

# Introduction to R

## Session 01: Introduction to R

Álvaro Pérez<sup>1</sup>

Instituto Tecnológico Autónomo de México

Fall 2024

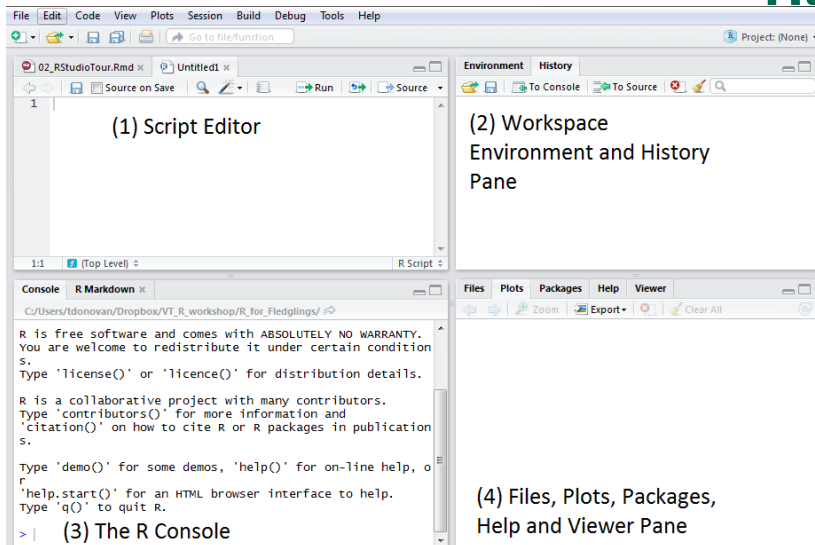
---

<sup>1</sup>Based on PhD Romero Londoño's notes.

- ▶ R works on pretty much every platform available, including the widely available Windows, Mac OS X, and Linux systems. You can download R from the CRAN website
- ▶ There is also an integrated development environment available for R that is built by RStudio.

- ▶ R is a language for working with data
- ▶ RStudio is an environment for working with that language
- ▶ Excel/Sheets is a great tool for accountants, not for working with data
- ▶ Learning to program is a highly valuable skill (regardless of what you want to do)
- ▶ Programming: a language to communicate with the computer
- ▶ Programming requires you to be very precise: Computer will do exactly as told

- ▶ Console
- ▶ Environment Pane
- ▶ Browser Pane
- ▶ Source Editor



The screenshot shows the RStudio IDE interface with four panels labeled (1) through (4):

- (1) Script Editor:** The top-left panel, containing a script editor with a file named "02\_RStudioTour.Rmd" and an "Untitled1" tab. The script content is: 

```
1
```
- (2) Workspace Environment and History Pane:** The top-right panel, showing the "Environment" and "History" tabs. The "Environment" tab is active, showing a list of objects in the workspace.
- (3) The R Console:** The bottom-left panel, showing the R console output. The output text is: 

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
> |
```
- (4) Files, Plots, Packages, Help and Viewer Pane:** The bottom-right panel, showing the "Files", "Plots", "Packages", "Help", and "Viewer" tabs. The "Files" tab is active, showing a file explorer view of the current project directory.

```
1 #Generate 500 heads and tails
2 data <- sample(c("Heads","Tails"),500,replace=TRUE)
3 #Calculate the proportion of heads
4 mean(data=="Heads")
5 #This line should give an error - it didn't work!
6 data <- sample(c("Heads","Tails"),500,replace=BLUE)
7 #This line should give a warning
8 #It did SOMETHING but maybe not what you want
9 mean(data)
10 #This line won't give an error or a warning
11 #But it's not what we want!
12 mean(data=="heads")
```

- ▶ See the code that we've run
- ▶ See the output of that code, if any
- ▶ See any errors or warnings (in **red**)
- ▶ Errors mean it didn't work
- ▶ Warnings mean it **maybe** didn't work.
- ▶ Just because there is no error or warning does **not** mean it **did** work! Always check

- ▶ Two important tabs: Environment and History
- ▶ History: Log of what we have done
  - ▶ Can re-run commands by double-clicking them or hitting Enter
  - ▶ Send to console with double-click/enter
  - ▶ Send to source pane with Shift+double-click/Enter
  - ▶ Or use "To Console." or "To Source" buttons
- ▶ Environment: Objects we have created
  - ▶ All the objects we have in memory
  - ▶ For example, we created the "data" object, so we can see that in Environment
  - ▶ It shows us lots of useful information about that object too (e.g., size)
  - ▶ You can erase everything with the little broom (equivalent to `"rm(list=ls())"`)



- ▶ When you create something that must be viewed, like a plot, it will show up here
- ▶ For example:

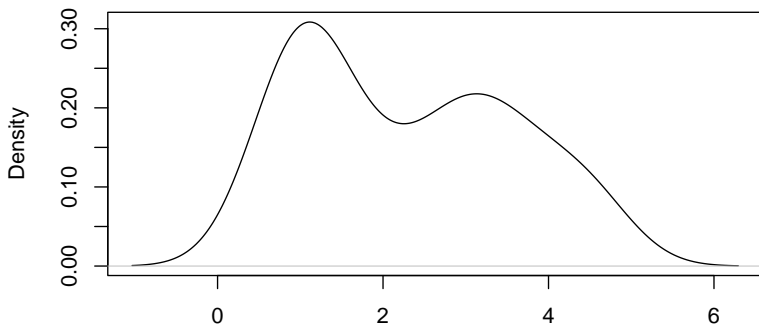
```
1 data(LifeCycleSavings)
2 plot(density(LifeCycleSavings$pop75),
3      main='Percent of Population over 75')
```

- ▶ For example (using ggplot):

```
1 data(LifeCycleSavings)
2 library(ggplot2)
3 ggplot(LifeCycleSavings, aes(x=pop75)) +
4   stat_density(geom='line') +
5   ggtitle('Percent of Population over 75')
```

- ▶ “Export” button here - save plots you’ve created (better to do this via code)

### Percent of Population over 75



N = 50 Bandwidth = 0.5312

- ▶ You should be working with code from this pane, not the console
- ▶ Also, comments “#” lets you write a comment
- ▶ Switch between tabs like a internet browser
- ▶ Select a chunk of code and hit the “Run” button (ctrl+enter)
- ▶ Going one line at a time lets you check for errors more easily

```
1 data(mtcars)
2 mean(mtcars$mpg)
3 mean(mtcars$wt)
4 372+565
5 log(exp(1))
6 2^9
7 (1+1)^9
```

- ▶ Let's create an object. Do this with the assignment operator "<-" (a.k.a., "gets")

```
1 a <- 4
```

- ▶ Why store it as an object? To look at it and manipulate it
- ▶ We can do more complex calculations before storing it, too.

```
1 b <- sqrt(16)+10
```

- ▶ We already determined that 'a' was a number
- ▶ What else could it be? What other kinds of variables are there?
- ▶ Some basic object types:
  - ▶ Numeric: A single number
  - ▶ Character: A string of letters, like "hello"
  - ▶ Logical: 'TRUE' or 'FALSE' (or 'T' or 'F')
  - ▶ Factor: A category, like "left handed", "right handed", or "ambidextrous"
  - ▶ Vector: A collection of objects of the same type

- ▶ Data is basically a bunch of variables all put together
- ▶ A lot of R works with vectors, which are a bunch of objects all put together!
- ▶ Use 'c()' (concatenate) to put objects of the same type together in a vector
- ▶ Use square brackets to pick out parts of the vector

```
1 d <- c(5,6,7,8)
2 c(is.numeric(d),is.vector(d))
3 d[2]
```

- ▶ Statistics helps us make sense of lots of different measurements of the same thing
- ▶ Thus, lots of statistical functions look at multiple objects

```
1 mean(d)
2 c(sum(d), sd(d), prod(d))
```

- ▶ We can perform the same operation on all parts of the vector at once

```
1 d + 1
2 d + d
3 d > 6
```



► Factors make a lot more sense as a vector

```
1 continents <- as.factor(c('Asia', 'Asia', 'Asia',  
2                             'N_America', 'Europe',  
3                             'Africa', 'Africa'))  
4 table(continents)  
5 continents[4]
```