R Lesson 1

Federico Reali

2024-10-04

Federico Reali, PhD

Email: federico.reali@unitn.it

Senior researcher @Fondazione COSBI (www.cosbi.eu)

Mathematician by training, PhD in Mathematical modelling for systems biology.

Focus on Diabetes and Insulin resistance, Rare and Neglected Diseases.

Why R

R is a free software environment for statistical computing and graphics.
It compiles and runs on a wide variety of UNIX platforms, Windows, and MacOS.
The R Foundation

R Overview

R is a versatile programming language, popular among statisticians. There are numerous online resources for installation and usage. It is open-source and available on platforms like Windows, Mac, and Linux.

Steps to Install R

1. Install R

2. Install an interface (e.g., RStudio)
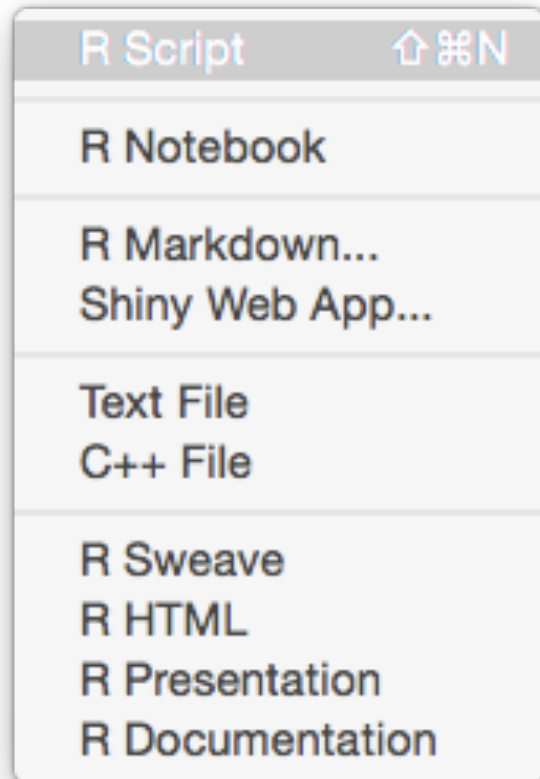
RStudio Interface

RStudio consists of three main panels: - **Console** - **Environment & History** - **Files, Plots, Packages, Help, Viewer**
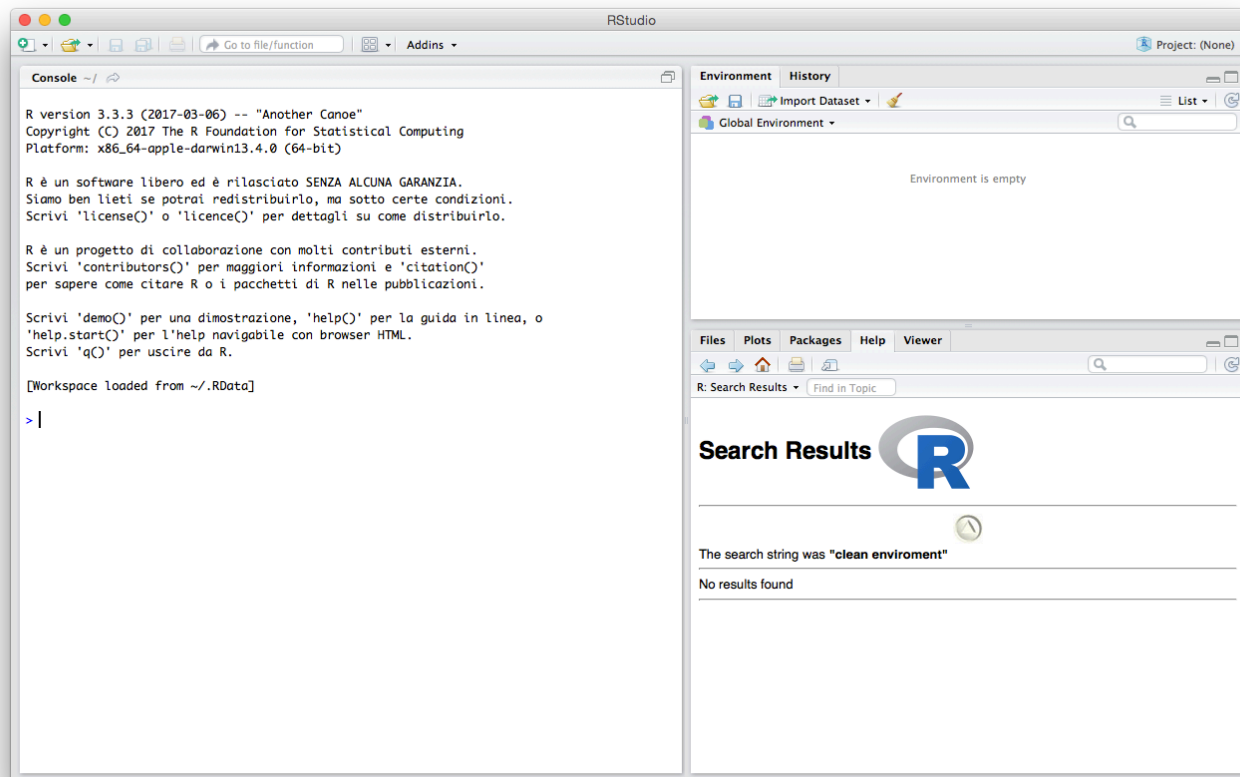
Running Scripts

Create scripts to store and run code. Open a new script:

`File -> New File -> R Script`

R Script	⇧⌘N
R Notebook	
R Markdown...	
Shiny Web App...	
Text File	
C++ File	
R Sweave	
R HTML	
R Presentation	
R Documentation	

Useful Links

- R Intro Manual

- R Notes (Italian)

- R Commands by Topic (Italian)

R basic syntax

Folders (where are we now?)

When working with datasets and files, it is fundamental to ensure that you're working in the correct directory. We can use the following R command to check which folder we are currently in:

```
1  getwd()
```

This command returns the path of the current working directory. To set the working directory to a different location, use the command `setwd()`:

```
1  setwd('your-path/StatLearn/')
```

You can navigate directories using relative paths (e.g., `./` for the current folder, `../` for the parent folder).

Arithmetical Operations

Using standard notation, R can solve basic arithmetic:

```
1  2+2
2  2-2
3  2*2
4  2/2
```

R supports division using `%/%` for integer division and `%%` for the remainder. You can also compute powers with `^` and roots using `sqrt()`.

Exponential, logarithms, and trigonometric functions

- Exponential: `exp()`

- Logarithms: `log()`, `log10()`, `log2()`

- Trigonometric: `sin()`, `cos()`, `tan()`

The functions expect angles in radians. You can use functions like `sinpi()` to work with multiples of pi.

Assignation, memory, and types

Variables can be assigned using arrows **<-** or **=**:

```
1  x <- 6
2  x
```

R automatically allocates memory based on the type of data assigned (e.g., numeric, character, logical).

Logical and relational operators

R supports logical comparisons:

```
1  6 > 10
2  6 <= 10
```

Logical operators: & (and), | (or), ! (not), and xor().

Vectors

Create vectors using the `c()` function:

```r
1  y <- c(1,2,3,4)
2  y[1]
```

R operates on vectors element-wise by default. You can also transpose vectors with `t()`.

Matrices

Matrices can be created with `cbind()` and `rbind()`:

```
1  a <- cbind(c(1, 2, 3), c(4, 5, 6))
2  dim(a)
```

You can access elements using square brackets `a[2, 3]`.

Strings

R handles character strings using double quotes:

```
1  names <- c("Francesco", "Sofia", "Alessandro")
2  names[1]
```

Use `paste()` to concatenate strings.

Lists and data frames

- **Lists** are collections of objects.

- **Data frames** are tabular data structures that allow mixed types.

Example:

```
1  L3 <- LETTERS[1:3]
2  fac <- sample(L3, 10, replace = TRUE)
3  data.frame(x = 1, y = 1:10, fac = fac)
```

Tidy Data

Tidy datasets have a structure where each variable is a column, each observation is a row. This helps with data manipulation and modeling.

Read more about Tidy Data by Hadley Wickham.

Packages

R comes with core functions, but additional functionalities can be added using packages:

```
1  install.packages('hflights')
2  library(hflights)
```

You can also explore installed packages with `library()`.

Let's play with the data

R comes with several datasets, especially in packages like *MASS* or *hflights*. You can practice data exploration with these datasets.

In addition, we can download data from website, for example:

```
1  df_from_web <- read.csv('https://web.stanford.edu/class/archive/cs/cs109/cs
```