Machine Learning.

An Introduction to R

Antonio Muñoz, Fernando San Segundo

January 2023.

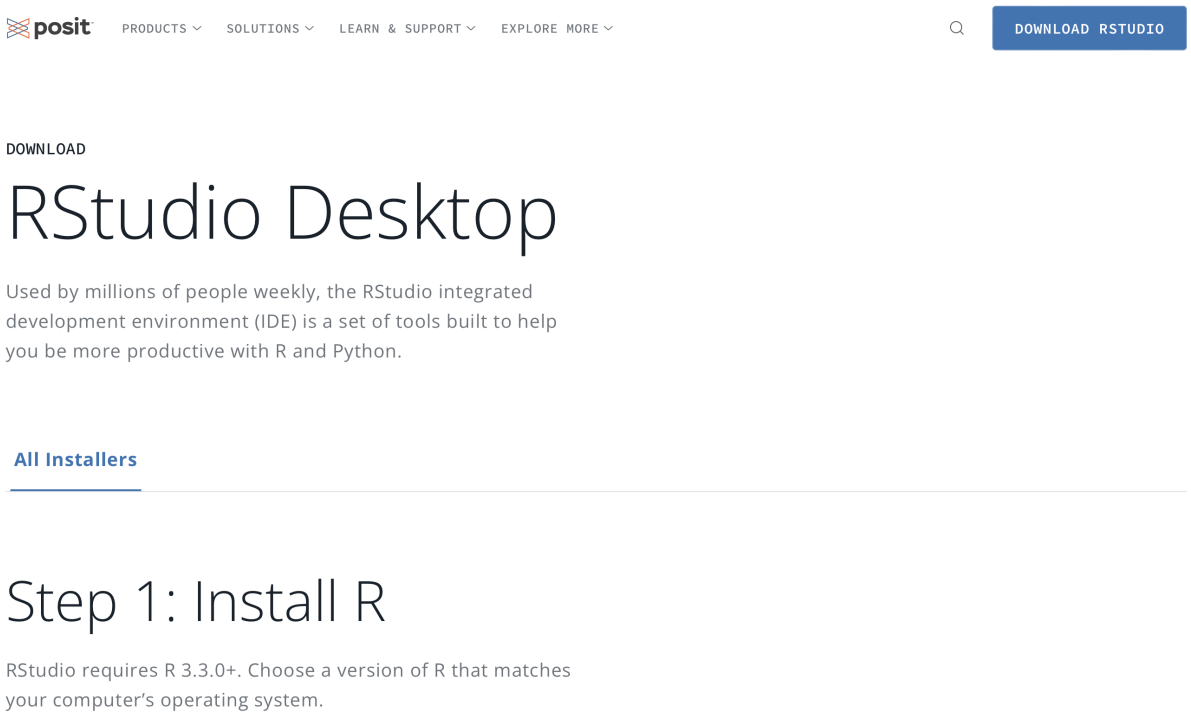
1. Setting up R and RStudio
2. First steps in R and RStudio
3. The RStudio editor and R scripts.

Section 1

Setting up R and RStudio

Setup R

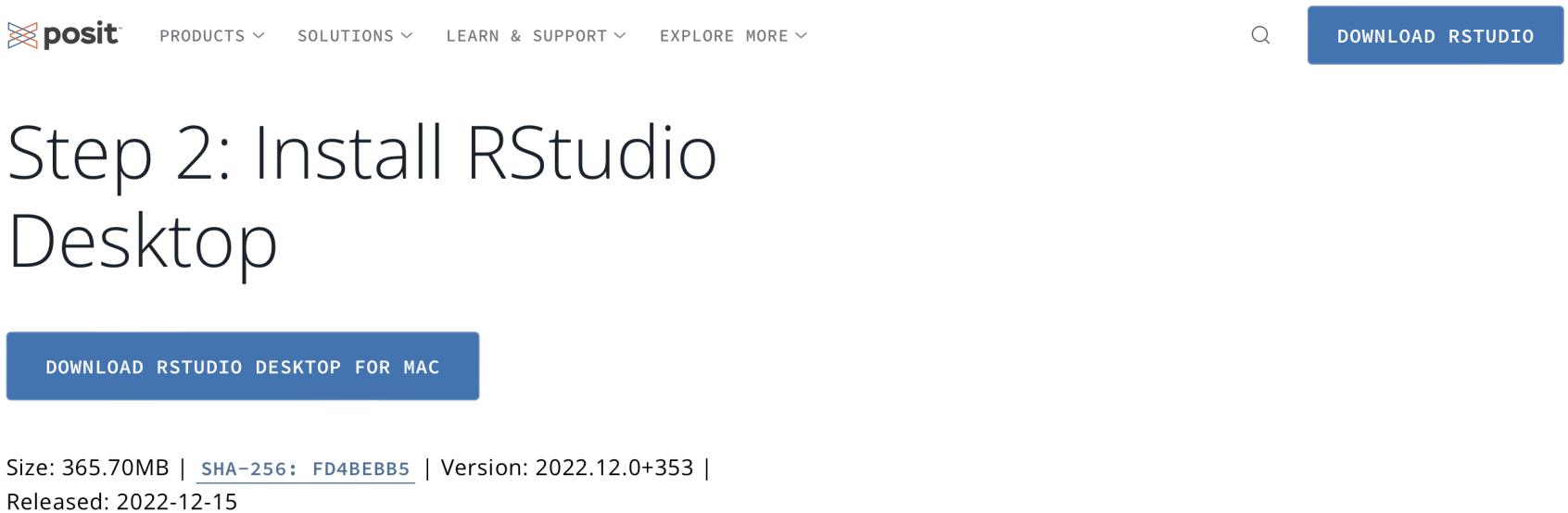
- The download page for R and RStudio that we will be using is https://posit.co/download/rstudio-desktop/



- Download the R installer from that page. **Be careful to pick the right version if you are using a Mac silicon M1 system**. You may also want to check this link. **Install R before moving on!**

RStudio setup

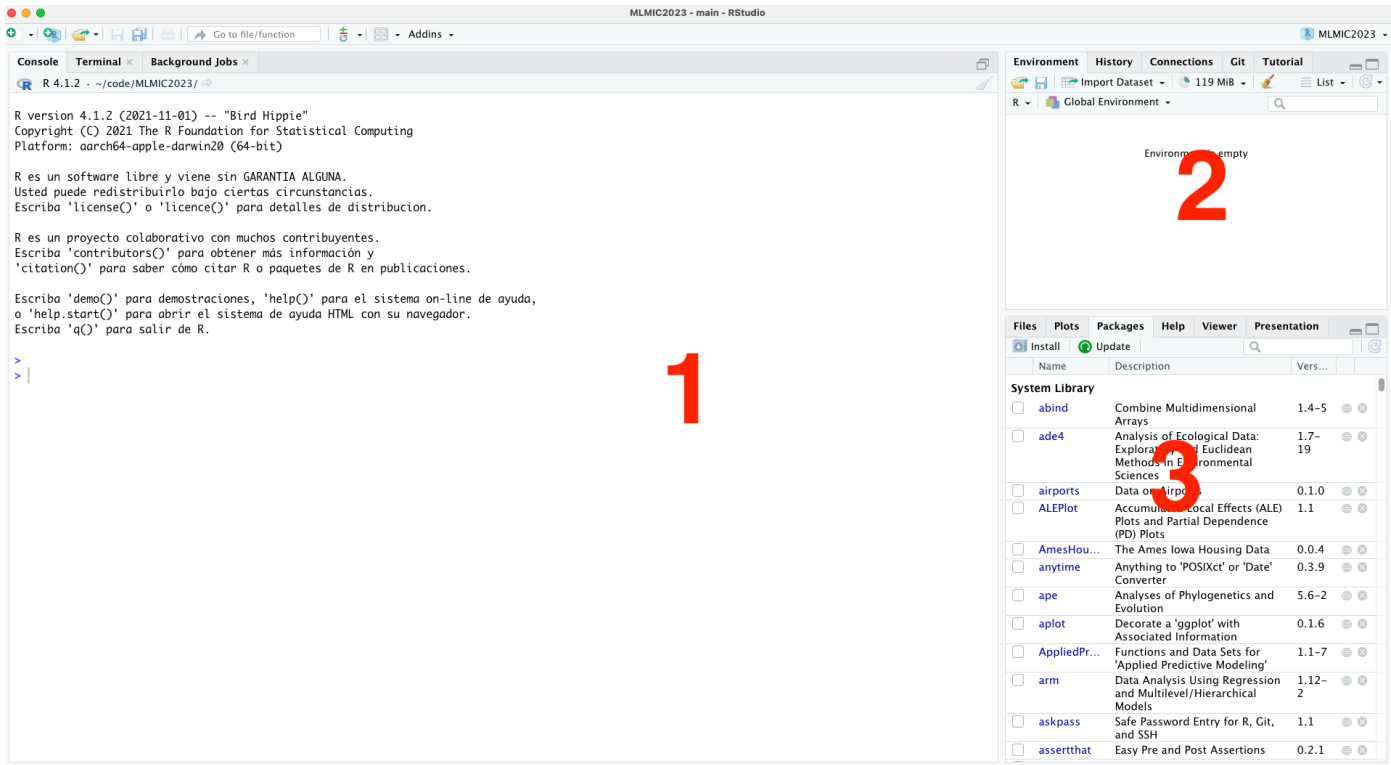
Directly working with R is not recommended, and we will instead use an integrated development environment for R called https://www.rstudio.com/.



K

The Rstudio interface

- When you first open RStudio you will be confronted with three windows or *panes*:
  1. The *R Console* or Command Window (also Terminal)
  2. The Environment / History.
  3. Files / Plots / Packages / Help
  4. Eventually we will also find a fourth pane, the *Editor*, above the Console.

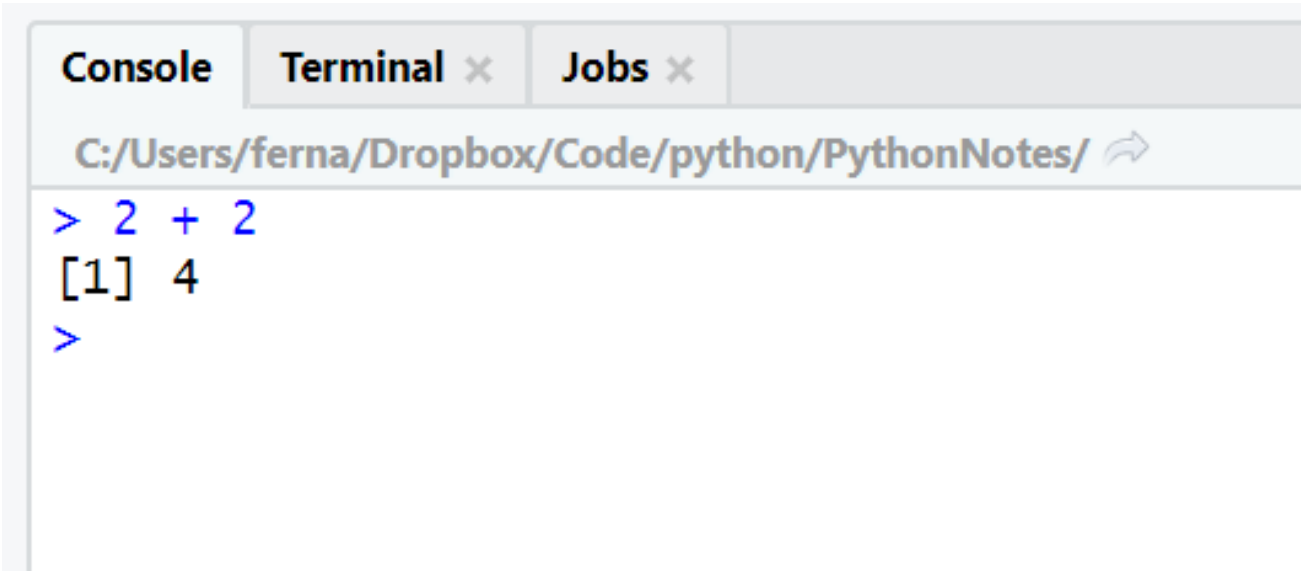


Section 2

First steps in R and RStudio

Working with the console

- We will start working in the *Console* in Rstudio (left pane). Click with your mouse in the line with the prompt (symbol >) and make sure you see the cursor blink in that line. Then type 3^2 a hit RETURN



- That number one between brackets [1] means that this is line number 1 of the answer. We will soon get answers spanning multiple lines. A new prompt appears waiting for further commands.

Use R as a calculator

- Type and execute each of the following lines in the prompt, one by one.

```
2 + 3
15 - 7
4 * 6
13/5
1/3 + 1/5
sqrt(25)
sqrt(26)
sin(pi)
sin(3.14)
```

Notes:

- Use parenthesis to control the execution order.
- Spaces are normally irrelevant. For example: `3 + 5` is equivalent to `3+5`. Use spacing wisely to improve the readability of your code! But **do not type** `sqrt (25)` instead of `sqrt(25)`. It will work but it is ugly, hard to read and error prone.
- R is a **numeric language** (as opposed to *symbolic*). In R `sin(pi)` is not 0 (the exact answer) but rather a very small approximate number in scientific notation:

```
1.224606e-16
```

Errors.

- Try now typing the following incomplete expression:

```
3 +
```

  and hit RETURN. The + symbols that appears is the way that R uses to tell us that it is still waiting for *"something more"* (it has nothing to do with a sum). Your best choice is usually to hit sc Esc and complete the expression.

- Type and execute the following expressions to see different kinds of errors:

```
4/*3
2/0
sqrt(-4)
Sqrt(4)
```

  Regarding the last example: always keep in mind that **R is case-sensitive.**

Further details about RStudio.

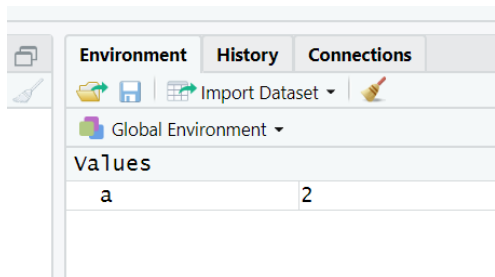
- While you are working at the Console you can use the *up arrow key* to go back in your *command history*. And the *down arrow key* navigates the history forward. The (top right) `History` pane contains that same list and you can use it to copy / paste commands or to save tour command history to a text file.
- *Console cleanup:* Hit $\textsc{Ctrl} + \textsc{L}$. This will clear the console but it has no effect on your command history.
- *Getting help:* The (bottom right) `Help` pane provides access to the R help system. You can also get help on a specific function from the console, using the function key $\textsc{fn1}$ with the cursor placed at the end of the function name. Also, try executing `?log` in the console.

Variables and assignments in R

- Execute the following *assignment* command in the console

```
a <- 2
```

This command binds the *variable (name)* a to that value 2. Apparently nothing happens because there is no output for this command. But note that the top right pane *Environment* displays the value of the variable:



Thus if you execute

```
a + 1
```

you will get the answer 3. Check the environment pane while you execute `a <- a + 1`.

Operations, assignments and displaying results.

- Execute this sequence of commands, checking the results and the values of the variables after each command.

```
a <- 2
b <- 3
c <- a + b
a <- b * c
b <- (c - a)^2
c <- a * b
```

- A command such as `c <- a + b` is an **assignment**; and we have seen that the default behavior of R is *no output* for assignments. But if we want to see the output we can put parenthesis around *the whole assignment command*, like this

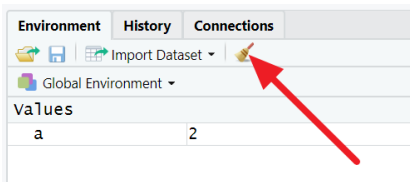
```
(c <- a + b)
```

```
[1] 115
```

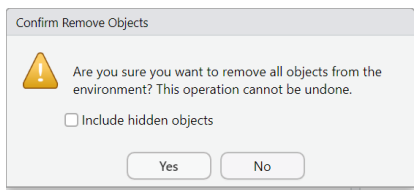
- To get the assignment operator <- with RStudio use the keyboard shortcut Alt + _.

Cleaning your environment.

- Sometimes we will need to start R with a clean environment, free from any object lingering from our previous work. IN order to accomplosh that we start using the Menu sequence Session -> Restart R. The message `Restarting R session...` will appear in the Console.
- We are not done cleaning. If you look at the environment pane tou will see that restarting the session has not cleared the objects we have created. To remove them you can use the broom icon on top of the environment pane



After clicking that icon a confirmation message appears:



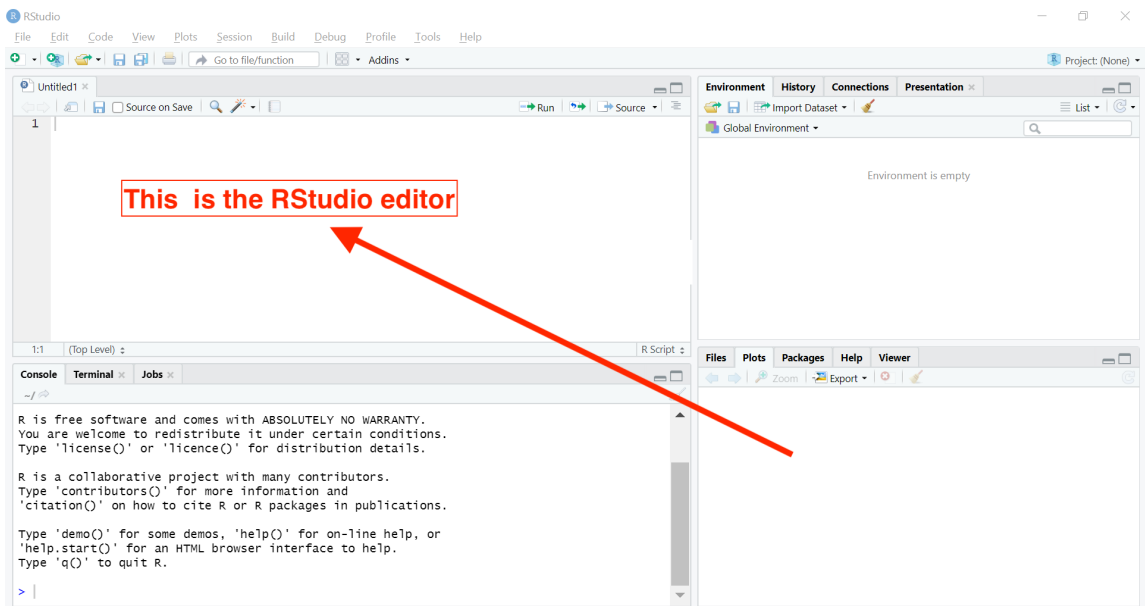
Click *Yes* to accept and you are ready to move on. We will see an alternative cleaning procedure below.

Section 3

The RStudio editor and R scripts.

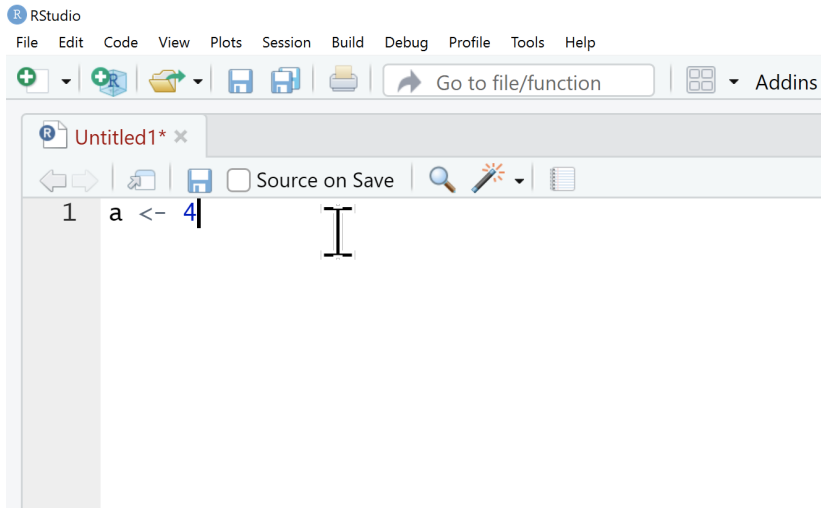
Opening the editor

- Using the console is convenient for some basic tasks but it becomes increasingly awkward as our work with R gets more complex. Lets see a better woy to work in RStudio.
- Use the following keyboard shortcut: under Windows Ctrl + Shift + N, under MacOs Cmd + Shift + N .
  Alternatively use the menu File -> New File -> R Script. A new pane will open in the top left part of the window; that is the RStudio *Editor* where we will do most of our work.



Code execution in the editor

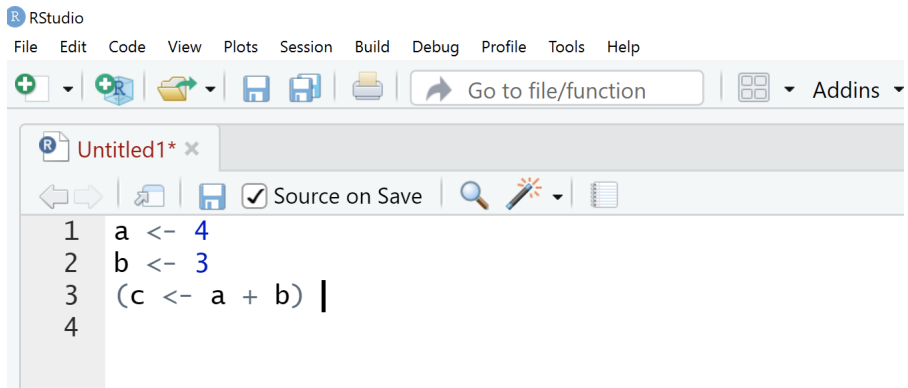
- Let us begin by writing this assignment in the editor:



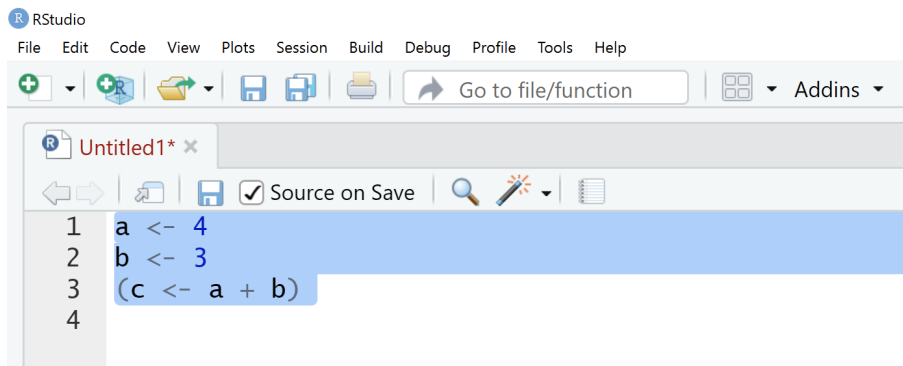
If you hit the RETURN key you will see that the command is not executed (check both the console and environment), the only result is that the cursor has moved to the line below in the editor. To execute your code click again anywhere in the line with `a <- 4` and now use the key combination CTRL + RETURN. A copy of the code appears in the console just as if you executed your code there. Check the environment to see that the code has in fact run.

Executing several commands at once.

- Type some more lines of code in the editor:



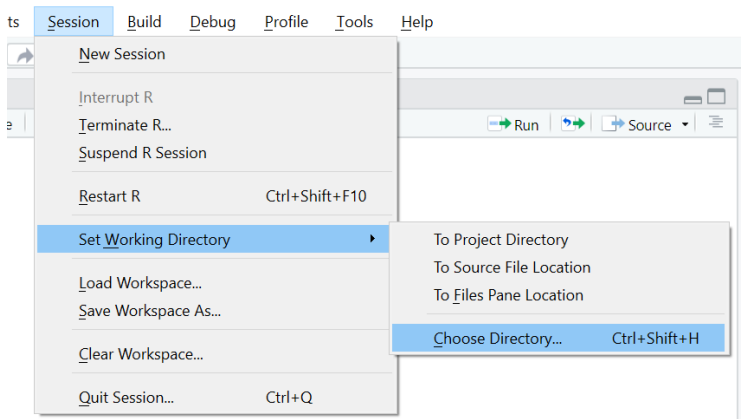
Now select all the lines (using the keyboard or the mouse). MAke sure that all the three lines are completeley selected as shown here:



Use the key combination Ctrl + Return again and check in the console that the three lines of code get executed one by one.

Script files and working directory.

- We will learn how to save to a text file (*R script*) the work we have done in the editor. But first we have to understand the concept of *Working Directory*.
- Every session with R uses a folder in your computer called *Working Directory* to store data, script files, etc. You should get into the habit of telling R about your choice of working directory as one of the first (if not the first) steps of every session. We strongly recommend that you pick a folder for this course (say a folder called *MachineLearning*) somewhere under your user folder. Then you can create subfolders for different sections of the course, assignments, lab sessions, exams, etc. At the beginning of every session you pick one such subfolder. But you have to tell R about that choice. You do that by using the *Session* menu in RStudio as illustrated below:



- After choosing the working directory use the menu File -> Save As to save the R script R with a name such as `sesion01_00` in that working directory. RStudio will automatically add the extension `.R` to the file name.
- We can of course open a script written by someone else. And in fact in the course you will be using many scripts designed for each lecture. Let us begin by downloading the script

  P1_1_R_Introduction.R

  from Moodle. Save it to your working folder for this session. Use the *Files* pane to open it in the editor. We will continue our work for today in that file. Return here when you have completed your work with the script.

Practice for this session

- Using what you have learnt in this session and with the help pf that cheatsheet try to work on the exercises in the
  LabPractice_1_1_Statement.pdf
  that you will find in Moodle.

References

- base-R cheatsheet. You may also take a look at the many other cheatsheets https://www.rstudio.com/resources/cheatsheets/.
- (2022) R.A.Irizarry. *Introduction to Data Science*. Online version at https://rafalab.github.io/dsbook
- H. Wickham & G. Grolemund (2021). *R for Data Science*. Online version at https://r4ds.had.co.nz.
- R. Peng (2020) *R Programming for Data Science* Online version at https://bookdown.org/rdpeng/rprogdatascience/
- W. Chang (2022) *R Graphics Cookbook* Online version at https://r-graphics.org/index.html
- H. Wickham (2022) *Advanced R* Online version at https://adv-r.hadley.nz/
- B. Boehmke (2016) *Data Wrangling with R.* Springer ISBN: 978-3-319-45599-0