Chapter 3

Getting R, getting started

3.1 Finding R

Either Google CRAN R (the **C**omprehensive **R** **A**rchive **N**etwork) or use

http://cran.r-project.org/

which is where to begin. You are directed to pages that tell you how to install R on various platforms, and more information, if needed, is provided in the FAQs. R is updated frequently. Documentation on R is comprehensive and much of it is free. It is well worth looking at what is available in CRAN at an early stage

3.2 Data entry

3.2.1 General

For other than very small data sets it is best to import data from an external source. This can be done in different ways. Although not the preferred method of the developers of R, many users may find it simplest, if starting from scratch, to create and import an Excel file.

Create the data file; it is assumed below that headers naming the columns are given. Spaces in headers must be avoided (and if other illegal characters are used an error message in R will inform you). Next, highlight the data you want to import; copy it (to the clipboard) and go into R. You name the data file when reading it into R; for illustration the data for *A Night at the Opera* (1935), submitted by James Cutting, is used (see next section for more detail), which will be named Night_at_the_Opera in R. Type

```
Night_at_the_Opera <- read.table(file = "clipboard", header = T)
```

and type Night_at_the_Opera to see the result. Here <- is the assigment operator, and note that clipboard must be enclosed in double inverted commas or an error message results. If data are missing R requires that the offending cell be filled with NA. To use read.table a rectangular table of data is expected. Commands are preceded by the R prompt >. [1]

It is best to keep headers informative but short (in writing-up an analysis or captioning a figure a key can always be provided). Headers beginning with a number are allowed, but column names in R will not be quite as you expect.

The writers of the R manual for data import/export prefer you to write the Excel file to a Tab or comma-separated file and use read.delim or read.csv. Use ?read.table in R for the documentation.

[1] You will notice, by the way, that R is directive- or command-driven rather than menu-driven. That is, you have to tell it what to do by writing things down. People (i.e. students I've taught over the years) can find this a confusing and challenging idea. Anyone who has ever word-processed a sentence, paying attention to spelling and syntax, is more than equipped to meet this challenge.

3.2.2 Using the Cinemetrics database

Some familiarity with the Cinemetrics data base, `http://www.cinemetrics.lv/database.php`, is assumed here. Firstly, films recorded in the 'Simple' mode are used.

Within the database select *A Night at the Opera*; show the raw data, select it, and copy into an Excel file. The following protocol will be adopted here of editing the three column headers to read `Id`, `SL` and `Cut`. This should leave you with an Excel file of three columns with these headers. Copy the file, go into `R` and read the data in as described above.

The shot length, `SL`, and cut point, `cut`, data are recorded in deci-seconds. For these notes it is convenient both to convert these to seconds and have a separate variable for shot lengths only. Either of the following commands will do this.

```
SL.Night_at_the_Opera <- Night_at_the_Opera$SL/10
SL.Night_at_the_Opera <- Night_at_the_Opera[,2]/10
```

The first of these picks out the variable of interest by name, the second by column number. The first version is generally neater. Conversion to seconds can be done before reading the data into `R`, in which case omit the division by 10.

Secondly, for films recorded in advanced mode, there is a fourth column corresponding to shot type that it is assumed will be named `Type`. For illustration Charles O'Brien's submission of *Lights of New York* (1928) is used. The `Type` variable identifies whether the shot is 'action', 'dialog' or a title, 'exp.tit'. If the information on the type is not needed just proceed as already described, naming the table of data `LONY` and creating a shot length variable `SL.LONY`, say. If the type variable is wanted create it using one of the following.

```
SL.LONY.Type <- LONY$Type
SL.LONY.Type <- LONY[,4]
```

How this might be used is taken a little further in Section 4.4

3.3 Packages

Packages are collections of *functions* that, together with *arguments* provided to them, control the analyses undertaken. Some packages are loaded automatically with `R` and the functions in them are immediately accessible. Others are bundled with `R` and must be loaded before use. Yet others need to be imported before they can be loaded.

For illustration the bundled `MASS` package, associated with the book *Modern Applied Statistics with S: Fourth Edition* (Venables and Ripley, 2002), is used. In the following code everything that precedes # should be typed; everything after is my comment. If writing your own code # can be used either to annotate it or comment out parts you don't want to use for any particular analysis.

```
library(MASS)           # loads MASS
library(help = MASS)    # lists available functions
?truehist               # information on the function truehist in MASS
```

Importing packages is done in two stages. In `R` from the *Packages* menu select *Set CRAN mirror* to choose a site to import from then, from the same menu, select *Install package(s)* and the select the package you want to import. The package then needs to be loaded before use with `library(packagename)` as above.

There are an overwhelming number of packages available. If you know what you want, all well and good. If not, Google is invaluable if the search terms are 'R' and the name of the technique of interest.

3.4 Reading

This is enough to get started, which is all that is being attempted here, Section 4.3 in the next chapter provides an introduction to writing functions in R, at a very basic level, which is useful to know about early. Subsequent chapters provide code to go with the analyses illustrated and introduce useful things like the labelling of graphs, in an incremental fashion.

There is, as noted, a vast amount of help around, much of it free. The CRAN site lists over a hundred books, in various languages and at all levels, many on specialised topics. I found the Venables and Ripley (2002) book and earlier editions invaluable when teaching myself about R and the related and earlier commercial package S-Plus. The title of the book reflects its earlier origins, but it works with R and I continue to find it invaluable. It is not an introductory text for non-statisticians and Dalgaard (2008) would be more suitable as an elementary-level introduction. A strength of R is its graphical capabilities, and Murrell (2006) collects together useful information on this, not always easily found elsewhere.