

First R Refresher

Getting started

Log-in to a PC in the usual way. Now open Rstudio; 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 open in 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.

In the questions below, the important part is **understanding** what’s going on rather than just typing the R commands. Make sure you answer the questions under each piece of R code. If you can’t answer the question, then ask! Where possible, write your answer down on the worksheet. You should create the folder Practical1 in a directory called MAS2906 – this is where you can save your R session and any other files associated with this practical.

Question 1

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!

```
1 x = "1"
2 y = log(2)
3 z = 10
4 y = y*z
5 a = y^y
6 b = a/0
7 d = x*z
8 A = seq(0, 1001, by=0.5)
9 z = Z+1
```

Listing 1: Assignment and simple error messages

- (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) What has R done to produce the vector `A`?
- (k) Why does the last line of code give an error?
- (l) Use R to generate the vector `-1, -0.5, 0.0, ..., 10`.
- (m) Look at the top-right window. Make sure you understand what the “Environment” and “History” tabs are showing.

Question 2

Vectors:

```
1 #You will find out more about the runif command in a few weeks.
2 x = runif(5000, 1, 8)
3
```

Listing 2: Vectors

Carefully type the code in listing 3 into R and then answer the following questions.

- (a) What is the length of `x`?
- (b) How many values in your vector `x` are below 2?
- (c) How many values in your vector `x` are above 7?
- (d) How many values in your vector `x` are below 3 or above 8?
- (e) How many times does the value 2 occur?
- (f) What are the 3400th – 3402nd elements in your vector?
- (g) What is the smallest value in `x`?
- (h) What is the largest value in `x`?
- (i) What are the first and third quartiles of `x`?
- (j) Create a new vector called `y` which contains all values of `x` that are between 2 and 6.
- (k) **Tricky:** Which value in `x` is the closest to 5, i.e.

$$\min(|5 - x_i|)?$$

You can use the `abs` to work out absolute values.

Question 3

- (a) Create a vector of characters (strings), containing the names of your best friends – call this vector `ID` (minimum three friends, maximum ten friends).
- (b) Create a vector of characters (strings) containing the sex of these friends – call this vector `sex`.
- (c) Create a vector of logicals specifying whether or not each friend likes Marmite – call this vector `marmite`. If they don't know, the entry should be a missing value (`NA`).
- (d) Estimate the height of your friends, to the nearest cm, and create a vector of doubles containing these estimated heights – call this vector `heights`.
- (e) Use the R function `sort` on each of your four vectors. What happens?
- (f) Use the R function `rev` on each of your four vectors. What happens?
- (g) What does the code in Listing 3 below do?

```
1 ID[marmite == TRUE]
2 D = data.frame(ID, sex, marmite, heights)
3 D1 = D[order(heights),]
```

Listing 3: Friends

Question 4

Read through Section 2.4 of the lecture notes (pages 10–13), trying out the R code and making sure you can install the `ncldata` package and read-in the `movies` database. Don't rush through this – take time now, in the practical, to make sure you understand the basics of dataframe manipulation in R! Once you're done, try Example 2.2 from the notes, which is set out below for your convenience:

Follow the steps at the start of Section 2.4.2 of the lecture notes to load the `ncldata` package and to read in the `movies` dataframe. Then answer the following questions. If you encounter problems installing the package, then you can use the following code to read in the `movies` dataframe direct from my MAS2906 personal webpage – do not copy-and-paste this code, but make sure you type it exactly as it appears below!

```
1 movies = read.table("https://www.mas.ncl.ac.uk/~nlf8/teaching/mas2906/movies.
  txt", header=TRUE)
```

- (a) How many people submitted votes for the films "*10 Things I Hate About You*", "*Midnight in the Garden of Good and Evil*" and "*Zoolander*"?
- (b) How many films are classified as "Action"? How many are classified as "Romance"? How many "Short" films are there in the database?

- (c) How many films were made between the Years 1970 and 2000 (inclusive)?
- (d) How many movies are longer than two hours in duration?
- (e) How many movies made since the year 2005 (not including 2005) are longer than three hours in duration?

Question 5

Load and explore the `music` dataset from the `ncldata` package by typing the following lines of code (Note: This will only work if you have correctly installed the `ncldata` package and read the package into your working RStudio space using the `library` command; see Section 2.4.2 of the lecture notes. Alternatively, you can install the dataset directly from my MAS2906 personal webpage using the same line of code as given in question 4, replacing `movies` with `music`):

```
1 data(music)
2 # What's this dataset all about? Let's find out
3 ?music
```

- (a) How many tracks do we have data for, and on how many variables?
- (b) How many tracks do we have from *The Beatles* in this dataframe? What about *Lady GaGa*?
- (c) How many tracks have a tempo of at least 130 beats per minute?
- (d) Of the tracks with a tempo of at least 130 beats per minute, how many have been released since the year 2000 (including the year 2000)?
- (e) What is the most popular key for tracks in this database?