Help on the R software

This little tutorial gives you some tips to learn how to use R and to be able to do the exercises offered in this course.For a better understanding of the software, you can see the book *R for Statistics*[1].

Contents

---

[1]R for Statistics. (2012). Cornillon *et al.*, 305 pages, CRC/PRESS

1   Installing R

Simply go to the CRAN website at `http://cran.r-project.org/` and install the version of R best suited to your computer's operating system. For comprehensive documentation on the installation process, visit `http://cran.r-project.org/doc/manuals`.

You can use the free and open-source integrated development environment (IDE) for R called Rstudio: `www.rstudio.com`.

2   Opening and Closing R

In Windows and Mac OS X, look for R in Program Files. In Linux, simply type R in a command window. To close R, either use the menu or type the command **q()**. When exiting the program, the software will ask if you want to save the work session. This means that the next time that you open R, you will be able to retrieve all the objects you have constructed. Most of the time it is preferable to save the instructions rather than all the objects. To do so, write and save the instructions in a text file, known as a script.

3   The Command Prompt

When R is launched, a window opens and waits for some instructions with a prompt `>`. R can be used immediately as a calculator, for example,

```
> 2+3.2
[1] 5.2
```

The `[1]` indicates that the first (and only) coordinate of the resulting vector is `5.2`. When R is waiting for the next instruction, the command prompt becomes `+`. For example, if you type

```
> 1 -
```

R will wait for the second part of the subtraction and the command prompt will be `+`. To substract 2 from 1, type

```
+ 2
[1] -1
```

Generally, either a `)` or `"` has been forgotten. Simply suggest one (or more) bracket(s) or speech mark(s) to end the command.

4 Attribution, Objects, and Function

R makes calculations using functions (which will be written in bold):

```
> sqrt(2)
[1] 1.414214
```

Results can be attributed to more or less complex objects using <- or =. Thus we create vector x with integer values from 3 to 7:

```
> x <- 3:7
> x
[1] 3 4 5 6 7
```

The content can be displayed simply by typing the name of the object.

5 Selection

To construct the vector y with coordinates 2 and 4 of vector x, type

```
> y <- x[c(2,4)]
> y
[1] 4 6
```

Select columns 1 and 3 of a matrix m and rows 2 and 5 respectively using

```
> m <- matrix(1:15,ncol=3)   #creates the matrix
> m[,c(1,3)]                  #selects columns 1 and 3
> m[c(2,5),]                  #selects rows 2 and 5
```

In combination, rows 4 and 2 of columns 2 and 3 are obtained as follows:

```
> m[c(4,2),c(2,3)]
     [,1] [,2]
[1,]    9   14
[2,]    7   12
```

We can also assign objects of different kinds to lists:

```
> mylist <- list(vector=x,mat=m)
```

The component mat from the list mylist is selected using

```
> mylist$mat
```

6 Aids for a function

All commands can be stopped using the shortcut `Ctrl + c` (or the STOP icon). It is possible to reuse preceding commands using the up or down arrows. Additionally, to obtain help for the **mean** function, type

> **help**(**mean**)

7 Installing a Package

In this section we download a package and install it. Packages need only be installed once. Once they are installed, simply call the package using the **library** function each time you open an R session:

> **library**(package.name)

Packages are available on the CRAN network (http://cran.r-project.org/). A number of mirror sites, or exact copies of the CRAN site, are available. To install a package from CRAN, simply run the **install.packages** function:

> **install.packages**(dependencies=TRUE)

Then choose the nearest mirror to you and select the package to install, such as the FactoMineR package. You then simply need to load it in order to use it in an R session:

> **library**(FactoMineR)

8 Graphical User Interface

8.1 The Rcmdr Package

The graphic interface R Commander is available in the package Rcmdr. This interface means that R can be used simply, via a scroll-down menu. The aim of this package is also to help people to learn to use the software as it also provides the lines of code for the corresponding analyses. As with any package, it only needs to be installed once, and then loaded when needed, using:

> **library**(Rcmdr)

8.2 The package RcmdrPlugin.FactoMineR

A graphic interface is also available and can be installed as a user interface in the interface for the Rcmdr package. To do so, the package RcmdrPlugin.FactoMineR must be installed once. Then, every time one wishes to use the FactoMineR scroll-down menu, Rcmdr must be loaded. To do so, click on `Tools → Load Rcmdr`

`plug-in(s) ....` Choose the FactoMineR plug-in from the list; Rcmdr must then be restarted to take the new plug-in into account. If you want to have the FactoMineR scroll-down menu available in Rcmdr all the time, you can do: `Tools →  Tools ==> Save Rcmdr options`. The use of the scroll-down menu for PCA is detailed below.

Importing Data: The Rcmdr scroll-down menu offers a number of formats for importing data. When the file is in text format (.txt, .csv), it is impossible to specify that the first column contain the individuals' identities (which is often the case in data analysis). It is therefore preferable to import using the FactoMineR menu
`FactoMineR → Import data from txt file`
Click on `Row names in the first column` (if the names of the individuals are present in the first column), and then specify the column separator (field separator) and the decimal separator.

8.3   The package Factoshiny

The Factoshiny package also offers a graphical interface which enables users to construct interactive graphs directly from FactoMineR output. This interface can be used on the one hand to set parameters for the method, by choosing supplementary quantitative and qualitative variables in PCA for example, and on the other hand to improve graphs by altering font size, elements to be labelled, the identification of individuals according to the categories of a categorical variable, etc.

9   Graphs

The functions **print**, **plot** and **summary** are generic functions, they give results that are unique depending on the class of the object to which they are applied. For example, **plot.PCA**, **plot.CA**, **plot.MCA**, can all be applied using the generic instruction **plot**. Depending on the class of the object (output resulting from PCA, CA, MCA), the outputs or graphs will be specific to these analyses. To access assistance for the function of writing a given object, use, for example PCA: `help("plot.PCA")`.

10   Statistical Analysis

Different functions can be used to conduct various statistical analyses. The easiest way to know how to use a function is to refer to its help section by typing **help**`(myfunction)`. The help section demonstrates how the function is defined, all of its arguments and, finally, usage examples that can be directly copied and pasted into R. If the outputs of the function are assigned to an object named `result`, it

is possible to list all the objects present in `result` using **names**(`result`). For example, after a principal component analysis on a dataset `X`,

```
> resultat <- PCA(X)
> names(resultat)
 [1]  "eig"  "var"  "ind"  "svd"  "call"
```

The `result$ind` command gives access to the object containing the results on the individuals. The function **summary** can be used to summarise the main results.