

R Studio and R Markdown

Louis Nass

1/29/2021

R and R Studio

- What are *R* and *R Studio*?
 - The difference between an engine (*R*) and the dashboard (*R Studio*)

R: Engine



RStudio: Dashboard



- *R*: the machinery, language that performs computations
- *R Studio*: the pretty stuff, integrated development environment (IDE), neatly organizes your work

Installation

BOTH *R* and *R Studio* are needed in order to run

1. To download *R* follow the instructions here, instructions vary depending on type of computer (Mac, Window, Linux)
2. To download *R Studio* follow the instructions here, again follow instructions for your computer

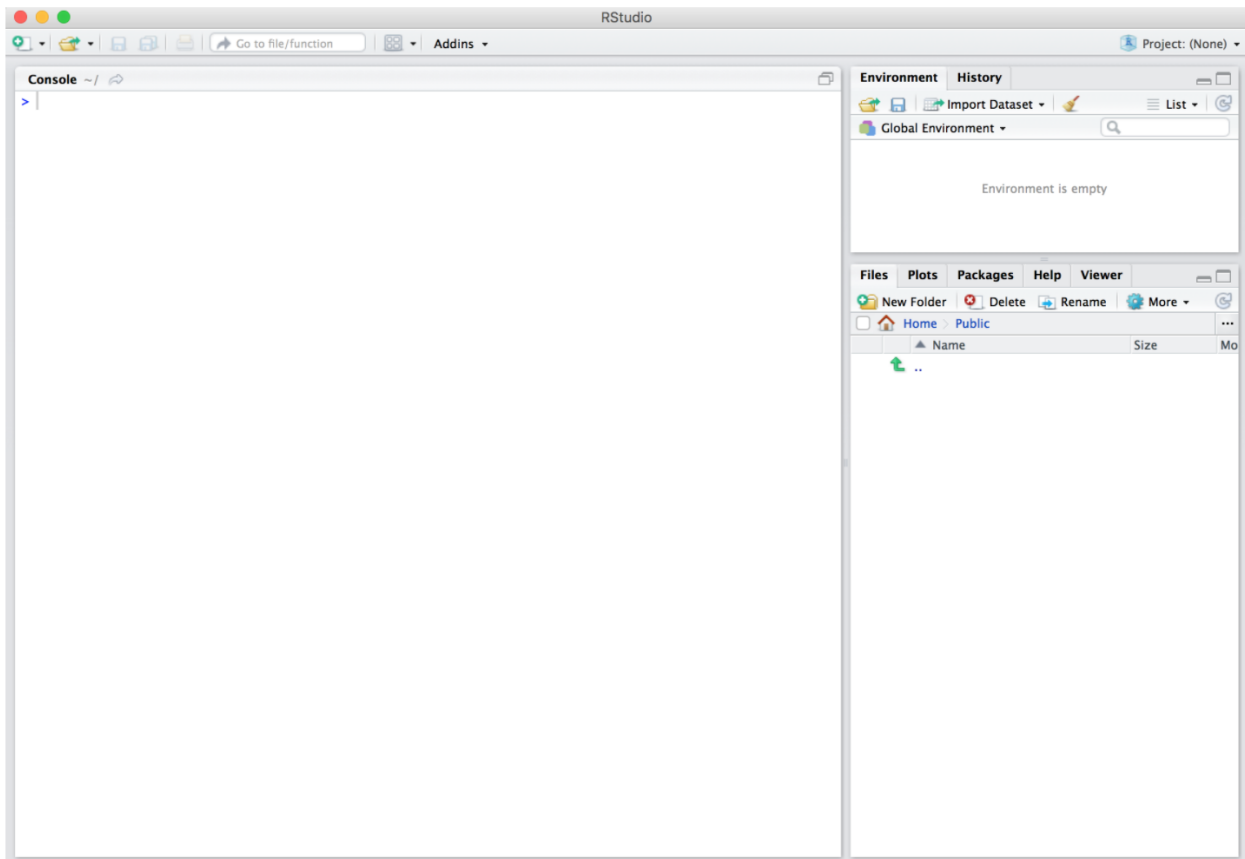
R: Do not open this



RStudio: Open this



Once you click the correct icon, your window will look like this:



We observe 3 different panes, the **console pane** (left), **enviornment pane** (top right), **file pane** (bottom right).

R Basics

- **Console pane:** Enter commands
- **Running code:** Telling *R* what to do
- **Objects:** Values that you have assigned in *R*

```
object <- 1  
object = 3  
object
```

```
## [1] 3
```

- **Data types:** Integers (whole numbers), doubles/numerics (“real numbers”), logicals (T/F), characters (words or phrases) can all be assigned to objects
- **Vectors:** An object that has multiple dimensions

```
vector <- c(1, 2, 3)
```

```
vector
```

```
## [1] 1 2 3
```

- **Factors:** Categorical data, sometimes known as strings
- **Data Frames:** spreadsheets where rows are data and columns are variables
- **Conditionals:** Results in a logical result (T/F)
 - Testing equality, use ‘==’

```
2+1 == 3
```

```
## [1] TRUE
```

```
2+2 == 3
```

```
## [1] FALSE
```

- Boolean algebra with operators like ‘<’ (less than), ‘<=’ (less than or equal to), ‘!=’ (not equal to) will return T/F as above.
- Logical operators to combine terms like ‘&’ (and), ‘|’ (or)
- **Functions:** Also called commands, take inputs and produce outputs. One can write their own functions for R to compute or can use built in functions from R.

```
seq(from = 2, to = 5)
```

```
## [1] 2 3 4 5
```

```
Diff<- function(a,b){  
  x<-b-a  
  x  
}  
Diff(3,4)
```

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

Errors Warnings and Messages

R can communicate back to you. Particularly if you give it something it does not like

3 forms of communication

- **Errors:** When you see red text in *R Studio* that is prefaced with ‘Error in...’, think of this like a red light. Something is wrong and needs to be corrected.

- **Warnings:** When you see red text in *R Studio* that is prefaced with ‘Warning:...’, think of this like a yellow light. Something is not necessarily wrong, but there can be an improvement made to help the code run more smooth.
- **Messages:** When you see red text in *R Studio* that does NOT start with ‘Error...’ or ‘Warning...’, think of this like a green light. Simply *R Studio* talking at you while it is working. Smile and wave back!
 - Particularly when you download packages, some developers have fun little messages for you

Coding rules of thumb

- **Computers are not smart:** They are only as smart as their instructor (you). It is up to you to find any mistakes and fix them
- **Copy, paste, tweak:** It helps to modify existing code that can suit you, especially when learning. If you have access to a function that you want to use, simply copy, paste, and make necessary adjustments to serve your purpose. Coding ‘training wheels’
- **Learn by doing:** Making mistakes is frustrating, but read the *Error* and *Warning* messages. They will provide insight towards solving the problem. Google is also your friend
- **Practice:** Practice makes perfect!

Packages

Think apps on a smartphone. Programs we can install that have a variety of functions!

R: A new phone



R Packages: Apps you can download

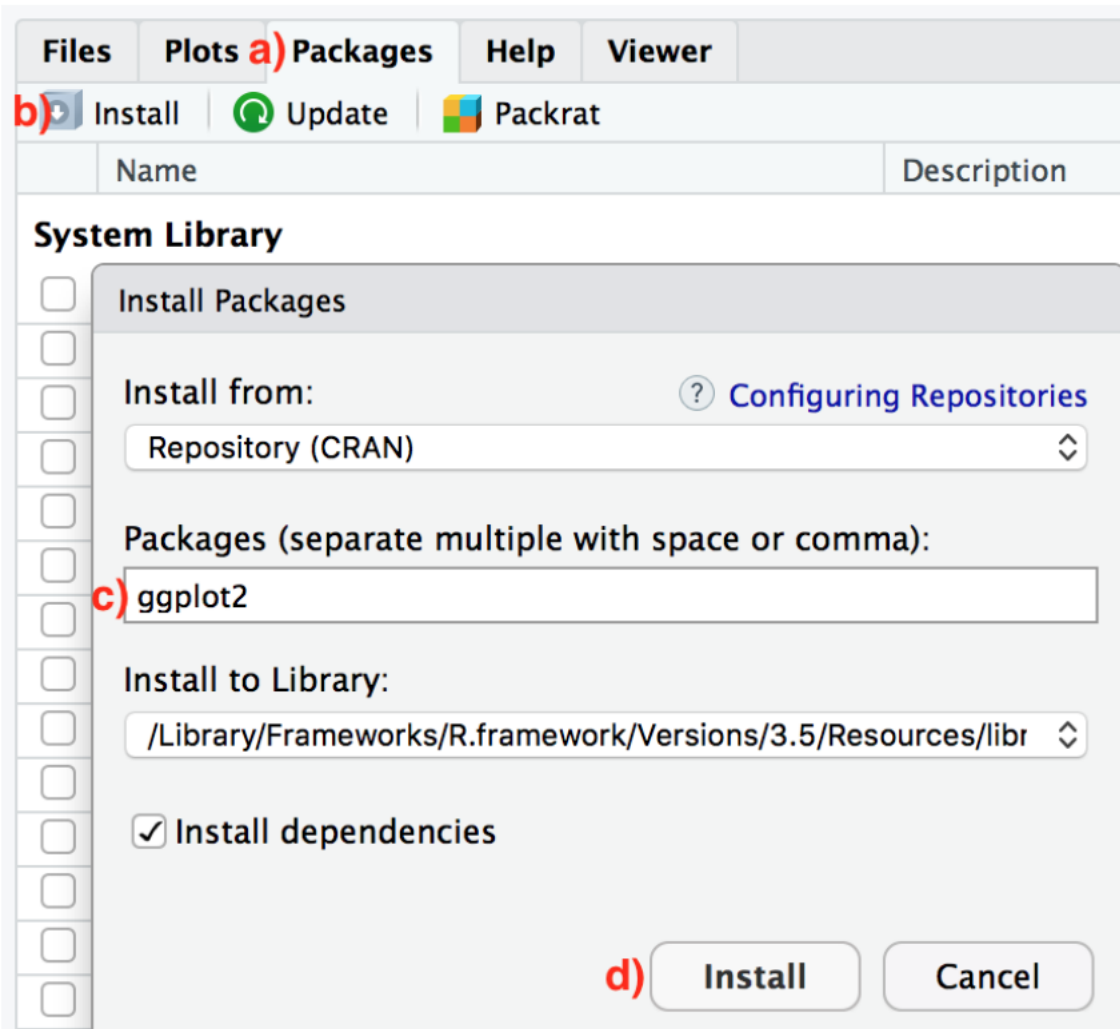


In order to use the app we must first: 1. **Install** the app/package 2. **Load** the app/package

2 ways to install packages:

- ‘Easy way’: using the *files pane* we can quickly install packages
 - a. Click on *Packages* tab in *file pane*

- b. Click on *Install* next to Update
- c. Type name of package under *Packages* (separate multiple with a space or comma)
- d. Click *Install*



- ‘Advanced way’: The true easy way, type in ‘install.packages(“name of package”)’ on your console

```
install.packages("ggplot2", repos="http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
## /var/folders/jl/6gk425js7kv_x6n_kzy1vxmc0000gn/T//RtmpuDLnid/downloaded_packages
```

Now we need to load our package. Use the ‘library()’ command

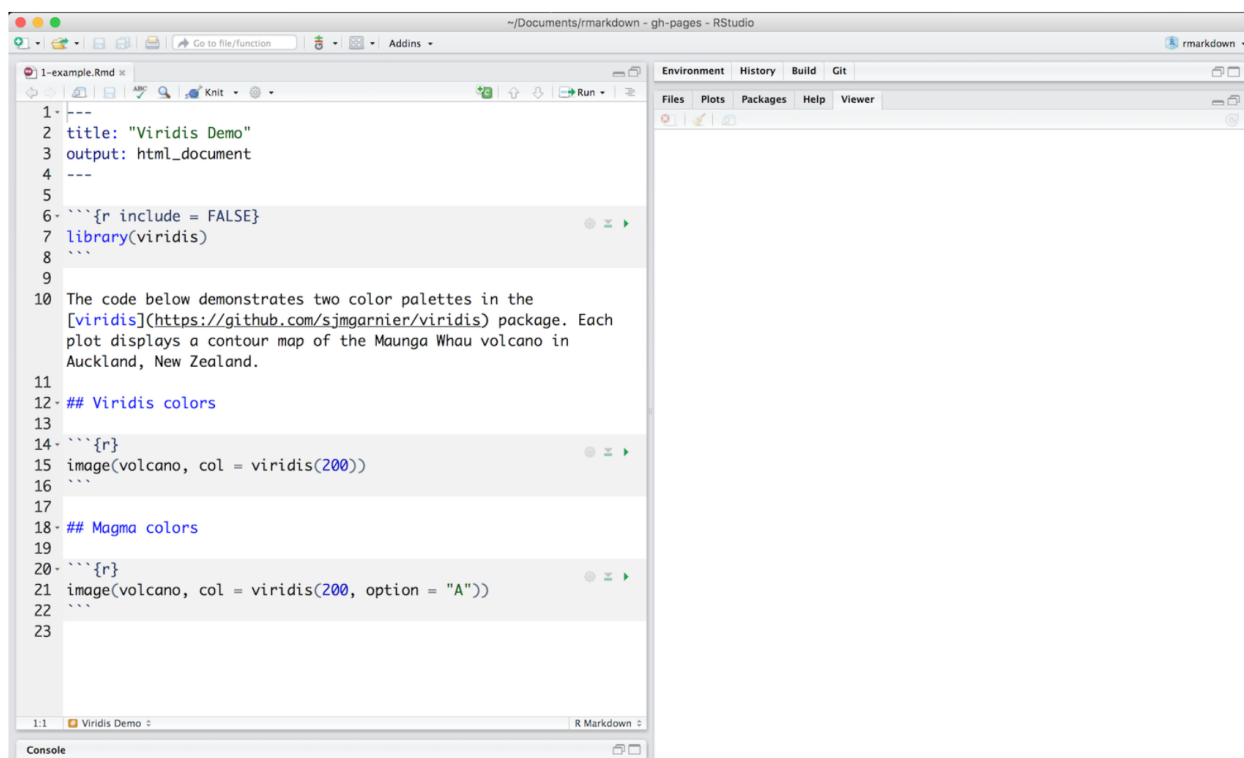
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

Now we have access to the functions in the package ‘ggplot2’!

R Markdown

A type of file you can create using *R Studio*. Allows the user to publish their work in a convenient fashion. Used to make the document you are currently reading!



We can see three types of content

- *Headers* signified using `'—'`s
- *Chunks* signified using `''''`s
- Standard text that is formatted

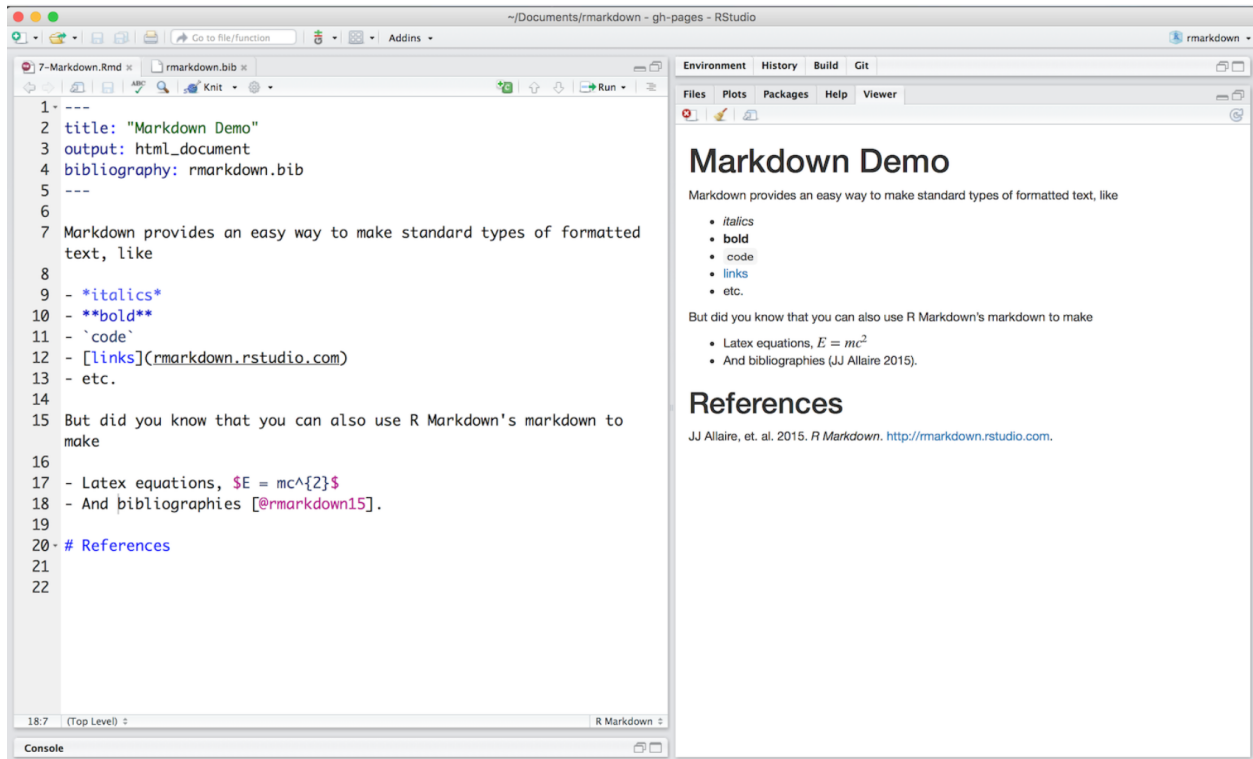
We render our *R Markdown* file using the 'Knit' button. This runs all of the code through our console and creates the desired document (html, pdf, docx, etc...)

Chunks

Lines of code that we want incorporated with our final document are called 'Chunks'. We define them in our document by initializing the chunk with three dashes above below our desired code.

- Chunk options
 - `include = FALSE`: hides code in final document
 - `echo = FALSE`: hides code but not results in final document
 - `message = FALSE`: hides messages from *R* in final document
 - `warning = FALSE`: hides warnings from *R* in final document

Markdown Basics



For a quick reference guide to *R Markdown* check out the R Markdown cheat sheet.