Welcome to General Assembly

- **WiFi** GA Guest
- **Password** yellowpencil

GENERAL ASSEMBLY

DATA SCIENCE

10 WEEK PART TIME COURSE

Lesson 14:
- Introduction to R
- Introduction to Time Series

Course Plan

UNITS

UNIT 1: FOUNDATIONS OF DATA MODELING	Introduction to Data Science	Lesson 1
	Elements of Data Science	Lesson 2
	Data Visualisation	Lesson 3
	Linear Regression	Lesson 4
	Logistic Regression	Lesson 5
	Model Evaluation	Lesson 6
	Regularisation	Lesson 7
	Clustering	Lesson 8
UNIT 2: DATA SCIENCE IN THE REAL WORLD	Recommendations	Lesson 9
	SQL + Productivity	Lesson 10
	Decision Trees	Lesson 11
	Ensembles	Lesson 12
	Natural Language Programming	Lesson 13
Final Countdown!	Cloud Computing	Lesson 14
	Time Series	Lesson 15
	Soft Skills	Lesson 16
	Network Analysis	Lesson 17
	Neural Networks	Lesson 18
	Final Projects Presentations	Lesson 19
	Final Projects Presentations	Lesson 20

Git & GitHub – 1 Pager Guide!

(Part B) EVERY CLASS:

At the START of the class, you'll need to sync the latest materials from the COURSE repo:

(1) Make sure you are in the dat11syd directory:
```
cd ~/workspace/dat11syd
```
(2) Make sure to select the "master" branch of your repo:
```
git checkout master
```
(3) Fetch the latest changes from the UPSTREAM repo (i.e the course repo)
```
git fetch upstream
```
(4) Merge the changes from the upstream repo to your master branch:
```
git merge upstream/master
```

DURING the class:

(5) Before editing, either copy files to your "students/" folder, or rename them

At the END of every class:

(6) Make sure you are in the dat11syd directory:
```
cd ~/workspace/dat11syd
```
(7) Add any files that you've updated to your git registry:
```
git add -A
```
(8) Commit the changes with a sensible comment:
```
git commit -m "my updates for lesson 7"
```
(9) Push your changes to your PERSONAL repo:
```
git push origin master
```

DONE!!!!!

AGENDA

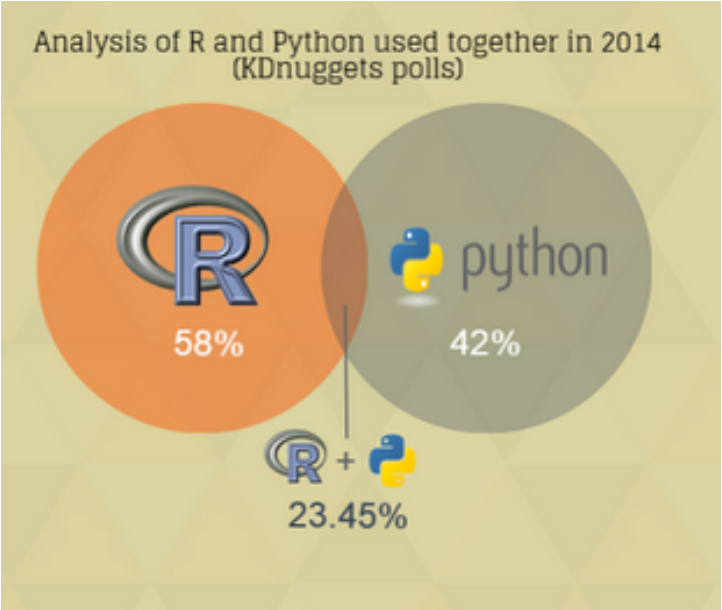
1. R – installing & setup
2. R – introduction
3. Time Series Introduction
4. Lab
5. Discussion – Final Projects

DATA SCIENCE PART TIME COURSE

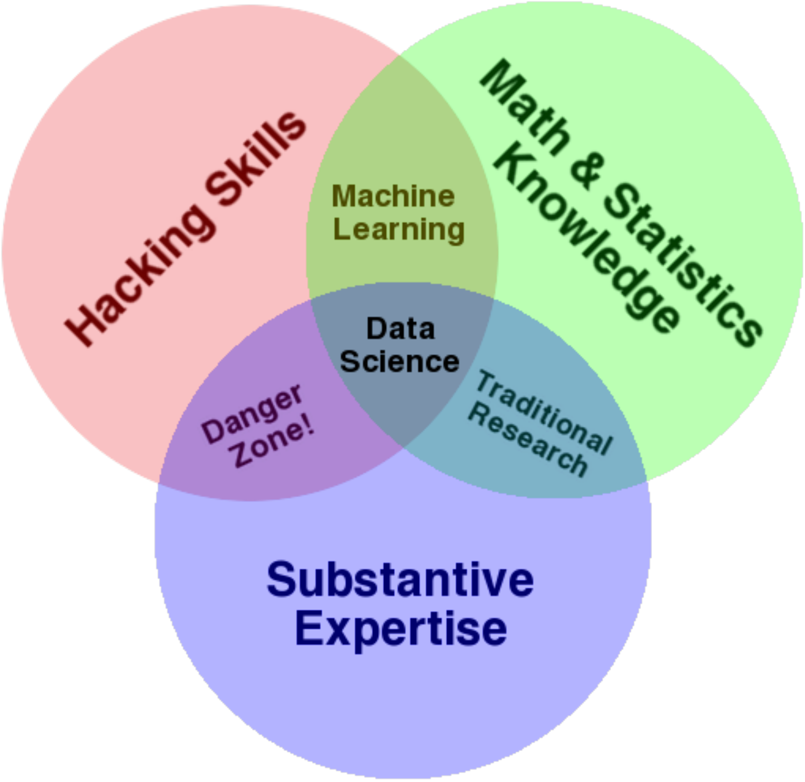
PYTHON & R

Language Rank		Types	Spectrum Ranking
1.	Java		100.0
2.	C		99.9
3.	C++		99.4
4.	Python		96.5
5.	C#		91.3
6.	R		84.8
7.	PHP		84.5
8.	JavaScript		83.0
9.	Ruby		76.2
10.	Matlab		72.4

IEEE Spectrum Survey 2015



DataCamp Infographic
2015


Hacking Skills
Math & Statistics Knowledge
Machine Learning
Data Science
Danger Zone!
Traditional Research
Substantive Expertise

To-Do List

1. Getting started with R:

Option 1 – Use R-Studio on the server: http://paulgoodall.tech/rstudio/

Option 2 – Download & Install R and R-Studio (15mins)

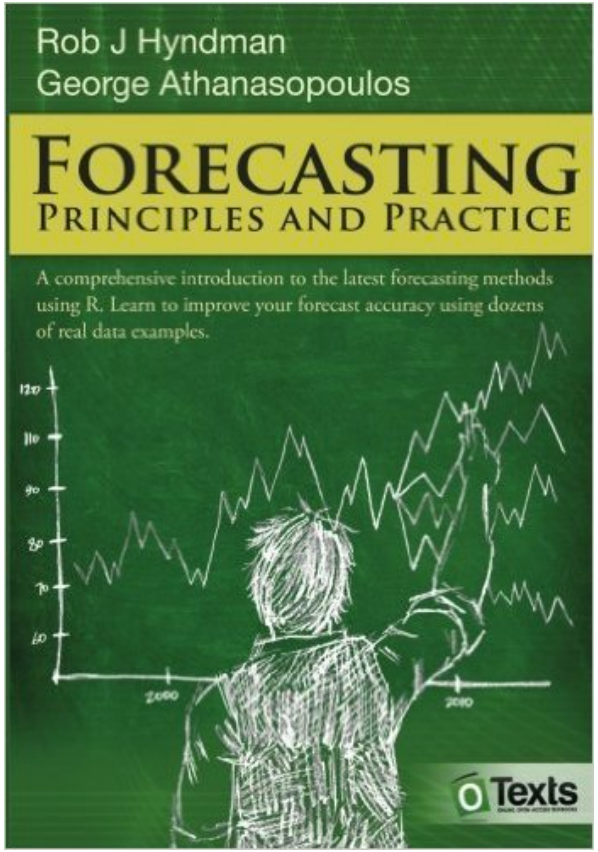
2. SWIRL intro to R: (includes install instructions – can skip to step 4)

http://swirlstats.com/students.html

Go through the courses: [**R Programming**] [**Getting and Cleaning Data**]

3. Read first 2 chapters of Forecasting Principles and Practice

https://www.otexts.org/fpp **(20 mins)**


Rob J Hyndman
George Athanasopoulos
FORECASTING
PRINCIPLES AND PRACTICE
A comprehensive introduction to the latest forecasting methods using R. Learn to improve your forecast accuracy using dozens of real data examples.
OTexts

DATA SCIENCE PART TIME COURSE

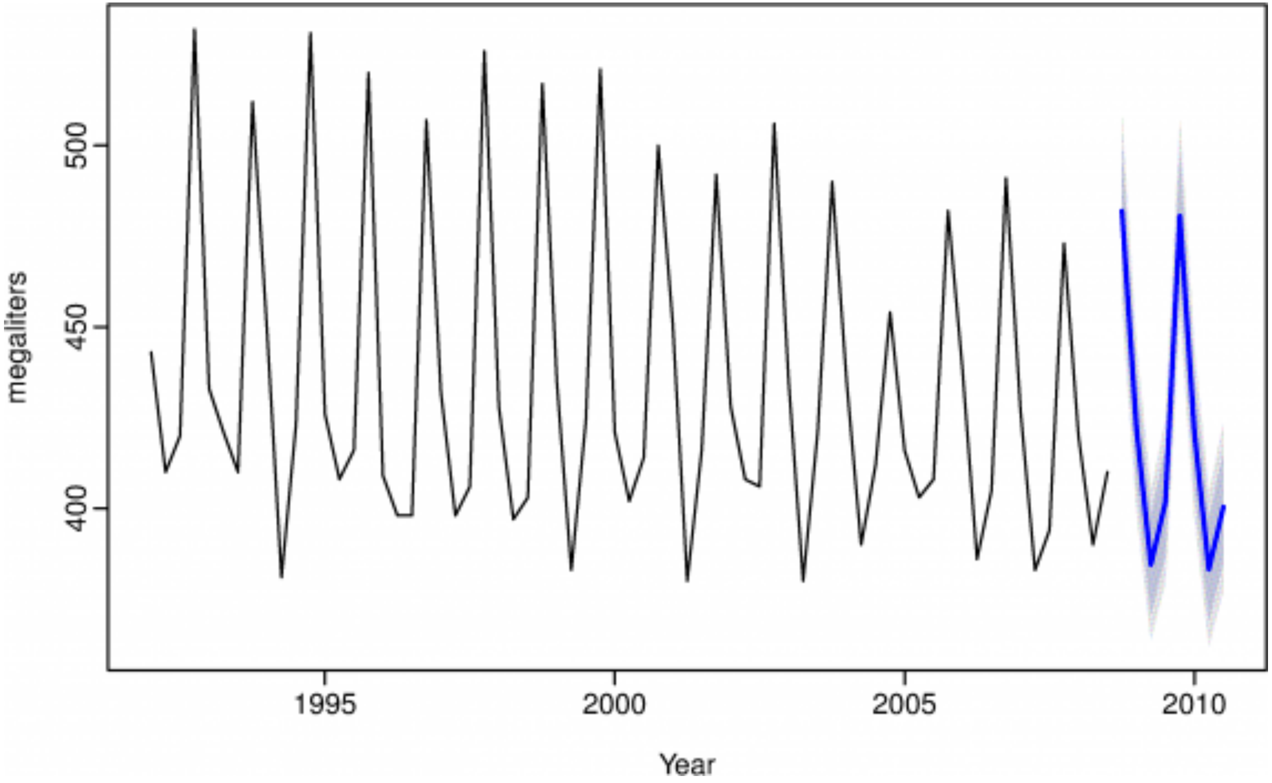
WHAT IS A TIME SERIES?

WHAT IS A TIME SERIES?

A time series is a series of data that is observed sequentially over time.

Examples include:

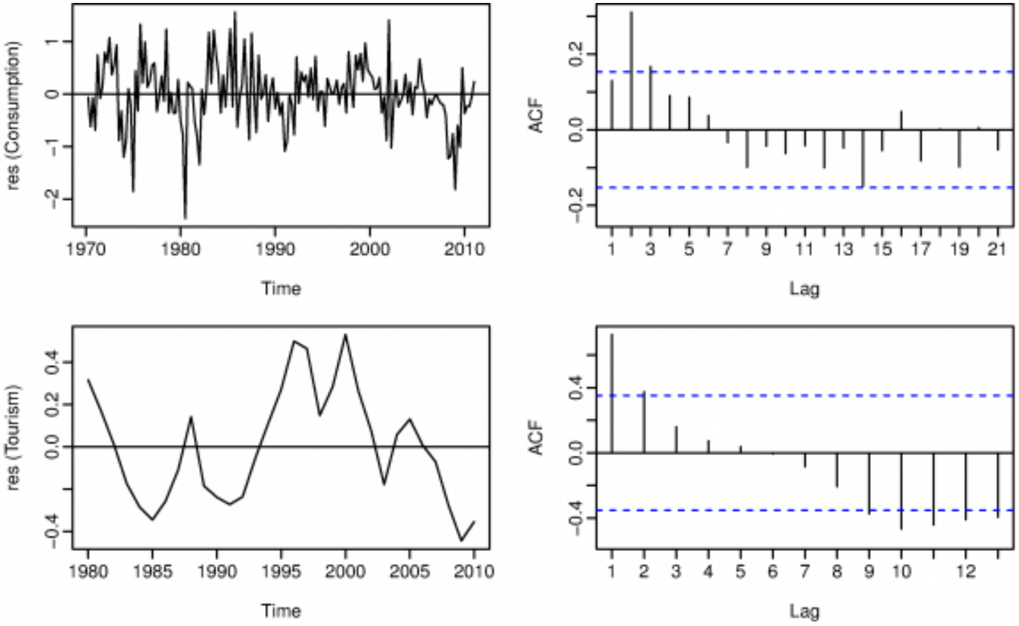
- **Weekly Rainfall**
- **Daily Stock price of Atlassian**
- **Quarterly oil import figures**


megaliters
500
450
400
1995
2000
2005
2010
Year

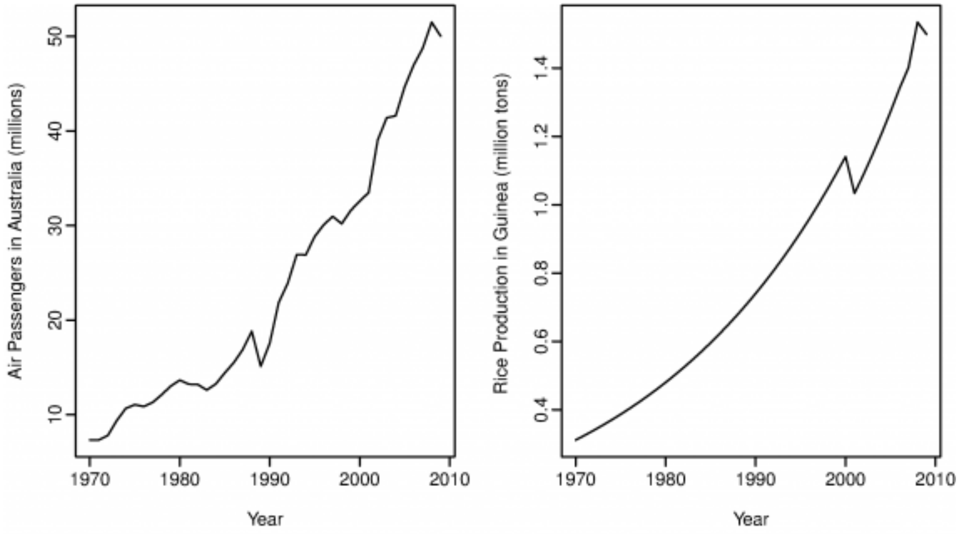
In other words, why wouldn't we just use linear regression and have the time variable as our X values?

$$y=\beta_0+\beta_1 x+\varepsilon.$$

With time series data it is highly likely that the value of a variable observed in the current time period will be influenced by its value in the previous period, or even the period before that, and so on…



Time Series will often be 'non-stationary', that means that you do not have a zero mean or constant variance. This can lead to spurious regressors (factors considered influential in a model that are actually not).



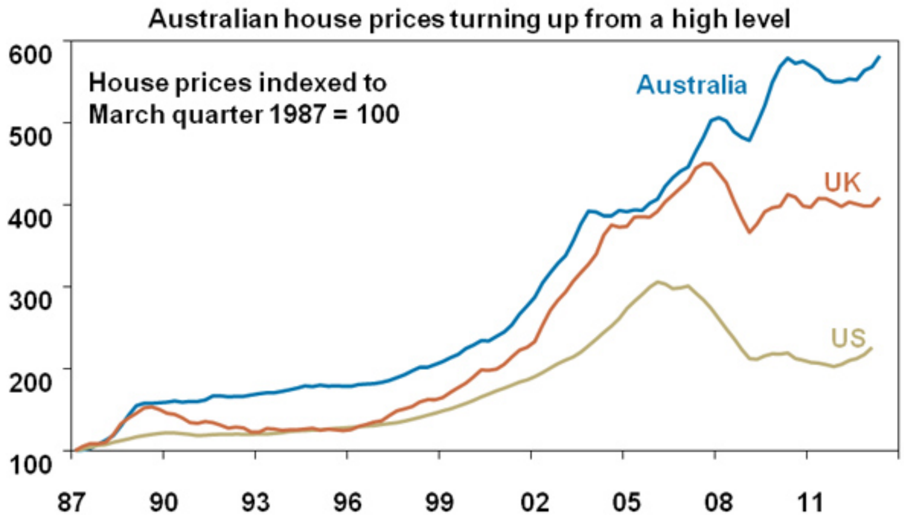
There are three time series components we will use to describe a time series

- **Trend**
- **Seasonal**
- **Cyclic**

TIME SERIES COMPONENTS - 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.

Example:

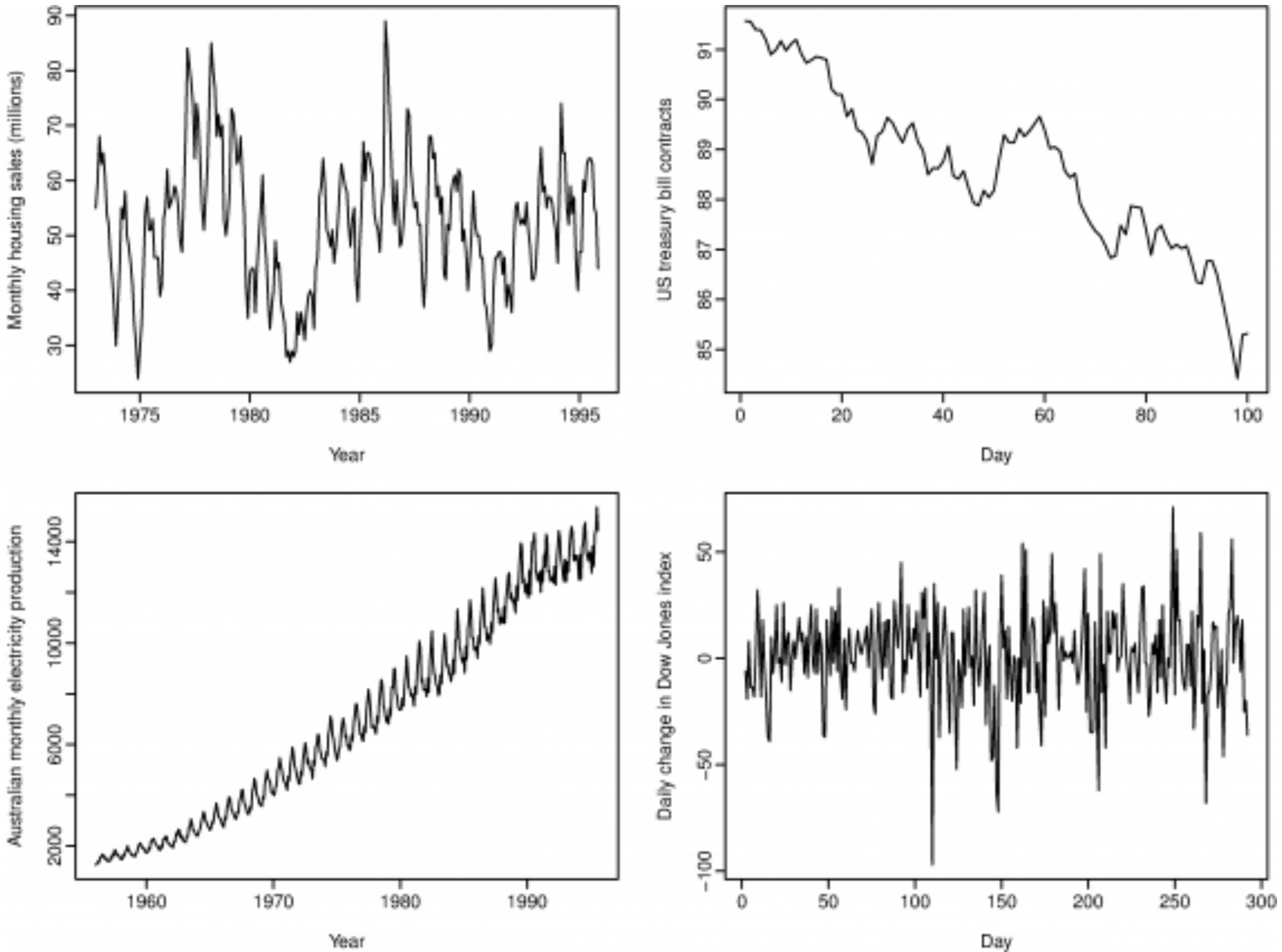


TIME SERIES COMPONENTS - SEASONAL

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.

TIME SERIES COMPONENTS - CYCLIC

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.



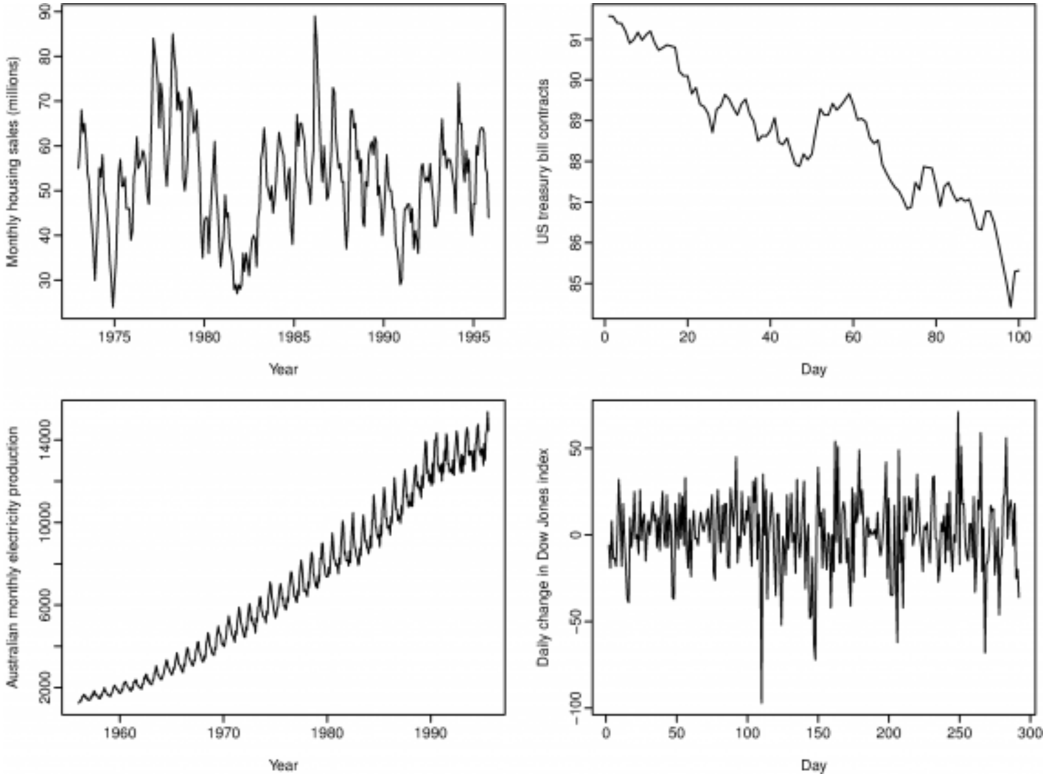
Monthly housing sales (millions)
Year
US treasury bill contracts
Day
Australian monthly electricity production
Year
Daily change in Dow Jones index
Day

Top Left: strong seasonality within each year, as well as some strong cyclic behaviour with period about 6–10 years. No Trend

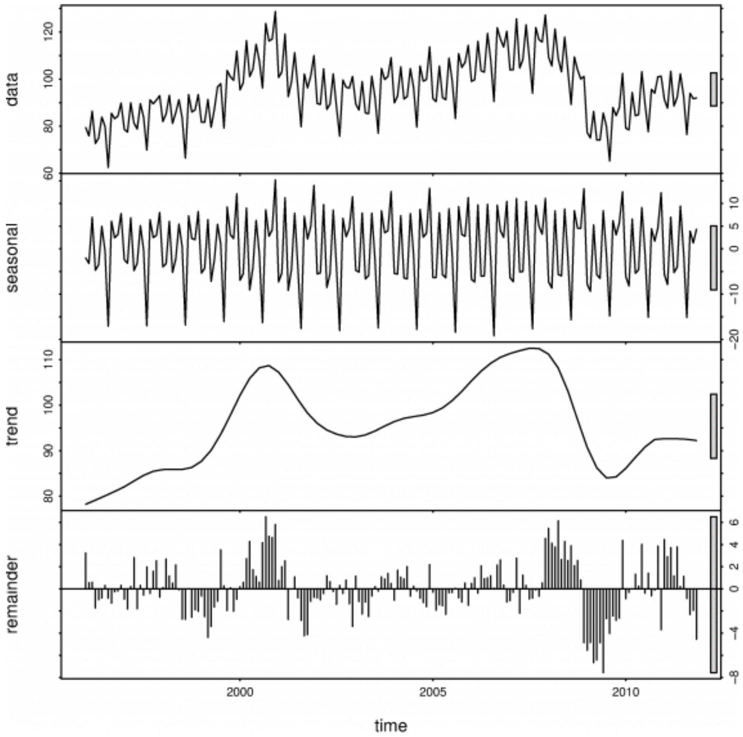
Top Right: no seasonality, but an obvious downward trend. If we had more data we may be able to observe a cycle

Bottom Left: strong increasing trend, with strong seasonality. No cycle

Bottom Right: no trend, seasonality or cyclic behaviour



Time Series Decomposition is a way to break down a time series into the Season, Trend (which includes the cycle) and Remainder.



We can consider these to be weighted averages of past observations. This means that the more recent the observation, the higher the weighting of that observation.

The Naive model is the case where the forecast is equal to the last observed value,

$$Y_{t+1} = Y_t$$

What if we were to weight the observations to have decreasing weights as the observations got older? What would the equation look like?

$$\hat{y}_{t+1 \mid t} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots$$

where $0 \leq \alpha \leq 1$ is the smoothing parameter.

The Holt-Winters method captures level, trend and seasonality.

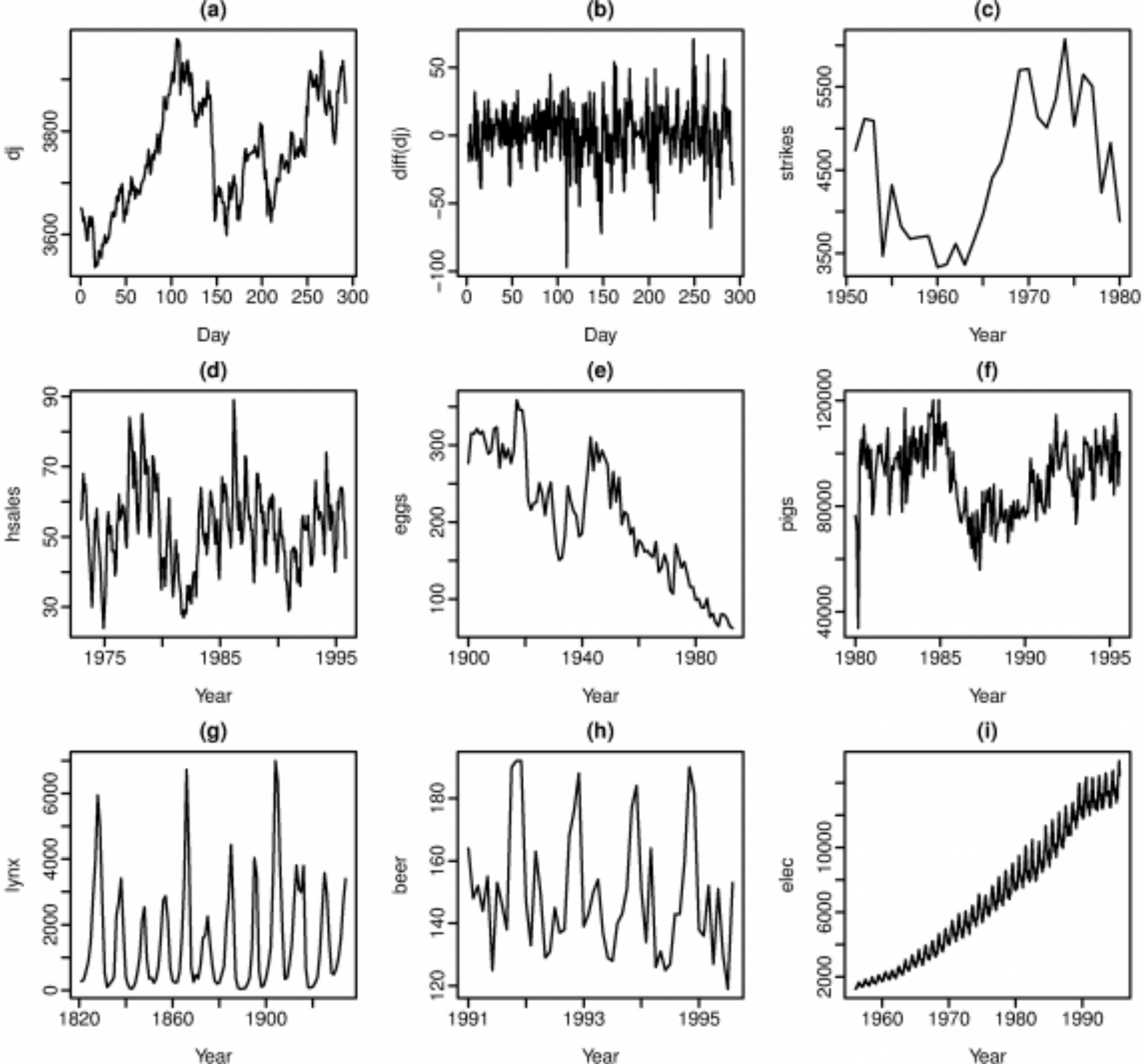
There are two variations to this method that differ in the nature of the seasonal component. The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series.

$$\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t-m+h_m^+} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},
\end{aligned}$$

$$\begin{aligned}
\hat{y}_{t+h|t} &= (\ell_t + hb_t)s_{t-m+h_m^+}. \\
\ell_t &= \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma\frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}
\end{aligned}$$

A stationary time series is one whose properties do not depend on the time at which the series is observed. So 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. On the other hand, a white noise series is stationary — it does not matter when you observe it, it should look much the same at any period of time.

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.



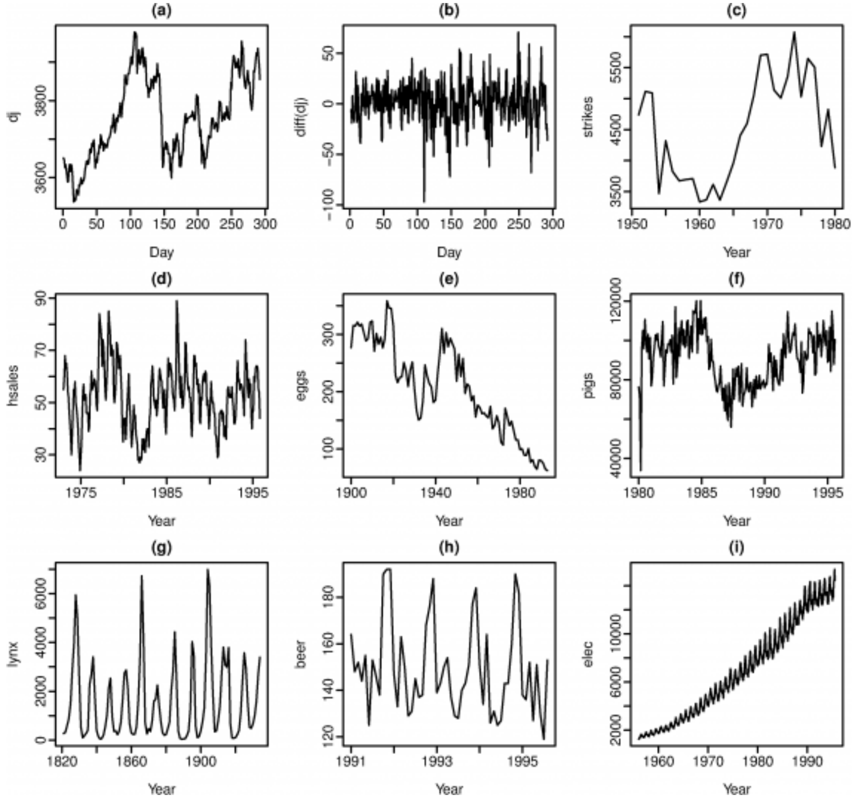
(a)
(b)
(c)
dj
3600
3800
0 50 100 150 200 250 300
Day
diff(dj)
-100
-50
0
50
0 50 100 150 200 250 300
Day
strikes
3500
4500
5500
1950 1960 1970 1980
Year
(d)
(e)
(f)
hsales
30
50
70
90
1975 1985 1995
Year
eggs
100
200
300
1900 1940 1980
Year
pigs
40000
80000
120000
1980 1985 1990 1995
Year
(g)
(h)
(i)
lynx
0
2000
4000
6000
1820 1860 1900
Year
beer
120
140
160
180
1991 1993 1995
Year
elec
2000
6000
10000
1960 1970 1980 1990
Year

seasonality rules out series (d), (h) and (i).

Trend rules out series (a), (c), (e), (f) and (i).

Increasing variance also rules out (i).

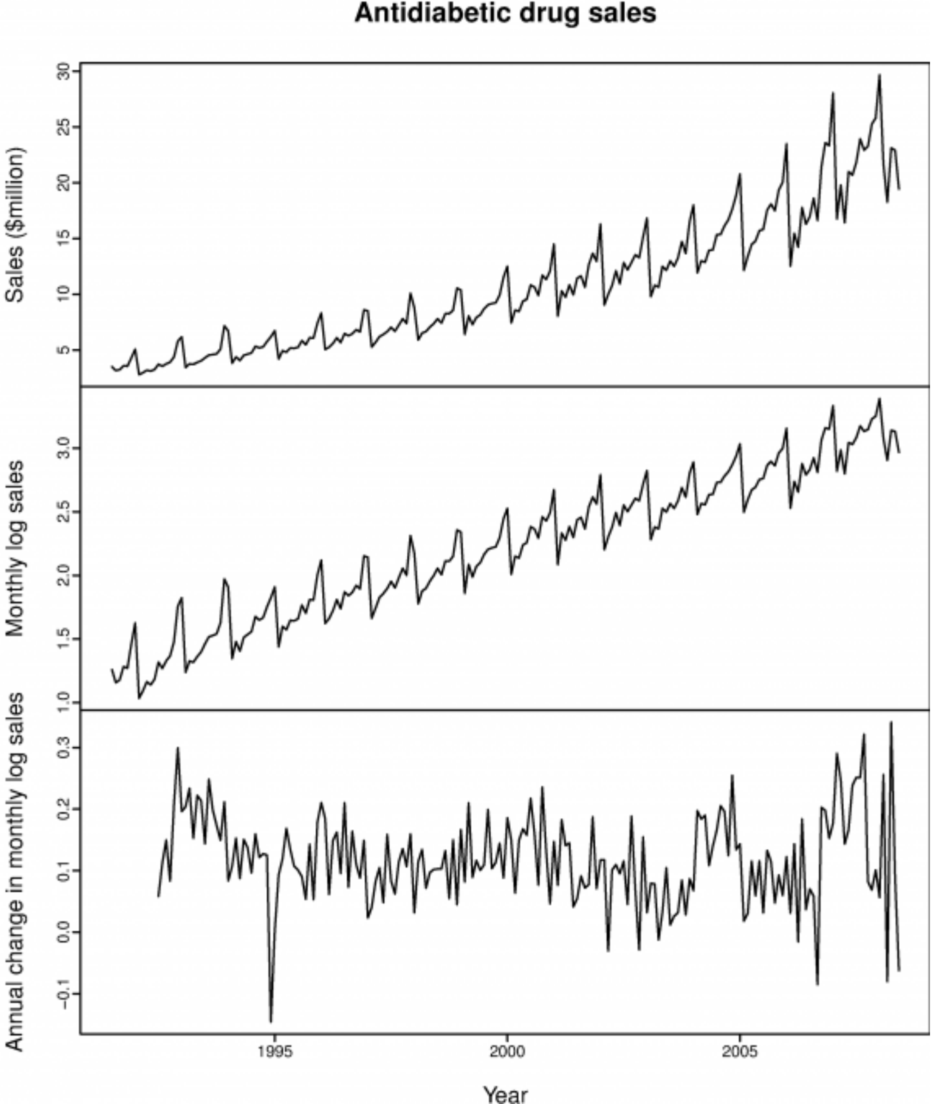
That leaves only (b) and (g) as stationary series. At first glance, the strong cycles in series (g) might appear to make it non-stationary. But these cycles are aperiodic



One way to make a time series stationary is to compute the differences between consecutive observations.

Differencing can help stabilize the mean of a time series by removing changes in the level of a time series, and so eliminating trend and seasonality.

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


Antidiabetic drug sales
Sales ($million)
Monthly log sales
Annual change in monthly log sales
Year
1995
2000
2005
5
10
15
20
25
30
1.0
1.5
2.0
2.5
3.0
−0.1
0.0
0.1
0.2
0.3

The backward shift operator BB is a useful notational device when working with time series lags:

$$By_t = y_{t-1}$$

Two applications of BB to y_t shifts the data back two periods:

$$B(By_t) = B^2 y_t = y_{t-2}$$

(This is also known as a Lag operator)

In an autoregression model, we forecast the variable of interest using a linear combination of past values of the variable. The term autoregression indicates that it is a regression of the variable against itself.

This is like a multiple regression but with lagged values of y_t as predictors. We refer to this as an AR(p) model

Rather than use past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model.

We refer to this as an MA(q) model

If we combine differencing with autoregression and a moving average model, we obtain a non-seasonal ARIMA model. ARIMA is an acronym for AutoRegressive Integrated Moving Average model ("integration" in this context is the reverse of differencing).

We call this an ARIMA(p,d,q) model, where

p= order of the autoregressive part;

d= degree of first differencing involved;

q= order of the moving average part.

The forecast error is simply $e_i = y_i - \hat{y}_i$, which is on the same scale as the data. Accuracy measures that are based on e_i are therefore scale-dependent and cannot be used to make comparisons between series that are on different scales.

$$\text{Mean absolute error: MAE} = \text{mean}(|e_i|),$$

$$\text{Root mean squared error: RMSE} = \sqrt{\text{mean}(e_i^2)}.$$

EVALUATING A TIME SERIES MODEL

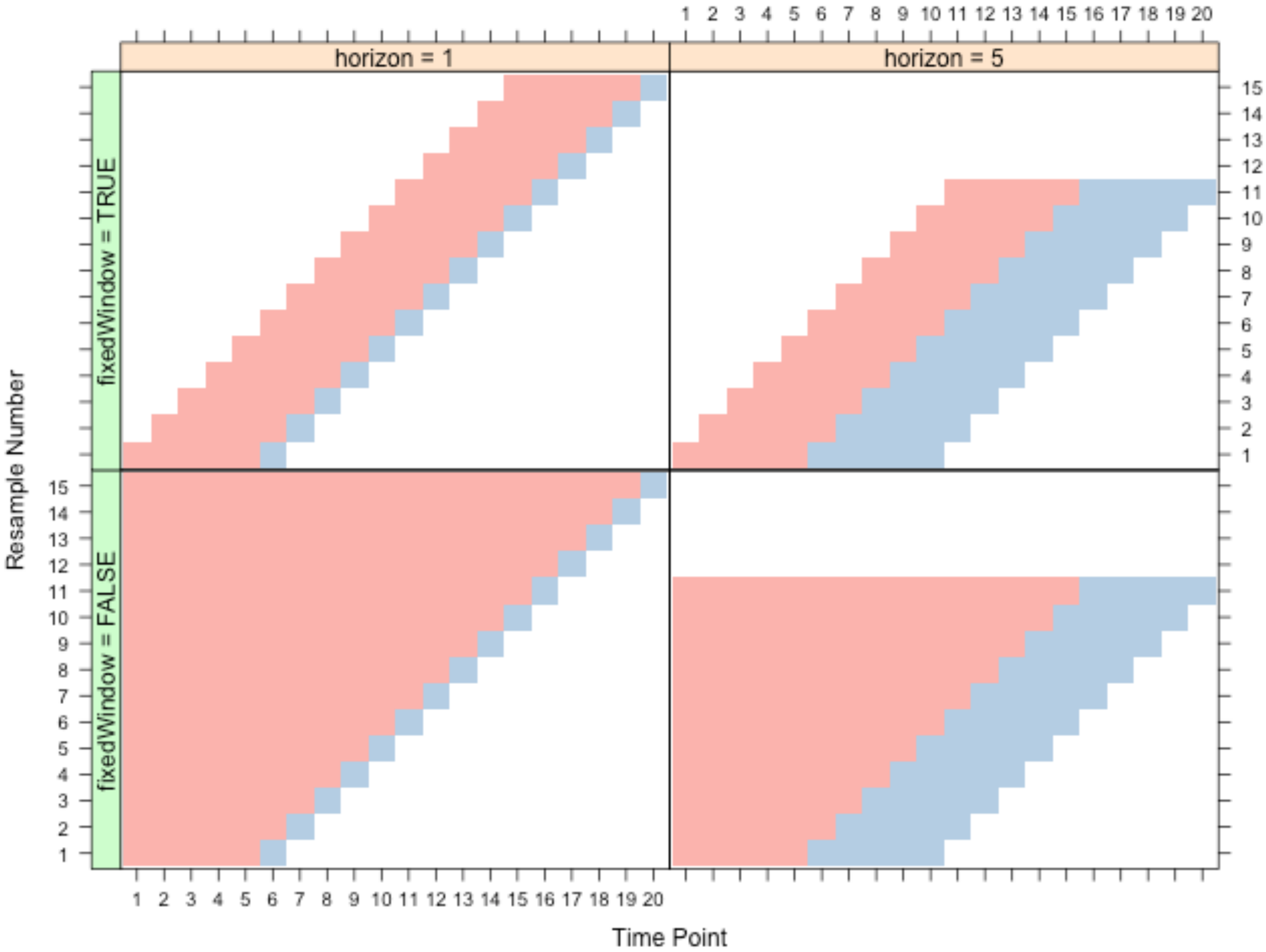
The percentage error is given by $p_i=100e_i/y_i$. Percentage errors have the advantage of being scale-independent, and so are frequently used to compare forecast performance between different data sets.

Mean absolute percentage error: MAPE=mean($|p_i|$)

CROSS-VALIDATION FOR TIME SERIES

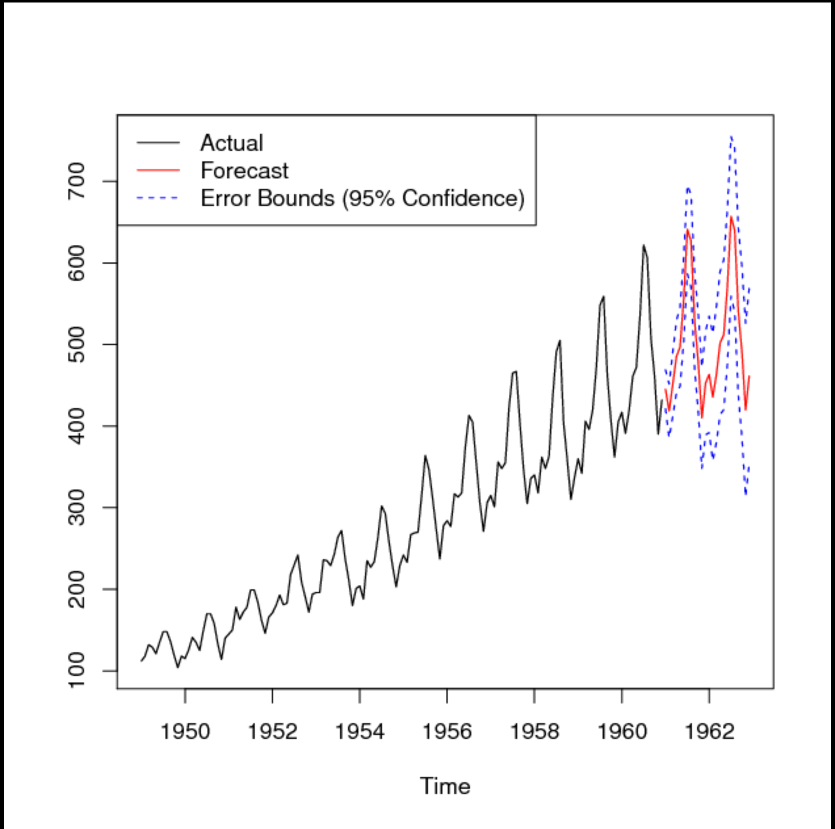
The three parameters for this type of splitting in the R Caret package are:

- **initialWindow: the initial number of consecutive values in each training set sample**
- **horizon: The number of consecutive values in test set sample**
- **fixedWindow: A logical: if FALSE, the training set always start at the first sample and the training set size will vary over data splits.**

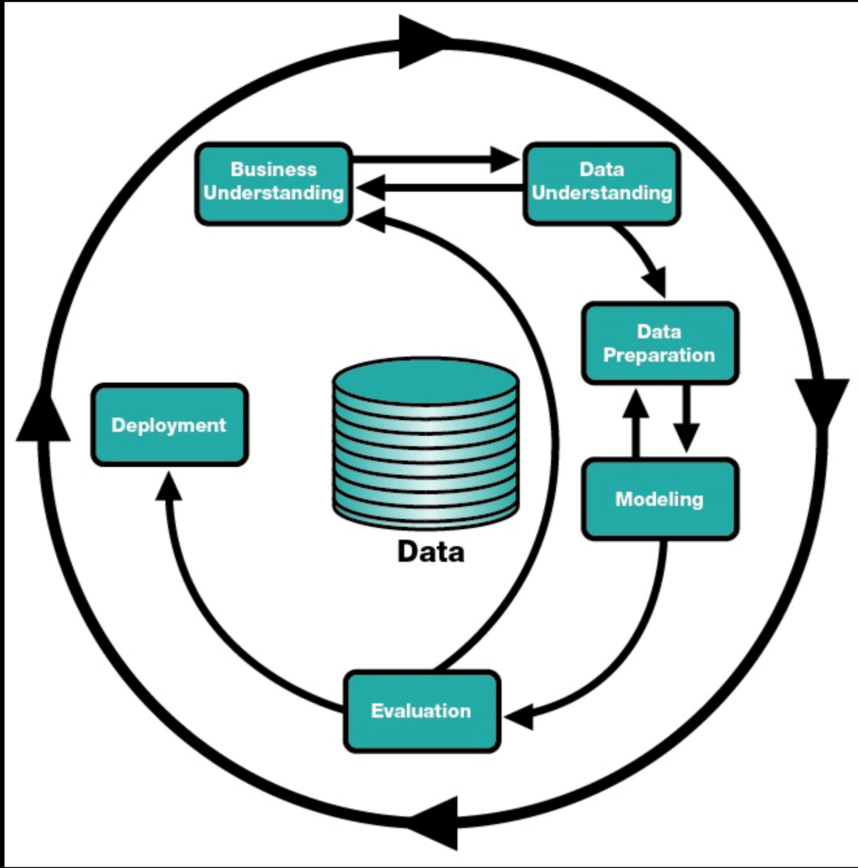

horizon = 1
horizon = 5
fixedWindow = TRUE
fixedWindow = FALSE
Resample Number
Time Point

DATA SCIENCE PART TIME COURSE

LAB



DATA SCIENCE CYCLE – Lesson 14



Lesson 14 - Review

FINAL PROJECT

- Final Project split into 4-parts: [Review with James 5mins]
    - (a) Real-world Problem Identification [Lesson 14]
    - (b) Data Cleaning [Lesson 15]
    - (c) Model & Validation [Lesson 16]
    - (d) Presentation & Storytelling [Lesson 17]

Lesson 18 – any final queries

Lesson 19 & 20 – Presenting final project back to class