Tutorial 3- R and R Studio

Brainhack Magdeburg
07.-08.12.2021
- virtual -

programming language	integrated development environment
not elaborate like RStudio	
used to do statistical computing of programs	Development of statistical programs
works independently	compulsory needs R language
.pkg extension	.tgz extension

RStudio *is* an application like Excel or Word—except that instead of helping you creating tables or writing , RStudio helps you write in R.

- Open-source.
- A Large Variety of Libraries
- Excellent for Statistical Computing and Analysis
- Supports various Data Types
- Powerful Graphics.
- Highly Active Community.

Even if you use RStudio, you’ll still need to download R to your computer. RStudio helps you use the version of R that lives on your computer, but it doesn’t come with a version of R on its own.

Forget about the actual R application (until you update it in a few months).

https://rstudio-education.github.io/hopr/starting.html

DOWNLOAD: how and from where

DOWNLOAD

1.For Windows :
Download the binary setup file for R from the following link.(R for Windows)
2.Open the downloaded .exe file and Install R

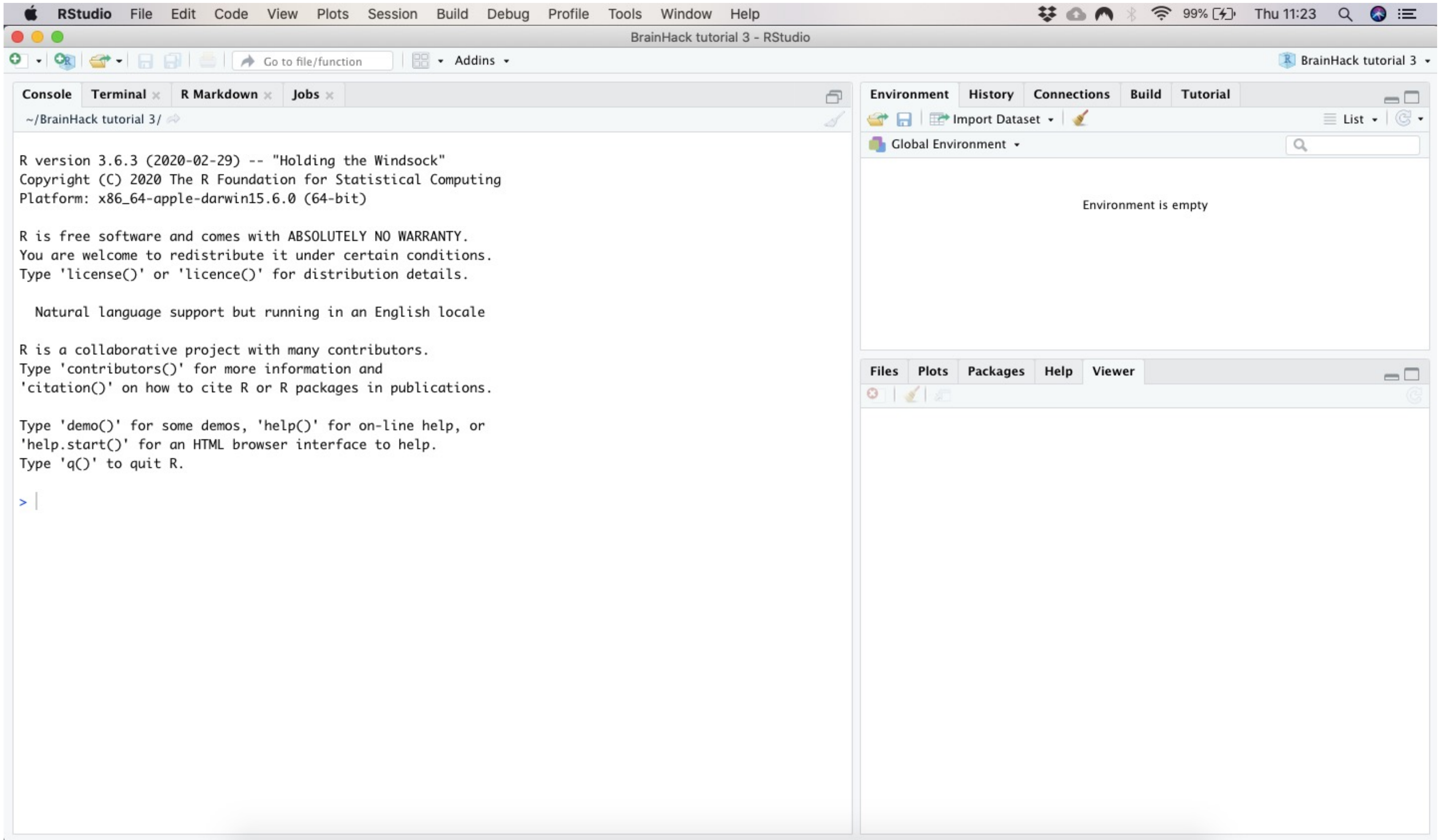
For Mac :
Download the appropriate version of .pkg file form the following link. (R for Mac)
1.Open the downloaded .pkg file and Install R

For Linux :
For complete R System installation in Linux, follow the instructions on the following link (Link)
1.For Ubuntu with Apt-get installed, execute *sudo apt-get install r-base* in terminal.

On the following link Download R Studio choose the appropriate installer file for your operating system, download it and then run it to install R-studio.

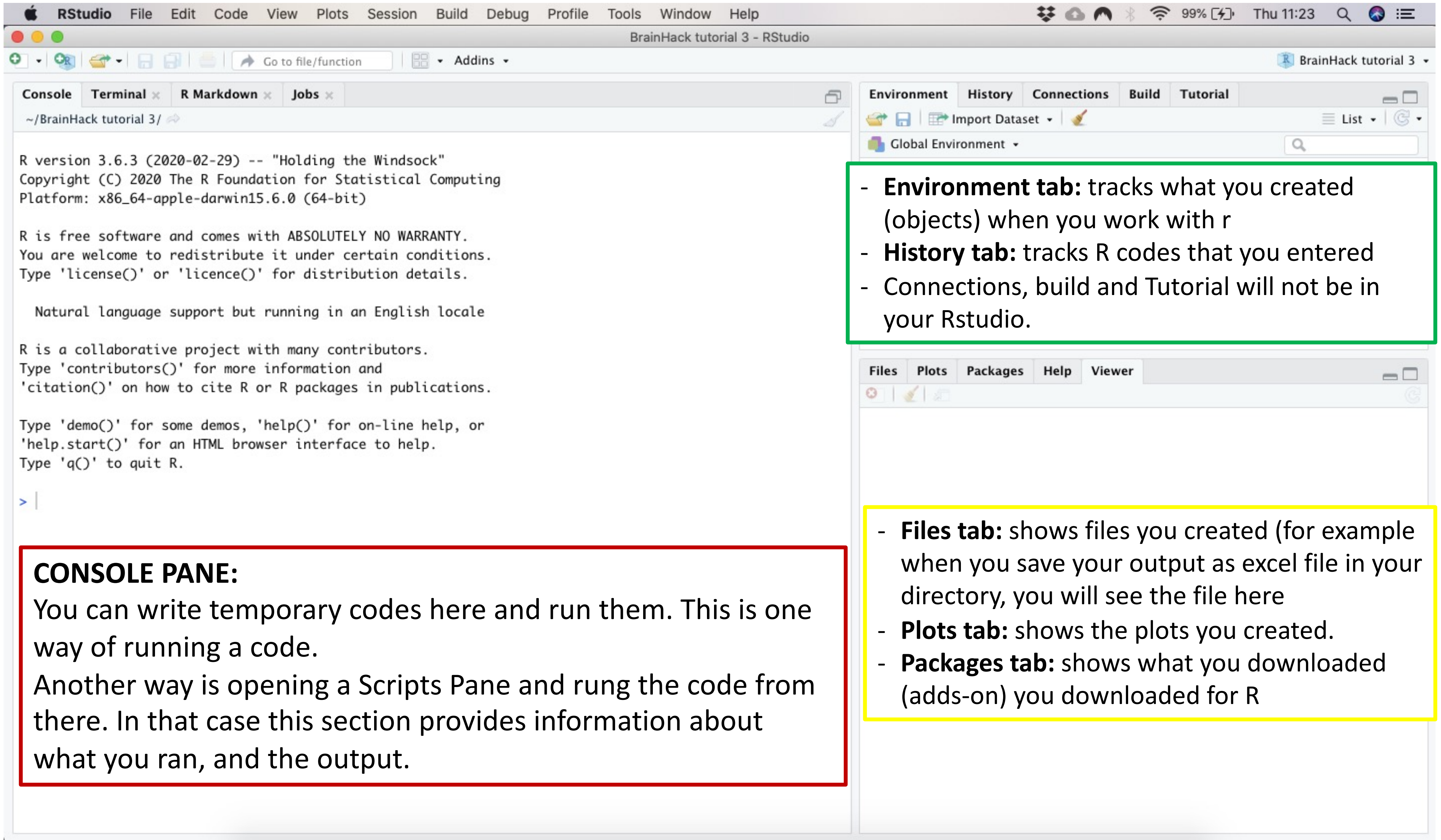
R studio environment

How Rstudio looks after you installed it / open it



```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 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.

>
```

How Rstudio looks after you installed it / open it


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
99% Thu 11:23
BrainHack tutorial 3 - RStudio
Go to file/function
Addins
BrainHack tutorial 3
Console Terminal R Markdown Jobs
~/BrainHack tutorial 3/
Environment History Connections Build Tutorial
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer

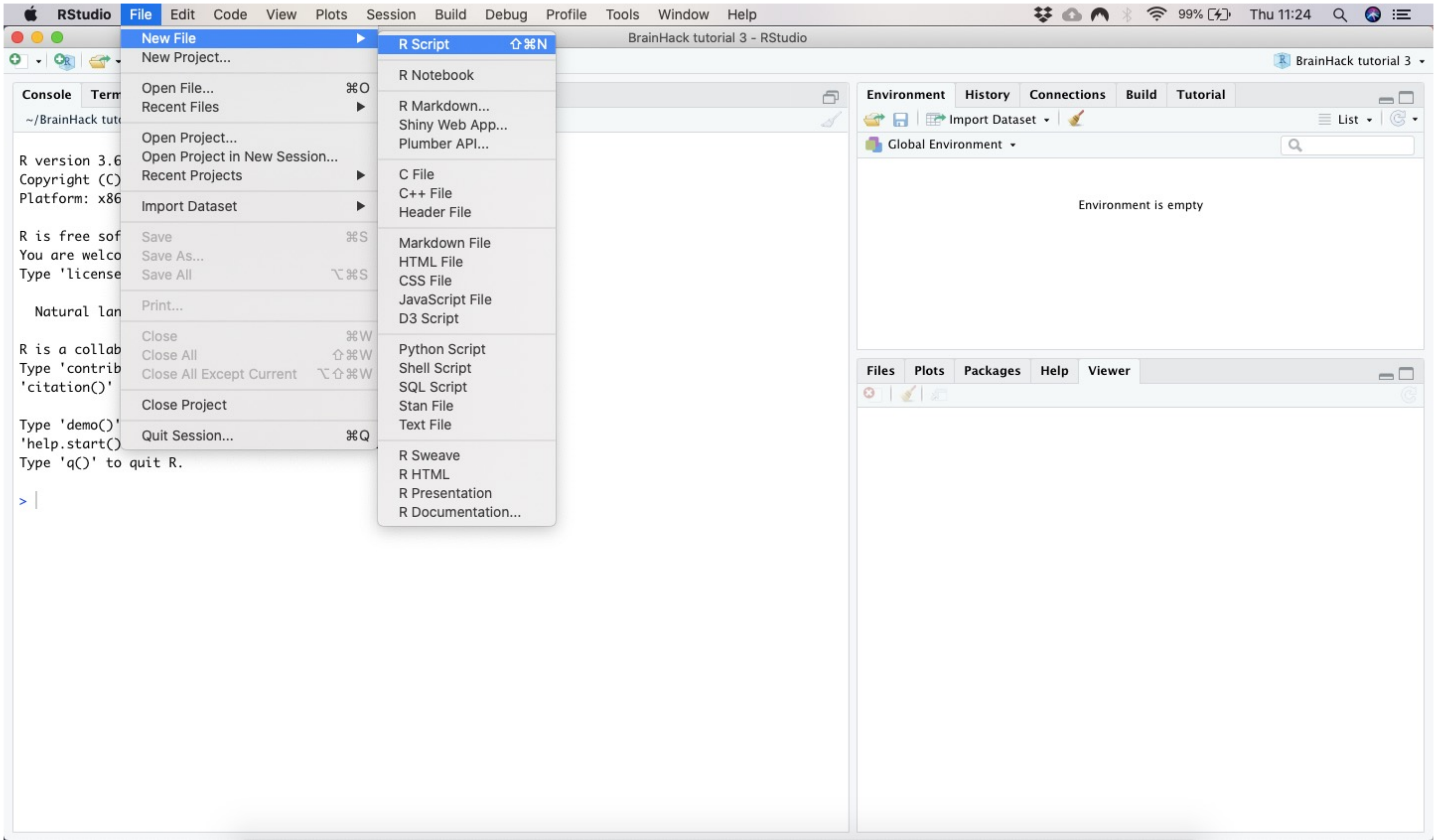
```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

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.

> |
```

- **Environment tab:** tracks what you created (objects) when you work with r
- **History tab:** tracks R codes that you entered
- Connections, build and Tutorial will not be in your Rstudio.

CONSOLE PANE:
You can write temporary codes here and run them. This is one way of running a code.
Another way is opening a Scripts Pane and rung the code from there. In that case this section provides information about what you ran, and the output.

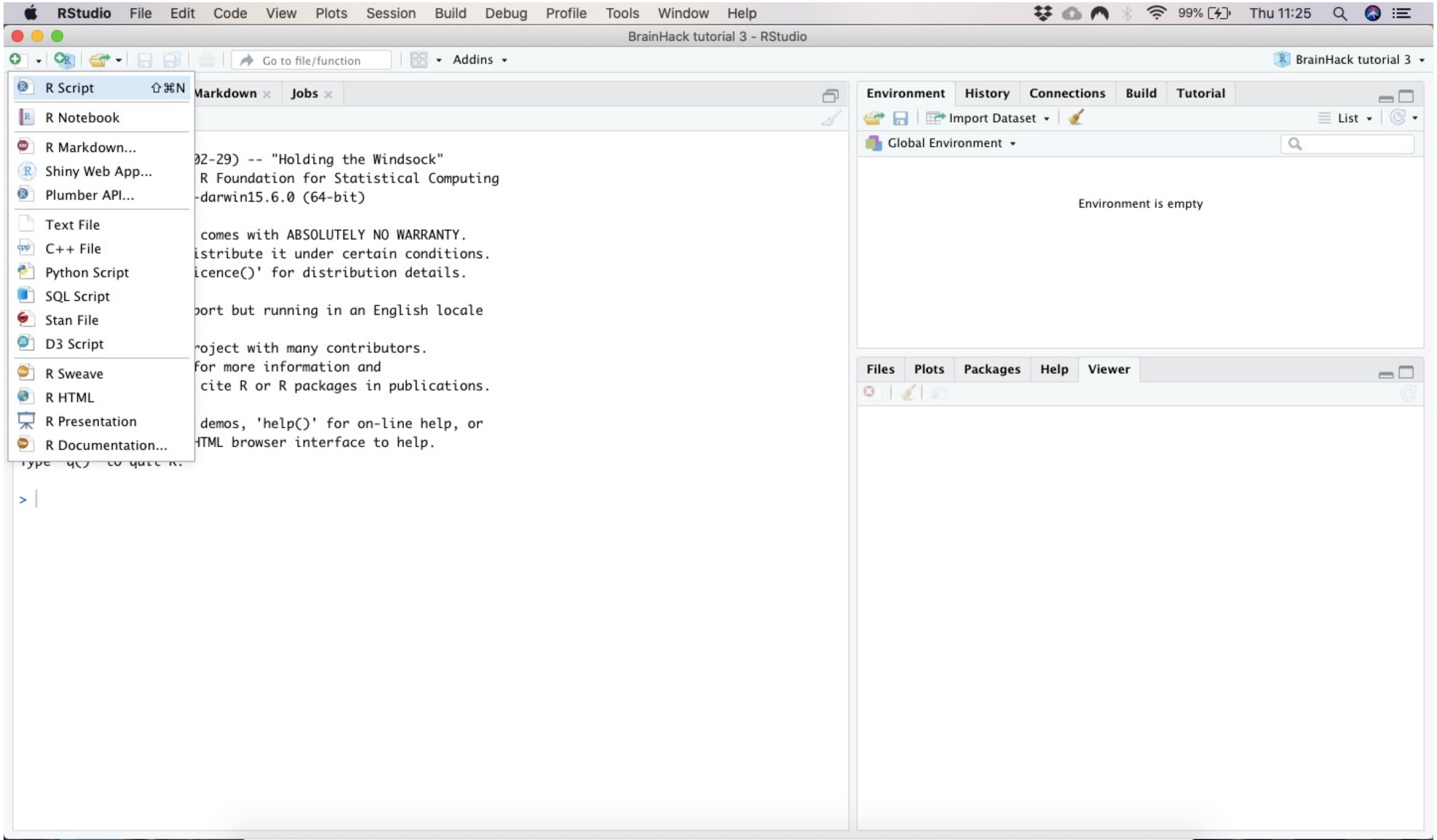
- **Files tab:** shows files you created (for example when you save your output as excel file in your directory, you will see the file here
- **Plots tab:** shows the plots you created.
- **Packages tab:** shows what you downloaded (adds-on) you downloaded for R

Create a script (or open an existent one) OPTION 1



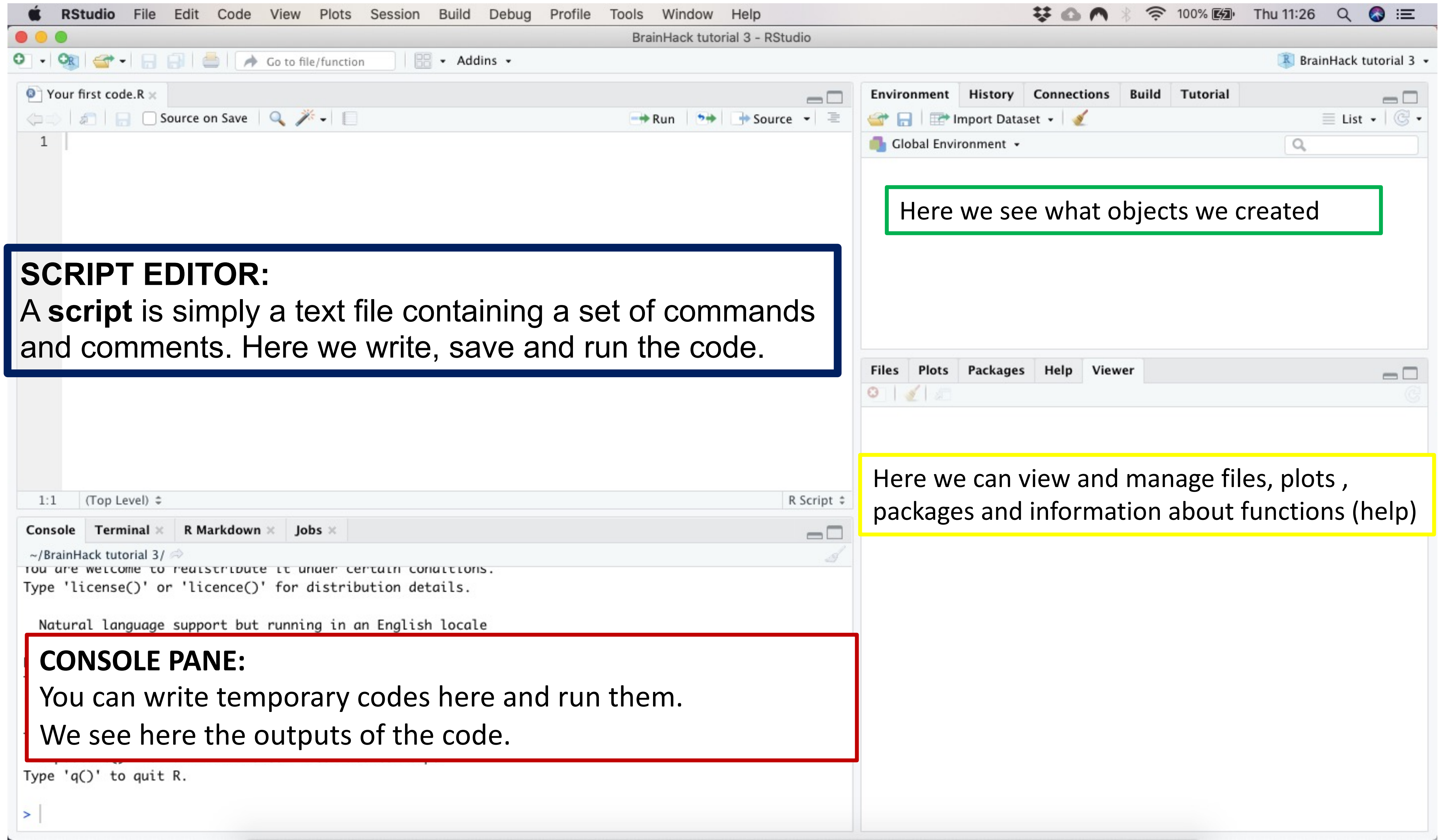
```
R version 3.6
Copyright (C)
Platform: x86

R is free sof
You are welco
Type 'license

  Natural lan

R is a collab
Type 'contrib
'citation()'

Type 'demo()'
'help.start()
Type 'q()' to quit R.

>
```

Create a script (or open an existent one) OPTION 2



```
02-29) -- "Holding the Windsock"
 R Foundation for Statistical Computing
-darwin15.6.0 (64-bit)

 comes with ABSOLUTELY NO WARRANTY.
istribute it under certain conditions.
icence()' for distribution details.

oort but running in an English locale

roject with many contributors.
for more information and
 cite R or R packages in publications.

 demos, 'help()' for on-line help, or
HTML browser interface to help.
Type 'q()' to quit R.

> 
```

ENVIRONMENT


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
BrainHack tutorial 3 - RStudio
Go to file/function
Addins
BrainHack tutorial 3
Your first code.R
Source on Save
Run
Source
1:1
(Top Level)
R Script
Environment History Connections Build Tutorial
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
Console Terminal R Markdown Jobs
~/BrainHack tutorial 3/

```
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

Type 'q()' to quit R.

>
```

SCRIPT EDITOR:
A **script** is simply a text file containing a set of commands and comments. Here we write, save and run the code.

Here we see what objects we created

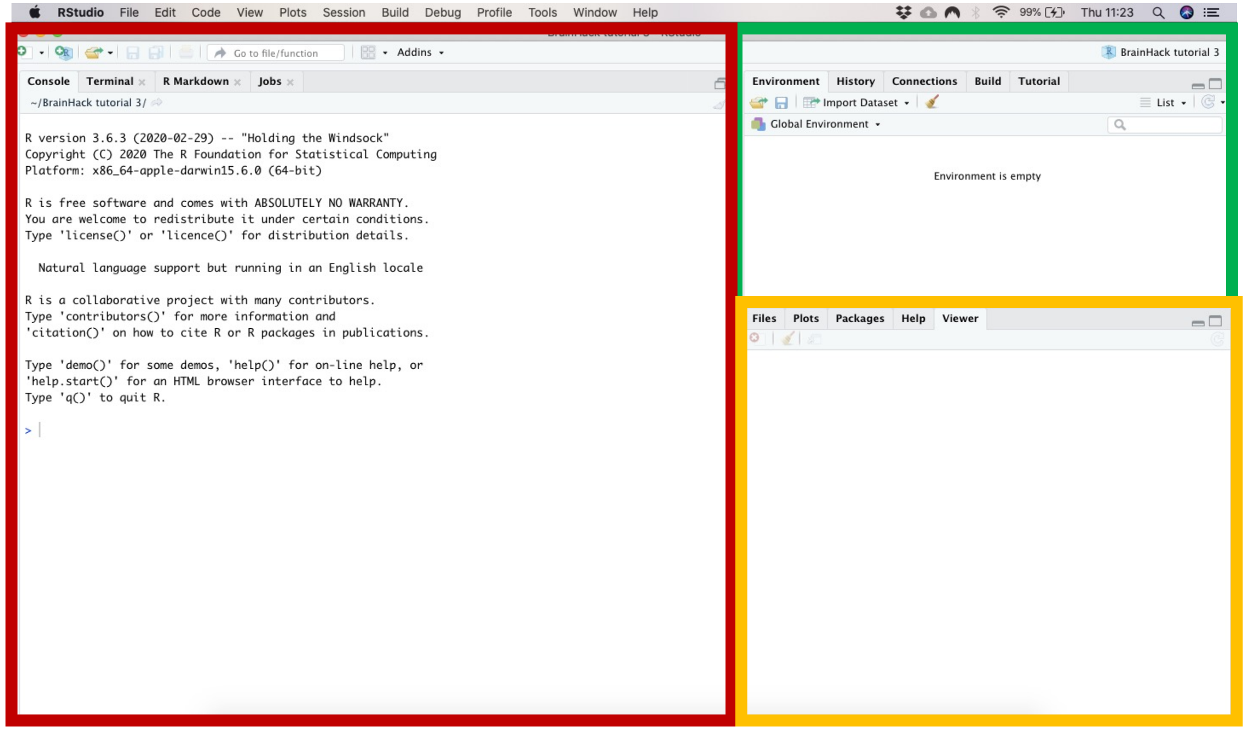
Here we can view and manage files, plots , packages and information about functions (help)

CONSOLE PANE:
You can write temporary codes here and run them.
We see here the outputs of the code.

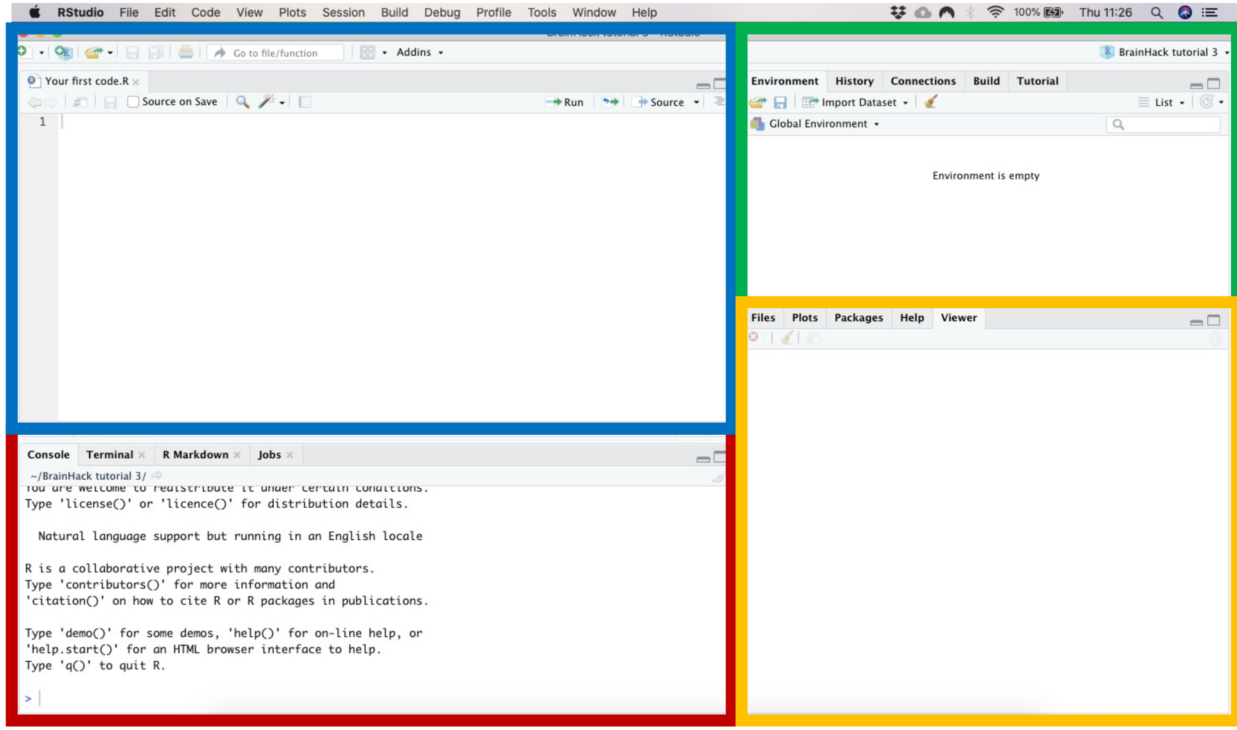
ENVIRONMENT

How Rstudio looks after you installed it / open it

How Rstudio looks after you open a script

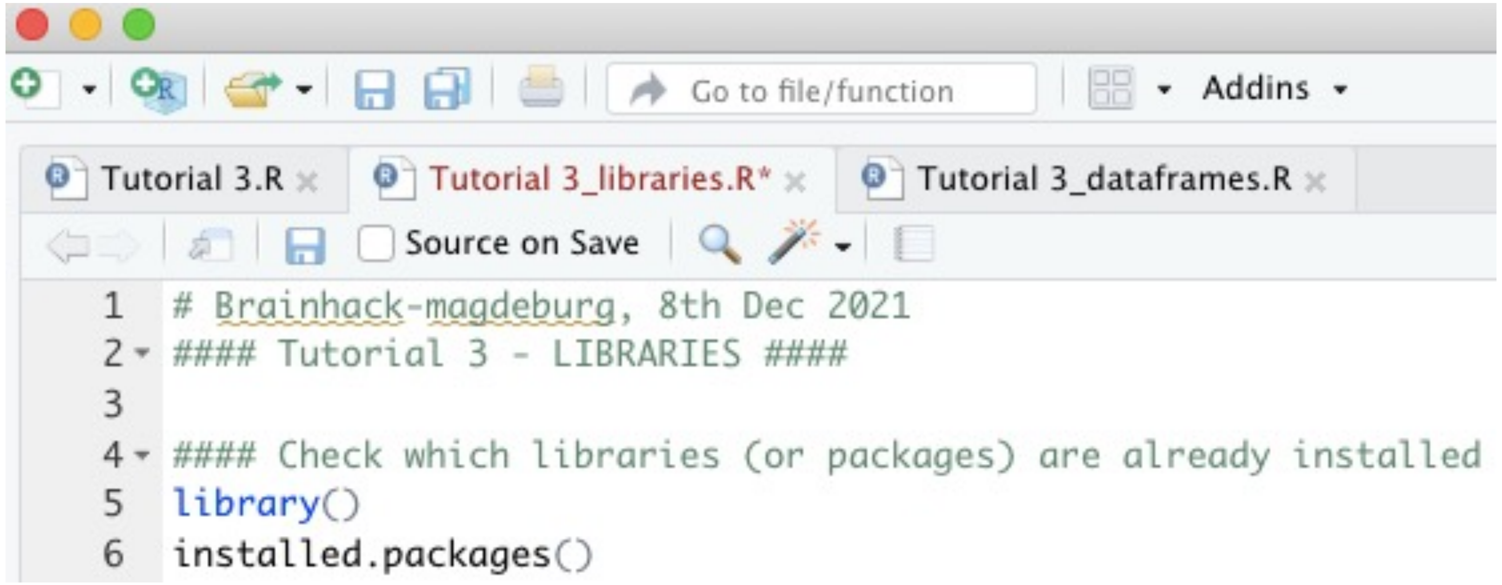


```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

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.

>
```

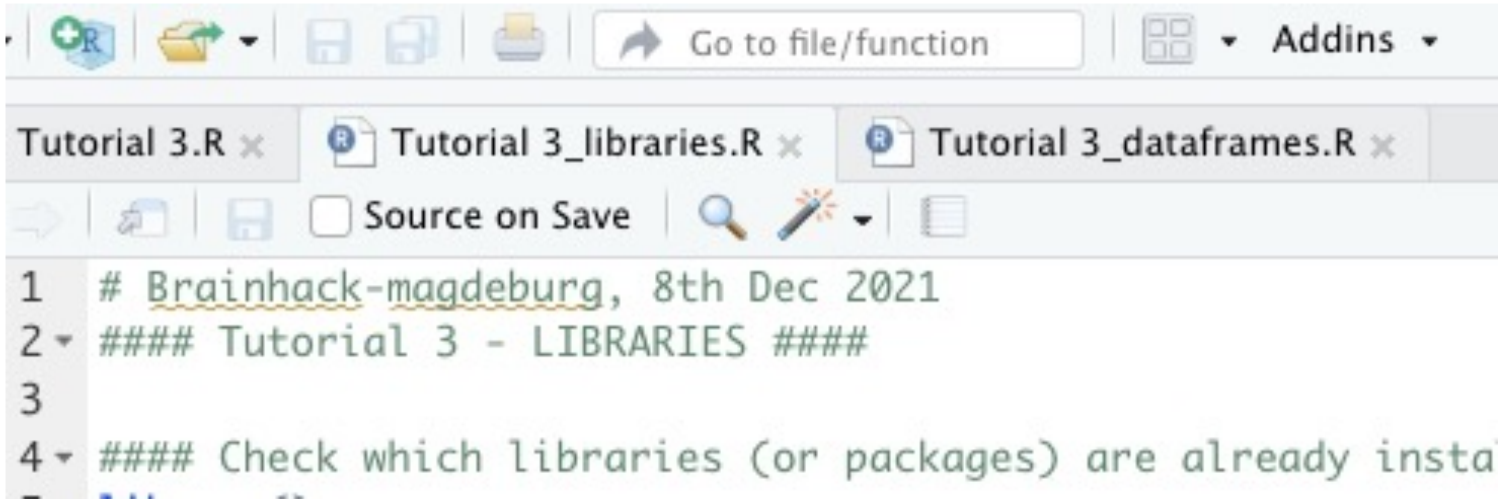


```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  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.

>
```

Save your script and exit

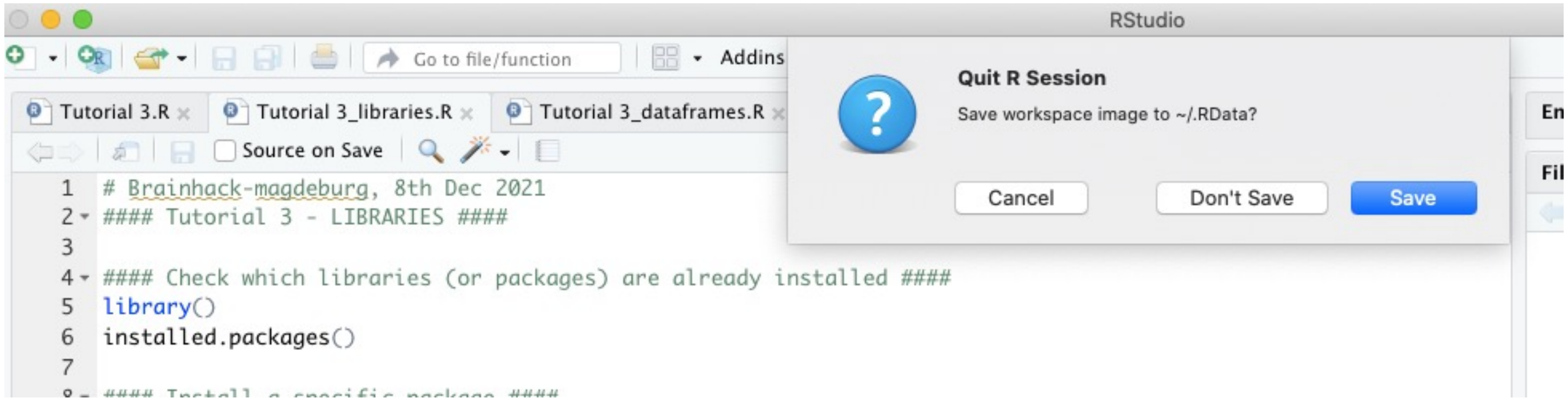


```r
# Brainhack-magdeburg, 8th Dec 2021
#### Tutorial 3 - LIBRARIES ####

#### Check which libraries (or packages) are already installed
library()
installed.packages()
```



```r
# Brainhack-magdeburg, 8th Dec 2021
#### Tutorial 3 - LIBRARIES ####

#### Check which libraries (or packages) are already insta
```

Save your script and exit

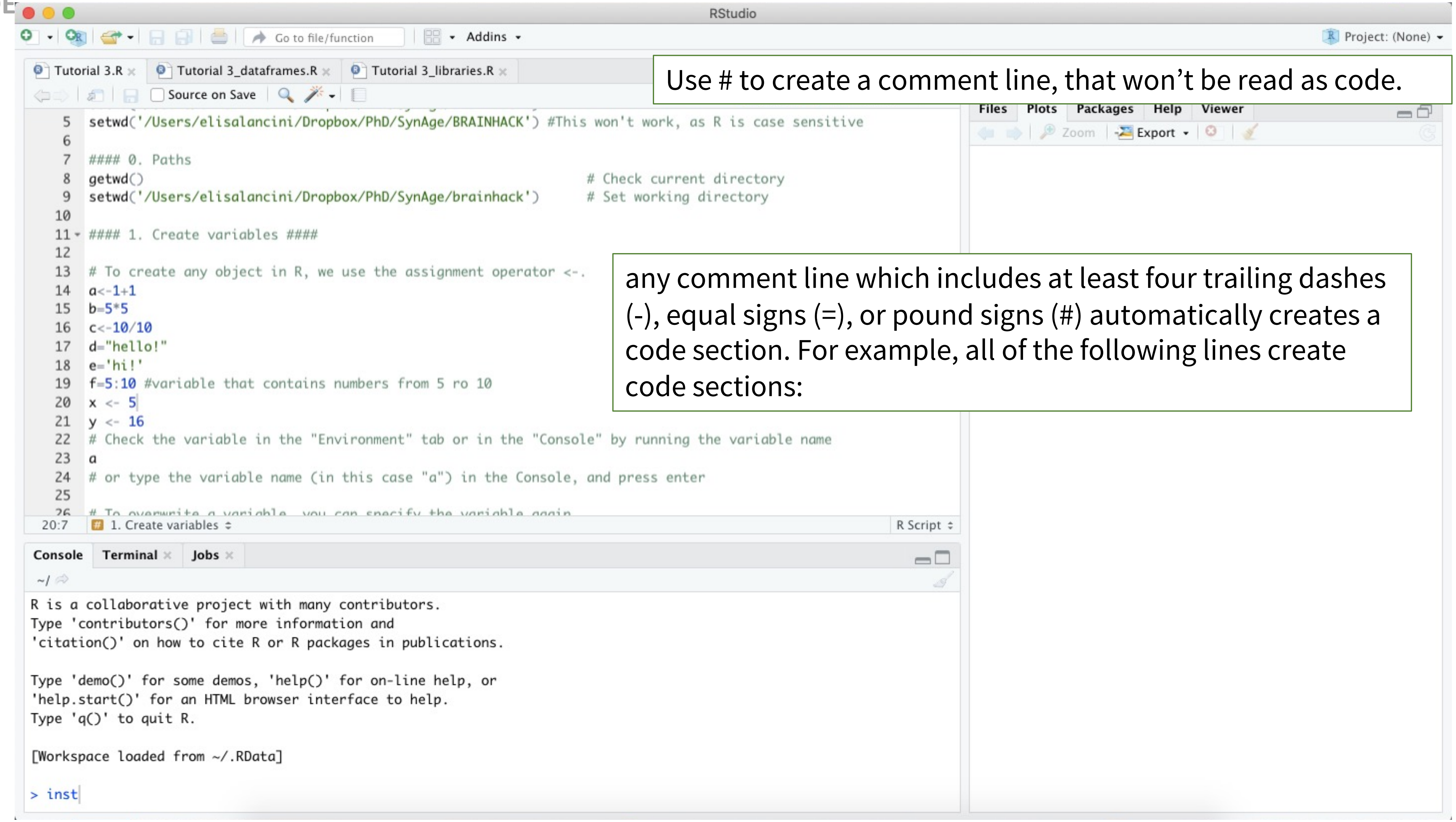


Quit R Session

Save workspace image to ~/.RData?

```
# Brainhack-magdeburg, 8th Dec 2021
#### Tutorial 3 - LIBRARIES ####

#### Check which libraries (or packages) are already installed ####
library()
installed.packages()

```

A CODE: how does it look like and how to run it

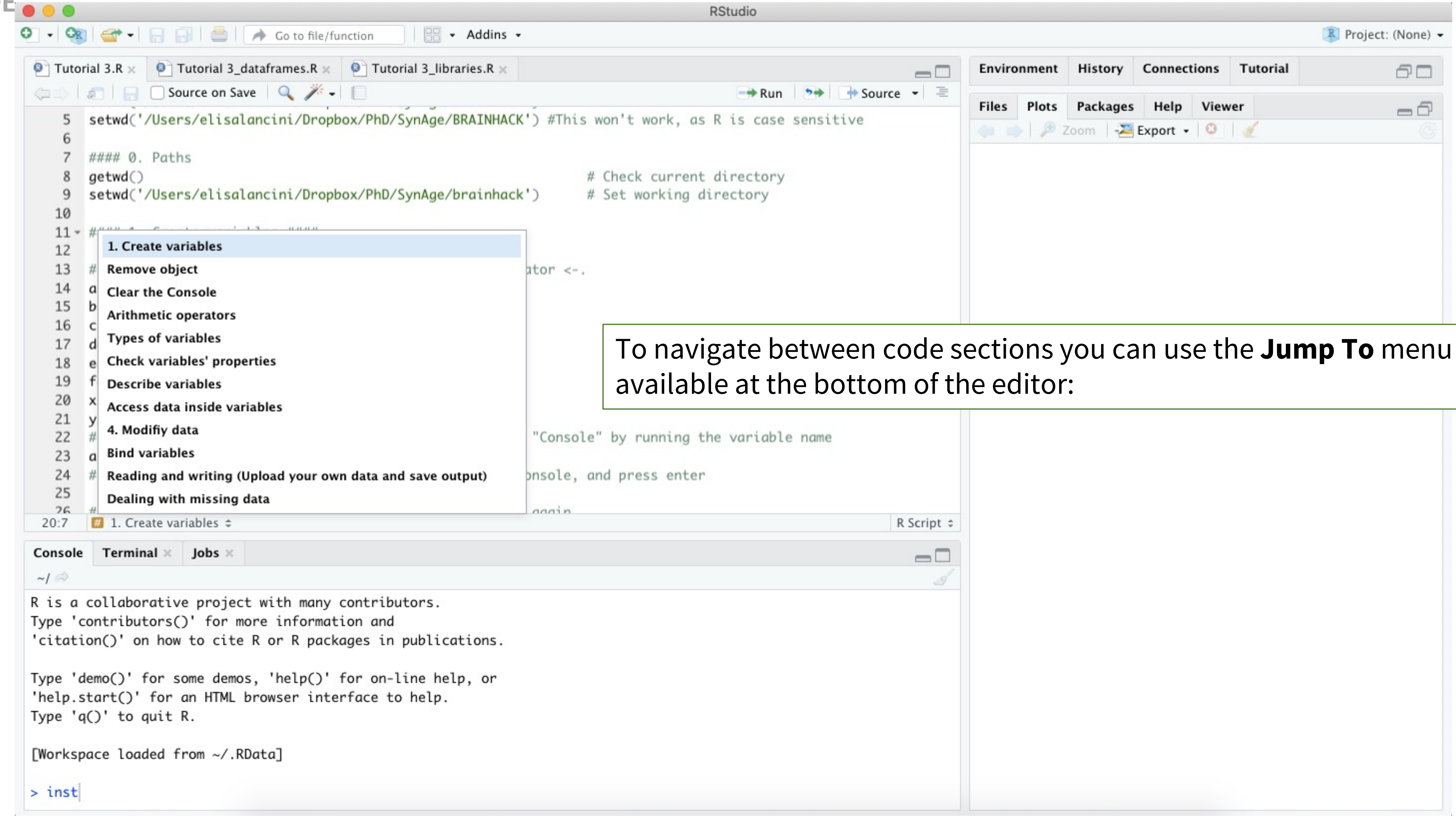


```
setwd('/Users/elisalancini/Dropbox/PhD/SynAge/BRAINHACK') #This won't work, as R is case sensitive

#### 0. Paths
getwd()                                                    # Check current directory
setwd('/Users/elisalancini/Dropbox/PhD/SynAge/brainhack')      # Set working directory

#### 1. Create variables ####

# To create any object in R, we use the assignment operator <-.
a<-1+1
b=5*5
c<-10/10
d="hello!"
e='hi!'
f=5:10 #variable that contains numbers from 5 ro 10
x <- 5
y <- 16
# Check the variable in the "Environment" tab or in the "Console" by running the variable name
a
# or type the variable name (in this case "a") in the Console, and press enter

# To overwrite a variable, you can specify the variable again
```

```
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.

[Workspace loaded from ~/.RData]

> inst
```

Use # to create a comment line, that won't be read as code.

any comment line which includes at least four trailing dashes (-), equal signs (=), or pound signs (#) automatically creates a code section. For example, all of the following lines create code sections:

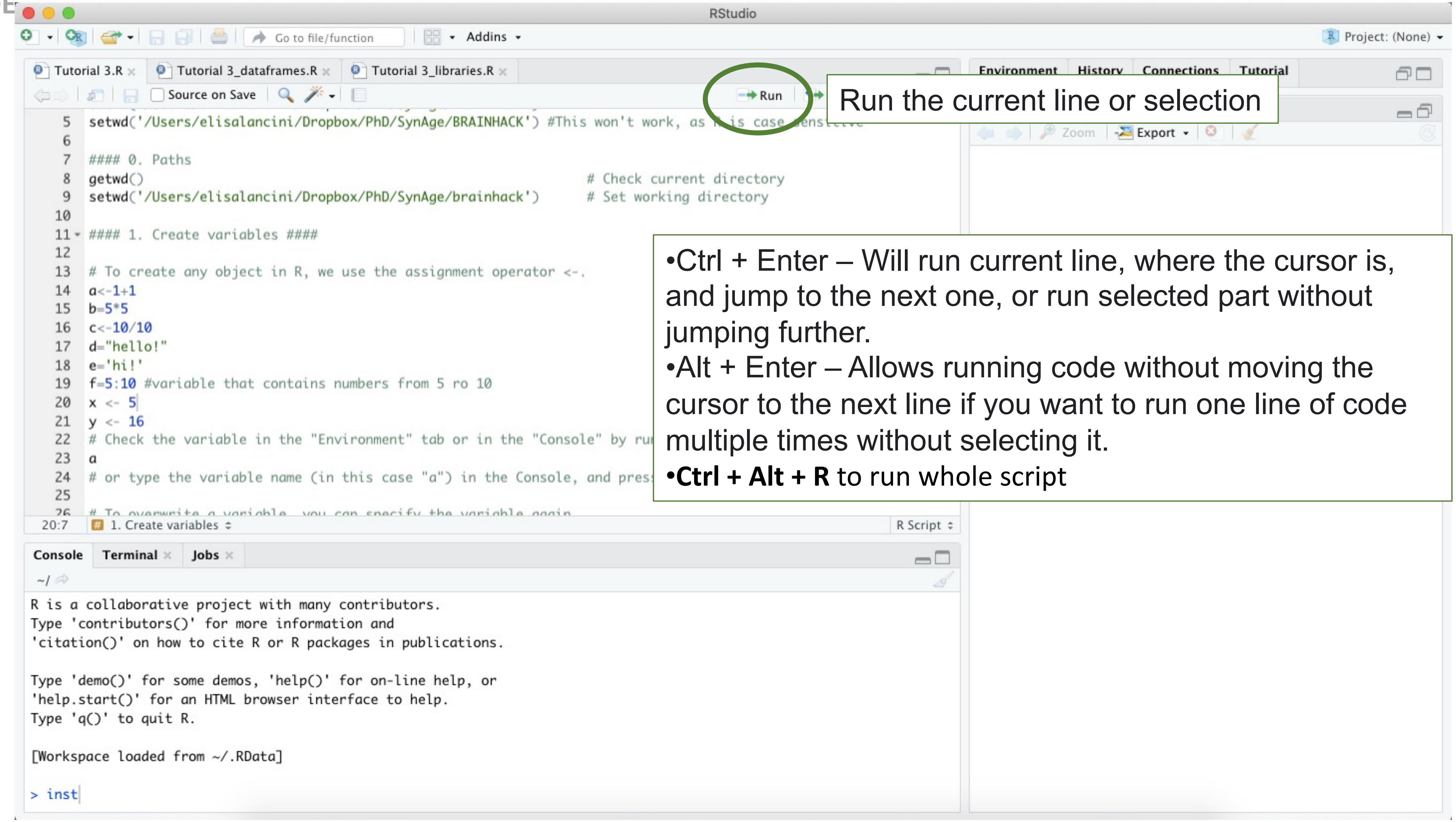


```
setwd('/Users/elisalancini/Dropbox/PhD/SynAge/BRAINHACK') #This won't work, as R is case sensitive

#### 0. Paths
getwd()                                                    # Check current directory
setwd('/Users/elisalancini/Dropbox/PhD/SynAge/brainhack')  # Set working directory

#
```

```
~/
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.

[Workspace loaded from ~/.RData]

> inst
```

To navigate between code sections you can use the **Jump To** menu available at the bottom of the editor:

CODE



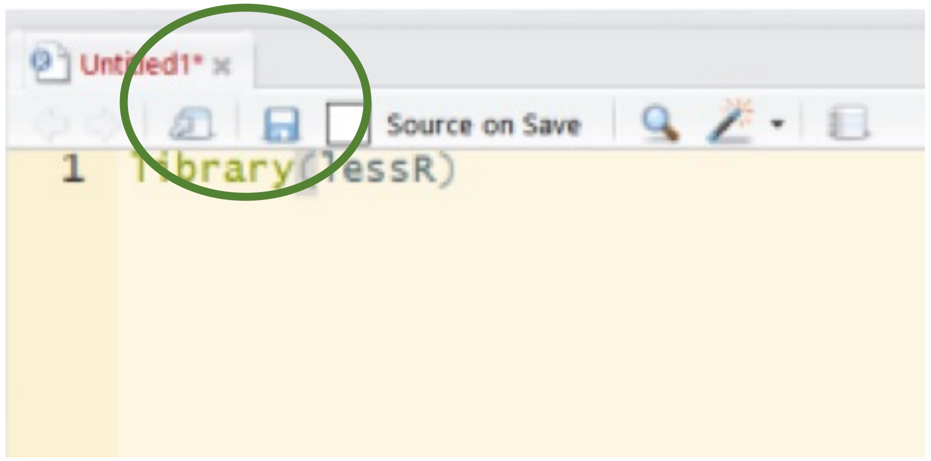
```
setwd('/Users/elisalancini/Dropbox/PhD/SynAge/BRAINHACK') #This won't work, as R is case sensitive

#### 0. Paths
getwd()                                                # Check current directory
setwd('/Users/elisalancini/Dropbox/PhD/SynAge/brainhack')      # Set working directory

#### 1. Create variables ####

# To create any object in R, we use the assignment operator <-.
a<-1+1
b=5*5
c<-10/10
d="hello!"
e='hi!'
f=5:10 #variable that contains numbers from 5 ro 10
x <- 5
y <- 16
# Check the variable in the "Environment" tab or in the "Console" by ru
a
# or type the variable name (in this case "a") in the Console, and pres

# To overwrite a variable, you can specify the variable again
```

```
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.

[Workspace loaded from ~/.RData]

> inst
```

Run the current line or selection

- Ctrl + Enter – Will run current line, where the cursor is, and jump to the next one, or run selected part without jumping further.
- Alt + Enter – Allows running code without moving the cursor to the next line if you want to run one line of code multiple times without selecting it.
- **Ctrl + Alt + R** to run whole script

You will see the code's name in red, and with an asterisk, whenever you change something in it.
This means that the code has been modified but the not saved.
Save it before closing it!



```
library(lessR)
```

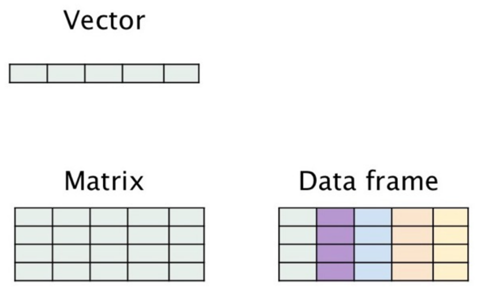
WHAT DO YOU FIND IN A CODE

WHAT DO YOU FIND IN A CODE

1. VARIABLES

Different types and structures

Variables	Example
integer	100
numeric	0.05
character	"hello"
logical	TRUE
factor	"Green"



2. FUNCTIONS

piece of code written to carry out a specified task. You need to specify some details in order to use it. Those details are called "arguments". Every function comes with a documentation, where you can check which arguments are required.

mean(x, trim = 0, na.rm = FALSE, ...)

3. PACKAGES (or LIBRARIES)

Collection of functions developed by the community to improve R functionalities or to add new ones.

You can install them once, and they will remain in your R studio.

However, everytime you want to use them, you shoul load them.

ggplot2

ggplot 2 is an enhanced data visualization package for R. Create stunning multi-layered graphics with ease.

Project Site Link

tidyr

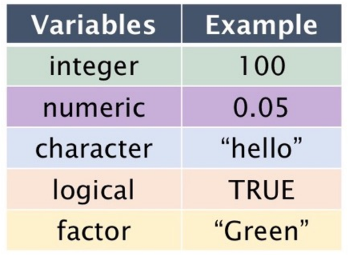
tidyr makes it easy to "tidy" your data. Tidy data is data that's easy to work with: it's easy to munge (with dplyr), visualise (with ggplot2 or ggvis) and model (with R's hundreds of modelling packages).

Project Paper Link

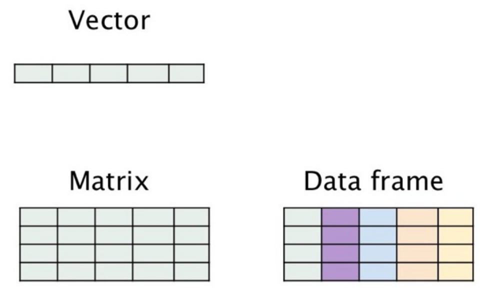
WHAT DO YOU FIND IN A CODE

1. VARIABLES

Different types and structures



Variables	Example
integer	100
numeric	0.05
character	“hello”
logical	TRUE
factor	“Green”



2. FUNCTIONS

piece of code written to carry out a specified task. You need to specify some details in order to use it. Those details are called “arguments”. Every function comes with a documentation, where you can check which arguments are required.

```
mean(x, trim = 0, na.rm = FALSE, …)
```

3. PACKAGES (or LIBRARIES)

Collection of functions developed by the community to improve R functionalities or to add new ones.

You can install them once, and they will remain in your R studio.

However, everytime you want to use them, you shoul load them.

ggplot2

ggplot 2 is an enhanced data visualization package for R. Create stunning multi-layered graphics with ease.

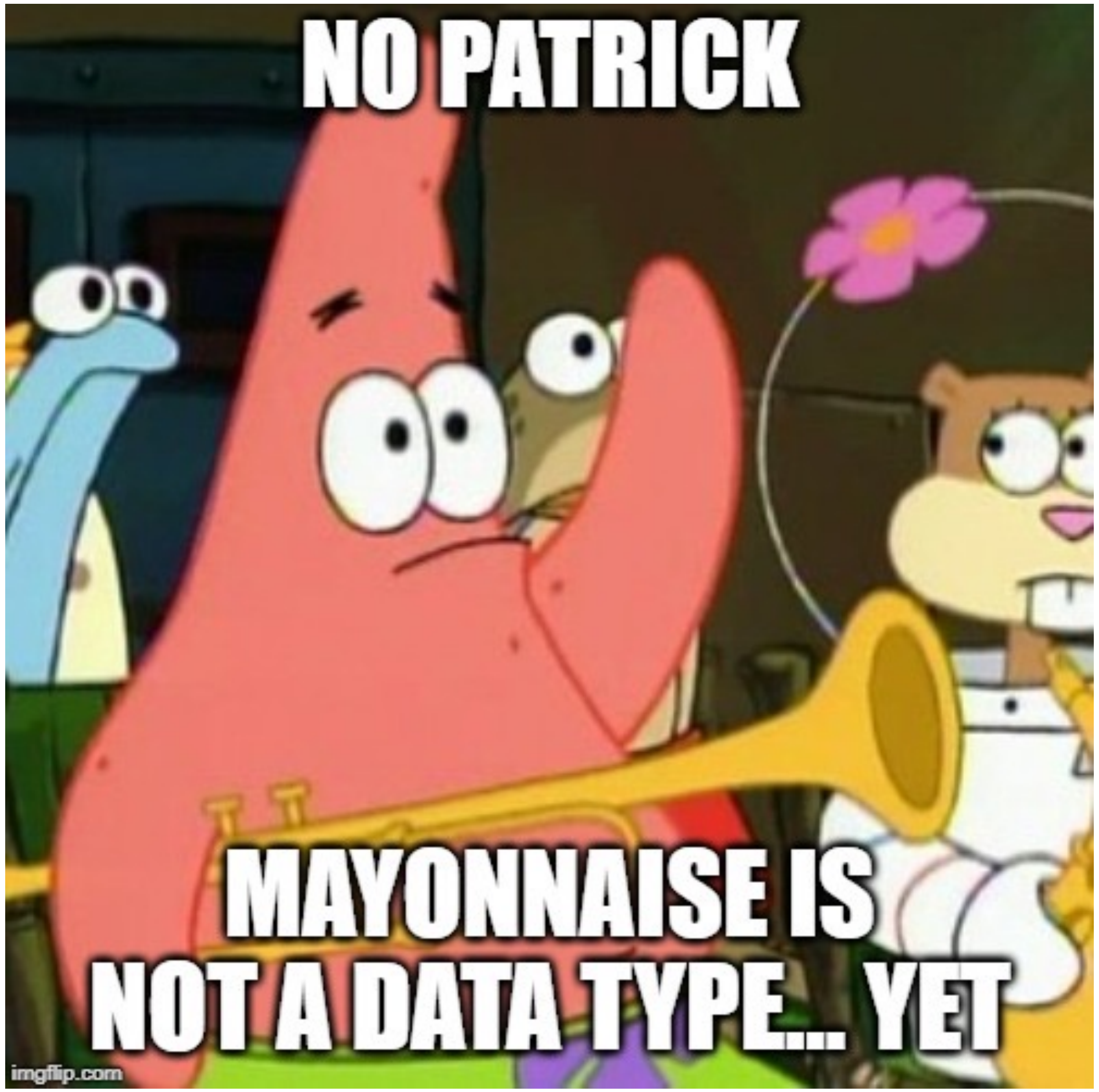
Project Site Link

tidyr

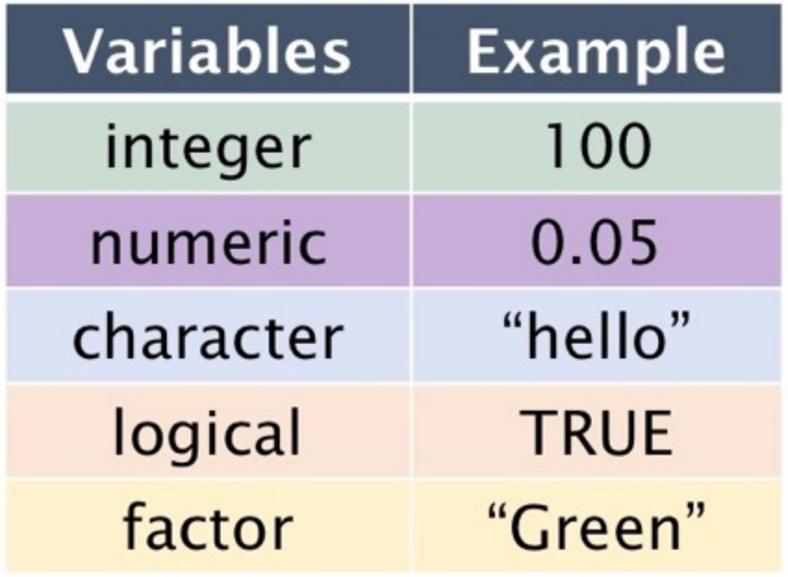
tidyr makes it easy to “tidy” your data. Tidy data is data that’s easy to work with: it’s easy to munge (with dplyr), visualise (with ggplot2 or ggvis) and model (with R’s hundreds of modelling packages).

Project Paper Link

DATA: Types and Structures



DATA TYPES



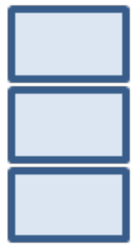
Variables	Example
integer	100
numeric	0.05
character	“hello”
logical	TRUE
factor	“Green”

Variable type	Type	Example
integer	Whole numbers	1, 100, -9
numeric	Decimals	0.1, -0.09, 234.567
character	Text	“A”, “hello”, “welcome”
logical	Booleans	TRUE or FALSE
factor	Categorical	“green”, “blue”, “red”, “purple”
missing	Logical	NA
empty	-	NULL

Elements of these data types may be combined to form data structures

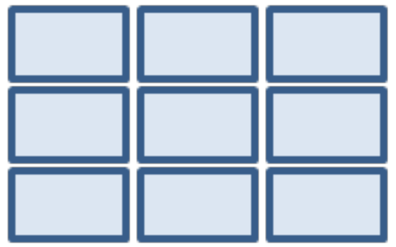
DATA STRUCTURES

Vector



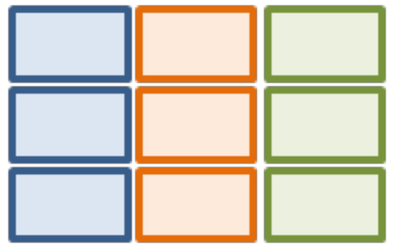
- 1 column or row of data
- 1 type (numeric or text)

Matrix

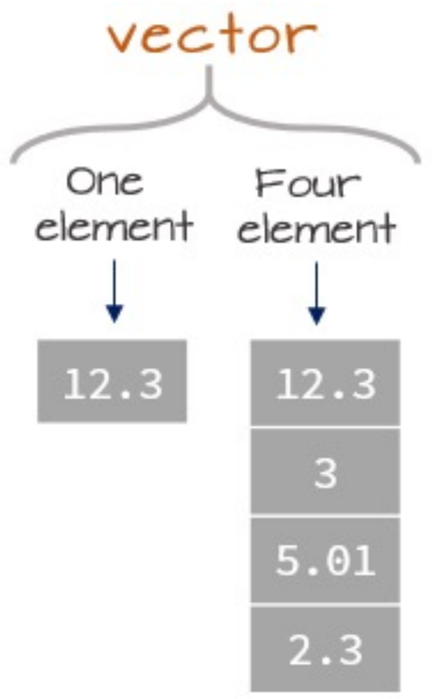


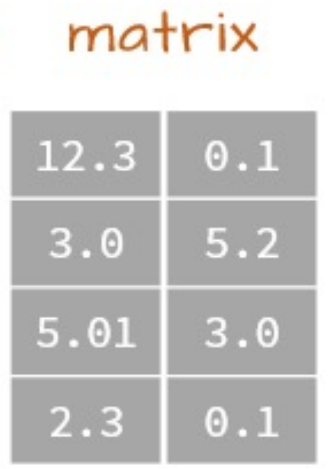
- multiple columns and/or rows of data
- 1 type (numeric or text)

Data Frame

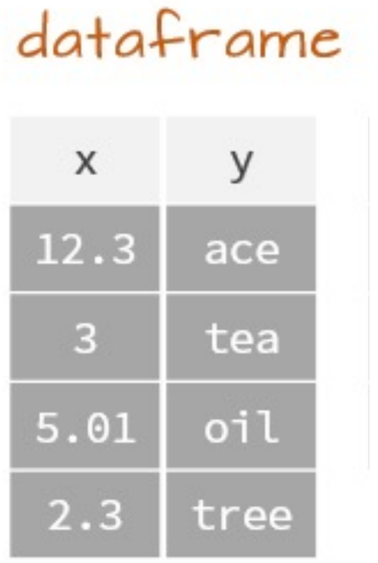


- multiple columns and/or rows of data
- multiple types




matrix

12.3	0.1
3.0	5.2
5.01	3.0
2.3	0.1


dataframe

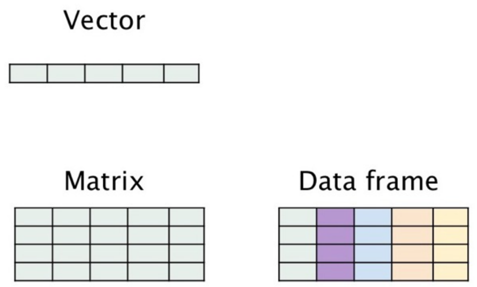
x	y
12.3	ace
3	tea
5.01	oil
2.3	tree

WHAT DO YOU FIND IN A CODE

1. VARIABLES

Different types and structures

Variables	Example
integer	100
numeric	0.05
character	“hello”
logical	TRUE
factor	“Green”



2. FUNCTIONS

piece of code written to carry out a specified task. You need to specify some details in order to use it. Those details are called “arguments”. Every function comes with a documentation, where you can check which arguments are required.

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

3. PACKAGES (or LIBRARIES)

Collection of functions developed by the community to improve R functionalities or to add new ones.

You can install them once, and they will remain in your R studio.

However, every time you want to use them, you should load them.

ggplot 2 is an enhanced data visualization package for R. Create stunning multi-layered graphics with ease.

Project Site Link

tidyr makes it easy to “tidy” your data. Tidy data is data that’s easy to work with: it’s easy to munge (with dplyr), visualise (with ggplot2 or ggvis) and model (with R’s hundreds of modelling packages).

Project Paper Link

FUNCTIONS

WHAT IS A FUNCTION

a function is **an object** so the R interpreter is able to pass control to the function, along with arguments that may be necessary for the function to accomplish the actions. The function in turn performs its task and returns control to the interpreter as well as any result which may be stored in other objects.

- Use a pre-existing one (Built-in functions)

```
mean()
```

- Create your own (User-defined Function)

```
function_name <- function(arg_1, arg_2, ...) {
Function body
}
```

https://www.tutorialspoint.com/r/r_functions.htm

Create your own (User-defined Function)

This is the actual name of the function.

Arguments are optional; that is, a function may contain no arguments. Also arguments can have default values.

```
function_name <- function(arg_1, arg_2, ...) {
Function body
}
```

Contains a collection of statements that defines what the function does.

Use a pre-existing one (Built-in functions)

```
mean(x, trim = 0, na.rm = FALSE, …)
```

```
makecake(flour, chocolate, need_the_oven= TRUE, …)
```

Use a pre-existing one (Built-in functions)

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Use a pre-existing one (Built-in functions)

This is the actual name of the function.

Arguments are optional; that is, a function may contain no arguments. Also arguments can have default values.

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Use a pre-existing one (Built-in functions)

This is the actual name of the function.

Arguments are optional; that is, a function may contain no arguments. Also arguments can have default values.

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Use a pre-existing one (Built-in functions)

See R documentation of this function to understand specific arguments
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/mean

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

An R object.

The fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

The logical value indicating whether NA values should be stripped before the computation proceeds.

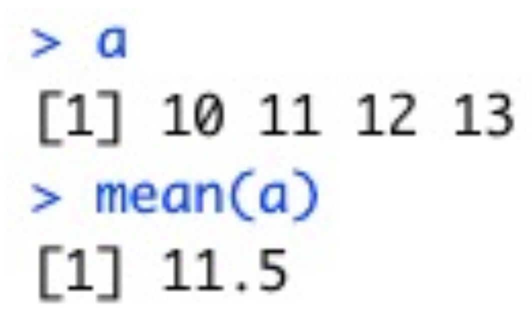
Use a pre-existing one (Built-in functions)

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

An R object.

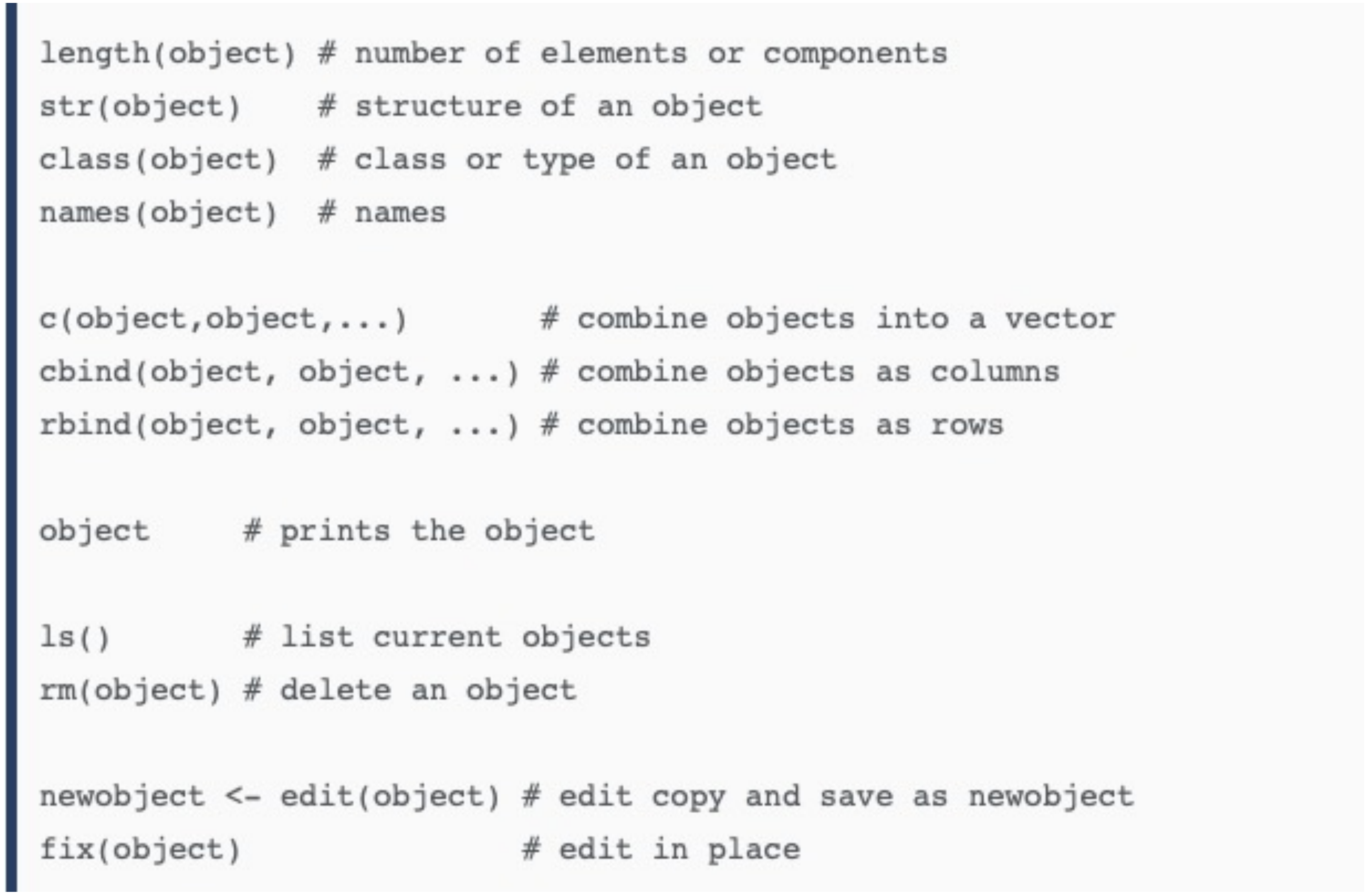
the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

a logical value indicating whether NA values should be stripped before the computation proceeds.



```
> a
[1] 10 11 12 13
> mean(a)
[1] 11.5
```

Use a pre-existing one (Built-in functions)



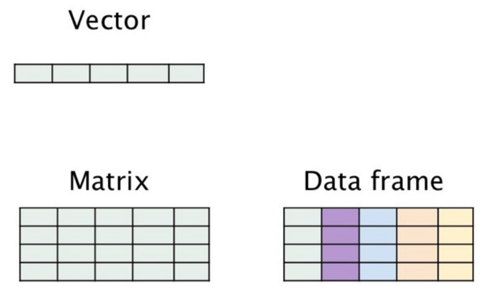
```
length(object) # number of elements or components
str(object)    # structure of an object
class(object)  # class or type of an object
names(object)  # names

c(object,object,...)       # combine objects into a vector
cbind(object, object, ...) # combine objects as columns
rbind(object, object, ...) # combine objects as rows

object     # prints the object

ls()       # list current objects
rm(object) # delete an object

newobject <- edit(object) # edit copy and save as newobject
fix(object)               # edit in place
```

WHAT DO YOU FIND IN A CODE

1. VARIABLES

Different types and structures

Variables	Example
integer	100
numeric	0.05
character	“hello”
logical	TRUE
factor	“Green”



2. FUNCTIONS

piece of code written to carry out a specified task. You need to specify some details in order to use it. Those details are called “arguments”. Every function comes with a documentation, where you can check which arguments are required.

mean(x, trim = 0, na.rm = FALSE, ...)

3. PACKAGES (or LIBRARIES)

Collection of functions developed by the community to improve R functionalities or to add new ones.

You can install them once, and they will remain in your R studio.

However, every time you want to use them, you shoul load them.

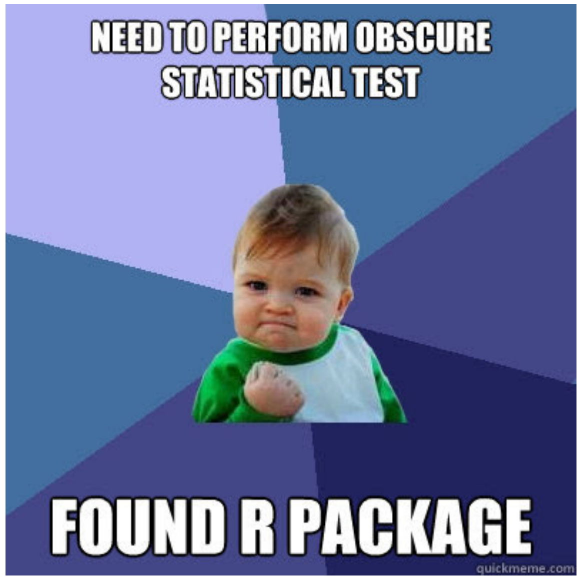
ggplot 2 is an enhanced data visualization package for R. Create stunning multi-layered graphics with ease.

Project Site Link

tidyr makes it easy to “tidy” your data. Tidy data is data that’s easy to work with: it’s easy to munge (with dplyr), visualise (with ggplot2 or ggvis) and model (with R’s hundreds of modelling packages).

Project Paper Link

PACKAGES (LIBRARIES)

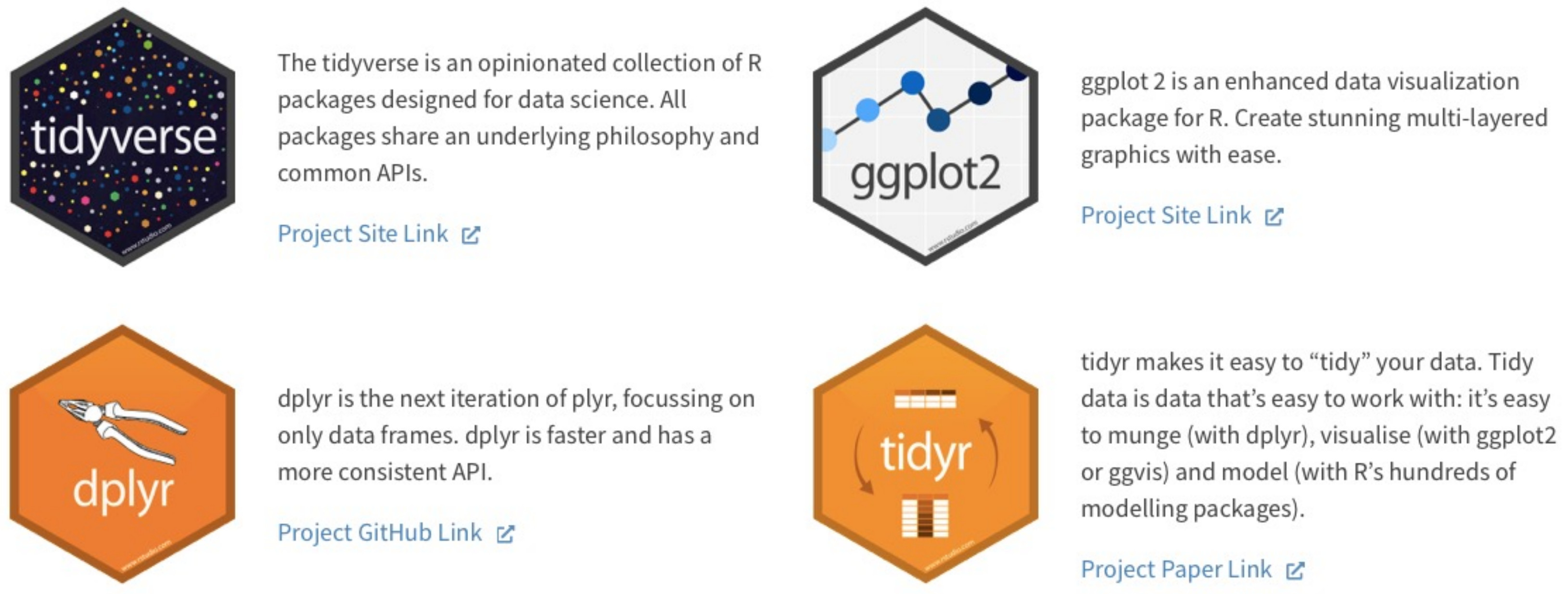


- R packages are **collections of functions and data sets developed by the community**. They increase the power of R by improving existing base R functionalities, or by adding new ones

- They are stored under a directory called "library" in the R environment..

- Packages which are already installed have to be loaded explicitly to be used by the R program that is going to use them.

https://www.rstudio.com/products/rpackages/



The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying philosophy and common APIs.

Project Site Link

ggplot 2 is an enhanced data visualization package for R. Create stunning multi-layered graphics with ease.

Project Site Link

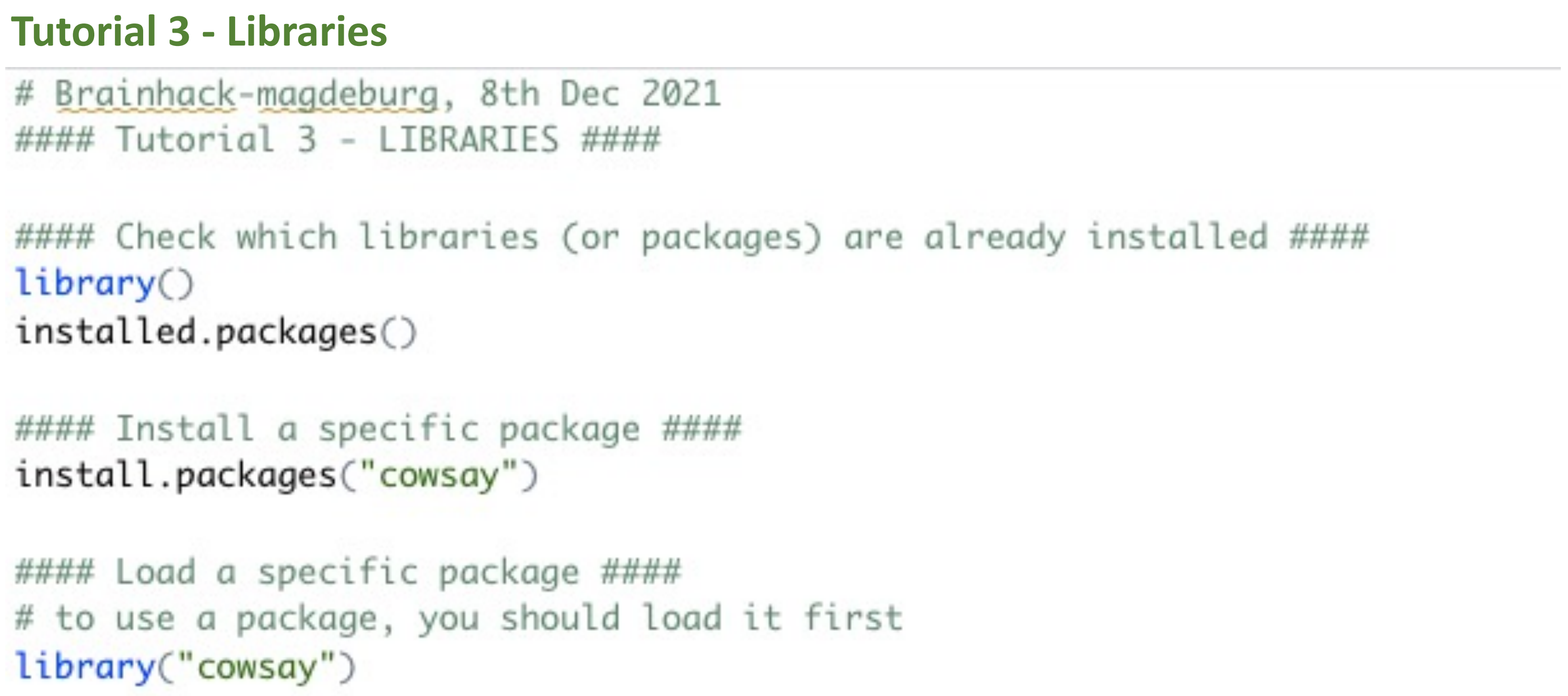
dplyr is the next iteration of plyr, focussing on only data frames. dplyr is faster and has a more consistent API.

Project GitHub Link

tidyr makes it easy to "tidy" your data. Tidy data is data that's easy to work with: it's easy to munge (with dplyr), visualise (with ggplot2 or ggvis) and model (with R's hundreds of modelling packages).

Project Paper Link

- All installed packages needs to be **load before being used.**
- This has to be done every time
- No need to install it again, just load it



Tutorial 3 - Libraries

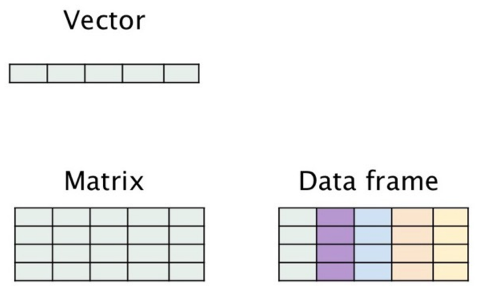
```r
# Brainhack-magdeburg, 8th Dec 2021
#### Tutorial 3 - LIBRARIES ####

#### Check which libraries (or packages) are already installed ####
library()
installed.packages()

#### Install a specific package ####
install.packages("cowsay")

#### Load a specific package ####
# to use a package, you should load it first
library("cowsay")
```

WHAT DO YOU FIND IN A CODE

1. VARIABLES

Different types and structures

Variables	Example
integer	100
numeric	0.05
character	“hello”
logical	TRUE
factor	“Green”



2. FUNCTIONS

piece of code written to carry out a specified task. You need to specify some details in order to use it. Those details are called “arguments”. Every function comes with a documentation, where you can check which arguments are required.

mean(x, trim = 0, na.rm = FALSE, ...)

3. PACKAGES (or LIBRARIES)

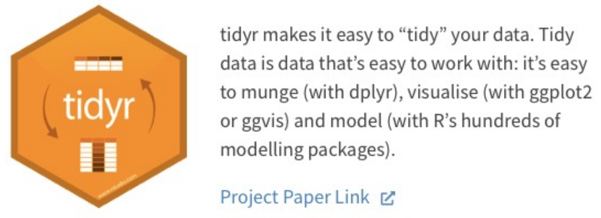
Collection of functions developed by the community to improve R functionalities or to add new ones.

You can install them once, and they will remain in your R studio.

However, every time you want to use them, you shoul load them.

ggplot 2 is an enhanced data visualization package for R. Create stunning multi-layered graphics with ease.

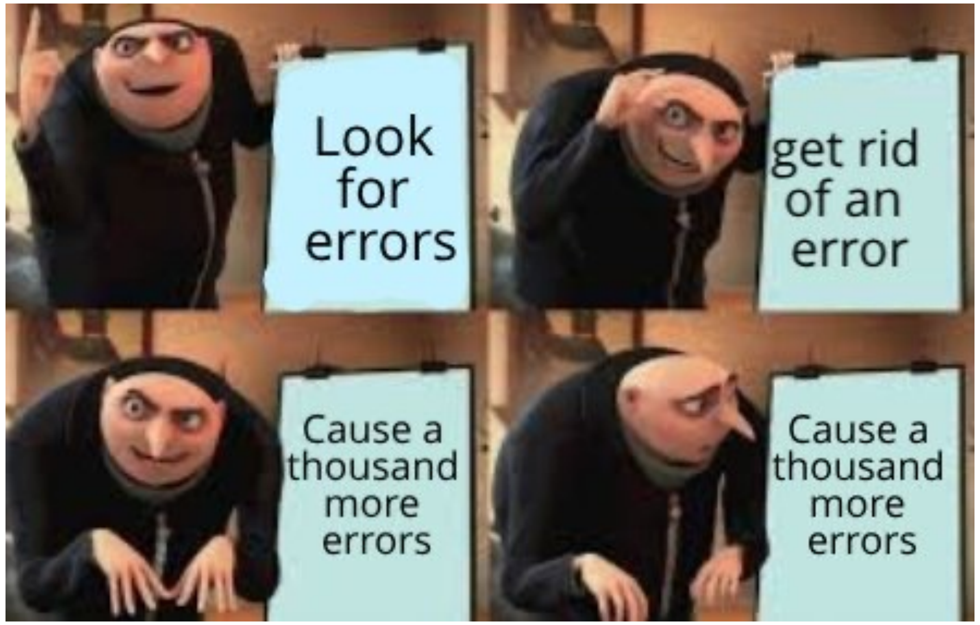
Project Site Link



tidyr makes it easy to “tidy” your data. Tidy data is data that’s easy to work with: it’s easy to munge (with dplyr), visualise (with ggplot2 or ggvis) and model (with R’s hundreds of modelling packages).

Project Paper Link

COMMON ERRORS



Read the error message

- The most common error in RStudio is Syntax errors

Read the Documentation

- Use the Help pane within RStudio
- Type in the console: ?help or ?(package) or ?(function)

Google it!

- But do not copy paste the entire error message, with your unique variable names!

Reproduce the Error

- Start a whole new code and make it very small so that you can isolate your Error

Ask for Help

- Github, Stackoverflow, Twitter, Slack/Discord communities or you may also ask for help from R and RStudio users on community.rstudio.com.

http://community.rstudio.com

What does my error mean?

•**‘could not find function’**.
•This error happen when an R package is not loaded properly or due to missing object like misspelling of the functions or data set name.

•**‘object not found’**. Check if the variable / object your refer to is actually present (you can do it easily by looking in the Environment Tab

•**‘non-numeric argument to a binary operator’**. T
This happen when we mix different vector value in calculation, for the example : numeric x characters.

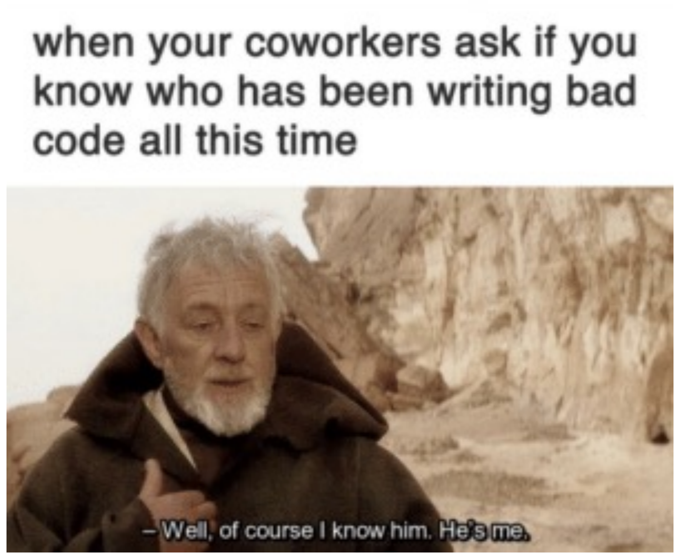
•**“replacement has”**.
•This error occurs when one tries to assign a vector of values to a subset of an existing object and the lengths do not match up.

•**‘Error in if’**.
It generally means the logical statement in “if (xxx) { …” is not yielding a logical value. Most of these have missing value where TRUE/FALSE is needed, meaning that the variable in xxx has NA in it.

•**“subscript out of bounds”**.
This error is likely to occur when one there is an error in a loop.

https://ismayc.github.io/rbasics-book/6-errors.html

CODE WRITING GOOD PRACTICE


when your coworkers ask if you know who has been writing bad code all this time
– Well, of course I know him. He's me.

GOOD PRACTICE

Understand your data

START EACH CODE WITH A DESCRIPTION OF WHAT IT DOES.

BE CONSISTENT WITH VARIABLES NAMES

Be consistent throughout your code

DOCUMENT

Although you must spend some extra time doing this, it pays up in the long run

BREAK THE CODE IN (WELL DESCRIVED) SECTIONS

By using # or #-

USE AUTOMATIC PACKAGES TO HELP CLEAN UP THE CODE

“lintr” and “goodpractice” are an example

SPECIFY WHAT IS NEEDED

Load the libraries at the beginning

Be explicit about input and output files

DO NOT REPEAT YOURSELF- AUTOMATE!

Use a loop or a function for recurrent actions

USE THE SAME DIRECTORY THROUGHOUT THE CODE

DO NOT SAVE THE SESSION HISTORY

so that older objects don’t remain in your environment **BUT SAVE THE CODE**

GOOD PRACTICE

Understand your data

START EACH CODE WITH A DESCRIPTION OF WHAT IT DOES.

BE CONSISTENT WITH VARIABLES NAMES

Be consistent throughout your code

DOCUMENT

Although you must spend some extra time doing this, it pays up in the long run

BREAK THE CODE IN (WELL DESCRIVED) SECTIONS

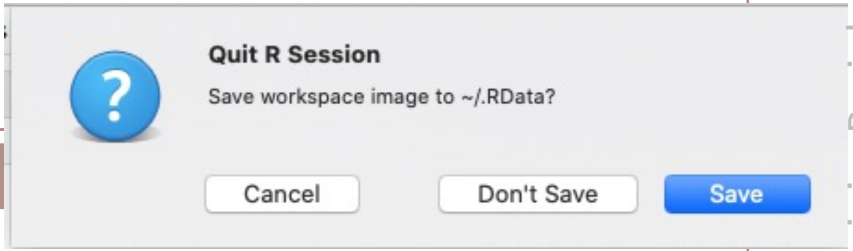
By using # or #-

USE AUTOMATIC PACKAGES TO HELP CLEAN UP THE CODE

"lintr" and "goodpractice" are an example

SPECIFY WHAT IS NEEDED

Load the libraries at the beginning
Be explicit about input and output files



Quit R Session

Save workspace image to ~/.RData?

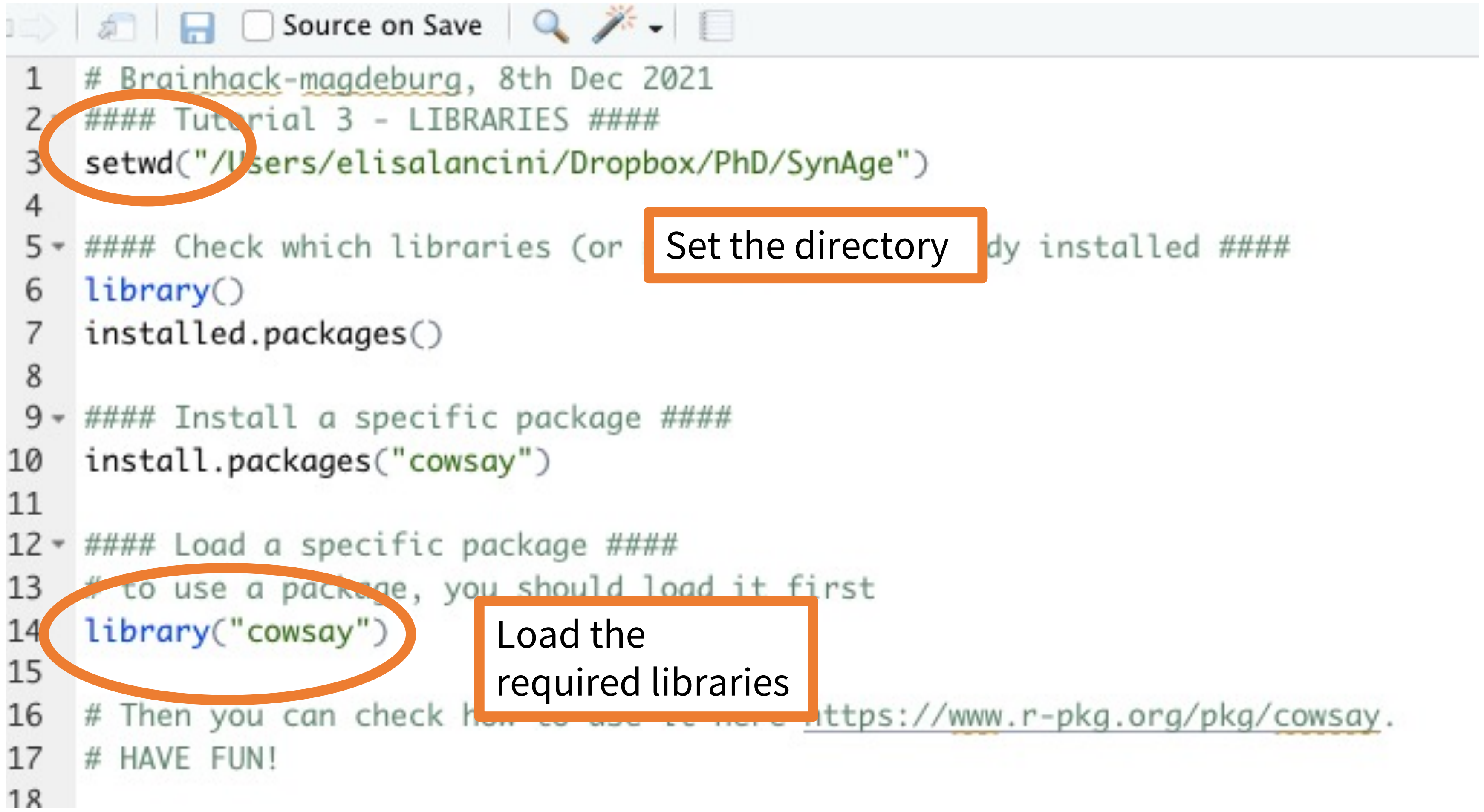
DO NOT REPEAT YOURSELF- AUTOMATE!

Use a loop or a function for recurrent actions

USE THE SAME DIRECTORY THROUGHOUT THE CODE

DO NOT SAVE THE SESSION HISTORY

so that older objects don't remain in your environment **BUT SAVE THE CODE**



```
# Brainhack-magdeburg, 8th Dec 2021
#### Tutorial 3 - LIBRARIES ####
setwd("/Users/elisalancini/Dropbox/PhD/SynAge")

#### Check which libraries (or                dy installed ####
library()
installed.packages()

#### Install a specific package ####
install.packages("cowsay")

#### Load a specific package ####
# to use a package, you should load it first
library("cowsay")

# Then you can check how to use it here https://www.r-pkg.org/pkg/cowsay.
# HAVE FUN!
```

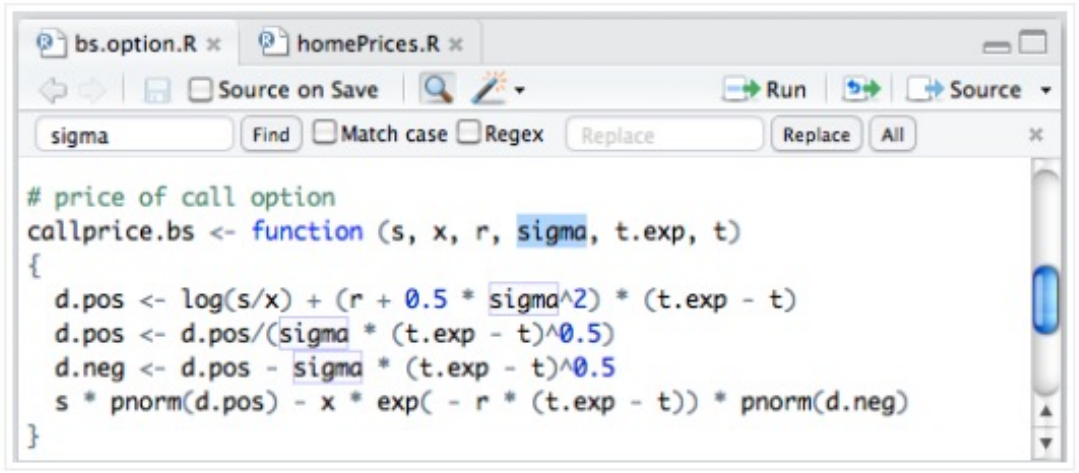
Set the directory

Load the required libraries

USEFUL TO KNOW

- https://support.rstudio.com/hc/en-us/articles/200484448-Editing-and-Executing-Code-in-the-RStudio-IDE
- https://www.r-bloggers.com/2019/04/r-studio-shortcuts-and-tips-2/
- https://www.rstudio.com/resources/cheatsheets/

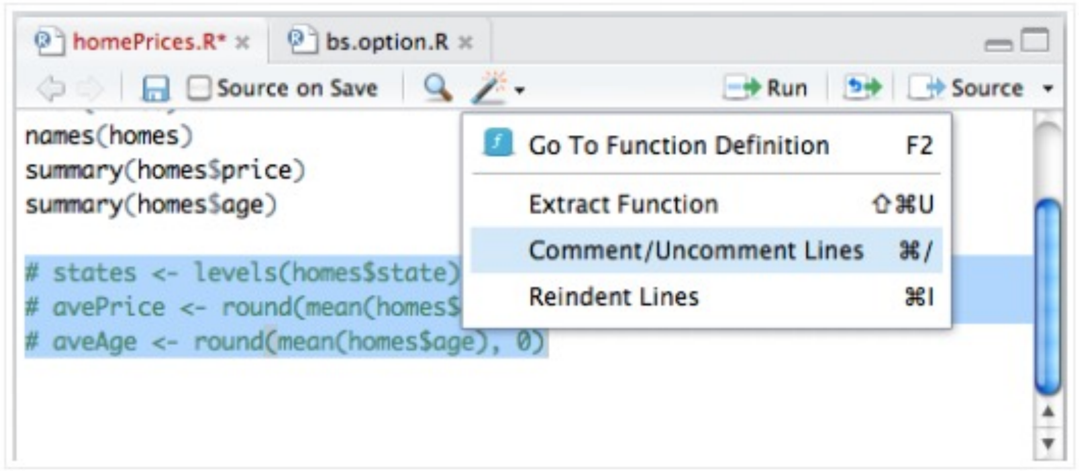
FIND AND REPLACE



```
# price of call option
callprice.bs <- function (s, x, r, sigma, t.exp, t)
{
  d.pos <- log(s/x) + (r + 0.5 * sigma^2) * (t.exp - t)
  d.pos <- d.pos/(sigma * (t.exp - t)^0.5)
  d.neg <- d.pos - sigma * (t.exp - t)^0.5
  s * pnorm(d.pos) - x * exp( - r * (t.exp - t)) * pnorm(d.neg)
}
```

Find and replace can be opened using the **Ctrl+F** shortcut key, or from the **Edit -> Find...** menu item.

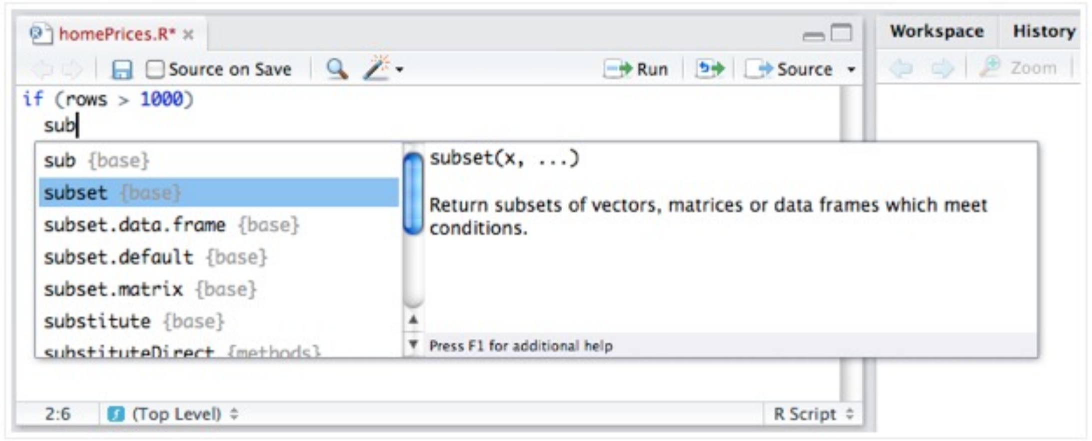
COMMENT / UNCOMMENT



```
names(homes)
summary(homes$price)
summary(homes$age)

# states <- levels(homes$state)
# avePrice <- round(mean(homes$
# aveAge <- round(mean(homes$age), 0)
```

You can comment and uncomment entire selections of code using the **Code -> Comment/Uncomment Lines** menu item (you can also do this using the **Command+Shift + C** keyboard shortcut)

AUTOMATIC COMPLETION



```
if (rows > 1000)
  sub
```

```
sub {base}
subset {base}
subset.data.frame {base}
subset.default {base}
subset.matrix {base}
substitute {base}
substituteDirect {methods}
```

subset(x, ...)

Return subsets of vectors, matrices or data frames which meet conditions.

Press F1 for additional help

RStudio supports the automatic completion of code using the **Tab** key. For example, if you have an object named pollResults in your workspace you can type poll and then **Tab**and RStudio will automatically complete the full name of the object.

SUMMARY

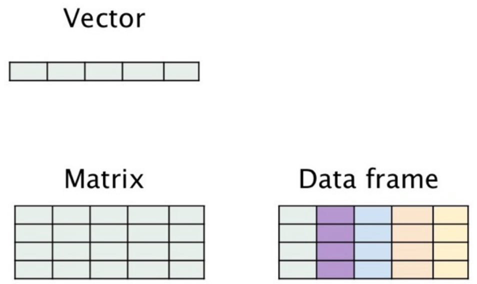
SUMMARY

TUTORIAL CODES: https://github.com/ElisaLancini/brainhack_magdeburg_2021

1. VARIABLES :

Different types and structures

Variables	Example
integer	100
numeric	0.05
character	“hello”
logical	TRUE
factor	“Green”



2. FUNCTIONS

piece of code written to carry out a specified task. You need to specify some details in order to use it. Those details are called “arguments”. Every function comes with a documentation, where you can check which arguments are required.

Tutorial 3.R ; Tutorial 3 dataframes.R

mean(x, trim = 0, na.rm = FALSE, …)

3. PACKAGES (or LIBRARIES)

Collection of functions developed by the community to improve R functionalities or to add new ones.

You can install them once, and they will remain in your R studio.

However, every time you want to use them, you should load them.

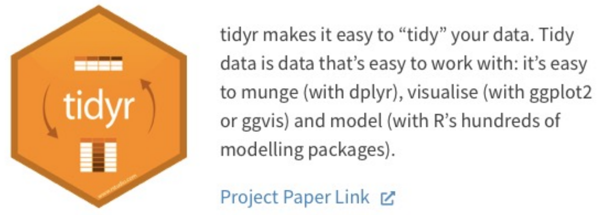
Tutorial 3 libraries.R

Tutorial 3 plots.R

ggplot2

ggplot2 is an enhanced data visualization package for R. Create stunning multi-layered graphics with ease.

Project Site Link

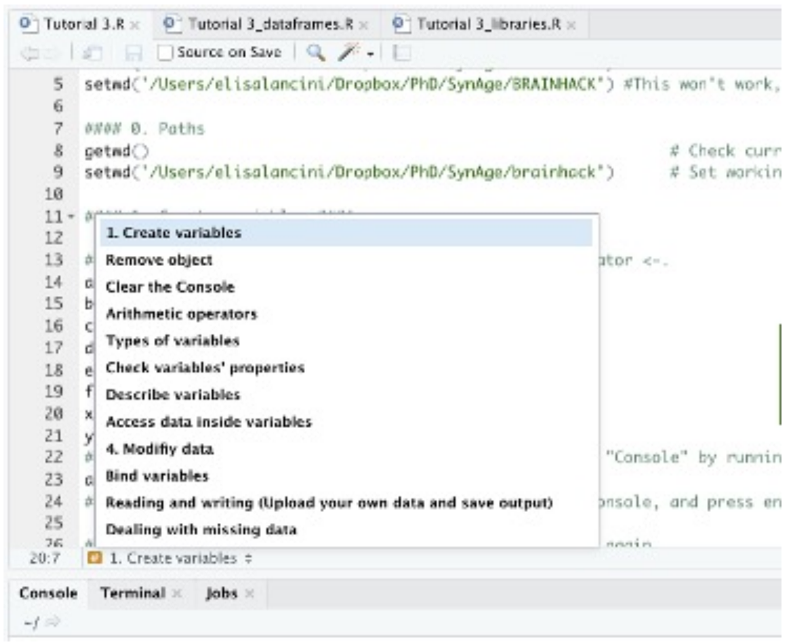

tidyr

tidyr makes it easy to “tidy” your data. Tidy data is data that’s easy to work with: it’s easy to munge (with dplyr), visualise (with ggplot2 or ggvis) and model (with R’s hundreds of modelling packages).

Project Paper Link

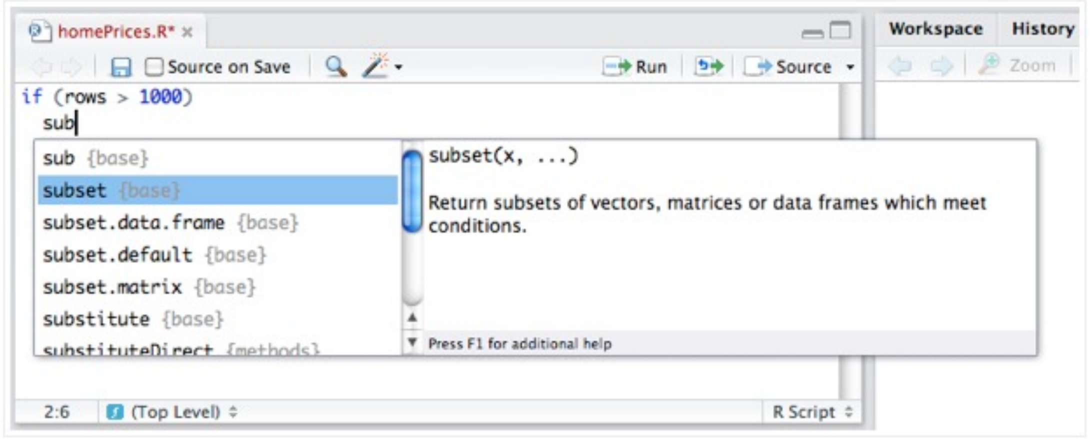
SUMMARY

4. # COMMENT and
DIVIDE YOUR CODE IN SECTIONS



```
setwd('/Users/elisalancini/Dropbox/PhD/SynAge/BRAINHACK') #This won't work,

#### 0. Paths
getwd()                                                  # Check curr
setwd('/Users/elisalancini/Dropbox/PhD/SynAge/brainhack')      # Set workin
```

5. USE THE AUTOMATIC COMPLETION



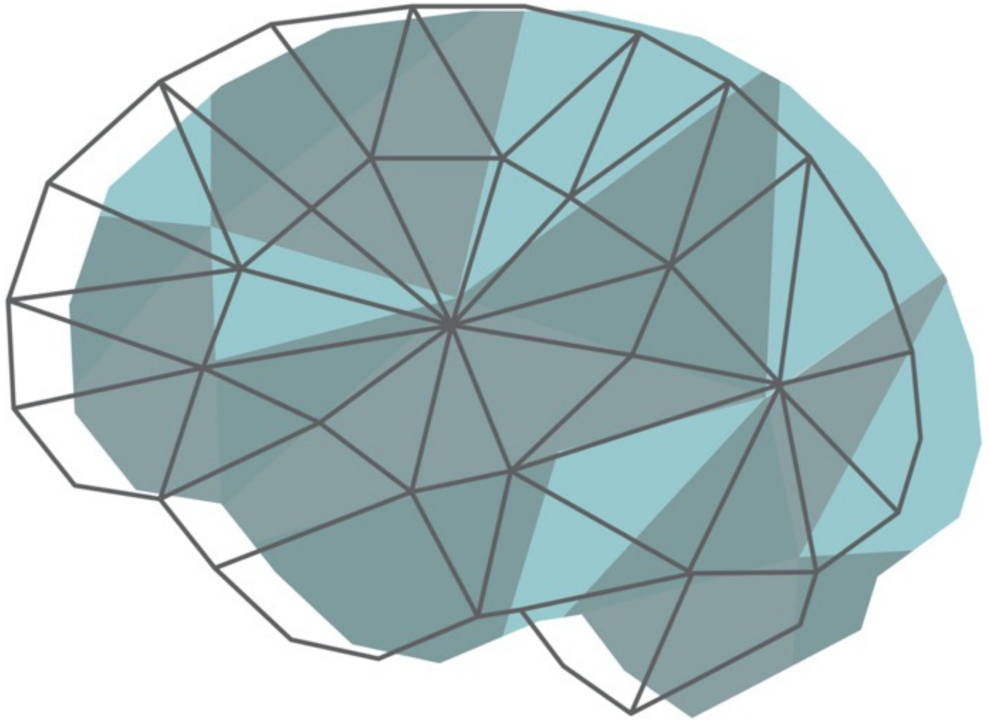
```
if (rows > 1000)
  sub
```

subset(x, ...)

Return subsets of vectors, matrices or data frames which meet conditions.

Tutorial 3 – Introduction to R(Studio)

TUTORIAL CODES: https://github.com/ElisaLancini/brainhack_magdeburg_2021



Brainhack Magdeburg
07.-08.12.2021

Elisa Lancini

elisa.lancini@dzne.de

@e_lancini

ElisaLancini