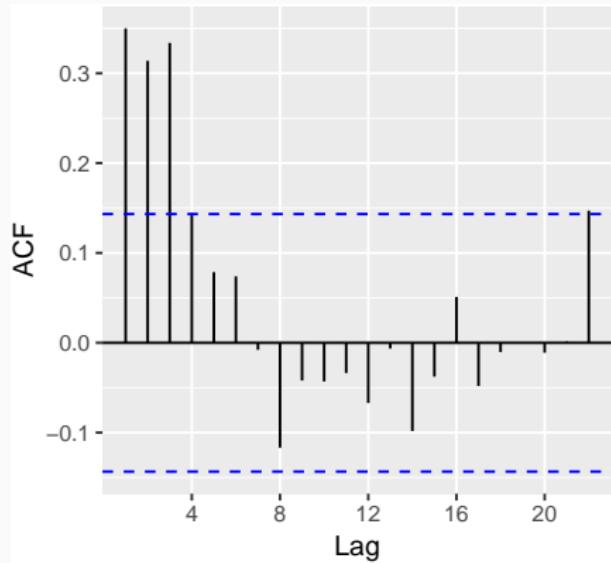
$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_k y_{t-k}.$$

- Varying number of terms on RHS gives  $\alpha_k$  for different values of  $k$ .
- There are more efficient ways of calculating  $\alpha_k$ .
- $\alpha_1 = \rho_1$

# Example: US consumption



# Example: US consumption



# ACF and PACF interpretation

## AR(1)

$$\rho_k = \phi_1^k \quad \text{for } k = 1, 2, \dots;$$

$$\alpha_1 = \phi_1 \quad \alpha_k = 0 \quad \text{for } k = 2, 3, \dots.$$

So we have an AR(1) model when

- autocorrelations exponentially decay
- there is a single significant partial autocorrelation.

# ACF and PACF interpretation

## AR( $p$ )

- ACF dies out in an exponential or damped sine-wave manner
- PACF has all zero spikes beyond the  $p$ th spike

So we have an AR( $p$ ) model when

- the ACF is exponentially decaying or sinusoidal
- there is a significant spike at lag  $p$  in PACF, but none beyond  $p$

# ACF and PACF interpretation

## MA(1)

$$\begin{aligned}\rho_1 &= \theta_1 & \rho_k &= 0 & \text{for } k = 2, 3, \dots; \\ \alpha_k &= -(-\theta_1)^k\end{aligned}$$

So we have an MA(1) model when

- the PACF is exponentially decaying and
- there is a single significant spike in ACF

# ACF and PACF interpretation

## MA( $q$ )

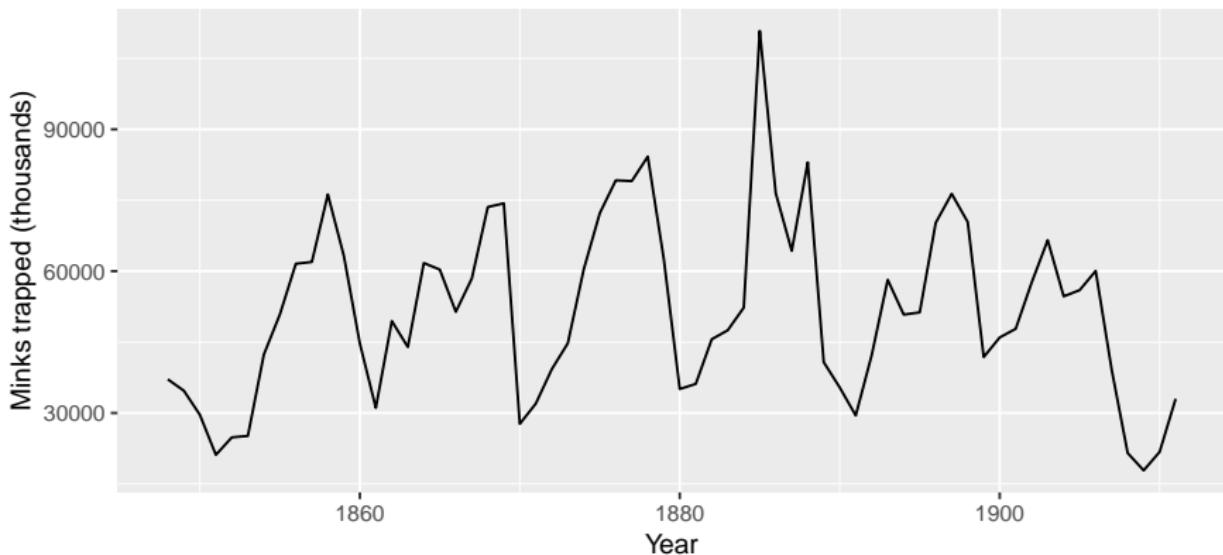
- PACF dies out in an exponential or damped sine-wave manner
- ACF has all zero spikes beyond the  $q$ th spike

So we have an MA( $q$ ) model when

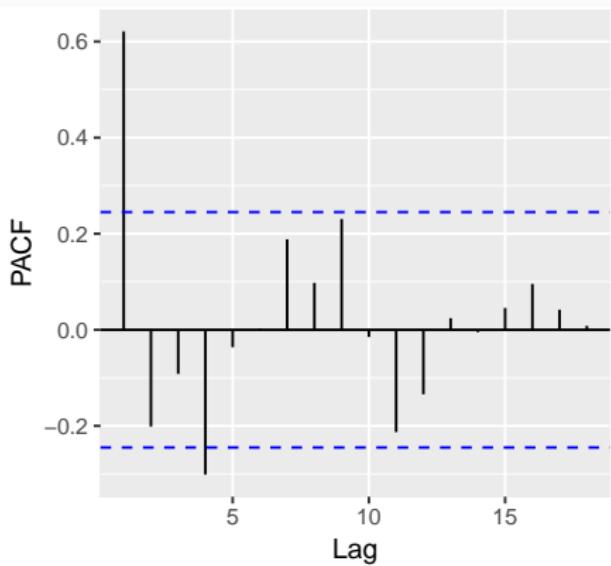
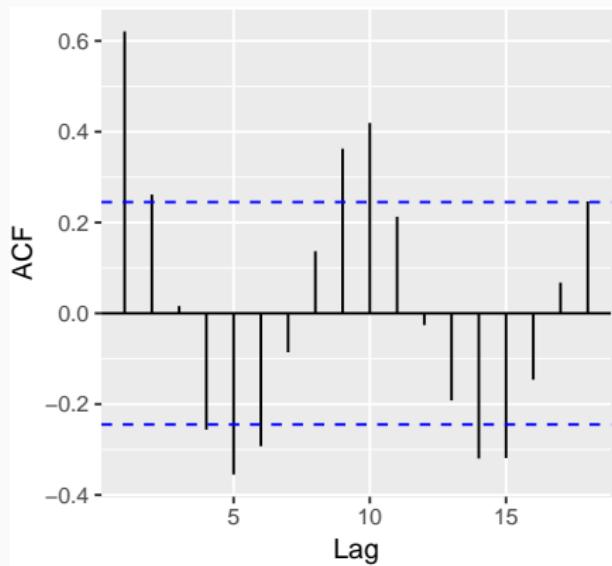
- the PACF is exponentially decaying or sinusoidal
- there is a significant spike at lag  $q$  in ACF, but none beyond  $q$

# Example: Mink trapping

Annual number of minks trapped



# Example: Mink trapping



# Information criteria

## Akaike's Information Criterion (AIC):

$$AIC = -2 \log(L) + 2(p + q + k + 1),$$

where  $L$  is the likelihood of the data,

$k = 1$  if  $c \neq 0$  and  $k = 0$  if  $c = 0$ .

# Information criteria

## Akaike's Information Criterion (AIC):

$$\text{AIC} = -2 \log(L) + 2(p + q + k + 1),$$

where  $L$  is the likelihood of the data,

$k = 1$  if  $c \neq 0$  and  $k = 0$  if  $c = 0$ .

## Corrected AIC:

$$\text{AICc} = \text{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}.$$

# Information criteria

## Akaike's Information Criterion (AIC):

$$\text{AIC} = -2 \log(L) + 2(p + q + k + 1),$$

where  $L$  is the likelihood of the data,

$k = 1$  if  $c \neq 0$  and  $k = 0$  if  $c = 0$ .

## Corrected AIC:

$$\text{AICc} = \text{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}.$$

## Bayesian Information Criterion:

$$\text{BIC} = \text{AIC} + [\log(T) - 2](p + q + k - 1).$$

# Information criteria

## Akaike's Information Criterion (AIC):

$$\text{AIC} = -2 \log(L) + 2(p + q + k + 1),$$

where  $L$  is the likelihood of the data,

$k = 1$  if  $c \neq 0$  and  $k = 0$  if  $c = 0$ .

## Corrected AIC:

$$\text{AICc} = \text{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}.$$

## Bayesian Information Criterion:

$$\text{BIC} = \text{AIC} + [\log(T) - 2](p + q + k - 1).$$

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

# Outline

- 1 Stationarity and differencing**
- 2 Non-seasonal ARIMA models**
- 3 Estimation and order selection**
- 4 ARIMA modelling in R**
- 5 Forecasting**
- 6 Seasonal ARIMA models**
- 7 ARIMA vs ETS**

# How does auto.arima() work?

## A non-seasonal ARIMA process

$$\phi(B)(1 - B)^d y_t = c + \theta(B)\varepsilon_t$$

Need to select appropriate orders:  $p, q, d$

## Hyndman and Khandakar (JSS, 2008) algorithm:

- Select no. differences  $d$  and  $D$  via KPSS test and seasonal strength measure.
- Select  $p, q$  by minimising AICc.
- Use stepwise search to traverse model space.

# How does auto.arima() work?

$$\text{AICc} = -2 \log(L) + 2(p + q + k + 1) \left[ 1 + \frac{(p+q+k+2)}{T-p-q-k-2} \right].$$

where  $L$  is the maximised likelihood fitted to the *differenced* data,  
 $k = 1$  if  $c \neq 0$  and  $k = 0$  otherwise.

# How does auto.arima() work?

$$\text{AICc} = -2 \log(L) + 2(p + q + k + 1) \left[ 1 + \frac{(p+q+k+2)}{T-p-q-k-2} \right].$$

where  $L$  is the maximised likelihood fitted to the *differenced* data,  
 $k = 1$  if  $c \neq 0$  and  $k = 0$  otherwise.

**Step1:** Select current model (with smallest AICc) from:

ARIMA(2, d, 2)

ARIMA(0, d, 0)

ARIMA(1, d, 0)

ARIMA(0, d, 1)

# How does auto.arima() work?

$$\text{AICc} = -2 \log(L) + 2(p + q + k + 1) \left[ 1 + \frac{(p+q+k+2)}{T-p-q-k-2} \right].$$

where  $L$  is the maximised likelihood fitted to the *differenced* data,  
 $k = 1$  if  $c \neq 0$  and  $k = 0$  otherwise.

**Step 1:** Select current model (with smallest AICc) from:

ARIMA(2, d, 2)

ARIMA(0, d, 0)

ARIMA(1, d, 0)

ARIMA(0, d, 1)

**Step 2:** Consider variations of current model:

- vary one of  $p, q$ , from current model by  $\pm 1$ ;
- $p, q$  both vary from current model by  $\pm 1$ ;
- Include/exclude  $c$  from current model.

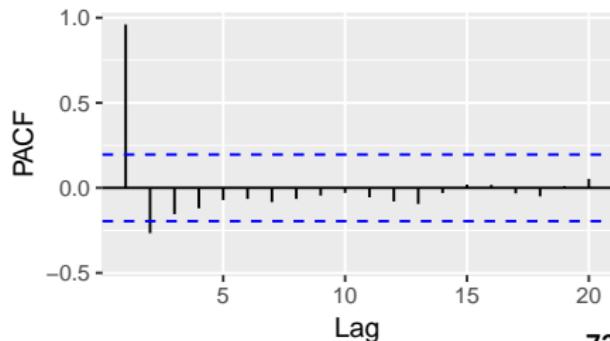
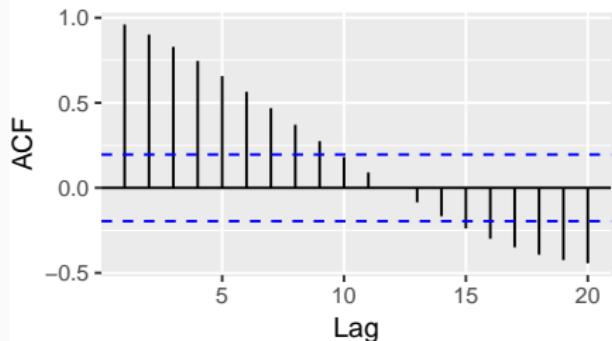
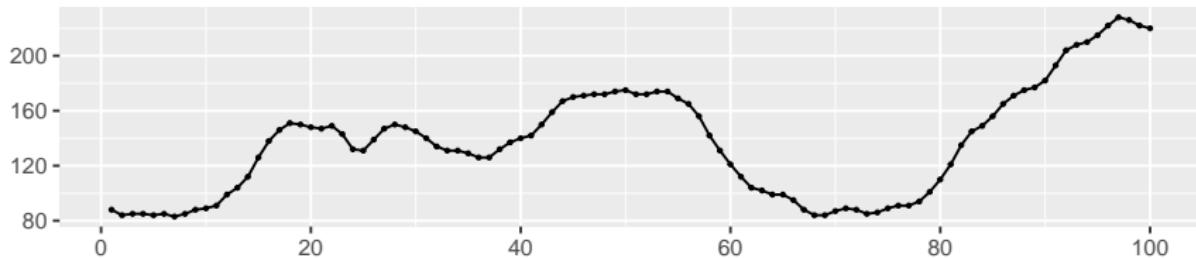
Model with lowest AICc becomes current model.

**Repeat Step 2 until no lower AICc can be found.**

# Choosing your own model

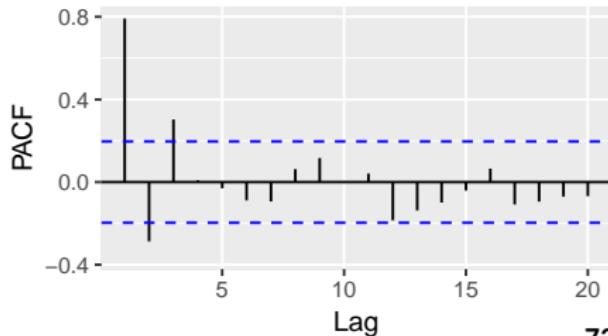
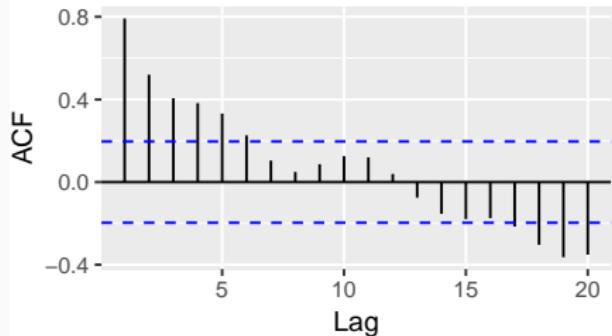
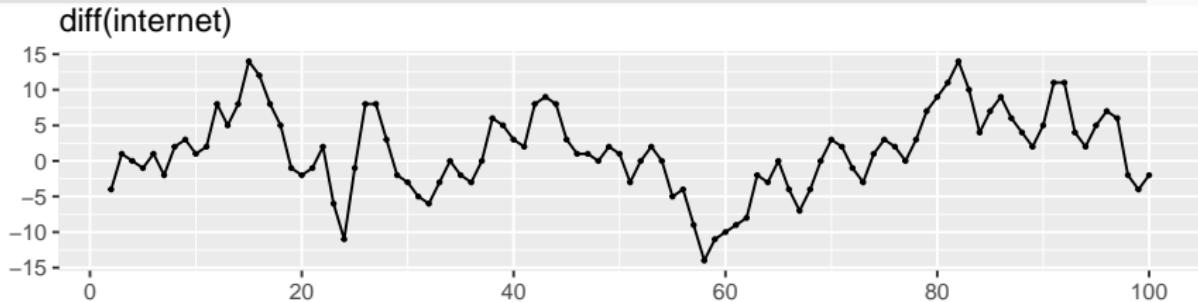
ggtsdisplay(internet)

internet



# Choosing your own model

```
ggttsdisplay(diff(internet))
```



# Choosing your own model

```
(fit <- Arima(internet,order=c(3,1,0)))  
  
## Series: internet  
## ARIMA(3,1,0)  
##  
## Coefficients:  
##           ar1       ar2       ar3  
##      1.1513 -0.6612  0.3407  
## s.e.  0.0950  0.1353  0.0941  
##  
## sigma^2 estimated as 9.656:  log likelihood=-  
252  
## AIC=511.99    AICC=512.42    BIC=522.37
```

# Choosing your own model

```
auto.arima(internet)
```

```
## Series: internet
## ARIMA(1,1,1)
##
## Coefficients:
##             ar1      ma1
##             0.6504   0.5256
## s.e.    0.0842   0.0896
##
## sigma^2 estimated as 9.995:  log likelihood=-254.15
## AIC=514.3    AICc=514.55    BIC=522.08
```

# Choosing your own model

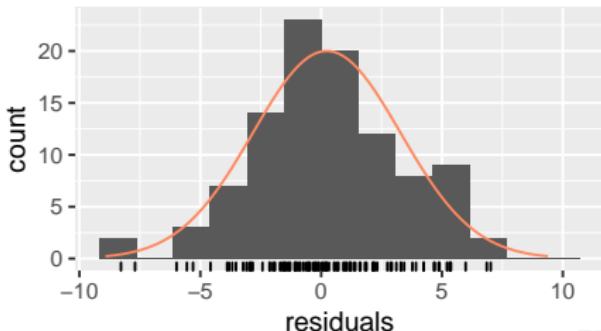
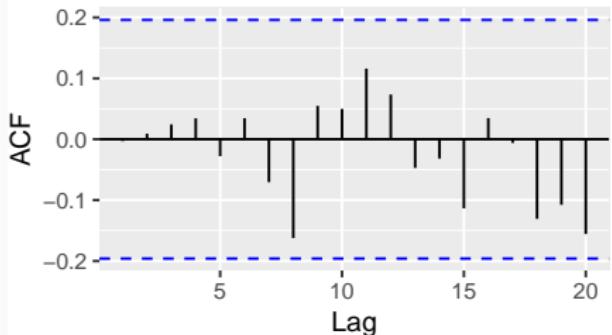
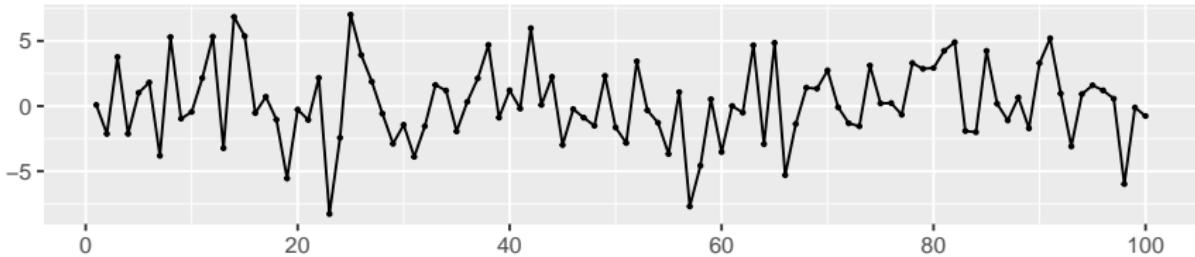
```
auto.arima(internet, stepwise=FALSE,  
approximation=FALSE)
```

```
## Series: internet  
## ARIMA(3,1,0)  
##  
## Coefficients:  
##          ar1      ar2      ar3  
##          1.1513 -0.6612  0.3407  
## s.e.  0.0950  0.1353  0.0941  
##  
## sigma^2 estimated as 9.656:  log likelihood=-  
252  
## AIC=511.99    AICC=512.42    BIC=522.37
```

# Choosing your own model

## checkresiduals(fit)

Residuals from ARIMA(3,1,0)



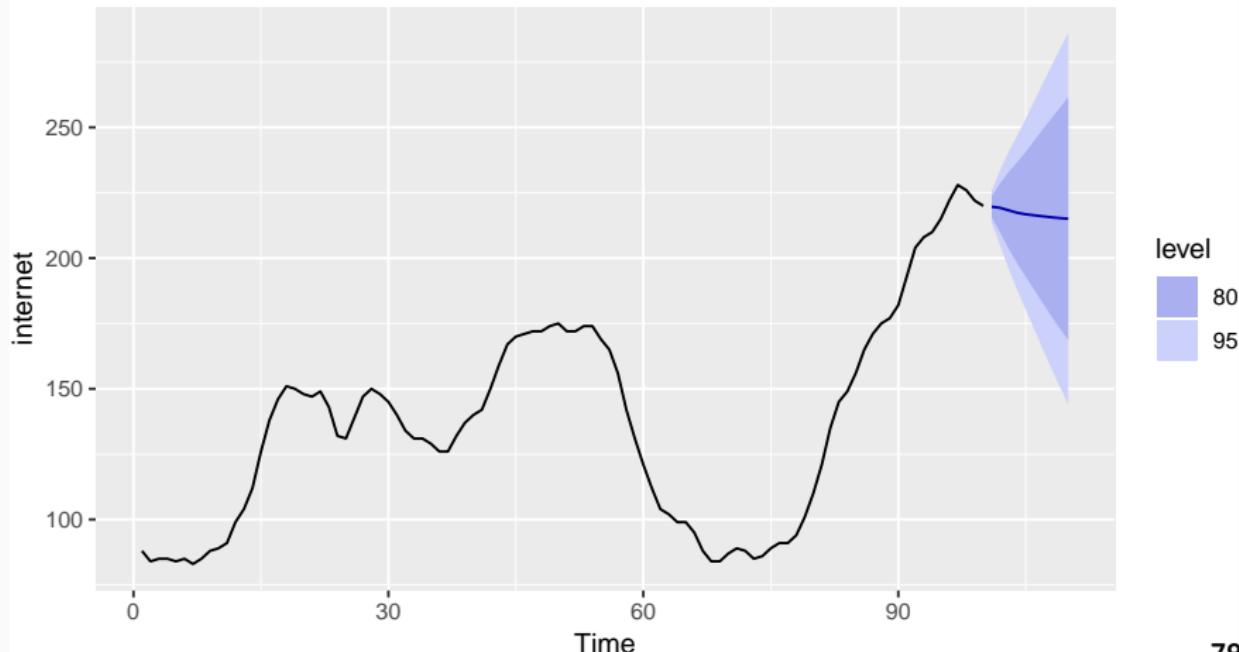
# Choosing your own model

```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(3,1,0)  
## Q* = 4.4913, df = 7, p-value = 0.7218  
##  
## Model df: 3.    Total lags used: 10
```

# Choosing your own model

```
fit %>% forecast %>% autoplot
```

Forecasts from ARIMA(3,1,0)



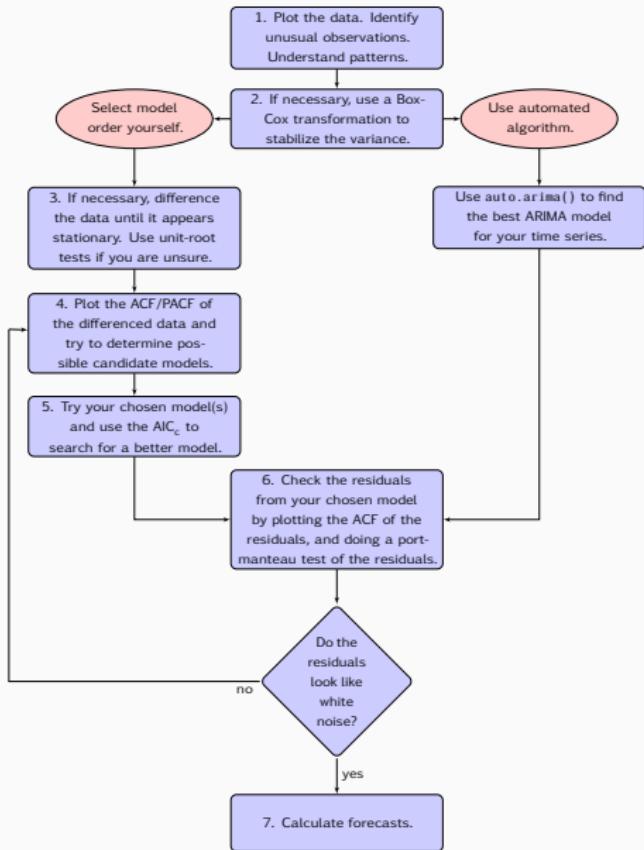
# Modelling procedure with Arima

- 1 Plot the data. Identify any unusual observations.
- 2 If necessary, transform the data (using a Box-Cox transformation) to stabilize the variance.
- 3 If the data are non-stationary: take first differences of the data until the data are stationary.
- 4 Examine the ACF/PACF: Is an AR( $p$ ) or MA( $q$ ) model appropriate?
- 5 Try your chosen model(s), and use the AICc to search for a better model.
- 6 Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
- 7 Once the residuals look like white noise, calculate forecasts.

# Modelling procedure with `auto.arima`

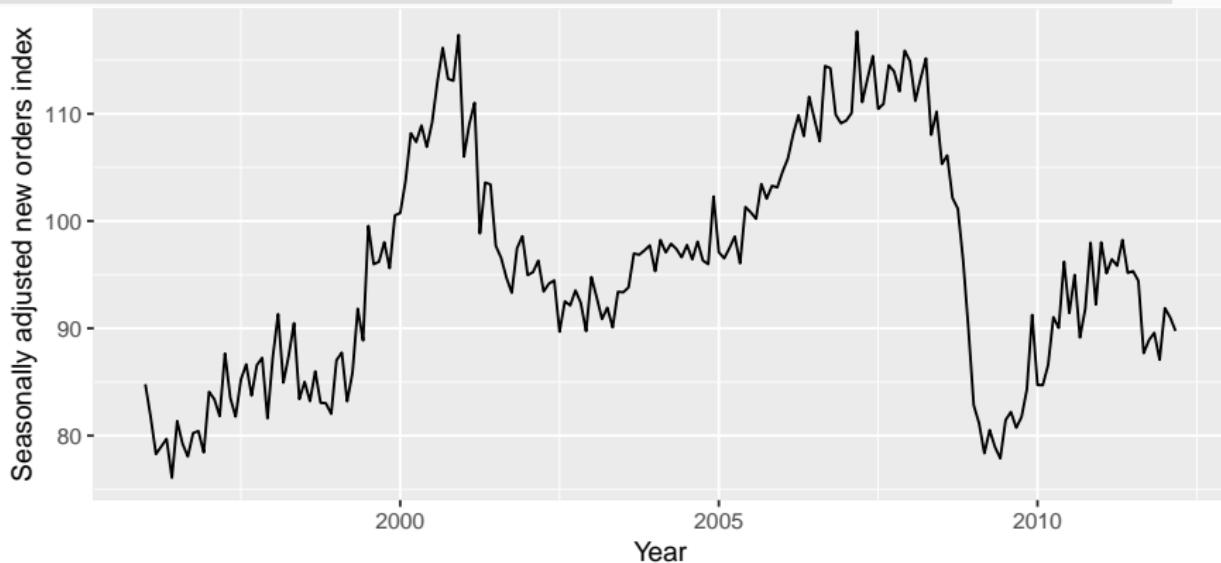
- 1 Plot the data. Identify any unusual observations.
- 2 If necessary, transform the data (using a Box-Cox transformation) to stabilize the variance.
- 3 Use `auto.arima` to select a model.
- 6 Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
- 7 Once the residuals look like white noise, calculate forecasts.

# Modelling procedure



# Seasonally adjusted electrical equipment

```
eeadj <- seasadj(stl(elecequip, s.window="periodic"))
autoplot(eeadj) + xlab("Year") +
ylab("Seasonally adjusted new orders index")
```



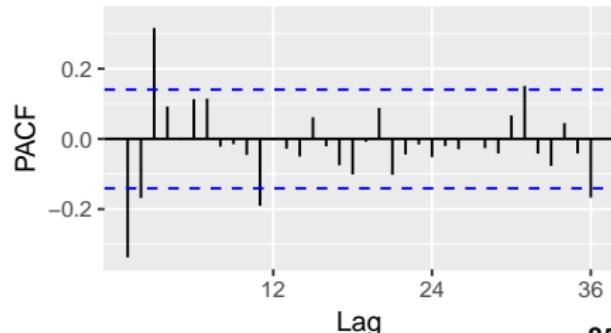
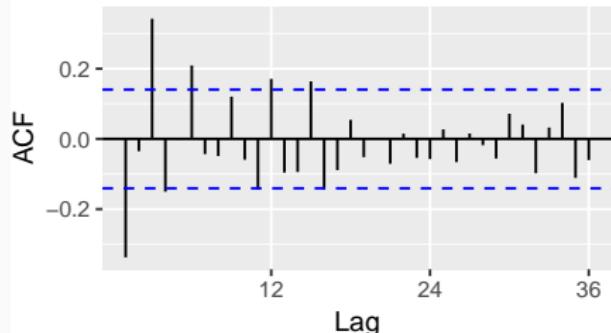
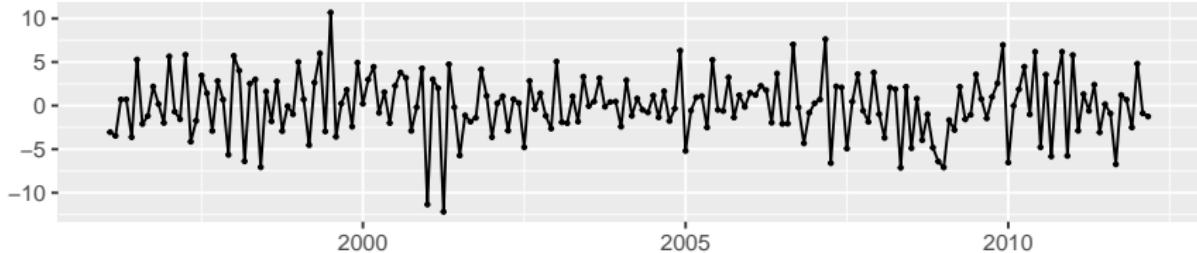
# Seasonally adjusted electrical equipment

- 1 Time plot shows sudden changes, particularly big drop in 2008/2009 due to global economic environment. Otherwise nothing unusual and no need for data adjustments.
- 2 No evidence of changing variance, so no Box-Cox transformation.
- 3 Data are clearly non-stationary, so we take first differences.

# Seasonally adjusted electrical equipment

```
ggttsdisplay(diff(eeadj))
```

diff(eeadj)



# Seasonally adjusted electrical equipment

- 4 PACF is suggestive of AR(3). So initial candidate model is ARIMA(3,1,0). No other obvious candidates.
- 5 Fit ARIMA(3,1,0) model along with variations: ARIMA(4,1,0), ARIMA(2,1,0), ARIMA(3,1,1), etc. ARIMA(3,1,1) has smallest AICc value.

# Seasonally adjusted electrical equipment

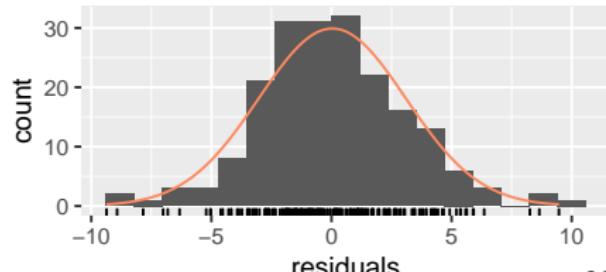
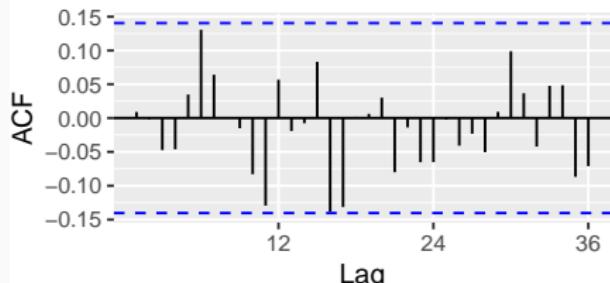
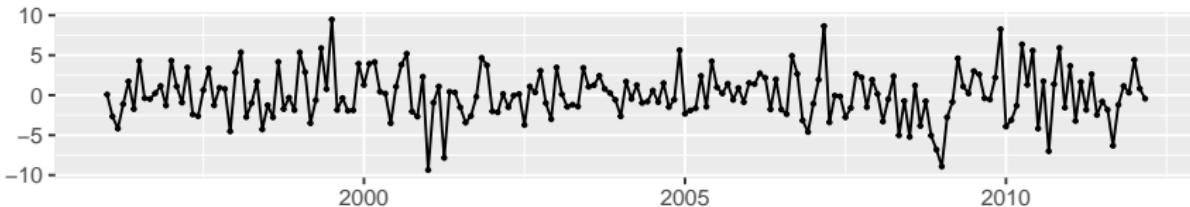
```
(fit <- Arima(eeadj, order=c(3,1,1)))  
  
## Series: eeadj  
## ARIMA(3,1,1)  
##  
## Coefficients:  
##          ar1      ar2      ar3      ma1  
##          0.0044  0.0916  0.3698 -0.3921  
## s.e.  0.2201  0.0984  0.0669  0.2426  
##  
## sigma^2 estimated as 9.577: log likelihood=-492.69  
## AIC=995.38    AICc=995.7    BIC=1011.72
```

# Seasonally adjusted electrical equipment

- 6 ACF plot of residuals from ARIMA(3,1,1) model look like white noise.

```
checkresiduals(fit)
```

Residuals from ARIMA(3,1,1)



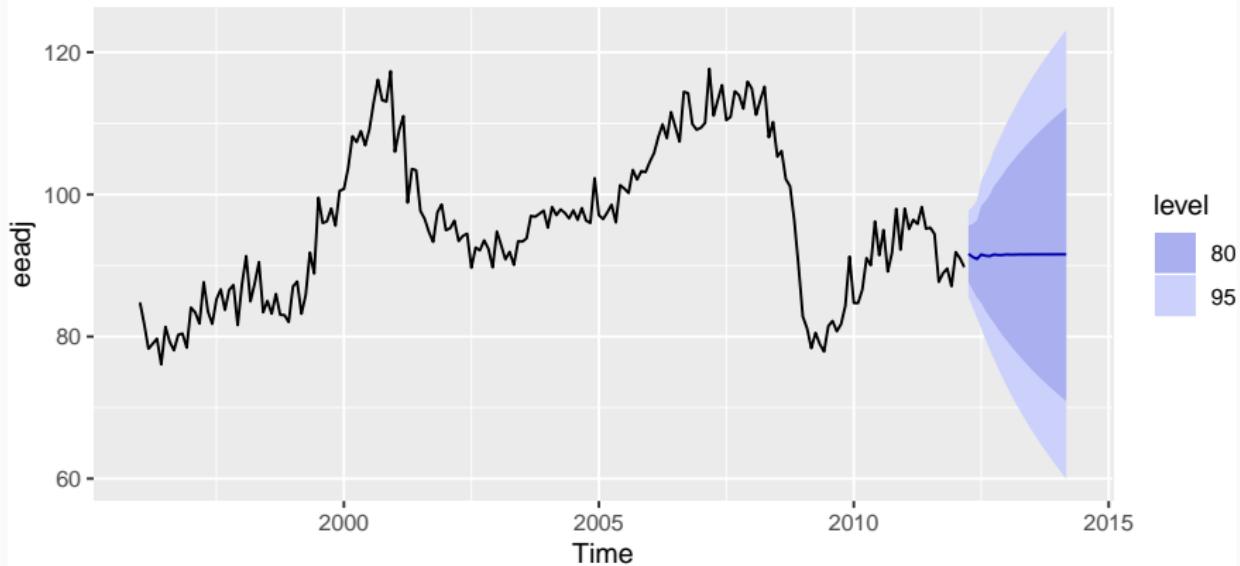
# Seasonally adjusted electrical equipment

```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(3,1,1)  
## Q* = 24.034, df = 20, p-value = 0.2409  
##  
## Model df: 4. Total lags used: 24
```

# Seasonally adjusted electrical equipment

```
fit %>% forecast %>% autoplot
```

Forecasts from ARIMA(3,1,1)



# Outline

- 1 Stationarity and differencing
- 2 Non-seasonal ARIMA models
- 3 Estimation and order selection
- 4 ARIMA modelling in R
- 5 Forecasting
- 6 Seasonal ARIMA models
- 7 ARIMA vs ETS

## Point forecasts

- 1 Rearrange ARIMA equation so  $y_t$  is on LHS.
- 2 Rewrite equation by replacing  $t$  by  $T + h$ .
- 3 On RHS, replace future observations by their forecasts, future errors by zero, and past errors by corresponding residuals.

Start with  $h = 1$ . Repeat for  $h = 2, 3, \dots$

# Point forecasts

## ARIMA(3,1,1) forecasts: Step 1

$$(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3)(1 - B)y_t = (1 + \theta_1 B)\varepsilon_t,$$

# Point forecasts

## ARIMA(3,1,1) forecasts: Step 1

$$(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3)(1 - B)y_t = (1 + \theta_1 B)\varepsilon_t,$$

$$\begin{aligned} [1 - (1 + \phi_1)B + (\phi_1 - \phi_2)B^2 + (\phi_2 - \phi_3)B^3 + \phi_3 B^4] y_t \\ = (1 + \theta_1 B)\varepsilon_t, \end{aligned}$$

# Point forecasts

## ARIMA(3,1,1) forecasts: Step 1

$$(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3)(1 - B)y_t = (1 + \theta_1 B)\varepsilon_t,$$

$$\begin{aligned} [1 - (1 + \phi_1)B + (\phi_1 - \phi_2)B^2 + (\phi_2 - \phi_3)B^3 + \phi_3 B^4] y_t \\ = (1 + \theta_1 B)\varepsilon_t, \end{aligned}$$

$$\begin{aligned} y_t - (1 + \phi_1)y_{t-1} + (\phi_1 - \phi_2)y_{t-2} + (\phi_2 - \phi_3)y_{t-3} \\ + \phi_3 y_{t-4} = \varepsilon_t + \theta_1 \varepsilon_{t-1}. \end{aligned}$$

# Point forecasts

## ARIMA(3,1,1) forecasts: Step 1

$$(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3)(1 - B)y_t = (1 + \theta_1 B)\varepsilon_t,$$

$$\begin{aligned} [1 - (1 + \phi_1)B + (\phi_1 - \phi_2)B^2 + (\phi_2 - \phi_3)B^3 + \phi_3 B^4] y_t \\ = (1 + \theta_1 B)\varepsilon_t, \end{aligned}$$

$$\begin{aligned} y_t - (1 + \phi_1)y_{t-1} + (\phi_1 - \phi_2)y_{t-2} + (\phi_2 - \phi_3)y_{t-3} \\ + \phi_3 y_{t-4} = \varepsilon_t + \theta_1 \varepsilon_{t-1}. \end{aligned}$$

$$\begin{aligned} y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} \\ - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}. \end{aligned}$$

## Point forecasts (h=1)

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} \\ - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

## Point forecasts (h=1)

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} \\ - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

### ARIMA(3,1,1) forecasts: Step 2

$$y_{T+1} = (1 + \phi_1)y_T - (\phi_1 - \phi_2)y_{T-1} - (\phi_2 - \phi_3)y_{T-2} \\ - \phi_3 y_{T-3} + \varepsilon_{T+1} + \theta_1 \varepsilon_T.$$

## Point forecasts (h=1)

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

### ARIMA(3,1,1) forecasts: Step 2

$$y_{T+1} = (1 + \phi_1)y_T - (\phi_1 - \phi_2)y_{T-1} - (\phi_2 - \phi_3)y_{T-2} - \phi_3 y_{T-3} + \varepsilon_{T+1} + \theta_1 \varepsilon_T.$$

### ARIMA(3,1,1) forecasts: Step 3

$$\hat{y}_{T+1|T} = (1 + \phi_1)y_T - (\phi_1 - \phi_2)y_{T-1} - (\phi_2 - \phi_3)y_{T-2} - \phi_3 y_{T-3} + \theta_1 e_T.$$

## Point forecasts (h=2)

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} \\ - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

## Point forecasts (h=2)

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

### ARIMA(3,1,1) forecasts: Step 2

$$y_{T+2} = (1 + \phi_1)y_{T+1} - (\phi_1 - \phi_2)y_T - (\phi_2 - \phi_3)y_{T-1} - \phi_3 y_{T-2} + \varepsilon_{T+2} + \theta_1 \varepsilon_{T+1}.$$

## Point forecasts (h=2)

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} - \phi_3 y_{t-4} + \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

### ARIMA(3,1,1) forecasts: Step 2

$$y_{T+2} = (1 + \phi_1)y_{T+1} - (\phi_1 - \phi_2)y_T - (\phi_2 - \phi_3)y_{T-1} - \phi_3 y_{T-2} + \varepsilon_{T+2} + \theta_1 \varepsilon_{T+1}.$$

### ARIMA(3,1,1) forecasts: Step 3

$$\hat{y}_{T+2|T} = (1 + \phi_1)\hat{y}_{T+1|T} - (\phi_1 - \phi_2)y_T - (\phi_2 - \phi_3)y_{T-1} - \phi_3 y_{T-2}.$$

# Prediction intervals

## 95% prediction interval

$$\hat{y}_{T+h|T} \pm 1.96 \sqrt{v_{T+h|T}}$$

where  $v_{T+h|T}$  is estimated forecast variance.

# Prediction intervals

## 95% prediction interval

$$\hat{y}_{T+h|T} \pm 1.96 \sqrt{v_{T+h|T}}$$

where  $v_{T+h|T}$  is estimated forecast variance.

- $v_{T+1|T} = \hat{\sigma}^2$  for all ARIMA models regardless of parameters and orders.
- Multi-step prediction intervals for ARIMA(0,0,q):

$$y_t = \varepsilon_t + \sum_{i=1}^q \theta_i \varepsilon_{t-i}.$$

$$v_{T|T+h} = \hat{\sigma}^2 \left[ 1 + \sum_{i=1}^{h-1} \theta_i^2 \right], \quad \text{for } h = 2, 3, \dots$$

# Prediction intervals

## 95% Prediction interval

$$\hat{y}_{T+h|T} \pm 1.96 \sqrt{v_{T+h|T}}$$

where  $v_{T+h|T}$  is estimated forecast variance.

- Multi-step prediction intervals for ARIMA(0,0,q):

$$y_t = \varepsilon_t + \sum_{i=1}^q \theta_i \varepsilon_{t-i}.$$

$$v_{T|T+h} = \hat{\sigma}^2 \left[ 1 + \sum_{i=1}^{h-1} \theta_i^2 \right], \quad \text{for } h = 2, 3, \dots$$

# Prediction intervals

## 95% Prediction interval

$$\hat{y}_{T+h|T} \pm 1.96 \sqrt{v_{T+h|T}}$$

where  $v_{T+h|T}$  is estimated forecast variance.

- Multi-step prediction intervals for ARIMA(0,0,q):

$$y_t = \varepsilon_t + \sum_{i=1}^q \theta_i \varepsilon_{t-i}.$$

$$v_{T|T+h} = \hat{\sigma}^2 \left[ 1 + \sum_{i=1}^{h-1} \theta_i^2 \right], \quad \text{for } h = 2, 3, \dots$$

- AR(1): Rewrite as MA( $\infty$ ) and use above result.
- Other models beyond scope of this subject.

# Prediction intervals

- Prediction intervals **increase in size with forecast horizon.**
- Prediction intervals can be difficult to calculate by hand
- Calculations assume residuals are **uncorrelated** and **normally distributed**.
- Prediction intervals tend to be too narrow.
  - the uncertainty in the parameter estimates has not been accounted for.
  - the ARIMA model assumes historical patterns will not change during the forecast period.
  - the ARIMA model assumes uncorrelated future errors

## Your turn

For the usgdp data:

- if necessary, find a suitable Box-Cox transformation for the data;
- fit a suitable ARIMA model to the transformed data using `auto.arima()`;
- check the residual diagnostics;
- produce forecasts of your fitted model. Do the forecasts look reasonable?

# Outline

- 1 Stationarity and differencing**
- 2 Non-seasonal ARIMA models**
- 3 Estimation and order selection**
- 4 ARIMA modelling in R**
- 5 Forecasting**
- 6 Seasonal ARIMA models**
- 7 ARIMA vs ETS**

# Seasonal ARIMA models

ARIMA	$\underbrace{(p, d, q)}$	$\underbrace{(P, D, Q)_m}$
	↑	↑
	Non-seasonal part of the model	Seasonal part of the model

where  $m$  = number of observations per year.

## Seasonal ARIMA models

E.g., ARIMA(1, 1, 1)(1, 1, 1)<sub>4</sub> model (without constant)

## Seasonal ARIMA models

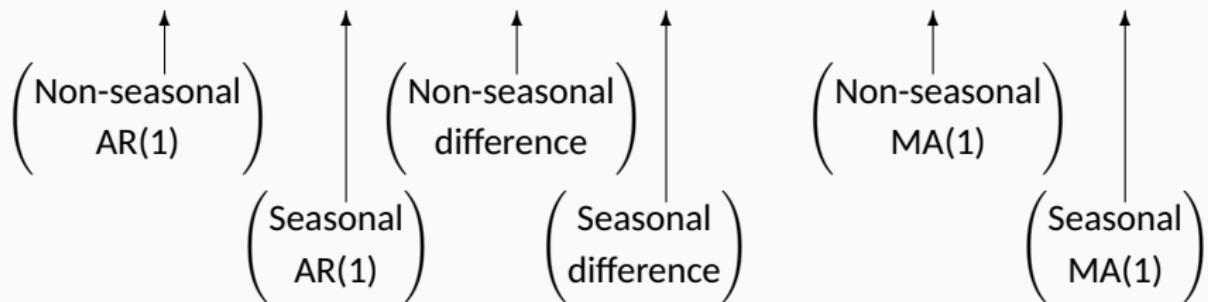
E.g., ARIMA(1, 1, 1)(1, 1, 1)<sub>4</sub> model (without constant)

$$(1-\phi_1B)(1-\Phi_1B^4)(1-B)(1-B^4)y_t = (1+\theta_1B)(1+\Theta_1B^4)\varepsilon_t.$$

# Seasonal ARIMA models

E.g., ARIMA(1, 1, 1)(1, 1, 1)<sub>4</sub> model (without constant)

$$(1 - \phi_1 B)(1 - \Phi_1 B^4)(1 - B)(1 - B^4)y_t = (1 + \theta_1 B)(1 + \Theta_1 B^4)\varepsilon_t.$$



## Seasonal ARIMA models

E.g., ARIMA(1, 1, 1)(1, 1, 1)<sub>4</sub> model (without constant)

$$(1 - \phi_1 B)(1 - \Phi_1 B^4)(1 - B)(1 - B^4)y_t = (1 + \theta_1 B)(1 + \Theta_1 B^4)\varepsilon_t.$$

All the factors can be multiplied out and the general model written as follows:

$$\begin{aligned}y_t = & (1 + \phi_1)y_{t-1} - \phi_1 y_{t-2} + (1 + \Phi_1)y_{t-4} \\& - (1 + \phi_1 + \Phi_1 + \phi_1\Phi_1)y_{t-5} + (\phi_1 + \phi_1\Phi_1)y_{t-6} \\& - \Phi_1 y_{t-8} + (\Phi_1 + \phi_1\Phi_1)y_{t-9} - \phi_1\Phi_1 y_{t-10} \\& + \varepsilon_t + \theta_1\varepsilon_{t-1} + \Theta_1\varepsilon_{t-4} + \theta_1\Theta_1\varepsilon_{t-5}.\end{aligned}$$

## Common ARIMA models

The US Census Bureau uses the following models most often:

- |                                  |                         |
|----------------------------------|-------------------------|
| ARIMA(0,1,1)(0,1,1) <sub>m</sub> | with log transformation |
| ARIMA(0,1,2)(0,1,1) <sub>m</sub> | with log transformation |
| ARIMA(2,1,0)(0,1,1) <sub>m</sub> | with log transformation |
| ARIMA(0,2,2)(0,1,1) <sub>m</sub> | with log transformation |
| ARIMA(2,1,2)(0,1,1) <sub>m</sub> | with no transformation  |

# Seasonal ARIMA models

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF.

**ARIMA(0,0,0)(0,0,1)<sub>12</sub> will show:**

- a spike at lag 12 in the ACF but no other significant spikes.
- The PACF will show exponential decay in the seasonal lags; that is, at lags 12, 24, 36, ....

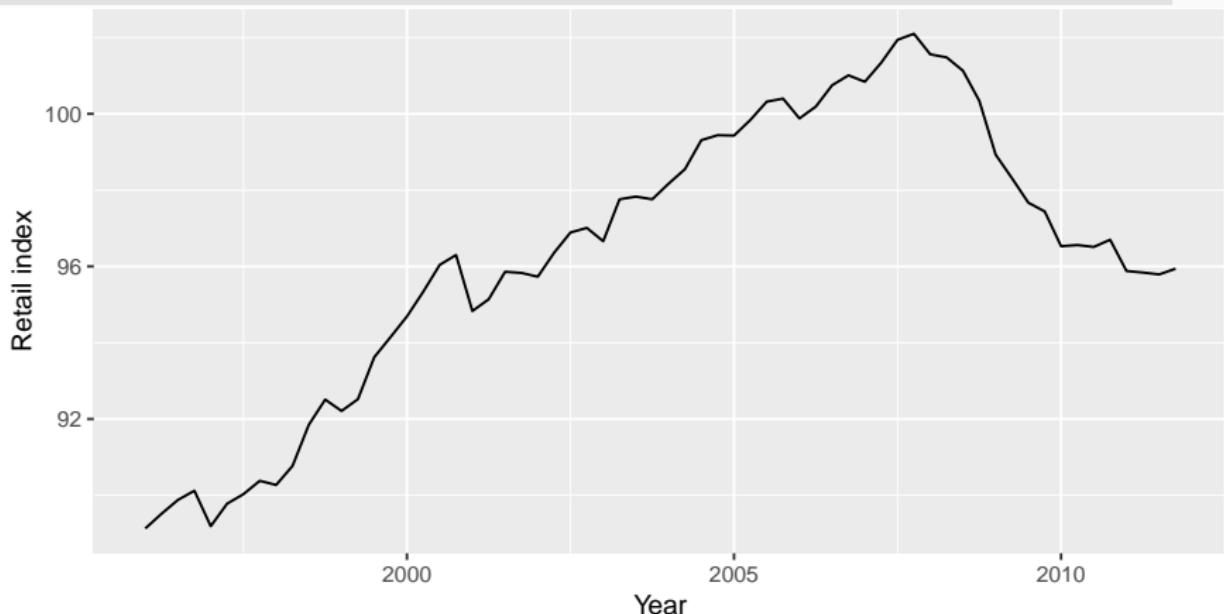
**ARIMA(0,0,0)(1,0,0)<sub>12</sub> will show:**

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

# European quarterly retail trade

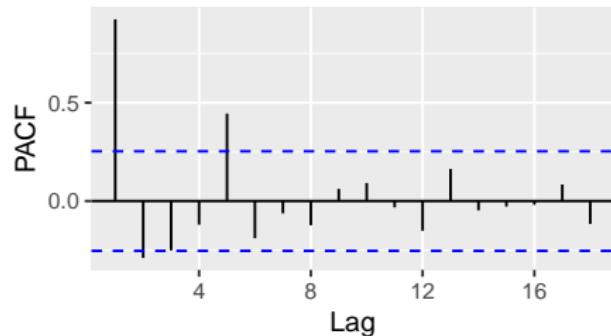
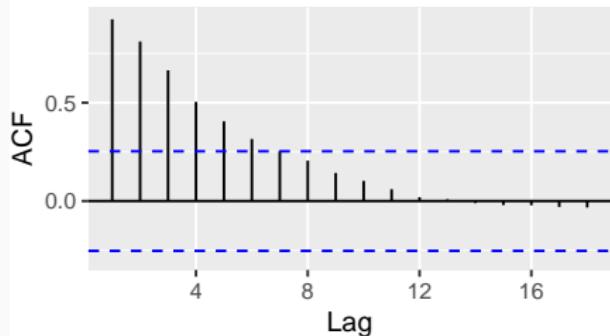
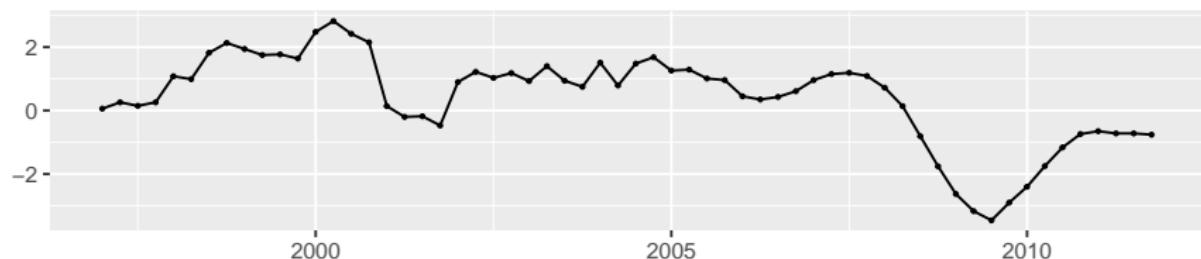
```
autoplot(euretail) +
```

```
xlab("Year") + ylab("Retail index")
```



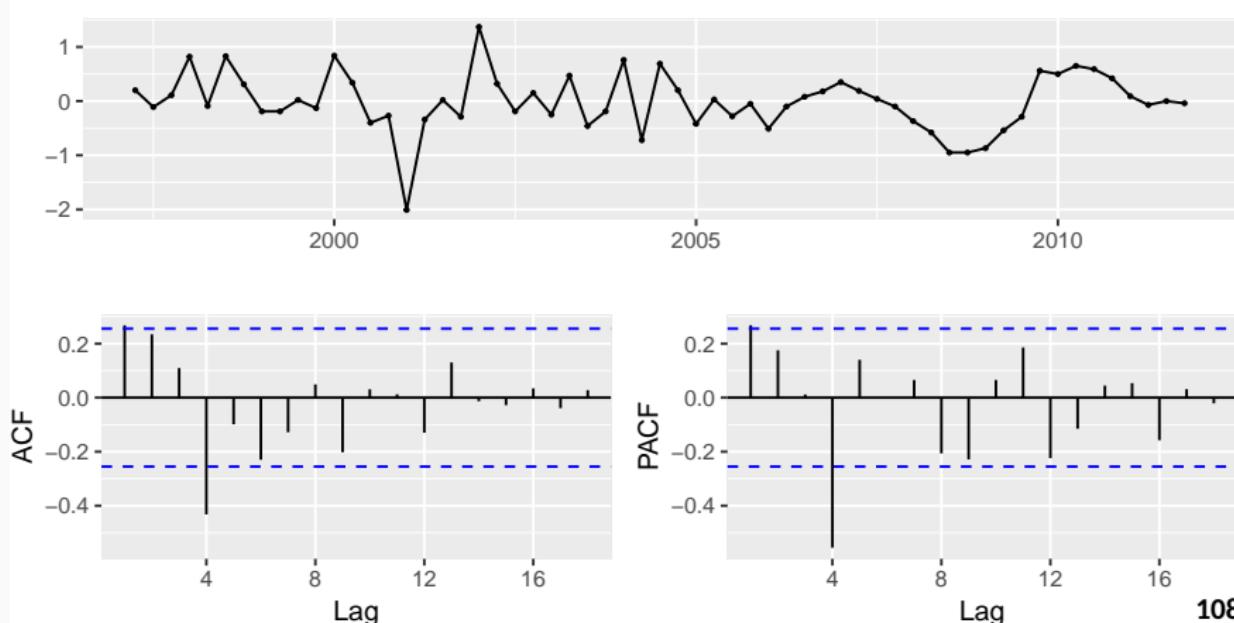
# European quarterly retail trade

```
euretail %>% diff(lag=4) %>% ggtsdisplay()
```



# European quarterly retail trade

```
euretail %>% diff(lag=4) %>% diff() %>%  
ggtsdisplay()
```



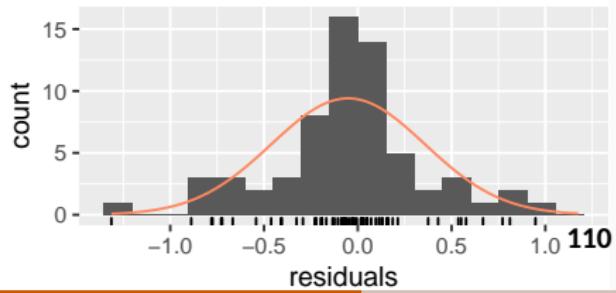
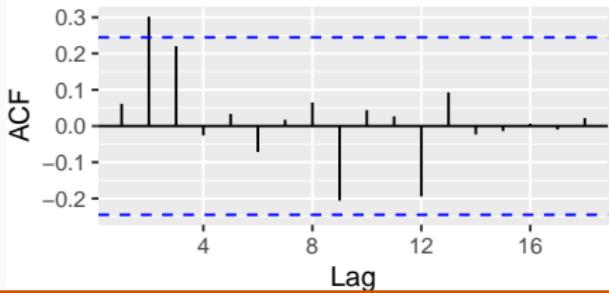
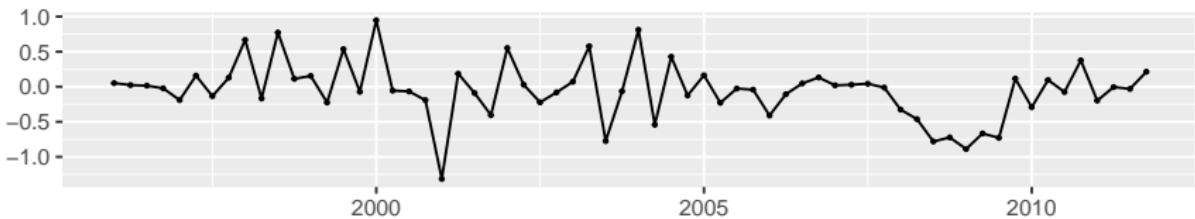
## European quarterly retail trade

- $d = 1$  and  $D = 1$  seems necessary.
- Significant spike at lag 1 in ACF suggests non-seasonal MA(1) component.
- Significant spike at lag 4 in ACF suggests seasonal MA(1) component.
- Initial candidate model: ARIMA(0,1,1)(0,1,1)<sub>4</sub>.
- We could also have started with ARIMA(1,1,0)(1,1,0)<sub>4</sub>.

# European quarterly retail trade

```
fit <- Arima(euretail, order=c(0,1,1),  
             seasonal=c(0,1,1))  
  
checkresiduals(fit)
```

Residuals from ARIMA(0,1,1)(0,1,1)[4]



# European quarterly retail trade

```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,1)(0,1,1)[4]  
## Q* = 10.654, df = 6, p-value = 0.09968  
##  
## Model df: 2. Total lags used: 8
```

## European quarterly retail trade

- ACF and PACF of residuals show significant spikes at lag 2, and maybe lag 3.
- AICc of ARIMA(0,1,2)(0,1,1)<sub>4</sub> model is 74.27.
- AICc of ARIMA(0,1,3)(0,1,1)<sub>4</sub> model is 68.39.

## European quarterly retail trade

- ACF and PACF of residuals show significant spikes at lag 2, and maybe lag 3.
- AICc of ARIMA(0,1,2)(0,1,1)<sub>4</sub> model is 74.27.
- AICc of ARIMA(0,1,3)(0,1,1)<sub>4</sub> model is 68.39.

```
fit <- Arima(euretail, order=c(0,1,3),  
             seasonal=c(0,1,1))  
checkresiduals(fit)
```

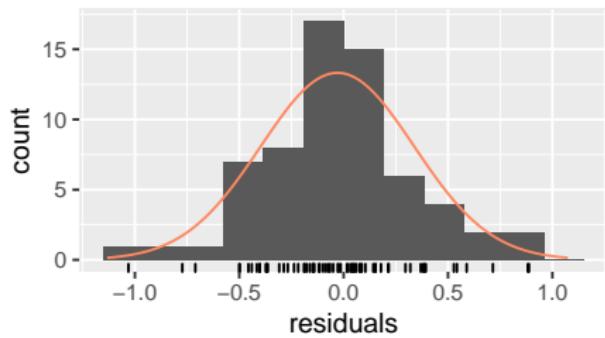
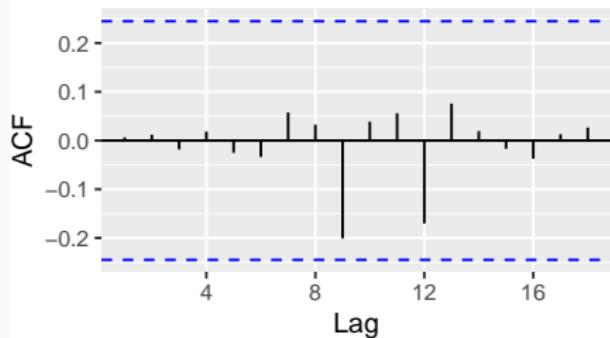
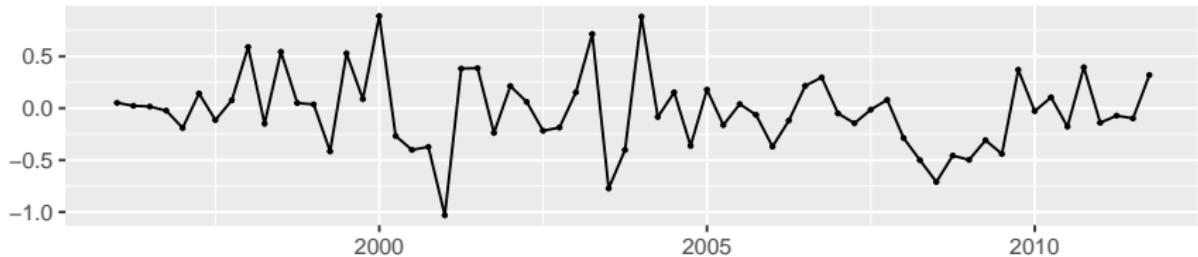
# European quarterly retail trade

```
## Series: euretail
## ARIMA(0,1,3)(0,1,1)[4]
##
## Coefficients:
##          ma1      ma2      ma3     sma1
##          0.2630   0.3694   0.4200  -0.6636
## s.e.  0.1237   0.1255   0.1294   0.1545
##
## sigma^2 estimated as 0.156:  log likelihood=-28.63
## AIC=67.26    AICc=68.39    BIC=77.65
```

# European quarterly retail trade

**checkresiduals(fit)**

Residuals from ARIMA(0,1,3)(0,1,1)[4]



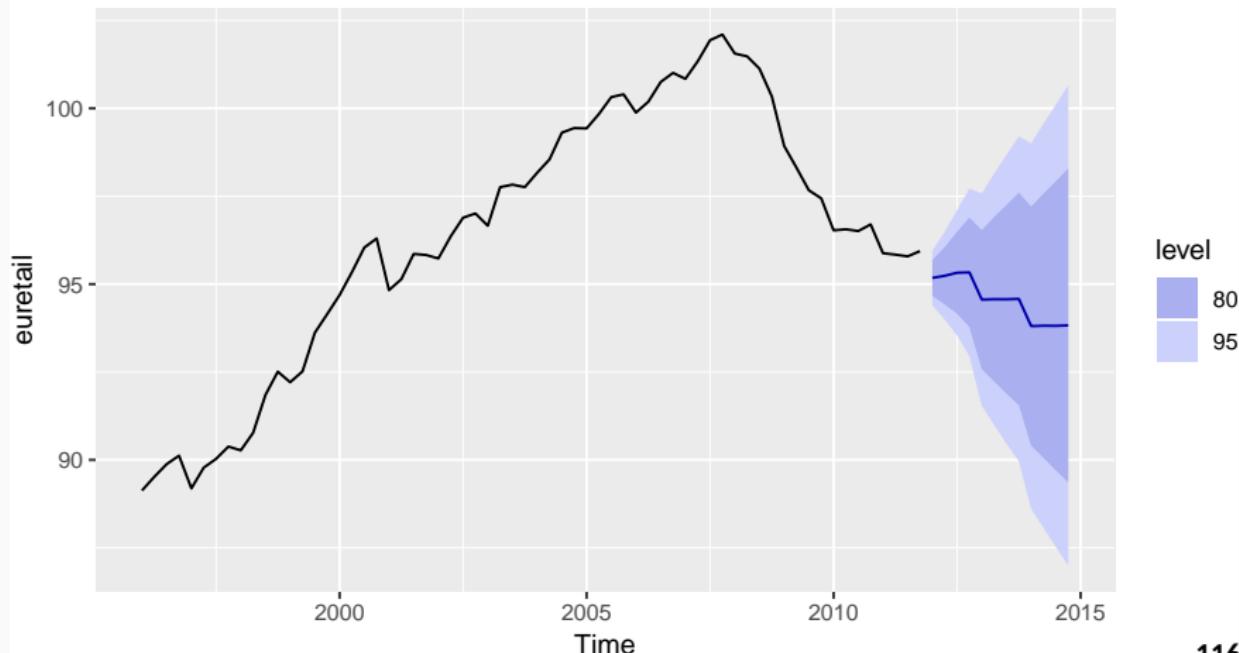
# European quarterly retail trade

```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,3)(0,1,1)[4]  
## Q* = 0.51128, df = 4, p-value = 0.9724  
##  
## Model df: 4. Total lags used: 8
```

# European quarterly retail trade

```
autoplot(forecast(fit, h=12))
```

Forecasts from ARIMA(0,1,3)(0,1,1)[4]



# European quarterly retail trade

```
auto.arima(euretail)

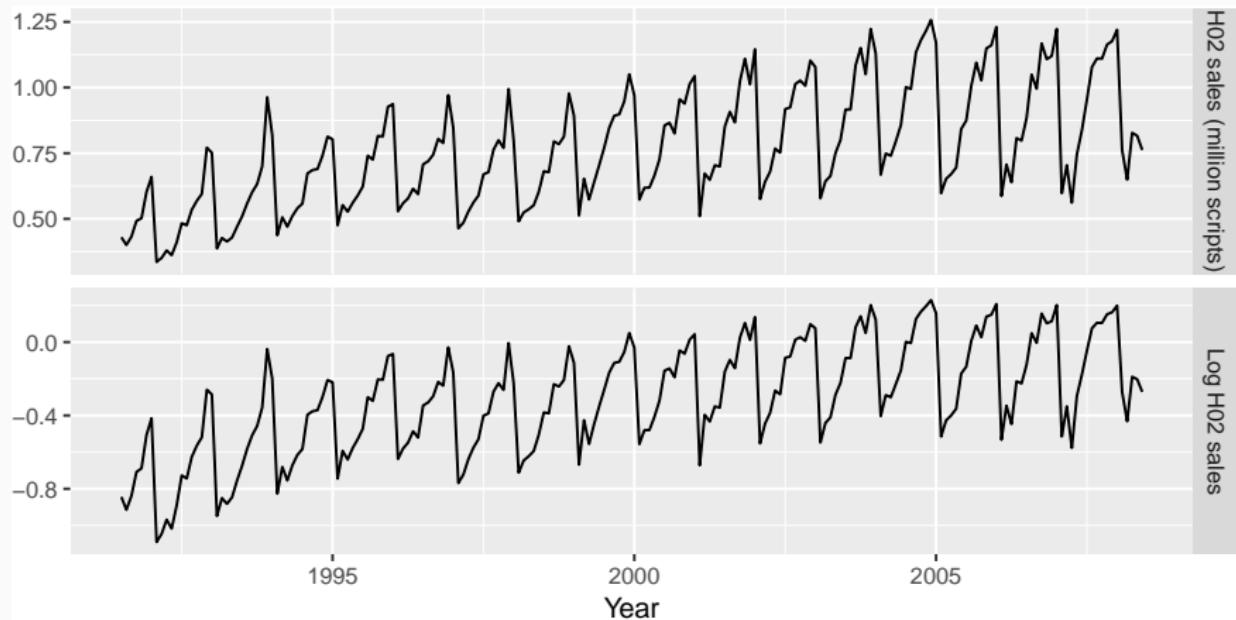
## Series: euretail
## ARIMA(1,1,2)(0,1,1)[4]
##
## Coefficients:
##             ar1      ma1      ma2      sma1
##             0.7362  -0.4663  0.2163  -0.8433
## s.e.    0.2243   0.1990  0.2101   0.1876
##
## sigma^2 estimated as 0.1587:  log likelihood=-29.62
## AIC=69.24    AICc=70.38    BIC=79.63
```

# European quarterly retail trade

```
auto.arima(euretail,
stepwise=FALSE, approximation=FALSE)

## Series: euretail
## ARIMA(0,1,3)(0,1,1)[4]
##
## Coefficients:
##          ma1      ma2      ma3      sma1
##          0.2630   0.3694   0.4200  -0.6636
## s.e.  0.1237   0.1255   0.1294   0.1545
##
## sigma^2 estimated as 0.156: log likelihood=-28.63
## AIC=67.26    AICc=68.39    BIC=77.65
```

# Cortecosteroid drug sales



# Cortecosteroid drug sales



## Cortecosteroid drug sales

- Choose  $D = 1$  and  $d = 0$ .
- Spikes in PACF at lags 12 and 24 suggest seasonal AR(2) term.
- Spikes in PACF suggests possible non-seasonal AR(3) term.
- Initial candidate model: ARIMA(3,0,0)(2,1,0)<sub>12</sub>.

## Cortecosteroid drug sales

Model	AICc
ARIMA(3,0,1)(0,1,2) <sub>12</sub>	-485.48
ARIMA(3,0,1)(1,1,1) <sub>12</sub>	-484.25
ARIMA(3,0,1)(0,1,1) <sub>12</sub>	-483.67
ARIMA(3,0,1)(2,1,0) <sub>12</sub>	-476.31
ARIMA(3,0,0)(2,1,0) <sub>12</sub>	-475.12
ARIMA(3,0,2)(2,1,0) <sub>12</sub>	-474.88
ARIMA(3,0,1)(1,1,0) <sub>12</sub>	-463.40

# Cortecosteroid drug sales

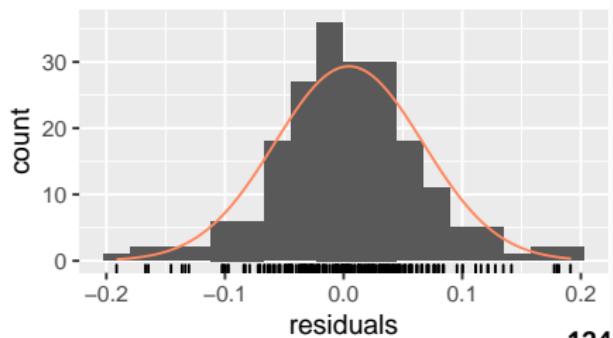
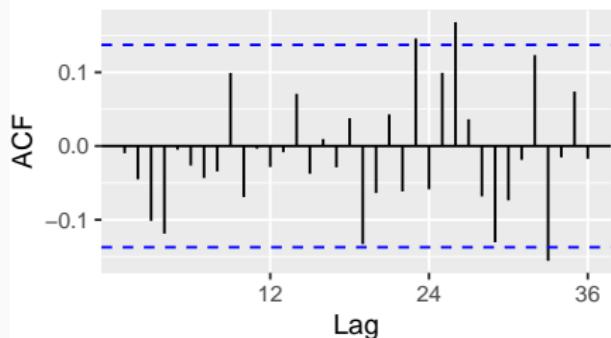
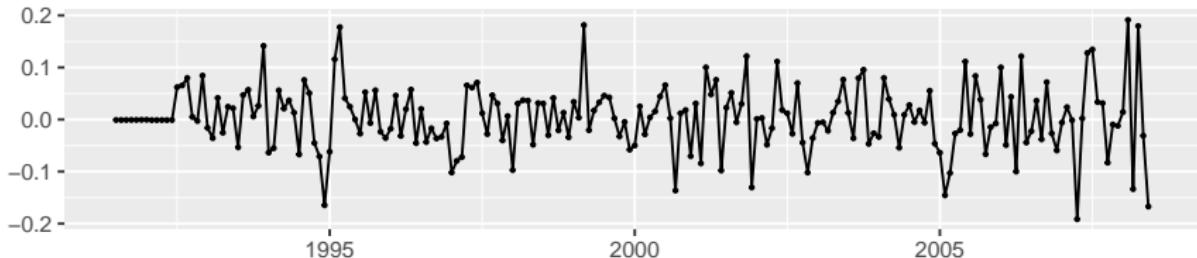
```
(fit <- Arima(h02, order=c(3,0,1), seasonal=c(0,1,2),
lambda=0))

## Series: h02
## ARIMA(3,0,1)(0,1,2)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##             ar1      ar2      ar3      ma1      sma1      sma2
##             -0.1603   0.5481   0.5678   0.3827  -0.5222  -0.1768
## s.e.      0.1636   0.0878   0.0942   0.1895   0.0861   0.0872
##
## sigma^2 estimated as 0.004278: log likelihood=250.04
## AIC=-486.08    AICc=-485.48    BIC=-463.28
```

# Cortecosteroid drug sales

`checkresiduals(fit, lag=36)`

Residuals from ARIMA(3,0,1)(0,1,2)[12]



# Cortecosteroid drug sales

```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(3,0,1)(0,1,2)[12]  
## Q* = 50.712, df = 30, p-value = 0.01045  
##  
## Model df: 6. Total lags used: 36
```

# Cortecosteroid drug sales

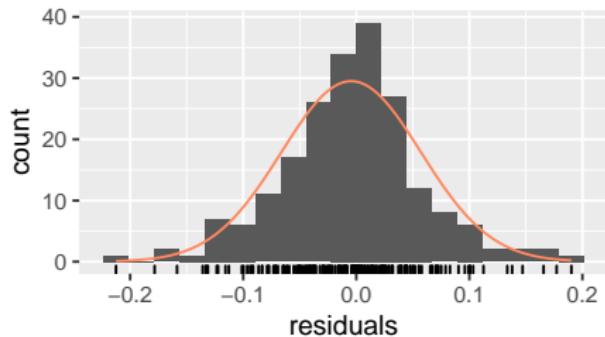
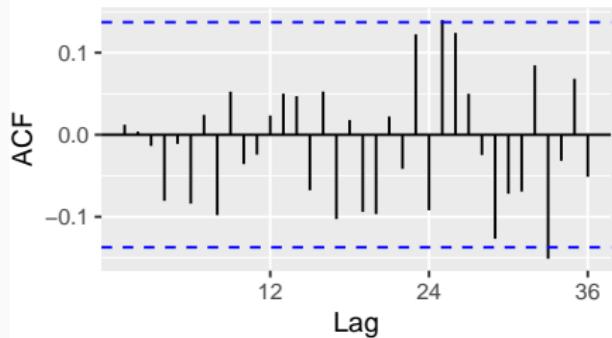
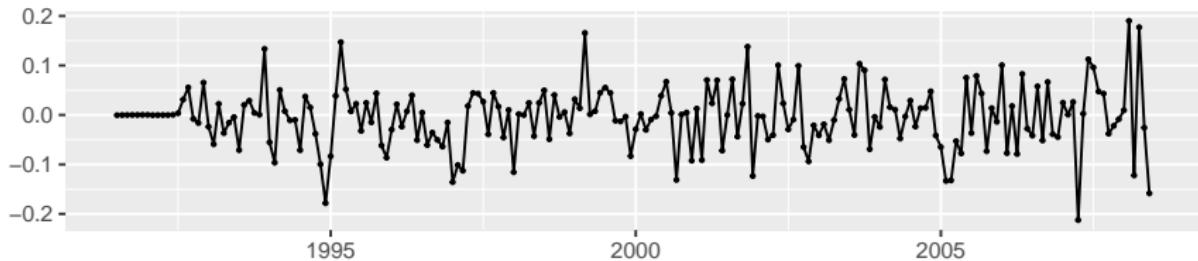
```
(fit <- auto.arima(h02, lambda=0))

## Series: h02
## ARIMA(2,1,3)(0,1,1)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##          ar1      ar2      ma1      ma2      ma3      sma1
##         -1.0194  -0.8351  0.1717  0.2578  -0.4206  -0.6528
## s.e.    0.1648   0.1203  0.2079  0.1177   0.1060   0.0657
##
## sigma^2 estimated as 0.004203: log likelihood=250.8
## AIC=-487.6   AICc=-486.99   BIC=-464.83
```

# Cortecosteroid drug sales

```
checkresiduals(fit, lag=36)
```

Residuals from ARIMA(2,1,3)(0,1,1)[12]



# Cortecosteroid drug sales

```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,1,3)(0,1,1)[12]  
## Q* = 46.149, df = 30, p-value = 0.03007  
##  
## Model df: 6. Total lags used: 36
```

# Cortecosteroid drug sales

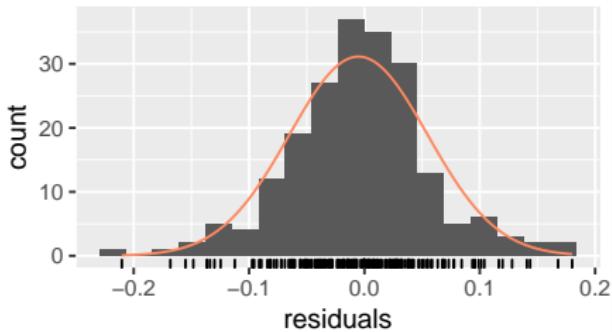
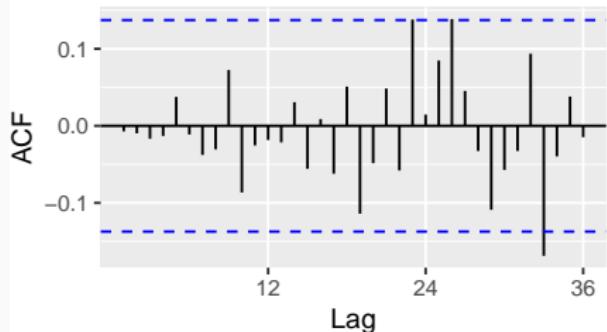
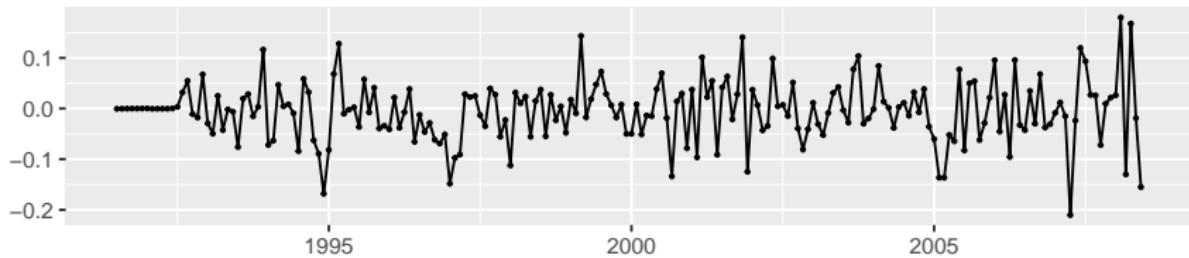
```
(fit <- auto.arima(h02, lambda=0, max.order=9,
  stepwise=FALSE, approximation=FALSE))

## Series: h02
## ARIMA(4,1,1)(2,1,2)[12]
## Box Cox transformation: lambda= 0
##
## Coefficients:
##             ar1      ar2      ar3      ar4      ma1      sar1      sar2      sma1
##             -0.0425  0.2098  0.2017  -0.2273  -0.7424  0.6213  -0.3832  -1.2019
## s.e.      0.2167  0.1813  0.1144   0.0810   0.2074  0.2421   0.1185  0.2491
##             sma2
##             0.4959
## s.e.      0.2135
##
## sigma^2 estimated as 0.004049:  log likelihood=254.31
## AIC=-488.63  AICc=-487.4  BIC=-456.1
```

# Cortecosteroid drug sales

```
checkresiduals(fit, lag=36)
```

Residuals from ARIMA(4,1,1)(2,1,2)[12]



```
##
```

130

```
## Living Box test
```

# Cortecosteroid drug sales

```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(4,1,1)(2,1,2)[12]  
## Q* = 36.456, df = 27, p-value = 0.1057  
##  
## Model df: 9. Total lags used: 36
```

# Cortecosteroid drug sales

Training data: July 1991 to June 2006

Test data: July 2006–June 2008

```
getrmse <- function(x,h,...)
{
  train.end <- time(x)[length(x)-h]
  test.start <- time(x)[length(x)-h+1]
  train <- window(x,end=train.end)
  test <- window(x,start=test.start)
  fit <- Arima(train,...)
  fc <- forecast(fit,h=h)
  return(accuracy(fc,test)[2,"RMSE"])
}
getrmse(h02,h=24,order=c(3,0,0),seasonal=c(2,1,0),lambda=0)
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(2,1,0),lambda=0)
getrmse(h02,h=24,order=c(3,0,2),seasonal=c(2,1,0),lambda=0)
getrmse(h02,h=24,order=c(3,0,1),seasonal=c(1,1,0),lambda=0)
```

# Cortecosteroid drug sales

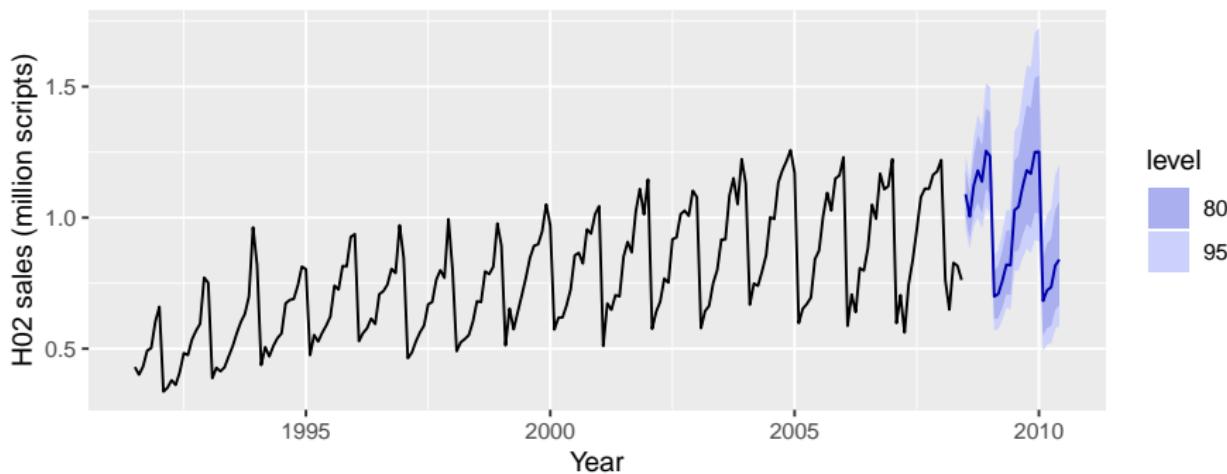
Model	RMSE
ARIMA(4,1,1)(2,1,2)[12]	0.0615
ARIMA(3,0,1)(0,1,2)[12]	0.0622
ARIMA(3,0,1)(1,1,1)[12]	0.0630
ARIMA(2,1,4)(0,1,1)[12]	0.0632
ARIMA(2,1,3)(0,1,1)[12]	0.0634
ARIMA(3,0,3)(0,1,1)[12]	0.0639
ARIMA(2,1,5)(0,1,1)[12]	0.0640
ARIMA(3,0,1)(0,1,1)[12]	0.0644
ARIMA(3,0,2)(0,1,1)[12]	0.0644
ARIMA(3,0,2)(2,1,0)[12]	0.0645
ARIMA(3,0,1)(2,1,0)[12]	0.0646
ARIMA(3,0,0)(2,1,0)[12]	0.0661

# Cortecosteroid drug sales

- Models with lowest AICc values tend to give slightly better results than the other models.
- AICc comparisons must have the same orders of differencing. But RMSE test set comparisons can involve any models.
- Use the best model available, even if it does not pass all tests.

# Cortecosteroid drug sales

```
fit <- Arima(h02, order=c(3,0,1), seasonal=c(0,1,2),  
lambda=0)  
autoplot(forecast(fit)) +  
ylab("H02 sales (million scripts)") + xlab("Year")  
Forecasts from ARIMA(3,0,1)(0,1,2)[12]
```



# Outline

- 1 Stationarity and differencing**
- 2 Non-seasonal ARIMA models**
- 3 Estimation and order selection**
- 4 ARIMA modelling in R**
- 5 Forecasting**
- 6 Seasonal ARIMA models**
- 7 ARIMA vs ETS**

## ARIMA vs ETS

- Myth that ARIMA models are more general than exponential smoothing.
- Linear exponential smoothing models all special cases of ARIMA models.
- Non-linear exponential smoothing models have no equivalent ARIMA counterparts.
- Many ARIMA models have no exponential smoothing counterparts.
- ETS models all non-stationary. Models with seasonality or non-damped trend (or both) have two unit roots; all other models have one unit root,

# Equivalences

---

ETS model	ARIMA model	Parameters
ETS(A,N,N)	ARIMA(0,1,1)	$\theta_1 = \alpha - 1$
ETS(A,A,N)	ARIMA(0,2,2)	$\theta_1 = \alpha + \beta - 2$ $\theta_2 = 1 - \alpha$
ETS(A,A,N)	ARIMA(1,1,2)	$\phi_1 = \phi$ $\theta_1 = \alpha + \phi\beta - 1 - \phi$ $\theta_2 = (1 - \alpha)\phi$
ETS(A,N,A)	ARIMA(0,0,m)(0,1,0) <sub>m</sub>	
ETS(A,A,A)	ARIMA(0,1,m+1)(0,1,0) <sub>m</sub>	
ETS(A,A,A)	ARIMA(1,0,m+1)(0,1,0) <sub>m</sub>	

---

## Your turn

For the condmilk series:

- Do the data need transforming? If so, find a suitable transformation.
- Are the data stationary? If not, find an appropriate differencing which yields stationary data.
- Identify a couple of ARIMA models that might be useful in describing the time series.
- Which of your models is the best according to their AIC values?
- Estimate the parameters of your best model and do diagnostic testing on the residuals. Do the residuals resemble white noise? If not, try to find another ARIMA model which fits better.
- Forecast the next 24 months of data using your preferred model.



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# **STU33010: Forecasting Lecture: Kalman filter**

A. Benavoli

October 19, 2021

# Outline

## 1 Introduction

## 2 Bayesian filter

- Kalman filter

## 3 ETS

# Focus

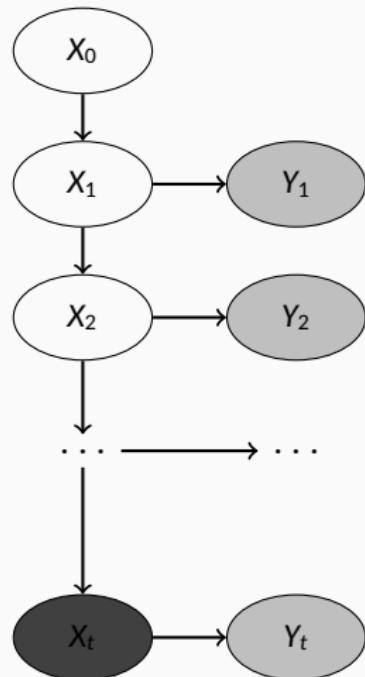
This lecture is about:

**1** Hidden Markov Models

More precisely:

- estimate of the state  $X_t$  given all observations  $y_1, \dots, y_t$ .

**2** Why/How is this related to time-series forecasting?



## Example: the Sinking of the Ship

A crew is involved in a sinking



- to be rescued, they need to communicate via radio an accurate position of the raft;
- they know the position before the sinking but, in the meantime, the raft keeps moving on a wavy sea;
- they have a sextant on-board.

A discussion begins about the optimal way to obtain the best accurate position of the raft.

## An intuitive solution

We know the GPS position of the ship and its variance:

$$x_0 \leftarrow \hat{x}_0, \sigma_0^2$$

before the sinking (time 0)

Assume that the raft is moving along the wind direction, so its position is:

$$x_{k+1} = x_k + d + w_k \quad \text{for } k = 0, 1, 2, \dots, t$$

where  $d$  is a drift term (due to the wind) and

$$w_k \leftarrow 0, \sigma_w^2$$

is a disturbance (due to the waves).

Then at time  $k = 1$ ,

$$x_1 \leftarrow \hat{x}_{1|0}, \sigma_{1|0}^2$$

with  $\hat{x}_{1|0} = \hat{x}_0 + d$ ,  $\sigma_{1|0}^2 = \sigma_0^2 + \sigma_w^2$

## An intuitive solution

Without measurements, the uncertainty in the position will keep growing until we have no clue anymore about where we are.

With the sextant, we can measure our position:

$$y_1 = x_1 + v_1$$

where  $v$  is the measurement error. Assuming that

$$v_1 \leftarrow 0, \sigma_v^2,$$

if we use the measurements, we can improve our previous estimate:

$$x_1 \leftarrow \hat{x}_{1|0}, \sigma_{1|0}^2$$

as follows:

## An intuitive solution

$$x_1 \leftarrow \hat{x}_1, \sigma_1^2$$

where

$$\hat{x}_1 = \frac{\frac{1}{\sigma_{1|0}^2}}{\frac{1}{\sigma_{1|0}^2} + \frac{1}{\sigma_v^2}} \hat{x}_{1|0} + \frac{\frac{1}{\sigma_v^2}}{\frac{1}{\sigma_{1|0}^2} + \frac{1}{\sigma_v^2}} y_1$$

the estimate is a weighted average the measurement  $y_1$  and  $\hat{x}_{1|0}$ , and

$$\frac{1}{\sigma_1^2} = \frac{1}{\sigma_{1|0}^2} + \frac{1}{\sigma_v^2}$$

For instance, if  $\sigma_{1|0}^2 = \sigma_v^2 = \sigma^2$  then  $\hat{x}_1 = \frac{\hat{x}_{1|0} + y_1}{2}$  and  $\sigma_1^2 = \frac{\sigma^2}{2}$

# An intuitive solution

Summing up: dynamic and measurement model

$$x_{k+1} = x_k + d + w_k, \quad y_k = x_k + v_k$$

assumptions:

$$w_k \leftarrow 0, \sigma_w^2, \quad v_k \leftarrow 0, \sigma_v^2,$$

Initialization:

$$x_0 \leftarrow \hat{x}_0, \sigma_0^2,$$

Repeat for  $k = 1, \dots, t$ :

$$\left\{ \begin{array}{lcl} \hat{x}_{k|k-1} & = & \hat{x}_{k-1} + d \\ \sigma_{k|k-1}^2 & = & \sigma_{k-1}^2 + \sigma_w^2 \\ \\ \hat{x}_k & = & \hat{x}_{k|k-1} + K_k(y_k - \hat{x}_{k|k-1}) \\ \sigma_k^2 & = & (1 - K_k)\sigma_{k|k-1}^2 \\ K_k & = & \frac{\sigma_{k|k-1}^2}{\sigma_{k|k-1}^2 + \sigma_v^2} \end{array} \right.$$

# Outline

1 Introduction

2 Bayesian filter

- Kalman filter

3 ETS

# The Bayesian solution

Model:

$$\begin{cases} x_{k+1} = x_k + d + w_k \\ y_k = x_k + v_k \end{cases}$$

Goal: to estimate  $x_t$  given the measurements  $y_1, y_2, \dots, y_t$  and the prior knowledge:

$$x_0 \leftarrow \hat{x}_0, \sigma_0^2, \quad w_k \leftarrow 0, \sigma_w^2, \quad v_k \leftarrow 0, \sigma_v^2.$$

Since

$v_k \in \mathbb{R}$  is a measurement error

a natural choice is:

$$p(v_k) = N(v_k; 0, \sigma_v^2)$$

and, thus, for mathematical convenience:

$$p(x_0) = N(x_0; \hat{x}_0, \sigma_0^2), \quad p(w_k) = N(w_k; 0, \sigma_w^2)$$

## The Bayesian solution

From the assumption  $p(v_k) = N(v_k; 0, \sigma_v^2)$ ,  $p(w_k) = N(w_k; 0, \sigma_w^2)$  and

$$\begin{cases} x_{k+1} &= x_k + d + w_k \\ y_k &= x_k + v_k \end{cases}$$

it follows that:

$$p(x_{k+1}|x_k) = N(x_{k+1}; x_k + d, \sigma_w^2), \quad p(y_k|x_k) = N(y_k; x_k, \sigma_v^2)$$

### Goal

to estimate  $x_t$  given the measurements  $y_1, y_2, \dots, y_k$ , which means to compute

$$p(x_t|y^k)$$

here  $y^k = \{y_1, y_2, \dots, y_k\}$ .

## The Bayesian solution

Solution: assuming that we have already computed

$$p(x_{k-1}|y^{k-1})$$

then

- Prediction:

$$p(x_k|y^{k-1}) = \int_{x_{k-1}} p(x_k|x_{k-1})p(x_{k-1}|y^{k-1}) dx_{k-1}$$

- Correction:

$$p(x_k|y^k) = \frac{p(y_k|x_k)p(x_k|y^{k-1})}{\int_{x_k} p(y_k|x_k)p(x_k|y^{k-1}) dx_k}$$

We can then compute recursively the previous steps starting from the initialization:

$$p(x_0|y^{-1}) = p(x_0).$$

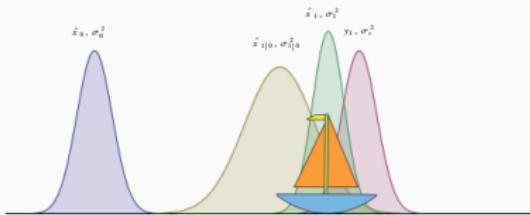
# The Bayesian solution

In the Gaussian case, solution:

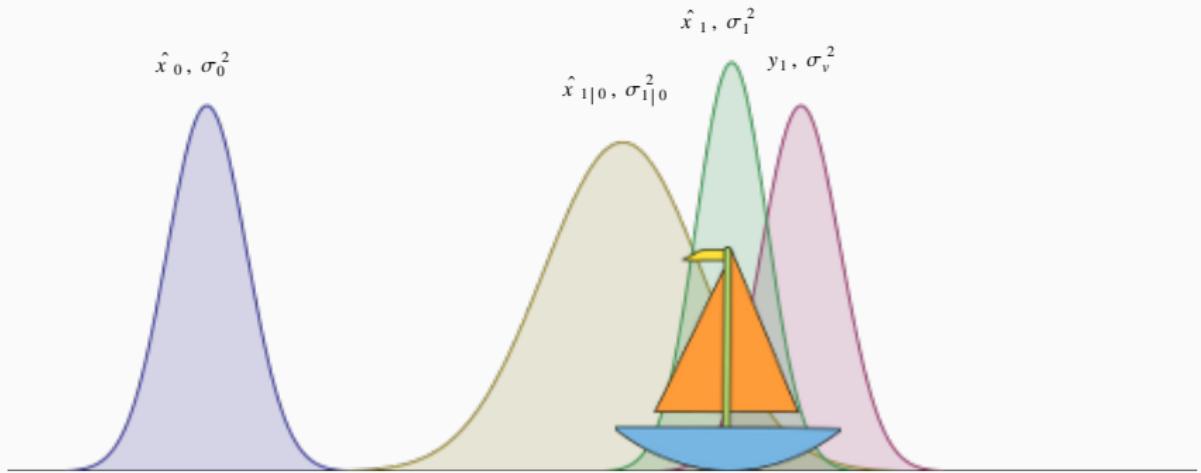
$$p(x_k | y^k) = N(x_k; \hat{x}_k, \sigma_k^2)$$

with

$$\left\{ \begin{array}{lcl} \hat{x}_{k|k-1} & = & \hat{x}_{k-1} + d \\ \sigma_{k|k-1}^2 & = & \sigma_{k-1}^2 + \sigma_w^2 \\ \\ \hat{x}_k & = & \hat{x}_{k|k-1} + K_k(y_k - \hat{x}_{k|k-1}) \\ \sigma_k^2 & = & (1 - K_k)\sigma_{k|k-1}^2 \\ K_k & = & \frac{\sigma_{k|k-1}^2}{\sigma_{k|k-1}^2 + \sigma_v^2} \end{array} \right.$$



# The Bayesian solution



## More general model

This model is not very realistic

$$\begin{cases} x_{k+1} = x_k + d + w_k \\ y_k = x_k + v_k \end{cases}$$

because the drift (due to the wind) won't be a constant

$$d_k = \Delta_k v_k$$

where  $\Delta_k$  is the “sampling time”, that is the time between two measurements..  
The velocity  $v_k$  is a function of time and changes. So a better model is

$$\begin{cases} x_{k+1} = x_k + \Delta_k v_k + w_k^{(1)} \\ v_{k+1} = v_k + w_k^{(2)} \\ y_k = x_k + v_k \end{cases}$$

this time we aim two unknowns  $x_k, v_k$ , which we need to estimate from the position measurement.

## The Bayesian solution

We first write it in matrix form

$$\mathbf{z}_k = \begin{bmatrix} x_k \\ v_k \end{bmatrix}, \quad F_k = \begin{bmatrix} 1 & \Delta_k \\ 0 & 1 \end{bmatrix}, \quad H_k = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{w}_k = \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \end{bmatrix}$$

We rewrite

$$\begin{cases} x_{k+1} &= x_k + \Delta_k v_k + w_k^{(1)} \\ v_{k+1} &= v_k + w_k^{(2)} \\ y_k &= x_k + v_k \end{cases}$$

as

$$\begin{cases} \mathbf{z}_{k+1} &= F_k \mathbf{z}_k + \mathbf{w}_k \\ y_k &= H_k \mathbf{z}_k + v_k \end{cases}$$

where

$$\mathbf{w}_k \sim N \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_{w^{(1)}} & 0 \\ 0 & \sigma_{w^{(2)}} \end{bmatrix} \right) = N(\mathbf{0}, Q_k), \quad v_k \sim N(0, R_k)$$

## Kalman filter - Problem formulation

The dynamic state evolution model is linear

$$\mathbf{z}_{k+1} = F_k \mathbf{z}_k + \mathbf{w}_k$$

The measurement model is linear

$$y_k = H_k \mathbf{z}_k + v_k$$

- $\mathbf{w}_k$  and  $v_k$  are zero-mean, white Gaussian, mutually independent:

$$\mathbf{w}_k = N(0, Q_k) \quad v_k = N(0, R_k)$$

- Initial state PDF:  $\mathbf{z}_0 = N(\hat{\mathbf{z}}_0, P_0)$ ;
- Goal: recursive computation of posterior density

$$p(\mathbf{z}_k | y^k)$$

# Kalman filter

Solution:

$$p(\mathbf{z}_k | \mathbf{y}^k) = N(\hat{\mathbf{z}}_k, P_k)$$

where:

$$\hat{\mathbf{z}}_k = E[\mathbf{z}_k | \mathbf{Z}^k] \quad \text{and} \quad P_k = E[(\mathbf{z}_k - \hat{\mathbf{z}}_k)(\mathbf{z}_k - \hat{\mathbf{z}}_k)']$$

Mean and covariance can be calculated recursively:

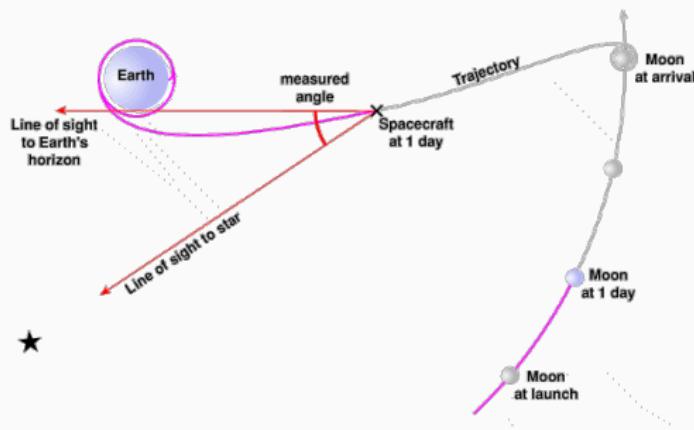
$$\left\{ \begin{array}{ll} \hat{\mathbf{z}}_{k|k-1} = F_{k-1}\hat{\mathbf{z}}_{k-1} & \text{state estimate prediction} \\ P_{k|k-1} = F_{k-1}P_{k-1}F'_{k-1} + Q_{k-1} & \text{covariance predicted estimate} \\ \hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k|k-1} + K_k(y_k - H_k\hat{\mathbf{z}}_{k|k-1}) & \text{state estimate updating} \\ P_k = P_{k|k-1} - K_kH_kP_{k|k-1} & \text{covariance updated estimate} \\ K_k = P_{k|k-1}H'_kS_k^{-1} & \text{Kalman gain} \\ S_k = H_kP_{k|k-1}H'_k + R_k & \text{innovation covariance} \end{array} \right.$$

# Kalman filter

The Kalman filter is named after Rudolf E. Kalman. Kalman first published his ideas on:

Kalman, R.E., A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering, Trans. ASME, Series D, Vol. 82, No. 1, 1960, pp. 35-45,

First important application, Apollo missions:



## Many Other Applications

- GPS and inertial navigation systems;
- radar tracking of ships and air-planes;
- finance applications (e.g., estimate of volatility);
- biological applications (e.g., estimate of eggs' hatch);
- weather forecast applications (e.g., estimate of ocean surface temperature);
- etc.

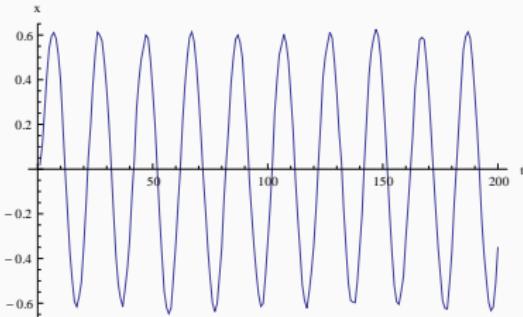
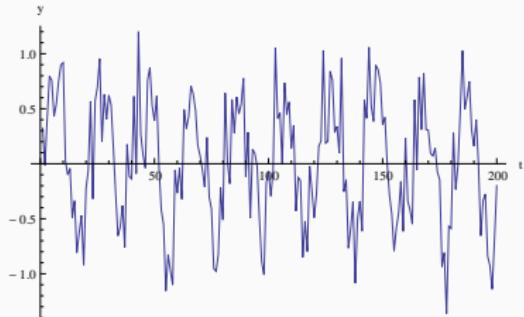
# Why the term filter?

Consider the dynamic and measurement model:

$$x_{t+1} = \frac{1}{2}x_t + \frac{1}{3} \sin\left(\frac{2\pi t}{20}\right) + w_t, \quad y_t = x_t + v_t,$$

with

$$\sigma_w^2 = 0.01, \quad \sigma_v^2 = 0.1,$$



# Filtering: general Dynamic State Estimation Problem

## Problem:

Estimate the state  $x_k$  of a non-linear dynamic stochastic system from noisy measurements  $y^k = [y_1, y_2, \dots, y_{m_k}]$ .

The dynamic (state evolution) model

$$x_{k+1} = f_k(x_k) + w_k$$

The measurement model

$$y_k = h_k(x_k) + v_k$$

Known:

- functional form of  $f_k(\cdot)$  and  $h_k(\cdot)$ ;
- noise statistics  $p(w_k)$  and  $p(v_k)$ ;
- initial state pdf  $p(x_0)$ .

One can calculate  $p(x_k|x_{k-1})$  and  $p(y_k|x_k)$

## Kalman filter

The importance of KF is that is one of the few cases for which a closed form solution exists for the *optimal recursive Bayesian estimator*.

The most important case:

- dynamic and measurement models are linear functions of the state;
- distributions of the noises are Gaussian.

Unfortunately, in the most of cases, these assumptions do not hold.

We cannot use Kalman filter

Example:

$$x_{k+1} = x_k + d + w_k, \quad y_k = x_k^2 + v_k$$

then  $p(x_k|y^k)$  is not Gaussian anymore.

# Outline

1 Introduction

2 Bayesian filter

- Kalman filter

3 ETS

## How is KF related to time-series?

Consider the model ETS(A,N,N)

State equation

$$\ell_t = \ell_{t-1} + \alpha \varepsilon_t$$

Measurement equation

$$y_t = \ell_{t-1} + \varepsilon_t$$

this does not look a state space model like the one we used for the KF, because the error in the state equation is proportional (and so dependent on) to the error on the measurement equation. Also the time index does not look aligned.

## Auxiliary variable

State equation 1

$$\ell_{k+1} = \ell_k + \alpha \varepsilon_k$$

State equation 2

$$\tilde{y}_{k+1} = \ell_k + \varepsilon_k$$

Measurement equation

$$y_k = \tilde{y}_k + v_k$$

We define

$$\mathbf{z}_k = \begin{bmatrix} \ell_k \\ \tilde{y}_k \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad \mathbf{w}_k = \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \end{bmatrix}$$

$$\left\{ \begin{array}{lcl} \mathbf{z}_{k+1} & = & F\mathbf{z}_k + \mathbf{w}_k \\ y_k & = & H\mathbf{z}_k + v_k \end{array} \right.$$

where

$$\mathbf{w}_k \sim N \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha^2 \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix} \right) = N(\mathbf{0}, Q), \quad v_k \sim N(0, R) \text{ with } R = 0$$

## Likelihood and parameters

The KF returns

$$p(y_1, \dots, y_T)$$

that is the marginal probability of the data which is also Gaussian (it can easily be computed). It is also called “marginal likelihood”.

This probability depends on the parameters of the model  $\alpha, \sigma_w^2, \ell_0$  and, therefore, we can choose these parameters by

$$\max_{\alpha, \sigma_w^2, \ell_0} p(y_1, \dots, y_T)$$

or equivalently

$$\max_{\alpha, \sigma_w^2, \ell_0} \log p(y_1, \dots, y_T)$$

## Kalman filter - Problem formulation

The dynamic state evolution model is linear

$$\mathbf{z}_{k+1} = F_k \mathbf{z}_k + \mathbf{w}_k$$

The measurement model is linear

$$y_k = H_k \mathbf{z}_k + v_k$$

- $\mathbf{w}_k$  and  $v_k$  are zero-mean, white Gaussian, mutually independent:

$$\mathbf{w}_k = N(0, Q_k) \quad v_k = N(0, R_k)$$

- Initial state PDF:  $\mathbf{z}_0 = N(\hat{\mathbf{z}}_0, P_0)$ ;
- Goal: recursive computation of posterior density

$$p(\mathbf{z}_k | y^k)$$

# Kalman filter

Solution:

$$p(\mathbf{z}_k | \mathbf{y}^k) = N(\hat{\mathbf{z}}_k, P_k)$$

where:

$$\hat{\mathbf{z}}_k = E[\mathbf{z}_k | \mathbf{Z}^k] \quad \text{and} \quad P_k = E[(\mathbf{z}_k - \hat{\mathbf{z}}_k)(\mathbf{z}_k - \hat{\mathbf{z}}_k)']$$

Mean and covariance can be calculated recursively:

$$\left\{ \begin{array}{ll} \hat{\mathbf{z}}_{k|k-1} = F_{k-1}\hat{\mathbf{z}}_{k-1} & \text{state estimate prediction} \\ P_{k|k-1} = F_{k-1}P_{k-1}F'_{k-1} + Q_{k-1} & \text{covariance predicted estimate} \\ \hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k|k-1} + K_k(y_k - H_k\hat{\mathbf{z}}_{k|k-1}) & \text{state estimate updating} \\ P_k = P_{k|k-1} - K_kH_kP_{k|k-1} & \text{covariance updated estimate} \\ K_k = P_{k|k-1}H'_kS_k^{-1} & \text{Kalman gain} \\ S_k = H_kP_{k|k-1}H'_k + R_k & \text{innovation covariance} \end{array} \right.$$

## Exercise 1

Consider the following state and measurement equation

$$z_{k+1} = z_k + w_k, \quad y_k = z_k + v_k$$

where  $w_k$  is Gaussian, zero mean and standard deviation  $\sigma_w = 1$ ,  $v_k$  is Gaussian, zero mean and standard deviation  $\sigma_v = 0.8$ . We know that the initial state  $z_0 \sim N(\hat{z}_0 = 1, P_0 = 3)$ .  $z_k, w_k, v_k$  are assumed to be mutually independent.

Given the measurements  $[y_1, y_2] = [0.5, 0.7]$ , compute  $p(z_2|y_1, y_2)$ ,  $p(z_3|y_1, y_2)$  and  $p(z_4|y_1, y_2)$ .

## Solution

Given everything is Gaussian and state and measurement models are linear in the state, we can use the KF to answer those questions.

We start computing

$$p(z_1|y_1) = N(\hat{z}_1, P_1)$$

This is the probability distribution of the state  $z_1$  given the measurement  $y_1$ . It is a Gaussian distribution and  $\hat{z}_1, P_1$  are its mean and variance. We use the equations of the KF to compute this mean and variance.

## Prediction step

Target

$$p(z_1|y_1) = N(\hat{z}_1, P_1)$$

First we apply the prediction equations:

$$\begin{cases} \hat{z}_{1|0} = F_0 \hat{z}_0 & \text{state estimate prediction} \\ P_{1|0} = F_0 P_0 F_0' + Q_0 & \text{covariance predicted estimate} \end{cases}$$

We know that  $\hat{z}_0 = 1$  and  $P_0 = 3$ . We also know  $Q_0 = \sigma_w^2 = 1$ .

From  $z_{k+1} = z_k + w_k$  we know that  $F_0 = 1$ .

$$\begin{cases} \hat{z}_{1|0} = 1 & \text{state estimate prediction} \\ P_{1|0} = 3 + 1 = 4 & \text{covariance predicted estimate} \end{cases}$$

and so

$$p(z_1|y_0) = N(\hat{z}_{1|0}, P_{1|0}) = N(1, 4)$$

This is the predicted state based on the measurements before  $y_1$ , which are none

## Update step

We update the prediction  $\hat{z}_{1|0}, P_{1|0}$  with the measurement  $y_1$  to get:

$$p(z_1|y_1) = N(\hat{z}_1, P_1)$$

$$\left\{ \begin{array}{ll} S_1 = H_1 P_{1|0} H_1' + R_1 & \text{innovation covariance} \\ K_1 = P_{1|0} H_1' S_1^{-1} & \text{Kalman gain} \\ \hat{z}_1 = \hat{z}_{1|0} + K_1(y_1 - H_1 \hat{z}_{1|0}) & \text{state estimate updating} \\ P_1 = P_{1|0} - K_1 H_1 P_{1|0} & \text{covariance updated estimate} \end{array} \right.$$

We know that  $H_1 = 1$  and  $R_1 = 0.8^2 = 0.64$  (variance of the measurement noise). We computed  $\hat{z}_{1|0} = 1, P_{1|0} = 3$  and  $y_1 = 0.5$  so

$$\left\{ \begin{array}{ll} S_1 = 4 + 0.64 = 4.64 & \text{innovation covariance} \\ K_1 = 4/4.64 = 0.86207 & \text{Kalman gain} \\ \hat{z}_1 = 1 + 0.86207(0.5 - 1) = 0.568965 & \text{state estimate updating} \\ P_1 = 4 - 0.86207 \cdot 4 = 0.55172 & \text{covariance updated estimate} \end{array} \right.$$

and so

$$p(z_1|y_1) = N(\hat{z}_1, P_1) = N(0.568965, 0.55172)$$

## Another prediction step

Current

$$p(z_1|y_1) = N(\hat{z}_1, P_1) = N(0.568965, 0.55172)$$

We apply the prediction equations:

$$\begin{cases} \hat{z}_{2|1} = F_1 \hat{z}_1 & \text{state estimate prediction} \\ P_{2|1} = F_1 P_1 F_1' + Q_1 & \text{covariance predicted estimate} \end{cases}$$

We know  $Q_1 = \sigma_w^2 = 1, F_1 = 1$ .

$$\begin{cases} \hat{z}_{2|1} = 0.568965 & \text{state estimate prediction} \\ P_{2|1} = 0.55172 + 1 = 1.55172 & \text{covariance predicted estimate} \end{cases}$$

and so

$$p(z_2|y_1) = N(\hat{z}_{2|1}, P_{2|1}) = N(0.568965, 1.55172)$$

This is the predicted state based on the measurements before  $y_2$ , which are none

## Another update step

Current

$$p(z_2|y_1) = N(\hat{z}_{2|1}, P_{2|1}) = N(0.568965, 1.55172)$$

We apply the update equations with  $y_2 = 0.7$ :

$$\left\{ \begin{array}{ll} S_2 = 1.55172 + 0.64 = 2.19172 & \text{innovation covariance} \\ K_2 = 1.55172 / 2.19172 = 0.70799 & \text{Kalman gain} \\ \hat{z}_1 = 0.568965 + 0.70799(0.7 - 0.568965) = 0.66174 & \text{state estimate updating} \\ P_1 = 1.55172 - 0.70799 \cdot 1.55172 = 0.4532 & \text{covariance updated estimate} \end{array} \right.$$

and so

$$p(z_2|y_1, y_2) = N(\hat{z}_2, P_2) = N(0.66174, 0.4532)$$

This is the filtered state based on the measurements up to  $y_2$ .

## Summarising

<i>measurement</i>	0.5	0.7
<i>prediction</i>	1	0.5689
<i>filtered</i>	0.5689	0.66173

How do we compute the predictions  $p(z_3|y_1, y_2)$  and  $p(z_4|y_1, y_2)$ ?

Current

$$p(z_2|y_1, y_2) = N(\hat{z}_2, P_2) = N(0.66174, 0.4532)$$

We apply the prediction equations:

$$\begin{cases} \hat{z}_{3|2} = F_2 \hat{z}_2 = 0.66174 & \text{state estimate prediction} \\ P_{3|2} = F_2 P_2 F_2' + Q_2 = 1.4532 & \text{covariance predicted estimate} \end{cases}$$

and so

$$p(z_3|y_1, y_2) = N(\hat{z}_{3|2}, P_{3|2}) = N(0.66174, 1.4532)$$

and similarly

$$p(z_4|y_1, y_2) = N(\hat{z}_{4|2}, P_{4|2}) = N(0.66174, 2.4532)$$

## Exercise 2

Given the measurement  $y_1 = 0.5$ , compute the estimated level and forecast for ETS(A,N,N) using the Kalman filter. Assume that  $\alpha = 0.3$ ,  $\sigma_w^2 = 0.5$ ,  $\ell_0 = 10$ ,  $\tilde{y}_0 = 0$ ,  $P_0 = \text{diag}(100, 100)$ .

## Solution

We know that:

State equation 1

$$\ell_{k+1} = \ell_k + \alpha \varepsilon_k$$

State equation 2

$$\tilde{y}_{k+1} = \ell_k + \varepsilon_k$$

Measurement equation

$$y_k = \tilde{y}_k + v_k$$

We define

$$\mathbf{z}_k = \begin{bmatrix} \ell_k \\ \tilde{y}_k \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad \mathbf{w}_k = \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \end{bmatrix}$$

$$\begin{cases} \mathbf{z}_{k+1} = F\mathbf{z}_k + \mathbf{w}_k \\ y_k = H\mathbf{z}_k + v_k \end{cases}$$

where

$$\mathbf{w}_k \sim N \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha^2 \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix} \right) = N(\mathbf{0}, Q), \quad v_k \sim N(0, R) \text{ with } R = 0 \quad 11$$

## Prediction step

First we apply the prediction equations:

$$\begin{cases} \hat{\mathbf{z}}_{1|0} = F\hat{\mathbf{z}}_0 & \text{state estimate prediction} \\ P_{1|0} = F_0 P_0 F'_0 + Q_0 & \text{covariance predicted estimate} \end{cases}$$

We know that

$$\hat{\mathbf{z}}_0 = \begin{bmatrix} 10 \\ 0 \end{bmatrix}, \quad P_0 = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$$

$$\begin{cases} \hat{\mathbf{z}}_{1|0} = \begin{bmatrix} 10 \\ 10 \end{bmatrix} & \text{state estimate prediction} \\ P_{1|0} = \begin{bmatrix} 100 & 100 \\ 100 & 100 \end{bmatrix} + \begin{bmatrix} 0.045 & 0 \\ 0 & 0.5 \end{bmatrix} = \begin{bmatrix} 100.045 & 100 \\ 100 & 100.5 \end{bmatrix} & \text{cov. predicted estimate} \end{cases}$$

and so

$$p(\mathbf{z}_1 | y_0) = N(\hat{\mathbf{z}}_{1|0}, P_{1|0})$$

## Update step

We update the prediction  $\hat{z}_{1|0}, P_{1|0}$  with the measurement  $y_1$  to get:

$$p(\mathbf{z}_1 | y_1) = N(\hat{\mathbf{z}}_1, P_1)$$

$$\left\{ \begin{array}{ll} S_1 = H_1 P_{1|0} H_1' + R_1 & \text{innovation covariance} \\ K_1 = P_{1|0} H_1' S_1^{-1} & \text{Kalman gain} \\ \hat{\mathbf{z}}_1 = \hat{\mathbf{z}}_{1|0} + K_1 (y_1 - H_1 \hat{\mathbf{z}}_{1|0}) & \text{state estimate updating} \\ P_1 = P_{1|0} - K_1 H_1 P_{1|0} & \text{covariance updated estimate} \end{array} \right.$$

$$\left\{ \begin{array}{l} S_1 = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 100.045 & 100 \\ 100 & 100.5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 100.5 \\ K_1 = \frac{1}{100.5} \begin{bmatrix} 100.045 & 100 \\ 100 & 100.5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{100}{100.5} \\ \frac{100.5}{100.5} \end{bmatrix} \\ \hat{\mathbf{z}}_1 = \begin{bmatrix} 10 \\ 10 \end{bmatrix} + \begin{bmatrix} \frac{100}{100.5} \\ \frac{100.5}{100.5} \end{bmatrix} (0.5 - 10) = \begin{bmatrix} 0.5472 \\ 0.5 \end{bmatrix} \\ P_1 = \begin{bmatrix} 100.045 & 100 \\ 100 & 100.5 \end{bmatrix} - \begin{bmatrix} 0 & \frac{100}{100.5} \\ 0 & \frac{100.5}{100.5} \end{bmatrix} \cdot \begin{bmatrix} 100.045 & 100 \\ 100 & 100.5 \end{bmatrix} = \begin{bmatrix} 0.5425 & 0 \\ 0 & 0 \end{bmatrix} \end{array} \right.$$



# Coláiste na Tríonóide, Baile Átha Cliath Trinity College Dublin

Ollscoil Átha Cliath | The University of Dublin

## Faculty of Engineering, Mathematics and Science

### School of Computer Science & Statistics Statistics

### Semester 1 Assessment

#### STU33010 : Forecasting

**Mock exam**

**2 hours**

**Dr. Alessio Benavoli:** benavola@tcd.ie

#### Instructions to Candidates:

This is an open book examination. This is an individual assessment and collaboration is **not** permitted. Submission is to be made through the Blackboard page of the module.

- There are 3 questions, for a total of 60 marks (see below rubric). Answer all questions. **You need to give clear derivations of your solution for full credits.**
- You must upload pictures of your hand-written solutions. Please upload all the material as a single zip file using the following name for your file “surname\_studentID.zip” and add your firstname, surname and studentID at the top of all files you upload. Before submitting the zip file via BlackBoard, please **check** you can unzip the file and open its content.

Rubric	Exercise 1	Exercise 2	Exercise 3
Question A	4	30	5
Question B	4		5
Question C	6		6
Total	14	30	16

**Exercise 1**

Consider the following quarterly time-series for the year 2020.

Month	$t_1$	$t_2$	$t_3$	$t_4$
Value	10	20	10	15

Compute the forecast for  $t_5, t_6, t_7$  based on all the data up to  $t_4$  (included).

**Question (a): ETS(N,N)** [4 marks]

**Question (b): ARIMA(2,0,1)** [4 marks]

**Question (c):** Linear regression with trend and Fourier terms up to the degree

2. For this model, you do not need to compute the predictions but only to write the 4 rows of the covariates matrix  $X$ . You also need to compute the covariates  $X^*$  for the prediction of  $y$  at  $t_5, t_6, t_7$ . Is the matrix  $XX^T$  invertible? If no, change both  $X$  and  $X^*$  to guarantee invertibility. [6 marks]

The parameters for ETS(N,N) are  $\alpha = 0.1$  and  $\ell_0 = 5$ . The parameters for ARIMA(2,0,1) are  $\phi_1 = 0.3$ ,  $\phi_2 = \theta_1 = 0.1$ ,  $e_4 = 5$ .

**Exercise 2** A state-space model has the following general form

$$\mathbf{z}_{k+1} = A\mathbf{z}_k + \mathbf{w}_k, \quad y_k = H\mathbf{z}_k + v_k$$

where  $\mathbf{z}_0 \sim N(0, P_0)$ ,  $\mathbf{w}_k = N(0, Q)$ ,  $v_k = N(0, R)$  and  $\mathbf{z}_0, \mathbf{w}_k, v_k$  for  $k = 1, 2, \dots, T$  are all assumed to be independent.

**Question (a):** Given the measurement  $y_1 = 0.5$ , compute the estimated level  $\ell_1$  and  $b_1$ , the mean and covariance of the forecast of the observation  $y_2, y_3$  for ETS(A,A,N) using the Kalman filter.

The parameters of ETS(A,A,N) are:  $\alpha = 0.3$ ,  $\beta = 0.1$ ,  $\sigma_w^2 = 0.5$ ,  $\ell_0 = 10$ ,  $b_0 = 1$ ,  $P_0 = \text{diag}(100, 100, 100)$ . [30 marks]

**Exercise 3**

Answer the following questions.

**Question (a):** When is an ARIMA model stationary? When is an ARIMA model invertible? [2 marks]

**Question (b):** Consider the following time series model  $y_t = y_{t-1} + 0.25\epsilon_{t-1} + \epsilon_t$ , where  $\epsilon_k$  is a Gaussian noise with zero mean and variance  $\sigma^2$ .

Is the model invertible? [1 marks]

Is it stationary? [1 marks]

Is the model obtained after first order differentiation,  $x_t = y_t - y_{t-1}$ , stationary? [1 marks]

**Question (c):** Consider the state space model corresponding to a random walk:

$$z_{t+1} = z_t + w_t$$

$$y_{t+1} = z_t + v_t$$

where  $z_k, y_k, w_k, v_k$  are all scalar quantities. We assume that  $z_0 \sim N(0, p_0 - 1)$ ,  $w_k, v_k \sim N(0, 1)$  and they are independent for all  $k = 1, 2, \dots, T$ . Prove that the Kalman gain

$$K_t = \frac{f_{2t-2} + f_{2t-1}p_0}{f_{2t-1} + f_{2t}p_0} \quad \forall t \geq 1$$

where  $f_k$  is the  $k$ -th Fibonacci number

$$f_0 = 0, \quad f_1 = 1, \quad f_k = f_{k-1} + f_{k-2} \quad \forall k > 1$$

[11 marks]

**Solution 1** The forecast for the ETS(N,N) method is computed as follows:

$$\begin{aligned}\hat{y}_{t+h} &= \ell_t \\ \ell_t &= \alpha y_t + (1 - \alpha)\ell_{t-1}\end{aligned}$$

We need to run the recursion from  $t_1$  to  $t_4$ .

$$\begin{aligned}\ell_1 &= \alpha y_1 + (1 - \alpha)\ell_0 &= 0.1 * 10 + 0.9 * (5) &= 5.5 \\ \ell_2 &= \alpha y_2 + (1 - \alpha)\ell_1 &= 0.1 * 20 + 0.9 * (5.5) &= 6.95 \\ \ell_3 &= \alpha y_3 + (1 - \alpha)\ell_2 &= 0.1 * 10 + 0.9 * (6.95) &= 7.255 \\ \ell_4 &= \alpha y_4 + (1 - \alpha)\ell_3 &= 0.1 * 15 + 0.9 * (7.255) &= 8.0295\end{aligned}$$

and so the forecast is  $\hat{y}_5 = \hat{y}_6 = \hat{y}_7 = \ell_4 = 8.0295$ .

ARIMA(2,0,1) model is equal to

$$(1 - \phi_1 B - \phi_2 B^2)y_t = (1 + \theta_1 B)\epsilon_t$$

and applying the backshift operator gives

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \epsilon_t + \theta_1 \epsilon_{t-1}$$

Hence the forecast equation for the first step is

$$y_{T+1} = \phi_1 y_T + \phi_2 y_{T-1} + \epsilon_{T+1} + \theta_1 \epsilon_T$$

where  $T$  is the length of time-series from  $t_1$  to  $t_4$ . We then put  $\epsilon_{T+1} = 0$  and  $\epsilon_T = e_4$ , that is the residual error for  $t_4$ .

This leads to

$$y_{T+1} = \phi_1 y_T + \phi_2 y_{T-1} + \theta_1 e_T$$

Given that  $\phi_1 = 0.3$ ,  $\phi_2 = \theta_1 = 0.1$  and  $e_T = 3$ , the prediction is

$$\hat{y}_4 = 0.3y_3 + 0.1y_2 + 0.1 * 3 = 0.3 * 10 + 0.1 * 20 + 0.1 * 3 = 5.3$$

For the next step, we keep running the recursion by replacing the future observations with the corresponding predictions and so:

$$\hat{y}_5 = 0.3\hat{y}_4 + 0.1y_3 = 0.3 * 5.3 + 0.1 * 10 = 2.59$$

$$\hat{y}_6 = 0.3\hat{y}_5 + 0.1\hat{y}_4 = 0.3 * 2.59 + 0.1 * 5.3 = 1.307$$

For the linear regression model, we need to construct  $X$ . The linear model is

$$y_t = \beta_0 + \beta_1 t + \beta_2 \cos(2\pi t) + \beta_3 \sin(2\pi t) + \beta_4 \cos(4\pi t) + \beta_5 \sin(5\pi t)$$

Given the time series is monthly, the time step is  $1/4 = 0.25$  and so the matrix  $X$  is

$$X = \begin{bmatrix} 1 & 0.25 & \cos(2\pi 0.25) & \sin(2\pi 0.25) & \cos(4\pi 0.25) & \sin(4\pi 0.25) \\ 1 & 0.5 & \cos(2\pi 0.5) & \sin(2\pi 0.5) & \cos(4\pi 0.5) & \sin(4\pi 0.5) \\ 1 & 0.75 & \cos(2\pi 0.75) & \sin(2\pi 0.75) & \cos(4\pi 0.75) & \sin(2\pi 0.75) \\ 1 & 1 & \cos(2\pi) & \sin(2\pi) & \cos(4\pi) & \sin(4\pi) \end{bmatrix}$$

The matrix  $XX^T$  is not invertible because the last column of  $X$  is always zero and, therefore, linear dependent on the other columns. We must remove it in order to guarantee invertibility for  $XX^T$ :

$$X = \begin{bmatrix} 1 & 0.25 & \cos(2\pi 0.25) & \sin(2\pi 0.25) & \cos(4\pi 0.25) \\ 1 & 0.5 & \cos(2\pi 0.5) & \sin(2\pi 0.5) & \cos(4\pi 0.5) \\ 1 & 0.75 & \cos(2\pi 0.75) & \sin(2\pi 0.75) & \cos(4\pi 0.75) \\ 1 & 1 & \cos(2\pi) & \sin(2\pi) & \cos(4\pi) \end{bmatrix}$$

Therefore, the predictions can be computed using the following covariates

$$X^* = \begin{bmatrix} 1 & 1.25 & \cos(2\pi 1.25) & \sin(2\pi 1.25) & \cos(4\pi 1.25) \\ 1 & 1.5 & \cos(2\pi 1.5) & \sin(2\pi 1.5) & \cos(4\pi 1.5) \\ 1 & 1.75 & \cos(2\pi 1.75) & \sin(2\pi 1.75) & \cos(4\pi 1.75) \end{bmatrix}$$

**Solution 2** The ETS(A,A,N) model is

$$\begin{aligned} y_t &= \ell_{t-1} + b_{t-1} + \varepsilon_t \\ \ell_t &= \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t \\ b_t &= b_{t-1} + \beta \varepsilon_t \end{aligned}$$

We rewrite it in a state space form which is suitable for the KF:

$$\begin{array}{ll} \text{State equation 1} & \ell_{k+1} = \ell_k + b_k + \alpha \varepsilon_k \\ \text{State equation 2} & b_{k+1} = b_k + \beta \varepsilon_k \\ \text{State equation 3} & \tilde{y}_{k+1} = \ell_k + b_k + \varepsilon_k \\ \text{Measurement equation} & y_k = \tilde{y}_k + v_k \end{array}$$

where  $v_k$  is equal to zero, so there is no measurement noise ( $R = 0$  in the KF equations).

We define

$$\mathbf{z}_k = \begin{bmatrix} \ell_k \\ b_k \\ \tilde{y}_k \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{w}_k = \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \\ w_k^{(3)} \end{bmatrix}$$

$$\left\{ \begin{array}{lcl} \mathbf{z}_{k+1} & = & F\mathbf{z}_k + \mathbf{w}_k \\ y_k & = & H\mathbf{z}_k + v_k \end{array} \right.$$

where

$$\mathbf{w}_k \sim N \left( \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha^2 & \alpha\beta & \alpha \\ \alpha\beta & \beta^2 & \beta \\ \alpha & \beta & 1 \end{bmatrix} \sigma_w^2 \right) = N(\mathbf{0}, Q), \quad v_k \sim N(0, R) \text{ with } R = 0$$

The above covariance matrix for  $\mathbf{w}_k$  has been computed by calculating the expected value of the components of  $\mathbf{w}_k$  for instance

$$E[w_k^{(1)} w_k^{(2)}] = E[\alpha \beta \epsilon_k^2] = \alpha \beta \sigma_w^2$$

which corresponds to the element (2,1) (and (1,2)) in the matrix.

First we apply the prediction equations:

$$\begin{cases} \hat{\mathbf{z}}_{1|0} = F\hat{\mathbf{z}}_0 & \text{state estimate prediction} \\ P_{1|0} = FP_0F' + Q & \text{covariance predicted estimate} \end{cases}$$

We know that

$$\hat{\mathbf{z}}_0 = \begin{bmatrix} 10 \\ 1 \\ 0 \end{bmatrix}, \quad P_0 = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$$

$$\begin{cases} \hat{\mathbf{z}}_{1|0} = \begin{bmatrix} 11 \\ 1 \\ 11 \end{bmatrix} \\ P_{1|0} = \begin{bmatrix} 200 & 100 & 200 \\ 100 & 100 & 100 \\ 200 & 100 & 200 \end{bmatrix} + \begin{bmatrix} 0.09 & 0.03 & 0.3 \\ 0.03 & 0.01 & 0.1 \\ 0.3 & 0.1 & 1 \end{bmatrix} 0.5 = \begin{bmatrix} 200.045 & 100.015 & 200.15 \\ 100.015 & 100.005 & 100.05 \\ 200.15 & 100.05 & 200.5 \end{bmatrix} \end{cases}$$

The update step:

$$\begin{cases} S_1 = HP_{1|0}H' + R & \text{innovation covariance} \\ K_1 = P_{1|0}H'S_1^{-1} & \text{Kalman gain} \\ \hat{\mathbf{z}}_1 = \hat{\mathbf{z}}_{1|0} + K_1(y_1 - H\hat{\mathbf{z}}_{1|0}) & \text{state estimate updating} \\ P_1 = P_{1|0} - K_1HP_{1|0} & \text{covariance updated estimate} \end{cases}$$

$$\begin{cases} \hat{z}_1 = \begin{bmatrix} 11 \\ 1 \\ 11 \end{bmatrix} + \frac{1}{200.5} \begin{bmatrix} 200.15 \\ 100.05 \\ 200.5 \end{bmatrix} (0.5 - 11) = \begin{bmatrix} 0.518 \\ -4.23 \\ 0.5 \end{bmatrix} \\ P_1 = \begin{bmatrix} 0.244 & 0.139 & 0 \\ 0.139 & 50 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases}$$

For the prediction, we first apply the prediction equations:

$$\begin{cases} \hat{\mathbf{z}}_{2|1} = F\hat{\mathbf{z}}_1 & \text{state estimate prediction} \\ P_{2|1} = FP_1F' + Q & \text{covariance predicted estimate} \end{cases}$$

and so

$$\begin{cases} \hat{\mathbf{z}}_{2|1} = \begin{bmatrix} -3.72 \\ -4.23 \\ -3.72 \end{bmatrix} \\ P_{2|1} = \begin{bmatrix} 51.64 & 50.23 & 50.75 \\ 50.23 & 50.08 & 50.27 \\ 50.75 & 50.27 & 51.10 \end{bmatrix} \end{cases}$$

To get the forecast for the measurement  $y_2$ , we notice that

$$\begin{cases} \hat{y}_{2|1} = H\hat{\mathbf{z}}_{2|1} & \text{measurement estimate prediction} \\ Q_{2|1} = HP_{2|1}H' + R & \text{covariance predicted estimate} \end{cases}$$

and therefore,  $\hat{y}_{2|1} = -3.72$  and  $Q_{2|1} = 51.10$ . Therefore, the 95% CI is

$$-3.71 \pm 2\sqrt{51.10}$$

To get the forecast for the measurement  $y_3$ , we predict again

$$\begin{cases} \hat{\mathbf{z}}_{3|1} = F\hat{\mathbf{z}}_{2|1} & \text{state estimate prediction} \\ P_{3|1} = FP_{2|1}F' + Q & \text{covariance predicted estimate} \end{cases}$$

and so

$$\hat{\mathbf{z}}_{3|1} = \begin{bmatrix} -7.95 \\ -4.23 \\ -7.95 \end{bmatrix} \quad P_{3|1} = \begin{bmatrix} 201.2472 & 100.3343 & 201.3522 \\ 100.3343 & 50.0898 & 100.3693 \\ 201.3522 & 100.3693 & 201.7022 \end{bmatrix}$$

and computer

$$\begin{cases} \hat{y}_{3|1} = H\hat{\mathbf{z}}_{3|1} & \text{measurement estimate prediction} \\ Q_{3|1} = HP_{3|1}H' + R & \text{covariance predicted estimate} \end{cases}$$

and therefore,  $\hat{y}_{3|1} = -7.95$  and  $Q_{3|1} = 201.7022$ . Therefore, the 95% CI is

$$-7.95 \pm 2\sqrt{201.7022}$$

**Solution 3** Question a. Given the transfer function of the ARIMA model

$$y_t = \frac{n(B)}{d(B)}\epsilon_t$$

where  $n(B), d(B)$  are polynomials of  $B$ . The model is stationary if the roots of  $d(B)$  are outside the unit circle. The model is invertible if the roots of  $n(B)$  are outside the unit circle.

Question b. The model  $y_t = y_{t-1} + 0.25\epsilon_{t-1} + \epsilon_t$  has transfer function

$$\frac{1 + 0.25B}{1 - B}$$

Therefore, the model is not stationary because the root of the denominator is one. The model is invertible because the root of the numerator  $1/0.25$  is outside the unit circle. The first difference model is  $y'_t = 0.25\epsilon_{t-1} + \epsilon_t$ , which is stationary.

Question c. The equations of the Kalman gain are determined by the following recursion.

$$\begin{cases} P_{k|k-1} = FP_{k-1}F' + Q = P_{k-1} + 1 \\ S_k = HP_{k|k-1}H' + R = P_{k|k-1} + 1 \\ K_k = P_{k|k-1}H'S_k^{-1} = \frac{P_{k|k-1}}{1+P_{k|k-1}} \\ P_k = P_{k|k-1} - K_kHP_{k|k-1} = P_{k|k-1} - \frac{P_{k|k-1}}{1+P_{k|k-1}}P_{k|k-1} = K_k \end{cases}$$

We prove the statement by recursion.

Note that

$$p_{1|0} = p_0 - 1 + 1 = p_0$$

and so

$$K_1 = \frac{p_{1|0}}{1 + p_{1|0}} = \frac{p_0}{1 + p_0}$$

Note that

$$p_{2|1} = 1 + \frac{p_0}{1 + p_0} = \frac{1 + 2p_0}{1 + p_0}$$

and so

$$K_2 = \frac{p_{2|1}}{1 + p_{2|1}} = \frac{1 + 2p_0}{2 + 3p_0}$$

Assume it holds up to  $t - 1$

$$K_{t-1} = \frac{f_{2t-4} + f_{2t-3}p_0}{f_{2t-3} + f_{2t-2}p_0}$$

we aim to prove for  $t + 1$ .

$$p_{t|t-1} = 1 + \frac{f_{2t-4} + f_{2t-3}p_0}{f_{2t-3} + f_{2t-2}p_0} = \frac{f_{2t-3} + f_{2t-2}p_0 + f_{2t-4} + f_{2t-3}p_0}{f_{2t-3} + f_{2t-2}p_0} = \frac{f_{2t-2} + f_{2t-1}p_0}{f_{2t-3} + f_{2t-2}p_0}$$

and

$$K_t = \frac{f_{2t-2} + f_{2t-1}p_0}{f_{2t-2} + f_{2t-1}p_0 + f_{2t-3} + f_{2t-2}p_0} = \frac{f_{2t-2} + f_{2t-1}p_0}{f_{2t-1} + f_{2t}p_0}$$

which ends the proof.