Introduction to R

Part 1: R and RStudio

QantArch Week 1 | 07-02-2022

Universiteit Leiden

Who am I?

PhD student, Archaeological Sciences
R user since 2017

OsteoBjorn

bbartholdy

bjorn.rbind.io

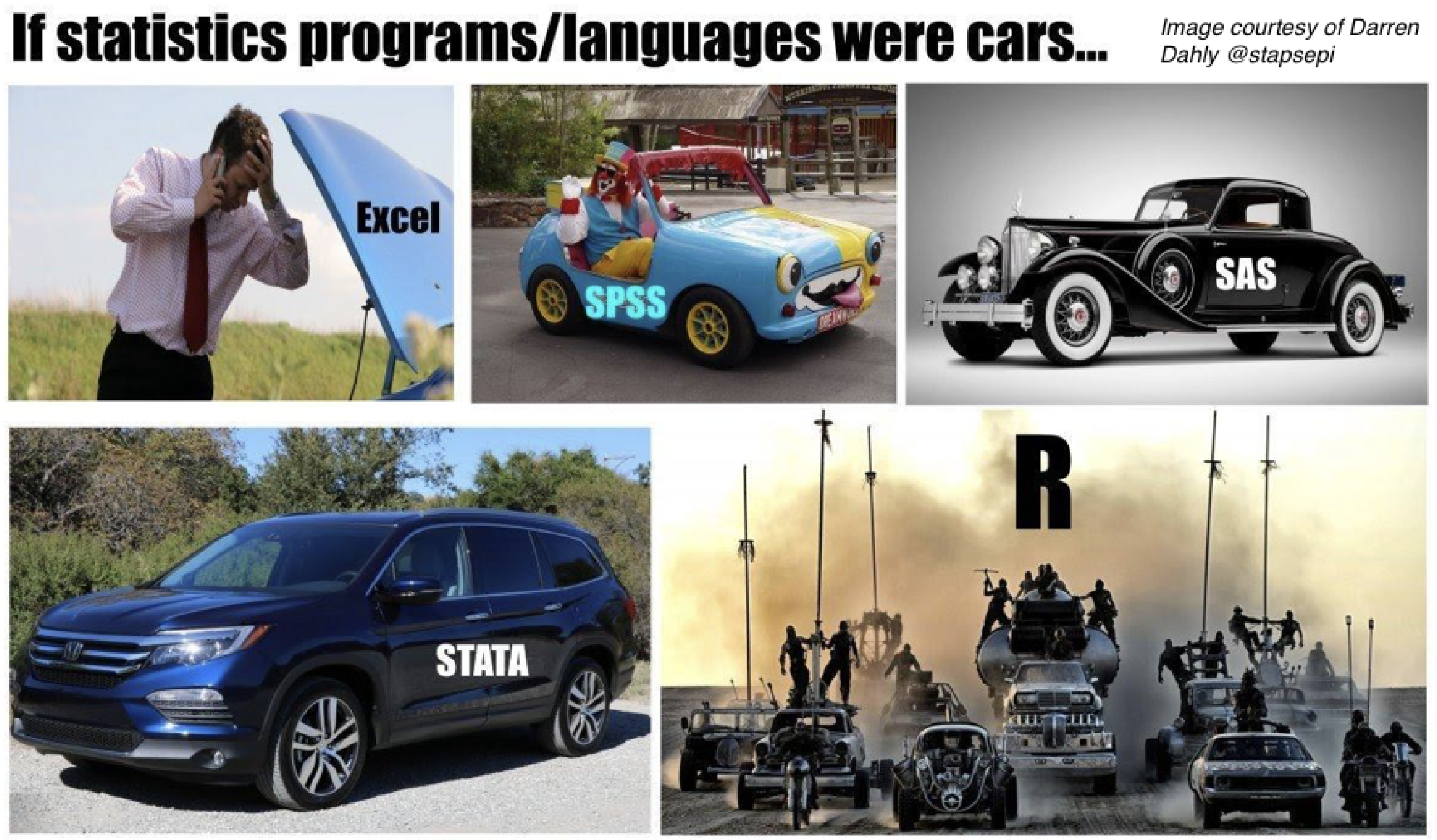
R

/'Arrrgh/

Why R?

R is **free** and **open source**,

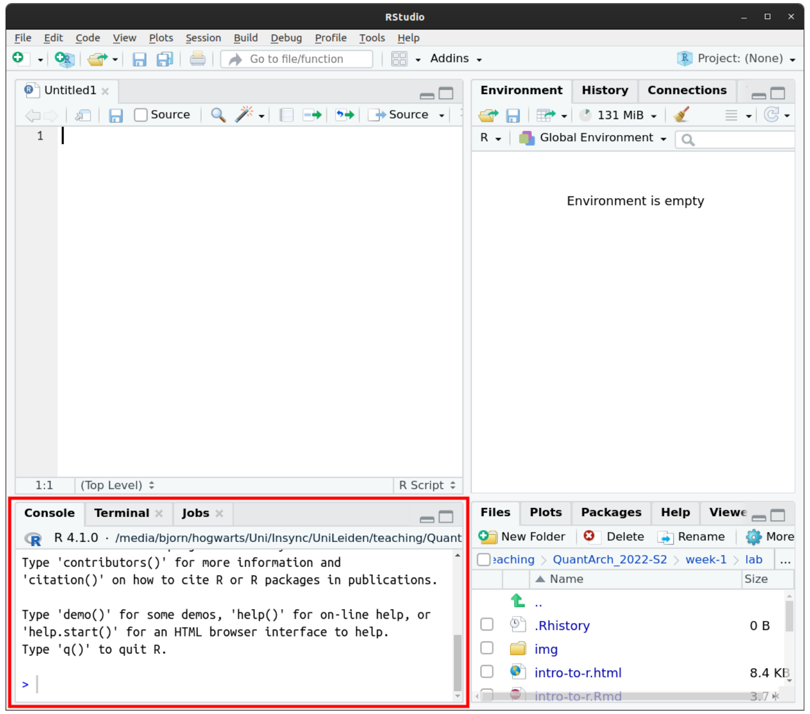
created primarily for statistics.


If statistics programs/languages were cars...
Image courtesy of Darren Dahly @stapsepi
Excel
SPSS
SAS
STATA
R

RStudio

RStudio is an Integrated Development Environment (IDE) for R. You will only have to interact with RStudio, there is no need to ever open R, as it is integrated into RStudio.

You will be able to interact with R in the **Console** pane in RStudio (left).



Environment is empty

```
R 4.1.0 · /media/bjorn/hogwarts/Uni/Insync/UniLeiden/teaching/Quant
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.

>
```

NOTE: Your console may take up the entire left side of RStudio; this is totally fine

Basic usage

The console can be used for basic math. For example:

```
3 + 5
```

```
## [1] 8
```

Or

```
12 / 7
```

```
## [1] 1.714286
```

Or even one of the dreaded internet equations

```
8 / 2 * (2 + 2) # which we can solve without breaking the internet
```

```
## [1] 16
```

Basic usage

When using the console, the `>` symbol indicates where you type your code. The symbol will not be present where you have your output. Instead, you may see `[1]`, which indicates the first element of your output.

See what happens when we have an output with multiple elements; for example, we can print the whole numbers from 1 to 42:

```
1:42
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
```

If you don't see the `>` in the console, it may still be busy with something. Wait until the symbol returns before you run more code.

If, instead, you see the `+` symbol, it means R is waiting for more input. To get back to `>`, just press [Esc].

RStudio Projects

Super convenient!

RStudio Projects are a great tool to help you stay organised and allow you to work with **relative paths** instead of **absolute paths**.

```
"~/Path/Is/Probably/Unique/To/You/QuantArch_2022/week-1" # absolute path
"./week-1" # path relative to QuantArch_2022
```

Combined with the **here** package you can always make sure the project root is the working directory, even when storing scripts and R Markdown files in folders within the root directory (more on this later).

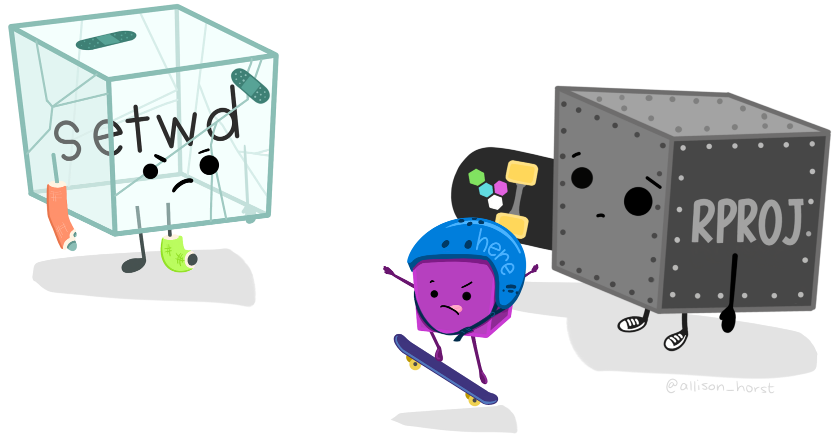


Image credit: Allison Horst

Yes yes, let's get started already!

Create a new project in the top right corner of RStudio:

- Project: (None) > New Project… > New Directory > New Project

You can call the project QuantArch_2022 (or something like that)

Once the project is created, a fresh RStudio environment will open.

Project Organisation

You can organise your folder by weeks. Create a new folder for each week. This can also be done in the lower right pane. Select the **Files** tab and then click on **New Folder** and call it week_1 (or something similar).

If you know you want to create 7 files and you are lazy (like me), you can run the following in the **Console**:

```r
sapply(paste0("week-", 1:7), dir.create)
```

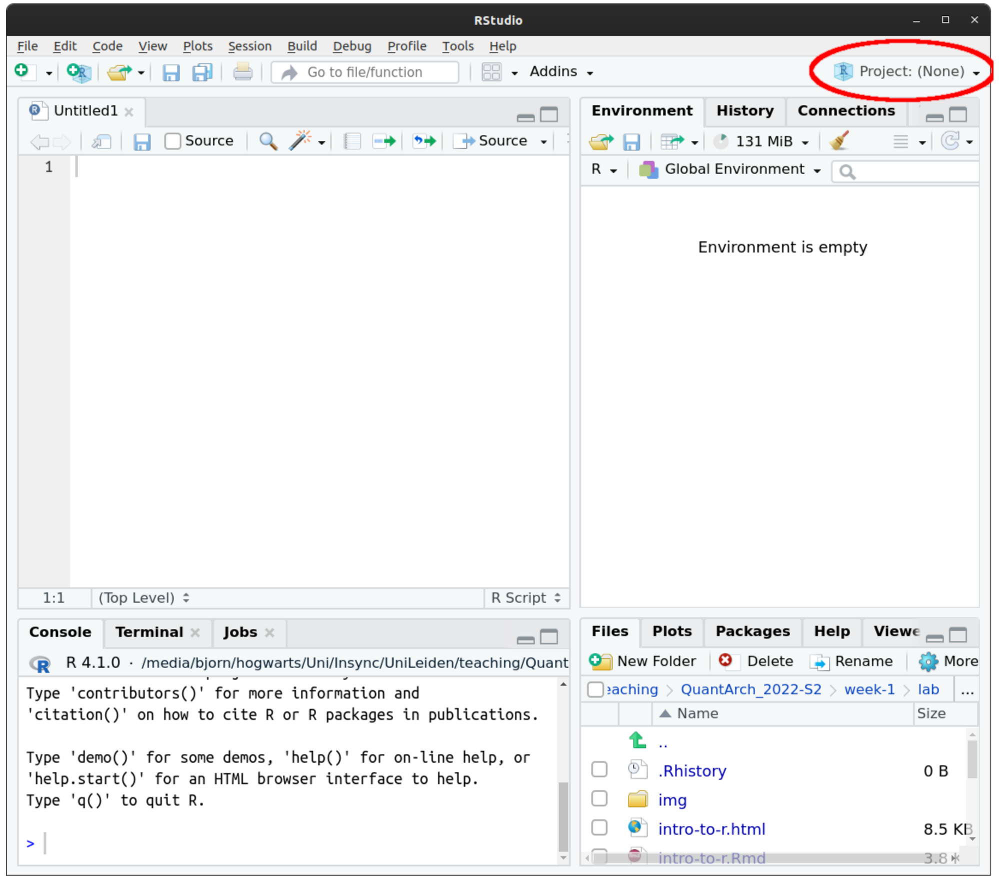
Within the directories for each week, you may also want to create a 'lab' and 'lecture' folder.

Did I mention convenience?

Using a Project also allows you to conveniently pick up where you left off.

When you want to continue your work in the next lab,

you can select your project from the drop-down menu in the top right corner.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1
Source
Source
1:1
(Top Level)
R Script
Environment History Connections
131 MiB
R
Global Environment
Environment is empty
Console Terminal Jobs
Files Plots Packages Help Viewer
New Folder Delete Rename More
aching > QuantArch_2022-S2 > week-1 > lab
Name
Size

```
R 4.1.0 · /media/bjorn/hogwarts/Uni/Insync/UniLeiden/teaching/Quant
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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

>
```

Name	Size
..	
.Rhistory	0 B
img	
intro-to-r.html	8.5 KB
intro-to-r.Rmd	3.8 K

R Scripts

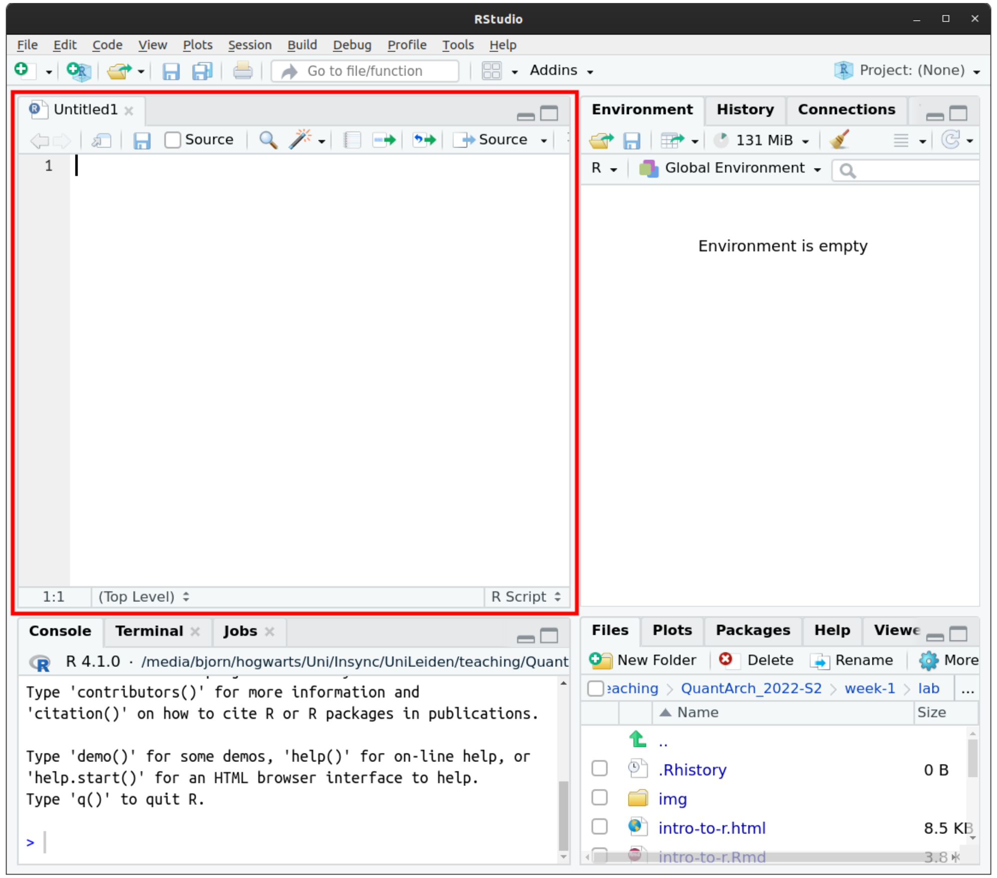
Learning good (enough) practice

The best way to work with R is using **scripts**.

This way you will have a record of everything you have done, which is convenient and reproducible.

To open a new R Script: File > New File > R Script

An empty R Script will show up in the **Source** pane.



```
R 4.1.0 · /media/bjorn/hogwarts/Uni/Insync/UniLeiden/teaching/Quant
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.

>
```

:aching > QuantArch_2022-S2 > week-1 > lab

Name	Size
..	
.Rhistory	0 B
img	
intro-to-r.html	8.5 KB
intro-to-r.Rmd	3.8

Universiteit Leiden

Let's write some code!

For now, we can enter the calculations from earlier:

```
3 + 5
12 / 7
8 / 2 * (2 + 2)
```

You can run a line of code with [Ctrl] + [Enter] on Linux and Windows, and [Cmd] + [Enter] on Mac. The output will show up in the **Console** pane.

```
## [1] 8
```

```
## [1] 1.714286
```

```
## [1] 16
```

You can run the whole script with [Ctrl] + [Alt] + [R] on Linux and Windows, and [Cmd] + [Alt] + [R] on Mac.

You can also find these and others in Code > Run Region > …

Annotation, annotation, annotation...

You can add comments to an R Script with the 'hashtag', `#`. Anything written after this will not be executed.

```r
# Some very basic calculations
3 + 5
```

```
## [1] 8
```

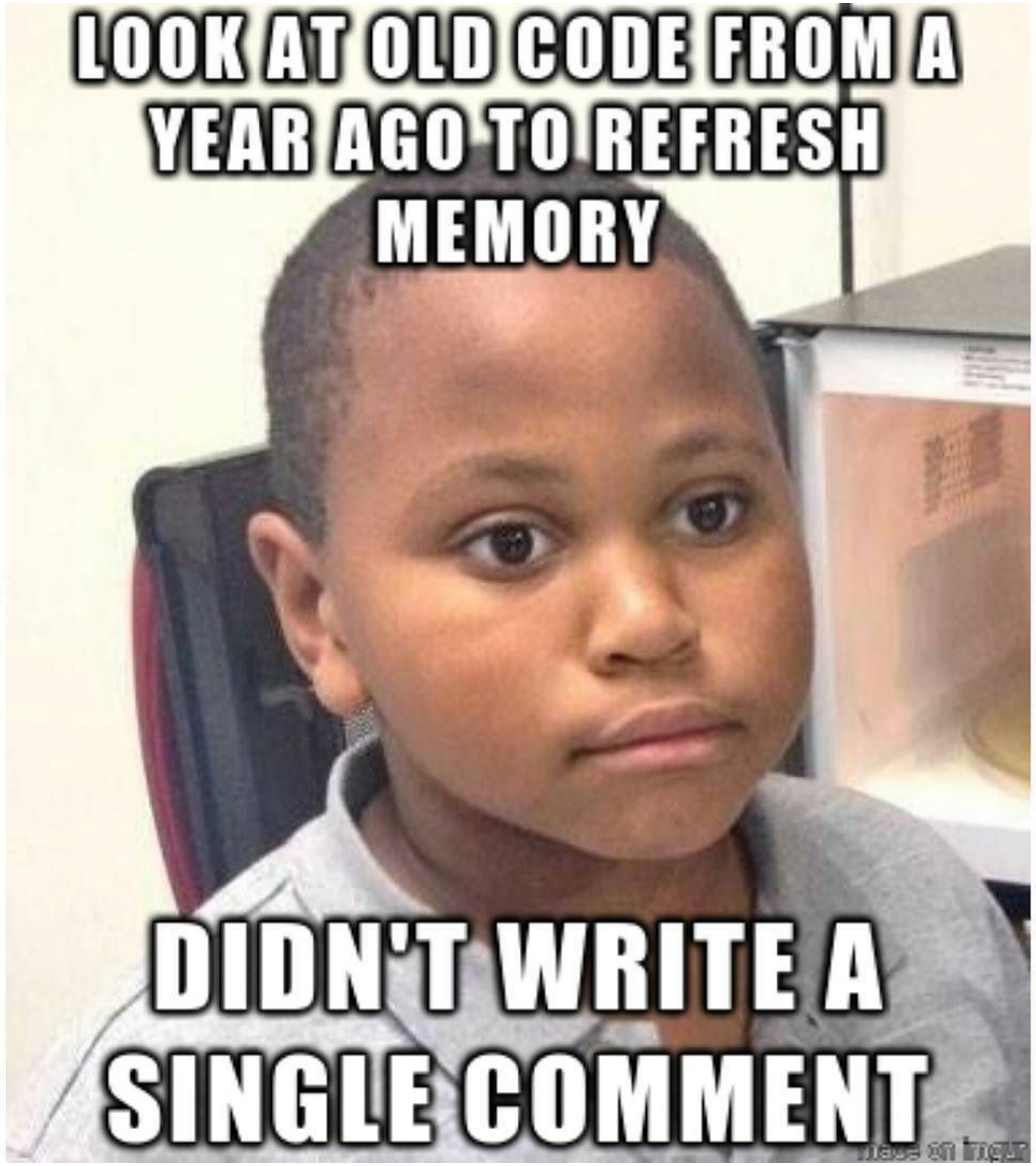
```r
12 / 7 # although this one I can't actually do in my head...
```

```
## [1] 1.714286
```

```r
8 / 2 * (2 + 2)
```

```
## [1] 16
```

This is a good way to let others (and yourself) know what's going on in your script; otherwise…


LOOK AT OLD CODE FROM A YEAR AGO TO REFRESH MEMORY
DIDN'T WRITE A SINGLE COMMENT
made on imgur

(Smooth) Operators

R has the following operators (non-exhaustive):

operator	function
`+`	addition
`-`	subtraction
`*`	multiplication
`/`	division
`^` or `**`	exponentiation/order
`()`	brackets
`%%`	modulus

Storing objects

So far, none of our code output has been stored/saved, i.e. if we wanted to recall the calculation answers, we would have to run the code again.

To store an object in your **Global Environment**, use the assignment operator, `<-`, with the name of the object on the left and the values on the rights.

When naming an object, it is important to consider the following:

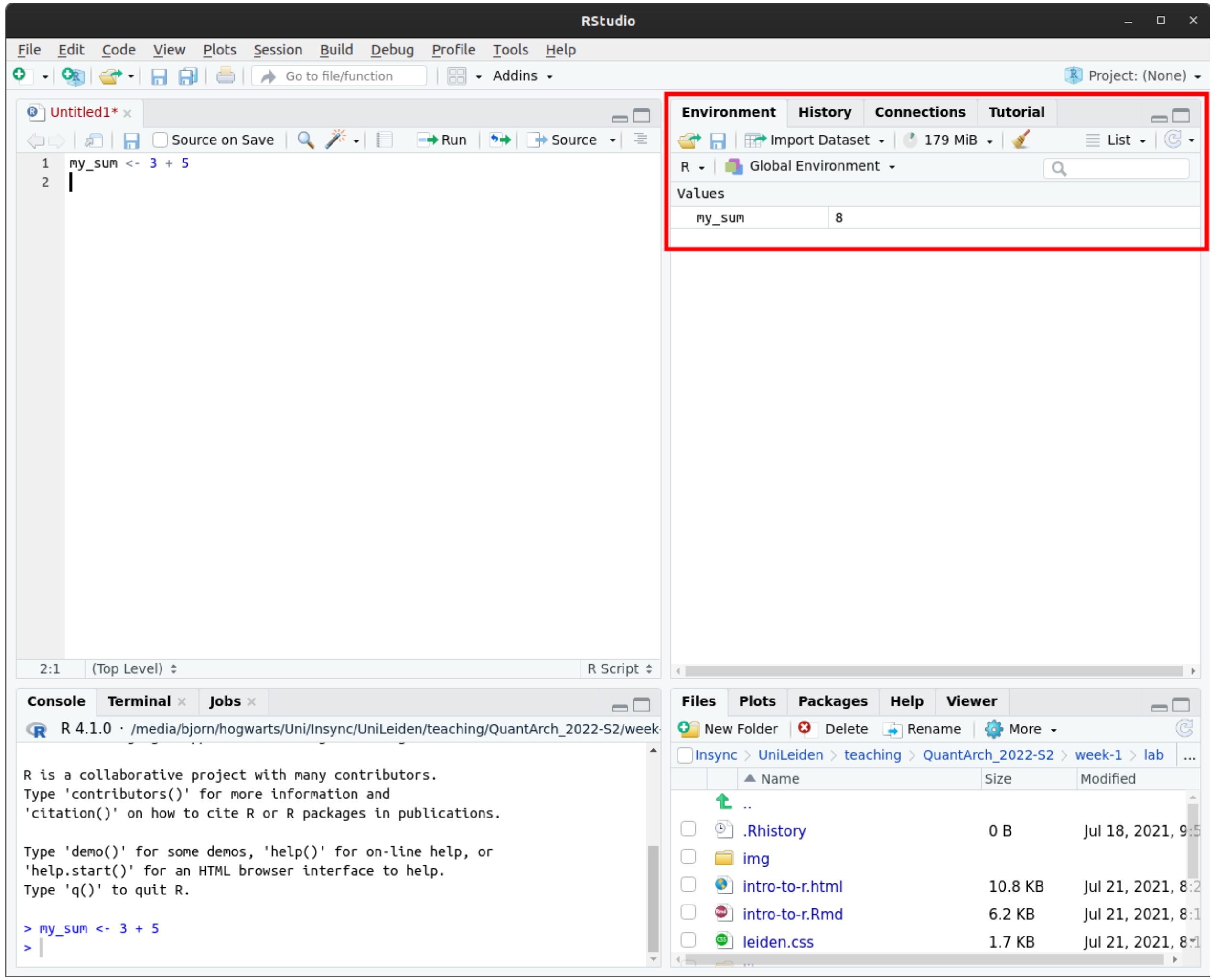
- Must only contain alphanumerics, decimal point `.`, or, underscore `_`
  - must start with a letter
- should make sense to what you are storing
- should NOT interfere with any predefined R objects
  - knowing what these are comes with experience

More storing objects

```
my_sum <- 3 + 5
# call it 'my_sum' because 'sum' is
```

When you store an object, you won't get any output in the console.

Instead, the object will show up in your **Global Environment** pane.



```
my_sum <- 3 + 5
```

Values

my_sum	8

```
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.

> my_sum <- 3 + 5
>
```

Name	Size	Modified
..		
.Rhistory	0 B	Jul 18, 2021, 9:5
img		
intro-to-r.html	10.8 KB	Jul 21, 2021, 8:2
intro-to-r.Rmd	6.2 KB	Jul 21, 2021, 8:1
leiden.css	1.7 KB	Jul 21, 2021, 8:1

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1*
Source on Save
Run
Source
2:1
(Top Level)
R Script
Environment History Connections Tutorial
Import Dataset
179 MiB
List
R
Global Environment
Console Terminal Jobs
R 4.1.0 · /media/bjorn/hogwarts/Uni/Insync/UniLeiden/teaching/QuantArch_2022-S2/week
Files Plots Packages Help Viewer
New Folder Delete Rename More
Insync › UniLeiden › teaching › QuantArch_2022-S2 › week-1 › lab

Universiteit Leiden

You guessed it, more storing objects

You can then call the output by entering the name of the object in the console (or script), or using the `print` function,

```
my_sum
```

```
## [1] 8
```

```
print(my_sum)
```

```
## [1] 8
```

and perform further operations on the object:

```
my_sum * 2
```

```
## [1] 16
```

The real benefit of storing an object is with larger data types and functions.

Final notes on objects

You can change an object by overwriting it with a new value:

```r
my_sum <- my_sum * 2
my_sum # show new value of object
```

```
## [1] 16
```

You can remove objects with the `rm()` function by entering the name of the object:

```r
rm(my_sum)
```

Or remove everything from your environment (which can be useful if you are testing whether or not an R Script can run from scratch):

```r
rm(list = ls()) # ls() is a function that lists everything in your environment
```

Break