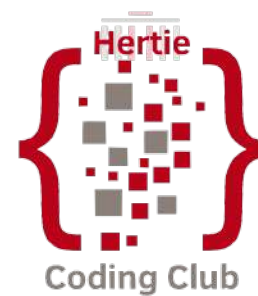# Hertie Coding Club

(Re) Introduction to R

Jorge Roa

Welcome

Hertie Coding Club

Agenda for today

- Objective of the club. Why this club?

- Who we are?

- Motivation

- Dinamic of the sessions

- Slack

- GitHub and GitHub Desktop

- R and RStudio

- Customize our RStudio appearance

- Create our first project

- RStudio elements

- Working directory

- Packages

- Objects

- Summary

- Next session

(Re) Introduction to R

{ } Hertie Coding Club

Objective of the club.
Why this club?

{ } Hertie Coding Club

Objective of the club. Why this club?

- 💻 Data is a relevant trend and us, as future policy makers, must have those skills.

- 👷 **Programming**: You are developing your problem-solving skills. The time you spend learning how to code and resolving code problems makes us think more logically.

- 📣 **Gap**: A lot of people want to learn code, but you must take Stats I and Stats II to have the right to apply to the Introduction to Data Science Course. Why should we not start learning from the first semester?

- 🚂 **Democratize learning code**: No matter if you are from the MPP, MIA, or another program, everyone should learn how to code and possess those abilities; we must exploit our potential.

{ } Hertie Coding Club

Who are we?

Hertie Coding Club

Who are we?

Instructors

Gabriel Zech

MDS 2023

Project Manager

(Data Science Lab)
Bertelsmann Stiftung

Carmen Garro

MDS

2024

Jorge Roa

MDS

2024

Lukas Warode

MDS 2023

Data Researcher

POLITICO

Rodrigo Dornelles

MDS

2024

{ } Hertie Coding Club

Who are we?

Communications

Ileana Marroquin (MPP 2023)

Coordinator alligator

Ridhima Singh (MPP 2024)

Coordinator alligator

Mary Nguyen (MPP 2023)

Celebrity promoter (Social Media)

Chhengpor Aun (MIA 2024)

Celebrity promoter (Social Media)

Lisa Bewermeier (MPP 2024)

Celebrity promoter (Social Media)

Natascha Schoepl (MPP 2023)

Central office emissary

Abigail Pena Alejos (MPP 2023)

Central office emissary

Kaja Buxrud (MPP 2024)

Promotion emotion

Issac John (MPP 2024)

Promotion emotion

Fernando Segovia (MIA 2024)

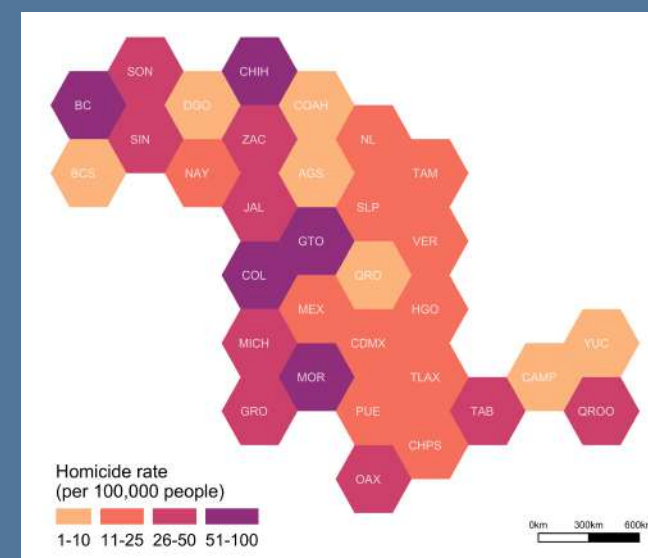Promotion emotion

Aayran Salman (MPP 2024)

Canva Wizard

{ } Hertie Coding Club

Motivation

Hertie Coding Club

Motivation

This was my first graph in R

How I improved

{ } Hertie Coding Club

Motivation

Hertie Coding Club

Motivation

Hertie Coding Club

Hertie Coding Club

What is the key to learn how to code
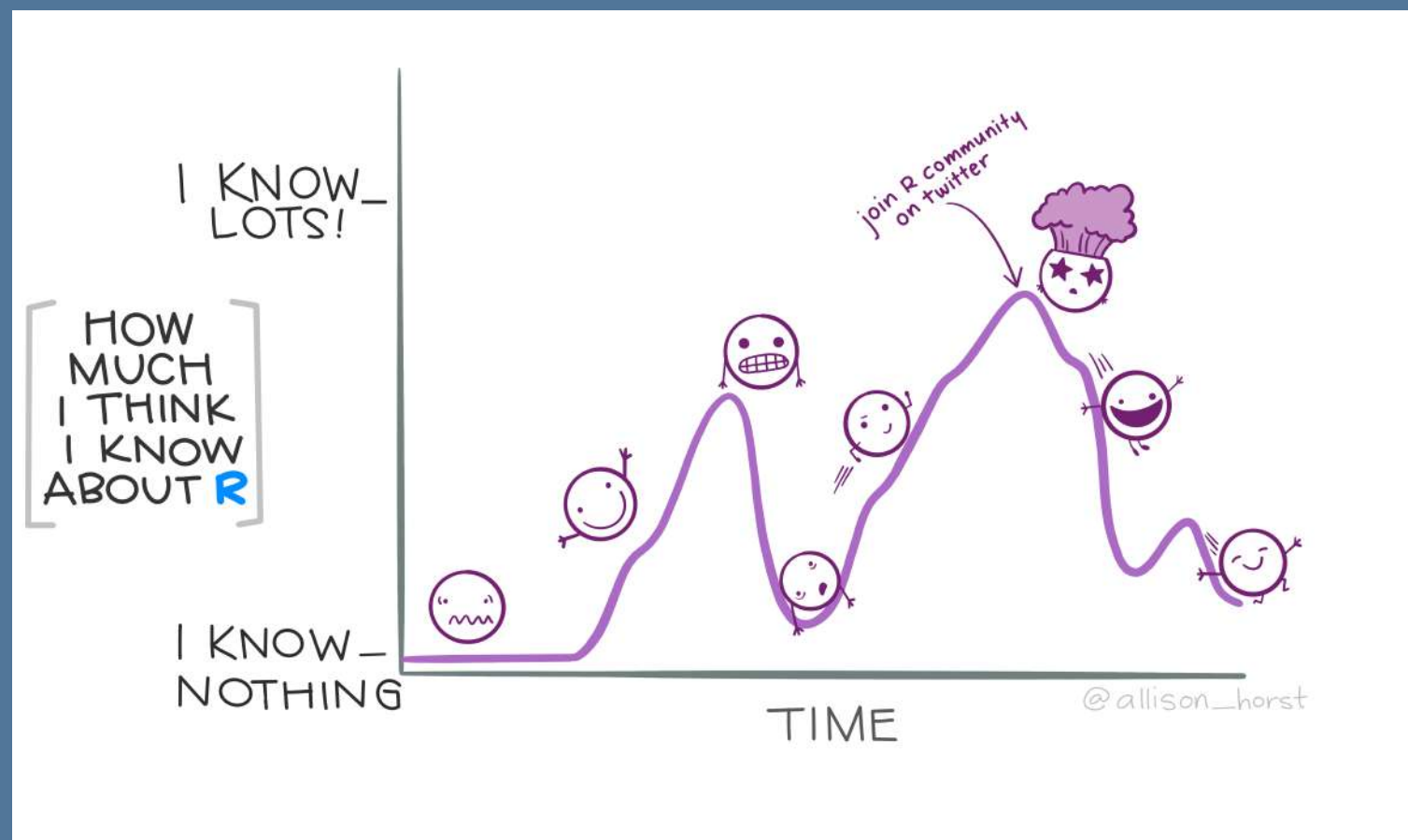
- Learning a new thing can be difficult sometimes.
- Getting errors can be scary: don't worry, we are here for you.
- In my opinion: the more errors you get, the better. Why? Because you learn from those errors. (It doesn't apply to real life :p)
- From my own experience: patience.
- The goal of the club is help us each other learn how to code.

Hertie Coding Club

Dynamic of the sessions

Hertie Coding Club

Dynamic of the sessions

- In the short run, we will have sessions for introduction to R (weekly)

- We hope this semester, we can have specific sessions for advanced R users and other languages (Python, SQL).

- We will record the sessions for future reference.

{ } Hertie Coding Club

Slack
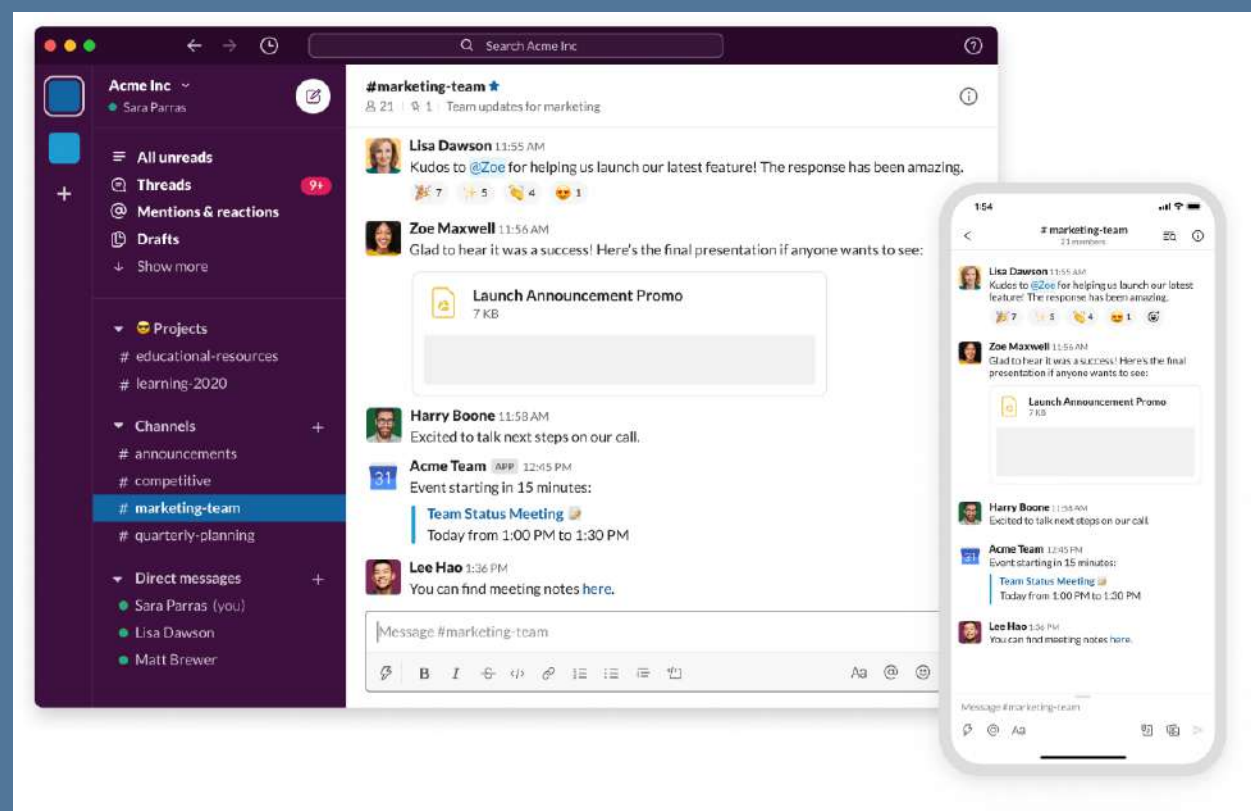
- **Permanent platform for Hertie Community**
- Like a WhatsApp for code.
- Share code, share files, and ask questions.
- One of the most popular platforms in the data science world.

Hertie Coding Club

GitHub and GitHub Desktop
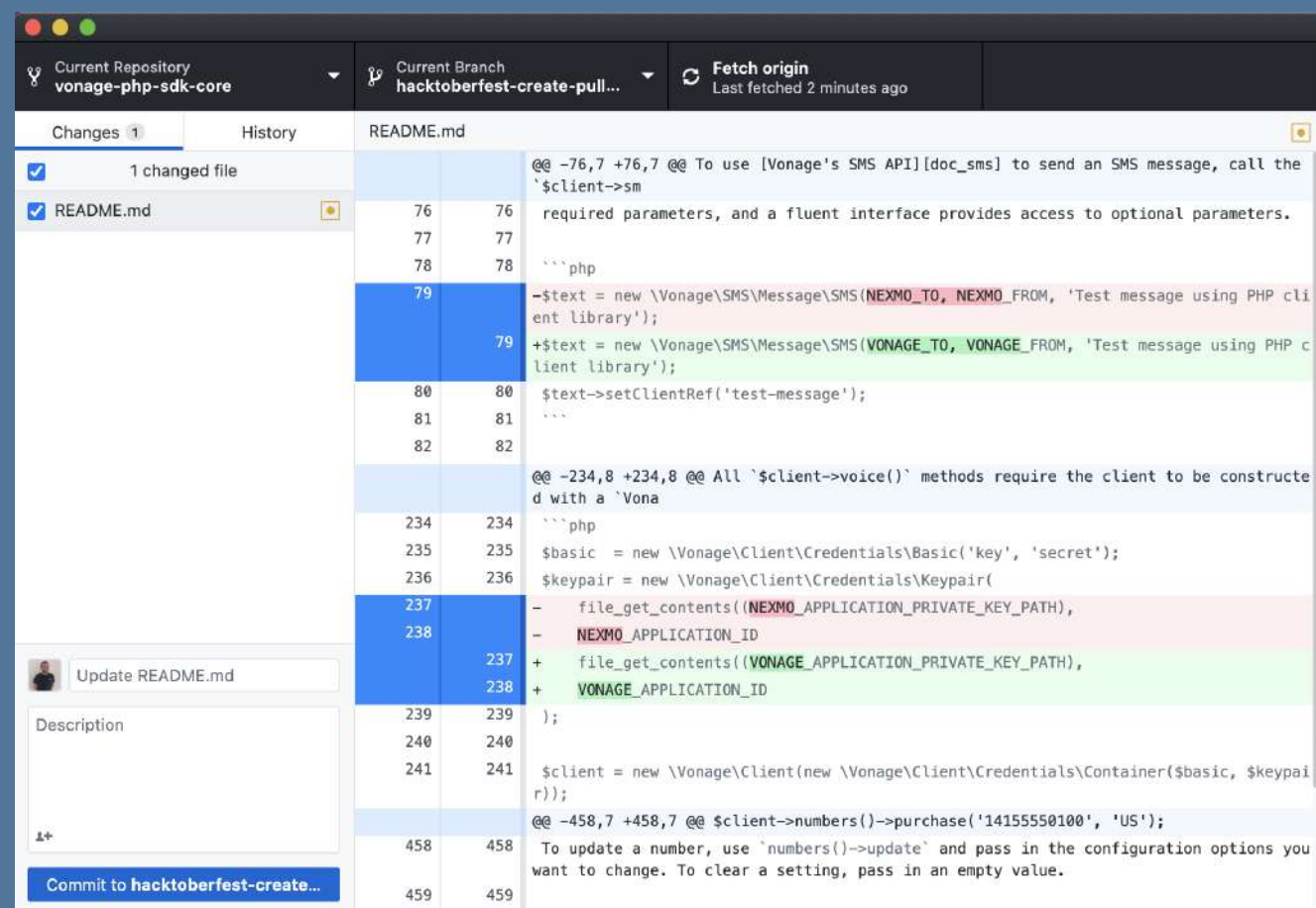
- Website and cloud-based service to store and manage code

- Git IDE: used in the programming world. It is used for tracking changes in the source code during software development.

- It makes it easier for individuals and teams to use Git for version control and collaboration.

Hertie Coding Club

R and RStudio

Hertie Coding Club

R and RStudio

Why R?

- Open source

- One of the most used language programs in the world

- Works for every type of data (shape and sizes)

- Easy to learn it (We will help you with that)

- More than 18,000 packages

Customize our RStudio appearance

Hertie Coding Club

Customize our RStudio appearance

Welcome to RStudio

- First window that appears when you open RStudio

- Let's change the appearance of RStudio

Hertie **C**oding **C**lub
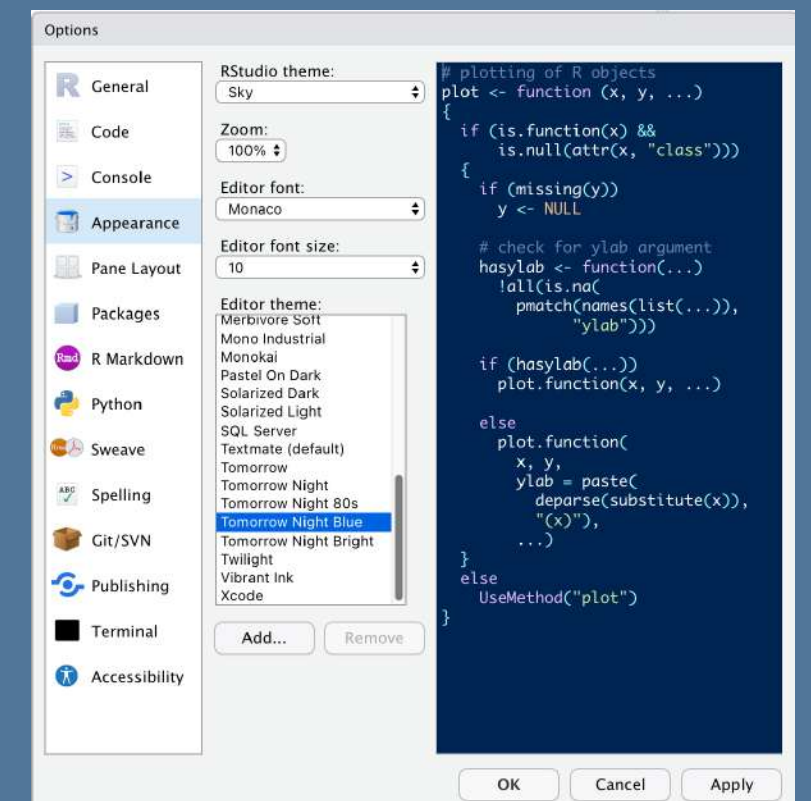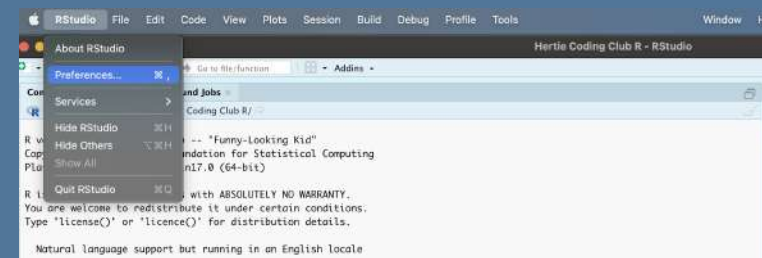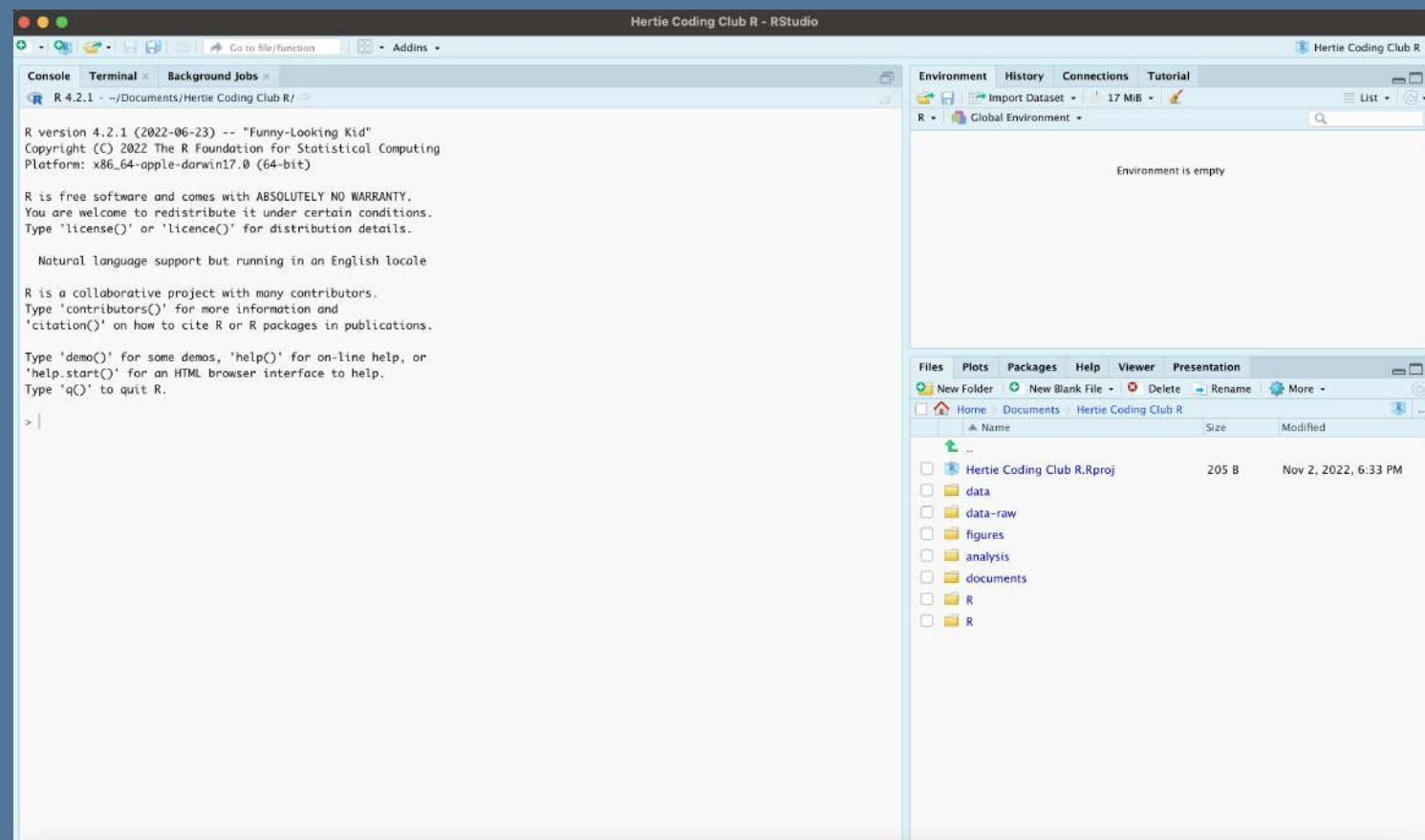
Customize our RStudio appearance
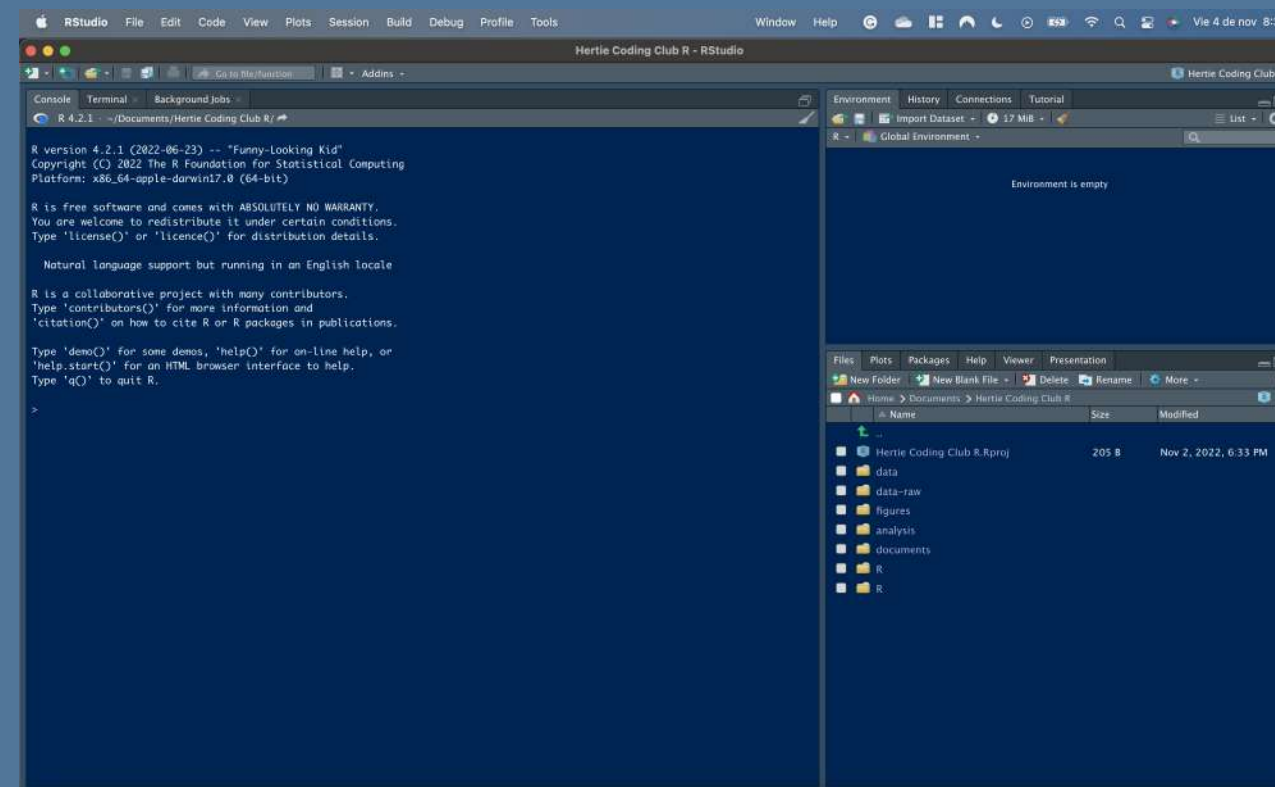
- Blue is my favorite color. For me looks awesome! What about you?

Hertie Coding Club

Create our first project

{ } Hertie Coding Club

Create our first project (Step 1)
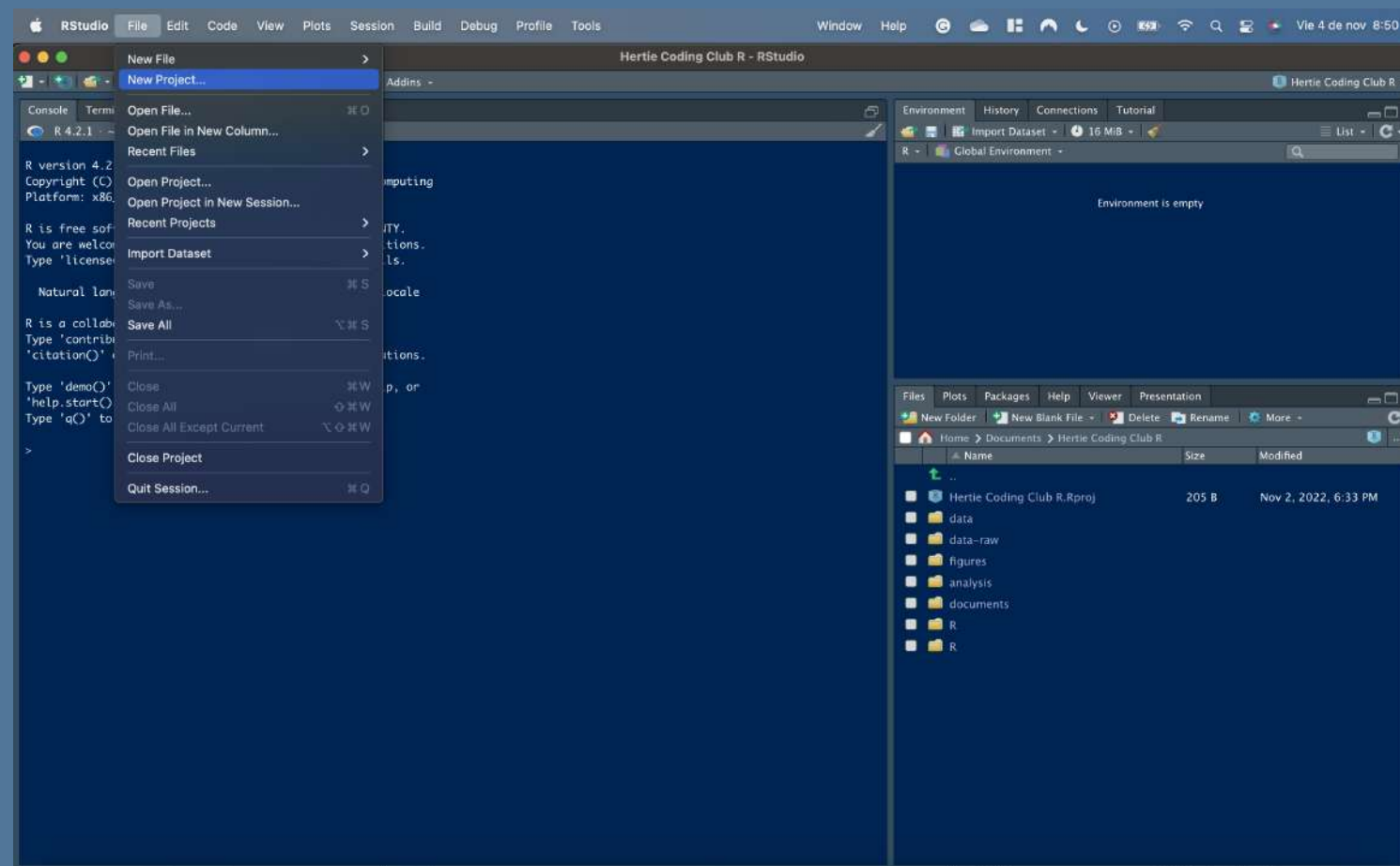
Go to the next steps:

- Create a new directory (Where we will store our project)

- Select "New Project"

Hertie Coding Club
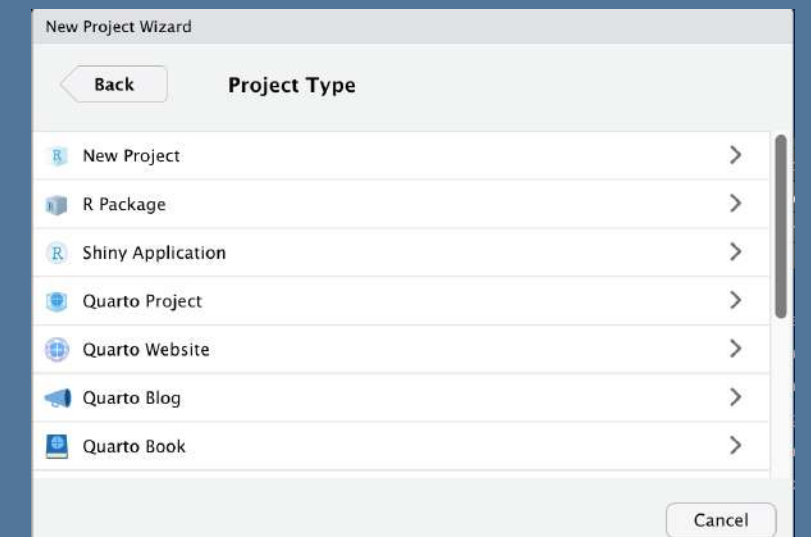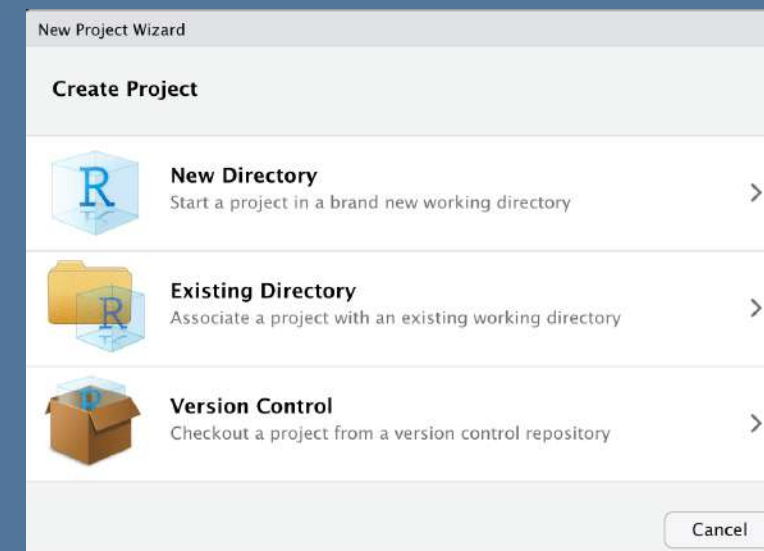
Create our first project (Step 2)
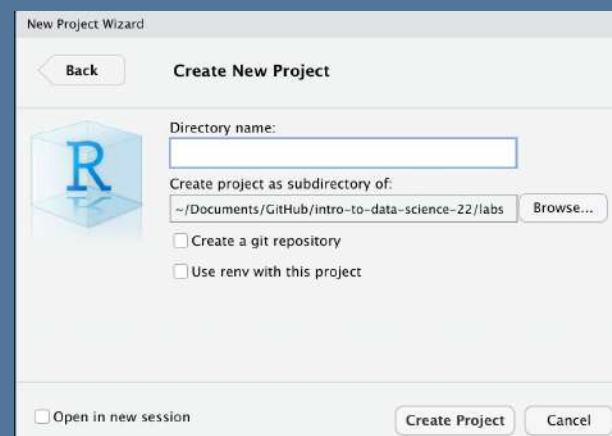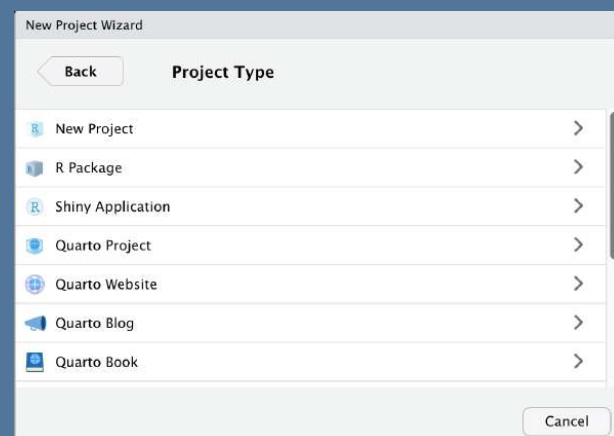
Go to the next steps:

- Select again New Project

- Choose the path where you want to create your project (It should be a folder)

Hertie Coding Club

Working directory (How we load and save things in R)

Go to the next steps:

- data/: for processed data
- data-raw/: for raw data and intermediate datasets (Internet, f.e.)
- figures/: for save your plots
- analysis/: for save your scripts
- documents/: for outlines, drafts, other text
- R/: for functions

Alarid et. al. (2019)

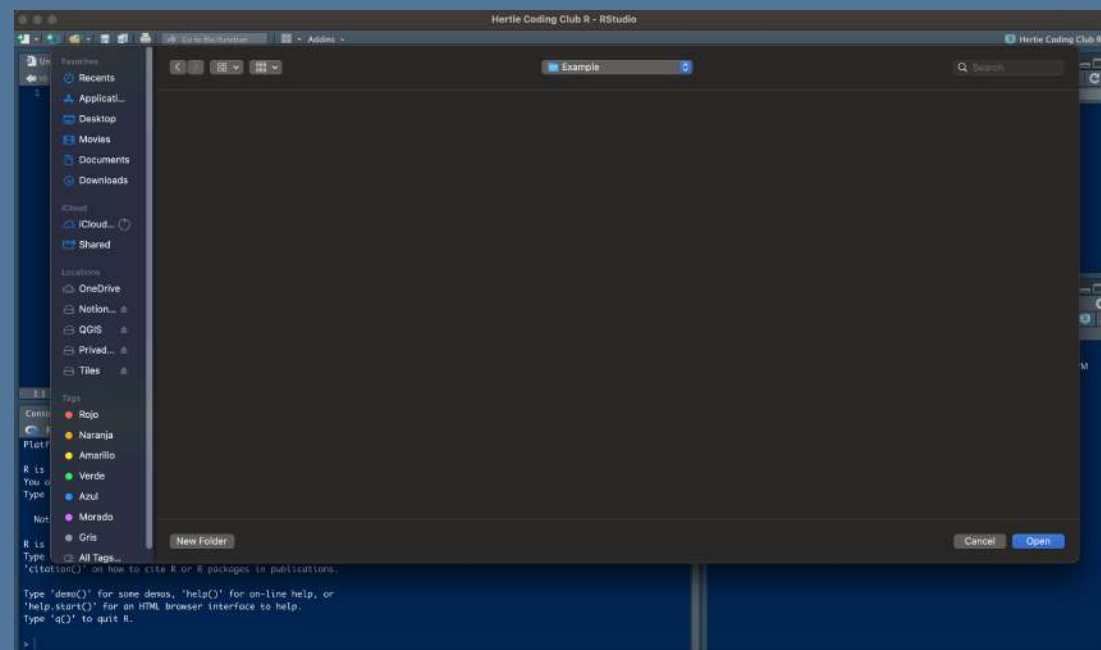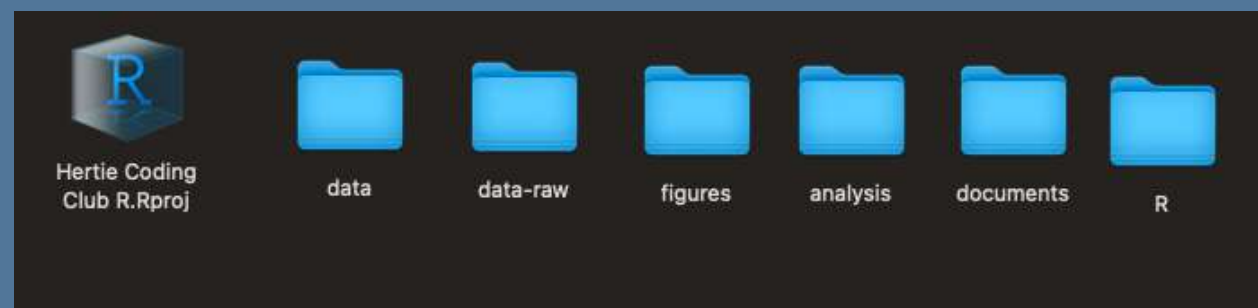| Folder name	Folder function
data-raw	This is where raw data is stored alongside ".R" scripts that read in raw data, process these data, and calls use_this::use_data(<processed data>) to save .rda formatted data files in the "data" folder. These data could include a ".csv" file with input parameters derived from the published literature, as well as internal R data files (with .RData, .rds, or .rda extensions) containing primary data from which model input values will be estimated through statistical models embedded into the analysis.
data	This is where input data is stored to be used in the different components of the CEA. These data could be generated from raw data stored in the "data-raw" folder. Essentially, this folder stores the cleaned or processed versions of raw data that has been gathered from elsewhere
R	This is where ".R" files that define functions to be used as part of the analysis are stored. These are functions that are specific to the analysis. The model will be one such function; however, other functions will likely be used, such as computing the fit of the model output to the specific calibration targets of the analysis. This folder also stores ".R" scripts that document the datasets in the "data" folder.
analysis	This is where interactive scripts of the analysis would be stored. These scripts control the overall flow of the analysis. This is also where many operations that ultimately become functions will be developed and debugged.
output	This is where output files of the analysis should be stored. These files may be internal R data files (".RData", ".rds", ".rda") or external data files (such as ".csv"). Examples of files stored here would be the output of the model calibration component or the PSA dataset generated in the uncertainty analysis component. These data files can then be loaded by other components without having to first rerun previous components (e.g. the calibrated model values can be loaded for a base case analysis without re-running the calibration).
figs	For analyses that will include figures, we generally create a separate figures folder. Though these could be stored in the output folder, it can be helpful to have a separate folder so that the images of the figure files can be easily previewed. This is particularly important for analyses that generate a large number of figures.
tables	This folder includes tables to be included in a publication or report, such as the table of intervention costs and effects and ICERs.
report	A report folder could be used to store R Markdown files to describe in detail the model-based CEA by using all the functions and data of the framework, run analyses and display figures. The R Markdown files can be compiled into .html, .doc or .pdf files to generate a report of the CEA. This report could be the document submitted to HTA agencies accompanying the R code of the model-based CEA.
vignettes	A vignettes folder could be used to describe the usage of the functions and data of each of some or all components of the framework through accompanying R Markdown files as documentation. The R Markdown file can use all the functions, outputs, and figures to integrate the R code into the Markdown text.
tests	A tests folder includes ".R" scripts that runs all the unit tests of the functions in the framework. A good practice is to have one file of tests for each complicated function or for each of the components of the framework.

Hertie Coding Club

YOU ARE READY

{●} Hertie Coding Club

RStudio

Create a script (This is the place where we write code)
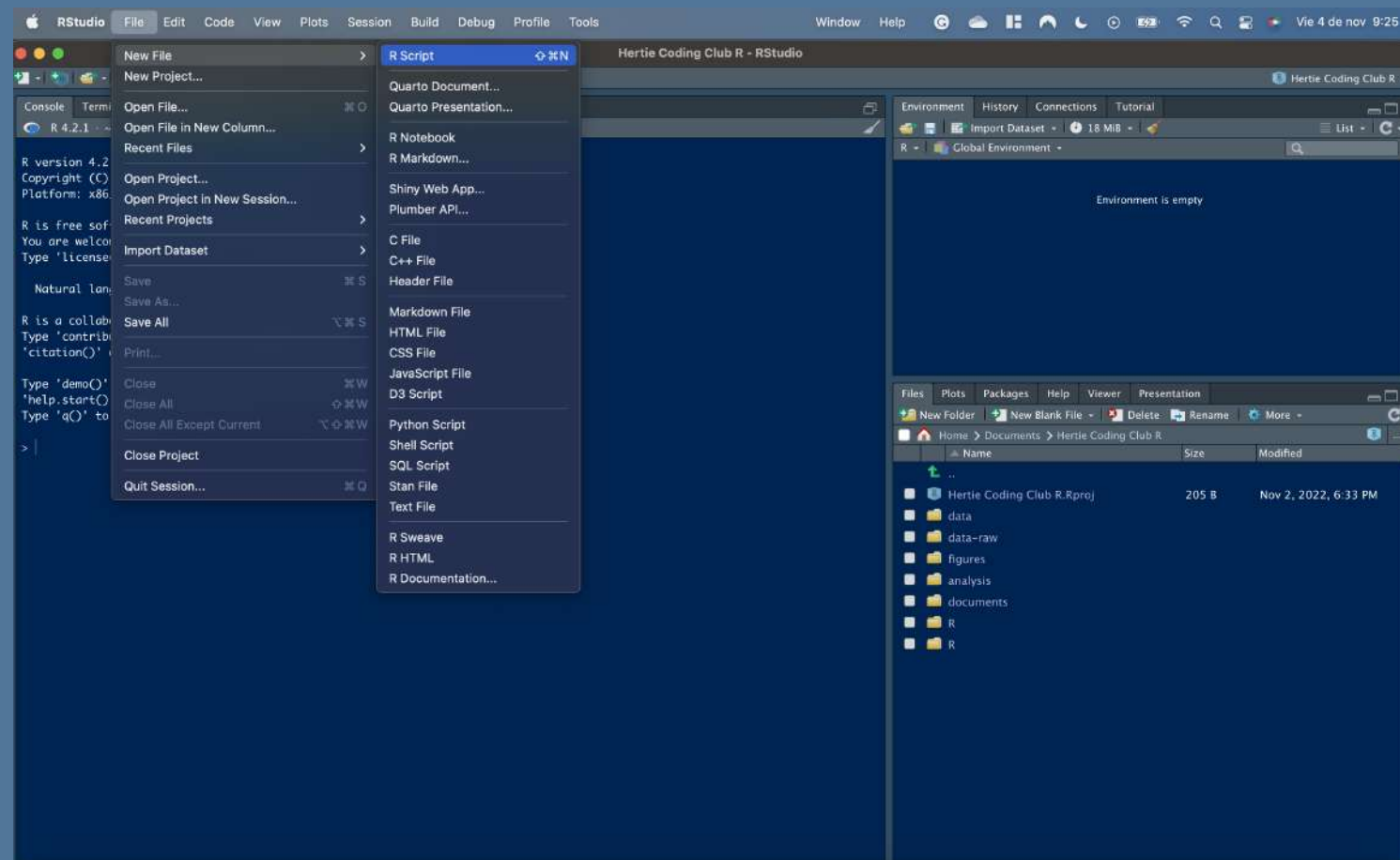
Go to the next steps:

- In the file section, create an R script

- Save inside your analysis folder

Hertie Coding Club
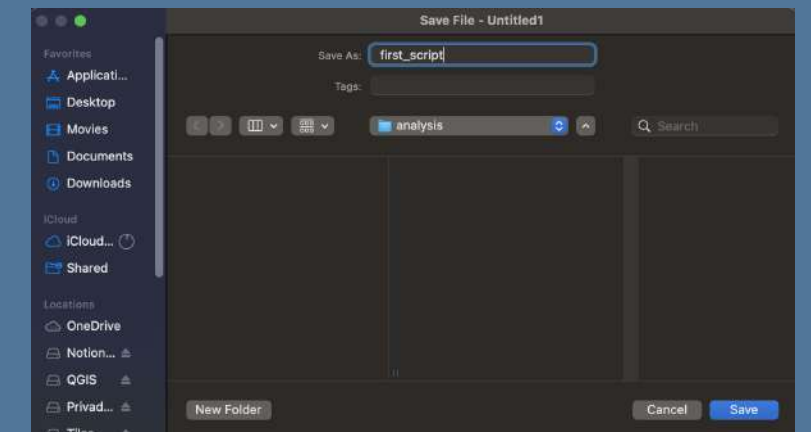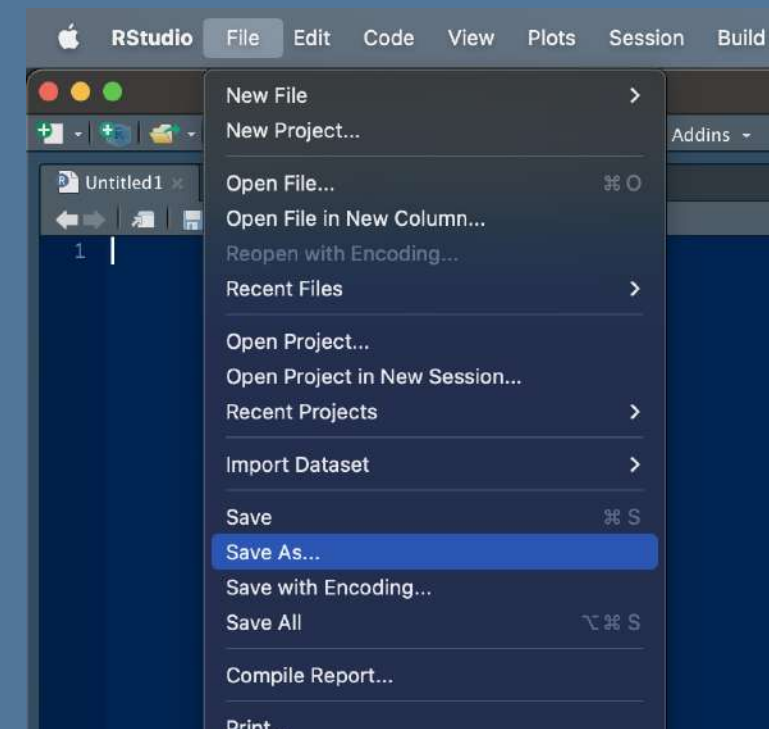
Source

The source pane is where you create and edit R Scripts

- "Command + Return" on Mac, or "Control + Enter" on PC to send all highlighted code to the console.

- Save it inside your analysis folder

Hertie Coding Club
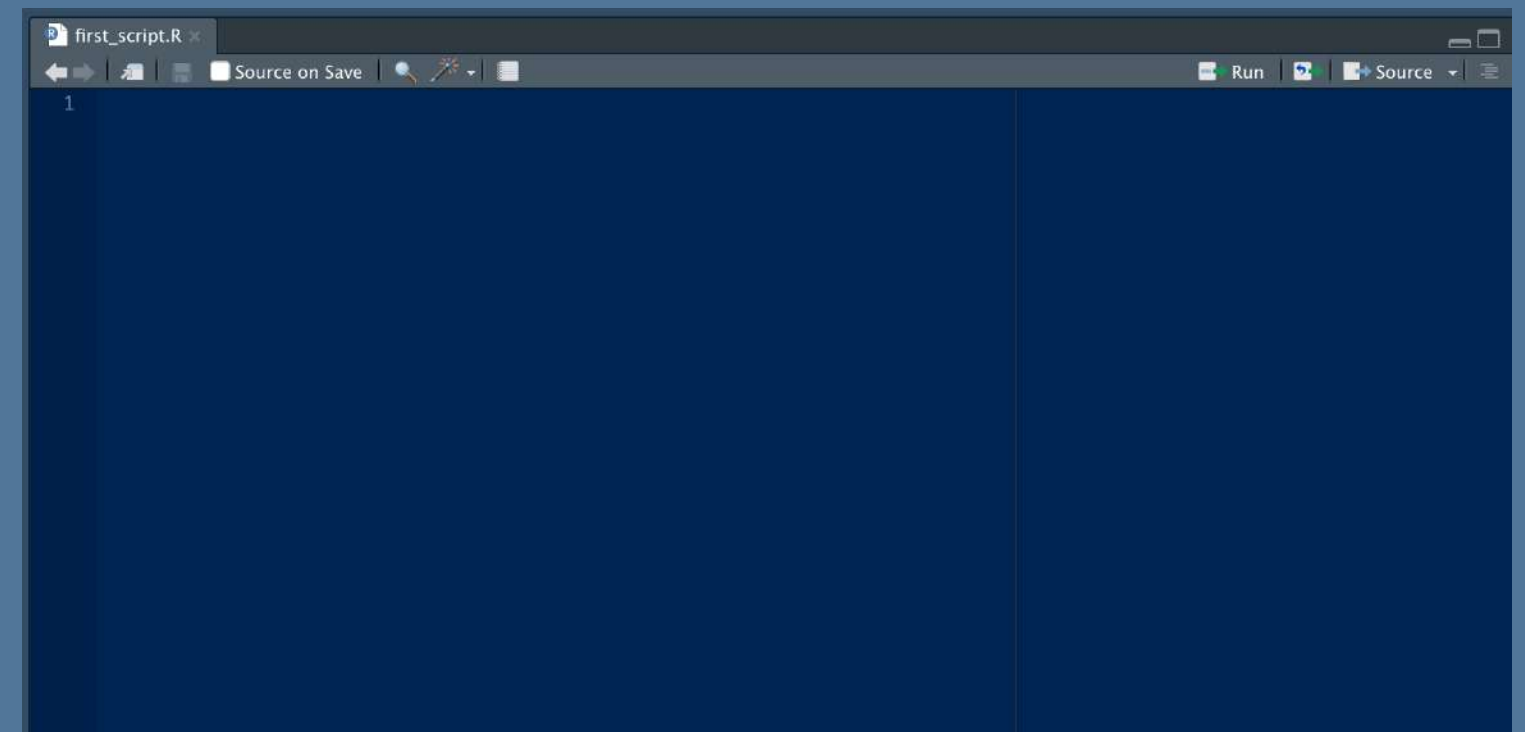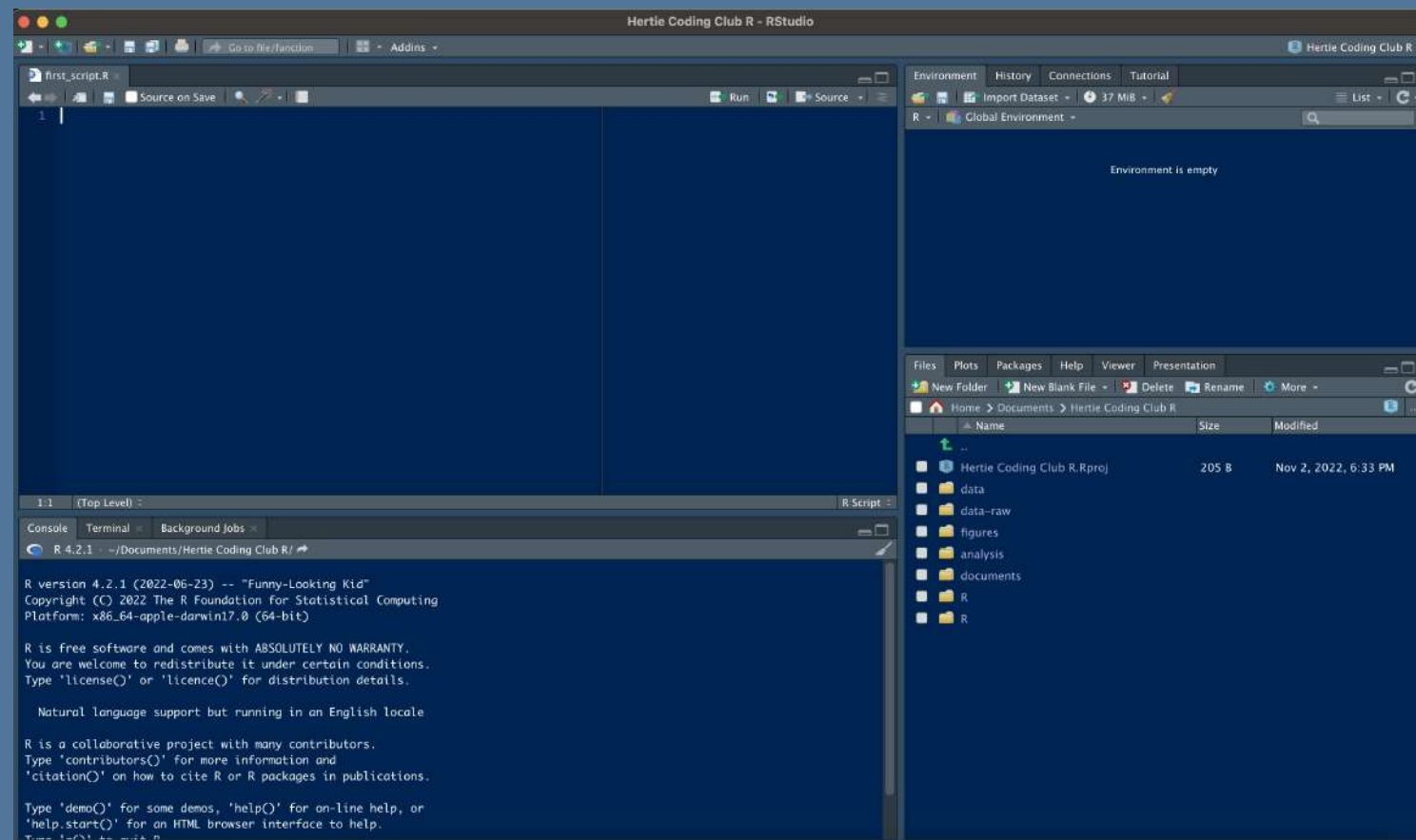
Console

The console is the heart of R.
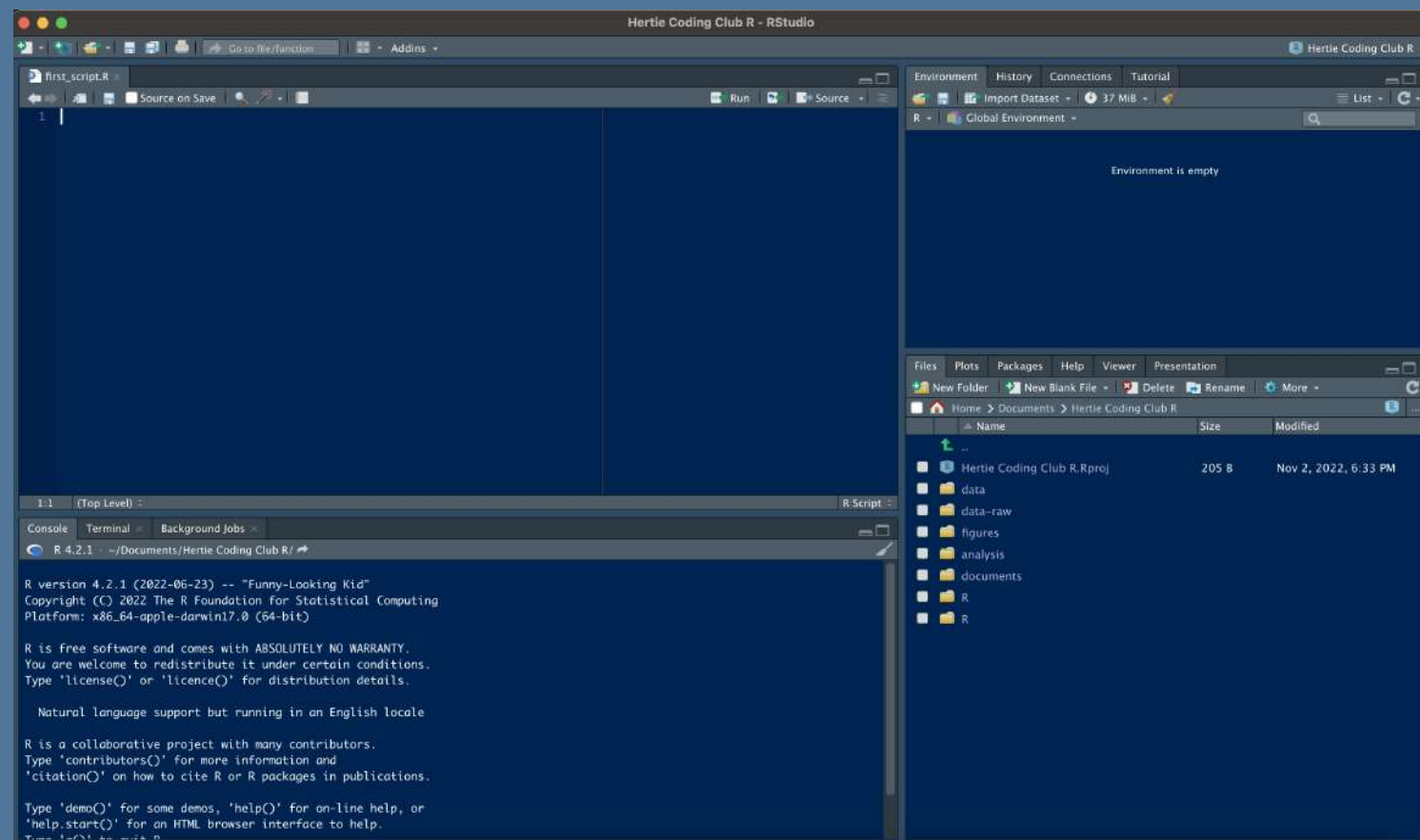
- Here R actually evaluates your code

- Try to write most of your code in a document in the Source. Only type directly into the Console to de-bug or do quick analyses.

- When ready: > and If waiting: +

- Cancel commands by pressing Esc.

Hertie Coding Club

Environment/History

The Environment tab of this panel shows you the names of all the data objects

- You can also see information like the number of observations and rows in data objects

Files and more

The Files / Plots / Packages / Help panel shows you lots of helpful information

- Files - The files panel gives you access to the file directory on your hard drive.

- Plots - The Plots panel shows all your plots.

{ } Hertie Coding Club

Packages

Hertie Coding Club
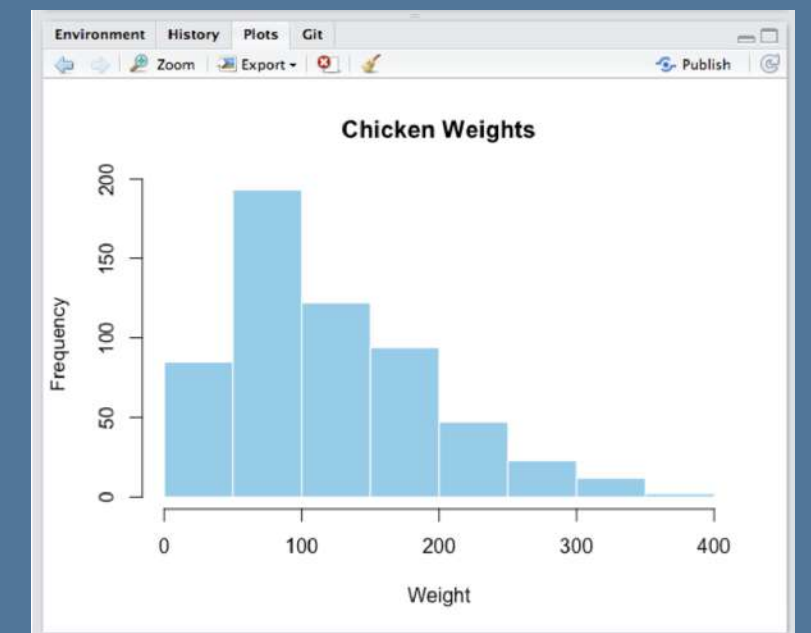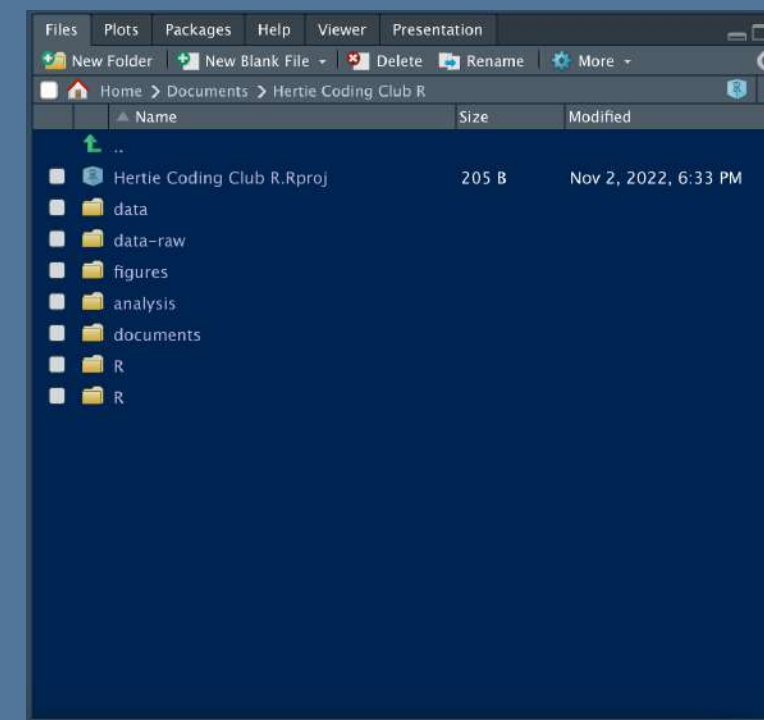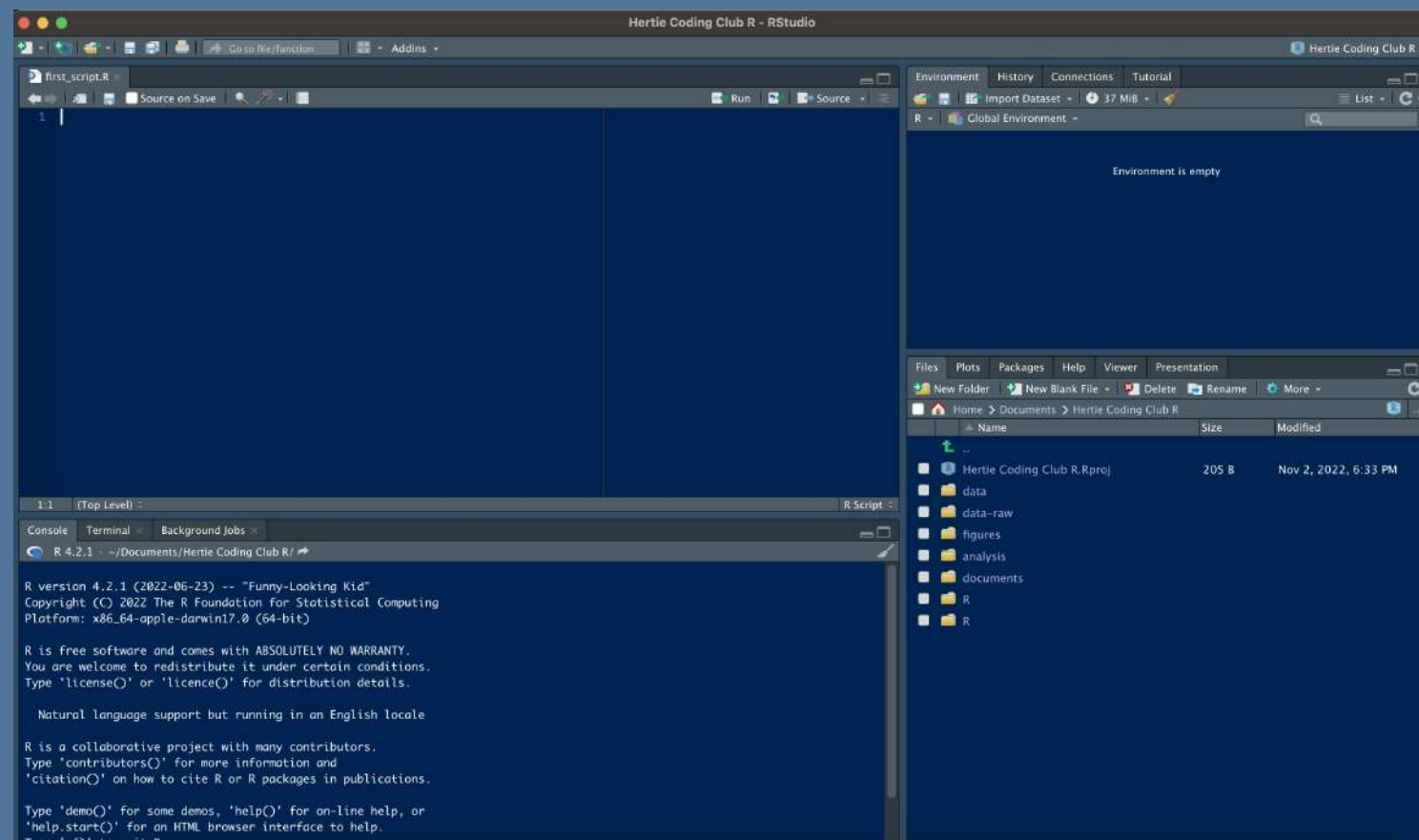
Packages

Why R?

- Packages in R Programming language are a set of R functions, compiled code, and sample data. These are stored under a directory called "library" within the R environment.

- R packages provide a simple way to distribute R code and documentation.

- Packages are the fundamental units of reproducible R code. They include reusable R functions, the documentation that describes how to use them, and sample data

- More than 18,000 packages

- Extends the capacity of R and allows us to do much more

HertieCoding Club

How we install packages
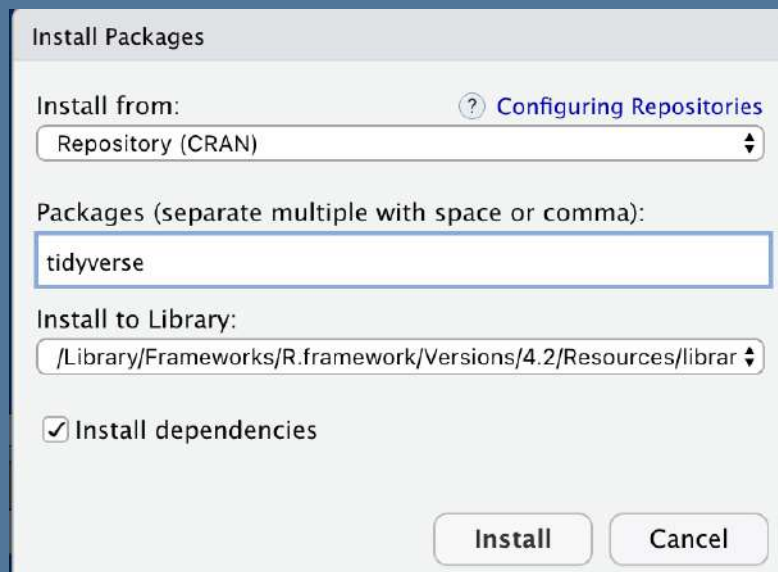
The easy way

- Go to the "Packages" tab

- Press the "Install" button
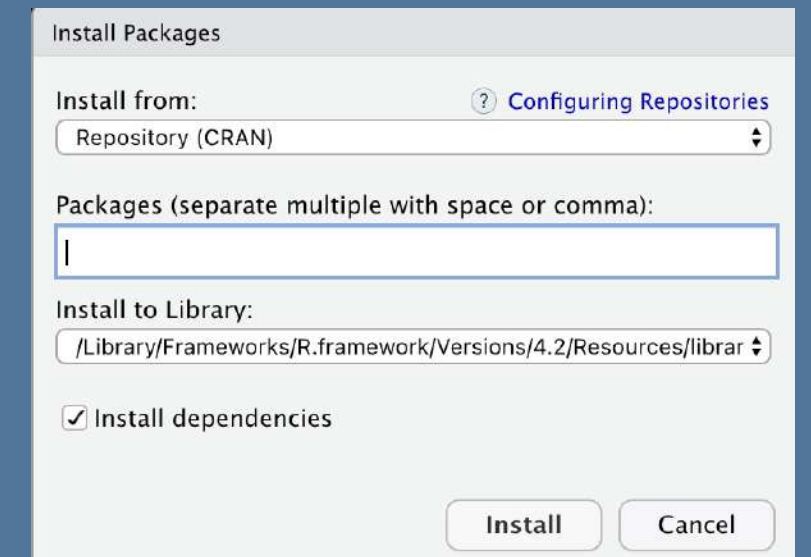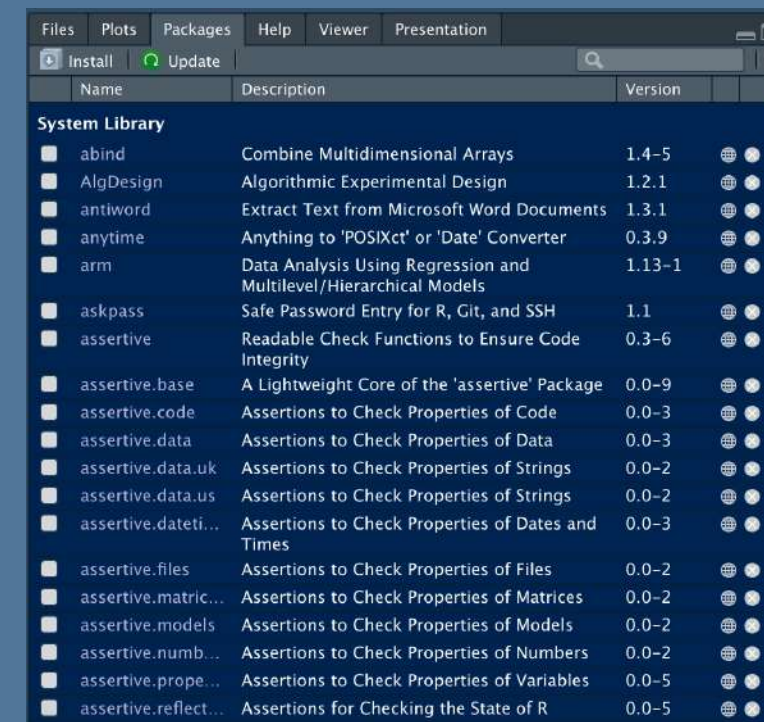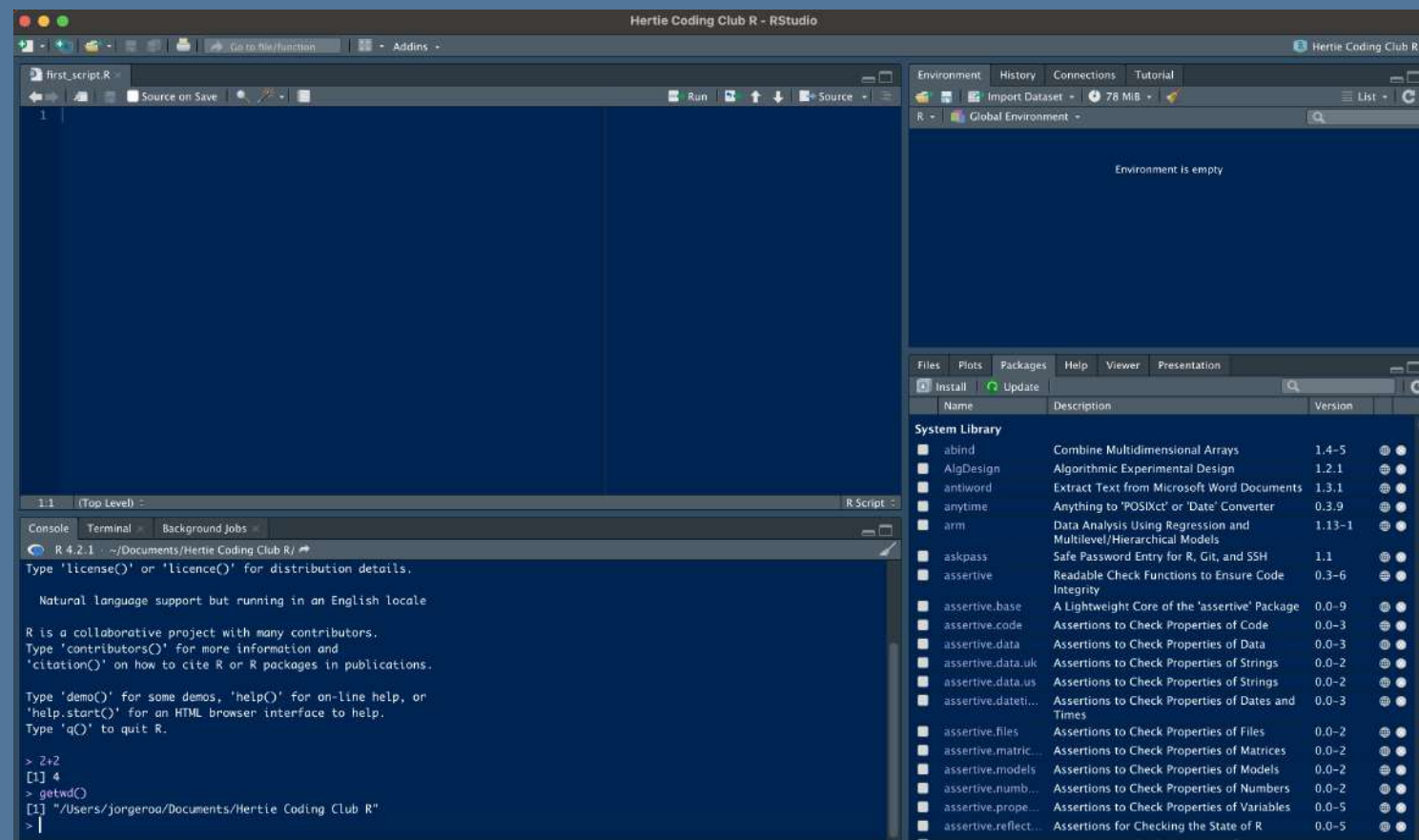
Hertie Coding Club

How we install packages

The easy way

- Other way is type in the console
- install.packages("tidyverse")

{ } Hertie Coding Club

Objects

Hertie Coding Club

Objects

Why objects?

- R works with objects (vectors, dataframes, lists, numbers, etc)

- To do useful and interesting things in R, we need to assign values to objects.

- To create an object, give it a name followed by the assignment operator, followed by the value.

- Assignment operator <-

- Can also use = but not recommended

- Shortcut: "Alt + -" on PC, "Option + -" on Mac

```
1  x <- 2 + 2
2
3  x
```
```
[1] 4
```

{} Hertie Coding Club

Type of objects

There are 5 basic types of objects in the R language:

- **Atomic vectors** are one of the basic types of objects in R programming. Atomic vectors can store homogeneous data types such as character, doubles, integers, raw, logical, and complex.

- **List** is another type of object in R programming. List can contain heterogeneous data types such as vectors or another lists.

```r
1  #Numeric vector
2  numbers <- c(1, 2, 3, 4)
3
4  #String vector
5  characters <- c("a", "b", "c", "d")
6
7  #Numeric value
8  value <- 5
9
10 #List
11 my_list <- list(c(1, 2, 3, 4), list("a", "b", "c"))
```

```r
1  print(numbers)
```
```
[1] 1 2 3 4
```
```r
1  print(characters)
```
```
[1] "a" "b" "c" "d"
```
```r
1  print(value)
```
```
[1] 5
```
```r
1  print(my_list)
```
```
[[1]]
[1] 1 2 3 4

[[2]]
[[2]][[1]]
[1] "a"

[[2]][[2]]
[1] "b"

[[2]][[3]]
[1] "c"
```

Hertie Coding Club

More type of objects

- **Matrices**: To store values as 2-Dimensional array, matrices are used in R. Data, number of rows and columns are defined in the matrix() function.

- **Factors**: Factor object encodes a vector of unique elements (levels) from the given data vector.

- **Arrays**: array() function is used to create n-dimensional array. This function takes dim attribute as an argument and creates required length of each dimension as specified in the attribute.

```r
1   x <- c(1, 2, 3, 4, 5, 6)
2
3   # Matrix
4   mat <- matrix(x, nrow = 2)
5
6   # Matrix
7   s <- c("spring", "autumn", "winter", "summer",
8   "spring", "autumn")
9
10  arr <- array(c(1, 2), dim = c(3, 3))
```

```r
1   print(mat)
```

```
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```r
1   print(factor(s))
```

```
[1] spring autumn winter summer spring autumn
Levels: autumn spring summer winter
```

```r
1   print(nlevels(factor(s)))
```

```
[1] 4
```

```r
1   print(arr)
```

```
     [,1] [,2] [,3]
[1,]    1    2    1
[2,]    2    1    2
[3,]    1    2    1
```

{⟩ Hertie Coding Club

Finally: dataframes

- Data frames are 2-dimensional tabular data object in R programming.

- Data frames consists of multiple columns and each column represents a vector.

- Columns in data frame can have different modes of data unlike matrices.

```r
1  # Create vectors
2  who <- c("Mom", "Sister", "Myself", "Dad", "Brother", "Brother", "Our dog (:")
3  age <- c(58, 17, 25,60, 29, 27, 5)
4  names <- c("Carmen", "Fernanda", "Jorge", "Arturo", "Ale", "Eduardo", "Rocky")
5
6  # Create data frame of vectors
7  df_my_family <- data.frame(who, age, names)
```

```r
1  print(df_my_family)
```

```
         who  age     names
1        Mom   58    Carmen
2     Sister   17  Fernanda
3     Myself   25     Jorge
4        Dad   60    Arturo
5    Brother   29       Ale
6    Brother   27   Eduardo
7 Our dog (:    5     Rocky
```

Hertie Coding Club

Now it's your turn

Hertie Coding Club

Exercise

- Create a dataframe of your family with multiple vectors.

- Include age, position (mom, dad, etc), their names and their birthdays.

- Hint:

```
1  # Create vectors
2  who <- c()
3  age <- c()
4  names <- c()
5  birthday <- "?"
6
7  # Create data frame of vectors
8  df_my_family <- data.frame("Here you put your vectors")
```

REMEMBER: They must have the same length.

LET'S GET READY TO

{☰} Hertie Coding Club

Summary

- Customize our RStudio appearance

- Create our first project

- RStudio elements

- Working directory

- Packages

- Objects

- Summary

- Next session

{ } Hertie Coding Club

Next session

- Base R

- Load and save objects

- Pipelines

- Tidyverse

- And more...

- Same channel (Forum)

- Same day and hour Friday, November 11 (14:00 hrs - 16:00 hrs)

- Join Slack if you haven't.

- Use Slack for questions.

- Follow us in our social media:

- 📷 Instagram: hertiecodingclub

- 🔮 Twitter: @HertieCodingCLB

{ } Hertie Coding Club

Thanks for your time and welcome

Remember that everybody can learn how to code!!

1200 lines of code where created for this presentation.