

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

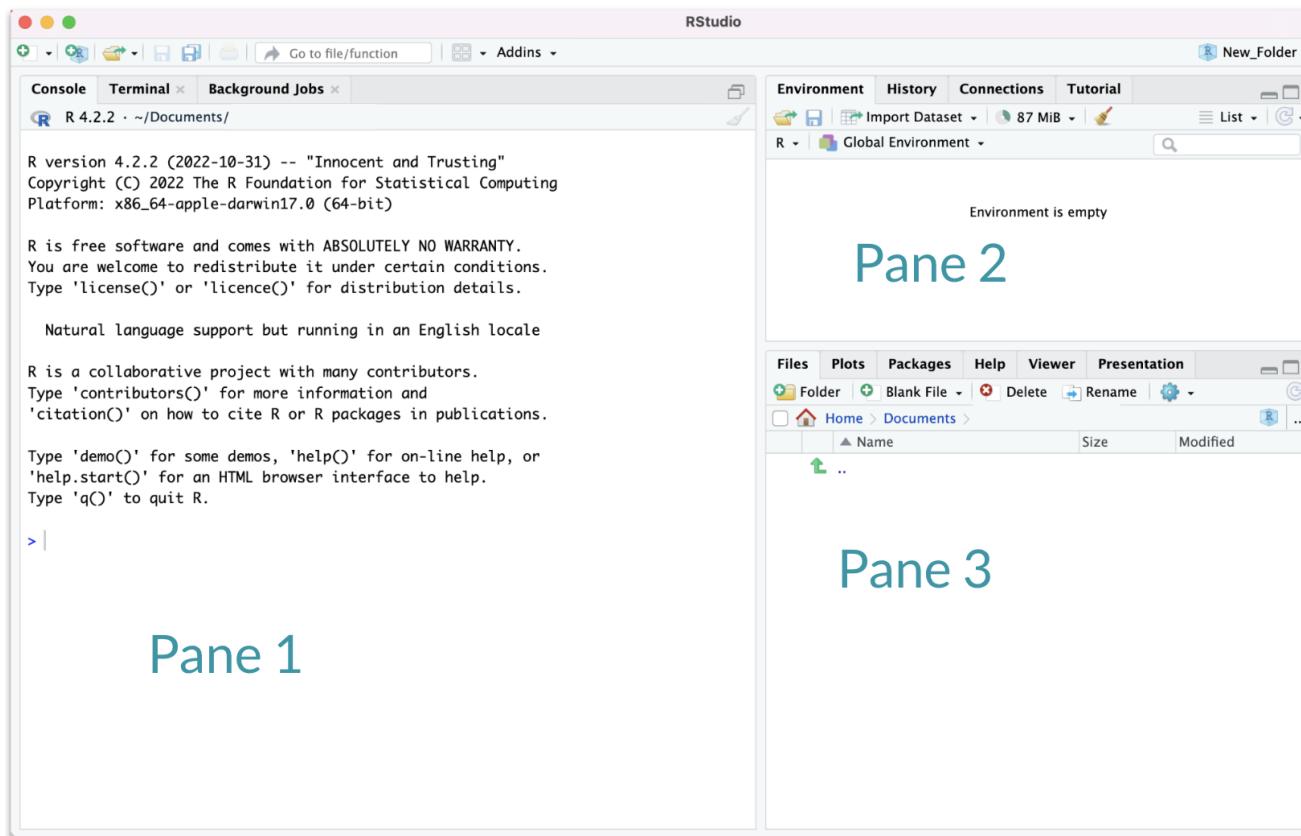
- Syntax highlighting, code completion, and smart indentation
- Easily manage multiple working directories and projects

More information

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

RStudio

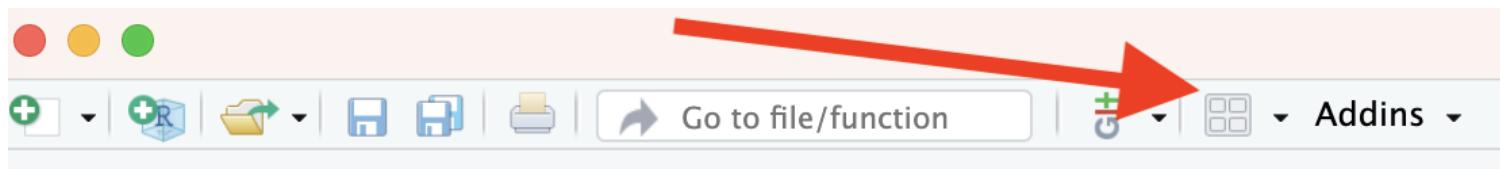
First it is important to be familiar with the layout. When you first open RStudio, you will see 3 panes.



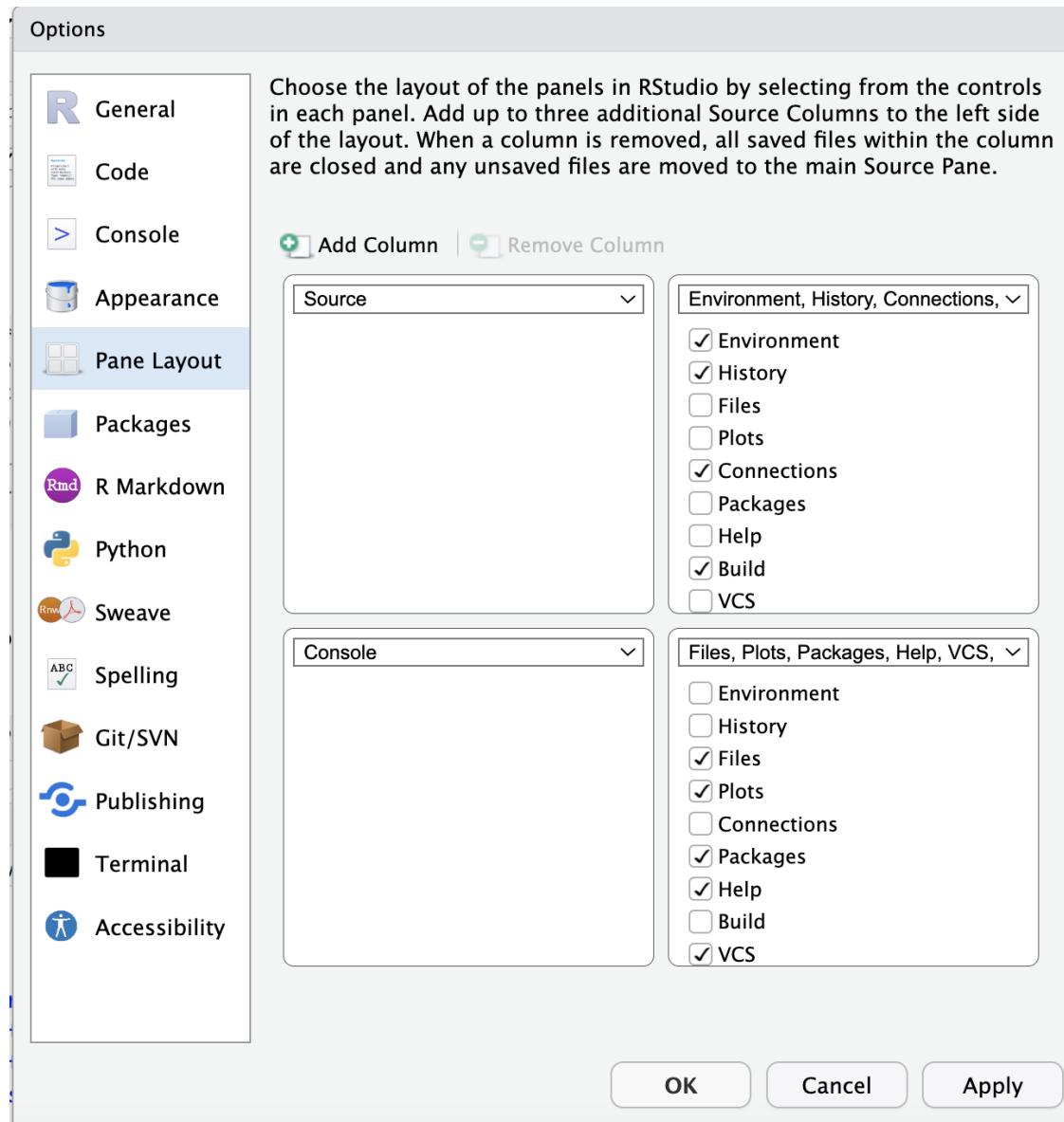
RStudio Layout

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

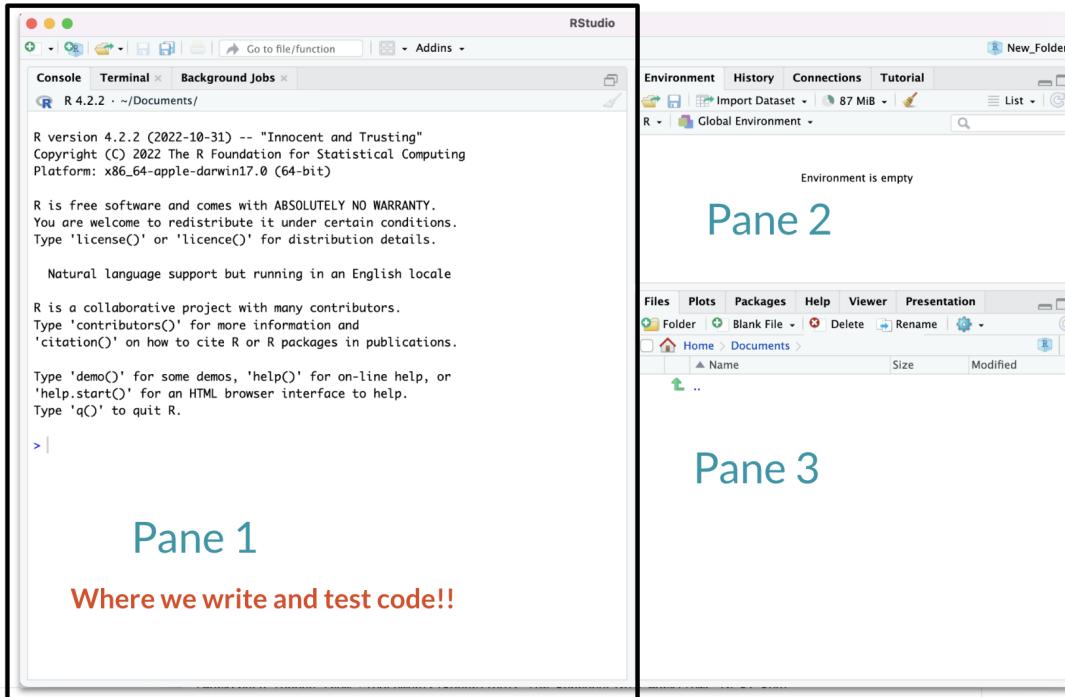
Click on the pane button, which looks like a waffle with 4 indentations. Scroll down to "Pane Layout".



Default Layout

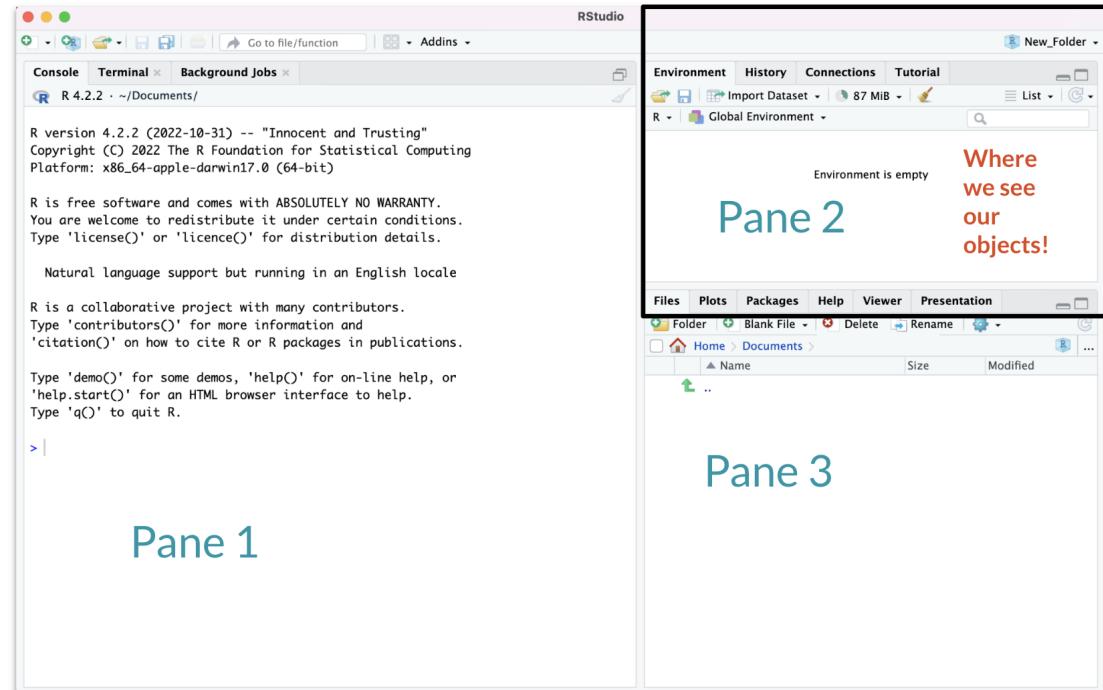


Pane 1 (Left side) for writing code

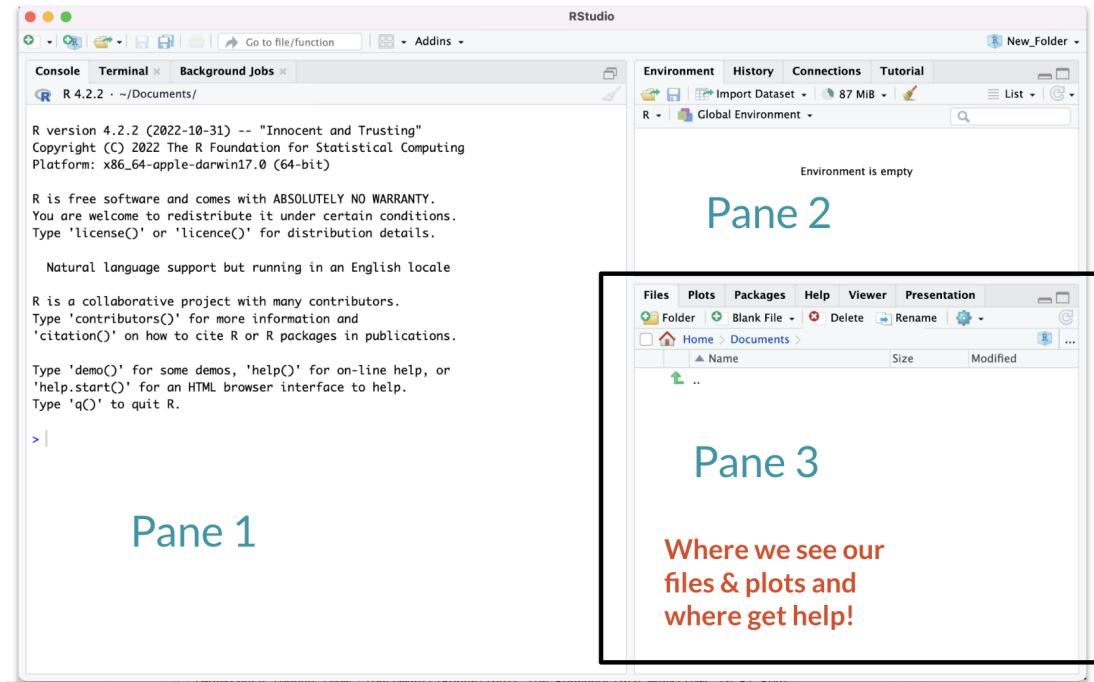


Pane 2 - where objects will be

We will start seeing this tomorrow!

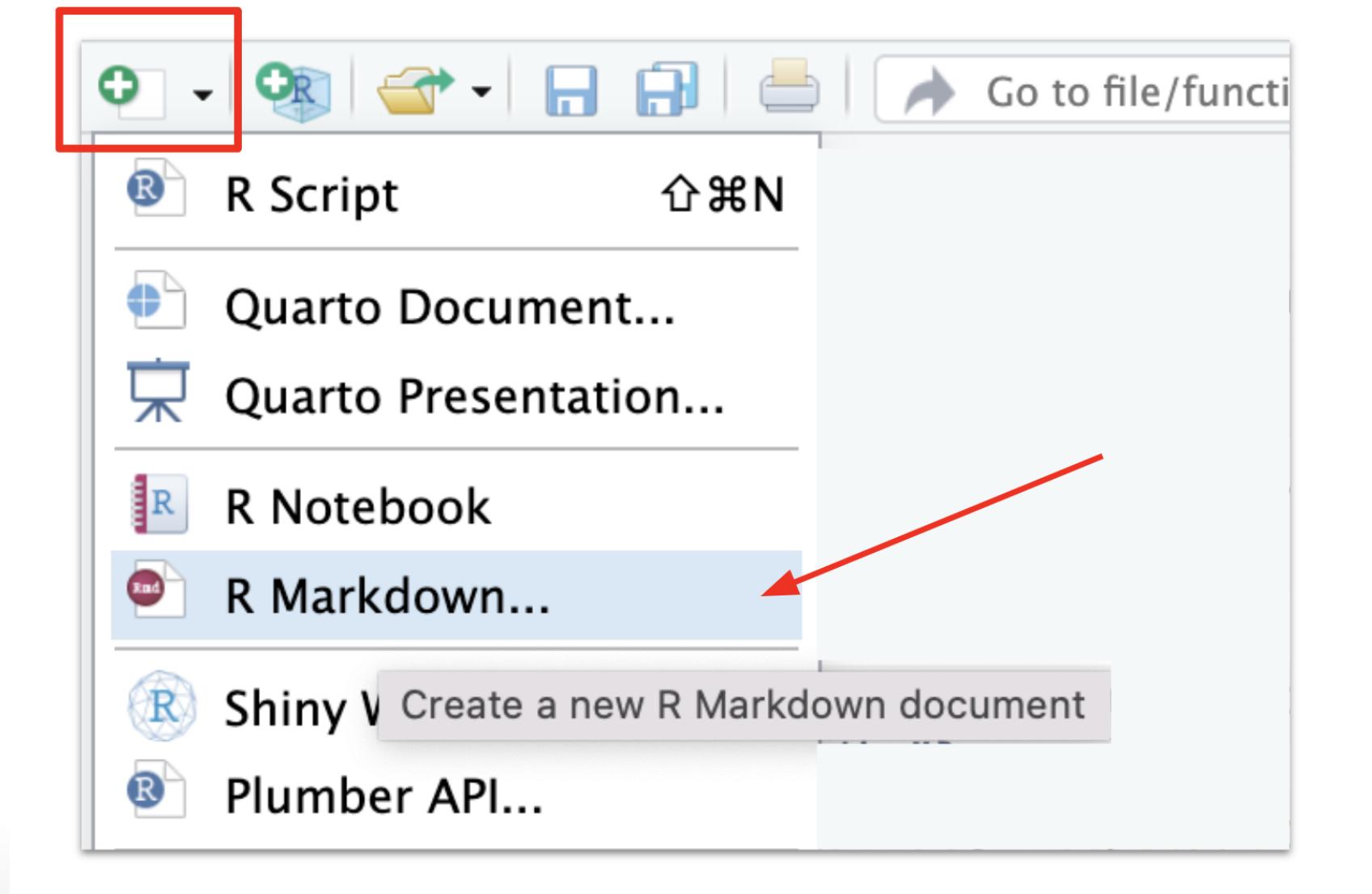


Pane 3 - where we get help and see plots



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:

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.

The screenshot shows the RStudio interface with three main panes:

- Pane 1 (Squished down):** The bottom-left pane, which is very narrow, displays the R console output. It shows the R version information, the license terms, and a note about natural language support.
- Pane 2 (Empty Environment):** The top-right pane, which is also very narrow, displays the R environment. It shows the message "Environment is empty".
- Pane 3 (New Pane):** The top-left pane, which is wider than the others, displays an R Markdown document titled "Untitled". The document contains code for setting up the environment, including knitr options, and a brief introduction to R Markdown.

Annotations in the image:

- "New Pane!" is written in blue text above the top-left pane.
- "Pane 1 squished down" is written in blue text above the bottom-left pane.
- "Pane 2" is written in blue text above the top-right pane.
- "Pane 3" is written in blue text above the top-left pane.

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

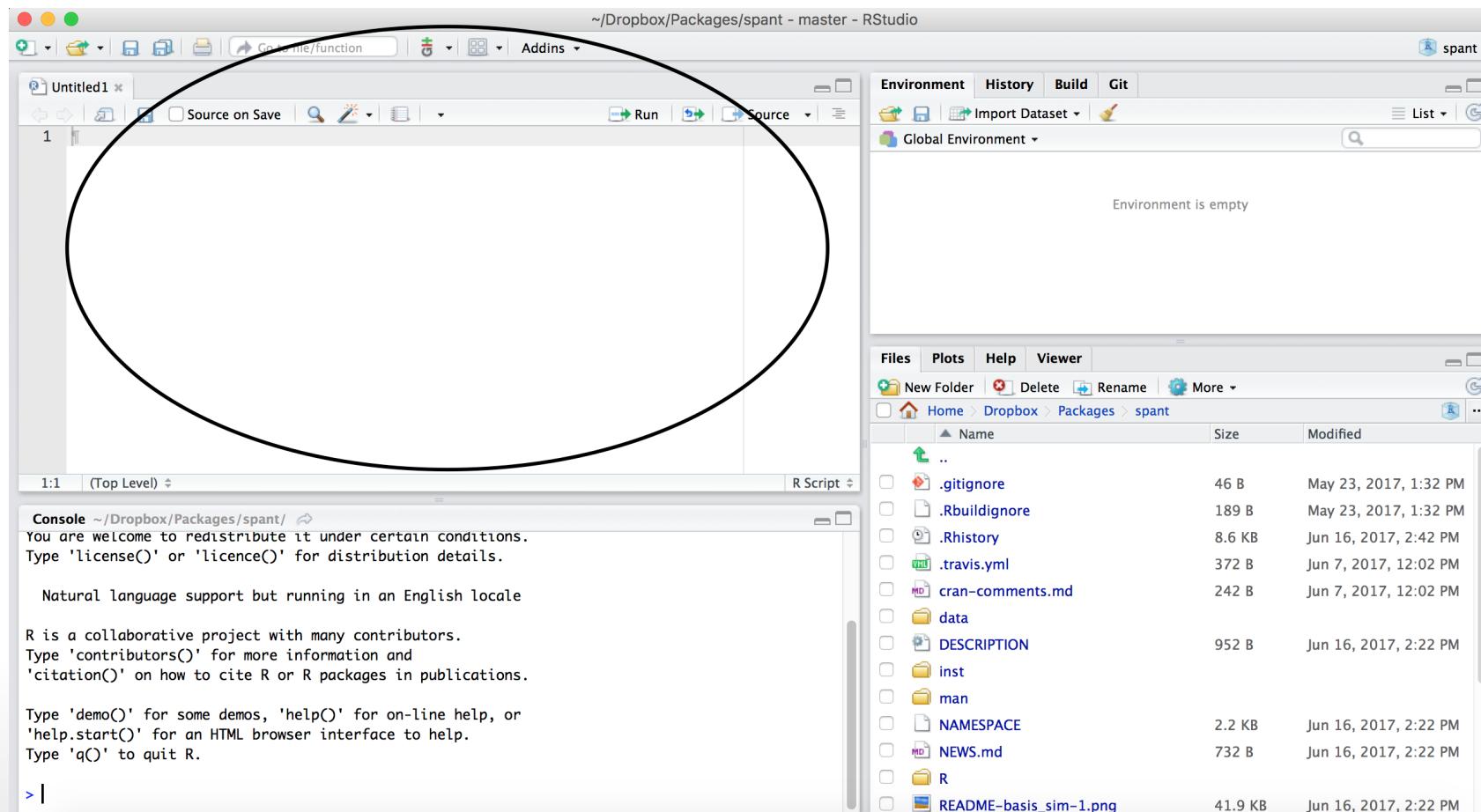
2. The **R Console**: "interprets" whatever you type

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

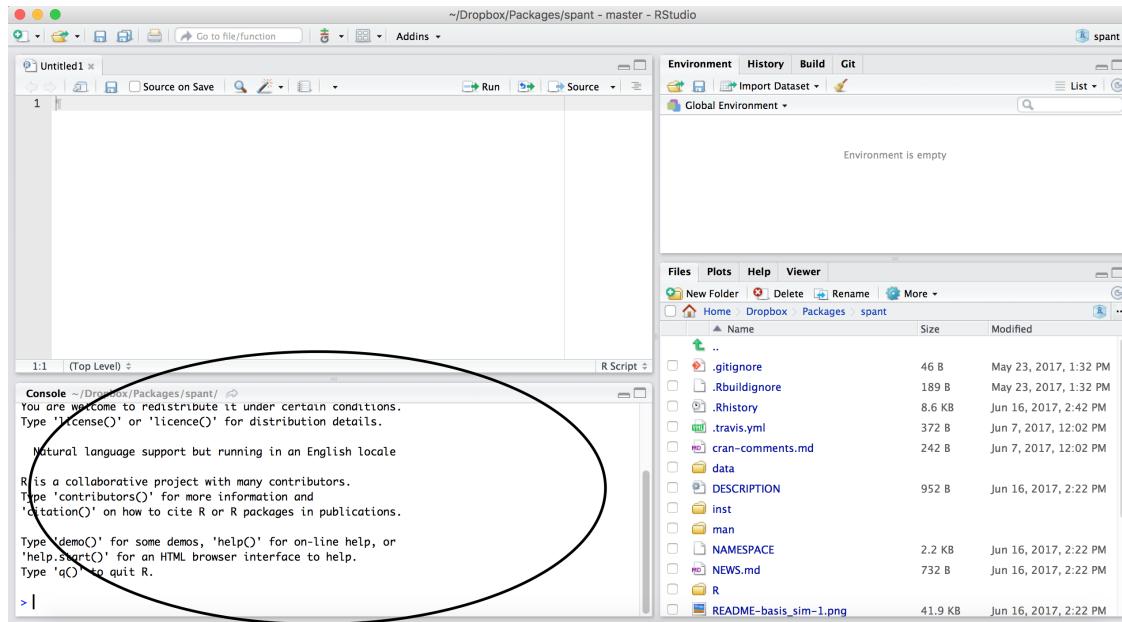
Source / Editor

- 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



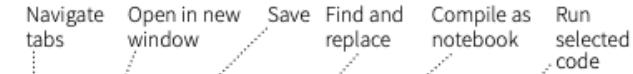
- Where code is executed (where things happen)
- You can type here for things interactively to test code
- Code is **not saved** on your disk

RStudio

Super useful “cheat sheet”:

<https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf>

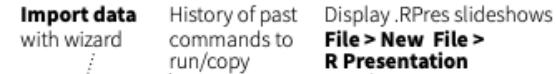
Write Code



The screenshot illustrates several key features of the RStudio IDE:

- File Bar:** Shows tabs for RMD, app.R, Split.R, and Addins.
- Code Editor:** Displays R code for a function named `get_digit`. Callouts explain:
 - Multiple cursors/column selection with **Alt + mouse drag**.
 - Code diagnostics in the margin; hover over symbols for details.
 - Syntax highlighting based on file extension.
 - Tab completion for function names, file paths, arguments, and more.
 - Jump to function in file.
 - Multi-language code snippets to quickly use common blocks of code.
- Environment Tab:** Shows the global environment with objects like `iris`, `foo`, and `a`. Callouts explain:
 - Choose environment to display from list of parent environments.
 - Load workspace.
 - Save workspace.
 - Delete all saved objects.
 - Search inside environment.
 - Display objects as list or grid.
- File Browser:** Shows a tree view of files and folders. Callouts explain:
 - View in data viewer.
 - View function source code.
 - New Folder.
 - Upload.
 - Delete.
 - Rename.
 - More.
 - Create folder.
 - Upload file.
 - Delete file.
 - Rename file.
 - Copy...
 - Move...
 - Export...
 - Set As Working Directory.
 - Go To Working Directory.
 - Change directory.
- Console:** Shows command history and output. Callouts explain:
 - Working Directory.
 - Press **↑** to see command history.
 - Maximize, minimize panes.
 - Drag pane boundaries.
- Top Status Bar:** Shows the user name (garrett), session ID (Sessions), and R version (R 3.2.2).

R Support



The screenshot shows the RStudio IDE interface. At the top is the menu bar with tabs: Environment, History, Build, Git, and Presentation. Below the menu bar is the Global Environment pane, which displays a list of saved objects: iris (150 obs. of 5 variables), values (1), and functions (foo). To the right of the environment list are two buttons: "View in data viewer" and "View function source code".

Below the Global Environment is the File browser, titled "ath to displayed directory". It lists a single file named "hello.R". The status bar at the bottom indicates the file size as 19 B and the date/time as Apr 13, 2016, 11:17 AM.

Dotted lines from the text annotations point to specific elements in the interface:

- "Choose environment to display from list of parent environments" points to the Global Environment pane.
- "Display objects as list or grid" points to the "Search inside environment" button.
- "Displays saved objects by type with short description" points to the list of objects in the Global Environment.
- "View in data viewer" and "View function source code" both point to their respective buttons in the "View" menu.
- "Change directory" points to the "Change directory" button in the toolbar.
- "ath to displayed directory" points to the title bar of the File browser.
- "File browser keyed to your working directory. Click on file or directory name to open." points to the "hello.R" file entry in the File browser.

R Markdown files look different from scripts

It will look like this with text in it, unlike a script.

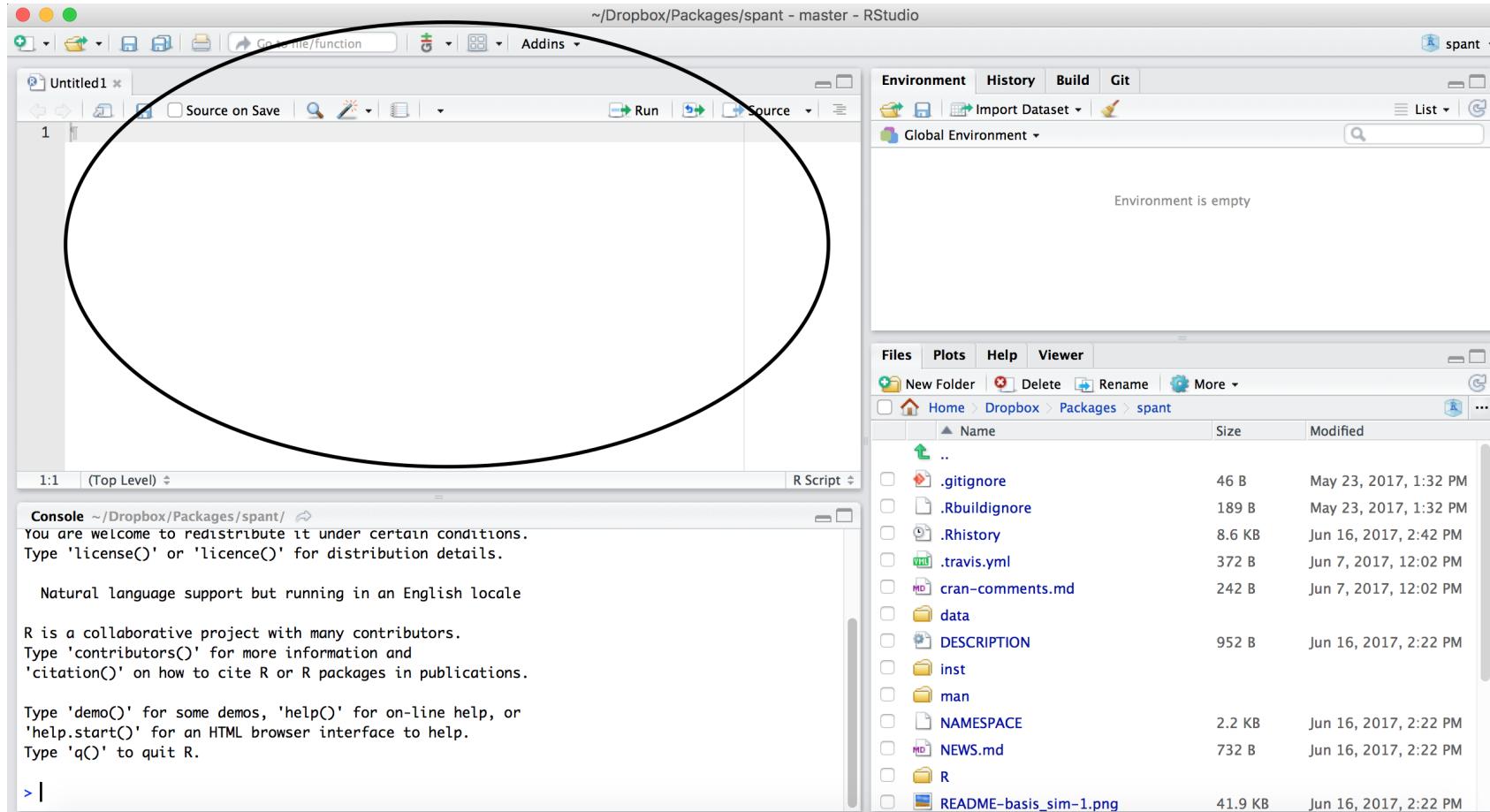
The screenshot shows the RStudio interface with a red box highlighting the main code editor area. The code editor contains the following R Markdown code:

```
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 includes a title, output type, and a code chunk setup. The text part describes what R Markdown is and how it works. The code editor also shows the current file path: ~/Documents/GitHub/Teaching/intro_to_r/.

The RStudio interface also includes a Global Environment panel showing "Environment is empty", a Files browser listing various project files, and a Console window displaying R startup messages and locale information.

Recall that a script was just empty

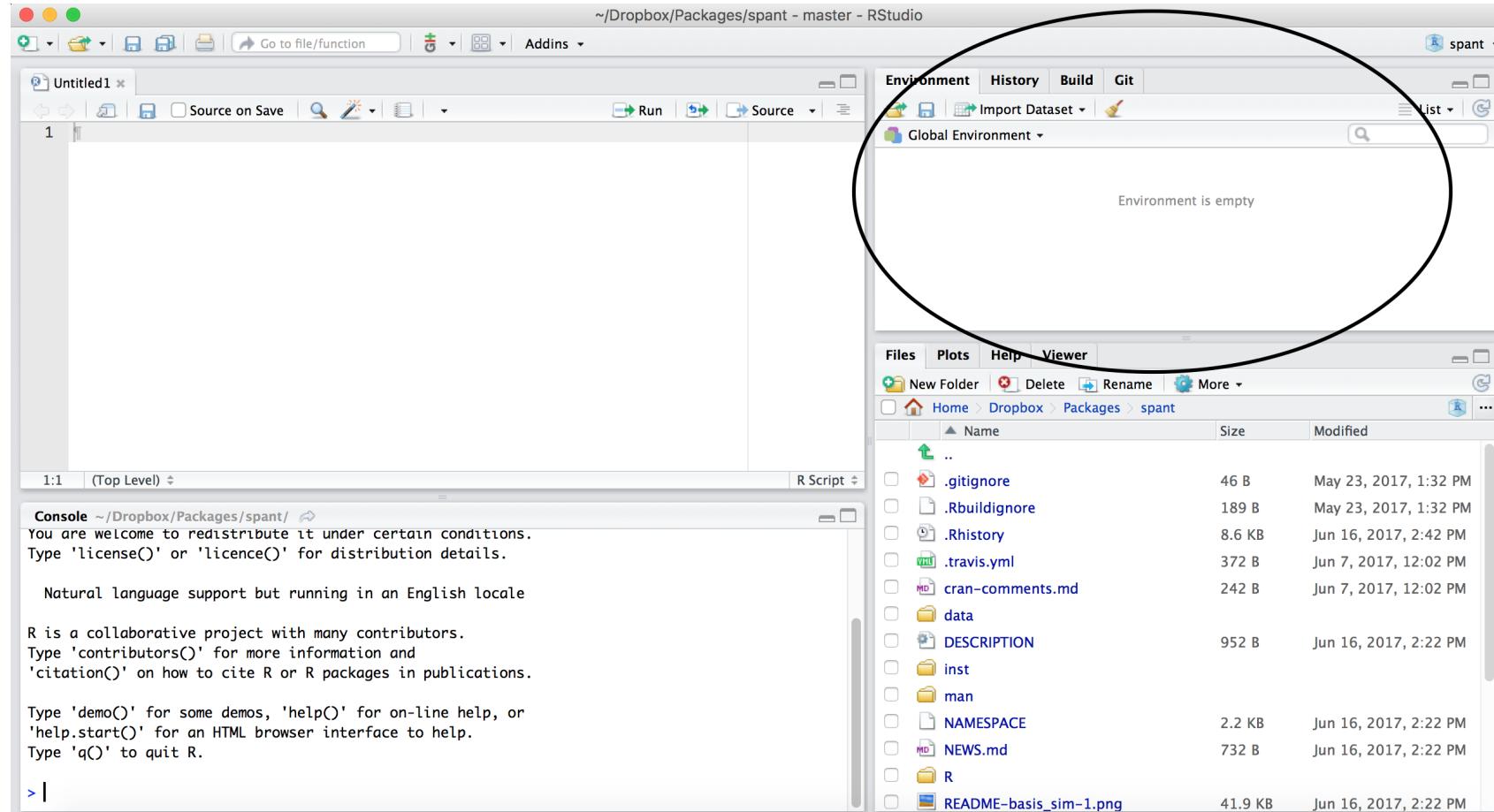


Scripts and R Markdown

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.

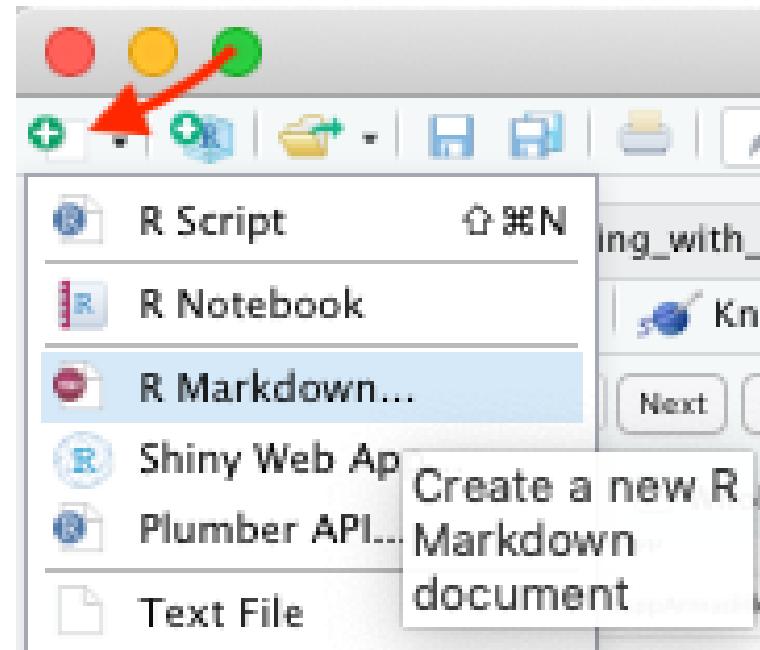
R Markdown file

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

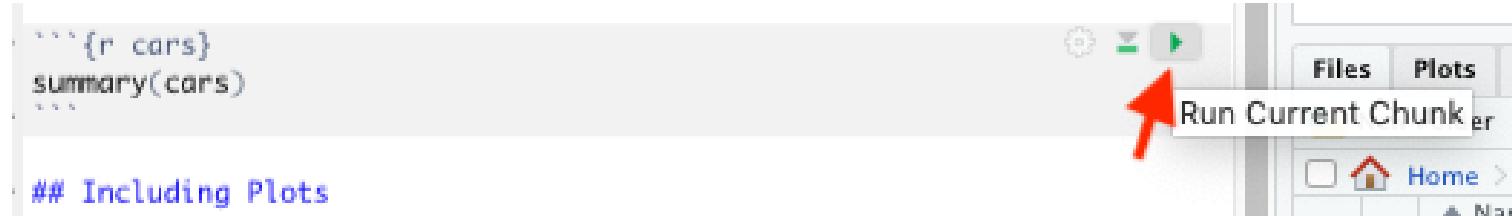
Go to File → New File → R Markdown or click the green add file button.



Code chunks

Within R Markdown files are code “chunks”.

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



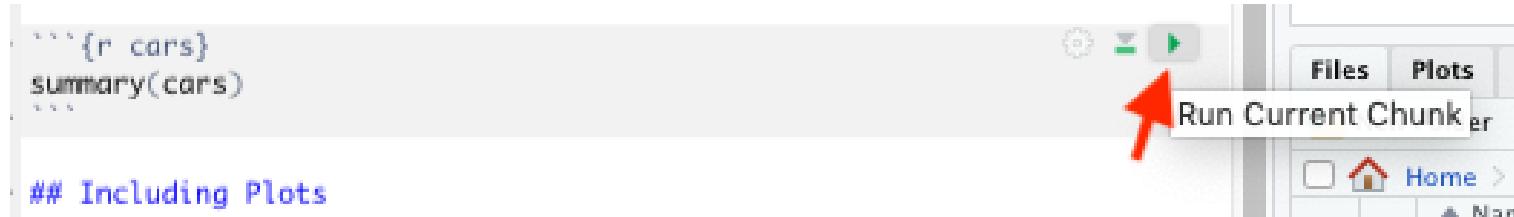
The image shows a screenshot of the RStudio IDE. On the left, there is a code editor window containing R code. The code includes a code chunk indicator (three backticks) followed by `summary(cars)`, and a comment `## Including Plots`. On the right, the RStudio toolbar is visible, featuring various icons for file operations like 'New File', 'Open', 'Save', and 'Run'. A red arrow points specifically to the 'Run' icon, which is a green square with a white play symbol. The menu bar at the top has 'File', 'Plots', 'Edit', 'View', 'Tools', 'Help', and 'More' options.

```
```{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 panes: "Source" (top) and "Visual" (bottom). In the Source pane, 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 ```
```

In the Visual pane, the output of the `summary(cars)` chunk is displayed as a table:

| | 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 |

At the bottom of the RStudio window, the "Console" tab is active, showing the R session history:

```
>  
>  
>  
> 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:

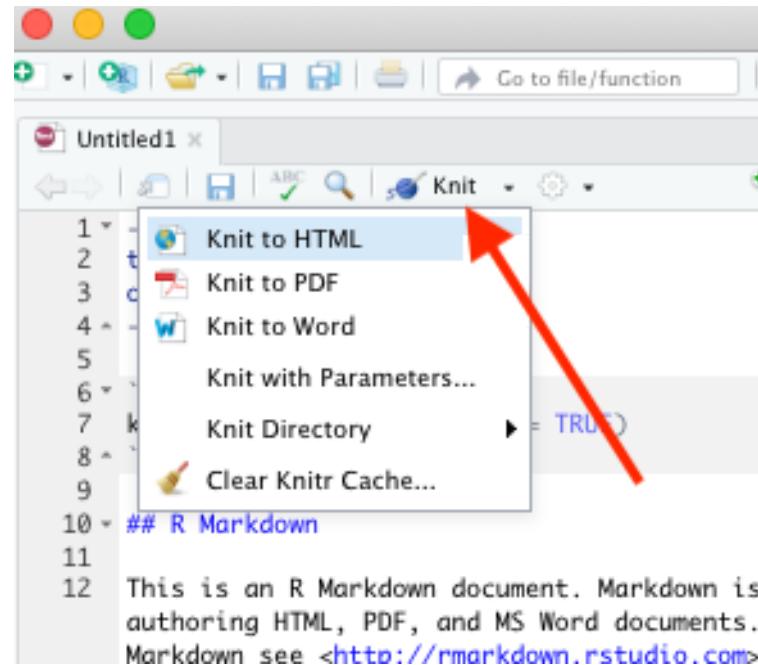
- Show document outline by default
- Soft-wrap R Markdown files
- Show in document outline:
- Show output preview in:
- Show output inline for all R Markdown documents
- Show equation and image previews:
- Evaluate chunks in directory:

Under the 'R Notebooks' section, two options are shown:

- 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 window titled 'Untitled.html' with the URL ' ~/Documents/Roger/New_Folder/Untitled.html'. The window includes standard OS X window controls (red, yellow, green) and a toolbar with 'Untitled.html', 'Open in Browser', 'Find', and a 'Publish' button. The main content area displays the following:

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:

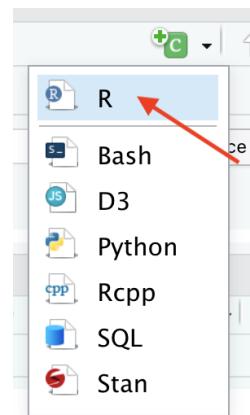
Create Chunks

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

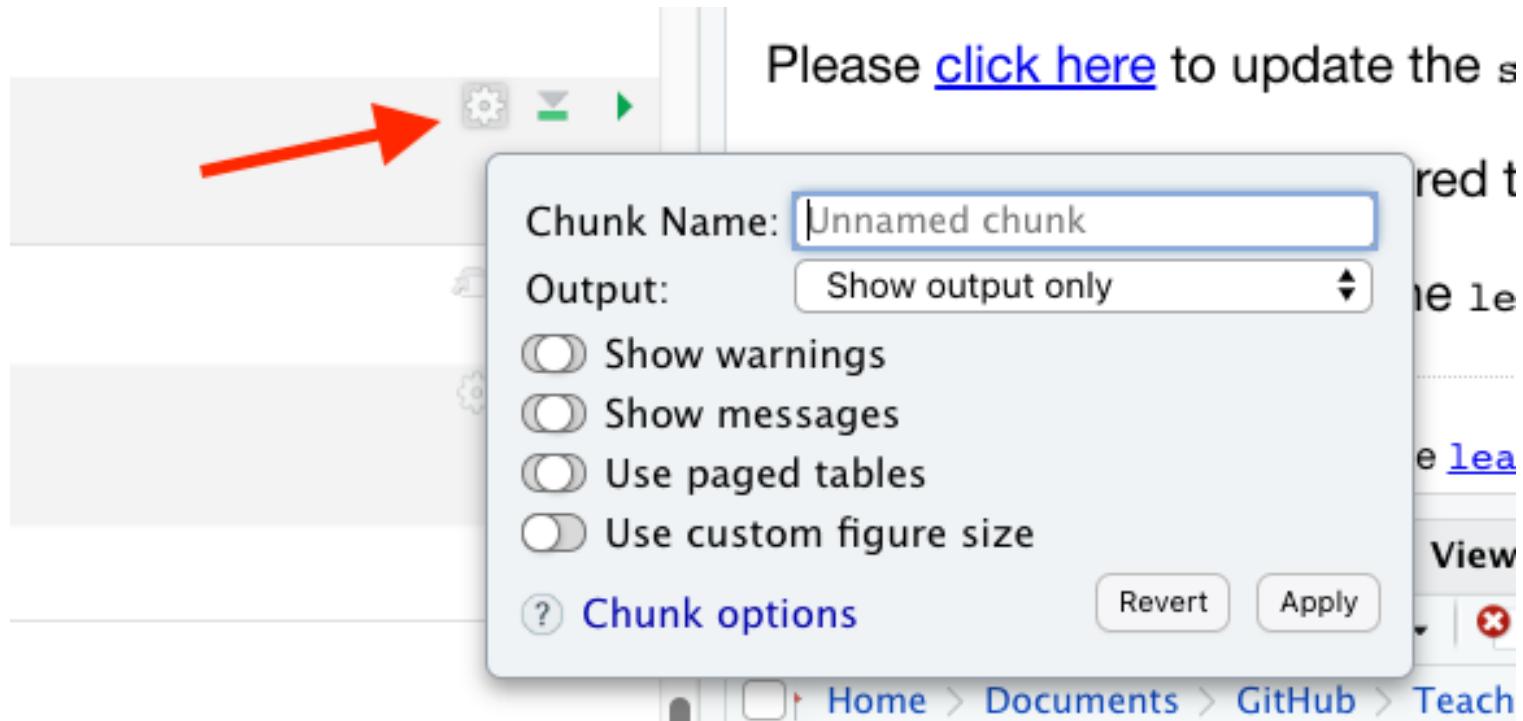
I is for insert.

Run previous chunks button

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

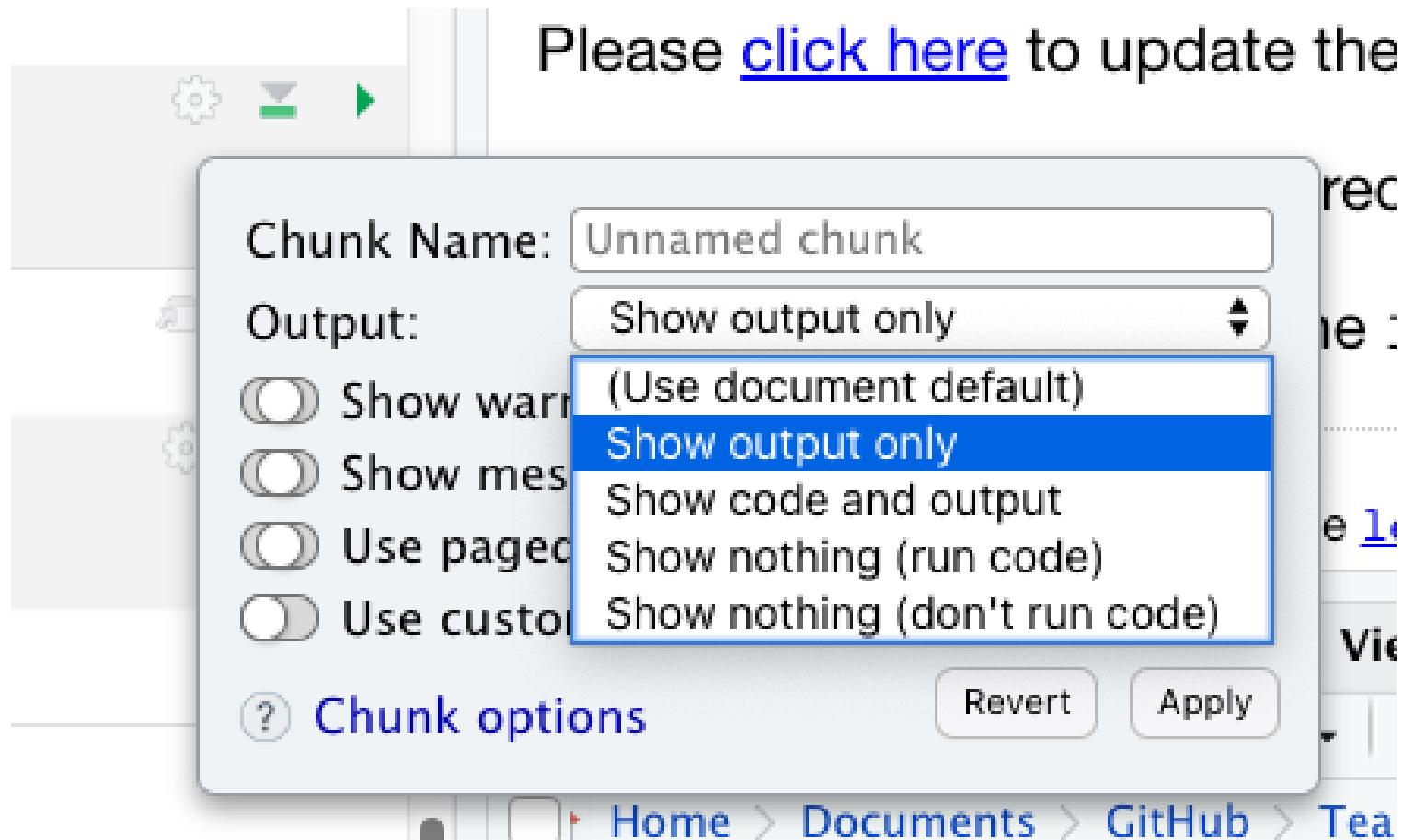


Chunk settings



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.



A screenshot of the RStudio IDE interface. The code editor shows the following R code:

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

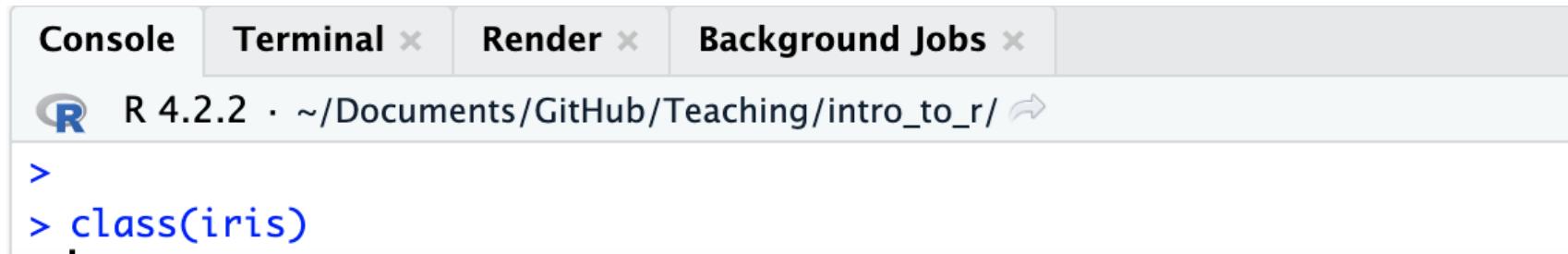
The line "print(x))" is highlighted with a red error indicator (a red circle with a white exclamation mark) and a yellow background. A tooltip or callout box is displayed below the error line, containing the two error messages: "unexpected token ')'". The RStudio toolbar at the top right includes icons for settings, file, and run.

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

Recap of where code goes

- you can test code in the console



The screenshot shows the RStudio interface with the 'Console' tab selected. The title bar indicates 'R 4.2.2 · ~/Documents/GitHub/Teaching/intro_to_r/'. The console window displays the following R session:

```
R 4.2.2 · ~/Documents/GitHub/Teaching/intro_to_r/
>
> class(iris)
[1] "data.frame"
```

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

```
## R Markdown
```

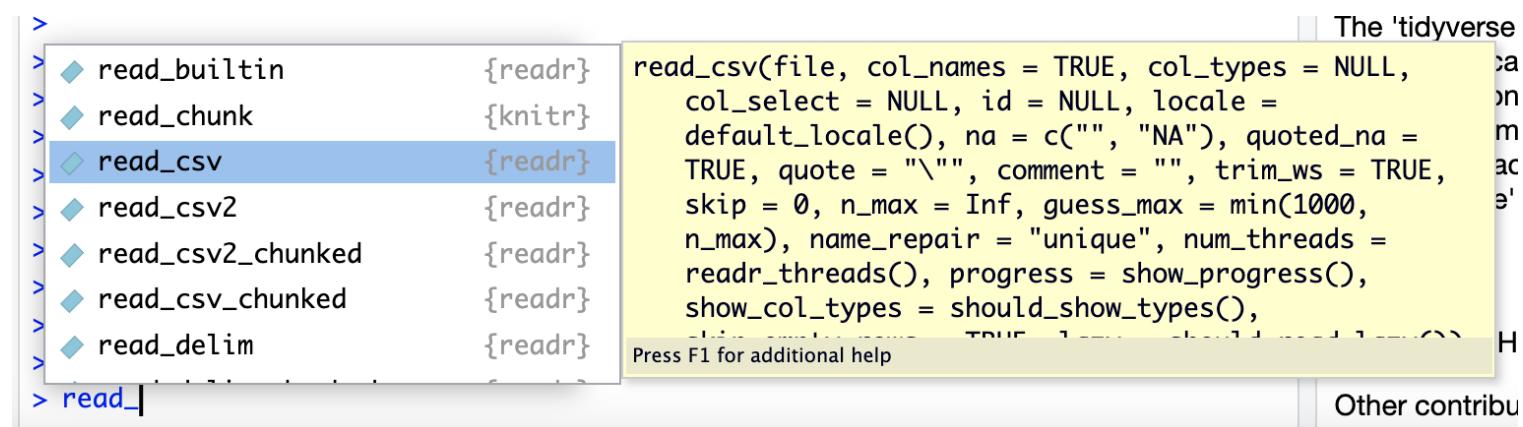
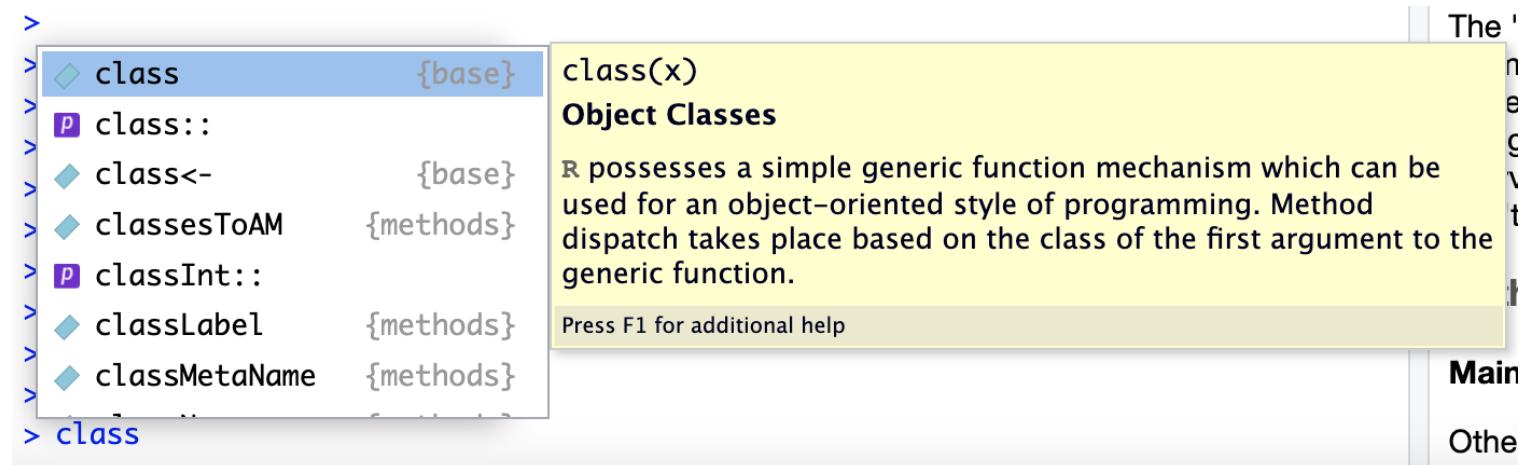
Code does not go here and instead goes within the grey chunks like this:

```
```{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!



Get help with the help pane



The screenshot shows the R help pane interface. At the top, there is a menu bar with tabs: Files, Plots, Packages, Help, Git, Viewer, and Presentation. Below the menu 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 text "class". Below the search bar is a button with a magnifying glass icon and another button with an 'X' icon. Underneath the search bar, the text "R: Object Classes" is displayed next to a dropdown arrow, and to its right is a "Find in Topic" button. 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

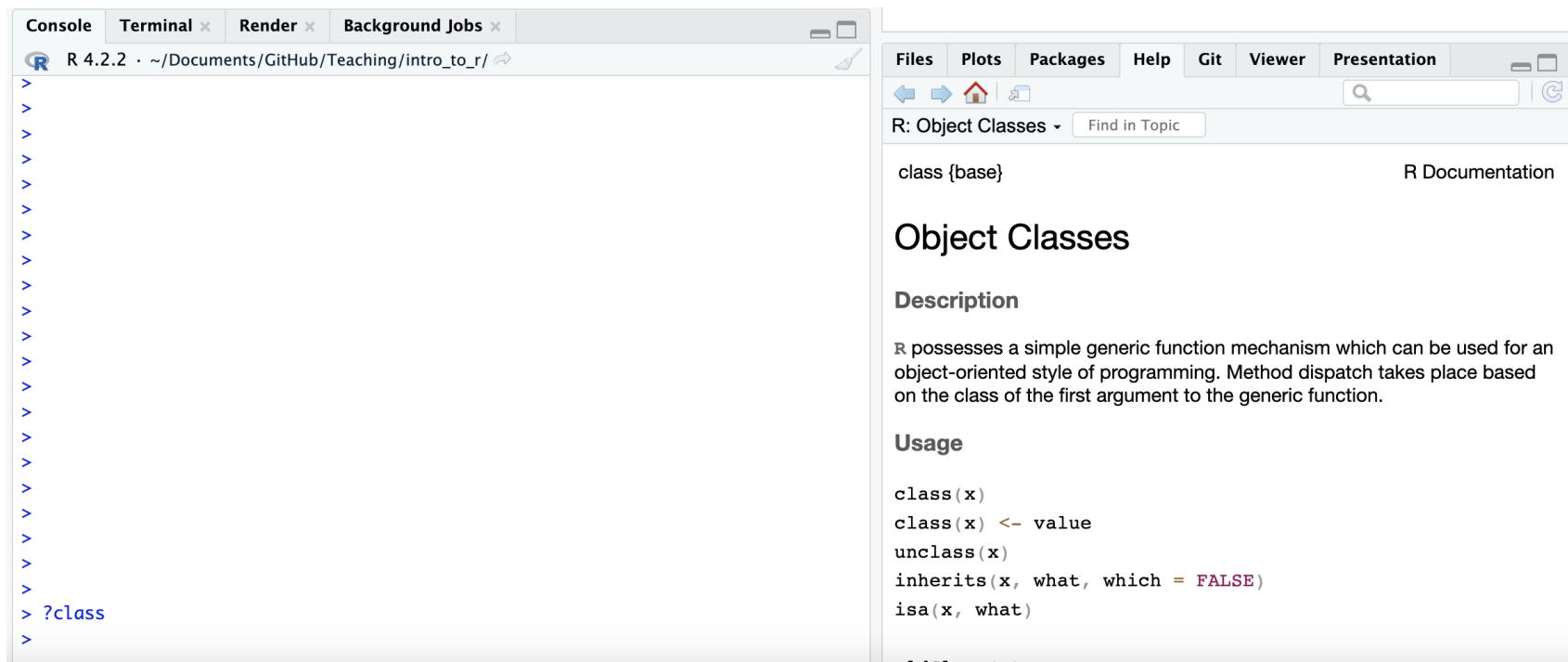
```
class(x)
class(x) <- value
unclass(x)
inherits(x, what, which = FALSE)
isa(x, what)
oldClass(x)
```

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.



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 there is no documentation for 'tidyverse' in specified packages and libraries, but suggests trying `??tidyverse`. The user then loads the tidyverse package using `library(tidyverse)`. The output shows the packages attached and conflicts resolved. On the right, the Help tab is active, displaying the documentation for the tidyverse package. The title is 'tidyverse: Easily Install and Load the 'Tidyverse''. It includes sections for 'Description', 'Author(s)', and 'Maintainer'. A 'tidyverse' logo is visible on the right side of the help window.

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

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](#)