

Introduction to R

Jason Matthiopoulos 2011

Overview: R is a software environment for statistical analysis, modelling and simulation. It is also a fully-fledged programming language. A prerequisite to analysing your data is to import it into R from an external file. You will also need to know how to organise your data within R and how to save your results to file for future work. Of particular importance, is the ability to export data in formats that are readable by other types of software such as a spreadsheet.

1. Introductory References

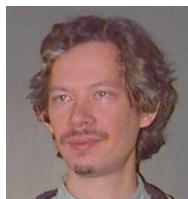
There are several quick-start guides for R. Before you attempt this practical, I suggest that you read "Chapter 0: Introduction to R" from the coursebook. If you feel that you need still more reading material, you will find references at the end of that document.



R logo



Douglas Bates



Peter Dalgaard



Martin Maechler



Ross Ihaka



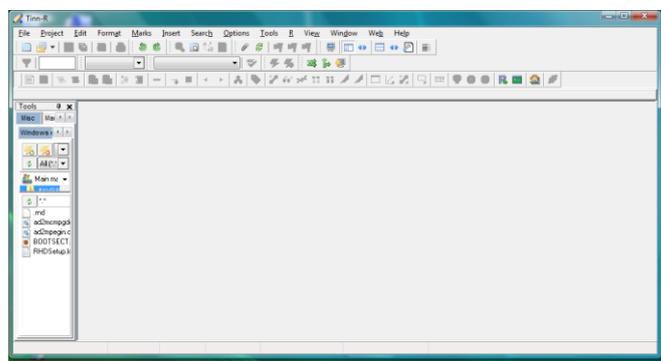
Robert Gentleman

The R logo and mugshots of a few members of the R core developers' team (pictures are property of their owners).

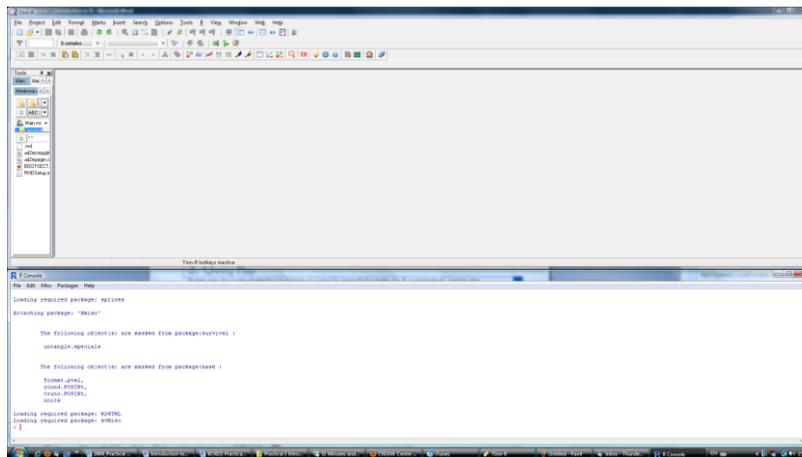
R commands are sometimes difficult to remember. You will often find yourselves remembering that there is a command that does a particular job but being unable to remember what it is called. So, make sure you keep handy a printout of Tom Short's, R-reference card. You can download it directly from <http://cran.r-project.org/doc/contrib/Short-refcard.pdf>.

2. Start Your Engines

For this entire course we will be using R, Tinn-R and Excel. Although you can use R interactively, without Tinn-R, it is a better idea to always launch R from within Tinn-R, so that if you need to use the text-editing facilities of Tinn-R, it is there, as part of your current R session. So, launch Tinn-R (either from the start menu, or using a quick-start icon). The Tinn-R application looks something like this:



Look for the button . Once pressed, this launches R and arranges the two windows in an expanded, non-overlapping configuration. This looks a bit like this:



Feel free to ignore the start-up information printed in the R console window. Now, go to the R Console window and type `1+1`. The characters appear automatically next to the active prompt `>`. Press "return" to see the result of this calculation. This brief interaction should look a bit like this

```
> 1+1
[1] 2
```

To achieve the same result from within Tinn-R: Click with your mouse in the Tinn-R window. Go to the "File" drop-down menu. Select "New". This starts a new, blank sheet for you to write your code in. Type `1+1` in that sheet. You can type anywhere, like you would in a word-processor, but ideally start at the top left. The next thing is to send your bit of code to R. You can send your entire document, by pressing the button . Alternatively, you can send selected bits by first highlighting them with your mouse and then pressing the button .

3. Getting help

There are an overwhelming number of ways to search for help on R commands, some are accessible through the "Help" menu in the R console, others directly by typing in the console. You will initially find R help files somewhat... unhelpful. The way they present information takes some getting used to but, thankfully, extracting what you need becomes easier with practice. One way to look for help is to use the command `help()`. Let's assume that we want to know how to use the addition sign "+" in R. Typing `help("+")`, or alternatively, `?"+` in the R console, brings up the relevant help file on arithmetic operators. The sections of all help files tend to be similar. There is usually a section summarising the purpose of a command ("Description"), its correct syntax ("Usage"), its inputs ("Arguments") and outputs ("Value"). Most importantly, towards the bottom, all help files have examples of usage ("Examples") and hyperlinks to relevant commands ("See also").

If you are unsure of the name of the command for which you want help but you vaguely remember that it pertains to or contains some keyword, try typing `help.search("keyword")`, or alternatively `??keyword`. This performs a search through the help files for related commands.

4. R ingredients

You will find a section of R ingredients, such as this one, in every practical handout. Using these ingredients in addition to the ones presented in previous practicals, you should be able to perform all the tasks needed to complete the practical.

- *Assignments*: Numerical and mathematical assignments are made by the arrow symbol (e.g. `n<-10` sets the variable `n` to the value 10). In R syntax, the equality sign “=” is reserved for specifying options within commands and for testing equality “==”. If you wish, you can make assignments using “=” but it makes the directionality of the assignment difficult to guess. The thing to remember about assignments is that they are not the same as mathematical equations. Hence, the job of the assignment `a<-a+1` is to increment the value of `a` (whatever that happens to be) by one. Mathematically, the equation $a = a + 1$ is nonsense.
- *Concatenation*: This term describes the action of bundling things together. Say, for example, that you have three temperature readings in degrees Celsius, 10,12 and 15 that you would like to combine into a single list called `temps`. The concatenation command `c()` can be used to do this as follows: `temps<-c(10,12,15)`. Now, assume that you have two additional measurements, 9 & 12, that you would like to add to your existing data set. You can do this as follows `temps<-c(temps, 9,12)`.
- *Importing files*: We will be using the command `read.table()`. This requires you to specify a file path for the data to be imported. Examples of its usage can be found in §0.9 of Matthiopoulos.
- *Simple mathematical operations*: All the basic operations of arithmetic are available. Addition (+), subtraction (-), multiplication (*), division (/), powers (^). Powers can be used to estimate roots (e.g. $\sqrt{4}$ can be found by typing `4^0.5`) but there is also a square root command (`sqrt()`).
- *Vectorised operations*: One of the great thing about R is the flexibility with which it can operate on entire data sets. For example, if you have a list, called `temps`, of temperature readings measured in Celsius, and wanted to convert these to Fahrenheit and store them in a new list called `tempsF`, you can do it without dealing with each measurement individually. The general conversion formula is $[^{\circ}\text{F}] = [^{\circ}\text{C}] \times \frac{9}{5} + 32$. In R, you would simply type `tempsF<-temps*9/5+32`.
- *Data frames*: Data frames are structures for storing and manipulating data. Make sure you have read §0.10 in Matthiopoulos before you attempt the tasks in the following section.
- *Merging data frames*: If you have two data frames (say `x`, `y`) that you would like to combine into one, you can do it by typing `merge(x,y)`. R will use the column names of the dataframes to make sure that similar data are listed under the same headings. If some column entries are present in one data frame but missing from the other, R will tend to create a minimal dataset by keeping only the rows with all the column entries. You can by-pass this and create a maximal data set by specifying `merge(x,y,all=TRUE)`. This results in `NA` entries wherever the information was not previously available.
- *Adding columns or rows to data frames*: If you have a list of values (say, `x`) that has the same length as the row or column of a data frame `dat`. You can append it as a row by typing `dat<-rbind(dat,x)` or, as a column by typing `dat<-cbind(dat,x)`.
- *Commenting your code*: On any given line, R will ignore everything that follows the character `#`. Use this to make sure that the task performed by each line of code is appropriately commented. See examples in §0.5 in Matthiopoulos.

5. Practical tasks

We consider two field seasons of a tagging study. In 2008 and 2009, highly visible marks were placed on several individuals from a species of terrestrial animals. This was followed up by spatially uniform re-sighting effort during the three months after tagging. Records were made of the ID of the re-sighted animals, their behavioural state and the type of habitat where they were seen. Additionally, in 2009, records were kept of the ambient temperature in degrees Celsius.

- 1■ Read carefully the material in sections 1-4 above.
- 2■ Read through the tasks in this section.
- 3■ Read through the assessment components in the next section before you carry out any tasks
- 4■ Create a folder somewhere on your hard disc called "Practical_1". Write down the path to that folder (e.g. "C:\My Documents\BL5021\Practical_1").
- 5■ Copy the Excel file "Sight.xls" into that folder. During the practical, there will also be several memory sticks available with this file. Just ask for one if you need it.
- 6■ Open the file in Excel. You will find that it contains two separate worksheets called Sight2008 and Sight2009. Save each of them as Tab-delimited files (see §0.9 in Matthiopoulos) with the names "Sight2008.txt" and "Sight2009.txt".
- 7■ In Tinn-R, write a line of code to import "Sight2008.txt" into a data frame called `dat08`. Run this line from Tinn-R and then, in the R Console, type `dat08` to see the contents of the data frame. Make sure that the data have been imported successfully by checking back at the Excel spreadsheet.
- 8■ Do the same to import "Sight2009.txt" into the data frame `dat09`.
- 9■ In Tinn-R write a line of code to merge the two data frames into one called `dat`.
- 10■ In Tinn-R, write code to extract the temperature column from the data frame `dat` and convert it to Fahrenheit. Store these results into a list called `tempsF`.
- 11■ In Tinn-R, write a line of code to append `tempsF` to `dat` as a new column.
- 12■ In Tinn-R, write a line of code to export `dat` into a new file on your hard-disk (call it "SightE.xls") that can be read by Excel. Check that this has worked by opening the file with Excel.

6. Assessment

For your report, print the Tinn-R code that you used to carry out the above tasks. Make sure that your code is fully commented. This should take up no more than half a side of A4 (probably a lot less).