**MAST7866 — Foundations of Data Science**

**Computing Session 3 — Graphical methods 2**

Within this worksheet we are going to continue look at producing some basic graphs in R. Make sure you have finished the worksheet for Computing Session 2 before moving on to this worksheet.

## Task 1 – Histograms

A histogram provides a visual representation of continuous data. The height of the bars in the histogram show how many of our sample data values occur in a particular interval.

**Example 1**

Suppose we have data on the distances (in miles) 42 employees of a particular company have to travel to work:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 15.2 | 7.1 | 3.8 | 5.2 | 28.3 | 17.2 | 27.4 | 29.9 | 13.4 | 17.0 | |
| 31.6 | 36.6 | 10.9 | 4.9 | 20.1 | 18.4 | 8.9 | 4.8 | 4.4 | 16.0 | 3.7 |
| 12.2 | 15.5 | 32.5 | 8.2 | 6.4 | 7.5 | 11.9 | 14.6 | 20.0 | 15.3 | 5.6 |
| 22.5 | 7.1 | 4.6 | 23.3 | 12.3 | 13.7 | 6.3 | 11.8 | 4.1 | 5.0 | |

These data are available as a text file called distance.txt. Load the data into R.

To plot a basic histogram of these data in R we type:

```
> hist(distance$dist)
```

This plot has used all of the default settings.

Type `?hist` to investigate some of the other options within this function.

What commands would you use to change the histogram to plot the histogram in colour, with blue borders and orange fill?

Suppose we want to put the breaks of the histogram at intervals of 7 miles rather than default 5. We can do this using the command:

```
> hist(distance$dist,breaks=c(0,7,14,21,28,35,42))
```

Alternatively, you can specify:

```
> hist(distance$dist,breaks=7)
```

**Challenge 1**

The file parkrun.txt gives the finishing times of the fastest male and fastest female runners for 230 Canterbury parkruns (a 5km run which takes place every Saturday morning). Plot histograms for the male and female data and provide a commentary on the differences.

Note if you are comparing two histograms it is often useful to set the x-axes to have the same limits in order to make easier comparisons.

## Task 2 – Scatter Plots

A scatterplot displays the value of 2 sets of data on 2 dimensions. Each dot represents an observation. The position on the x (horizontal) and y (vertical) axis represents the values of the 2 variables. It is really useful to study the relationship between variables.

**Example 2**

We are going to use a data set called mtcars which is already available in R.

Type

```
> ?mtcars
```

to find out more about the data.

We want to plot the weight variable versus the miles per gallon variable. Plot a chart for cars with weight between 2.5 and 5 and mileage between 15 and 30:

```
> par(mfrow=c(1,1))
```

```
> plot(mtcars$wt, mtcars$mpg, xlab = "Weight", ylab = "Mileage", xlim = c(2.5,5), ylim = c(15,30), main="Weight vs Mileage")
```

When we have more than two variables and we want to find the correlation between one variable versus the remaining ones we use scatterplot matrix. We use the `pairs()` function to create matrices of scatterplots.

Let us consider the relationship between variables mtcars variables wt, mpg, disp and cyl:

```
> pairs(~wt+mpg+disp+cyl,data=mtcars,main="Scatterplot Matrix")
```

Consider what each of the plots shows – note that we are displaying each graph twice.

## Example 3

We are going to explore the iris data set. Use the command `?iris` to find out more about the data.

Look at the first few rows of the iris data:

```
> head(iris)
```

Plot a scatter plot of the first 4 columns of data:

```
> pairs(iris[,1:4], pch = 19)
```

`pch` controls the symbol used. Find out more about this command using `?pch`. Adjust the pch variable to see what changes.

```
> pairs(iris[,1:4], pch = 23)
```

This plot includes each graph twice – to only show the plots the once we can use the following command:

```
> pairs(iris[,1:4], pch = 19, lower.panel = NULL)
```

Now suppose we want to colour points by groups (species).

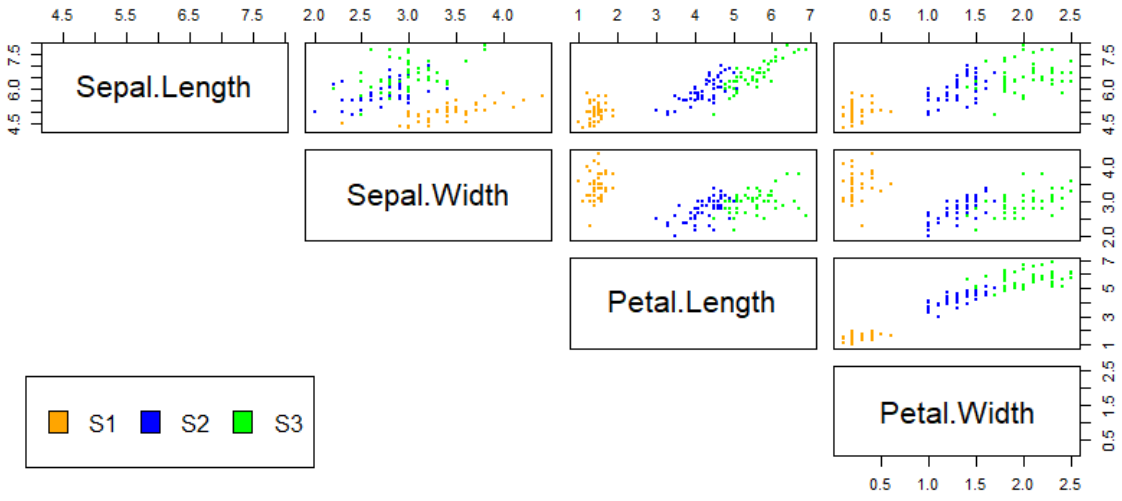
```
> dev.off() # removes the previous plot settings
```

```
> my_cols <- c("orange", "blue", "green")
```

```
> pairs(iris[,1:4], pch = 19, cex = 0.5, col =
my_cols[iris$Species], lower.panel=NULL)
```

```
> par(xpd = TRUE, oma=c(1,1,1,1))
```

```
> legend("bottomleft",fill =
unique(my_cols[iris$Species]),legend=c("S1","S2","S3"),horiz=T
RUE)
```



## Challenge 2

The text file temp_obs.txt contains minimum air and ground temperatures taken over 93 nights.  Use scatter plots to visualise the temperature data, using colour coding to distinguish between the different observers recording the measurements.

*Hint: If your legend does not display try including the dev.off() command at the start of the code*