Installing and running R

R on the web

The R website:

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

here you can find information on the software, download the current version “R-2.9.2” (released on 2009-08-24), packages, tutorials and manuals.

Installing and running R

To download R first choose a Comprehensive R Archive Network (CRAN) mirror:

A CRAN is a network of ftp and web servers around the world that store identical, up-to-date versions of code and documentation for R) -> Use the CRAN mirror nearest to you to minimize network load

Download the R version suitable to your operative system, following the installation instructions.

Windows:

the executable file is auto-installing, just follow the instructions. Create an icon on the Desktop!
Double click on it to run R

Linux:

Download the R-2.9.1.tar.gz file and unpack it with
tar xvfz R-*x.y.z*.tar.gz
(or gzip -dc R-*x.y.z*.tar.gz | tar xvf -)
./configure
make
make install
Once installed, to run the program just type R on shell.

It will appear a prompt “>”, meaning that R is waiting for you to input a command.

Basic Commands

R commands

R commands consists of either *expressions* or *assignments*.

An *expression* is evaluated, printed on the terminal and its value is "*lost*". No memory of the output is kept.

Example:

```
> objects()
character(0)
> (5+3/2)*0.1
[1] 0.65
> objects()
character(0)
>
```

An *assignment* also evaluate an expression but passes the value to a variable and the result is not automatically printed.

To assign a value to a variable it is used the simbol "<-" pointing to the variable which receives the value:

Example

```
> objects()
character(0)
> a <- (5+3/2)*0.1
> objects()
[1] "a"
>
```

Commands are separeted by a semi-colon(“;”) or by a new line.

If a command is not complete at the end of the line, R will give a different prompt, “+” by default, at any subsequent lines untill the command is syntactically complete

Comments have to be preceeded by a hashmark (“#”): everything following “#” to the end of the line is ignored.

Recall and correction of previous commands

R provides a mechanism for recalling and re-executing previous commands.

Vertical arrow keys on the keyboard of the computer can be used to scroll forward and backward the list of commands (history) you input.

Once you find the command you are looking for you can modify it using the keys.

Executing commands from a file

Commands can be typed on the R console or they can be stored in an external file, say “commands.r”

The list of commands in “commands.r” can be executed at any time in a R session typing:

source(“commands.r”)

The output is printed on the console

Printing output to a file

Sometimes it is useful to divert the output from the console to an external file.

Using the command:

```
sink("output.txt")
```

all the subsequent output will be printed to the external file "output.txt".

The command:

```
sink()
```

will restore the output to the console again.

R objects

The entities that R creates and manipulates are known as *objects*. They may be:

- variables
- vectors
- arrays of numbers
- character strings
- functions

or more general structures built from these components.

Once objects are created, the R commands:

```
objects()
```

or alternatively:

```
ls()
```

can be used to display the names of the objects which are currently stored within R.

The collection of objects currently stored is called the *workspace*.

To remove objects, use the function rm():

```
rm(x)            to remove x
rm(list=ls())    to remove all the objects of the workspace
```

Saving Data

When you quit an R Session with the command

q()

you will be asked if you want to save the workspace: typing “y”
two files will be created in the working directory:

- **“.RData”** containg all the objects created during the session
- “**.Rhistory**” containing the list of commands you typed during the session

The same files are created if you type

save.image(); savehistory()

before quitting the session

It is possible to save the workspace in an other directory in a specific file:

save.image(file="mydirectory/myfile.RData")

or

save(file="mydirectory/myfile.RData", list=ls())

To access the objects in "myfile.RData" in the following R sessions you can type on the console:

load("mydirectory/myfile.RData")

and the objects in myfile.RData will be added in the current workspace.

The same for saving the list of commands in an other directory in a specific file:
savehistory(file="mydirectory/myfile.Rhistory")

To load the commands saved in myfile.Rhistory type:
loadhistory(file="mydirectory/myfile.Rhistory")

As well you can access the history typing:
history() to display the last 25 commands or
history(n) to display the last n commands

NB: .Rhistory files can be opened by text editors

Managing directories

During a R session it can be necessary to get information on the working directory or to change the current diretory.

Use:

- **getwd()** to get the working directory
- **list.files()** or **dir()** to see the files in the working directory
- **setwd("path/mydir")** to set the working directory

R help

R has an excellent help.

- The inbuilt help facility (like “man” in Linux)

  It can be accessed from the command line using:

  > *?function_name*

  or

  > *help(function_name)*

  The help window will appear and you will find information on the function, its arguments, examples, correlated topics and further details.

- The HTML format help

  The command:

  > help.start()

  will launch a Web browser that allows the help pages to be browsed with hyperlinks.

Further help…

The command **example(***function_name***)** runs the examples contained in the help pages.

Example:

```
>example(array)


array> dim(as.array(letters))
[1] 26


array> array(1:3, c(2,4)) # recycle 1:3 "2 2/3 times"
     [,1] [,2] [,3] [,4]
[1,]    1    3    2    1
[2,]    2    1    3    2
```

Now you able to:

- download,
- install,
- and run R
- do the first exercise!