Using R as a Research Tool.
Part 2: R Markdown and basic statistics.

Dr Susan Johnston: Susan.Johnston@ed.ac.uk

Demonstrators: Gergana Daskalova, John Godlee.
Hat-Tips to Kyle Dexter, The Coding Club and R4all.

November 26, 2017

1 Introduction

This practical will follow on from the previous practical in data manipulation and visualisation, exploring how to write reports in **R** Markdown and how to conduct simple statistical tests in **R**. By the end of the practical, you should be able to:

- Write, embed and render code and results into an HTML document.
- Carry out basic statistics and visualisations, including:
  - Linear regression with `lm()`
  - Chi-squared test with `chisq.test()`
  - 2-sample t-test with `t.test()`

2 Writing reports with R Markdown.

R Markdown is a tool for writing, reproducible reports in **R**. It can be used to produce documents with embedded code and figures in HTML, Word and PDF format, and can also be used to create webpages and slideshows. The current version of the **R** Markdown Cheatsheet from RStudio is included in the github repository `https://github.com/susjoh/Intro_to_Stats_in_R`.

2.1 Creating an R Markdown Document.

Open RStudio and create a new markdown document by going to **File > New File > R Markdown...**. In the window, name your document, select **HTML** and click **OK**. RStudio should automatically create a template as in Figure 1 (if not - it is saved in the file `R_Markdown_Template.Rmd`).

To render the document, click the button that says **Knit** (you may have to save it first). As you can see, it produces a formatted HTML document with embedded figures.

Another thing to note is the **Show document outline** in the top right corner of the window, which shows the document structure.

Figure 1: R Markdown Template.

2.2 Formatting an R Markdown Document.

It is possible to carry out basic formatting, tables, headers, as well as adding tables, lists and external figures. See Part 3 of the R Markdown CheatSheet from RStudio, which is attached to the back of this document.

Exercise 1.

1. Create a new **R** Markdown Script with the following header:

```
--
title: "Introduction to Statistics in R"
author: "Your Name"
date: "28 November 2017"
output:  html_document
--
```

Add some text to it and familiarise yourself with how to make *italic*, **bold** and superscript text. Add headers for "Introduction", "Linear regression", "Chi-squared test" and "Two-sample t-test"..

2.3 Embedding code in R Markdown.

Code can be embedded into the document in two ways:

Code Chunks:

````
```{r}
head(iris)
```
````

Code chunks can be named, e.g. ```` ```{r iris} ````

Inline code:

```
Two plus two equals `r 2 + 2`.
```

Which will print "Two plus two equals 4.".

There are also a number of display options for each chunk in section 5 of the cheat sheet. For example,

```
```{r echo = F, results = "hide", fig.width = 4, fig.height = 3}
head(iris)

library(ggplot2)
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) + geom_point()
```
```

will show the code output and the figure (dimensions specified with `fig.width` and `fig.height`), but not the code itself (`echo = F`) or the console output (i.e. `head(iris)`; `results = "hide"`).

2.4 Points to note.

- Rendering the document with **Knit**, that R is opening and creating a new working environment. Each code chunk that is run is saved in the workspace - therefore, libraries and new objects created in a chunk will remain in the workspace for all subsequent chunks.
- It's good to put a chunk at the start loading the libraries and data.
- Code in R Markdown chunks can be run in the console as in the previous practical. The default option is to show the code output inline in the document. Some people like this, others don't: to switch this function on and off, go to **Global options > R Markdown > Show output inline for all Markdown documents** and select your preferred setup.
- R Markdown documents are not flexible to errors - the code must be error free, or it will not render.
- Want to learn more? More detailed information on this can be found at the Coding Club tutorial at `https://ourcodingclub.github.io/2016/11/24/rmarkdown-1.html`.

For the rest of the practical, it is expected that you create an R Markdown document that contains text, code, graphs and inline reporting of results.

3 χ^2 contingency table

A χ^2 contingency table analyses count data, and looks at the association between two or more categorical variables. In this example, we will examine the differences in the frequency of red and black ladybirds (*Adalia bipunctata*) in rural and industrial habitats. Our question is: are dark morphs more likely to reside in dark (industrial) backgrounds? The null hypothesis is that there is no association between ladybird colour morph and habitat type [1].

Load the data file `ladybirds.csv` into R using `read.csv()` and examine it using `glimpse()` from the `dplyr` package.

```
library(dplyr)

ladybirds <- read.csv("data/ladybirds.csv", header = T)
glimpse(ladybirds)
```

```
Observations: 20
Variables: 4
$ Habitat      <fctr> Rural, Rural, Rural, Rural, Rural, Rural, Rural, Rura...
$ Site         <fctr> R1, R2, R3, R4, R5, R1, R2, R3, R4, R5, U1, U2, U3, U...
$ morph_colour <fctr> black, black, black, black, black, red, red, red, red...
$ number       <int> 10, 3, 4, 7, 6, 15, 18, 9, 12, 16, 32, 25, 25, 17, 16,...
```

There are multiple lines for each category, with the column number giving the count details. We ultimately want four numbers, corresponding to the 2×2 categories: red industrial, black industrial, red rural and black rural.

This can be done using the `dplyr` functions `group_by()` amd `summarise()`.

```
> totals <- group_by(ladybirds, Habitat, morph_colour)
> totals <- summarise(totals, total.number = sum(number))
> totals
```

[1] The approach and dataset here is based on the example presented in the book "Getting Started with R" (2nd Edition, 2017) by Beckerman, Childs and Petchey (`http://www.r4all.org`)

```
# A tibble: 4 x 3
# Groups:   Habitat [?]
     Habitat morph_colour total.number
      <fctr>       <fctr>        <int>
1 Industrial        black          115
2 Industrial          red           85
3      Rural        black           30
4      Rural          red           70
```

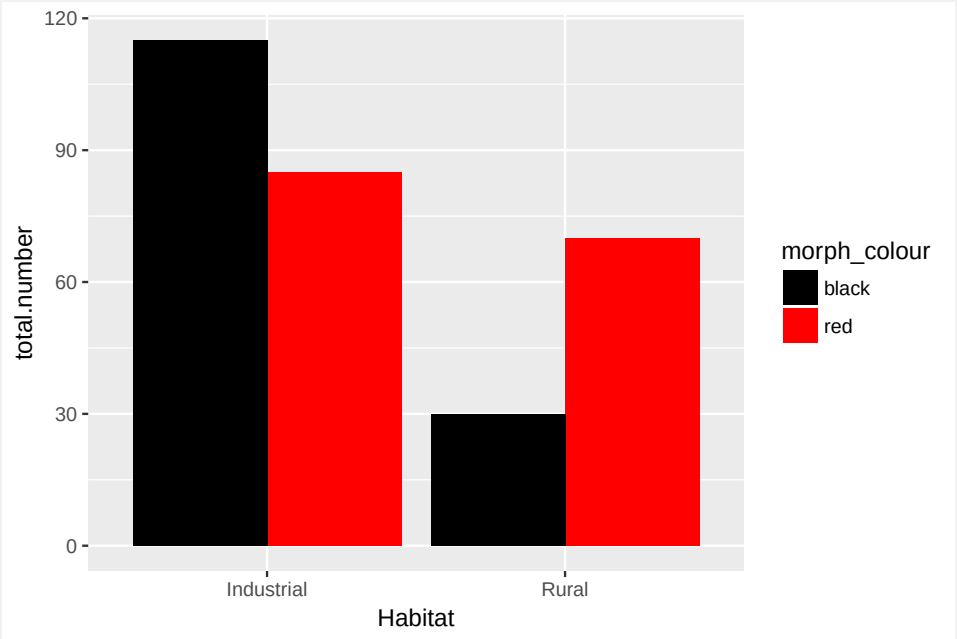
3.1 Plot the data

One visualisation for this type of data is a bar chart using `geom_col()` in `ggplot2`.

```
> library(ggplot2)
> ggplot(totals, aes(x = Habitat, y = total.number, fill = morph_colour)) +
+   geom_col()
```

There are two edits we can make to this to improve the visualisation: to add `geom_col(position = "dodge")` to place bars side by side, and also by changing the colours of the bars to black and red to match the colour of the morphs in real life (using `scale_fill_manual()`):

```
> ggplot(totals, aes(x = Habitat, y = total.number, fill = morph_colour)) +
+   geom_col(position = "dodge") +
+   scale_fill_manual(values = c(black = "black", red = "red"))
```



3.2 Test the hypothesis with `chisq.test()`.

The χ^2 is run using the function `chisq.test()`. As this is a 2×2 contingency test, we must convert the data into a matrix. Looking at the data `ladybirds`, a matrix can be made using the function `xtabs()`, which is similar to creating pivot table cross-tabulation in Excel:

```
> lady.mat <- xtabs(number ~ Habitat + morph_colour, data = ladybirds)
> lady.mat

            morph_colour
Habitat      black red
  Industrial   115  85
  Rural         30  70
```

Now run the test:

```
> chisq.test(lady.mat)
```

```
        Pearson's Chi-squared test with Yates' continuity correction

data:  lady.mat
X-squared = 19.103, df = 1, p-value = 1.239e-05
```

This provides a statistic indivating that there is a very small probability that the observed pattern arose by chance. Therefore, we can reject the null hypothesis. We can extract more information from the statistic if we save the `chisq.test(ladymat)` as an object:

```
> lady.chisq <- chisq.test(lady.mat)
```

Running `lady.chisq` gives the same output as before, but we can explore the object in detail using the `$` notation:

```
> names(lady.chisq)

[1] "statistic" "parameter" "p.value"   "method"    "data.name" "observed"
[7] "expected"  "residuals" "stdres"

> # str(lady.chisq) # not run here to save space - please run it!
>
> lady.chisq$statistic

X-squared
 19.10289

> lady.chisq$p.value

[1] 1.238571e-05
```

In the **R** Markdown document, it is possible to quote statistics inline using the `` `r` `` notation e.g. `` `r lady.chisq$statistic` `` and `` `r lady.chisq$p.value` `` will print the χ^2 statistic and P value inline, respectively.

Exercise 2.

Create a short report in the **R** Markdown document with an inline report of the test statistics and P-value. This can be done as follows:

1. Run a code chunk for loading and manipulating the ladybird data, and running the χ^2 test that does not print the code or results to the compiled document (hint: define `echo` and `results` in the chunk options).

2. Write a few lines of text stating the hypothesis, the test statistic and interpretation. E.g. *The null hypothesis is... Ladybird morphs are not equally distributed in the two habitats (Chi squared = ..., df = ..., P = ...), with black morphs being more frequent in.... (Figure 1)*

3. Run another code chunk to create a barplot for Figure 1.

4 Two-sample t-test.

A two-sample t-test is one of the most simple and commonly used hypothesis tests. It determines whether the mean of two groups of numeric values are significantly different or due to random chance. Here, we will test whether the sepal length differs between two *Iris* species, *I. virginica* and *I. versicolor* (Figure 2).

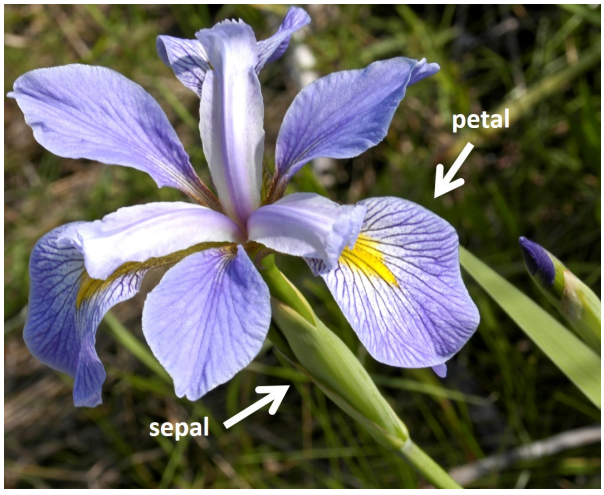


Figure 2: *Iris virginica.*

This test makes two assumptions about the data - that both groups are normally distributed

and that the variances are equal in each category.

First, load the data:

```
sepals <- read.csv("../data/iris.edited.csv", header = T)
str(sepals)

'data.frame':        100 obs. of  2 variables:
 $ Species     : Factor w/ 2 levels "versicolor","virginica": 1 1 1 1 1 1 1 1 1 1 ..
 $ Sepal.Length: num  7 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 ...
```

The first step is to create a figure, as it is important to visualise the data before analysing it. One approach is to use a boxplot, which is more visually appealing (but based on the median rather than the mean) - another is to use histograms. These can help us to assess if the means seem different between the two categories, and if the data is normally distributed with a similar variance. It can also provide an inidication of whether the null hypothesis can be accepted or rejected.

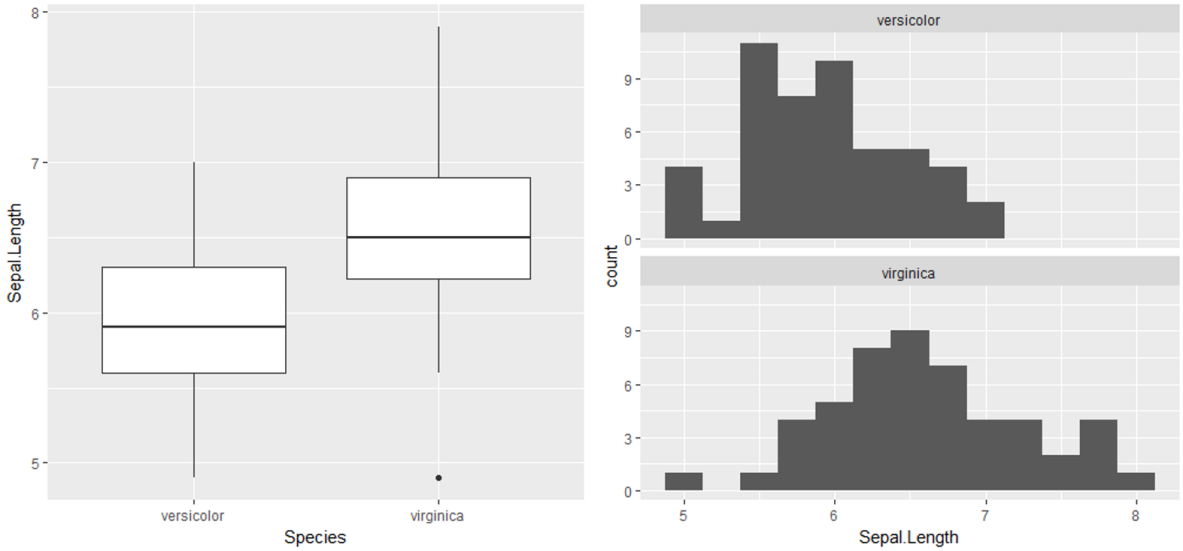
```
# boxplot

ggplot(sepals, aes(Species, Sepal.Length)) + geom_boxplot()

# histogram with facet_wrap

ggplot(sepals, aes(Sepal.Length)) +
  geom_histogram(binwidth = 0.25) +
  facet_wrap(~Species, ncol = 1)
```

To carry out the t-test, we will use the `t.test()` function. We can find out the details of the test using `?t.test` as before. The syntax requires a formula `Sepal.Length ~ Species` and the data frame which contains the data `data = sepals`. This should reflect the hypothesis - how does sepal length vary as a function of species?




```
> t.test(Sepal.Length ~ Species, data = sepals)

        Welch Two Sample t-test

data:  Sepal.Length by Species
t = -5.6292, df = 94.025, p-value = 1.866e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.8819731 -0.4220269
sample estimates:
mean in group versicolor  mean in group virginica
                   5.936                    6.588
```

The function has automatically used the Welch version of the t-test, which relaxes the assumption of equal variances - this is fine for the purposes of this practical (see Beckerman et al for a discussion of this in more detail). The output provides the `t`, `df` and `p-value` for the test, as well as the mean value in each of the two groups. The 95% confidence interval shows the interval between the difference between the two means - if this overlapped 0, then we would retain the null hypothesis.

Therefore, given the output, we can reject the null hypothesis, and can conclude that *I. virginica* has longer sepals than *I. versicolor*.

Exercise 3.

1. Create a short report in the **R** Markdown document with an inline report as for Exercise 2.
2. Use the CheatSheet (Hint: Section 3) to add the image of *Iris virginica* to the Markdown document ("data/Irisvirginica.jpg")

5 Simple Linear regression.

The last model we will tackle is a linear regression. This is the most basic of a class of models called 'general linear models' which also includes multiple regression and ANOVA.