

RStudio

Working with R – RStudio

RStudio is an Integrated Development Environment (IDE) for R it helps you:

- write code - makes suggestions
- view the output of your code, including plots
- find errors
- manage files
- View documentation
-

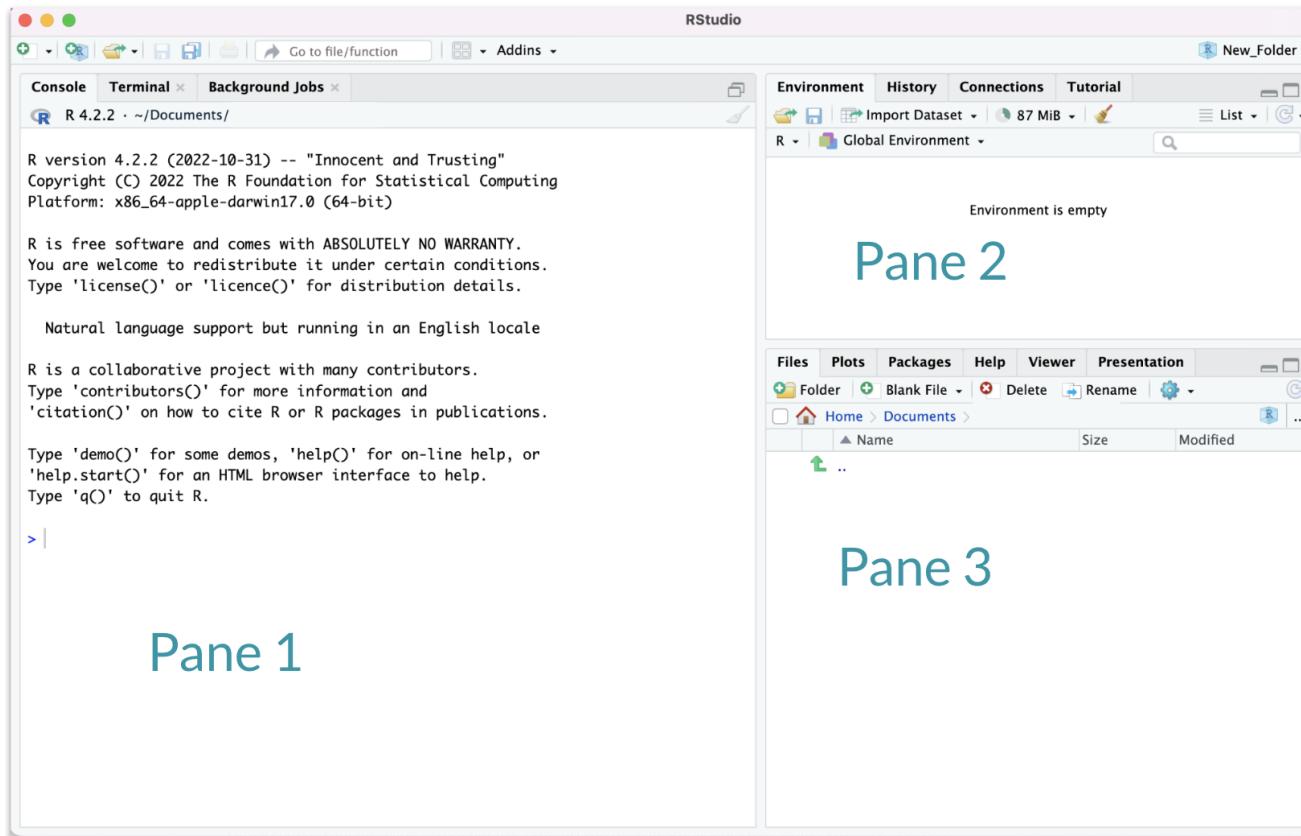


[\[source\]](#)

RStudio used to be the name of a company that is now called [Posit](#).

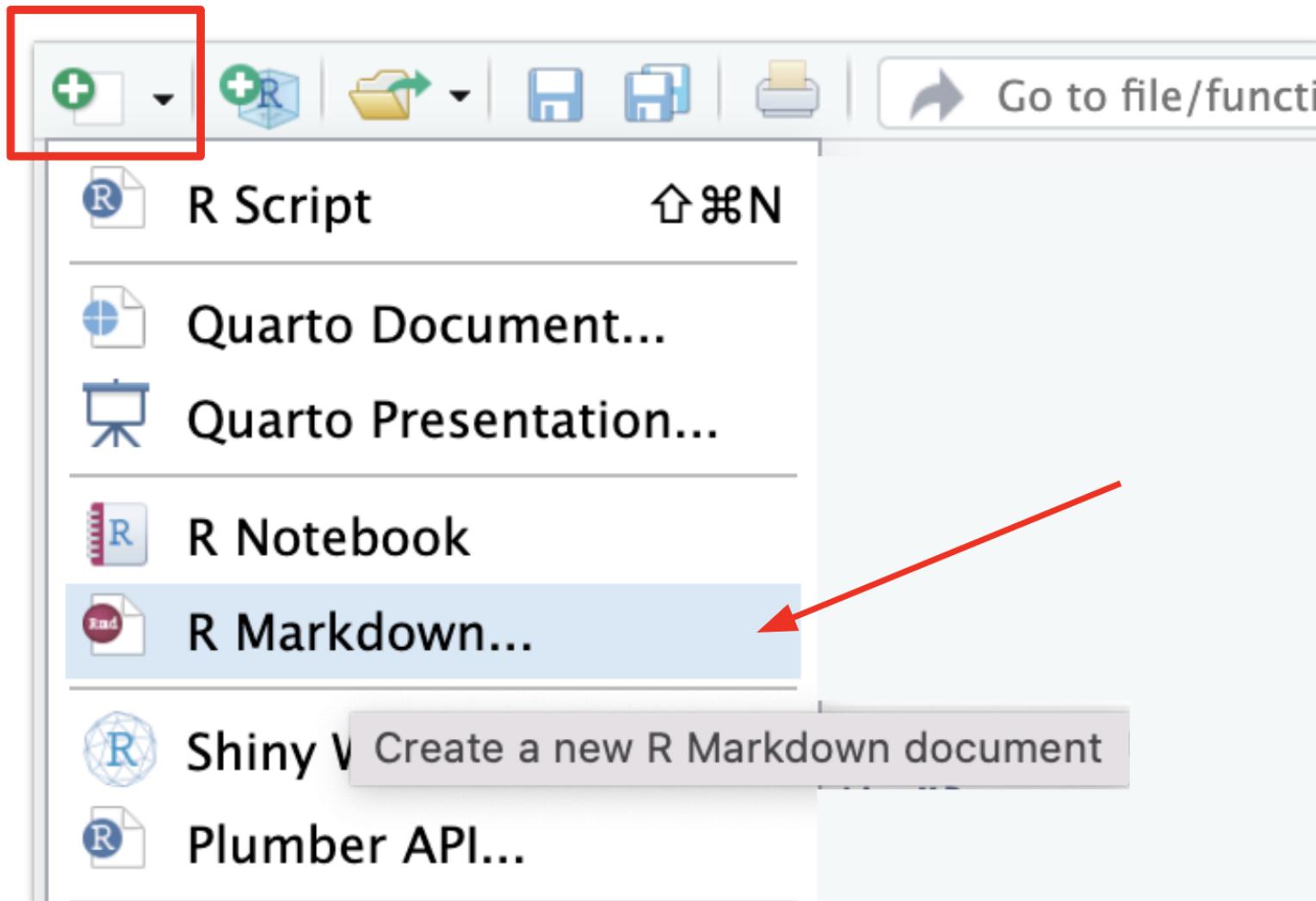
RStudio

First it is important to be familiar with the layout. When you first open RStudio, you will see 3 panes.



Hidden Pane

To save a copy of your code. You must open a file first - this will open a 4th pane. These files include Scripts or what are called R Markdown files.



Hidden Pane

You will see a popup that you can just say "OK" to for now.

Hidden Pane

Nice! now we have a place to save code! This is where we will mostly be working.

The screenshot shows the RStudio interface with three main panes:

- Pane 1 (Squished Down):** The bottom-left pane, which is currently the active one, displays the R console output. It shows the standard R startup message, followed by a note about natural language support, and a reminder that it is a collaborative project.
- Pane 2 (Empty Environment):** The top-right pane, titled "Environment", shows that the environment is currently empty.
- Pane 3 (Hidden Source):** The top-left pane, titled "Source", contains the R Markdown code for a new document. The text "New Pane!" is overlaid on this pane.

Working with R in R Studio - 2 major panes:

1. The **Source/Editor**:

- Static copy of what you did (reproducibility)
- Top by default
- **saves your code**

2. The **R Console**:

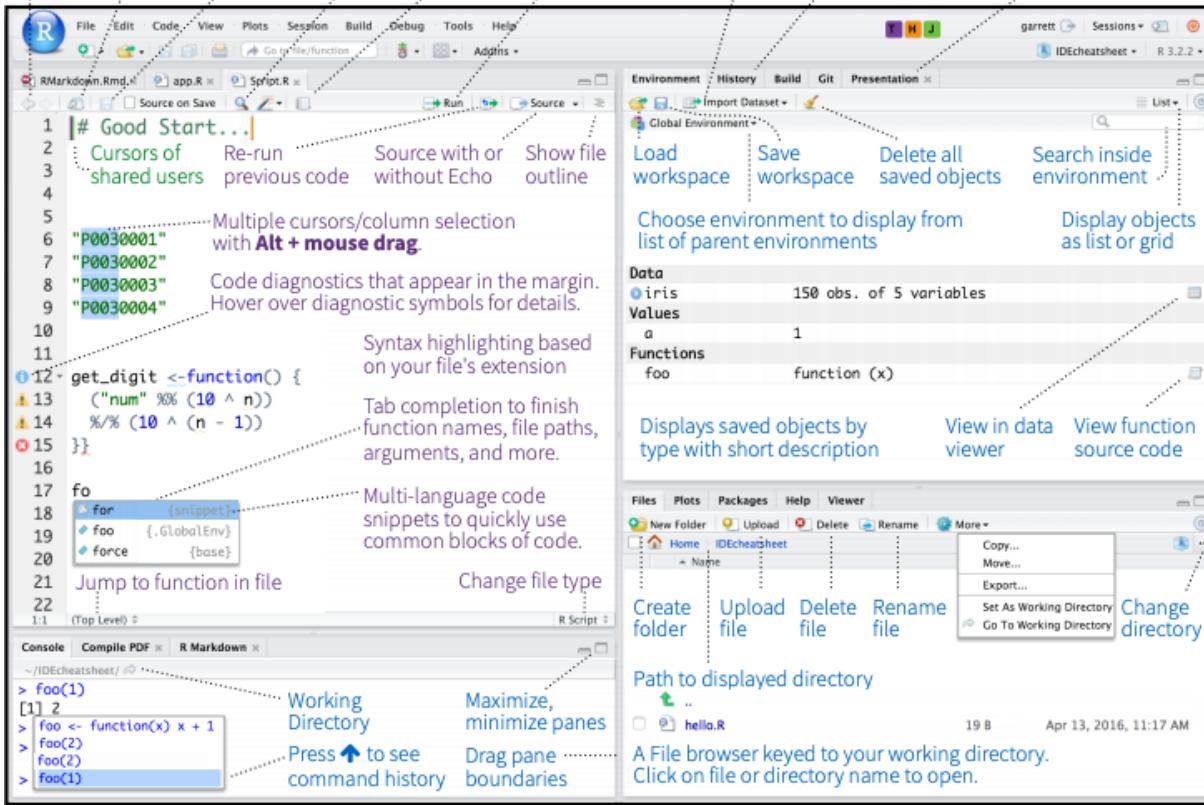
- Calculator
- Try things out interactively, then add to your editor
- Bottom by default
- **doesn't save your code**

RStudio

Super useful “cheatsheet”: [LINK](#)

Write Code

Navigate tabs Open in new window Save Find and replace Compile as notebook Run selected code



R Support

R Markdown files look different from scripts

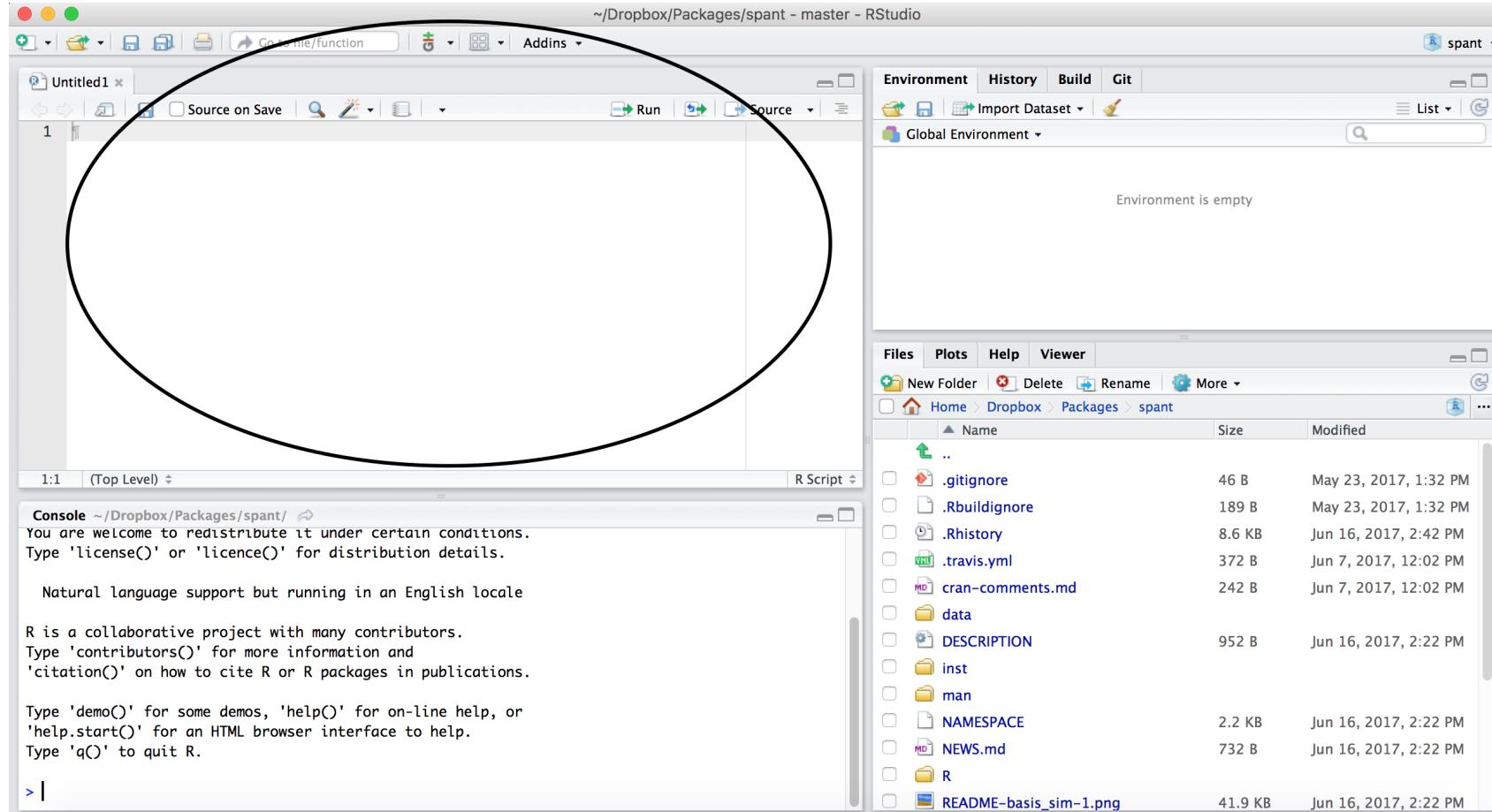
It will look like this with text in it.

The screenshot shows the RStudio interface with the following components:

- Top Bar:** Shows the title "intro_to_r - rstudio - RStudio".
- Left Panel (Code Editor):** Contains the code for "Untitled1". A red box highlights the first 14 lines of the document, which include the YAML front matter and a code chunk. Lines 10 through 14 are part of the R Markdown content.

```
1 ---  
2 title: "first_markdown"  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE}  
7 knitr::opts_chunk$set(echo = TRUE)  
8 ````  
9  
10 ## R Markdown  
11  
12 This is an R Markdown document. Markdown is a simple formatting syntax for  
authoring HTML, PDF, and MS Word documents. For more details on using R  
Markdown see <http://rmarkdown.rstudio.com>.  
13  
14 When you click the **Knit** button a document will be generated that includes  
both content as well as the output of any embedded R code chunks within the  
document. You can embed an R code chunk like this:  
15  
16 ```{r cars}  
2:23 # first_markdown
```
- Right Panel (Environment):** Shows the Global Environment tab with the message "Environment is empty".
- Bottom Panel (Console):** Displays the R startup message and the command "R Markdown".
- File Explorer:** Shows the project structure under "GitHub > Teaching > intro_to_r".

Scripts will just be empty

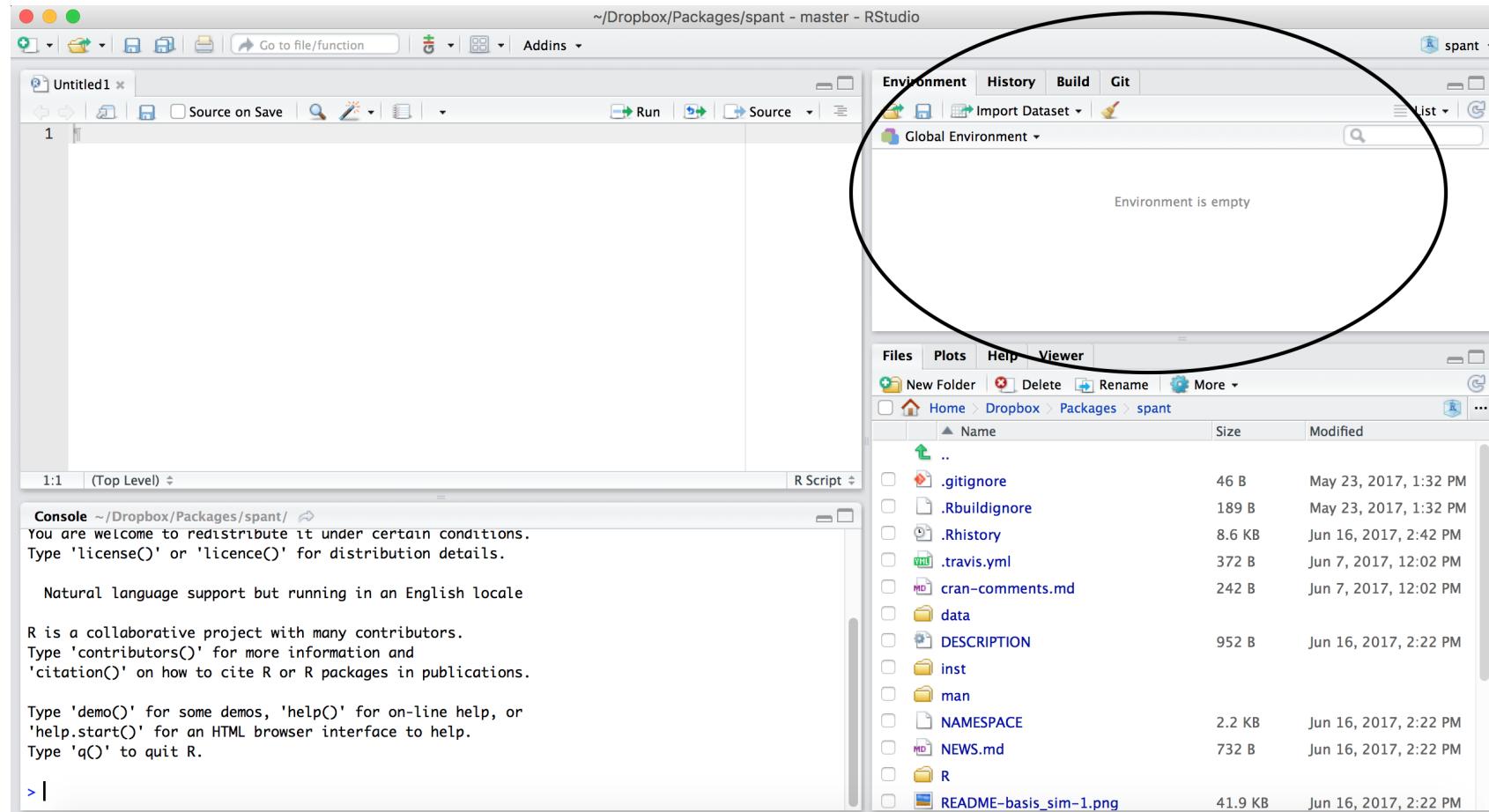


Scripts and R Markdown

Although people will use scripts often, and they are good for more programmatic purposes, we generally don't recommend them for Public Health Researchers.

For data analyses, R Markdown files are generally superior because they allow you to check your code and write more info about your code.

Workspace/Environment



Workspace/Environment

- Tells you what **objects** are in R
- What exists in memory/what is loaded?/what did I read in?

History

- Shows previous commands. Good to look at for debugging, but **don't rely** on it.
Instead use RMarkdown!
- Also type the “up” key in the Console to scroll through previous commands

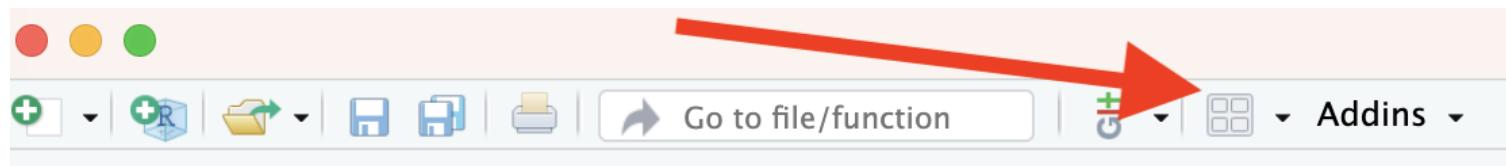
Lower right pane

- **Files** - shows the files on your computer or the directory you are working in
- **Help** - shows help of R commands
- **Plots** - pictures and figures

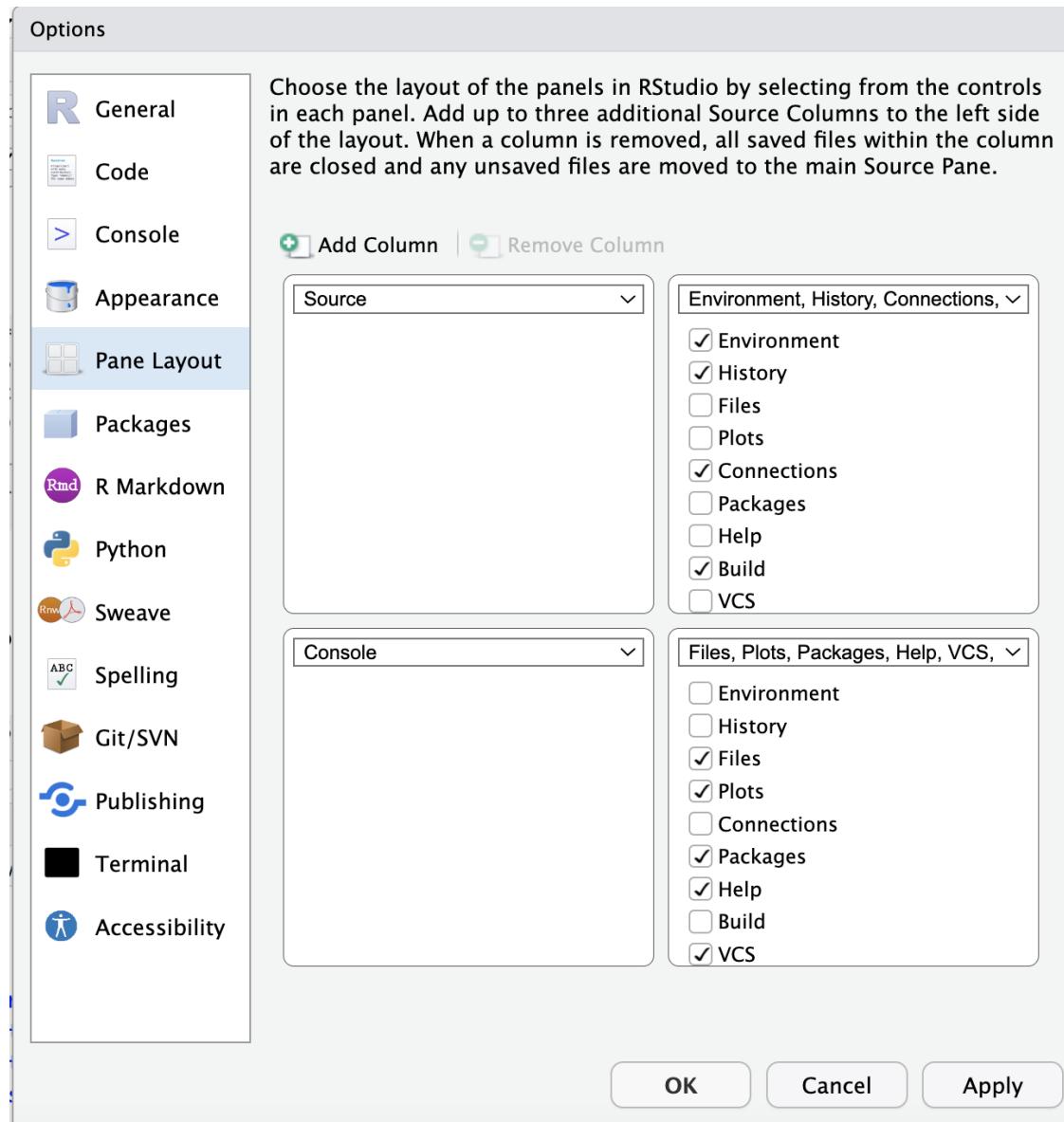
RStudio Layout

If RStudio doesn't look the way you want (or like our RStudio), then:

Click on the pane button, which looks like a waffle with 4 indentations. Scroll down to "Pane Layout".



Default Layout



Let's take a look at R Studio
ourselves!

R Markdown file

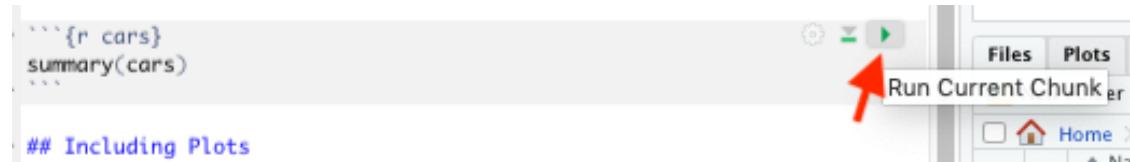
R Markdown files (.Rmd) help generate reports that include your code and output.

1. Helps you describe your code
2. Allows you to check the output
3. Can create many different file types

Code chunks

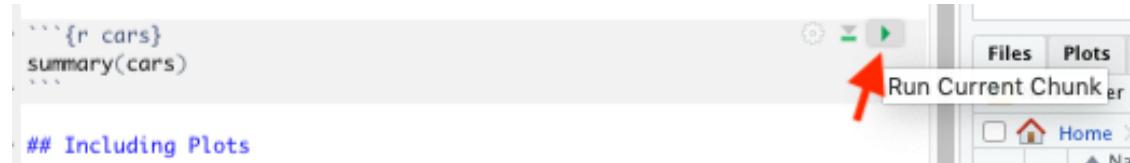
Within R Markdown files are code “chunks”.

This is where you can type R code and run it!



Run code in a chunk

Clicking the run (play) button runs the code in the chunk.



Ctrl + Enter on Windows or Command + Enter on Mac in your script evaluates that line of code

Running a chunk executes the code

- generally see a preview of the output of the code just below the chunk
- see the code in the console

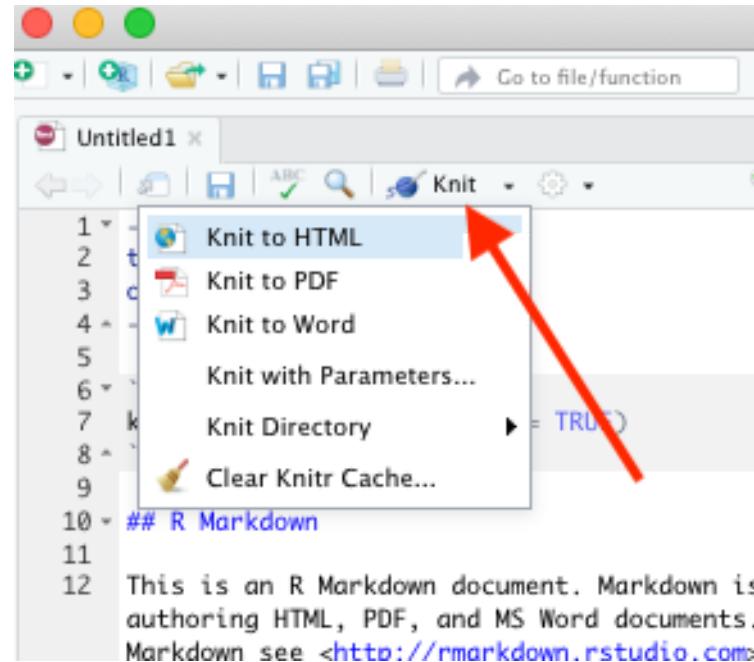
If you get annoyed by code previews in Markdown files...

See the [Help page](#) of the website. You can adjust this and change your RStudio settings:

Tools > Global Options > R Markdown

Knit file to html

Running all chunks - this will create a report from the R Markdown document! Note that it can't use anything not included in the file, it can't use objects in your environment that you were modifying interactively.



Nice report!

This generates a nice report that you can share with others who can open in any browser.

The screenshot shows a window from RStudio displaying an R Markdown document. The title bar reads "Untitled.html" and the path "~/Documents/Roger/New_Folder/Untitled.html". The toolbar includes standard Mac OS X controls (red, yellow, green buttons) and menu items like "Open in Browser" and "Find". On the right, there's a "Publish" button. The main content area contains:

Untitled

Your Name
2023-03-29

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
## Min.   : 4.0   Min.   : 2.00
## 1st Qu.:12.0   1st Qu.: 26.00
## Median :15.0   Median : 36.00
## Mean   :15.4   Mean   : 42.98
## 3rd Qu.:19.0   3rd Qu.: 56.00
## Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:

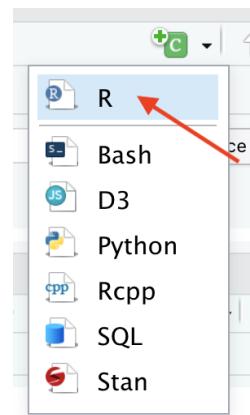
Create Chunks

To create a new R code chunk:

- Use the insert code chunk button at the top of RStudio.



- Select R (default) as the language:



Create Chunks

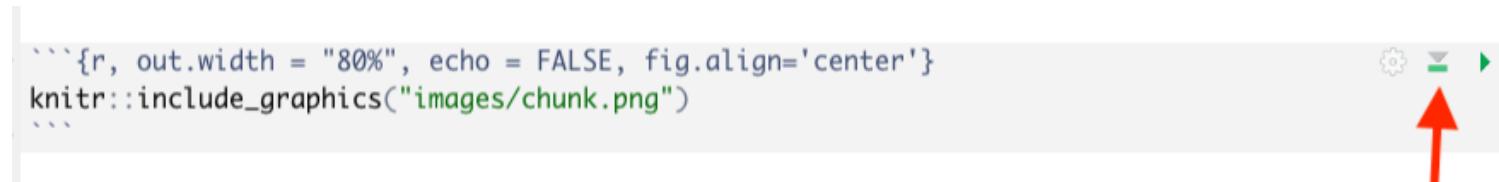
If you like keyboard shortcuts:

- Windows & Linux use Ctrl+Alt+I
- Mac use Command+Option+I

I is for insert.

Run previous chunks button

You can run all chunks above a specific chunk using this button:



Errors

R studio can help you find issues in your code. Note that sometimes the error occurs earlier than RStudio thinks.

The screenshot shows a code editor in RStudio with the following code:

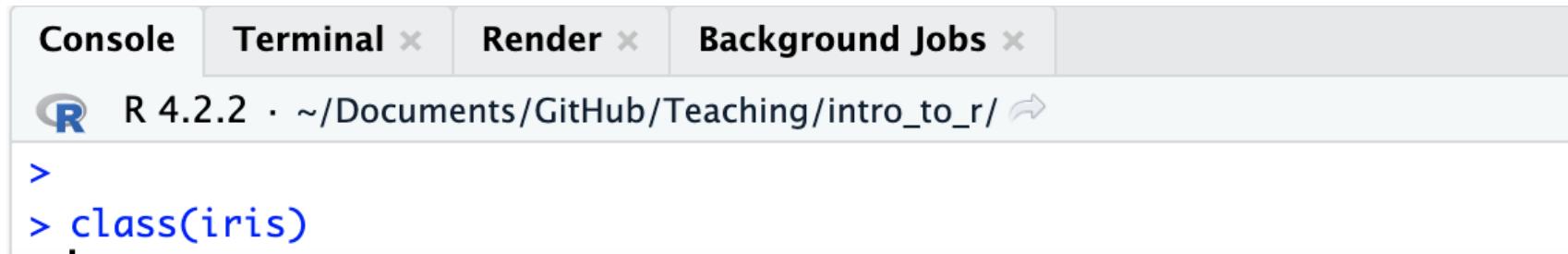
```
```
265
266 ````{r}
267 ✖ class(er))
268 ````
```

A red error icon is positioned next to the line number 267. A tooltip or message box is displayed below the code area, containing the error message:

Error: unexpected ')' in "class(er))"

## Recap of where code goes

- you can test code in the console



A screenshot of the RStudio interface. At the top, there are four tabs: "Console" (which is selected and highlighted in grey), "Terminal x", "Render x", and "Background Jobs x". Below the tabs, the main area shows an R session. It starts with the R logo icon and the text "R 4.2.2 · ~/Documents/GitHub/Teaching/intro\_to\_r/". Then, there are two blue-highlighted lines of code: "> class(iris)" and "> .".

- you can save code in a chunk in the editor (Markdown file)

```
R Markdown
```

Code does not go here and instead goes within the grey chunks like this:

```
```{r}
summary(cars)
```
```



# Getting help from the preview

When you type in a function name, a pop up will preview documentation to help you. It also helps you remember the name of the function if you don't remember all of it!

The screenshot shows two examples of RStudio's code completion and documentation preview feature.

**Top Example:** The user has typed the beginning of a function name, and a dropdown menu lists several options. The option "class" is highlighted. A tooltip window is open over the "class" entry, displaying the following information:

```
class(x)
Object Classes
R possesses a simple generic function mechanism which can be used for an object-oriented style of programming. Method dispatch takes place based on the class of the first argument to the generic function.
```

**Bottom Example:** Similar to the top example, the user has typed "read\_". A dropdown menu lists several functions starting with "read\_". The function "read\_csv" is highlighted. A tooltip window is open over "read\_csv", displaying its usage:

```
read_csv(file, col_names = TRUE, col_types = NULL,
 col_select = NULL, id = NULL, locale =
 default_locale(), na = c("", "NA"), quoted_na =
 TRUE, quote = "\"", comment = "", trim_ws = TRUE,
 skip = 0, n_max = Inf, guess_max = min(1000,
 n_max), name_repair = "unique", num_threads =
 readr_threads(), progress = show_progress(),
 show_col_types = should_show_types(),
```

Both examples include a "Press F1 for additional help" link at the bottom of the tooltip window.

# Get help with the help pane

# Getting Help with ?

If you know the name of a package or function:

Type `?package_name` or `?function_name` in the console to get information about packages and functions.

For example: `?readr` or `?read_csv`.

The screenshot shows the RStudio interface. On the left, the Console tab is active, displaying the R command `?class`. On the right, the Help pane is open, showing the documentation for the `Object Classes` in R. The title is `Object Classes`, and the description explains R's generic function mechanism. Below the description, the `Usage` section lists several functions: `class(x)`, `class(x) <- value`, `unclass(x)`, `inherits(x, what, which = FALSE)`, and `isa(x, what)`.

# Double Question Mark

If you haven't loaded a package yet into R than you may get a response that there is no documentation.

Typing in `??package_name` can show you packages that you haven't loaded yet.

The screenshot shows the RStudio interface. On the left, the R console window displays the following session history:

```
>
>
>
>
>
>
> ?class
> ?tidyverse
No documentation for 'tidyverse' in specified packages and libraries:
you could try '??tidyverse'
> ??tidyverse
> library(tidyverse)
— Attaching packages ——————— tidyverse 1.3.2 —
✓ ggplot2 3.4.0 ✓ dplyr 1.0.10
✓ tibble 3.1.8 ✓ stringr 1.5.0
✓ tidyr 1.2.0 ✓ forcats 0.5.1
✓ purrr 1.0.0
— Conflicts ——————— tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag() masks stats::lag()
> ?tidyverse
>
```

On the right, the help viewer window is open for the `tidyverse` package. The title is "tidyverse: Easily Install and Load the 'Tidyverse'". The description states: "The 'tidyverse' is a set of packages that work in harmony because they share common data representations and 'API' design. This package is designed to make it easy to install and load multiple 'tidyverse' packages in a single step. Learn more about the 'tidyverse' at <https://www.tidyverse.org>". The maintainer is listed as Hadley Wickham ([hadley@rstudio.com](mailto:hadley@rstudio.com)). A "tidyverse" logo, which is a dark hexagon with numerous small colored dots, is displayed. The "R Documentation" tab is selected at the top of the help viewer.

## Gut Check

Why are R Markdown files so useful?

1. They let you test your code
2. They let you view the output of your code
3. They let you generate cool reports
4. All of the above

## Gut Check

Where does code go typically in an Rmd file?

A

```
```{r}
```



B

```
...
```

C

Gut Check

Which button do you click to run the code in a current chunk?

```
```{r}  
library(tidyverse)
```
```



Lab: Getting started

To do this lab we need to:

- Download the file at the link above by clicking on it or go to the website schedule page
- Find the downloaded file on your computer
- Open the file in RStudio (double clicking the file name typically works, otherwise right click)
- Might need to restart RStudio

These videos can help if you aren't sure where your downloads are:

If you have a PC: <https://youtu.be/we6vwB7DsNU>

If you have a Mac: <https://www.youtube.com/watch?v=Ao9e0cDzMrE>

You can find these on the resource page of the class website.

Summary

- RStudio makes working in R easier
- the Editor (top) is for static code like scripts or R Markdown documents
- The console is for testing code (bottom) - best to save your code though!
- R markdown documents are really helpful for lots of reasons!
- R code goes within what is called a chunk (the gray box with a green play button)
- Code chunks can be modified so that they show differently in reports

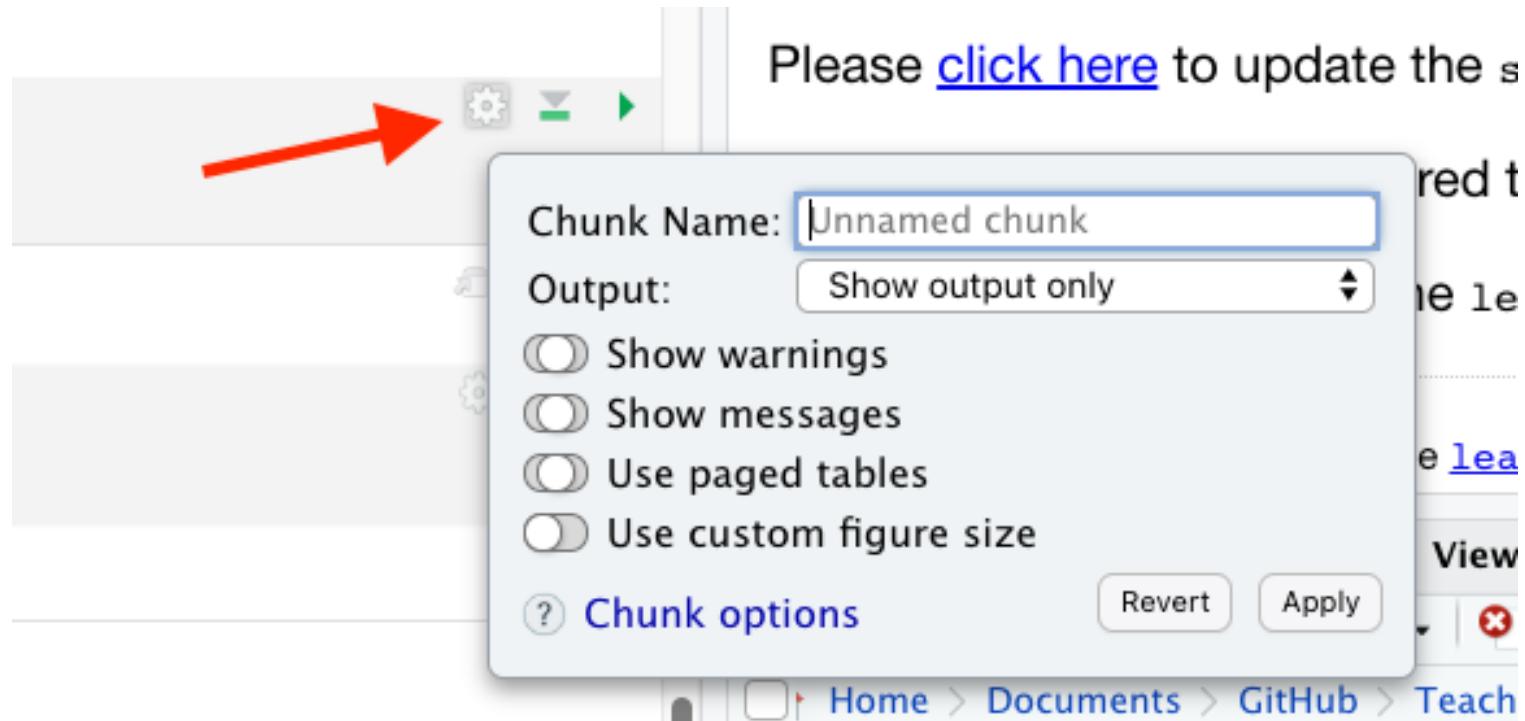
□ [Class Website.](#) □ [Lab.](#) □ [Posit Cheatsheet.](#) □ [Day 1 Cheatsheet.](#)



Image by [Gerd Altmann from Pixabay](#)

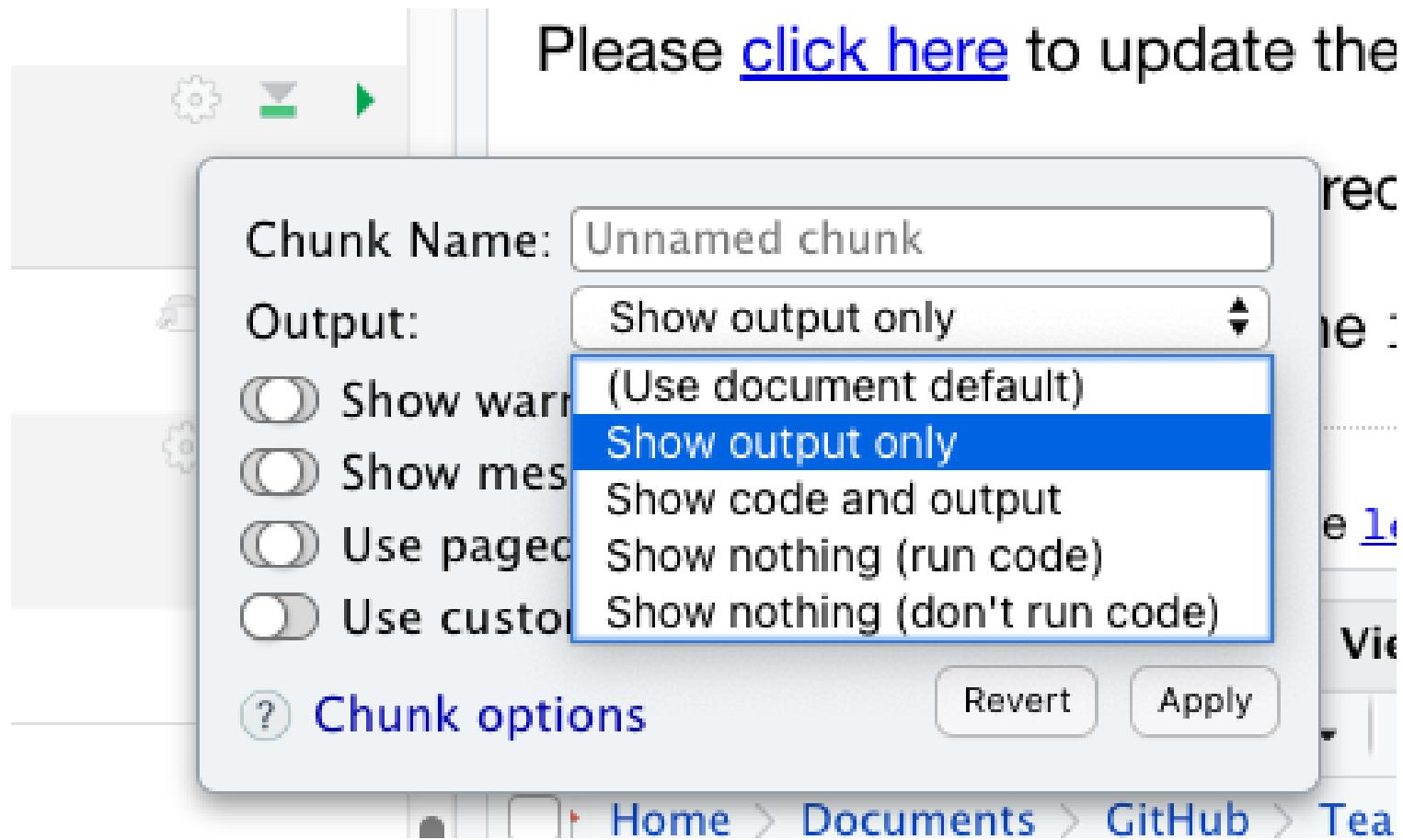
Extra Slides

Chunk settings



Chunk settings

You can specify if a chunk will be seen in the report or not.



Sometimes you want to hide your code

If you want to keep your code so people can see it if they want to there is a nice option called code folding - check it out here:

<https://stackoverflow.com/questions/69326576/show-output-but-hide-code-when-sending-rmd-to-other-people>

Rainbow Parentheses

Tools -> Global Options -> Code -> Display -> Use rainbow parentheses

This can help you see your code more easily.

Press enter to save this setting and get out of this menu.

(((((({{{{{{{{ Enjoy your colorful code! }}}}}}}}})))))