

Getting started with R and RStudio





The engine



The dashboard

A tour of RStudio

~/Dropbox/Teaching/Workshops/2020-gpl/01_welcome-tidyverse - RStudio

Go to file/function

Addins

01_welcome-tidyverse

Console Terminal x Jobs x

~/Dropbox/Teaching/Workshops/2020-gpl/01_welcome-tidyverse/

```
R version 4.0.0 (2020-04-24) -- "Arbor Day"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
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.

> |
```

Environment History Connections Tutorial

Import Dataset

Global Environment

Environment is empty

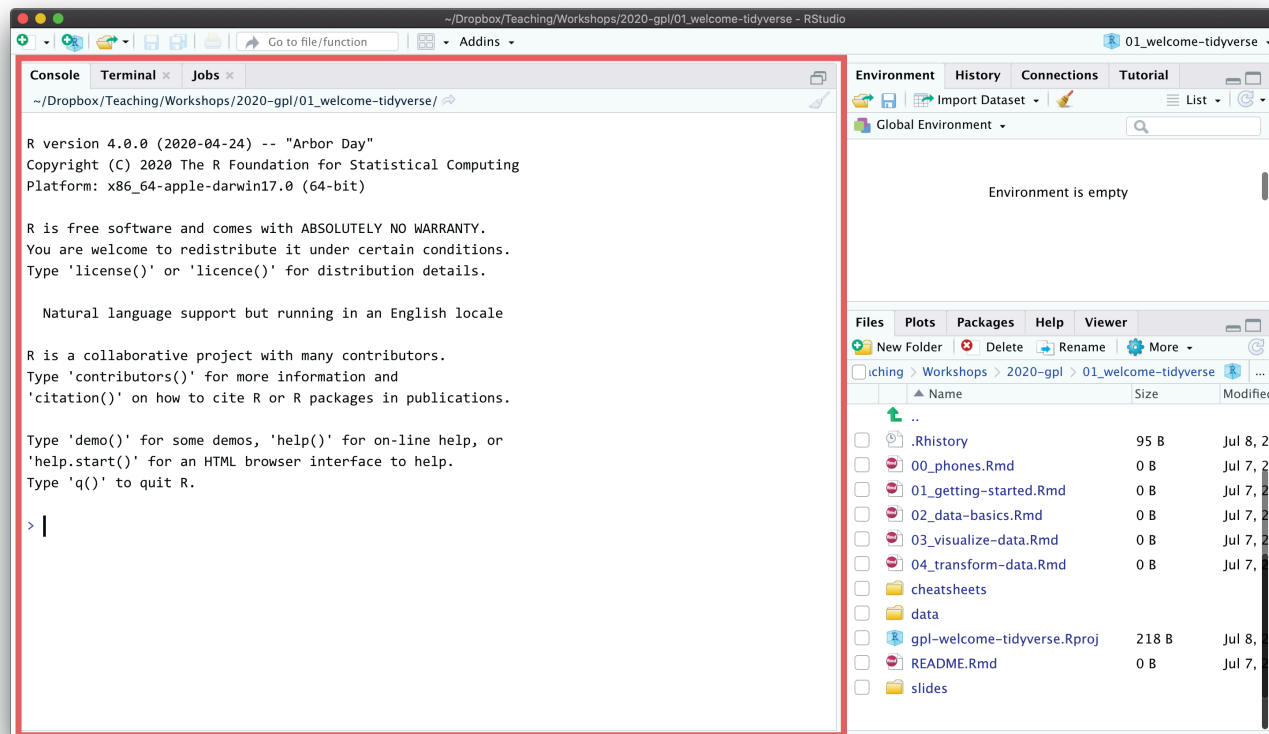
Files Plots Packages Help Viewer

New Folder Delete Rename More

iching > Workshops > 2020-gpl > 01_welcome-tidyverse

	Name	Size	Modified
↑	..		
☐	.Rhistory	95 B	Jul 8, 2
☐	00_phones.Rmd	0 B	Jul 7, 2
☐	01_getting-started.Rmd	0 B	Jul 7, 2
☐	02_data-basics.Rmd	0 B	Jul 7, 2
☐	03_visualize-data.Rmd	0 B	Jul 7, 2
☐	04_transform-data.Rmd	0 B	Jul 7, 2
☐	cheatsheets		
☐	data		
☐	gpl-welcome-tidyverse.Rproj	218 B	Jul 8, 2
☐	README.Rmd	0 B	Jul 7, 2
☐	slides		

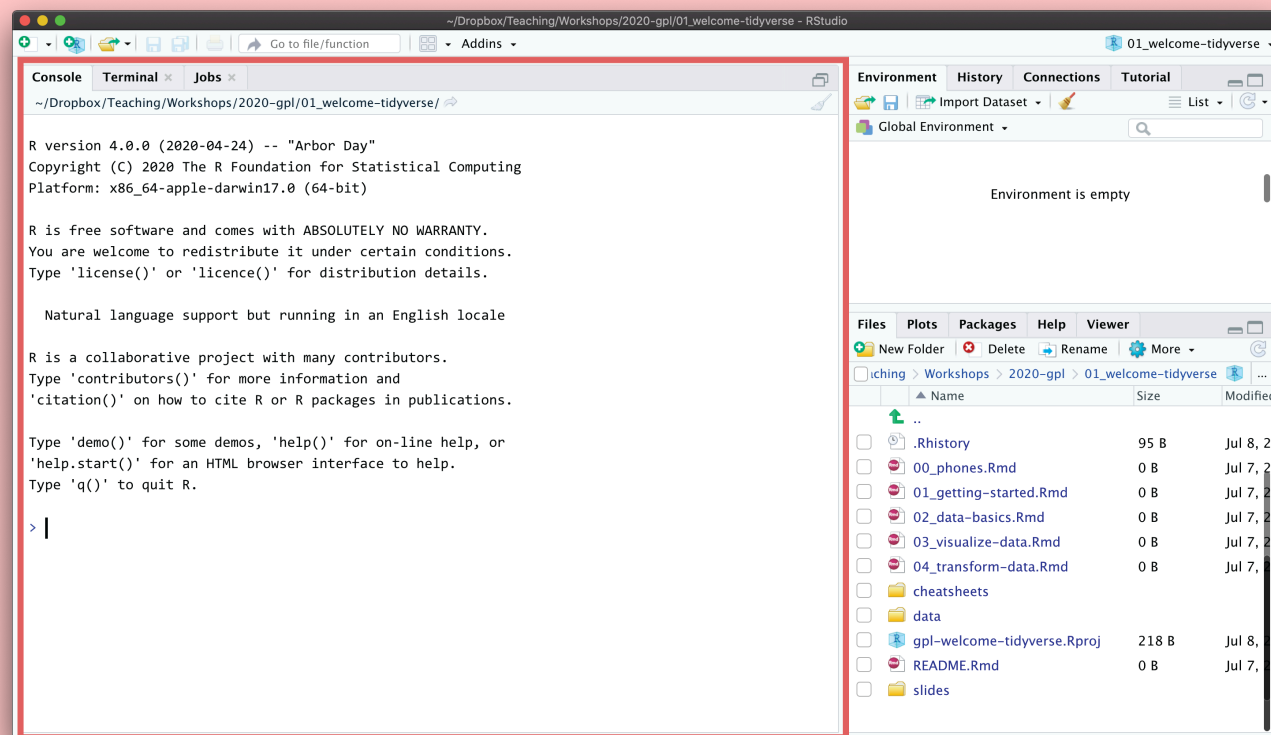
Console



R is awaiting your instructions

Type code here, press enter, and R will run it

Your turn



Type `2 + 2` in the
console

Press enter

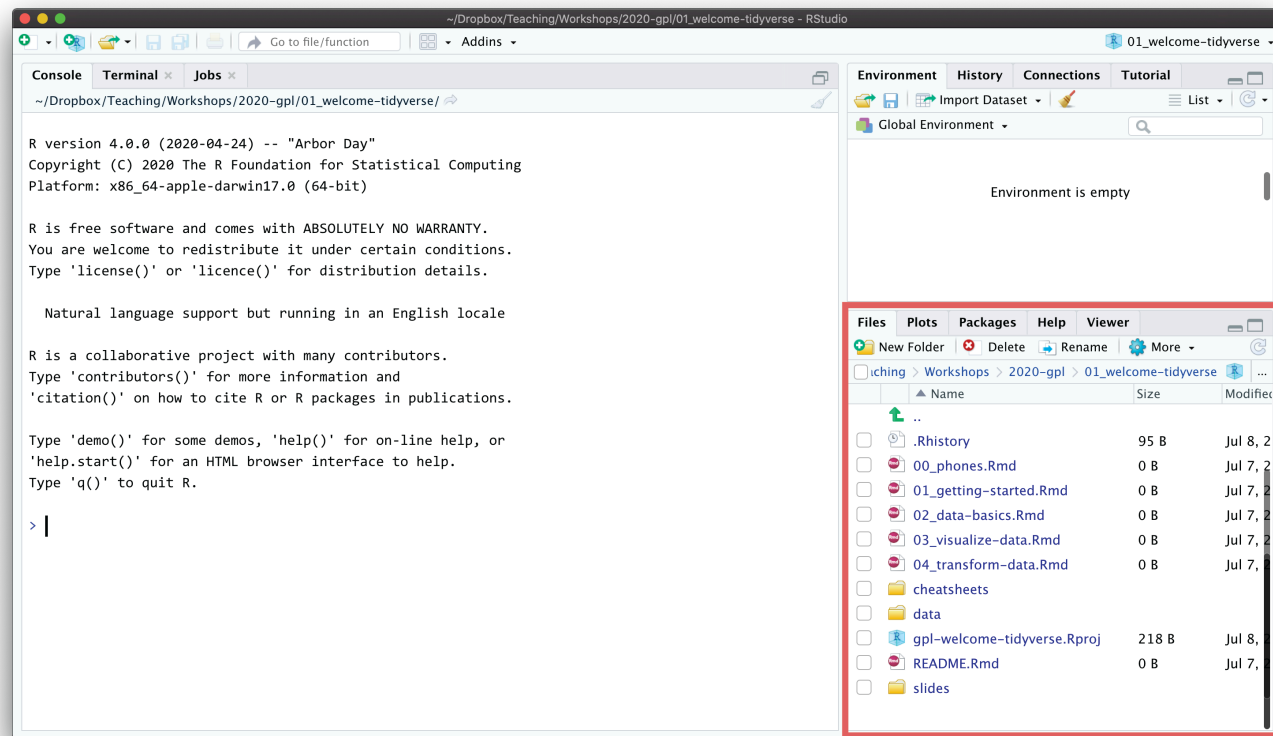
```
2 + 2
```

```
## [1] 4
```

**This is ephemeral though.
If you want to run this again, you'll have to type it again.**

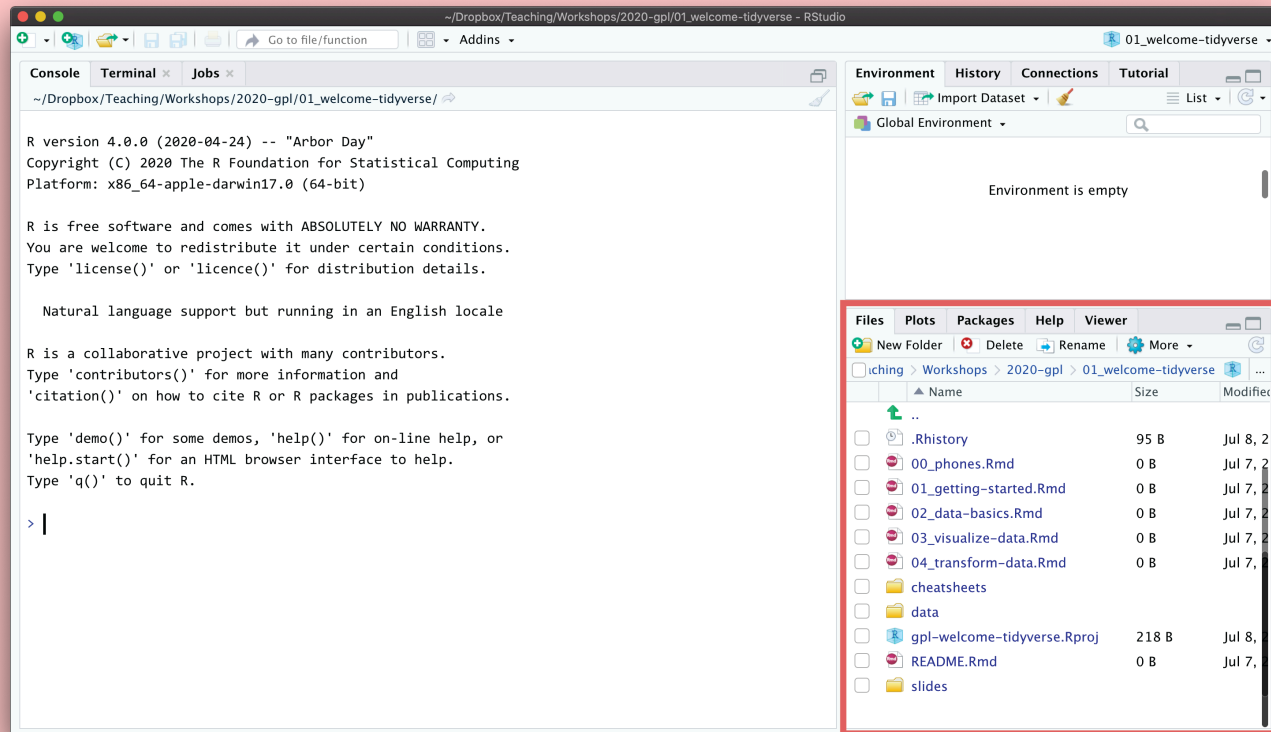
Store R code in a document instead

Files pane



All the files in your
current working
directory

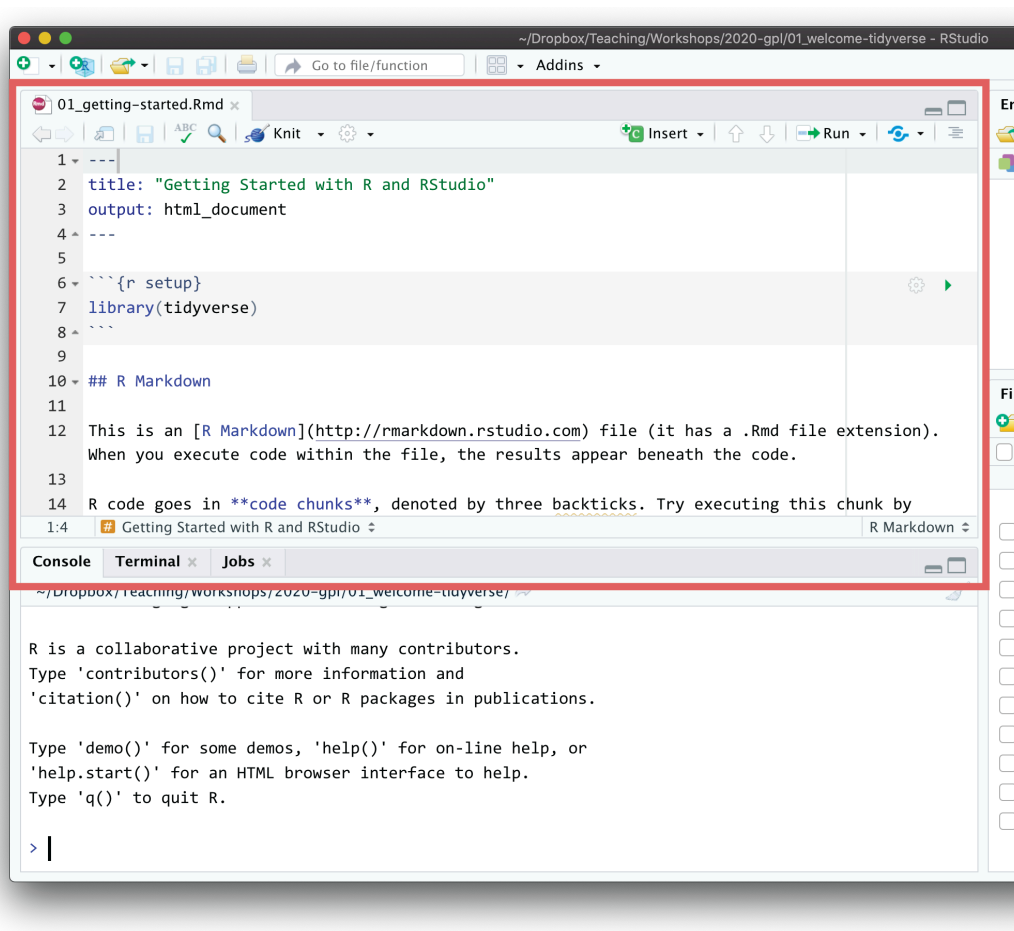
Your turn



Find 01_getting-started.qmd

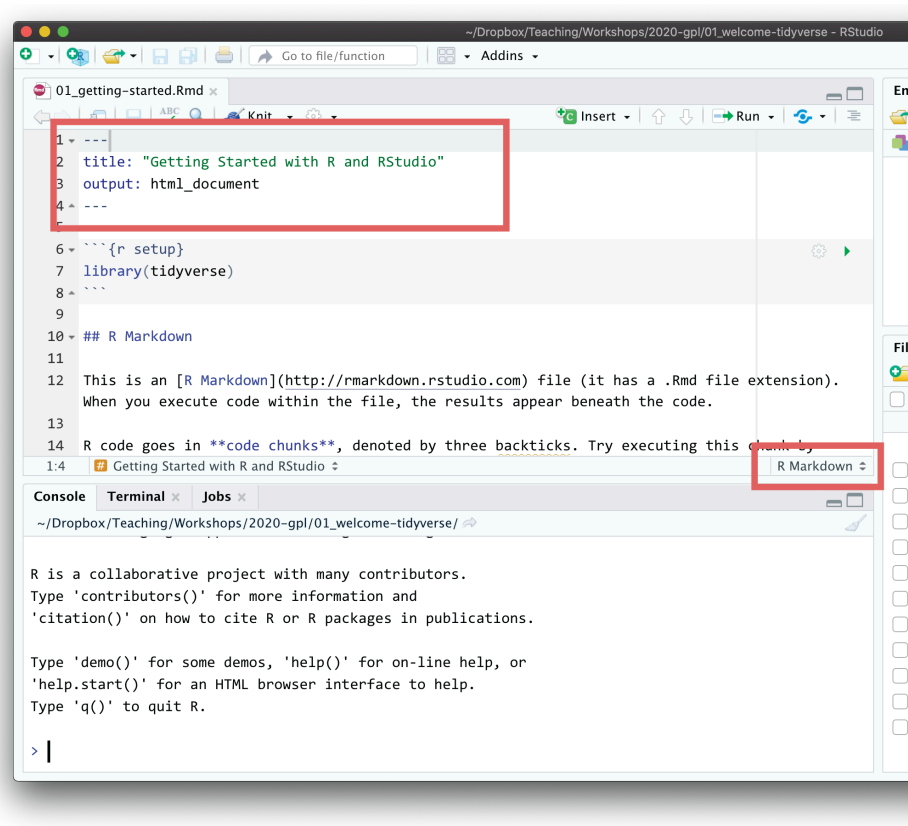
Click on its name to open the file

Source pane



Documents
open here

Quarto



Document format that
combines text and code

Acts like a notebook
for your analysis

Quarto



Text

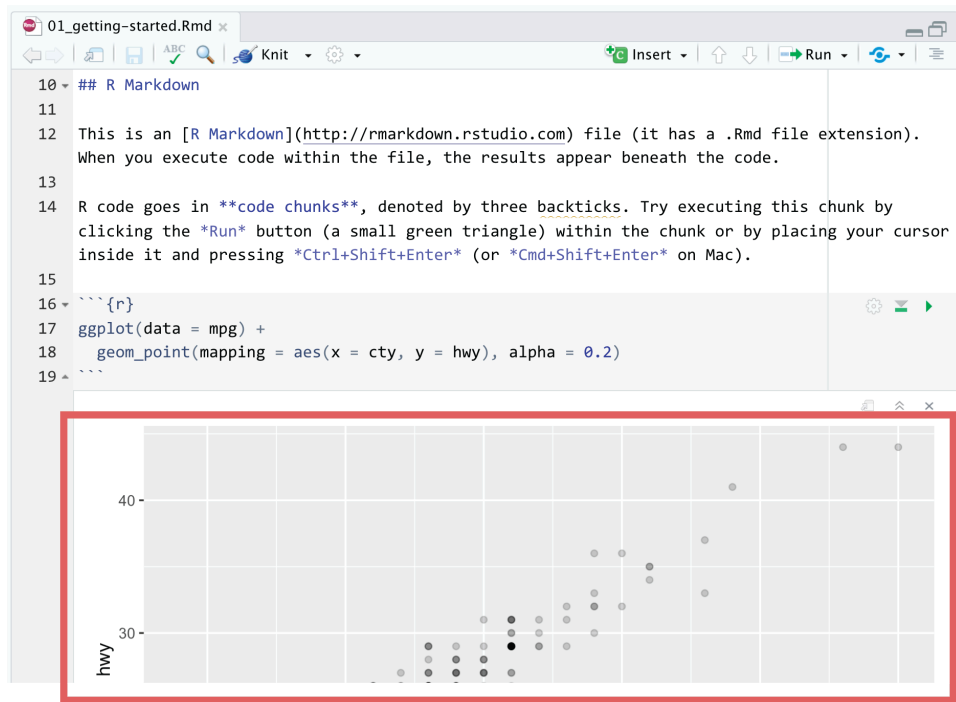
Quarto



Text

Code

Quarto



Text

Code

Output

Your turn

Read the instructions

Run the code chunk by clicking the play button



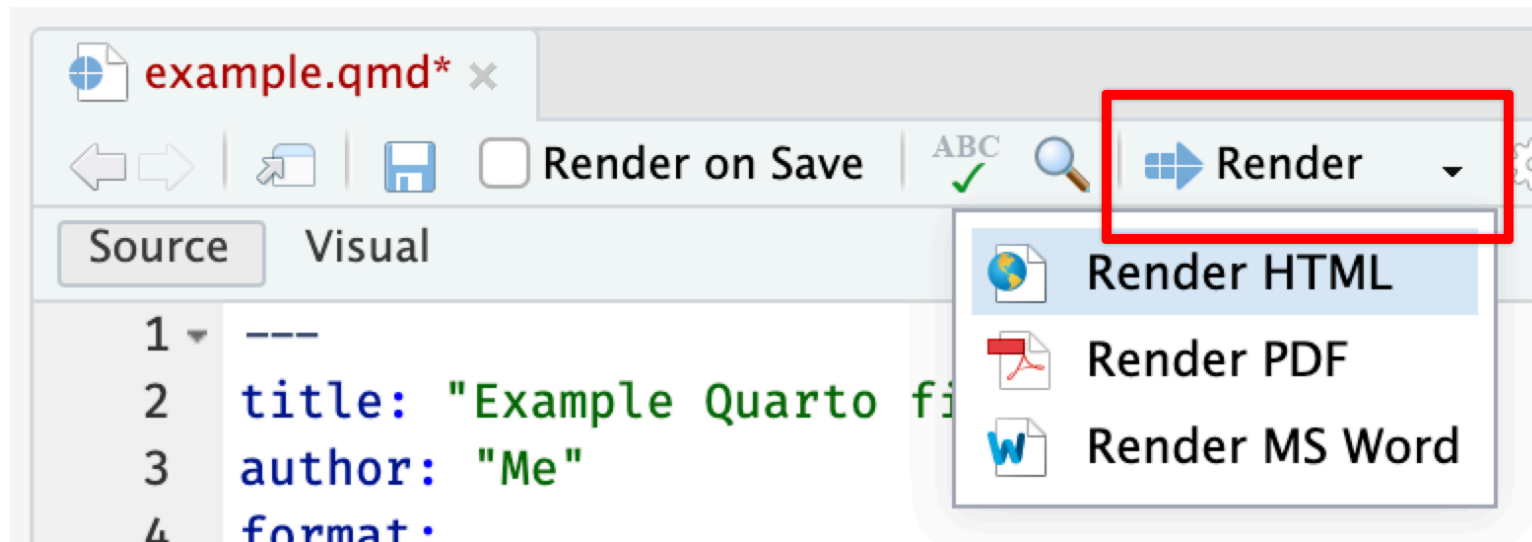
Your turn

Add a new chunk

Put $2 + 2$ in the chunk and run it

Render

Render a Quarto document into a standalone shareable file



Quarto

The best way to combine R code and narrative

We'll use it throughout the class:

I'll provide starter code

You'll complete "Your turns"

In the end, you'll have an annotated record for yourself

Your turn

Spot the difference:

```
filter(mtcars, cyl == 4)
```

```
four_cyls <- filter(mtcars, cyl == 4)
```

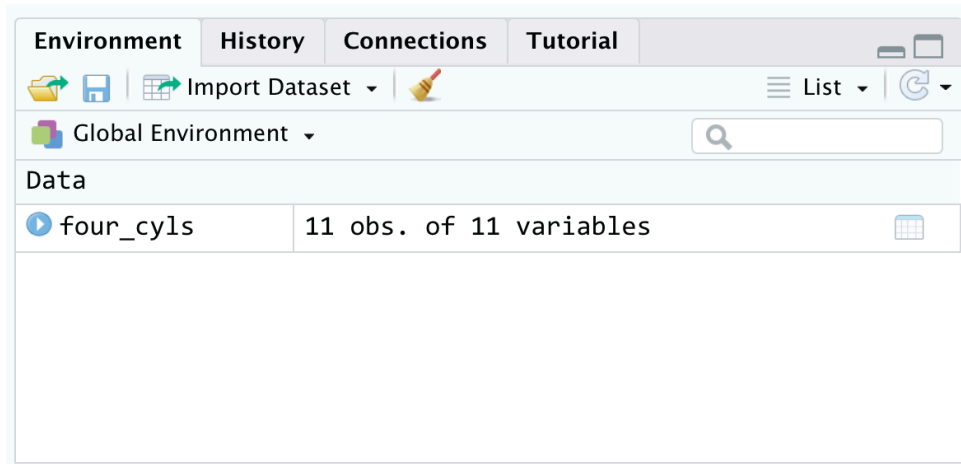
**Find these chunks in the notebook and run them.
What's different about what happens?**

Assignment

<- assigns the output from the righthand side to a variable with the name on the lefthand side

```
four_cyls <- filter(mtcars, cyl == 4)
```

Environment pane



**List of all the
variables you've created**

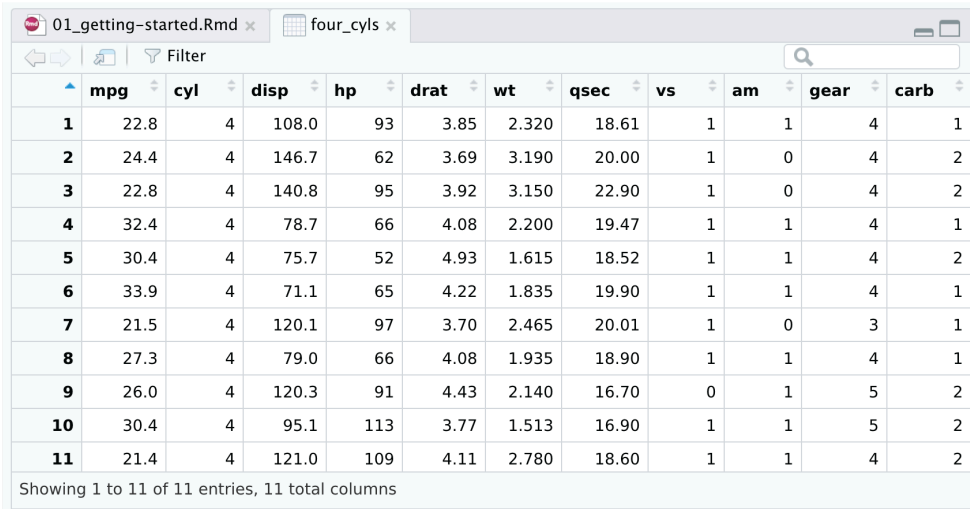
Your turn

Find `four_cyls` in the environment pane.
Click on the name `four_cyls`

What happens?

Viewer

Clicking on an object in the environment panel opens it an interactive viewer tab



The screenshot shows the RStudio interface with a tab titled '01_getting-started.Rmd' and a sub-tab 'four_cyls'. Below the tabs is a toolbar with navigation icons and a 'Filter' search bar. The main area displays a data table with 11 rows and 12 columns. The columns are: an index (1-11), 'mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am', 'gear', and 'carb'. Each column header has a small upward arrow icon. The data rows contain numerical values for each variable. At the bottom of the table, a status bar reads 'Showing 1 to 11 of 11 entries, 11 total columns'.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
2	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
3	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
4	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
5	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
6	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
7	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
8	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
9	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
10	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
11	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Showing 1 to 11 of 11 entries, 11 total columns

Functions

Functions do things

Functions take arguments, output results

If you want to keep the output, assign it to a variable

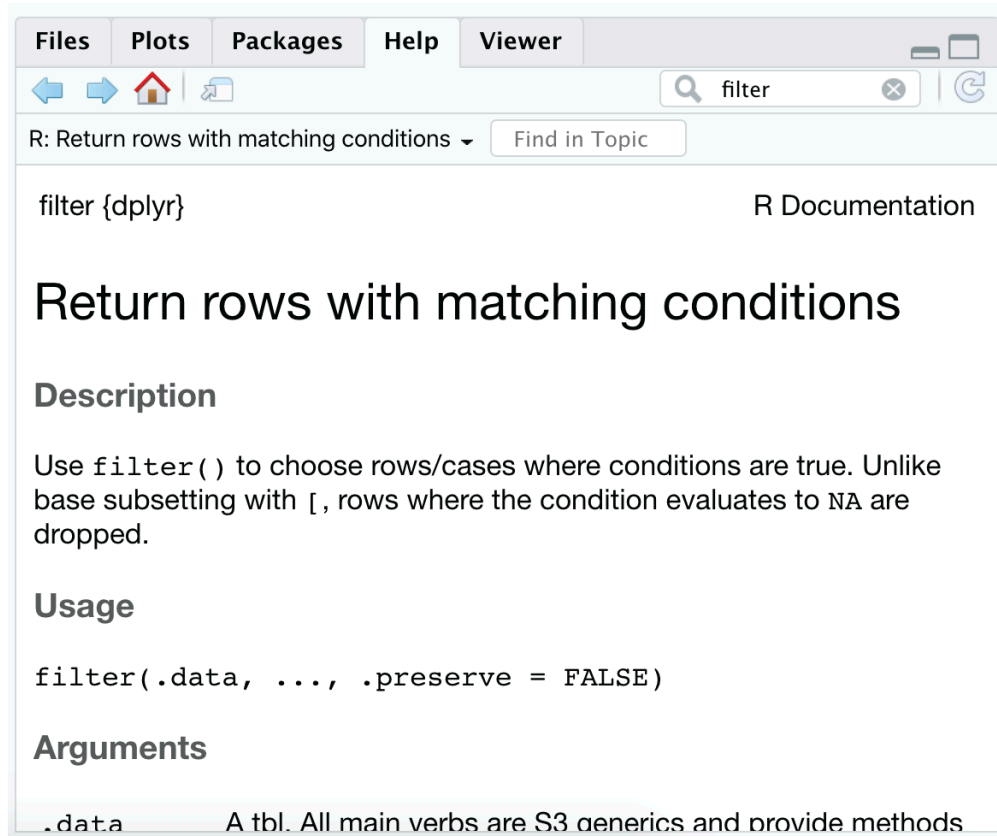
Help

To look up the help page for an R function,
type this in the console:

```
?function_name
```

(Or google it!)

Help pane



These help pages provide details about the arguments you can supply a function

Often full of examples at the bottom

Your turn

Look at the help page for `seq`

Add a chunk that uses `seq()` to create a list of numbers from 5 to 100, spaced by 5 (so 5, 10, 15, 20, ...)

02:00

```
seq(from = 5, to = 100, by = 5)
```

```
## [1] 5 10 15 20 25 30 35 40 45 50 55 60 65 70  
## [20] 100
```

Common syntax problem #1

Missing closing parentheses or quotes

```
mean(mtcars
```

```
"Oops this is wrong
```

Common syntax problem #2

Surrounding something in quotes when it should be (or vice versa)

```
mean("mtcars")
```

```
## Warning in mean.default("mtcars"): argument is not numeric or  
## returning NA
```

```
## [1] NA
```

Your turn

There are three chunks under "Syntax gone wrong"

Run each, read the error message, and try to fix the syntax

Cheatsheets

Go to Help > Cheatsheets to find quick reference guides to different packages

[illegible]

Next up

Data basics