

# Intro to R

RStudio

## Help! Office hours

Office hours will always be held at the *same Zoom link*.

# Working with R – RStudio

RStudio is an Integrated Development Environment (IDE) for R

- Helps you write code - makes suggestions
- Helps you view the output of your code
- Helps you find errors
- Is NOT a dropdown statistical tool (such as Stata)
  - See [Rcmdr](#) or [Radiant](#)



[[source](#)]

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

# RStudio

## Easier working with R

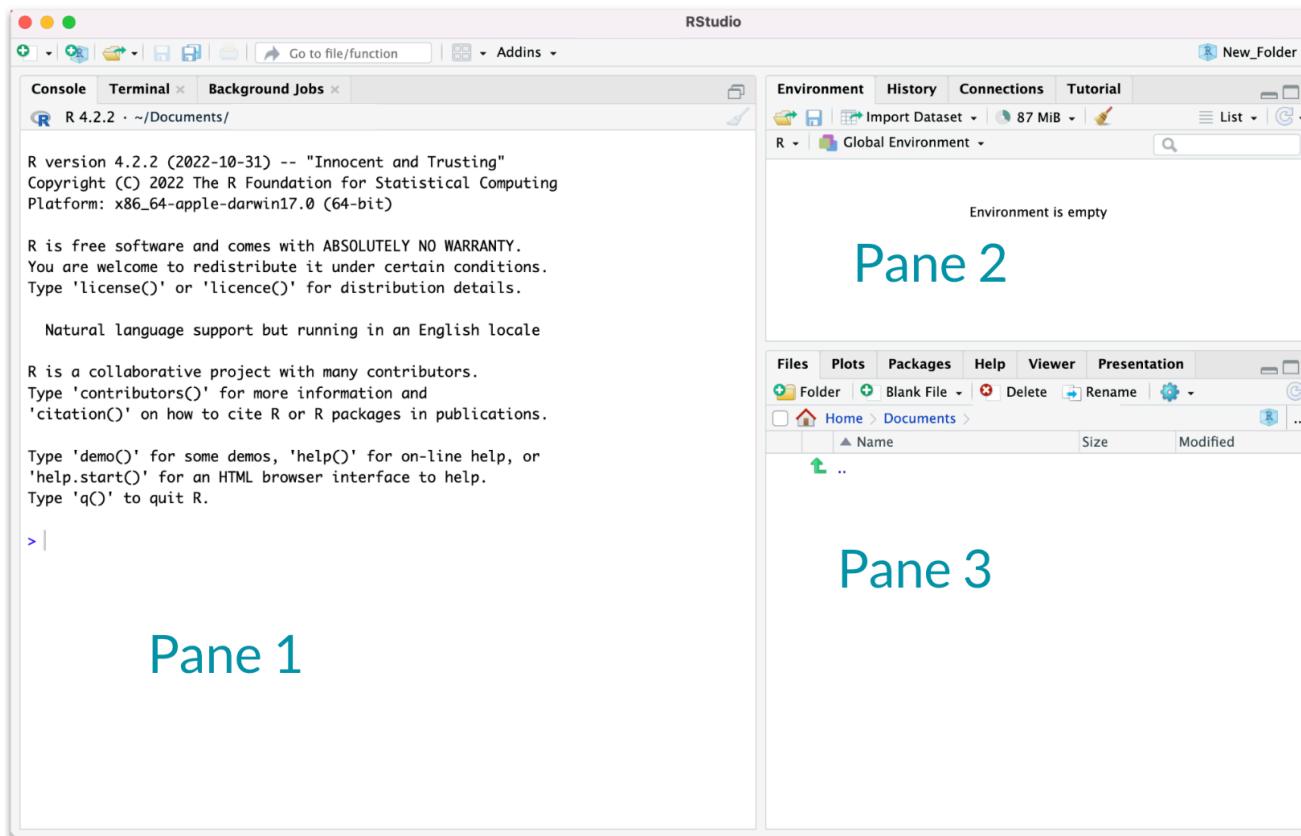
- Syntax highlighting, code completion, and smart indentation
- Easily manage multiple working directories and projects

## More information

- Workspace browser and data viewer
- Plot history, zooming, and flexible image and file export
- Integrated R help and documentation

# RStudio

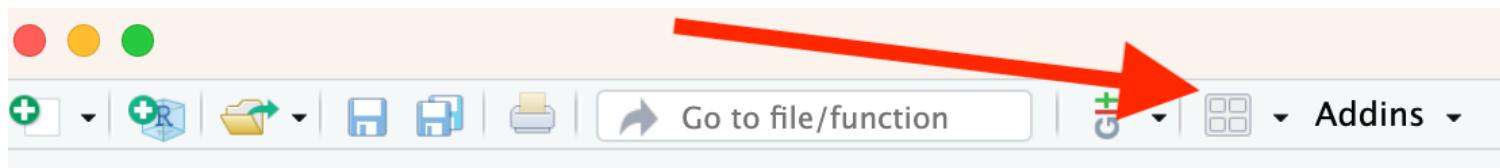
First it is important to be familiar with the layout. When you first open RStudio, you will see 3 panes.



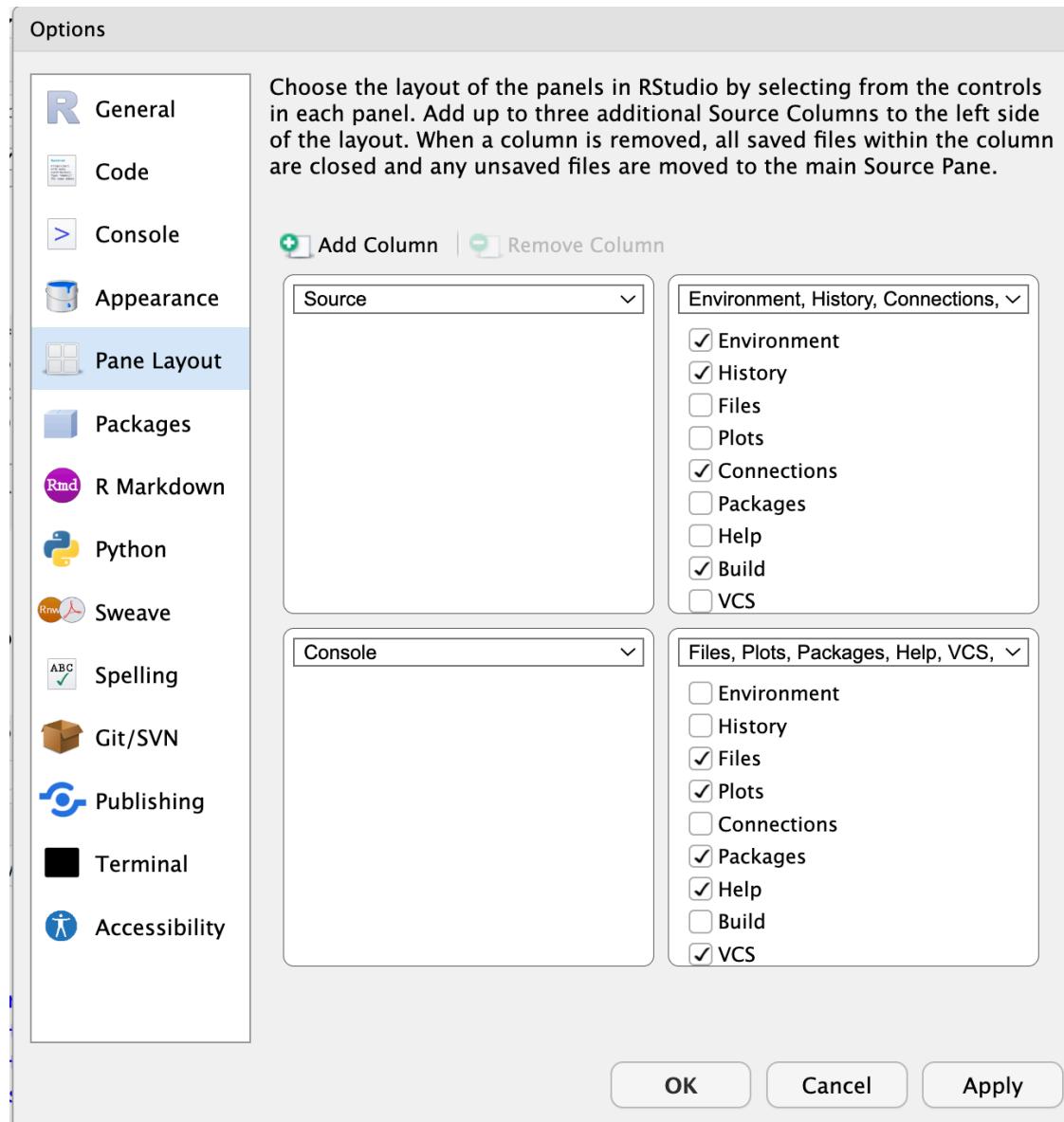
## RStudio Layout

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

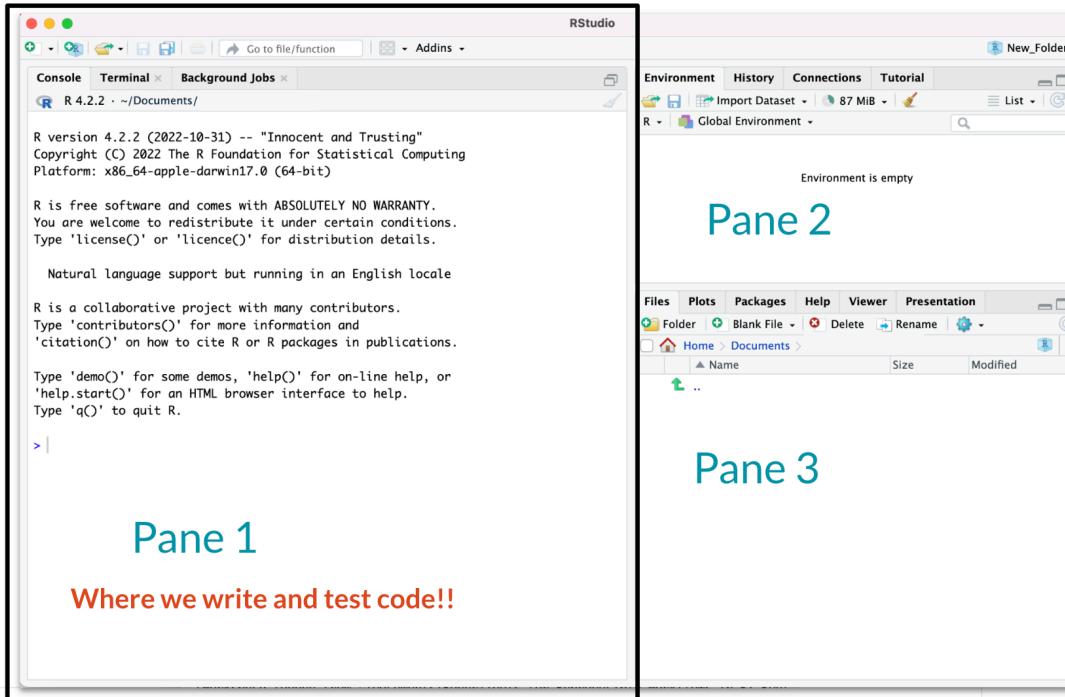
Click on the pane button, which looks like a waffle with 4 indentations. Scroll down to "Pane Layout".



# Default Layout

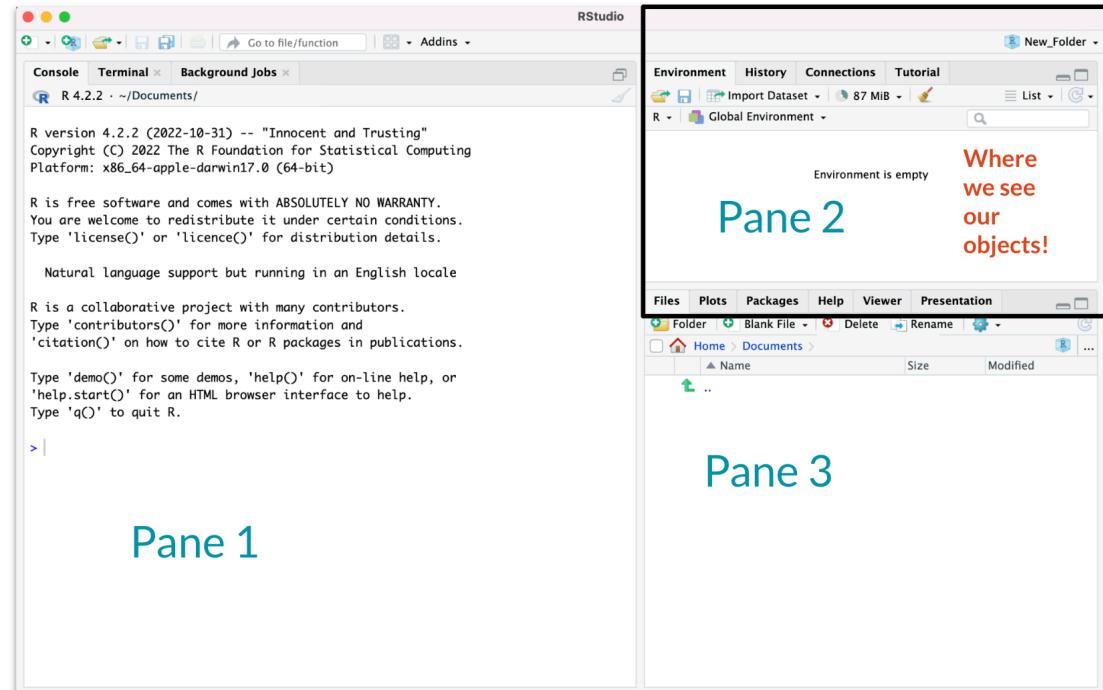


# Pane 1 (Left side) for writing code

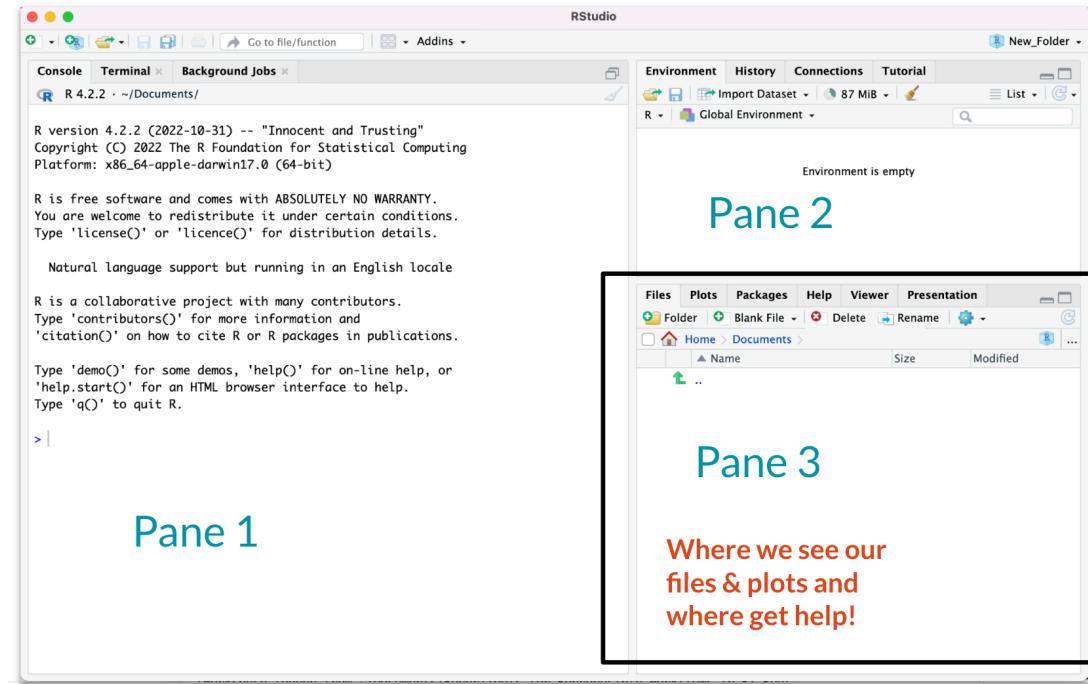


# Pane 2 - where objects will be

We will start seeing this tomorrow!

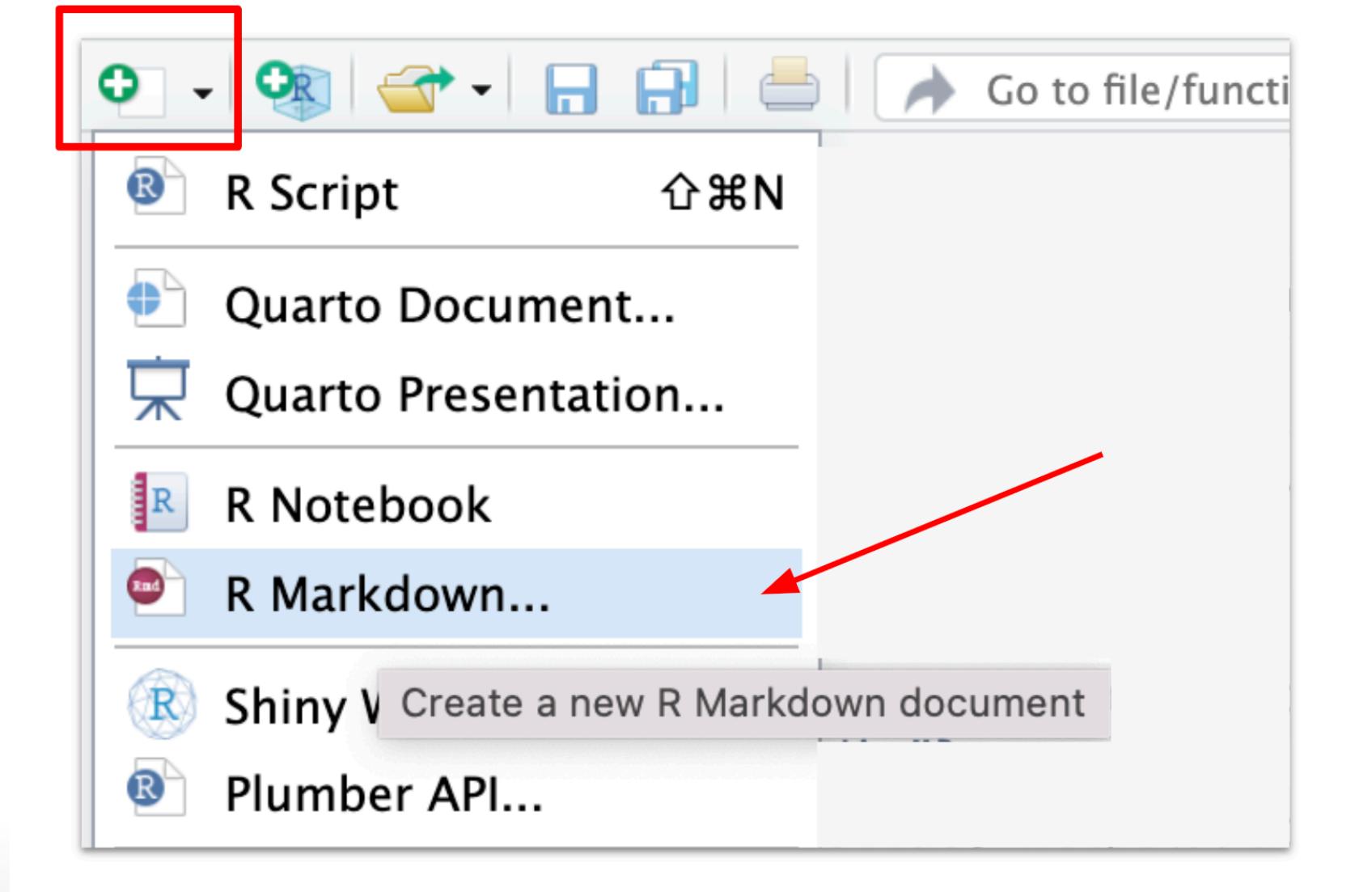


# Pane 3 - where we get help and see plots



## 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.

New R Markdown

Document

Presentation

Shiny

From Template

Title: Untitled

Author:

Date:

Use current date when rendering document

**Default Output Format:**

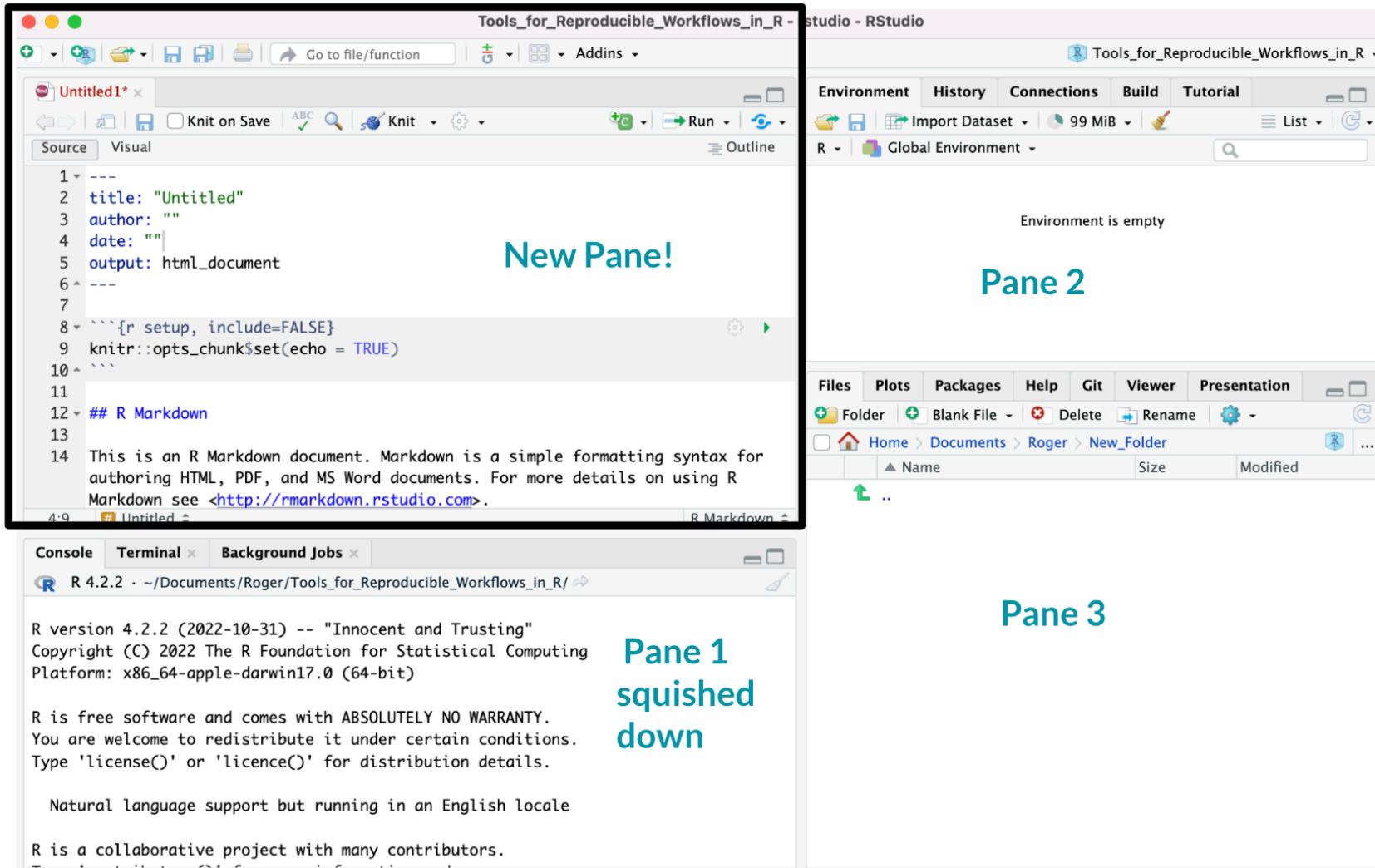
HTML  
Recommended format for authoring (you can switch to PDF or Word output anytime).

PDF  
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

Word  
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

# Hidden Pane

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



## Working with R in R Studio - 2 major panes:

1. The **Source/Editor**: "Analysis" Script + Interactive Exploration

- Static copy of what you did (reproducibility)
- Top by default

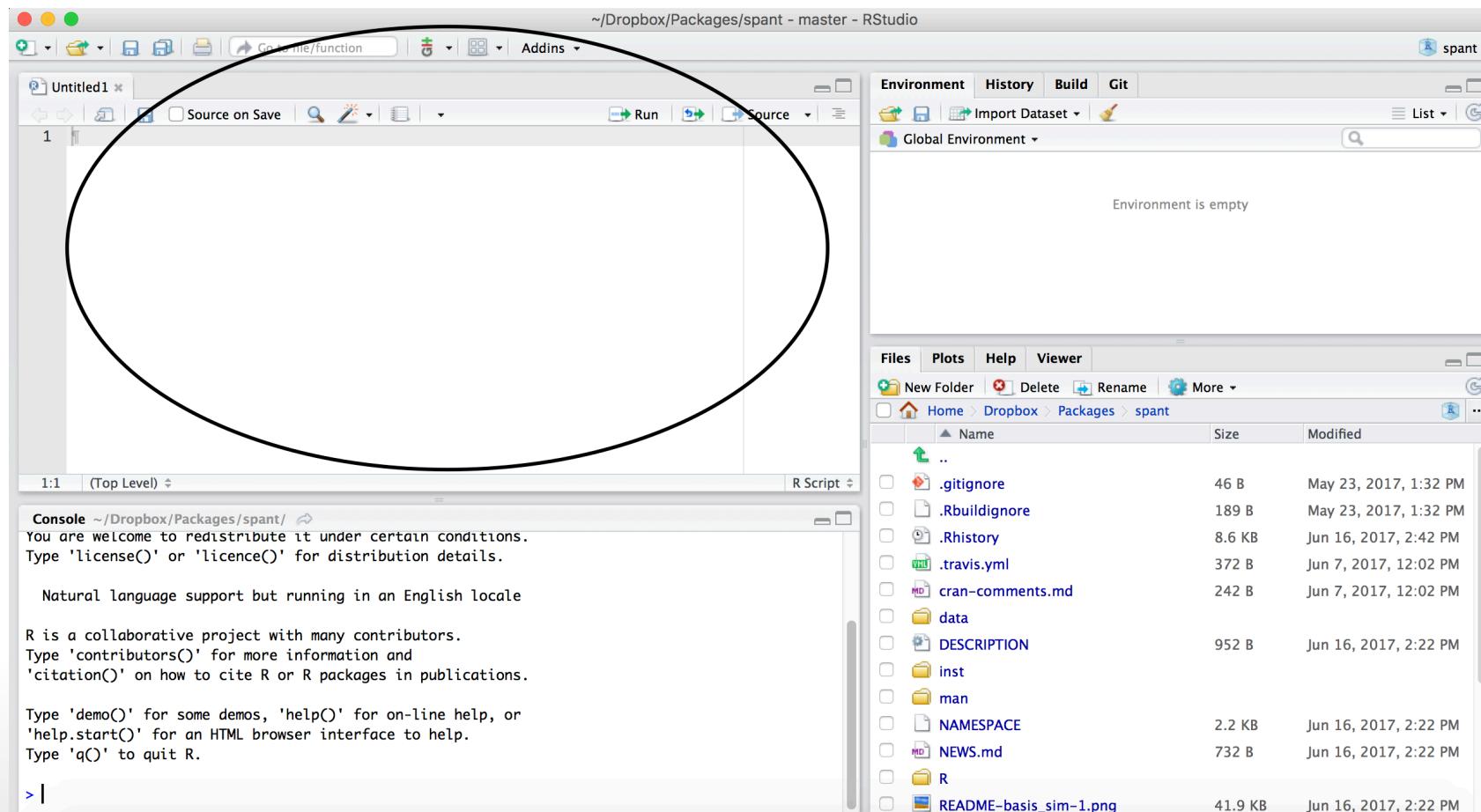
2. The **R Console**: "interprets" whatever you type

- Calculator
- Try things out interactively, then add to your editor
- Bottom by default

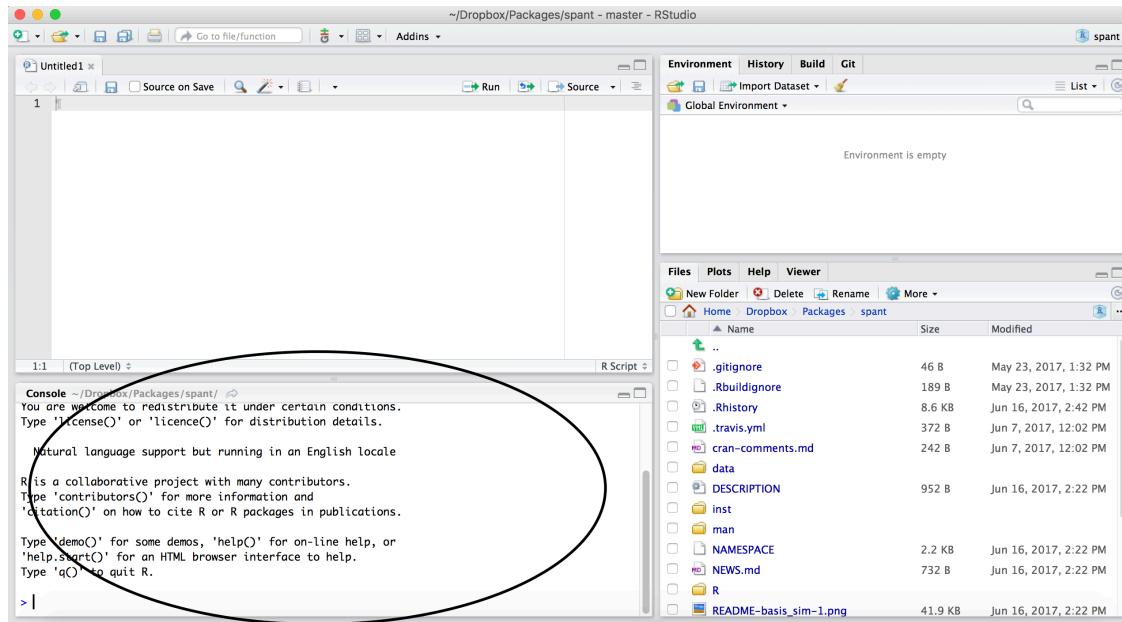
# Source / Editor

- Where files open to
- Have R code and comments in them
- Can highlight and press (CMD+Enter (Mac) or Ctrl+Enter (Windows)) to run the code

In a .R file (we call a script), code is saved on your disk



# R Console



- Where code is executed (where things happen)
- You can type here for things interactively to test code
- Code is **not saved** on your disk

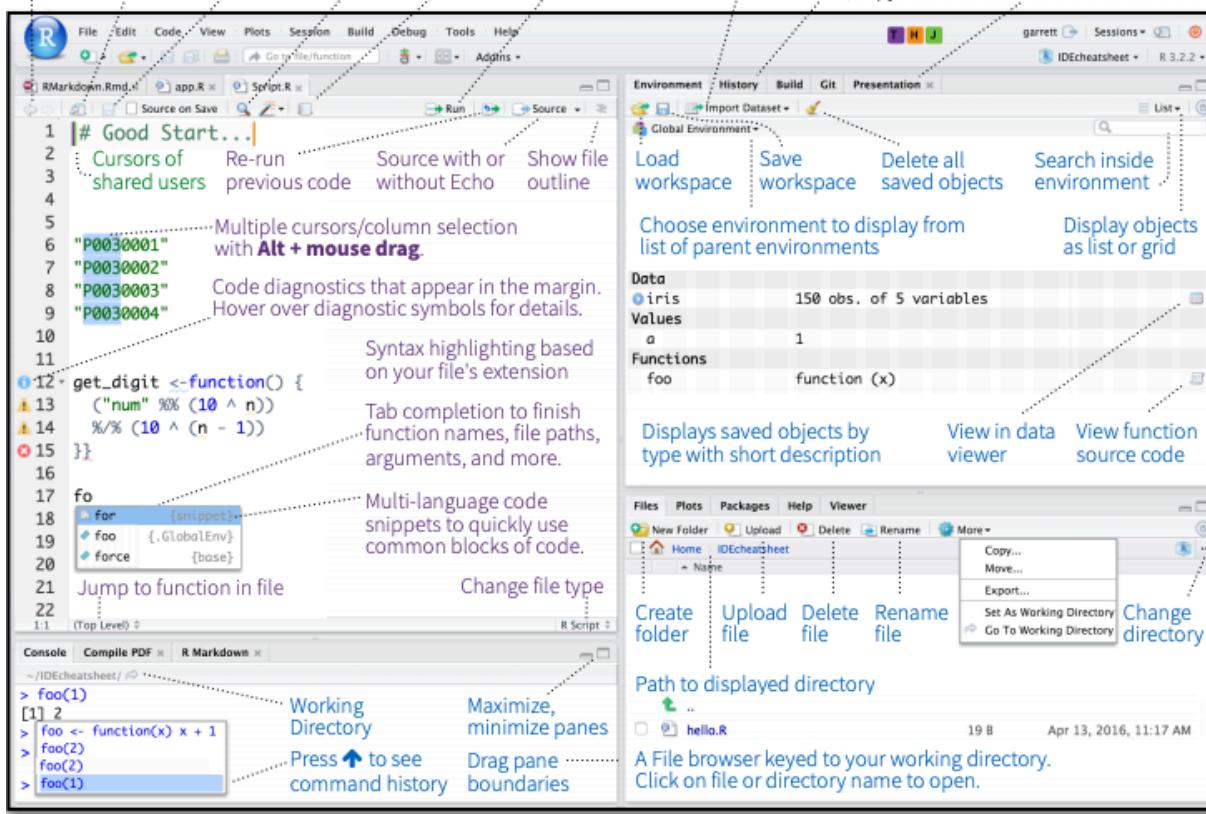
# RStudio

Super useful “cheat sheet”:

<https://raw.githubusercontent.com/rstudio/cheatsheets/main/rstudio-ide.pdf>

## 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, unlike a script.

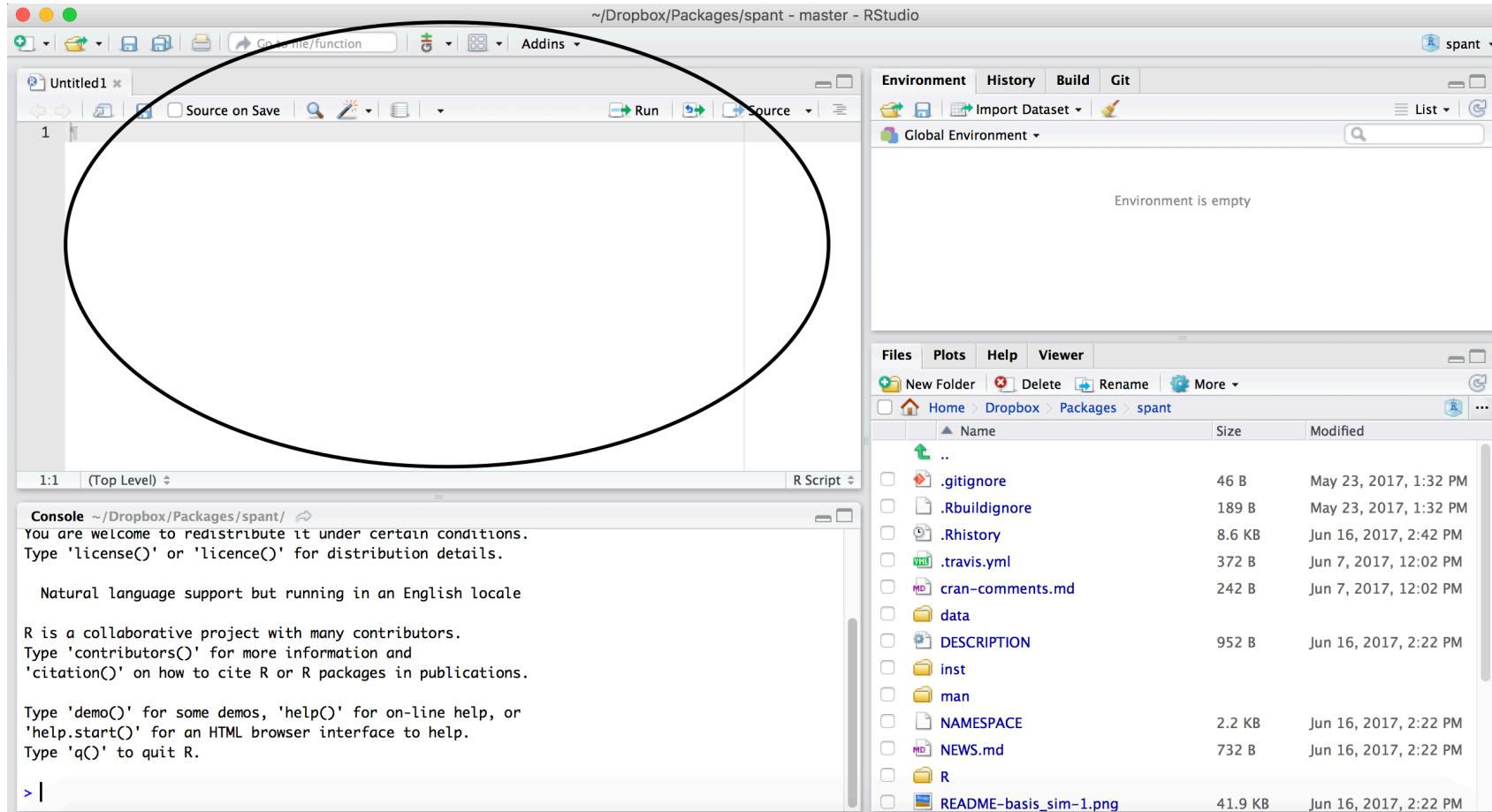
The screenshot shows the RStudio interface with a red box highlighting the code editor area. The code editor contains the following R Markdown code:

```
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
```

The R Markdown code is highlighted in blue. The RStudio environment pane shows the Global Environment is empty. The file browser pane shows the project structure:

Name	Size	Modified
.gitignore	245 B	May 18, 2021, 11:53 AM
.Rbuildignore	16 B	May 18, 2021, 11:53 AM
.Rhistory	43 B	Jun 10, 2021, 11:53 AM
.travis.yml	666 B	Jun 9, 2021, 11:53 AM
all_functions.xlsx	13.4 KB	Jun 8, 2021, 3:11 PM
all_the_functions.csv	57.3 KB	Jun 8, 2021, 3:11 PM
all_the_packages.txt	211 B	May 18, 2021, 11:53 AM
Arrays_Split		
Basic_R		
Best_Model_Coefficients.csv	587 B	May 18, 2021, 11:53 AM
Best_Model_Coefficients.xlsx	3.8 KB	May 18, 2021, 11:53 AM
bibliography.bib	599 B	May 18, 2021, 11:53 AM
black_and_white_theme.pdf	45.1 KB	May 18, 2021, 11:53 AM
bloomberg logo small horizontal	25.4 KB	May 18, 2021, 11:53 AM

# Recall that a script was just empty

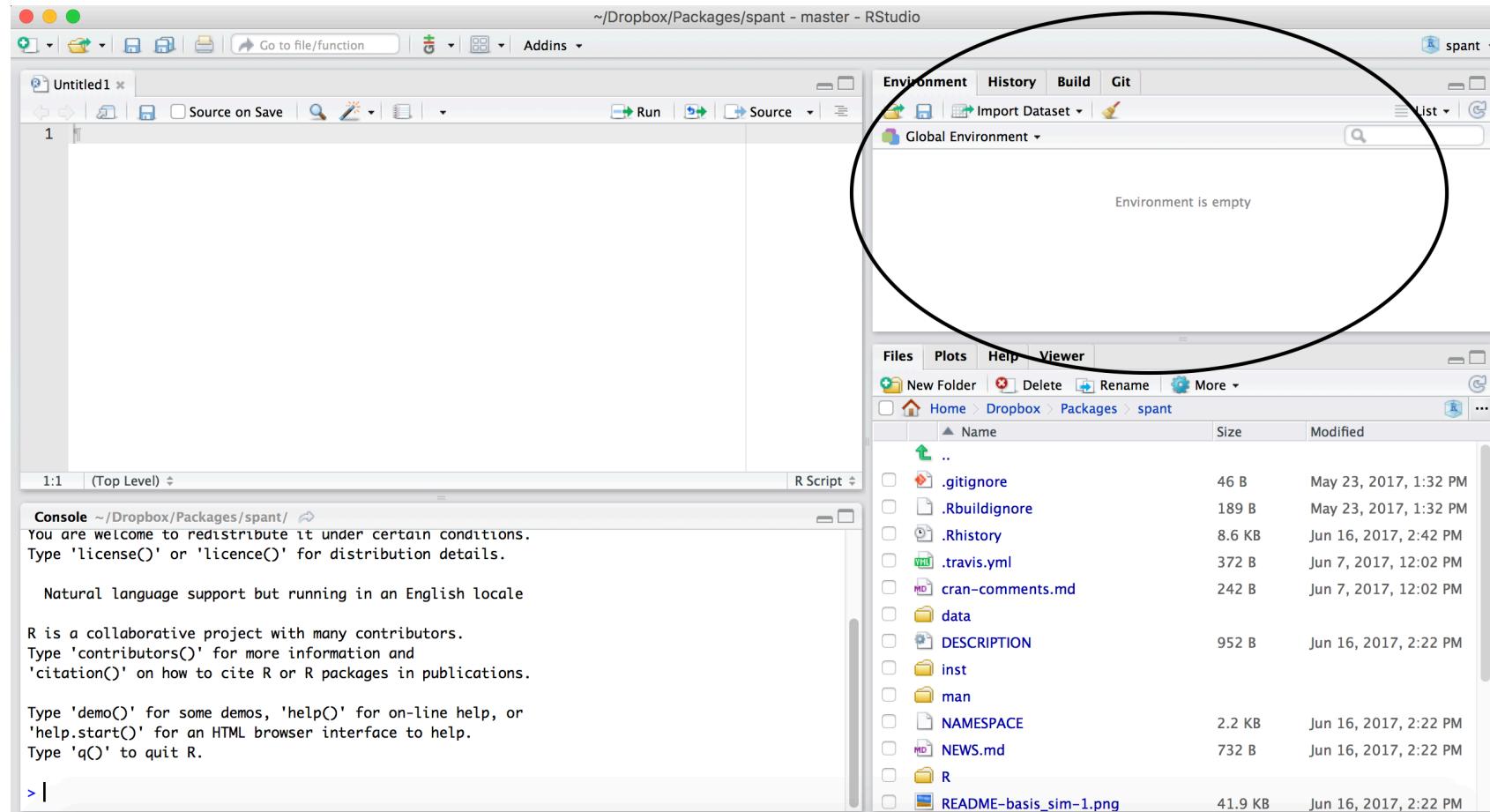


## Scripts and R Markdown

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

## Other Panes

- **Files** - shows the files on your computer or the directory you are working in
- **Viewer** - can view data or R objects
- **Help** - shows help of R commands
- **Plots** - pictures and figures
- **Packages** - list of R packages that are loaded in memory

Let's take a look at R Studio  
ourselves!

# Lab: Starting with R and RMarkdown

## RStudio Lab

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)

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.

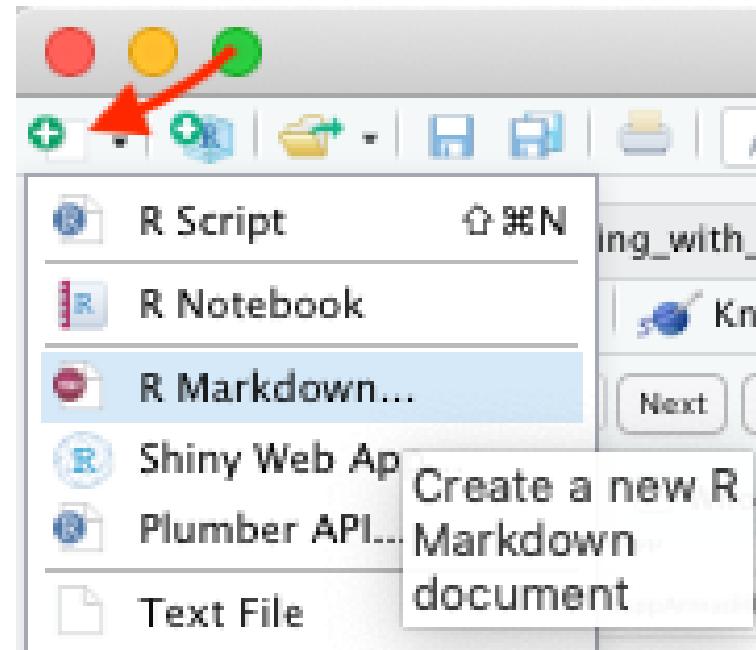
## R Markdown file

R Markdown files (.Rmd) help generate reports that include your code and output. Think of them as fancier scripts.

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

# Create an R Markdown file

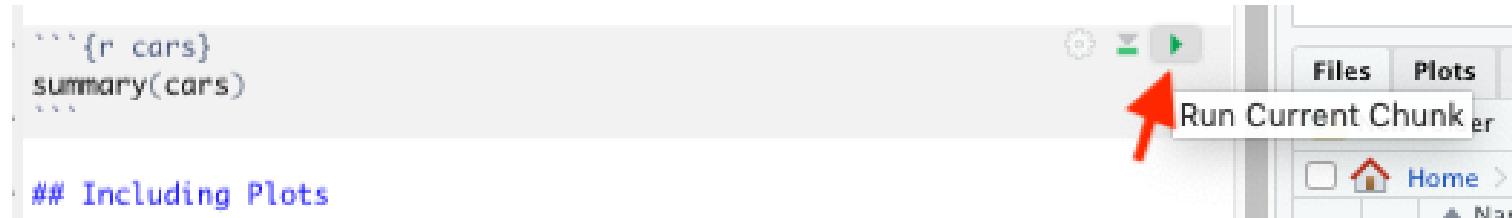
Go to File → New File → R Markdown or click the green add file button.



## Code chunks

Within R Markdown files are code “chunks”.

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

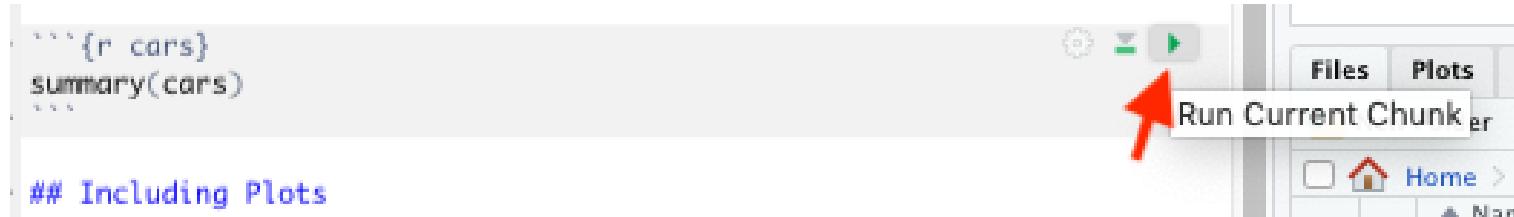


The image shows a screenshot of the RStudio IDE. On the left, there is a code editor window containing R code. The code includes a code chunk with three dots at the top, followed by `summary(cars)`, another three dots, and then a comment `## Including Plots`. On the right, the RStudio toolbar is visible, featuring various icons for file operations like Open, Save, and Print, as well as tabs for 'Files' and 'Plots'. A prominent red arrow points upwards from the bottom of the image towards the 'Run Current Chunk' button in the toolbar. This button is highlighted with a green border and has a small green play icon on it. The text 'Run Current Chunk' is also visible next to the button.

```
```{r cars}
summary(cars)
```
## Including Plots
```

## 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

The screenshot shows the RStudio interface. At the top, there are two tabs: "Untitled2" and "RStudio.Rmd". Below the tabs is a toolbar with icons for back, forward, knit, run, and other functions. The main area is divided into two sections: "Source" (active) and "Visual". In the Source tab, the R Markdown code is visible:

```
14 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>.  
15  
16 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:  
17  
18 ```{r cars}  
19 summary(cars)  
20 ```
```

Below the code, the Visual tab displays the rendered output of the R code chunk. It shows a table of summary statistics for the "cars" dataset:

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

At the bottom of the RStudio window, the "Console" tab is active, showing the R command and its output:

```
>  
>  
>  
> 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
```

# If you get annoyed by code previews in Markdown files:

In RStudio Click the Edit tab → scroll down to Preferences... → R Markdown

Uncheck the following:

The screenshot shows the 'Options' dialog in RStudio. On the left, a sidebar lists various sections: General, Code, Console, Appearance, Pane Layout, Packages, R Markdown (which is selected and highlighted in blue), Python, Sweave, and Spelling. The main area contains tabs for Basic, Advanced, Visual, and Citations, with 'Basic' selected. Under the 'R Markdown' section, there are several configuration options:

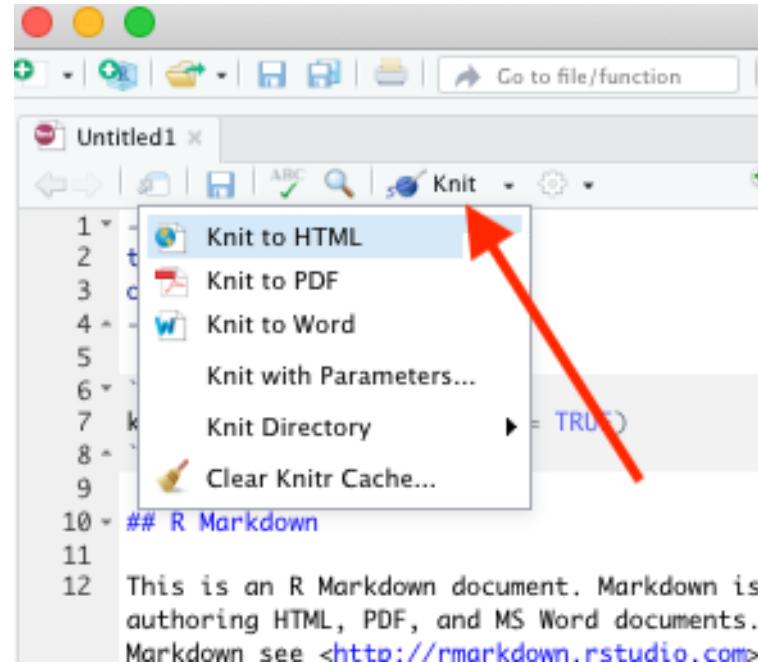
- Show document outline by default
- Soft-wrap R Markdown files
- Show in document outline:
- Show output preview in:
- Show output inline for all R Markdown documents (this option is highlighted with a red box)
- Show equation and image previews:
- Evaluate chunks in directory:

Below this, under 'R Notebooks', are two more options:

- Execute setup chunk automatically in notebooks
- Hide console automatically when executing notebook chunks

## Knit file to html

Running all chunks - this will create a report from the R Markdown document!



# Nice report!

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

The screenshot shows a web browser window with the following details:

- Address Bar:** ~/Documents/Roger/New\_Folder/Untitled.html
- Toolbar:** Untitled.html | Open in Browser | Find | Publish
- Content Area:**
  - # 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)
```

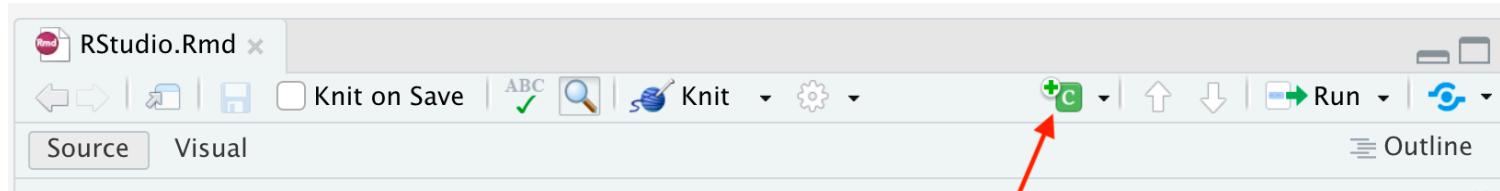
```
##      speed      dist
## Min.   : 4.0   Min.   : 2.00
## 1st Qu.:12.0   1st Qu.: 26.00
## Median :15.0   Median : 36.00
## Mean   :15.4   Mean   : 42.98
## 3rd Qu.:19.0   3rd Qu.: 56.00
## Max.   :25.0   Max.   :120.00
```
  - ## Including Plots

You can also embed plots, for example:

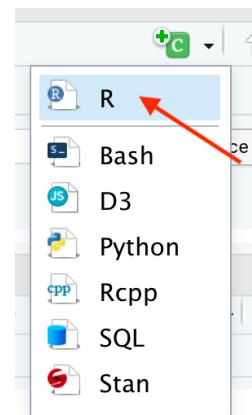
# Create Chunks

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

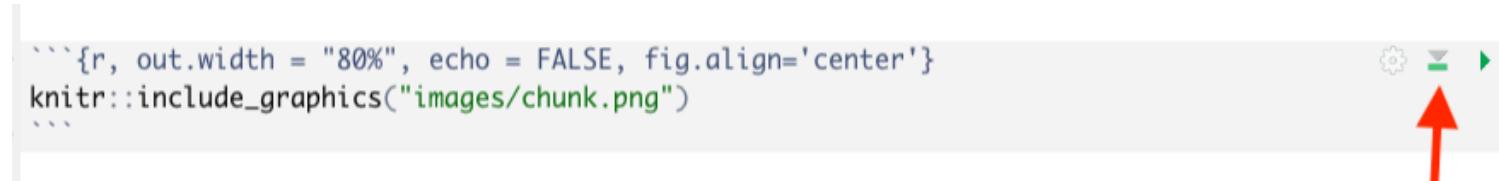
If you like keyboard shortcuts:

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

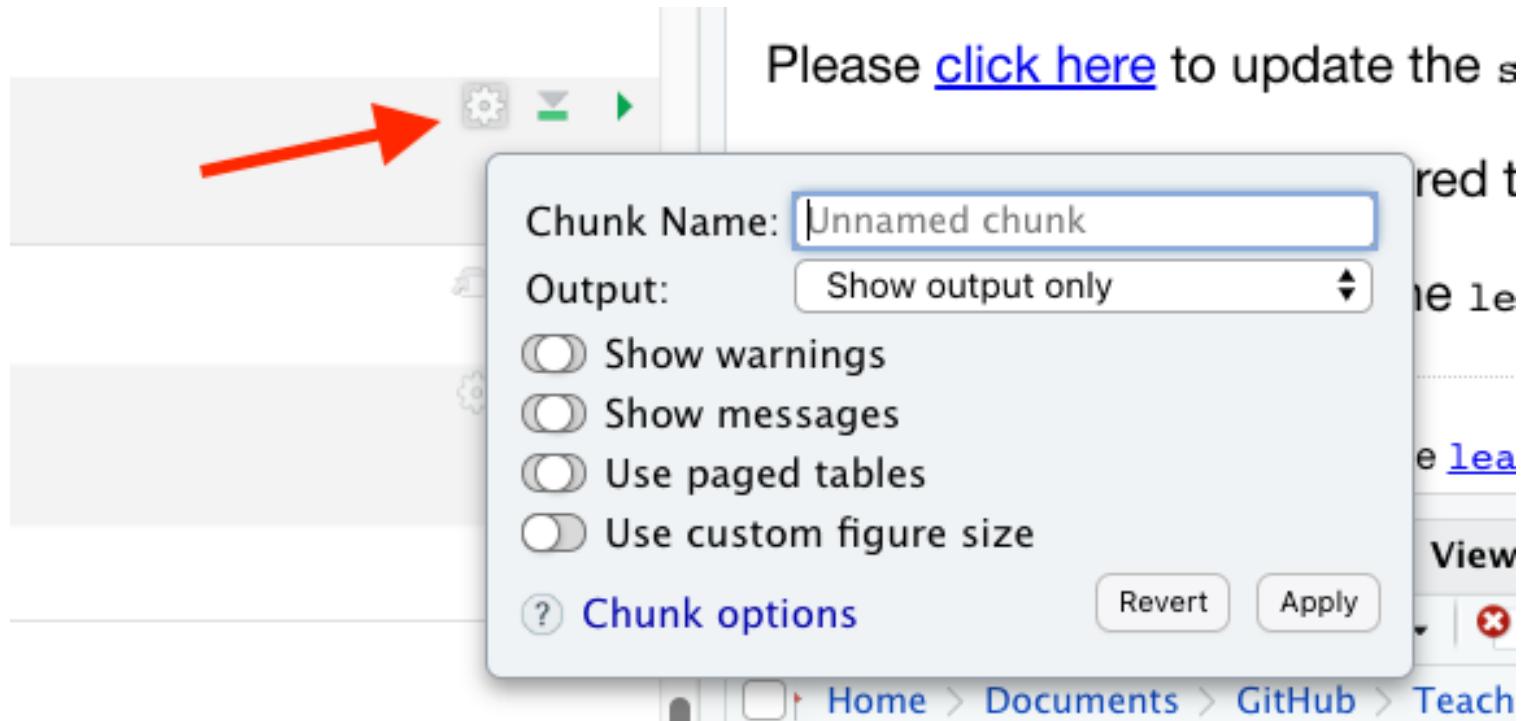
I is for insert.

## Run previous chunks button

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

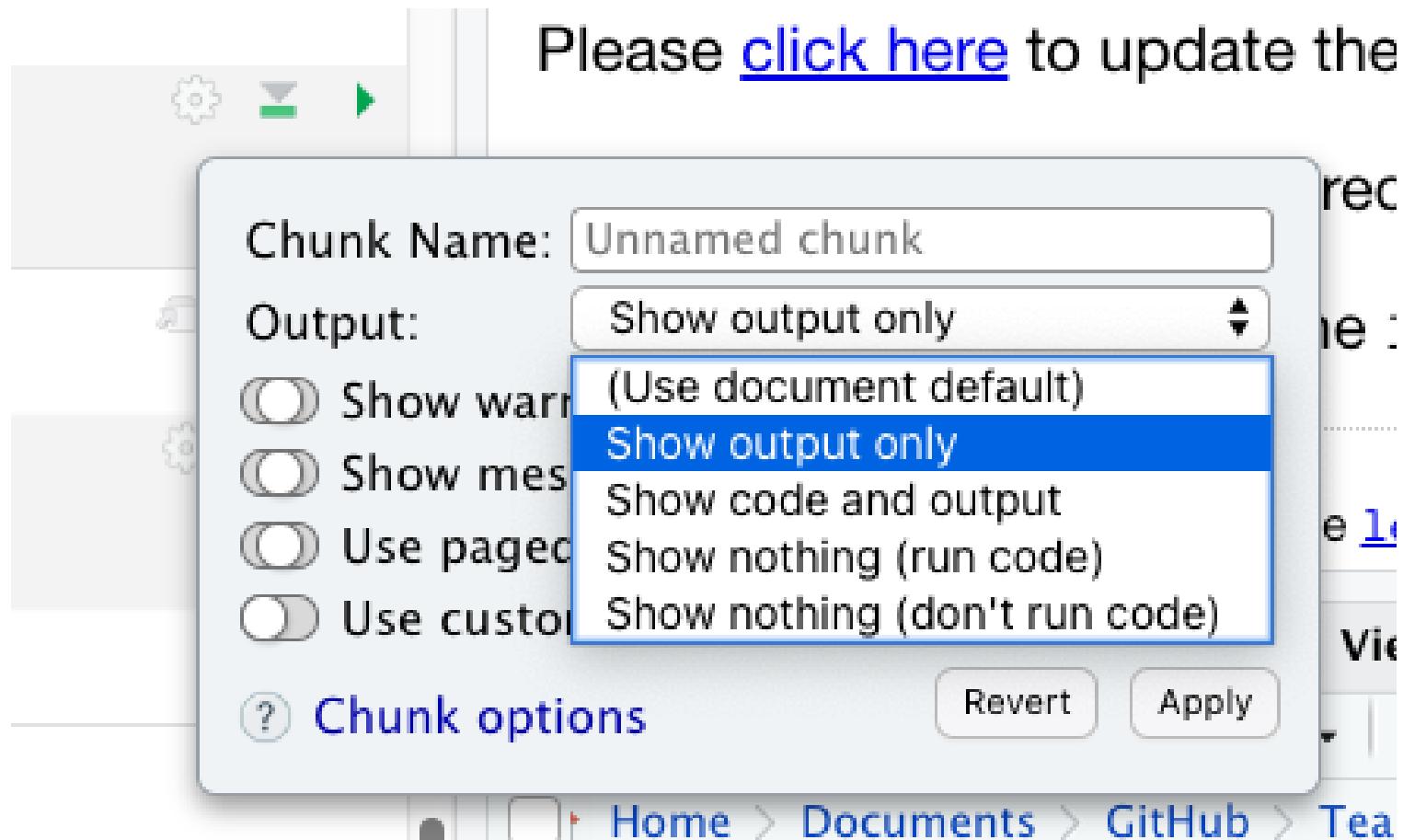


## Chunk settings



## Chunk settings

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



## Errors

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



The screenshot shows an RStudio interface with a code editor containing the following R code:

```
305 print(x, ...)  
306 - {r}  
✖ 307 print(x))  
308 ...  
3 unexpected token ')'  
3 unexpected end of document
```

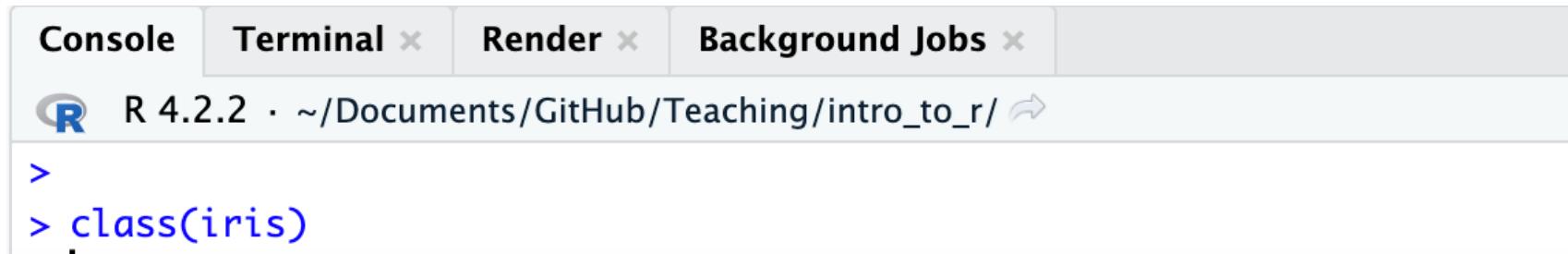
A yellow box highlights the closing parenthesis ')' at line 307. A red circle with a white 'x' marks the line number 307, indicating an error. The status bar at the bottom shows three icons: a gear, a downward arrow, and a green arrow pointing right.

## Useful R Studio Shortcuts

- **Ctrl + Enter** on Windows or **Command + Enter** on Mac in your script evaluates that line of code
  - It's like copying and pasting the code into the console for it to run.
- **Ctrl+1** on Windows or **Command + 1** on Mac takes you to the script page
- **Ctrl+2** on Windows or **Command + 2** on Mac takes you to the console
- [http://www.rstudio.com/ide/docs/using/keyboard\\_shortcuts](http://www.rstudio.com/ide/docs/using/keyboard_shortcuts)

## Recap of where code goes

- you can test code in the console



A screenshot of the RStudio interface showing the Console tab selected. The console window displays the following text:

```
R 4.2.2 · ~/Documents/GitHub/Teaching/intro_to_r/ ↗  
>  
> class(iris)  
.
```

- 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 documentation preview feature.

**Top Example:** The user has typed "`> class`". A tooltip appears over the first result in the completion dropdown, which is `class(x)`. The tooltip contains the following text:  
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.  
Press F1 for additional help

**Bottom Example:** The user has typed "`> read_`". A tooltip appears over the first result in the completion dropdown, which is `read_csv`. The tooltip contains the following text:  
The 'tidyverse'  
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(),  
 ...)  
Press F1 for additional help

# Get help with the help pane



The screenshot shows the RStudio interface with the 'Help' tab selected in the top navigation bar. Below the navigation bar are several icons: a left arrow, a right arrow, a house icon, and a refresh/circular arrow icon. To the right of these icons is a search bar containing the word 'class'. Below the search bar is a dropdown menu labeled 'R: Object Classes' with a 'Find in Topic' button next to it. The main content area displays the title 'class {base}' and the text 'R Documentation'.

## Object Classes

### Description

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.

### Usage

```
class(x)
class(x) <- value
unclass(x)
inherits(x, what, which = FALSE)
isa(x, what)
oldClass(x)
```

# 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 with two main panes. The left pane is the Console, displaying the command `?class` entered by the user. The right pane is the Help viewer, showing the documentation for the `class` object. The title of the documentation is "Object Classes". The "Description" section states: "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." The "Usage" section lists the following functions: `class(x)`, `class(x) <- value`, `unclass(x)`, `inherits(x, what, which = FALSE)`, and `isa(x, what)`. The R Documentation logo is visible in the top right corner of the Help viewer.

# 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 Console tab is active, displaying R code and its output. The user has run several commands related to the tidyverse package, including `?class`, `?tidyverse`, and `??tidyverse`. The output for `??tidyverse` shows that no documentation was found in the specified packages and libraries, but suggests trying `??tidyverse`. The user then loads the tidyverse package using `library(tidyverse)`. The tidyverse documentation page is open in the Help viewer on the right. The title is "tidyverse: Easily Install and Load the 'Tidyverse'". The description explains that the tidyverse is a set of packages designed to work in harmony, sharing common data representations and API design. It lists the core packages: ggplot2, dplyr, tibble, stringr, tidyr, and forcats. The maintainer is Hadley Wickham. A "tidyverse" hexagonal logo is visible on the right.

Console Terminal × Render × Background Jobs ×

R 4.2.2 · ~/Documents/GitHub/Teaching/intro\_to\_r/ ↵

```
>
>
>
>
>
>
> ?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
> |
```

Files Plots Packages Help Git Viewer Presentation

R: tidyverse: Easily Install and Load the 'Tidyverse' Find in Topic

tidyverse-package {tidyverse} R Documentation

## tidyverse: Easily Install and Load the 'Tidyverse'

### Description

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>.

### Author(s)

Maintainer: Hadley Wickham [hadley@rstudio.com](mailto:hadley@rstudio.com)

Other contributors:



## 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](#)



Image by [Gerd Altmann](#) from [Pixabay](#)