A collection of notes on R, Git and statistics

Olalla Diaz-Yañez

2018-03-03

Contents

Chapter 1

A work in progress

This is a collection of notes (a book?) with useful resources and instructions related to R, RStudio, Git and Statistics with R. Right now it mainly works as a personal repository. But I hope that litle by litle it will have more meaninful content with detailed instructions to more complex concepts and examples.

Chapter 2

Introduction

2.1 What is R

R is a system for statistical computation and graphics. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in script files. To be able to use and understand R you will also need to know some basic concepts that all programming depends on. But do not be scared!

Here you can read a longer deffinition of what is R.

R vs. Python

I really like this comparison of python and R. The idea is that R is Batman and Python is Superman. Batman (R) does better detective work, has a more developed intelligence, or in other words Batmand is more brain than muscles. On the other hand Superman (Python) has muscle power and super strength, you could consider him more elegant, but in general words is more muscles than brain.

FUN FACT: The "Python" programming language name derived from the series Monty Python's Flying Circus.

2.1.1 Why R

There are several reason why R:

- it´s free
- it´s well-documented and has an amazing user community
- it runs almost everywhere
- it has a large user base among researchers, data scientists, companies
- it has a extensive library of packages helping to solve different tasks
- it´s not a black box

The best way to learn a tool is to use it for something useful, for example analyze data. Thats why this is the tool prefered in our courses. The goal it is not to master the tool or actually teach you R but to have the enough knowledge to be able to find your way to suceed in your goals and to have a basic knowledge that will aloow you to explore independtly and find the solutions that you need.

2.2 What is RStudio

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, debugging and workspace management.

2.2.1 Why RStudio

My friend usually sais that we have not come to this world to suffer, using Rstudio instead of just R makes your life easier, so it will avoid some unnecesary suffering. There is nothing wrong of just using R, without RStudio, and actually some people prefer to learn R without RStudio, not me. Some of the reasons why I prefer to use RStudio are:

- window docking (all necessary things in one window)
- full-featured text editor
- tab-completion of filenames, function names and arguments (you do not need to remember everything)
- Rmarkdown and knitr integration

2.3 What is Git

Git is a **version control system**. Probably the best ever description og Git is whaty XXX wrote: "it is as the "Track Changes" features from Microsoft Word on steroids". It was created to help groups of developers to deal with big and complex projects. For a data science user, it is basically a clever way to ovoid having hundred "final version" files as described here.

It is both benefitial when working alone as you can delete that not-so-clever code that you wrote and never used, without been scared, as if future you need that piced of code you will be able to go back and take it. But Git benefits increase exponinetally when you include collaborators in the equation, use git it is a smart way to collaborate and to be up-to-date with each others work and at the same time have a version control of your and others people work. Some people think that hthe Git-pain is only worth it when collaborating, but even in that case, I am sure you are going to work with others in some point, and avoid taken that into consideration from the beginning will mean a delay in the implementation of it in your workflow and a higher pain than just Git-pain.

2.4 What is GitHub

A way of hosting your work online.

2.5 What it is in for me?

Maybe you are still wondering: what it is really in for me? I just wanted to do my statistical anaylisis and be done with it, do really all the gains possibly justify the inevitable pain of start using R, Rstudio, GitHub and Git?…

My personal experience is that doing things from the beginning with the correct aproach, although painful, may avoid a bigger pain in the future. As soon as you get into your best workflow easier for you would be to do things right and realised early about mistakes. Of course your workflow won't be a static thing, you will continiouly learn new aproaches and techniques that will improve the way you do things. Saying that I also think that one thing is what it is the best workflow for each of us as individuals (if you like to write

your essays on a paper and then in a word document, thats fine for me), but a different thing is what is the best way to collaborate and work with others. In the first case you chose, in the second case chosing should be always on what is the best for the team-work, meaning higher productivity, less chances of errors, easier collaboration etc. And yes, you are going to work with others most of the times. There is were Git and RStudio are going to make your life easier!

Chapter 3

Applications

Some *significant* applications of the tools presented

3.1 Example one

3.2 Example two

Chapter 4

Sessions

4.1 Session 0

1. Install R and R studio

- Install pre-compiled binary of R for your operating system: https://cloud.r-project.org
- Install Preview version RStudio Desktop: https://www.rstudio.com/products/rstudio/download/preview/

If you have a pre-existing installation of R and/or RStudio, it is highly recommend that you reinstall both. You will face more dificulties if you run an old softwared version. If you upgrade R, you will need to update any packages you have installed.

2. Test it

Launch RStudio, you should see something simmilar to this but a bit emptier as you have not written anythign yet. Put your cursor in the pane labelled Console, which is where you interact with the live R process. In console, write something like: 2 + 1 and press return and you should get a 3. If you get a 3, you've succeeded in the installation of R and RStudio, congrats!

Some extra resources:

https://cran.r-project.org/doc/manuals/R-admin.html

https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f

https://cran.r-project.org/doc/FAQ/R-FAQ.html

4.2 Session 1

4.2.1 Learning objectives

- Learn what is programming
- Learn what is R
- Learn the basics of RStudio
- Learn the basics of R:
  - Variable assigment
  - Basic data types
  - Vectors
  - Data frames

4.2.2 Contents

Basics of R:

- Program
- Language: Not compiled, simple sintaxys
- R "flow":
  - Variables, data, functions, results, etc, are stored in the active memory of the computer in the form of objects which have a name.
  - You can do actions on these objects with operators (arithmetic, logical, comparison, . . .) and functions (which are themselves objects).

Practice

- Using Rstudio
  - Open a project
  - Organize your folder (rstudio projet / Code / Data / Figures /) **Super tip of the day!**
  - Console
  - Files
  - Script
  - Keyboard shortcuts
- Variable assigment: the "assign" operator
  - concept overwiting a variable (s in the active memory, not the data on the disk)
  - Note that R is case sensitive!
  - "#" is used for comments
- Basic data types
  - Decimals values like 4.5 are called numerics.
  - Natural numbers like 4 are called integers. Integers are also numerics.
  - Boolean values (TRUE or FALSE) are called logical.
  - Text (or string) values are called characters. The quotation marks indicate that is a character.
- Vectors
  - create vectors
  - name a vector
  - select elements from the vector
  - compare different vectors
  - combine vectors
- Data frames
  - creating a data frame
  - quick look to the data frame: structure, rownames, number of columns, number of rows, summary,
  - select data fram elements
  - subset in a data frame
  - ordering
  - sorting

The code of this session can be found here

4.2.3 Carry on learning

Data camp (still) free basic r course

Base R Cheat Sheet

4.3 Session 2

4.3.1 Learning objectives

- Learn the basics of R:
  - Variable assigment (review)
  - Basic data types (review)
  - Vectors (review)
  - Data frames (review & new)
  - Lists
  - Matrices
  - Factors
- Basic math functions
- Basic visualizations of the data
- Basic statistics
- Using Libraries
- File path
- Working directory

4.3.2 Contents

*Data frames + Understanding a data frame + Count rows and columns + Add rows and columns + Subset a data frame + list of variables + name of the columns + Vector functions (sort, counts of values, unique values)

- Lists
  - Lists, as opposed to vectors, can hold components of different types
  - create a list
  - list subsetting
  - name lists
- Matrices
  - create matrices and to understand how you can do basic computations with them.
- Factors
  - create, subset and compare
- Maths functions
  - Maximun value
  - Minimun value
  - Mean value
  - Median
  - Variance
  - Standart deviation
  - Correlation
  - Round values
- Basic visualizations of the data
  - Ploting
  - histograms
- Basic statistics
  - linear model
  - summary of the linear model
- Using Libraries
- File path
- Working directory

Chapter 5

Final Words

We have finished a nice book.