



# Week 11

# R

# Day 1

INTRO TO R

# ¿Qué es R?

R es un lenguaje de programación funcional enfocado en la estadística.  
R es software libre descargable desde CRAN.

**The Comprehensive R Archive Network**

CRAN es el "gran almacén" de R donde están guardados todos los paquetes (y sus versiones) desarrollados en R con sus documentaciones.

El método más común de utilizar R es a través del entorno de desarrollo RStudio.

# Un poco de historia



---

1991

## **En Nueva Zelanda**

Ross Ihaka y Robert Gentleman

---

1993

## **Primer anuncio sobre R**

---

2000

## **Se lanza la version 1.0.0 de R**

---

1976

## **R es la implementación moderna de S**

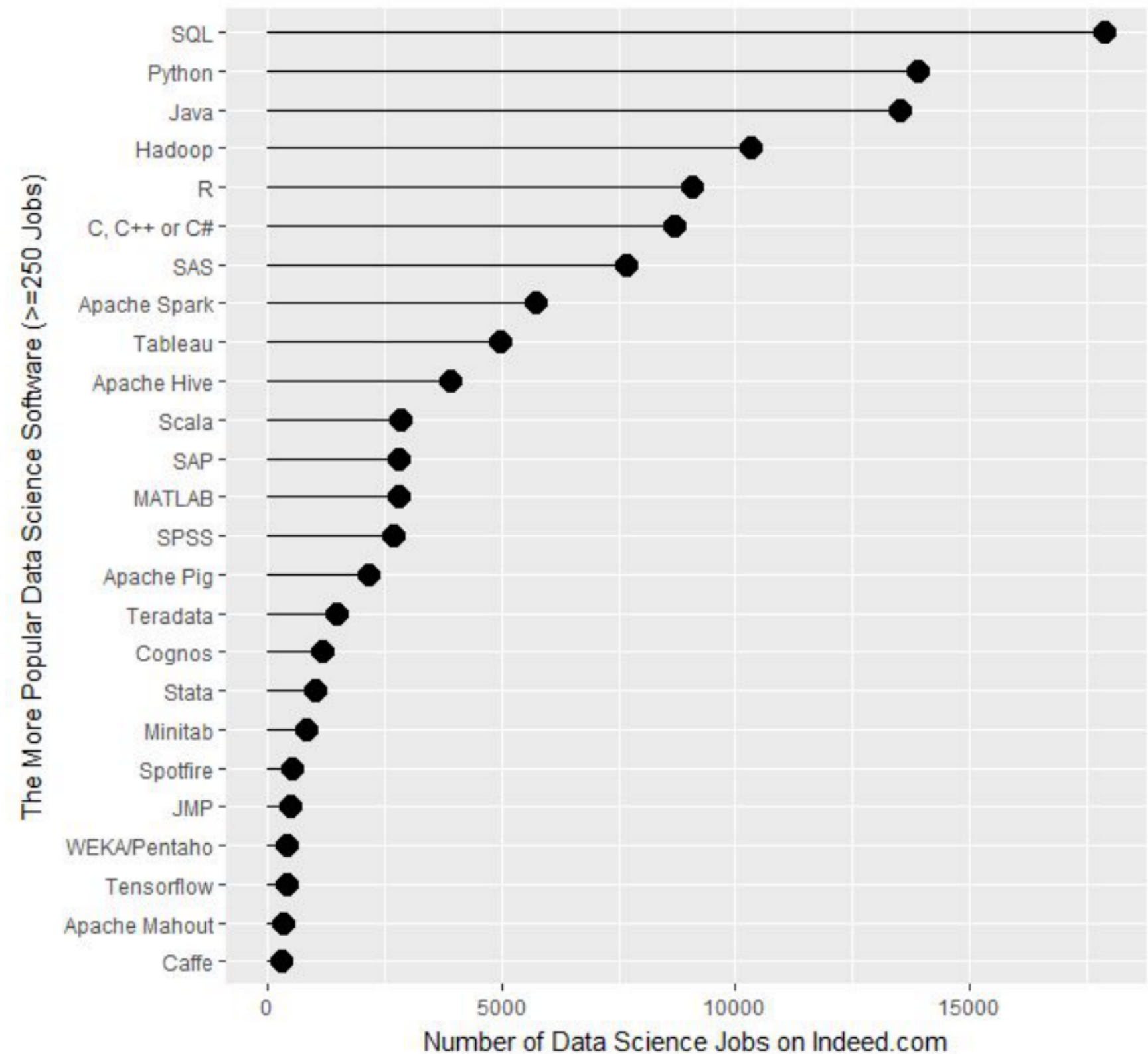
S es uno de los varios lenguajes de computación estadística.

# ¿Por qué R?

01.

## Su popularidad

R se está convirtiendo rápidamente en el lenguaje utilizado para los análisis estadísticos. Esto hace de R un lenguaje que aprender ya que cuanto más popular se hace un lenguaje, más rápido se desarrollan nuevas funcionalidades, más poderoso se vuelve y mejor soporte encontramos. Además, como se puede ver en el gráfico, R es uno de los cinco lenguajes más pedidos en las ofertas de trabajo de data scientist.



<http://r4stats.com/articles/popularity/>

<https://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future>

# ¿Por qué R?

02.

## Su precio

Es gratis, R, RStudio y todas las librerías.

Cuando hablamos de Software libre:

- Libre de usar el programa.
- Libre de estudiar como funciona el programa y adaptarlo a tus necesidades.
- Libre de redistribuir copias, y así ayudes al vecino.
- Libre de mejorar el programa y transmitir estas mejoras al público.

03.

## Su amplia funcionalidad

R es un lenguaje muy versátil - una aplicación amplia en estadística y gráficos, pero su uso puede extenderse a otras muchas funcionalidades como hacer webs, hacer mapas con GIS data, analizar el lenguaje natural... Para cualquier cosa que tengas en mente hay una librería que hace exactamente eso.

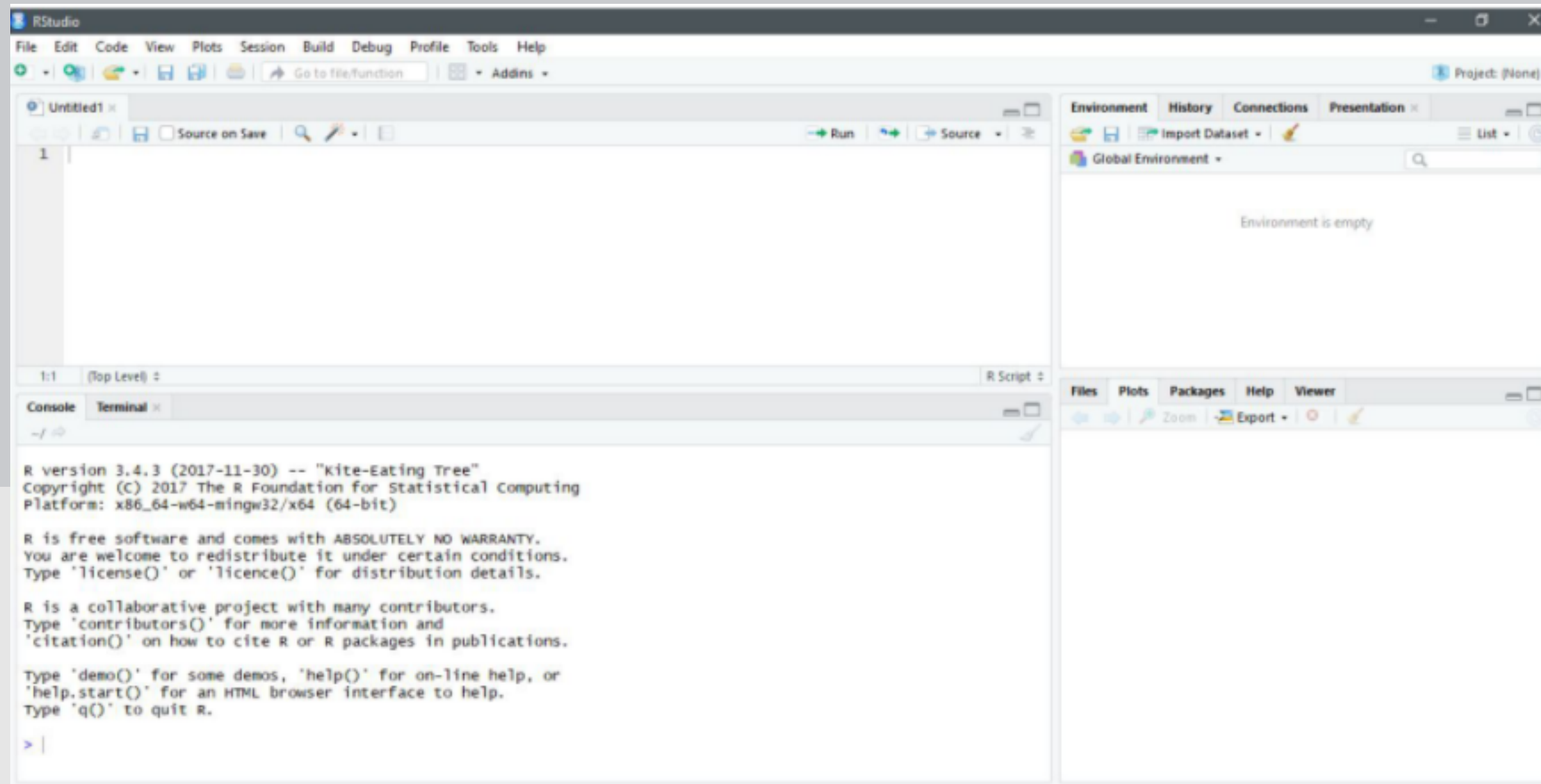
04.

## Su comunidad

Y la razón por la que su funcionalidad es tan extensa es por la comunidad que hay detrás de R. Muchas personas se han unido para crear librerías que sin duda añaden funcionalidades y valor a este lenguaje. ¡Y esto solo acaba de empezar!

# ¿Qué es RStudio?

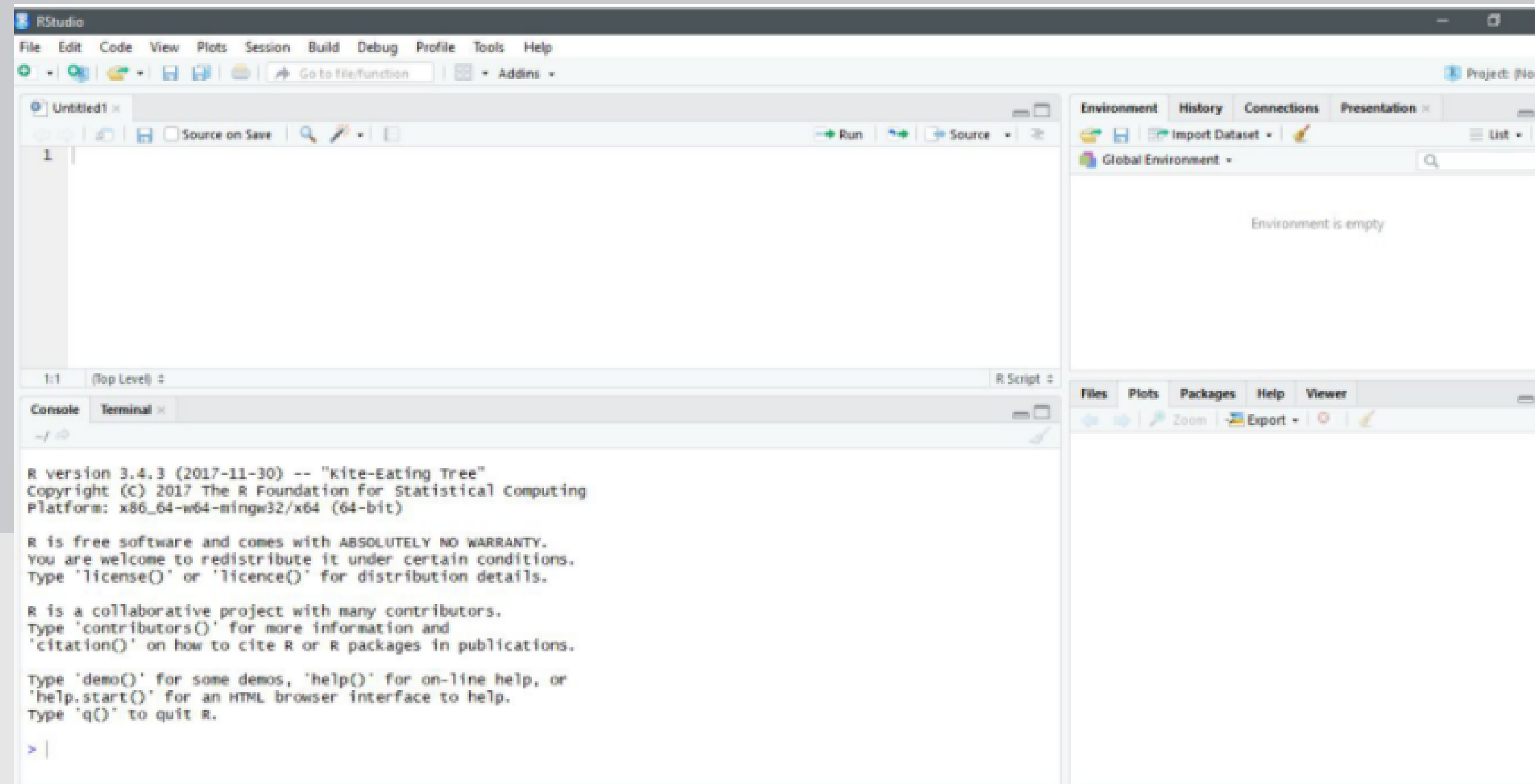
- RStudio es una interfaz gráfica de R, que permite entre otras cosas,
- escribir y editar código, generar, ver y guardar gráficas,
  - gestionar archivos, objetos y dataframes,
  - además de integrar sistemas de control de versiones. Para los amigos, Git.



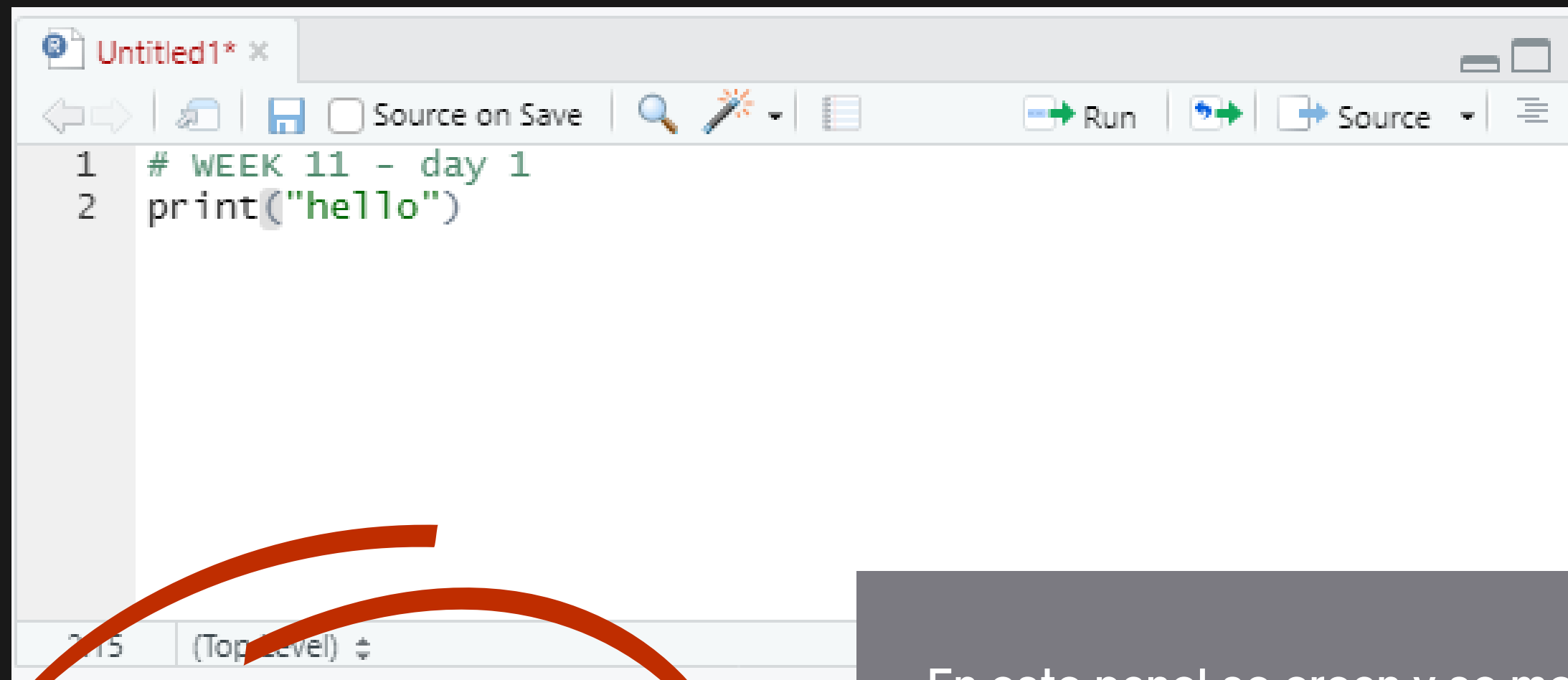
# Interfaz RStudio

Consta de 4 paneles:

- La consola
- El entorno de variables
- El editor
- Las utilidades



# El editor

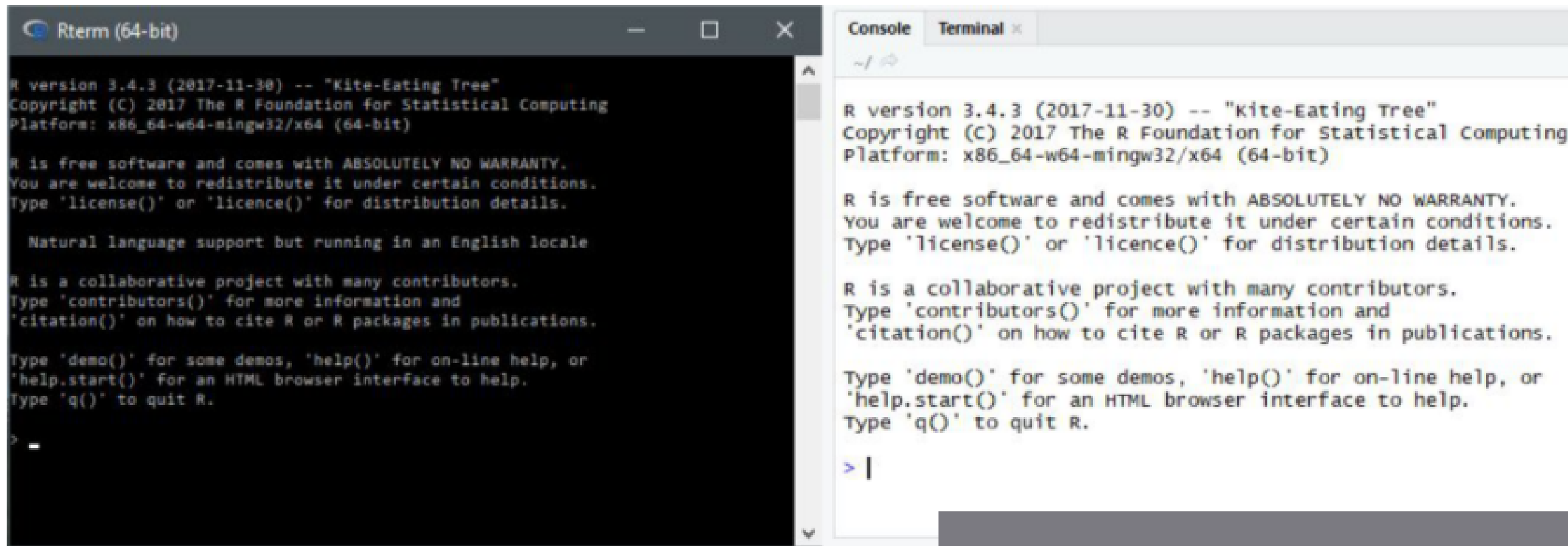


En este panel se crean y se modifican los scripts.

Para ejecutar un script deberás seleccionar lo que quieras ejecutar y o bien pulsar **Run**, o **Ctrl+Enter**



# La consola



```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 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.

  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 version 3.4.3 (2017-11-30) -- "kite-Eating Tree"
Copyright (C) 2017 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.

> |
```





Aquí es donde escribes y ejecutas comandos o donde obtendrás los outputs de los scripts escritos en el editor.

# El editor de variables

Este punto lo trataremos más adelante




EnvironmentHistoryConnectionsGitTutorial

  |  Import Dataset ▾ | 

Global Environment ▾

Data

 mtcars

32 obs. of 11 variables

Values

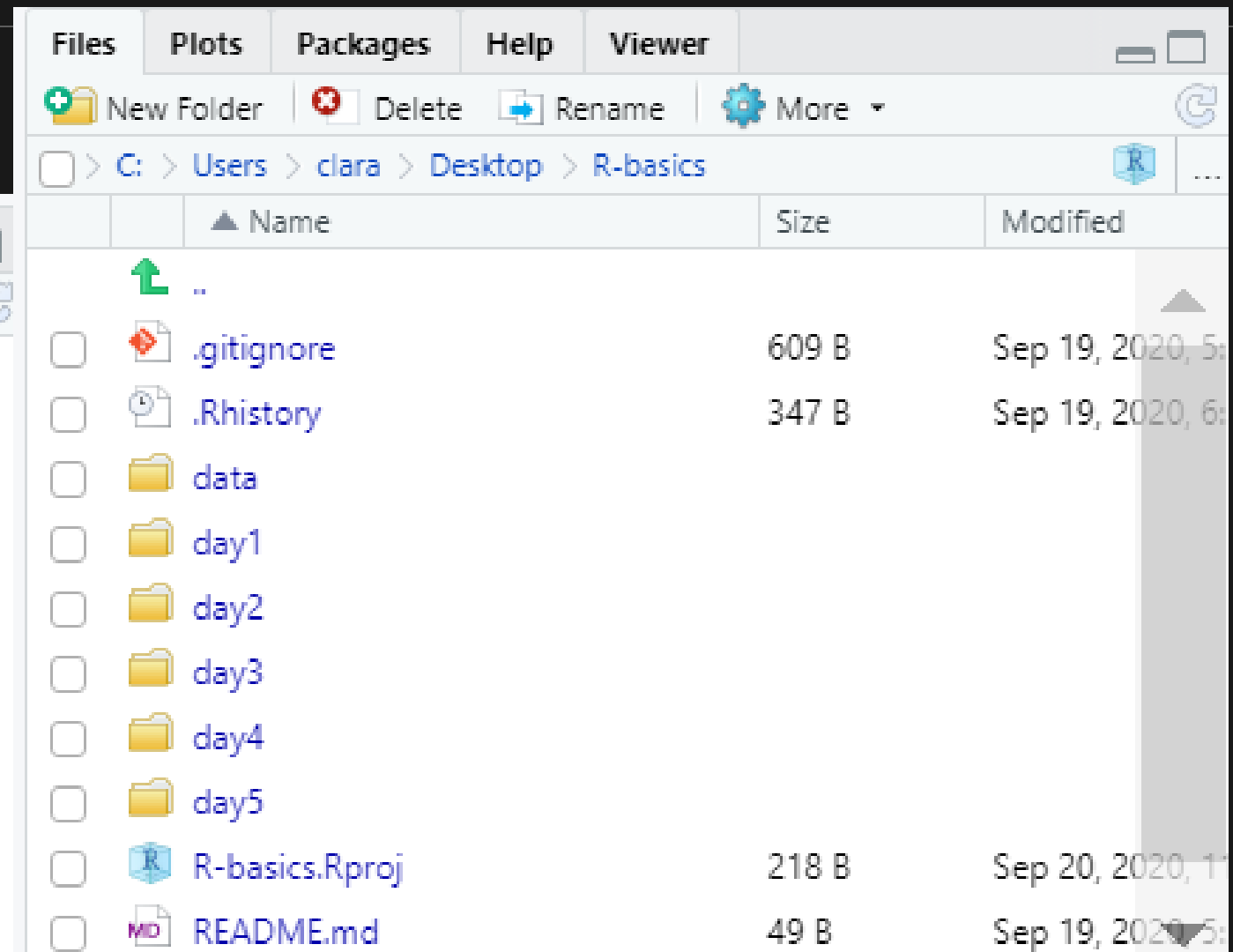
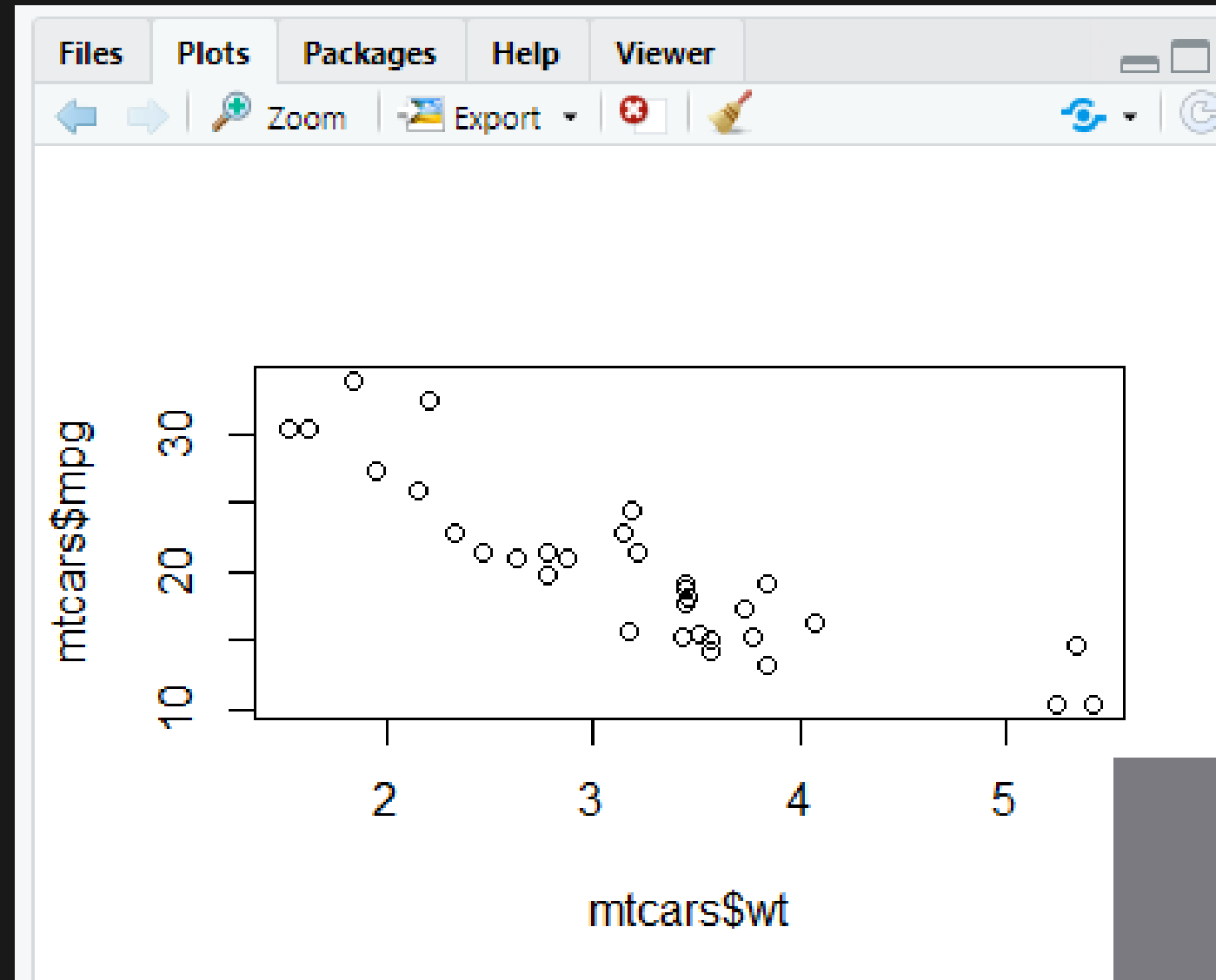
num\_alumnos

12

Si clicas encima de la flechita azul se mostrarán los detalles del dataframe, para ver el data.frame, pulsa sobre su nombre.

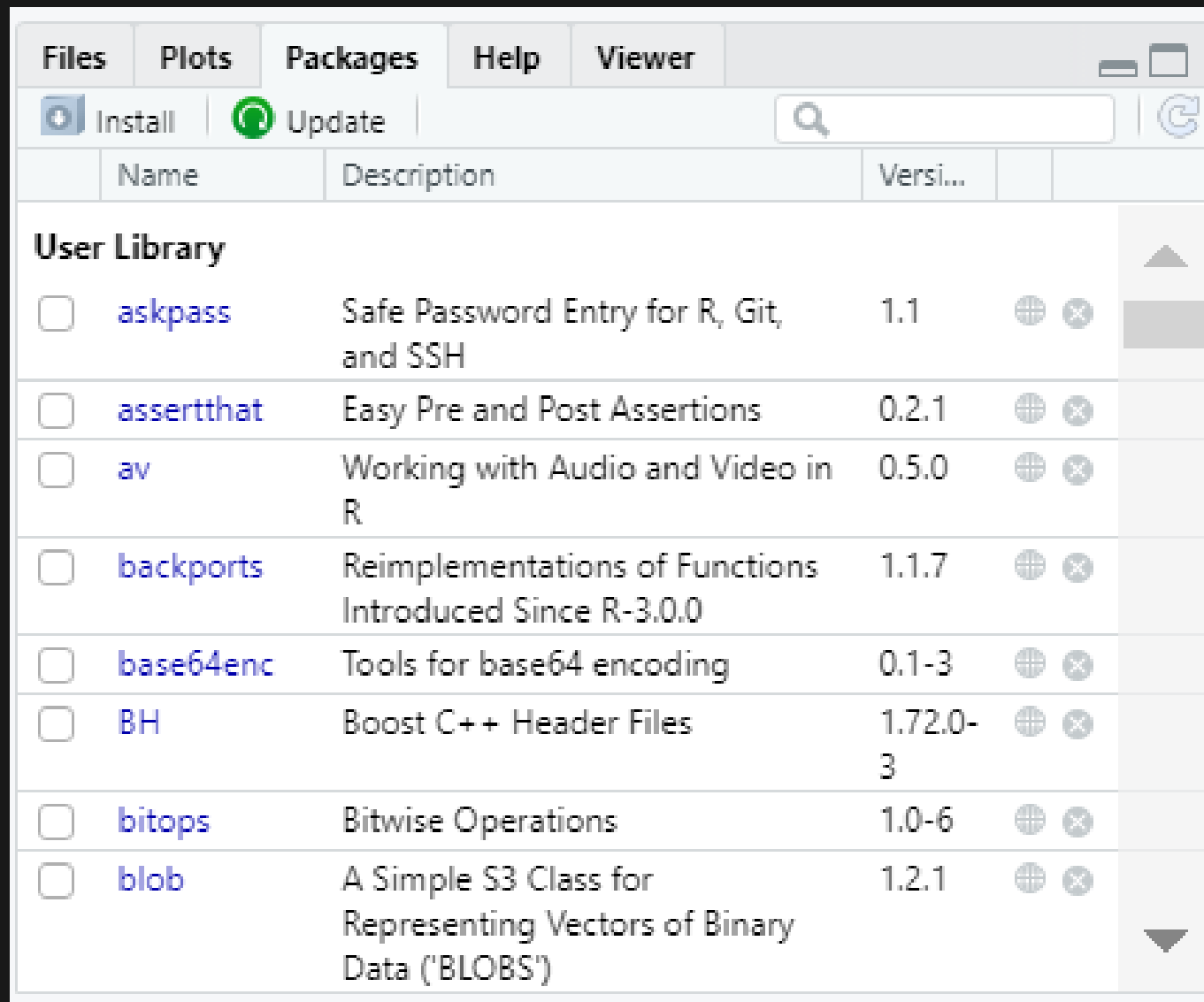
Aquí encontraremos aquellas variables a las que hayamos asignado valor y que se encuentren cargadas en el entorno

# Las utilidades



Aquí se muestran las diferentes gráficas que se hayan cargado en este archivo, además de mostrar nuestro entorno de trabajo

# Las utilidades



The screenshot shows the RStudio interface with the 'Packages' tab selected. It displays a list of packages in the 'User Library'. Each row includes a checkbox, the package name, a description, the version number, and icons for installation, update, and removal. The packages listed are askpass, assertthat, av, backports, base64enc, BH, bitops, and blob.

	Name	Description	Versi...		
<b>User Library</b>					
<input type="checkbox"/>	askpass	Safe Password Entry for R, Git, and SSH	1.1	⊕	⊗
<input type="checkbox"/>	assertthat	Easy Pre and Post Assertions	0.2.1	⊕	⊗
<input type="checkbox"/>	av	Working with Audio and Video in R	0.5.0	⊕	⊗
<input type="checkbox"/>	backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.7	⊕	⊗
<input type="checkbox"/>	base64enc	Tools for base64 encoding	0.1-3	⊕	⊗
<input type="checkbox"/>	BH	Boost C++ Header Files	1.72.0-3	⊕	⊗
<input type="checkbox"/>	bitops	Bitwise Operations	1.0-6	⊕	⊗
<input type="checkbox"/>	blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.1	⊕	⊗

Un paquete en R es un conjunto de funciones, datos y código. La localización donde los paquetes están contenidos es a lo que se le llama librerías. Si hay alguna funcionalidad concreta que necesites, deberás descargarte el paquete que se guardará en tu librería. Para poder utilizar el paquete tendrás que cargar dicho paquete desde tu librería usando el comando `library(package)`, lo que hará que sus funciones estén disponibles para su uso.

Para instalar paquetes nuevos, usa el comando `install.packages("package")`.

# Las utilidades

Files	Plots	Packages	Help	Viewer
Install	Update			
Name	Description	Versi...		
<b>User Library</b>				
<input type="checkbox"/> <a href="#">askpass</a>	Safe Password Entry for R, Git, and SSH	1.1	⊞ ⊗	
<input type="checkbox"/> <a href="#">assertthat</a>	Easy Pre and Post Assertions	0.2.1	⊞ ⊗	
<input type="checkbox"/> <a href="#">av</a>	Working with Audio and Video in R	0.5.0	⊞ ⊗	
<input type="checkbox"/> <a href="#">backports</a>	Reimplementations of Functions Introduced Since R-3.0.0	1.1.7	⊞ ⊗	
<input type="checkbox"/> <a href="#">base64enc</a>	Tools for base64 encoding	0.1-3	⊞ ⊗	
<input type="checkbox"/> <a href="#">BH</a>	Boost C++ Header Files	1.72.0-3	⊞ ⊗	
<input type="checkbox"/> <a href="#">bitops</a>	Bitwise Operations	1.0-6	⊞ ⊗	
<input type="checkbox"/> <a href="#">blob</a>	A Simple S3 Class for Representing Vectors of Binary Data ("BLOBS")	1.2.1	⊞ ⊗	

¿qué paquetes tengo instalados?

```
installed.packages()  
library()
```

actualizar los paquetes instalados

```
old.packages()  
update.packages()  
install.packages("package")
```

para desinstalar paquetes

```
remove.packages("package")
```

# Proyectos en RStudio

Al crear un proyecto en R, se creará una carpeta que se asignará como tu carpeta de trabajo **<working directory>**, de modo que todos los archivos generados se guardarán en esta carpeta.

Esto puede ser de mucha utilidad, ya que al abrir el proyecto, RStudio recordará todos los archivos que estaban abiertos y reestablecerá todo el entorno de trabajo **<working enviroment>** tal y como estaba cuando lo cerraste la última vez. Esto incluye variables o librerías cargadas.

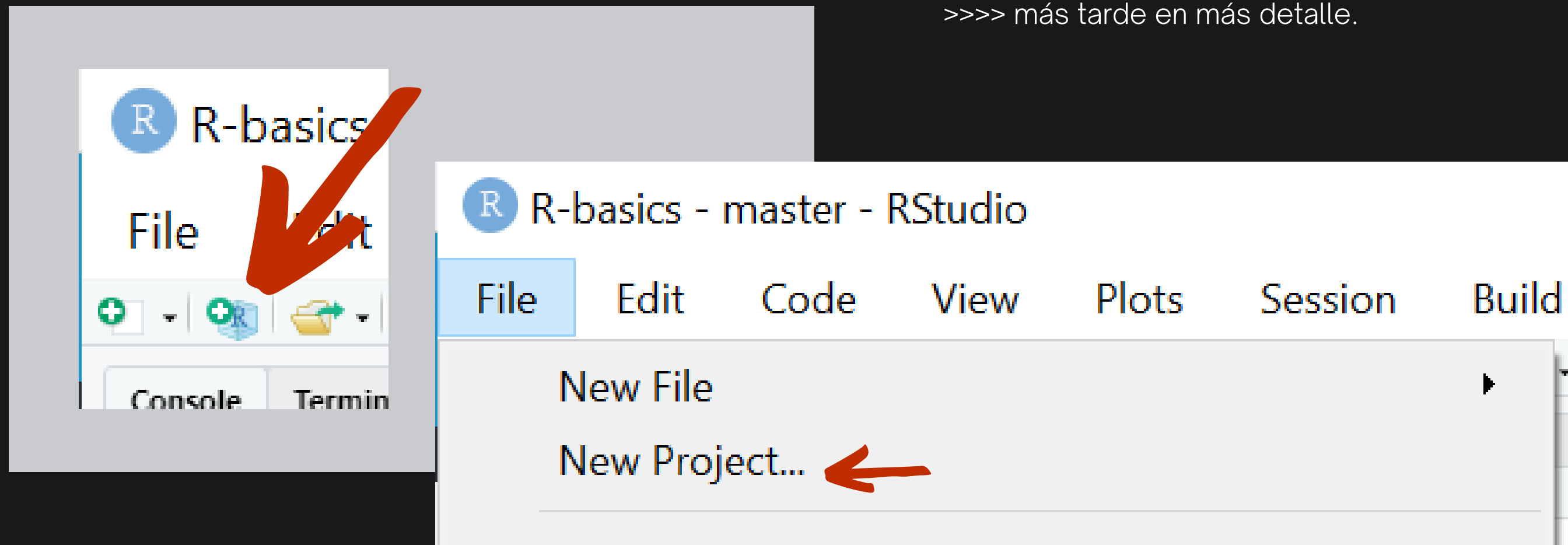


<https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects>

# Crear un proyecto

## Desde cero

1. Abre RStudio
2. En File, selecciona "New Project"



## Vinculándolo a Github

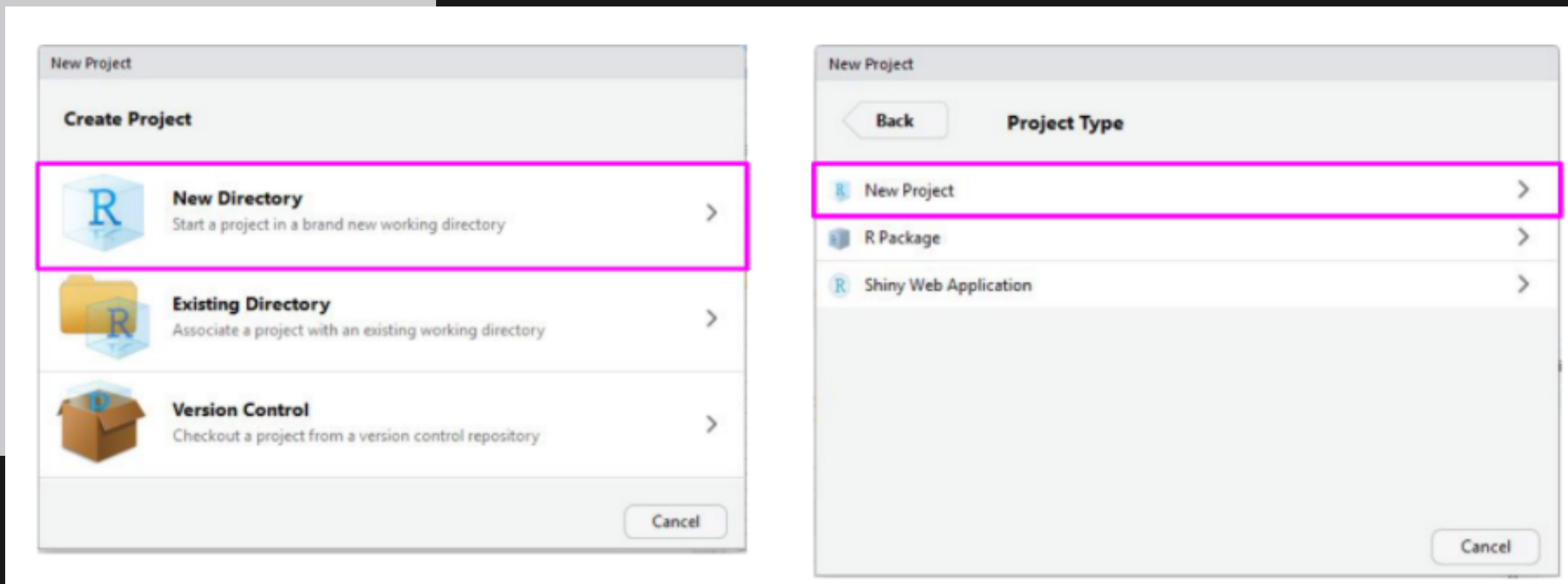
Esto "clonará" el repositorio y creará un proyecto de R.

>>>> más tarde en más detalle.

# Crear un proyecto

## Desde cero

- 3. New Directory
- 4. New Project






# Crear un proyecto

## Desde cero

5. Elije un nombre para tu proyecto y guárdalo en la carpeta que elijas. Esto creará una carpeta en la localización elegida, donde se guardarán todos los archivos vinculados a este proyecto.



New Project

[Back](#) **Create New Project**

Directory name:

Create project as subdirectory of:  [Browse...](#)

☐ Create a git repository

☐ Open in new session

**Create Project** [Cancel](#)

# Crear un proyecto

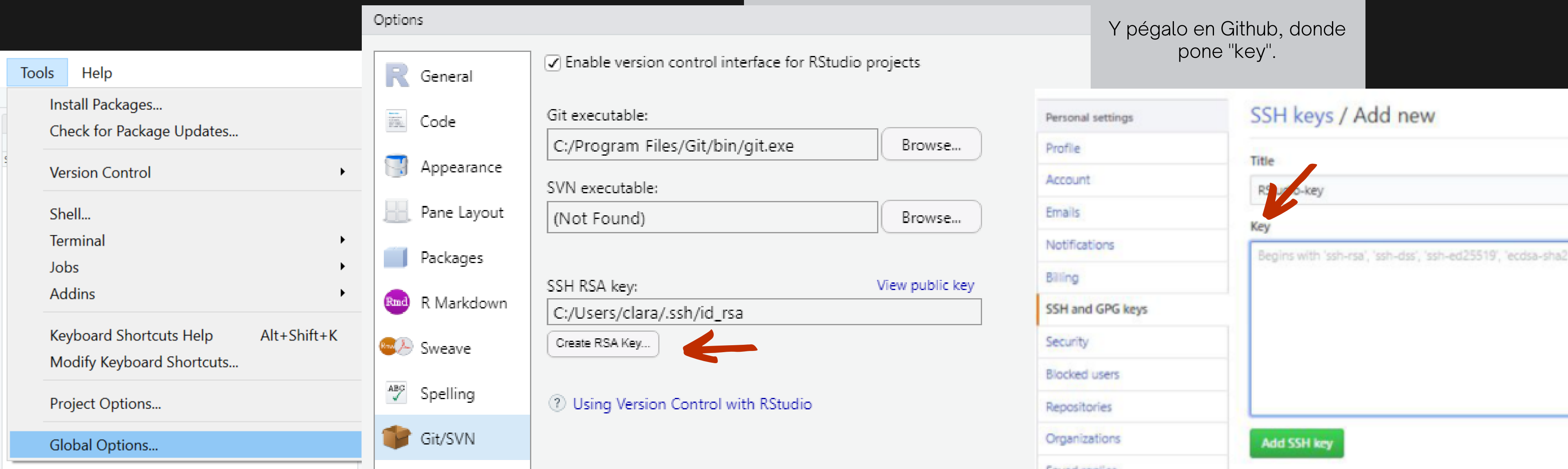
## Vinculándolo a Github

Esto "clonará" el repositorio y creará un proyecto de R.

Si no tienes clave .ssh que vincule tu ordenador con Github, sigue estos pasos:

Pulsa "Create RSA Key" y en "View public key" copia todo el texto.

Y pégalo en Github, donde pone "key".



The image shows two side-by-side screenshots illustrating the steps to set up SSH for RStudio.

**Left Screenshot: RStudio Options dialog**

- The **Tools** menu is open, showing **Global Options...** at the bottom.
- The **Options** dialog is open, with the **Git/SVN** tab selected.
- The **SSH RSA key:** field shows `C:/Users/clara/.ssh/id_rsa`.
- The **Create RSA Key...** button is highlighted with a red arrow.
- The **Enable version control interface for RStudio projects** checkbox is checked.

**Right Screenshot: GitHub SSH key setup**

- The **SSH keys / Add new** page is shown.
- The **Title** field is filled with `RSudio-key`.
- The **Key** field is empty, with a red arrow pointing to it from the text above.
- The **Add SSH key** button is visible at the bottom.

# Crear un proyecto

## Vinculándolo a Github

1. Crea un repositorio en Github. Copia la URL del repositorio
2. Abre RStudio
3. En File, selecciona "New Project"
4. Selecciona Version Control
5. Selecciona Git
6. Pega la URL de tu repositorio como en la imagen y elige dónde quieres ubicar la carpeta.

Abrir un Proyecto una vez creado es tan simple como hacer doble click sobre el archivo .Rproj en tu ordenador o bien, desde RStudio, File > Open Project

New Project Wizard

Back

Create Project from Version Control



Git

Clone a project from a Git repository

New Project Wizard

Back

Clone Git Repository



Repository URL:

<https://github.com/Clapiniella/R-basics>

Project directory name:

R-basics

Create project as subdirectory of:

C:/Users/clara/Desktop

Browse...

☐ Open in new session

Create Project

Cancel

# Guardar, commit & push

Aquí puedes elegir entre hacerlo desde la terminal como hasta ahora o bien hacerlo a través de RStudio. Si eliges esta última opción, sigue estos pasos:

The screenshot displays the RStudio interface with four numbered steps indicating the workflow for committing and pushing changes:

- Step 1:** The **Review Changes** window is open, showing the file `ejemplo.R` staged for commit. The **Commit** button is visible in the bottom toolbar.
- Step 2:** The **Environment** pane shows the file `ejemplo.R` with a question mark icon, indicating it is a new file.
- Step 3:** The **Commit** button in the bottom toolbar is highlighted.
- Step 4:** The **Commit message** field in the **Review Changes** window is filled with the text: "Tras seleccionar el archivo y darle a commit, aparecerá esta ventana. Ahora solo queda escribir aquí un mensaje de commit, hacer clic en Commit y pulsar Push. ¡Hecho!". The **Commit** button is also visible in this window.

The code editor shows the following R code:

```
1 print("abre un script nuevo, haz cambios, ctrl+S para guardar")
```

# Ejercicio práctico

---

01.

## **Crea un repositorio en Github**

Llámallo `Learning-R` de ese modo reconocerás tu espacio de trabajo sin problema. Si decides elegir tú el nombre, ponle un nombre coherente.

---

02.

## **Crea un proyecto y vinculalo al repo**

---

03

## **Abre un nuevo script en tu proyecto**

---

04.

## **Copia en este script el contenido del archivo `EjercicioW11D1.R`**

Este archivo se encuentra en `R-basics/day1`

---

05.

## **Ejecuta el archivo**

Para ejecutar este script necesitarás instalar y cargar el paquete "tidyverse"

---

06.

## **Guarda, commit & push los cambios desde RStudio**

Llama al nuevo script `first\_script.R`

# You got this!



**Relax & keep coding!**