Introduction to time series analysis

(fpp) Chapter 1:
Sections 1.1 – 1.7

FREE companion online resource: Forecasting: Principles and Practice by Rob J Hyndman and George Athanasopolous

(https://www.otexts.org/book/fpp3)

I highly recommend this textbook as excellent supplementary reading that delves a bit deeper into topics that we discuss here (Parts of chapters 1, 2, 3, 4, 5, 8,9)

Whenever you see '*(fpp)*' in the slides, it is referring to the Forecasting: Principles and Practice online textbook

Introduction to Time Series Analysis

Time Series vs. Cross-Sectional Data

- **Cross-sectional** data is data observed or measured at one point in time

  e.g. Final STA2020 marks

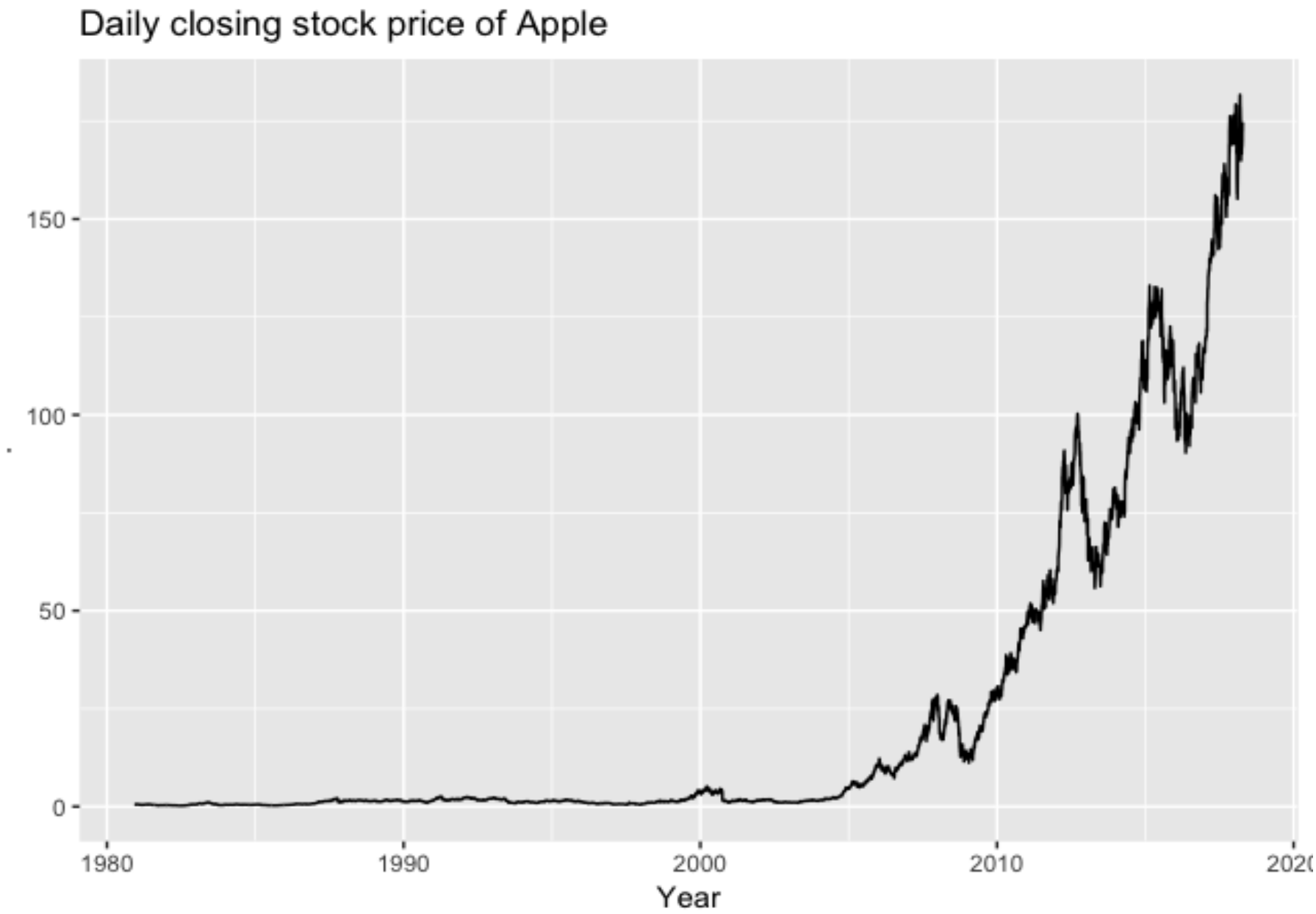
- Any variable that is regularly measured over time in sequential order

(e.g. hourly, daily, weekly, monthly, quarterly, or yearly) is called a ***time series***

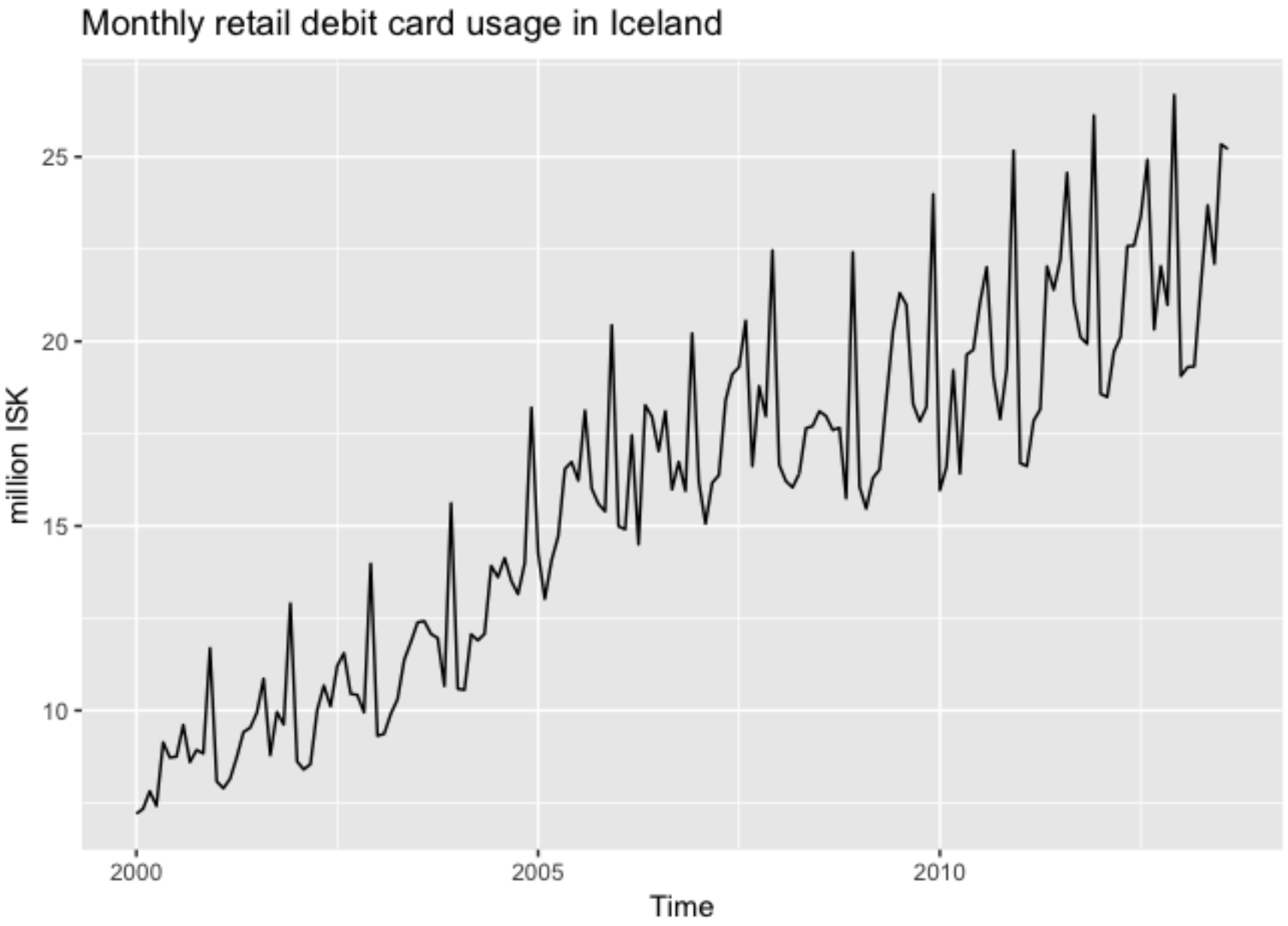
Definition: ***A time series is a sequence of observations collected at regular equally spaced intervals over a period of time***

- Time series data are extremely common. They arise in virtually every application field, such as e.g.:
- Business
  - Sales figures, production numbers, customer frequencies
- Economics
  - Stock prices, exchange rates, interest rates
- Official Statistics
  - Census data, personal expenditures, road casualties

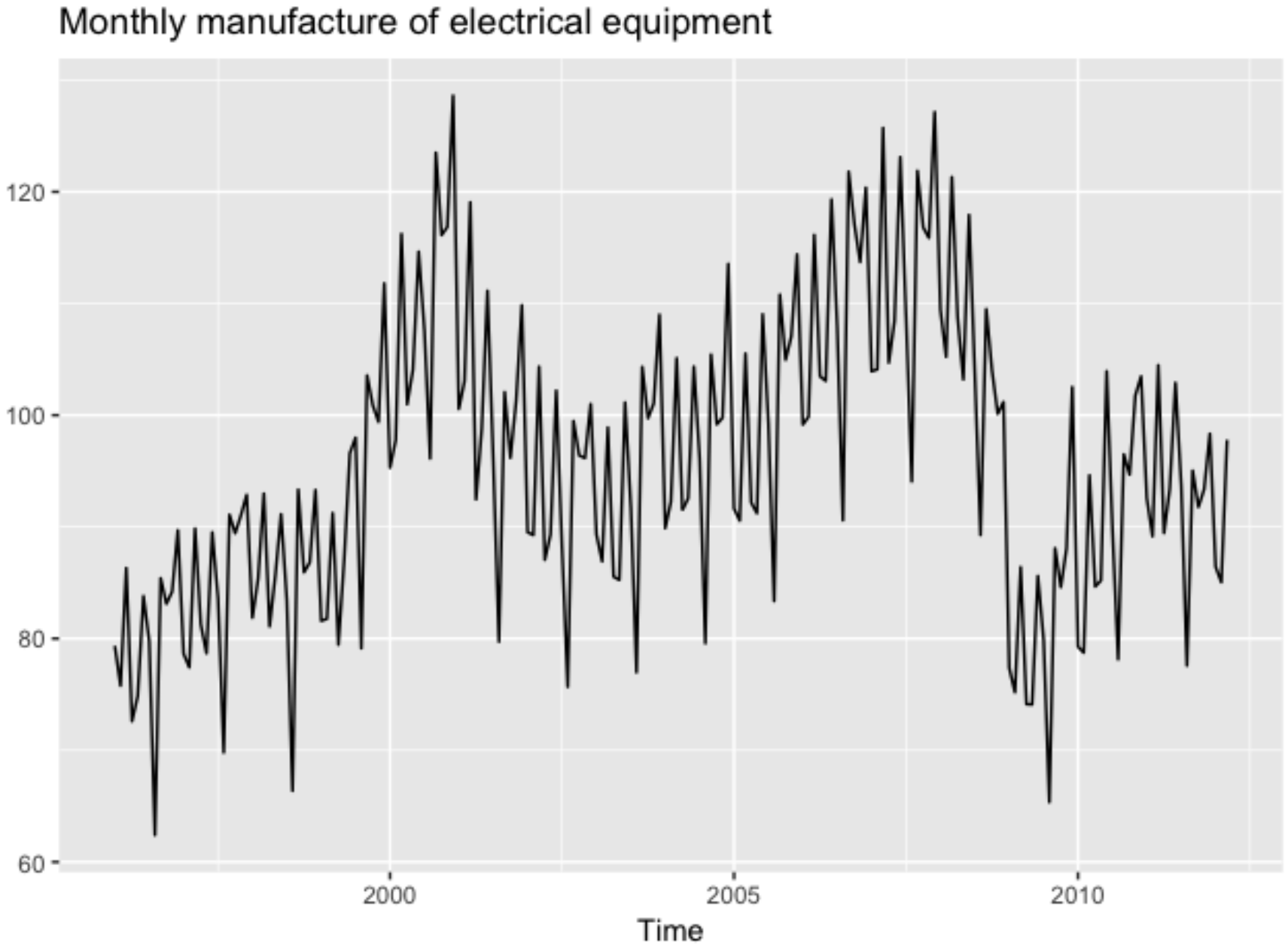
Introduction to Time Series Analysis



Introduction to Time Series Analysis



Introduction to Time Series Analysis



Introduction to Time Series Analysis

Why Time Series Analysis?

- Economic conditions vary over time. The success of many businesses is dependent on being able to foresee future events and plan accordingly.

- Time series analysis is a collection of statistical techniques that attempt to *isolate and quantify the influence of these events and changes in conditions* in order to build a model that utilizes this information to forecast future values of the time series

- Standard inferential techniques which assume independence of observations (e.g. regression) do not work well when *data is collected at regular or equally spaced time intervals* because the observations are likely to be dependent. When this dependence occurs between observations g time periods apart, it is called autocorrelation at lag g.

- So we cannot assume that the data constitute a random sample. Much of the methodology in time series analysis is aimed at explaining this correlation using appropriate statistical models

Basic Assumption Underlying Time Series Forecasting

The factors that influenced patterns of activity in the past and present will continue to do so in more or less the same manner in the future (i.e. we assume that past patterns will continue into the future).

Thus the overall purpose of time series analysis is to identify and isolate these influencing factors from the past in order to better understand the process underlying the time series, for predictive purposes.

We can conduct a time series analysis to:

a) Develop a better understanding of the pattern of behavior present and the factors/components that have given rise to that pattern

b) Develop a model that captures as much relevant information/patterns in the time series as possible to forecast future values of the time series

c) … and other uses which we don't consider in this course

Two fundamental time series concepts:

Autocorrelation and Stationarity

1. Autocorrelation

- Suppose we have a data $X_1, X_2 \ \ldots X_T$ where T is the length (number of observations) in the time series. The most basic assumption in inferential statistics is that X_i are independent, i.e. we have a random sample. The independence is a nice property, since using it we can derive a lot of useful results.

- The problem is that frequently this property does not hold for time series data since the observations are all from the same variable and recorded at equally spaced intervals of time. This often results in the observations of a time series being correlated with one another. This is referred to as autocorrelation or serial correlation.

- The assumption of independence of observations in a multiple linear regression model commonly fails when the sample data have been collected over time and the regression model fails to effectively capture any trends. Thus, standard regression models are often not appropriate to model time series due to the autocorrelation in the data.

- **The majority of time series are dependent** i.e. they exhibit significant autocorrelation at some lag *g*, where 'lag' refers to the number of time periods between observations at which we measure the autocorrelation. ***This autocorrelation represents useful information about patterns in the data that a time series model can use to make forecasts.***

2. Stationarity

A time series is said to be stationary if its statistical properties are **constant** over time i.e. they are independent of time. This implies that:

1) The time series has a constant *mean*, ***AND***
2) The *variability* of the time series is constant over time

"A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary — the trend and seasonality will affect the value of the time series at different times…

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

Most time series we encounter are non-stationary and we often need to transform them so that they exhibit stationarity (see later in course). These transformations enable us to consider what other information (i.e. autocorrelation) exists in the data after we have removed the effect of a trend or seasonality and/or changing variance.

Time Series Graphics

(fpp) Chapter 2: Sections 2.1–2.3, 2.8–2.9

Time series plot

1) The first step in a time series analysis is to plot the data and observe any patterns that have occurred over time using a time series plot/line graph

--- a time series plot is a line graph of the observed data y_t against time t.

2) The time series plot enables us to detect and to describe patterns/factors/components (of the past behavior) of the series. The successive changes in values are comparable because they all relate to a common time interval between observations.

3) The identified components help in finding a suitable statistical model to describe the data, which enables us to forecast future values of the time series, on the *basis that past patterns will continue into the future*.

Components of a Non-Stationary Time Series

4 components:

1) **TREND**
2) **CYCLICAL** variation
3) **SEASONAL** variation and the
4) **IRREGULAR/RANDOM** variation

You need to be able to both describe each of these components and identify them in a time series plot.

Components of a Non-Stationary Time Series

1) Trend (also known as *Secular* Trend) (**T**)

- The long-term tendency of a time series (pattern observed may move steadily in an upward or downward direction, or stay the same over time)

- It is usually the result of long-term factors e.g. population increases/decreases, consumer preferences etc.

- Can be linear or nonlinear over time.

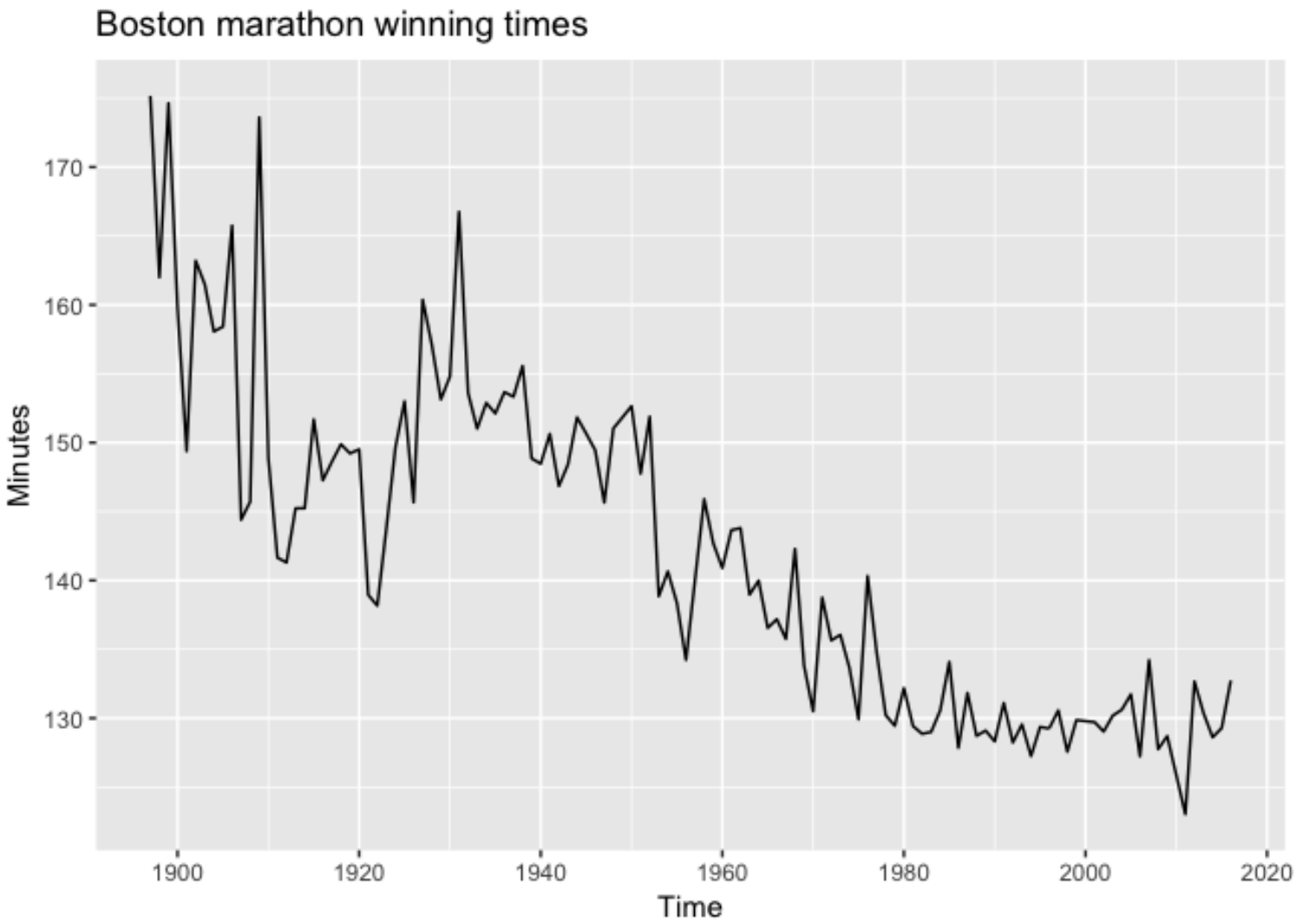
- Duration of *the trend* over the entire window of observation is much longer than a 1 time period.

- We assume it can be predicted into the future.

Components of a Non-Stationary Time Series

Overall, a linear trend is evident here since the overall general upwards movement is linear, with lots of random variation.

Components of a Non-Stationary Time Series



Decreasing, non-linear trend and random variation evident in this time series.

Always **be specific** when describing the trend – its direction (increasing/decreasing) and whether it is linear or non-linear

Components of a Non-Stationary Time Series --- The CYCLICAL variation

2) Cyclical variation (**C**)

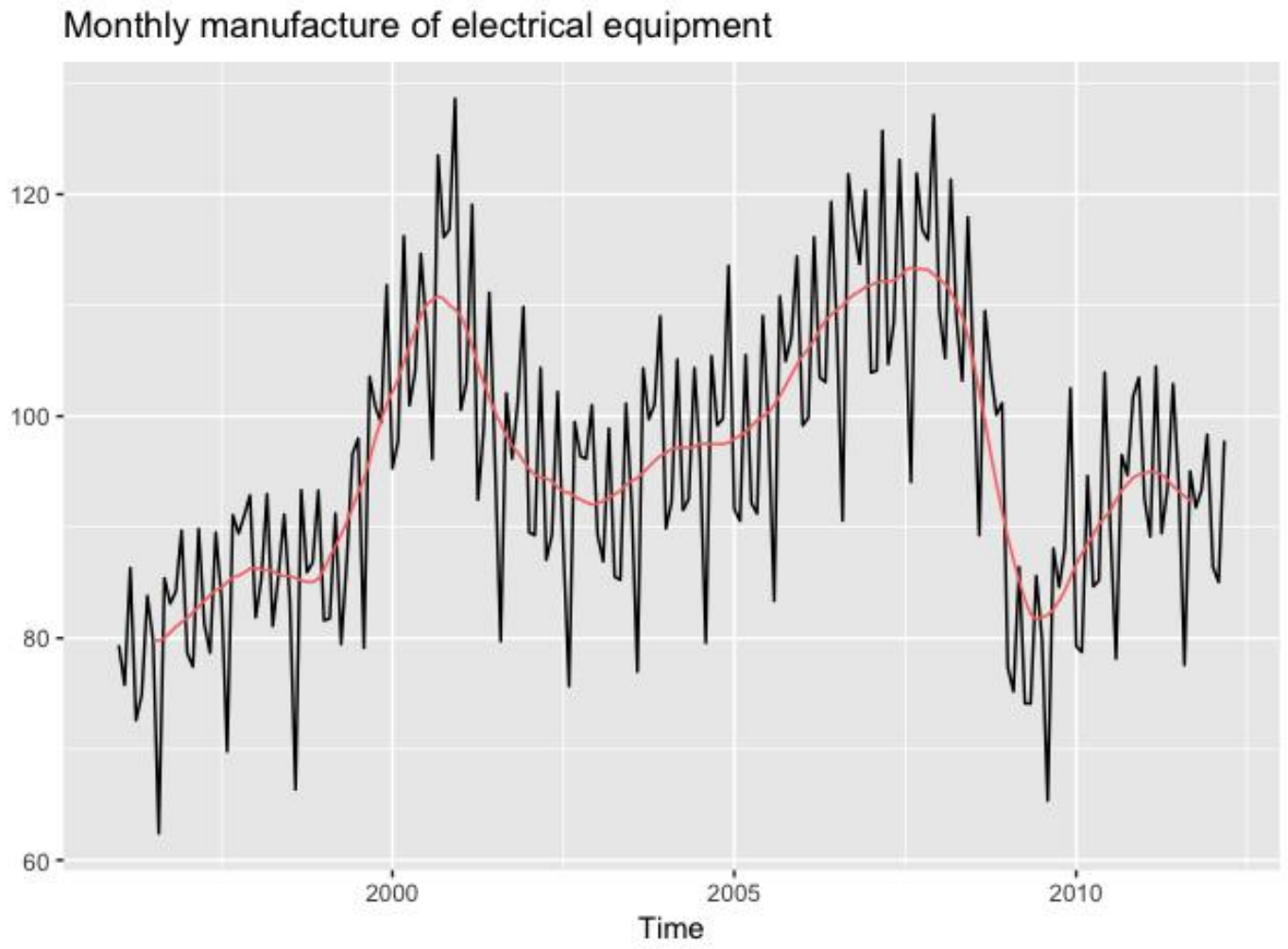
- Irregular **long-term** wavelike movements through a time series (due to extended periods of prosperity/booms followed by extended periods of recession, depression/troughs and recovery in the economy)

- Duration of a cycle is not fixed (i.e. successive cycles are usually not the same length). A cycle usually lasts at least 2 years or more (a full cycle can last from 2 to 10 years).

- Generally, the length of the cyclical variation is **more than a year**, and **the duration is not fixed. Cyclicality does not repeat continuously throughout a time series. If it is present in a time series, there are usually only a few cycles - sometimes only one.**

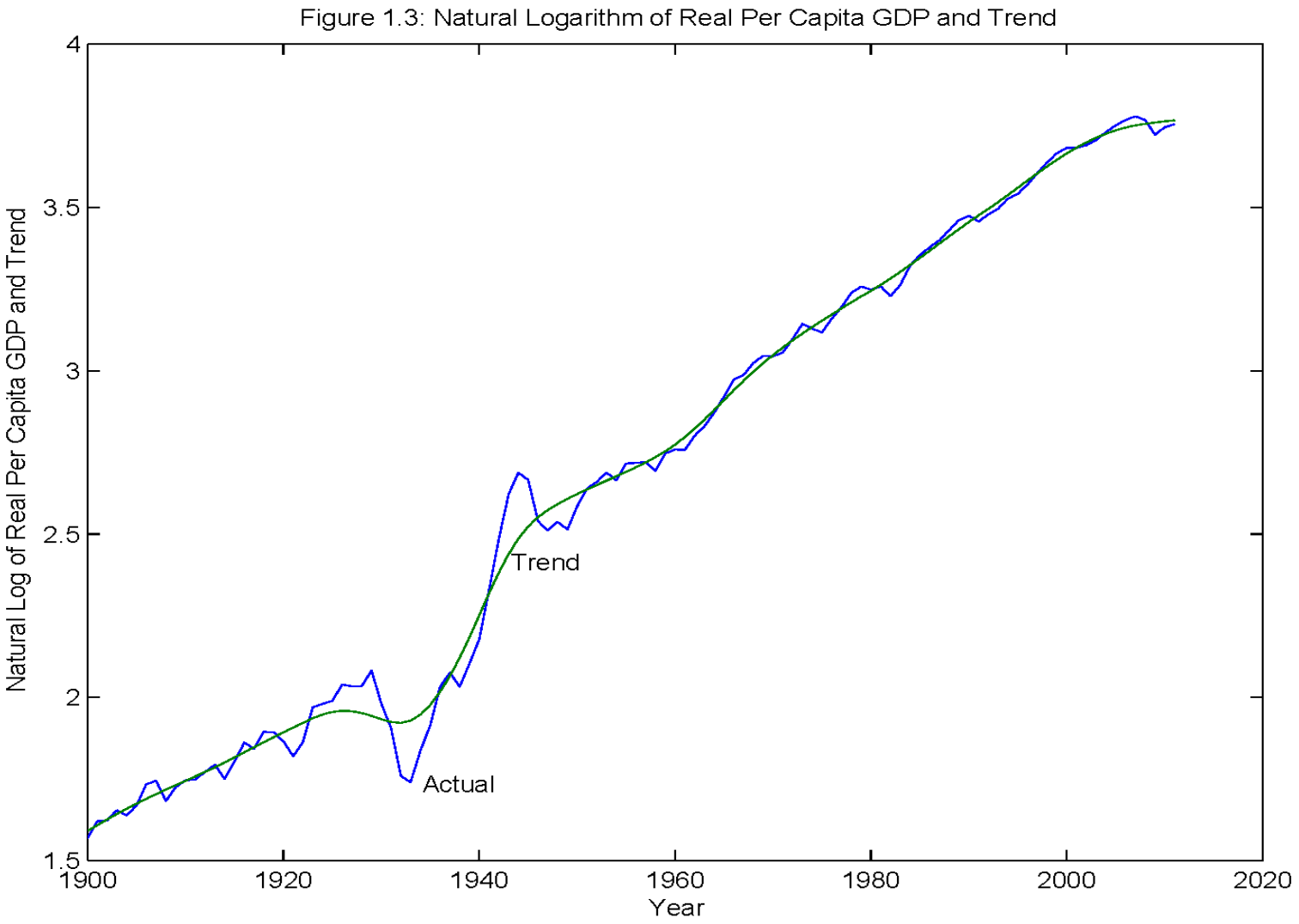
- We assume it can be predicted because it appears to have a repetitive pattern.

Components of a Non-Stationary Time Series



Note: There are two cycles here – from approximately 1998 to 2003 and then from approximately 2006 to 2011. The cycles/waves are not necessarily of same length and are longer than a year. Also, they do not repeat throughout the entire time series.

Components of a Non-Stationary Time Series



Again, the here we have just one cycle and then no more throughout the time series.

Components of a Non-Stationary Time Series --- The SEASONAL variation

3) Seasonal variation (**S**)

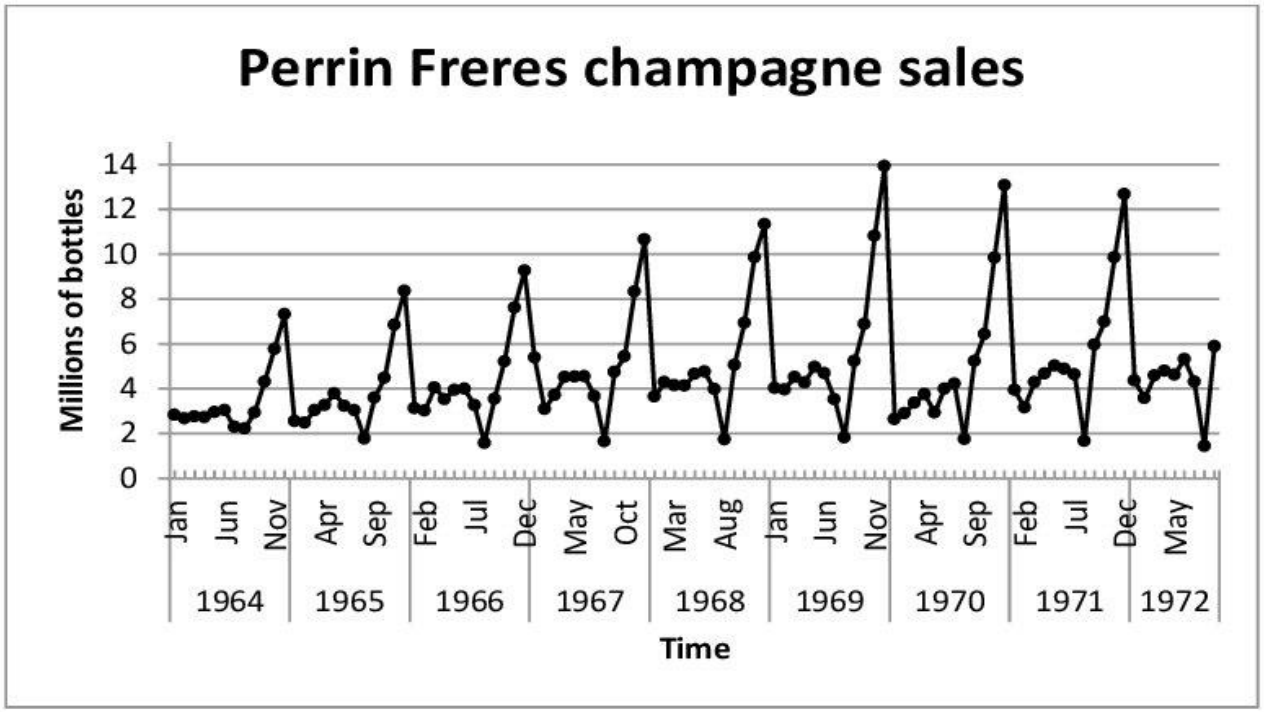
- Regular *short-term* repetitive wavelike movements through a time series, often when data is recorded hourly, daily, weekly, monthly or quarterly. ***It repeats itself throughout the time series.***

- **The variations are often short term, a year or less in length**, and usually repeat themselves during the 'same' calendar periods *e.g. daily traffic volume in Cape Town will display within-the-day "seasonal" behaviour, with peak levels occurring between 8-9am and 4-5:30pm, and moderate flow during the rest of the day.* ***The duration of the seasonal variation is typically fixed (i.e. the same) e.g. the same pattern repeats itself each year or each week etc.***

- Can be predicted because the pattern is repeated many times. E.g.
  - 24 hours in a day (hourly data)
  - 7 days in a week (weekly data)
  - 12 months in a year (monthly data)
  - 4 seasons in a year (quarterly data)

Components of a Non-Stationary Time Series

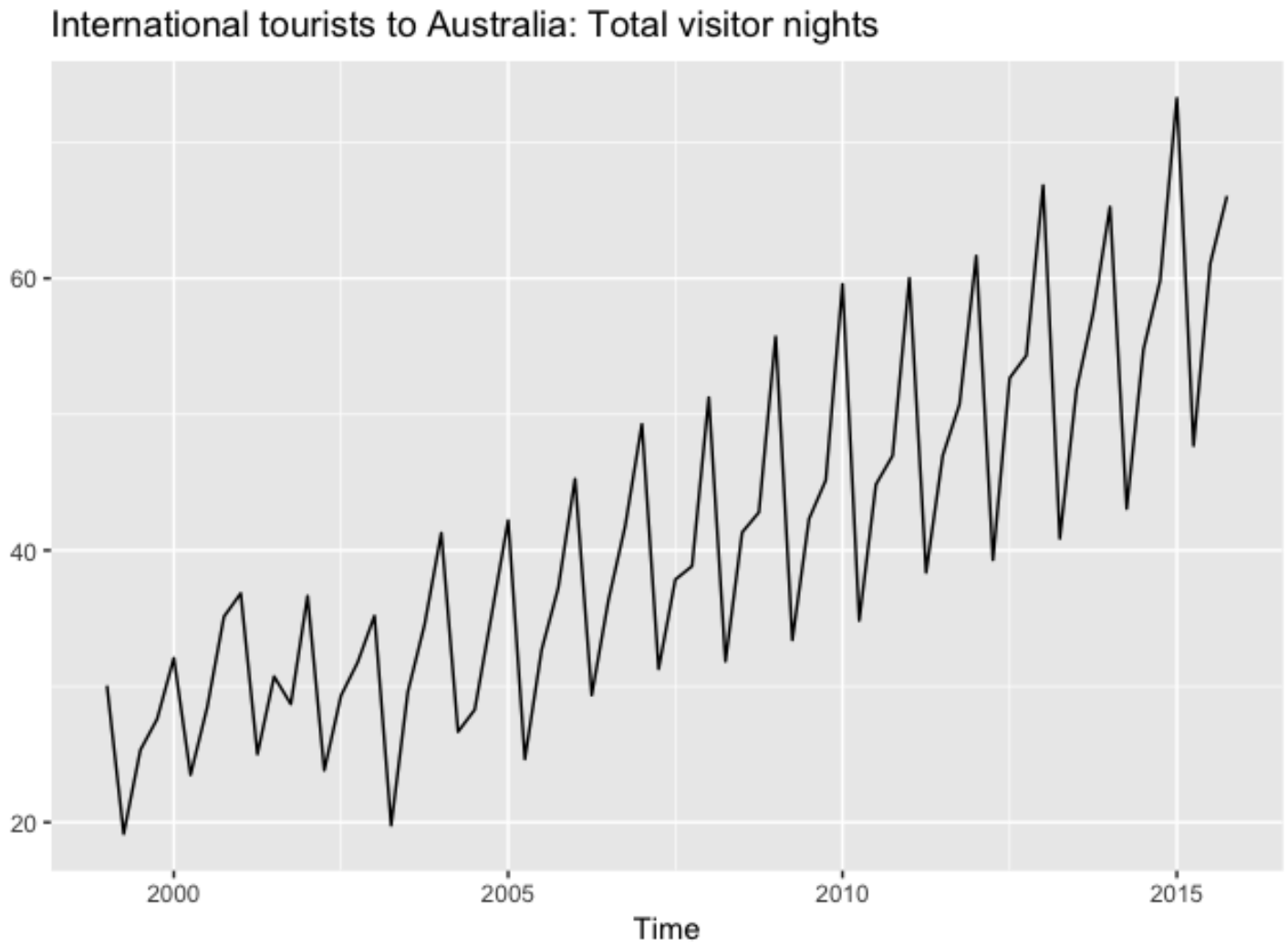
Perrin Freres champagne sales



There are 12 observations each year, so this is showing monthly champagne sales

The same pattern repeats itself every 12 months (i.e. one year). It has the same length each time and repeats itself throughout the time series. This is characteristic of seasonality.

Components of a Non-Stationary Time Series



There are 4 observations each year, so this is quarterly data. The same pattern repeats itself every 4 quarters, and this pattern is the same length each time and repeats throughout the time series. This is seasonality.

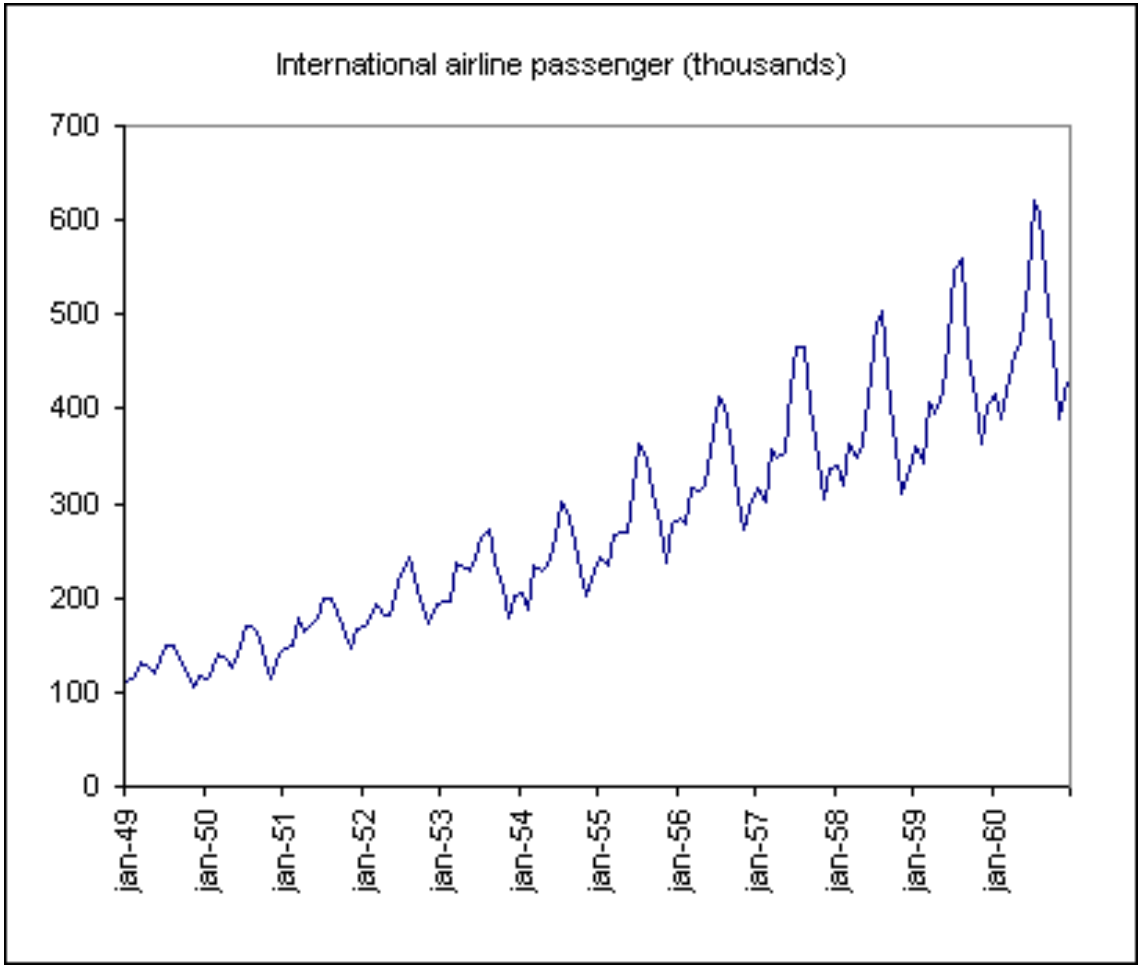
Components of a Non-Stationary Time Series -- - The IRREGULAR/RANDOM variation

4) Irregular/Random variation (**R**)

- Random variations in data due to the combined effects of all unforeseen events such as war, strikes, natural disasters, power cuts, etc.
- Duration is short, and non repeating, emerges as variance
- There is no defined statistical technique for estimating random fluctuations in a time series i.e. they cannot be predicted --- **NB**
- Tends to hide the existence of the other predictable components of a time series, especially when it is large relative to the pattern in the data

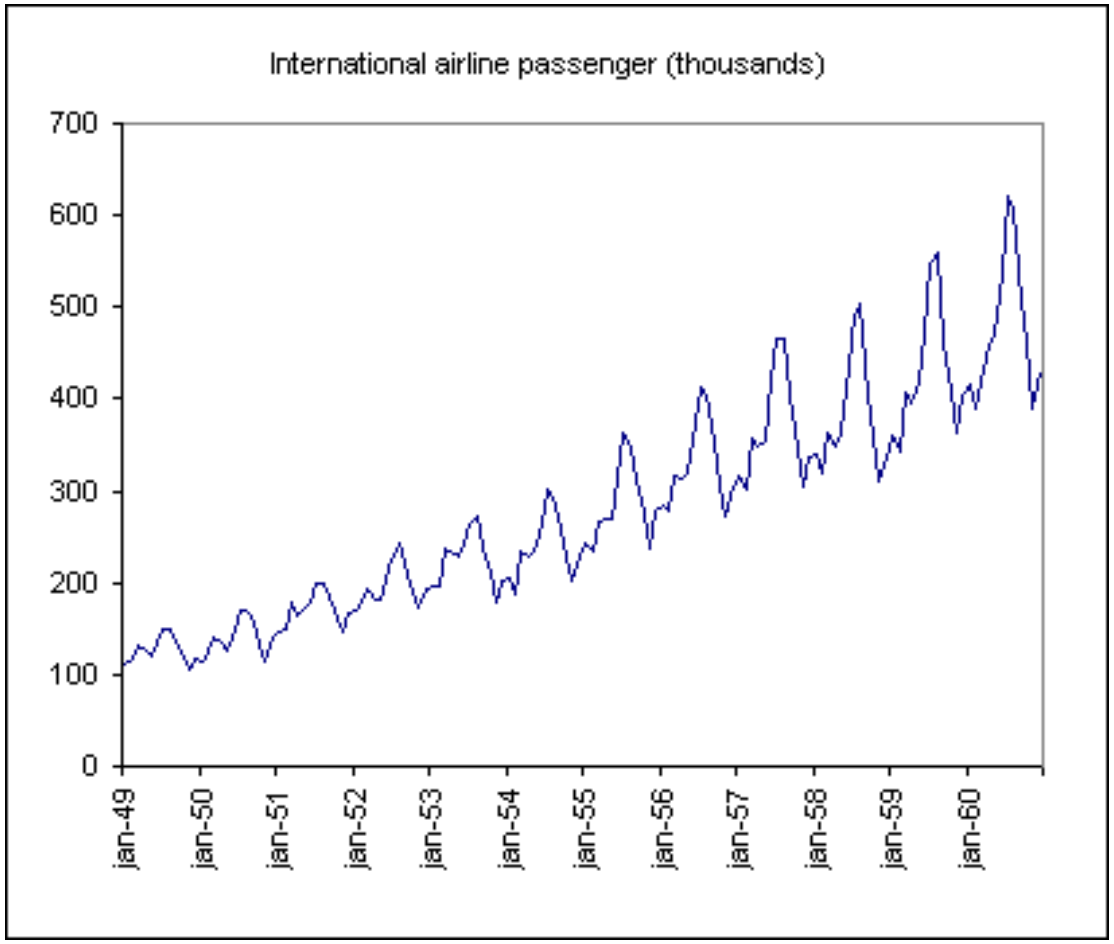
Note: **All time series include random variation**. In addition, a time series may include none, one, two or all the other three components – trend, cyclical and seasonal variation

Components of a Non-Stationary Time Series ---Examples



Time Series Components?

Components of a Non-Stationary Time Series ---Examples



Time Series Components?

Upward non-linear TREND,

RANDOM VARIATION and

SEASONALITY lasting 12 months at a time.

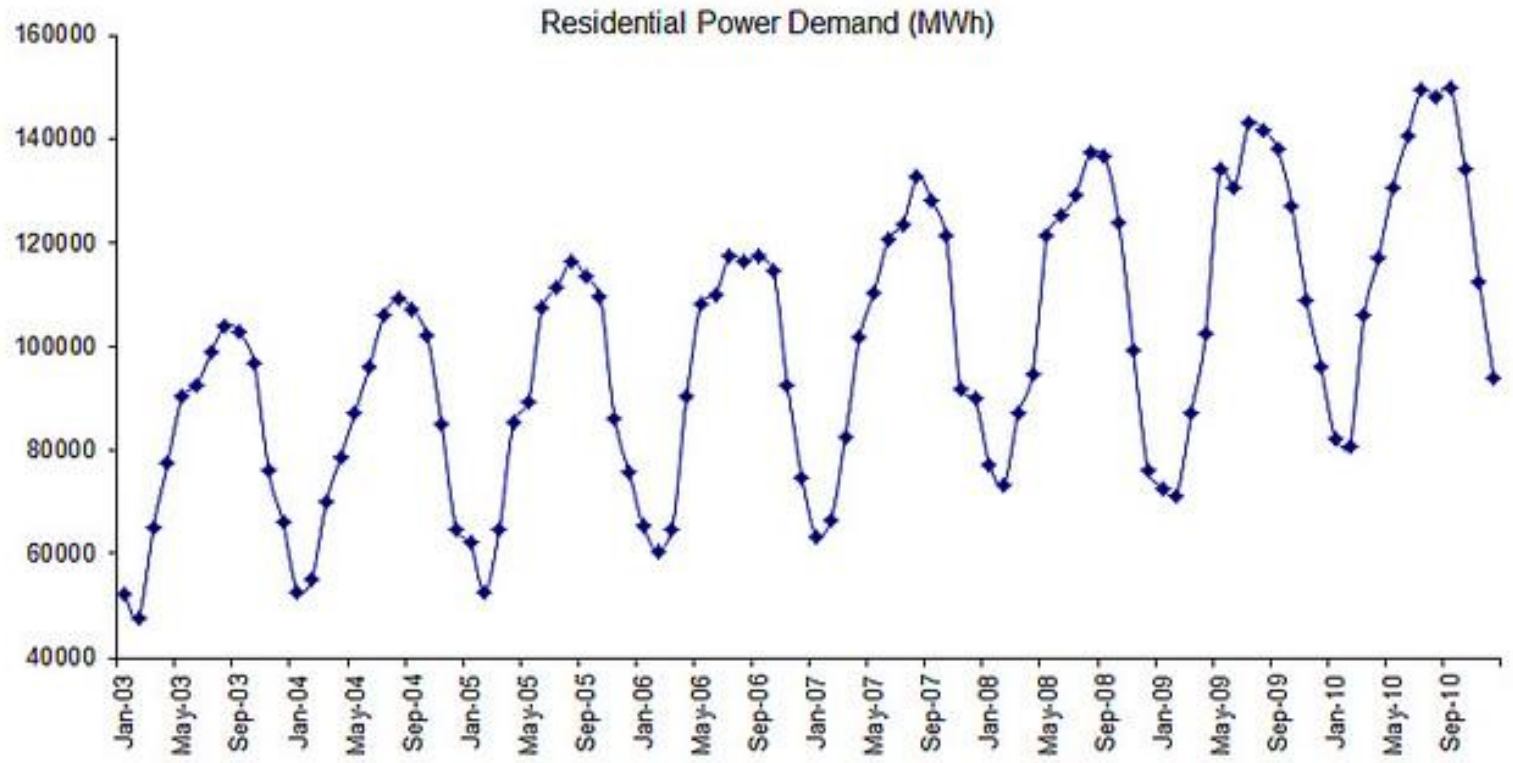
Components of a Non-Stationary Time Series ---Examples

Time Series Components?

Components of a Non-Stationary Time Series ---Examples

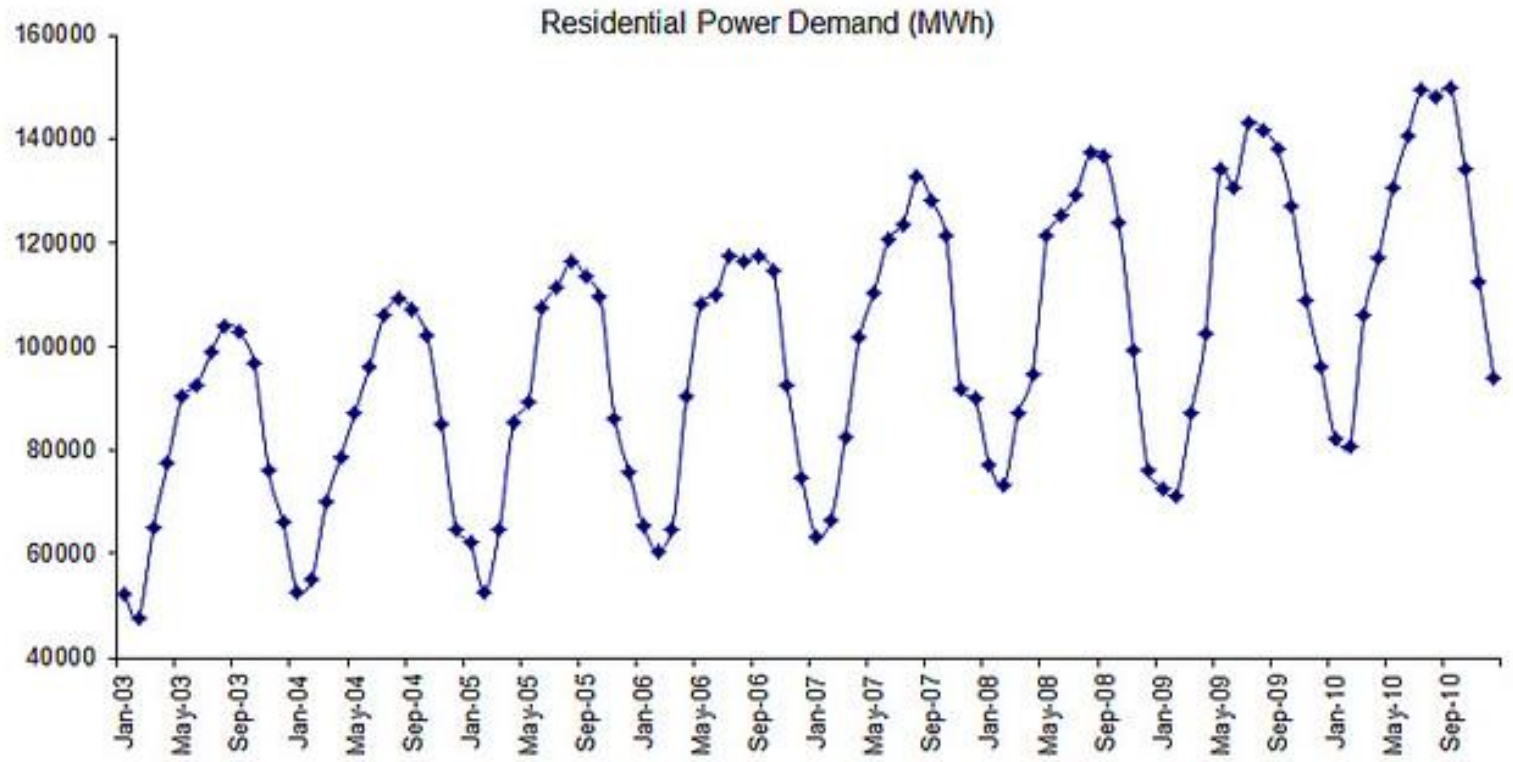
Time Series Components? Upward linear TREND, RANDOM VARIATION and CYCLICAL VARIATION

Components of a Non-Stationary Time Series ---Examples



Time Series Components?

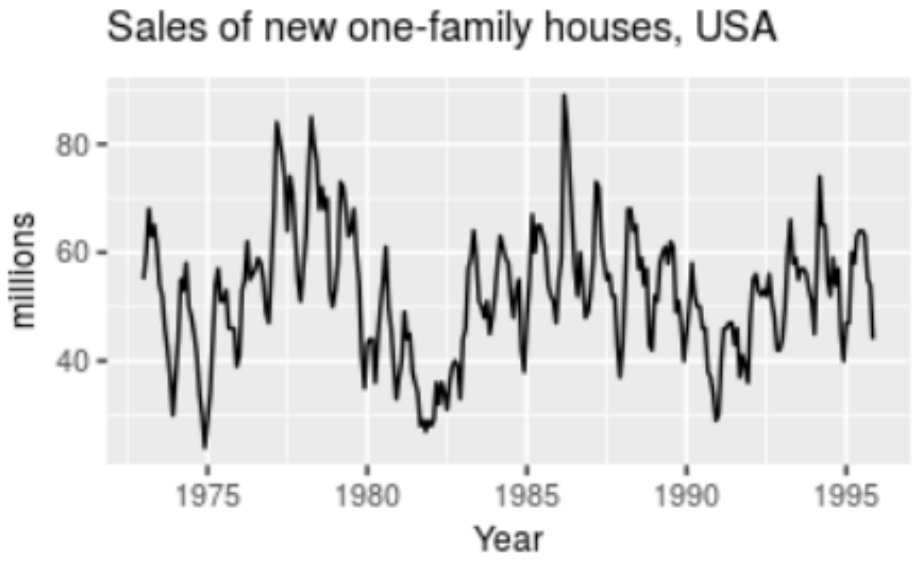
Components of a Non-Stationary Time Series ---Examples

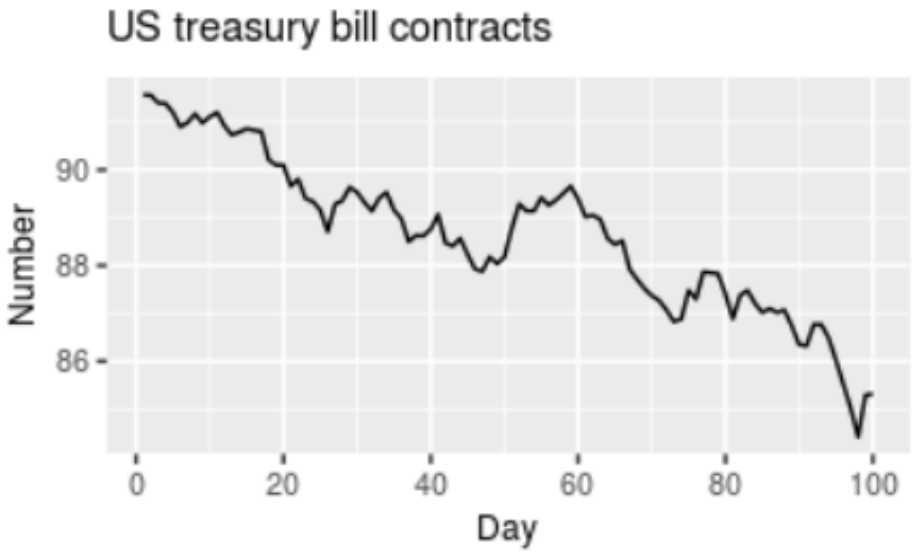


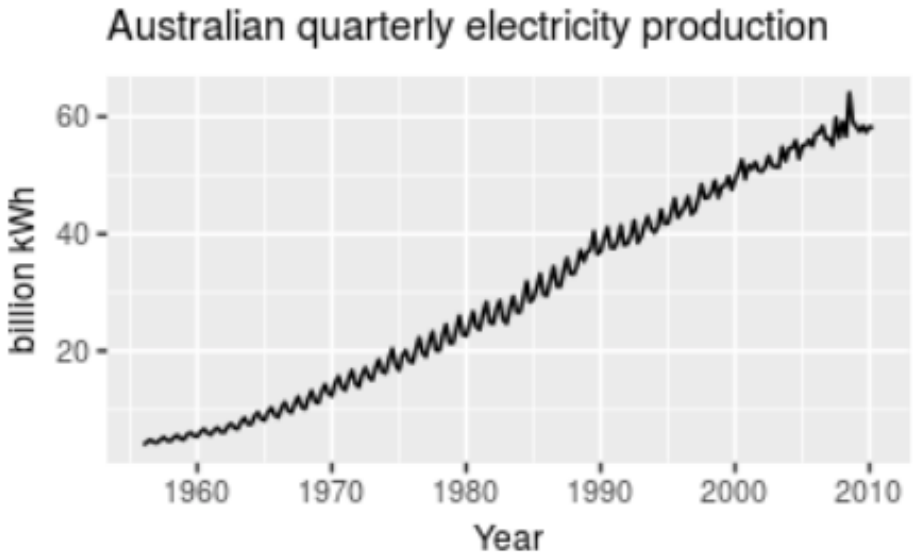
Time Series Components?

Upward linear TREND, RANDOM VARIATION and SEASONALITY lasting 12 months at a time.

Components of a Non-Stationary Time Series --







***(fpp)*: Time Series Components? Go and read the fpp online textbook to find out!**

General time series model & moving average smoothing

Gaining a better understanding of the patterns in the historical time series data

Time Series Model

- There are many models that are available for forecasting a time series.
- The explanatory variable, time *(t)*, is often recoded from its real world value to make the equations easier to interpret.
  - For example $t = 1$ for the first time period (year, quarter, week, day etc.), and all successive time periods are assigned consecutively in increasing integer codes (2, 3, 4 , … etc.)

- Note that in a time series model we are modelling the dependent variable on past values of itself. We are not incorporating other explanatory variables in the model, we are only seeking to understand the past pattern in the data and use the information we glean to forecast future values

- Why? First, the process generating the data may not be understood, and it may be difficult to measure the relationships that give rise to its behavior. Second, in an explanatory model you need to know or forecast the various predictors in order to be able to forecast the variable of interest, and this may be too difficult. Third, the main concern may be only to predict what will happen, not to know why it happens. Finally, the time series model may give more accurate forecasts than an explanatory model.

Time Series Model

- The simplest assumption about the relationship between the components in a time series is that they are additive and independent of each other.

- We write the **additive model** as: $Y_t = T_t + C_t + S_t + R_t$

where: t is the time period we are interested in (t starts at 1)

y_t is the observed value of the time series at time period t

T_t is the value of the trend component at time t

C_t is the value of the cyclical component at time t

S_t is the value of the seasonal component at time t

R_t is the value of the random component at time t

- Alternatively, we assume that the four components of a time series are not necessarily independent and they can affect one another. This is captured by the **multiplicative model** as: $Y_t = T_t \times C_t \times S_t \times R_t$

- That is, the observed value (Y_t) is a product of the four time series components

Time Series Model

The multiplicative model is the most preferred forecasting model, because:

- It can be made additive by taking the logarithm of the series
- If we consider any component (e.g. T), then the other components (C, S, and R) can be interpreted as *indexes* relative to that component

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

Note: **For STA2020, we *only consider the multiplicative model*, AND we assume that the CYCLICAL component is negligible i.e. . Hence,** $C_t = 1$ *and* $Y_t = T_t \times S_t \times R_t$

Time Series Model

- What tools are available to us if our main goal is *not forecasting future values*, but *understanding the patterns in the current time series* i.e. the trend and seasonal components?

- It is often impossible or very difficult to identify components of a time series by simply graphing the series over time, because of too much random variation.

- If there is no obvious pattern, then **moving average smoothing** can be used to smooth the series. This is done to *dampen down the error terms* to provide an overall long-term impression of the pattern of predictable movement in the data. *Removing random variation can help us to better estimate/understand the underlying components, thus enabling models to make more accurate forecasts.*

- If on the other hand there is an obvious pattern, then a variety of forecasting methods can be considered depending on the nature of the data (i.e. whether it is monthly, quarterly or annual) (see later)

Moving Averages

Since observations that are close together in time are likely to have similar values, the moving average method "*smoothes*" the data (**removes some of the random variation to get a clearer understanding of the effect of the components present in the time series**) by "moving" the arithmetic mean over a window of values through the time series.

i.e. We calculate the mean of the first "window" of observations (e.g. 3 observations) and then "move" the window to exclude the first observation and include the next three observations (i.e. the 2nd, 3rd and 4th observations)

- Thus to apply the moving average technique to a time series, we compute the mean of a series of observations taken over *k* consecutive time periods. This is done over the entire time series

- We use notation MA(*k*), to denote a moving average given by a window of length of *k* consecutive time periods

Moving Averages

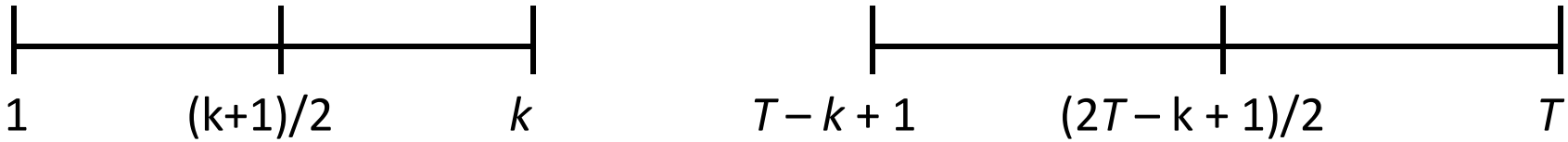
To compute MA(*k*):

- The first moving average of this sequence is obtained by calculating the average of the first consecutive *k* values ($y_1, y_2, \ldots, y_k$)
- The second moving average is obtained by calculating the average of the next batch of consecutive *k* values excluding the first value ($y_2, y_3, \ldots, y_{k+1}$)
- The third moving average is obtained by calculating the average of the next batch of consecutive *k* values excluding the first and second values ($y_3, y_4, \ldots, y_{k+2}$)
- The process continues until the average of the last batch of *k* consecutive values is computed. **There will be T - *k*+1 smoothed values**

Moving Averages

Note: When plotting moving averages on a chart, each of the computed values is plotted against the middle period of the sequence of periods used to compute it.

- That is, each value is associated with the mid-point of the time window as illustrated below:



For Example:

Year	t	Revenue	MA(3)
2003	1	**34**	
2004	2	**12**	**37.667**
2005	3	**67**	
2006	4	87	
2007	5	22	
2008	6	66	
2009	7	77	
2010	8	90	
2011	9	34	
2012	10	22	

Thus the MA(3) for 2004 is calculated by taking the average the values within 1 period either side of 2004

Moving Averages

Year	t	Revenue	MA(3)
2003	1	34	
2004	2	**12**	37.667
2005	3	**67**	**55.333**
2006	4	**87**	
2007	5	22	
2008	6	66	
2009	7	77	
2010	8	90	
2011	9	34	
2012	10	22	

Moving Averages

Year	t	Revenue	MA(3)
2003	1	34	
2004	2	12	37.667
2005	3	67	55.333
2006	4	87	58.667
2007	5	22	58.333
2008	6	66	55.000
2009	7	77	77.667
2010	8	**90**	67.000
2011	9	**34**	**48.667**
2012	10	**22**	

$T - k + 1$

$= 10 - 3 + 1$

$= 8$

MA(3) values

Year	t	Revenue	MA(5)
2003	1	**34**	
2004	2	**12**	
2005	3	**67**	**44.400**
2006	4	**87**	
2007	5	**22**	
2008	6	66	
2009	7	77	
2010	8	90	
2011	9	34	
2012	10	22	

Year	t	Revenue	MA(5)
2003	1	34	
2004	2	**12**	
2005	3	**67**	44.400
2006	4	**87**	**50.800**
2007	5	**22**	
2008	6	**66**	
2009	7	77	
2010	8	90	
2011	9	34	
2012	10	22	

Year	t	Revenue	MA(5)
2003	1	34	
2004	2	12	
2005	3	67	44.400
2006	4	87	50.800
2007	5	22	63.800
2008	6	**66**	68.400
2009	7	**77**	57.800
2010	8	**90**	**57.800**
2011	9	**34**	
2012	10	**22**	

$T - k + 1$

$= 10 - 5 + 1$

$= 6$

MA(5) values

Year	t	Revenue	MA(4)
2003	1	**34**	
2004	2	**12**	**50.000**
2005	3	**67**	
2006	4	**87**	
2007	5	22	
2008	6	66	
2009	7	77	
2010	8	90	
2011	9	34	
2012	10	22	

Year	t	Revenue	MA(4)
2003	1	34	
2004	2	**12**	50.000
2005	3	**67**	**47.000**
2006	4	**87**	
2007	5	**22**	
2008	6	66	
2009	7	77	
2010	8	90	
2011	9	34	
2012	10	22	

Moving Averages

Note:

- We want to assign each moving average to an instant at which an observation was made.

- This alignment of time points for the observed and smoothed values is important for plotting the smoothed series

- When k is even, the moving average does not correspond to any of the time periods in the original series

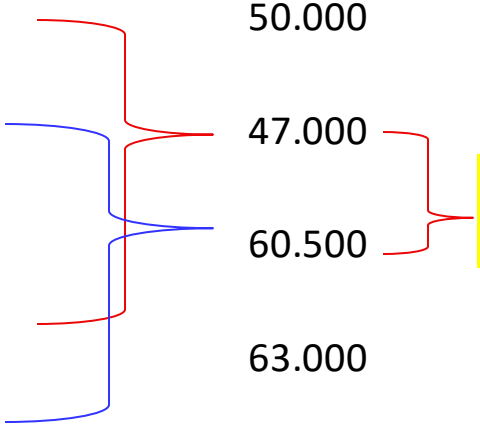
- **Hence when *k* is even it becomes necessary to average out 2 adjacent MA(*k*) values (i.e. we *centre* the moving averages)**

Moving Averages

Year	t	Revenue	MA(4)	CMA(4)
2003	1	34		
2004	2	12	50.000	
2005	3	67	47.000	48.500
2006	4	87		
2007	5	22		
2008	6	66		
2009	7	77		
2010	8	90		
2011	9	34		
2012	10	22		

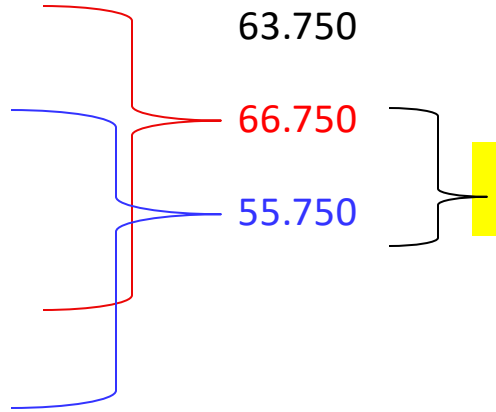
Moving Averages

Year	t	Revenue	MA(4)	CMA(4)
2003	1	34		
2004	2	12	50.000	
2005	3	67	47.000	48.500
2006	4	87	60.500	53.750
2007	5	22	63.000	
2008	6	66	63.750	
2009	7	77	66.750	
2010	8	90	55.750	
2011	9	34		
2012	10	22		



Moving Averages

Year	t	Revenue	MA(4)	CMA(4)
2003	1	34		
2004	2	12	50.000	
2005	3	67	47.000	48.500
2006	4	87	60.500	53.750
2007	5	22	63.000	61.750
2008	6	66	63.750	63.375
2009	7	77	66.750	65.250
2010	8	90	55.750	**61.250**
2011	9	34		
2012	10	22		



Note: There are 7 MA(4) values, but only 6 CMA(4) values because of averaging

Moving Averages

Year	t	Revenue	MA(3)	MA(5)	CMA(4)
2003	1	34			
2004	2	12	37.667		
2005	3	67	55.333	44.400	48.500
2006	4	87	58.667	50.800	53.750
2007	5	22	58.333	63.800	61.750
2008	6	66	55.000	68.400	63.375
2009	7	77	77.667	57.800	65.250
2010	8	90	67.000	57.800	61.250
2011	9	34	48.667		
2012	10	22			

Moving Averages --- Example 1

The following data represent total revenues (in millions of Rands) by a car rental agency in South Africa over the 11 year period 1992 to 2002.

4.0	5.0	7.0	6.0	8.0	9.0	5.0	2.0	3.5
5.5	6.5							

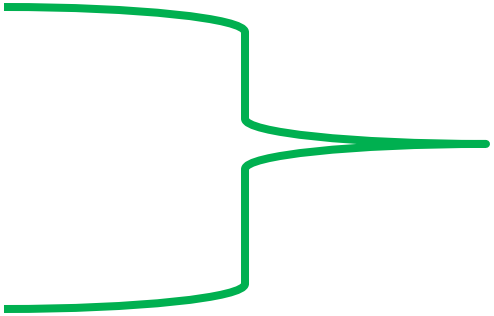
Compute the 3-year and the 4-year moving averages.

Moving Averages --- Example 1

Year	t	Data (Y_t)	MA(3)	MA(4)	CMA(4)
1992	1	**4**			
1993	2	**5**	**5.33**		
1994	3	**7**			
1995	4	6			
1996	5	8			
1997	6	9			
1998	7	5			
1999	8	2			
2000	9	3.5			
2001	10	5.5			
2002	11	6.5			

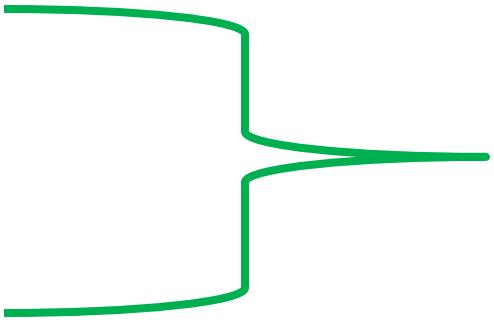
Moving Averages --- Example 1

Year	t	Data (Y_t)	MA(3)	MA(4)	CMA(4)
1992	1	**4**			
1993	2	**5**		**5.50**	
1994	3	**7**			
1995	4	**6**			
1996	5	8			
1997	6	9			
1998	7	5			
1999	8	2			
2000	9	3.5			
2001	10	5.5			
2002	11	6.5			



Moving Averages --- Example 1

Year	t	Data (Y_t)	MA(3)	MA(4)	CMA(4)
1992	1	4			
1993	2	5		5.50	
1994	3	7		**6.50**	
1995	4	6			
1996	5	8			
1997	6	9			
1998	7	5			
1999	8	2			
2000	9	3.5			
2001	10	5.5			
2002	11	6.5			



Moving Averages --- Example 1

Year	t	Data (Y_t)	MA(3)	MA(4)	CMA(4)
1992	1	4			
1993	2	5		5.50	
1994	**3**	7		6.50	**6.00**
1995	4	6			
1996	5	8			
1997	6	9			
1998	7	5			
1999	8	2			
2000	9	3.5			
2001	10	5.5			
2002	11	6.5			

Moving Averages --- Example 1

Year	t	Data (Y_t)	MA(3)	MA(4)	CMA(4)
1992	1	4			
1993	2	5	5.33	5.50	
1994	3	7	6.00	6.50	6.00
1995	4	6			
1996	5	8			
1997	6	9			
1998	7	5			
1999	8	2		4.00	
2000	9	3.5	3.67	4.38	4.19
2001	10	5.5	5.17		
2002	11	6.5			

Moving Averages --- Example 1

Year	t	Data (Y_t)	MA(3)	MA(4)	CMA(4)
1992	1	4			
1993	2	5	5.33	5.50	
1994	3	7	6.00	6.50	6.00
1995	4	6	7.00	7.50	7.00
1996	5	8	7.67	7.00	7.25
1997	6	9	7.33	6.00	6.50
1998	7	5	5.33	4.88	5.44
1999	8	2	3.50	4.00	4.44
2000	9	3.5	3.67	4.38	4.19
2001	10	5.5	5.17		
2002	11	6.5			

Moving Averages

Subjective choice of *k*

- *k* should be chosen to minimize the fluctuations (random variation) as best as possible. *This is done to get a better estimate/understanding of the effect of the underlying components in the time series (since random variation cannot be predicted).*

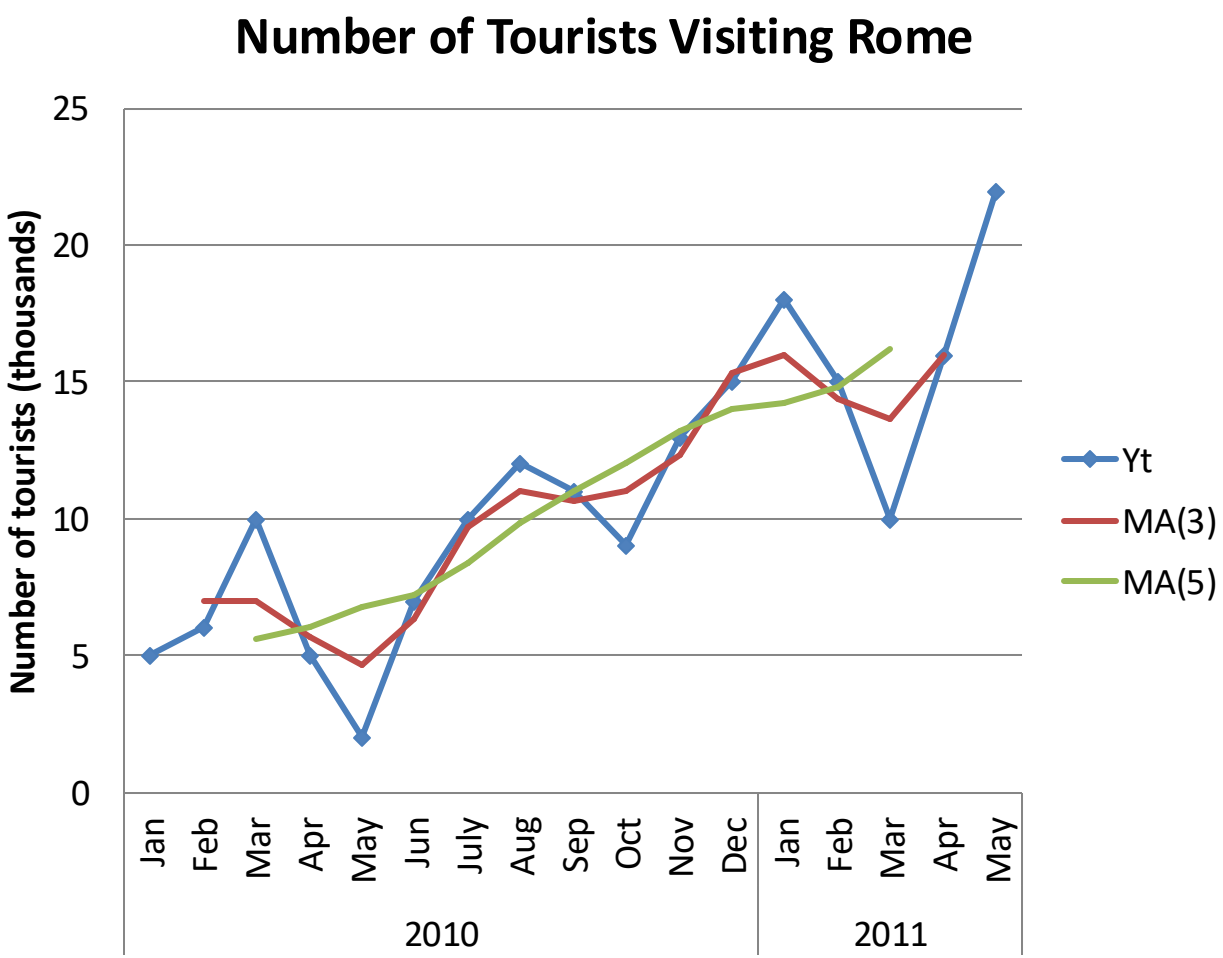
- If the data are quarterly, then $k = 4$ since there are four quarters in a year

- If the data are daily, then $k = 7$ since there are seven days in a week

- The bigger *k* gets, the smoother the series becomes

- If *k* is too large, the smoothed series tends towards a straight line (which may defeat the purpose of identifying components of a TS, unless you think that there is only a trend present OR you want to get a better idea as to the influence of the underlying trend)

Moving Averages --- Example



There is seasonality and increasing linear trend here in this time series.

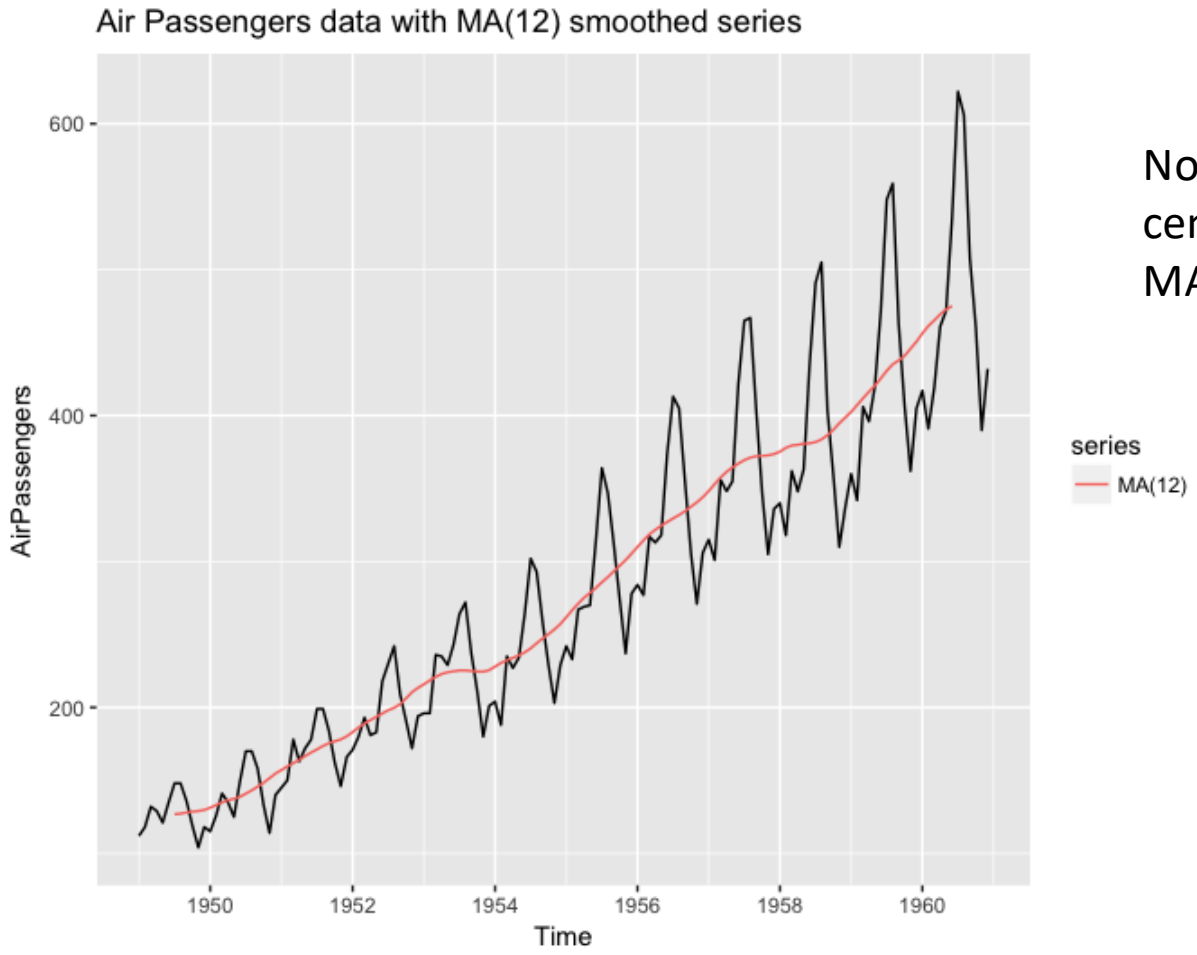
The MA(3) series removes some random variation so we can get a better understanding of the seasonality and trend.

The MA(5) series has removed some random variation and the seasonality – giving us a better picture of the underlying linear trend.

So, the choice of k also depends on what you want to achieve.

Moving Averages --- Example

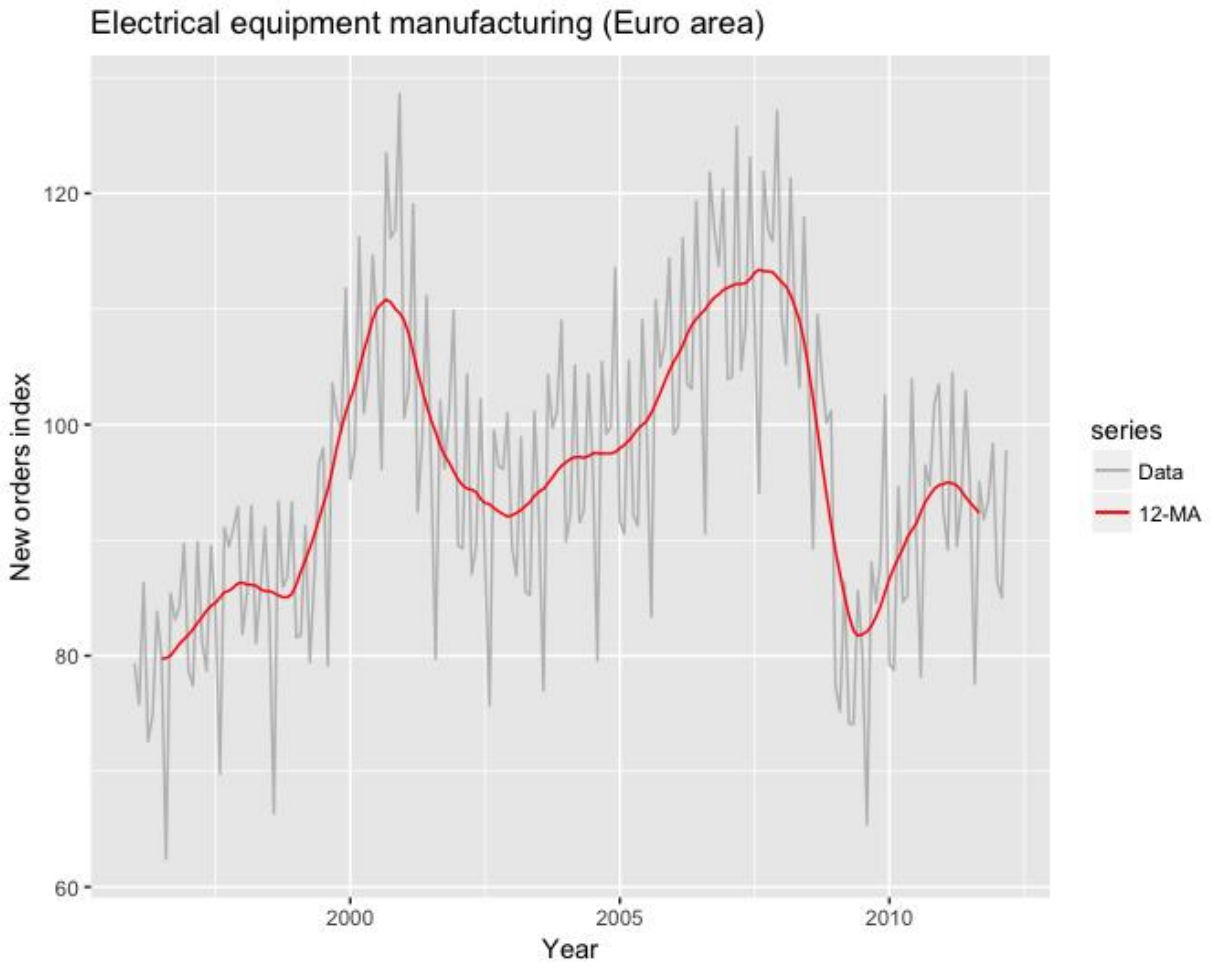
```
autoplot(AirPassengers) +
  autolayer(ma(AirPassengers, 12), series = "MA(12)") +
  ggtitle("Air Passengers data with MA(12) smoothed series")
```



Note that RStudio automatically centers the smoothed values, so MA(12) here is actually CMA(12)

Moving Averages --- Example

```
autoplot(elecequip, series="Data") +
  autolayer(ma(elecequip, 12), series="12-MA") +
  xlab("Year") + ylab("New orders index") +
  ggtitle("Electrical equipment manufacturing (Euro area)") +
  scale_colour_manual(values=c("Data"="grey","12-MA"="red"),
                      breaks=c("Data","12-MA"))
```



Moving Averages – Advantages/Disadvantages

- Advantage of moving average smoothing is that it is quick and easy to apply

- The problem with the moving average technique is that you will **lose *k* -1 observations if *k* is odd, and *k* observations if *k* is even** (after the averages have been centered). For example: For MA(3) there is no moving average value for the first and last time period and the corresponding observed value

- Also, once an observation falls out of the window of time periods for which the moving average is calculated, it is never considered again – which has implications when we use the moving average series to forecast future values

- The larger *k* is, the fewer number of moving averages that can be computed and plotted, thus sometimes making it difficult to obtain an overall impression of the entire series

Time Series Decomposition

Gaining a better understanding of the patterns in the historical time series data

(fpp) Chapter 3:
Sections 3.2 – 3.4

Classical decomposition

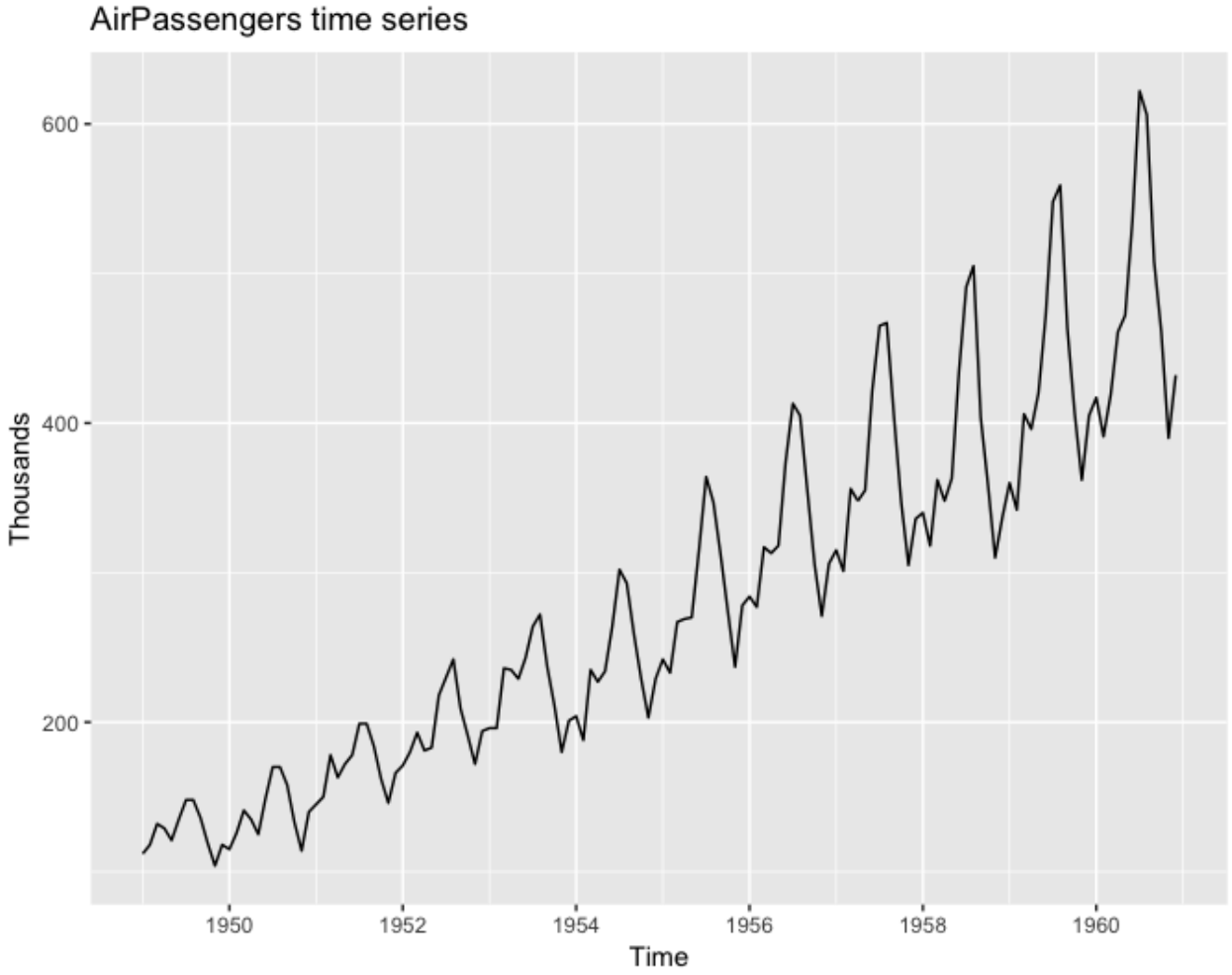
- We decompose a time series primarily to get a better understanding of the underlying components (i.e. trend and/or seasonality) as to how they are affecting the variable of interest.

- It is important for us to develop a clear understanding of what seasonal variation is. **Seasonal variation does NOT just refer to the 4 weather seasons**!

- The 4 weather seasons we naturally observe may **contribute** towards the *seasonal component/variation* we observe in a time series data set, but they (in and of themselves) are **not the seasonal component**

- Consider two examples:

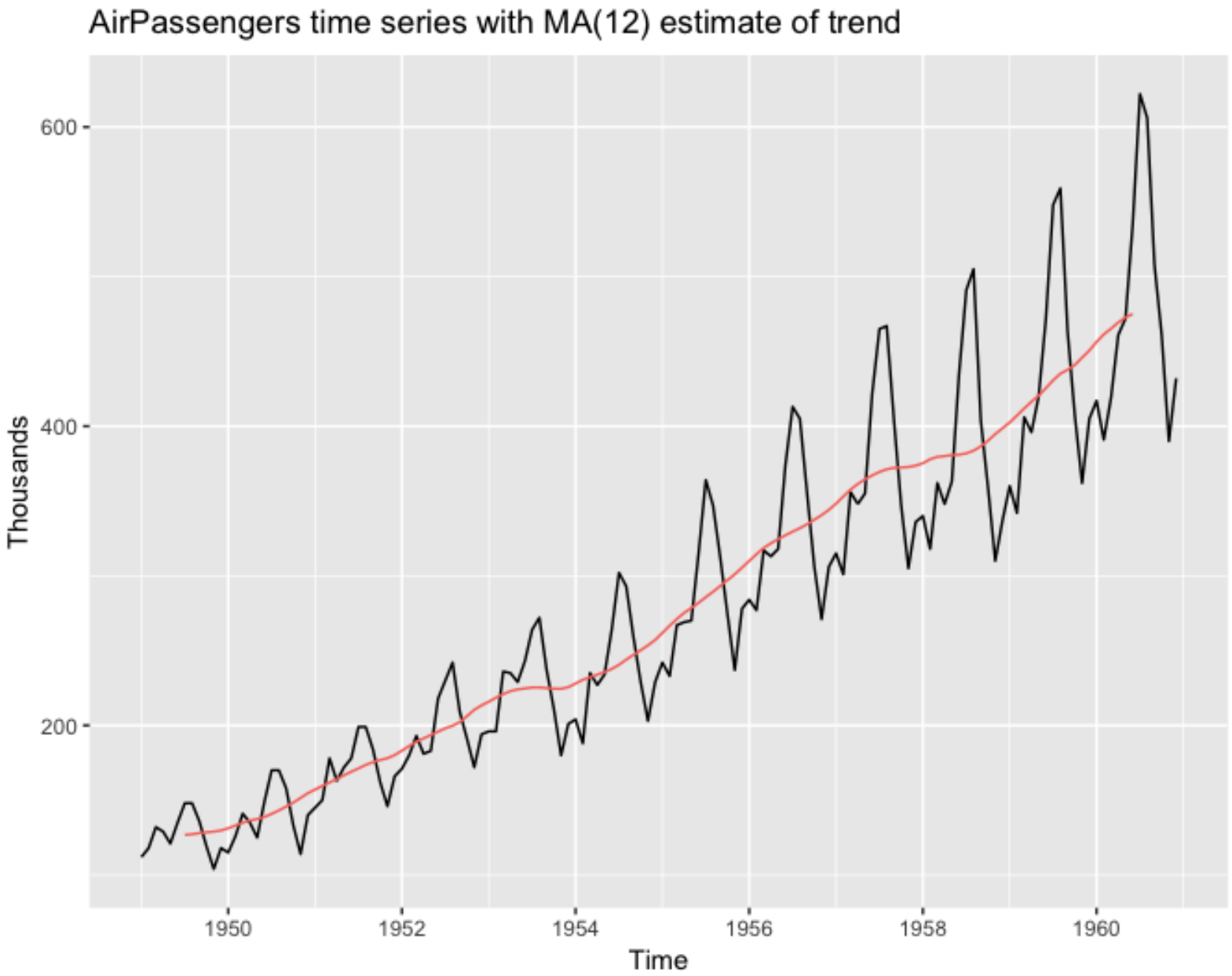
Classical decomposition



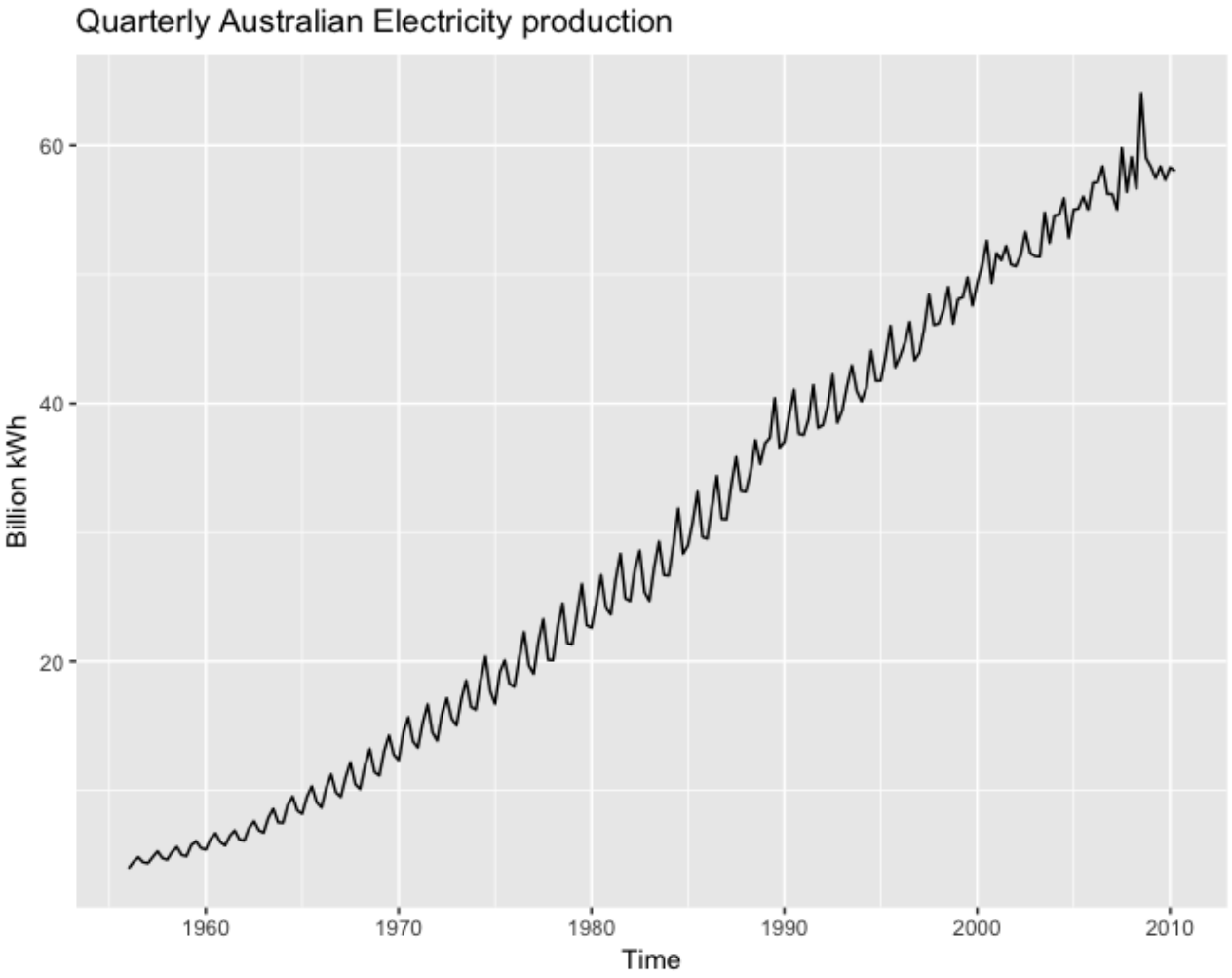
Here the data is recorded in MONTHS. We can deduce that the length (or amount of time periods) it takes for one seasonal variation/fluctuation is equivalent to 12 months i.e. one year.

We can say that, in this example, the length of the seasonal variation is one year, and that it is made up of 12 months or 12 "SEASONS". An appropriate k value for MA smoothing to remove the seasonality to estimate the underlying trend would thus be k = 12 (the length of one seasonal variation)

Classical decomposition



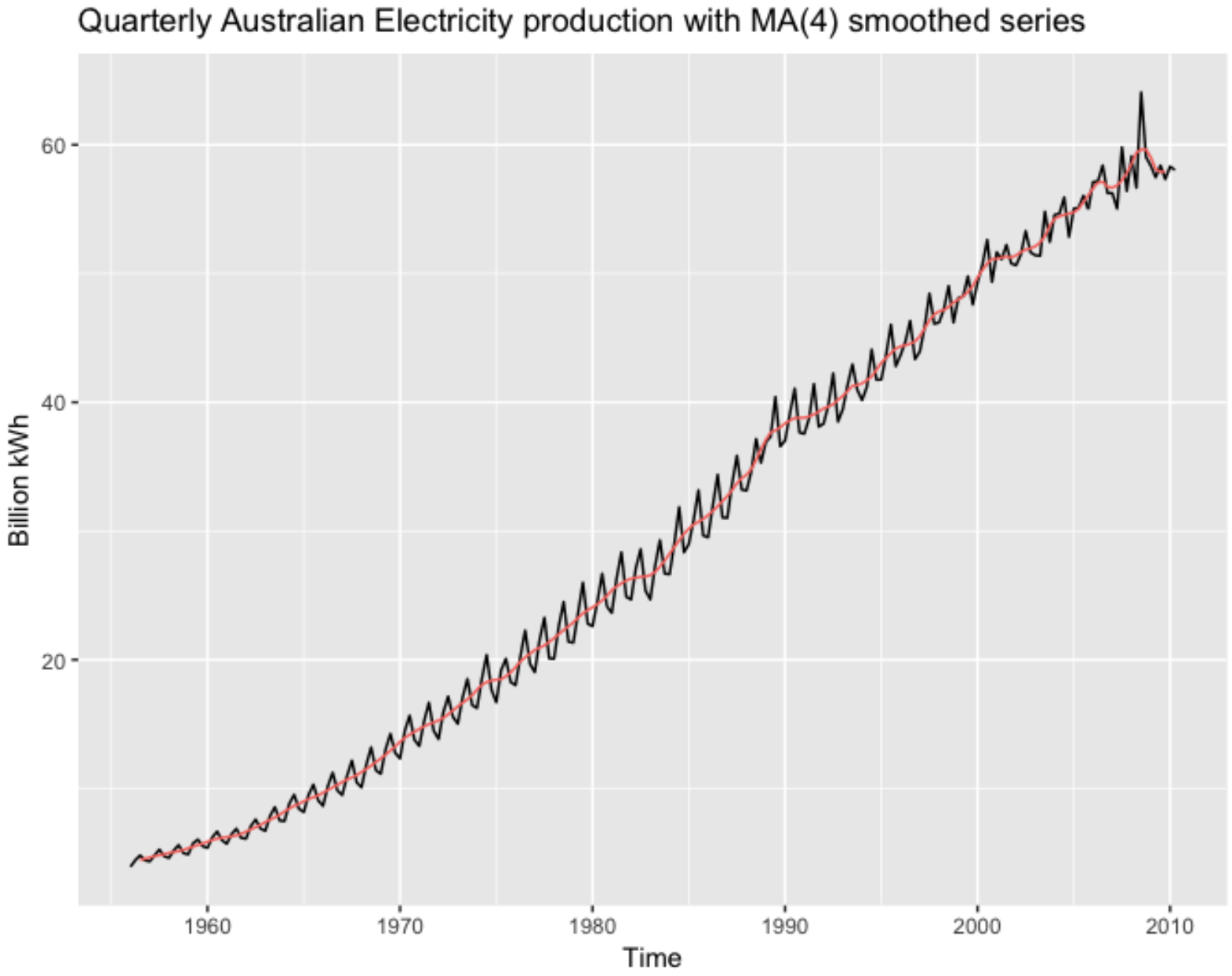
Classical decomposition



Here the data is recorded in quarters. We can deduce that the length (or amount of time periods) it takes for one seasonal variation/fluctuation is equivalent to 4 quarters i.e. one year.

We can say that, in this example, the length of the seasonal variation is one year, and that it is made up of 4 quarters or "SEASONS". An appropriate k value for our MA smoothing to remove the seasonality to better estimate the underlying trend would thus be k = 4 (the length of one seasonal variation)

Classical decomposition



Classical decomposition – the entire process

1) Develop an estimate of the trend component using centered moving averages

2) De-trend the original time series (i.e. isolate the seasonal component)

3) Calculate seasonal indices and then **adjust them if necessary** (precise estimates of seasonal component)

4) Calculate the random component

5) Deseasonalize the original time series (i.e. isolate the trend component)

6) If the trend is linear, develop a precise estimate of trend component using a trend analysis on the deseasonalized time series

Classical decomposition

- As previously mentioned, most often we encounter time series that have both trend and seasonality.
  - In such cases, to better understand each component, we need to isolate them (i.e. "separate" them from each other) in the original time series

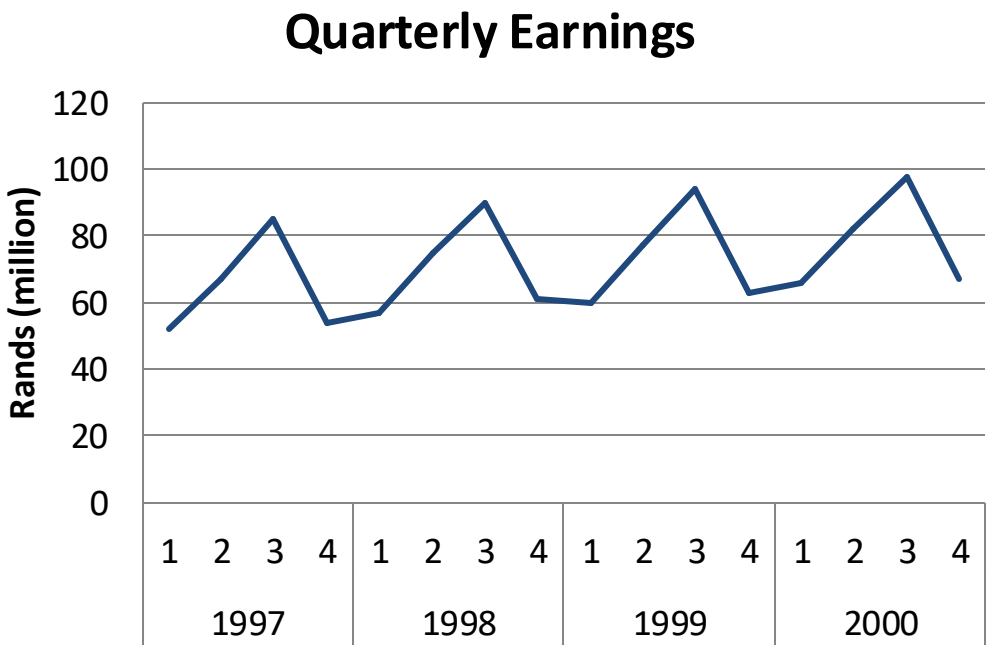
- We isolate the seasonal variation by removing the trend component (i.e. "de-trending" the original time series) and computing seasonal indexes, which are simply the average of all the values in each "season". These indices help gauge the degree to which seasons differ from one another, relative to the overall seasonal average.
  - Recall that we have assumed that C_t is negligible for the purposes of this course:

$$Y_t = T_t \times S_t \times R_t$$

Classical decomposition --- Example

The quarterly earning (in R millions) of a large drink manufacturer have been recorded for the years 1997 to 2000. The data are listed below:

	Quarter			
Year	**1**	**2**	**3**	**4**
1997	52	67	85	54
1998	57	75	90	61
1999	60	77	94	63
2000	66	82	98	67



Decompose the time series into its components, estimate the components and then build a model that could be used to forecast future quarterly earnings values.

Classical decomposition

1) We first develop a crude estimate of the trend component in the series by calculating (centered) moving averages (CMA's) which allow us to smooth the seasonal and some of the random component out of the time series:

$$CMA_t \approx T_t$$

Year	Quarter	t	Quarterly Earnings (Y_t)	$T_t \approx CMA(4)_t$
1997	1	1	52	
1997	2	2	67	
1997	3	3	85	65.125
1997	4	4	54	
1998	1	5	57	
1998	2	6	75	
1998	3	7	90	
1998	4	8	61	
1999	1	9	60	
1999	2	10	77	
1999	3	11	94	
1999	4	12	63	
2000	1	13	66	
2000	2	14	82	77.750
2000	3	15	98	
2000	4	16	67	

Year	Quarter	t	Quarterly Earnings (Y_t)	$T_t \approx CMA(4)_t$
1997	1	1	52	
1997	2	2	67	
1997	3	3	85	65.125
1997	4	4	54	66.750
1998	1	5	57	68.375
1998	2	6	75	69.875
1998	3	7	90	71.125
1998	4	8	61	71.750
1999	1	9	60	72.500
1999	2	10	77	73.250
1999	3	11	94	74.250
1999	4	12	63	75.625
2000	1	13	66	76.750
2000	2	14	82	77.750
2000	3	15	98	
2000	4	16	67	

Classical decomposition

2) We "DE-TREND" the original time series i.e. for each time period, compute the ratio:

$$\frac{Y_t}{CMA(k)_t}$$

De-trending the time series results in a measure of seasonal variation and some random variation. That is, the multiplicative model is now:

$$\frac{Y_t}{CMA(k)_t} \approx \frac{T_t \times S_t \times R_t}{T_t} = S_t \times R_t$$

Year	Quarter	t	Earnings (Y_t)	(T_t)	De-trended Data (Y_t / T_t)
1997	1	1	52		
1997	2	2	67		
1997	3	3	85	65.125	1.305
1997	4	4	54	66.750	
1998	1	5	57	68.375	
1998	2	6	75	69.875	
1998	3	7	90	71.125	
1998	4	8	61	71.750	
1999	1	9	60	72.500	
1999	2	10	77	73.250	
1999	3	11	94	74.250	
1999	4	12	63	75.625	
2000	1	13	66	76.750	
2000	2	14	82	77.750	1.055
2000	3	15	98		
2000	4	16	67		

Year	Quarter	t	Earnings (Y_t)	(T_t)	De-trended Data (Y_t / T_t)
1997	1	1	52		
1997	2	2	67		
1997	3	3	85	65.125	1.305
1997	4	4	54	66.750	0.809
1998	1	5	57	68.375	0.834
1998	2	6	75	69.875	1.073
1998	3	7	90	71.125	1.265
1998	4	8	61	71.750	0.850
1999	1	9	60	72.500	0.828
1999	2	10	77	73.250	1.051
1999	3	11	94	74.250	1.266
1999	4	12	63	75.625	0.833
2000	1	13	66	76.750	0.860
2000	2	14	82	77.750	1.055
2000	3	15	98		
2000	4	16	67		

Classical decomposition

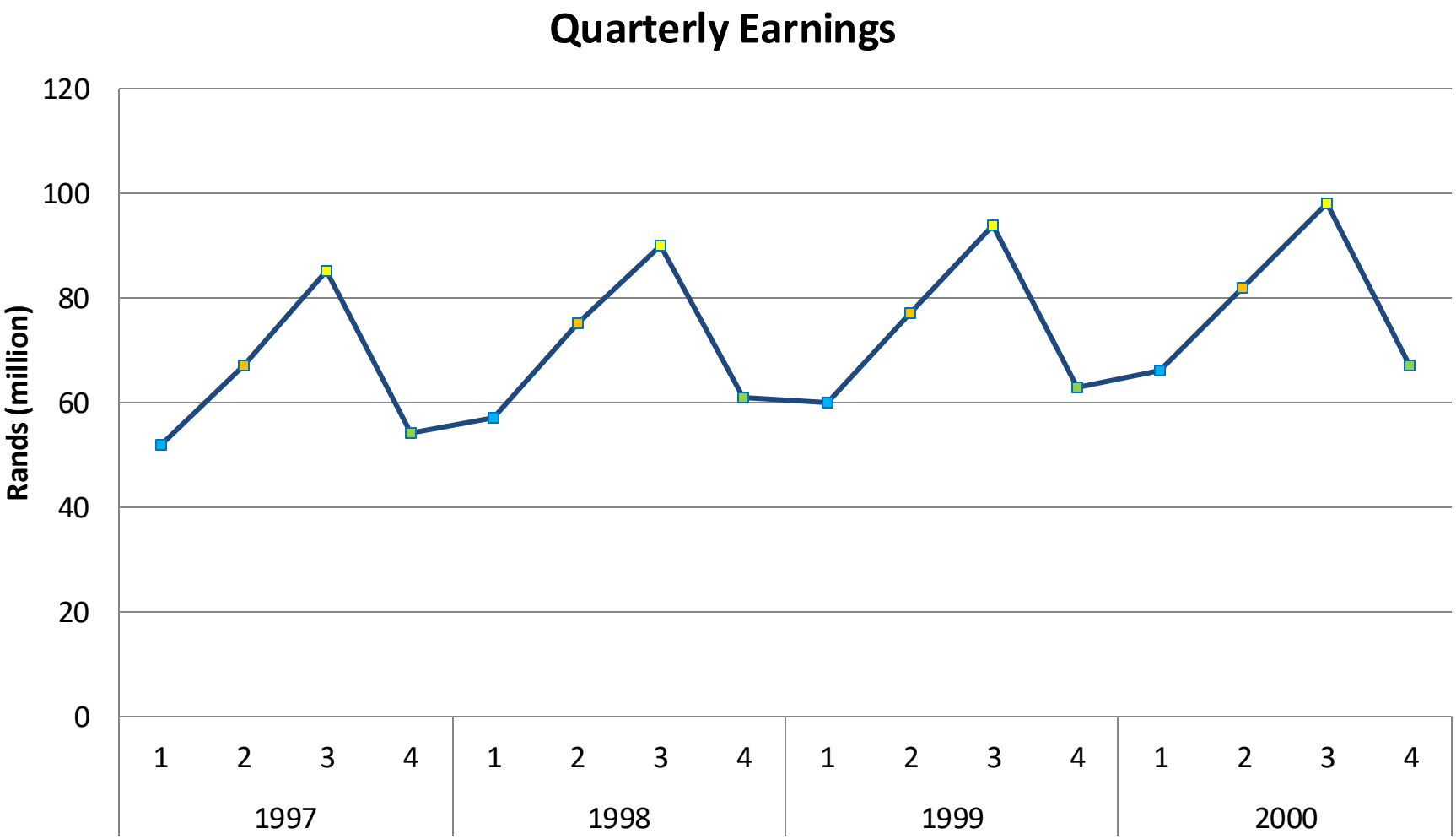
3) Calculate seasonal indices, then adjust them if necessary

- If we group the resulting de-trended data by the corresponding "seasons" e.g. by quarters or by months, and take the average of the data values in each "season" to get seasonal indices

- In other words, we average the de-trended data for each *m* types of "season" present in the time series

- In this example our data is recorded in quarters, so we group the **corresponding** quarters together and take the averages of these groups of de-trended data values to give us our seasonal indices

Classical decomposition



Year	Quarter	t	Earnings (Y_t)	(T_t)	De-trended Data (Y_t / T_t)
1997	1	1	52		
1997	2	2	67		
1997	3	3	85	65.125	1.305
1997	4	4	54	66.750	0.809
1998	1	5	57	68.375	0.834
1998	2	6	75	69.875	1.073
1998	3	7	90	71.125	1.265
1998	4	8	61	71.750	0.850
1999	1	9	60	72.500	0.828
1999	2	10	77	73.250	1.051
1999	3	11	94	74.250	1.266
1999	4	12	63	75.625	0.833
2000	1	13	66	76.750	0.860
2000	2	14	82	77.750	1.055
2000	3	15	98		
2000	4	16	67		

Classical decomposition --- Example

		Quarter			
Year	1	2	3	4	
1997			1.305	0.809	
1998	0.834	1.073	1.265	0.850	
1999	0.828	1.051	1.266	0.833	
2000	0.860	1.055			
Seasonal Index (SI)					

Classical decomposition --- Example

		Quarter			
Year	1	2	3	4	
1997			1.305	0.809	
1998	0.834	1.073	1.265	0.850	
1999	0.828	1.051	1.266	0.833	
2000	0.860	1.055			
Seasonal Index (SI)					

Classical decomposition --- Example

		Quarter			
Year	**1**	**2**	**3**	**4**	**Total**
1997			1.305	0.809	
1998	0.834	1.073	1.265	0.850	
1999	0.828	1.051	1.266	0.833	
2000	0.860	1.055			
Seasonal Index (SI)	0.841	1.060	1.279	0.831	4.011
Adjusted SI					

Classical decomposition

- Under the multiplicative model, **IF there was no seasonal component** then $S_t = 1$ for all *m* "seasons" and hence the sum of all the S_t would be equal to:

  (1 x m) = *m.* This property is also applicable to time series in which *there is a seasonal component* – here the seasonal indices fluctuate around 1.

- Hence, we must adjust the *m* seasonal indices so that their sum is equal to the number of "seasons" in the time series (*m*) i.e. we adjust the indices so that their average for the m "seasons" is 1 . The reason why the seasonal indices often don't sum exactly to 'm' is because of the influence of random variation.

- In this example, as we have 4 "seasons", we adjust only if the indices do not sum to **4**, using the following correction factor:

$$\textbf{Correction factor} = \frac{\boldsymbol{m}}{\textbf{sum of m seasonal indices}}$$

Classical decomposition --- Example

		Quarter			
Year	1	2	3	4	Total
1997			1.305	0.809	
1998	0.834	1.073	1.265	0.850	
1999	0.828	1.051	1.266	0.833	
2000	0.860	1.055			
Seasonal Index (SI)	0.841	1.060	1.279	0.831	4.011
Adjusted SI					

$$Correction\ factor = \frac{m}{sum\ of\ m\ seasonal\ indices}$$

$$= \frac{4}{4.011}$$

Classical decomposition --- Example

		Quarter			
Year	**1**	**2**	**3**	**4**	**Total**
1997			1.305	0.809	
1998	0.834	1.073	1.265	0.850	
1999	0.828	1.051	1.266	0.833	
2000	0.860	1.055			
Seasonal Index (SI)	0.841	1.060	1.279	0.831	4.011
Adjusted SI	0.839	1.057	1.275	0.829	4.000

This leaves us with estimates of the seasonal component S_t $--- \widehat{S_t}$ which are the ***adjusted*** **seasonal indices**

Note: Interpreting the seasonal indices (relative to overall seasonal average) is important!

Classical decomposition --- Example

How do we interpret the seasonal indices?

Recall that if there was no seasonal component in a time series, then the value of S_t would equal 1 in every time period. Clearly, when seasonality is present the seasonal indices fluctuate above and below 1.

$S_t > 1$ implies that, **ON AVERAGE/IN GENERAL across the time series** values in that "season" are $100 \times |S_t - 1|\%$ ***ABOVE*** the seasonal average, where seasonal is the length of a seasonal variation i.e. it could be annual, weekly, daily etc. depending on the frequency of how your data is recorded.

$S_t < 1$ implies that, **ON AVERAGE/IN GENERAL across the time series** values in that "season" are $100 \times |S_t - 1|\%$ ***BELOW*** the seasonal average.

Classical decomposition --- Example

	Quarter				
Year	**1**	**2**	**3**	**4**	**Total**
1997			1.305	0.809	
1998	0.834	1.073	1.265	0.850	
1999	0.828	1.051	1.266	0.833	
2000	0.860	1.055			
Seasonal Index (SI)	0.841	1.060	1.279	0.831	4.011
Adjusted SI	0.839	1.057	1.275	0.829	4.000

In this example, each seasonal variation is one year in length. Hence, we would interpret each seasonal index as follows:

$S_1 = 0.839$ implies that, on average, earnings in the first quarter are 100*|0.839-1| = 16.1% below the annual average earnings

$S_2 = 1.057$ implies that, on average, earnings in the second quarter are 100*|1.057-1| = 5.7% above the annual average earnings

$S_3 = 1.275$ implies that, on average, earnings in the third quarter are 100*|1.275-1| = 27.5% above the annual average earnings

$S_4 = 0.829$ implies that, on average, earnings in the fourth quarter are 100*|0.829-1| = 17.1% below the annual average earnings

Classical decomposition

4) We compute the random component by dividing the ORIGINAL time series by the product of the estimates of the trend and seasonal components:

$$R_t = \frac{Y_t}{\widehat{S_t} \times \widehat{T_t}} = \frac{Y_t}{\widehat{S_t} \times \widehat{CMA(k)_t}}$$

i.e. divide the observed time series value by (its corresponding adjusted seasonal index)*(the CMA(k) estimate of the trend component)

error term

Year	Quarter	t	Y_t	$CMA(4)_t$	De-Trended Data	Adjusted Seasonal Index	Random Component
1997	1	1	52			0.839	
	2	2	67			1.057	
	3	3	85	65.125	1.305	1.275	1.024
	4	4	54	66.750	0.809	0.829	0.976
1998	1	5	57	68.375	0.834	0.839	0.994
	2	6	75	69.875	1.073	1.057	1.015
	3	7	90	71.125	1.265	1.275	0.992
	4	8	61	71.750	0.850	0.829	1.026
1999	1	9	60	72.500	0.828	0.839	0.986
	2	10	77	73.250	1.051	1.057	0.995
	3	11	94	74.250	1.266	1.275	0.993
	4	12	63	75.625	0.833	0.829	1.005
2000	1	13	66	76.750	0.860	0.839	1.025
	2	14	82	77.750	1.055	1.057	0.998
	3	15	98			1.275	
	4	16	67			0.829	

Classical decomposition

- Seasonal indices allow as to compare a time series across seasons. However, we need to get a better estimate of the trend movement (in the past). To do this it is necessary to isolate the trend component. Thus:

5) We deseasonalize the ORIGINAL time series. The removal of seasonal variations results in a smoother **seasonally-adjusted** time series which makes it easier to estimate the trend component. We deseasonalize the time series as follows:

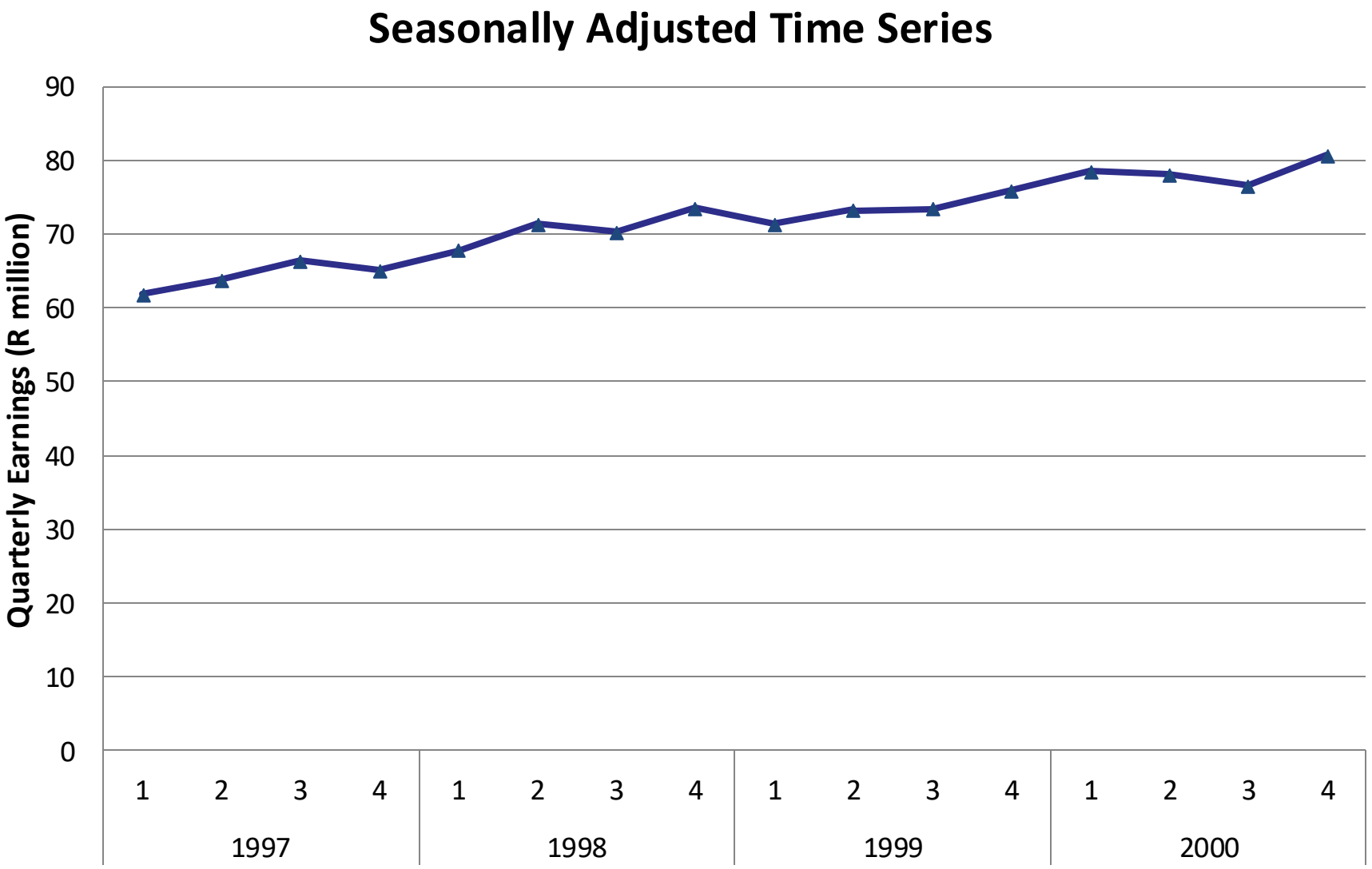
Divide the observed time series value by its corresponding adjusted seasonal index, resulting in a deseasonalized multiplicative model:

$$\frac{Y_t}{\hat{S}_t} = \frac{T_t \times S_t \times R_t}{\hat{S}_t} \approx T_t \times R_t = (DS)_t = T_t + \epsilon_t$$

Year	Quarter	t	Y_t	$CMA(4)_t$	De-Trended Data	Adjusted Seasonal Index	Seasonally Adjusted Time Series
1997	1	1	52			0.839	61.979
	2	2	67			1.057	63.387
	3	3	85	65.125	1.305	1.275	66.667
	4	4	54	66.750	0.809	0.829	
1998	1	5	57	68.375	0.834	0.839	
	2	6	75	69.875	1.073	1.057	
	3	7	90	71.125	1.265	1.275	
	4	8	61	71.750	0.850	0.829	
1999	1	9	60	72.500	0.828	0.839	
	2	10	77	73.250	1.051	1.057	
	3	11	94	74.250	1.266	1.275	
	4	12	63	75.625	0.833	0.829	
2000	1	13	66	76.750	0.860	0.839	
	2	14	82	77.750	1.055	1.057	77.578
	3	15	98			1.275	76.863
	4	16	67			0.829	80.820

Year	Quarter	t	Y_t	$CMA(4)_t$	De-Trended Data	Adjusted Seasonal Index	Seasonally Adjusted Time Series
1997	1	1	52			0.839	61.979
	2	2	67			1.057	63.387
	3	3	85	65.125	1.305	1.275	66.667
	4	4	54	66.750	0.809	0.829	65.139
1998	1	5	57	68.375	0.834	0.839	67.938
	2	6	75	69.875	1.073	1.057	70.956
	3	7	90	71.125	1.265	1.275	70.588
	4	8	61	71.750	0.850	0.829	73.583
1999	1	9	60	72.500	0.828	0.839	71.514
	2	10	77	73.250	1.051	1.057	72.848
	3	11	94	74.250	1.266	1.275	73.725
	4	12	63	75.625	0.833	0.829	75.995
2000	1	13	66	76.750	0.860	0.839	78.665
	2	14	82	77.750	1.055	1.057	77.578
	3	15	98			1.275	76.863
	4	16	67			0.829	80.820

Classical decomposition --- Example



Classical decomposition

6) After we have deseasonalized the ORIGINAL time series, **IF THE TREND LOOKS LINEAR** we can regress the deseasonalized data on time *t:*

$$\widehat{T}_t = \widehat{\beta}_0 + \widehat{\beta}_1 t$$

In summary: We need to isolate the TREND component by removing the seasonal component from the original time series. Once we have deseasonalized the original time series, we are left with only the TREND component and some random variation.

IF the trend looks linear, a trend analysis (via linear regression) will provide us with a good estimate of the trend component. Hence, we conduct a simple linear regression analysis on the (deseasonalized) seasonally-adjusted time series:

Classical decomposition --- Example

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.96
R Square	0.93
Adjusted R Square	0.92
Standard Error	1.43
Observations	15.00

$$\widehat{T}_t = 62.56 + 1.10t$$

ANOVA

	df	*SS*	*MS*	*F*
Regression	1.00	336.34	336.34	163.96
Residual	13.00	26.67	2.05	
Total	14.00	363.00		

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	62.56	0.85	73.22	0.00
t	1.10	0.09	12.80	0.00

Classical decomposition

We now have more precise estimates of both the *seasonal component* (via SEASONAL INDICES) and the *trend component* (via a regression analysis of the deseasonalized data) of the time series, which is the main goal of conducting a classical decomposition.

However, should we wish to forecast future values of a time series that has a linear trend using a classical decomposition model, we are now able to forecast using ***seasonally-adjusted trend estimates***, with the following multiplicative model:

$$\hat{Y}_t = \hat{T}_t \times \hat{S}_t = (\hat{\beta}_0 + \hat{\beta}_1 t) \times \hat{S}_t$$

We don't cover an example of forecasting with classical decomposition here since it is not recommended to use it as a forecasting model (see next slide), but do try the past examples in the Tuts and tests to get the hang of how it is done.

Classical decomposition – drawbacks

While classical decomposition is still widely used, it is no longer necessary as more sophisticated methods have been developed in recent years. It is also not recommended for the following reasons:

- The estimate of the trend is unavailable for the first few and last few observations, due to the use of moving average smoothing to estimate the trend. Consequently, there is also no estimate of the remainder component for the same time periods.

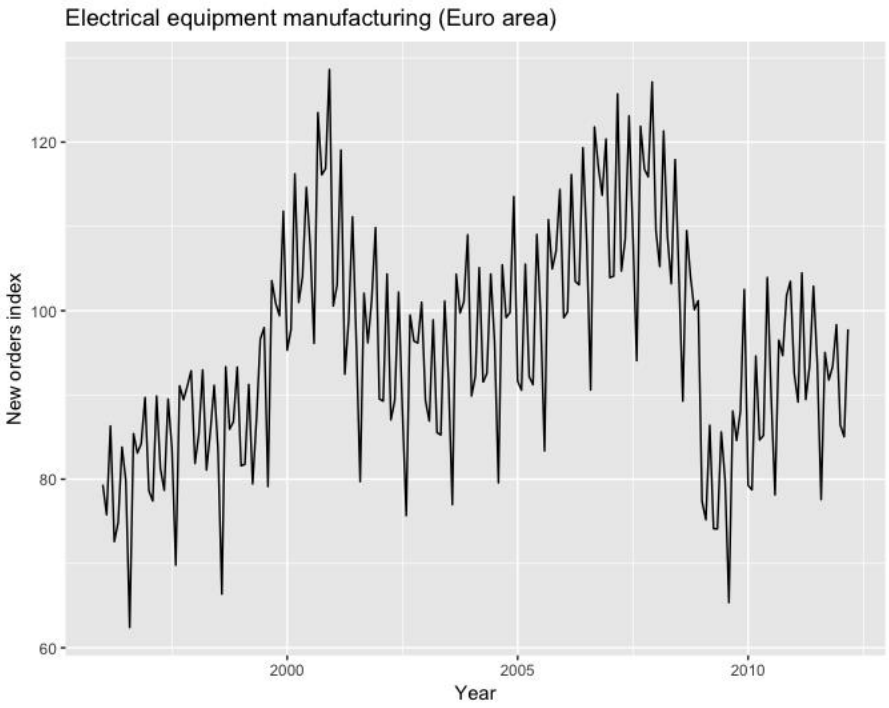
- The trend estimate tends to over-smooth rapid rises and falls in the data

- Classical decomposition methods assume that the seasonal component repeats from year to year. For many series, this is a reasonable assumption, but for some longer series it is not. For these series, classical decomposition methods are unable to capture the seasonal changes over time.

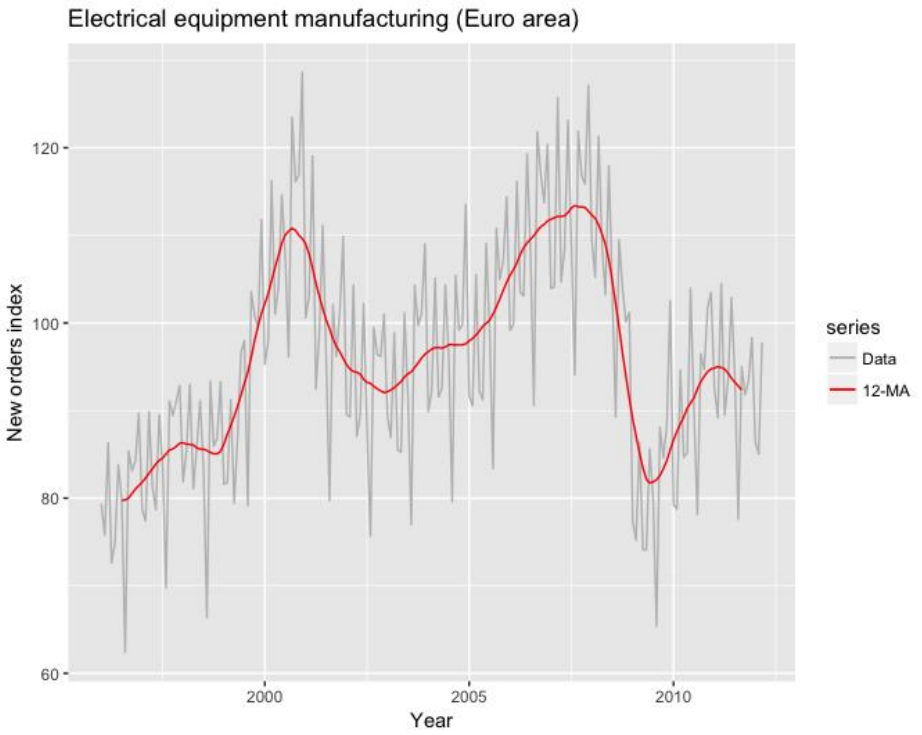
- Occasionally, the values of the time series in a small number of periods may be particularly unusual. Classical decomposition is not robust to these kinds of unusual values.

Classical decomposition – RStudio

```
autoplot(elecequip) +
  xlab("Year") + ylab("New orders index") +
  ggtitle("Electrical equipment manufacturing (Euro area)")
```



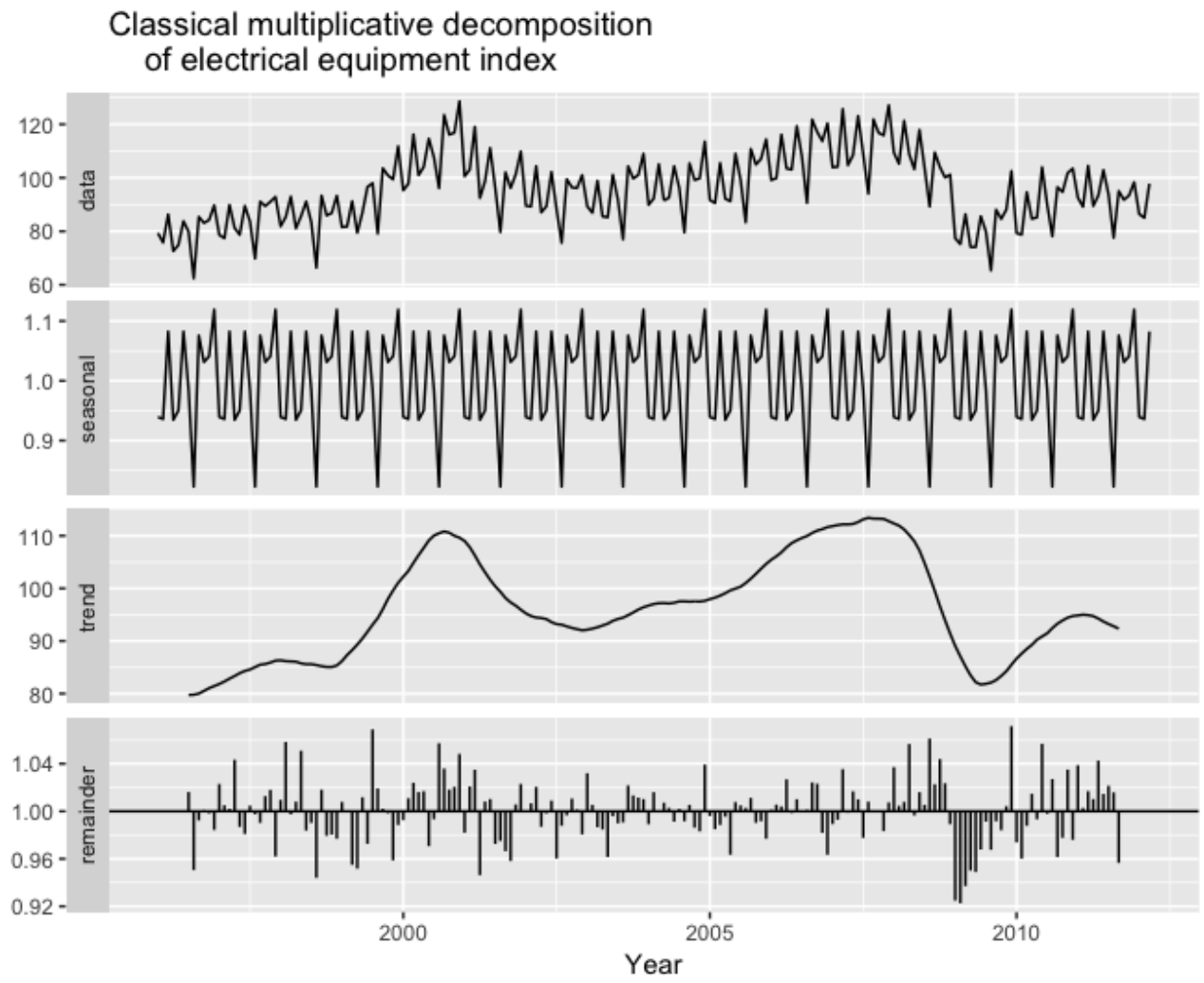

```
autoplot(elecequip, series="Data") +
  autolayer(ma(elecequip, 12), series="12-MA") +
  xlab("Year") + ylab("New orders index") +
  ggtitle("Electrical equipment manufacturing (Euro area)") +
  scale_colour_manual(values=c("Data"="grey","12-MA"="red"),
                      breaks=c("Data","12-MA"))
```



Note that the ma() function in RStudio automatically centers the smoothed values when k is even.

Classical decomposition – RStudio

```
elecequip %>% decompose(type="multiplicative") %>%
  autoplot() + xlab("Year") +
  ggtitle("Classical multiplicative decomposition
    of electrical equipment index")
```



The forecaster's toolbox

A selection of tools to use in conducting a time series analysis

(fpp) Chapter 5:
Sections 5.2 – 5.4 & 9.1

Some simple forecasting methods

- There are some forecasting methods that are simple, yet remarkably effective for some time series.

- We fit simple forecasting methods to **use as 'benchmarks' against which we compare the other more complex forecasting methods** i.e. if a more complicated forecasting method does not yield better forecasts than one of the simple methods here, there is no need to use it for the particular time series being analysed

- There are four simple forecasting methods that we consider:
  - The average method
  - The naïve method
  - The seasonal naïve method
  - The drift method

Some simple forecasting methods

Average method: the forecasts of all future values is the mean of the historical data:

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

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

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

Naïve method: the forecasts of all future values is the value of the last observation:

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

```
naive(y, h)
rwf(y, h) # Equivalent alternative
```

As previously mentioned, the naïve methods performs well for many financial and economic time series.

Some simple forecasting methods

Seasonal naïve method: this is similar to the naïve method, used when we have a time series with a strong seasonal component. The forecasted value for a particular 'season' is simply the value corresponding to the previous 'season':

$$\hat{y}_{T+h|T} = y_{T+h-m(k+1)},$$

Where m is the seasonal period and k is the integer part of (h-1)/m i.e. the number of complete seasonal variations that have passed since the end of the original time series data.

For example, if our time series is monthly data and a seasonal variation lasts 12 months, then the future forecasts for all April values is simply the previous April value. If our time series is quarterly data and a season lasts 4 quarters, then the future forecast for all quarter 3 values is the previous quarter 3 value etc.

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

Some simple forecasting methods

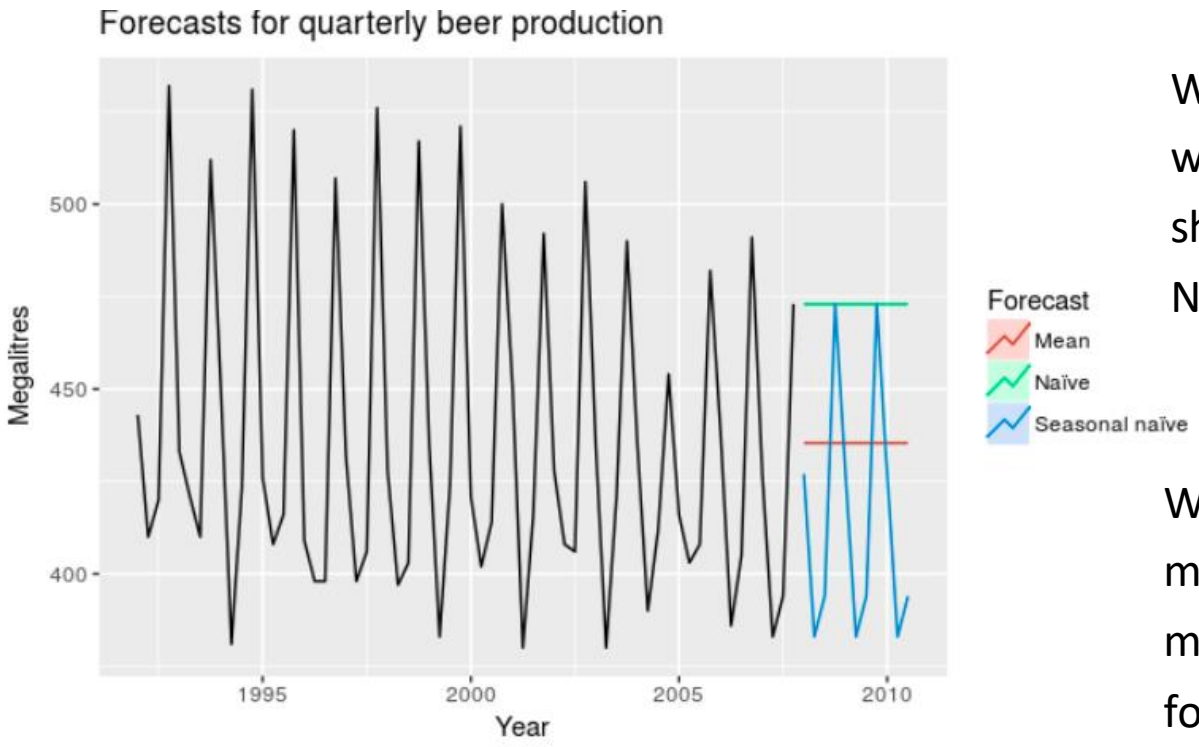
Drift method: An extension to the naïve method to allow for the presence of a linear trend in the data. The amount of change over time (called the 'drift') is equal to the average change in the historical data:

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

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

Some simple forecasting methods – Example 1

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

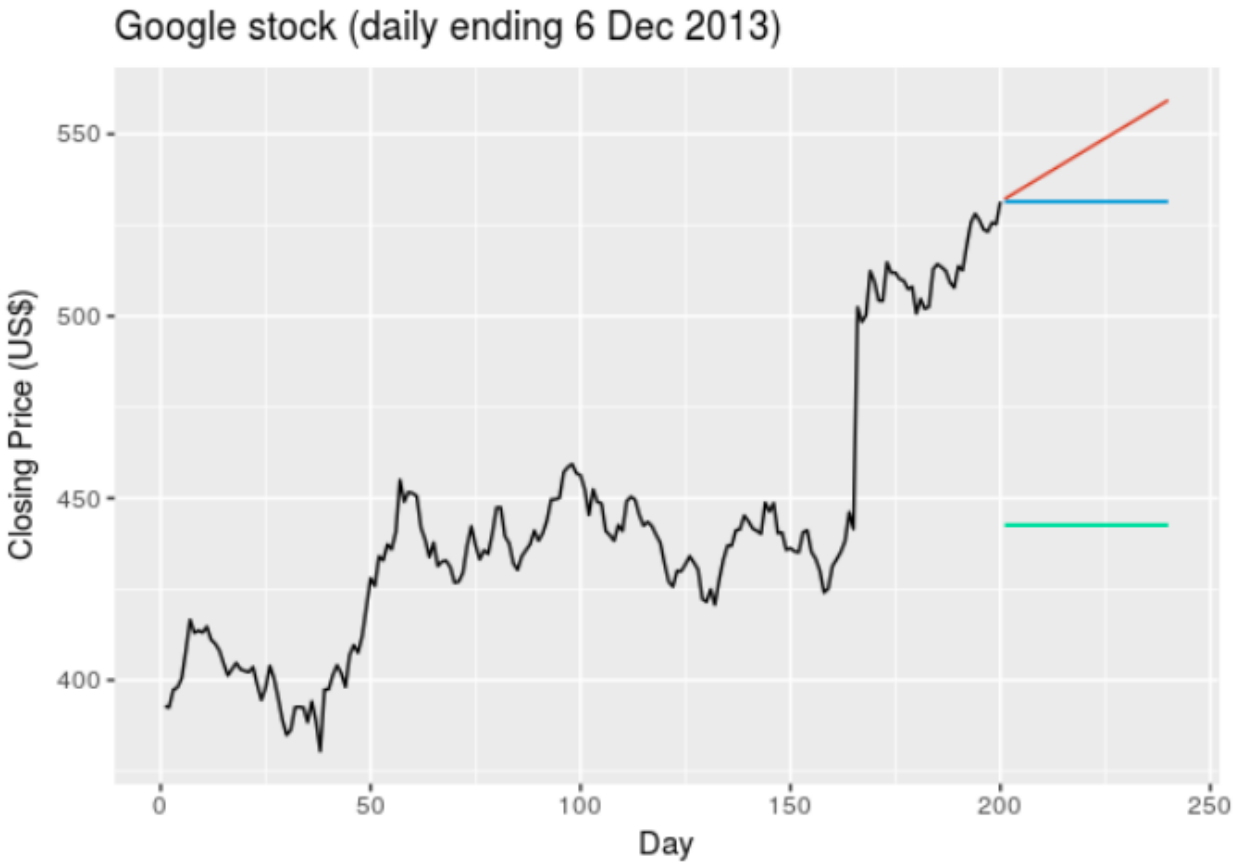


Which of these fitted models will provide the most accurate short-term forecasts? Seasonal Naïve. Why?

Which simple forecasting method will provide the most accurate long-term forecasts? Drift. Why?

Some simple forecasting methods – Example 2

```
# Plot some forecasts
autoplot(goog200) +
autolayer(meanf(goog200, h=40), series="Mean", PI=FALSE) +
autolayer(rwf(goog200, h=40), series="Naïve", PI=FALSE) +
autolayer(rwf(goog200, drift=TRUE, h=40),series="Drift", PI=FALSE) +
ggtitle("Google stock (daily ending 6 Dec 2013)") +
xlab("Day") + ylab("Closing Price (US$)") +
guides(colour=guide_legend(title="Forecast"))
```



Which method will provide the most accurate short-term forecasts? Perhaps naïve or drift.

Which method will provide the most accurate long-term forecasts? Drift.

Why?

Transformations

- The stationarity condition is quite strong – it implies that the mean **and** variance must be constant over time. We have already noted that in many series this is not true, since they have either trend or seasonality (or both), and/or variation that changes with time.
- The condition of a constant mean rules out many regression problems where the general trend is up or down over time. We have also noted many time series also display seasonality. Often, the variance of a time series changes over time.
- There are several ways in which a time series can be ***transformed***, some involving estimating the components and subtracting them from the data, and others depending on differencing the data. *Whichever method is used, the aim is to produce a stationary time series that displays constant mean (i.e. no trend or seasonality) and constant variance. This is done for models that want to understand what information/patterns other than the trend and/or seasonality is present in the data. These models hope to capture that information (autocorrelation) and combine it with the information represented by the trend and or seasonality when making forecasts. In this course, we consider ARIMA models that do this.* ***Hence, transformations to time series form part of the ARIMA model-fitting process. We don't need to transform time series before fitting exponential smoothing models, since they focus on modelling the components of the time series.***
- We usually start transforming a time series by applying variance stabilising transformations if they are necessary, before removing trend and/or seasonality.

Transformations

a) Stabilising the variance

- If we observe that the variance of the time series increases or decreases with the level of the time series, then we need to apply some sort of transformation to stabilise the variance.

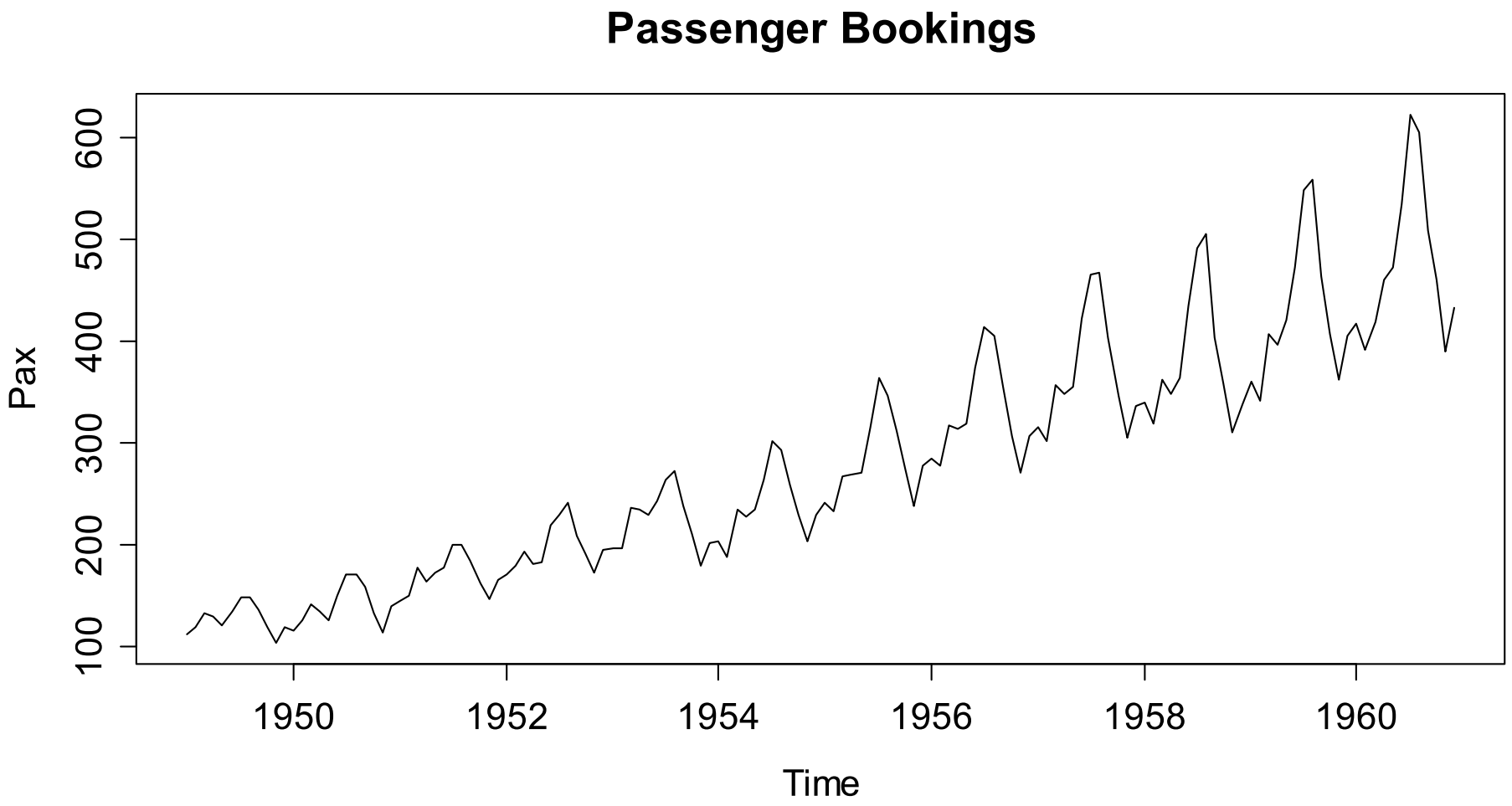
- For example, if the magnitude of the fluctuations appear to grow roughly linearly with the mean of the series, then applying logarithms to the data will result in the transformed series $\{\ln(y_1), \ln(y_2). . . , \ln(y_T)\}$ that will have fluctuations of more constant magnitude.

- A square root or cube root transformation can also be useful in stabilizing the variance of a time series when the variance of the series is proportional to the (changing) mean. These transformations are referred to as power transformations as they can be expressed as $w_t = y_t^p$ where $p = \frac{1}{2}$ $for\ a\ square\ root\ and\ p = \frac{1}{3}$ $for\ a\ cube\ root$

- **NB – for the purposes of this course, we assess changes in variation graphically, and only consider time series where it is clear that the variance is changing (or not).** *Always look at the entire time series plot when assessing whether the variance is changing.*

Transformations

a) Stabilising the variance - Example



Here is it obvious that the variance in passenger bookings is increasing with time

Transformations

a) Stabilising the variance - Example

- From the time series plot, we can see that the variance is changing (increasing) over time, as the mean changes. Hence, this time series is non-stationary (in both mean and variance).

- We thus need to transform the data in some way in order to stabilise the variance. We could either seek to apply a natural logarithm to the data series:

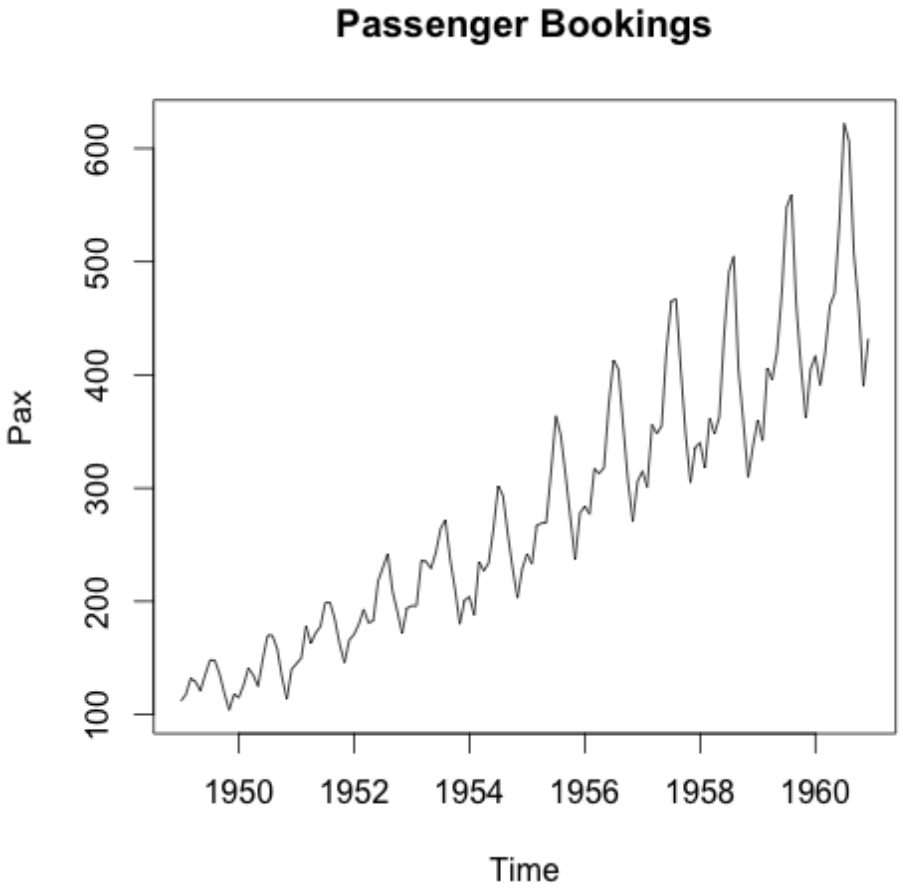
$w_t = \{\ln(y_t)\}\ \mathrm{t} = 1, 2, \ldots, \mathrm{T}$ i.e. $w_1 = \ln(\mathrm{y}_1)\ ;\ w_2 = \ln(\mathrm{y}_2); \ldots; w_n = \ln(\mathrm{y}_\mathrm{T})$

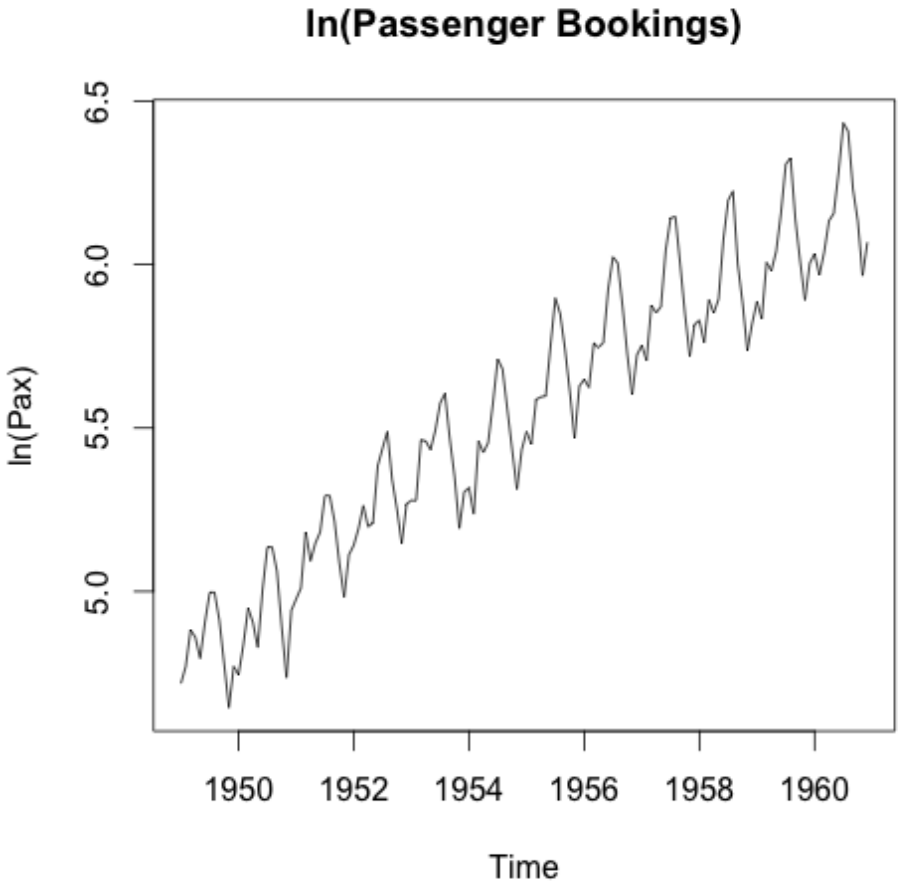
- OR we could apply a square root transformation to the time series:

$w_t = \{\sqrt{y_t}\}\ \mathrm{t} = 1, 2, \ldots, \mathrm{T}$ i.e. $w_1 = \sqrt{y_1};\ w_2 = \sqrt{y_2}; \ldots; w_n = \sqrt{y_T}$

Transformations

a) Stabilising the variance - Example

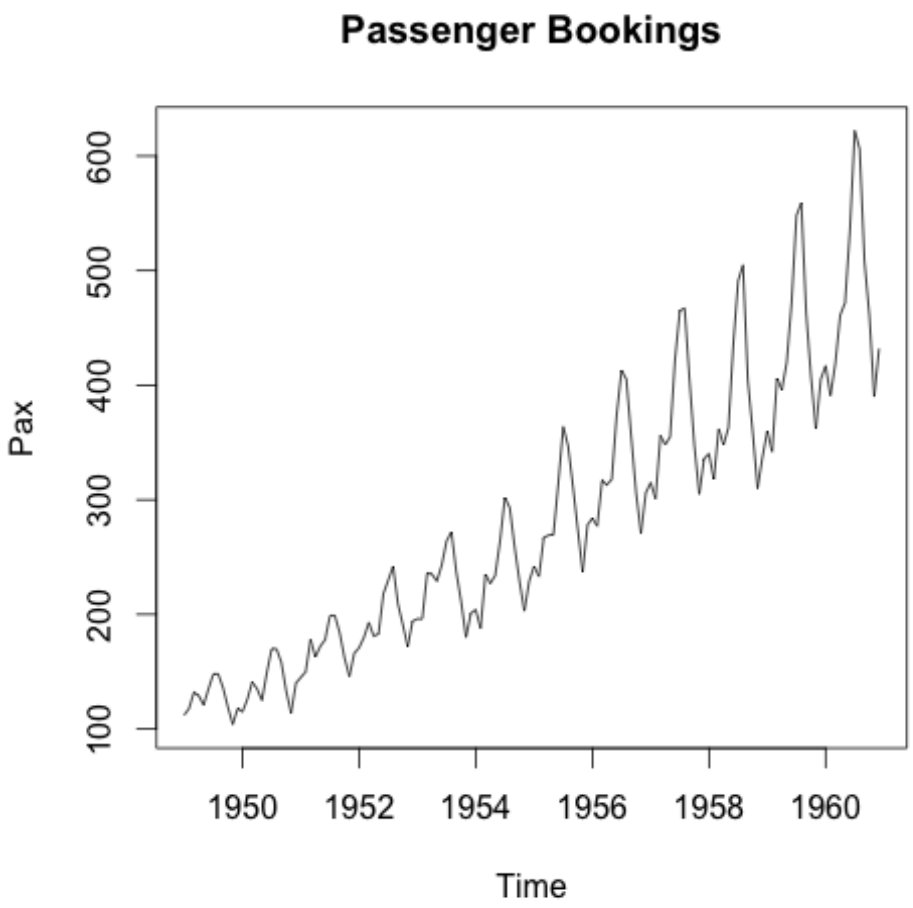


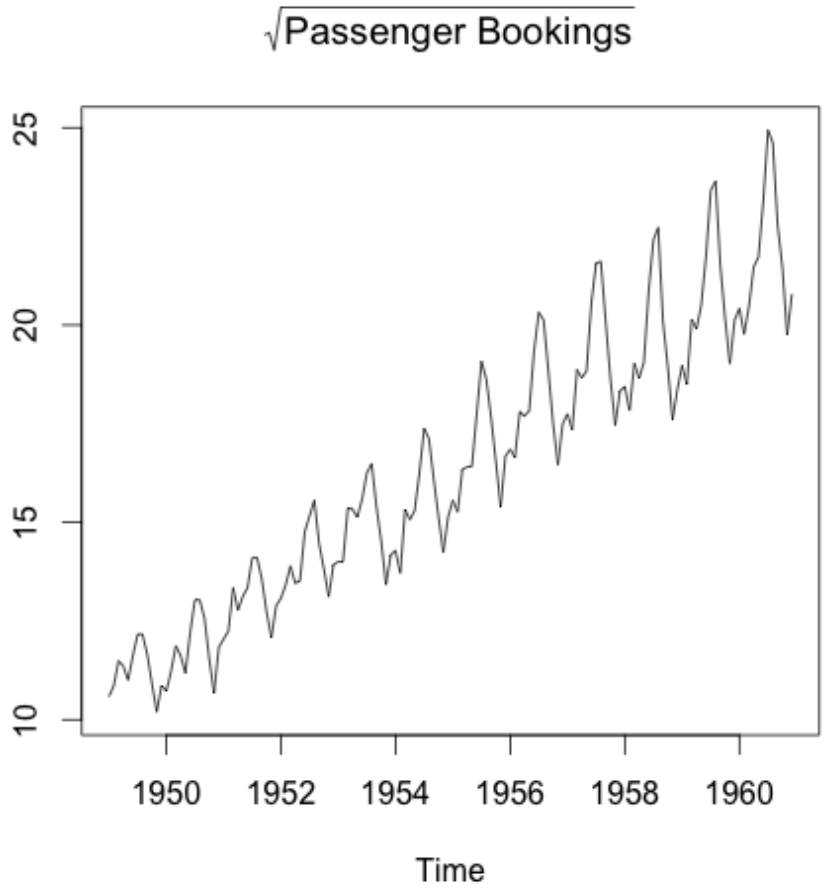


Note that the variance of the transformed Passenger bookings series is much more constant than before.

Transformations

a) Stabilising the variance - Example





Note that the variance of the transformed Passenger bookings series is only a little more constant than before. Hence, taking the natural logarithm of the original series would be a better variance stabilising transformation here.

Transformations

Q: Isn't there a more formal way to decide on what variance stabilising transformation to apply? Well, I'm glad you asked… YES!

The logarithm, square root and cube root variance stabilising transformations form part of a broader family called Box-Cox transformations that depend on the parameter λ:

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

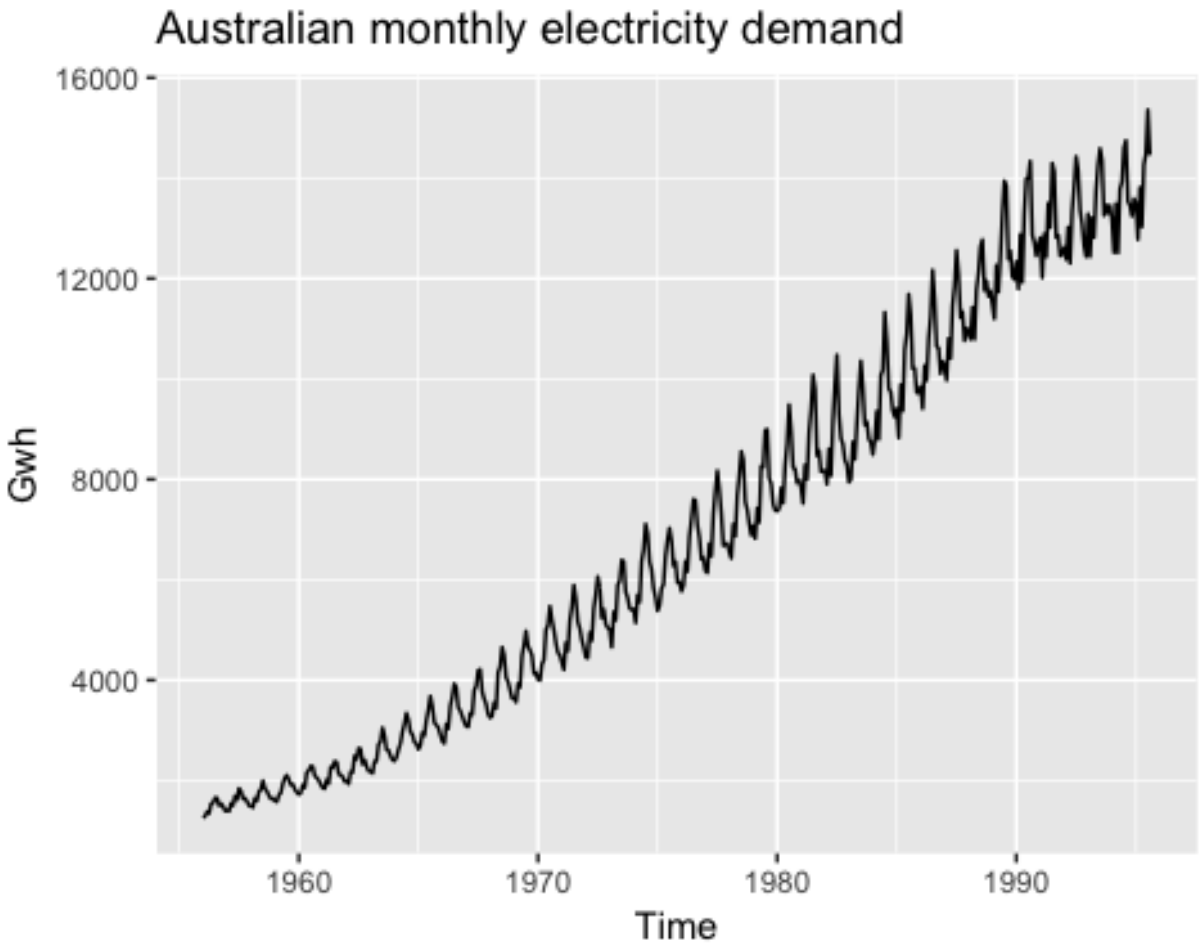
- The Box-Cox transformations allow us to experiment with a wide variety of λ values. A good value of λ is one the makes the variation in the data constant through time.
- In *RStudio*, the BoxCox.Lambda() function will choose a value of λ for you.
- Once you have chosen a transformation, you will forecast the transformed data. Thus, you need to reverse the transformation (or *back-transform*) to obtain forecasts on the original scale. The reverse Box-Cox transformation is given by

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

Transformations

a) Stabilising the variance – *RStudio* Example

```
autoplot(elec) +
  ggtitle("Australian monthly electricity demand") + ylab("Gwh")
(lambda <- BoxCox.lambda(elec))
#> [1] 0.2654
```



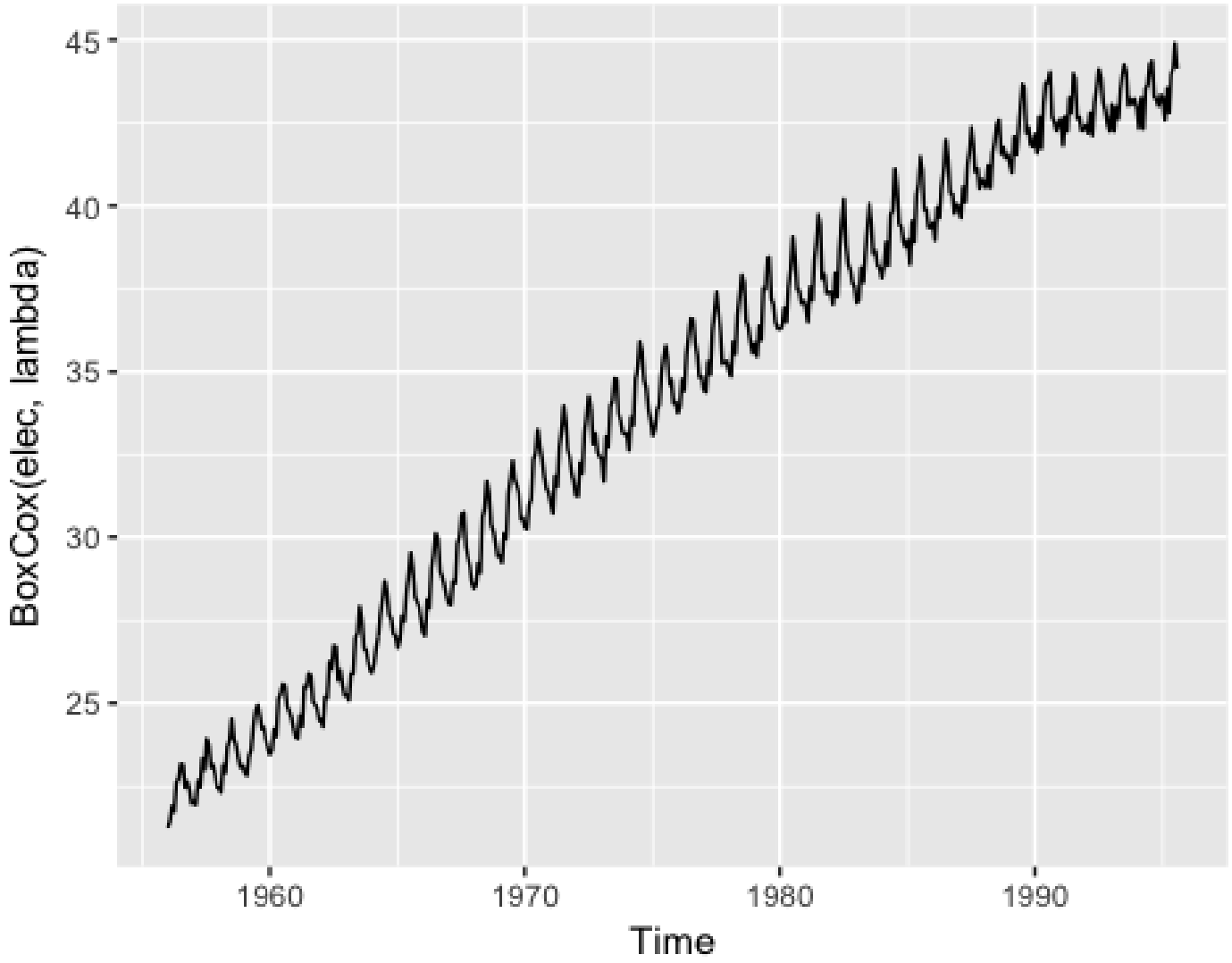
λ = 1 means no variance stabilizing is needed

$\lambda = 0.2654$ results in a transformation that is close to a 4th root transformation

($\lambda = 0$ would be a logarithm transformation;
$\lambda = 0.33$ would be a cube root transformation;
$\lambda = 0.5$ would be a square root transformation)

Transformations

```
autoplot(BoxCox(elec,lambda))
```



The variance is now constant. Constant variance does not mean that it is EXACTLY the same throughout, just that there are no obvious changes in variance over time

Transformations

b) Detrending via differencing ()

- If a time series has a trend, it is non-stationary because the mean will be changing with time (because of the trend).

- If we observe only a linear trend and random variation (i.e. no seasonality), and the variance of the time series appears to be constant, then we can "difference" the series to remove the trend. To achieve this:

- **First-order differencing at lag 1 is performed**, as the name implies, as:

$$w_t = y_t - y_{t-1} \qquad t = 2, 3, \ldots, T$$

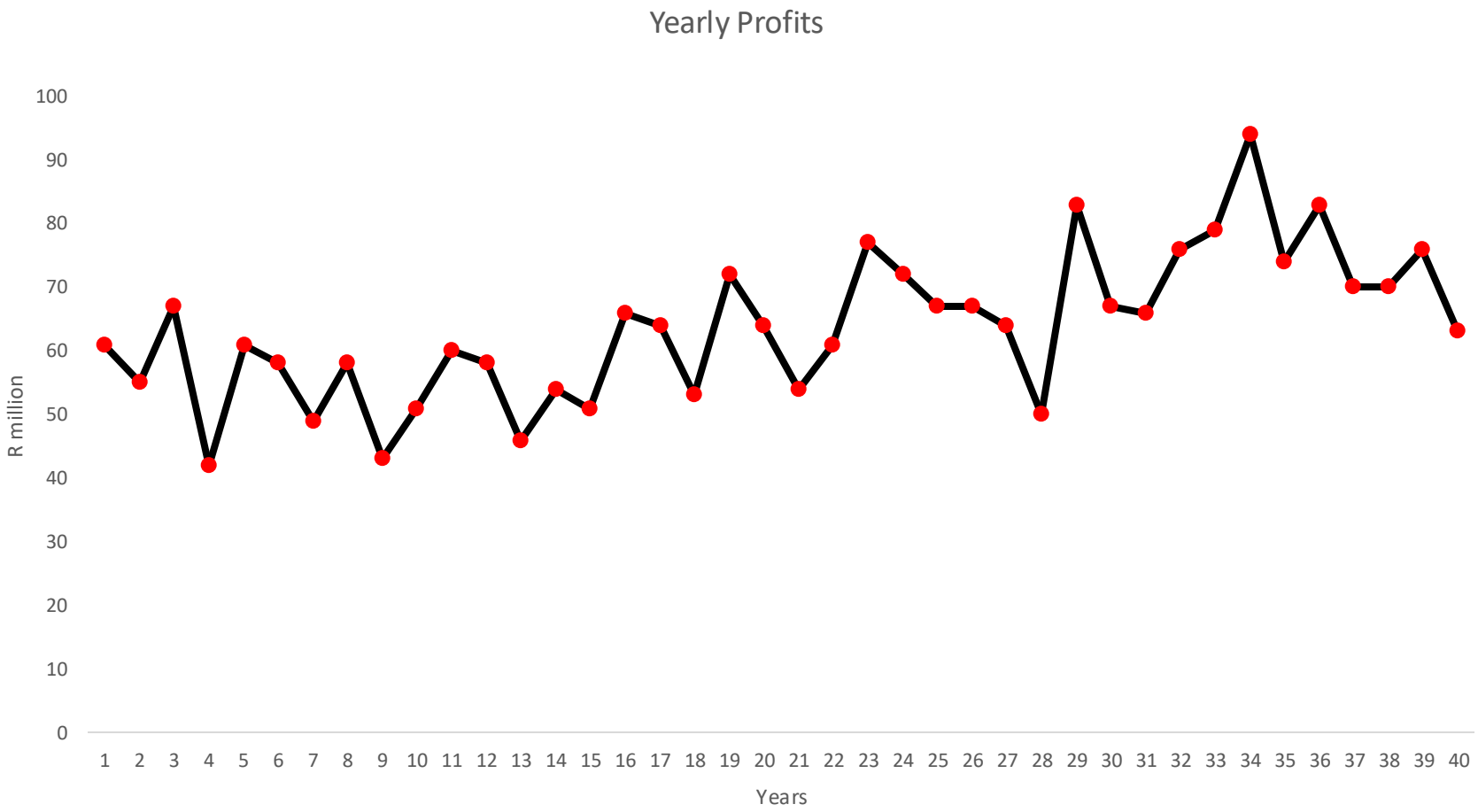
  i.e. subtract the previous value from the current value. You will lose one data point in the process as the difference for the first observation cannot be computed.

- Second differences, i.e. differences of the differences, can be used to remove a quadratic trend. Rarely is more than second differencing performed in practice.

- **Note that since we do not consider non-linear trend in this course,** we won't perform more than one order of differencing.

Transformations

b) Detrending via differencing - Example



There is an increasing linear trend here so the mean is not constant and the time series is not stationary. We can remove the linear trend with first-order differencing at lag 1:

Differenced data series, $w_t = y_t - y_{t-1} \quad t = 2, 3, \ldots, T$ (see next slide)

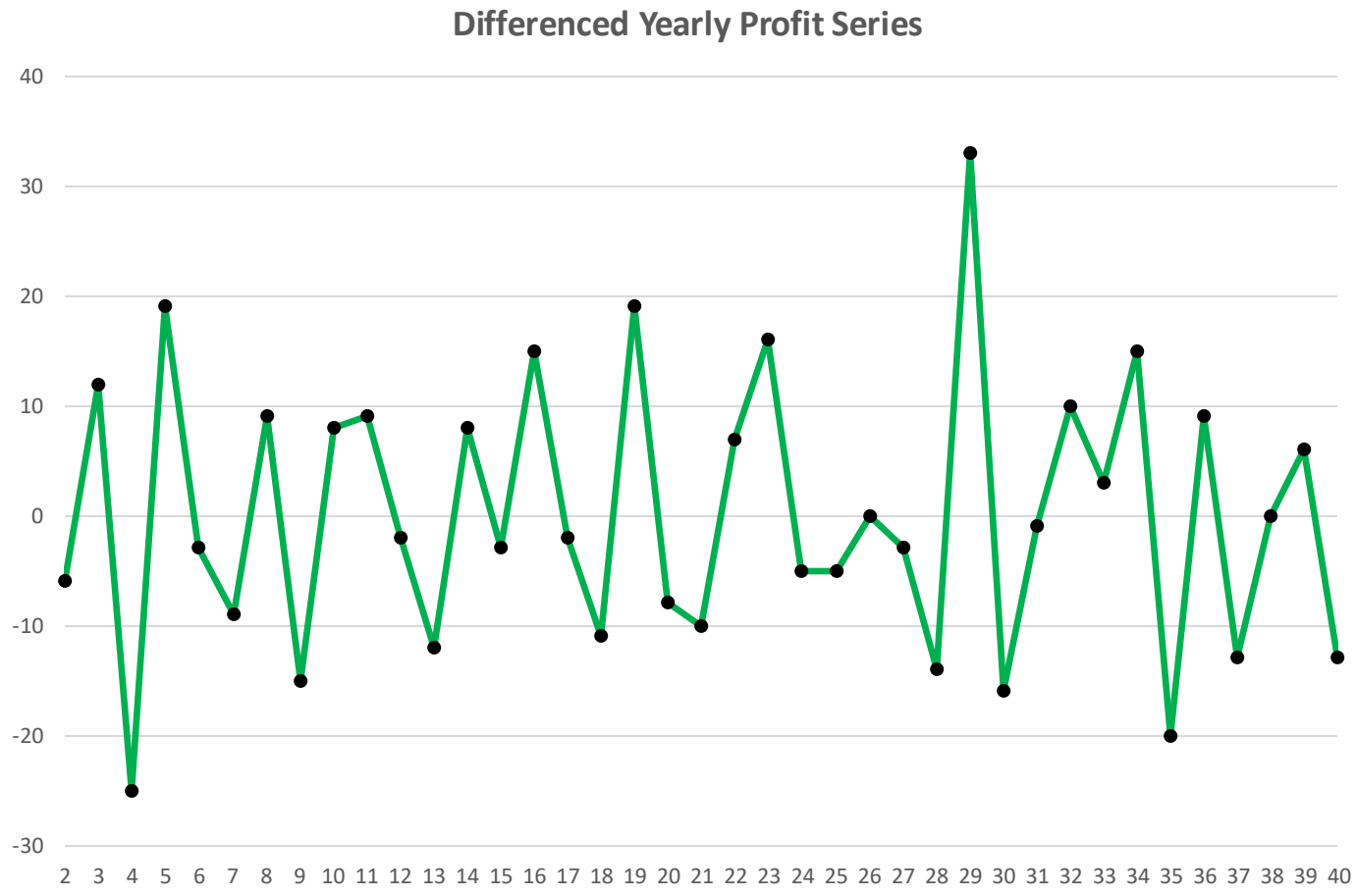
Transformations

b) Detrending via differencing - Example

$w_t = y_t - y_{t-1} \quad t = 2, 3, \dots, T \qquad w_3 = y_3 - y_2 = 67 - 55 = 12$

$\vdots$

$w_2 = y_2 - y_1 = 55 - 61 = -6 \qquad w_{40} = y_{40} - y_{39} = 63 - 76 = -13$

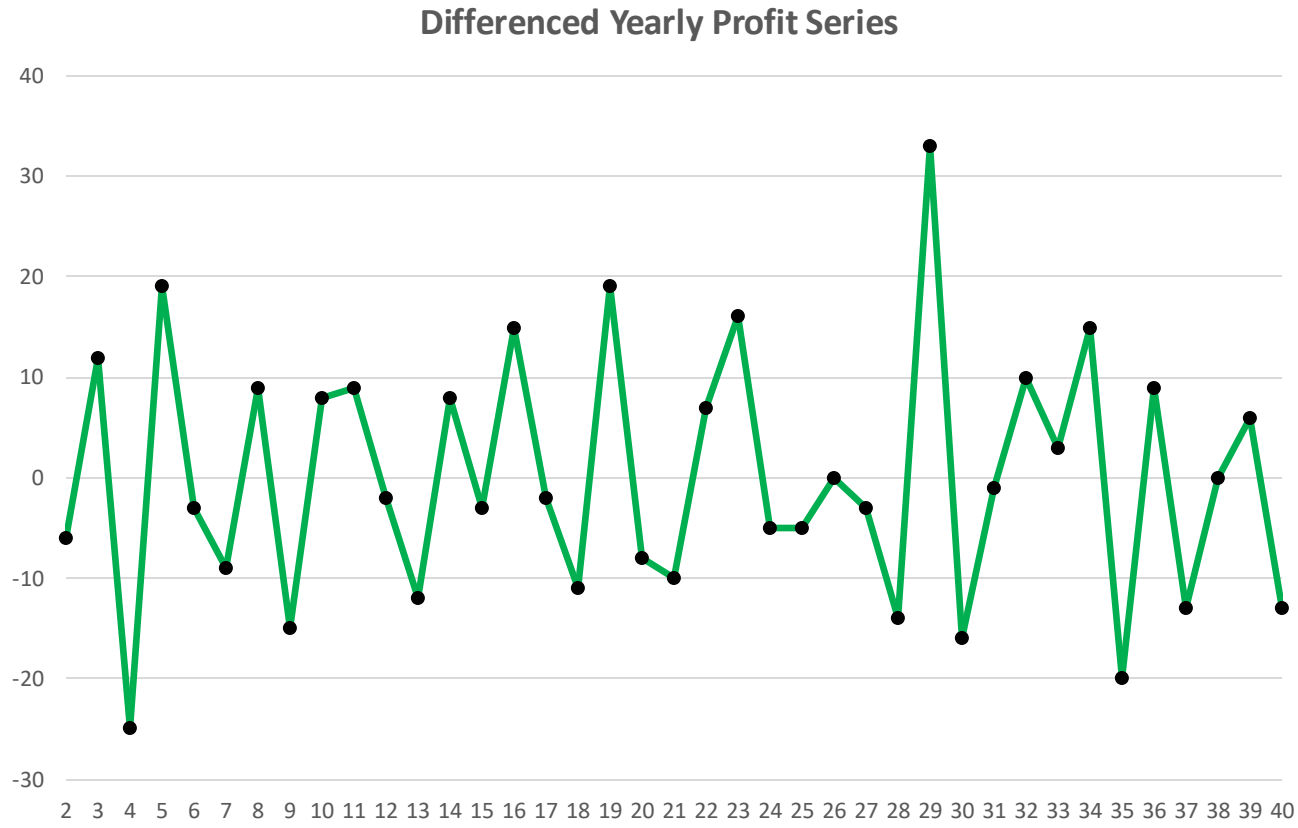


Year	Yearly'Profit	Diff(Yearly'Profit)
1	61	
2	55	%6
3	67	12
4	42	%25
5	61	19
6	58	%3
7	49	%9
8	58	9
9	43	%15
10	51	8
11	60	9
12	58	%2
13	46	%12
14	54	8
15	51	%3
16	66	15
17	64	%2
18	53	%11
19	72	19
20	64	%8
21	54	%10
22	61	7
23	77	16
24	72	%5
25	67	%5
26	67	0
27	64	%3
28	50	%14
29	83	33
30	67	%16
31	66	%1
32	76	10
33	79	3
34	94	15
35	74	%20
36	83	9
37	70	%13
38	70	0
39	76	6
40	63	%13

Transformations

b) Detrending via differencing - Example

- We can see that the differenced series does not appear to have an upward or downward trend. The variance of the differenced series is also relatively constant. We would thus say that this series looks stationary



IF the trend was perfectly linear, then this graph would just be a straight line with value m in every time period, i.e.

$$w_t = y_t - y_{t-1} = m \; \forall t$$

Q: Why is this not so?

Note: You won't be asked to difference an entire time series, but you need to know what it means to difference a series and *when* and *why* you should difference a time series.

Transformations

c) Removing Seasonality by Differencing

- For many time series seasonal effects are very common. If a time series has seasonality, it is non-stationary because the mean will be changing with time due to the seasonality.

- Using an appropriate form of differencing, it is possible to remove these, as well as potential trends. **We take first-order differences with lag g:**

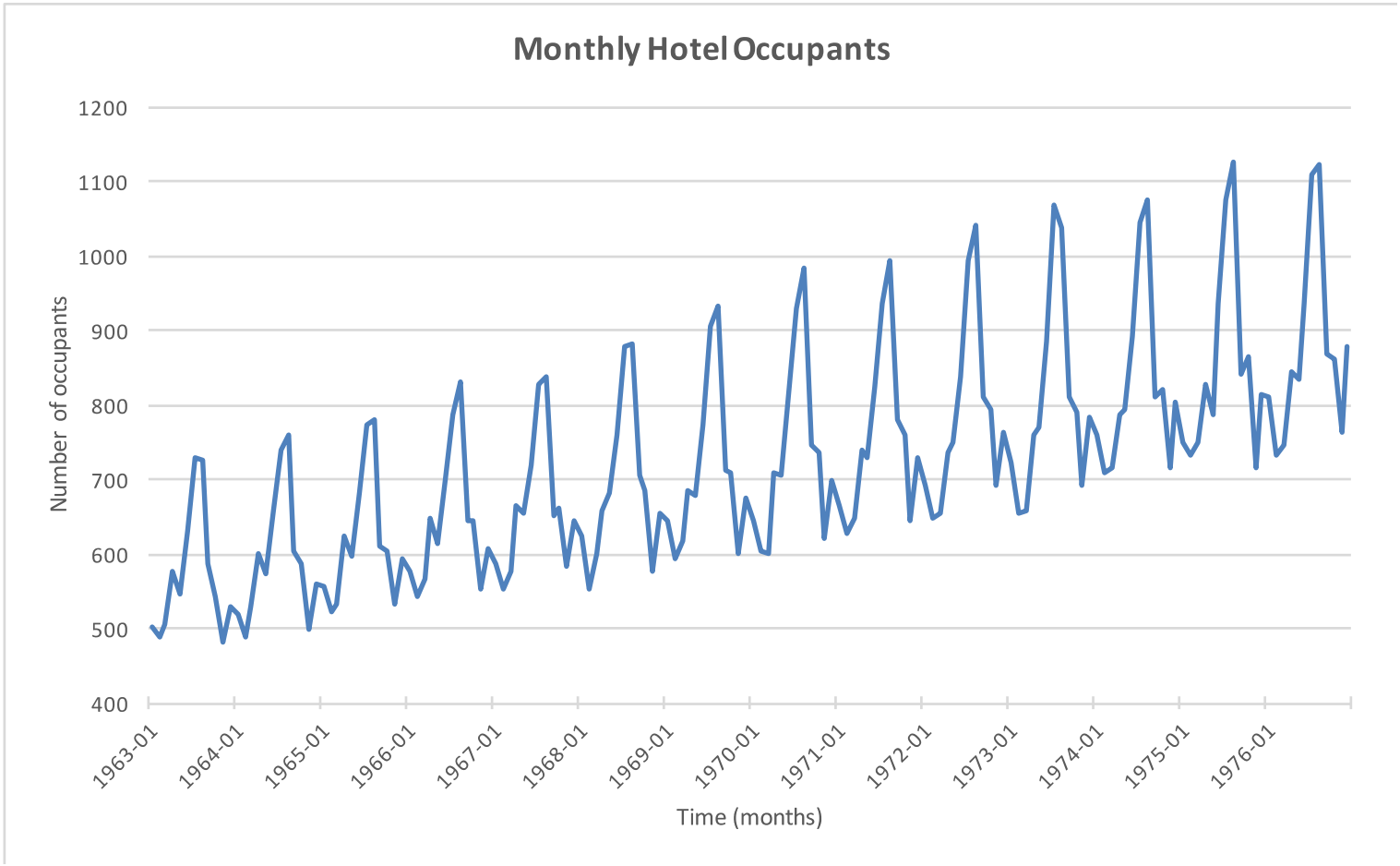
$$w_t = y_t - y_{t-g} \quad t = g+1, g+2, \dots, T$$

- **Here, g is the length of the seasonal variation. i.e. the number of time periods or 'seasons' in one seasonal variation.** The series Y_t then is made up of the changes compared to the previous period's value, e.g. the previous year's value. *Also, from the definition, it is evident that not only the seasonal variation, but also a strictly linear trend will be removed.*

- Note that, similarly to before, there are no differenced values for the first g periods of the series, so you lose g values of data when applying the seasonal differencing.

Transformations

c) Removing Seasonality by Differencing - Example

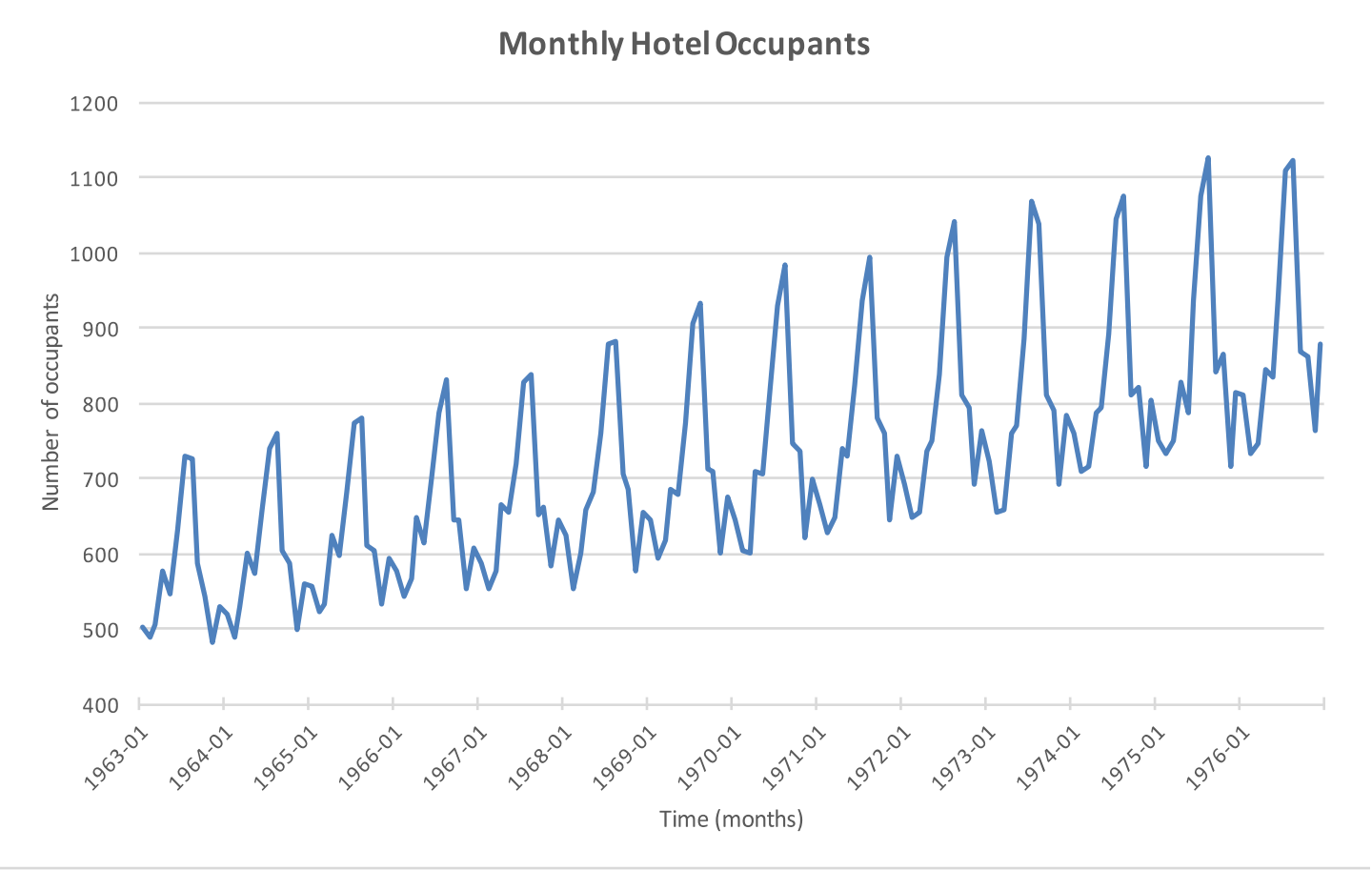


Note: the x-axis here only shows the label for the start of each year, but the time series is recorded **every month** between January 1963 and December 1976

QUESTION: DO WE NEED TO APPLY A VARIANCE STABILISNG TRANSFORMATION?

Transformations

c) Removing Seasonality by Differencing - Example



It appears as if the seasonal variation is one year (12 months in length) i.e., there are 12 periods or 12 'seasons' in one seasonal variation in this time series.

We can also always examine the data itself to guide our choice of the length of the seasonal variation in periods, g, which will be the lag at which we apply our difference:

Transformations

c) Removing Seasonality by Differencing - Example

	Month											
Year	*Jan*	*Feb*	*Mar*	*April*	*May*	*Jun*	*Jul*	*Aug*	*Sep*	*Oct*	*Nov*	*Dec*
1963	501	488	504	578	545	632	728	725	585	542	480	530
1964	518	489	528	599	572	659	739	758	602	587	497	558
1965	555	523	532	623	598	683	774	780	609	604	531	592
1966	578	543	565	648	615	697	785	830	645	643	551	606
1967	585	553	576	665	656	720	826	838	652	661	584	644
1968	623	553	599	657	680	759	878	881	705	684	577	656
1969	645	593	617	686	679	773	906	934	713	710	600	676
1970	645	602	601	709	706	817	930	983	745	735	620	698
1971	665	626	649	740	729	824	937	994	781	759	643	728
1972	691	649	656	735	748	837	995	1040	809	793	692	763
1973	723	655	658	761	768	885	1067	1038	812	790	692	782
1974	758	709	715	788	794	893	1046	1075	812	822	714	802
1975	748	731	748	827	788	937	1076	1125	840	864	717	813
1976	811	732	745	844	833	935	1110	1124	868	860	762	877

Examination of the data confirms that the length of the seasonal variation is 12 months, with the low point in November and the high point in August of the following year. Hence, a choice of g = 12 is appropriate for the lag at which we conduct the differencing:

$$w_t = y_t - y_{t-g} \quad t = g+1, g+2, \ldots, T$$

$$w_t = y_t - y_{t-12} \quad t = 13,14, \ldots, 156$$

$$w_{13} = y_{13} - y_1 = 518 - 501 = 17$$

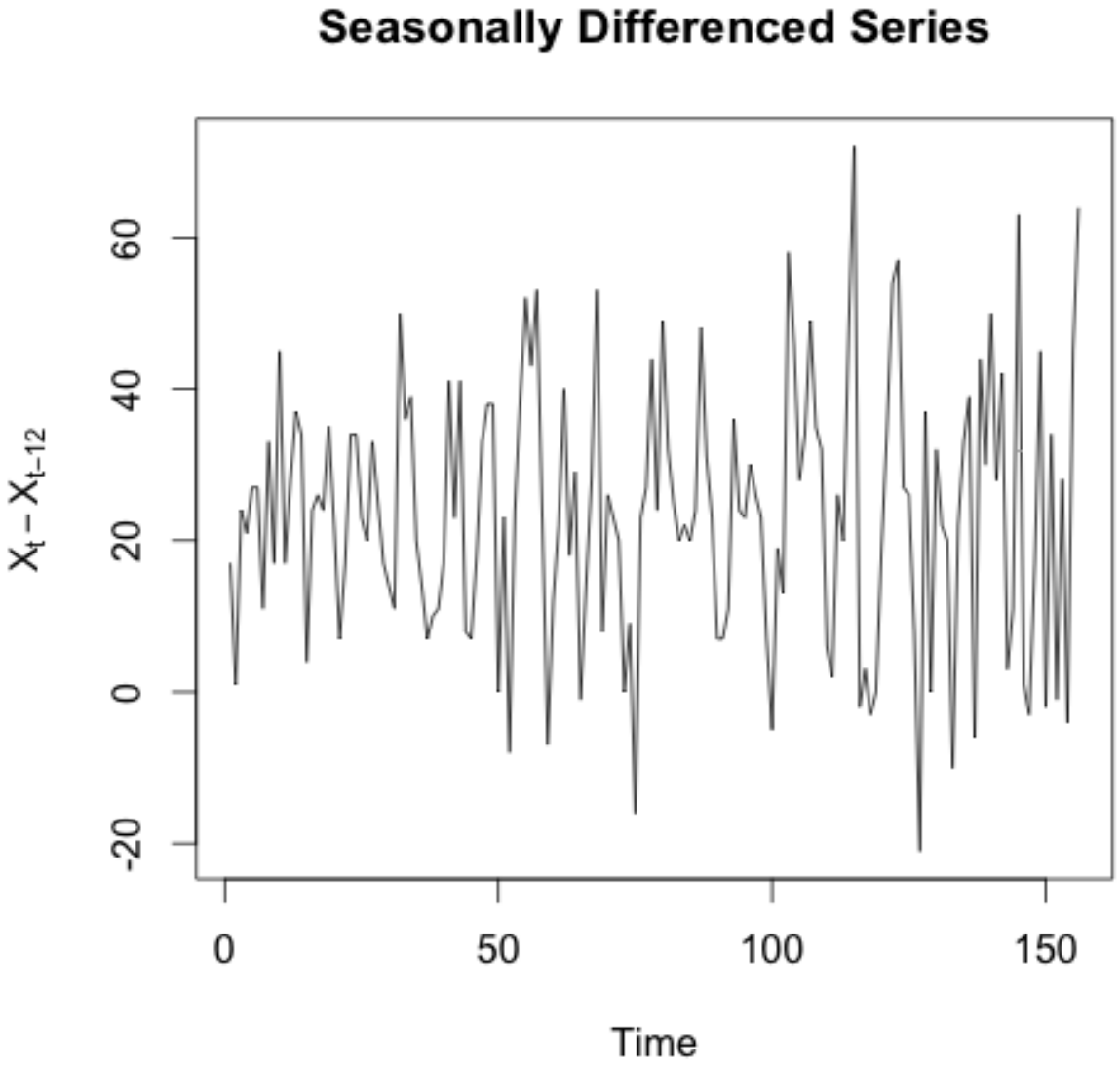
$$w_{14} = y_{14} - y_2 = 489 - 488 = 1$$

$$\vdots$$

$$w_{156} = y_{156} - y_{144} = 877 - 813 = 64$$

Transformations

c) Removing Seasonality by Differencing - Example



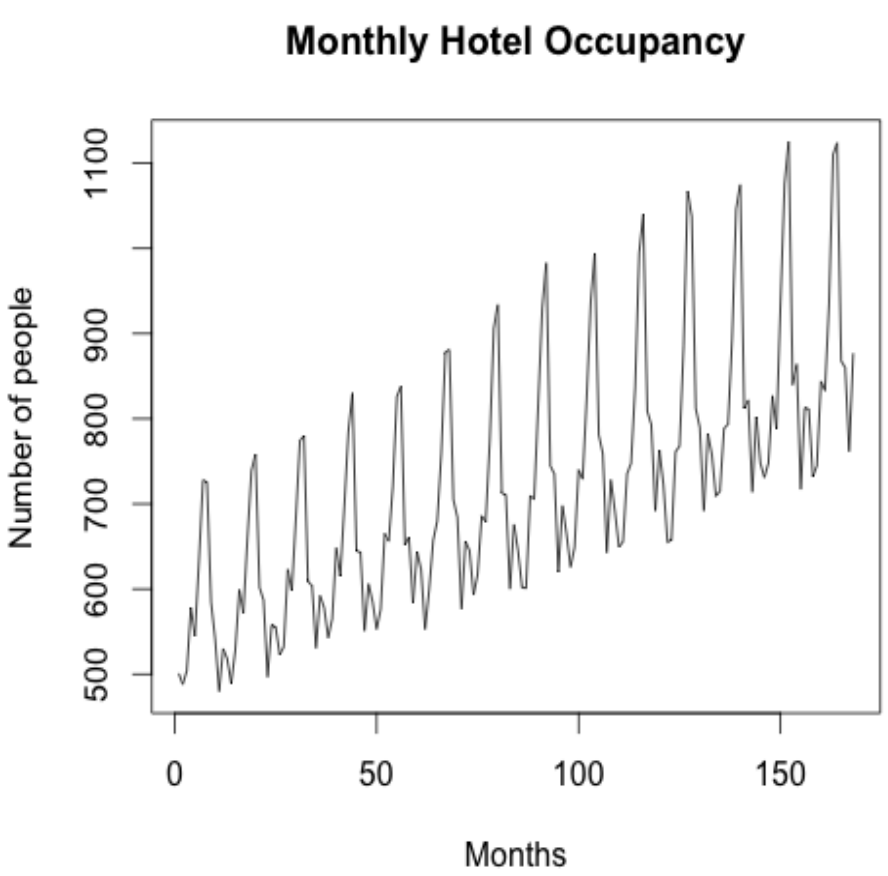
A first-order differencing at lag g = 12, the length of a seasonal variation, has removed both the seasonality and the trend…

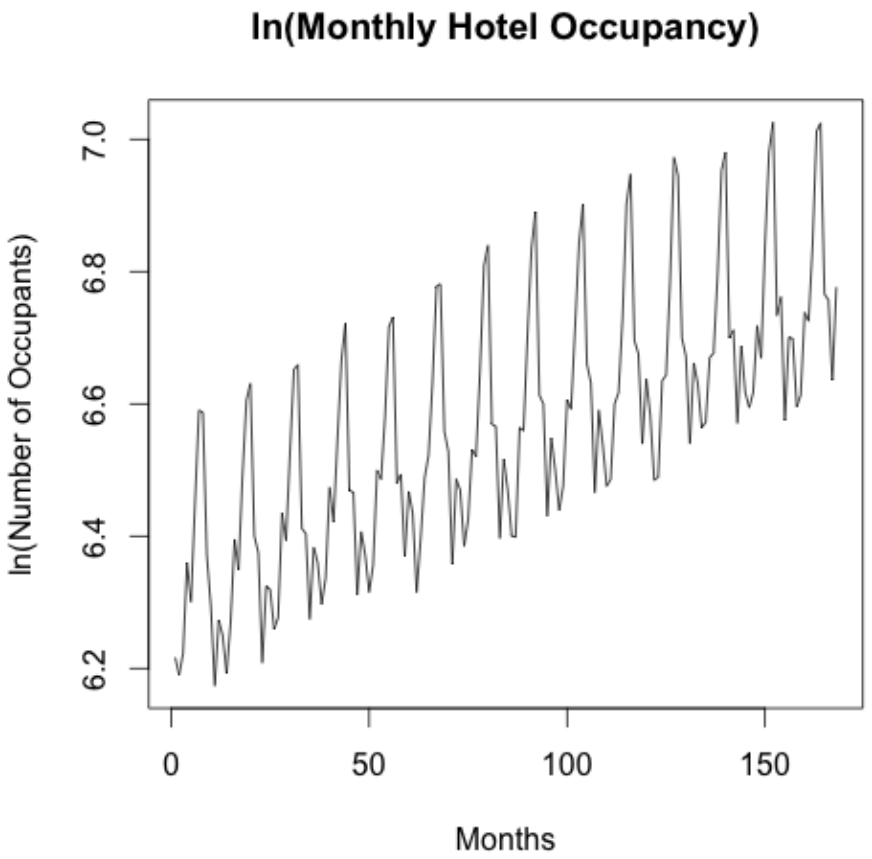
Q: But is the seasonally differenced series stationary in variance?

It appears that the variance is slightly increasing over time. Hence, a better approach may have been to first apply a variance stabilising transformation to the data before taking seasonal differences:

Transformations

c) Removing Seasonality by Differencing - Example

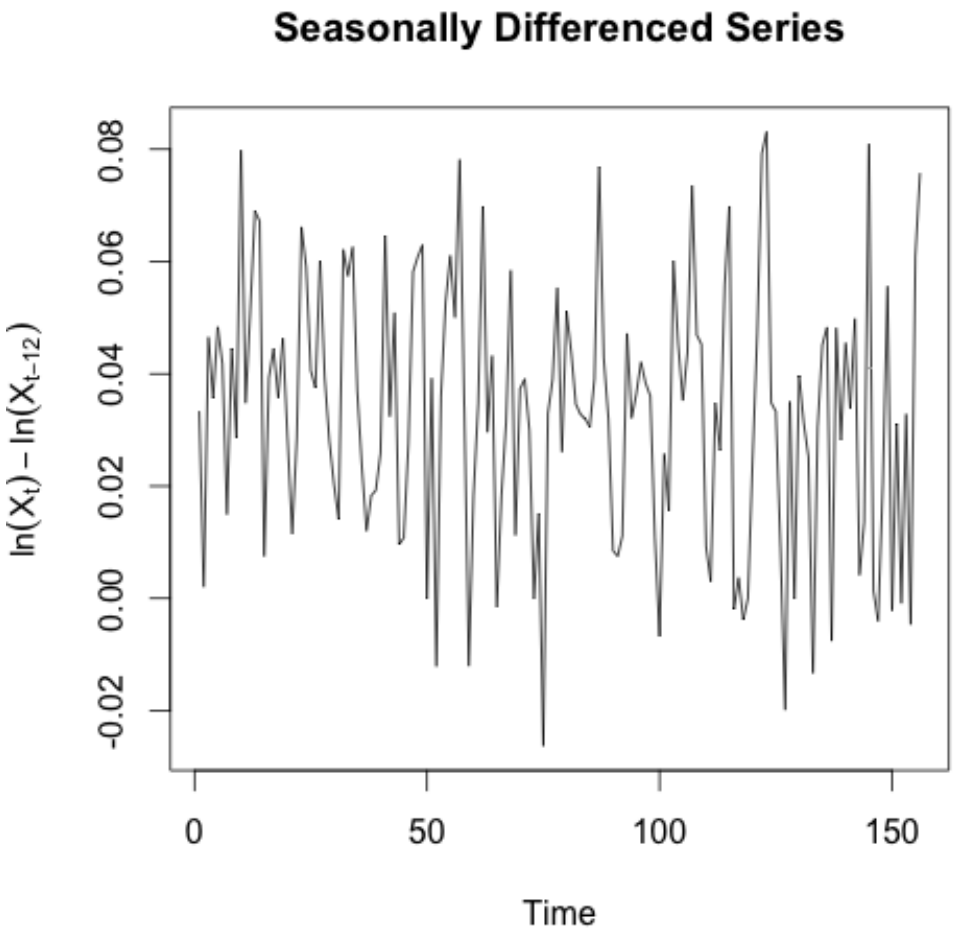




Applying logarithms to the original series does appear to stabilise the variance of the series somewhat. We can now proceed to seasonally difference the logged series in order to remove the seasonality and trend, as before with lag g = 12:

Transformations

c) Removing Seasonality by Differencing - Example



The seasonally differenced logged data appears to be stationary in its mean AND variance, and thus looks stationary.

Hence, by applying two transformations to the data:

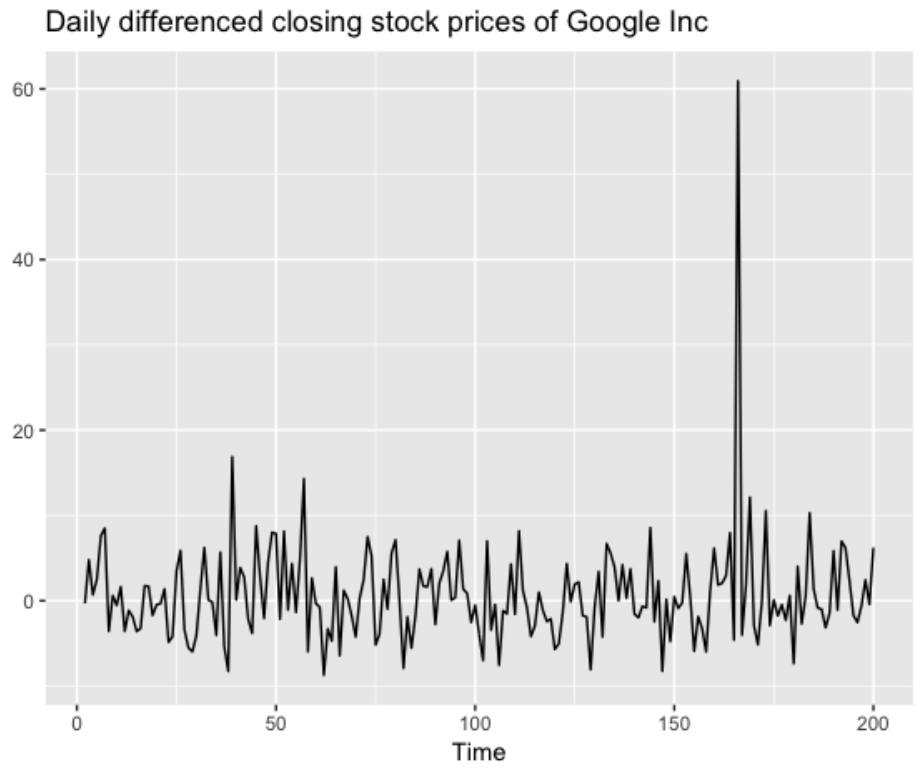
- Logarithms to stabilise the variance
- Differencing at lag g = 12 to remove the seasonality and trend

We have successfully transformed the data from one that we know is non-stationary into one that appears to be stationary

Transformations

Daily closing stock price of Google Inc. – Example of first-order differencing at lag 1

```
autoplot(goog200) + ggtitle("Daily closing stock prices of Google Inc") + ylab("")
autoplot(diff(goog200)) + ggtitle("Daily differenced closing stock prices of Google Inc") + ylab("")
```

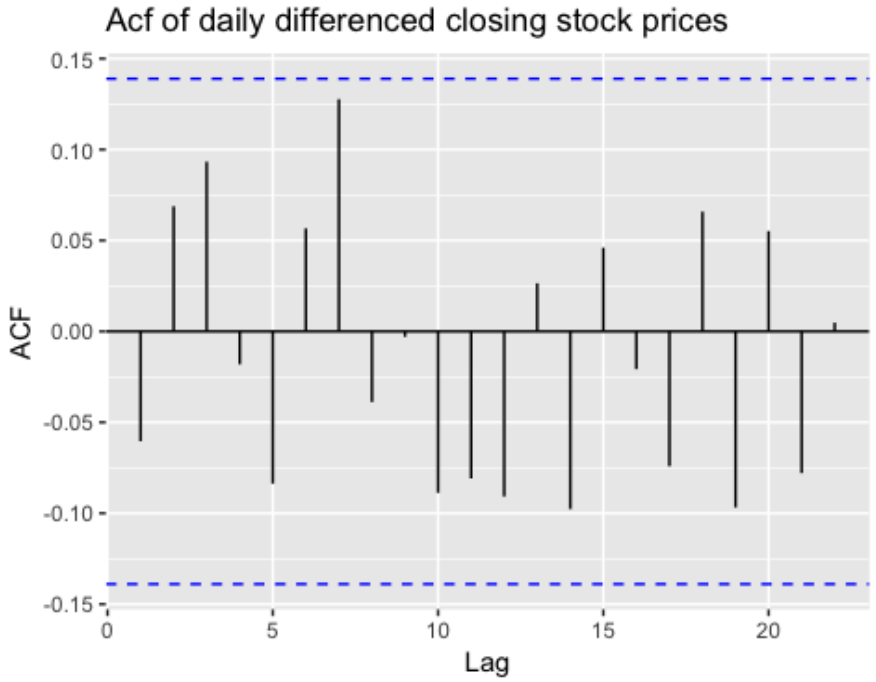


First order differencing at lag 1 has successfully removed the linear trend from the data. There is no pattern in the differenced data and the variance is relatively constant (other than the outlier value), so it is stationary.

Transformations

Daily closing stock price of Google Inc. – Example of first-order differencing at lag 1

```
ggAcf(goog200) + ggtitle("Acf of daily closing stock prices")
ggAcf(diff(goog200)) + ggtitle("Acf of daily differenced closing stock prices")
```



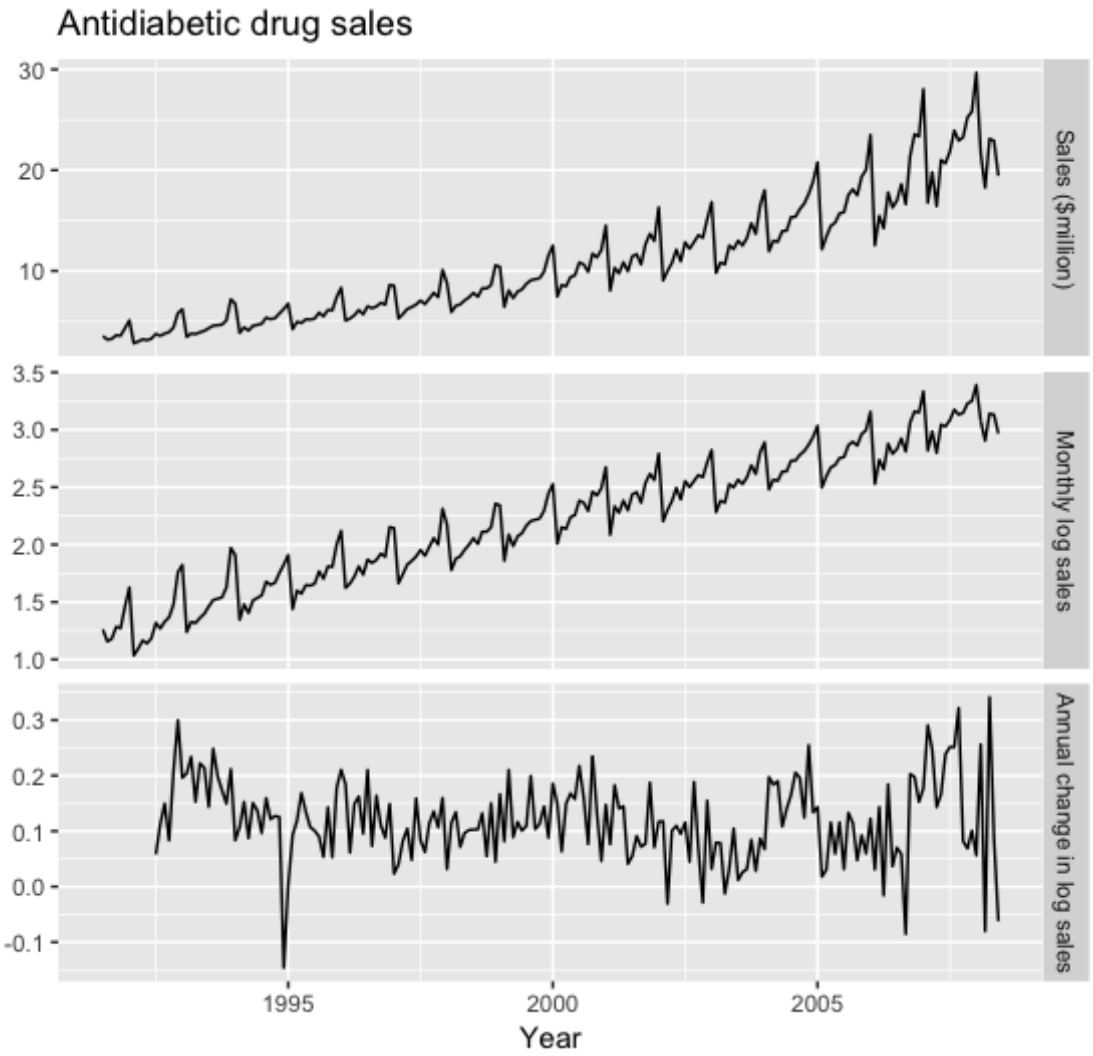
The ACF plot (aka correlogram) shows the values of the sample autocorrelations at each lag. In the left-had plot, the autocorrelation at lag 1 is very strong and positive (approx. 0.95). At lag 20, the autocorrelation is about 0.43 (so moderate and positive). We would say that observations 20 time periods apart have a moderate positive relationship.

The dashed blue lines represent the upper and lower limits of a 95% confidence interval of the hypothesis that there is no autocorrelation. So, those autocorrelation values that lie between the dashed blue lines are likely 0. The plot on the right shows that there is no autocorrelation at any lag, all the way up to lag 22. Since autocorrelation represents information about potential patterns in the data the plot on the right indicates that there is likely no information/pattern left in the differenced stock prices. What does this imply about the differencing transformation? A plot like the one on the right is what you want to see in the ACF plot of the residuals of a fitted model. Can you explain why?

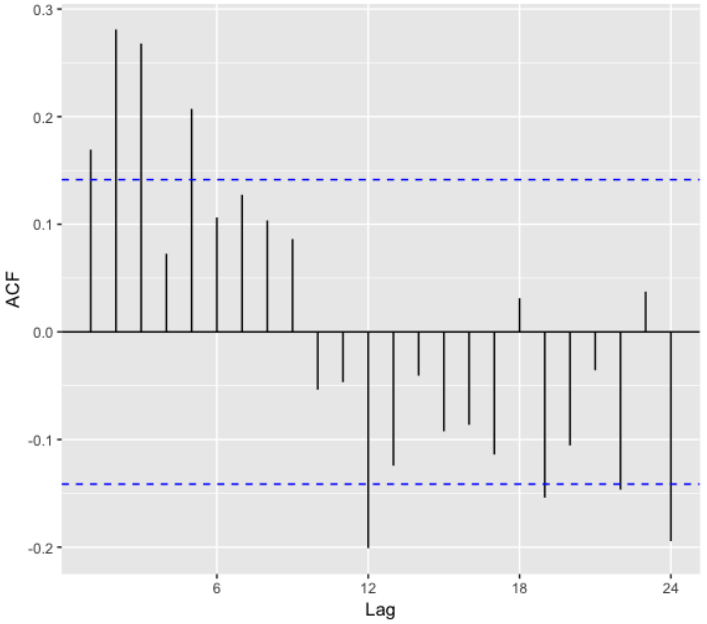
Transformations

Antidiabetic drug sales – Example of first-order seasonal differencing at lag g

```
cbind("Sales ($million)" = a10,"Monthly log sales" = log(a10),"Annual change in log sales" = diff(log(a10),12)) %>%
  autoplot(facets=TRUE) +  xlab("Year") + ylab("") +  ggtitle("Antidiabetic drug sales")
```




```
a10 %>% log() %>%
  diff(12) %>%
  ggAcf()
```



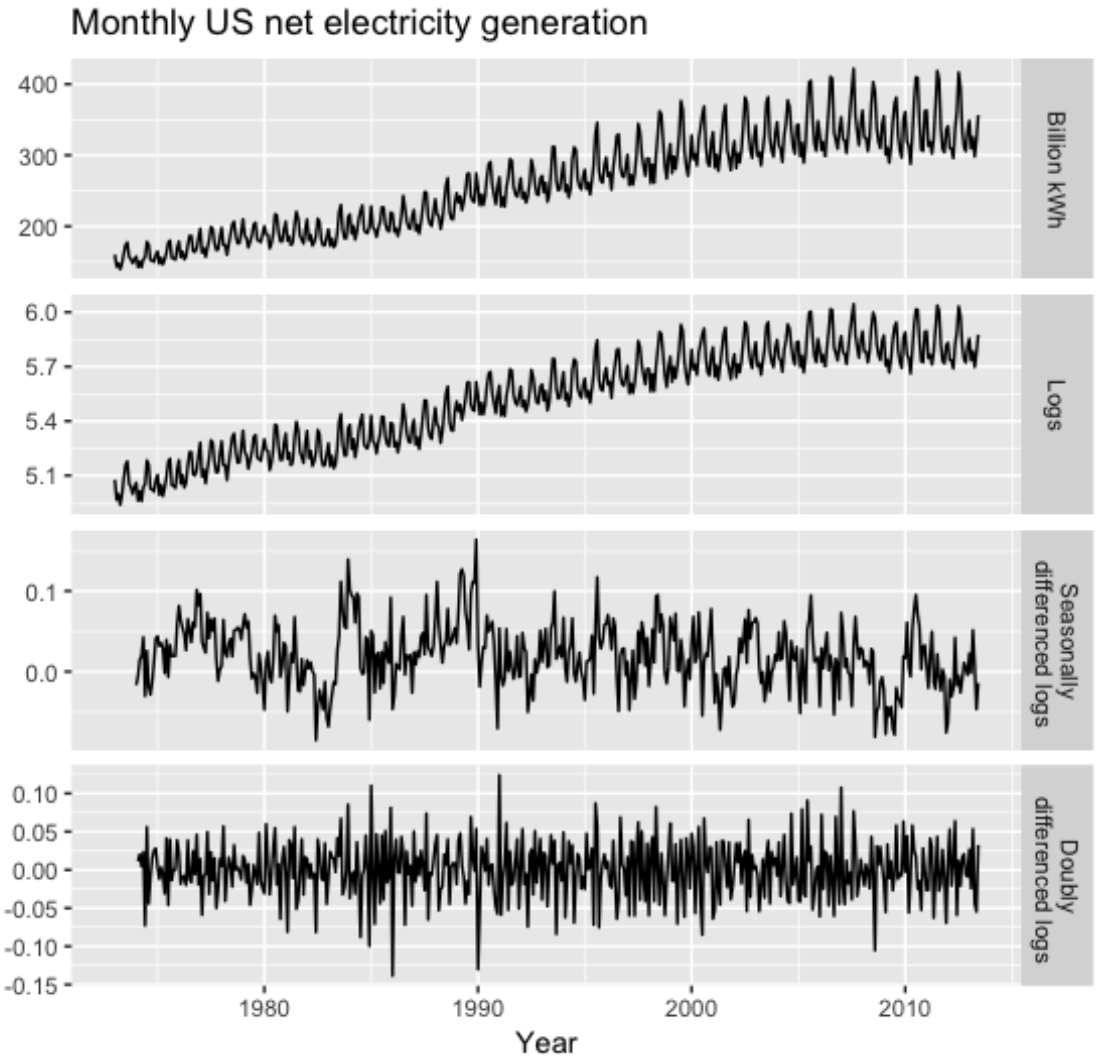
There is autocorrelation in the transformed data at multiple lags.

What does this mean?

Transformations

Monthly US net electricity generation – Example of both seasonal and first differences

```
cbind("Billion kWh" = usmelec, "Logs" = log(usmelec), "Seasonally\n differenced logs" = diff(log(usmelec),12),
      "Doubly\n differenced logs" = diff(diff(log(usmelec),12),1)) %>%
  autoplot(facets=TRUE) + xlab("Year") + ylab("") + ggtitle("Monthly US net electricity generation")
```



Transformations

Q: How do we know how many orders of differencing to perform?

(a) By inspecting the time series and correlogram plots

- Before differencing, examine a plot of the data. A linear increase or decrease will suggest that a first difference is appropriate. In the absence of any obvious order of differencing, look at the correlogram.
- A "nice" correlogram decays fairly rapidly to zero, either from above or below. If the autocorrelations are positive out to a high number of lags (e.g. 10 or more), then it likely needs a differencing. ***If the lag 1 autocorrelation is zero or even negative, then the series does not need further differencing.***
- One of the most common errors in time series modelling is "over-differencing". ***As a rule of thumb if the first lag autocorrelation is more negative than -0.5 this may mean the series has been over-differenced.***
- Finally, another symptom of over-differencing is an increase in the standard deviation, rather than a reduction, when the order of differencing is increased. *A common rule of thumb is that you stop differencing when the standard deviation is at its lowest*.

Transformations

Q: How do we know how many orders of differencing to perform?

(b) By conducting a unit root test of the stationarity of the data

- If stationarity is a hypothesis that is tested on data, are there are any formal tests we can conduct? The answer to this question is yes – we can conduct a unit root test which is a statistical hypothesis test of stationarity that is designed for determining whether differencing is required.

- The *Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests whether a time series is stationary. The null hypothesis is that the time series is stationary, and the alternative that it is not stationary. The null hypothesis is rejected for sufficiently small p-values.*

- *There are two useful R functions to help in deciding how many differences are needed:*
  - *ndiffs()*, which uses a sequence of KPSS tests to determine the appropriate number of first differences required for a non-seasonal time series.
  - *nsdiffs()*, to determine the appropriate number of seasonal differences required.

- See ***RStudio* Examples**

Testing Stationarity of a time series

Some important notes on tests for stationarity:

- There are several tests for the stationarity of a time series, each based on different assumptions and often leading to conflicting answers.
- **However**, we will not discuss any more of these tests here for a variety of reasons:
  - First and foremost, they all focus on some very specific non-stationarity aspects, but do not test stationarity in a broad sense. While they may reasonably do their job in the narrow field they are aimed for, they have low power to detect general non-stationarity and in practice often fail to do so.
  - The theory of these tests is quite complex, and thus beyond the scope of this course.
- ***Thus, we recommend generally assessing stationarity by visual inspection***. The primary tool for this is the time series plot, but the correlogram is helpful as a second check.

Diagnostic Checks of the Residuals

- ***If there is significant dependence among the residuals, then we must use the theory of stationary processes to develop a model for the errors that accounts for the dependence***. Dependence means in that past observations can assist in predicting future values.

- Recall: The residual sequence, $\hat{\epsilon}_t$ *t = 1, 2, …, T*, in the time series model is obtained by subtracting the fitted values from the observed values in each time period. The residuals represent the true error (noise) sequence i.e. they represent the random variation component R_t

- There are several methods we can use to analyse the residuals. The main objective is to determine whether these residuals can be regarded as a sequence of independent, identically distributed random variables (*i.e. white noise*), or if they exhibit dependence (autocorrelation), since this means that the model has captured the relevant information/patterns in the data. They should:
  - **i.** **Have no autocorrelation**. *If there is autocorrelation present, it means that there is information left in the residuals that represents potential patterns in the data.* ***This means that the model has not captured all the relevant information/patterns in the data.***
  - **ii.** **Have zero mean**. If they do not have zero mean, the forecasts will be biased
  - iii. Have constant variance
  - iv. Are normally distributed

Properties (i) and (ii) are more important than (iii) and (iv). If the residuals of a model do not satisfy (i) and (ii), that model can be improved. Properties (iii) and (iv) make the calculation of prediction intervals easier

Diagnostic Checks of the Residuals

QUESTION: How do we go about checking the residuals?

1. *A first step in analysing the residuals is to plot them*

We look for the presence/absence of any discernible pattern/trend as well as how "smooth" the plot looks. In particular, if there are long stretches of residuals that have the same sign. This would be very unlikely if the residuals were observations of a white noise series with zero mean (i.e. that they are Independent and Identically Distributed (iid))

2. Tests for normality

If there is evidence that the data are generated by normal random variables, one can create the QQ plot to check for normality. One can also plot a histogram of the residual data to see if they follow a normal distribution. Both these tools are based on a visual inspection of the residuals.

3. Examine the correlogram for presence of significant autocorrelation at any lag

The dashed blue lines represent 95% confidence intervals of the hypothesis that there is no autocorrelation in the data (in this case, the residuals). This means that, by chance, 1 in 20 of the autocorrelations could be slightly outside these lines. However, one wants to see ALL of the autocorrelations within the two lines as this is interpreted as there being no autocorrelation in the residuals. **Q: If the correlogram shows autocorrelation in the residuals, what does this imply about your fitted model?**

Diagnostic Checks of the Residuals

4. The Ljung-Box Test

The Ljung-Box test is based on the test statistic:

$$Q_{LB}(m) = T(T+2) \times \sum_{g=1}^{m} \frac{\hat{\rho}(g)^2}{T-g}$$

- T is the length of the time series, $\hat{\rho}(g)$ are the sample autocorrelation coefficients at lag h and m is the lag up to which the test is performed. It is typical to use *m* = 1, 3, 5, 10 or 20.

- It tests the joint hypothesis that all the ACF coefficients up to lag m are simultaneously equal to zero, thus implying that the time series is stationary i.e. it evaluates whether there is any significant autocorrelation in a series.

- The null and alternative hypotheses of the test are:

  H_0: There is no autocorrelation present in any of the first m lags
  H_1: There is autocorrelation present in at least one of the first m lags

Diagnostic Checks of the Residuals

QUESTION: How do we go about checking the residuals?

- If you are conducting the Ljung-Box test on the residuals of a fitted model, then Q will have a $\chi 2$ distribution with (m−K) degrees of freedom, where K is the number of parameters in the model. If you are conducting the test on time series data (rather than the residuals from a model), then set K=0 and Q will have m degrees of freedom.

- *Rejection region:* We reject the hypothesis of independent and identically distributed residuals at the level α if $Q > \chi^2_{1-\alpha}(m-K)$ OR if the p-value of the test statistic < α. *i.e. only if the test statistic is sufficiently LARGE. Just like the previous Chi-square goodness-of-fit tests that we have covered in this course, small values of the test statistic serve as evidence that the null hypothesis is TRUE. Only LARGE values of the test statistic serve as evidence that the null hypothesis is FALSE.*

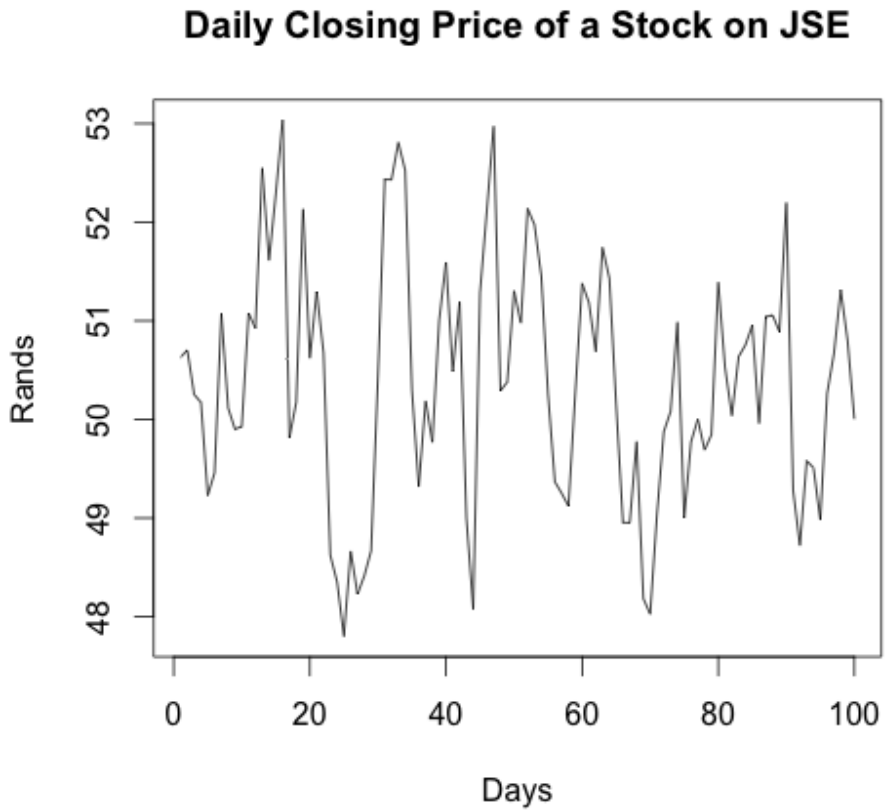
- ***The general strategy in applying the above diagnostic checks*** is to check them all and to proceed with caution if any of them suggests a serious deviation from the iid hypothesis.

- It should be noted that if a model satisfies these checks, it does not mean that it cannot be improved. It is possible to have several different forecasting methods for the same data set, all of which satisfy these properties. Checking these properties is important to see whether a method is using all the available information, but it is not a good way to select a forecasting method. We will consider other tools later that help guide model selection.

Diagnostic Checks of the Residuals

4. The Ljung-Box Test – Example

Consider 100 daily closing prices of a stock on the JSE. The sample autocorrelation coefficients up to lag 15 were computed. Is there any autocorrelation between the daily closing price and any of the previous 5 days' closing prices? Conduct an appropriate test at the 5% significance level.



ACF of Daily Stock Price

Lag	*Correlation*
0	1
1	0.593
2	0.267
3	0.099
4	-0.109
5	-0.116
6	-0.116
7	-0.138
8	-0.129
9	-0.184
10	-0.237
11	-0.177
12	-0.061
13	0.019
14	0.026
15	0.006

Diagnostic Checks of the Residuals

H_0: There is no autocorrelation at any lag up to lag 5 in the data
H_1: There is significant autocorrelation at at least one lag in the first 5 lags

Test statistic: $Q_{LB}(m) = T(T+2) \times \sum_{h=1}^{m} \frac{\hat{\rho}(g)^2}{T-g}$

$$Q_{LB}(5) = 100(102) \times \sum_{p=1}^{5} \frac{\hat{\rho}(g)^2}{100-g}$$

$$= 10200 \times \left(\frac{(0.593)^2}{99} + \frac{(0.267)^2}{98} + \frac{(0.099)^2}{97} + \frac{(-0.109)^2}{96} + \frac{(-0.116)^2}{95} \right)$$

$$= 47.39$$

Critical region: Reject H_0 if $47.39 > \chi^2_{0.05;5}$ OR if p-value < 0.05

ACF of Daily Stock Price

Lag	*Correlation*
0	1
1	0.593
2	0.267
3	0.099
4	-0.109
5	-0.116
6	-0.116
7	-0.138
8	-0.129
9	-0.184
10	-0.237
11	-0.177
12	-0.061
13	0.019
14	0.026
15	0.006

Diagnostic Checks of the Residuals

Critical region:

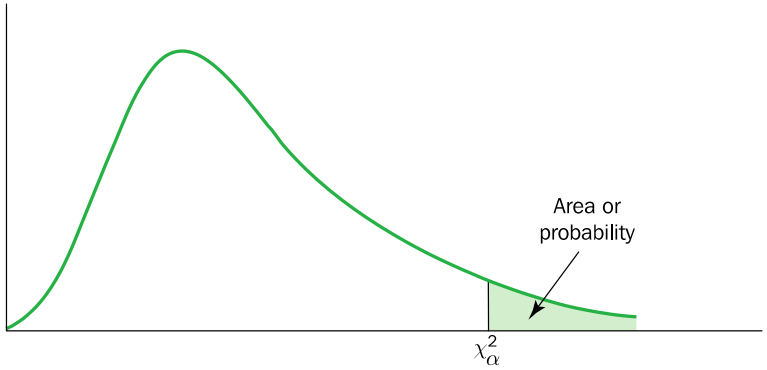
Reject H_0 if $47.39 > \chi^2_{0.05;5}$

OR if p-value < 0.05

$\chi^2_{0.05;5} = 11.070$

p-value of test statistic < 0.005

TABLE 3 Chi-squared distribution



Entries in the table give χ^2_α values, where α is the area or probability in the upper tail of the chi-squared distribution. For example, with ten degrees of freedom and a.01 area in the upper tail, $\chi^2_{0.1} = 23.209$

Degrees of freedom	*Area in upper tail*									
	.995	*.99*	*.975*	*.95*	*.90*	*.10*	*.05*	*.025*	*.01*	*.005*
1	.000	.000	.001	.004	.016	2.706	3.841	5.024	6.635	7.879
2	.010	.020	.051	.103	.211	4.605	5.991	7.378	9.210	10.597
3	.072	.115	.216	.352	.584	6.251	7.815	9.348	11.345	12.838
4	.207	.297	.484	.711	1.064	7.779	9.488	11.143	13.277	14.860
5	.412	.554	.831	1.145	1.610	9.236	11.070	12.832	15.086	16.750
6	.676	.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.647	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188

Diagnostic Checks of the Residuals

H_0: There is no autocorrelation at any lag up to lag 5 in the data
H_1: There is significant autocorrelation at at least one lag in the first 5 lags

ACF of Daily Stock Price	
Lag	*Correlation*
0	1
1	0.593
2	0.267
3	0.099
4	-0.109
5	-0.116
6	-0.116
7	-0.138
8	-0.129
9	-0.184
10	-0.237
11	-0.177
12	-0.061
13	0.019
14	0.026
15	0.006

Test statistic: $Q_{LB}(m) = T(T+2) \times \sum_{h=1}^{m} \frac{\hat{\rho}(g)^2}{T-g}$

$$Q_{LB}(5) = 100(102) \times \sum_{h=1}^{5} \frac{\hat{\rho}(g)^2}{100-g}$$

$$= 10200 \times \left(\frac{(0.593)^2}{99} + \frac{(0.267)^2}{98} + \frac{(0.099)^2}{97} + \frac{(-0.109)^2}{96} + \frac{(-0.116)^2}{95} \right)$$

$$= 47.39$$

Critical region: Reject H_0 if $47.39 > \chi^2_{0.05;5}$ OR if p-value < 0.05

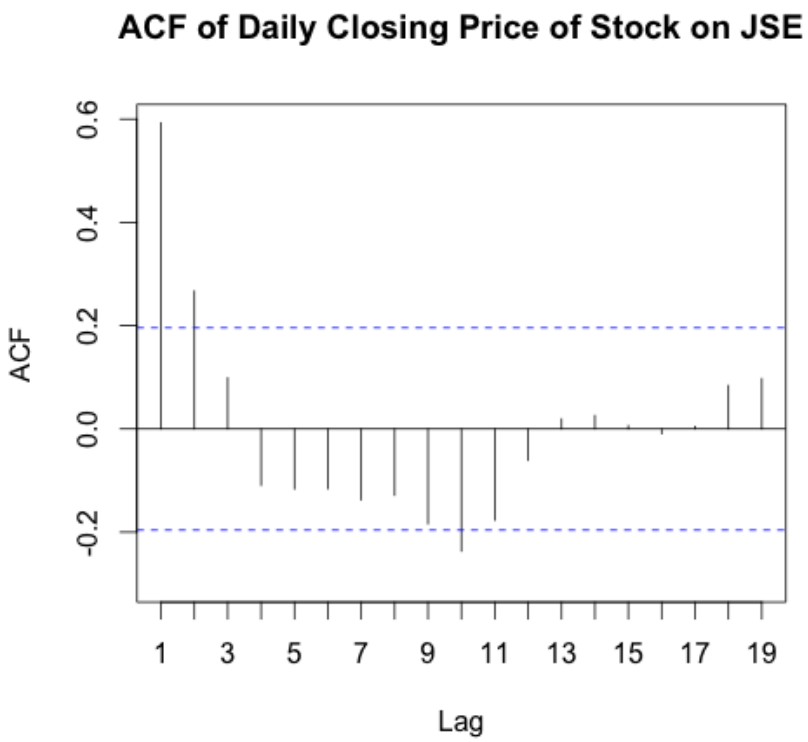
$\chi^2_{0.05;5} = 11.070$

p-value of test statistic < 0.005

Decision & Conclusion: Since 47.39 > 11.070 (OR p-value < 0.005) we Reject H_0 and conclude that there is autocorrelation between the daily closing price and at least one of the previous 5 days' closing prices

Diagnostic Checks of the Residuals

Decision & Conclusion: Since 47.39 > 11.070 (OR p-value < 0.005) we Reject H_0 and conclude that there is autocorrelation between the daily closing price and at least one of the previous 5 days' closing prices



Under the null hypothesis that there is no autocorrelation, we should not see more than 1 autocorrelation coefficient outside of the dotted lines.

Here, we see 3. Hence, the result of the Ljung-Box test makes sense.

Also, take note of the strength of the autocorrelations. At lag 1, there is moderate positive autocorrelation.

ACF of Daily Stock Price

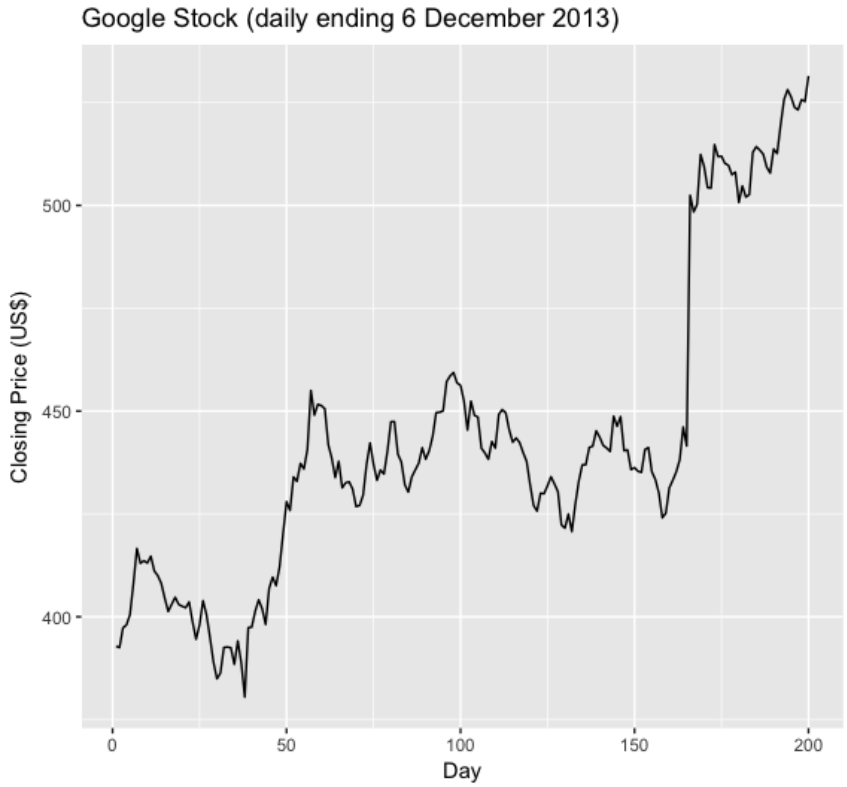
Lag	*Correlation*
0	1
1	0.593
2	0.267
3	0.099
4	-0.109
5	-0.116
6	-0.116
7	-0.138
8	-0.129
9	-0.184
10	-0.237
11	-0.177
12	-0.061
13	0.019
14	0.026
15	0.006

A reminder that the dashed blue lines represent a 95% confidence interval of the hypothesis that there is no autocorrelation. So, sometimes it is possible that you get an ACF there there is one autocorrelation 'spike' that is just outside the dashed blue lines, but all other spikes are within. In such a case, that spike might be outside the lines due to chance. So, we can still conclude that there is no autocorrelation.

Diagnostic Checks of the Residuals

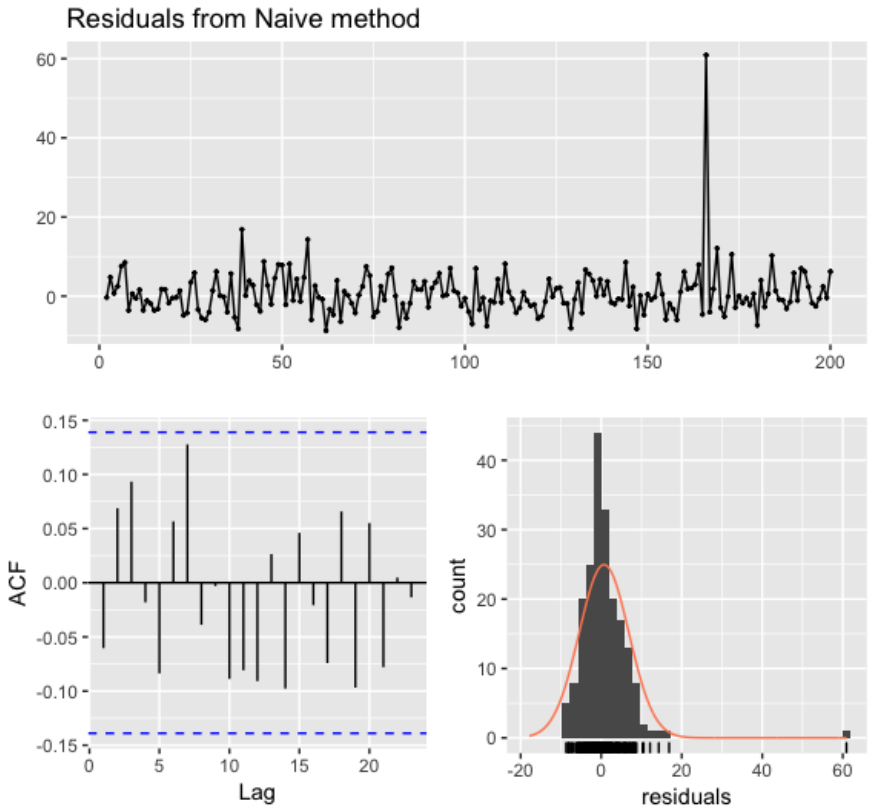
In *RStudio* the `checkresiduals()` ction performs all the above diagnostic checks of the residuals of a fitted model

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



What is NB is for you to discuss EACH of the three plots **AND** the Ljung-Box test output when diagnosing the residuals to assess the adequacy of a fitted model.

```
checkresiduals(naive(goog200))
```



The ACF plot here shows that there is **no autocorrelation** in the residuals, which means *that the model has captured all the relevant information/patterns* in the data.

```
        Ljung-Box test

data:  Residuals from Naive method
Q* = 11.031, df = 10, p-value = 0.3551

Model df: 0.   Total lags used: 10
```

Forecasting --- forecast accuracy

- There are many forecasting methods available. When more than one forecasting model is applicable in a particular situation, one way of deciding which method to use is to select the technique that results in the best forecast accuracy (***retrospectively***)

- Two of the most common measures of forecast accuracy are the **mean absolute error (MAE)** and the **mean squares error (MSE).**

$$MAE = \sum_{t=1}^{T} \frac{|Y_t - \widehat{Y_t}|}{T} \qquad MSE = \sum_{t=1}^{T} \frac{(Y_t - \widehat{Y_t})^2}{T}$$

Where:

- $\hat{Y}_t$ is the forecasted value in period t
- Y_t is the actual observation in period t
- T is the number of time periods
- $\widehat{e_t} = Y_t - \hat{Y}_t$ is the forecast error

Forecasting --- forecast accuracy

MAE is a measure of the average of the absolute differences between the actual and fitted values for a given set of data:

- If a model predicts the data perfectly, the MAE will be zero, and if a model predicts the data poorly, the MAE will be large (*relative to your data range*)

MSE is a measure of the average of squared differences between the actual and fitted values for a given set of data:

- If a model predicts the data perfectly, the MSE will be zero, and if a model predicts the data poorly, the MSE will be large (*relative to your data range*)

Choosing the measure of forecasting accuracy to use depends on circumstances:

- If it is important to avoid (even a few) large errors, then use MSE because it penalizes large deviations more heavily (through the squaring process) than does MAE. Otherwise, use MAE.
- Note that MAE and MSE are both scale-dependent accuracy measures i.e. they can only be used to compare models fitted on data that has the same scale
- When comparing models using forecast accuracy, we prefer models with smaller values for these measures. Can you explain why?

Forecasting – forecast accuracy Example

Compare the accuracy of Model 1 and Model 2 over 3 months

			Model 1			Model 2		
Month	**t**		$\widehat{Y_t}$	$\lvert Y_t - \widehat{Y_t} \rvert$	$(Y_t - \widehat{Y_t})^2$	$\widehat{Y_t}$	$\lvert Y_t - \widehat{Y_t} \rvert$	$(Y_t - \widehat{Y_t})^2$
Jan	1	45	51			46		
Feb	2	54	48			57		
Mar	3	34	42			39		
				MAE	***MSE***		***MAE***	***MSE***

Forecasting – forecast accuracy Example

Which model would we prefer here? Model 2 because it has ***lower/smaller values for either MAE or MSE***. What do these smaller values for Model 2 when compared to Model 1 imply?

			Model 1			**Model 2**		
Month	**t**		$\widehat{Y_t}$	$\lvert Y_t - \widehat{Y_t} \rvert$	$(Y_t - \widehat{Y_t})^2$	$\widehat{Y_t}$	$\lvert Y_t - \widehat{Y_t} \rvert$	$(Y_t - \widehat{Y_t})^2$
Jan	1	45	51	6	36	46	1	1
Feb	2	54	48	6	36	57	3	9
Mar	3	34	42	8	64	39	5	25
				6.67	*45.33*		*3*	*11.67*
				MAE	***MSE***		***MAE***	***MSE***

Forecasting – forecast accuracy Example

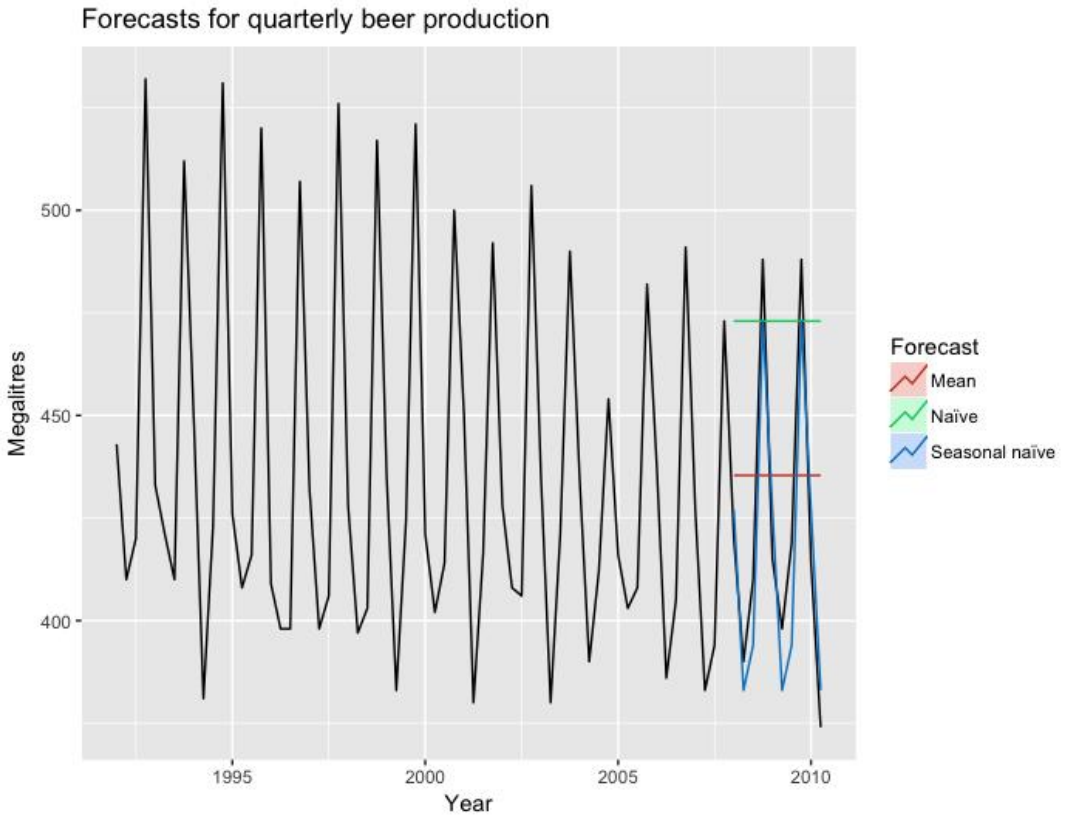
RStudio Example: Australian quarterly beer production

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

To get an idea as to how accurately our model can forecast 'unseen' data, we split our observed data into a training dataset and a 'test' dataset.

NB NB – the model NEVER sees the test data. It is fitted to the training data and then the model is used to make forecasts for the test data time periods.

These forecasts are then compared to the values in the test set to calculate measures of forecast accuracy.



Forecasting – forecast accuracy Example

RStudio Example: Australian quarterly beer production

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

Note that R outputs RMSE which is simply the square root of MSE.

```
> accuracy(beerfit1, beer3)
                ME     RMSE      MAE        MPE     MAPE     MASE        ACF1 Theil's U
Training set   0.000 43.62858 35.23438 -0.9365102 7.886776 2.463942 -0.10915105        NA
Test set     -13.775 38.44724 34.82500 -3.9698659 8.283390 2.435315 -0.06905715  0.801254
> accuracy(beerfit2, beer3)
                   ME     RMSE      MAE         MPE     MAPE     MASE        ACF1 Theil's U
Training set   0.4761905 65.31511 54.73016  -0.9162496 12.16415 3.827284 -0.24098292        NA
Test set     -51.4000000 62.69290 57.40000 -12.9549160 14.18442 4.013986 -0.06905715  1.254009
> accuracy(beerfit3, beer3)
                  ME     RMSE  MAE        MPE     MAPE      MASE       ACF1 Theil's U
Training set -2.133333 16.78193 14.3 -0.5537713 3.313685 1.0000000 -0.2876333        NA
Test set      5.200000 14.31084 13.4  1.1475536 3.168503 0.9370629  0.1318407  0.298728
```

The training set accuracy measures have little value, since its not informative to know how well the model predicted values that it has already seen! Its possible for the model to fit the training data perfectly but make poor forecasts for future time periods. This is a problem called 'overfitting'. The model essentially learns noise in the training data i.e., it learns things that aren't real patterns in the training data***. Thus, we only consider the Test set values when comparing the forecast accuracy of models.***

Here, model 3 would be the preferred model to forecast values. Why?

Forecasting – forecast accuracy Example

RStudio Example: Google daily stock prices

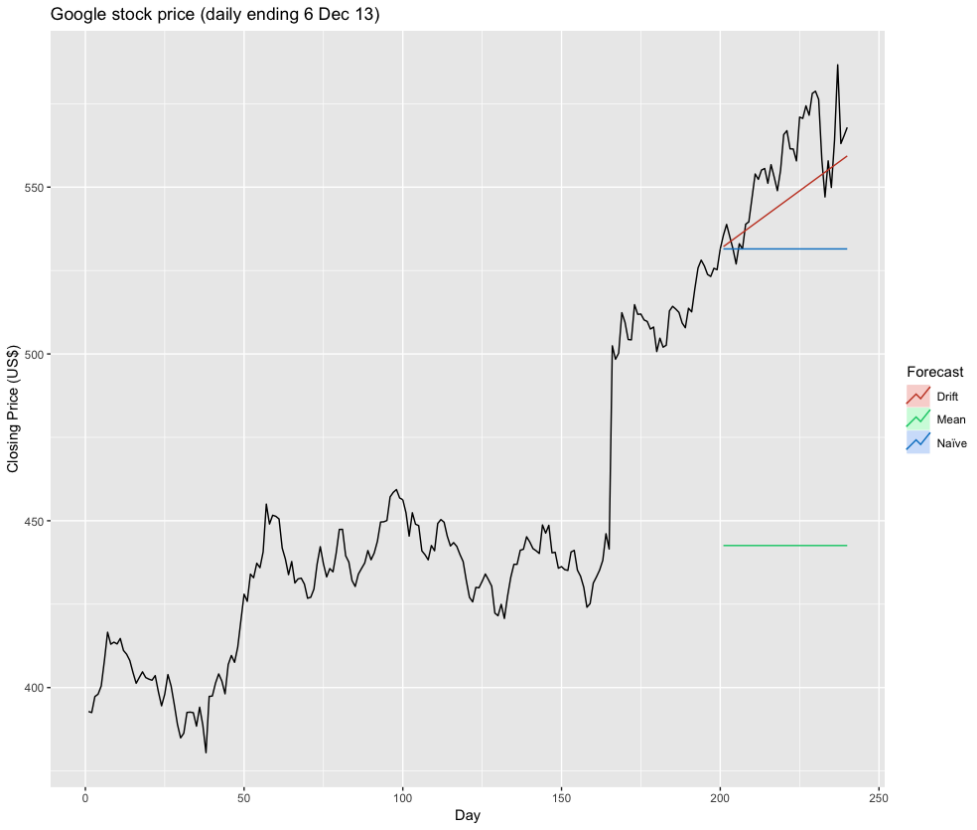
```
#Google daily stock price example:
googfc1 <- meanf(goog200, h=40)
googfc2 <- rwf(goog200, h=40)
googfc3 <- rwf(goog200, drift=TRUE, h=40)
autoplot(subset(goog, end = 240)) +
  autolayer(googfc1, PI=FALSE, series="Mean") +
  autolayer(googfc2, PI=FALSE, series="Naïve") +
  autolayer(googfc3, PI=FALSE, series="Drift") +
  xlab("Day") + ylab("Closing Price (US$)") +
  ggtitle("Google stock price (daily ending 6 Dec 13)") +
  guides(colour=guide_legend(title="Forecast"))
```

For how many time periods into the future are we making forecasts here?

In this example, we train the model on the first 200 daily stock prices, and then make predictions for the rest of the time periods. We then plot these predictions against the original data (before we split into train and test datasets).

Which simple forecasting model will provide the most accurate short-term forecasts here?

Which simple forecasting model will provide the most accurate long-term forecasts here?



Forecasting – forecast accuracy Example

RStudio Example: Google daily stock prices

```
googtest <- window(goog, start=201, end=240)
accuracy(googfc1, googtest)
accuracy(googfc2, googtest)
accuracy(googfc3, googtest)
```

```
> accuracy(googfc1, googtest)
                      ME      RMSE       MAE        MPE     MAPE      MASE      ACF1 Theil's U
Training set -4.296286e-15  36.91961  26.86941 -0.6596884  5.95376  7.182995 0.9668981        NA
Test set      1.132697e+02 114.21375 113.26971 20.3222979 20.32230 30.280376 0.8104340  13.92142
> accuracy(googfc2, googtest)
                     ME      RMSE       MAE       MPE      MAPE     MASE        ACF1 Theil's U
Training set  0.6967249  6.208148  3.740697 0.1426616 0.8437137 1.000000 -0.06038617        NA
Test set     24.3677328 28.434837 24.593517 4.3171356 4.3599811 6.574582  0.81043397  3.451903
> accuracy(googfc3, googtest)
                      ME      RMSE       MAE         MPE      MAPE     MASE        ACF1 Theil's U
Training set -5.998536e-15  6.168928  3.824406 -0.01570676 0.8630093 1.022378 -0.06038617        NA
Test set      1.008487e+01 14.077291 11.667241  1.77566103 2.0700918 3.119002  0.64732736  1.709275
```

Which of the three fitted models would you choose to make forecasts for the time series? Justify your answer by referring to a relevant piece of output.

Forecasting – prediction intervals

A prediction interval provides an interval within which we expect forecasted values to lie with a specified probability. For example, assuming that the forecast errors are normally distributed, a 95% prediction interval for the h-step forecast is:

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

where $\hat{\sigma}_h$ is an estimate of the standard deviation of the h-step forecast distribution, which we typically set equal to the standard deviation of the residuals for one-step ahead forecasts

More generally, a prediction interval can be written as

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

Where the multiplier c depends on the coverage probability. We usually calculate 80% intervals and 95% intervals, although any percentage may be used.

The following table gives the value of c for a range of coverage probabilities assuming normally distributed forecast errors:

Forecasting – prediction intervals

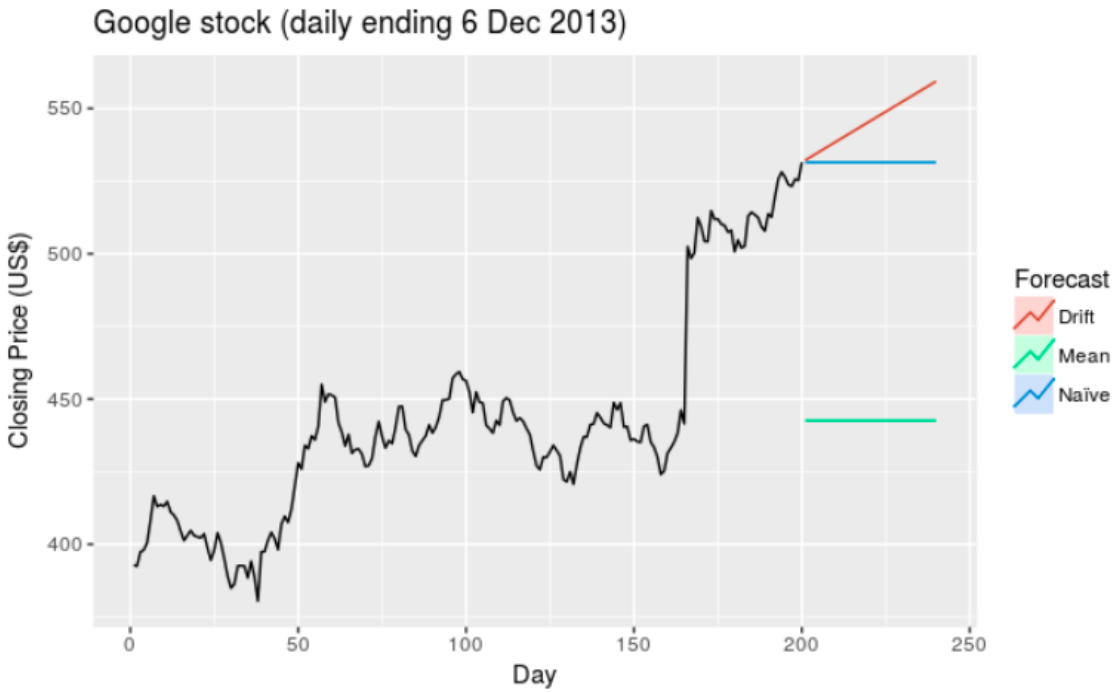
Percentage	Multiplier
50	0.67
55	0.76
60	0.84
65	0.93
70	1.04
75	1.15
80	1.28
85	1.44
90	1.64
95	1.96
96	2.05
97	2.17
98	2.33
99	2.58

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

Forecasting – prediction intervals

RStudio Example: Google daily stock prices

Recall the simple forecasts for the Google stock price data goog200.



For the naïve method, the last value of the observed series is 531.48, so the forecast of the next value of the stock price is 531.48.

The standard deviation of the residuals from the naïve method is 6.21.

Thus a 95% prediction interval for the next value of the stock price is:

$$531.48 \pm 1.96(6.21) = [519.3, 543.6]$$

Similarly, an 80% prediction interval is given by:

$$531.48 \pm 1.28(6.21) = [523.5, 539.4]$$

See RStudio Examples

Exponential smoothing

Forecasting based on the historical patterns in the data

(fpp) Chapter 8:
Sections 8.1 – 8.3

Exponential Smoothing

Exponential Smoothing derives its name from the fact that it consists of a series of exponentially weighted averages. Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older (i.e. the further you go back in time).

Throughout the series, each smoothing calculation is dependent on **all previously observed values** (to varying degrees – more emphasis is placed on more recent observations) i.e.

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

- In general,

$$\hat{Y}_{T+1|T} = \sum_{j=1}^{T-1} \alpha(1-\alpha)^j y_{T-j} + (1-\alpha)^T l_0 \qquad \textbf{(1)}$$

α is the smoothing constant **($0 \leq \alpha \leq 1$)** and l_0 is the estimated level in time period 0.

In the component form (slide 163), exponential smoothing models model each component as a weighted average of current information relevant to the component and previous component information.

Exponential smoothing generates reliable forecasts quickly and for a wide range of time series, which is a great advantage and of major importance to applications in industry.

Exponential Smoothing

The choice of α is somewhat subjective, and is made on the basis of how much smoothing is desired.

- a small value of α produces a great deal of smoothing
- a large value of α results in very little smoothing

The table below shows the weights attached to observations for four different values of α when forecasting using simple exponential smoothing

	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_T	0.2000	0.4000	0.6000	0.8000
y_{T-1}	0.1600	0.2400	0.2400	0.1600
y_{T-2}	0.1280	0.1440	0.0960	0.0320
y_{T-3}	0.1024	0.0864	0.0384	0.0064
y_{T-4}	0.0819	0.0518	0.0154	0.0013
y_{T-5}	0.0655	0.0311	0.0061	0.0003

If α is small (i.e., close to 0), more weight is given to observations from the more distant past. If α is large (i.e., close to 1), more weight is given to the more recent observations. For the extreme case where α=1, $\hat{Y}_{T+1|T} = \hat{Y}_T$, and the forecasts are equal to the naïve forecasts.

Exponential Smoothing

There are two different ways to express equation ***(1)***

(i) Weighted average form:

$$\hat{y}_{T+1|T} = \alpha y_T + (1-\alpha)\hat{y}_{T|T-1} \qquad \textit{(forecasts)}$$

$$\hat{y}_{t+1|t} = \alpha y_t + (1-\alpha)\hat{y}_{t|t-1} \qquad \textit{(fitted values)}$$

The process has to start somewhere, so we let the first fitted value at time 1 be denoted by l_0*, which we will need to estimate. Sometimes in simple exponential smoothing,* l_0 *is not estimated and* l_1 *is just set to be equal to the first observed value.*

Then:

$$\hat{y}_{2|1} = \alpha y_1 + (1-\alpha) l_0$$
$$\hat{y}_{3|2} = \alpha y_2 + (1-\alpha)\hat{y}_{2|1}$$
$$\hat{y}_{4|3} = \alpha y_3 + (1-\alpha)\hat{y}_{3|2}$$
$$\vdots$$
$$\hat{y}_{T|T-1} = \alpha y_{T-1} + (1-\alpha)\hat{y}_{T-1|T-2}$$
$$\hat{y}_{T+1|T} = \alpha y_T + (1-\alpha)\hat{y}_{T|T-1}$$

If you substitute each equation into the following equation, you will end up with equation ***(1)***

Simple Exponential Smoothing

(ii) Component form:

In general, the component form representations of exponential smoothing methods comprise a forecast equation and a smoothing equation for each of the components included in the method. For ***simple exponential smoothing***, the component form is given by:

Forecast equation: $\hat{y}_{T+k|t} = l_t$

Smoothing equation: $l_t = \alpha y_t + (1-\alpha) l_{t-1}$

where l_t is the level (or the smoothed value) of the series at time t. *The current level value is simply a weighted average of information relevant to the level and previous level information.*

Setting h=1 gives the fitted values, while setting t=T gives the true forecasts beyond the training data.

If we replace l_t with $\hat{y}_{t+1|t}$ and l_{t-1} with $\hat{y}_{t|t-1}$ in the smoothing equation and evaluate it recursively, we will get equation **(1)**

Simple Exponential Smoothing --- Example

The following data represent total revenues (in millions of Rands) for a car rental agency in South Africa over the 11 year period 1992 to 2002.

4.0	5.0	7.0	6.0	8.0	9.0	5.0	2.0	3.5
5.5	6.5							

a) Fit a simple exponential smoothing model with $\alpha = 0.25$ to the series.

Simple Exponential Smoothing --- Example

Year	t	Data (Y_t)	EXP(0.25)
1992	1	4	$l_1 = Y_1 = 4.00$
1993	2	5	
1994	3	7	
1995	4	6	
1996	5	8	
1997	6	9	
1998	7	5	
1999	8	2	
2000	9	3.5	
2001	10	5.5	
2002	11	6.5	

If one fits a simple exponential smoothing model by hand, you have to assume the first smoothed value is the first observed value

Simple Exponential Smoothing --- Example

Year	t	Data (Y_t)	EXP(0.25)
1992	1	4	4.00
1993	2	**5**	= (0.25)(**5**) + (1 - 0.25)(4.00) = 4.25
1994	3	7	
1995	4	6	
1996	5	8	
1997	6	9	
1998	7	5	
1999	8	2	
2000	9	3.5	
2001	10	5.5	
2002	11	6.5	

Simple Exponential Smoothing --- Example

Year	t	Data (Y_t)	EXP(0.25)
1992	1	4	4.00
1993	2	5	**4.25**
1994	3	**7**	= (0.25)(**7**) + (0.75)(**4.25**) = 4.94
1995	4	6	
1996	5	8	
1997	6	9	
1998	7	5	
1999	8	2	
2000	9	3.5	
2001	10	5.5	
2002	11	6.5	

Q: Can you see how each value of the level is a weighted average of current level information and previous level information? Hint: 0.25 = ¼ and 0.75 = ¾ .

Simple Exponential Smoothing --- Example

Year	t	Data (Y_t)	EXP(0.25)
1992	1	4	4.00
1993	2	5	4.25
1994	3	7	4.94
1995	4	6	5.20
1996	5	8	5.90
1997	6	9	6.68
1998	7	5	6.26
1999	8	2	5.19
2000	9	3.5	4.77
2001	10	5.5	4.95
2002	11	6.5	5.34

Simple Exponential Smoothing --- Forecasting

Simple exponential smoothing is only useful for forecasting time series that **do NOT display a trend or seasonal component**. This is because all future forecasted values are all set to be the value of the last level component, i.e.

$$\hat{y}_{T+h|T} = \hat{y}_{T+1|T} = l_T$$

This resembles the naïve forecasting method.

Optimization of components

When fitting any exponential smoothing model, the initial values and smoothing parameters must either be chosen or estimated. For simple exponential smoothing this would mean choosing the values of α and l_0.

Once we know the initial values of the level, smoothing parameter and other components, all forecasts can be computed from the observed data.

Simple Exponential Smoothing --- Forecasting

In certain cases, these values may be chosen subjectively, perhaps based on the forecasters knowledge/experience. A more reliable way is to estimate these values from the observed data by minimizing the sum of squared errors (SSE), just as was done in regression:

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

This is done via non-linear optimization, which is beyond the scope of this course. While we won't concern ourselves with estimating these component values, given the component and smoothing parameter values we will compute forecasted values using our observed data.

Simple Exponential Smoothing --- Forecasting Example

Oil production example:

```
oildata <- window(oil, start=1996)
autoplot(oildata) + ggtitle("Annual oil production in Saudi Arabia") +
  ylab("Oil (millions of tonnes)") + xlab("Year")
```

Simple Exponential Smoothing --- Forecasting Example

Oil production example:

```
#Estimate parameters
fc <- ses(oildata, h=5)
#View fitted model
fc[["model"]]
Simple exponential smoothing

Call:
 ses(y = oildata, h = 5)

  Smoothing parameters:
    alpha = 0.8339

  Initial states:
    l = 446.5759

  sigma:  28.1223

     AIC     AICc      BIC
178.1430 179.8573 180.8141
#Accuracy of one-step-ahead training errors: period 1-12
round(accuracy(fc),2)
                ME  RMSE   MAE MPE MAPE MASE  ACF1
Training set 6.4 28.12 22.26 1.1 4.61 0.93 -0.03
```

Considering the oil production series shows increasing non-linear trend, a simple exponential smoothing model would **NOT** be an appropriate model to forecast future values because it cannot model trend in a time series.

This example shows how to fit a simple exponential smoothing model in Rstudio, and also demonstrates how it is not appropriate for forecasting time series with trend. Hence why we need to consider an exponential smoothing model that can model trend.

Simple Exponential Smoothing --- Forecasting Example

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

Forecast equation:

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

Smoothing equation

$$l_t = \alpha y_t + (1-\alpha) l_{t-1}$$

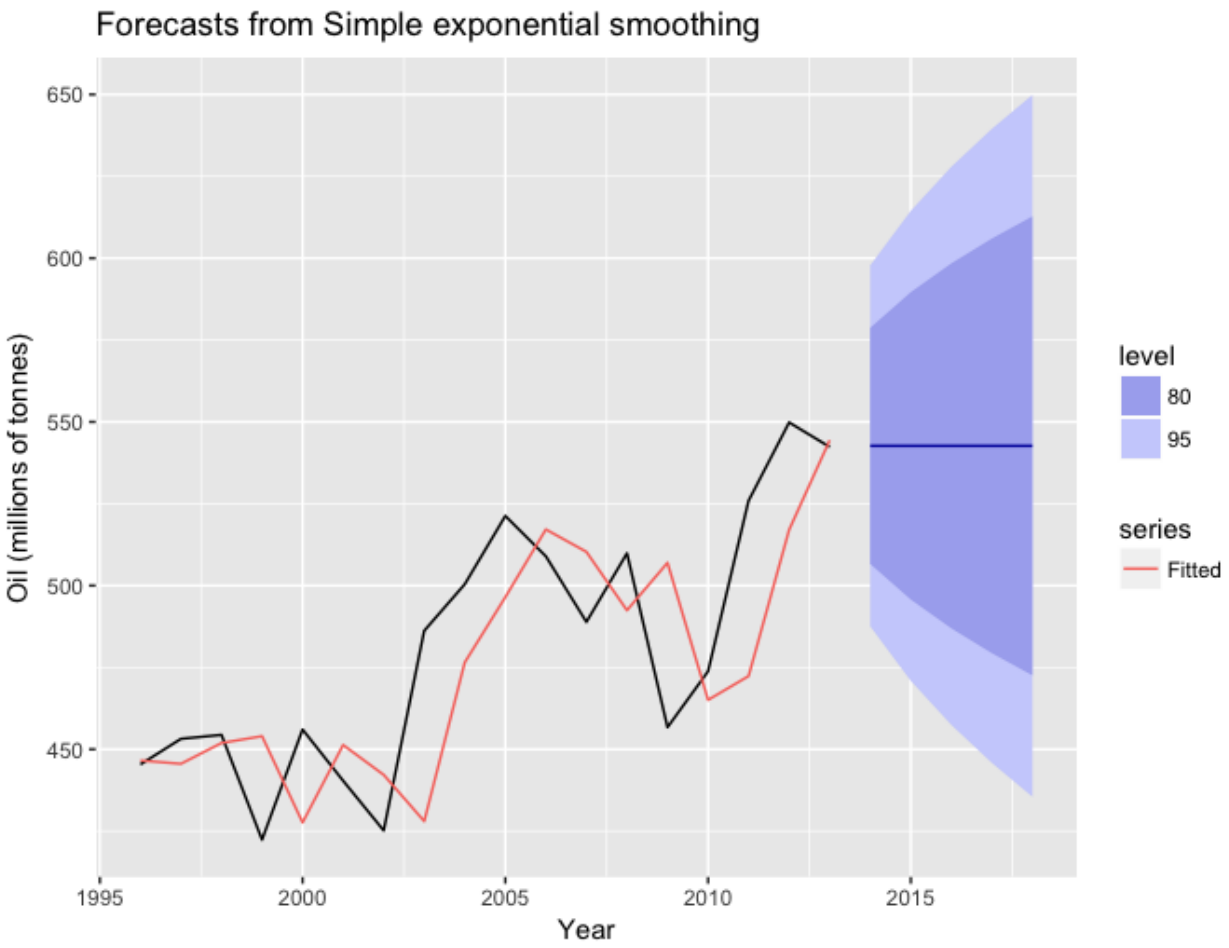
$\alpha = 0.8339$

Test yourself to see that you can obtain both the Level and Forecast values in this table using the above equations.

The 'forecasted' value within the observed time series periods is simply the previous time period's level value.

Simple Exponential Smoothing --- Forecasting Example

```
#Oil time series with fitted simple exponential model
autoplot(fc) +
  autolayer(fitted(fc), series="Fitted") +
  ylab("Oil (millions of tonnes)") + xlab("Year")
```



The large value of α (0.833) is evident in the large changes that take place in the estimated level l_t in each time period. A smaller value of α would lead to smaller changes over time, and so the series of fitted values would be smoother.

Exponential Smoothing--- Trend Methods --- Holt's Linear Trend

We can extend simple exponential smoothing to allow the forecasting of time series with a linear trend. This involves a forecasting equation and two smoothing equations, one for the level and one for the trend:

Forecast equation: $\hat{y}_{t+h|t} = l_t + hb_t$

Level equation: $l_t = \alpha y_t + (1-\alpha)(l_{t-1}+b_{t-1})$

Trend equation: $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$

l_t is the level (or the smoothed value) of the series at time t

b_t is the estimate of the trend (slope) at time t

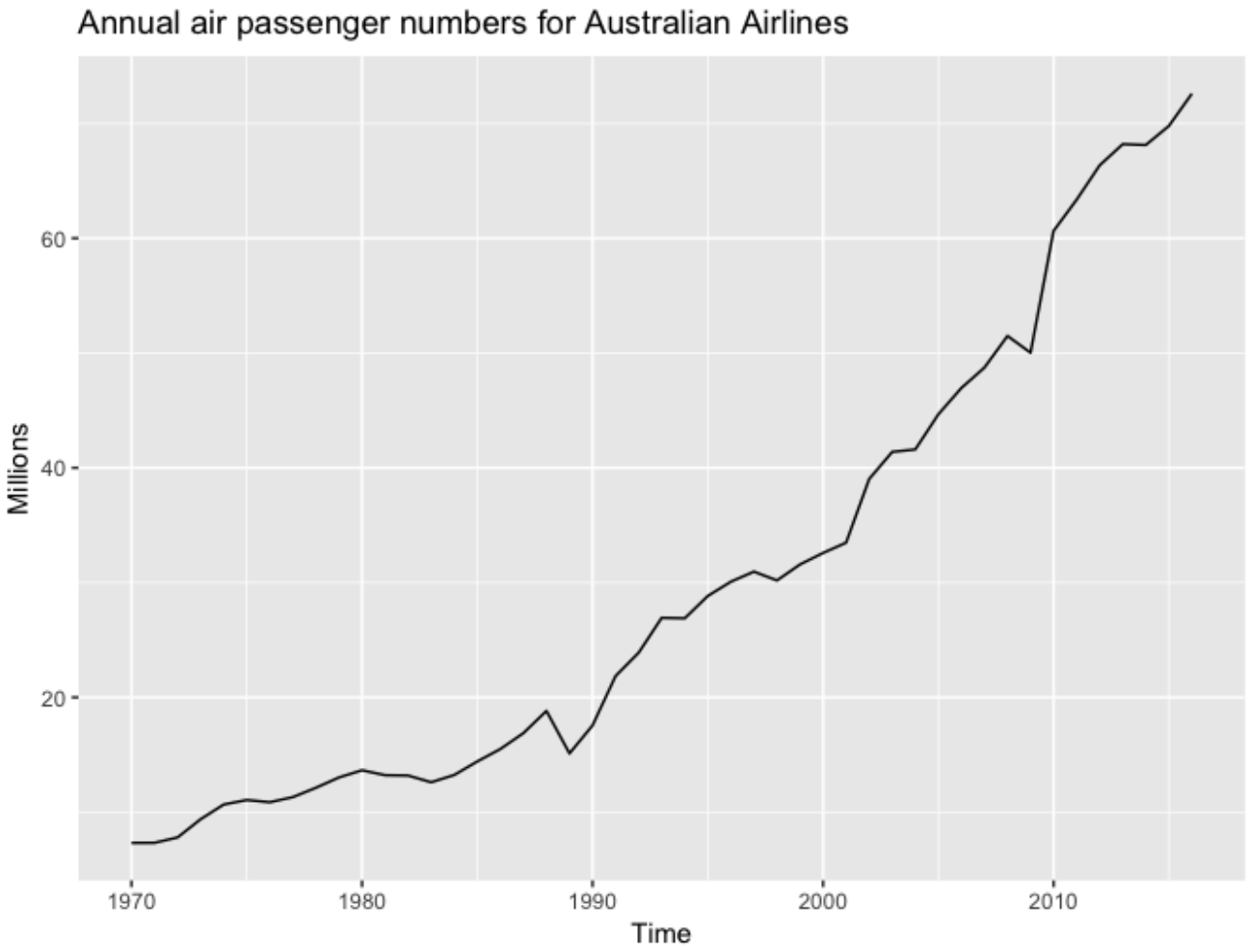
α is the smoothing parameter for the level, $0 \leq \alpha \leq 1$

β^* is the smoothing parameter for the trend, $0 \leq \beta^* \leq 1$

(Note that $\beta^*$is **NOT** the same as the coefficients from a regression model). The forecast function is no longer flat but trending. The h-step-ahead forecast is equal to the last estimated level plus h times the last estimated trend value. Hence the forecasts are a linear function of h.

Exponential Smoothing--- Holt's Linear Trend --- Example

```
autoplot(ausair) +
ggtitle("Annual air passenger numbers for Australian Airlines") +
  ylab("Millions")
```



Exponential Smoothing--- Holt's Linear Trend --- Example

```
air <- window(ausair, start=1990)

#Estimate parameters
fc <- holt(air, h=5)

#View fitted model
fc[["model"]]

Holt's method

Call:
 holt(y = air, h = 5)

  Smoothing parameters:
    alpha = 0.8302
    beta  = 1e-04

  Initial states:
    l = 15.5715
    b = 2.1017

  sigma:  2.3645

     AIC     AICc      BIC
141.1291 143.9863 147.6083
```

h is the forecast horizon value – the number of time periods into the future for which you want to forecast, typically equal to the number of time periods in the test set data

The very small value of β^* means that the slope hardly changes over time, since almost all the weight is placed on the previous slope value.

The initial states are the values of the level and slope components in time period 0. These would be used to calculate values in time period 1 – see the table on the next slide.

Year	Time	Observation	Level	Slope	Forecast
	t	y_t	ℓ_t	b_t	$\hat{y}_{t\|t-1}$
1989	0		15.57	2.102	
1990	1	17.55	17.57	2.102	17.67
1991	2	21.86	21.49	2.102	19.68
1992	3	23.89	23.84	2.102	23.59
1993	4	26.93	26.76	2.102	25.94
1994	5	26.89	27.22	2.102	28.86
1995	6	28.83	28.92	2.102	29.33
1996	7	30.08	30.24	2.102	31.02
1997	8	30.95	31.19	2.102	32.34
1998	9	30.19	30.71	2.101	33.29
1999	10	31.58	31.79	2.101	32.81
2000	11	32.58	32.80	2.101	33.89
2001	12	33.48	33.72	2.101	34.90
2002	13	39.02	38.48	2.101	35.82
2003	14	41.39	41.25	2.101	40.58
2004	15	41.60	41.89	2.101	43.35
2005	16	44.66	44.54	2.101	44.00
2006	17	46.95	46.90	2.101	46.65
2007	18	48.73	48.78	2.101	49.00
2008	19	51.49	51.38	2.101	50.88
2009	20	50.03	50.61	2.101	53.49
2010	21	60.64	59.30	2.102	52.72
2011	22	63.36	63.03	2.102	61.40
2012	23	66.36	66.15	2.102	65.13
2013	24	68.20	68.21	2.102	68.25
2014	25	68.12	68.49	2.102	70.31
2015	26	69.78	69.92	2.102	70.60
2016	27	72.60	72.50	2.102	72.02
	h				$\hat{y}_{t+h\|t}$
	1				74.60
	2				76.70
	3				78.80
	4				80.91
	5				83.01

Forecast equation:

$$\hat{\boldsymbol{y}}_{\boldsymbol{t+h|t}} = \boldsymbol{l_t} + \boldsymbol{hb_t}$$

(for your observed data, h = 1)

Level equation:

$$\boldsymbol{l_t} = \boldsymbol{\alpha y_t} + (\mathbf{1} - \boldsymbol{\alpha})(\boldsymbol{l_{t-1}} + \boldsymbol{b_{t-1}})$$

Can you see that the value of the level component is a weighted average of current relevant level information and previous level information?

Trend equation:

$$\boldsymbol{b_t} = \boldsymbol{\beta^*}(\boldsymbol{l_t} - \boldsymbol{l_{t-1}}) + (\mathbf{1} - \boldsymbol{\beta^*})\boldsymbol{b_{t-1}}$$

Can you see that the value of the slope component is a weighted average of current relevant slope information and previous slope information?

$\boldsymbol{\alpha} = 0.8302$

$\boldsymbol{\beta^*} = 0.0001$

Test yourself to see that you can obtain the Level, Slope and Forecast values in this table using the above equations with the estimated initial state and parameter values.

(Note that the values you get may be slightly different due to rounding.)

Exponential Smoothing --- Holt-Winters Seasonal Method

The Holt-Winters seasonal method is an extension of Holt's linear trend method that caters for time series that display seasonality in addition to a trend.

As with the general formulation for modelling a time series, there are both additive and multiplicative formulations of the Holt-Winters method. The decision of which to choose depends on whether the nature (e.g. amplitude) of the seasonality stays constant throughout the time series (additive formulation preferred) or changes with time (multiplicative formulation preferred). We only consider the multiplicative formulation in this course.

The Holt-Winters multiplicative seasonal method comprises the forecast equation and three smoothing equations:

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

Level equation: $$l_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1}+b_{t-1})$$

Trend equation: $$b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$$

Seasonal equation: $$s_t = \gamma \frac{y_t}{(l_{t-1}+b_{t-1})} + (1-\gamma)s_{t-m}$$

Exponential Smoothing --- Holt-Winters Seasonal Method

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

Level equation: $$l_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1}+b_{t-1})$$

Trend equation: $$b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$$

Seasonal equation: $$s_t = \gamma \frac{y_t}{(l_{t-1}+b_{t-1})} + (1-\gamma)s_{t-m}$$

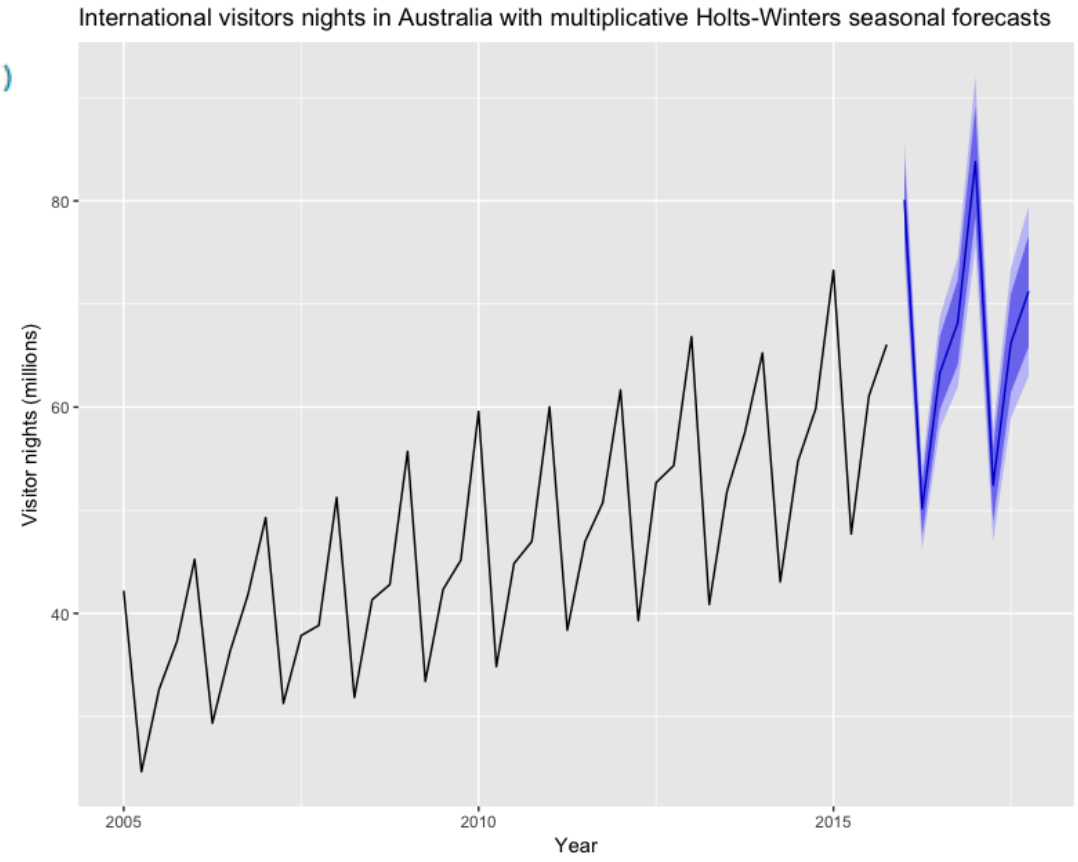
Where *m* denotes the frequency of the seasonality, i.e., the number of seasons in a year. For example, for quarterly data *m* = 4, and for monthly data *m* = 12.

k is the integer part of *(h-1)/m*, which ensures that the estimates of the seasonal indices used for forecasting come from the final year of the sample.

The level equation shows a weighted average between the seasonally adjusted observation and the non-seasonal forecast for time period t. The trend equation is identical to Holt's linear method. The seasonal equation shows a weighted average between the current seasonal index and the seasonal index of the same season last year (*m* time periods ago).

Exponential Smoothing --- Holt-Winters Seasonal Method --- Example

```
#International visitor nights to Australia example
aust <- window(austourists,start=2005)
fit <- hw(aust,seasonal="multiplicative")
autoplot(aust) +
  autolayer(fit) +
  xlab("Year") +
  ylab("Visitor nights (millions)") +
  ggtitle("International visitors nights
           in Australia with multiplicative
           Holts-Winters seasonal forecasts")
```



Exponential Smoothing --- Holt-Winters Seasonal Method --- Example

```
#test model by diving data into training and test sets:
train <- window(austourists, start = 2005, end = c(2013,4))
#print the traning data in the console
train
```

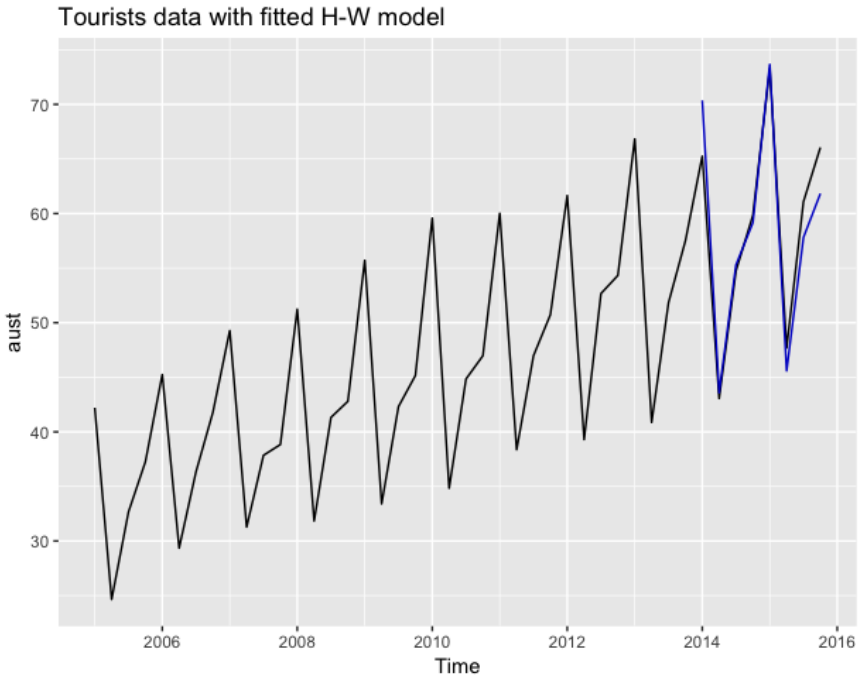
```
         Qtr1     Qtr2     Qtr3     Qtr4
2005 42.20566 24.64917 32.66734 37.25735
2006 45.24246 29.35048 36.34421 41.78208
2007 49.27660 31.27540 37.85063 38.83704
2008 51.23690 31.83855 41.32342 42.79900
2009 55.70836 33.40714 42.31664 45.15712
2010 59.57608 34.83733 44.84168 46.97125
2011 60.01903 38.37118 46.97586 50.73380
2012 61.64687 39.29957 52.67121 54.33232
2013 66.83436 40.87119 51.82854 57.49191
```

```
test <- window(austourists, start = 2014)
#print the test data in the console
test
```

```
         Qtr1     Qtr2     Qtr3     Qtr4
2014 65.25147 43.06121 54.76076 59.83447
2015 73.25703 47.69662 61.09777 66.05576
```

```
#fit Holt-winters to training data
fit <- hw(train, seasonal = "multiplicative", h = 8)

#plot H-W model forecasts against data
autoplot(aust) +
  autolayer(fit, PI = FALSE) +
  ggtitle("Tourists data with fitted H-W model")
```




```
#test accuracy of forecasts for test data
accuracy(fit, test)
```

```
                    ME     RMSE      MAE
Training set -0.2647464 1.366473 1.083639
Test set      0.4633637 2.747190 2.106150
```

	t	y_t	l_t	b_t	s_t	$\hat{y}_t$
2004 Q1	-3				0.9618	
2004 Q2	-2				0.7704	
2004 Q3	-1				1.2442	
2004 Q4	0		32.4875	0.6974	1.0237	
2005 Q1	1	42.2057	37.8980	0.7606	0.9625	31.9172
2005 Q2	2	24.6492	35.7228	0.7213	0.7701	29.7826
2005 Q3	3	32.6673	31.9551	0.6611	1.2434	45.3437
2005 Q4	4	37.2574	34.2811	0.6834	1.0240	33.3892
2006 Q1	5	45.2425	40.2696	0.7545	0.9633	33.6533
2006 Q2	6	29.3505	39.7413	0.7373	0.7700	31.5927
2006 Q3	7	36.3442	35.5223	0.6709	1.2426	50.3311
2006 Q4	8	41.7821	38.2242	0.6981	1.0243	37.0618
…	…	…	…	…	…	…
2014 Q4	40	59.8345	55.5289	0.6351	1.0252	54.6193
2015 Q1	41	73.2570	64.7175	0.7497	0.9701	54.4398
2015 Q2	42	47.6966	63.9644	0.7296	0.7685	50.3181
2015 Q3	43	61.0978	57.9607	0.6394	1.2358	79.9941
2015 Q4	44	66.0558	61.1697	0.6738	1.0254	60.0768
	h					
2016 Q1	1					59.9944
2016 Q2	2					48.0445
2016 Q3	3					78.0916
2016 Q4	4					65.4871
2017 Q1	5					62.6090
2017 Q2	6					50.1158

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

Level equation: $l_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1}+b_{t-1})$

Trend equation: $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$

Seasonal equation: $s_t = \gamma\frac{y_t}{(l_{t-1}+b_{t-1})} + (1-\gamma)s_{t-m}$

```
> fit[["model"]]
Holt-Winters' multiplicative method

Call:
 hw(y = aust, seasonal = "multiplicative")

  Smoothing parameters:
    alpha = 0.4406
    beta  = 0.0134
    gamma = 0.0023

  Initial states:
    l = 32.4875
    b = 0.6974
    s=1.0237 0.9618 0.7704 1.2442
```

The initial states here are for the level and slope in time period 0 and the previous seasonal component values

Test yourself to see that you can obtain the Level, Slope, Seasonal and Forecast values in this table using the equations and estimates for initial states and parameter values. For example:

$$l_1 = (0.4406)\frac{42.2057}{0.9618} + (0.5594)(32.4875 + 0.6974) = 37.8980$$

$$b_8 = (0.0134)(38.2246 - 35.5223) + (0.9866)(0.6709) = 0.6981$$

$$s_{44} = (0.0023)\frac{66.0558}{(57.9607 + 0.6394)} + (0.9977)(1.0252) = 1.0254$$

$$\hat{y}_{t+4|t} = (61.1697 + (4)(0.6738))(1.0254) = 65.4871$$

Again, can you see that in each time period t, each component is a weighted average between the most recent information relevant to that component and previous information relevant to that component?

Autocorrelation, Stationary, the ACF and PACF

Autocorrelation

Definitions

- **Autocorrelation** refers to the correlation of a variable with lagged values of itself. It is also called *serial correlation*

- Autocorrelation at lag g = 1 refers to the strength of the relationship between consecutive observations/values in a time series i.e. pairs of observations that are one period apart --- y_t , y_{t-1}

- Autocorrelation at lag g = 2 refers to the strength of the relationship between observations/values that are two periods apart --- y_t , y_{t-2}

- Autocorrelation at lag g = p refers to the strength of the relationship between observations/values that are *p* periods apart --- y_t , y_{t-p}

Autocorrelation

$$\rho(h) = \frac{\sum_{t=h+1}^{T}(y_t - \bar{y})(y_{t-h} - \bar{y})}{\sum_{t=1}^{T}(y_t - \bar{y})^2}$$

Where T is the length of the time series

$\rho(h)$ takes on a value on [-1;1] and is interpreted in a similar way to the correlation coefficient i.e. -1 is perfect negative autocorrelation, 1 is perfect positive autocorrelation and 0 is no autocorrelation.

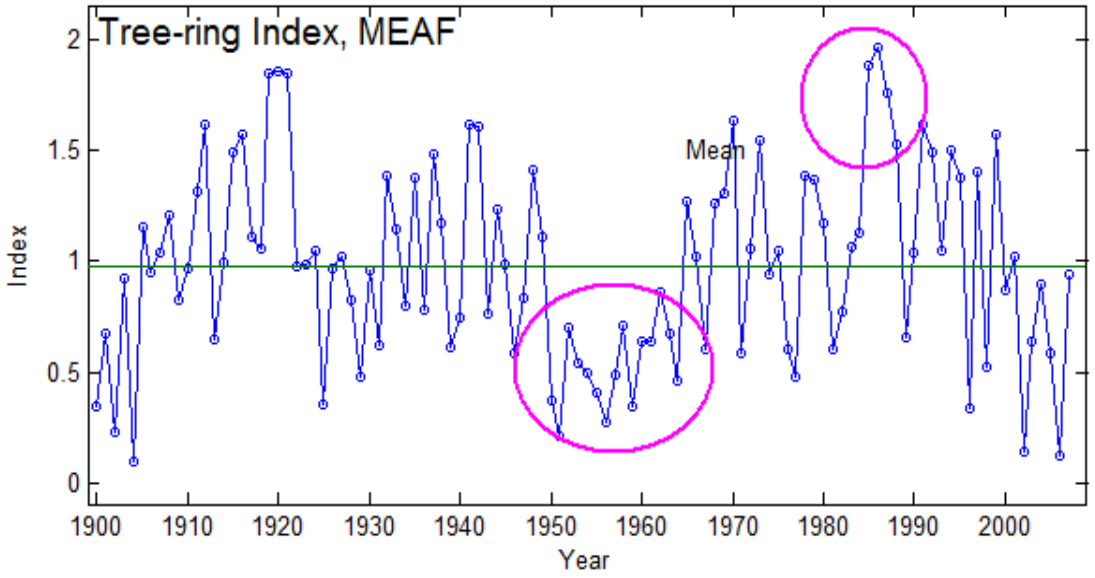
Recall that if autocorrelation exists in the data it is a both:

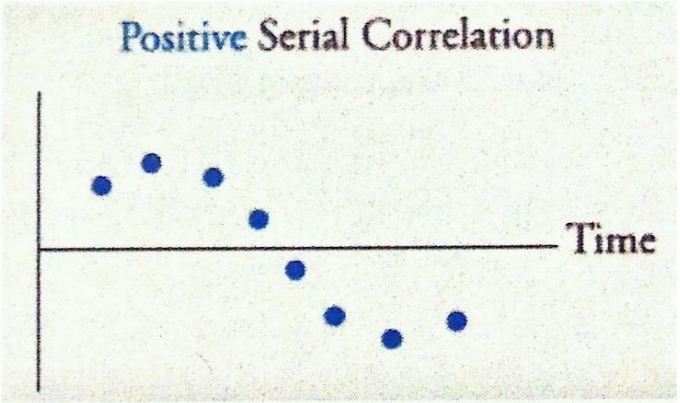
(i) **a problem** as regression-based forecasting models *may be incorrect or at least imprecise* because they assume independence between residuals. BUT, autocorrelation is also:

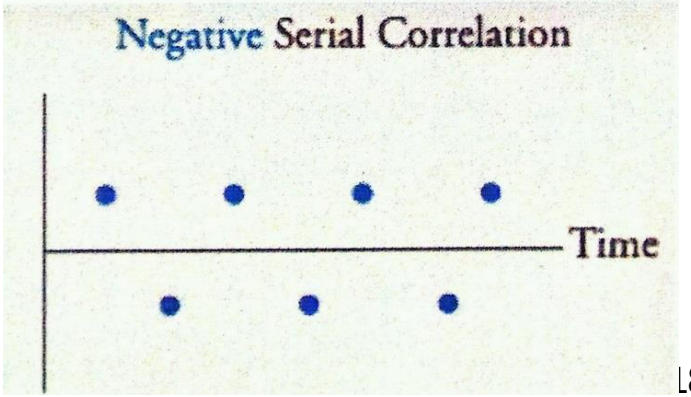
(ii) **important** as it represents information in the time series that we hope to capture in a model to forecast future values.

Autocorrelation

- Observations are positively autocorrelated if there is a large degree of similarity between observations close together in time, whilst observations are negatively autocorrelated if observations close together in time are more dissimilar (relative to the mean value of the time series)



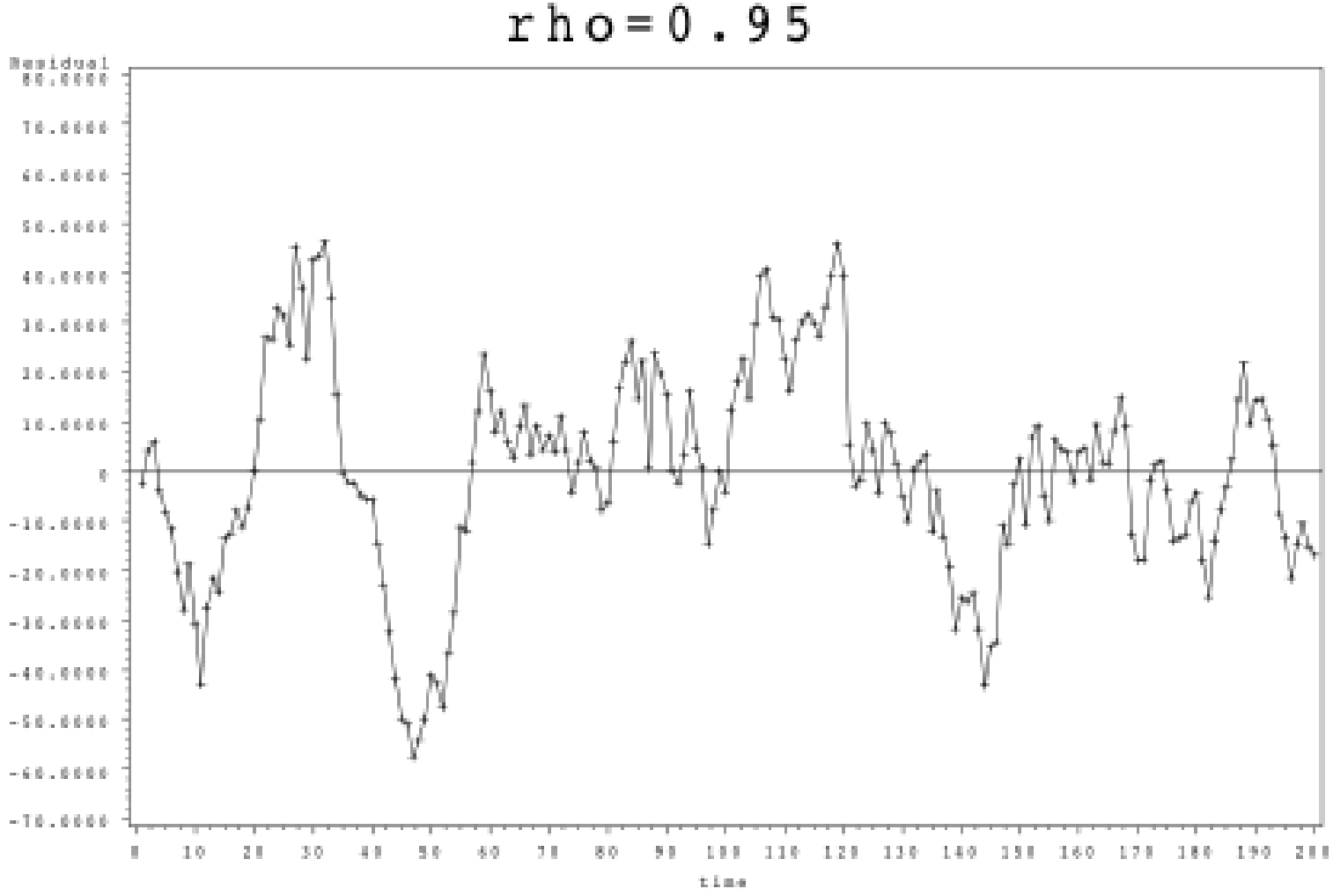




- The tendency for highs to follow highs or lows to follow lows is indicative of positive autocorrelation

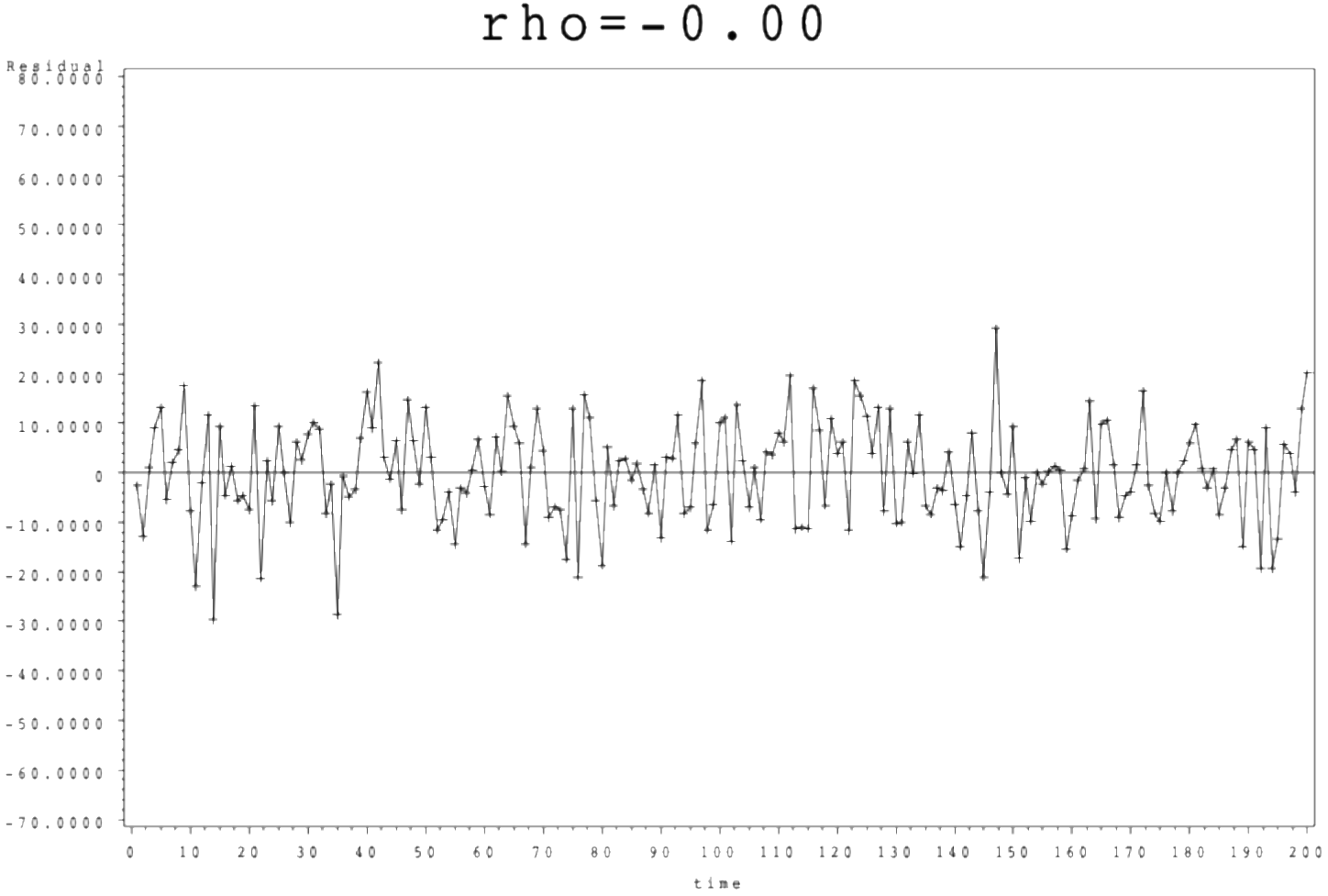
Autocorrelation

Positive autocorrelation implies that successive values of the time series are similar over short time intervals. A value of rho (ρ) of 0.95 is interpreted as strong positive autocorrelation.



Autocorrelation

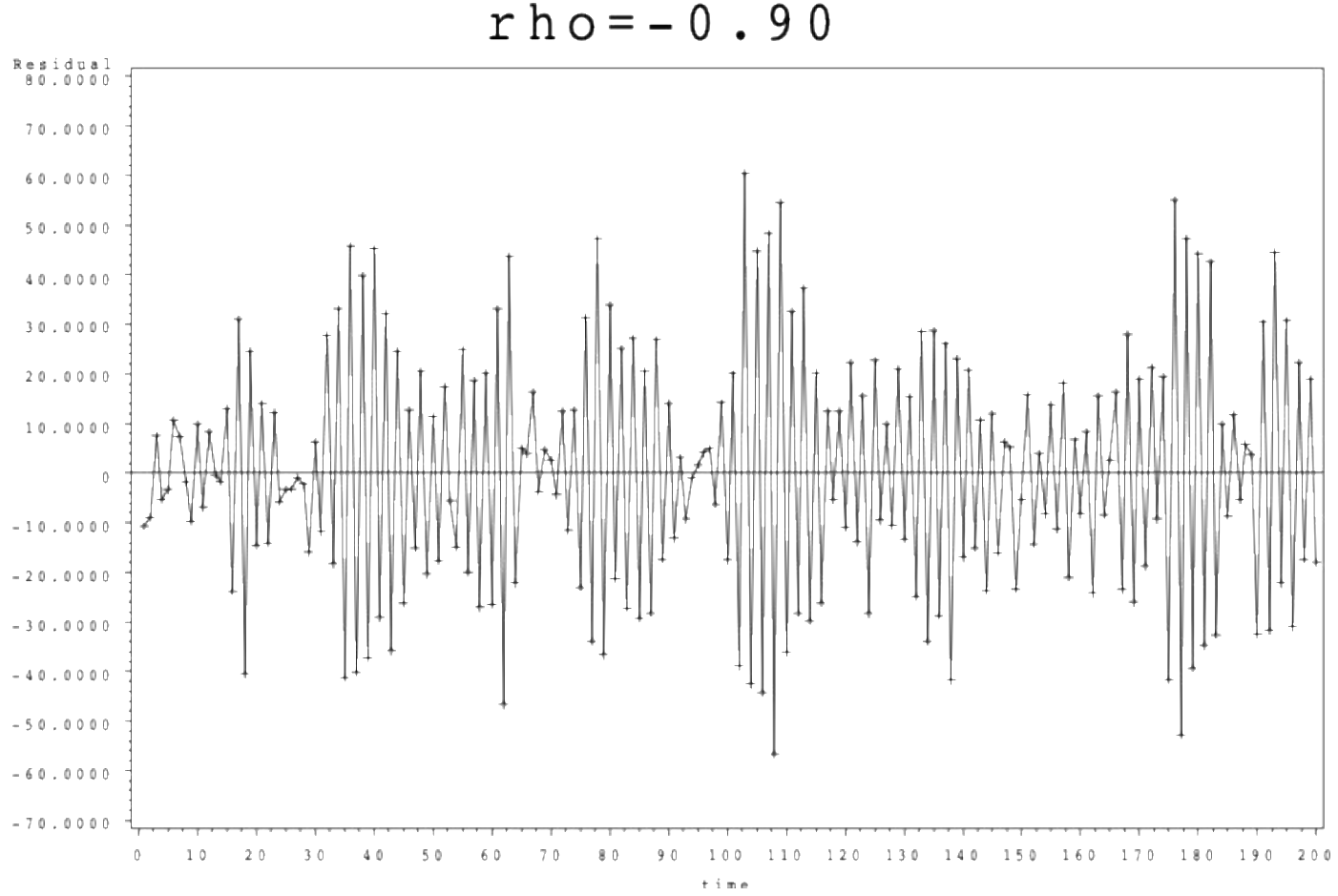
Values of auto-correlation close to 0 imply that the series jumps around the mean in a random fashion - also called **white noise**. There is no autocorrelation present in this data below.



Recall that this is what you want the residuals of a fitted model to resemble.

Autocorrelation

Negative values of autocorrelation implies that successive observations tend to oscillate above and below the mean in a sawtooth pattern. (Note that this sawtooth pattern should not be confused with seasonal effects.)



A value of rho (ρ) of -0.90 is interpreted as strong negative autocorrelation.

Partial autocorrelation

- **By now, many of you may have wondered: if we have first order autocorrelation i.e. X_t is correlated with X_{t+1} which is correlated with X_{t+2} etc., doesn't that imply that X_t and X_{t+g} will also be correlated?**

- In other words, if A is highly correlated to B, and B is highly correlated to C, then A is usually highly correlated to C as well. It would thus be useful to understand the direct relation between A and C, i.e. exploring what dependency there is in excess to the one associated to B. In a time series context, this is exactly what the partial autocorrelation measures.

- It is the correlation between two variables under the assumption that we know and consider the values of some other set of variables i.e**. it measures the excess correlation at lag g that is not accounted for by the first g - 1 lags**. i.e. the partial autocorrelation is the association between $\mathrm{X_t}$ and $\mathrm{X_{t+h}}$ with the linear dependence of $\mathrm{X_{t+1}}$ through $\mathrm{X_{t+g-1}}$ removed.

- Note that the 1st order partial autocorrelation is defined to be equal to the 1st order autocorrelation.

STATIONARITY

- *A key idea in time series is that of **stationarity***. Roughly speaking, a time series process is stationary if its behaviour does not change over time. This means, for example, that the values always tend to vary about the same level (mean) and that the variability is constant over time. It does not matter when you observe the series, it will look much the same at any point in time.

- ***If we wish to make predictions, then clearly we must assume that something does not vary with time***. Unfortunately, this is not always the case. Real-life data are often not stationary: e.g. they exhibit a trend over time, or they have a seasonal effect.

- Stationarity is a hypothesis which needs to be evaluated for every series. We may be able to reject this hypothesis with quite some certainty if the data strongly speak against it. However**, we can never prove stationarity with data**. At best, it is plausible that a series originated from a stationary process.

- Note: stationarity is a property of the process generating the data, and not actually the data itself.

- In general, a stationary time series will have no predictable patterns in the long-term. If we have a non-stationary time series (e.g. one with a trend or seasonal component) then we make it stationary by removing these components via a transformation (which we will consider later in the course)

The autocorrelation function (ACF)

- To assess the degree of dependence in the data and to select a model for the data that reflects this, one of the important tools we use is the sample autocorrelation function (sample ACF) of the data:

$$\rho(g) = \frac{\sum_{t=g+1}^{T}(y_t-\bar{y})(y_{t-g}-\bar{y})}{\sum_{t=1}^{T}(y_t-\bar{y})^2}$$

- The value of the sample autocorrelation function (aka the autocorrelation coefficient) at:
  - lag g = 1 is calculated using n-1 pairs of data (y_1, y_2), (y_2, y_3), (y_3, y_4),…(y_{n-1}, y_n)
  - lag g = 2 is calculated using n-2 pairs of data (y_1, y_3),(y_2, y_4),…,(y_{n-2}, y_n)
  - lag g = 3 is calculated using n-3 pairs id data (y_1, y_4),(y_2, y_5),…,(y_{n-3}, y_n)

    and so on…

Some Properties of the (theoretical) ACF:

- The ACF is an even function of lag i.e. $\rho(g) = \rho(-g)$. Therefore, the ACF is often considered just for p ≥ 0;
- It has the usual property of correlation that $|\rho(g)| \leq 1$ ∀ (for all) h i.e. $-1 \leq \rho(g) \leq 1$

The CORRELOGRAM (ACF plot)

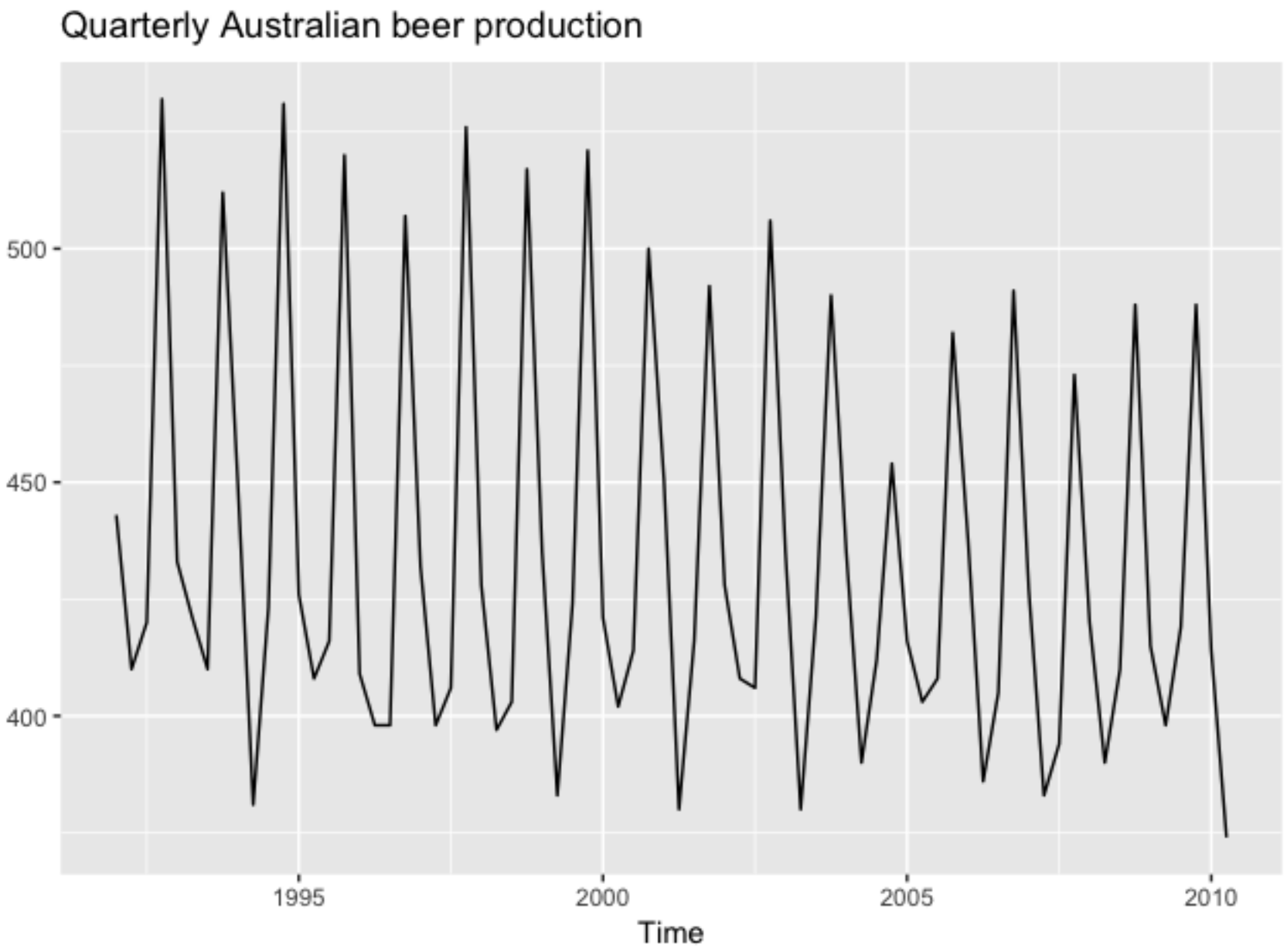
- ***The correlogram is probably the most useful tool in time-series analysis after the time series plot.*** **It is constructed by *graphing* the autocorrelation at lag g against g.** When constructing a correlogram, it has become a widely accepted standard to use vertical spikes for displaying the estimated autocorrelations.

- It can be used for two purposes: (i) either as a relatively simple descriptive tool of the dependency within a data series, OR (ii) as part of a more general procedure for identifying an appropriate model for a given time series.

- It is an important tool in assessing the behaviour and properties of a time series. It is typically plotted for the original series and also after differencing or transforming the data as necessary to make the series stationary and approximately normally distributed (more on this later).

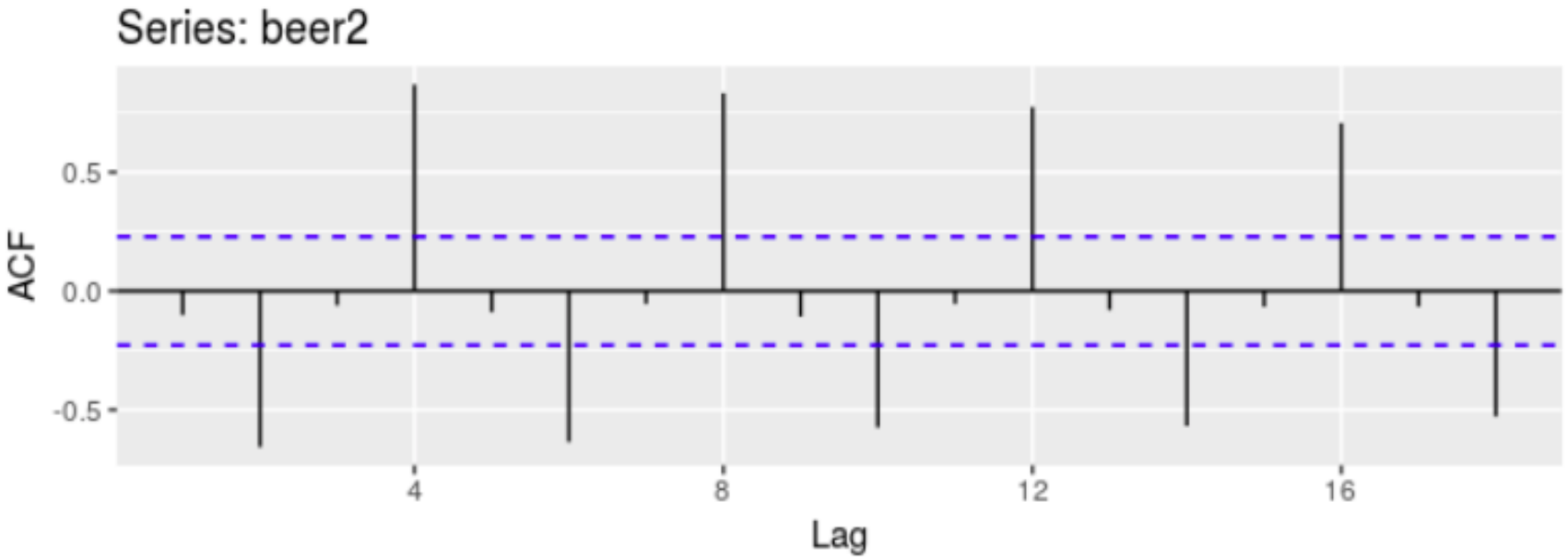
- For data from a stationary process, it can be shown that the correlogram generally provides an estimate of the theoretical ACF. It follows that, for data from a non-stationary process, the correlogram does not provide an estimate of anything! In that case, the values in the correlogram will typically not come down to zero except at high lags, and the only merit of the correlogram is to indicate that the series is not stationary.

The CORRELOGRAM (ACF plot) --- Example



The CORRELOGRAM (ACF plot) --- Example

```
ggAcf(beer2)
```

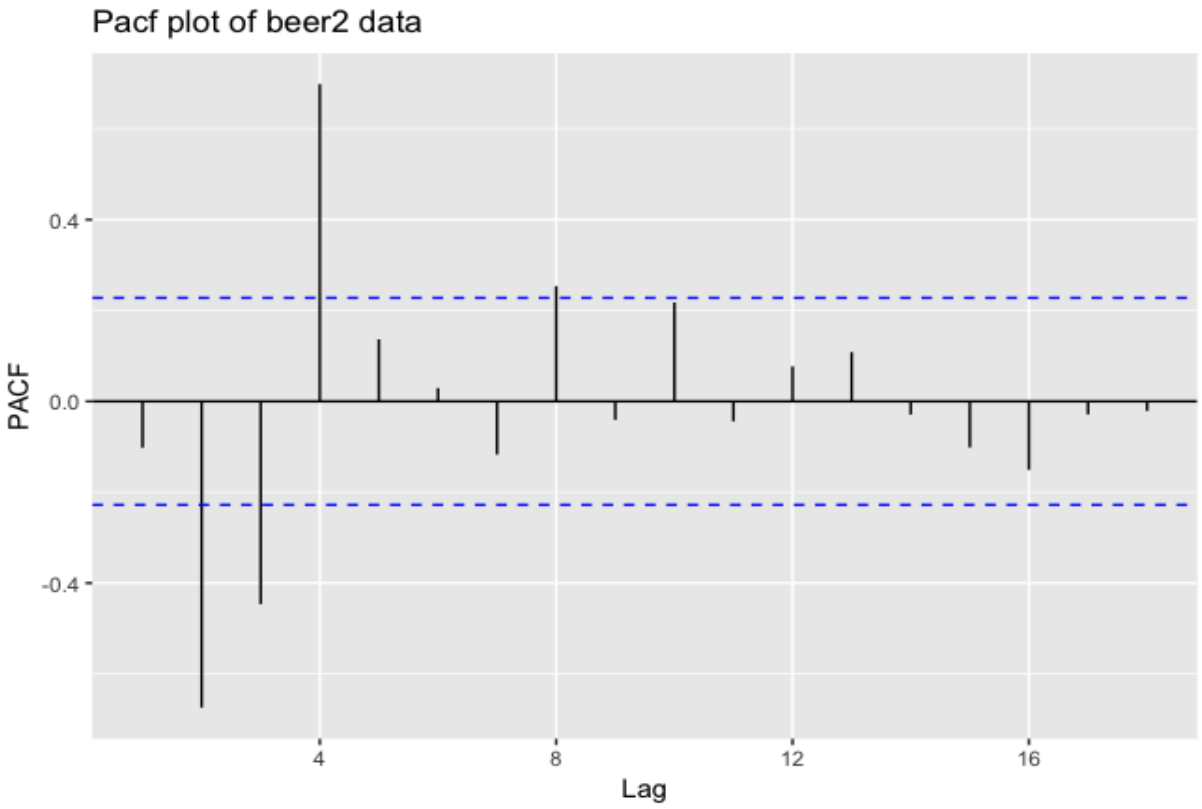


We see strong positive autocorrelation at lag 4, and every 4 lags after that. How do we know if the autocorrelation at lags 8,12,16 etc. is indicative of a relationship between observations at those lags OR is due to the autocorrelation at lag 4?

Similarly, we see moderate negative autocorrelation at lag 2, and every 4 lags after that. How do we know if the autocorrelation at lags 6,10,14 etc. is indicative of a relationship between observations at those lags OR is due to the autocorrelation at lag 2?

Plot the PACF to find out!

(PACF plot) --- Example



Note how different this plot looks to the ACF plot! What we can learn from this plot is that there is a moderate negative partial autocorrelation between values 2 time periods apart and a strong positive partial autocorrelation between values 4 time periods apart. This confirms that there is a direct relationship/dependence/autocorrelation between observations at these lags, independent of any relationship (autocorrelation) at lower lag values.

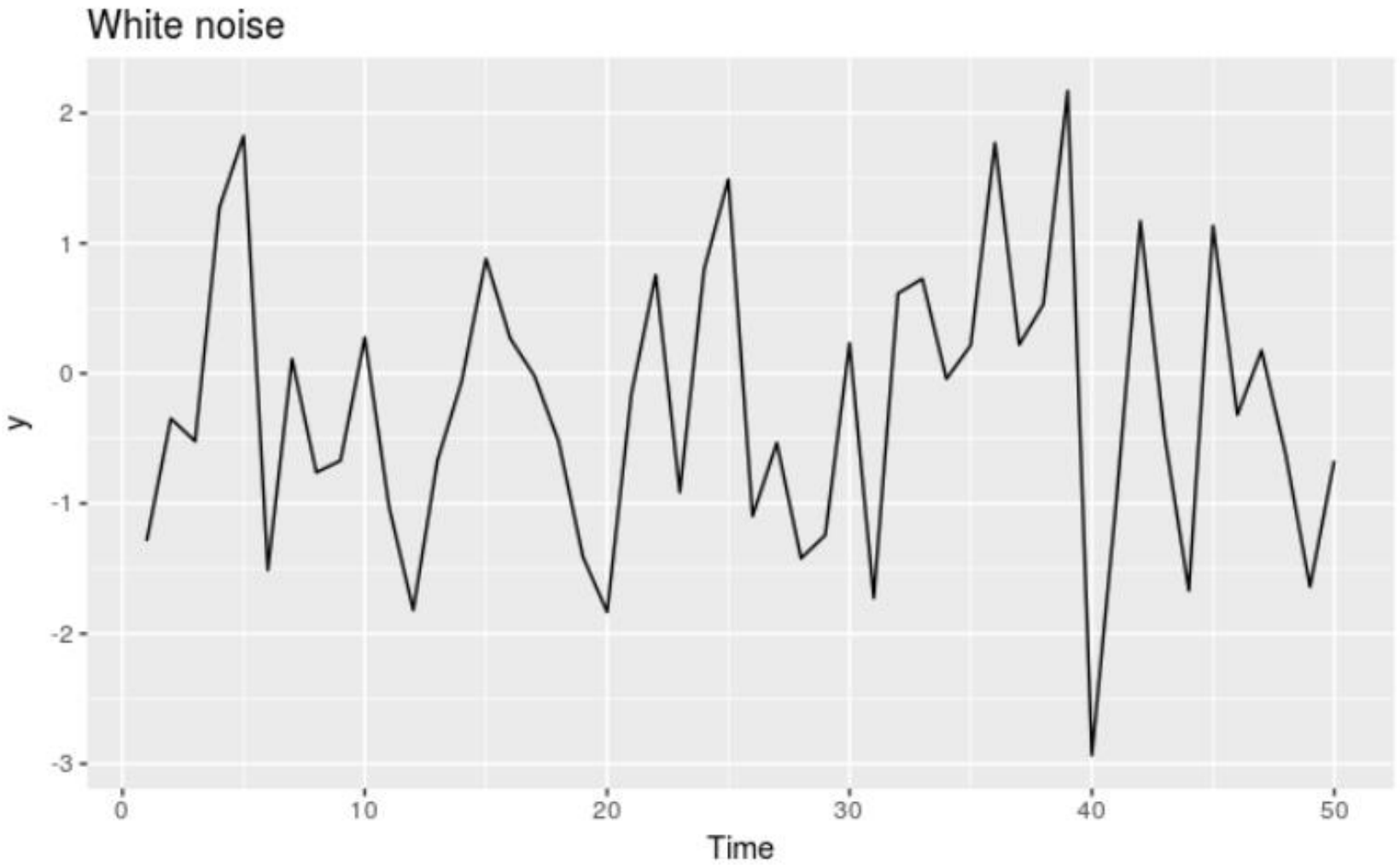
However, considering these partial autocorrelations at lags 2 and 4, we see that there is little/no direct relationship between observations 6 or 8 time periods apart. So that means what we saw at lags 6,10,14 etc. in the ACF was largely due to the autocorrelation at lag 2. Similarly, what we saw at lags 8,12,16 etc. in the ACF was largely due to the autocorrelation at lag 4. Again, this is due to the phenomenon of autocorrelation at lower lag values 'propagating' itself to higher lag values.

White noise process

- The term "white noise" refers to a stationary time series (i.e. no trend or seasonal component) where the observations are independent and identically distributed about a common mean with no autocorrelation at any lag i.e. **a time series that displays no autocorrelation**
- White noise is the implicit assumption made when doing regression analysis. The assumption that the residuals have mean 0 and constant variance and are independent implies that the residuals are white noise.
- Since there is no dependence between observations, knowledge of previous values is of no value for predicting the future behaviour of the series i.e. there is no available information in this data.
- *It should begin to make intuitive sense why we would want the residuals of any fitted model to resemble white noise – if they do, it implies that the model has captured most of the information/patterns that was available in the data. Conversely, if there is still autocorrelation present in the residuals, it implies that the fitted model has failed to capture all relevant information/patterns available in the data.*

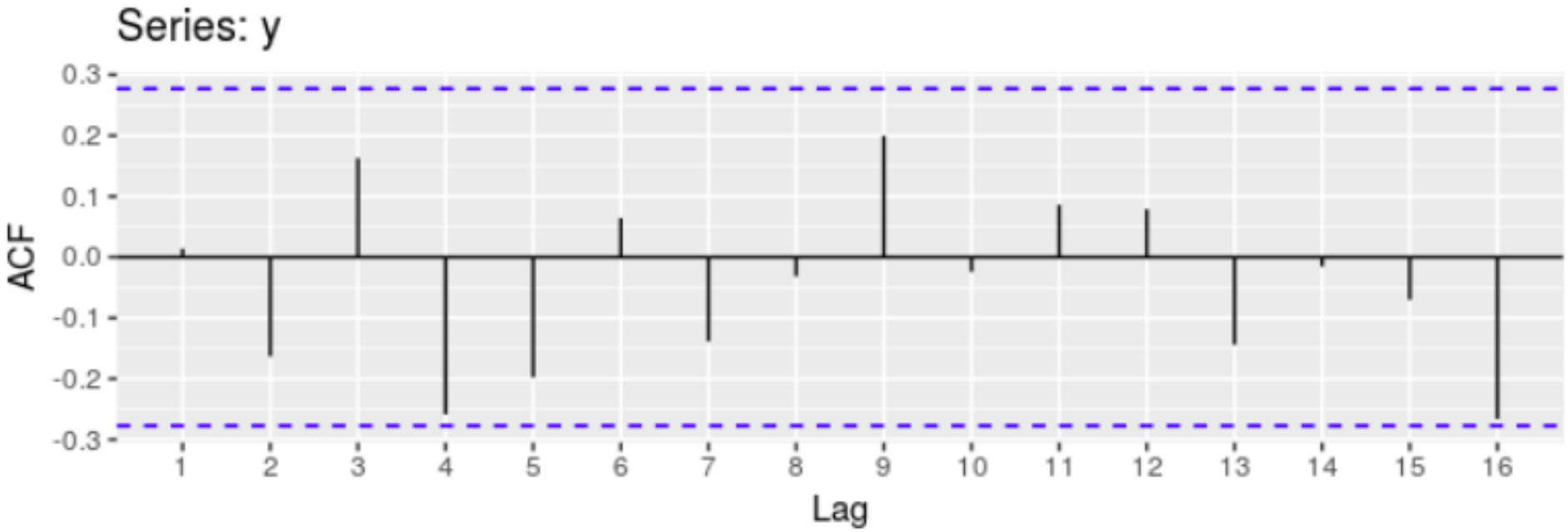
White noise process

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



White noise process

```
ggAcf(y)
```



Question: But what do the blue dashed lines mean???!!! Well, I'm glad you asked…

The CORRELOGRAM (ACF plot)

- The dashed lines on a correlogram represent the *95% Confidence Bands (i.e. upper and lower limits) of the hypothesis that* $\rho(h) = 0$ for all h (i.e. the hypothesis that there is no autocorrelation)

- It is obvious that even for a time series without any autocorrelation, and thus $\rho(h) = 0$ for all h, $\hat{\rho}\,(h)$ will generally not be 0 for all h. Hopefully they will be close, but the question is how close? The answer is provided by the confidence bands.

- These so-called confidence bands are obtained from an important result: for a long-time white noise series (i.e. values are independent and identically distributed), it can be shown that the $\hat{\rho}\,(h)$ approximately follow a N(0,1/T) distribution. Thus, each $\rho(h)$ lies within the interval of $\pm 1.96/\sqrt{T}$ with a probability of approximately 95%.

- This leads us to the following statement that facilitates interpretation of the correlogram: for any stationary time series, sample autocorrelation coefficients $\hat{\rho}\,(h)$ that fall within the confidence band $\pm 1.96\,/\sqrt{T}$ are considered to be different from 0 *only by chance*, while those outside the confidence band are considered *to be truly different from 0*.

- On the other hand, the above statement means that even for a white noise series, we expect 5% of the estimated ACF coefficients to exceed the confidence bounds (these correspond to type 1 errors) i.e. 1 in 20 ACF coefficients may be “significant” only by chance

The CORRELOGRAM (ACF plot)

- Please NOTE that while a white noise series of values will always be stationary, ***STATIONARITY of a time series does NOT imply INDEPENDENCE in the data*** i.e. you can have stationary time series that still exhibits autocorrelation.

- Interpreting a correlogram is one of the hardest tasks in time-series analysis, especially when T is less than 100 and the sample autocorrelations have relatively large variance.

- For seasonal series, there is likely to be a large positive value of $\rho(g)$ at the seasonal period, and this may still be present to a (much) lesser extent even after the seasonal effect has supposedly been removed. Thus the correlogram is often used to see if seasonality is present

- For series with a trend, the correlogram will not come down to zero until a high lag has been reached, perhaps upwards of half the length of the series. The correlogram provides little information in the presence of trend other than as an indicator that some form of trend-removal is necessary to make the series stationary.

The CORRELOGRAM (ACF plot)

USING THE CORRELOGRAM TO DETECT/CONFIRM NON-STATIONARY TIME SERIES

NB *- ACF Behaviours of common time series:*

- The ACF of a stationary time series decays to zero rapidly.
- The ACF of a time series with trend exhibits slow decay as the lag g increases.
- The ACF of a time series with seasonality displays the same wave-like structure as the series, with large autocorrelation coefficients at lag values equal to the period (length) of the seasonal variation

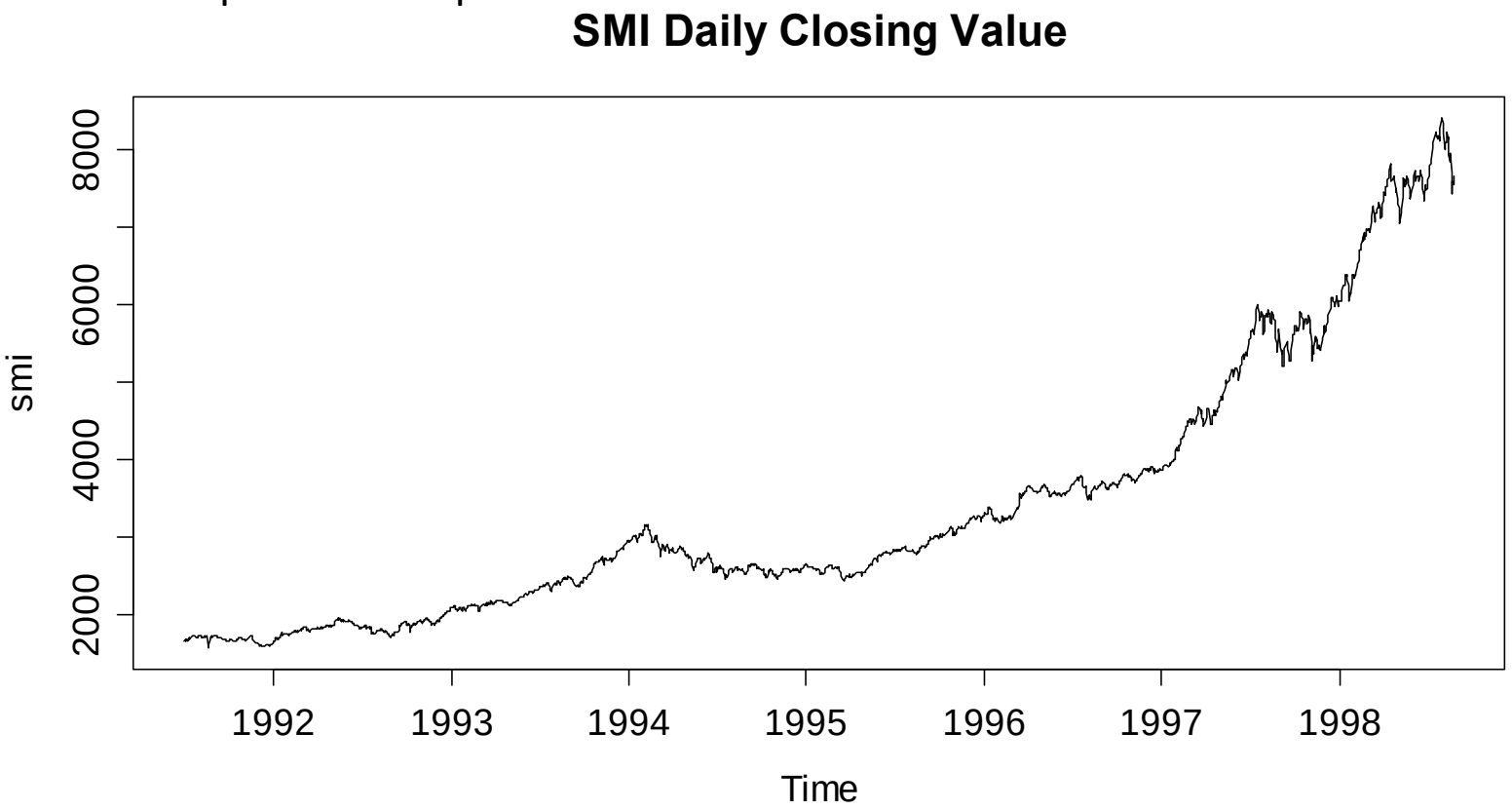
- Identification of a trend in a series using ACF
- We observe that the ACF decays very slowly. The reason is that if a time series features a trend, the observations at consecutive points in time will usually be on the same side of the series' global mean $\bar{y}$.

- For small to moderate lags p, most of the terms $(y_{t+g} - \bar{y})(y_t - \bar{y})$ are positive. For this reason, the sample autocorrelation coefficient will be positive as well, and is most often also close to 1

The CORRELOGRAM (ACF plot)

USING THE CORRELOGRAM TO DETECT/CONFIRM NON-STATIONARY TIME SERIES -
Identification of a trend in a series using ACF

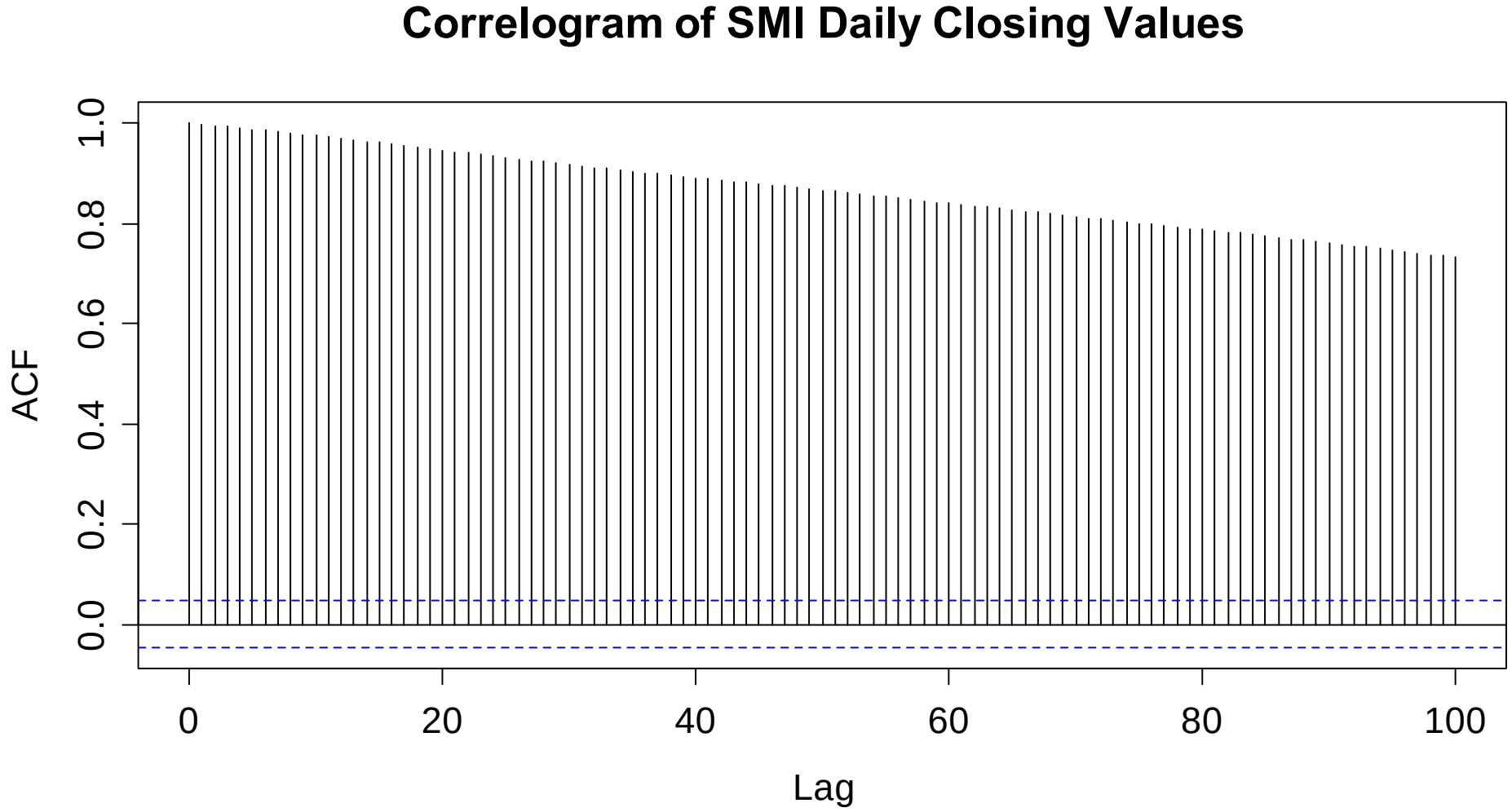
- Thus, a very slowly decaying ACF is an indicator for non-stationarity, i.e. a trend that was not removed before autocorrelations were estimated. For example, consider the daily closing values of the Swiss Market Index (SMI). It summarises the value of the shares of the 20 most important companies:



The CORRELOGRAM (ACF plot)

USING THE CORRELOGRAM TO DETECT/CONFIRM NON-STATIONARY TIME SERIES -

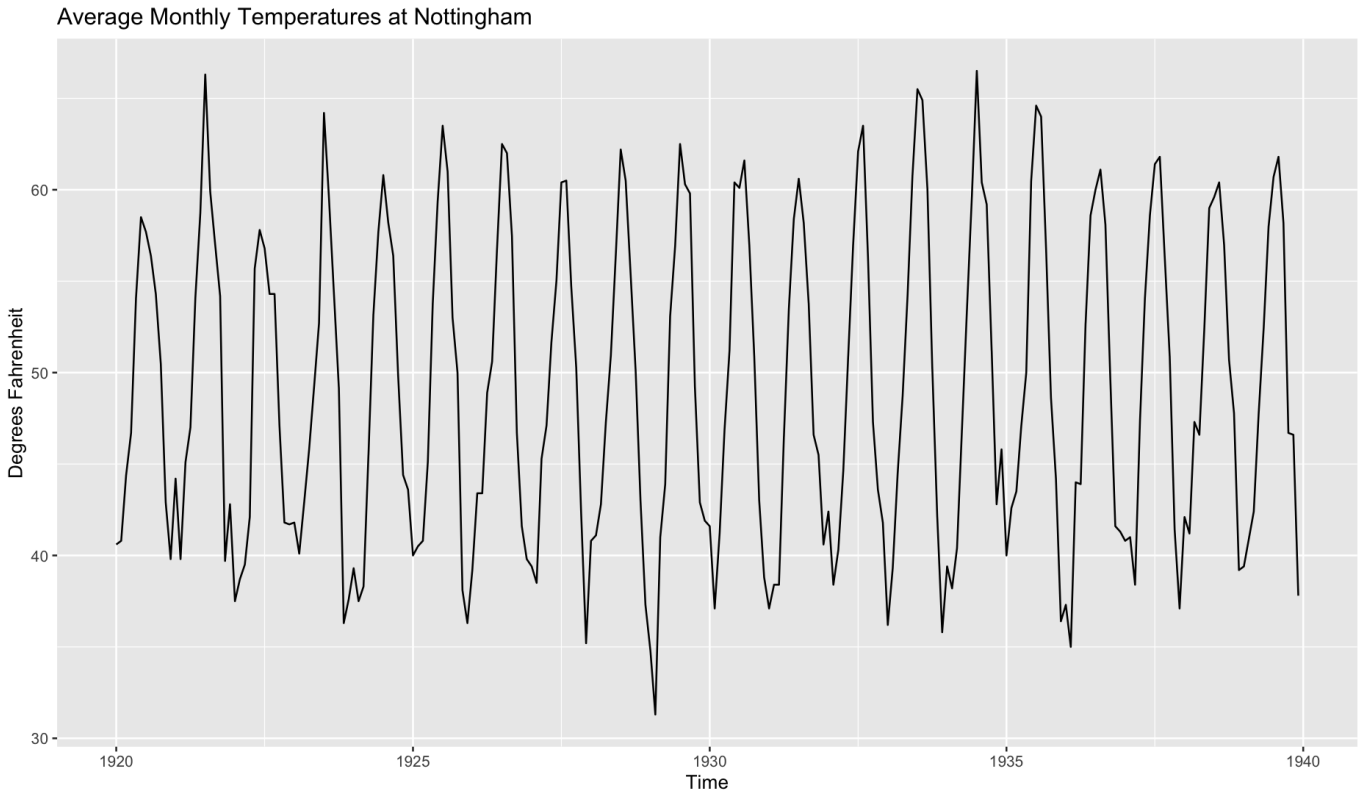
Identification of a trend in a series using ACF



The CORRELOGRAM (ACF plot)

USING THE CORRELOGRAM TO DETECT/CONFIRM NON-STATIONARY TIME SERIES -
Identification of a seasonal effect using ACF

The ACF is "seasonal", and owing to the recurring seasonality, again decays very slowly. Also note that for seasonal series, the correlogram often has periods rather than lags on the x-axis – often a matter of confusion. We conclude that a very slowly decaying periodicity in the correlogram is an indication of a seasonal effect that was not removed. For example:

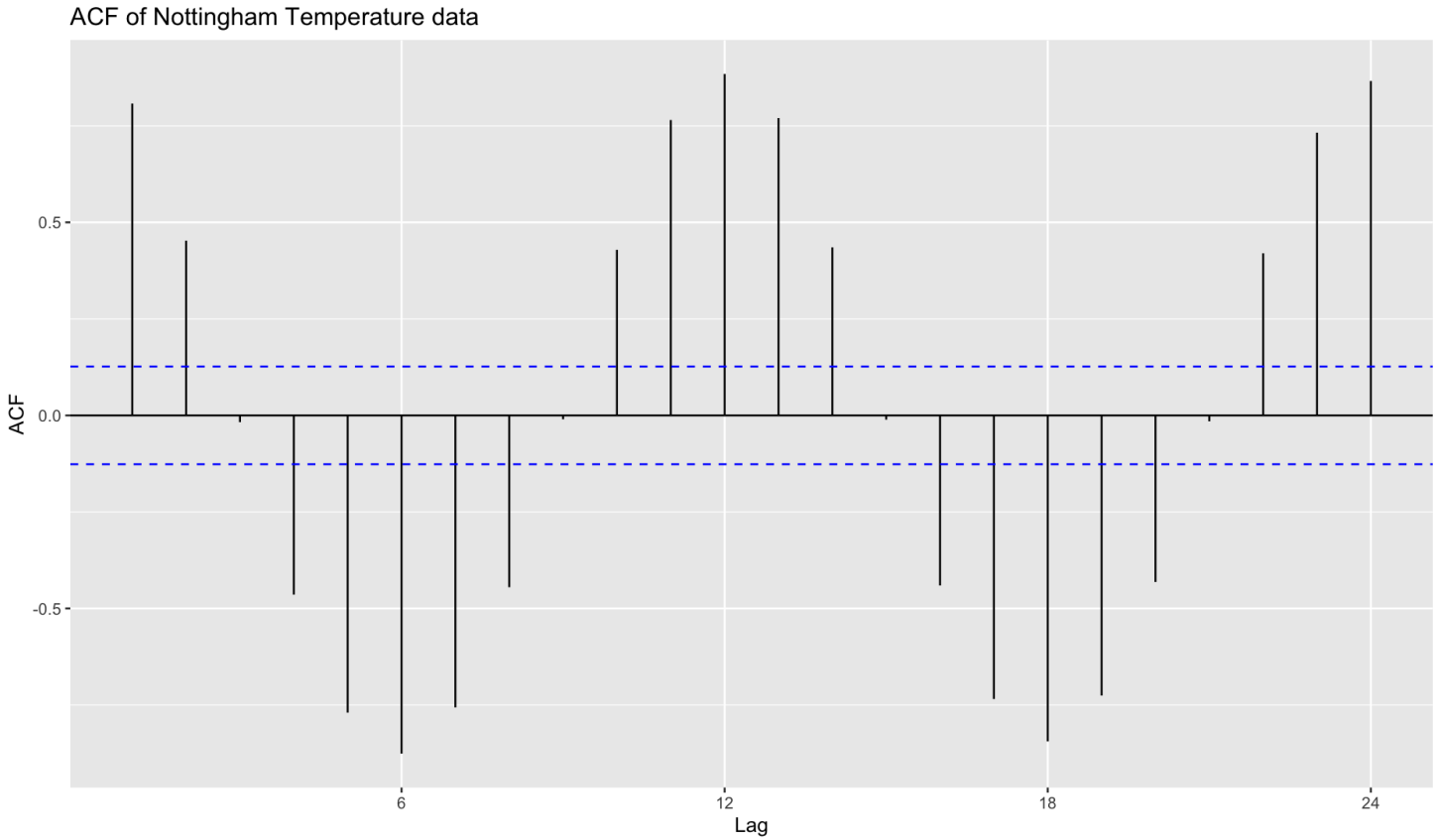


Q: What components are present in this time series?

Q: Is this time series stationary?

The CORRELOGRAM (ACF plot)

USING THE CORRELOGRAM TO DETECT/CONFIRM NON-STATIONARY TIME SERIES -
Identification of a seasonal effect using ACF

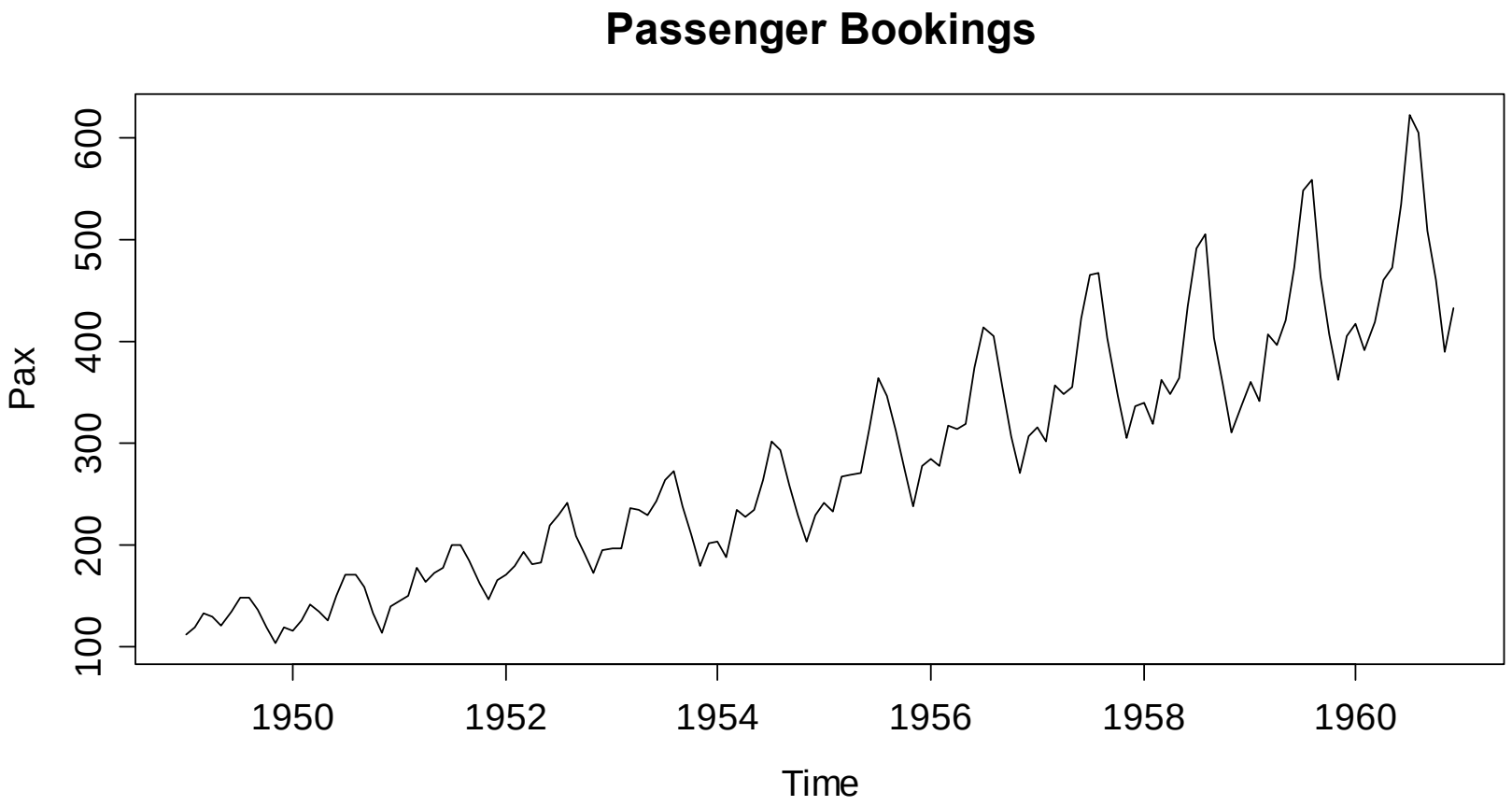


Note the strong positive autocorrelation at lags 12 and 24, and the strong negative autocorrelation at lags 6 and 18. How would you interpret those? *How would you determine if there is a direct relationship between observations at lag 24, or if the autocorrelation there is due to the autocorrelation at lag 12? How much of the autocorrelation at lag 18 is due to the autocorrelation at lag 6?*

The CORRELOGRAM (ACF plot)

USING THE CORRELOGRAM TO DETECT/CONFIRM NON-STATIONARY TIME SERIES –
ACF for series with trend and seasonality

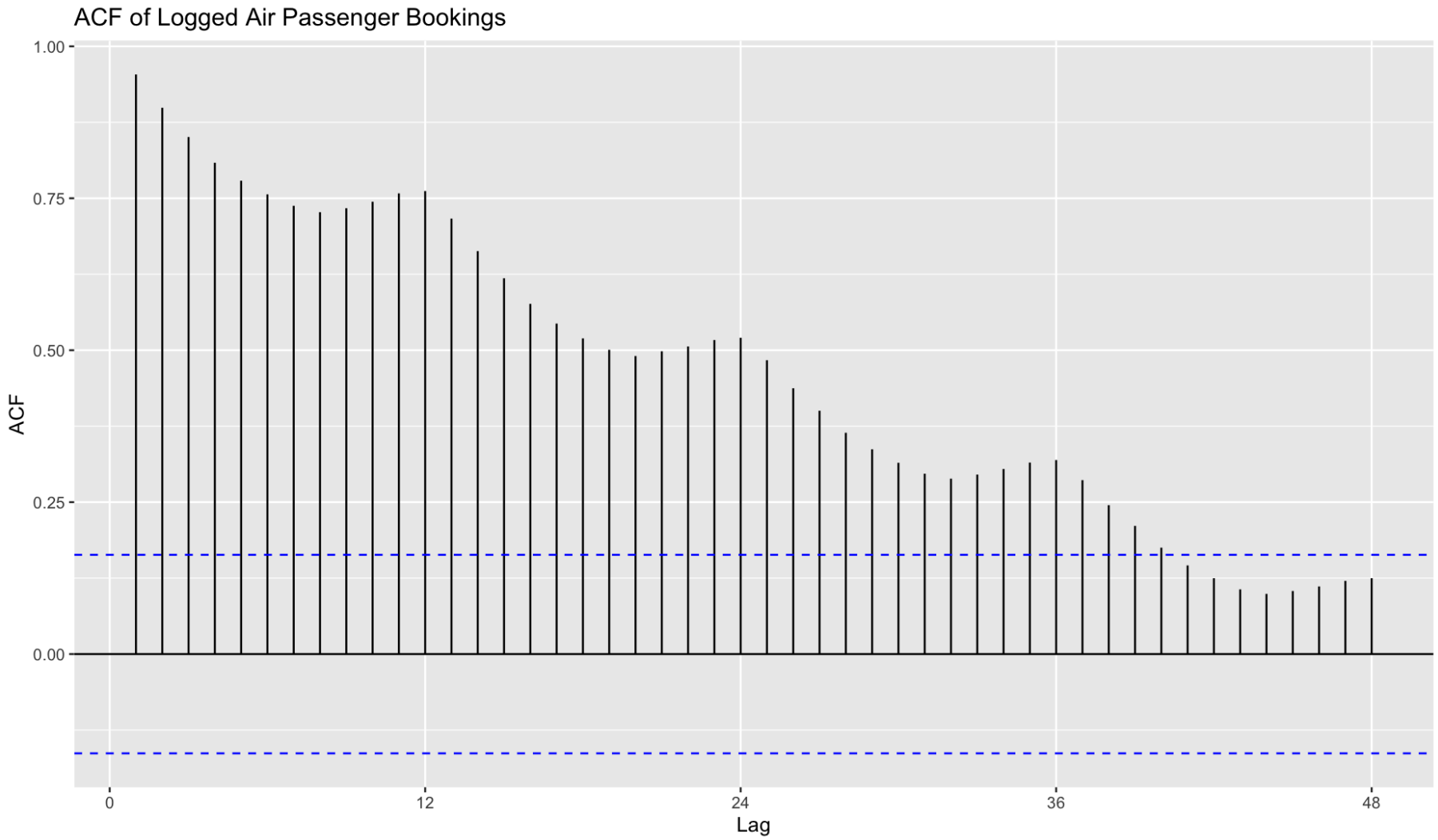
Here, the two effects described above are interspersed. We have a slow decay in the general level of the ACF, plus some seasonality. Again, this is an indication of a non-stationary series. For example:



The CORRELOGRAM (ACF plot)

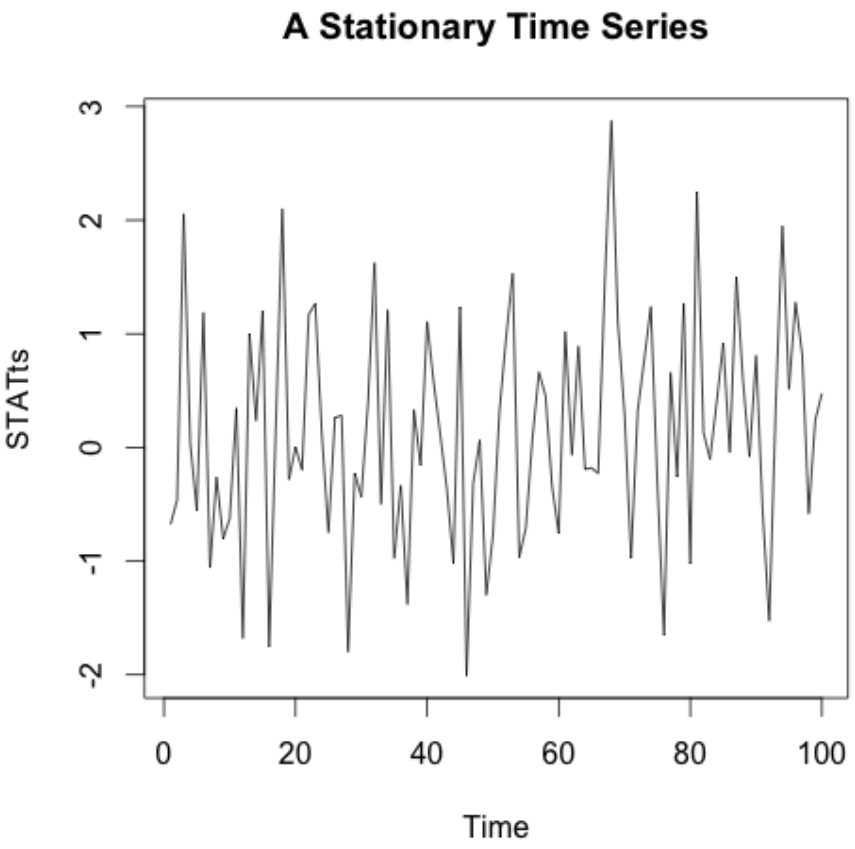
USING THE CORRELOGRAM TO DETECT/CONFIRM NON-STATIONARY TIME SERIES –

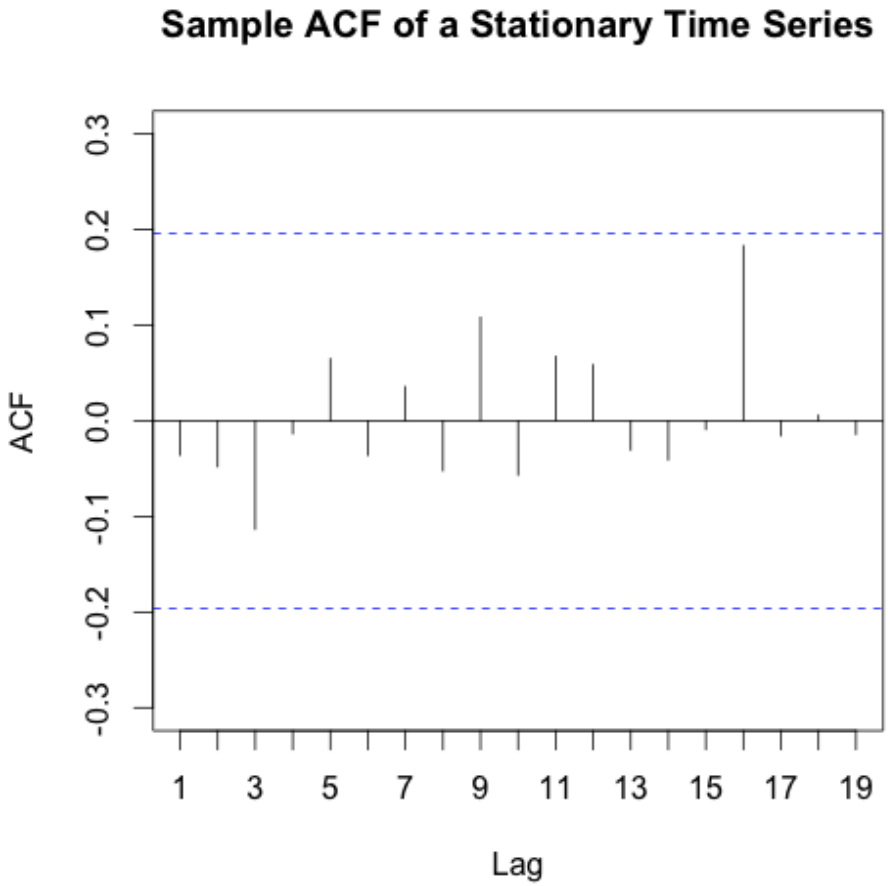
ACF for series with trend and seasonality



The CORRELOGRAM (ACF plot)

THE CORRELOGRAM OF A STATIONARY TIME SERIES – Example 1

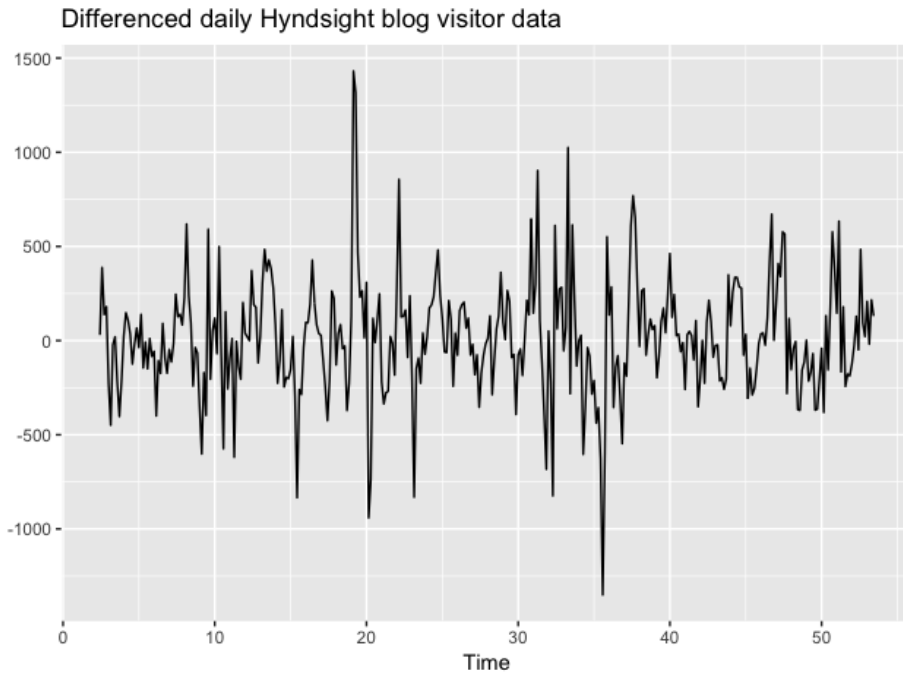


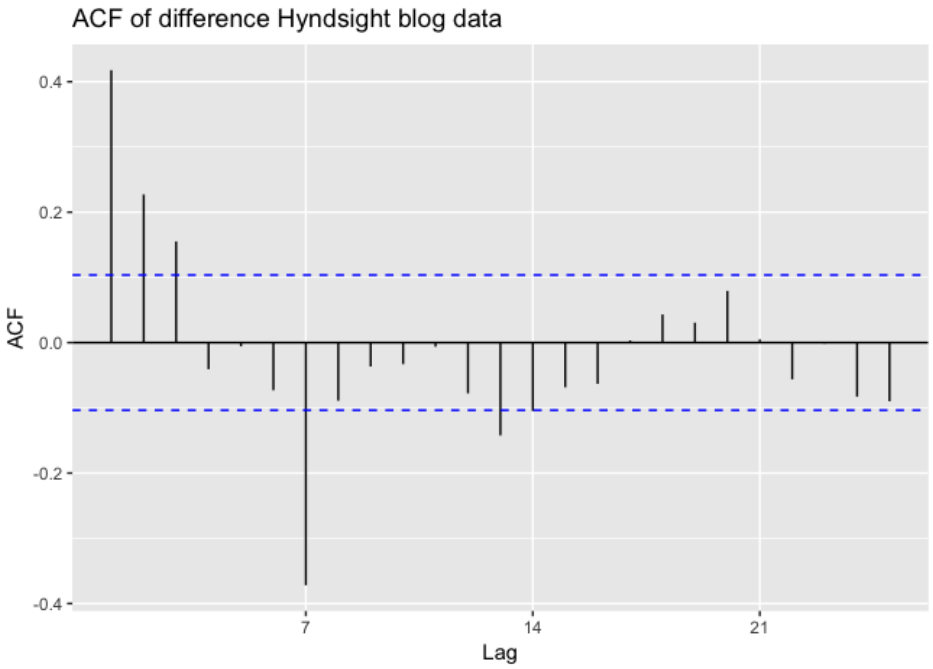


This stationary time series is white noise – can you explain why?

The CORRELOGRAM (ACF plot)

THE CORRELOGRAM OF A STATIONARY TIME SERIES – Example 2





Note that although the time series is plausibly second order stationary (constant mean and variance), **there is still autocorrelation present in the data** – indicated by the spikes at lags 1, 2, 3, 7 and 13.

Again, stationarity does NOT imply white noise!

The CORRELOGRAM (ACF plot)

THE CORRELOGRAM – EFFECT OF OUTLIERS AND ITS INTERPRETATION

- ACF and outliers: Large outliers can have a significant impact on the autocorrelation and hence the ACF. However, with a time series, you cannot simply omit a value as you will then break the evenly spaced intervals, so you must choose to either leave the outlier or:
  - i) Replace it with global mean
  - ii) Replace it with a local mean
  - iii) Use model-based imputation by forecasting (not covered in course)

- Interpreting the correlogram: The correlogram can be tricky to interpret at best, sometimes even misleading, or sometimes even wrong. ***However***, *it is the best means we have for understanding the dependency in a time series*. And we will base many if not most of our decisions in the modelling process on the correlogram.

ARIMA modelling

Capturing the nature of the autocorrelation in the data

(fpp) Chapter 9:
Sections 9.1, 9.3–9.8

Non-seasonal ARIMA models

- ARIMA models provide another approach to time series forecasting. Exponential smoothing and ARIMA models are the two most widely-used approaches in time series forecasting. They provide complementary approaches to the problem.

- While exponential smoothing models are based on a description of the trend and seasonality in the data, ARIMA models aim to describe the autocorrelation in the data.

- Thus, if we want to focus on fitting a model based simply on the pattern of the data (i.e. trend and seasonality) then we would look to Exponential smoothing models. However, if we are interested in finding a model that captures the nature of the autocorrelation in the data, ARIMA models are what we use.

- We have continuously emphasised that, in practice, many time series such as those related to socio-economic and business processes show non-stationary behaviour.

- ARIMA models are very useful for modelling a wide variety of stationary and non-stationary time series processes. ARIMA stands for '**A**uto**R**egressive **I**ntegrated **M**oving **A**verage' and refers to a family of models that are mixtures of autoregressive, AR(p) and moving average MA(q) models.

- Note that the moving average MA(q) models are NOT the same thing as moving average smoothing that we considered previously

Non-seasonal ARIMA models

- If a non-stationary time series is modelled by both an AR(p) and a MA(q) process, we say that it is modelled by an ARIMA(p, d, q) process. It takes the form:

$$X_t = \underbrace{c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p}}_{\text{AR(p) part}} + \underbrace{\epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}}_{\text{MA(q) part}}$$

- The "I" in ARIMA stands for "integrated", which basically means you're transforming the data by taking differences. (Integration is the reverse of differencing – which refers to the process of back-transforming your forecasts of your transformed data to the scale of the original time series.

- We say an ARIMA model has been "Integrated" at the order d (e.g., the dth difference) if we have had to apply a dth-order difference to the non-stationary time series in order to make it stationary.

- The ARIMA model is typically written as ARIMA(p, d, q) where:
  - p is the number of autoregressive terms
  - d is the order of differencing
  - q is the number of moving average terms

- For example, an ARIMA(1,1,0) is a first-order AR model with one order of differencing. An ARIMA(2,1,1) refers to a mixed model consisting of two AR terms and one MA term that has been fitted to a time series that has been differenced once

General Approach to Modelling a Time Series [that may have significant autocorrelation at some lag g]

The approach outline below serves as an outline to the general process to follow when performing a time series analysis when we suspect autocorrelation is present.

1. *Plot the series and examine the main features of the graph*, checking whether there is changing variance, a trend, a seasonal component, any apparent sharp changes in behaviour or any outlying observations.

TRANSFORM the series if it is non-stationary:
2. By stabilising the variance (if necessary) and

3. Removing any trend /seasonal components via differencing at the appropriate lag to get a stationary series

4. Choose an ARIMA(p,d,q) model to fit the transformed data/time series, based on the nature of the autocorrelation of the transformed data/ time series, by making use of various sample statistics including plots of the ACF and PACF. Note that it is unlikely you will be able to specify THE best ARIMA(p,d,q) model from simply examining the ACF and PACF plots. In practice, it is common to:

a. Estimate various ARIMA(p,d,q) models for a whole range of orders $0 \leq p \leq P$ and $0 \leq q \leq Q$
b. Compare all the models by information criteria (AIC)

General Approach to Modelling a Time Series [that may have significant autocorrelation at some lag g]

5. *Estimate the parameters of that model* (Computer software uses different methods available to do this e.g. MME, MLE) – **We do NOT concern ourselves with manually estimating the parameters in this course i.e. we let *RStudio* do the job!!!**

6. *Perform diagnostic checks of the residuals of the model*:
(i) Independence of error terms, (ii) Mean of 0, (iii) Normality of error terms, (iv) Constant error variance (Homoscedasticity)

7. *Forecast future values of the time series* using an appropriate fitted model.

NOTE: In practice, there may be a range of similar models (with different parameters) that may be appropriate for a given time series. Generally, we fit a few slightly different models and then compare the accuracy of the fit of each model ***relative to the other models*** using some form of information criterion (AIC) and measures of forecast accuracy (MSE/RMSE and MAE) to select the most appropriate model for forecasting purposes.

We will now proceed to consider each of the AR(p) and MA(q) models separately before discussing how to fit ARIMA(p,d,q) models

Autoregressive, AR(p), models

- The observed value at time *t* depends linearly on the last *p* observed values, and the model looks like a regression model – hence the term autoregression
  - i.e. the future value of a variable is assumed to be a linear combination of *p* past observations and a random error together with a constant term.

- A process $\{X_t\}$ is said to be an autoregressive process of order p (abbreviated AR(p)) if:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \phi_3 X_{t-3} + \cdots + \phi_p X_{t-p} + \epsilon_t$$

- Where ϵ_t is a white noise process with mean 0 and variance σ^2 that represents the error series, and c is a constant *related* to the mean of the process. Note that often we deal with AR(p) processes that have a mean of 0, corresponding to $c = 0$.

- **NOTE:** The β_t's we used previously are replaced by the ϕ_t's here in order to distinguish between the use of AR(p) models when a regression model has been fit, and the more general use we will be discussing here. The parameters ϕ_t here are **NOT** the same as used in Holt's methods as a dampening parameter for the slope.

Autoregressive, AR(p), models

Autoregressive series are important because they have a natural interpretation — the next value observed is a slight perturbation of a simple function of recent observations. Autoregressive models are remarkably flexible at handling a wide range of different time series patterns, and have a wide application:

- The study of mortality rates
- GDP dynamics
- Climatological issues like the el nino effect and measuring solar radiation
- The study of volcanic tremors and brain electrical activity mapping
- They are frequently applied by financial investors

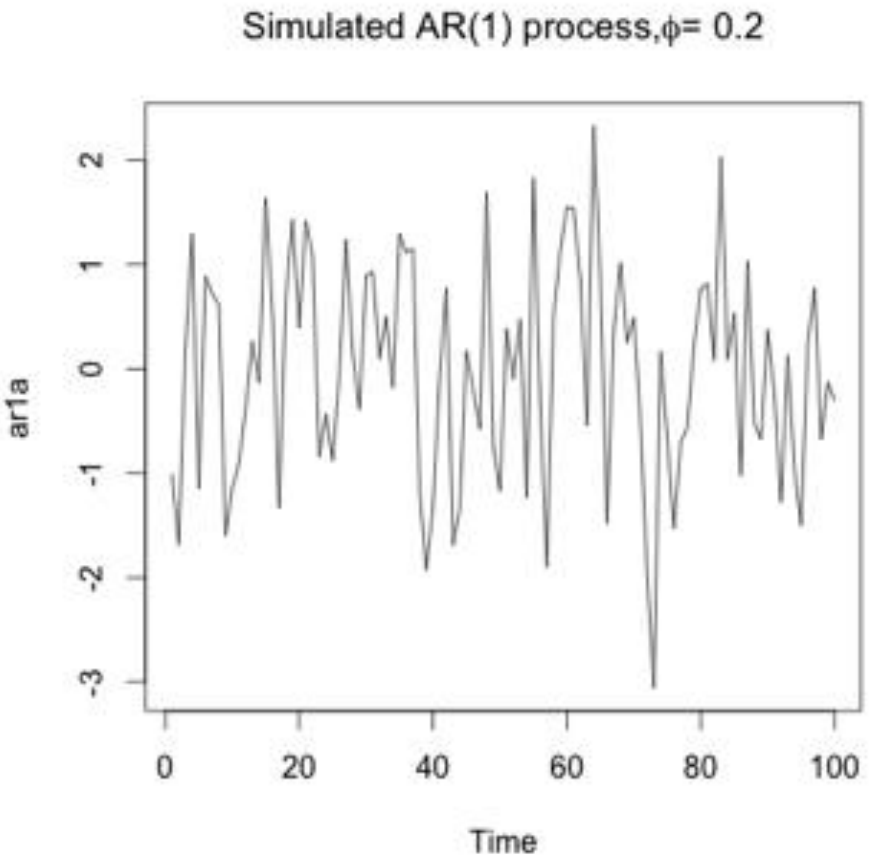
The AR(1) process: The simplest example of an AR process is the first-order case, denoted AR(1), given by:

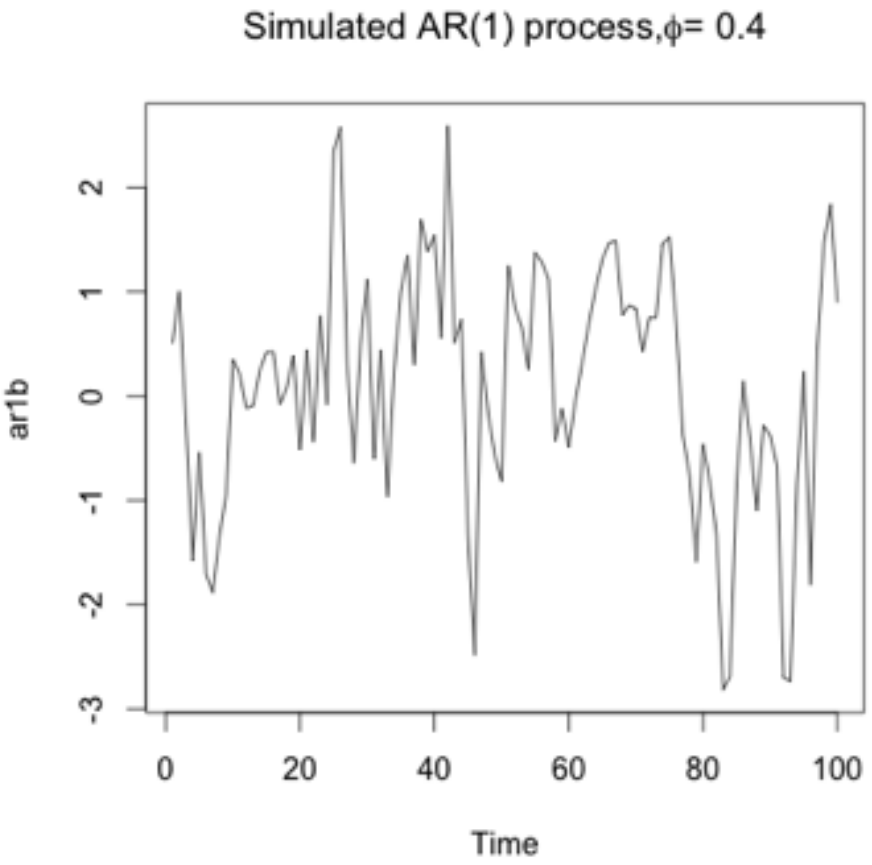
$$X_t = c + \phi_1 X_{t-1} + \epsilon_t$$

- We normally restrict autoregressive models to stationary data, in which case some constraints on the values of the parameters are required. The AR(1) process is stationary only when $|\phi_1| < 1$. As the value of ϕ_1 approaches either 1 or -1, the AR(1) process exhibits stronger non-stationary behaviour.

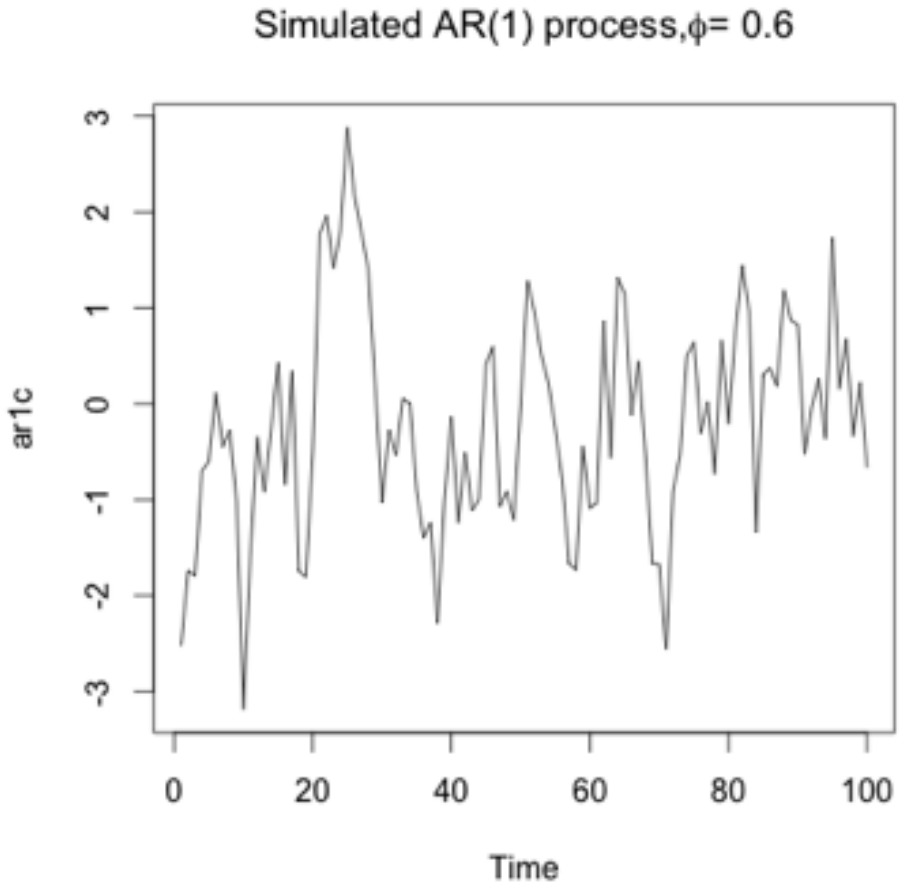
- Consider some examples of simulated AR(1) processes for different values of the coefficient ϕ_1:

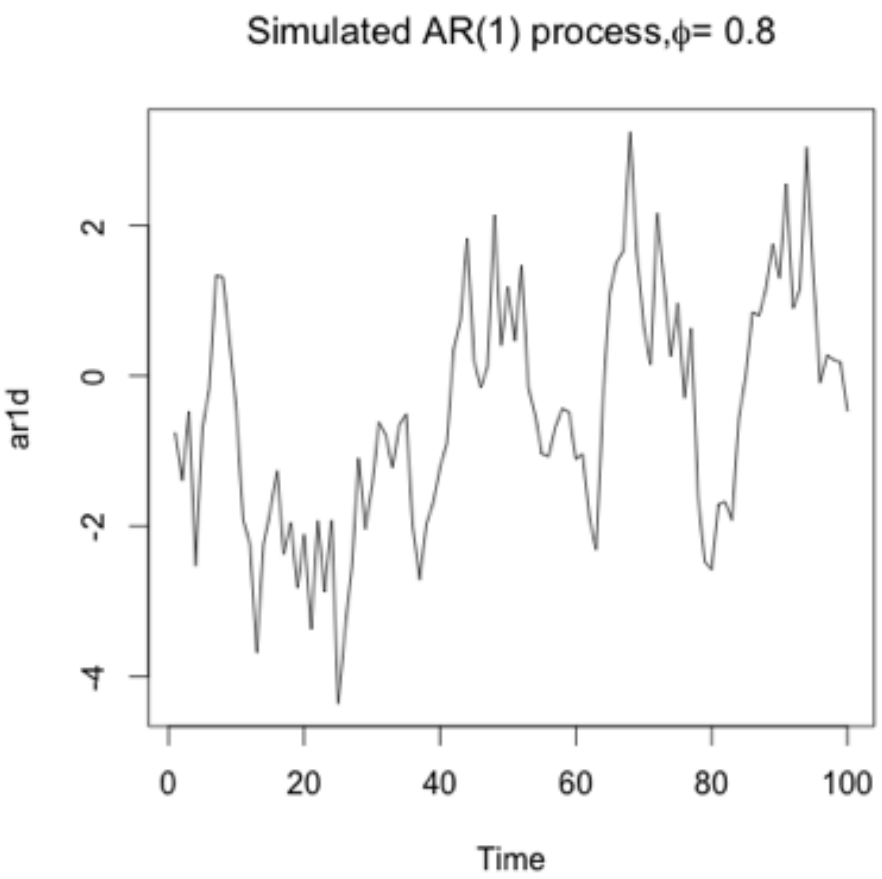
Autoregressive, AR(p), models



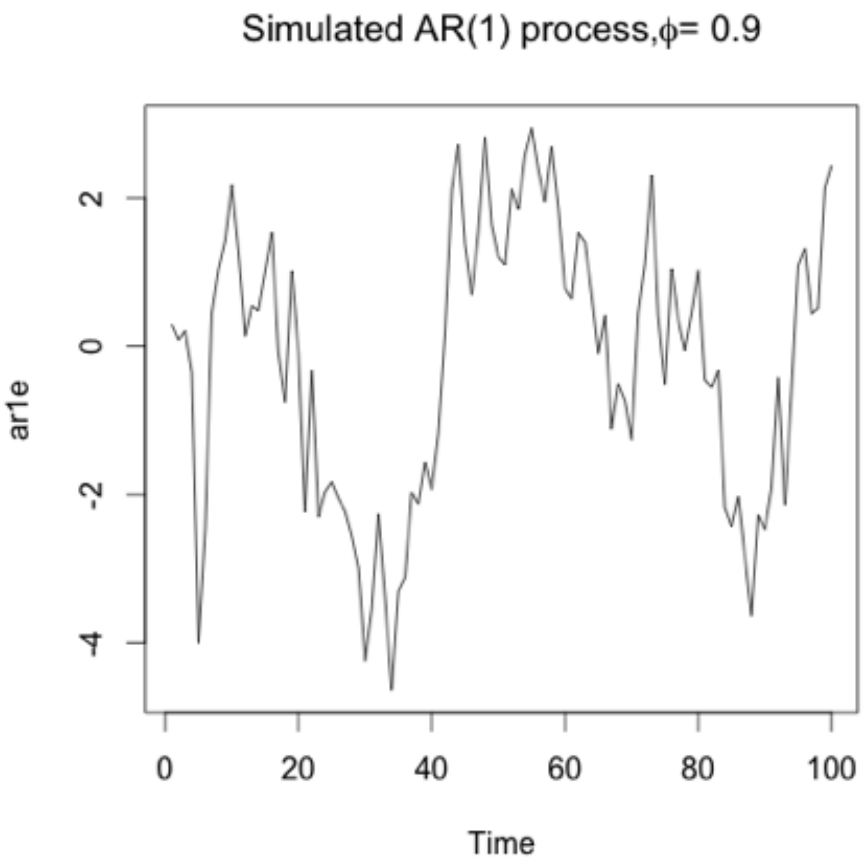


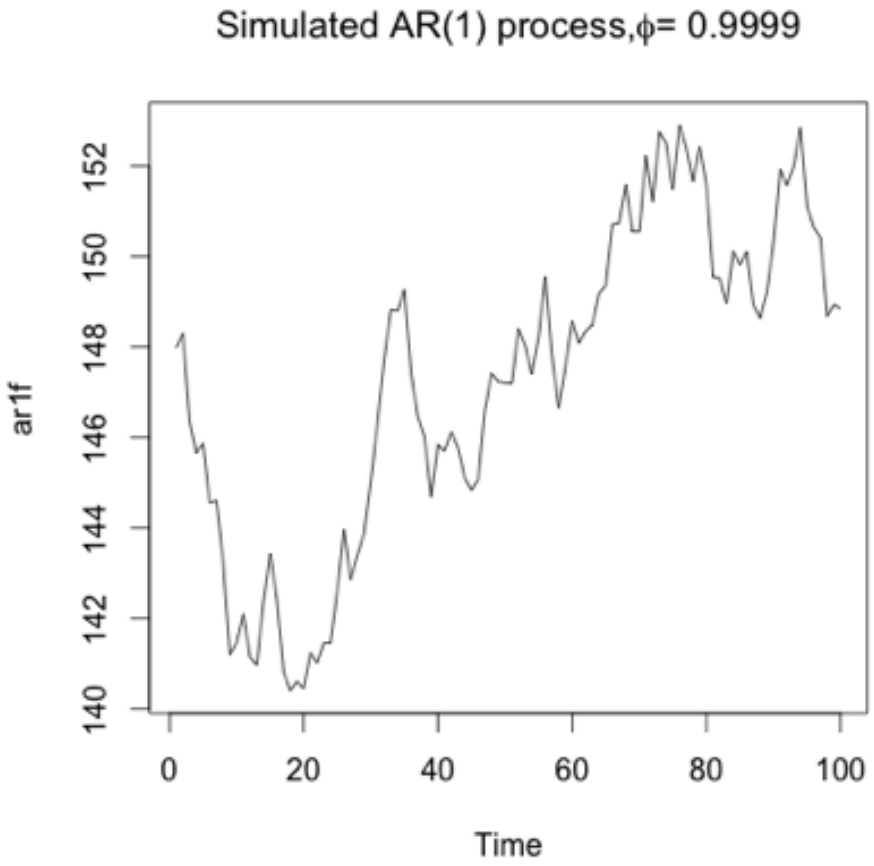
Autoregressive, AR(p), models





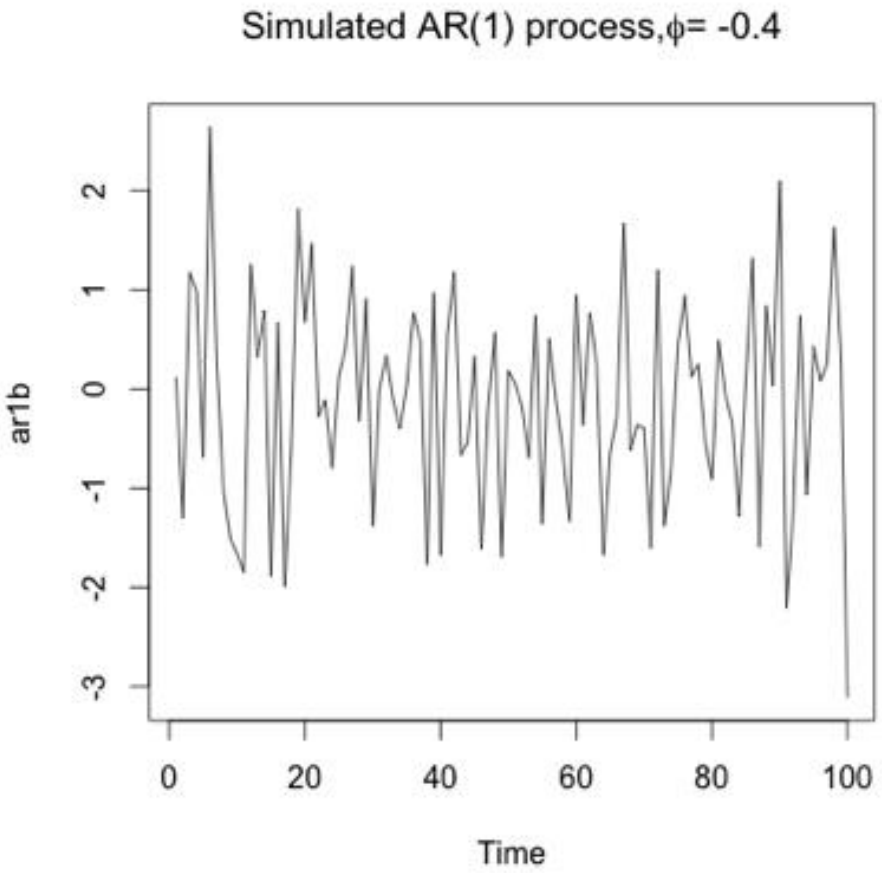
Autoregressive, AR(p), models

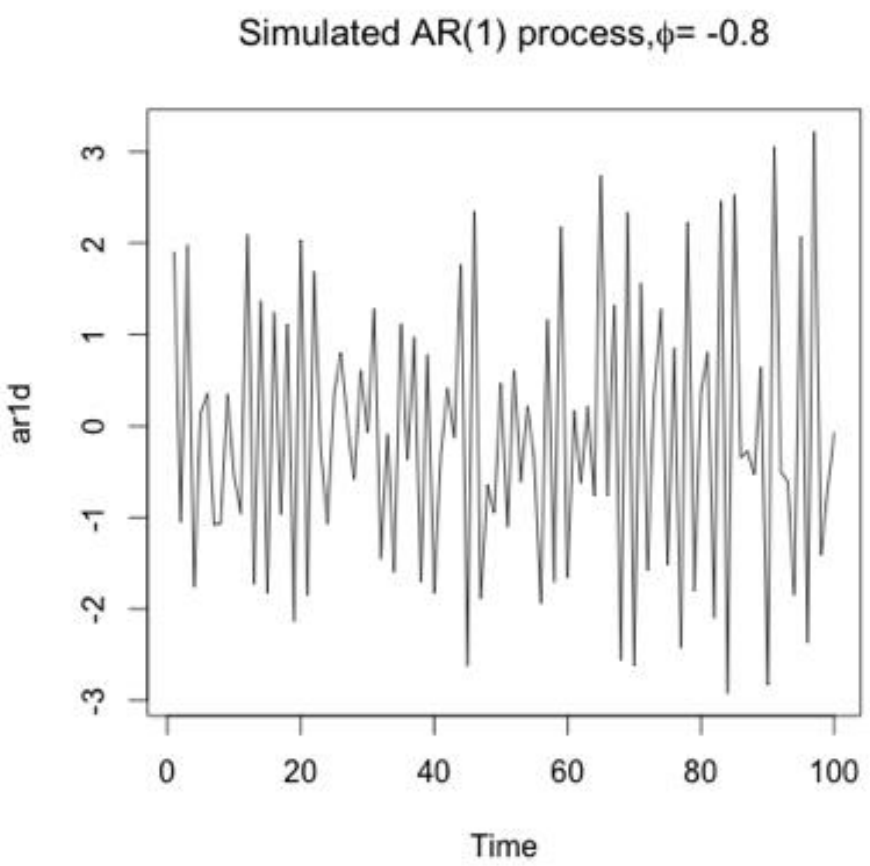




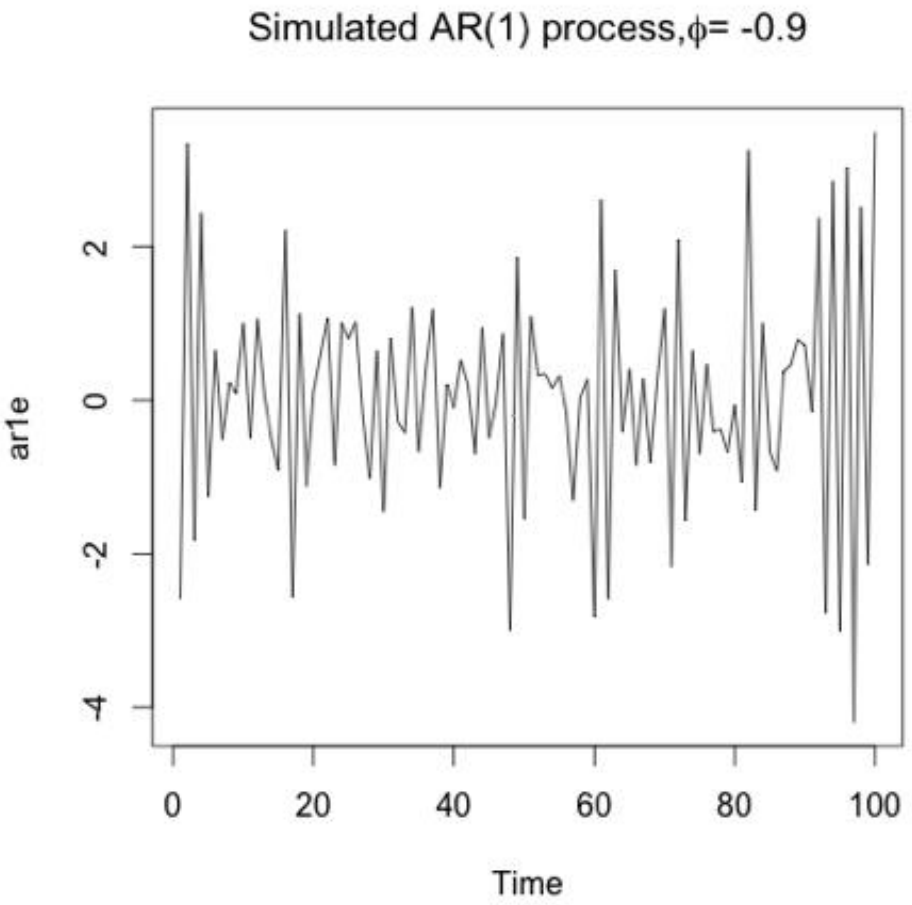
Note that all of the simulated AR(1) processes, barring the last one here, would be classified as stationary. However, as the value of the coefficient ϕ_1 approaches 1, the resulting time series becomes increasingly non-stationary in mean (i.e. an increasing trend becomes evident). For ϕ_1 = 1 you get a process referred to as a "random walk".

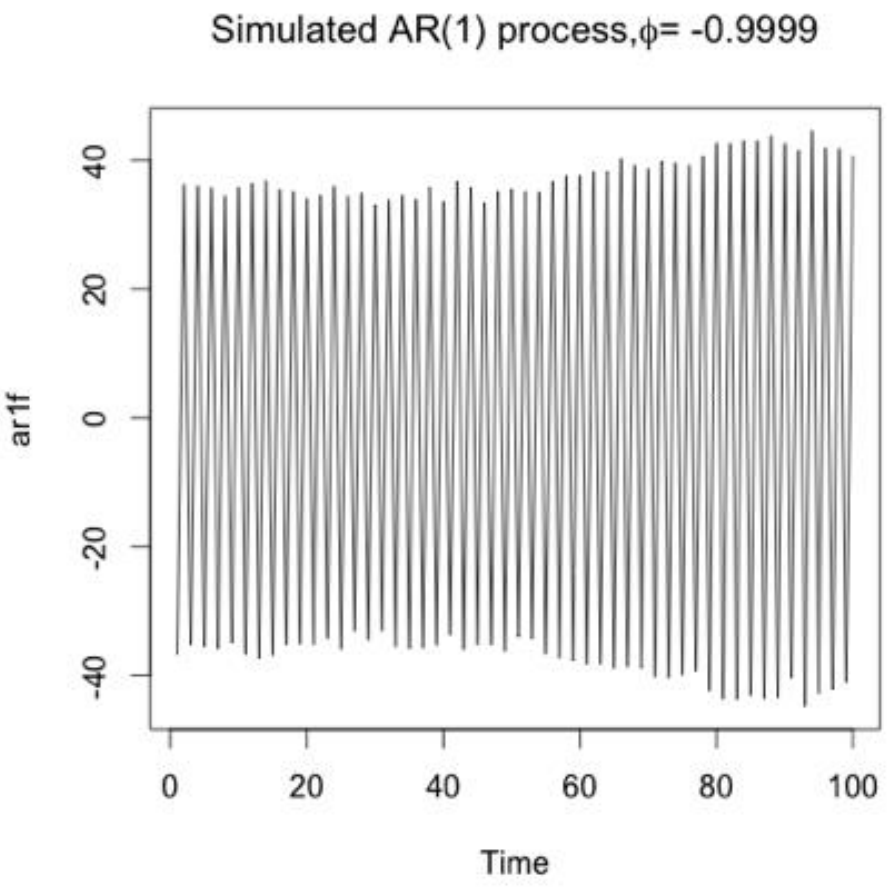
Autoregressive, AR(p), models





Autoregressive, AR(p), models





As the value of ϕ_1 approaches -1, the resulting time series becomes more non-stationary in its variance. For ϕ_1= -1, you get a process called a "random jump" which has a changing variance and is hence non-stationary. For $|\phi_1| > 1$ the mean increases exponentially and hence the AR process is non-stationary.

Autoregressive, AR(p), models

Properties of a stationary AR(1) process:

1. *The mean of a stationary AR(1) process is given by*:

$$E[X_t] = E(c + \phi_1 X_{t-1} + \epsilon_t) = c + \phi_1 E[X_{t-1}] + 0$$

Since the process is stationary, we have that $E[X_t] = E[X_{t-1}] = \mu$

$$\Longrightarrow E[X_t] = \mu = c + \phi_1 \mu$$

Therefore,

$$\mu - \phi_1 \mu = c$$

and finally:

$$\boldsymbol{\mu = \frac{c}{1-\phi_1}}$$

For the case where c $= 0$, , the above simply yields $E[X_t] = 0$. The corollary of this is also true – for AR(p) processes where the mean = 0, c = 0.

Autoregressive, AR(p), models

Properties of a stationary AR(1) process:

2. *The variance of a stationary AR(1) process*:

$$V[X_t] = V(c + \phi_1 X_{t-1} + \epsilon_t) = 0 + \phi_1^2 V[X_{t-1}] + \sigma^2$$

Since the process is stationary, we have that $V(X_t) = V(X_{t-1})$, and hence

$$V(X_t) - \phi_1^2 V(X_t) = \sigma^2$$

and finally:

$$\boldsymbol{V[X_t] = \frac{\sigma^2}{1-\phi_1^2}}$$

3. The ACF of a stationary AR(1) process: For the sake of simplicity, we consider an AR(1) process with mean 0 (i.e. c $= 0$):

$$X_t = \phi_1 X_{t-1} + \epsilon_t$$

If X_t is stationary, then there is only the autocorrelation at lag g=1: $\boldsymbol{\rho(1) = \phi_1}$

Autoregressive, AR(p), models

$$\rho(h) = \phi_1^h \qquad \text{h = 1,2,…,T}$$

NB: Thus, the ACF of an AR(1) model 'decays' exponentially to 0 as the lag g$\rightarrow \infty$

The following slides show some examples of the ACF of a stationary AR(1) process

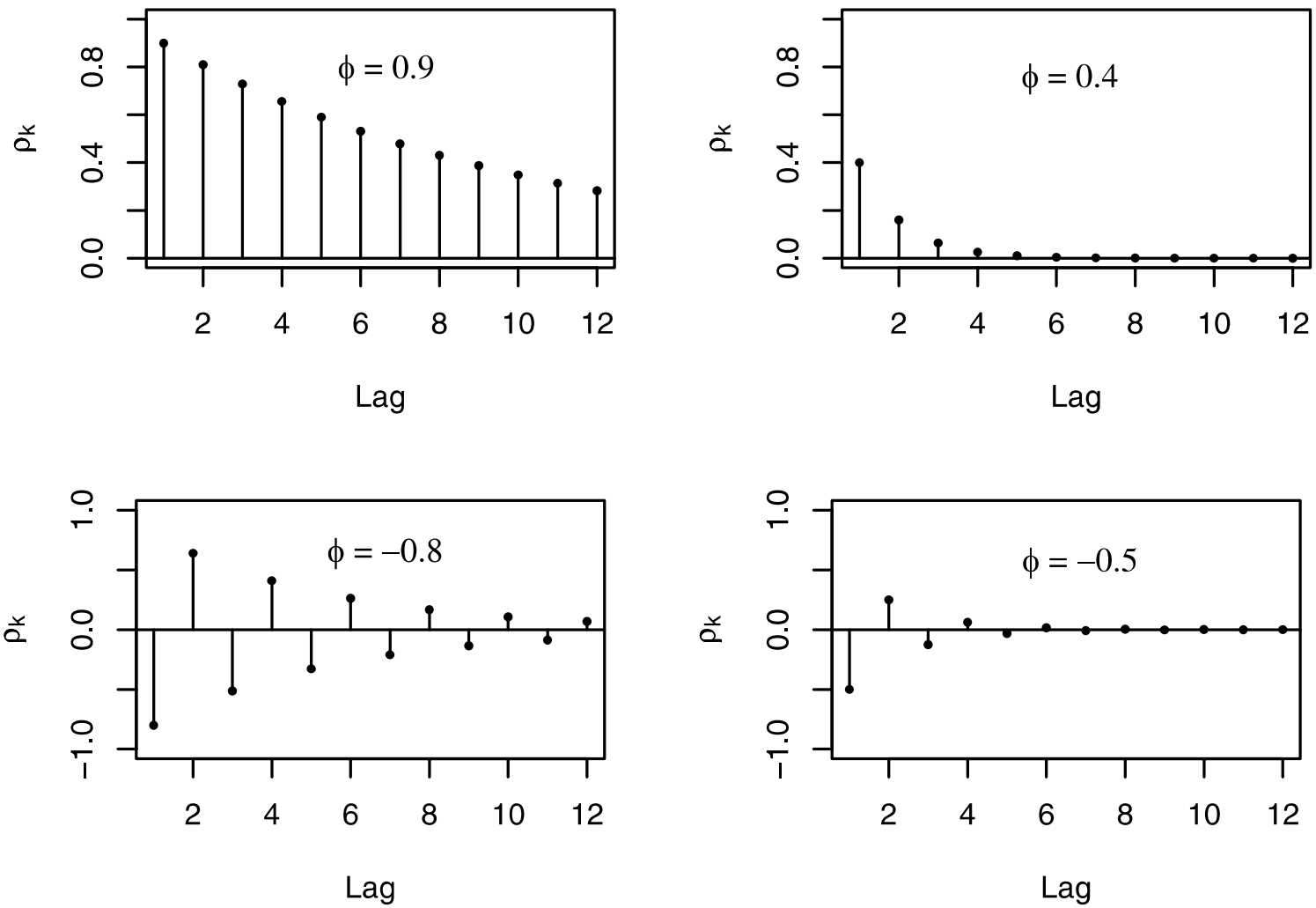
Properties of a **stationary AR(p) process**:

We can extend some of these results to an AR(p) process, where we obtain:

$$E[X_t] = \mu = \frac{c}{1 - \phi_1 - \phi_2 - \cdots - \phi_p}$$

Autoregressive, AR(p), models

Exhibit 4.12 Autocorrelation Functions for Several AR(1) Models



Autoregressive, AR(p), models

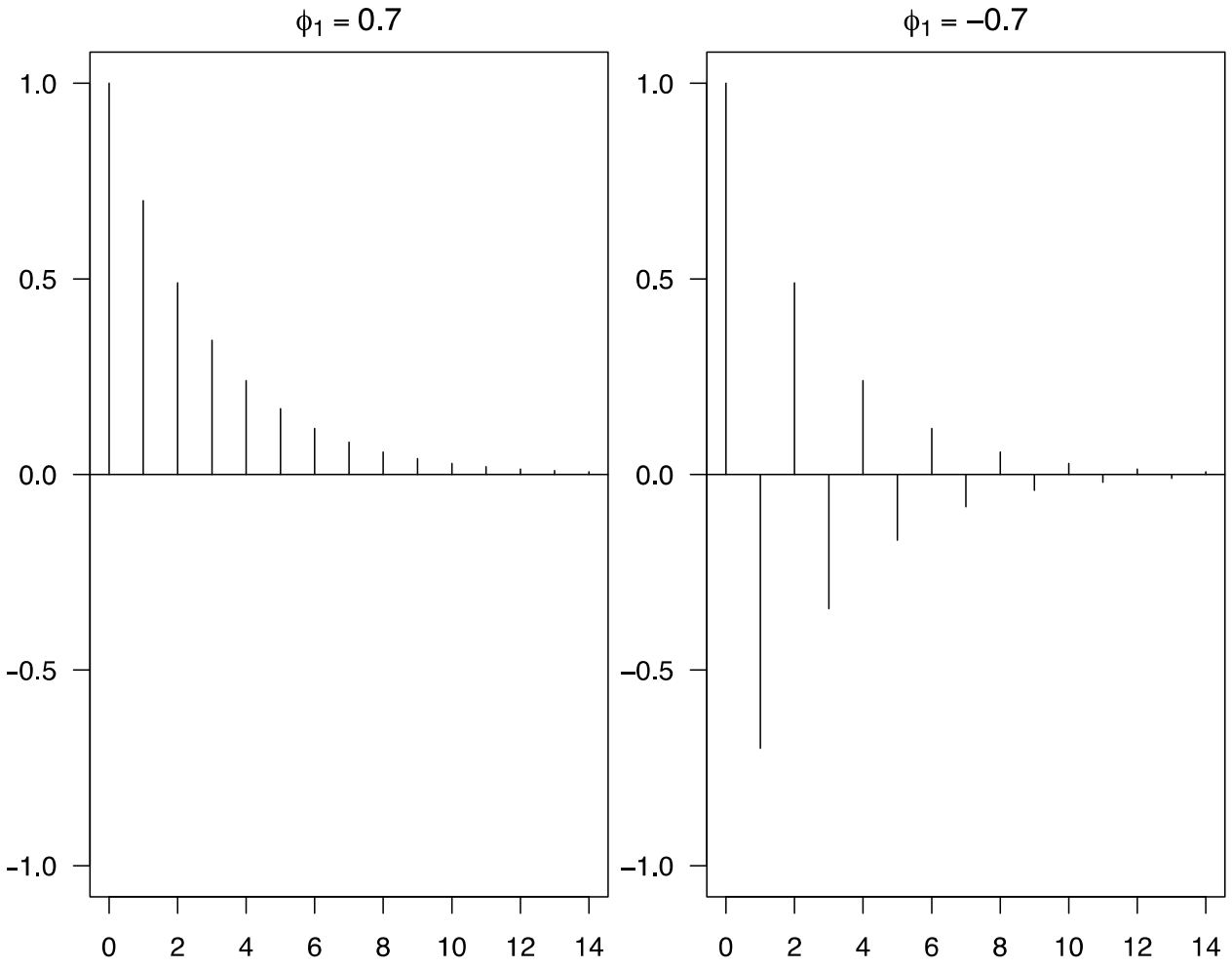


Figure 3.1: Autocorrelation functions for two of AR(1) models.

Moving Average, MA(q), models

Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model. A process $\{X_t\}$ is said to be a moving average process of order q (abbreviated MA(q)) if:

$$X_t = c + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q}$$

Where ϵ_t represents a white noise process (i.e. a series of independently and identically distributed values with mean 0 and variance σ^2)

In a MA(q) process, the current value of the series is a weighted average of past "white noise" terms. Notice that each value of X_t can be thought of as a weighted moving average of the past few 'forecast' errors.

- **NB NOTE: The moving average MA(q) process defined above is NOT the same thing as the moving average smoothing technique!**
  - The moving average smoothing technique, MA(k), is applied to a time series to smooth out random variation (or a seasonal component and random variation in the context of a Classical decomposition).
  - Here, the moving average process, MA(q) is seeking to model the dependency structure in a time series

Moving Average, MA(q), models

A moving average model is usually used to model a time series that shows short-term dependencies between successive observations. They are important because:

1. They are very useful in evaluating the effect of random events (e.g. natural disasters) on a wide variety of economic processes.
2. Any q-correlated process (i.e. any process correlated with its past q values) can be represented as a moving average process.
3. They are widely used in conjunction with autoregressive processes to model a vast range of time series processes.

So, what's the big difference between MA(q) and AR(p) processes?

- An AR(p) model includes lagged terms of the time series itself as explanatory variables, while a MA(q) model includes lagged terms on the forecast errors/random component
- In an autoregressive process, the actual values of the time series are used to form the next realization. In a moving average process, only the random component/forecast errors are used.
- Autoregressive processes depend upon the entire set of observations before time t. The moving average process depends directly on the forecast errors, and so the observations more than q periods in the past provide no information to the MA(q) process and can be ignored.

Moving Average, MA(q), models

Properties of MA(q) processes

1. *Mean:*

$$\boldsymbol{E[X_t] = E(c) + E[\epsilon_t] + \theta_1 E[\epsilon_{t-1}] + \theta_2 E[\epsilon_{t-2}] + \cdots + \theta_q E[\epsilon_{t-q}] = c}$$

as all the random errors have mean 0.

2. V*ariance*: $\boldsymbol{V[X_t] = \sigma^2(1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2)}$

3. *Stationarity:* MA(q) models are always stationary because they are finite linear combinations of a white noise sequence for which the mean and variance are constant (i.e. they do not depend on time)

4. A MA(q) series is only linearly related to its first q lagged values and hence is a "finite-memory" model. $X_t = c + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q}$

5. *The ACF of a MA(q) process has the distinctive feature of vanishing at lags greater than q*: This is because the observed value in time period t, X_t, is related to only the previous q random errors. Since the autocorrelation is a measure of linear association (dependence) it makes sense that it would be equal to 0 at lags greater than q.

Moving Average, MA(q), models

Properties of the MA(1) process

The simplest example of an MA process is the first-order case, denoted MA(1), given by:

$$X_t = c + \epsilon_t + \theta_1 \epsilon_{t-1}$$

1. The mean of a MA(1) process: $\boldsymbol{E[X_t]} = E[c + \epsilon_t + \theta_1 \epsilon_{t-1}] = c + 0 + \theta_1 \times 0 = c$

2. The variance of a MA(1) process: $\mathbf{V}[\boldsymbol{X_t}] = \boldsymbol{\sigma^2(1 + \theta_1^2)}$

The MA(1) process has an autocorrelation of 0 more than 1 lag away, as the process has a memory of only 1 period and "forgets" what happened in the further past. In theory, this implies that future forecasts only need very recent information to make predictions.

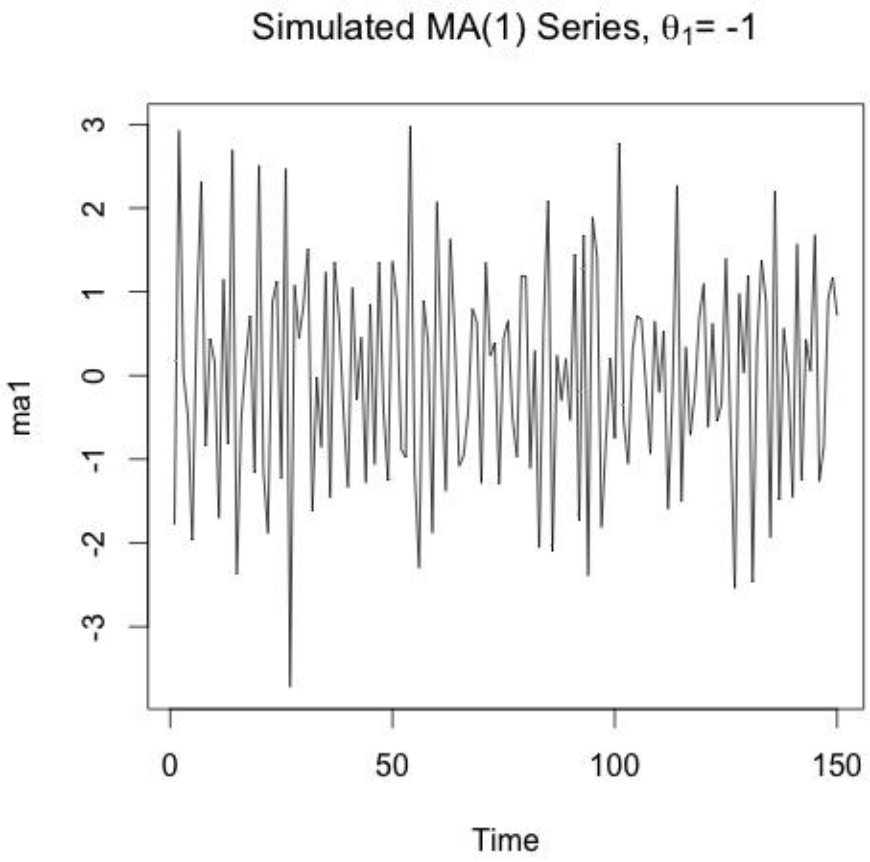
3. *The autocorrelation function* is given by:

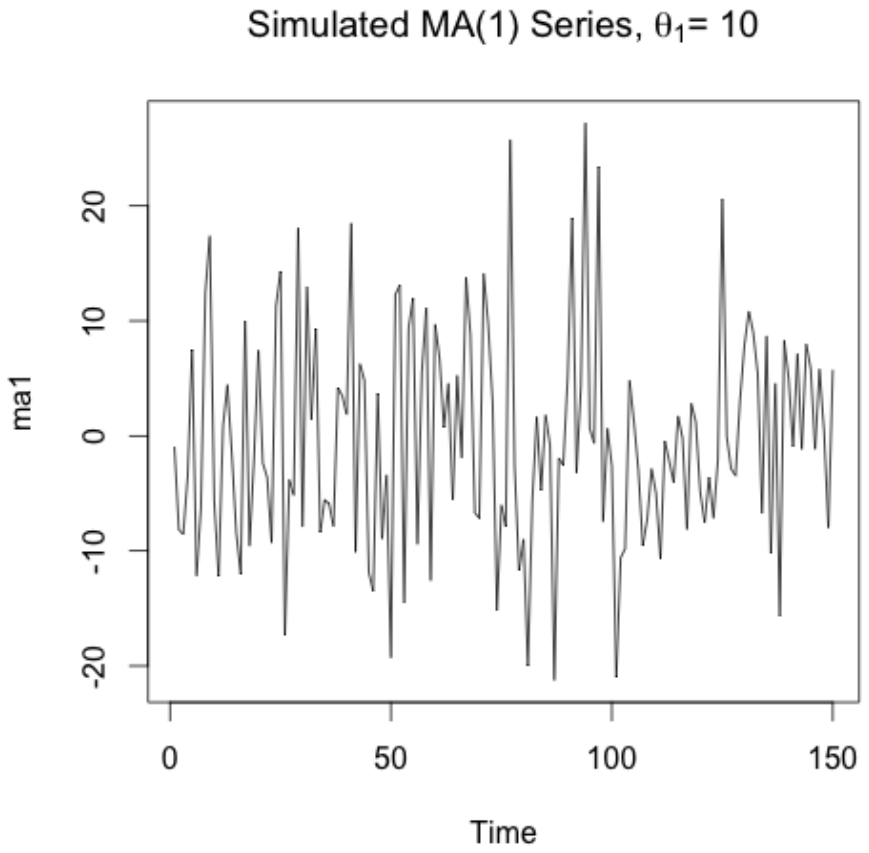
$$\rho(1) = \frac{\theta_1}{1 + \theta_1^2}$$

The autocorrelation plot thus shows a single non-zero autocorrelation at lag 1, with 0 autocorrelations at all higher lags.

Model choice: Using the ACF/PACF

Simulated MA(1) series with $\theta_1 = -1 \ \& \ 10, n = 150$:



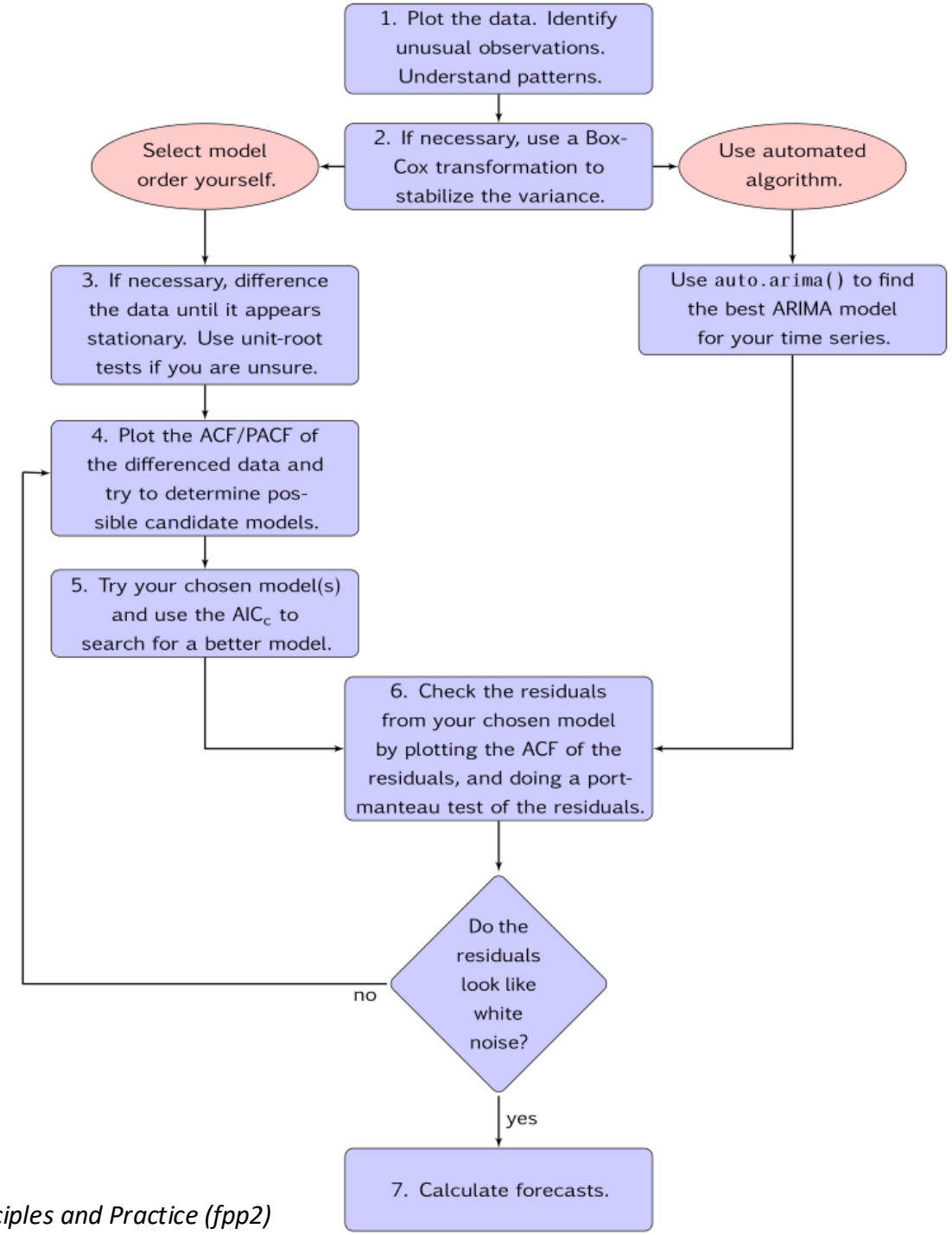


Note that only the scale of the values change as the magnitude of the coefficient θ_1 changes. Both are plausibly stationary time series.

Entire ARIMA modelling process overview:

When fitting an ARIMA model to a set of time series data, the following procedure provides a useful general approach:

1) Plot the data and 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 ARIMA(p,d,0) or ARIMA(0,d,q) model appropriate?
5) Try your chosen model(s) and use the AIC to search for a better model.
6) Check the residuals from your chosen model by plotting the ACF of the residuals and doing the diagnostic tests 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.



Taken from Forecasting: Principles and Practice (fpp2)

Model choice: Using the ACF/PACF

Determining what ARIMA(p,d,q) model to fit based on the behaviour of the ACF and PACF

Given a stationary series that has significant autocorrelation at some lag g, how can I determine (a) What ARIMA(p,d,q) model is appropriate to fit to the data? and (b) What order (i.e. the values of p and q) model to fit?

ANSWER? Use the sample ACF and PACF plots. It is usually not possible to tell, simply from a time plot, what values of p and q are appropriate for the data. However, it is often possible to use the behaviour of the ACF and PACF plots. ***ALWAYS LOOK AT THE COMBINED BEHAVIOUR OF THESE PLOTS.***

- If the time series is from an ARIMA(p,d,0) or ARIMA(0,d,q) process, then the ACF and PACF plots can help suggest the value of p or q. *If p and q are both positive, then the plots do not help in finding suitable values of p and q*.

- The data may follow an ARIMA(p,d,0) model (i.e. an AR(p) model) if the ACF and PACF plots of the stationary (differenced) data show the following patterns:
  i. the ACF is exponentially decaying or sinusoidal;
  ii. there is a significant spike at lag p in the PACF, but none beyond lag p.

- The data may follow an ARIMA(0,d,q) model (i.e. an MA(q) model) if the ACF and PACF plots of the stationary (differenced) data show the following patterns:
  i. the PACF is exponentially decaying or sinusoidal;
  ii. there is a significant spike at lag q in the ACF, but none beyond lag q.

- If (a) BOTH the ACF and PACF decay slowly to 0 OR (b) BOTH the ACF and PACF have a sharp cut-off at low lag values then one would fit a few low-order ARIMA(p,d,q) models and compare them with the AIC. *Watch the relevant lecture video recording for examples of each of the above behaviours.*

Model choice: Using the ACF/PACF

Recognizing an ARIMA(p,d,0) [i.e. an AR(p)] model:

- Recall that partial autocorrelation is a measure of how much correlation between two variables at lag g is explained solely by their relationship, i.e. the partial autocorrelation function (abbreviated PACF) measures the excess correlation at lag g that has not already been accounted for by autocorrelations at lower lags.
- Thus, we can utilise the PACF plot to guide our choice of how many autoregressive terms to include in the model (when we observe slow decay in the ACF

For example, the diagnostic patterns of the ACF and PACF of a AR(1) process:
PACF: cuts off abruptly after lag 1
ACF: declines/decays in a geometric/sinusoidal progression from its highest value at lag 1

- If we have significant partial autocorrelation coefficients up to lag 3, and slow decay to 0 in the ACF, then an AR(3) model may be appropriate.
- A PACF plot that displays no significant partial autocorrelations then an AR(p) model may not be appropriate for the data you are dealing with
- ***A further important result is that the values of the partial autocorrelation coefficients are the coefficients for the autoregressive terms you use in the AR(p) model***

Model choice: Using the ACF/PACF

Recognizing an ARIMA(0,d,q) [i.e. MA(q)] model:

- We have already noted that the ACF of the MA(q) process has the distinctive feature of vanishing at lags greater than q. Thus we can utilise the sample ACF plot to guide our choice of how many moving average terms (q) to include in the model.
- This is because the observed value in time period t, X_t, is related to only the previous q random errors. Since the autocorrelation measures of the strength of the linear association (dependence) it makes sense that its value would be equal to 0 at lags greater than q.
- Essentially, the number of autocorrelation coefficients that are significant (outside the dashed lines) on the sample ACF plot indicate how many terms we should include in a MA(q) model (because those terms will be correlated with the observed value).

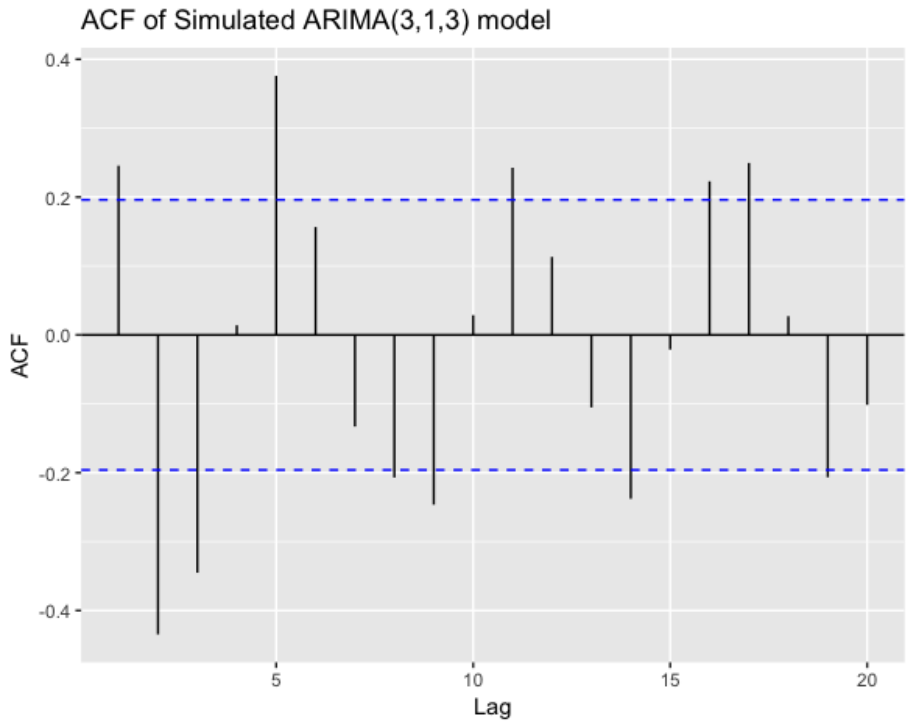
For example, the diagnostic patterns of the ACF and PACF of a MA(1) process:
ACF: cuts off abruptly after lag 1
PACF: declines/decays in a geometric/sinusoidal progression from its highest value at lag 1

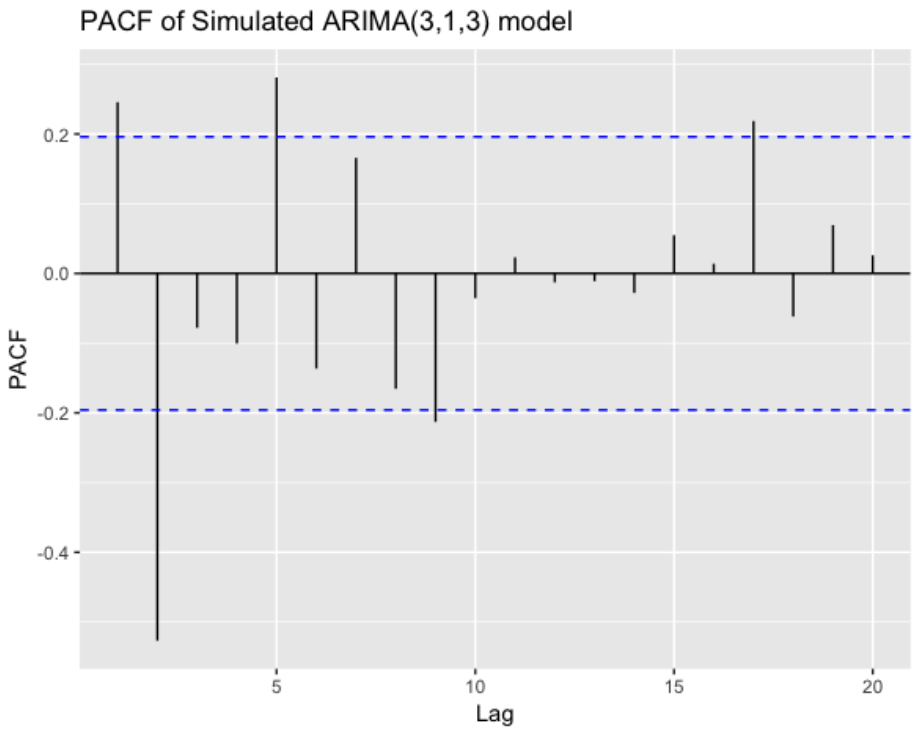
- If we have significant autocorrelation coefficients up to lag 3, then a MA(3) model may be appropriate.
- A sample ACF plot that displays no significant autocorrelations implies that a MA(q) model may not be appropriate for the data you are dealing with.

Model choice: Using the ACF/PACF

Recognizing an ARIMA(p,d,q) [i.e. A mixed AR(p) - MA(q)] model:

We have already noted that when there are both AR and MA terms in a model, then the behaviour of the ACF and PACF will not be helpful in determining the order of a model. At best, from their behaviour, we may be able to suggest that an ARIMA(p,d,q) model with $p > 0$ and $q > 0$ is needed. Consider the ACF and PACF plots below from a simulated ARIMA(3,1,3) model:

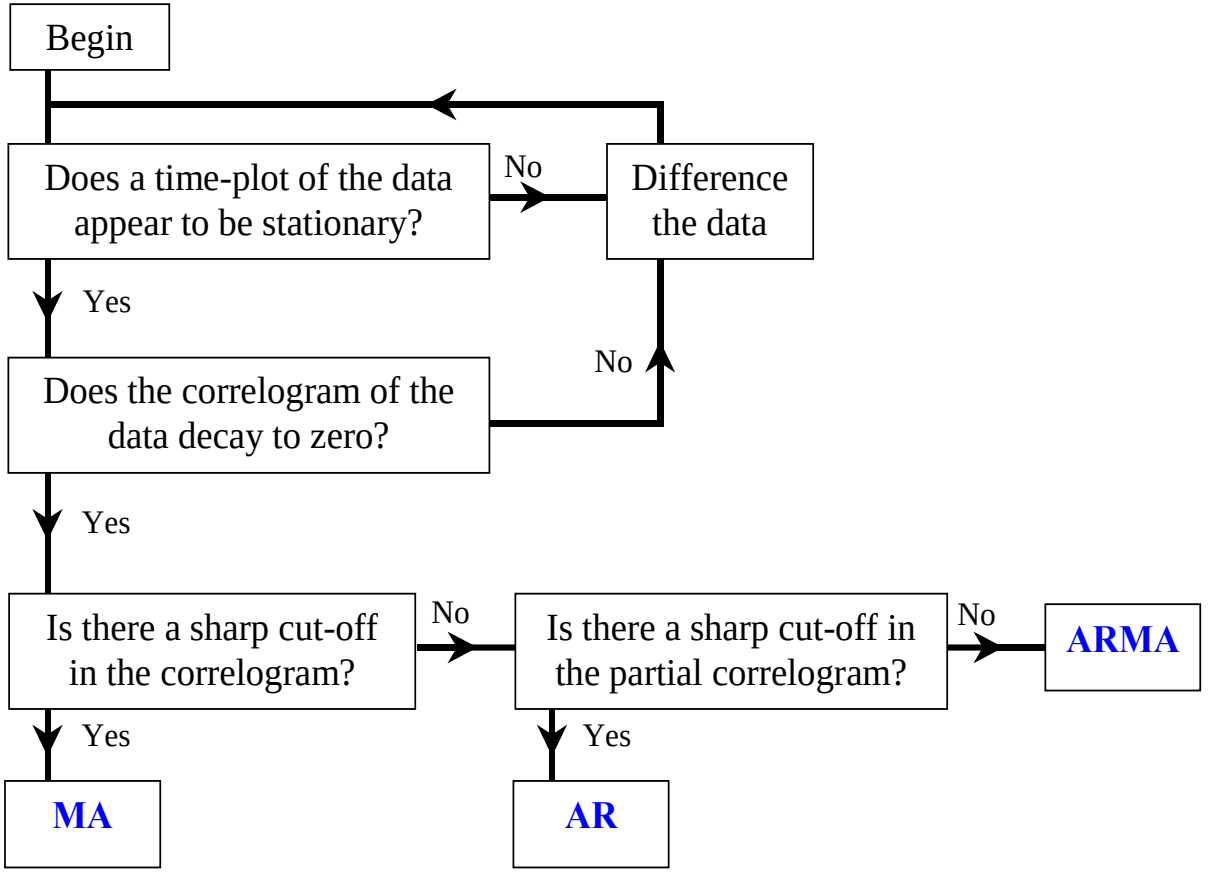




Neither the ACF nor the PACF shut off abruptly after a few lags, they both show slow decay to 0 (the ACF in a sinusoidal fashion, the PACF in a more geometric fashion). In a case like this, the best we can get from the behaviors shown here is that the must fit a mixed AR-MA model i.e. and ARIMA(p,d,q) model. We would start with low order models e.g. ARIMA(1,1,1), ARIMA(2,1,1), ARIMA(1,1,2), ARIMA(2,1,2) etc. and then compare them with AIC.

Model choice: Using the ACF/PACF

The flow diagram included here graphically displays the decision-making process involved when deciding on what type of ARIMA model to fit to a time series.

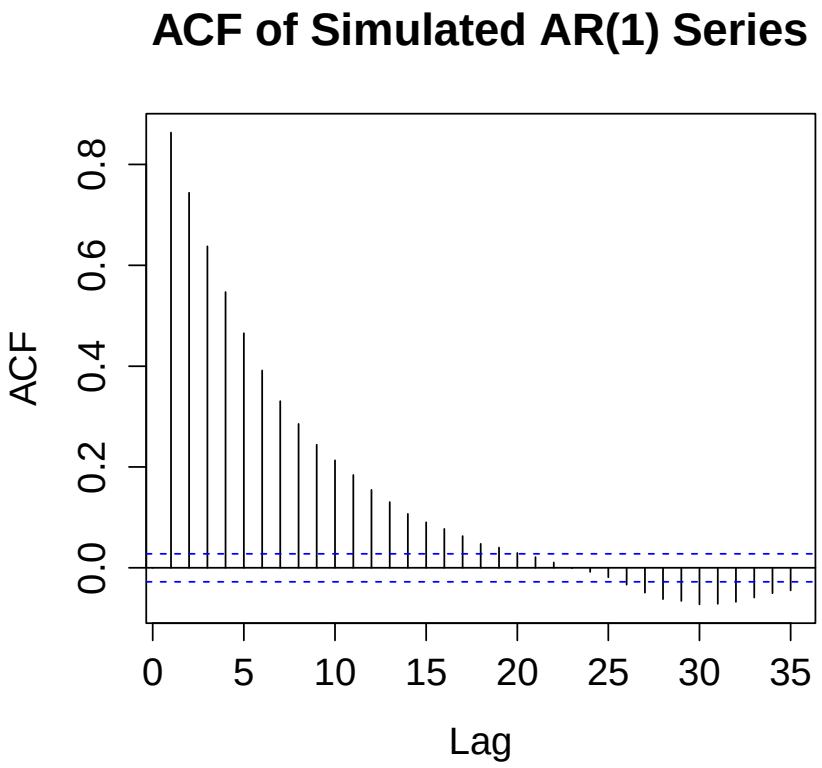


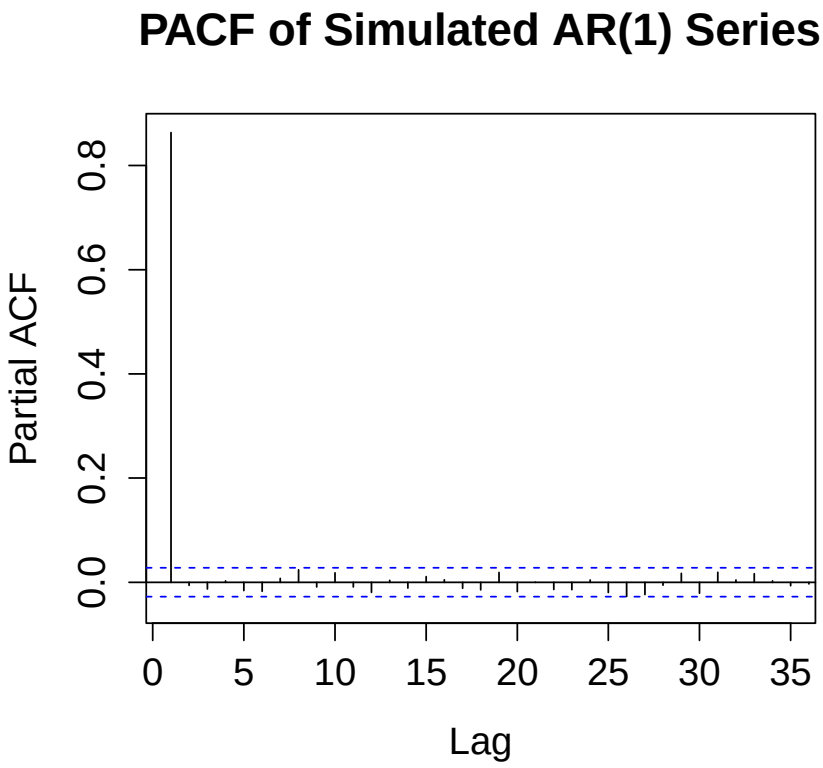
(From Diggle 2000, Fig. 6.2, page 169)

It is very important to understand that determining model order from ACF and PACF plots is ***NOT*** *an exact science*. They serve as a guide. Hence, we seldom only fit one ARIMA(p,d,q) model based on the behaviour of these plots. We always fit the suggested model and a few similar models and then compare them with AIC.

Model choice: Using the ACF/PACF

Consider the following example that displays the ACF and PACF of a simulated AR(1) time series for T = 5000, $\phi_1 = 0.86$:

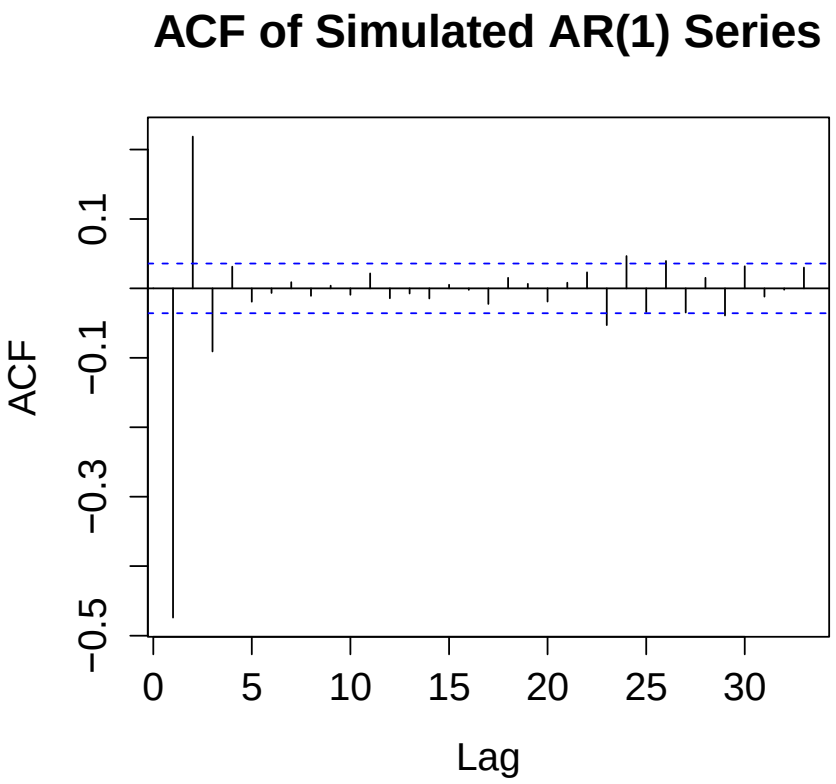


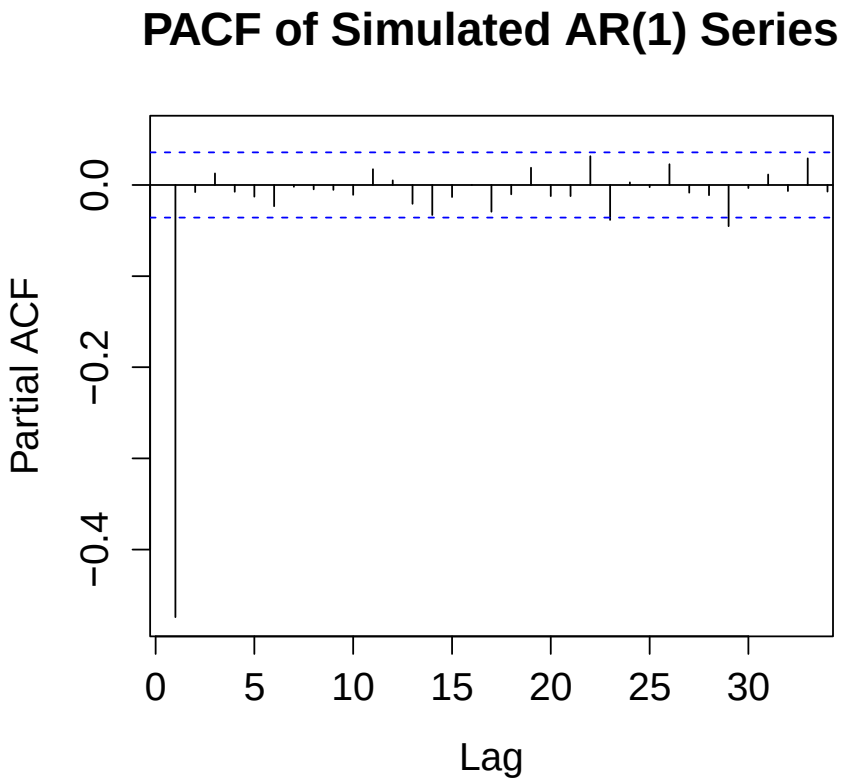


The ACF decays to 0 very slowly and in a sinusoidal fashion, but the PACF has a significant spike only at lag 1, after which it "shuts off" i.e. the partial autocorrelation function values at higher lags all lie between the dashed lines.

Model choice: Using the ACF/PACF

Consider the following example that displays the ACF and PACF of a simulated AR(1) time series for T = 3000, $\phi_1 = -0.46$:

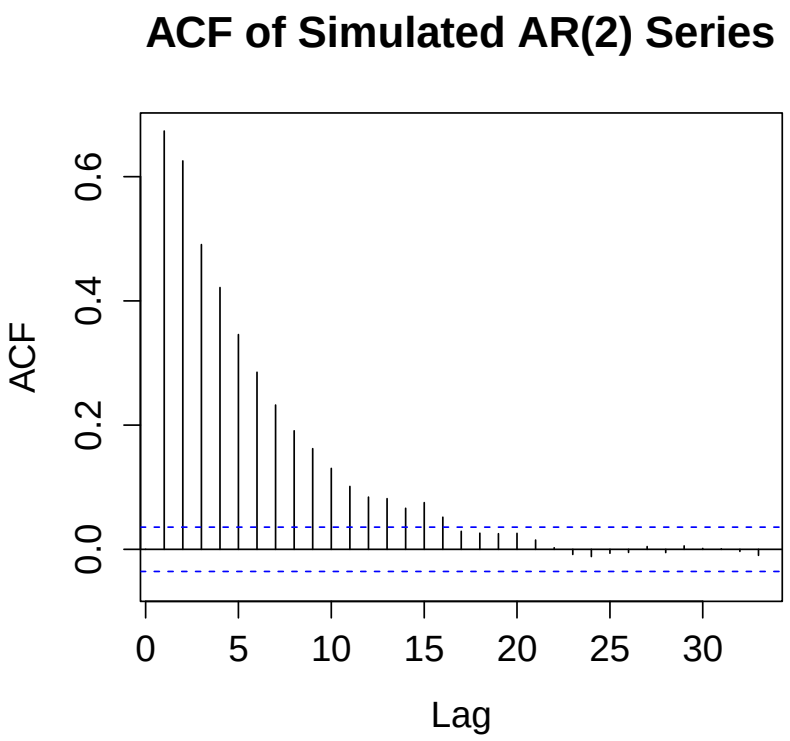


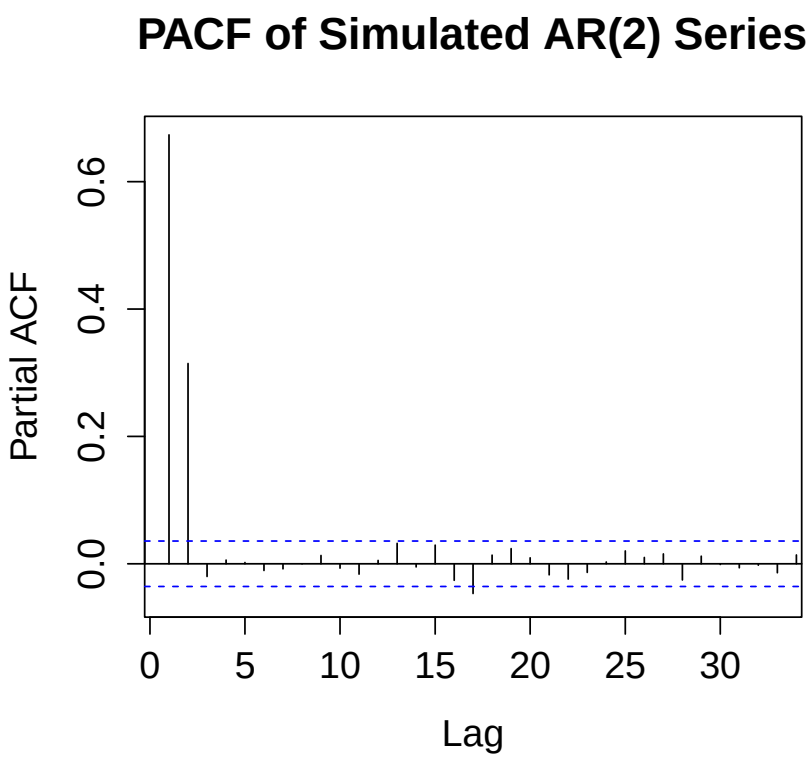


The ACF decays to 0 in a geometric fashion, but the PACF has a significant spike only at lag 1, after which it "shuts off" (the significant partial autocorrelation function value at lag 28 could be due to some form of sampling error).

Model choice: Using the ACF/PACF

Consider the following example that displays the ACF and PACF of a simulated AR(2) time series for T = 3000, $\phi_1 = 0.66$ and $\phi_2 = 0.32$:

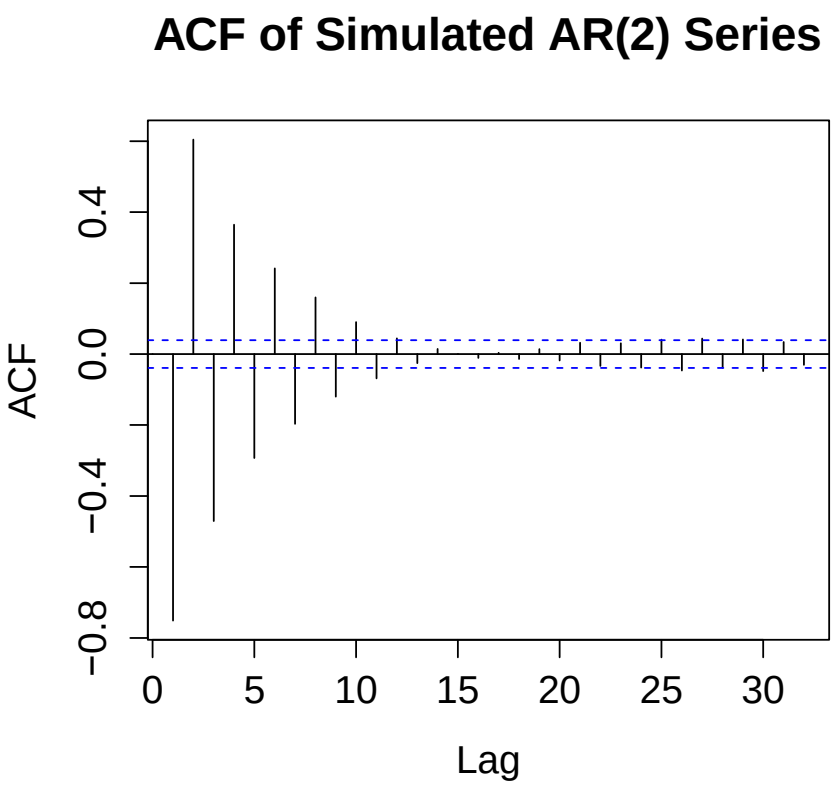


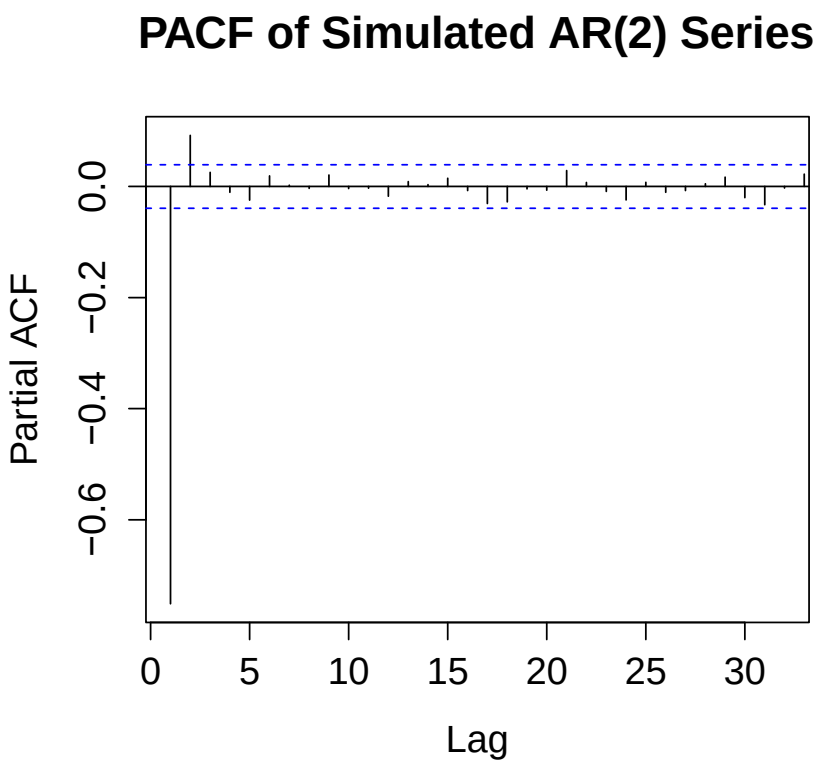


The ACF decays to 0 quite slowly, but the PACF has a significant spike only at lags 1 and 2, after which it "shuts off"

Model choice: Using the ACF/PACF

Consider the following example that displays the ACF and PACF of a simulated AR(2) time series for T = 3000, $\phi_1 = -0.66$ and $\phi_2 = 0.12$:

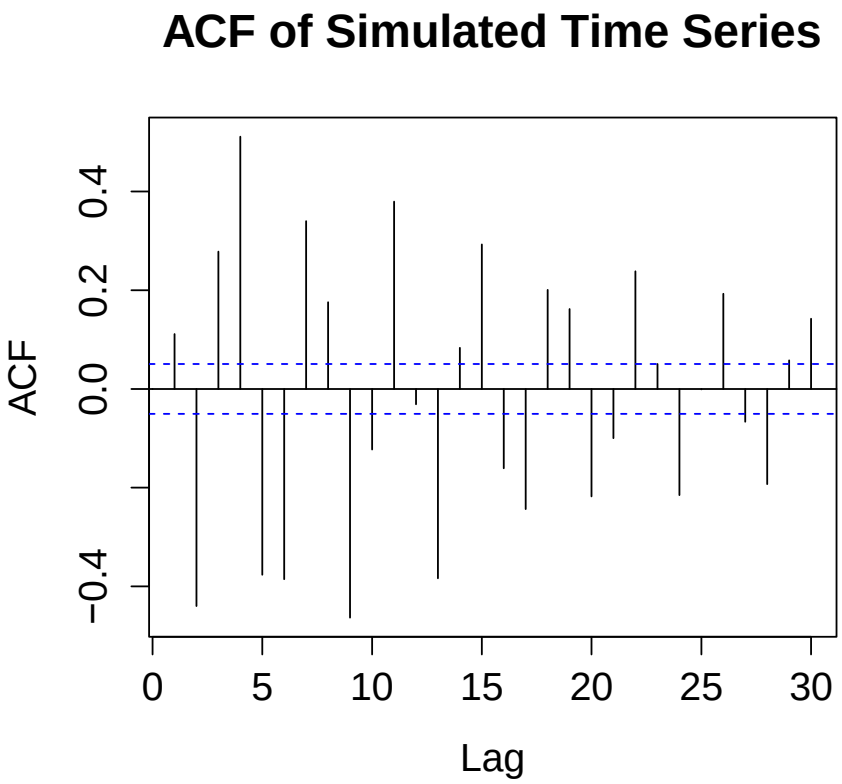


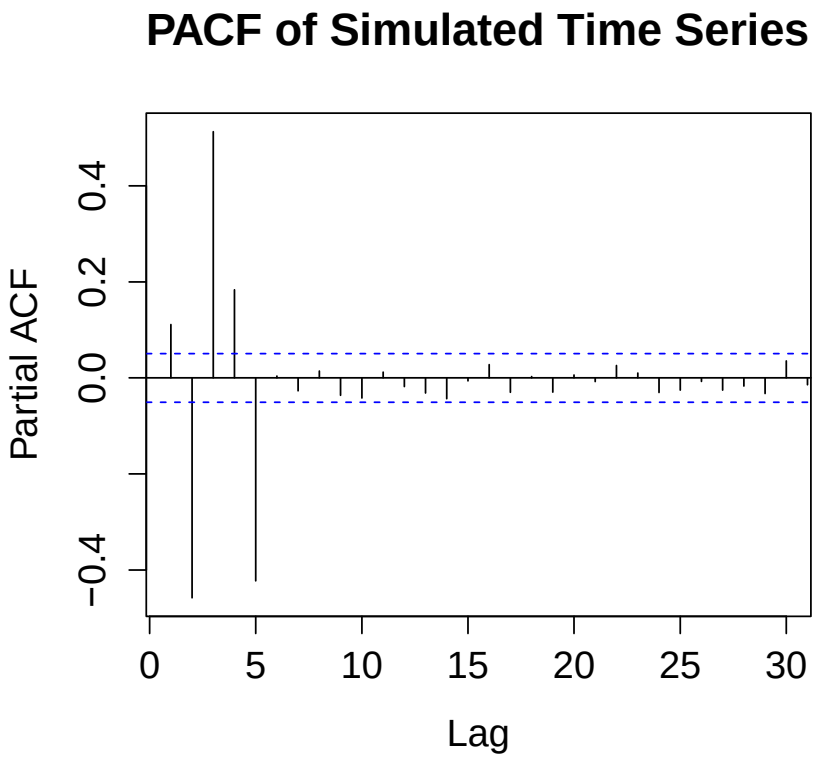


The ACF decays to 0 quite slowly, but the PACF has a significant spike only at lags 1 and 2, after which it "shuts off"

Model choice: Using the ACF/PACF

Consider the following example of the ACF and PACF of a stationary series of data:

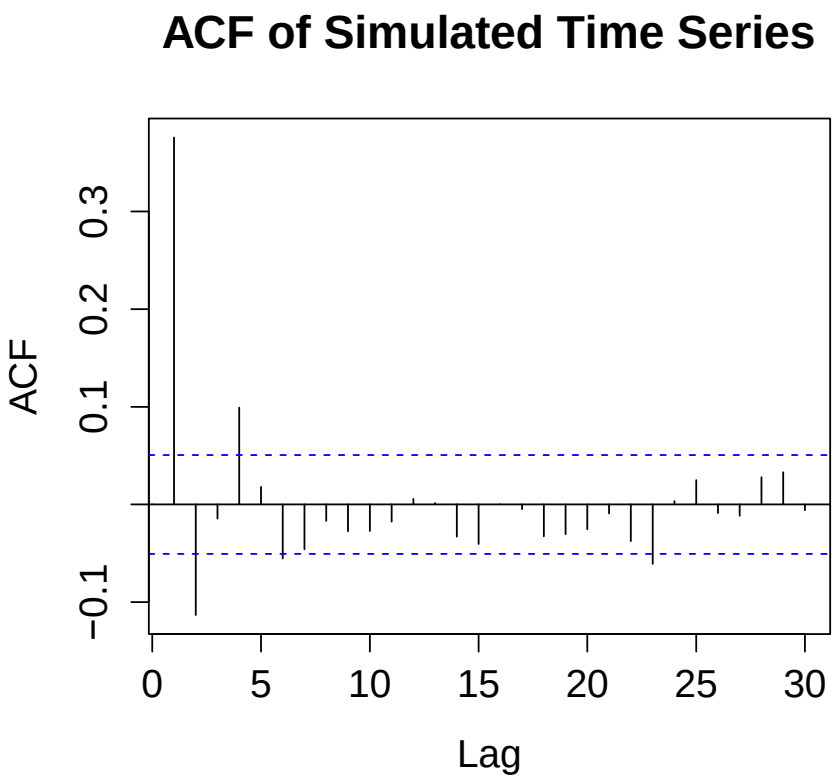


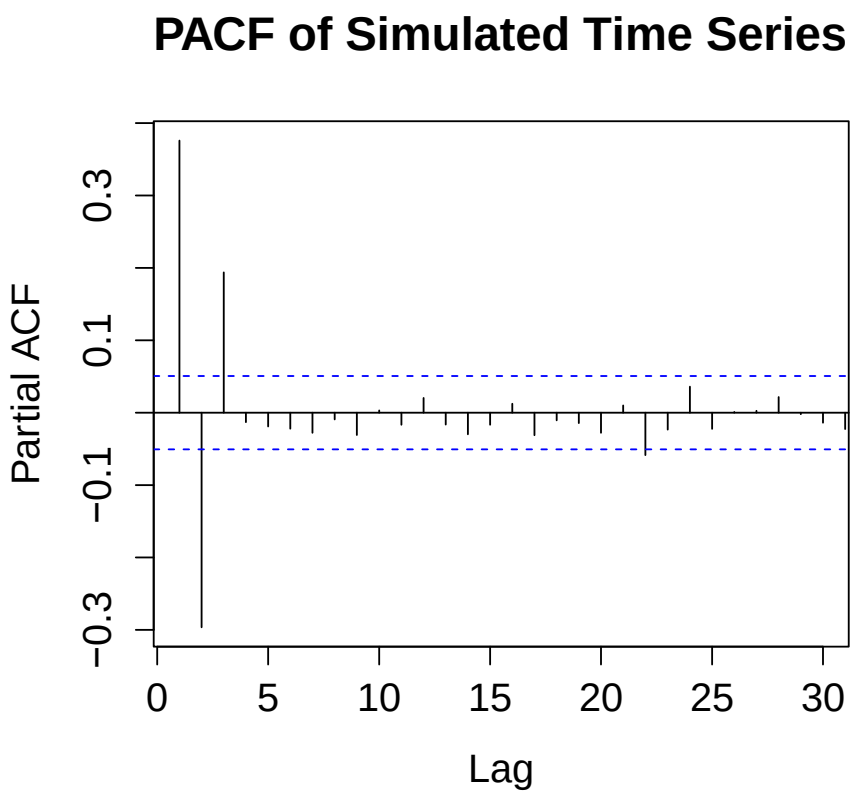


We see that the ACF slowly decays to 0, but the PACF shuts off abruptly after lag 5. Hence, an AR(5) model may be the appropriate order of AR model to fit to the data. So, we would fit an ARIMA(5,d,0) model and then fit some other ARIMA(p,d,q) models e.g. ARIMA(2,1,2), ARIMA(1,1,1) and then compare all models with AIC.

Model choice: Using the ACF/PACF

Consider another example of the ACF and PACF of a stationary series of data:



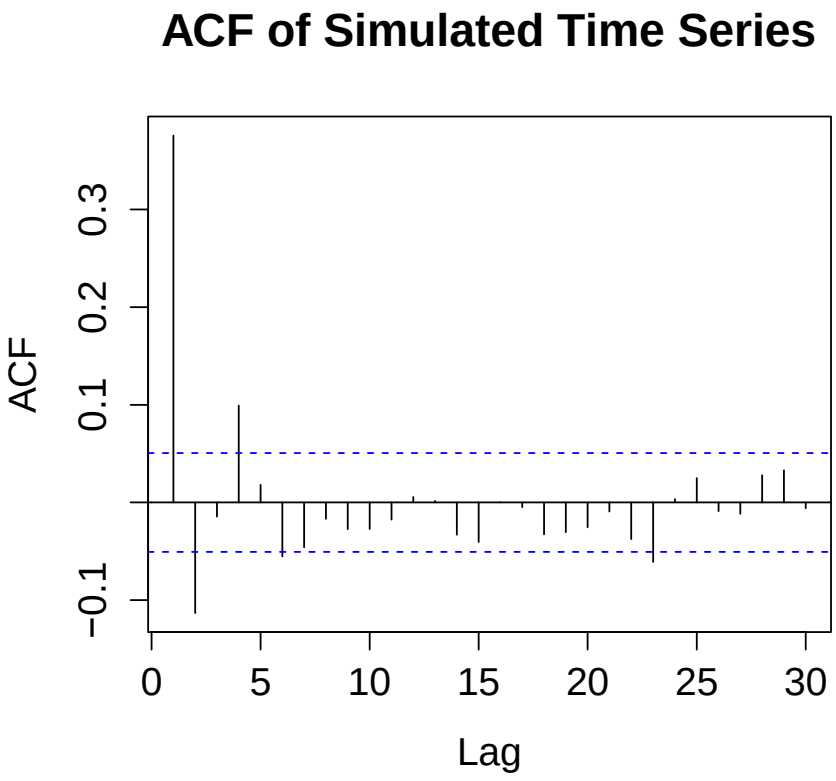


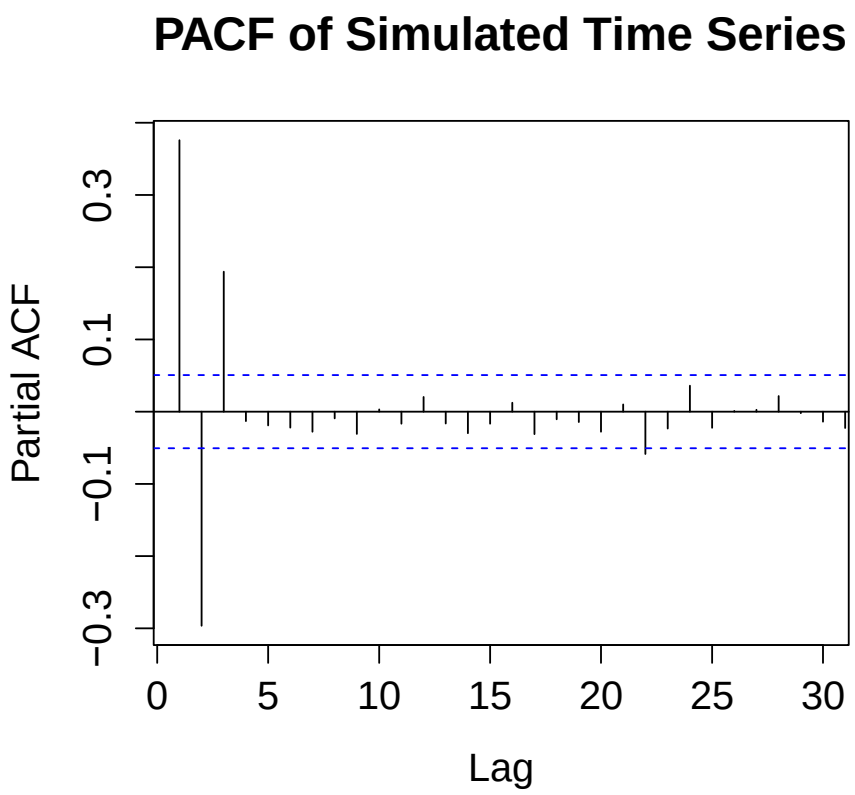
We see that the ACF takes a while to decay to 0 as the lag increases, but the PACF shuts off abruptly after lag 3. *Note the ACF at lags 6 & 23 is outside the 95% confidence bands. Its unlikely that both these are due to chance. There is also a potentially non-zero partial autocorrelation at lag 22, but as this is the only one just outside the bands at a higher lag value, we could say it is due to chance.*

Q: If we chose to fit an AR(p) model to this series, how many terms should be fit in the model?

Model choice: Using the ACF/PACF

Consider another example of the ACF and PACF of a stationary series of data:





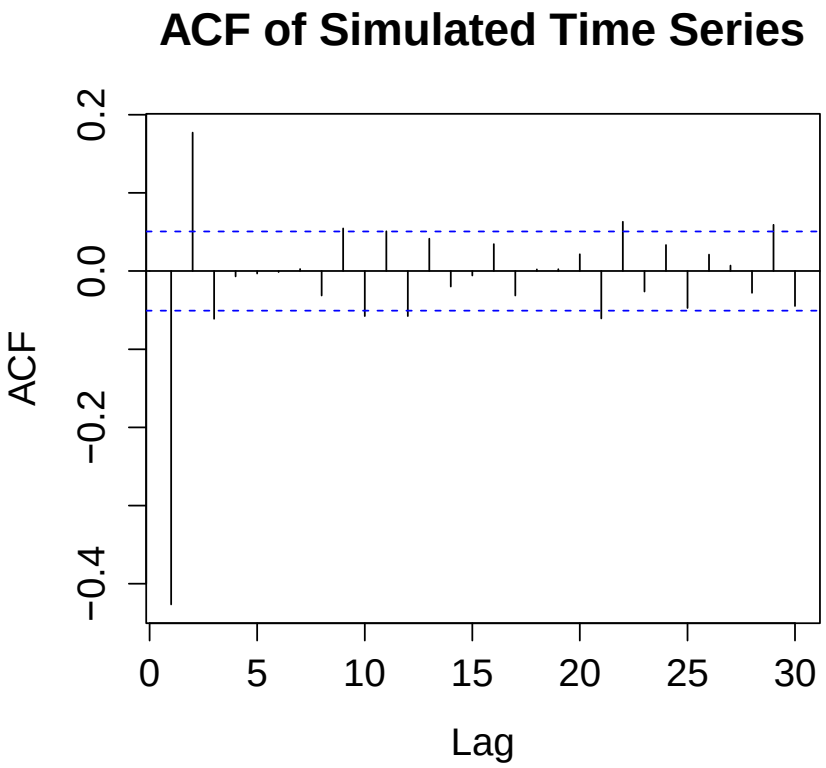
We see that the ACF looks like it decays to 0 as the lag increases, but the PACF shuts off abruptly after lag 3.

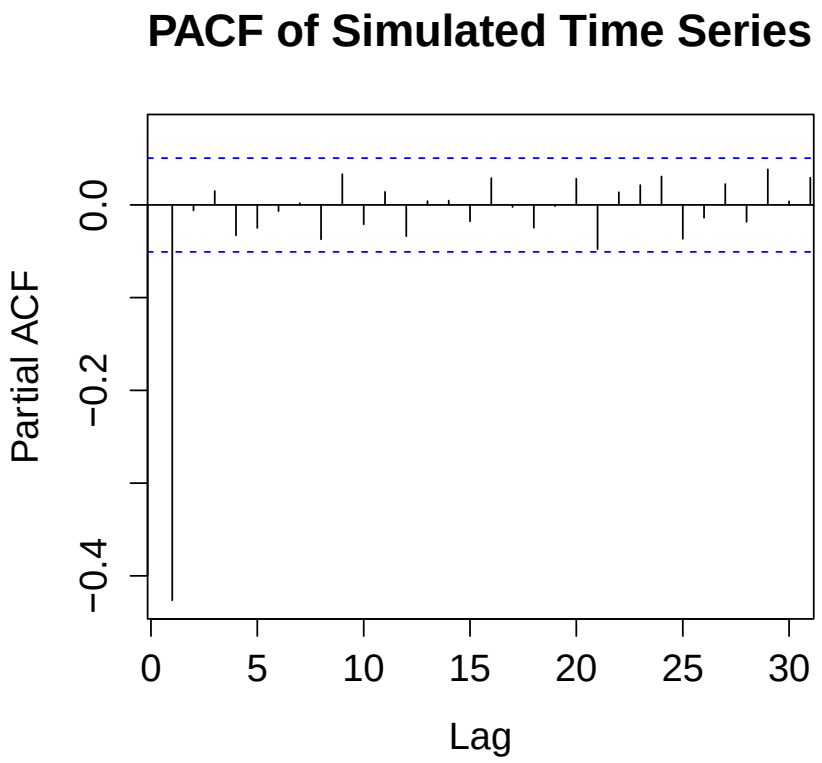
Q: If we chose to fit an AR(p) model to this series, how many terms should be fit in the model? An AR(3) model would be a good starting point.

Again, we would fit an ARIMA(3,d,0) model and then also some other ARIMA(p,d,q) models and then compare them on their AIC values.

Model choice: Using the ACF/PACF

Consider the ACF and PACF plots of a stationary series below:

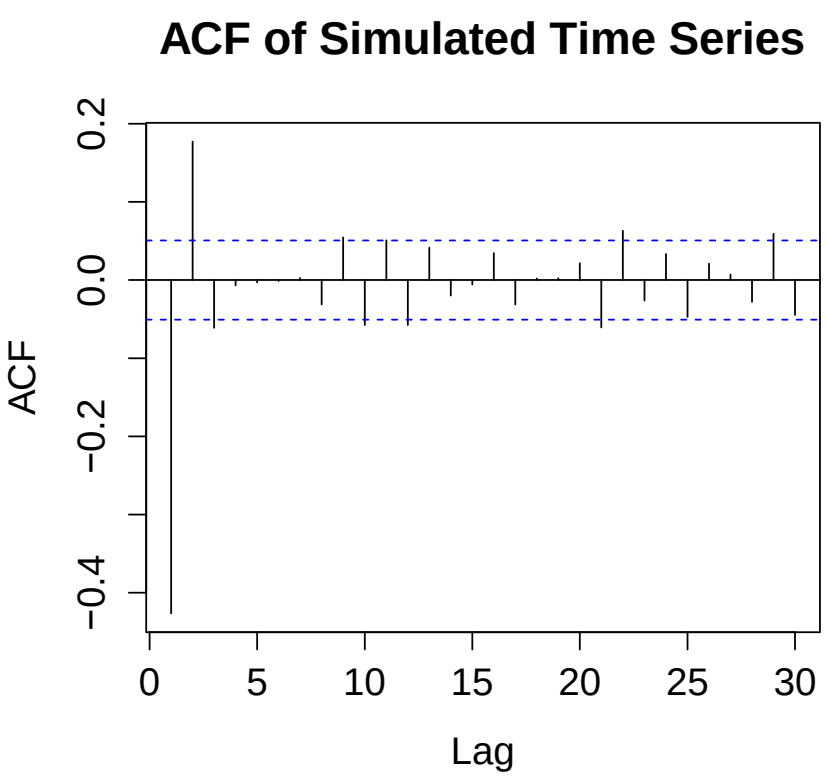


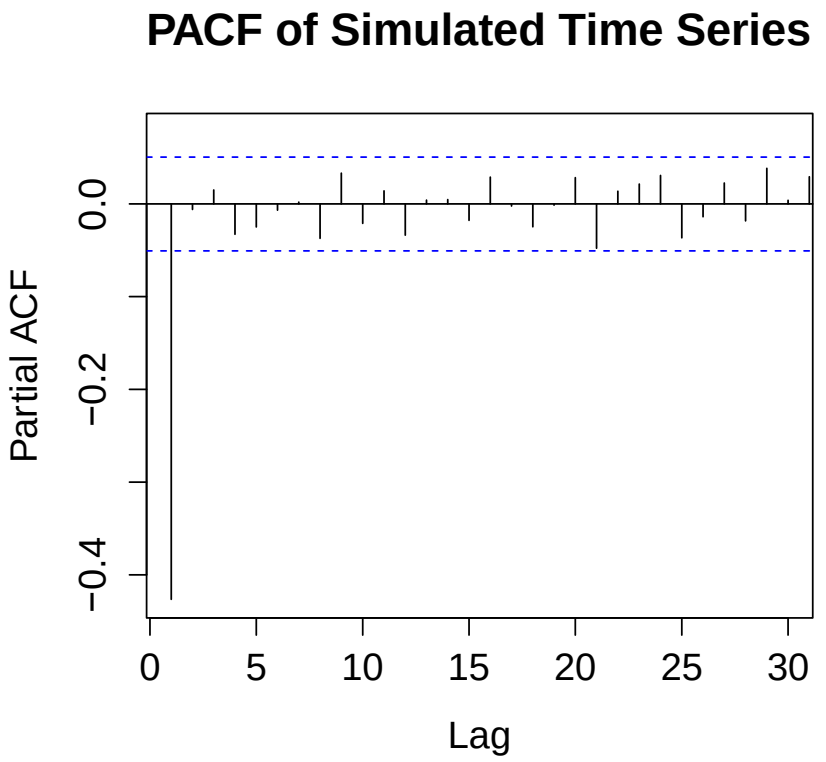


Q: What order of AR(p) model should be fit here?

Model choice: Using the ACF/PACF

Consider the ACF and PACF plots of a stationary series below:

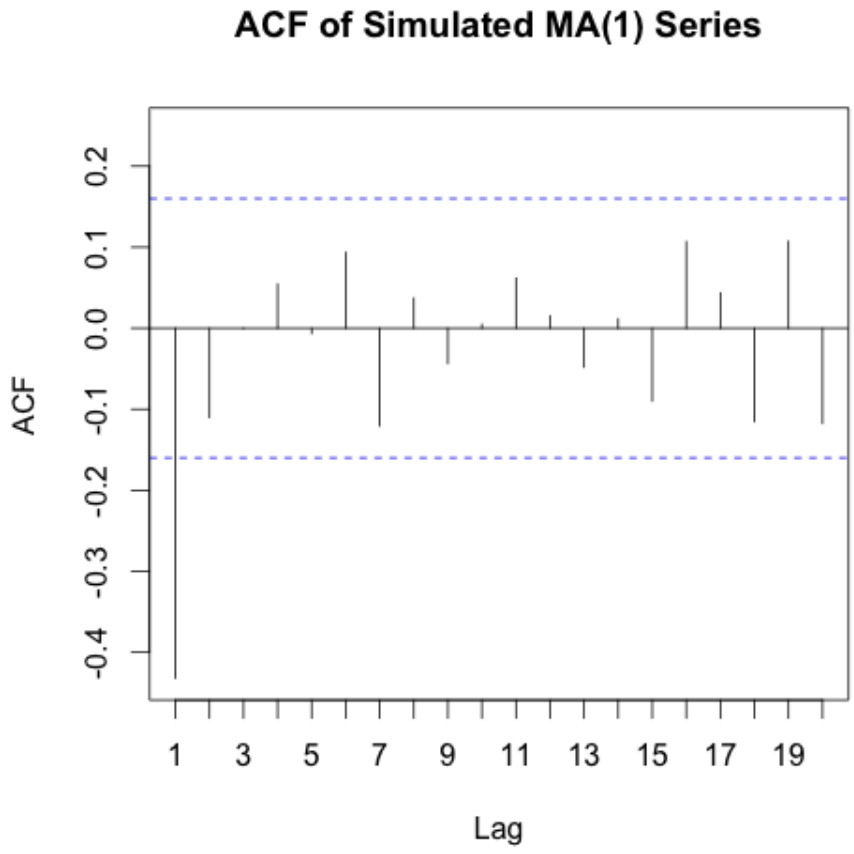


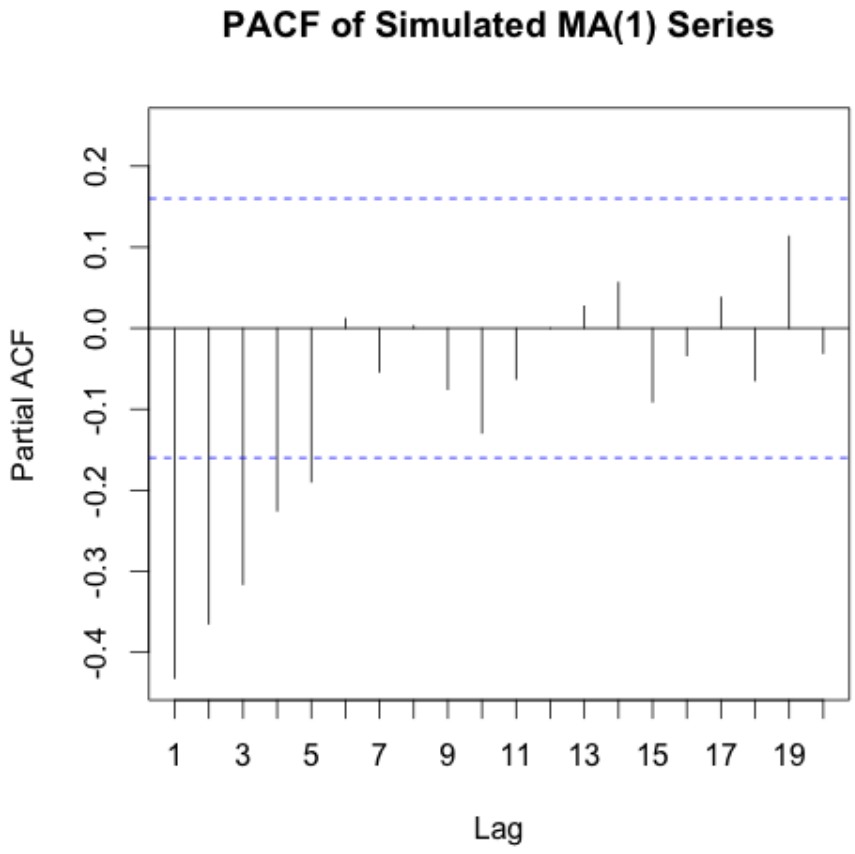


Q: What order of AR(p) model should be fit here? We see that the ACF decays slowly to 0 as the lag increases (there are several non-zero autocorrelations at higher lags in addition to the non-zero stronger autocorrelations at lags 1 and 2). The PACF shuts off abruptly after lag 1. Hence, an AR(**1**) model may be the appropriate order of AR model to fit to the data.

Model choice: Using the ACF/PACF

Consider the ACF and PACF plots of a simulated MA(1) series with $\theta_1 = -1, n = 150$:

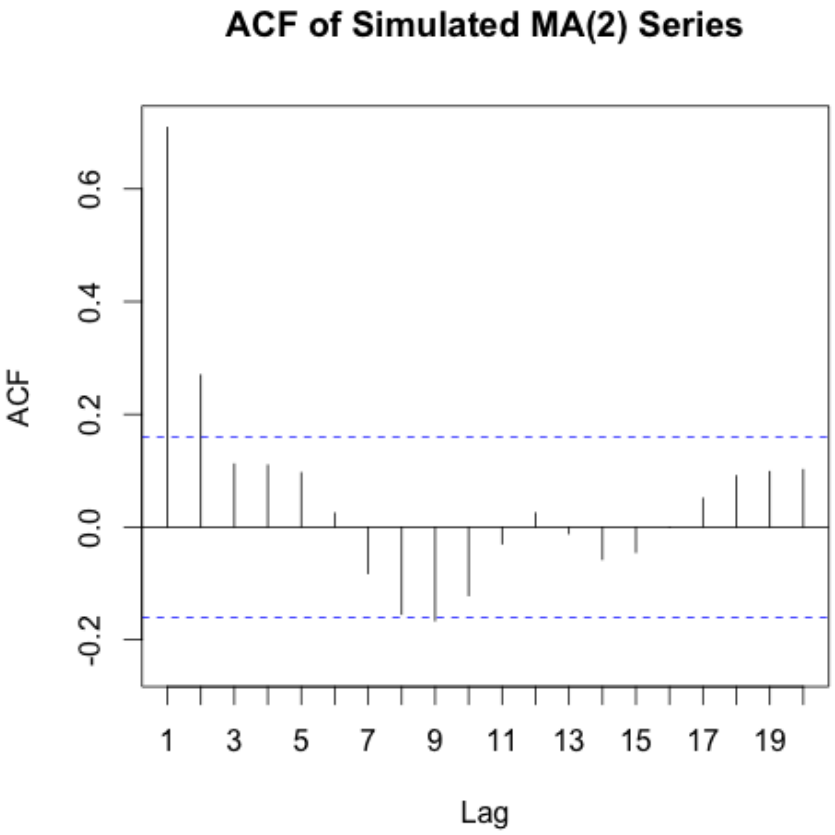


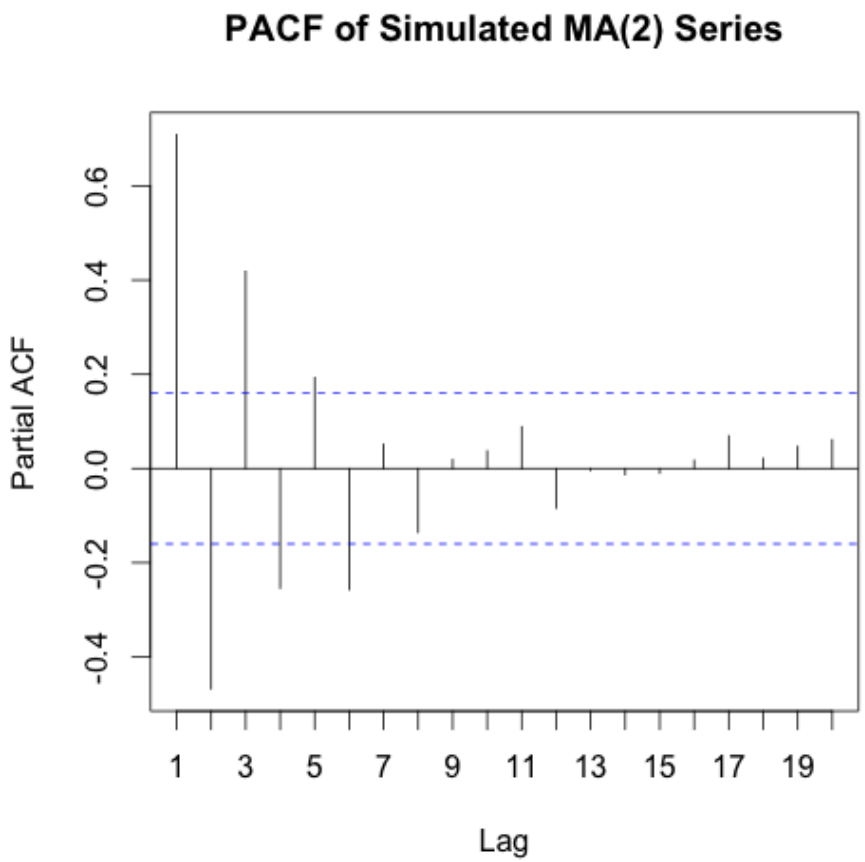


Note that the sample ACF has one significant spike at lag 1, after which it "shuts off". The sample PACF decays slowly to 0 in a geometric fashion. *Hence, a MA(1) model may be appropriate to fit to this series.*

Model choice: Using the ACF/PACF

Consider the ACF and PACF plots of a simulated MA(2) series with $\theta_1 = 2, \theta_2 = 1, n = 150$:

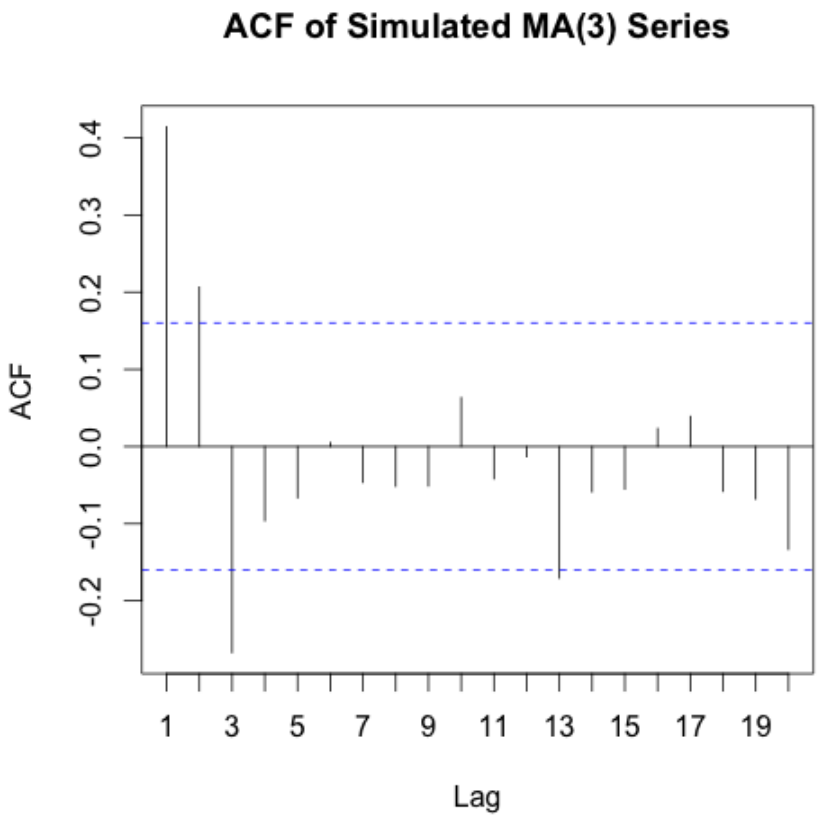


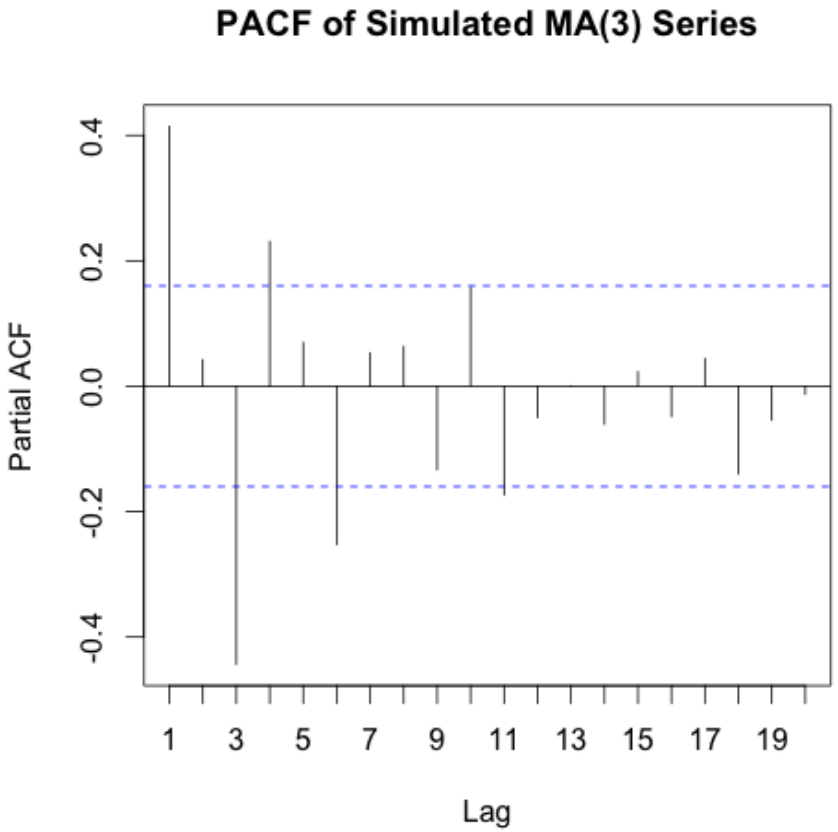


Note that the sample ACF is decaying sinusoidally to 0, with the strongest 2 autocorrelations at lags 1 and 2. Almost all the other autocorrelations are plausibly 0. The sample PACF decays slowly to 0 in a geometric fashion. *Hence, a MA(2) model would be one of the ARIMA(p,d,q) models that we fit.*

Model choice: Using the ACF/PACF

Consider the ACF and PACF plots of a simulated MA(3) series with $\theta_1 = 0.8, \theta_2 = 0.7, \theta_3 = -0.6, n = 150$:

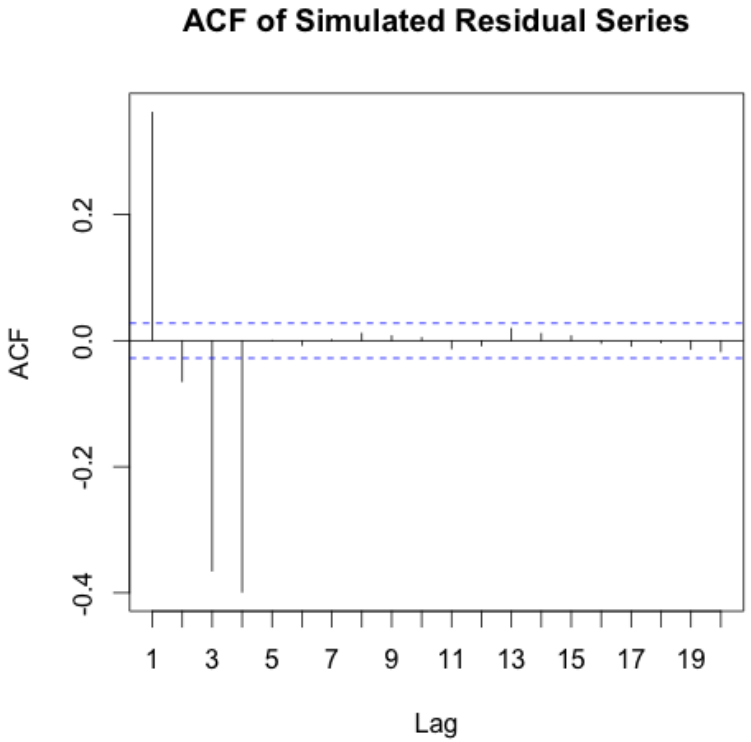


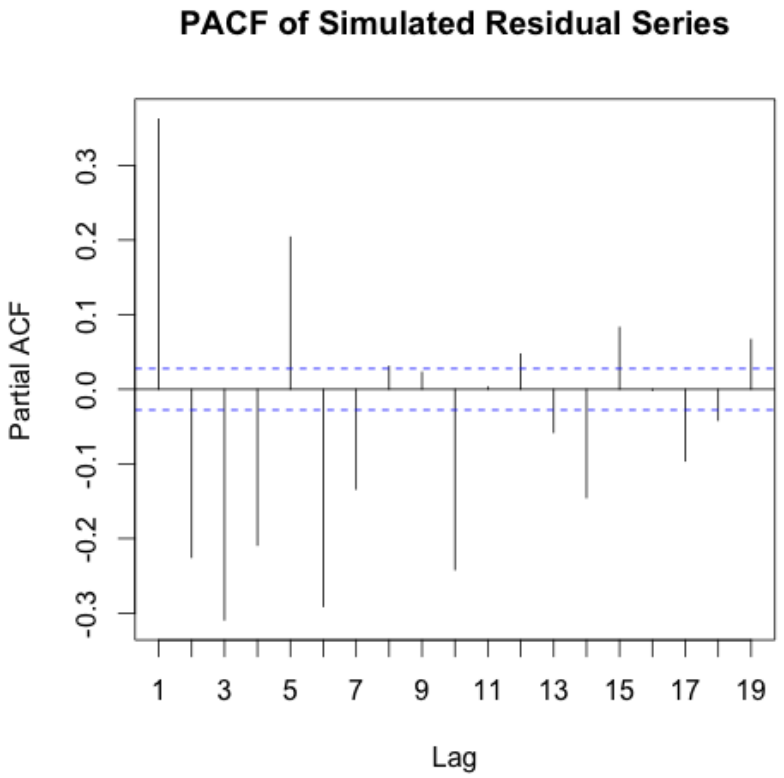


Note that the sample ACF has the three strongest autocorrelations at lags 1, 2 and 3. There is one non-zero autocorrelation at lag 13, which could be due to chance. The sample PACF decays slowly to 0 in a geometric fashion. Considering the combined behaviour of the plots, *an MA(3) model is the suggested model to be fit. So we would fit this model as well as some other ARIMA(p,d,q) models and then compare them on AIC.*

Model choice: Using the ACF/PACF

Consider the ACF and PACF plots of a stationary series below:





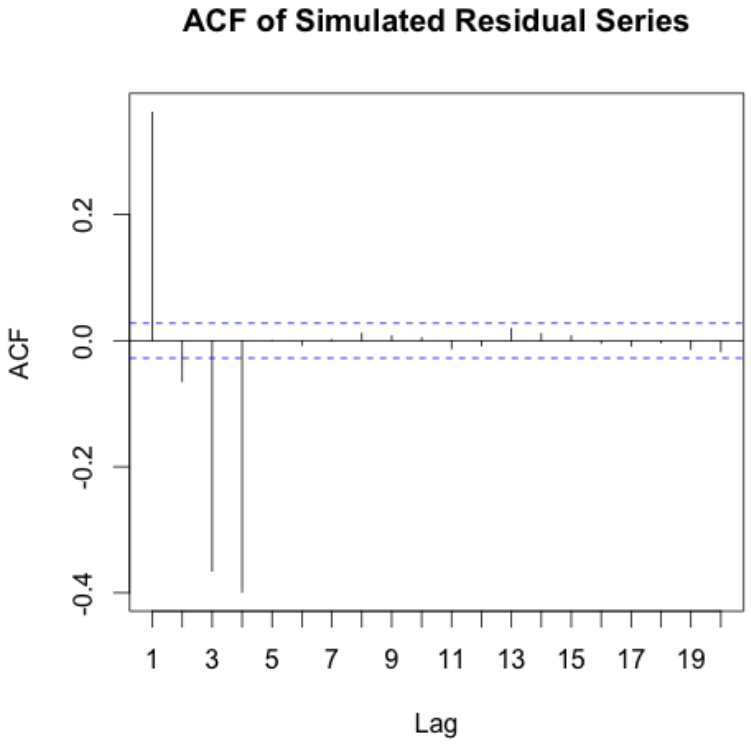
We see that the ACF shuts off abruptly after lag 4, but the PACF slowly decays to 0.

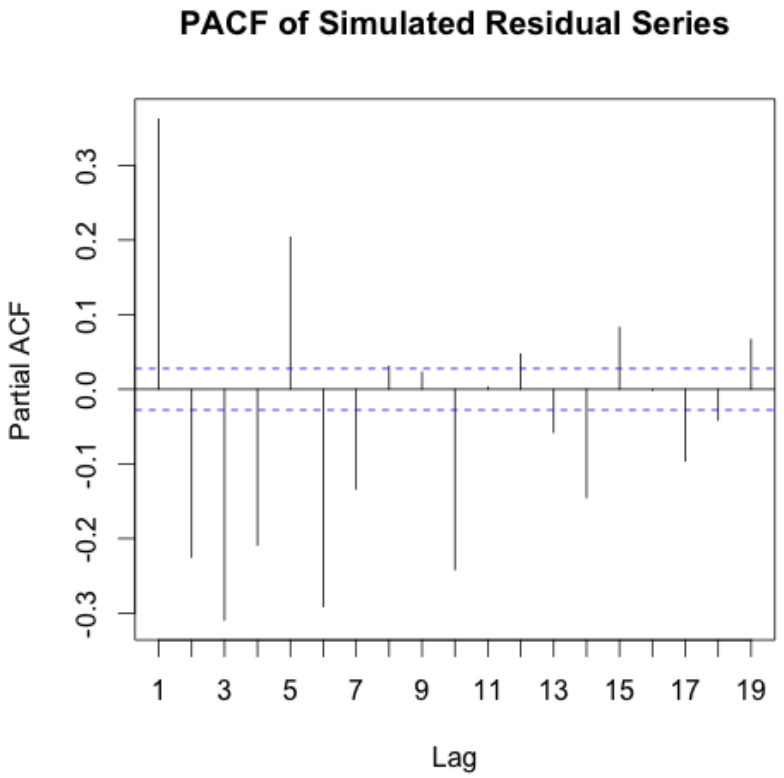
Q: Is it appropriate to fit a MA(q) model here?

Q: If yes, what order of MA(q) model should be fit?

Model choice: Using the ACF/PACF

Consider the ACF and PACF plots of a stationary series below:





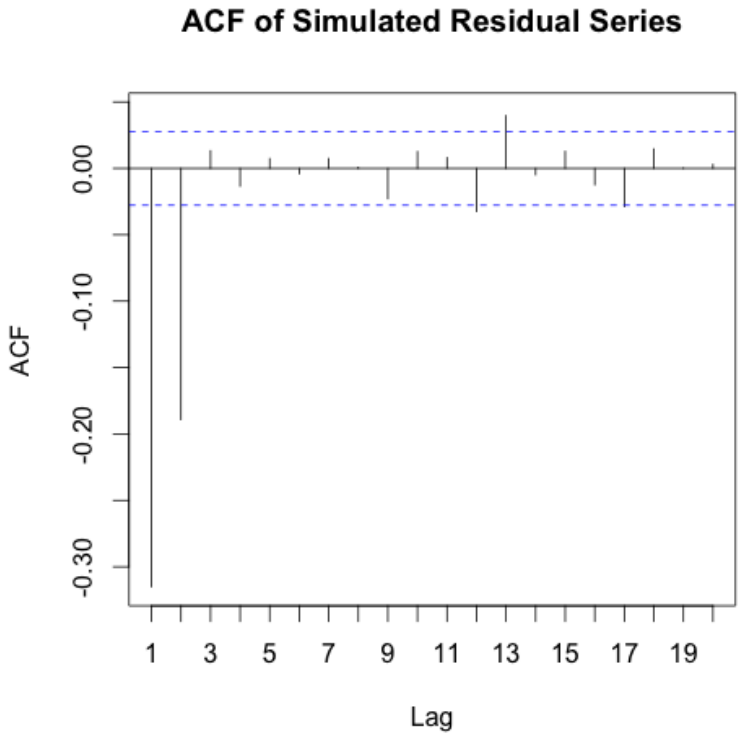
We see that the ACF shuts off abruptly after lag 4, but the PACF slowly decays to 0.

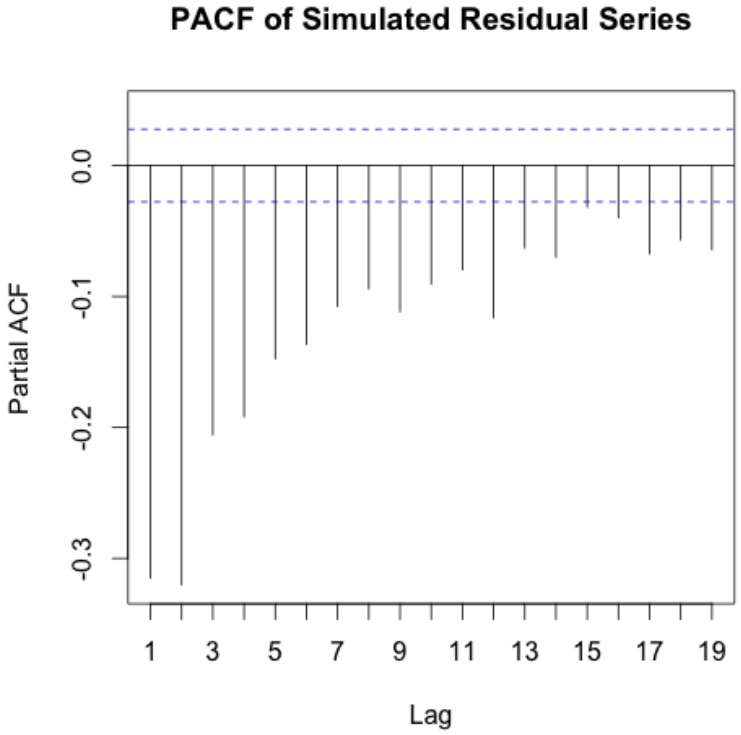
Q: Is it appropriate to fit a MA(q) model here? **YES**

Q: If yes, what order of MA(q) model should be fit? A **MA(4) model would be a good starting point. We would also fit a few other ARIMA(p,d,q) models.**

Model choice: Using the ACF/PACF

Consider the ACF and PACF plots of a stationary series below:



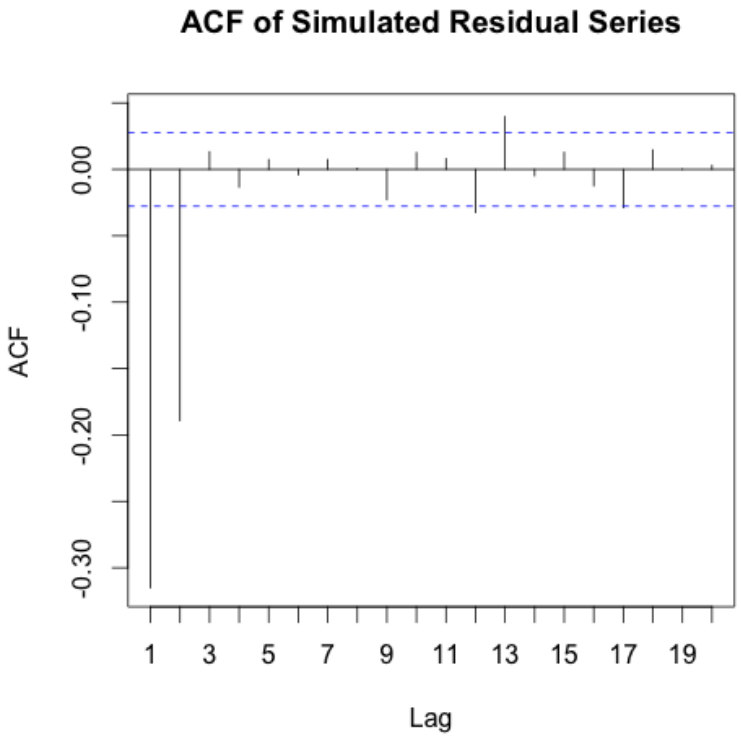


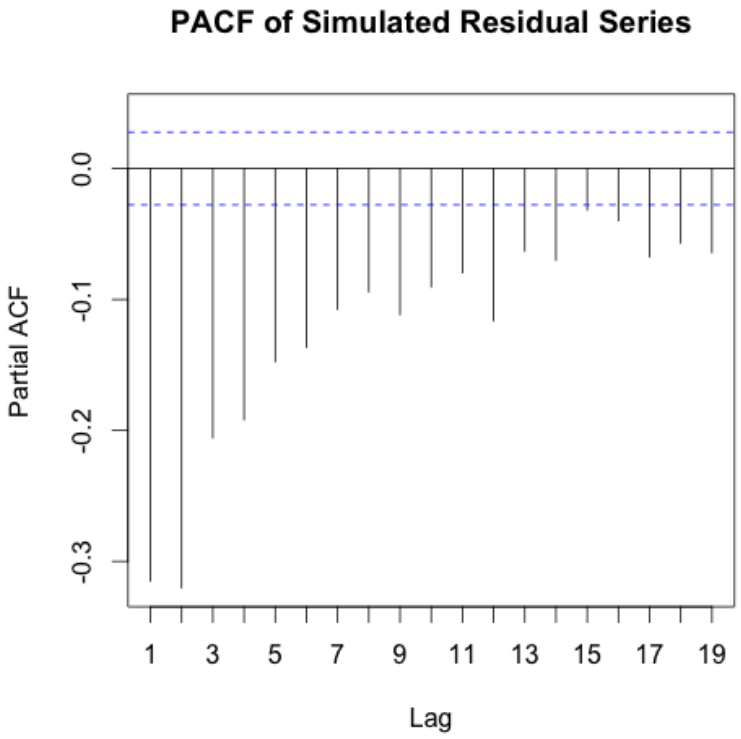
We see that the ACF has its strongest two autocorrelations at lags 1 and 2, with a few other non-zero (but very weak) autocorrelations at higher lag values. The PACF slowly decays to 0.

Q: What order MA(q) model should be fit here?

Model choice: Using the ACF/PACF

Consider the ACF and PACF plots of a stationary series below:





We see that the ACF has its strongest two autocorrelations at lags 1 and 2, with a few other non-zero (but very weak) autocorrelations at higher lag values. The PACF slowly decays to 0.

Q: What order MA(q) model should be fit here? **Considering the combined behaviour of the two plots, a MA(2) model would be appropriate to fit to this time series.**

Model fitting: Maximum Likelihood estimation

Once we have examined the sample ACF and PACF plots and have decided on a few candidate ARIMA(p,d,q) models, we fit them using maximum likelihood estimation (MLE). **We do NOT concern ourselves with estimating the parameters in this course by hand i.e. we let Rstudio do its job!!! However, some basic understanding of MLE is important:**

When R estimates the ARIMA model, it uses *maximum likelihood estimation* (MLE). This technique finds the values of the parameters (i.e. model coefficients) *which maximize the probability of obtaining the data that we have observed i.e. MLE adjusts the estimated coefficient values until the probability that the time series data we have observed comes from the specified ARIMA(p,d,q) model is* ***maximized.***

In practice, R will report the value of the *log likelihood* of the data; that is, the logarithm of the probability of the observed data coming from the estimated model. For given values of p, d and q, R will try to maximize the log likelihood when finding the coefficient estimates.

Given appropriate model output, you should be able to explain what the log-likelihood, $\log(L)$, is. You should also need to be able to state the fitted model and calculate its properties (e.g. mean, variance and autocorrelation at a certain lag. etc.)

Model selection: Using AIC

Once we have examined the sample ACF and PACF plots and have decided and fitted a few ARIMA(p,d,q) models to the time series, how are we to decide which model is the best one to fit to the data?

ANSWER: Use information criteria! Essentially, information criteria trade off the goodness-of-fit of the model with the complexity (i.e. order of the model). Lower order models that fit the data better than other models are preferred. The most commonly used information criteria are the Bayes Information Criterion (BIC), the Akaike Information Criterion (AIC) and the corrected Akaike Information Criterion (AICc). **We will use AIC in this course.**

The Akaike information criterion is a measure of the relative quality of a statistical model for a given set of data. That is, given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Hence, AIC provides a means for model selection:

$$\mathbf{AIC} = -\mathbf{2log(L)} + \mathbf{2(p + q + k + 1)}$$

Where:

L is the likelihood of the data

$k = 1\ if\ c \neq 0\ and\ k = 0\ if c = 0$

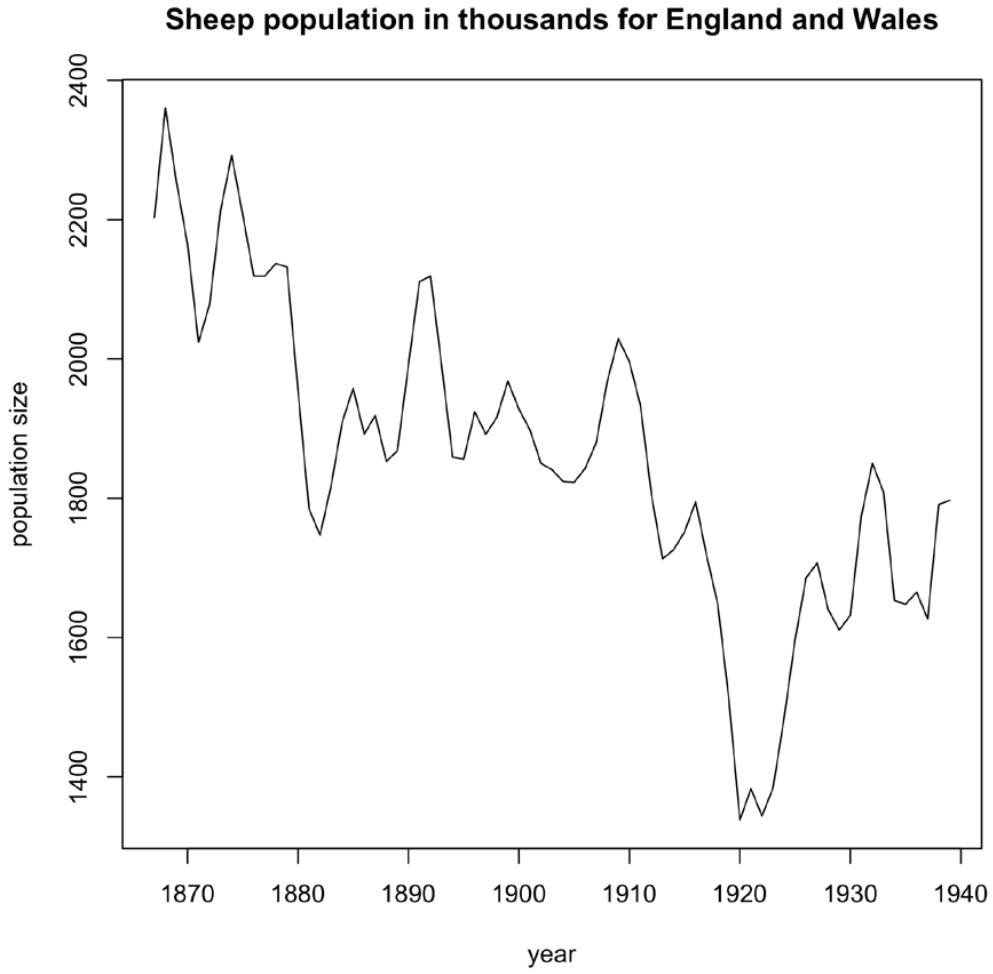
$p, d\ and\ q$ refer to the number of autoregressive terms, order of differencing and number of moving average terms

Model selection: Using AIC

- The first term of the expression is a measure of how well the model fits the data.
- The variance of the forecasts of a model (due to estimation error) increases with the overall number of terms in the model. This implies that you don't want a forecasting model with too many terms.
- But how many is "too many"? The second term represents a "penalty" term for the order (complexity) of the fitted model i.e. thus term is a "penalty" for using more terms, which increases the forecast variance.
- **Minimizing *AIC* trades off the goodness-of-fit with the number of fitted terms to determine a "best" model for your forecast**. This trade-off is a well-known phenomenon in statistics that is referred to as the bias-variance trade-off.
- A smaller value for AIC for a model (when compared to other models) implies a model with fewer parameters, a better fit OR both (i.e. fewer parameters and better fit).
- ***NB: AIC does not provide a test of a model in the sense of testing a null hypothesis; i.e. AIC can tell nothing about the quality of the model in an absolute sense. If all the candidate models fit poorly, AIC will not give any warning of that.***
- ***To use the AIC to select a model in practice***, one computes the AIC for a variety of model orders (e.g. for an ARIMA(2,1,1) model, the order = 3) and then select the model that has the ***minimum AIC value***.

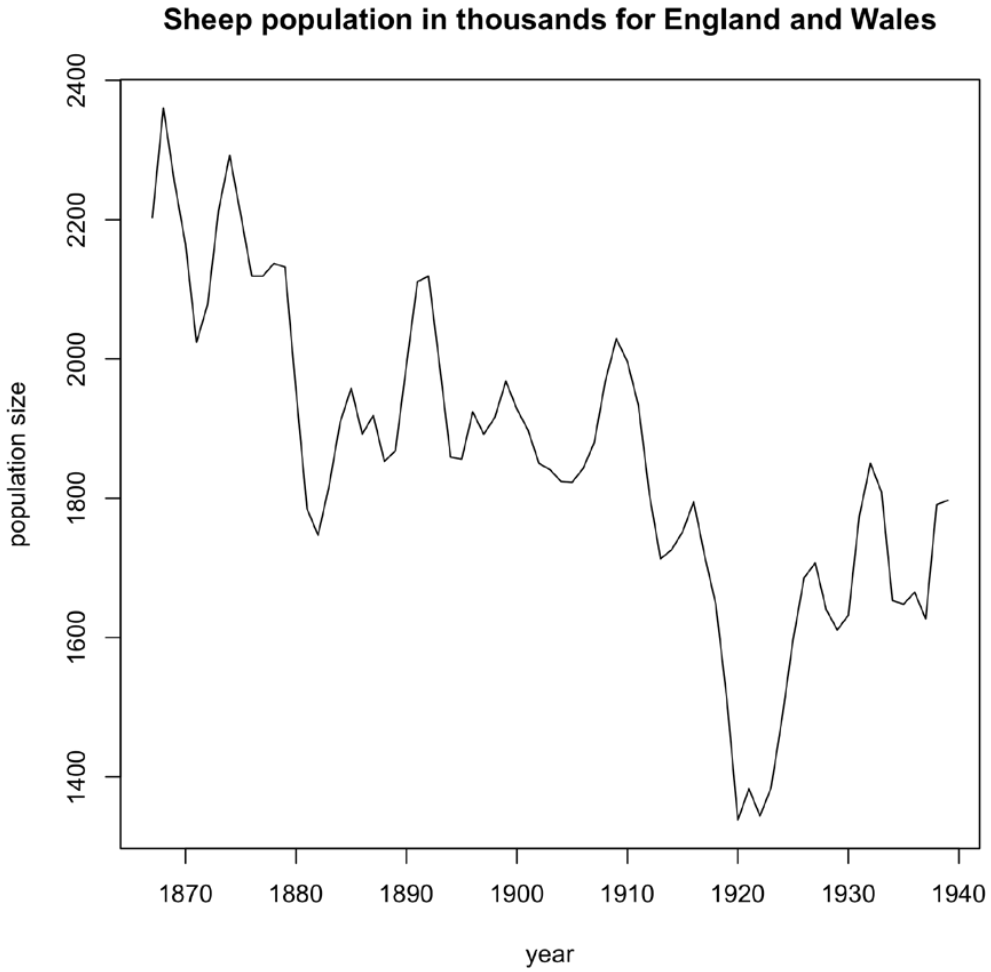
ARIMA time series approach --- Example

The time series of the annual sheep population (in thousands) of England and Wales is plotted for the years 1867 to 1939 in the diagram below:



ARIMA time series approach --- Example

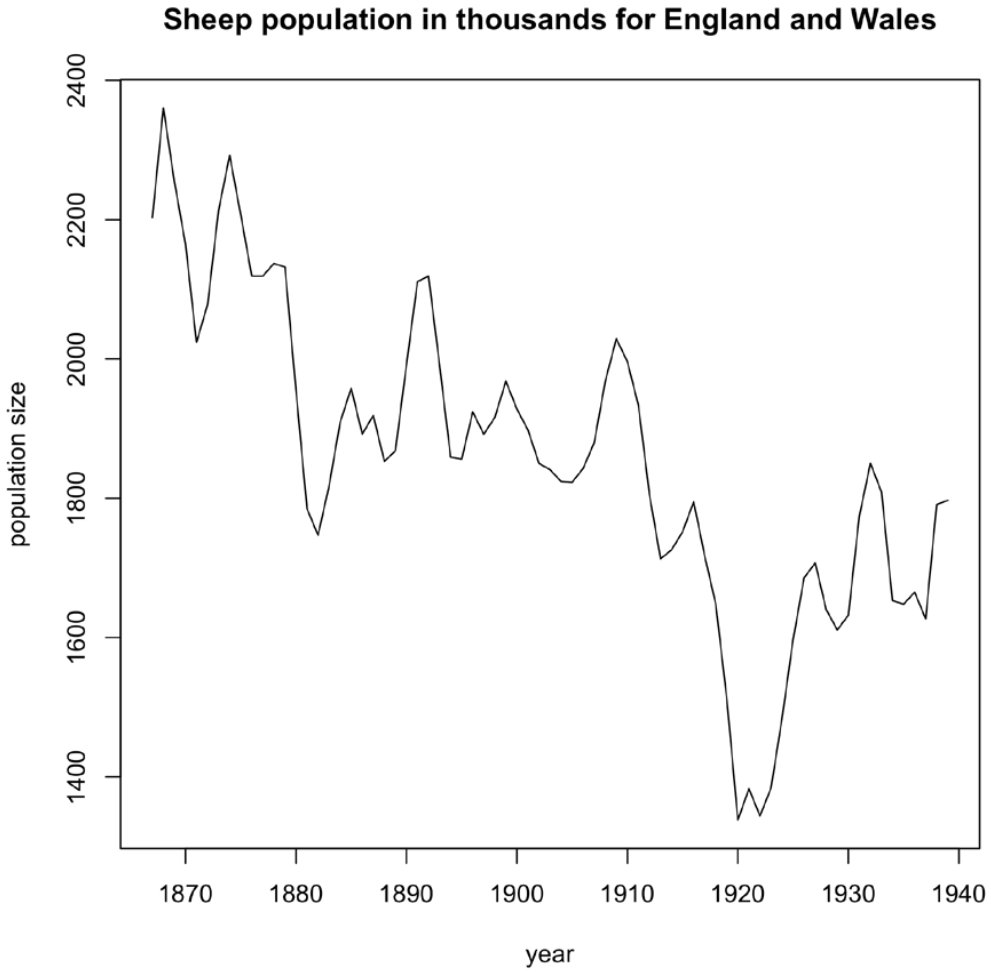
The time series of the annual sheep population (in thousands) of England and Wales is plotted for the years 1867 to 1939 in the diagram below:



a) Is the time series stationary? Justify your answer.

ARIMA time series approach --- Example

The time series of the annual sheep population (in thousands) of England and Wales is plotted for the years 1867 to 1939 in the diagram below:



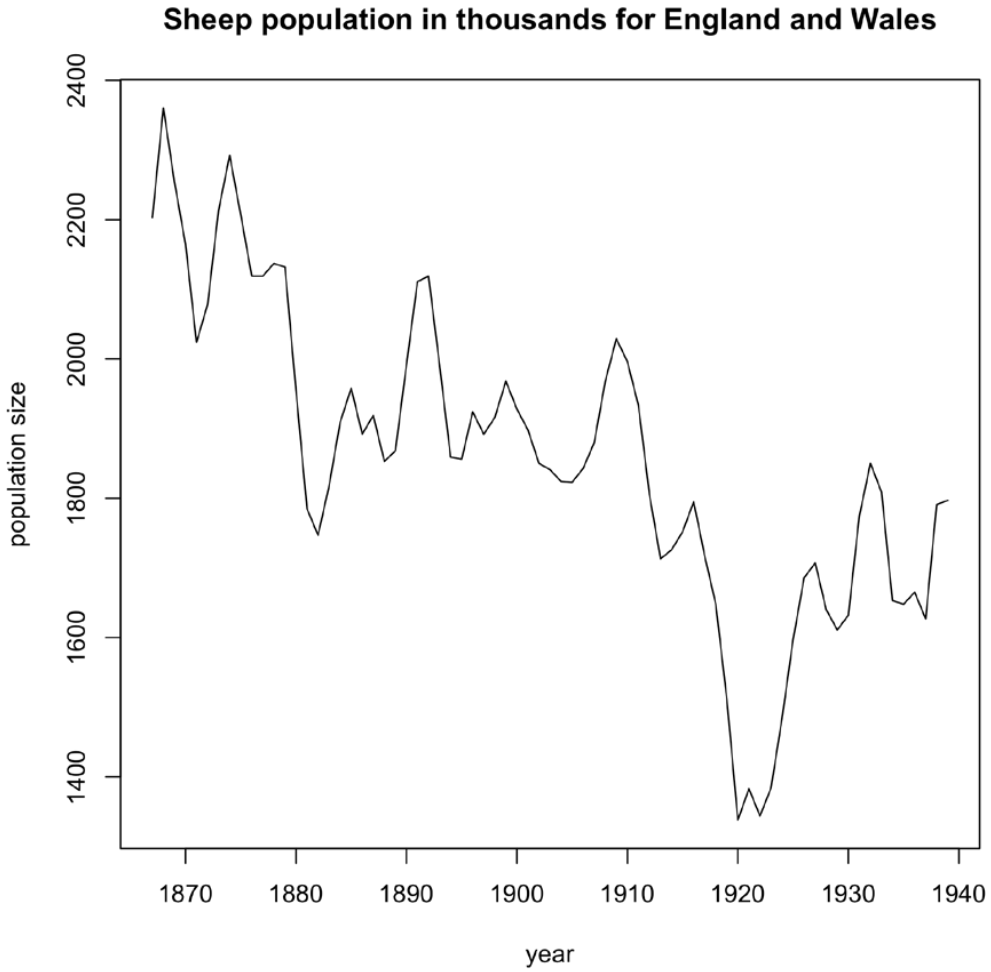
a) Is the time series stationary? Justify your answer.

No. There is a clear downward linear(potentially non-linear*) trend in the series. Hence, the series is not stationary in mean.

*For the purposes of this example, we go with the trend being linear and decreasing.

ARIMA time series approach --- Example

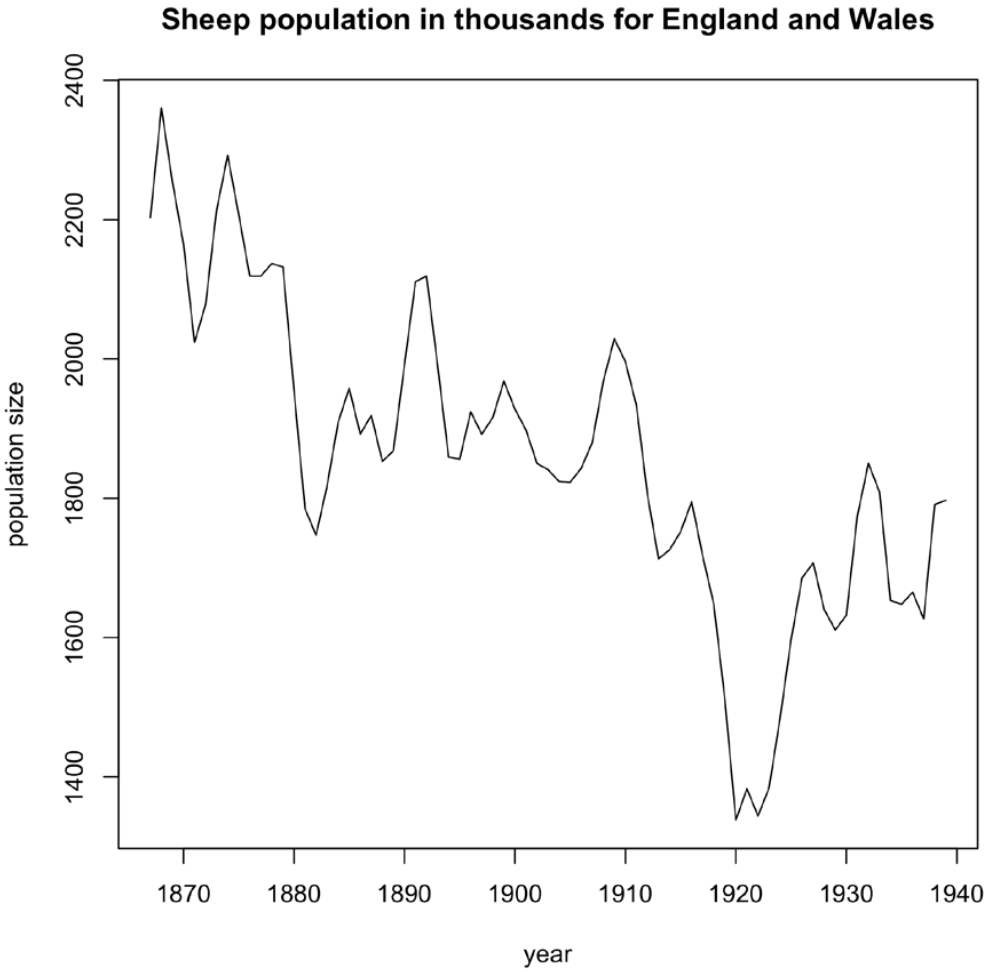
The time series of the annual sheep population (in thousands) of England and Wales is plotted for the years 1867 to 1939 in the diagram below:



b) What transformations should you apply to this series to result in a stationary series?

ARIMA time series approach --- Example

The time series of the annual sheep population (in thousands) of England and Wales is plotted for the years 1867 to 1939 in the diagram below:

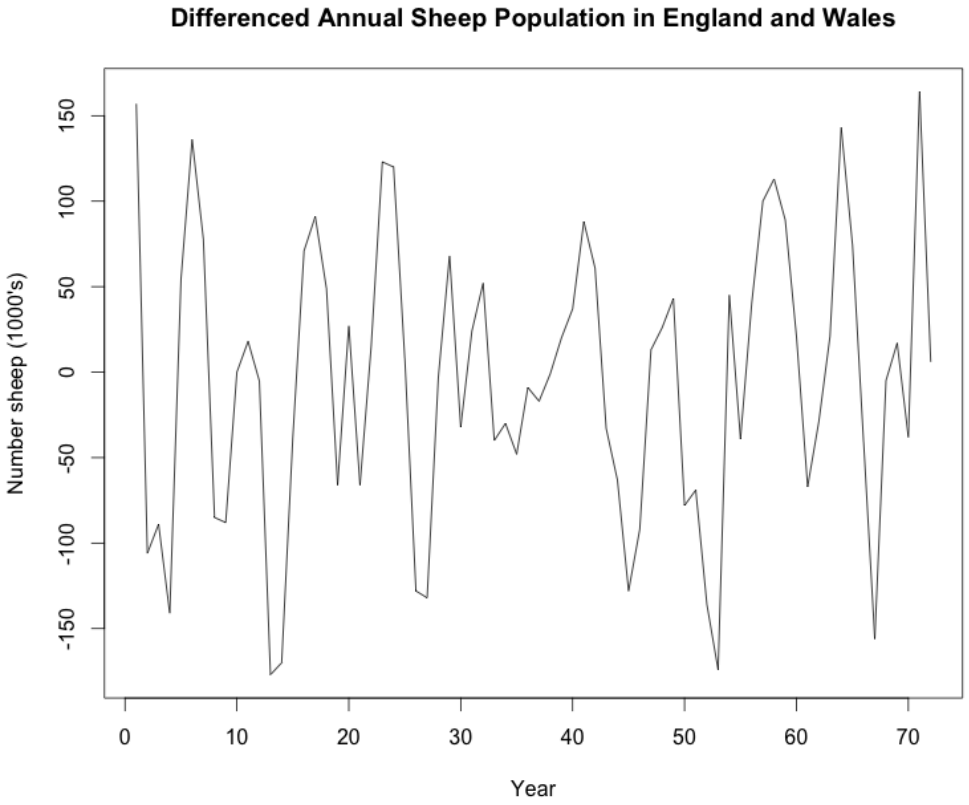


b) What transformations should you apply to this series to result in a stationary series?

The variance appears relatively constant, so there are no Box-Cox transformations needed in that regard. To remove the trend, you should apply a first-order of differencing at lag 1 i.e.

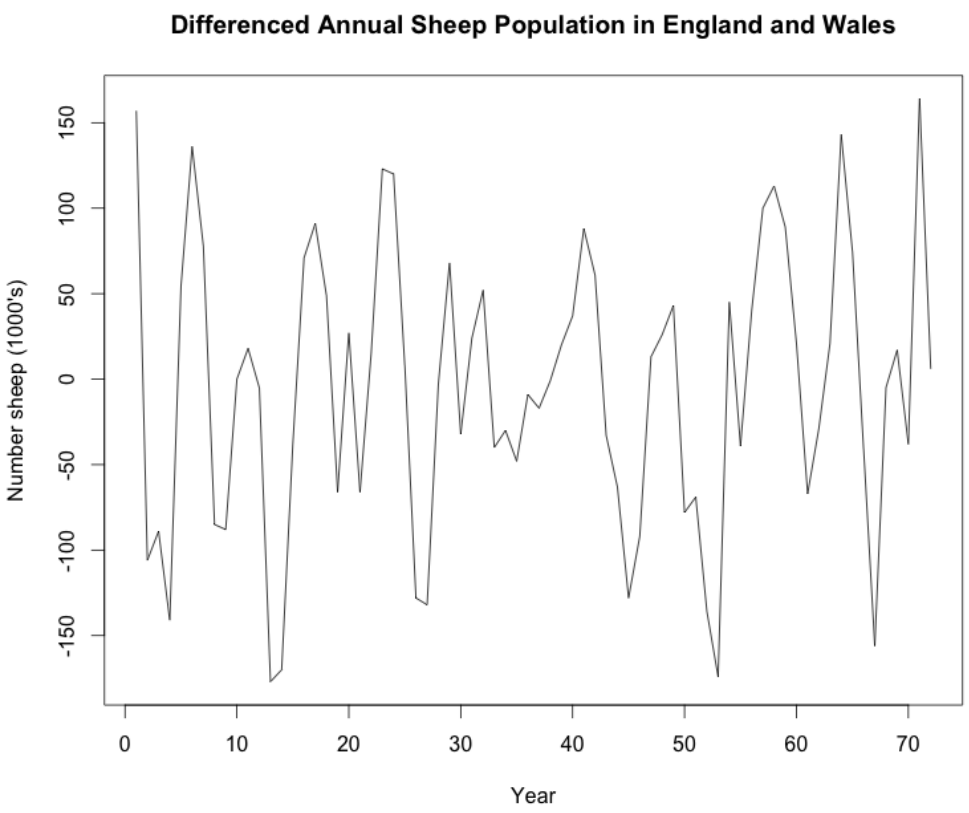
$$Y_t = X_t - X_{t-1}, \qquad t = 2, 3, \ldots, 73$$

ARIMA time series approach --- Example



c) A first-order difference at lag 1 was performed, and the differenced series plotted as shown in the figure to the left. Comment on the stationarity of the difference series. Are any more transformations required?

ARIMA time series approach --- Example



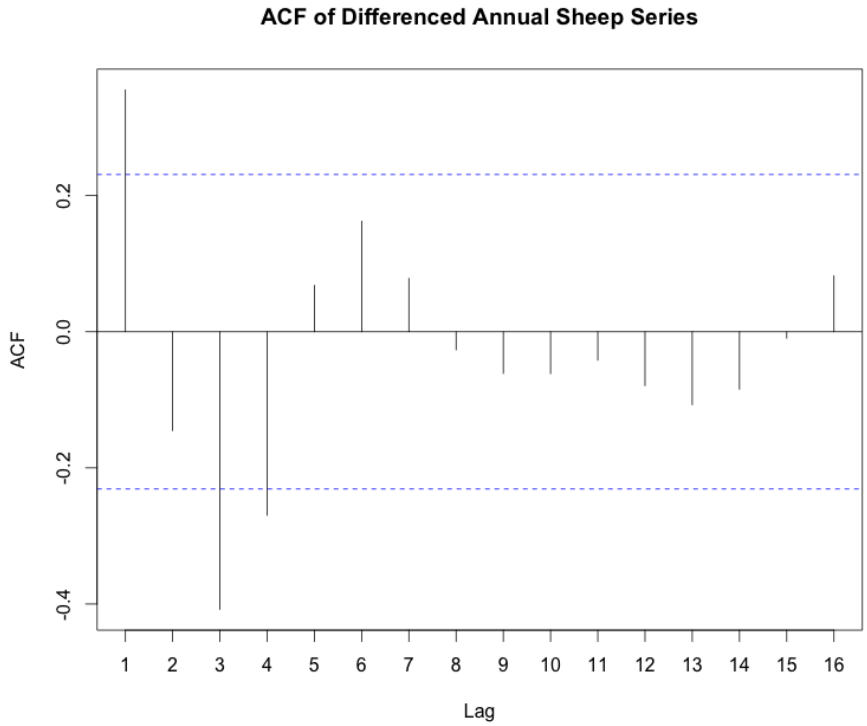
c) A first-order difference at lag 1 was performed, and the differenced series plotted as shown in the figure to the left. Comment on the stationarity of the difference series. Are any more transformations required?

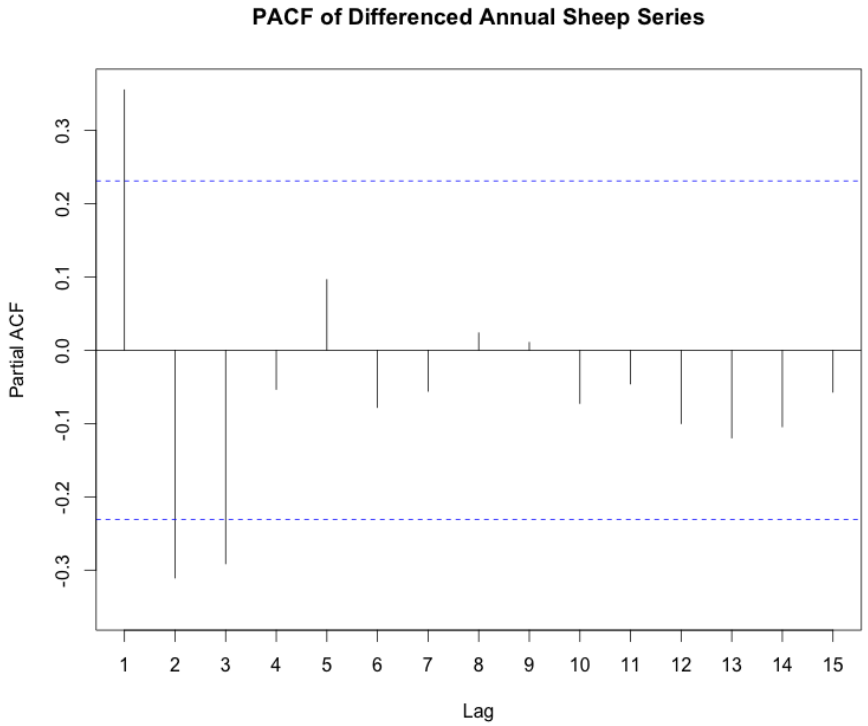
The differenced series appears stationary – it has a relatively constant mean and variance.

Therefore, no further transformations are necessary.

ARIMA time series approach --- Example

The sample ACF and PACF plots of the transformed series are plotted below:





d) Suggest an appropriate type of model to fit to this residual series, and hence an appropriate model to fit to the original data. Can you suggest what order of model to fit? Provide reasons for your answer

ARIMA time series approach --- Example

d) Suggest an appropriate type of model to fit to this transformed series, and hence an appropriate model to fit to the original data. Can you suggest what order of model to fit? Provide reasons for your answers.

Both the ACF and PACF shut off after lag 4 and 3 respectively. This behaviour is representative of a combined ARIMA(p,d,q) process, *Hence we should fit a few low-order ARIMA models and compare them with the AIC.*

If we were to fit an AR(p) model, an AR(3) model should be fit. Also, it must be noted that although the partial autocorrelations at lags 1, 2, 3 are significantly different from 0, they are only moderate in strength (just smaller than -0.3).

Since a first order difference was applied to result in a stationary series, we are fitting a non-seasonal ARIMA(3,1,0) model to the original data.

Let us suppose one of the ARIMA models fit was a non-seasonal ARIMA(3,1,0). The output from fitting the model is included below:

ARIMA(3,1,0)

Coefficients:	*ar1*	*ar2*	*ar3*	*mean*
	0.4134	*-0.2045*	*-0.3115*	*-5.8707*
s.e.	*0.1200*	*0.1367*	*0.1250*	*7.5076*

sigma^2 estimated as 4809: log likelihood=-401.6
AIC=813.2 AICc=814.11 BIC=824.58

ARIMA time series approach --- Example

ARIMA(3,1,0)

Coefficients:	*ar1*	*ar2*	*ar3*	*mean*
	0.4134	*-0.2045*	*-0.3115*	*-5.8707*
s.e.	*0.1200*	*0.1367*	*0.1250*	*7.5076*

sigma^2 estimated as 4809: log likelihood=-401.6

AIC=813.2 AICc=814.11 BIC=824.58

e) State the fitted model. What is the value of the constant term in the model?

ARIMA time series approach --- Example

ARIMA(3,1,0) with zero mean

Coefficients:	*ar1*	*ar2*	*ar3*	*mean*
	0.4134	*-0.2045*	*-0.3115*	*-5.8707*
s.e.	*0.1200*	*0.1367*	*0.1250*	*7.5076*

sigma^2 estimated as 4809: log likelihood=-401.6

AIC=813.2 AICc=814.11 BIC=824.58

e) State the fitted model. What is the value of the constant term in the model?

$$X_t = c + 0.4134X_{t-1} - 0.2045X_{t-2} - 0.3115X_{t-3} + Z_t$$

The estimated mean of the AR(3) series is given by -5.8707. Therefore, the constant is calculated as:

$$c = -5.8707(1 - 0.4134 + 0.2045 + 0.3115) = -6.4730$$

ARIMA time series approach --- Example

ARIMA(3,1,0) with zero mean

Coefficients:	*ar1*	*ar2*	*ar3*	*mean*
	0.4134	*-0.2045*	*-0.3115*	*-5.8707*
s.e.	*0.1200*	*0.1367*	*0.1250*	*7.5076*

sigma^2 estimated as 4809: log likelihood=-401.6

AIC=813.2 AICc=814.11 BIC=824.58

f) Interpret the value of the AIC

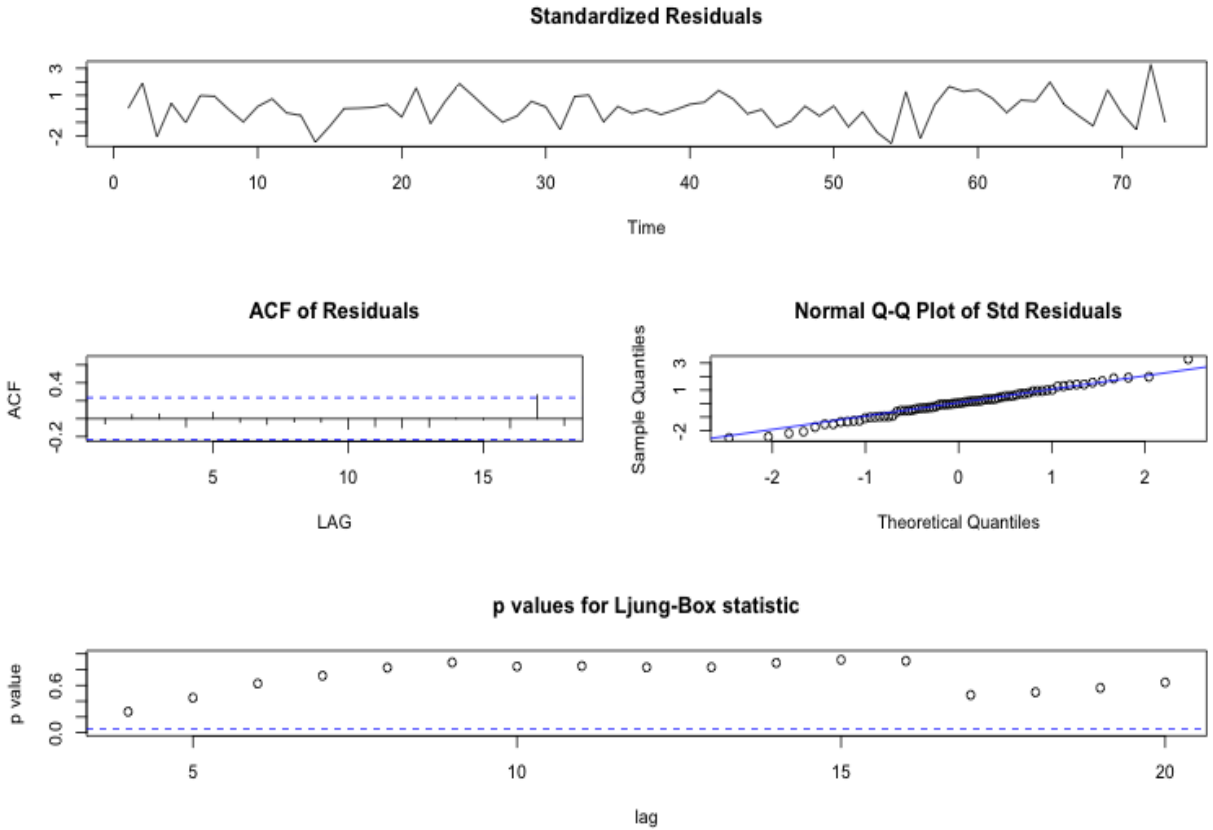
ARIMA time series approach --- Example

f) Interpret the values of AIC

The AIC value does not have a direct interpretation in this context as we fitted only 1 model, and thus were not comparing fitted models with one another.

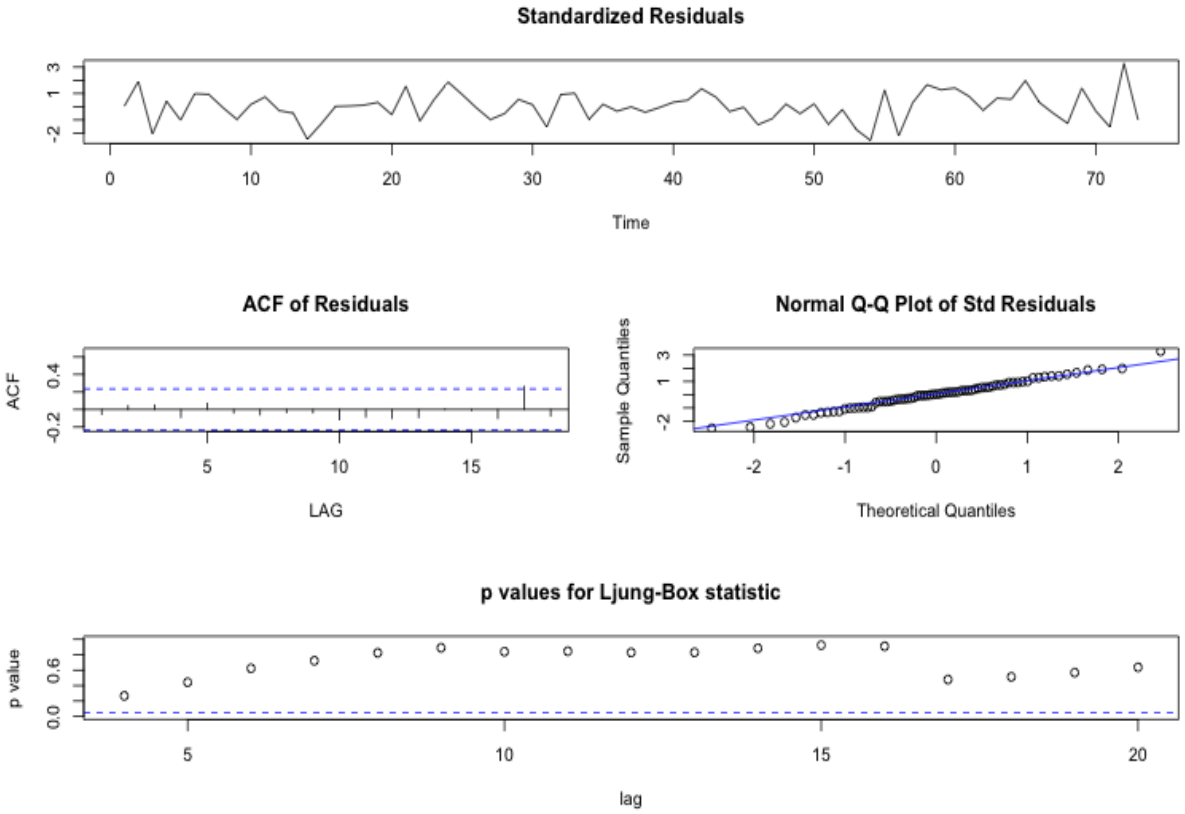
ARIMA time series approach --- Example

Residual diagnostics were carried out on the fitted ARIMA(3,1,0) model, as displayed in the figure below:



h) Is the ARIMA(3,1,0) an adequate model to forecast values of the Sheep population data? Justify your answer by referring to *each* element of the output

ARIMA time series approach --- Example

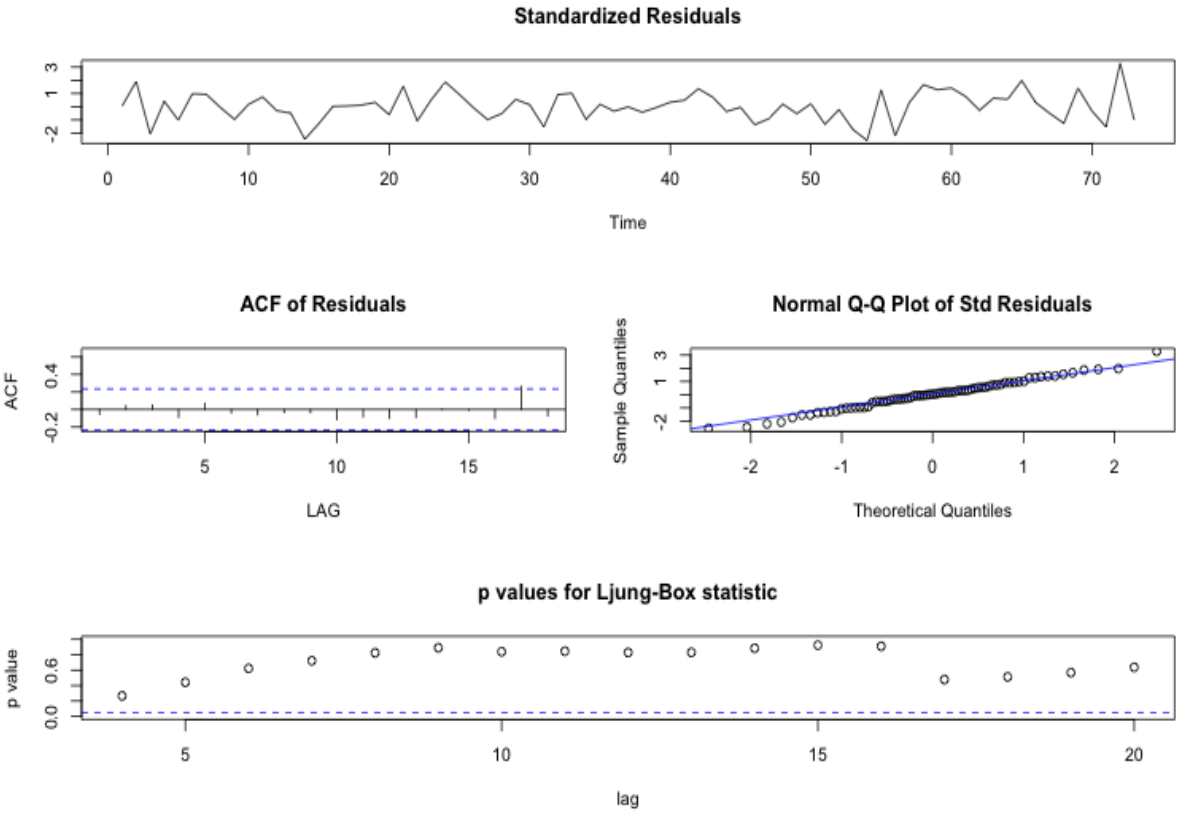


h) Is the ARIMA(3,1,0) an adequate fit to the Sheep population data? Justify your answer by referring to *each* element of the output

YES. The plot of the standardised residuals does not show a pattern, and they appear to have a mean of 0 and relatively constant variance.

The ACF of the residuals contains very small values apart from the one at lag 17. It is possible that this is significant due to chance or sampling error (there is only 1 out of 20 ACF values that are significantly different from 0).

ARIMA time series approach --- Example



The normal Q-Q plot does not indicate any serious deviations from normality.

The p-values for the Ljung—Box test are all > 0.05, indicating that we would not reject the null hypothesis that the residuals are independent at any of the first 20 lags.

Hence, the ARIMA(3,1,0) is an adequate fit to the original data.

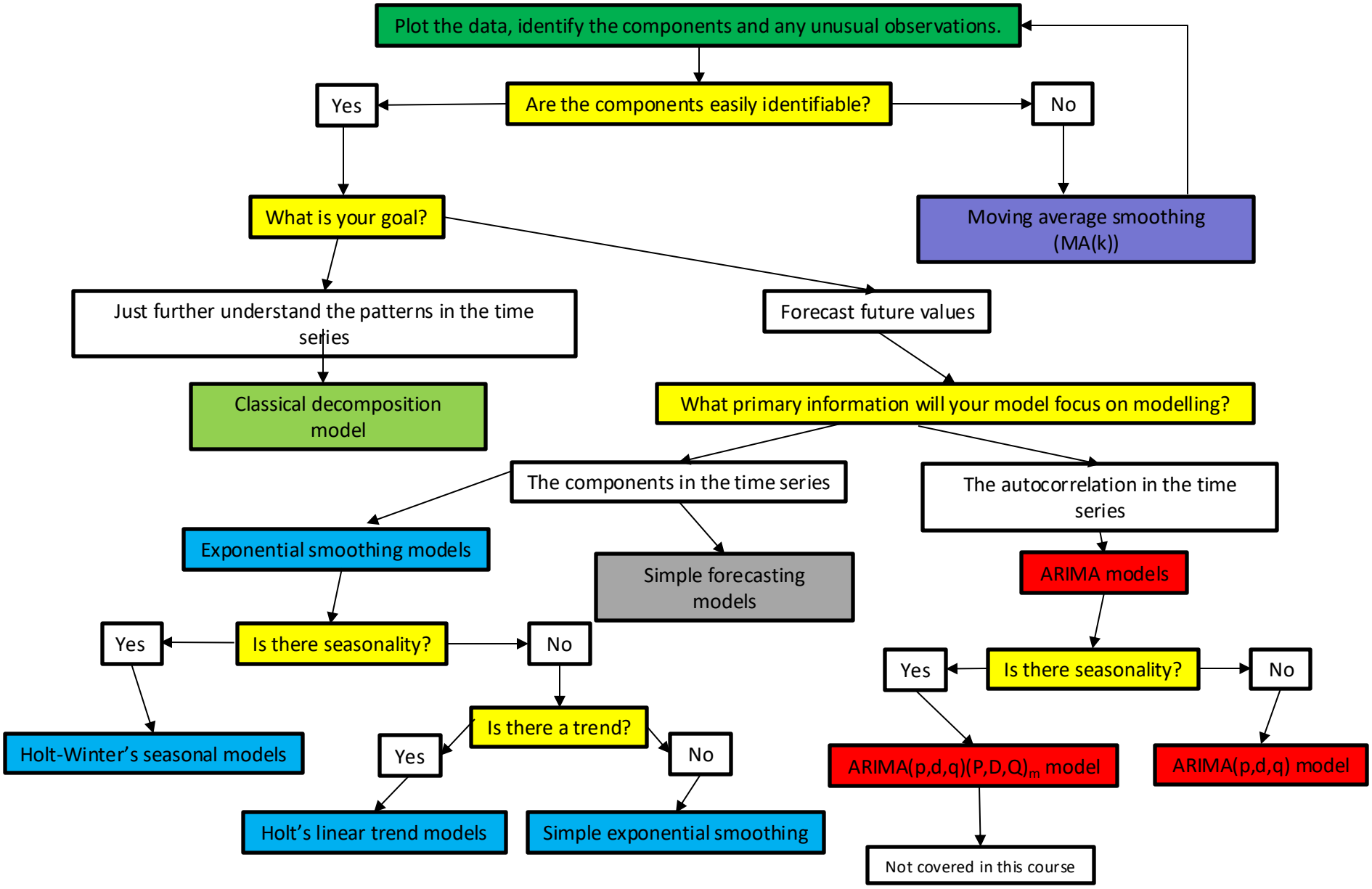
Note that you should be comfortable interpreting both residual diagnostic output like this and the output from the checkresiduals() function.

ALWAYS comment on each part of the output when assessing the residuals of a fitted model.

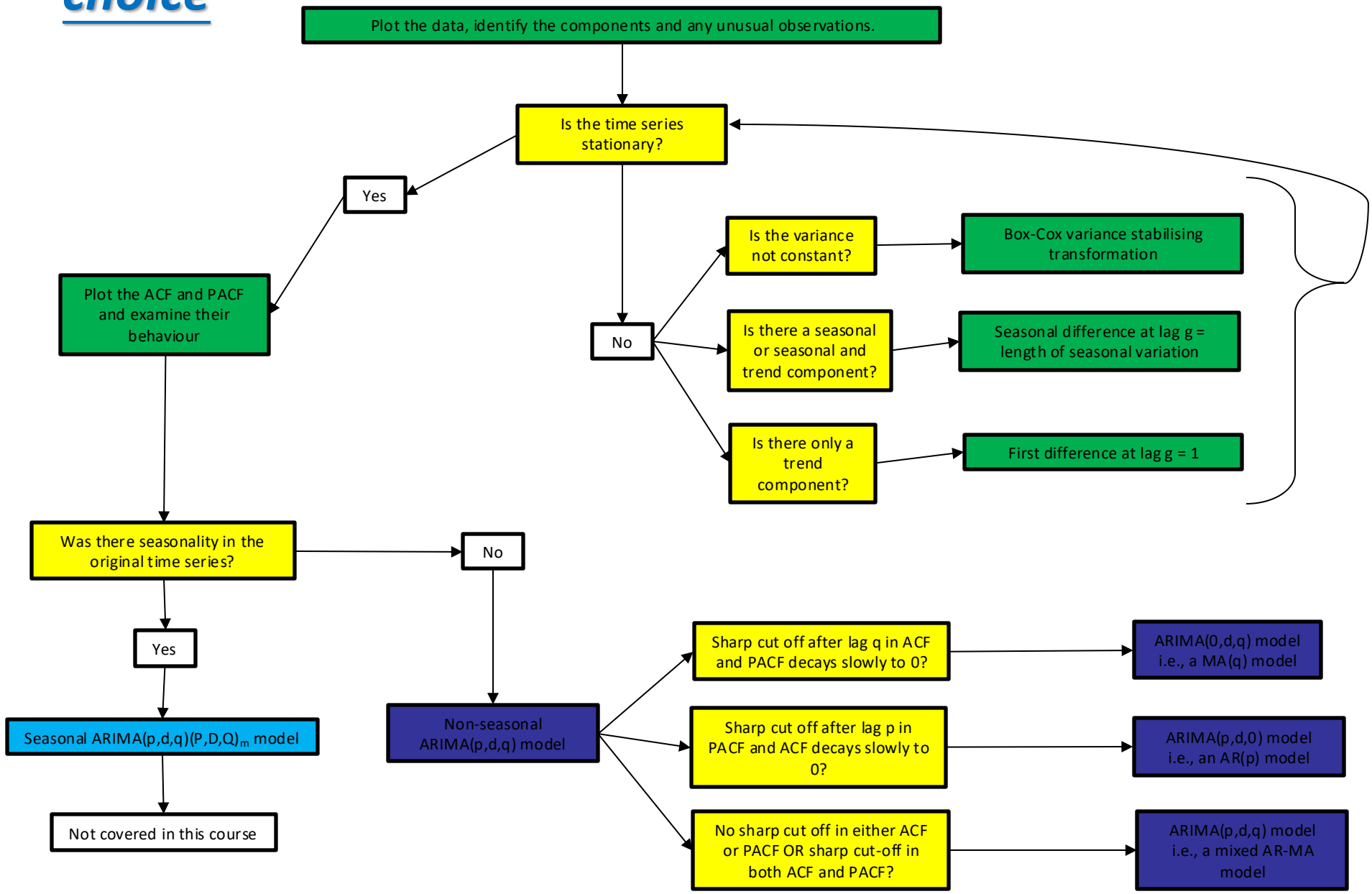
Overall time series process examples

Roadmaps & Conducting a time series analysis from start to finish

General time series approach --- Model choice



General time series approach --- ARIMA model choice



General time series approach --- Exponential Smoothing model choice (with simple forecasting methods)