

Introduction to R

Alexis Robert

Based on a script and session developed by
Alexis Robert & Sophie Meakin.

LONDON
SCHOOL of
HYGIENE
& TROPICAL
MEDICINE



Objectives of the module

- Gain a basic understanding of coding in R
 - Understand and use Rstudio
 - **Follow good coding practices**
 - Presentation of basic packages
- Implement a short epidemiological study in R (Data importation => Analysis => Visualisation)
- **Introduce notions of code and project management**

Objectives of the session

- Gain a basic understanding of coding in R
 - Check that R and Rstudio are working on all computers
 - How/Why use Rstudio?
 - Manipulate basic objects

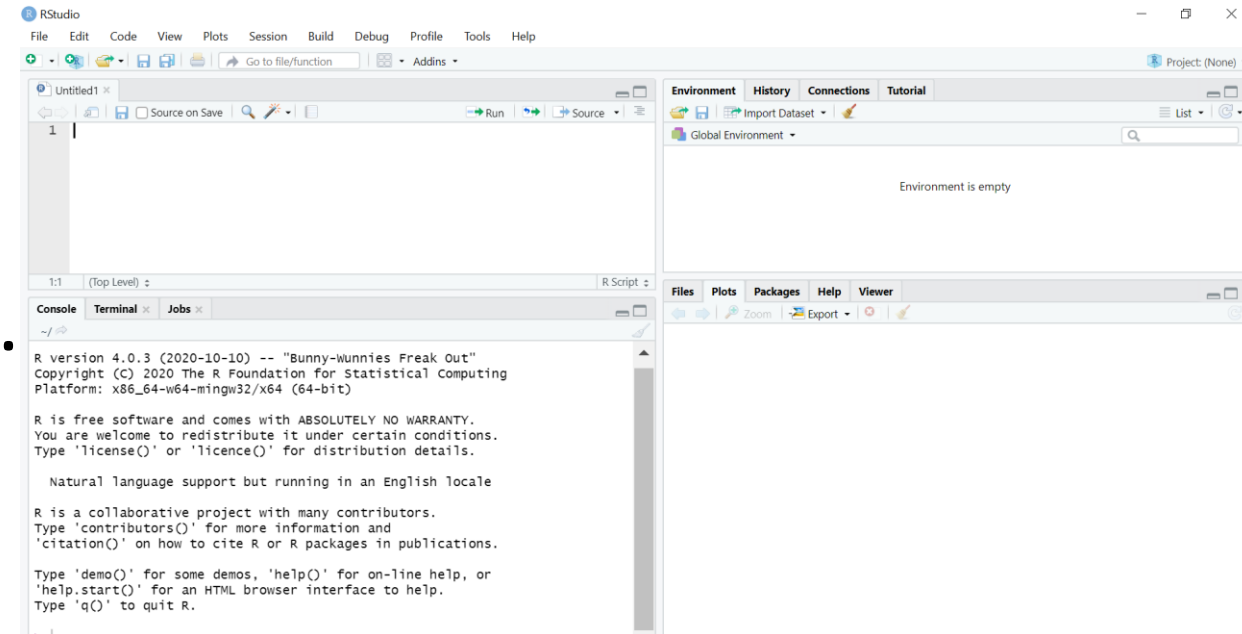
What is R?

LONDON
SCHOOL of
HYGIENE
& TROPICAL
MEDICINE



What is R, what is Rstudio?

- R designates both the language and the software using this language
 - Similar to STATA, C, Fortran, Python...
- R is (mostly) used for statistical computing (i.e. import, manipulate, analyse, and visualise data).
- R has no **graphical user interface**.
- **RStudio** is used as the graphical interface.



Why use R at all?

- Free and open-source.
- Versatile (packages developed for almost everything).
- More intuitive than C / C++.
- Highly popular:
 - Lots of learning resources.
 - Lots of forums / platforms providing help.

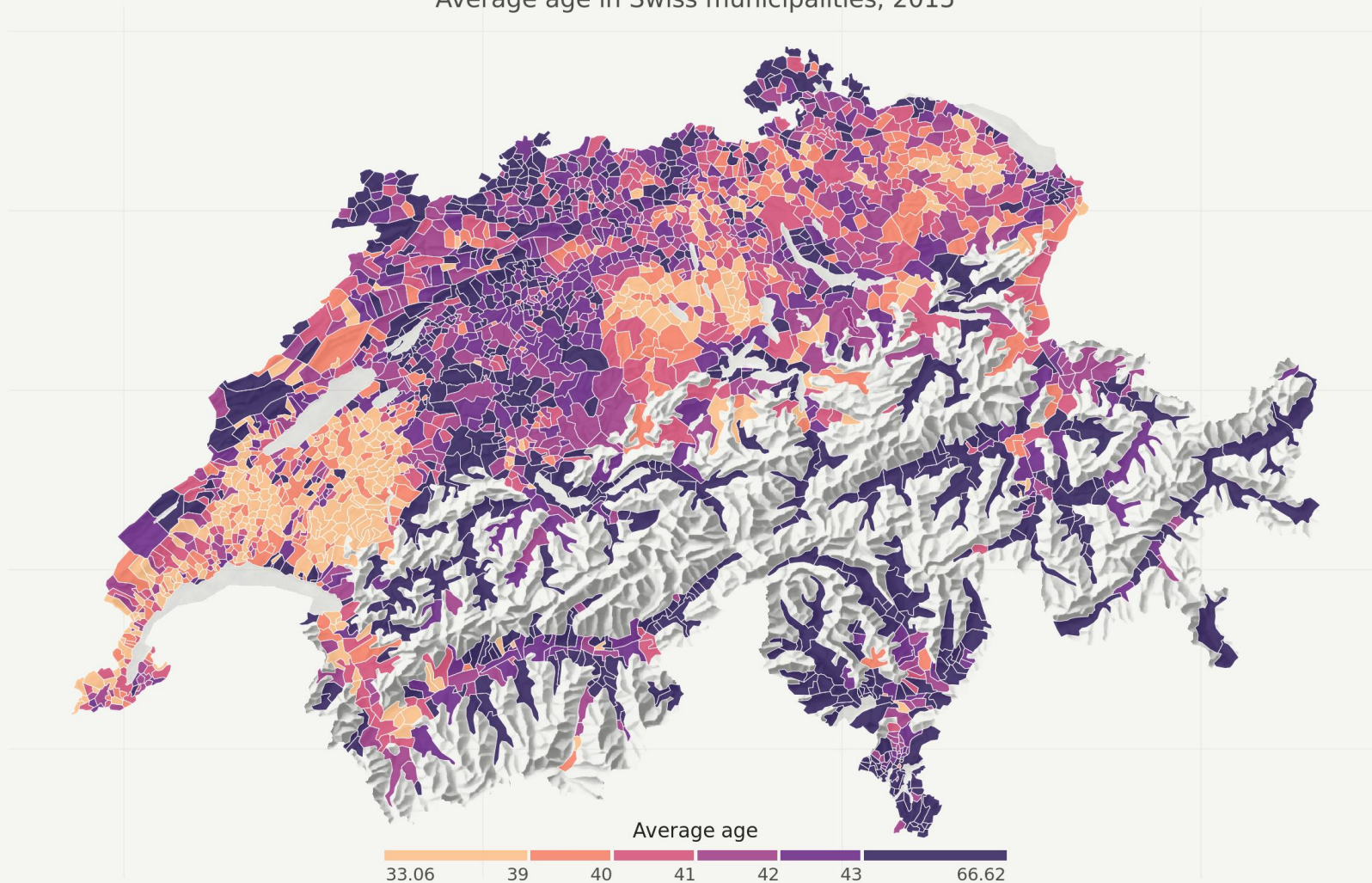
Why use R at all?

- Cool extensions!
 - Data visualization
 - Dashboards e.g. with R Shiny
 - Automated reports

Data visualisation e.g. with ggplot2



Switzerland's regional demographics
Average age in Swiss municipalities, 2015

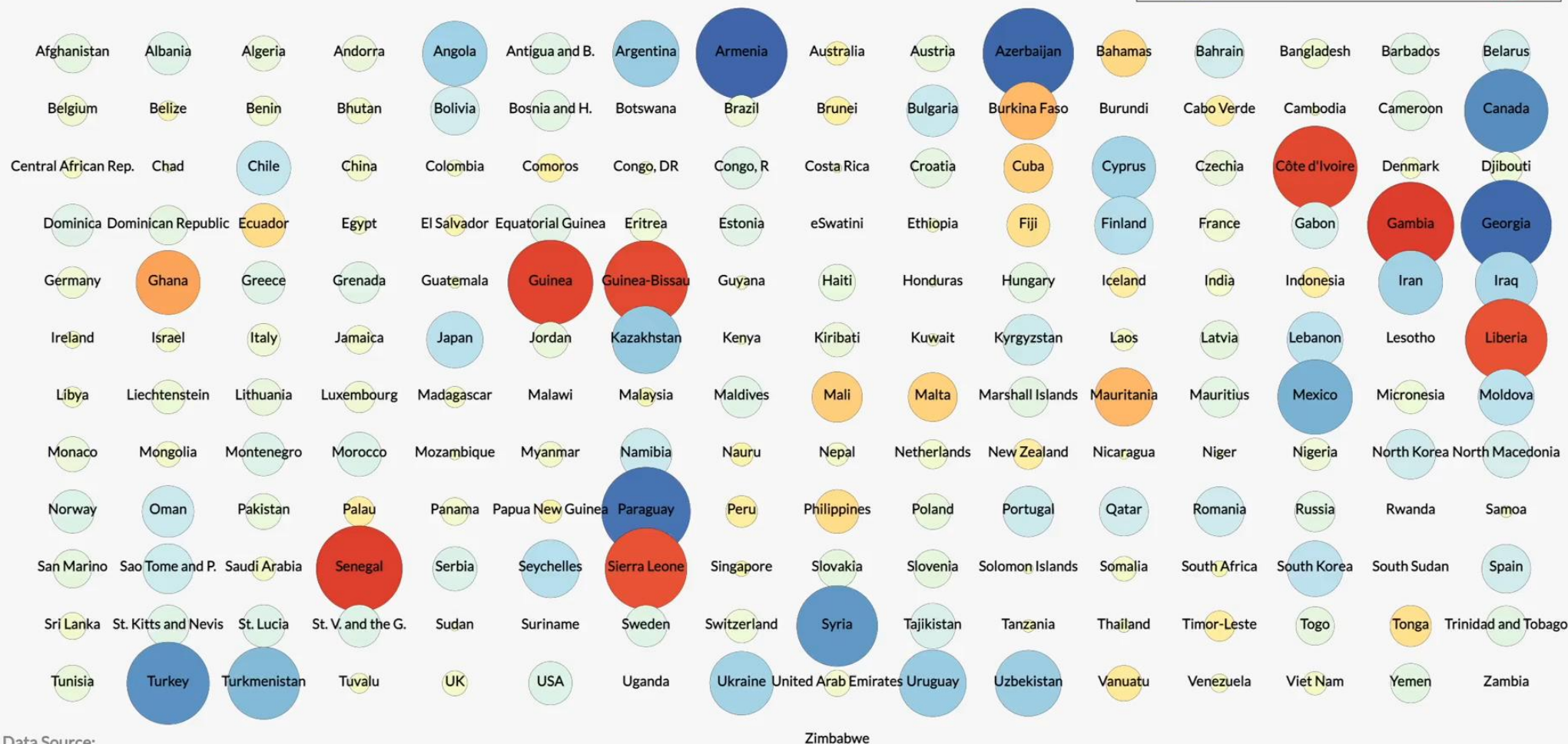
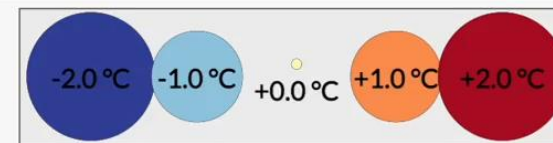


Map CC-BY-SA; Author: Timo Grossenbacher (@grssnbchr), Geometries: ThemaKart, BFS; Data: BFS, 2016; Relief: swisstopo, 2016

Data visualisation e.g. with ggplot2

Temperature Change by Country Years 1880 – 2020

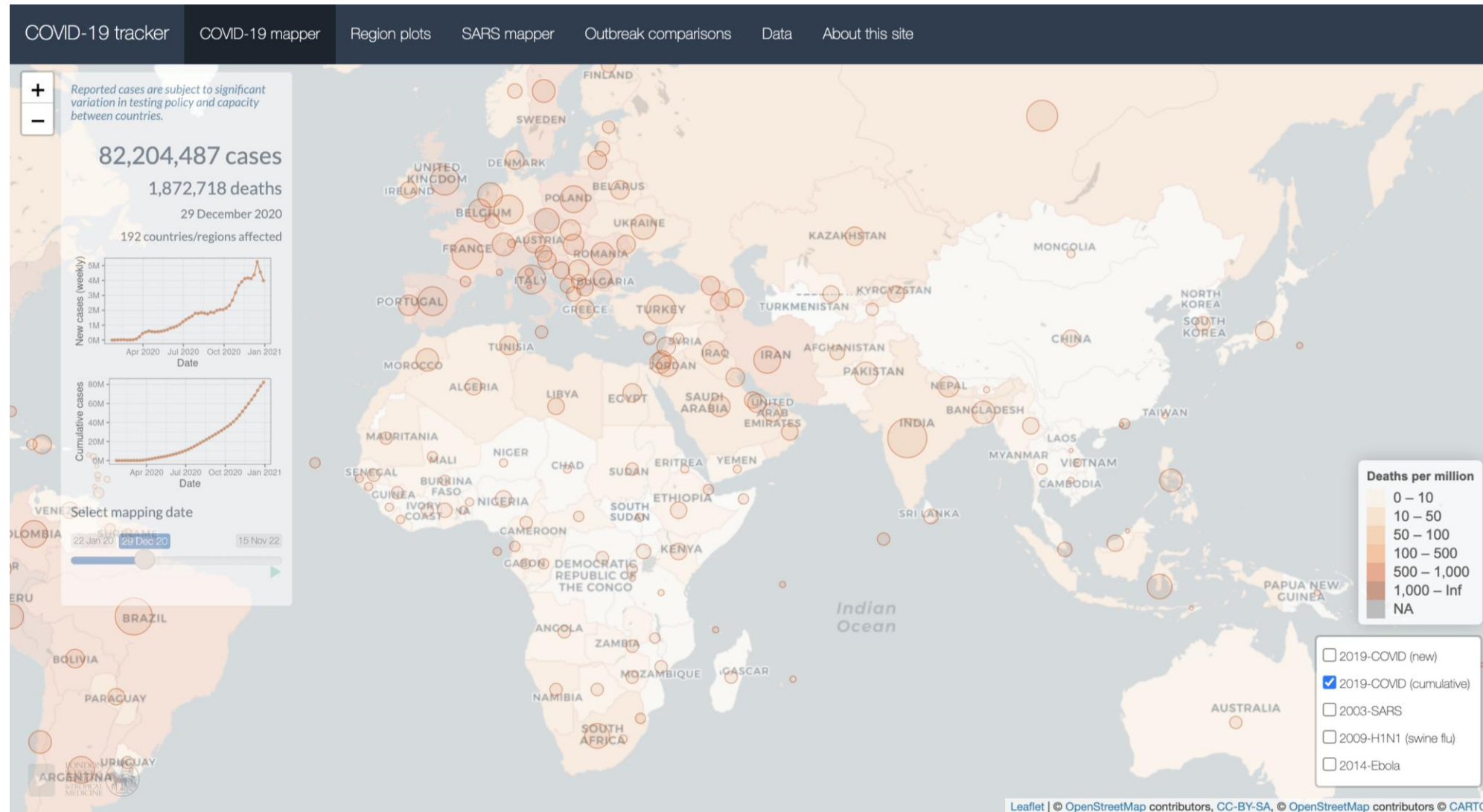
1880



Data Source:
NASA GISS, GISTEMP Land-Ocean Temperature Index, ERSSTv5, 1200km smoothing
<https://data.giss.nasa.gov/gistemp/>
Average of monthly temperature anomalies. GISTEMP base period 1951–1980.

Video license: CC-BY-4.0
Antti Lipponen (@anttilip)

Dashboards e.g. with R Shiny



Useful resources

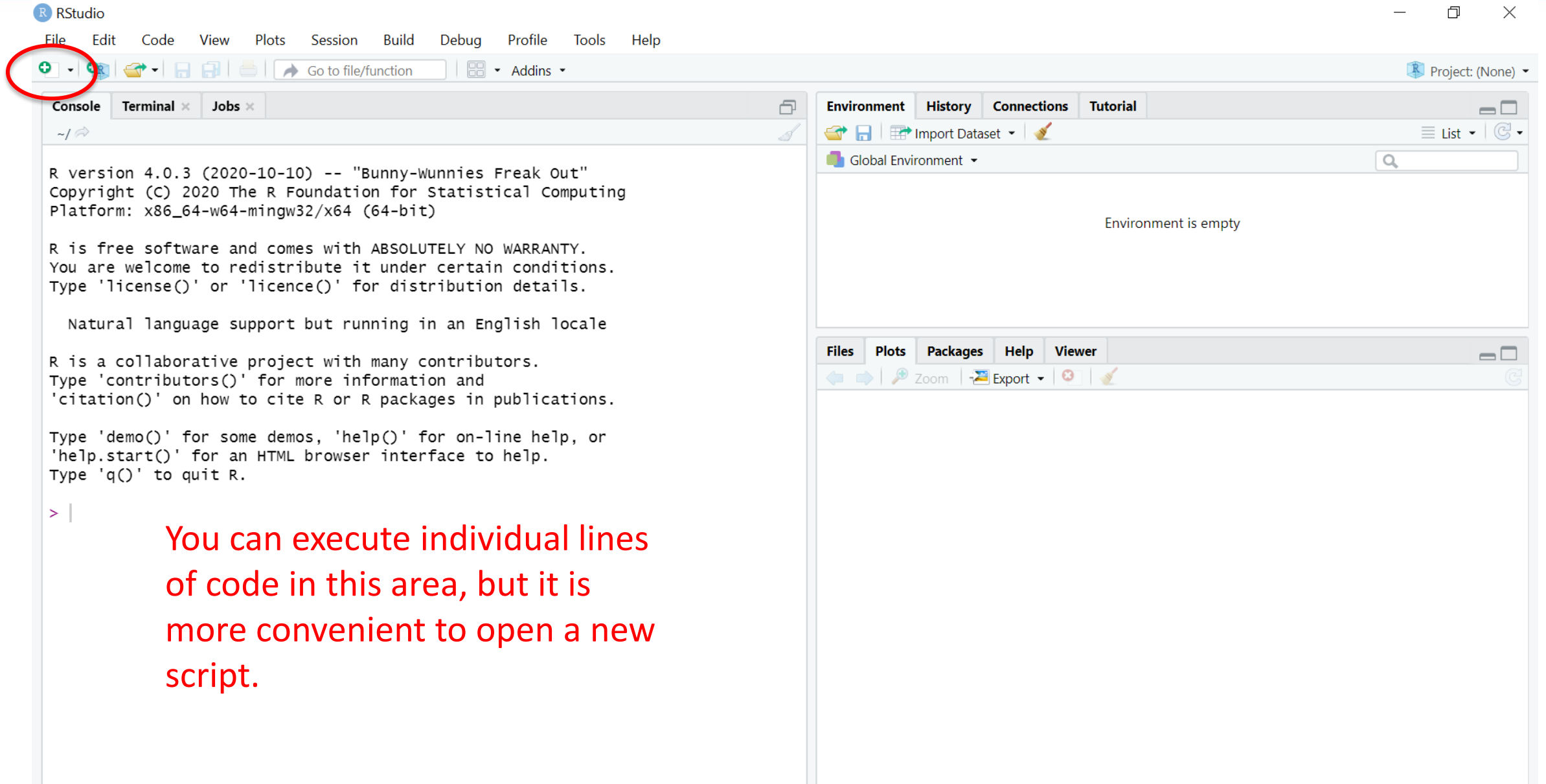
- <https://stackoverflow.com/questions/tagged/r> (questions related to R answered by the community)
- <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf> (Very thorough introduction to R, useful if you need more information on a specific aspect)
- <https://r4ds.had.co.nz/index.html> (R for data science)
- <https://www.epirhandbook.com/en/> (comprehensive introduction to R for applied epidemiology and public health)

Getting familiar with RStudio

LONDON
SCHOOL of
HYGIENE
& TROPICAL
MEDICINE



Opening RStudio and creating a new script



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

+ Go to file/function Addins

Project: (None)

Environment History Connections Tutorial

Global Environment

Environment is empty

Files Plots Packages Help Viewer

Zoom Export

```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 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

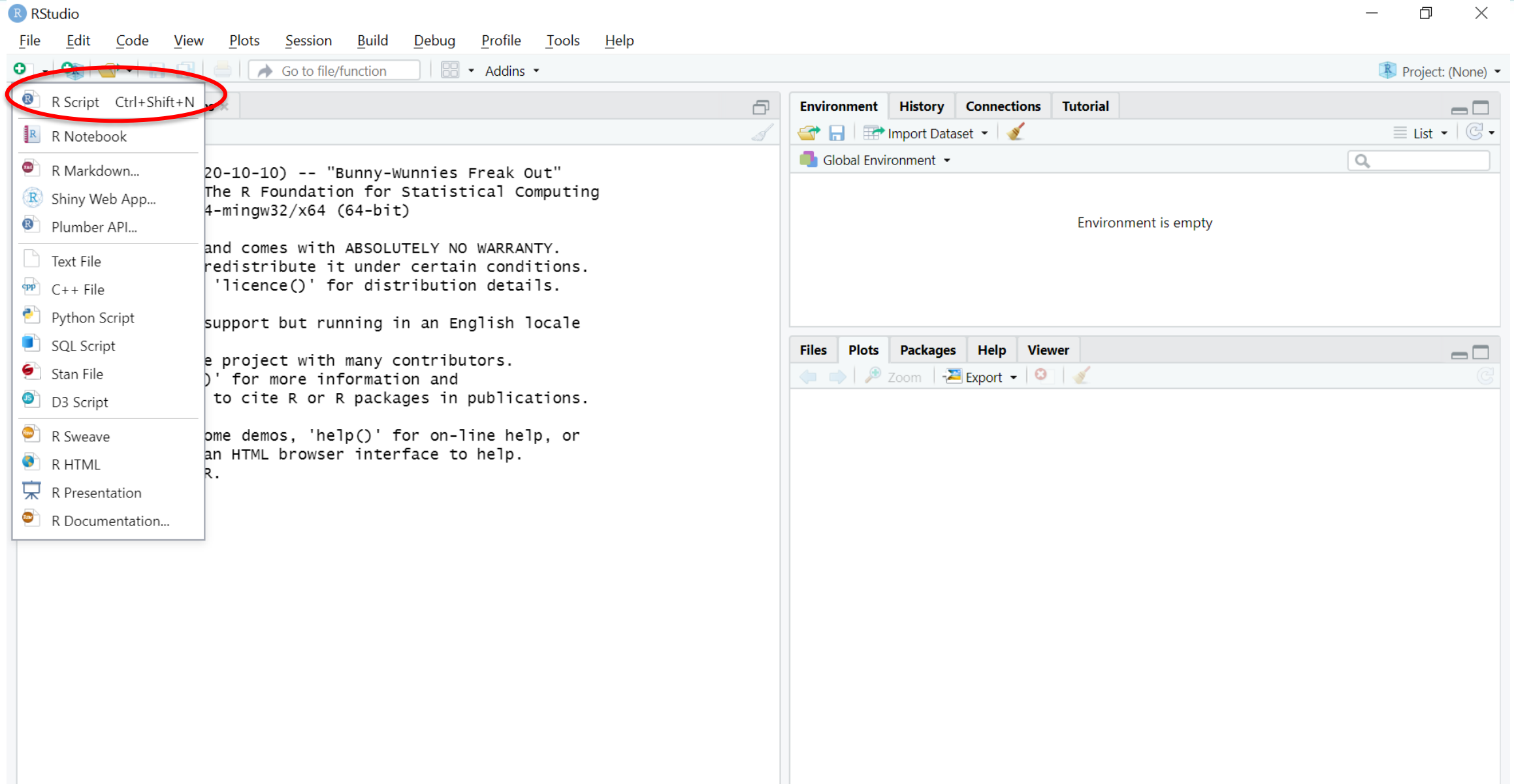
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

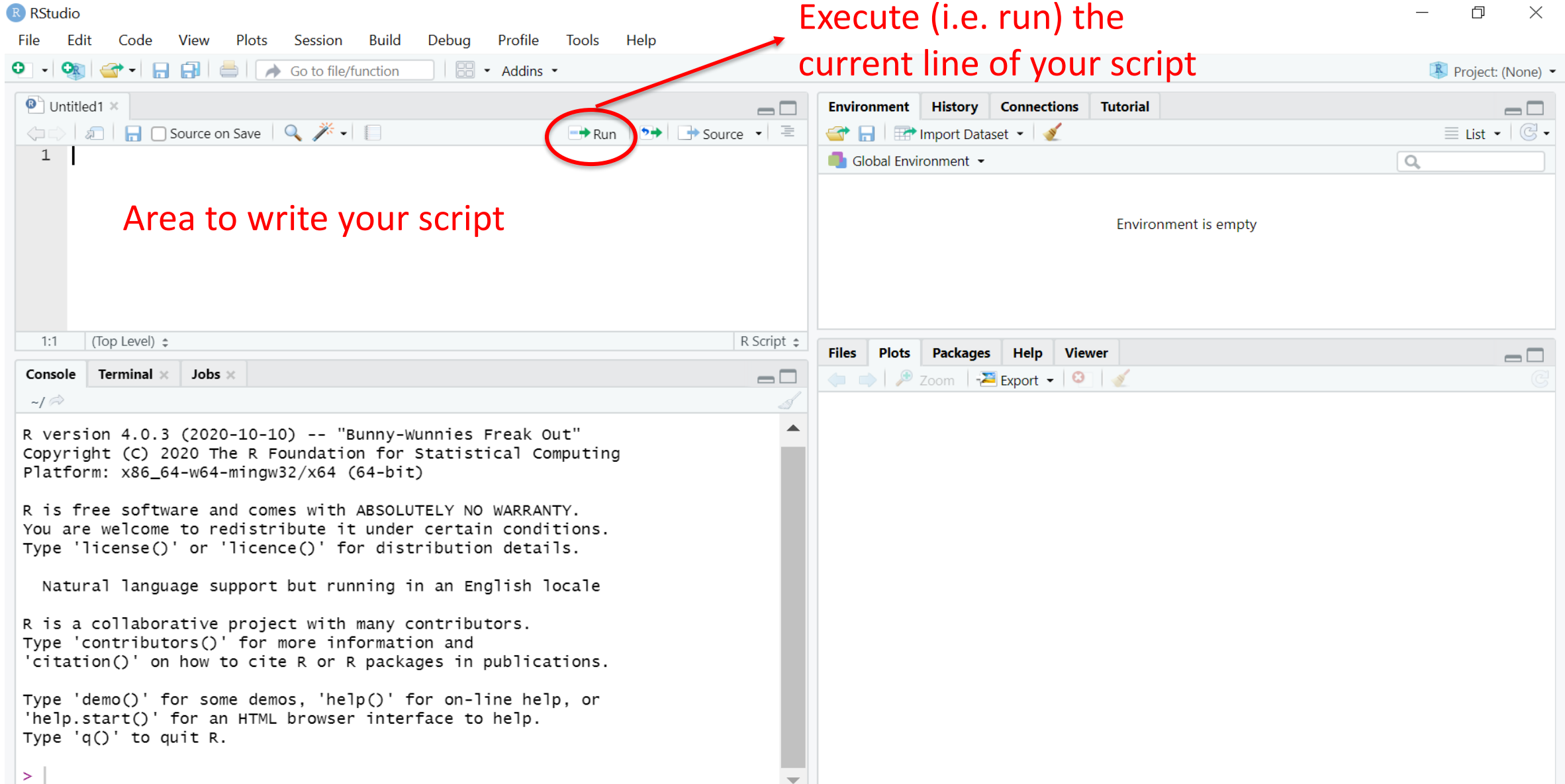
> |
```

You can execute individual lines of code in this area, but it is more convenient to open a new script.

Opening RStudio and creating a new script



The new script is open!



The screenshot shows the RStudio application window. The main editor pane on the left contains a single line of code, which is empty. A red circle highlights the 'Run' button (a green arrow) in the toolbar, with a red arrow pointing to it from the text 'Execute (i.e. run) the current line of your script'. The text 'Area to write your script' is written in red in the center of the editor pane. The console pane at the bottom shows the R startup message, including the version (4.0.3), copyright (2020 The R Foundation for Statistical Computing), and platform (x86_64-w64-mingw32/x64 (64-bit)). The environment pane on the right shows 'Global Environment' and 'Environment is empty'. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The bottom status bar shows '1:1 (Top Level)' and 'R Script'.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

Addins

Untitled1 x

Source on Save

Run

Source

1

Area to write your script

1:1 (Top Level) R Script

Console

Terminal x Jobs x

~/

R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 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.

>

Environment History Connections Tutorial

Import Dataset

Global Environment

Environment is empty

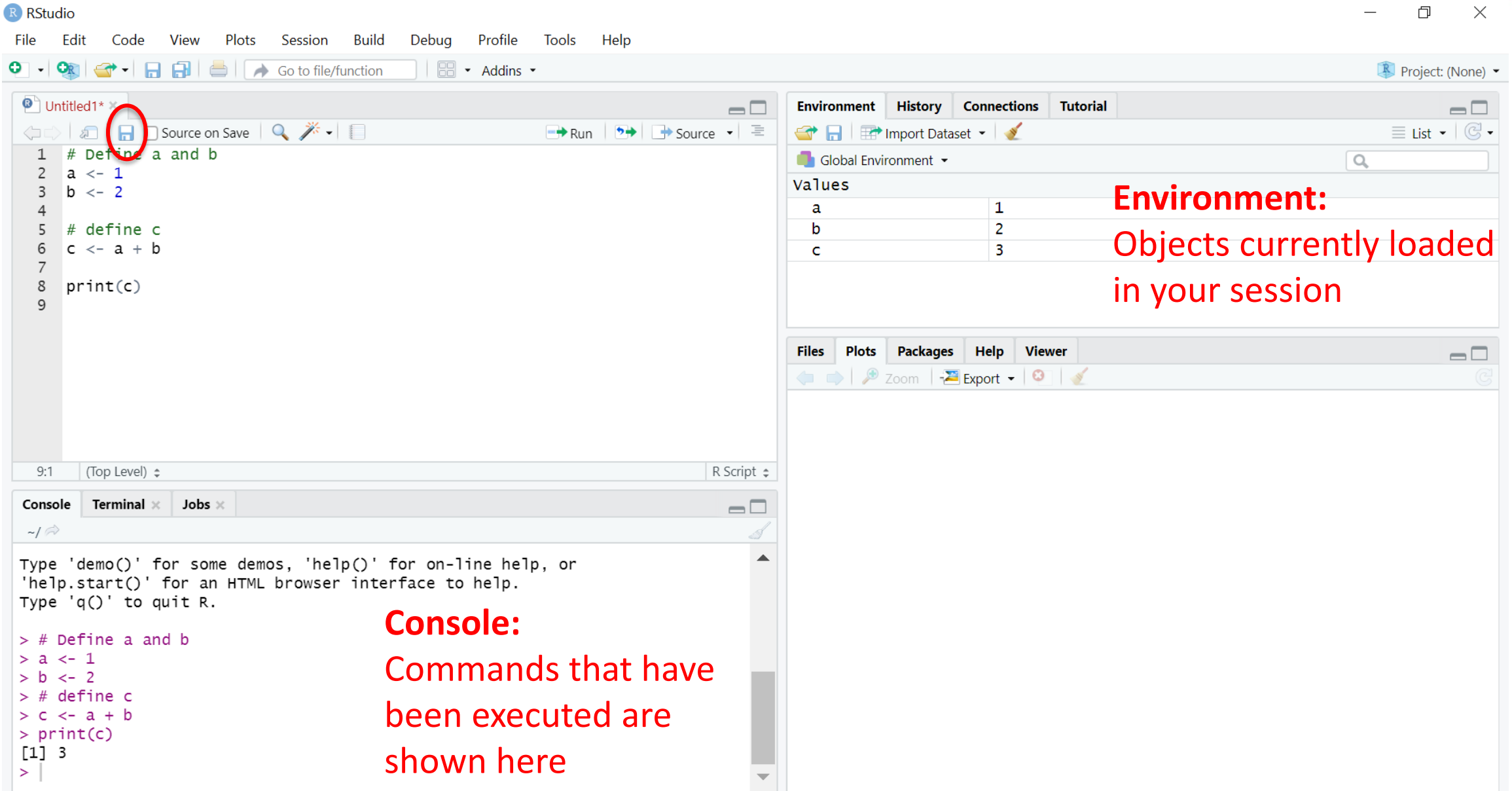
Files Plots Packages Help Viewer

Zoom Export

Why write in the script area?

- A script is the same as a “do-file” in STATA
- You can’t save what you write in the console section.
- Scripts can be saved so you don’t have to re-write code every time you re-open Rstudio.
- Scripts are much clearer for complex commands (i.e. when defining functions, generating plots..).

We can write and save the script



The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains an R script with the following code:

```
1 # Define a and b
2 a <- 1
3 b <- 2
4
5 # define c
6 c <- a + b
7
8 print(c)
9
```

The 'Save' icon (floppy disk) in the toolbar is circled in red.
- Environment Pane:** Shows the 'Global Environment' with the following values:

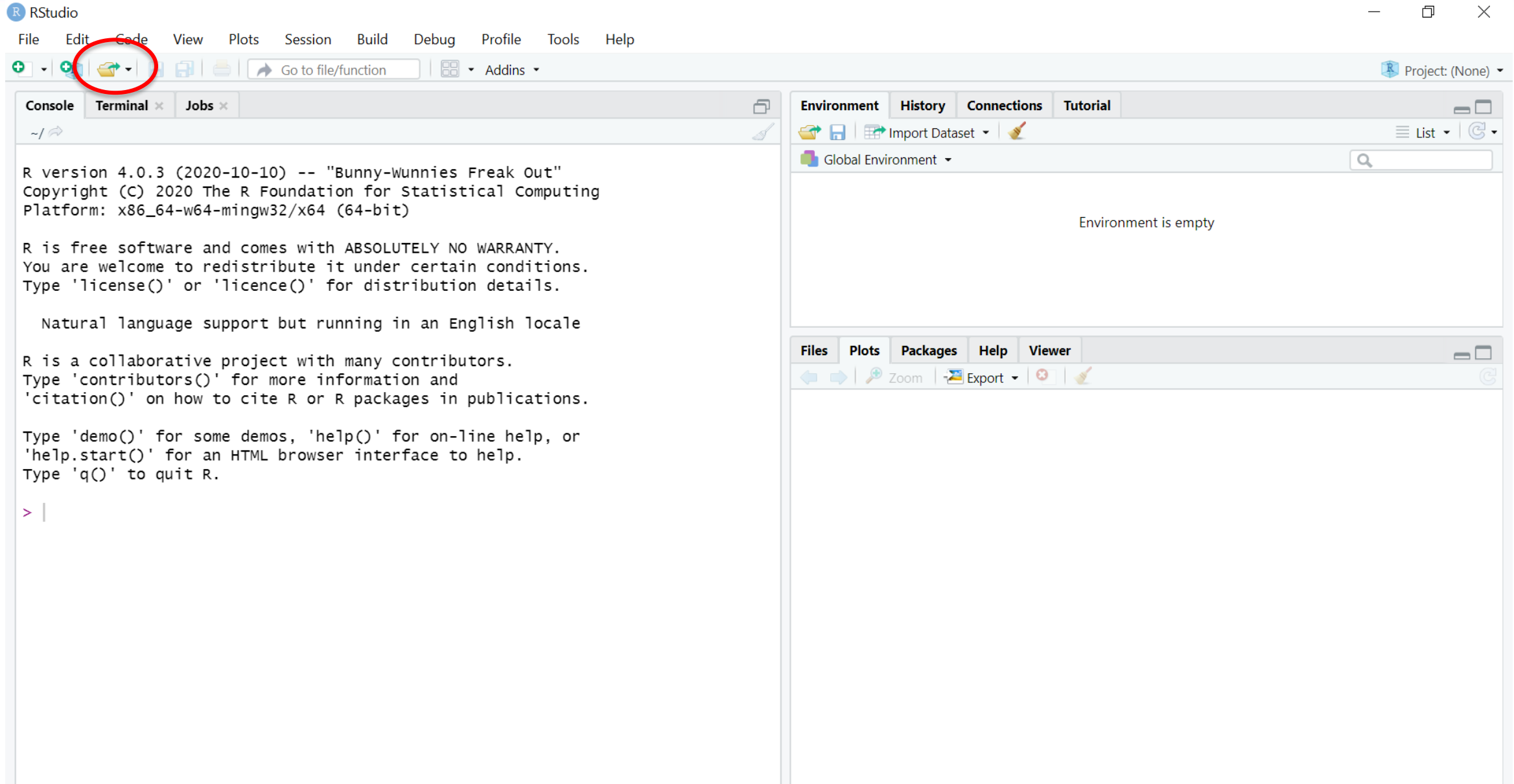
Variable	Value
a	1
b	2
c	3
- Console:** Shows the execution of the script:

```
> # Define a and b
> a <- 1
> b <- 2
> # define c
> c <- a + b
> print(c)
[1] 3
>
```

Environment:
Objects currently loaded
in your session

Console:
Commands that have
been executed are
shown here

Opening a file



The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The 'File' menu is open, and the 'Open' icon (a folder with a magnifying glass) is circled in red. Below the menu bar is a toolbar with various icons, including a 'Go to file/function' button. The main window is divided into four panes: Console, Environment, Plots, and Packages. The Console pane shows the R version 4.0.3 (2020-10-10) and the R Foundation for Statistical Computing logo. The Environment pane shows 'Global Environment' and 'Environment is empty'. The Plots pane shows 'Zoom' and 'Export' buttons. The Packages pane shows 'Export' and 'Help' buttons.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

Project: (None)

Environment History Connections Tutorial

Global Environment

Environment is empty

Files Plots Packages Help Viewer

Zoom Export

```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 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.

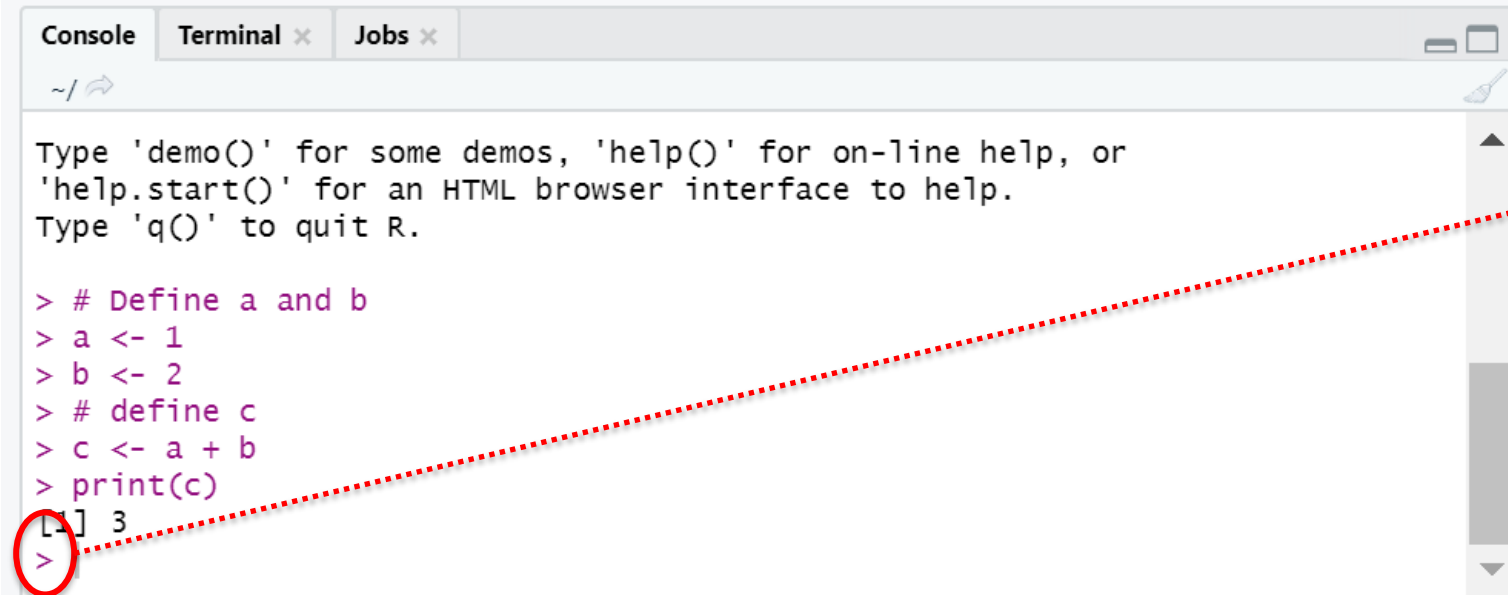
Natural language support but running in an English locale

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.

> |
```

Console lines start with “>”

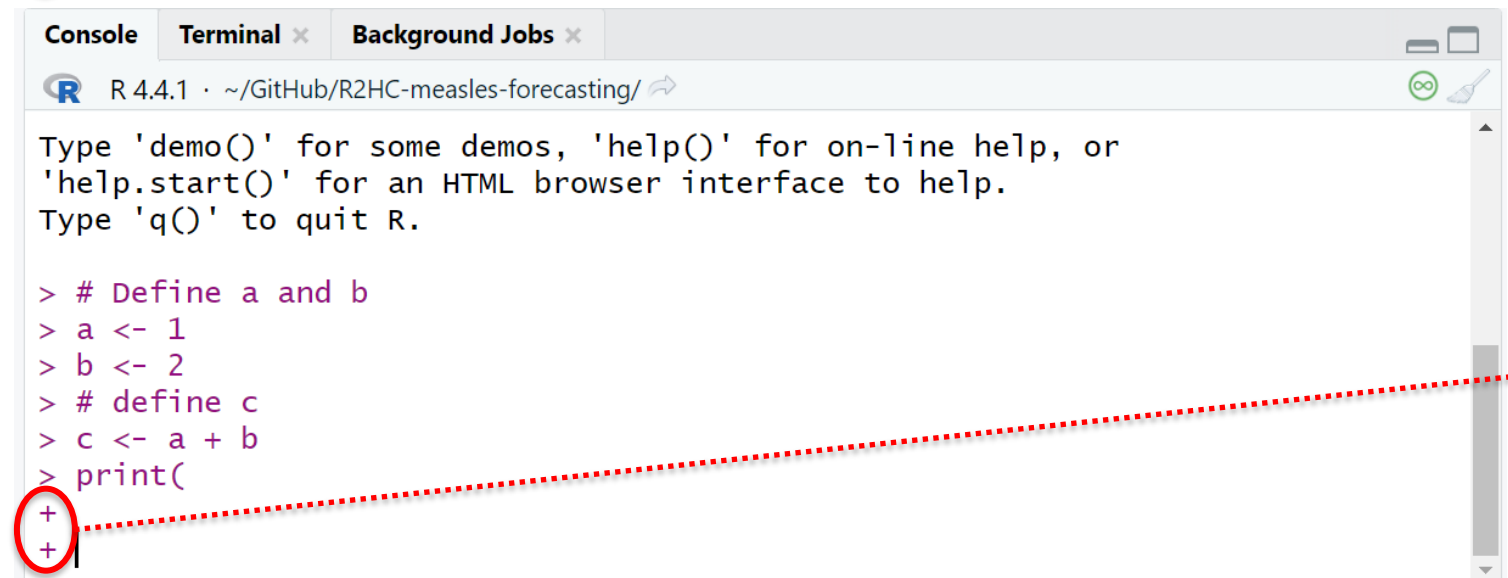


```
Console Terminal x Jobs x
~/
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> # Define a and b
> a <- 1
> b <- 2
> # define c
> c <- a + b
> print(c)
[1] 3
>
```

A screenshot of an R console window. The title bar shows 'Console', 'Terminal x', and 'Jobs x'. The window content shows the R prompt and a series of commands: defining variables a and b, calculating c as a + b, and printing c. The output is [1] 3. The next prompt is >. A red circle highlights the > prompt, and a red dotted arrow points from it to the text on the right.

Line starts with a >:
R will run the command



```
Console Terminal x Background Jobs x
R 4.4.1 · ~/GitHub/R2HC-measles-forecasting/
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.

> # Define a and b
> a <- 1
> b <- 2
> # define c
> c <- a + b
> print(
+
+
```

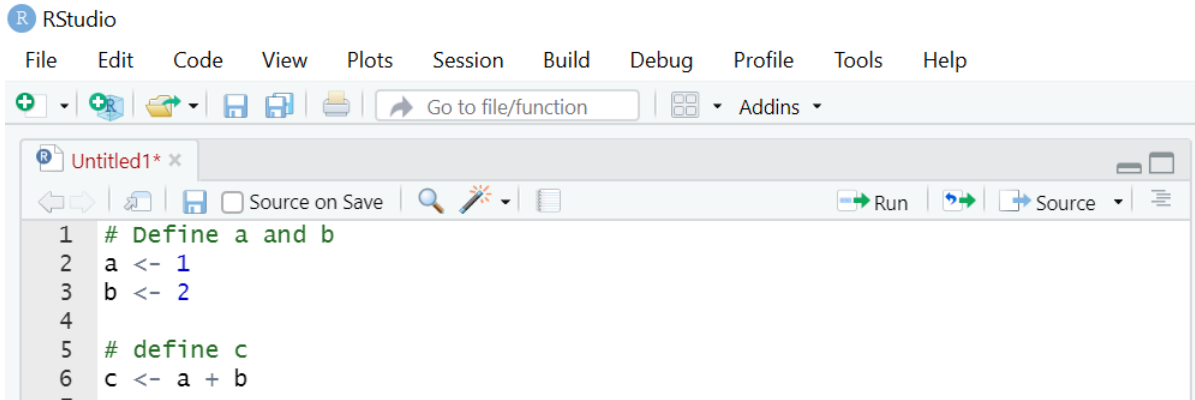
A screenshot of an R console window. The title bar shows 'Console', 'Terminal x', and 'Background Jobs x'. The window content shows the R prompt and a series of commands: defining variables a and b, calculating c as a + b, and starting to print c. The output is [1] 3. The next prompt is >. A red circle highlights the > prompt, and a red dotted arrow points from it to the text on the right.

Line starts with a +:
A parenthesis (or bracket, quotation mark..) is not closed, so R is continuing the previous line of code
Press ESC to exit

Basic objects and expressions



Basic R expressions



The screenshot shows the RStudio application window. The menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains icons for file operations and a search bar. The script editor shows a file named 'Untitled1*' with the following R code:

```
1 # Define a and b
2 a <- 1
3 b <- 2
4
5 # define c
6 c <- a + b
```

- The assign operator “<-”
 - Very similar to “=”
 - Shortcut “alt”+“-” (Mac: “Option”+“-”)
 - Use to assign a **value** to an **object**

- The most basic expressions are the same as would be anywhere else:
 - “+” to sum.
 - “-” to subtract.
 - “*” to multiply.
 - “^” to power.
 - “/” to divide.
- Others have functions integrated in the base version of R
 - “sqrt(5)” to compute the square root (of 5 in that case).
 - “exp(5)” to compute the exponential (of 5).
 - “log(5)” to compute the logarithm (of 5).

Data types and objects

- There are four main data types that you will frequently come across in R:
 - Character: "a", "hello"
 - Numeric: 2, 2.1, pi
 - Integer: 2L, 9:100.
 - Logical: True, False
- The function "class()" can be used to tell you the type of a given object.

Storing data

- Datasets (i.e. objects containing data) can take different shapes:
 - Vectors: Contain only one variable per element.
 - Matrices: Can contain several variables per element.
 - Data frames: Can contain several variables per element, of different types.

Vectors, matrices, dataframes: what do they look like?

- Datasets (i.e. objects containing data) can take different shapes:
 - Vectors: Contain only one variable per element.
 - Matrices: Can contain several variables per element.
 - Data frames: Can contain several variables per element, of different types.

Vector:

10	5	25	33	7	15	73	33	3
----	---	----	----	---	----	----	----	---

Matrix:

ID	Height
1	165
2	160
3	162
4	184
5	189
6	175

Data frame:

Country	Mean temperature
UK	8.45
Brazil	24.95
France	10.70
Morocco	17.10
Japan	11.15

- Datasets (i.e. objects containing data) can take different shapes:
 - Vectors: Contain only one variable per element.
 - Matrices: Can contain several variables per element.
 - Data frames: Can contain several variables per element, of different types.
- Base functions are used to compute the mean, median, or summary of a given dataset.
 - “mean()”
 - “median()”
 - “summary()”
 - “quantile()”

- Vectors contain one or more elements, usually of the same class.
- A vector is constructed using the function `'c()'`
 - `'vec <- c(1,2,3)'` initialises a numeric vector, called `vec`, that contains three elements.

1	2	3
---	---	---

- `'vec <- c("a","b","c")'` initialises a vector of characters, called `vec`, that contains three elements.

"a"	"b"	"c"
-----	-----	-----

`'vec <- c("a","b","c")'` initialises a vector of characters, called `vec`, that contains three elements.

"a"	"b"	"c"
-----	-----	-----

- Values of a vector are accessed using brackets `'[ ]'`:
 - `'vec[3]'` designates the third element of the vector `'vec'`.
- `'vec[-3]'` returns the vector `'vec'` minus the third element.

`vec[3] =`

"c"

`vec[-3] =`

"a"	"b"
-----	-----

Note: Brackets or parentheses?

- We have presented commands that require parenthesis `'()'` and brackets `'[]'`
- Brackets `[]` mean you are **accessing an element of an object**
 - `'X[3]'`: access the third element of the vector X
- Parentheses `()` mean that you are **calling a function**
 - `'X(3)'`: call a function called X
 - `'mean(Y)'`: compute the mean of the vector Y

Matrices: multiple vectors side-by-side

- A matrix is a two-dimensional group of vectors, containing elements of the same class.
- A matrix is constructed using the function '**matrix()**', which needs 3 arguments:
 - 'data': the values of the elements (by default set to NA).
 - 'nrow': the number of rows (default to 1).
 - 'ncol': the number of columns (default to 1).
- '`mat <- matrix(c(1,2,3,4), nrow = 2, ncol = 2)`' initialises a matrix, called 'mat', with 2 rows and 2 columns:

1	3
2	4

Matrices: multiple vectors side-by-side

- `'mat <- matrix(c(1,2,3,4), nrow = 2, ncol = 2)'` initialises a matrix, called 'mat'

1	3
2	4

- To access an element of a matrix, R needs the row(s) and column(s) we are interested in. We use `[,]` where
 - The value before the comma is the row number
 - the value after the comma is the column number

- `'mat[1,2]'` will return the element on the first row, and second column.

`mat[1,2] =`

3

- `'mat[1,]'` will return all elements on the first row

`mat[1, ] =`

1	3
---	---

- What if you want to store data describing various features for a set of individuals?
 - IDs number (numeric), name (character), gender (character), age (numeric), date of birth (Date) etc..
- Matrices can only contain one class of elements (all characters, all numeric..)
- Dataframes are similar to matrices, except that they can contain multiple data types.
- They are created using the command `'data.frame()'`
- Their elements can be accessed in the same way as matrices, plus specific functions.

Fixing your code and finding help

LONDON
SCHOOL of
HYGIENE
& TROPICAL
MEDICINE



Communicating with R: Errors and warnings

- Coding is learning to communicate with your computer, R will use **errors, warnings, and messages** to signal unintended behaviour and potential errors.
- Messages:
 - Inform users that R has performed certain actions; **do not prevent the code from running.**
- Warnings:
 - Indicate that something may have gone wrong, but **do not prevent the code from running, start with “Warning: [...]”.**
- Errors:
 - **Interrupt the execution of the code.** Indicate a severe problem in the code. Start with “Error: [...]”.

Communicating with R: Errors and warnings

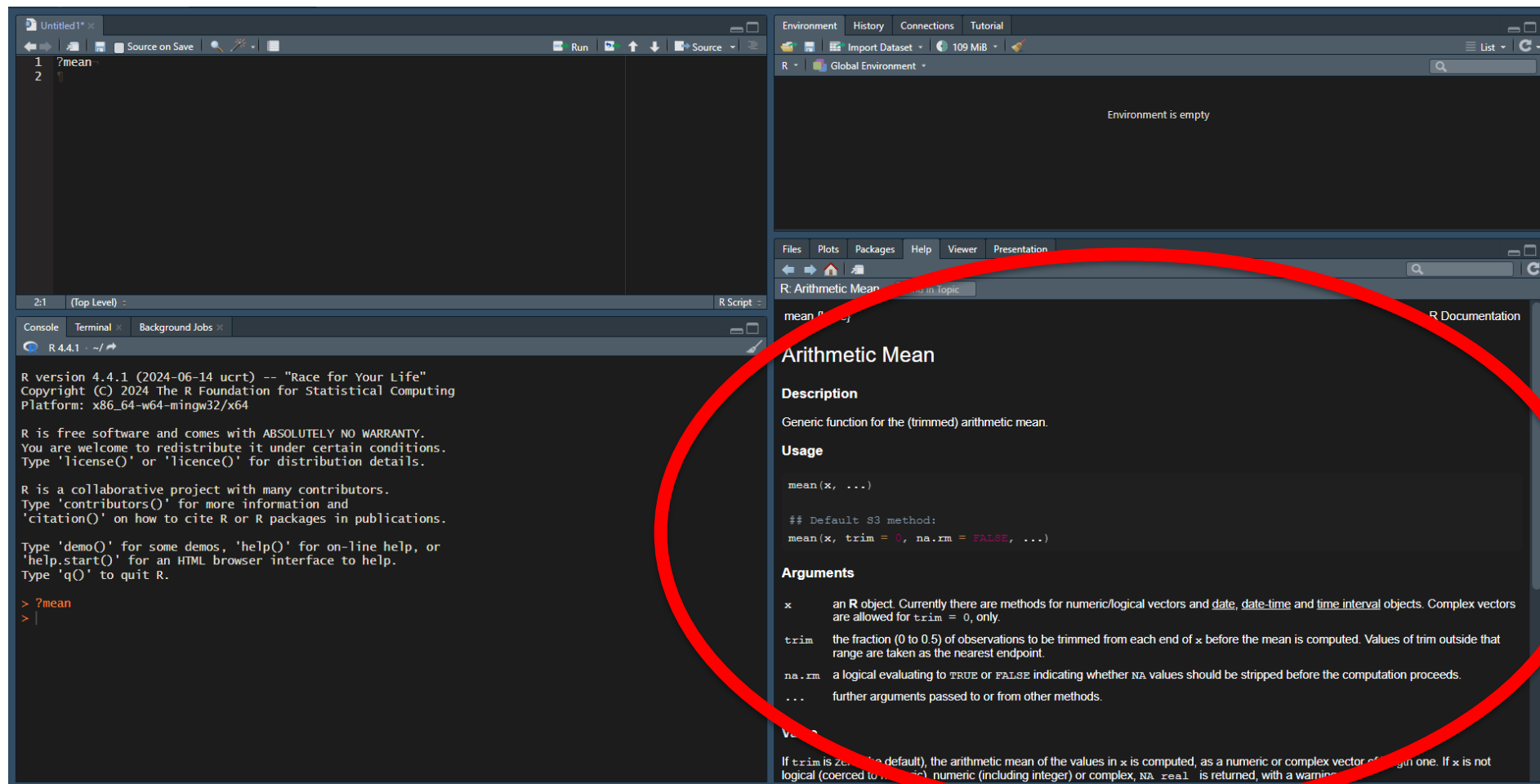
- Some warnings may not be relevant to you

```
> library(tidyr)  
Warning message:  
package 'tidyr' was built under R version 4.2.2
```

- Errors will have to be fixed

Helpfiles describe the arguments and outputs of functions

- Use helpfiles to see a description of the expected arguments and output of a function, using the operator `?` (e.g. `?mean`).



```
1 ?mean
2 |
```

R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

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.

```
> ?mean
> |
```

R: Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)

Arguments

x an R object. Currently there are methods for numeric/logical vectors and `date`, `date-time` and `time interval` objects. Complex vectors are allowed for `trim = 0`, only.

trim 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.

na.rm a logical evaluating to `TRUE` or `FALSE` indicating whether NA values should be stripped before the computation proceeds.

... further arguments passed to or from other methods.

Value

If `trim` is 0 (the default), the arithmetic mean of the values in `x` is computed, as a numeric or complex vector of length one. If `x` is not logical (coerced to logical), numeric (including integer) or complex, NA `real` is returned, with a warning.

Helpfiles describe the arguments and outputs of functions

- If helpfiles do not clarify the use of a function, look up the name of the function online (along with the error message if you are debugging).
- The online R community is very active and there are many posts and resources to understand the use of functions / why a part of the code is not running.
- AI & debugging

- Go through the practical.
- The main commands are given in the documents, try and manipulate the data and functions to make sure you understand how they work.