

RStudio

Working with R – RStudio

RStudio is an Integrated Development Environment (IDE) for R

- It helps the user effectively use R
- Makes things easier
- Is NOT a dropdown statistical tool (such as Stata)
 - See [Rcmdr](#) or [Radiant](#)
- All R Studio snapshots are taken from <http://ayeimanol-r.net/2013/04/21/289/>



[\[source\]](#)

RStudio

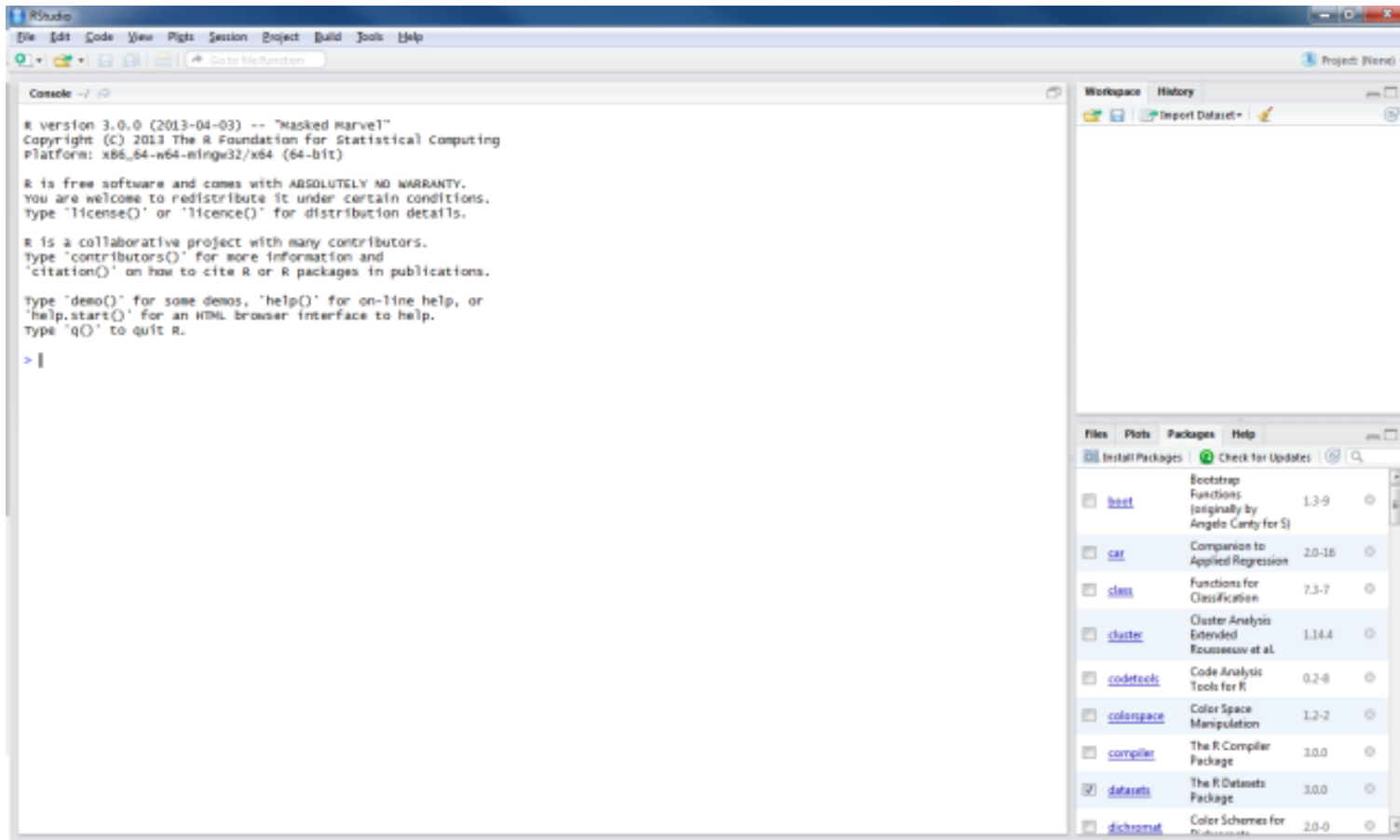
Easier working with R

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

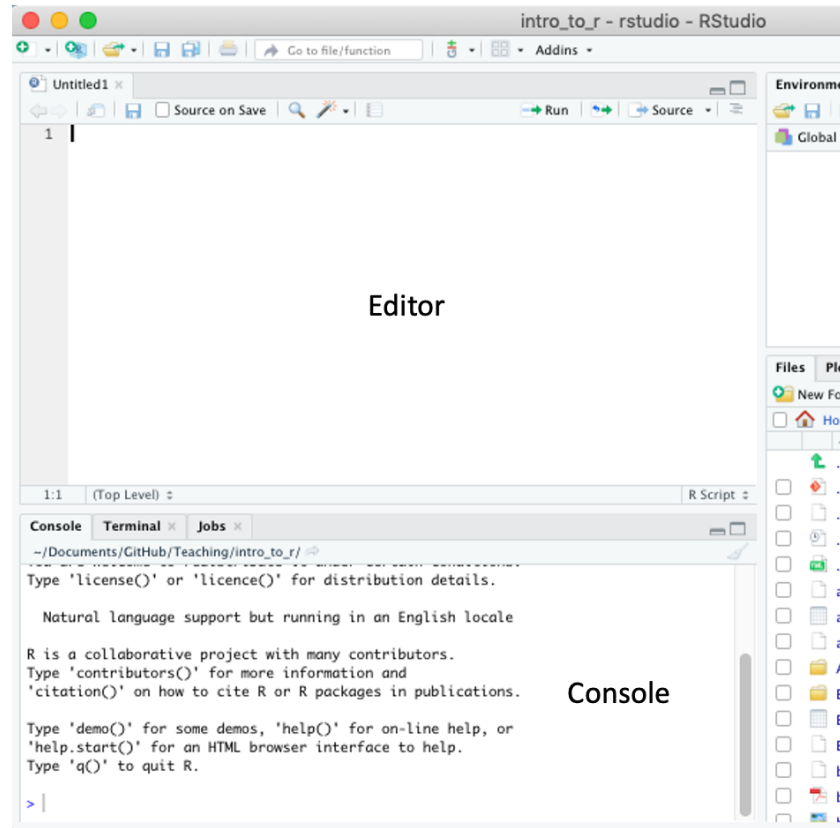
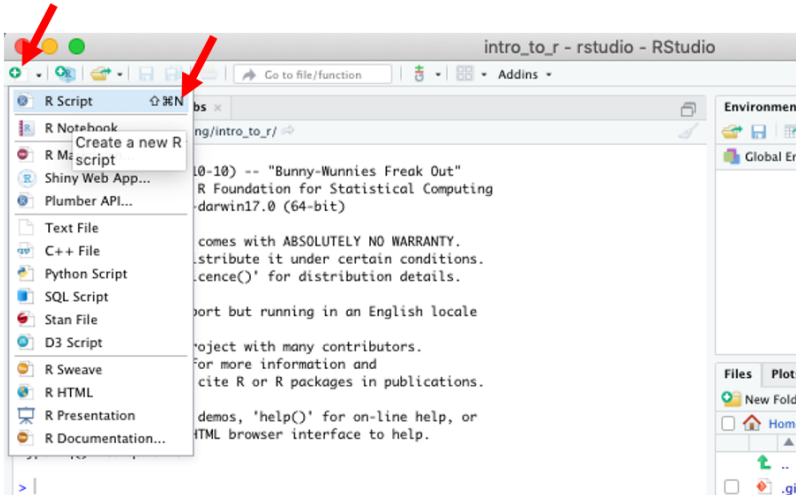
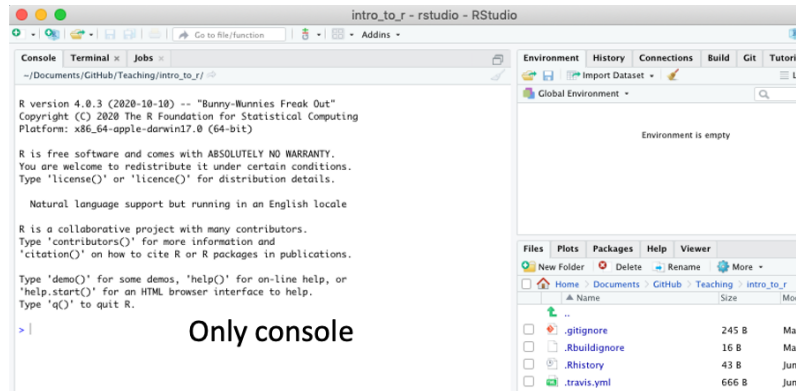
More information

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

RStudio



Getting the editor



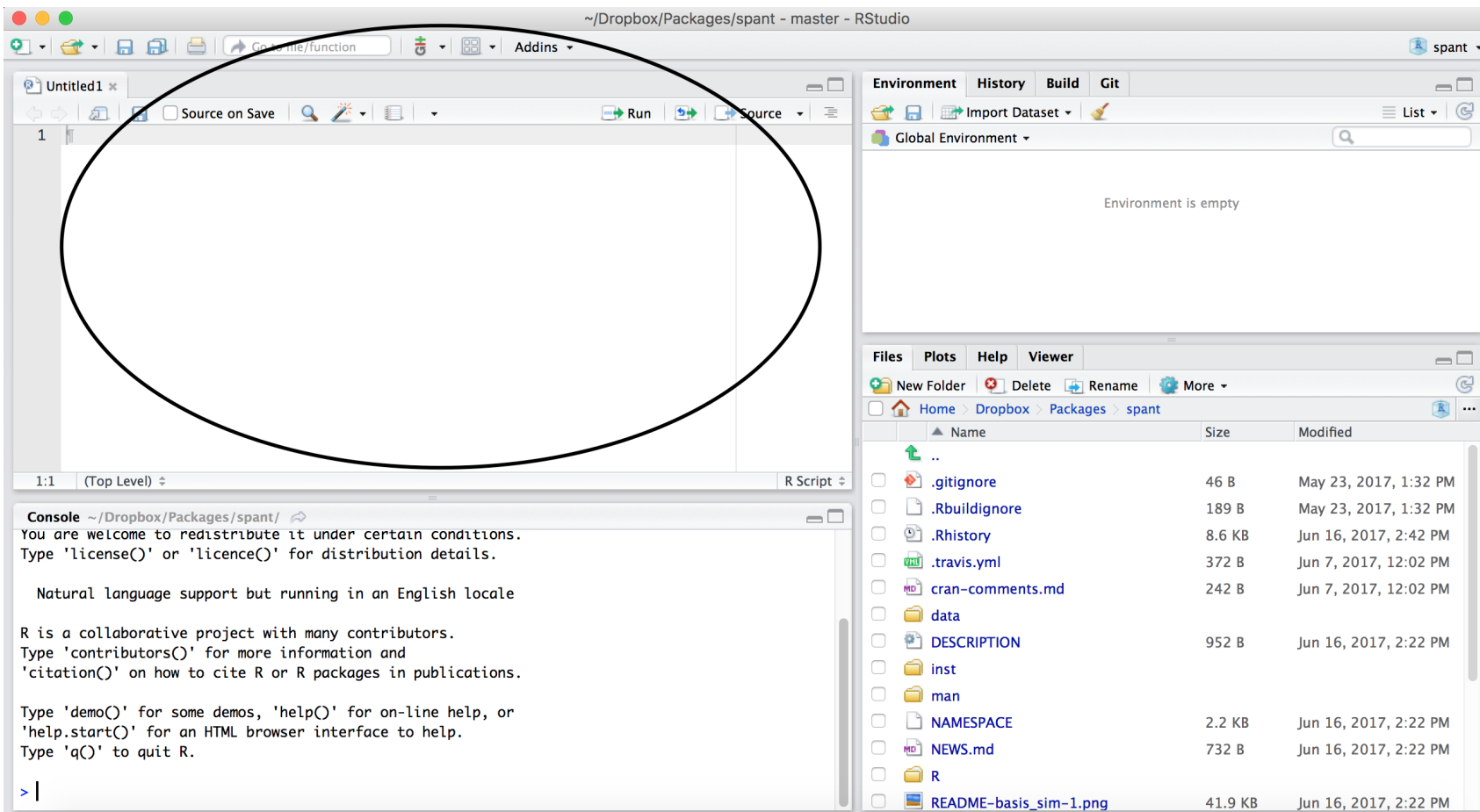
Working with R in R Studio - 2 major panes:

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

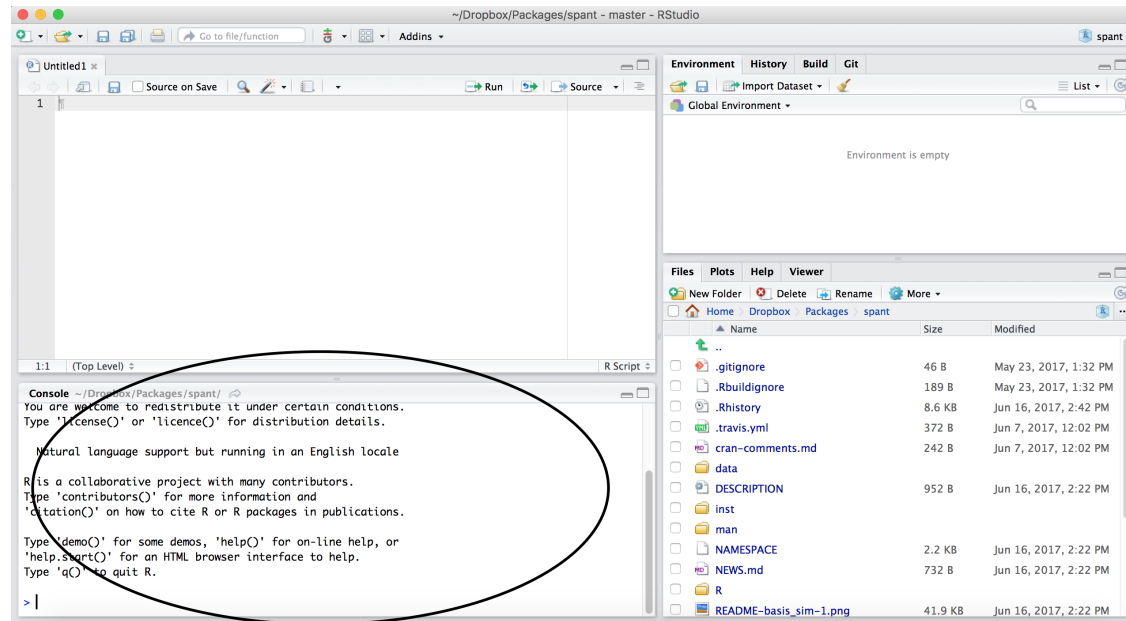
Source / Editor

- 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
- Code is **not saved** on your disk

RStudio

Super useful “cheat sheet”:

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

Write Code

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

R Support

Import data with wizard
History of past commands to run/copy
Display .RPres slideshows
File > New File > R Presentation

The screenshot shows the RStudio IDE interface with several panels and annotations:

- Source Editor:** Contains R code with annotations for cursors, multiple cursor selection (**Alt + mouse drag**), code diagnostics in the margin, syntax highlighting, tab completion, multi-language code snippets, and jumping to functions.
- Environment Panel:** Shows the Global Environment with objects like 'iris' and 'foo'. Annotations include 'Load workspace', 'Save workspace', 'Delete all saved objects', 'Search inside environment', and 'Display objects as list or grid'.
- Files Panel:** Shows the file browser with annotations for 'Create folder', 'Upload file', 'Delete file', 'Rename file', 'Set As Working Directory', 'Go To Working Directory', and 'Change directory'.
- Console:** Shows the command history with annotations for 'Working Directory' and 'Press ↑ to see command history'.
- Other Annotations:** 'Maximize, minimize panes', 'Drag pane boundaries', 'Path to displayed directory', and 'A file browser keyed to your working directory. Click on file or directory name to open.'

RStudio layout

The screenshot displays the RStudio IDE interface with the title bar "intro_to_r - rstudio - RStudio". The main editor window shows a new R Markdown document titled "Untitled1" with the following 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
13 authoring HTML, PDF, and MS Word documents. For more details on using R
14 Markdown see <http://rmarkdown.rstudio.com>.
15
16 When you click the Knit button a document will be generated that includes
17 both content as well as the output of any embedded R code chunks within the
18 document. You can embed an R code chunk like this:
19
20 ```{r cars}
21
```

The right-hand pane contains the "Environment" tab, which shows "Global Environment" and "Environment is empty". Below it is the "Files" pane, which shows the file explorer for the directory "Home > Documents > GitHub > Teaching > intro_to_r". The file list includes:

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

The bottom pane shows the "Console" tab with the following output:

```
~/Documents/GitHub/Teaching/intro_to_r/
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

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.

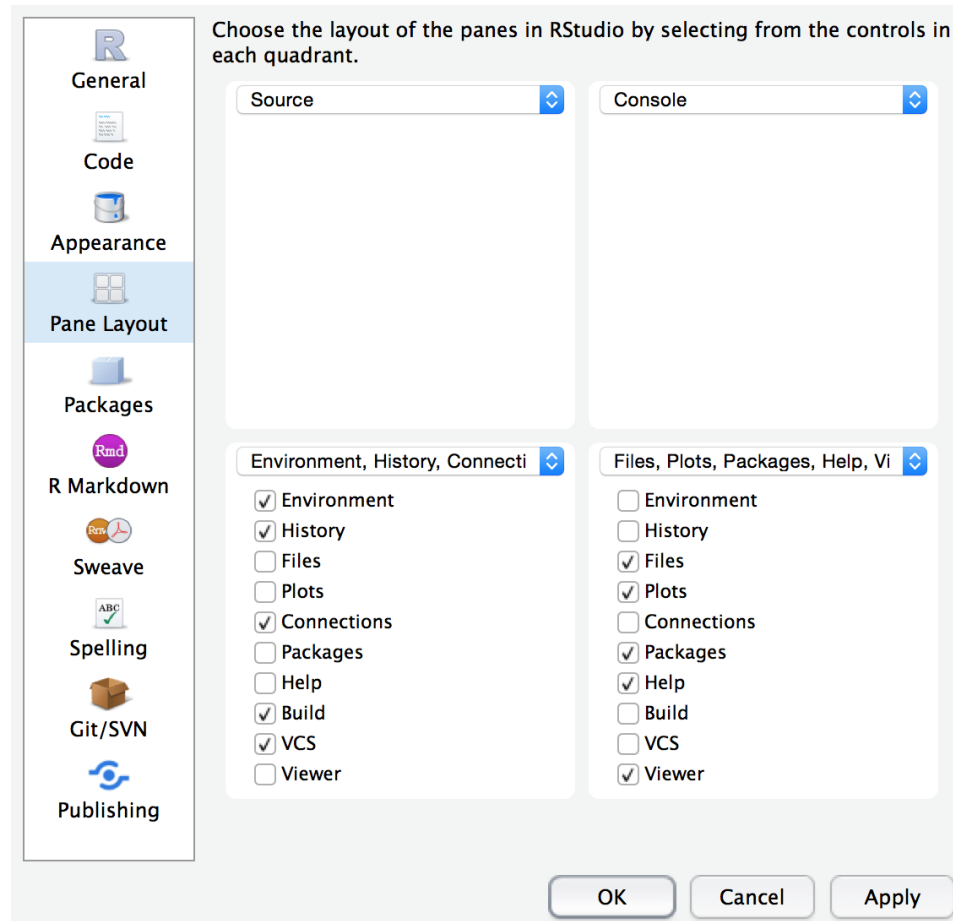
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

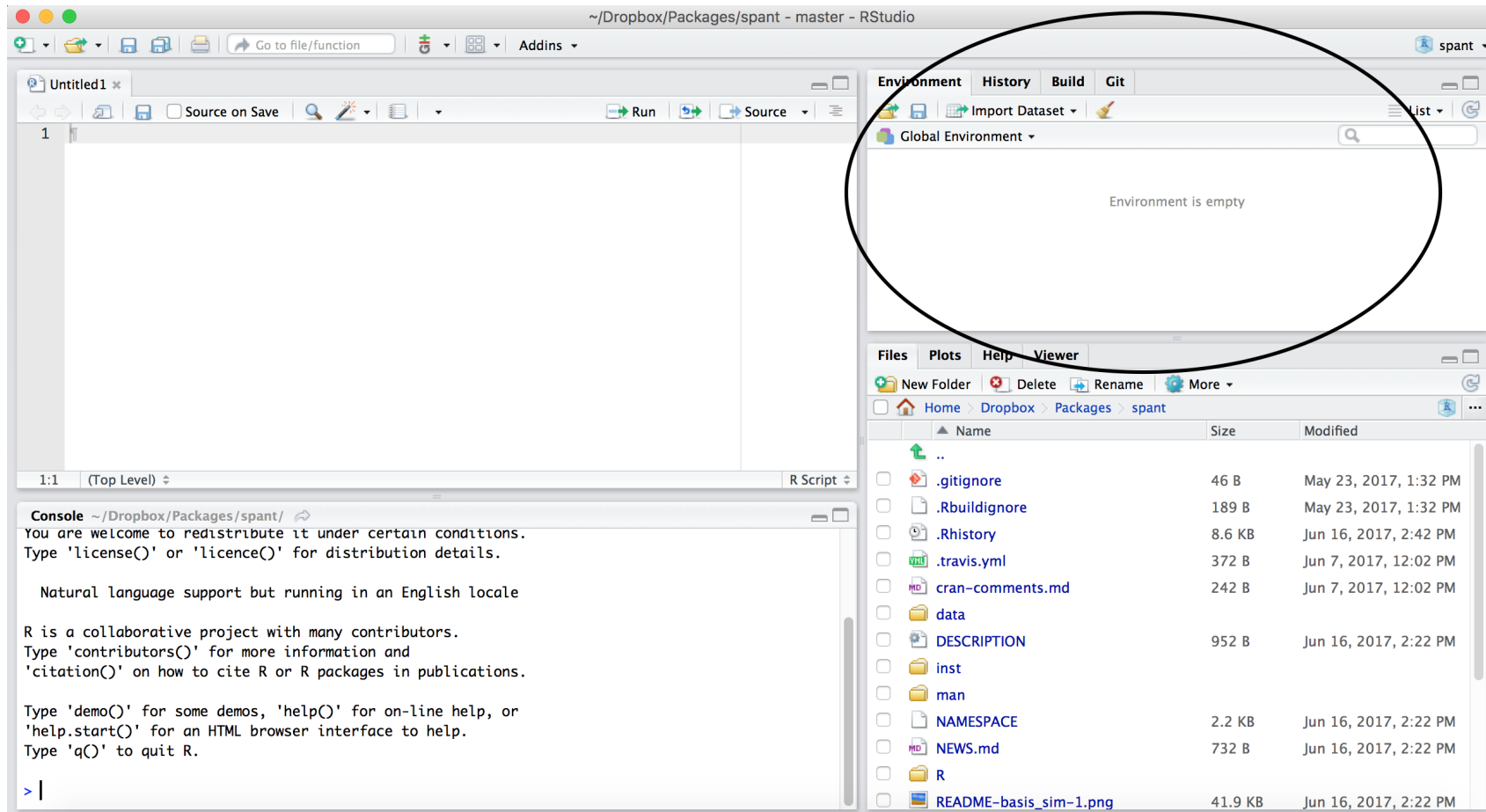
RStudio Layout

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

RStudio -> Preferences -> Pane Layout



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 of 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:

1. Download the file at the link above by clicking on the link or typing in:
https://jhudatascience.org/intro_to_r/modules/RStudio/lab/RStudio_L
(Also on the [website](#) schedule page - Lab for day 1)
2. Find the downloaded file on your computer
3. Open the file in RStudio

This may require finding your downloads on your computer.

Recall that these videos can help:

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

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

R Markdown file

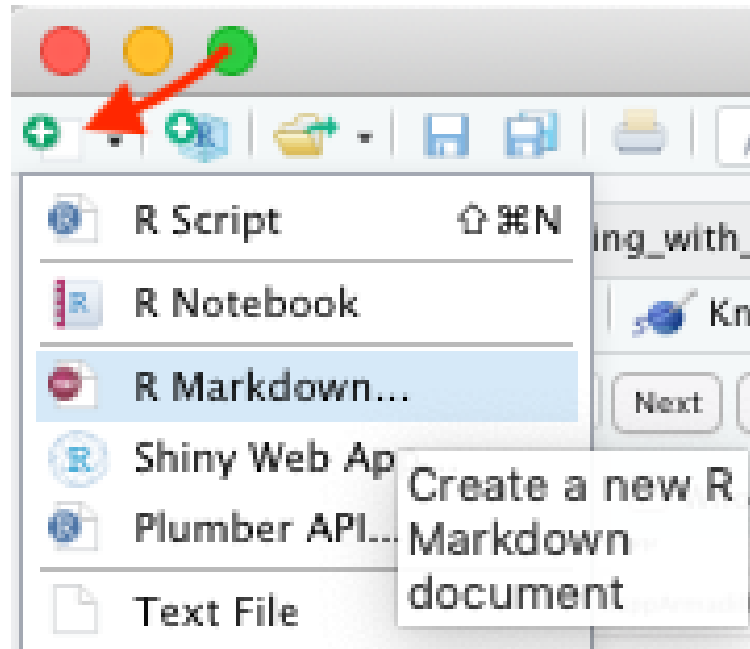
R Markdown files (.Rmd) help generate reports that include your code and output. Think of them as fancier scripts.

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

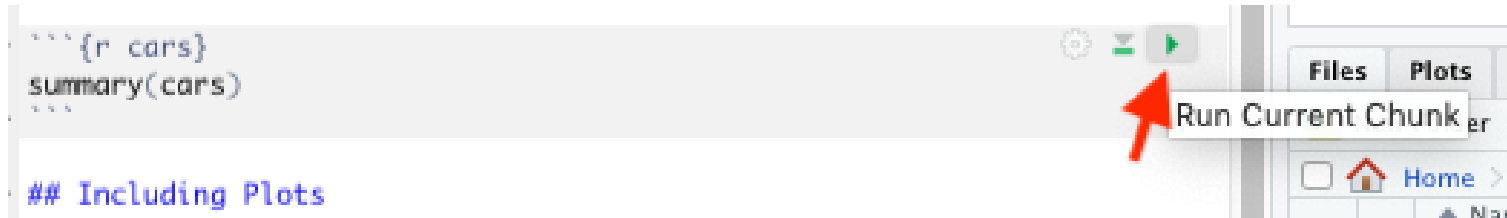
Call your file “first_markdown”



Code chunks

Within R Markdown files are code “chunks”

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

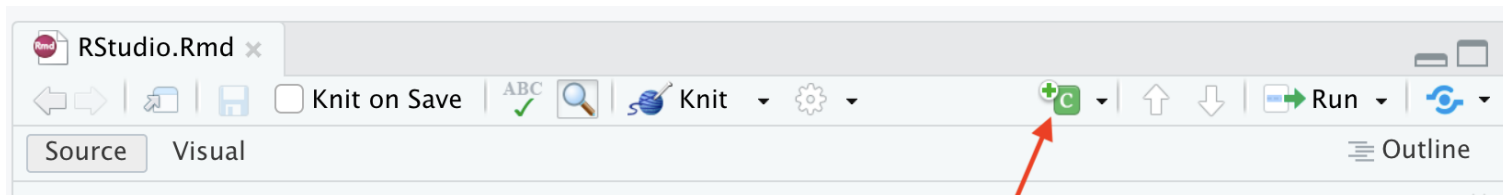


Create Chunks

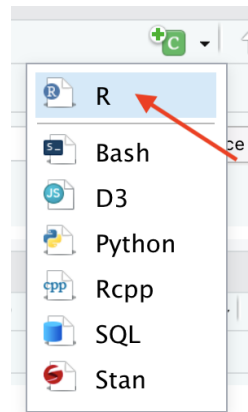
To create a new R code chunk:

Copy paste an existing chunk in the R Markdown file and replace the code **OR**

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



1. Select R as the language:



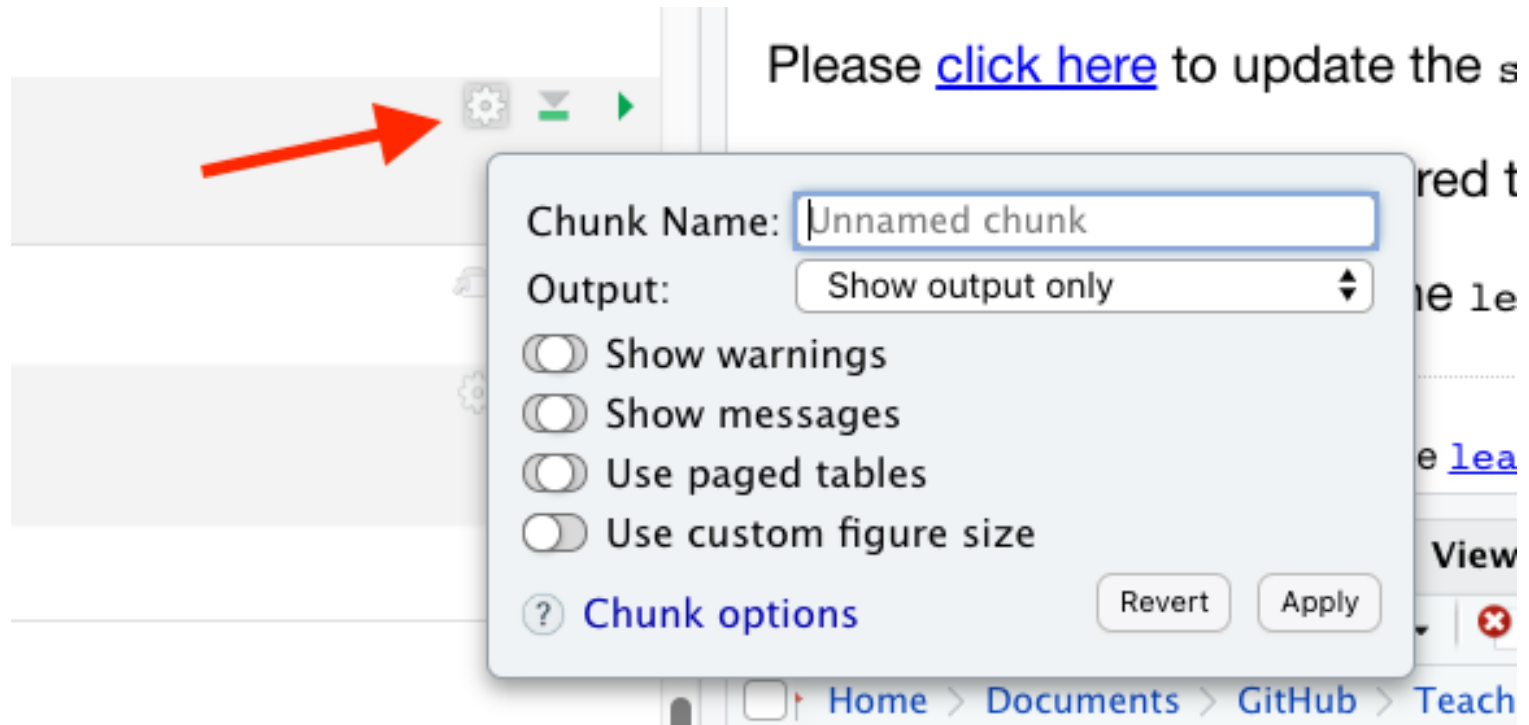
Run previous chunks button

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

```
```{r, out.width = "80%", echo = FALSE, fig.align='center'}  
knitr::include_graphics("images/chunk.png")
```
```

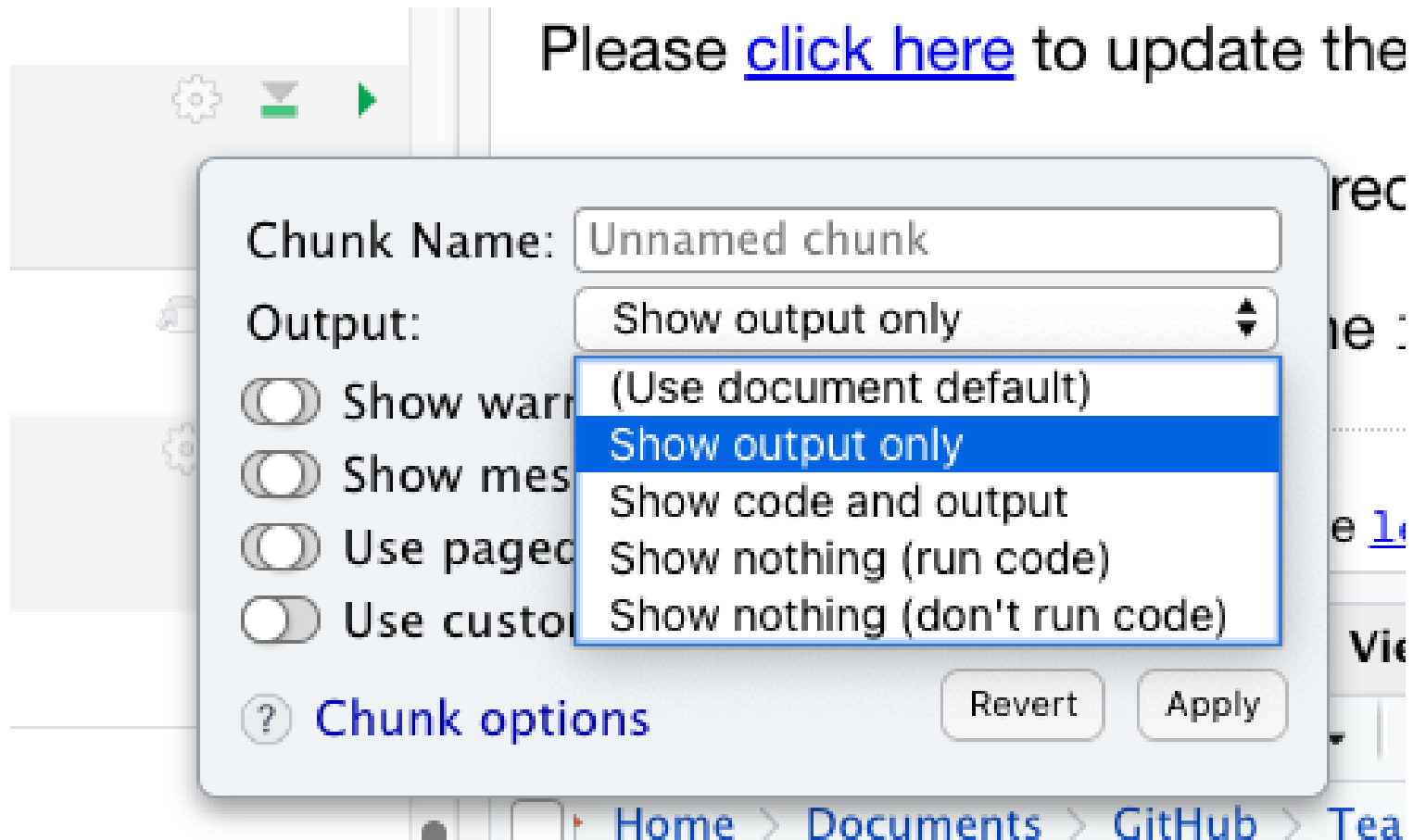


Chunk settings



Chunk settings

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



Knit file to html

This will create a report from the R Markdown document!

Useful R Studio Shortcuts

- `Ctrl + Enter` 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` takes you to the script page
- `Ctrl+2` takes you to the console
- http://www.rstudio.com/ide/docs/using/keyboard_shortcuts

Summary

- RStudio makes working in R easier
- the Editor is for static code like scripts or R Markdown documents
- The console is for testing code
- R markdown documents are really helpful for lots of reasons!
- R code goes within what is called a chunk
- Code chunks can be modified so that they show differently in reports

□ [Class Website](#)

□ [Lab](#)