Introductory practical: R basics

1 Aims

The main aim of this practical is to introduce you to the statistical software package **R**, and the Rstudio environment we will use throughout the computing part of this module. You should already be familiar with the basics from last week'ss introductory lecture; now it's your turn to get your hands dirty!

None of this work will be assessed – it's all practice at using **R**. The first 'proper' **R** practical will take place this coming Friday; Monday practicals are usually there for extra assistance with the work that has been set the previous Friday.

Please read through all of this handout *very carefully*, and follow the instructions *meticulously*. Whenever you are typing commands into **R**, make sure you do so *extremely carefully*. If you get an error message, try typing the code in again, just to make sure you've not made a spelling mistake or other simple typo. A common error is to forget that **R** is case sensitive, so use upper/lower cases exactly when you're told to do so.

For all **R** practicals, make sure you've got your lecture notes handy.

Attendance at computer practical sessions is vital if you are to succeed in this part of the module. To help you out here, an attendance register will be taken in every session.

2 Getting started

Log-in to a PC in the usual way. Now open Rstudio in exactly the same way Lee did in last week's intro lecture. If you weren't there, or can't remember how to do this, click on the "Start" button in the bottom left-hand-corner and search for "rstudio". Once you've opened Rstudio you should see four panels; if there is no text editor (usually the top-left panel), click on the green button with a white plus symbol and open a new **R** script. Remember – to execute a line of code in the text editor, hit `Ctrl-Enter`.

It is good practice to always use the text editor, rather than type into the console window (bottom- left panel) directly. This is especially true in future practicals when you will be building large functions; it is much easier to de-bug and edit code in the text editor than in the console window.

Hint: To commit several lines of code in one go, highlight your block of code in the text editor and hit `Ctrl-Enter` just once!

Now you've got Rstudio open, you're ready to go! Try out *all* of the following exercises.

3 Simple calculations

Use **R** to find the following. Remember, it's good practice to type your code into the text editor (top-left window), before hitting `Ctrl-Enter` to commit this code to the **R** console (bottom-left window).

(a) 232×4096

(b) The sum of 731, 519, 345 and 3512

(c) $\log(20)$ (unless otherwise stated, $\log = \log_e = \ln$)

(d) e^5

(e) 8! (what do you *think* the command will be? Don't be scared to try...)

(f) 20 mod 3 (Think back to the lecture, or Google it!)

(g) 7^{32} ...can you interpret the answer as given by **R**?

(h) $\frac{2}{7} \div \frac{13}{9}$

(i) $\sqrt{831}$

(j) $\lfloor\sqrt{831}\rfloor$, where $\lfloor\ \rfloor$ represent the integer part (e.g. $\lfloor 9.83 \rfloor = 9$; explore the `floor` command in **R**)

(k) The value of π to six decimal places

4 Assigning in R

Enter the code in listing 1, one line at a time, and then answer the following questions. Don't forget - it's good practice to enter the code into the text editor first, and then commit using `Ctrl-Enter`. As the listing header suggests, you will get some error messages!

Listing 1: Assignment and simple error messages

```
x = "1"
y = log(2)
z = 10
y = y*z
a = y^y
b = a/0
d = x*z
z = Z+1
```

(a) What is the current value of `y`?

(b) What is the current value of `z`?

(c) What is the current value of `a`?

(d) What is the current value of `b`?

(e) Why do you get an error message when you try to assign something to `d`?

(f) Make a new object `e` which is equal to `a` to the power `z`.

(g) Create a new object `f` which takes the string "Maths".

(h) Create a character vector `g` of length two which has `f` as the first element and `x` as the second element.

(i) What happens if you try to find the difference between `a` and `b`?

(j) Why does the last line of code give an error?

(k) Look at the top-right window. Make sure you understand what the "Environment" and "History" tabs are showing.

5 Exploring the movies database

In this question we will explore the movies database that was introduced in last week's lecture.

- The first line loads an **R** package associated with another course - MAS1343 - which we will find useful for this course
- Make sure you type everything in *very* carefully!
- Consult Chapter 1 of your lecture notes for a full description of this database

Listing 2: Loading and exploring the movies database

```
library(mas1343)
data(movies)
head(movies)
dim(movies)
```

(a) The command `head` displays the "head" of the dataframe - the first six rows. What do you think the command `dim` does? (If you're not sure, take a guess and see if it makes sense... clue: what do you think `dim` might be shorthand for?)

(b) In the movies database, what is the third film when all films are listed in alphabetical order?

(c) In what year was the film "*Abandon*" released?

(d) What was budget for the film "*Aberration*"?

(e) What was the length, in hours and minutes, of the film "*Alien Vs. Predator*"?

(f) Of the first six films listed in the database, which was rated most highly?

(g) What genre of film is "*AKA*"?

(h) What was the budget of the film "*Zoolander*"? (Hint: Experiment with the `tail` command in **R**)

(i) As we will see over the next week or so, if a dataframe has column headings you can extract column variables using the `$` symbol. For example, commit the code `movies$Budget` and see what happens.

(j) Assign the movie budgets to the object `a` in exactly the same way you did assignment in Section 4 of this worksheet.

(k) As we will see in a few weeks' time, **R** has many useful built-in functions for statistical analysis. What do you think the code in listing 3 is doing?

Listing 3: Using one of **R**'s built-in functions

```
mean(a)
mean(a[a!=(-1)])
```

You can save today's **R** session by clicking on `File-Save As`..., although don't feel compelled to as this work is not assessed. Make sure you keep your files organised. For example, you should create a folder called something like "`MAS1802`" and save this session as "`Practical0`" - future sessions can be saved as. e.g., "`Practical1`".