Regression (Part 2) and Time Series Decomposition (intro)

Forecasting: principles and practice (2nd edition)

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

Best subset regression

```
library(fpp2)
```

- Where possible, all potential regression models can be fitted and the best one selected based on one of the measures discussed here. This is known as "best subsets" regression or "all possible subsets" regression.
- It is recommended that one of CV, AIC or AICc be used for this purpose. If the value of T is large enough, they will all lead to the same model.
- While $\bar{R}^2$ is very widely used, and has been around longer than the other measures, its tendency to select too many variables makes it less suitable for forecasting than either CV, AIC or AICc.
- Also, the tendency of BIC to select too few variables makes it less suitable for *multiple regression models* for forecasting than either CV, AIC or AICc.

Stepwise regression

If there are a large number of predictors, it is not possible to fit all possible models. An approach that works quite well is backwards stepwise regression:

- Start with the model containing all potential predictors.
- Try subtracting one predictor at a time. Keep the model if it improves the measure of predictive accuracy.
- Iterate until no further improvement.

It is important to realize that a stepwise approach is not guaranteed to lead to the best possible model. But it almost always leads to a good model.

Beware of inference after selecting predictors

We do not discuss statistical inference of the predictors in this course (e.g., looking at p-values associated with each predictor).

The procedures the authors recommend for selecting predictors are helpful when the model is used for forecasting; they are not helpful if you wish to study the effect of any predictor on the forecast variable.

Forecasting with regression

Ex ante forecasting

Ex ante forecasts are those that are made using only the information that is available in advance. e.g., *ex-ante forecasts* for the percentage change in US consumption for quarters following the end of the sample in *data=uschange*, should only use information that was available up to and including 2016 Q3. These are genuine forecasts, made in advance using whatever information is available at the time. Therefore in order to generate *ex-ante forecasts*, the model requires future values (forecasts) of the predictors.

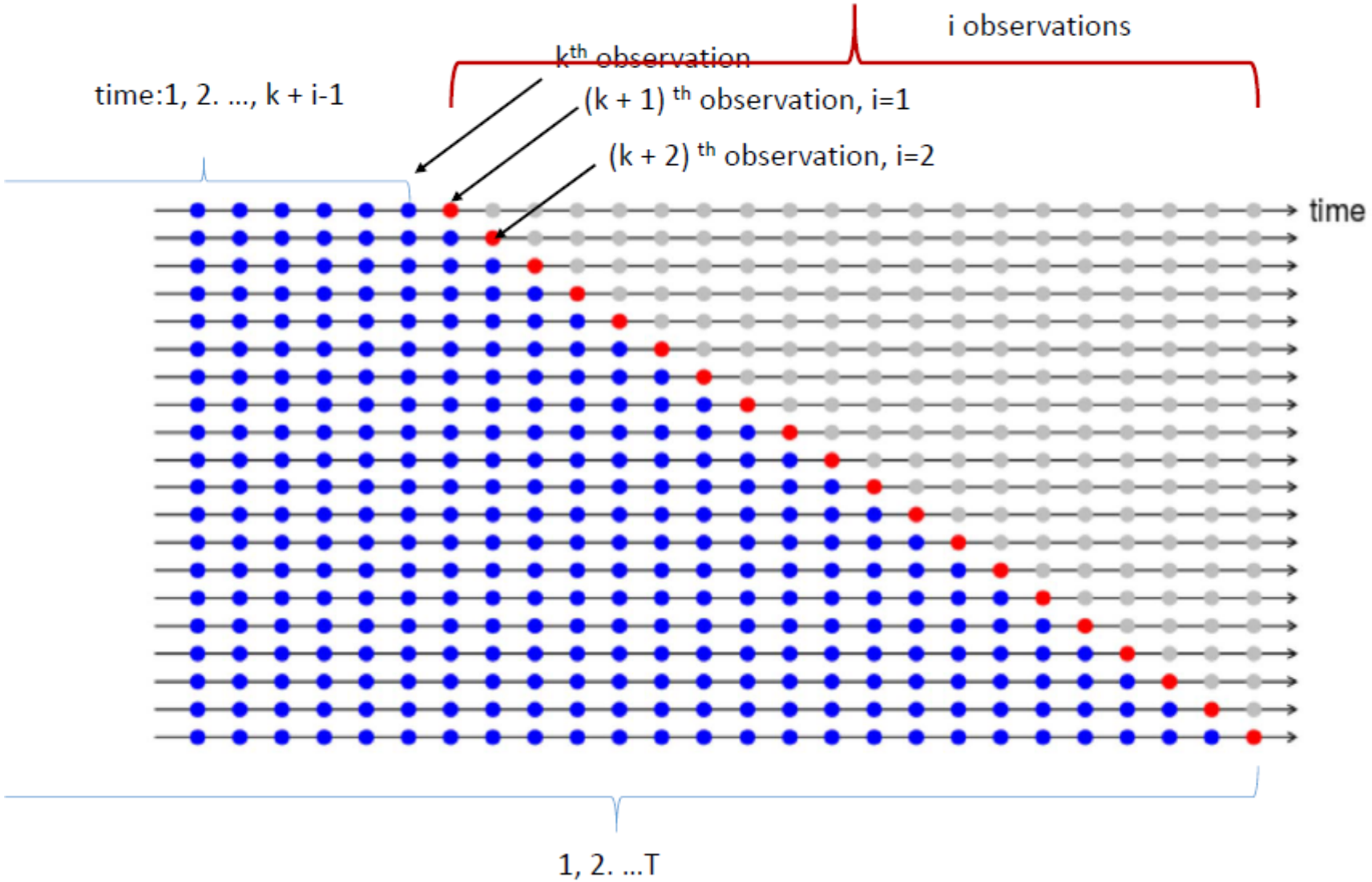
To obtain these we can use one of the simple methods introduced in Section 3.1 or more sophisticated pure time series approaches that follow in Chapters 7 (ETS models) and 8 (ARIMA models). Alternatively, forecasts from some other source, such as government agencies, may be available and can be used.

Ex post forecast

Ex post forecasts are those that are made using later information on the predictors. e.g., ex post forecasts of consumption may use the actual observations of the predictors, once these have been observed. These are not genuine forecasts, but are useful for studying the behaviour of forecasting models.

The model from which ex-post forecasts are produced should not be estimated using data from the forecast period. That is, ex-post forecasts can assume knowledge of the predictor variables (the x variables), but should not assume knowledge of the data that are to be forecast (the y variable).

Example of ex post forecast


i observations
k th observation
time:1, 2. ..., k + i-1
(k + 1) th observation, i=1
(k + 2) th observation, i=2
time
1, 2. ...T

Cross validation in regression models

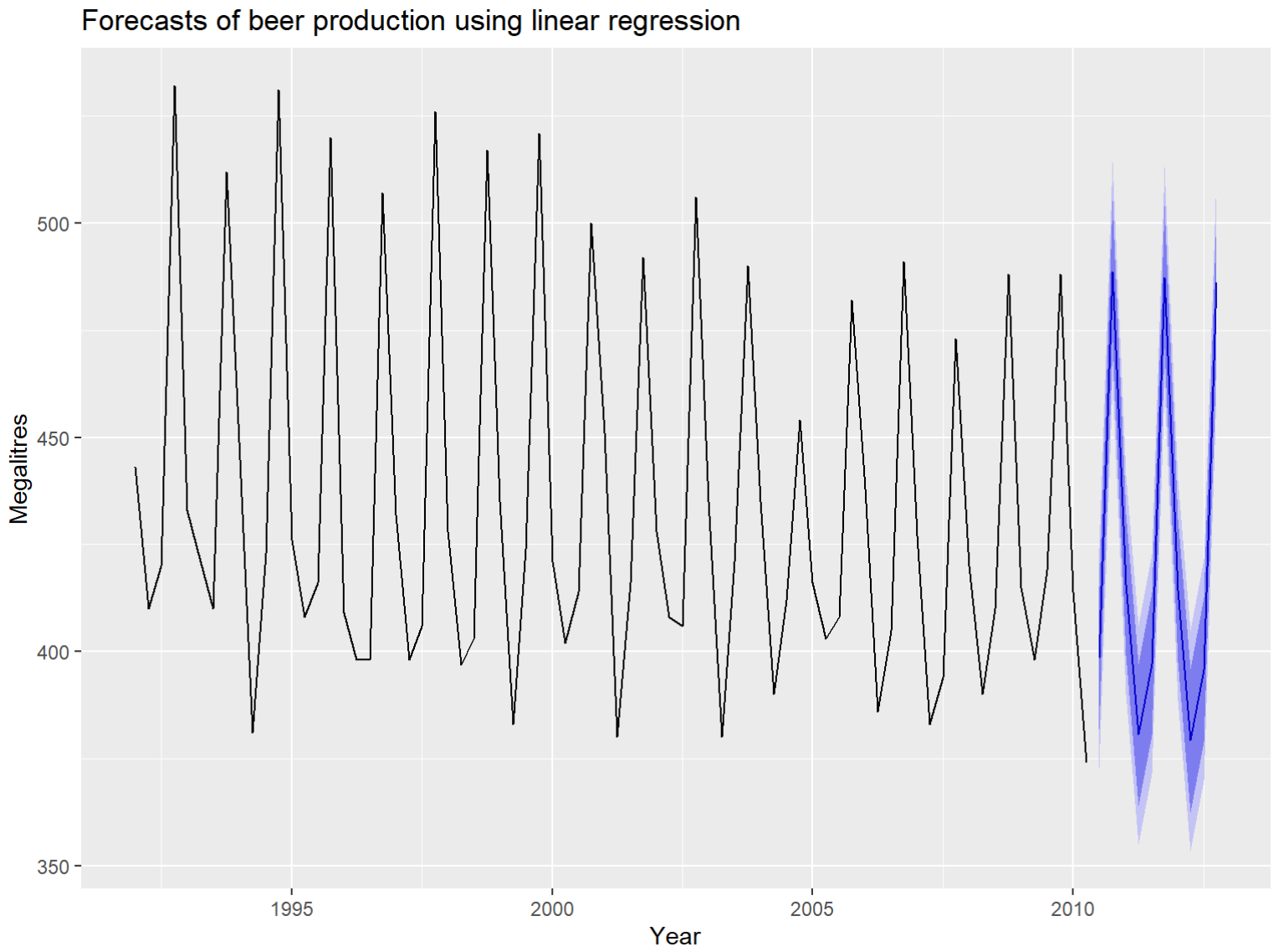
Ex ante and ex post forecasts

- A comparative evaluation of ex-ante forecasts and ex-post forecasts can help to separate out the sources of forecast uncertainty.
- This will show whether forecast errors have arisen due to poor forecasts of the predictor (ex ante forecast) or due to a poor forecasting model (ex post forecast).

Example: Australian quarterly beer production

Normally, we cannot use actual future values of the predictor variables when producing ex-ante forecasts because their values will not be known in advance. However, the special predictors introduced in Section 5.3 are all known in advance, as they are based on calendar variables (e.g., seasonal dummy variables or public holiday indicators) or deterministic functions of time (e.g. time trend). In such cases, there is no difference between ex ante and ex post forecasts.

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



Note: Data are only until 2010 Q2. 2010Q3 onward are forecast.

Scenario based forecasting

The forecaster assumes possible scenarios for the predictor variables:

scenario 1:

a US policy maker may be interested in comparing the predicted change in consumption when there is a constant growth of 1% and 0.5% respectively for *income* and *savings* with no change in the employment rate.

scenario 2:

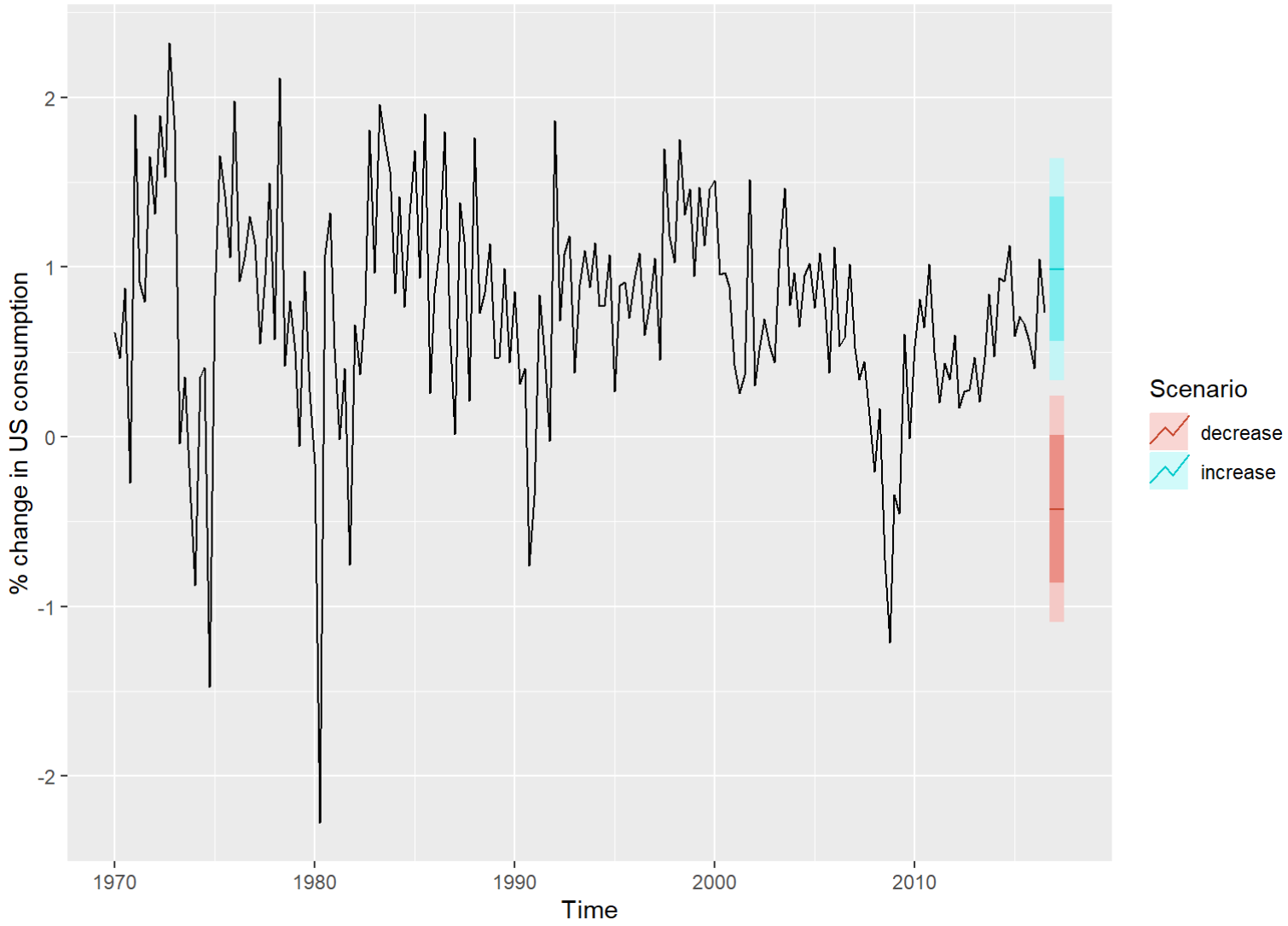
There are respective declines of 1% and 0.5%, respectively for *income* and *saving* for each of the four quarters with no change in unemployment following the end of the sample.

The resulting forecasts are calculated in the next slide

Note that prediction intervals for scenario based forecasts do not include the uncertainty associated with the future values of the predictor variables. They assume that the values of the predictors are known in advance

```
fit.consBest <- tslm(Consumption ~ Income + Savings + Unemployment,
data = uschange)
h <- 4
# first scenario
newdata <-
  cbind(Income=c(1,1,1,1),# assigns 1% rise in income for each h-ahead q
  Savings=c(0.5,0.5,0.5,0.5),# assigns 0.5 rise in income for each h-ahead q
  Unemployment=c(0,0,0,0)) %>% # unemployment is unchanged
as.data.frame()
fcast.up <- forecast(fit.consBest, newdata=newdata)
# second scenario
newdata <-
  cbind(Income=rep(-1,h),
  Savings=rep(-0.5,h),
  Unemployment=rep(0,h)) %>%
as.data.frame()
fcast.down <- forecast(fit.consBest, newdata=newdata)
```

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


% change in US consumption
2
1
0
-1
-2
1970
1980
1990
2000
2010
Time
Scenario
decrease
increase

Building a predictive regression model

The great advantage of regression models is that they can be used to capture important relationships between the forecast variable of interest and the predictor variables.

A major challenge however, is that in order to generate ex-ante forecasts the model requires future values of each predictor.

- If scenario based forecasting is of interest then these models are extremely useful.
- However, if ex-ante forecasting is the main focus, obtaining forecasts of the predictors can be very challenging (in many cases generating forecasts for the predictor variables can be more challenging than forecasting directly the forecast variable without using predictors).
- An alternative formulation is to use as predictors their lagged values. This is the topic of Chapter 9: *Dynamic regression models*.

Prediction intervals

- With each forecast for the change in consumption in the last figure (Figure 5.18 in the text) 95% and 80% prediction intervals are also included. (# please refer to the comments in the above codes)

- The general formulation of how to calculate the prediction intervals are in Section 5.7 of the online text (2nd edition) but will not be discussed in class.

- In section 5.7, it shows PI based on the normal distribution. If the distribution is far from normal, the bootstrap method is useful way to build PI (section 11.4).

PI: a simplified example

Consider our example where the model is a regression of US consumption against personal income:

$$\hat{y} = 0.545 + 0.28x_t\,.$$

Assuming that for the next four quarters

scenario 1:

-personal income will increase by its historical mean value of $\bar{x} = 0.72$,

scenario 2:

- personal income is assumed to increase by an extreme value of 10%

```
fit.cons <- tslm(Consumption ~ Income, data=uschange)
h <- 4
fcast.up <- forecast(fit.cons,
  newdata=data.frame(Income=rep(mean(uschange[,"Income"]), h)))
# income rises by its past mean
fcast.down <- forecast(fit.cons,
  newdata=data.frame(Income=rep(10,h))) #income rises by 10%
autoplot(uschange[,"Consumption"]) +
  ylab("% change in US consumption") +
  forecast::autolayer(fcast.up, PI=TRUE, series="Average increase") +
  forecast::autolayer(fcast.down, PI=TRUE, series="Extreme increase") +
guides(colour=guide_legend(title="Scenario"))
```


5.0
2.5
0.0
-2.5
% change in US consumption
1970
1980
1990
2000
2010
Time
Scenario
Average increase
Extreme increase

Scenario 1:

```
fcast.down
```

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2016 Q4       3.351116 2.389508 4.312725 1.876065 4.826167
## 2017 Q1       3.351116 2.389508 4.312725 1.876065 4.826167
## 2017 Q2       3.351116 2.389508 4.312725 1.876065 4.826167
## 2017 Q3       3.351116 2.389508 4.312725 1.876065 4.826167
```

Scenario 2:

```
fcast.up
```

```
##         Point Forecast       Lo 80    Hi 80    Lo 95    Hi 95
## 2016 Q4      0.7464708 -0.03063903 1.523581 -0.44557 1.938512
## 2017 Q1      0.7464708 -0.03063903 1.523581 -0.44557 1.938512
## 2017 Q2      0.7464708 -0.03063903 1.523581 -0.44557 1.938512
## 2017 Q3      0.7464708 -0.03063903 1.523581 -0.44557 1.938512
```

Non-linear regression

Sometimes the relationship between the forecast variable and a predictor is not linear, and then the usual multiple regression equation needs modifying. We discussed log transformations and a piecewise-linear trend in a model. Allowing other variables to enter in a nonlinear manner can be handled similarly.

To keep things simple, suppose we have only one predictor x:

$$y = f(x) + e.$$

In nonlinear regression, we allow f to be a nonlinear function of x.

The most commonly used transformation is the (natural) logarithmic. Recall that in order to perform a logarithmic transformation to a variable, all its observed values must be greater than zero.

A *log-log* functional form is specified as

$$\log y_t = \beta_0 + \beta_1 \log x_t + \varepsilon_t.$$

In this model, the slope β_1 can be interpeted as an elasticity: β_1 is the average percentage change in y resulting from a 1% change in x.

- Recall that in order to perform a logarithmic transformation to a variable, all of its observed values must be greater than zero.
- In the case that variable contains zeros, we use the transformation $\log(x+1)$; i.e., we add one to the value of the variable and then take logarithms.
- This has a similar effect to taking logarithms but avoids the problem of zeros.
- It also has the neat side-effect of zeros on the original scale remaining zeros on the transformed scale.

One of the simplest ways to do nonlinear regression is to make f piecewise linear. That is, we introduce points where the slope of f can change. These points are called *knots*. This can be achieved using the following letting $x_{1,t} = x$ and introducing a variable $x_{2,t}$ such that:

$$x_{2,t} = (x - c)_+ = \begin{cases} 0 & \text{if } x < c \\ x - c & \text{if } x \geq c \end{cases}$$

The notation $(x - c)_+$ means the value $x - c$, if it is positive and 0 otherwise. This forces the slope to bend at point c. Additional bends can be included in the relationship by adding further variables of the above form.

An example of this follows by considering $x = t$ and fitting a piecewise linear trend to a times series (this is a special case of *regression splines*)

Regression splines

In general, a linear regression spline is obtained using

$$x_1 = x \quad x_2 = (x - c_1)_+ \ \ldots \ x_k = (x - c_{k-1})_+.$$

where $c_1, \ \ldots, c_{k-1}$ are the knots (the points at which the line can bend). Selecting the number of knots $(k-1)$ and where they should be positioned can be difficult and somewhat arbitrary. Some automatic knot selection algorithms are available in some software, but are not yet widely used.

This is beyond the scope of the course but there are many references on regression splines if we search online including https://towardsdatascience.com/unraveling-spline-regression-in-r-937626bc3d96

A smoother result is obtained using piecewise cubics rather than piecewise lines. These are constrained so they are continuous (they join up) and they are smooth (so there are no sudden changes of direction as we see with piecewise linear splines).

In general, a cubic regression spline is written as

$$x_1 = x \;\; x_2 = x^2 \;\; x_3 = x^3 \;\; x_4 = (x - c_1)_+ \;\; \ldots \;\; x_k = (x - c_{k-3})_+.$$

Cubic splines usually give a better fit to the data. However, forecasting values of when it is outside the range of the historical data becomes very unreliable.

Forecasting with a nonlinear trend

The simplest way of fitting a nonlinear trend is using quadratic or higher order trends obtained by specifying

$$x_{1,t} = t, \;\; x_{2,t} = t^2, \ldots$$

However, it is not recommended that quadratic or higher order trends be used in forecasting. When they are extrapolated, the resulting forecasts are often very unrealistic.

A linear trend is easily accounted for by including the predictor $x_{1,t} = t$. A piecewise linear trend with a bend at time τ can be specified by including the following predictors in the model.

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

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

If the associated coefficients of $x_{1,t}$ and $x_{2,t}$ are β_1 and β_2, then β_1 gives the slope of the trend before time τ, while the slope of the line after time τ is given by $\beta_1 + \beta_2$. Additional bends can be included in the relationship by adding further variables of the form $(t - \tau)_+$ where is the "knot" or point in time at which the line should bend.

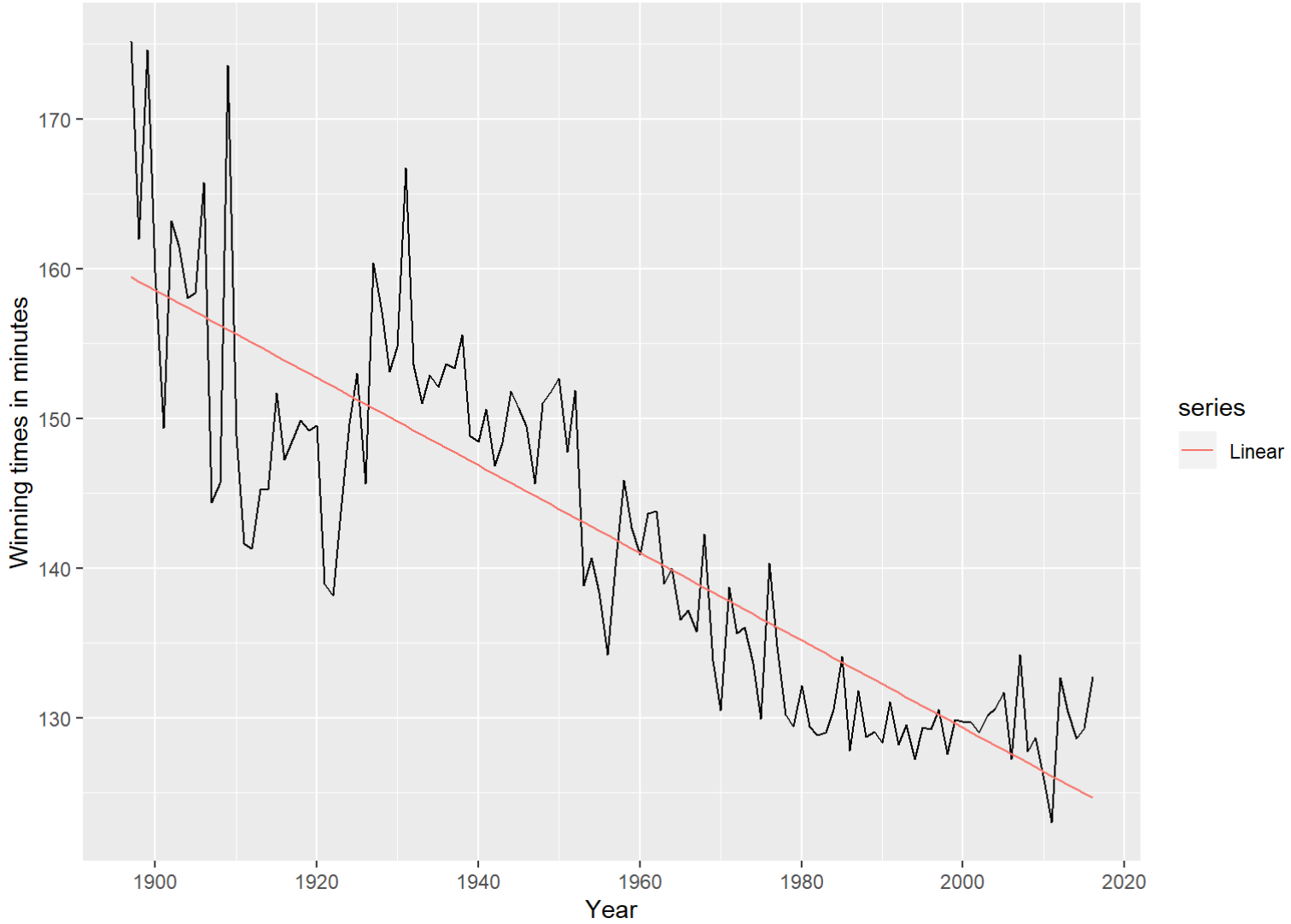
Example: Boston marathon winning times

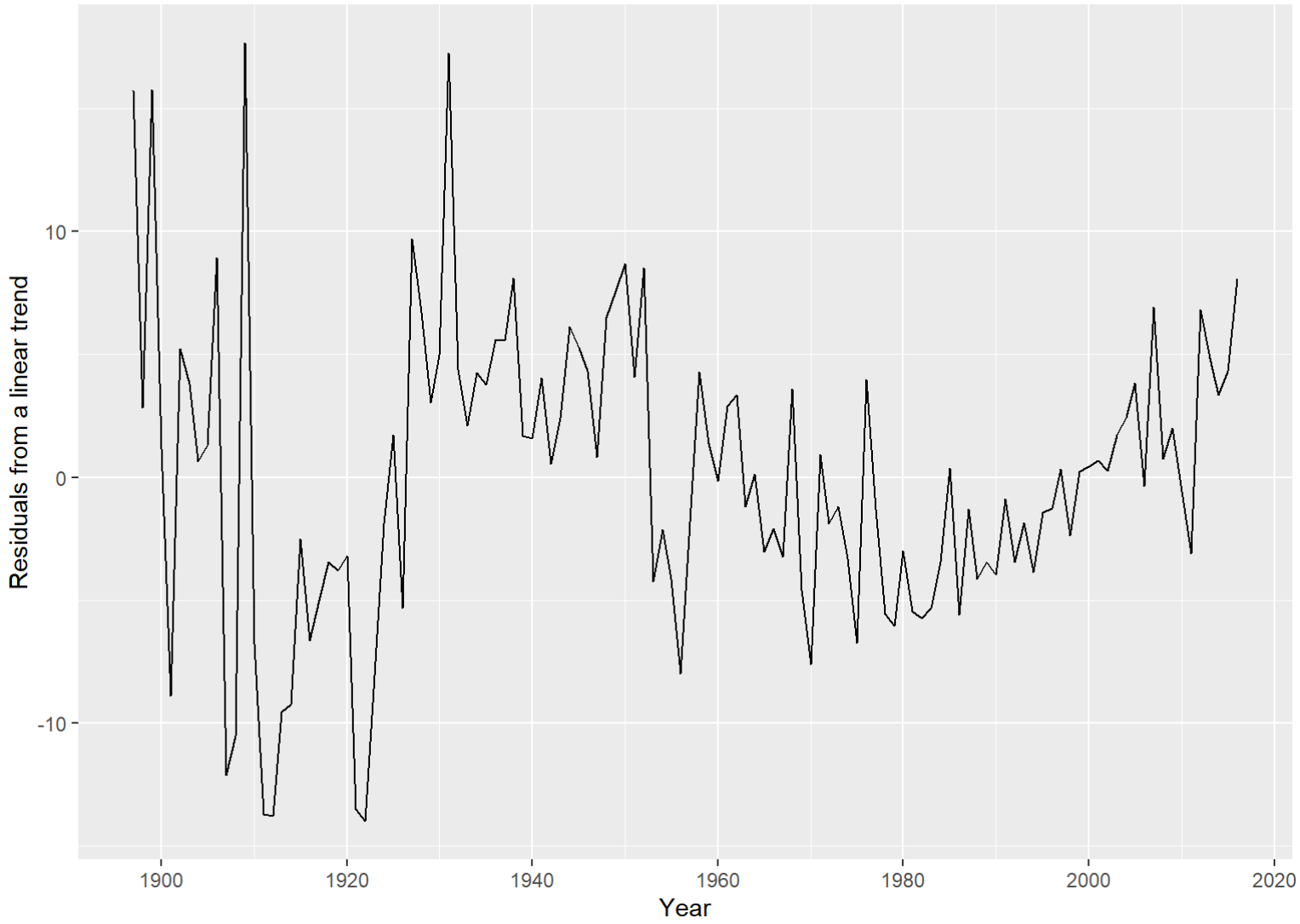
To show a graph of the Boston marathon winning times (in minutes) since it started in 1897 and a linear trend:

```
fit.lin <- tslm(marathon ~ trend)
autoplot(marathon) +
  forecast::autolayer(fitted(fit.lin), series = "Linear") +
  xlab("Year") + ylab("Winning times in minutes")
```

and to show a graph of its residuals from fitting a linear trend to the data:

```
res.mar <- residuals(tslm(marathon ~ trend))
autoplot(res.mar)+
  xlab("Year") + ylab("Residuals from a linear trend")
```


Winning times in minutes
170
160
150
140
130
1900
1920
1940
1960
1980
2000
2020
Year
series
Linear


Residuals from a linear trend
10
0
-10
1900
1920
1940
1960
1980
2000
2020
Year

The plots showed an obvious nonlinear pattern which has not been captured by the linear trend.

Fitting an exponential trend to the data can be achieved by transforming the variable so that the model to be fitted is,

$$\log y_t = \beta_0 + \beta_1 t + \varepsilon_t.$$

The fitted exponential trend and forecasts are shown in slide #38. Although the exponential trend does not seem to fit the data much better than the linear trend, it gives a more sensible projection in that the winning times will decrease in the future but at a decaying rate rather than a linear non-changing rate.

The plot of winning times will reveal three very different periods.

1. There is a lot of volatility in the winning times up to about 1940, with the winning times decreasing overall but with significant increases during the 1920s;

2. After 1940 there is a nearly-linear decrease in times;

3. The near linear decrease is followed by a flattening out after the 1980s with almost an upturn towards the end of the sample.

Following these observations we specify as knots: the years *1940* and *1980*.

Warning: Subjectively identifying suitable knots in the data can lead to over-fitting which can be detrimental to the forecast performance of a model and should be performed with caution. Knots in the case below were identified subjectively. There are methods in time series econometrics used to endogenously identify structural breaks but the gains from running these tests may be small in forecasting.

Plots of the exponential, linear and piecewise linear trends could be done with the following commands:

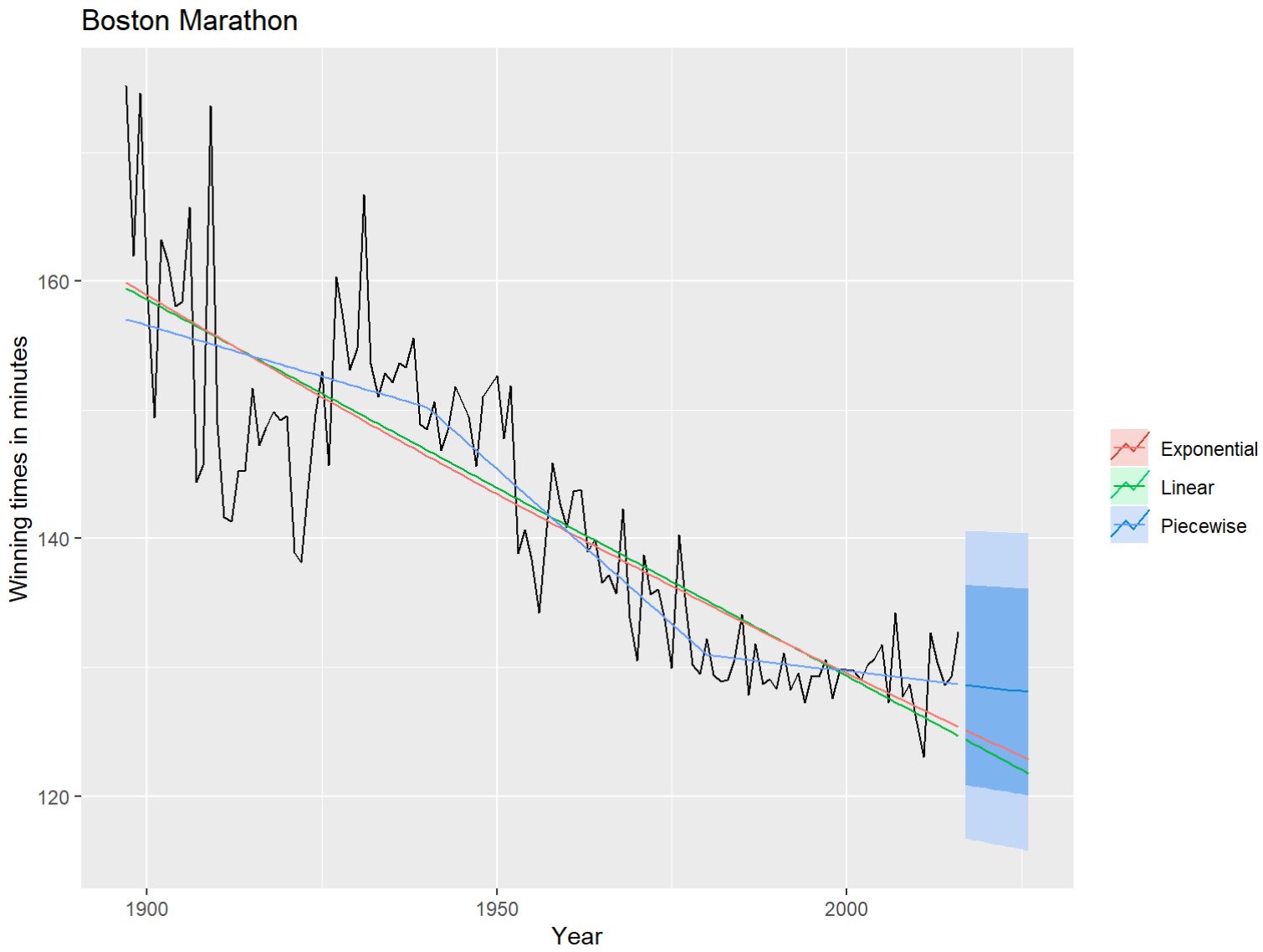
```
h <- 10
fit.lin <- tslm(marathon ~ trend)
fcasts.lin <- forecast(fit.lin, h=h)
fit.exp <- tslm(marathon ~ trend, lambda = 0)
fcasts.exp <- forecast(fit.exp, h=h)
```

```
t <- time(marathon)
t.break1 <- 1940
t.break2 <- 1980
t1 <- ts(pmax(0,t-t.break1), start=1897)
t2 <- ts(pmax(0,t-t.break2), start=1897)
```

```
fit.pw <- tslm(marathon ~ t + t1 + t2)
  t.new <- t[length(t)]+seq(h)
  t1.new <- t1[length(t1)]+seq(h)
  t2.new <- t2[length(t2)]+seq(h)
```

```
newdata <- cbind(t=t.new,t1=t1.new,t2=t2.new)%>%as.data.frame()
fcasts.pw <- forecast(fit.pw,newdata = newdata)
```

```
autoplot(marathon) +
  forecast::autolayer(fitted(fit.lin), series = "Linear") +
  forecast::autolayer(fitted(fit.exp), series="Exponential") +
  forecast::autolayer(fitted(fit.pw), series = "Piecewise") +
  forecast::autolayer(fcasts.pw, series="Piecewise") +
  forecast::autolayer(fcasts.lin$mean, series = "Linear") +
  forecast::autolayer(fcasts.exp$mean, series="Exponential") +
  xlab("Year") + ylab("Winning times in minutes") +
  ggtitle("Boston Marathon") +
  guides(colour=guide_legend(title=" "))
```

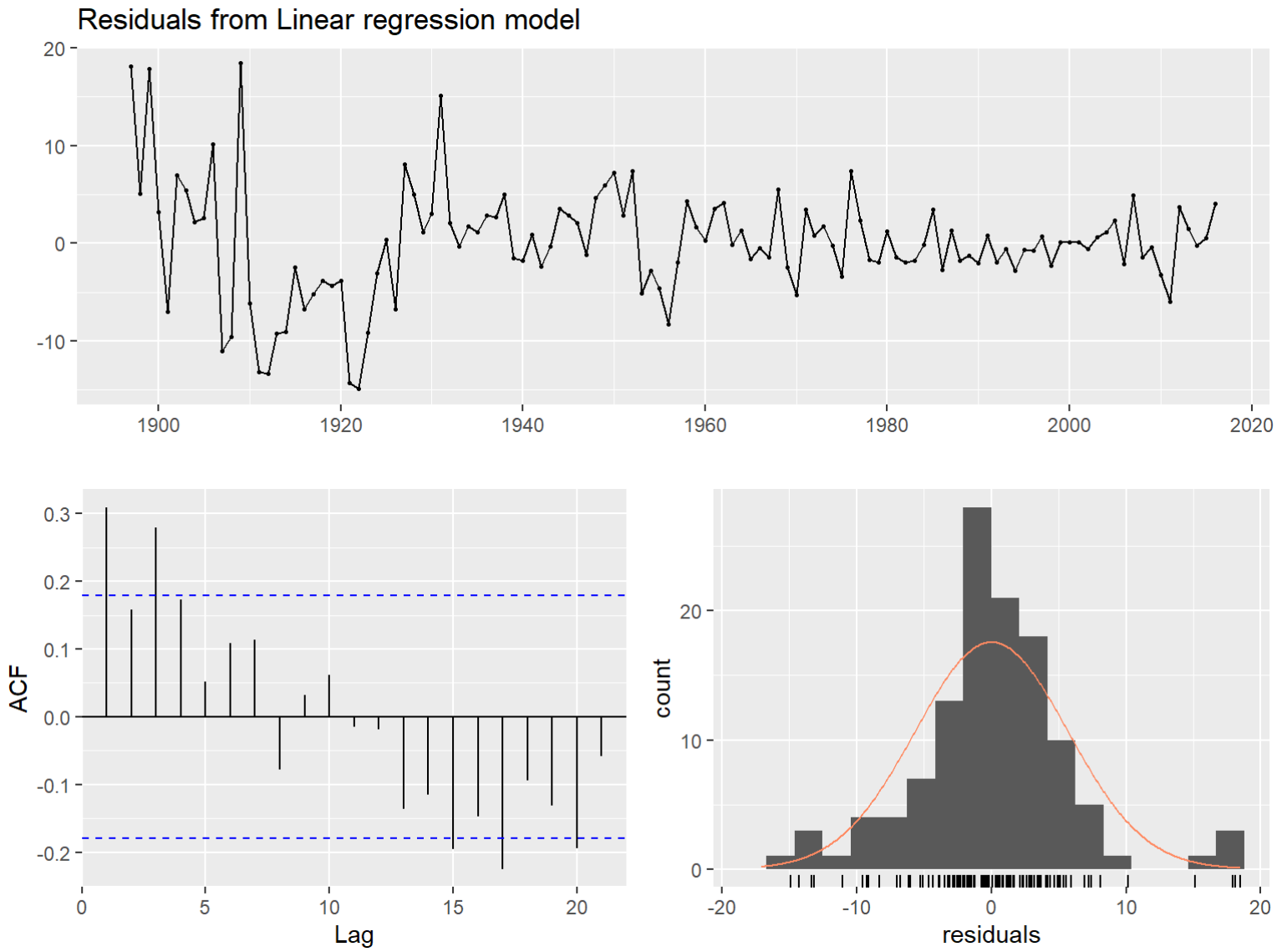

Boston Marathon
Winning times in minutes
160
140
120
1900
1950
2000
Year
Exponential
Linear
Piecewise

The piecewise linear trend fits the data better than both the exponential and linear trends and seems to return sensible forecasts. The wide prediction interval associated with the forecasts form the piecewise linear trend reflects the volatility observed in the historical winning times.

Note: The projection forward from the piecewise linear trend can be very sensitive to the choice of location of the knots, especially the ones located towards the end of the sample. For example moving the knot placed at 1980 even five years forward will reveal very different forecasts.

Residual diagnostics reveal some autocorrelation which will be dealt with in on line text in Chapter 8.

```
checkresiduals(fit.pw)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
```

```
## data:  Residuals from Linear regression model
## LM test = 21.597, df = 10, p-value = 0.0173
```


```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals from Linear regression model
## LM test = 21.597, df = 10, p-value = 0.0173
```

Correlation is not causation

A variable x may be useful for predicting a variable y, but that does not mean x is causing y.

For example, it is possible to model the number of drownings at a beach resort each month with the number of ice-creams sold in the same period. The two variables (ice-cream sales and drownings) are correlated (WHY?), but one is not causing the other. It is important to understand that correlations are useful for forecasting, even when there is no causal relationship between the two variables.

However, often a better model is possible if a causal mechanism can be determined.

Multicollinearity and forecasting

A closely related issue is multicollinearity which occurs when similar information is provided by two or more of the predictor variables in a multiple regression. It can occur in a number of ways.

- Two predictors are highly correlated with each other. In this case, knowing the value of one of the variables tells you a lot about the value of the other variable.
- A linear combination of predictors is highly correlated with another linear combination of predictors. In this case, knowing the value of the first group of predictors tells you a lot about the value of the second group of predictors.

When multicollinearity occurs in a multiple regression model, there are several consequences that you need to be aware of.

- If there is perfect correlation, it is not possible to estimate the regression model.
- If there is high correlation, then the estimation of the regression coefficients is computationally difficult.
- The uncertainty associated with individual regression coefficients will be large. This is because they are difficult to estimate. Consequently, statistical tests (e.g., t-tests) on regression coefficients are unreliable. Also, it will not be possible to make accurate statements about the contribution of each separate predictor to the forecast.
- Forecasts will be unreliable if the values of the future predictors are outside the range of the historical values of the predictors.

Otherwise, multicollinearity is not a problem for forecasting (i.e., it is used within the range of historical values of the predictors).

Homework 2

1. Exercise 1.8, Question 1 in the online text (https://otexts.com/fpp2/intro-exercises.html)

2. Exercise 5.10, Question 1 in the online text (https://otexts.com/fpp2/regression-exercises.html)

Due date: 25 February 2019 at 5 pm. Please submit in PDF in NTULearn under HW2. If you have any problems with the submission, please send me through e-mail.

Time series decomposition

Time series data can exhibit a huge variety of patterns and it is helpful to categorize some of the patterns and behaviours that can be seen in time series.

It is also sometimes useful to try to split a time series into several components, each representing one of the underlying pattern categories. Often this is done to help understand the time series better, but it can also be used to improve forecasts.

Time series patterns

We will refer to three types of time series patterns.

Trend

- A trend exists when there is a long-term increase or decrease in the data. It does not have to be linear. Sometimes we will refer to a trend changing direction when it might go from an increasing trend to a decreasing trend.

Seasonal

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

Cycle

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

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

Time series decomposition

Think of the time series y_t as consisting of three components: a seasonal component, a trend-cycle component (containing both trend and cycle), and a remainder component (containing anything else in the time series).

- Additive model

$$y_t = S_t + T_t + R_t,$$

where y_t is the data at period t, S_t is the seasonal component, T_t is the trend-cycle component and R_t is the remainder (or irregular or error) component at period t.

- Multiplicative model

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

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.

An alternative to using a multiplicative model, is to first transform the data until the variation in the series appears to be stable over time, and then use an additive model. When a log transformation has been used, this is equivalent to using a multiplicative decomposition because

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

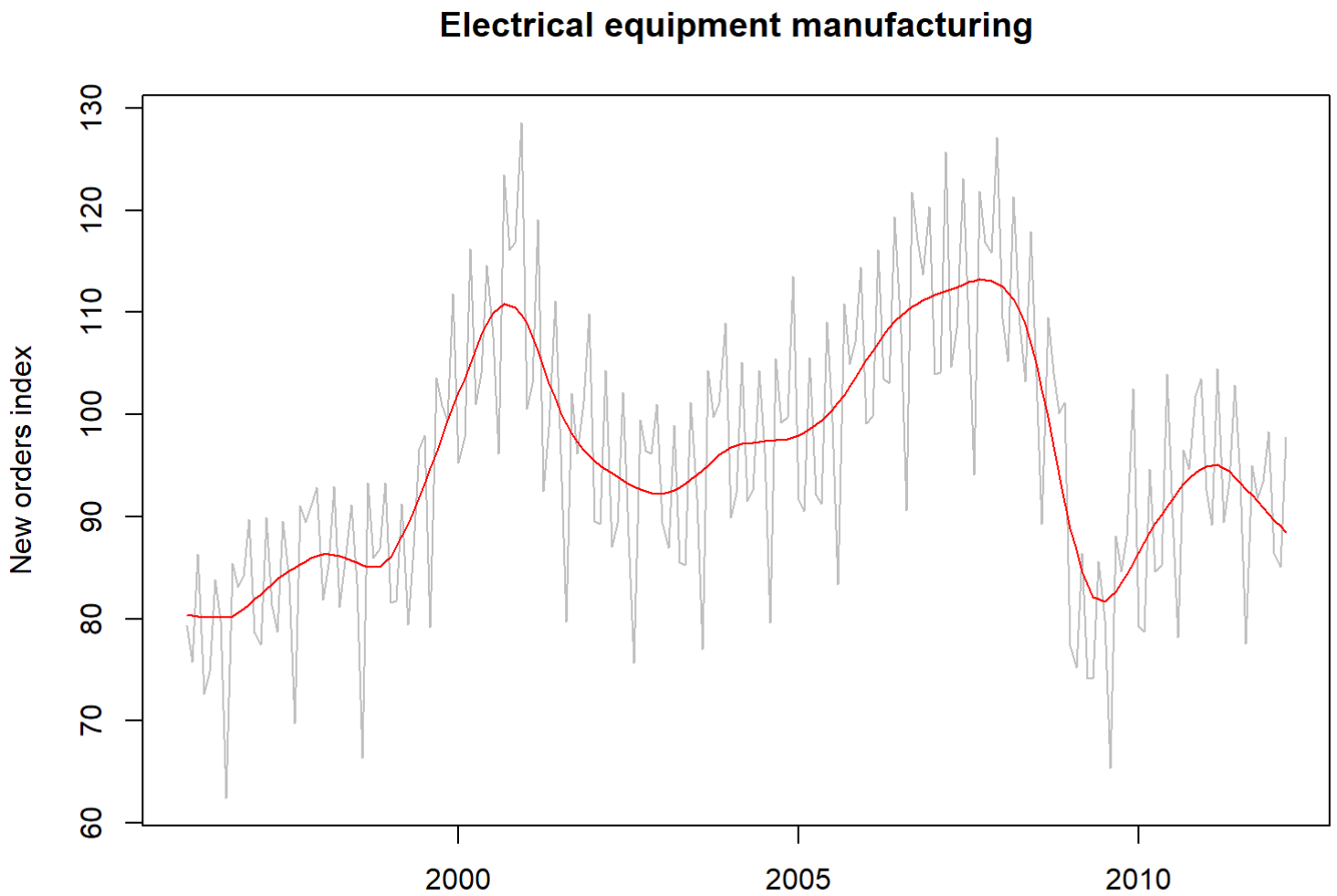
is equivalent to

$$\log y_t = \log S_t + \log T_t + \log R_t.$$

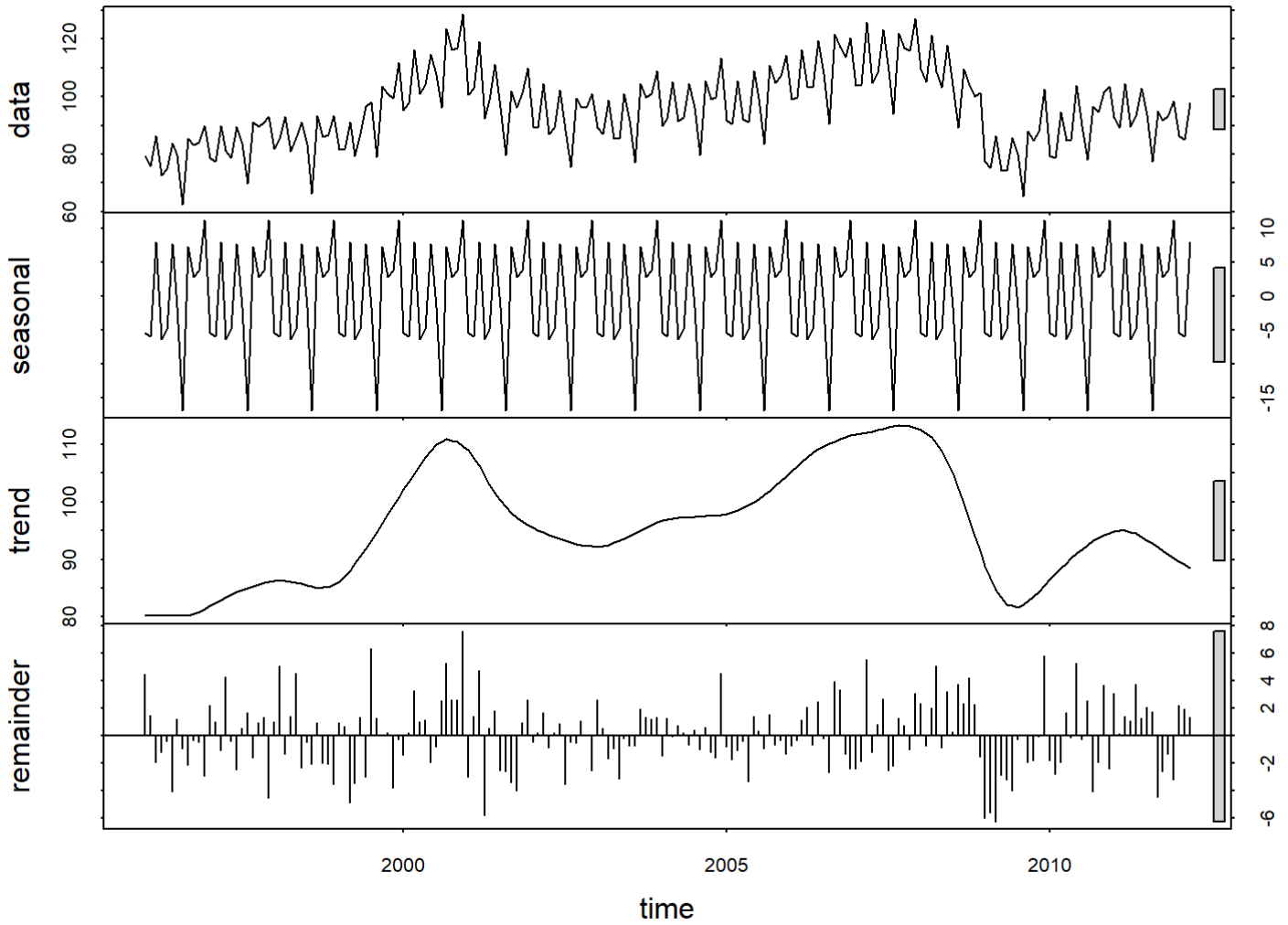
Sometimes, the trend-cycle component is simply called the trend component, even though it may contain cyclic behaviour as well.

Example

```
fit <- stl(elecequip, s.window="periodic")
plot(elecequip, col="gray", main="Electrical equipment manufacturing",
     ylab="New orders index", xlab="")
lines(trendcycle(fit),col="red", ylab="Trend-Cycle")
```


Electrical equipment manufacturing
New orders index
60
70
80
90
100
110
120
130
2000
2005
2010

```
plot(fit)
```


data
seasonal
trend
remainder
60
80
100
120
-15
-5
0
5
10
80
90
100
110
-6
-2
2
4
6
8
2000
2005
2010
time

The three components are shown separately in the bottom 3 panels of previous graph.

The grey bars to the right of each panel show the relative scales of the components. Each grey bar represents the same length but because the plots are on different scales, the bars vary in size.The large grey bar in the bottom panel shows that the variation in the remainder component is small compared to the variation in the data, which has a bar about one quarter the size.

Seasonally adjusted data

If the seasonal component is removed from the original data, the resulting values are called the seasonally adjusted data. For an additive model, the seasonally adjusted data are given by $y_t - S_t$, and for multiplicative data, the seasonally adjusted values are obtained using y_t / S_t.

Seasonally adjusted series contain the remainder component as well as the trend-cycle. Therefore they are not smooth and downturns or upturns can be misleading.

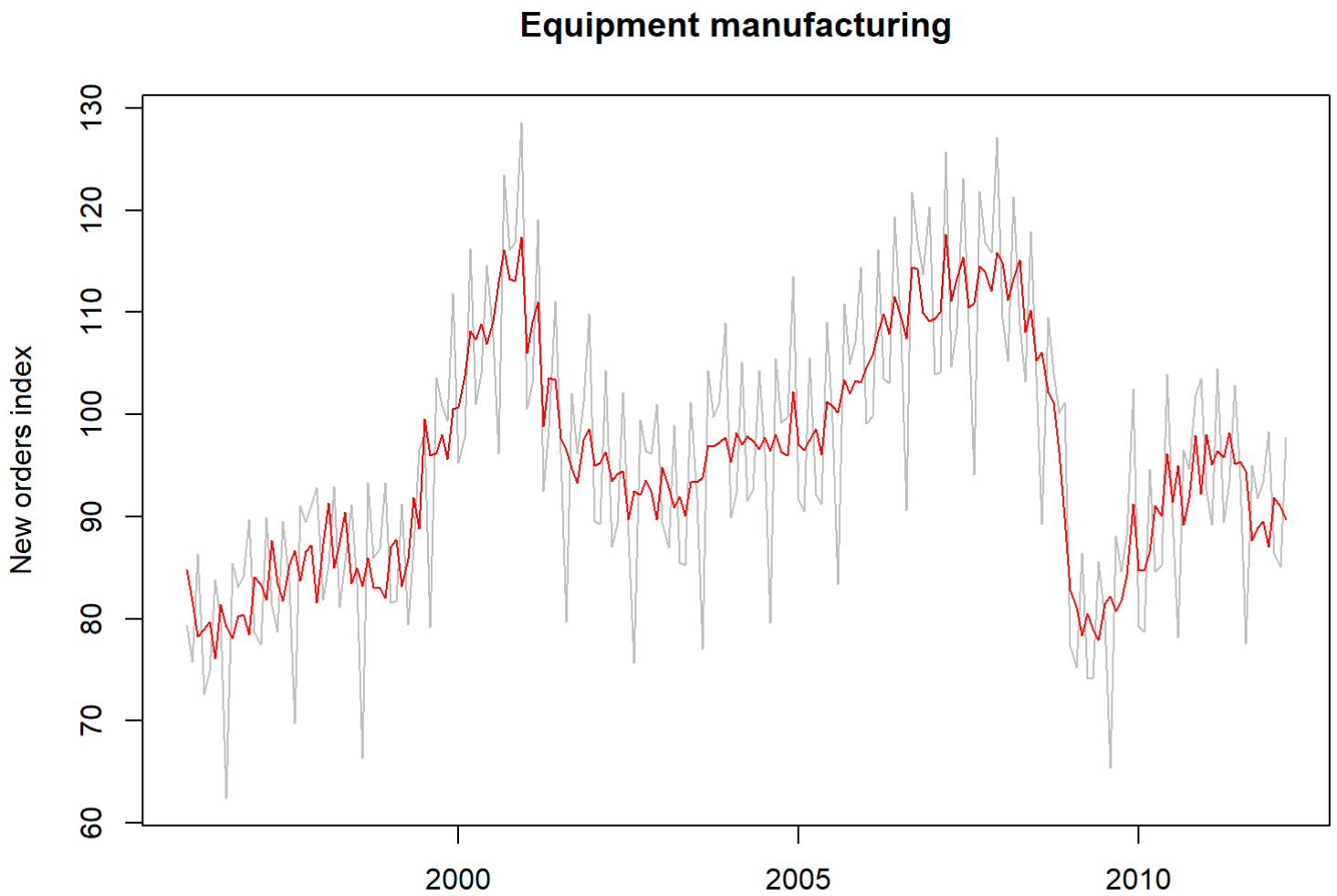
If the purpose is to look for turning points in the series, and interpret any changes in the series, then it is better to use the trend-cycle component rather than the seasonally adjusted data.

E.g., Unemployment data are usually seasonally adjusted to highlight the underlying state of the economy rather than seasonal variation.

Plot of seasonally adjusted data on electrical equipment manufacturing is shown in the next slide:

The seasonally adjusted series contain the remainder component and the trend-cycle component.

```
plot(elecequip, col="grey",
 main="Equipment manufacturing", xlab="", ylab="New orders index")
lines(seasadj(fit),col="red",ylab="Seasonally adjusted")
```



Moving averages

The classical method of time series decomposition originated in the 1920s. It still forms the basis of time series methods, and so it is important to understand how it works. The first step in a classical decomposition is to use a moving average method to estimate the trend-cycle.

A moving average of order m can be written as

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^{k} y_{t+j},$$

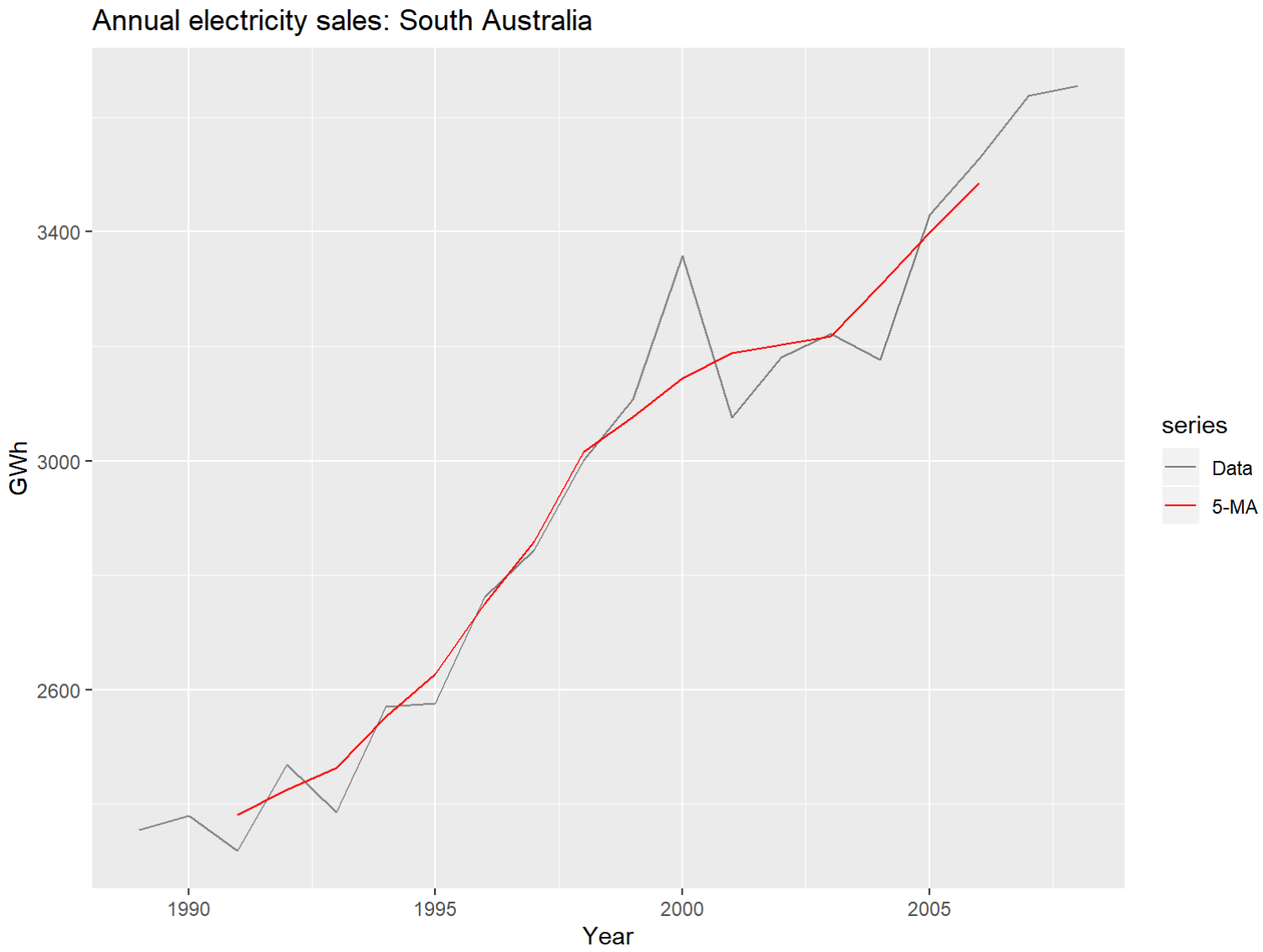
where $m = 2k + 1$. At time t it is obtained by averaging values of the time series within k periods of t. We call this an m-MA meaning a moving average of order m. The moving average eliminates some of the randomness, giving a smooth result.

```
data.vec <- round(rnorm(10)*100)
table.out <- rbind(data.vec, ma(data.vec, order=5))
rownames(table.out) <- c("data", "5-MA")
table.out
```

```
##      [,1] [,2]  [,3]  [,4] [,5]  [,6]  [,7] [,8] [,9] [,10]
## data  -44  -33 167.0 -70.0 38.0 170.0 128.0  141   81   -40
## 5-MA   NA   NA  11.6  54.4 86.6  81.4 111.6   96   NA    NA
```

Each value in the 5-MA row is the average of the observations in the five year period centered on the corresponding year. There are no values for the first two years or last two years because we don't have two observations on either side.

```
autoplot(elecsales, series="Data") +
  forecast::autolayer(ma(elecsales,5), series="5-MA") +
  xlab("Year") + ylab("GWh") +
  ggtitle("Annual electricity sales: South Australia") +
  scale_colour_manual(values=c("Data"="grey50","5-MA"="red"),
                      breaks=c("Data","5-MA"))
```

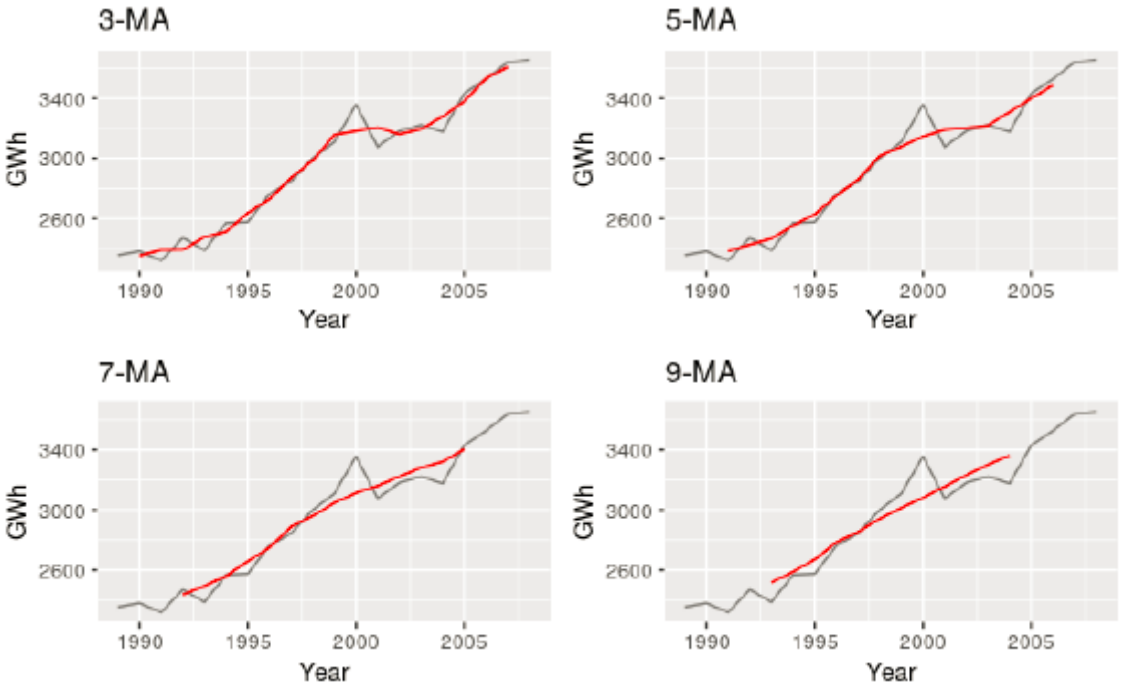

Annual electricity sales: South Australia
GWh
3400
3000
2600
1990
1995
2000
2005
Year
series
Data
5-MA

The data are shown in the next slide use the following commands (change rbind to cbind and rownames to colnames if you wish to reproduce Table 6.1):

```
ma5 <- ma(elecsales, 5)
table.out <- rbind(elecsales, ma5)
rownames(table.out) <- c("sales(GWh)", "5-MA")
table.out
```

```
##              [,1]    [,2]    [,3]     [,4]     [,5]     [,6]    [,7]
## sales(GWh) 2354.34 2379.71 2318.52 2468.990 2386.090 2569.470 2575.72
## 5-MA           NA      NA 2381.53 2424.556 2463.758 2552.598 2627.70
##               [,8]     [,9]    [,10]  [,11]   [,12]  [,13]   [,14]
## sales(GWh) 2762.720 2844.500 3000.700 3108.1 3357.50 3075.7 3180.60
## 5-MA       2750.622 2858.348 3014.704 3077.3 3144.52 3188.7 3202.32
##              [,15]    [,16]    [,17]    [,18]   [,19] [,20]
## sales(GWh) 3221.60 3176.200 3430.600 3527.480 3637.89  3655
## 5-MA       3216.94 3307.296 3398.754 3485.434      NA    NA
```

The order of the moving average determines the smoothness of the trend-cycle estimate. In general, a larger order means a smoother curve.



Effect of the order of the moving average on smoothness

Note: Simple moving averages such as these are usually of an odd order (e.g., 3, 5, 7, etc.) This is so they are symmetric: in a moving average of order $m = 2k + 1$, there are earlier observations, later observations and the middle observation that are averaged. But if it was even, it would no longer be symmetric.

Moving averages of moving averages

It is possible to apply a moving average to a moving average. One reason for doing this is to make an even-order moving average symmetric. `

For example, we might take a moving average of order 4, and then apply another moving average of order 2 to the results.

```
beer2 <- window(ausbeer,start=1992)
ma4 <- ma(beer2, order=4, centre=FALSE)# this cmd gives non-centred ma4
ma2x4 <- ma(beer2, order=4, centre=TRUE)# default. gives ma2X4
```

Looking at the data:

```
table.out <- cbind(beer2, ma4, ma2x4)
colnames(table.out) <- c("data", "4-MA", "2x4-MA")
table.out
```

```
##         data   4-MA  2x4-MA
## 1992 Q1  443     NA      NA
## 1992 Q2  410 451.25      NA
## 1992 Q3  420 448.75 450.000
## 1992 Q4  532 451.50 450.125
## 1993 Q1  433 449.00 450.250
## 1993 Q2  421 444.00 446.500
## 1993 Q3  410 448.00 446.000
## 1993 Q4  512 438.00 443.000
## 1994 Q1  449 441.25 439.625
## 1994 Q2  381 446.00 443.625
## 1994 Q3  423 440.25 443.125
## 1994 Q4  531 447.00 443.625
## 1995 Q1  426 445.25 446.125
## 1995 Q2  408 442.50 443.875
## 1995 Q3  416 438.25 440.375
## 1995 Q4  520 435.75 437.000
## 1996 Q1  409 431.25 433.500
## 1996 Q2  398 428.00 429.625
## 1996 Q3  398 433.75 430.875
## 1996 Q4  507 433.75 433.750
## 1997 Q1  432 435.75 434.750
## 1997 Q2  398 440.50 438.125
## 1997 Q3  406 439.50 440.000
## 1997 Q4  526 439.25 439.375
## 1998 Q1  428 438.50 438.875
## 1998 Q2  397 436.25 437.375
## 1998 Q3  403 438.00 437.125
## 1998 Q4  517 434.50 436.250
## 1999 Q1  435 439.75 437.125
## 1999 Q2  383 440.75 440.250
## 1999 Q3  424 437.25 439.000
## 1999 Q4  521 442.00 439.625
## 2000 Q1  421 439.50 440.750
## 2000 Q2  402 434.25 436.875
## 2000 Q3  414 441.75 438.000
## 2000 Q4  500 436.25 439.000
## 2001 Q1  451 436.75 436.500
## 2001 Q2  380 434.75 435.750
## 2001 Q3  416 429.00 431.875
## 2001 Q4  492 436.00 432.500
```

```
## 2002 Q1  428 433.50 434.750
## 2002 Q2  408 437.00 435.250
## 2002 Q3  406 438.75 437.875
## 2002 Q4  506 431.75 435.250
## 2003 Q1  435 435.50 433.625
## 2003 Q2  380 431.50 433.500
## 2003 Q3  421 431.50 431.500
## 2003 Q4  490 434.00 432.750
## 2004 Q1  435 431.75 432.875
## 2004 Q2  390 422.75 427.250
## 2004 Q3  412 418.00 420.375
## 2004 Q4  454 421.25 419.625
## 2005 Q1  416 420.25 420.750
## 2005 Q2  403 427.25 423.750
## 2005 Q3  408 432.75 430.000
## 2005 Q4  482 428.50 430.625
## 2006 Q1  438 427.75 428.125
## 2006 Q2  386 430.00 428.875
## 2006 Q3  405 427.25 428.625
## 2006 Q4  491 426.50 426.875
## 2007 Q1  427 423.75 425.125
## 2007 Q2  383 419.25 421.500
## 2007 Q3  394 417.50 418.375
## 2007 Q4  473 419.25 418.375
## 2008 Q1  420 423.25 421.250
## 2008 Q2  390 427.00 425.125
## 2008 Q3  410 425.75 426.375
## 2008 Q4  488 427.75 426.750
## 2009 Q1  415 430.00 428.875
## 2009 Q2  398 430.00 430.000
## 2009 Q3  419 429.75 429.875
## 2009 Q4  488 423.75 426.750
## 2010 Q1  414     NA      NA
## 2010 Q2  374     NA      NA
```

The notation 2×4-MA in the last column means a 4-MA followed by a 2-MA.

The values in the last column are obtained by taking a moving average of order 2 of the values in the previous column. For example, the first two values in the 4-MA row are 451.2= (443+410+420+532)/4 and 448.8=(410+420+532+433)/4. The first value in the 2×4-MA row is the average of these two: 450.0=(451.2+448.8)/2.

When a 2-MA follows a moving average of even order (such as 4), it is called a centered moving average of order 4. This is because the results are now symmetric. (note that this is implemented automatically in ma() by using the option centre=true). This is the default if option centre is omitted.

To see that this is the case, we can write the 2×4-MA as follows:

$$\begin{aligned}\hat{T}_t &= \frac{1}{2}\Big[\frac{1}{4}(y_{t-2} + y_{t-1} + y_t + y_{t+1}) \\ &\qquad + \frac{1}{4}(y_{t-1} + y_t + y_{t+1} + y_{t+2})\Big] \\ &= \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2}.\end{aligned}$$

It is now a weighted average of observations, but it is symmetric.

In general, an even order MA should be followed by an even order MA to make it symmetric. Similarly, an odd order MA should be followed by an odd order MA (e.g. 3×3-MA).

Estimating the trend-cycle with seasonal data

The most common use of centered moving averages is in estimating the trend-cycle from seasonal data. 2×4-MA:

$$\hat{T}_t = \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2}.$$

When applied to quarterly data, each quarter of the year is given equal weight as the first and last terms apply to the same quarter in consecutive years.

Consequently, the seasonal variation will be averaged out and the resulting values of $\hat{T}_t$ will have little or no seasonal variation remaining. A similar effect would be obtained using a 2×8-MA or a 2×12-MA.

In general, a $2 \times m$-MA is equivalent to a weighted moving average of order $m + 1$ with all observations taking weight $1/m$ except for the first and last terms which take weights $1/(2m)$ (more on this later).

So if the seasonal period is even and of order m, use a $2 \times m$-MA to estimate the trend-cycle. If the seasonal period is odd and of order m, use a m-MA to estimate the trend cycle.

For monthly data, we have an even-order MA so we centre it by taking 2×12-MA which could be used to estimate the trend-cycle of monthly data

For daily data, we have an odd-order 7-MA which could be used to estimate the trend-cycle of daily data.

Other choices for the order of the MA will usually result in trend-cycle estimates being contaminated by the seasonality in the data.

```
autoplot(elecequip, series="Data") +
  autolayer(ma(elecequip, 12, centre = FALSE), 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"))
```

Trend-cycle is contaminated by some seasonality

Weighted moving averages

Combinations of moving averages result in weighted moving averages. For example, the 2×4-MA discussed above is equivalent to a weighted 5-MA with weights given by $[\frac{1}{8}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}]$. In general, a weighted m-MA can be written as

$$\hat{T}_t = \sum_{j=-k}^{k} a_j y_{t+j},$$

where $k = (m-1)/2$ and the weights are given by $[a_{-k}, \dots, a_k]$. It is important that the weights all sum to one and that they are symmetric so that $a_j = a_{-j}$.

The simple m-MA is a special case where all the weights are equal to $1/m$. A major advantage of weighted moving averages is that they yield a smoother estimate of the trend-cycle. Instead of observations entering and leaving the calculation at full weight, their weights are slowly increased and then slowly decreased resulting in a smoother curve.

Name	a0	a1	a2	a3	a4	a5	a6	a7	a8
3-MA	.333	.333							
5 MA	.200	.200	.200						
2x12-MA	.083	.083	.083	.083	.083	.083	.042		
3x3-MA	.333	.222	.111						
3x5-MA	.200	.200	.133	.067					

The weighted moving average above is a linear filter that converts the observed time series into an estimate of the trend $\hat{T}_t$ through a linear operation (see above formula).

Other filters such as the Spencer weighted moving average and Henderson weighted moving average that leave higher order polynomials untouched (i.e. Henderson W MA preserves the cubic polynomial while Spencer WMA preserves the quadratic polynomial). These filters are typically incorporated in the seasonal adjustment packages available in R (eg., Henderson's WMA is used in the X-11 family of seasonal packages to be discussed later in this lecture). More on this next week ..

Thank you for your attention