Lab 01 - Intro to R

Jaime Montana

9/1/2021

Presentation

- *Short presentation:* Who I am?
- Objective of the labs
- Contact information: The most efficient way is to drop me an email to jaimem.montana@gmail.com. We can set up office hours. But I would prefer that you send me the question to the email. If the question is relevant for everyone I will use the moodle announcement instead of a personal email.
- How to ask a good question related to code: use the following guide.

Before starting. . .

Some General tips

1. Read Cal Newport's books about studying or read his blog
2. Think strategically:
   - This might not seem useful now, but is an important tool to excel in other core subjects.
   - Quantitative skills are highly valuated in the labor market (independent of the field).
   - Will give you the basis for understand new methods.
3. This will improve your abstract reasoning. Practice will also give you discipline which is very important in your current path.

Approach

- Reference to external resources (please take time and read the books)
- Help me find resources
- Collaborate (Dropbox, Slack, . . .)

What will this course teach you?

- Learn a new vocabulary that will give you a broader comprehension of analysis in other disciplines.
- Understand and make you a more critical consumer of statistical data (presentations, media, social media)
- Learn new methods of thinking and learn new tools to solve questions, and to provide quantitative support for your arguments.
- Even if the scope is not learning to program, we will learn to use a statistical software.

Some history

- Data collection - Ancient Babylonians recorded their crop yields on clay tablets, ancient Egiptian pharaons recorder their wealth on stone walls (First Census: Pharaon Amasis, 1557 BC). The word origin comes from latin status (same root as state). The collection of data have been associated to account for the power of a state (resources, military, population, wealth, . . .).
- Data analysis - Tabulation was common untill the XVIII century. It was untill the 1800 that there were significant advances in the field.
  1. William Playfair (1759 - 1823): develop the histogram to visualize data.
  2. Sir Francis Galton (1822 - 1911): Correlation and regression to the mean. "Discover" fingerprints.
  3. Karl Pearson (1857 - 1936): Standard deviation. Formalization of the correlation coeficient. Distribution.

. . . Gosset, Weldon, Tukey. . .

- Methodology. Data collection, analysis and interpretation.

Why R?

“R is a language and environment for statistical computing and graphics.” http://www.r-project.org/about.html

First you need to install R, which is the “engine” under which work RStudio. R is an interpreted language, meaning that all commands typed on the keyboard are directly executed without requiring to build a complete program. When a command is typed you will see the result in the console, or you can store the results in the memory of the program.

Why R?

- R is a free software, easy to install and runs in multiple OS.
- A lot of documentation and forums. Excellent documentation on packages.
- Very active community which allow to use other people codes and projects.
- Great visualizations thanks to ggplot or plotly packages.
- If you understand the logic behind R you will get into every statistical software very easy.
- Everything seems hard at the beginning. Just try and ask.

Why Rstudio?

R by itself is “user unfriendly” and “ugly”. That’s why we are going to use R accompanied with R studio.

From the description on his web page “RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history..”. Rstudio is a user interface that is more friendly and allow us to see the memory, code and execution so that working with R is easier.

If you have problems installing/reinstating you can follow this video: MacOS and Win10

RStudio basics

- Let's open RStudio and let's see the interface.
- create a R file.
- Create a Rmd file.

RMarkdown basics

Rmarkdown (Rmarkdown cheat sheet) will help you to write reports (also automatically) and to document your analysis. There are some simple rules:

- use '*' and '_' to make text italic and bold.
  - single *italics*
  - double **bold**
- use '#' to create a title, '##' a subtitle
- use '$' to insert an equation. Single is an **inline** equation,, while double insert a new line for the equation.
- You can also display code, inline or in block. Inline code uses back ticks: `mean(c(1,2))`. For a new line use the option `echo=TRUE` in order to see the code chunk in the document, and after the result.

```
mean(c(2,5,8))
```

```
## [1] 5
```

RMarkdown exercise

- “Random” link
    - create a paragraph. Emphasise a point.
    - create a section
    - Write an equation (see Latex math equations PDF for reference)

R rules

There is also an R cheat sheet for reference.

- use '#' to create a comment. **Everything** in the right of a 'hash' will be omited by the compiler.
- Comment all the things you (try to) do!
- Different kind of objects: numeric, character, dates, logical,..
- I can arrange in ordered structures such elements: a vector, a matrix, a list . . .
- The variables could define *groupings*, classifications, or characteristics. In R we call them *factors*. _ We make use of functions . . .

R in console vs. R in editor

You can use the *Console as a calculator*. But if you close and open a new session all the codes and work will not be there. It is a best practice to use the text editor, where we can pass the commands to the console easily.

```
1+1
```

```
## [1] 2
```

```
2*(10-2)*4
```

```
## [1] 64
```

```
8^(1/3)
```

```
## [1] 2
```

R in console vs. R in editor (functions)

```r
sqrt(25) # square root
```

```
## [1] 5
```

```r
exp(2)   # e^2
```

```
## [1] 7.389056
```

```r
8^(1/3)  # exponential
```

```
## [1] 2
```

Variables and environment

We can assign values to variables that are stored in memory.

“A variable provides us with named storage that our programs can manipulate. A variable in R can store an atomic vector, group of atomic vectors or a combination of many R objects. A valid variable name consists of letters, numbers and the dot or underline characters. The variable name starts with a letter or the dot not followed by a number.” R-variables

```
x    <- c(2,5,16,3.2)
y    <- c("2","5","16","3.2")
x_1  <- c(TRUE, FALSE, FALSE)

?class()
```

Functions and help

Functions are characterized to be followed by a parenthesis that encloses the arguments (inputs).

- Calculate the mean of 3 and 16.
- Search help and identify the usage, the arguments of the function,
- Calculate the mean of the vector `c(3, NA, 16)`
- calculate the standard deviation of 5,2,3.

Working directory and loading data

The working directory directs to a path in your computer/infrastructure.

```
getwd()
```

```
## [1] "/home/jaime/Dropbox/Catolica - Postdoc/Courses/BRM/
```

```
#setwd("C:/path/to/files/") # change the path to wd

#load("ceosal2.RData")   # If you have saved in wd
#load("C:/path/to/ceosal2.RData") # else
```