Handout R: Introduction to R for 20x

Russell Millar

R Background

Development of the R language began in the early 1990's by two lecturers at the University of Auckland, Robert Gentleman and Ross Ihaka. The initial form of R was loosely based on a commercial language called S that originated at AT&T Bell Laboratories in the mid 1970's.

The motivation for R was to provide an open source language that was free for all to use and to extend, and one of its first uses was for the teaching of STATS 20x.

From its humble beginnings, R is now the most widely used statistical language in the world. Moreover, in 2015, R became the 6th most widely used programming language (behind Java, C, C++, Python and C#).

R, RStudio, and R markdown

In the first week of 20x you may encounter some very unfamiliar looking programming code. Don't panic - there is plenty of help available and your lecturer will be careful to explain what is going on.

- R is the underlying language you will be running
- R markdown is an R add-on that enables you to create documents (i.e., the answers to your assignments)
- RStudio is the interface (GUI) that you will use to run R and R markdown

Tutorials will be provided in the computer lab during the first week of classes - be sure to check Canvas for times. Please take advantage of these tutorials as they have been designed to help you breeze through the more technical parts of your assignments.

Note that there are also Demonstrators in the Basement Lab of building 303s, room B91, who can help you from 9am-8pm Mon-Thurs and 9am-5pm on Friday. They are always willing to help students better understand the more technical side of this course and can help out with any problems or questions you have regarding R.

There are also plenty of online resources, e.g., See https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf

Assignments in 20x

You will be using R markdown to write your assignments. The great advantage of R markdown is that it automatically inserts the output of your statistical analyses into your homework document. **Copy and paste is now obsolete.**

This document was created by using R markdown to process the contents of file H0_R.Rmd. Your lecturer will show you how.

R markdown

If you look at file H0_R.Rmd you will see that some of the text has been interpreted as formatting. For example, the double hash ## is used at the start of a line to get a section header that uses the "Header 2" style.

In the next sections we see the use of R Chunks for running R commands, and how to excecute those commands.

Writing and running R Chunks

- A code chunk begins with ```{r}
- A code chunk terminates with ```

Anything inside the chunk will be interpreted as R code.

You can run selected lines or chunks of code in R using the Run Icon on the top right corner of the RStudio window, or alternatively the corresponding hotkeys. The R code and its output will automatically be inserted into your document.

The document is produced using a process called knitting, because it knits together the text and R code/output into a single document. The knit Icon at the top of the RStudio window lets you choose whether to produce the document in HTML, Word or PDF format.
Try Knitting this file now as a Word document.

Some example R chunks

This first example simply creates two numeric values, `y` and `z`.

```r
y = 3
y
```

```
## [1] 3
```

```r
z = y * 4
z
```

```
## [1] 12
```

Note that the lines of R output are prefixed by `##` (this is an option that can be changed).

It is important to be aware that R is an object-oriented language and everything that is created within an R session is some form of object. We refer to `y` and `z` as "objects". In this case they are numeric objects.

Following are a few more commands. Can you figure out what each one does?

```r
w = c(1, 2, 3, 4, 5, 6, 7, 8, 9)
w
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

```r
X = matrix(w, nrow = 3, ncol = 3, byrow = T)
X
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

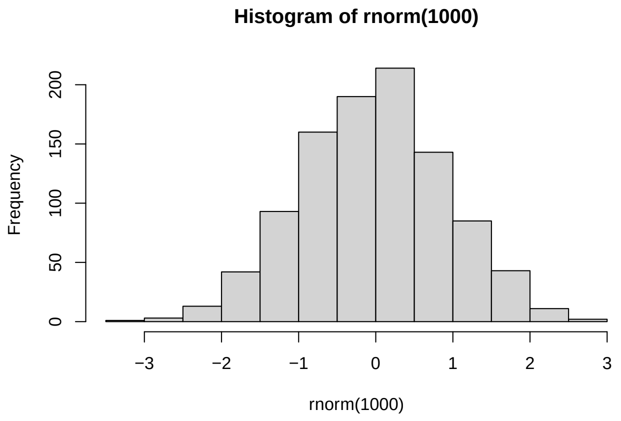
In the first line of this chunk we have used the `c` function, to create vector `w` from its arguments.

In the second line we have used the `matrix` function to create matrix `X` using the values in vector `w`.

R graphics

As you will discover, R is very good at producing plots. For example, here is a histrogram of one thousand randomly generated values from a normal distribution

```
#Using the hist function to draw a histogram
hist(rnorm(1000))
```



Note the use of # within the chunk to add a comment to the R code.

Reading data from a file

We will use the `read.table` function to input data from a data file. It is easiest if the data file is in the same folder as your R markdown file.

```
df = read.table("example.txt", header = T)
df
```

```
##   Height Weight      BMI
## 1   1.83     90 26.87450
## 2   1.72     89 30.08383
## 3   1.80     72 22.22222
```

Take a look at the `example.txt` file. The `header = T` (short for `TRUE`) argument tells R that the variable names `Height`, `Weight` and `BMI` are on the first line of this file. This is how data files will typically be provided in this course.

The data in `example.txt` have been saved in an object called `df`, which is a dataframe object.

Writing equations

R markdown can also write equations. Here is one you might recognize, $\bar{y} = \sum y_i/n$.

Challenges

The biggest challenge is to deal with the error messages that you will get from R and R markdown. They are often not very informative - though they typically give the line number in your R markdown file where the error occurred, and that is a good place to start looking.

Also, be aware that R is case sensitive, so `df` is different from `Df`.

- The R command `df` will print out the dataframe created above.
- The R command `Df` will cause an error and this document will not be produced.

Your lecturer will do his/her best to demonstrate a range of common errors that can occur!

Expect to see lots of error messages when you work on Assignment 1 - start this assignment early