# 301107 - Analytics Programming

*Nick Tothill*

*Practical class 1 - Excel; setting up R; R environment; basic data handling*

## Task 1. Clean up the Iris data

Use Excel to produce basic summaries of the iris data (given in vUWS, under Learning Materials; week02; practical). This famous (Fisher's or Anderson's) iris data set gives the measurements in centimetres of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

The data look like:

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

Hover, during the data collection process, something went wrong. As a result, some values are missing, and some values are incorrect (with values $\leq 0$).

This is a common problem in data science. So the first thing to do is to

- **clean up the data** by removing any record with **either missing values or incorrect values**
- copy the records with *good* values to another sheet and carry on the next task

## Task 2. Summarise the Iris data

Given now you have cleaned the data, try to work out some basic statistics of the flowers summarised according to different species. So you produce a table that looks likes the following (can be in different format):

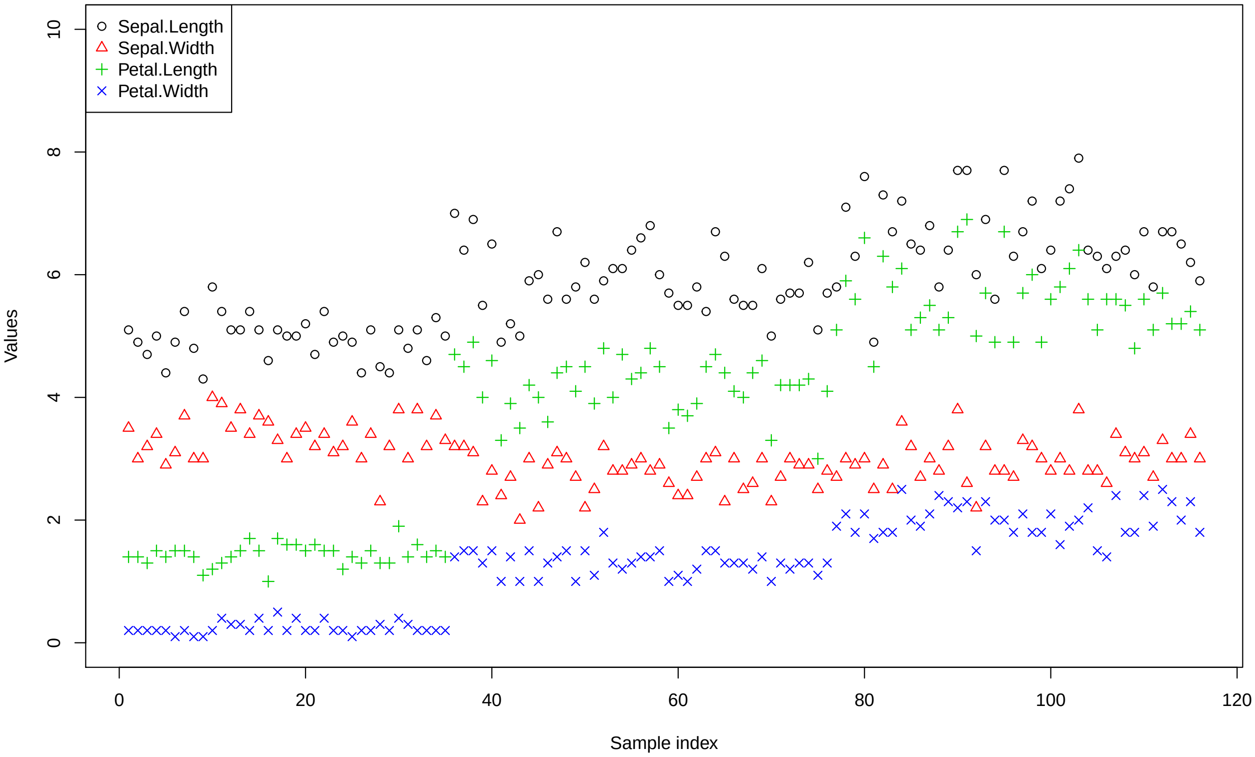
```
[1] "Species: setosa"
  Sepal#Length    Sepal#Width     Petal#Length    Petal#Width    
 Min.   :4.300   Min.   :2.300   Min.   :1.000   Min.   :0.1000  
 1st Qu.:4.750   1st Qu.:3.100   1st Qu.:1.350   1st Qu.:0.2000  
 Median :5.000   Median :3.400   Median :1.400   Median :0.2000  
 Mean   :4.957   Mean   :3.346   Mean   :1.437   Mean   :0.2371  
 3rd Qu.:5.100   3rd Qu.:3.600   3rd Qu.:1.500   3rd Qu.:0.3000  
 Max.   :5.800   Max.   :4.000   Max.   :1.900   Max.   :0.5000  
[1] "Species: versicolor"
  Sepal#Length    Sepal#Width     Petal#Length    Petal#Width   
 Min.   :4.900   Min.   :2.000   Min.   :3.000   Min.   :1.000  
 1st Qu.:5.500   1st Qu.:2.500   1st Qu.:3.900   1st Qu.:1.200  
 Median :5.800   Median :2.800   Median :4.200   Median :1.300  
 Mean   :5.885   Mean   :2.741   Mean   :4.171   Mean   :1.295  
 3rd Qu.:6.200   3rd Qu.:3.000   3rd Qu.:4.500   3rd Qu.:1.400  
 Max.   :7.000   Max.   :3.200   Max.   :4.900   Max.   :1.800  
[1] "Species: virginica"
```

```
  Sepal#Length     Sepal#Width     Petal#Length     Petal#Width
 Min.   :4.900   Min.   :2.20   Min.   :4.500   Min.   :1.400
 1st Qu.:6.175   1st Qu.:2.80   1st Qu.:5.100   1st Qu.:1.800
 Median :6.450   Median :3.00   Median :5.600   Median :2.000
 Mean   :6.590   Mean   :2.98   Mean   :5.575   Mean   :2.002
 3rd Qu.:7.125   3rd Qu.:3.20   3rd Qu.:5.825   3rd Qu.:2.225
 Max.   :7.900   Max.   :3.80   Max.   :6.900   Max.   :2.500
```

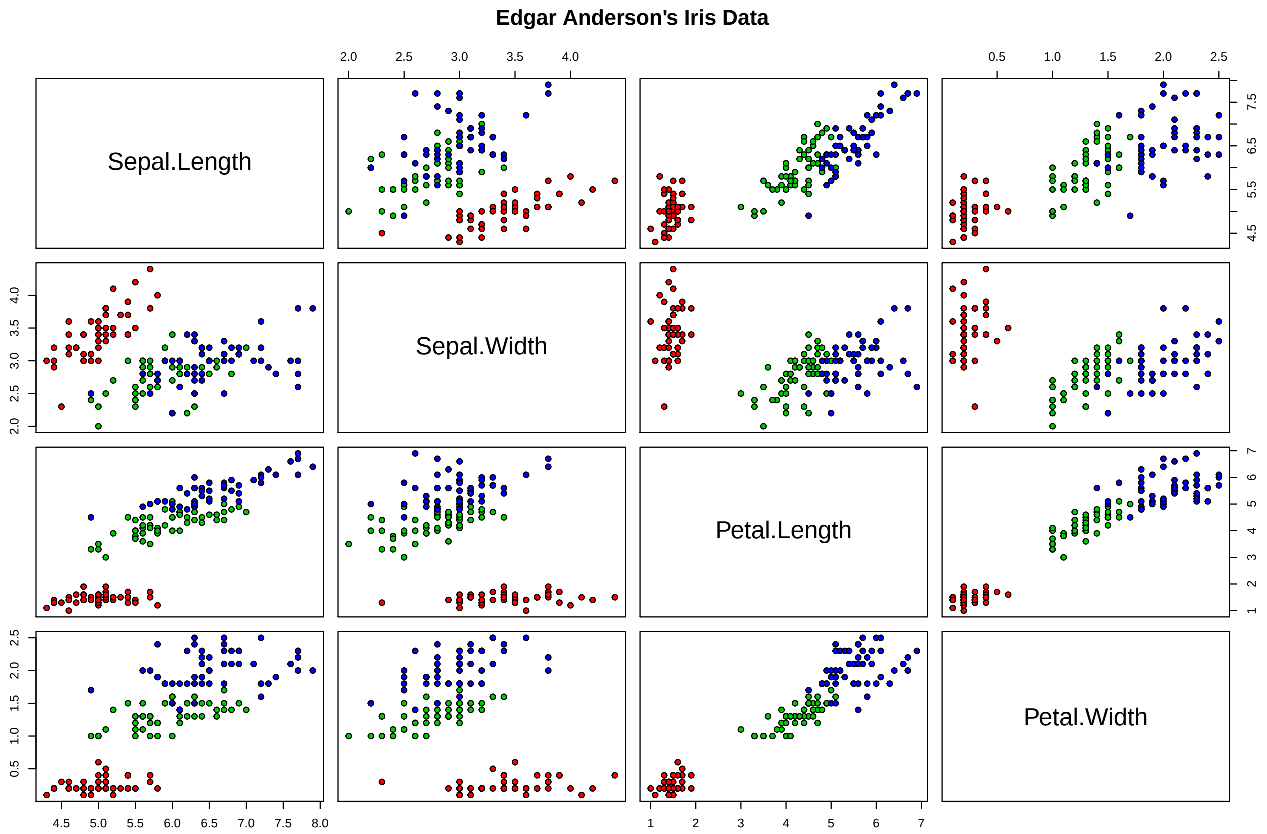
You may notice there are 6 statistics shown above. In this practical class, you *only need to find minimum, maximum and mean* for each variable.

## Task 3. Visualise the Data

Plot a figure. Use Excel's X Y (Scatter) chart to produce a plot similar to the following



However, it would be better to separate the species:



## Task 4. Starting R

Start Rstudio on your lab computer. If using your own computer, you should have Rstudio installed. If not, download and install while you're working on the practical on a lab computer.

In RStudio, try to run the following code

```
> 1+1
```

```
[1] 2
```

```
> sqrt(2)
```

```
[1] 1.414214
```

```
> print('Hello, world!')
```

```
[1] "Hello, world!"
```

## Task 5. Simple R session

Type in the following commands:

```
> x <- 1
> y <- 2
> z <- x+y
> ls()
> rm(x)
> ls()
> help(ls)
> help.start()
```

Look for *An Introduction to R*.

Go back to RStudio *R Console* panel and continue

```
> ls
> ls <- 1
> ls
> ls()
```

What happens with these different ways of writing `ls`? Notice the changes?

Continue

```
> getwd()
> save.image(file='test.RData')
> history()
```

What is `save.image(file='test.RData')` doing there?

What if I want to save all commands I used so far? (Hint: look up things related to `history` )

```
> rm(list=ls())
> load(file='test.RData')
> ls()
```

We cleared the current workspace by using `rm(list=ls())`. See what the argument `list` of function `rm()` means. What is `load()` doing?

If you have worked out how to save all commands you used in this session, you know how to save your work, which is likely to be important!

## Task 6. Practice with vectors

Try out the following code for vectors, and try to figure out what every line is doing

```
x <- seq(0,200,5)
print(paste('x is a vector of length',length(x)))
x
plot(1:length(x),x,main="Plot of vector x", xlab='Index', ylab='Values of elements in x',col=1)

print(paste('The 10th element in x is',x[10]))
cat('The first 5 elements in x are',x[1:5])
cat('The 5th, 7th, and 10th elements in x are',x[c(5,7,10)])
x1 <- x[1:4]
x2 <- x[(length(x)-3):length(x)]
x1
x1[-1]
x2
x1 + x2
-x1
x1 * x2
any(x1>5)
```

```r
which(x1>5)
x1[which(x1>5)]
x1[1] <- NA
x1
which(is.na(x1))
x1[-which(is.na(x1))]
all(x2<200)

x1 <- x1[-1]
x1
x1-5
x1*-1
sum(x1)
rep(1,length(x2))
```

There are quite a few indexing and manipulating functions in the above code, as well as a plotting function `plot()`. Find out what they do to a vector (hint: use help.)

After you understand the above, finish the following task, using the functions you have learned to make the code as simple as possible.

1. generate a random vector of length 10 (the function you need is `runif(10)`);
2. calculate the sum of the samples stored in the vector;
3. calculate the mean of the samples;
4. find all samples that are no less than 0.5 and calculate their mean;
5. find all samples that are less than 0.5 and calculate their mean.

You should have the results similar to the following:

```
My vector is
 0.0912563 0.7230549 0.7220213 0.8507309 0.48788 0.7993466
 0.2850184 0.0008180665 0.128345 0.5040901
```

```
Sum of samples: 4.592562
```

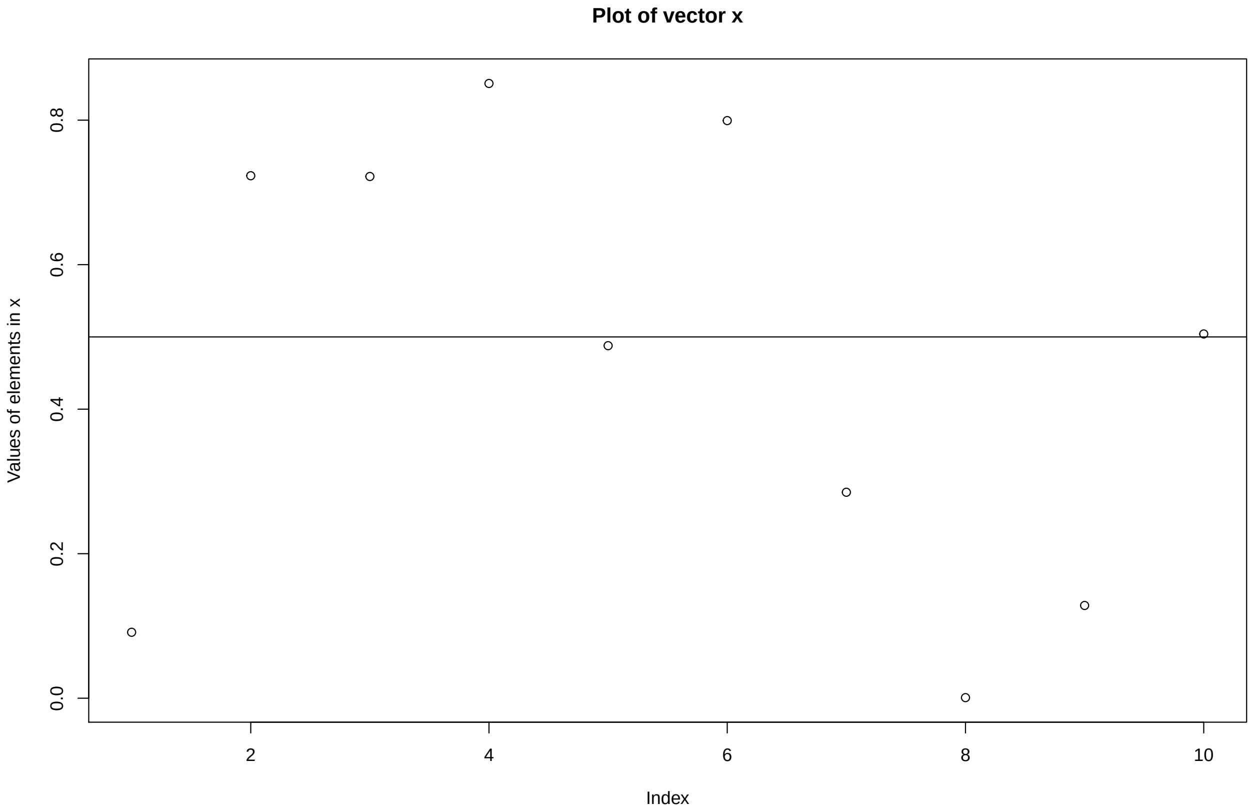
```
Mean of samples: 0.4592562
```

```
Mean of samples no less than 0.5: 0.7198488
```

```
Mean of samples less than 0.5: 0.1986636
```

your vector will have different elements in it, as, every time you run `runif` it will generate different values drawn from a uniform distribution between 0 and 1. So your results will be different from the above.

You can plot the vector here to get a visualisation of the vector like the following. The command to use is `plot`. The line in the middle (which is the mean of the data in the vector) is produced by another command `abline`. Use the help pages in R to see if you can figure out how to make a plot like the one shown below.



## Task 7. R self learning package swirl

A. install R package `swirl`
1. Start RStudio
2. Type in `install.packages('swirl')`. It will download and install the package.
3. load in the library by using `library(swirl)`
B. Have a `swirl` learning session

**N.B.** The learning module may take a long time to download.