Introduction to R Programming

Getting Started

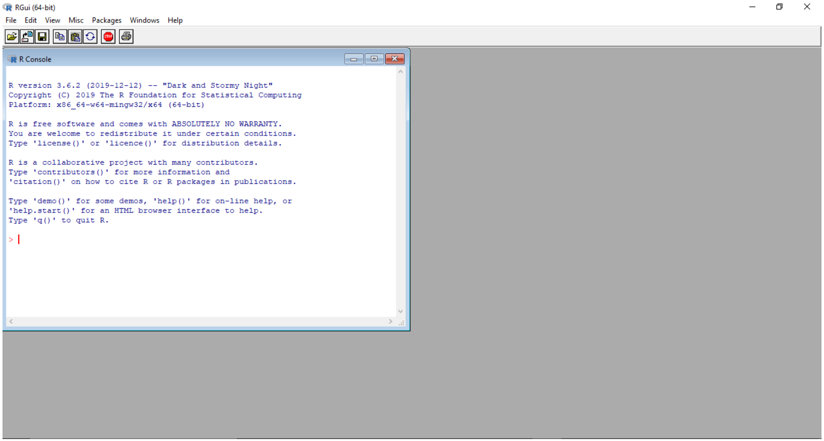
Pedro Fonseca

01 Maio 2020

R and Rstudio

- R is a programming language and free software environment for statistical computing and graphics.
- RStudio is an integrated development environment (IDE) for R.
- You can use R without using RStudio, but you can't use Rstudio without using R.

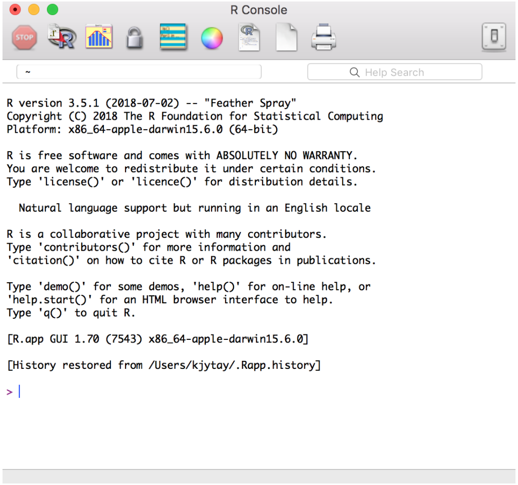
This is how R looks like



```
R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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.

>
```

Figure 1: R console on windows

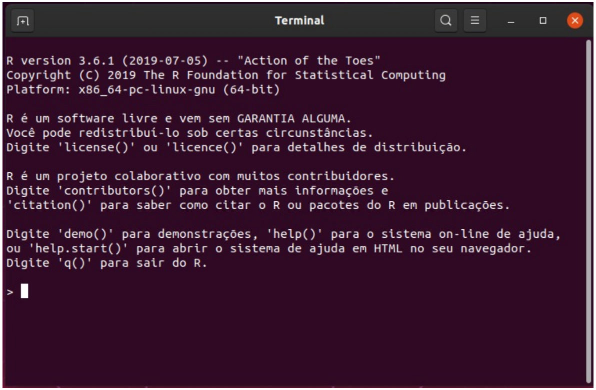
This is how R looks like



```
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

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.

  Natural language support but running in an English locale

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.

[R.app GUI 1.70 (7543) x86_64-apple-darwin15.6.0]

[History restored from /Users/kjytay/.Rapp.history]

> 
```

Figure 2: R console on MacOS

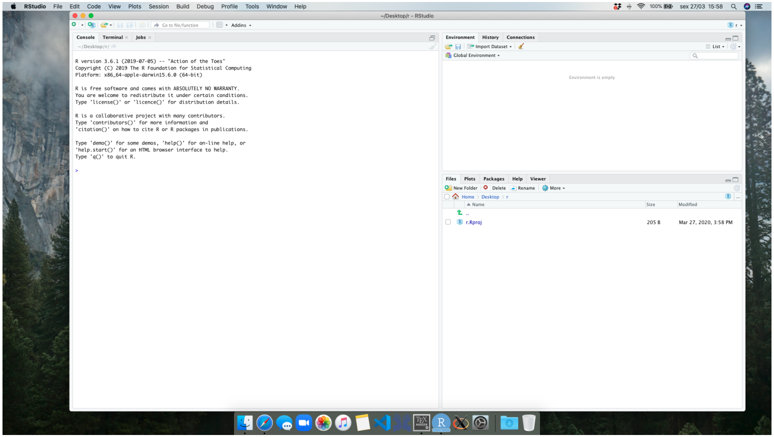
This is how R looks like



```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

>
```

Figure 3: R on Ununtu

This is how Rstudio looks like



```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

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.

>
```

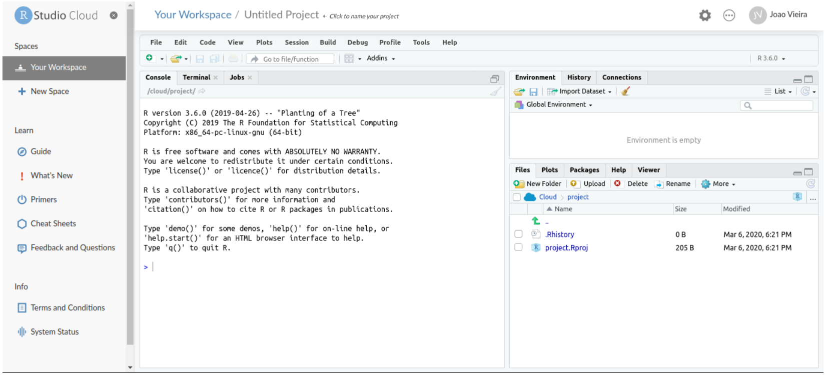
Figure 4: Rstudio on MacOS

Rstudio Cloud

If you don't want to install R and RStudio:

1. Go to RStudio Cloud
2. Create an account and login
3. Click "New Project"

Rstudio Cloud



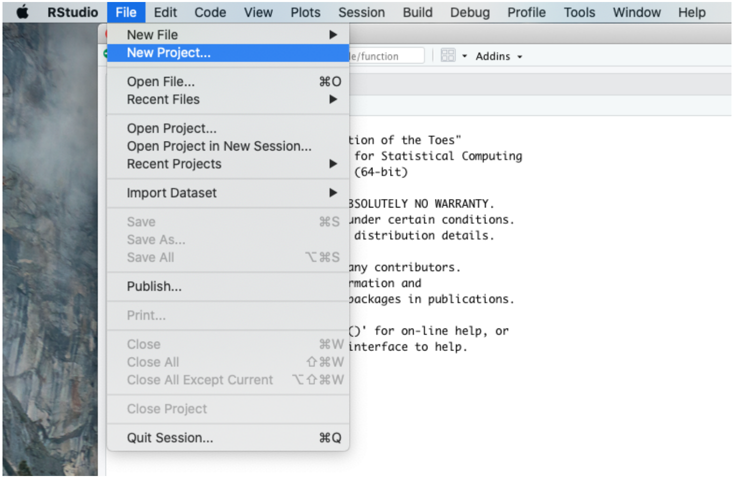
```
R version 3.6.0 (2019-04-26) -- "Planting of a Tree"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

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.

>
```

Name	Size	Modified
..		
.Rhistory	0 B	Mar 6, 2020, 6:21 PM
project.Rproj	205 B	Mar 6, 2020, 6:21 PM

Figure 5: Rstudio Cloud

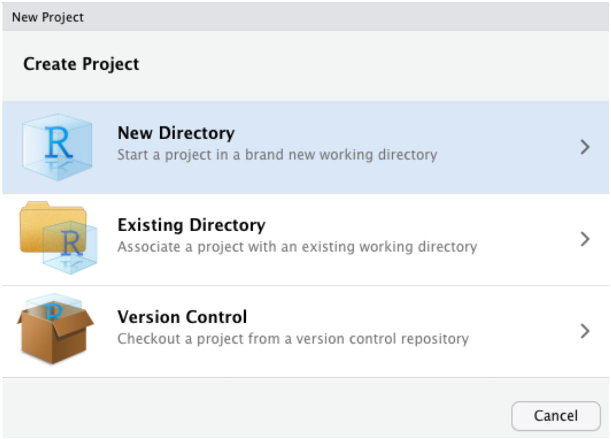
For additional information see https://rstudio.cloud/learn/guide

Your first Rstudio project

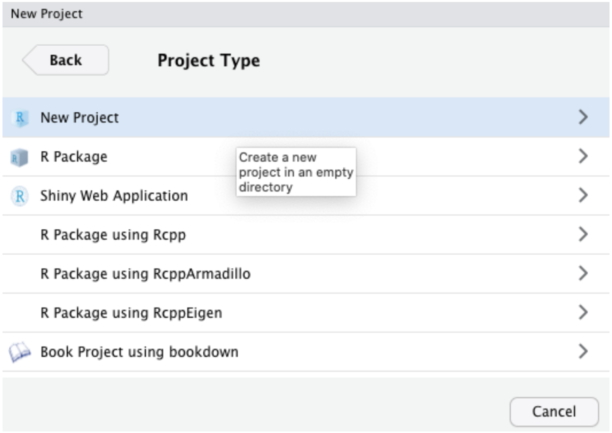


```
tion of the Toes"
 for Statistical Computing
 (64-bit)

BSOLUTELY NO WARRANTY.
under certain conditions.
 distribution details.

any contributors.
rmation and
packages in publications.

()' for on-line help, or
interface to help.
```

Your first Rstudio project

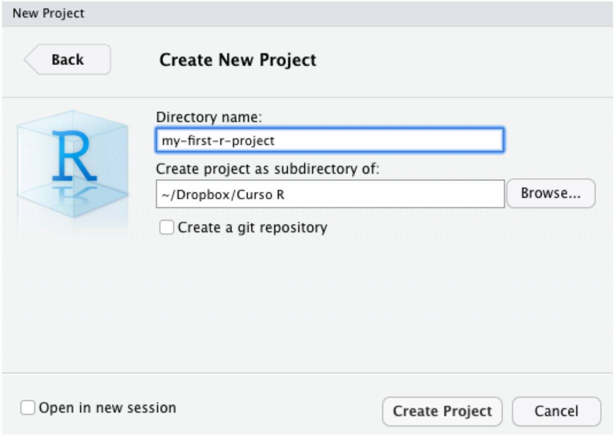

New Project
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel

Your first Rstudio project



Create a new project in an empty directory

Your first Rstudio project


New Project

Create New Project

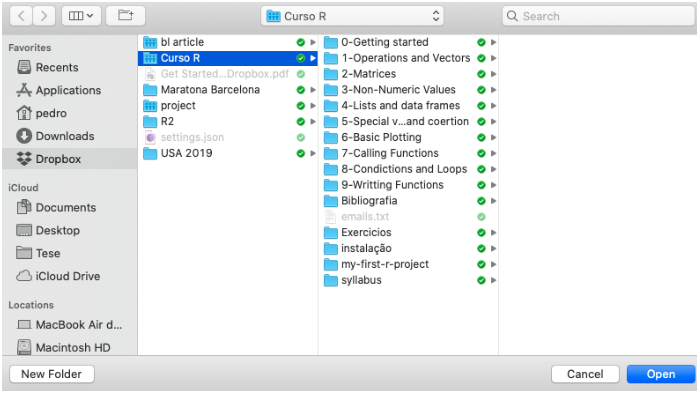
Directory name: my-first-r-project

Create project as subdirectory of: ~/Dropbox/Curso R

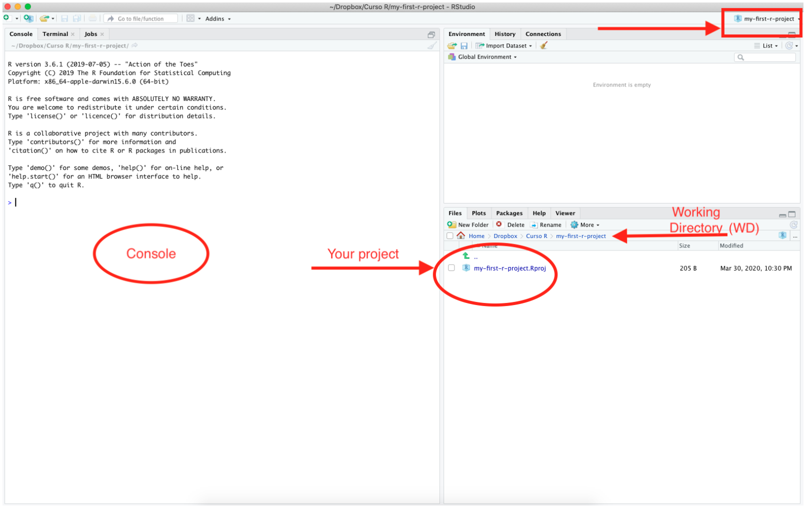
☐ Create a git repository

☐ Open in new session

Your first Rstudio project



Your first Rstudio project



```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

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.

>
```

R as a calculator

We can use R's console as a calculator:

```r
2+3
```

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

```r
3*5
```

```
## [1] 15
```

```r
14.5/6
```

```
## [1] 2.416667
```

```r
3^2
```

```
## [1] 9
```

R as a calculator

```r
(3^2)+14/(6+5)
```

```
## [1] 10.27273
```

```r
(3^2)+14/6+5
```

```
## [1] 16.33333
```

R as a calculator

```r
25^0.5
```

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

```r
sqrt(25)
```

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

R as a calculator

```r
log(5)
```

```
## [1] 1.609438
```

```r
log10(5)
```

```
## [1] 0.69897
```

R as a calculator

```r
pi
```

```
## [1] 3.141593
```

```r
cos(2*pi)
```

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

```r
tan(0.6)
```

```
## [1] 0.6841368
```

```r
sin(0.6)/cos(0.6)
```

```
## [1] 0.6841368
```

Functions

- R has a large collection of built-in functions.
- We´ve already used `log`, `log10`, `sqrt`, `sin`, `cos` and `tan`

Functions

This is how you call a function:

```
function_name(arg1 = val1, arg2 = val2, ...)
```

- Some arguments are mandatory.
- Some arguments are optional and have default values.
- Argument names are not mandatory.
- If you don't provide the names of the arguments, you must input the arguments in the correct order.
- As long as the argument's names are provided, the order is irrelevant.
- Help pages can be useful.

Getting Help

- If you don’t know what a functions does just put ,“?”, before the name of the function and send it to R’s console.
- In the help page a function you can find:
- Its arguments and respective admissible values
- The interpretation of its output
- Examples
- Related functions

```
?mean
?library
?sqrt
```

Some examples of functions

The exponential function is given by `exp()`.

```r
exp(x=3)
```

```
## [1] 20.08554
```

Some examples of functions

Logarithms can be calculated with the `log` function.

```r
log(x = 243, base = 3)
```

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

```r
log(x = 243)
```

```
## [1] 5.493061
```

The *base* argument is optional. The default value is e.

Some examples of functions

```r
log(243, exp(1))
```

```
## [1] 5.493061
```

```r
log(exp(1), 243)
```

```
## [1] 0.1820478
```

```r
log(base = exp(1), x = 243)
```

```
## [1] 5.493061
```

Tip: try

```r
?log
```

Some examples of functions

```r
log(x = 243, base = exp(1))
```

```
## [1] 5.493061
```

```r
log10(5)
```

```
## [1] 0.69897
```

```r
2^log2(6)
```

```
## [1] 6
```

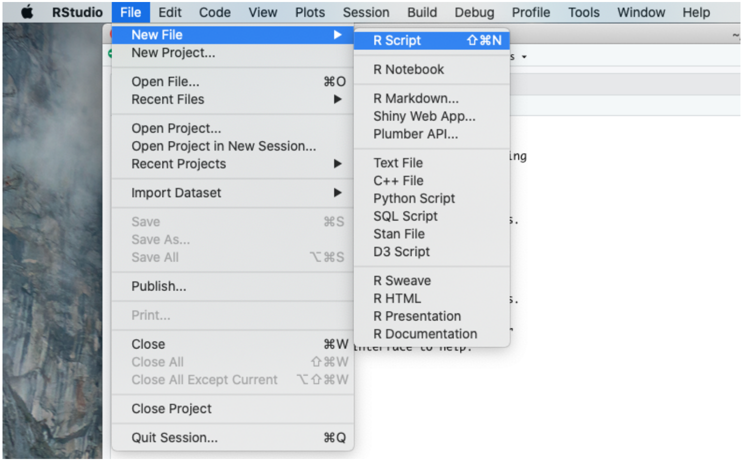
```r
10^log10(5)+1
```

```
## [1] 6
```

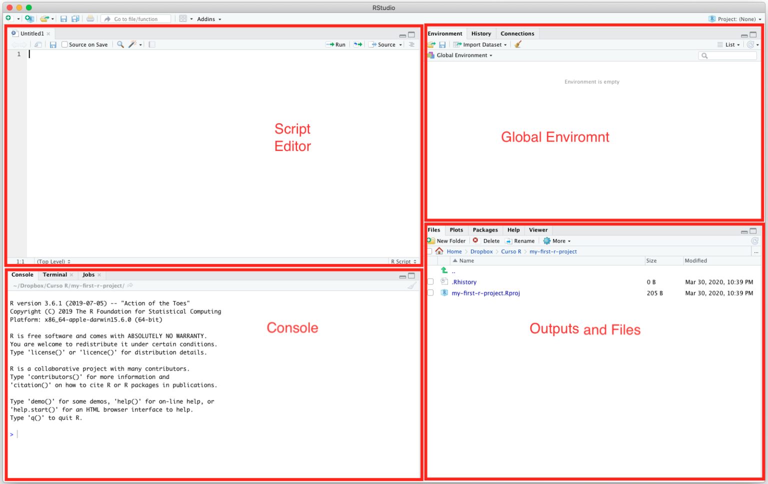
R scripts

- We´ve been using R's console
- Code sent directly to the console is executed but you won't be able to modify it or reuse it later
- Writting our code in scripts is a better option
- A script is just a text file we can use to write code

Your first R script



Rstudio Panes



```
~/Dropbox/Curso R/my-first-r-project/

R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

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.

>
```

Home > Dropbox > Curso R > my-first-r-project

Name	Size	Modified
..		
.Rhistory	0 B	Mar 30, 2020, 10:39 PM
my-first-r-project.Rproj	205 B	Mar 30, 2020, 10:39 PM

Editor

- R opens scripts in the editor pane
- This is where you should write your code
- In the editor you can modify, rerun and save your code at any time

Some Useful Shortcuts

- New script: Cmd/Ctrl + Shift + N
- Save the script: Cmd/Ctrl + S
- Send code from the editor to the console:
  - Cmd/Ctrl + Enter (current line or current selection)
  - Cmd/Ctrl + Shift + S (entire script)

More Shortcuts

- To see a list of Rstudio shortcuts try: Alt/Option + Shift + k
- Alternative: click here

Assigning values to objects

To store values in R's memory you need to assign them to objects. You can use the equal sign, the assign function, or the assign operator:

- The assignment operator is typically recommended.
- The equal sign should be reserved to provide arguments to functions.

```r
object_name_1 <- 5
object_name_1
```

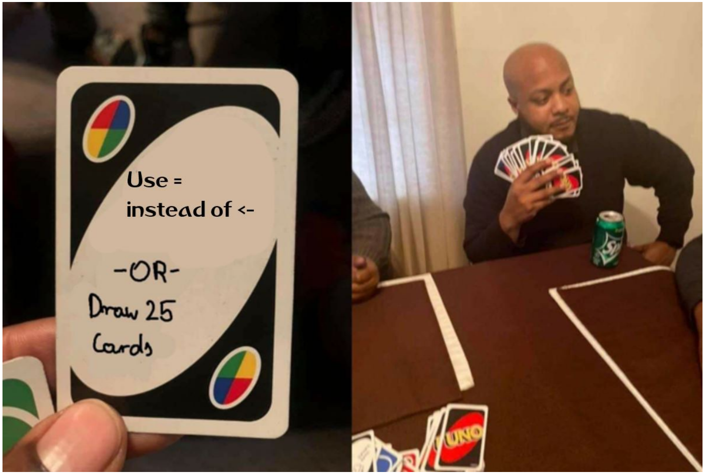
```
## [1] 5
```

```r
object_name_2 <- log(object_name_1) + exp(5)
object_name_2
```

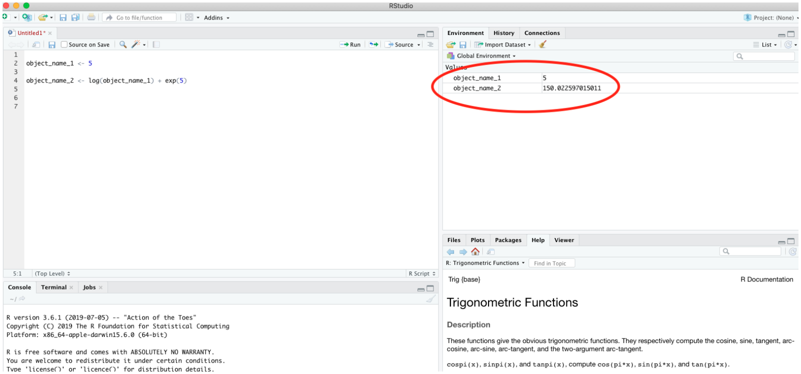
```
## [1] 150.0226
```

Rstudio's keyboard shortcut for the assign operator: "Alt/Option" + "_"

The assignment operator



The assignment operator



```
object_name_1 <- 5

object_name_2 <- log(object_name_1) + exp(5)
```

```
object_name_1      5
object_name_2      150.022597015011
```

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

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.
```

Trig {base} R Documentation

Trigonometric Functions

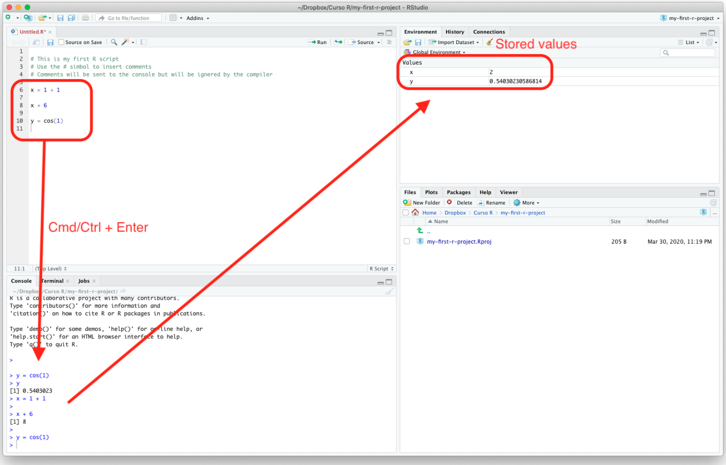
Description

These functions give the obvious trigonometric functions. They respectively compute the cosine, sine, tangent, arc-cosine, arc-sine, arc-tangent, and the two-argument arc-tangent.

`cospi(x)`, `sinpi(x)`, and `tanpi(x)`, compute `cos(pi*x)`, `sin(pi*x)`, and `tan(pi*x)`.

Figure 6: Stored objects are visible in the upper-right pane, under the "Environment" tab

Example



```r

# This is my first R script
# Use the # simbol to insert comments
# Comments will be sent to the console but will be ignered by the compiler

x = 1 + 1

x + 6

y = cos(1)

```

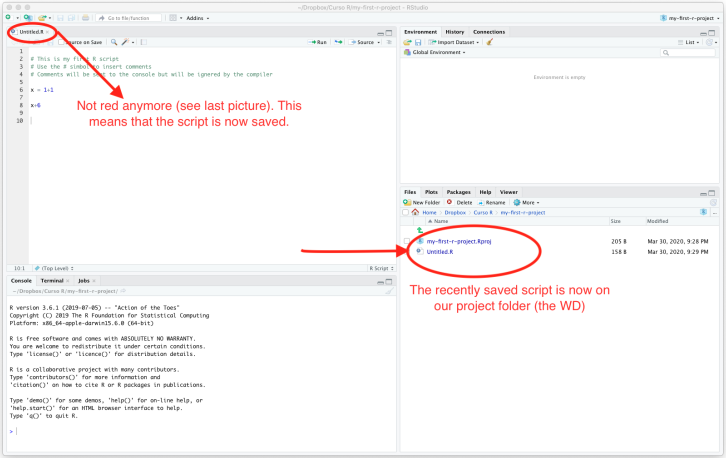
```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

>

> y = cos(1)
> y
[1] 0.5403023
> x = 1 + 1
>
> x + 6
[1] 8
>
> y = cos(1)
>
```

Values

x	2
y	0.54030230586814

Name	Size	Modified
..		
my-first-r-project.Rproj	205 B	Mar 30, 2020, 11:19 PM

Example (cont.)



```

# This is my first R script
# Use the # simbol to insert comments
# Comments will be sent to the console but will be ignored by the compiler

x = 1+1

x+6

```

Not red anymore (see last picture). This means that the script is now saved.

Name	Size	Modified
my-first-r-project.Rproj	205 B	Mar 30, 2020, 9:28 PM
Untitled.R	158 B	Mar 30, 2020, 9:29 PM

The recently saved script is now on our project folder (the WD)

```
~/Dropbox/Curso R/my-first-r-project/

R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

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.

> 
```

Naming Objects

Object names must start with a letter and can only contain letters, numbers, underscores and dots. You want your object names to be short, descriptive and consistent. Ideally, one should follow a convention:

- i_use_snake_case
- otherPeopleUseCamelCase
- some.people.use.periods
- And_aFew.People_RENOUNCEconvention

Case Matters

```r
pi
```

```
## [1] 3.141593
```

```r
r_rocks <- 2 * pi^2
r_rocks
```

```
## [1] 19.73921
```

```r
r_Rocks
```

```
## Error in eval(expr, envir, enclos): object 'r_Rocks' no
```

How to delete objects

To delete stored objects use the `rm` function:

```
rm(object_name_1)
object_name_1
```

```
## Error in eval(expr, envir, enclos): object 'object_name_
```

How to delete objects

- You can input as many objects as you want to `rm()`
- To remove all stored objects all once, use the following command:

```
rm(list = ls())
```

How to print the assigned value

If you make an assignment, you don't get to see the assigned value. You're then tempted to double-check the result:

```r
y <- log(2)+1
y
```

```
## [1] 1.693147
```

This common action can be shortened by surrounding the assignment with parentheses, which causes assignments to print:

```r
(y <- log(2)+1)
```

```
## [1] 1.693147
```

Overwritting stored values

```r
x <- -5
x
```

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

```r
x <- x + 1
x
```

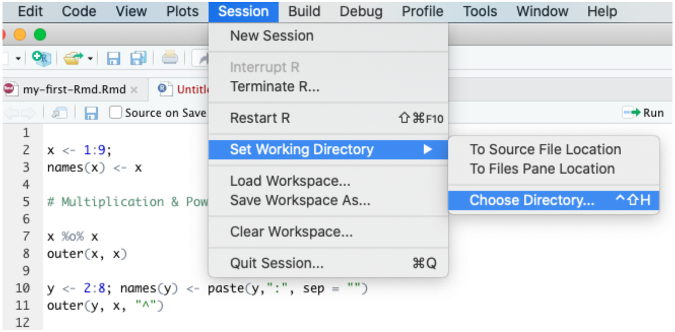
```
## [1] -4
```

Working directory

An active R session always has an associated working directory. R will use the working directory by default to:

- Search for files
- Save outputs (tables, plots, etc)

Setting the working directory



```

x <- 1:9;
names(x) <- x

# Multiplication & Pow

x %o% x
outer(x, x)

y <- 2:8; names(y) <- paste(y,":", sep = "")
outer(y, x, "^")

```

Figure 7: Setting the working directory

Setting and setting the working directory

You can also get or set the working directory in R's console:

```r
getwd()
```

```r
setwd("/folder1/folder2/folder3/")
```

- The problem with commands like this is that such paths will only exist on your computer
- Solution: Rstudio projects

Advantages of Rstudio projects

- Rstudio projects are self-contained.
- They put together all the files that are relevant for a particular project (article, book, research project) in the same folder.
- The project's working directory always points to that folder by default
- Rstudio projects can be moved around on your computer or onto other computers and will still "just work". No directory changes are needed.
- If you need to create additional folders or start moving around parts of you project around dont use the `setwd` function. It is safer to reference the full path.

Packages

- The more specialized functions and data sets are available on packages (also referred to as libraries).
- Installing R Packages:

```r
install.packages("ggplot2", dependencies = TRUE)
```

Loading R Packages:

```r
library("ggplot2")
```

Updating R Packages:

```r
update.packages()  # This is rarely necessary
```

- Packages are developed by the R core team and also by the community of R users.
- You can develop your own packages and make them available to the community on CRAN(The Comprehensive R Archive Network)

Packages

- It is typically recommend to start your scripts with the packages that you need.
- That way, if you share your code with others, they can easily see what packages they need to install.
- Note, however, that you should never include `install.packages` or `setwd` in a script that you share.
- It is very antisocial to change settings on someone else's computer!

Settings

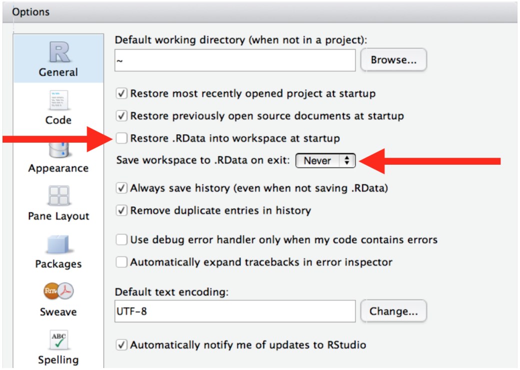
You can change Rstudio's default settings and appearence:

- Mac: Tools $->$ Global Option
- Windows and Linux: Rstudio $->$ preferences

Shortcut:

- Mac: "Cmd" $+$ ","
- Windows and Linux: "Ctrl" $+$ ","

Settings


Options

Default working directory (when not in a project):

~

Browse...

- [✓] Restore most recently opened project at startup
- [✓] Restore previously open source documents at startup
- [ ] Restore .RData into workspace at startup

Save workspace to .RData on exit: Never

- [✓] Always save history (even when not saving .RData)
- [✓] Remove duplicate entries in history
- [ ] Use debug error handler only when my code contains errors
- [ ] Automatically expand tracebacks in error inspector

Default text encoding:

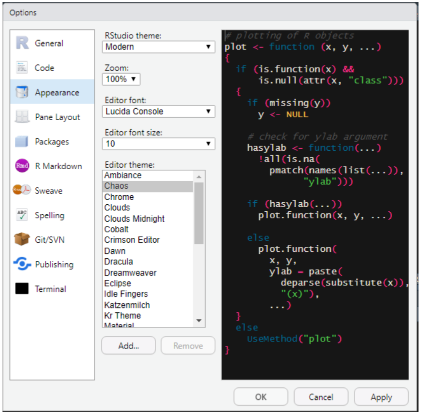
UTF-8

Change...

- [✓] Automatically notify me of updates to RStudio

Figure 8: These are the general settings that we recommend

Settings



RStudio theme: Modern
Zoom: 100%
Editor font: Lucida Console
Editor font size: 10
Editor theme: Chaos

```r
# plotting of R objects
plot <- function (x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...)),
              "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)"),
        ...)
  }
  else
    UseMethod("plot")
}
```

Figure 9: Changing Rstudio's appearence