



Introduction to R

Oswaldo Anacleto
Genetics and Genomics, Roslin Institute
osvaldo.anacleto@roslin.ed.ac.uk

27th January, 2016



THE UNIVERSITY *of* EDINBURGH



From the R website (www.r-project.org):

R is a language and environment for statistical computing and graphics

- Implementation of the S language (developed by AT&T)
- provides a wide variety of statistical and graphical techniques
- It's highly extensible (via packages)
- Why is R so popular?
 - ▶ It's free (open source)
 - ▶ There are many people contributing to it

R and its packages are installed by downloading them from The Comprehensive R Archive Network (CRAN):

<https://cran.r-project.org/mirrors.html>

R operates in two modes: **interactive** and **batch**

- **interactive mode:** commands are typed in in R, and then the software interprets and executes the command.
- **batch mode:** does not require interaction with the user. Very useful for long simulations and to automate processes (programs which must be run periodically)

We will only use R in the interactive mode during this course. This is done through the R Console in **R Graphical User interface** (RGui).

One of the simplest possible tasks in R is to enter an arithmetic expression and receive a result:

For example, by typing in the R console

```
2+3
```

R returns the value 5

- In R, values are assigned to **variables** through the assignment operator “ \leftarrow ” (or “ $=$ ”)
- **variables** are **names** that can be used to represent values

For example, the value of $2+3$ is assigned to the variable x by typing in

```
x <- 2+3
```

To show the value of the variable x , you can simply type in x in the R console

A simple way to store data in R is by using **vectors**
vectors can be defined using the **construct** (or **concatenate**) `c(...)`

example:

```
height <- c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
weight <- c(60, 72, 57, 90, 95, 72)
bmi <- weight/height**2 #body mass index
bmi
```

in R, text to the right of the hash key # are ignored
(very useful for including comments in R code)

- Rhas many functions for mathematical operations, statistical data analysis/modelling and for making graphs.

A simple example:

```
log(100) # returns the log of 100
```

- Function **arguments** are values passed to functions. They are specified within parenthesis after the function name.
- The `help` function can be used to obtain information about a function:

```
help(log)
```

alternatively,

```
?log
```

A crucial task in R programming is writing functions.

- A function is a group of instructions that
 - ▶ takes inputs (arguments)
 - ▶ uses them to compute values
 - ▶ returns a result

Functions can be created in R using `function` Example

```
someStats <- function(x)
{
  meanData <- mean(x)
  sdData <- sqrt(var(weight))
  answer<-c(meanData, sdData)
  return(answer)
}
# applying the function to the height vector
someStats(height)
```


- R uses **packages** to store groups of related functions (or datasets)
- Only when a package is loaded are its contents available
- Some (base) packages are loaded automatically when R is started. Use the function `search()` to see loaded packages in the current R session
- there are options in the “package” tab on Rgui to install and load packages
- Also, the functions `install.packages()` and `require()` are used to install and load packaged respectively

The R script `blabla.R` installs all packages required for the course.

- There are thousands of contributed packages for R ($\approx 8,000$ as of May 2016), written by many different authors
- These packages can:
 - ▶ implement statistical methods
 - ▶ provide access to data or hardware
 - ▶ complement textbooks or papers
- most packages are available for download on CRAN:
(<https://cran.r-project.org/>)
- there are also other repositories such as Bioconductor
(<https://www.bioconductor.org>) and Omegahat
(<http://www.omegahat.net/>)

CRAN task views provide a directory of links to R packages sorted by topic:

<https://cloud.r-project.org/web/views/>

R can be also used in conjunction with other graphical user interfaces (GUIs) or integrated development environments (IDEs):

- RStudio (<https://www.rstudio.com/>)
- Tinn-R (<https://sourceforge.net/projects/tinn-r/>)
- R commander - Rcmdr (<http://www.rcommander.com/>)
- ESS - Emacs speaks statistics (<http://ess.r-project.org/>)
- Deducer (<http://www.deducer.org/>)

These interfaces can provide a more user-friendly environment for using R

primary source:

```
help.start()
```

it is worth searching for answers to specific questions on these websites:

- Stack overflow:

<http://stackoverflow.com/questions/tagged/r>

- R-mailing list:

<https://stat.ethz.ch/mailman/listinfo/r-help>

(do read their posting guide before sending a question!)

- An introduction to R:
<https://cran.r-project.org/doc/manuals/r-release/R-intro.html>
- Advanced R, by Hadley Wickham
(<http://adv-r.had.co.nz/>)
- Matloff, Norman. The art of R programming: A tour of statistical software design. No Starch Press, 2011.

R and Statistical Analysis:

- Dalgaard, Peter. Introductory statistics with R. Springer, 2008.
- Venables, W. N., & Ripley, B. D. (2013). Modern applied statistics with S. Springer.
- Resources to help you learn and use R. UCLA: Statistical Consulting Group. (<http://www.ats.ucla.edu/stat/r/>)
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (with applications in R). Springer.