

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

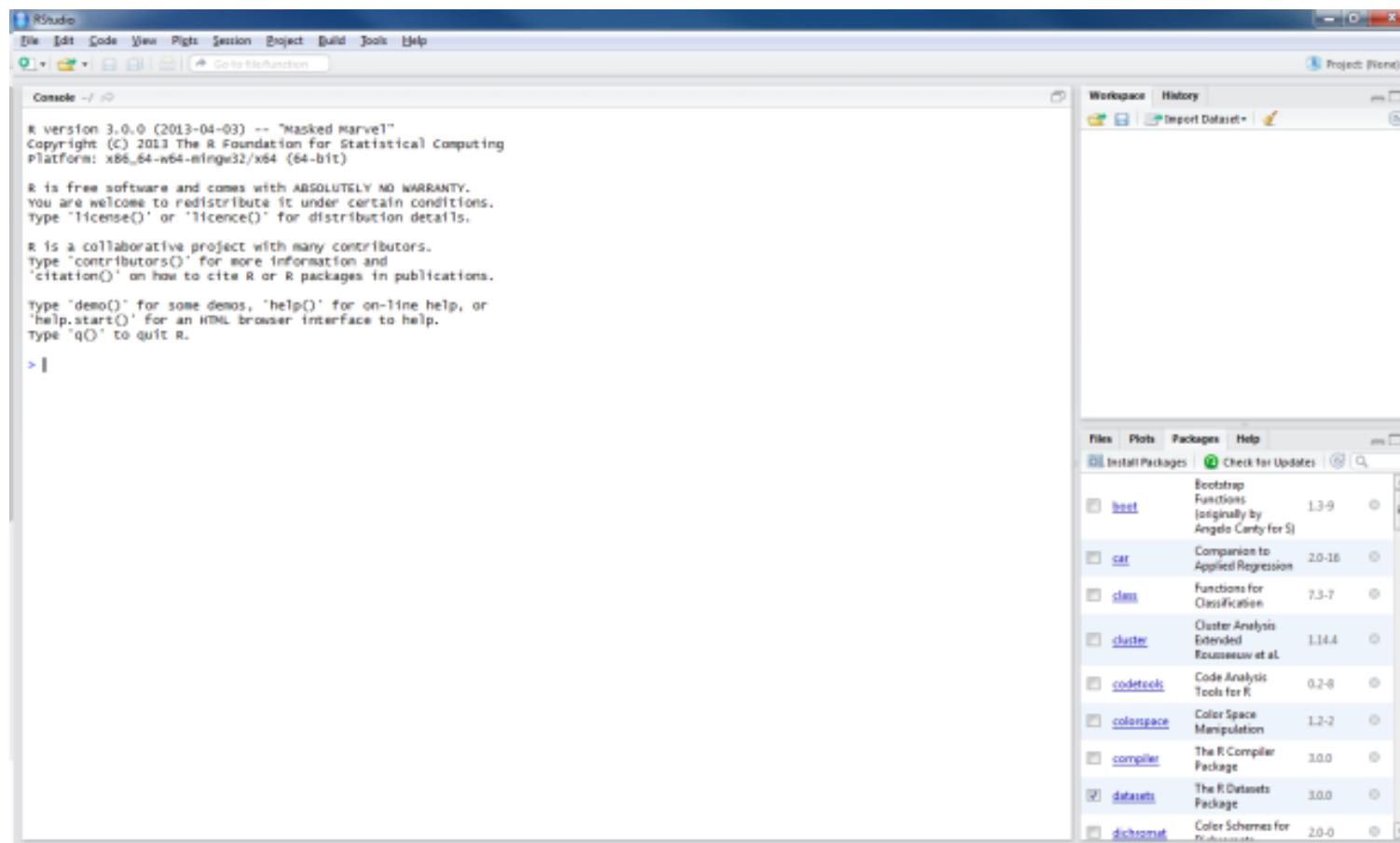
Easier working with R

- 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 PDF export
- Integrated R help and documentation
- Searchable command history

RStudio



Getting the editor

The image contains three side-by-side screenshots of the RStudio interface, each illustrating a different method to access the code editor.

- Only console:** This screenshot shows the RStudio interface with the "Console" tab selected. The title bar says "intro_to_r - rstudio - RStudio". The main area displays the R startup message and help text. The "Files" tab is visible at the bottom.
- Editor:** This screenshot shows the RStudio interface with the "Editor" tab selected. The title bar says "intro_to_r - rstudio - RStudio". The main area is empty, showing the text "1 |". The "Files" tab is visible at the bottom.
- Console:** This screenshot shows the RStudio interface with the "Console" tab selected. The title bar says "intro_to_r - rstudio - RStudio". The main area displays the R startup message and help text. The "Files" tab is visible at the bottom.

In the bottom-left screenshot, two red arrows point to the "New File" icons in the toolbar: one for an "R Script" and another for a "Text File".

Working with R in R Studio - 2 major panes:

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

- Static copy of what you did (reproducibility)
- Try things out interactively, then add to your script

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

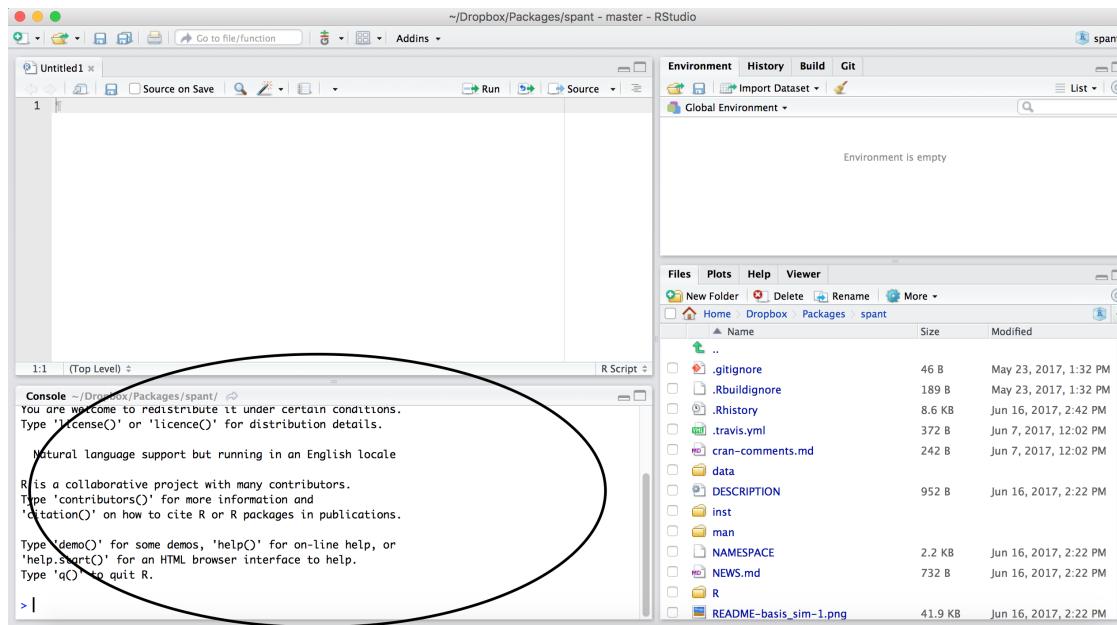
- Calculator
- Creating variables
- Applying functions

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

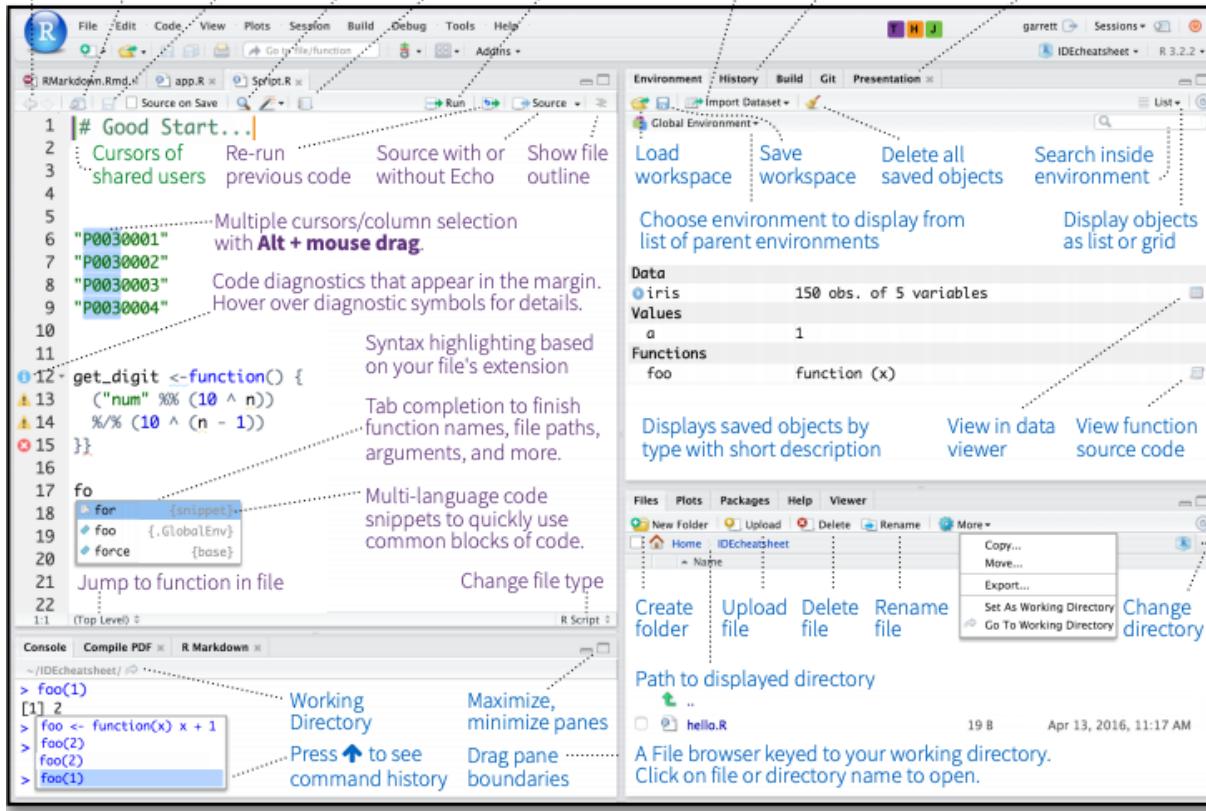
RStudio

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



More on Functions and Packages

- R revolves around **functions**
 - Commands that take input, performs computations, and returns results
- Functions are enclosed in **packages**
 - When you download R, it has a “base” set of functions/packages (**base R**)
 - You can install additional packages for your uses from [CRAN](#) or [GitHub](#)
 - These additional packages are written by RStudio or R users/developers (like us)
 - Think of them as “R Extensions”



Using Packages

- It helps to be somewhat familiar with base R - answers on Google commonly use it
- We will focus on newer and **more intuitive** ways to do things (tidyverse), not in base R
- RStudio (the company) makes a lot of great packages
- Not all packages available on CRAN or GitHub are trustworthy
- Who wrote it? **Hadley Wickham** is a major authority on R (Employee and Developer at RStudio)
- How to trust an R package: <http://simplystatistics.org/2015/11/06/how-i-decide-when-to-trust-an-r-package/>



(source: <https://twitter.com/hadleywickham>)

Let's take a look at R Studio
ourselves!

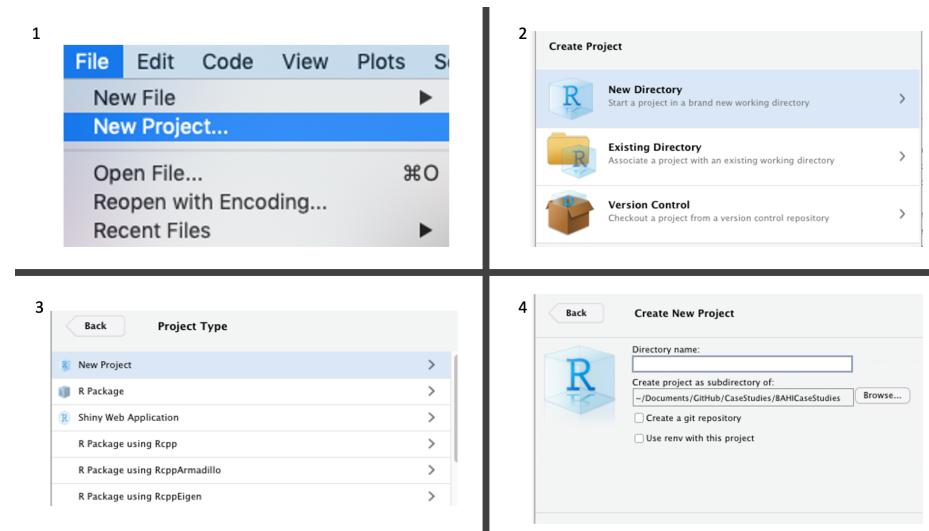
RStudio

Let's start by making an RStudio "Project".

1. Helps you organize your work.
2. Helps with working directories (discussed later).
3. Allows you to easily know which project you're on.

Go to File → New Project → New Directory → New Project

Call your Project "Intro_to_R"



R Markdown file

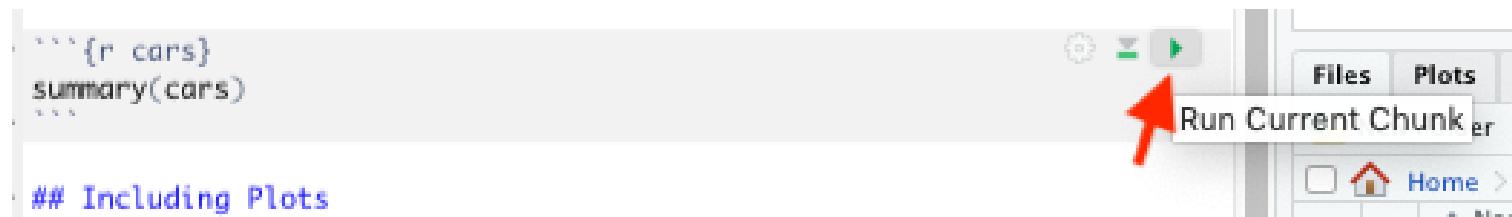
R Markdown files (.Rmd) help generate reports that include your code and output. Think of them as fancier scripts.

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

Code chunks

Within R Markdown files are code “chunks”

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

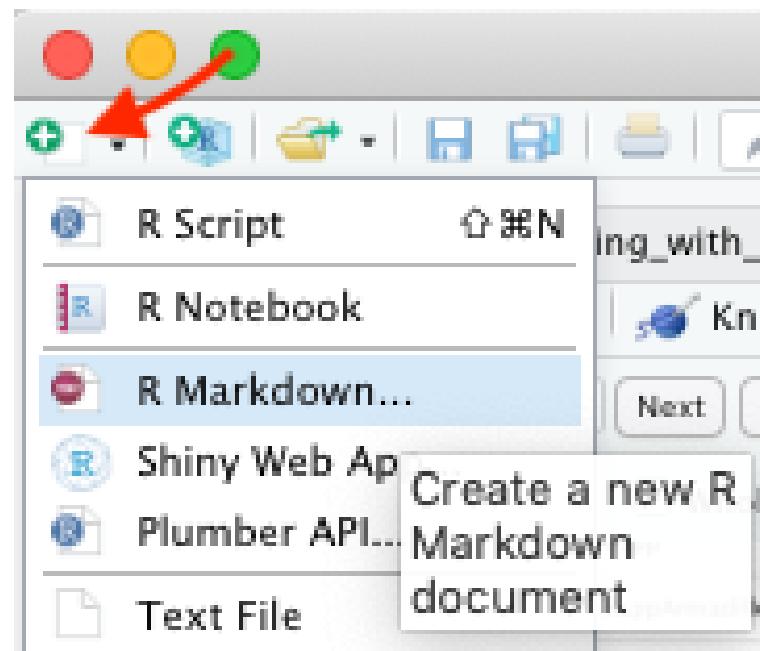


Knit

Create an R Markdown file

Go to File → New File → R Markdown

Call your file “first_markdown”



RStudio layout

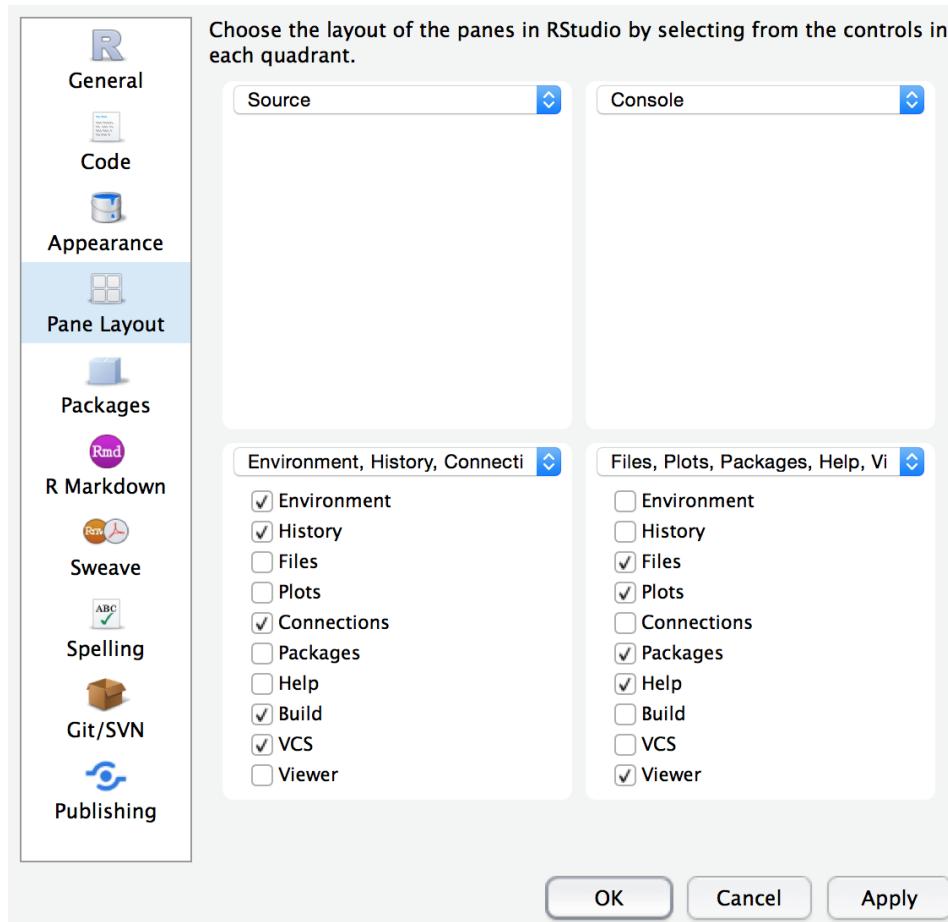
The screenshot displays the RStudio desktop application window with the following layout:

- Code Editor (Left Panel):** Shows an R Markdown document titled "first_markdown". The code includes setup chunks for knitr, a global environment message, and a code chunk illustrating R Markdown syntax.
- Environment (Top Right Panel):** Shows the Global Environment pane with the message "Environment is empty".
- Files (Bottom Right Panel):** Shows a file browser with the path: Home > Documents > GitHub > Teaching > intro_to_r. The contents include a .gitignore file and several CSV and TXT files related to arrays and basic R functions.
- Console (Bottom Left Panel):** Displays the R startup message, license information, natural language support, collaborative project details, demo instructions, and a prompt for quitting R.

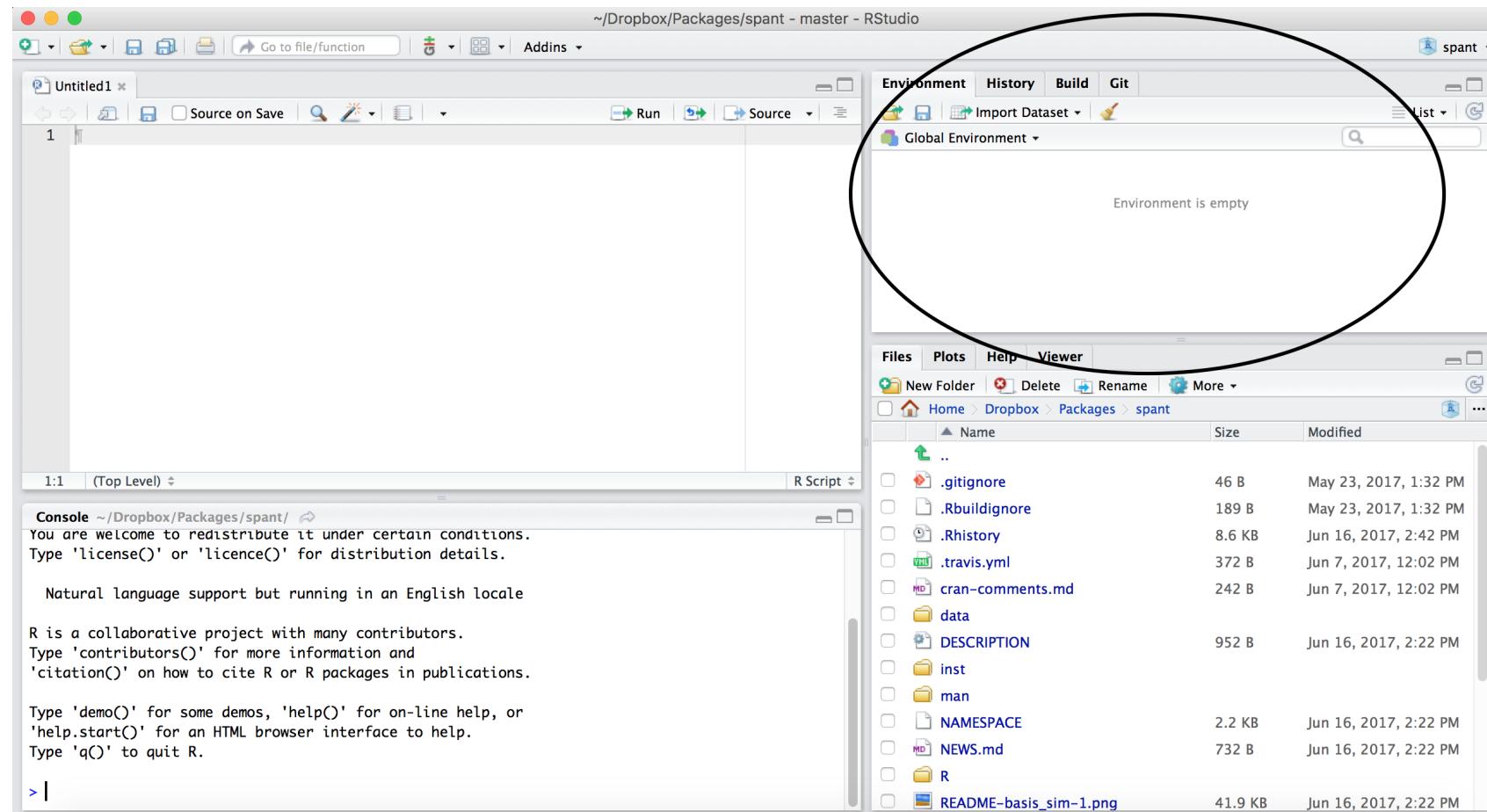
RStudio Layout

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 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

Useful R Studio Shortcuts

- `Ctrl + Enter` (`Cmd + Enter` on OS X) 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

Viewing data

The `View` command allows you to view data in a spreadsheet format.

```
View(mtcars)
```

```
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

```
tail(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.7	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.9	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.5	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.5	0	1	5	6

Lab: Starting with R and RMarkdown

Starting with R

Website

Website