

How to use R (short R tutorial)

Last Updated Thursday, 07 February 2008

R is an comprehensive statistical and graphical programming language. Unfortunately, it also has a steep learning curve. However, my goal with this website and the manual is to get a non-R-person able to use tnet quickly and correctly. This page will cover how to use R in general whereas the individual pages for each function deals them in detail.

There are many beginner guides on how to use R available. Below are a sample of them (this tutorial is mainly based on Quick-R):

- R wiki: <http://wiki.r-project.org/rwiki/doku.php>
- Quick-R: <http://www.statmethods.net/>
- An introduction to R: <http://www.stat.berkeley.edu/~spector/R.pdf>

General

R is an interpreted language. This means that you can enter code directly in R without having the need to compile the code you write. This allows the user to write code interactively. Commands can be entered one by one at the command prompt, or in a file. tnet is simply a collection of command lines stored in files.

There is a wide variety of data types, including vectors (numerical, character, logical), matrices, dataframes, and lists. Most functionality is provided through built-in and user-created functions and all data objects are kept in memory during an interactive session. Basic functions are available by default. Other functions, such as tnet, are contained in packages that can be attached to a current session as needed.

Importing data

R can import data form a range of programmes. The easiest is often a normal ASCII (or Notepad)-file. You can read these files using the read.table-command, e.g.:`data <- read.table("c:/folder/file_to_read.txt")`

This line will read the file "file_to_read.txt" in the folder "c:\folder", and store it as the R-object "data". Do note that the \ has become /.

You can also enter data by hand, e.g. a vector of 7 numbers:`data <- c(1, 2, 4, 7, 9, 6, 1)`

To read data from Excel, Stata, SPSS, SAS, Systat, see Quick-R: <http://www.statmethods.net/input/importingdata.html>

Viewing and manipulating data

To view an R-object, simply write the name of the object, e.g.:`data`

```
[1] 1 2 4 7 9 6 1
```

This will print the object on the screen. The [1] signals that the first number on the row of numbers is the first number in the object. If there were too many numbers to fit on one line, R would split it onto multiple lines, and the [?] would signal where you are in the sequence.

To change the object, use the object directly, e.g.:`data+1`

```
[1] 2 3 5 8 10 7 2
```

This would display the object, and add 1 to each of the numbers. However, it would not save the changed object. To do this, you need to save it onto itself:

```
data <- data+1
data
[1] 2 3 5 8 10 7 2
```

The <- means that what is written on the right, will become what is on the left. You can also use = but out of clarity, I prefer to use the <- .

To learn more about data management, see Quick-R: <http://www.statmethods.net/management/index.html>

Descriptive Statistics

To see descriptive statistics about a vector, you can use the summary-function, e.g.:`summary(data)`
 Min. 1st Qu. Median Mean 3rd Qu. Max.

2.000 2.500 5.000 5.286 7.500 10.000

You can also use this function on the results from a regression analysis, and then it will give you more information about the regression.

To learn more about statistics, see Quick-R: <http://www.statmethods.net/stats/index.html>

Graphs

R can also easily create graphs. The main function is called plot, e.g.:`plot(x, y)`

To add a regression line (the plot-window must stay open):`abline(lm(y~x))`

You can also add a title (the plot-window must stay open):`title("Regression of Y on X")`

When you are done with adding elements, and editing, you can save the graph by right-clicking on it. Alternatively, you can output the graph to a file instead of the screen. To read more about this and graphs in general, see Quick-R: <http://www.statmethods.net/graphs/creating.html>

Conclusion

R is a difficult to use in the begining, but very useful.