A (short) introduction to R

Emmanuelle Becker, University of Rennes 1

Contents

1 Introduction 1

2 Discovery of R and RStudio 1

2.1 Description of the interface 1

2.2 Working directory 2

2.3 Saving your work 2

2.4 First steps: R as a calculator 2

2.5 Help 3

3 Objects ans types 3

3.1 Three elementary types (or modes) 3

3.2 The operators 3

4 Vectors 4

4.1 Usual functions 4

4.2 Operations on vectors 4

4.3 Manipulating vectors 5

1 Introduction

What is R?

• R is a software for statistics and graphs;

• R is also a programming language, that’s what makes it special;

• R is free, available on all platforms (Windows, Mac, Linux ...);

• R is an open-source software, everyone can collaborate and enrich the software with new libraries.

2 Discovery of R and RStudio

2.1 Description of the interface

Launch RStudio. The window that opens is divided into 4 windows:

1. At the top left, the window of script, in which one writes the programs (files with an extension .R), comments (after the symbol #) ... When one wishes to execute a command line, or a block of command lines, one selects the entries to execute and sends them to the console with the key combination Ctrl + Entrer.
2. At the bottom right, the window of the console. In the console, we type the commands after the symbol >, they are executed after typing Enter. In the console, the symbol > means that R is ready to work. If this symbol is not displayed, it is because R has not finished calculating the result of the previous command.

3. At the top right, a visualization of the environment creates: defined variables, loaded tables of data etc..

4. At the bottom right, a window with different tabs to view the help, the graphics etc ...

If you’re working with R without RStudio, only the console is available. In this case, to visualize the environment, one uses the command `ls()`.

2.2 Working directory

R uses a working directory from which it will read the data files, save scripts, environments or figures. To know your working directory, use the `getwd()` command. To change your working directory, use the `setwd("path")` command. If you are under RStudio, you can do this via menu Session > Set Working Directory > Choose Directory.

2.3 Saving your work

When working with R, we have the possibility to save:

- The script with commands, comments etc: it’s the simplest way to work because by saving the script and the initial data, we can always reproduce results (if we did not use non-deterministic methods or random, of course). The extension of R scripts is simply .R.

- The script with the commands + all the environment of work. This solution has a major disadvantage: it is larger, but in return is very useful if the calculations are very long because we can directly store the result without having to re-run these calculations. The extension of R environments is .Rdata.

You are supposed to save your work regularly without being reminded. Thank you!

2.4 First steps: R as a calculator

In this part, we propose to enter your commands in the script window (top left) and then execute them. Observe what is going on in your Environment (window at the top right).

Type a command:
> 2 + 5.627
Now observe the environment: nothing changed.

Create an object x and affect it a value (operator = or <-):
> x <- 3
To see its content:
> x

You can modify the content of a variable and/or to affect it the result of a calculus:
> x <- 5 + 2 > x
> y <- x + 3
> y

To call a function:
Exercise 1: Construct an object called \( p \) that contains the value 3.78, and then computes the result of 
\[
\frac{4\sqrt{p} + 2}{5}
\]

The result should be 1.92333...

2.5 Help

All classical functions have a help page (in English). How to obtain help?

If you know the exact name of a function:

\[ ?\text{log} \]

The use of \( ? \) should be a reflex...

Otherwise:

\[ ??\text{logarithm} \]

The documentation is written in English, R is case sensitive...

Exercise 2: Perform the following operation: 4850 / 26. Display the result with only 3 decimals. You will need a rounding function...

3 Objects ans types

R manipulates only 3 types of elementary objects: numbers or numeric, text or character, and booleans or logical.

3.1 Three elementary types (or modes)

Numeric:
\[ x \leftarrow 3.69 \]

Character string (always enclosed in quotation marks):
\[ s1 \leftarrow "3.69" \]
\[ s2 \leftarrow "pierre" \]

Logical, TRUE (or T) / FALSE (or F):
\[ b1 \leftarrow \text{FALSE} \]
\[ b2 \leftarrow \text{T} \]
\[ b3 \leftarrow x > 2 \]

3.2 The operators

Arithmetic:
\[ + - * / ^ \text{ (power, or **)}, \%\% \text{ (modulo)} \]
as in a calculator...

Comparisons:
\[ <> <> == \text{(equality)} != \text{(different)} \]
returns a logical (or Boolean) value.
Logic:

! & | also return a logical value.

Exercise 3: Consider we have a triangle ABC, with $|AB| = 3$, $|AB| = 4$ and $|BC| = 5$. Imagine an expression that returns TRUE if and only if the triangle ABC is rectangle in A.

4 Vectors

A vector is a list of elements of the same type (only numeric, or only character...).

4.1 Usual functions

Create a vector with the function `c()` (read "combine"):

```r
> v1 <- c(3,8,25,4,9,1.5,71,20,5,0.02)
> v1
```

Size of a vector:

```r
> length(v1)
> length(x)
```

Generating suites:

```r
> v2 <- 1:15
> v3 <- seq(0.1,0.5,by=0.1)
> v4 <- seq(50,90,length=100)
```

Exercise 4: In a vector called v5, put a suite starting from 1 to 100 with all odd numbers (1,3,5,7,...)

4.2 Operations on vectors

We find the same operators as on the values, many functions are "generalized" on the vectors. Test with the following examples:

```r
> 5*v1
> 5*v1 + 10
> v1 > 5
```

To access one or more elements of a vector, the syntax is `vect[i]` to access the ith element of vect (at positions i). Warning: the numbering starts at 1!

```r
> v1[3]
> v1[5:8]
```

We can easily extract values greater than 5 with the command:

```r
> v1[v1>5]
```

Exercise 5: Extract from v1 all the values that are less than 10.

Exercise 6: Extract from v1 all the values that are multiple of 2.

Exercise 7: Extract from v1 all the values that are multiple of 2 and less than 10.
To delete values from a vector, the syntax is original:

```r
> v1
> v1[-3]
> v1
```

Explain what happens.

**Exercise 8:** Write a command that removes from v2 all items that are multiple of 2.

Function `sum()`:

```r
> sum(v1)
> sum(v1>5)
```

Try to explain the result.

### 4.3 Manipulating vectors

Some statistical functions that apply to a data vector: `mean()`, `min()`, `max()`, `var()`, `sd()`, `summary()`, `head()`.

What are they useful for? Try each one with one of your vectors, and see the help page for each command.

- `mean()`: 
- `min()`: 
- `max()`: 
- `var()`: 
- `sd()`: 
- `summary()`: 
- `head()`: 

**Exercise 9:** Calculate the mean of v1 using only the functions `sum` and `length`.

**Exercise 10:** For each value of the vector, computes its difference with the mean (with one unique R command).

Variance can be estimated from a sample using the following formula (μ is the mean of the sample):

\[
\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{\mu})^2
\]

**Exercise 11 (difficult):** Calculate the variance of v1 using only the functions `sum` and `length`.

You can check your calculation by comparing with `var(v1)`.

The end!

Next lesson: manipulation of data tables with R.