Forewords

> *I want to do something, not learn how to do everything.*
> John Carroll, "Minimal manual"

This manual is the reference for the classes of the course **Data Science using R** by Federico Roscioli. The real goal of this course is not to teach you the R language, but to teach you the basic concepts in order for you to be able to use and explore data using R.

R is a language and, as such, it should be practiced as much as possible in order to learn it by heart. The R language, however, is only a coding language, so you are not able to speak *Rish* with your friends nor write a letter using this language. The only option you have is to use it as much as possible in your work and **exercise, exercise, exercise**. Using R in your work, especially if you were accustomed to use another programs (SPSS or STATA) or if this is your first time coding, may be slow and tedious at the beginning. You will pass a lot of your time on Stack Overflow and on the R Help looking for a solution for your problem. The computer is going to seem to judge your mistakes harshly. You will learn the word "syntax error" a lot. And, if you're like me, I think back to the first time that I was programming, you will grow to kind of hate the creature that lives inside the computer because you will think that the creature does not like you. You will think the creature is value judging your code.

But the thing that you are going to have to learn is that syntax error doesn't mean that R thinks you're bad. It is not a judgment of your ability as a programmer, it is not a judgment of your potential to be a programmer. Syntax error means R is lost. It just does not have really friendly words when it says it's lost, it just goes syntax error, which literally is exactly what's happening. Your syntax is not something that R understands.

So, I want you to never loose your effort. It is not because you're a bad person, it is not because you're never going to figure it out. You will figure it out and you will get it. You only need to build some experience first. It is like when you learn to cycle, at the beginning you have to fall a lot, but in the end you will love cycling. I remember when I was first learning to program, I would be like, lost, lost, lost, lost, lost, oh wow, I love this. And I expect that many of you will go through that exact same feeling.

After this course you will be able to clearly understand the R code that someone else wrote. This means that you will be able to explore different and more complex solutions for your problems, explore new packages and become a real programmer!

Many people think of R as a statistics system. We prefer to think of it as an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern tools and statistics. R and its libraries implement various statistical and graphical techniques, including linear and nonlinear modeling, classical statistical tests, spatial and time-series analysis, classification, clustering, and others. R's data structures include vectors, matrices, arrays, data frames (similar to tables in a relational database) and lists.

The big difference that people coming from other programming tools will encounter is that in R, functional programming is much more important than object-oriented programming, because you typically solve complex problems by decomposing them into simple functions, not simple objects. But we will understand more using it.

Please, have fun, be curious, and exercise a lot!

All the contents in this book are available on my website (write me in order to ask for the password). I created an ad-hoc **R playground**. The R playground allows the student to exercise in order to reinforce her knowledge of R and data analysis. Exercises are fundamental in order to fix the knowledge acquired in class. The platform is structured in the same way as this manual, so for you it will be easy to understand what to do after each chapter/lesson.

Installation

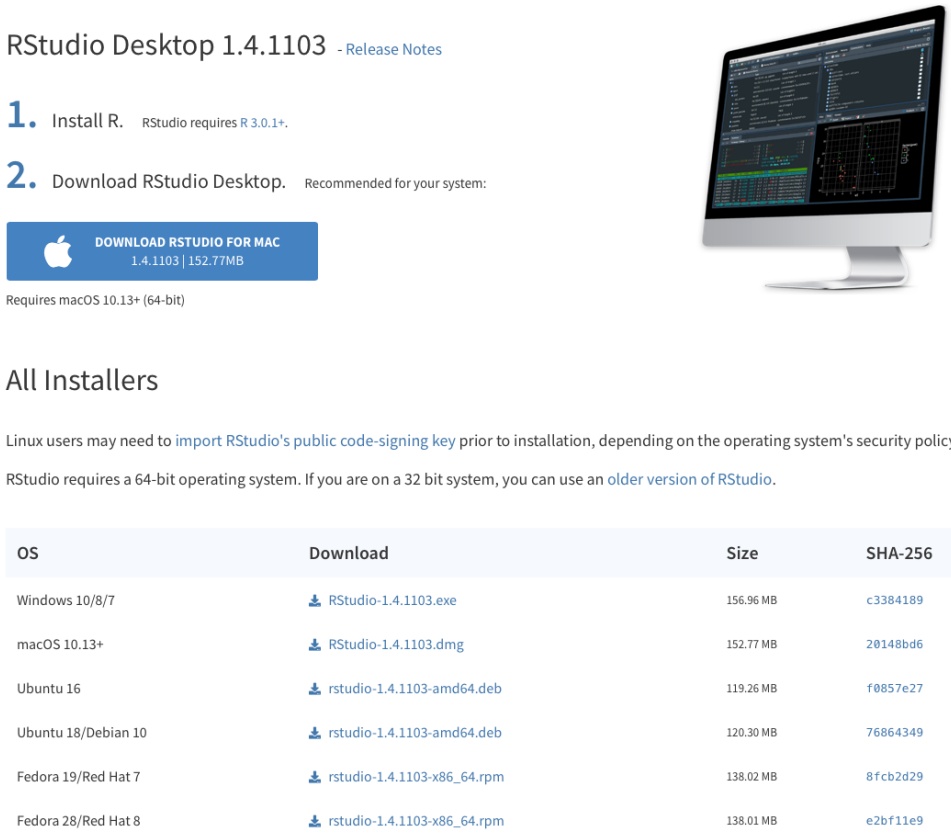
Introductory activities

First of all you need to install the software that will allow you to work with R. You will work with R-Studio, which is an interface software that runs on top of R and allows you to have some facilitation and suggestions while working. Additional to R and R-Studio you will need to install some packages. The packages are extension of the software that bring in additional functions and/or data.

Follow all the 9 steps below:

1. Download the R installer from CRAN.

2. Run the installer keeping the default settings. If you do not have admin rights on your computer, please ask you IT Support to give you full permissions to the R directories. Otherwise you will not be able to install packages afterwards.
3. Download the R-Studio installer from R-Studio.



RStudio Desktop 1.4.1103 - Release Notes

1. Install R. RStudio requires R 3.0.1+.

2. Download RStudio Desktop. Recommended for your system:

DOWNLOAD RSTUDIO FOR MAC
1.4.1103 | 152.77MB

Requires macOS 10.13+ (64-bit)

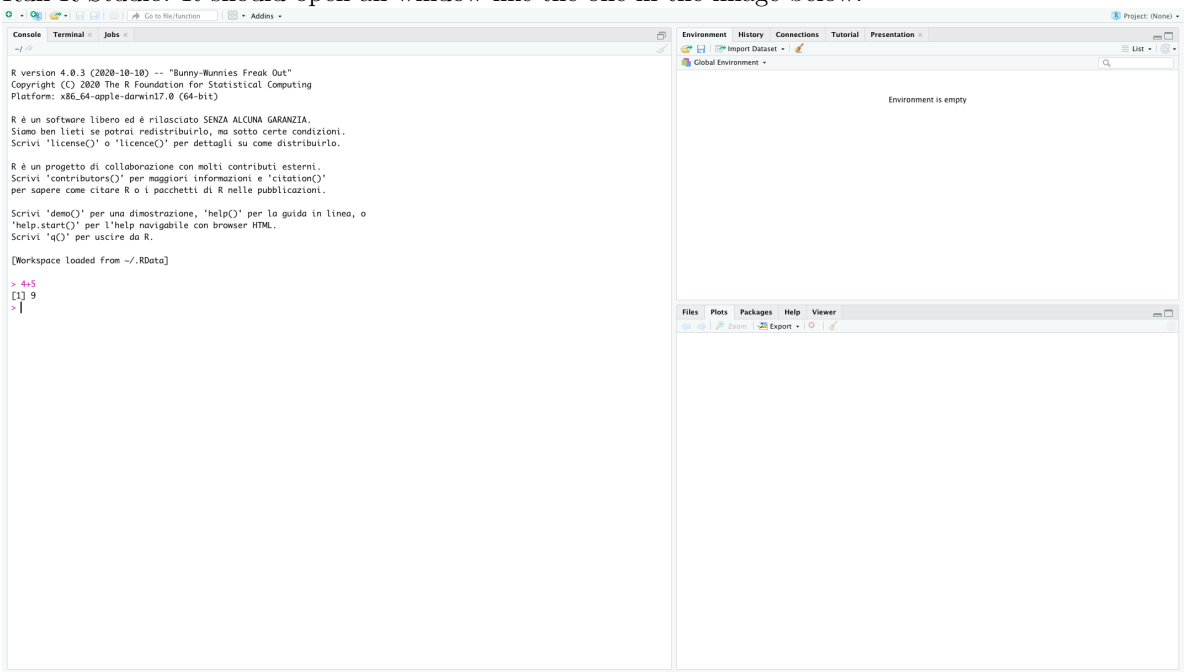
All Installers

Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.

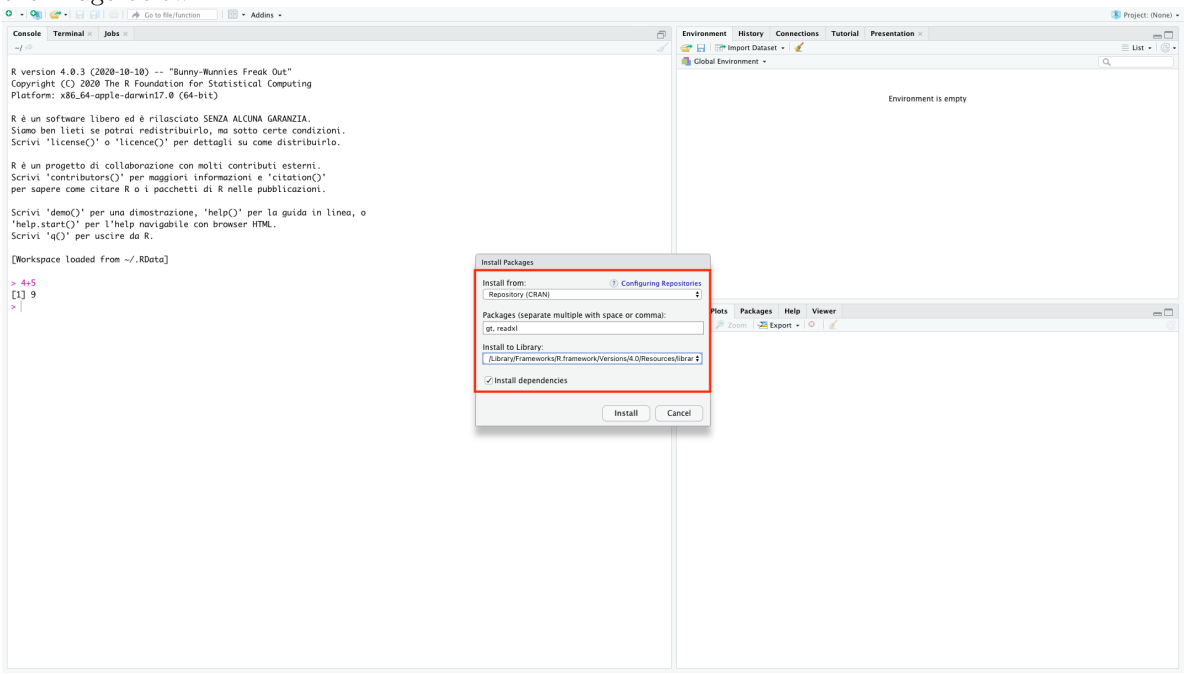
RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an older version of RStudio.

OS	Download	Size	SHA-256
Windows 10/8/7	RStudio-1.4.1103.exe	156.96 MB	c3384189
macOS 10.13+	RStudio-1.4.1103.dmg	152.77 MB	20148bd6
Ubuntu 16	rstudio-1.4.1103-amd64.deb	119.26 MB	f0857e27
Ubuntu 18/Debian 10	rstudio-1.4.1103-amd64.deb	120.30 MB	76864349
Fedora 19/Red Hat 7	rstudio-1.4.1103-x86_64.rpm	138.02 MB	8fcb2d29
Fedora 28/Red Hat 8	rstudio-1.4.1103-x86_64.rpm	138.01 MB	e2bf11e9
Debian 9	rstudio-1.4.1103-amd64.deb	120.45 MB	4a4d159c

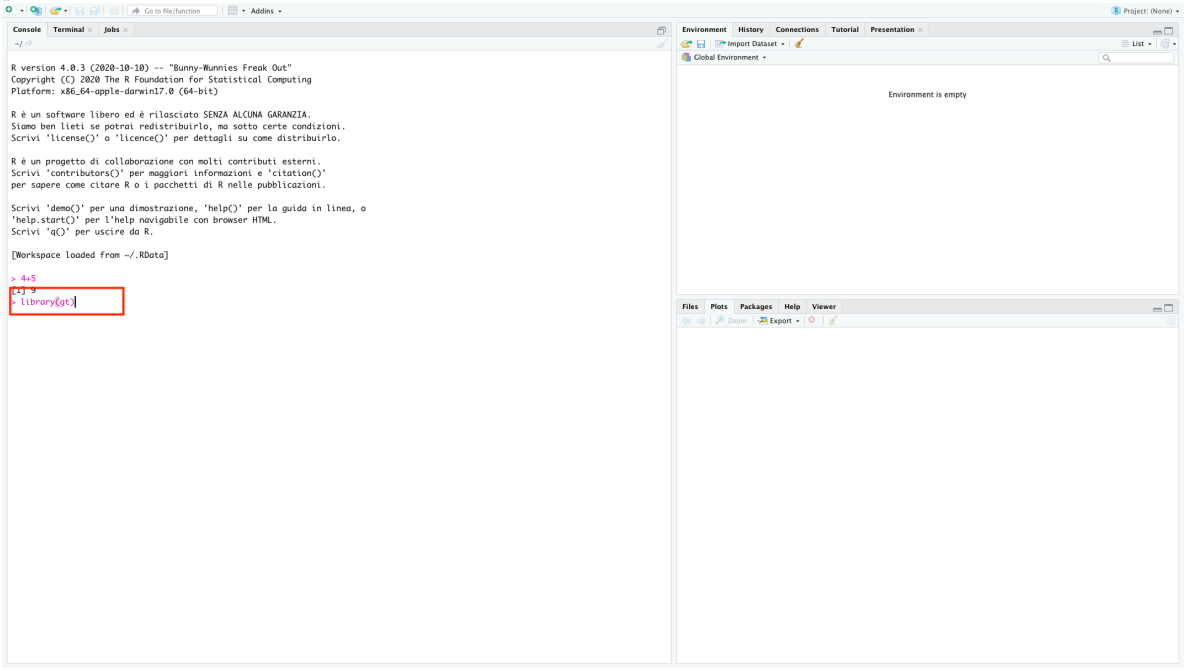
4. Once the installation of R is completed (NOT BEFORE), run the R-Studio installer keeping the default settings.
5. Run R-Studio. It should open an window like the one in the image below.



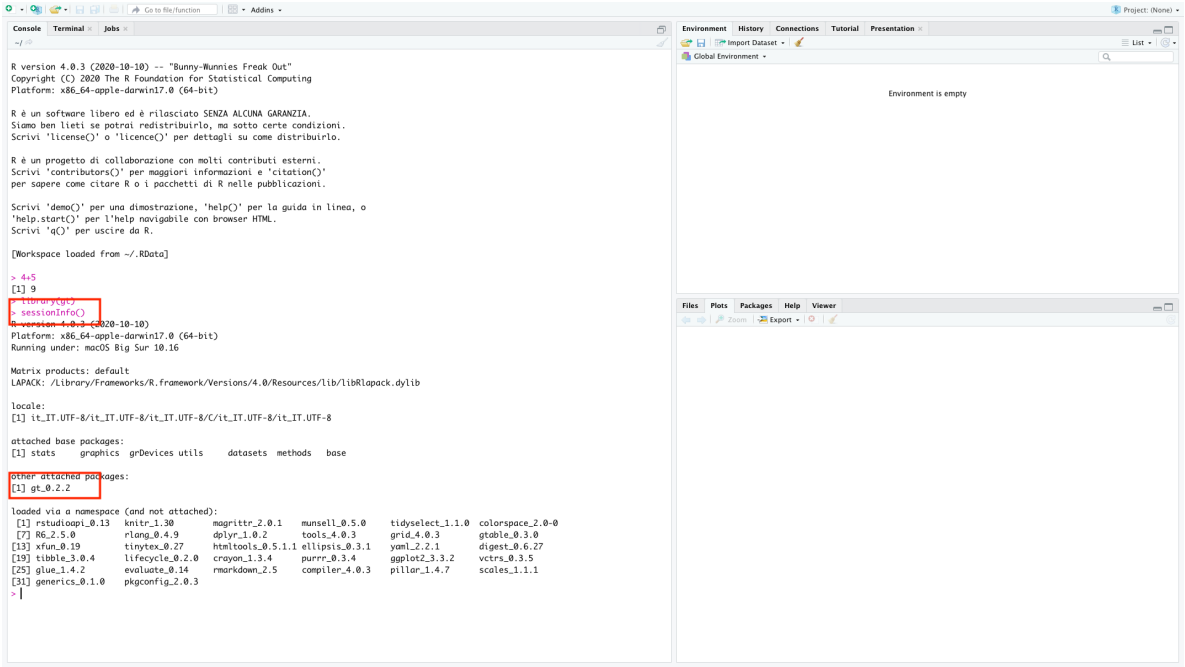
6. In the left hand window, by the sign ">", type "4+5" (without the quotes) and hit enter. An output line reading "[1] 9" should appear. This means that R and R-Studio are working properly. If this is not succesfull, please contact me.
7. Go to Tools -> Install Packages and install the packages requested for this lecture:"gt", "readxl". See the image below.



8. Check that the packages are installed by typing "library(gt)" (without the quotes) in the prompt and press enter.



9. Finally type "sessioninfo()" (without the quotes) and check that gt has been installed.



Visualization suggestions

Following some visualization suggestions that you may explore. Personally, I find them really helpful.

- Setting the workspace:
  - View -> Panes -> Panes Layout
  - clockwise from top-left you should have: Source, Environment, Files, Console
- Setting the appearance of the code:
  - Tools > Global Options -> Appearance -> Editor Theme -> Xcode

The workspace

The Source

The source is a text file with extension .R that can be saved and opened from every version of R and R-Studio. This file will allow you to run and rerun a bunch of code, modify some details if you made a mistake or if you want to change something. This will always be your best friend.

The Environment

The Environment is the place where R saves all the data that you tell "him" to save. The environment will never be clean or contain only the essential objects you need (maybe this will happen when you will be a great programmer, but not for now). You will see there all your datasets, variables, vectors, etc...

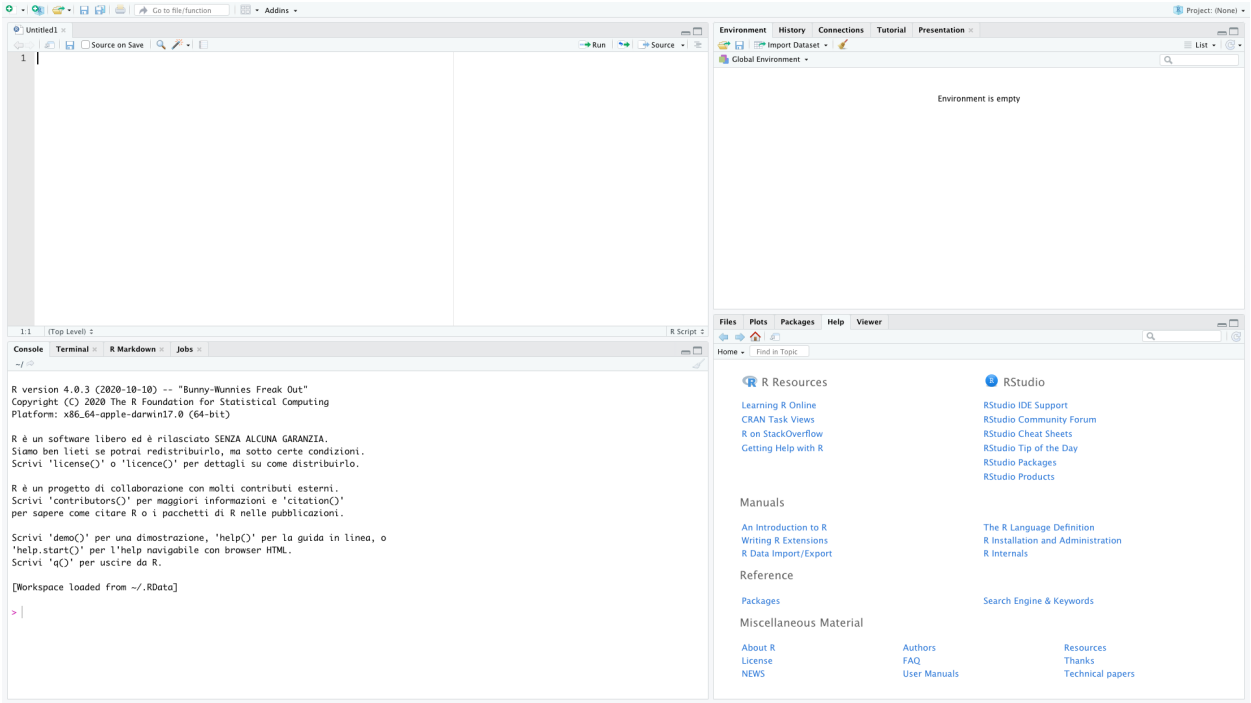


Figure 1: The Workspace.

The Files, Plot, Packages, Help, Vewer

This part of the screen is devoted to many things, as you can see from the title. Viewer, Plots and Help will be activated automatically to show you the requested output. Packages, instead, is useful only when you have to install new packages.

The Console

The console is where you can write some code that will not be saved, if not in the temporary history of the same console. This space is also where R give us the feedback of our inputs in the form of results, warnings and errors.