

1 Introduction and basics

1.1 What is R and why should you learn how to use it?

Used in its simplest way, R is a program to perform statistical analyses and produce graphical representations of data. However, R is also a programming language and an environment for statistical computing, allowing you to design and implement statistical procedures tailored to your own needs or simulate data.

R is open source software. This means that it is free, but also that it is developed further by a large international community of scientists and programmers. Accordingly, R tends to be at the forefront of statistical computing and is quickly becoming the standard statistical software in the life sciences and elsewhere.

Finally, R is the foundation of Bioconductor, a similar open-source project focussed on the development of bioinformatic analyses. Bioconductor rose to prominence when it became the standard environment for the analysis of microarray gene expression data, but it has maintained and extended this position with the advent of new technologies and the integration with different types of data.

For the reasons above, we believe that learning R will give you a solid foundation for a research career in the life sciences. It will also give you a lot of generic skills by making you more numerate and computer-savvy, and exposing you to some simple principles of computer programming.

All information about R can be found here www.r-project.org/. The Bioconductor project's website is here www.bioconductor.org/.

1.2 Getting, installing and starting R

R can be downloaded from the R project website which also contains links to installation instructions.

- Mac OS X : <http://cran.r-project.org/bin/macosx/>
- Windows : <http://cran.r-project.org/bin/windows/base/>
- Linux : <http://cran.r-project.org/bin/linux/>

Once installed R can be launched on Windows and Mac by running the R application. On Linux and Mac, R can also be run from a command window by typing R at the command prompt. To see which working directory R is running in you can do `getwd()` and also change working directories using the `setwd` function.

1.3 Communicating with R

You are probably used to interact with your computer or smartphone by clicking on icons and buttons. This, unfortunately, is not how R works. In order to tell R what you want it to do, you have to type commands. Here are some easy examples. You can ask R to open a webpage in your browser that provides some help on the program by typing

```
> help.start()
```

You can also use R as a calculator and type in some calculations, such as

```
> 2+2
[1] 4
> 8^2
[1] 64
> 2^0.5
[1] 1.414214
```

In this course document, each line of input will begin with the prompt symbol, `>`. Do not type this symbol, it just represents the start of the input line.

Finally, if you want to quit R, you can type

```
> q()
```

Lines of output will be shown indented and without a prompt symbol.

1.4 A few tips

- Don't repeat typing commands you already used: the upper arrow will bring back older commands that you already ran.
- If you start a command and do not close it (for example because of a missing parenthesis), you may end up somewhat stuck, with R not responding to what you are doing. Use the Escape key or Ctrl+C to get out of this if this happens.
- As you trial and error commands, your console window will quickly fill up. It can then become difficult to find the lines that worked or produced important output. A solution is to keep a text editor window open next to your R console, into which you copy commands and output that are important and that you want to keep. These files are then also an archive of useful commands that you can come back to later, for example when you need to perform a particular operation but can't quite remember how you did it last time.