Appendix A

Getting R, getting started

A.1 Finding R

Either google CRAN R (the Comprehensive R Archive Network) 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.

A.2 Data entry

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. The data file is named in R and for illustration the data of Table B.1 is used, which will be named tubb in R. Type

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

and type tubb to see the result. Here <- is the assigment operator, and note that clipboard must be enclosed in quotation marks 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 >.

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 may 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. For most of the data used in these notes row names were not supplied, and the default then produces case numbers as row names. For alternatives to this see ?read.table in R.

Once entered into R you may need to work with subsets of the data. Selecting subsets is discussed in Section 2.6.

A.3 Packages

Packages are collections of *functions* that, together with *arguments* provided to them, control the analyses undertaken[1]. 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* (Venables and Ripley, 2002), is used. In the following code comments follow #.

```
library(MASS)           # loads MASS
library(help = MASS)    # lists available functions
?kde2d                  # information on the function kde2
kde2d                   # prints source code for kde2
```

A list of available functions is provided by library(help = MASS) and ? lists the documentation for the function specified, in the example kde2d for 2-dimensional kernel density estimation. It is sometimes useful to look at the code to see what is going on. It can be edited to suit individual requirements. The '?' facility does not always work as you wish. For example the function biplot is contained in the stats package which is automatically loaded. If, however, ?biplot is typed then the following is obtained.

```
function (x, ...)
```

[1]There are numerous examples in the text. Code can be typed in directly but it is often more convenient to construct a function first. Further detail is given in Section 3.2.1.

```
UseMethod("biplot")
<environment: namespace:stats>
```

That is, the code is invisible; typing `stats:::biplot.default` should reveal the code. More generally, this requires the package where the function is located and the function name to be specified as indicated.

There are an enormous number of user-contributed packages available. These need to be imported before they can be used and this 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)`. The package then needs to be loaded, using the `library` function as shown in the example above.