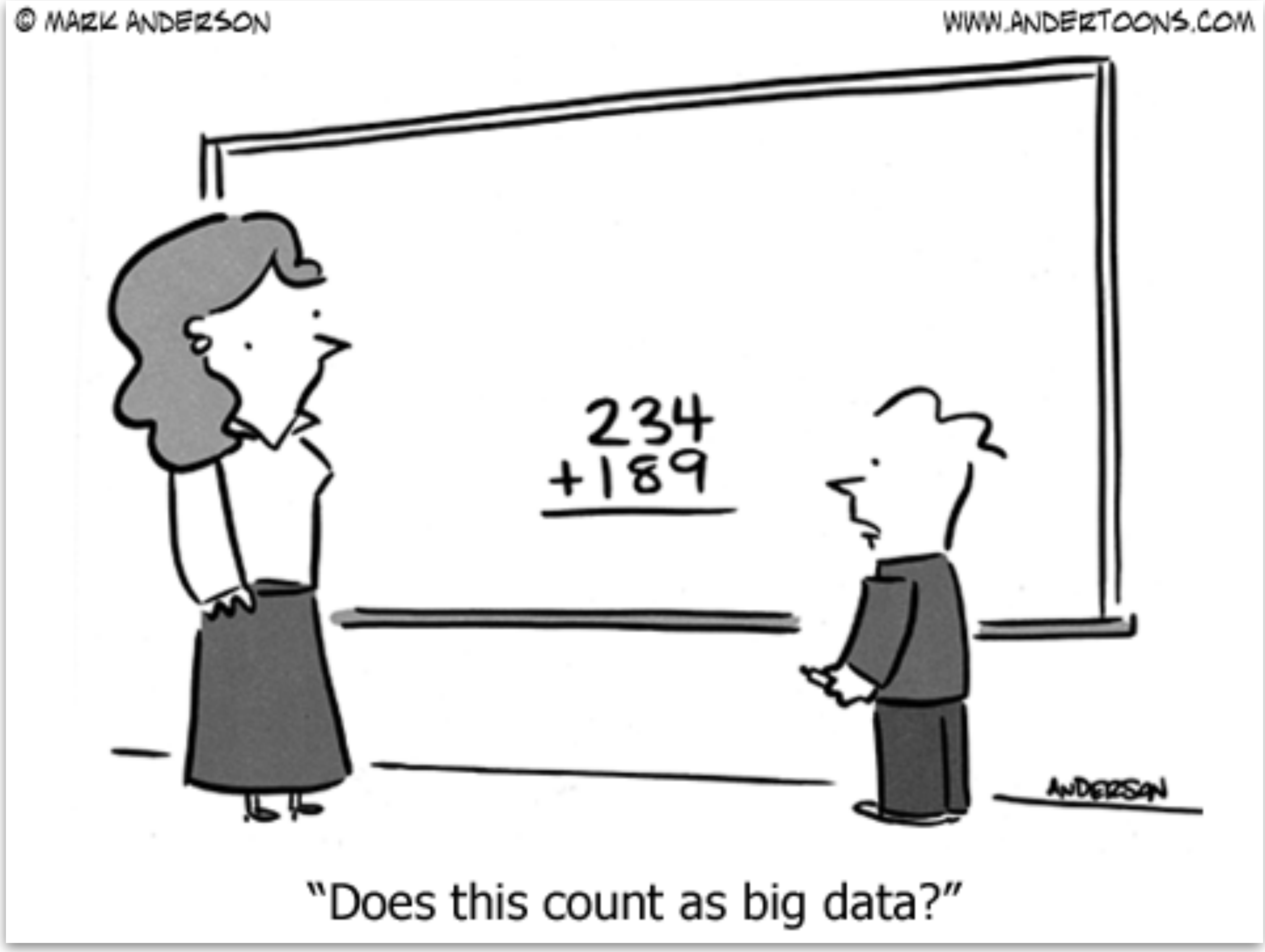
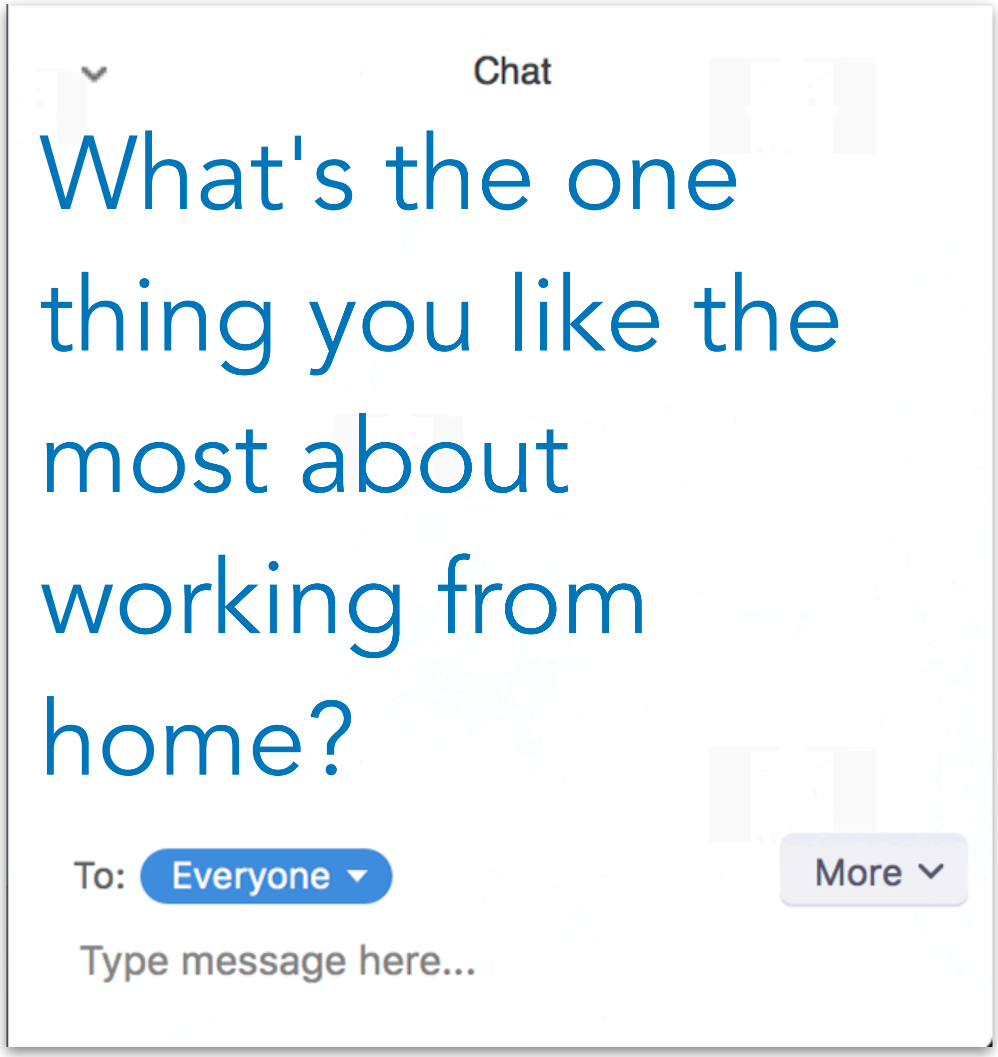
PSYCH 252: Statistical Methods for Behavioral and Social Sciences

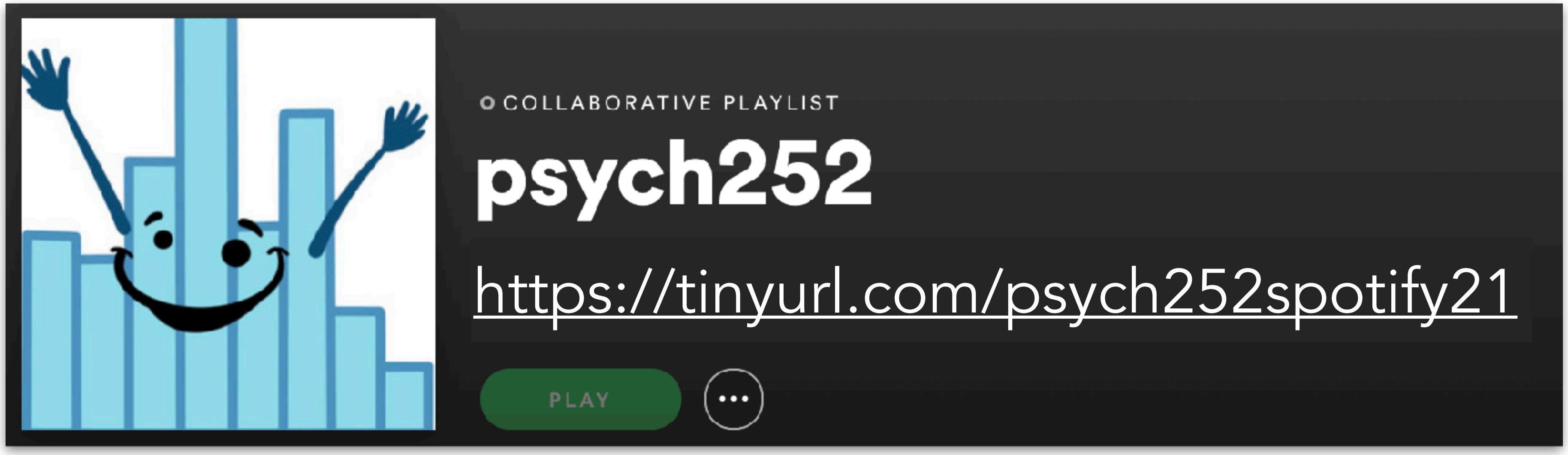
Wrangling data 1



"Does this count as big data?"



What's the one thing you like the most about working from home?


COLLABORATIVE PLAYLIST

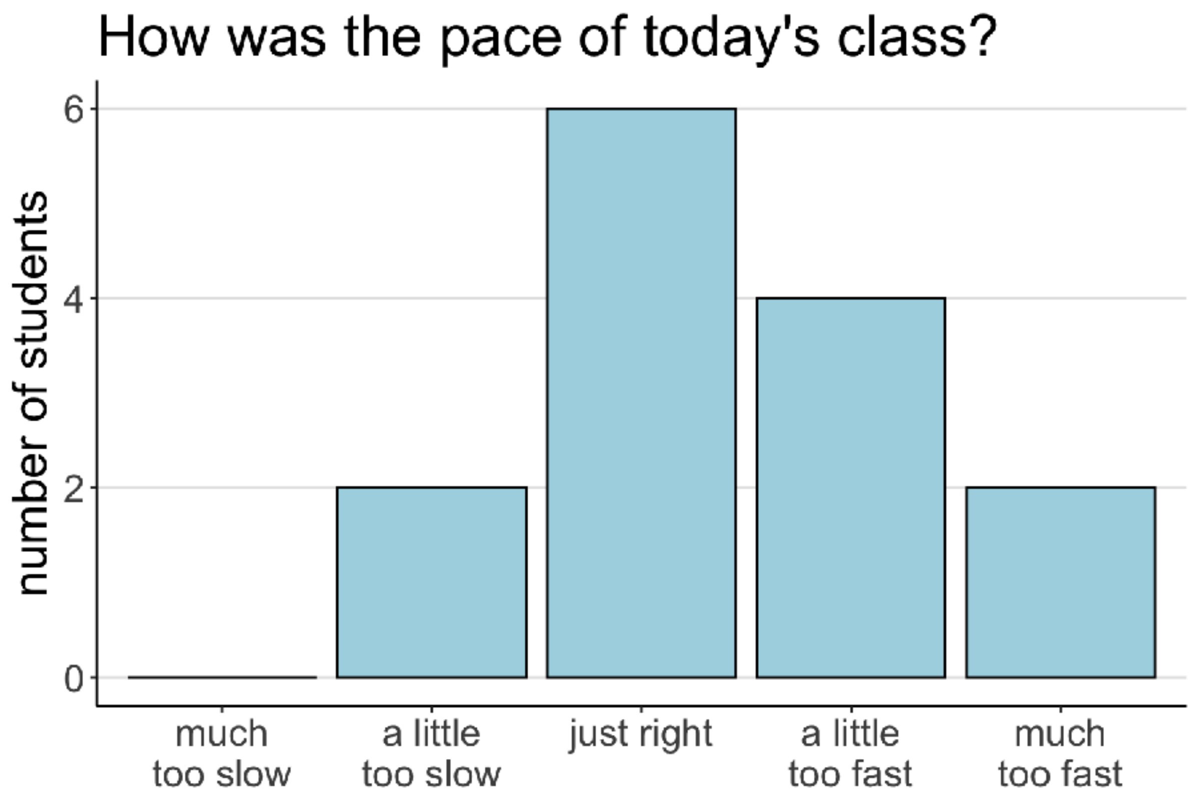
psych252

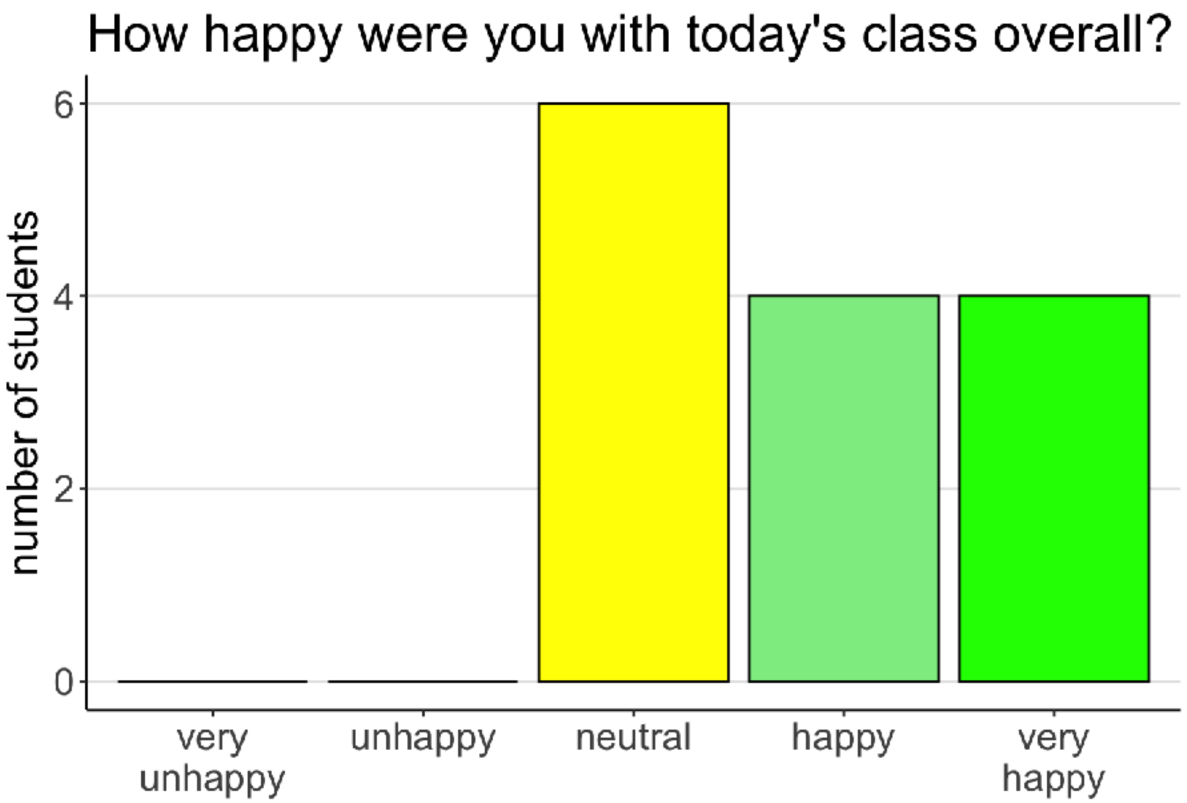
https://tinyurl.com/psych252spotify21

01/20/2021

Your feedback

Your feedback





A little fast and unsure if this is normal for the class. If we are finding the pace fast, is it useful to look over the Rmd notes and try the practice problems ahead of time? Or will we then be too ahead of class?

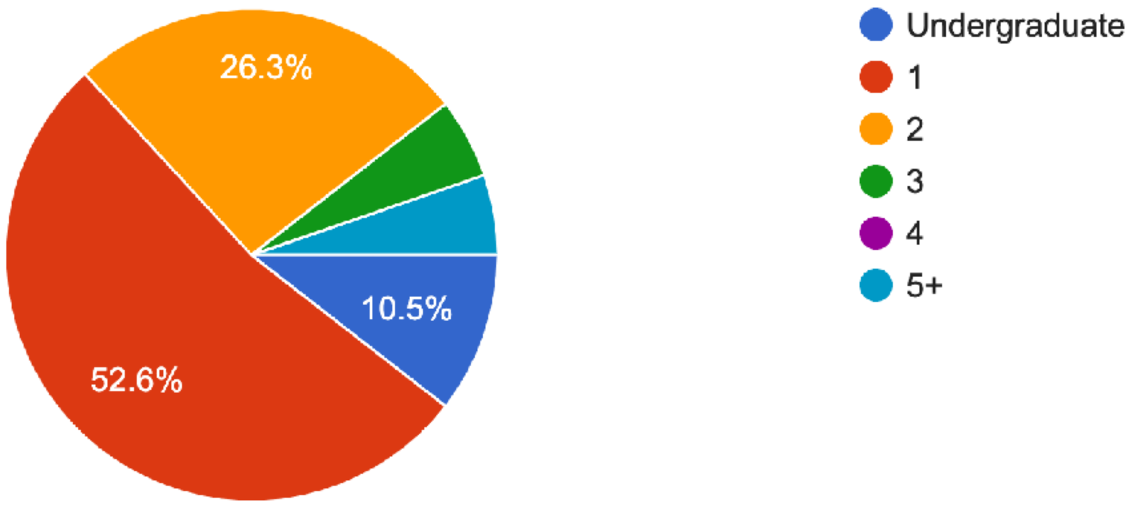
After scrolling to a line of code, give us 7 seconds to catch up to the same line of code.

I will try and go more slowly today

Introductory survey

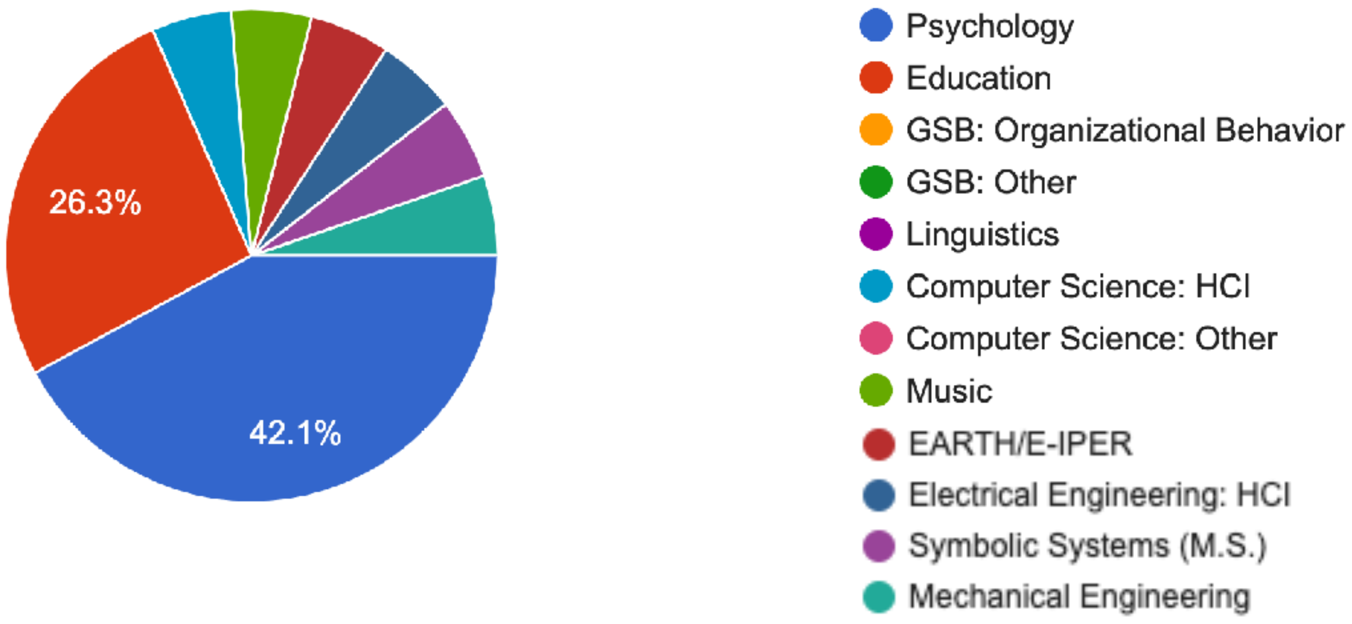
What year of graduate school are you in?

19 responses



What department are you in?

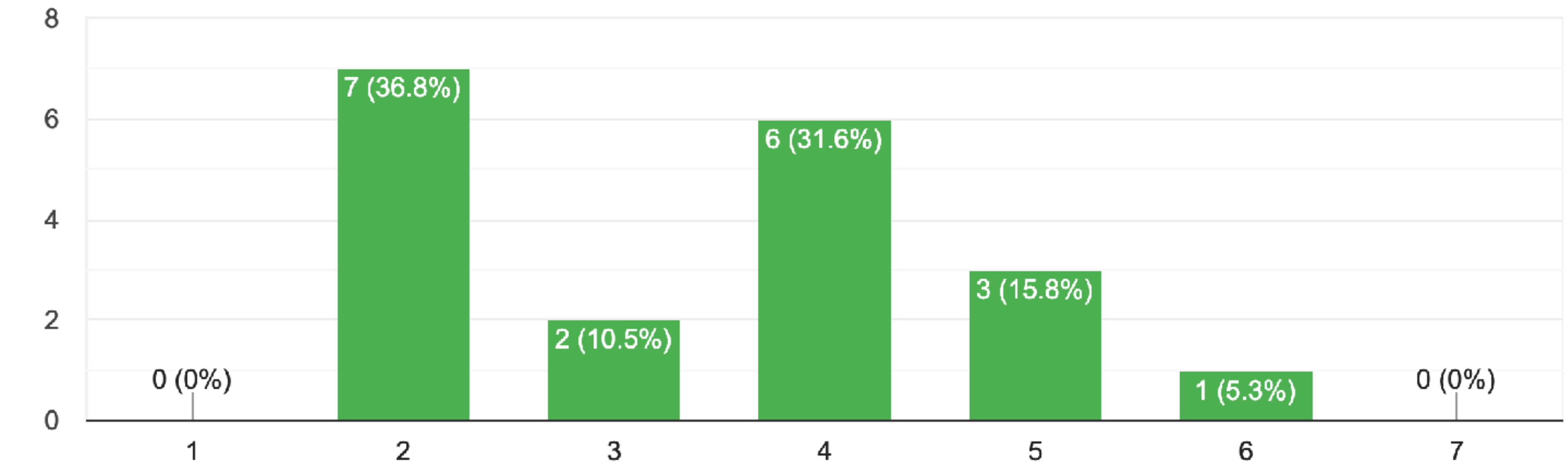
19 responses



Introductory survey

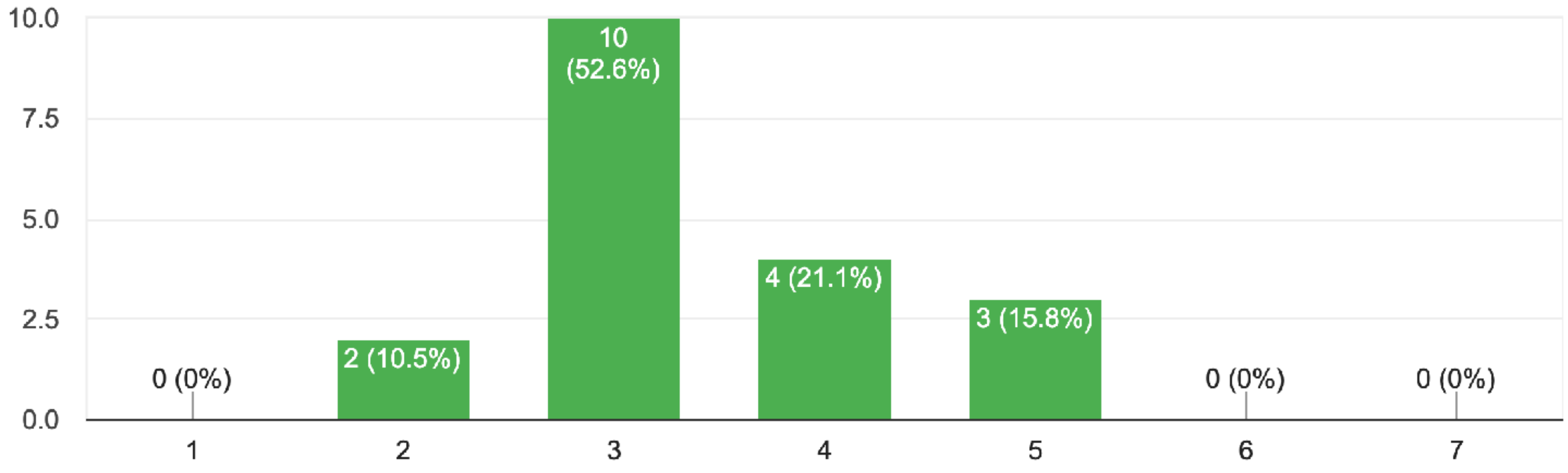
Please rate your level of experience with computer programming

19 responses



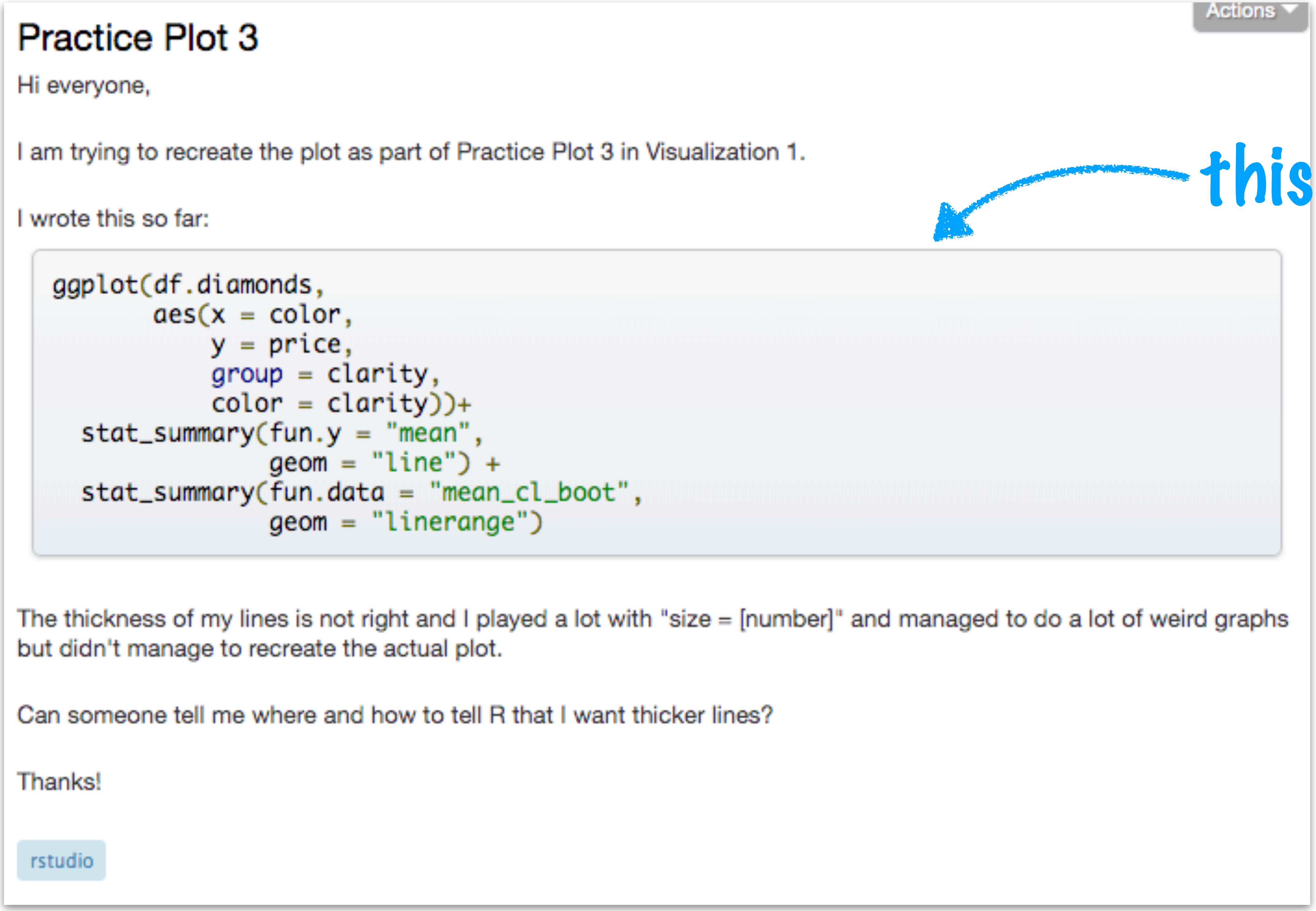
Please rate your level of experience with statistics

19 responses



Some tips and tricks

Piazza



Practice Plot 3

Hi everyone,

I am trying to recreate the plot as part of Practice Plot 3 in Visualization 1.

I wrote this so far:

```
ggplot(df.diamonds,
       aes(x = color,
           y = price,
           group = clarity,
           color = clarity))+
  stat_summary(fun.y = "mean",
               geom = "line") +
  stat_summary(fun.data = "mean_cl_boot",
               geom = "linerange")
```

The thickness of my lines is not right and I played a lot with "size = [number]" and managed to do a lot of weird graphs but didn't manage to recreate the actual plot.

Can someone tell me where and how to tell R that I want thicker lines?

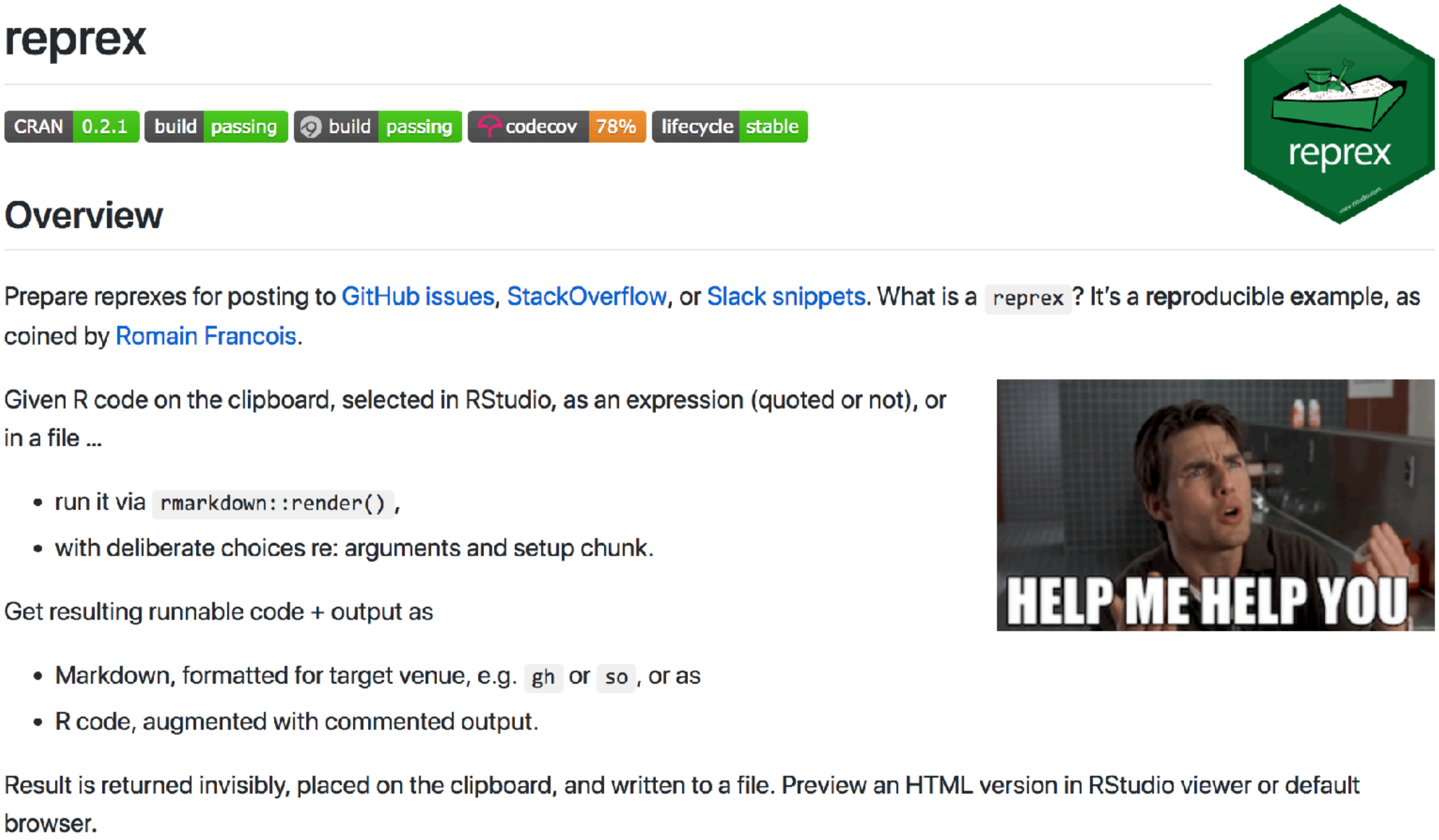
Thanks!

this is great!

Piazza

this is even better!

- best way to get help is by posting a **reprex**
- **reprex** = reproducible example



reprex

CRAN 0.2.1 | build passing | build passing | codecov 78% | lifecycle stable

Overview

Prepare reprexes for posting to GitHub issues, StackOverflow, or Slack snippets. What is a `reprex`? It's a reproducible example, as coined by Romain Francois.

Given R code on the clipboard, selected in RStudio, as an expression (quoted or not), or in a file ...

- run it via `rmarkdown::render()`,
- with deliberate choices re: arguments and setup chunk.

Get resulting runnable code + output as

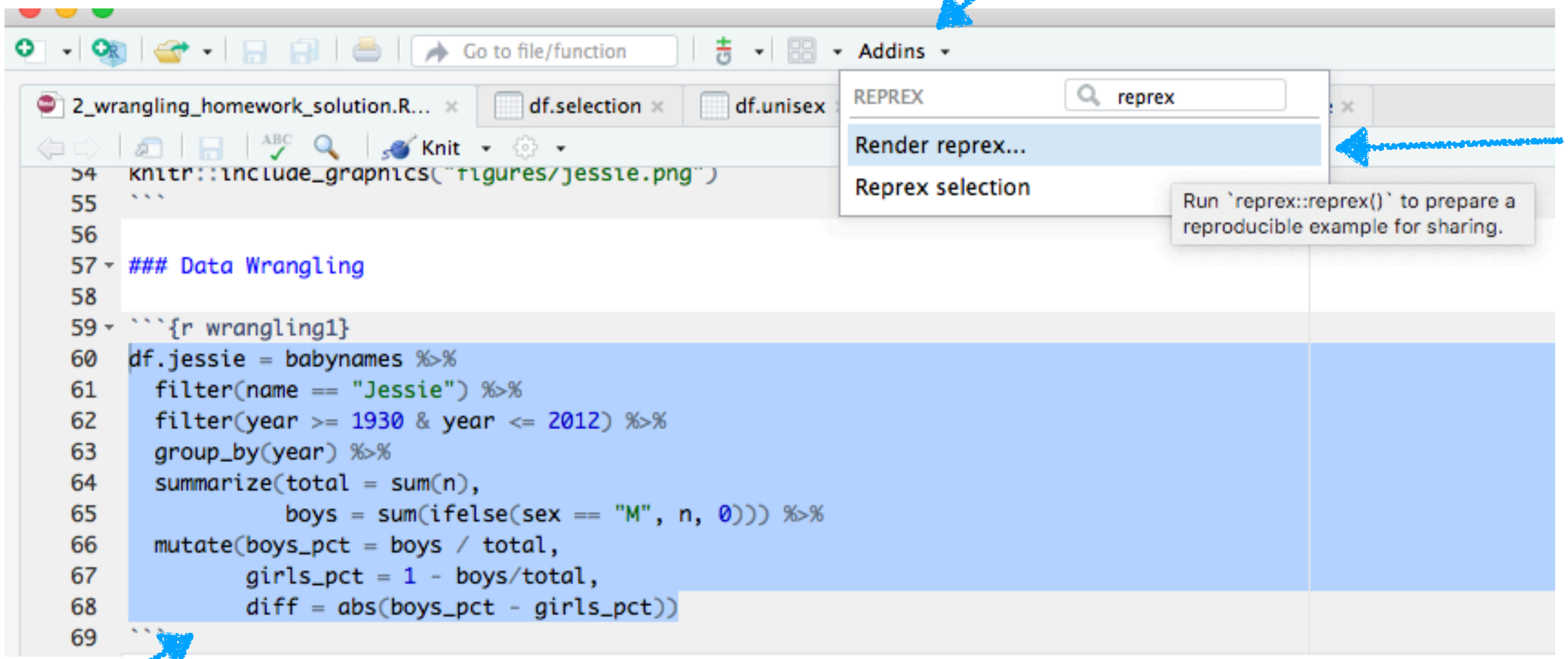
- Markdown, formatted for target venue, e.g. `gh` or `so`, or as
- R code, augmented with commented output.

Result is returned invisibly, placed on the clipboard, and written to a file. Preview an HTML version in RStudio viewer or default browser.

Piazza

```
install.package("reprex")
```

2. click on Addins



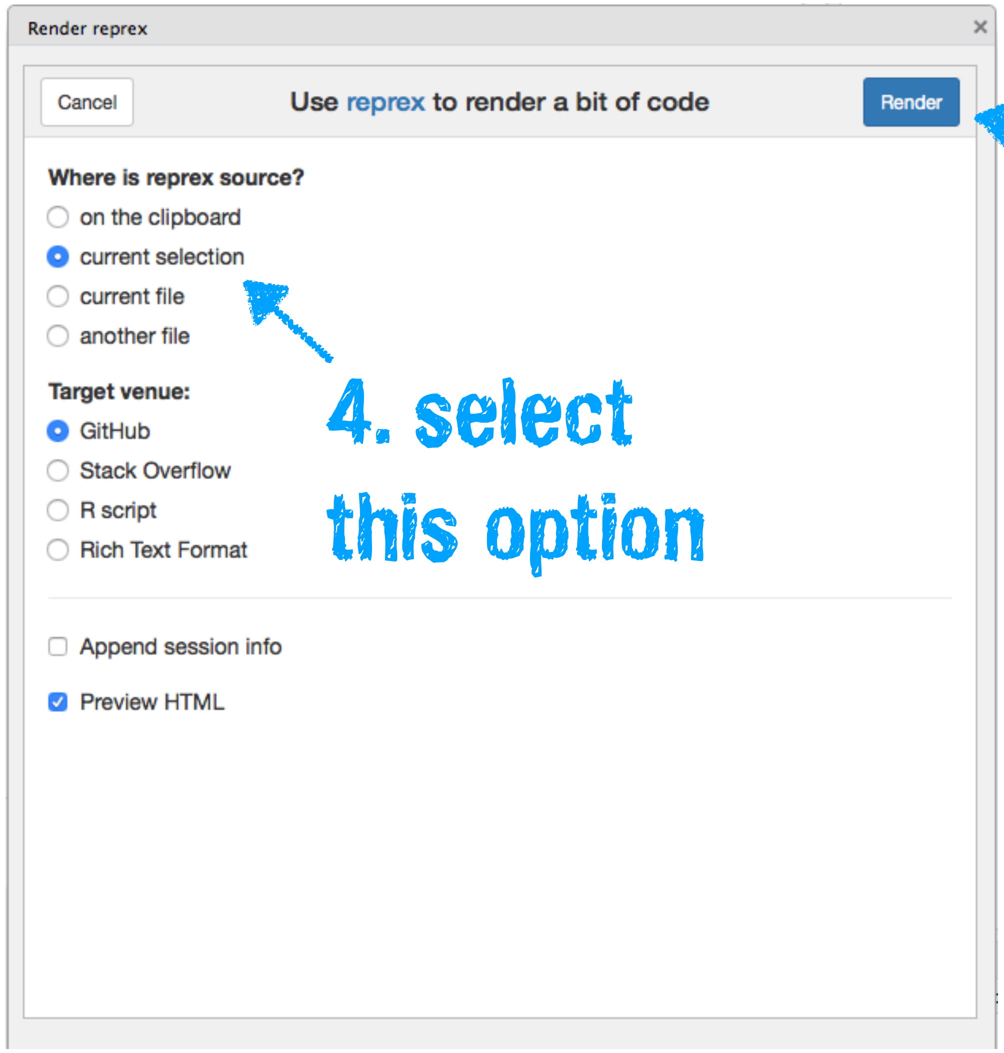
Run `reprex::reprex()` to prepare a reproducible example for sharing.

~~~
knitr::include_graphics("figures/jessie.png")
```

### Data Wrangling

```{r wrangling1}
df.jessie = babynames %>%
  filter(name == "Jessie") %>%
  filter(year >= 1930 & year <= 2012) %>%
  group_by(year) %>%
  summarize(total = sum(n),
            boys = sum(ifelse(sex == "M", n, 0))) %>%
  mutate(boys_pct = boys / total,
         girls_pct = 1 - boys/total,
         diff = abs(boys_pct - girls_pct))
```
~~~

3. Render reprex

1. select the text

Piazza



Use reprex to render a bit of code

Where is reprex source?

○ on the clipboard
● current selection
○ current file
○ another file

Target venue:

● GitHub
○ Stack Overflow
○ R script
○ Rich Text Format

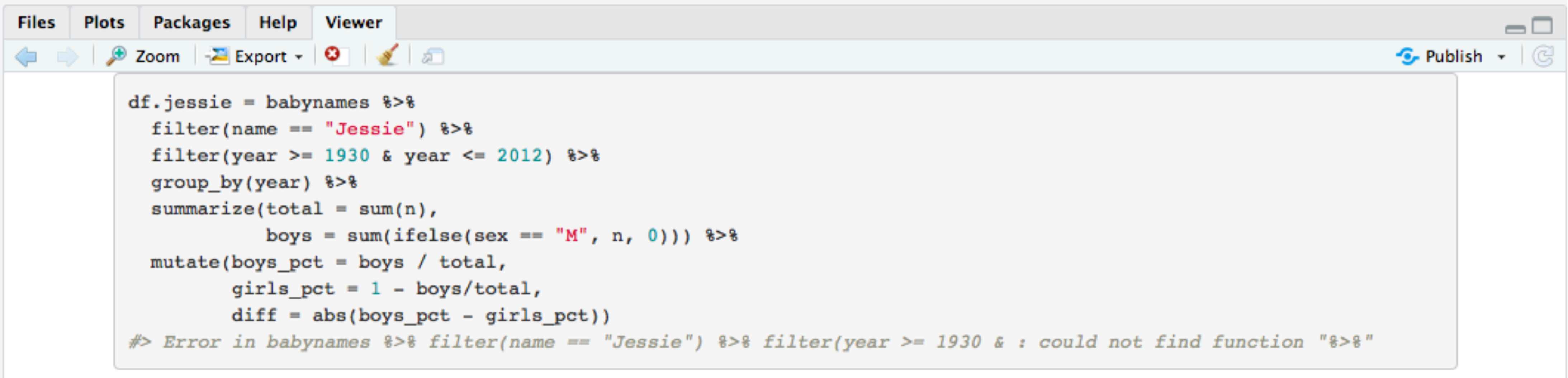
☐ Append session info

☑ Preview HTML

4. select this option

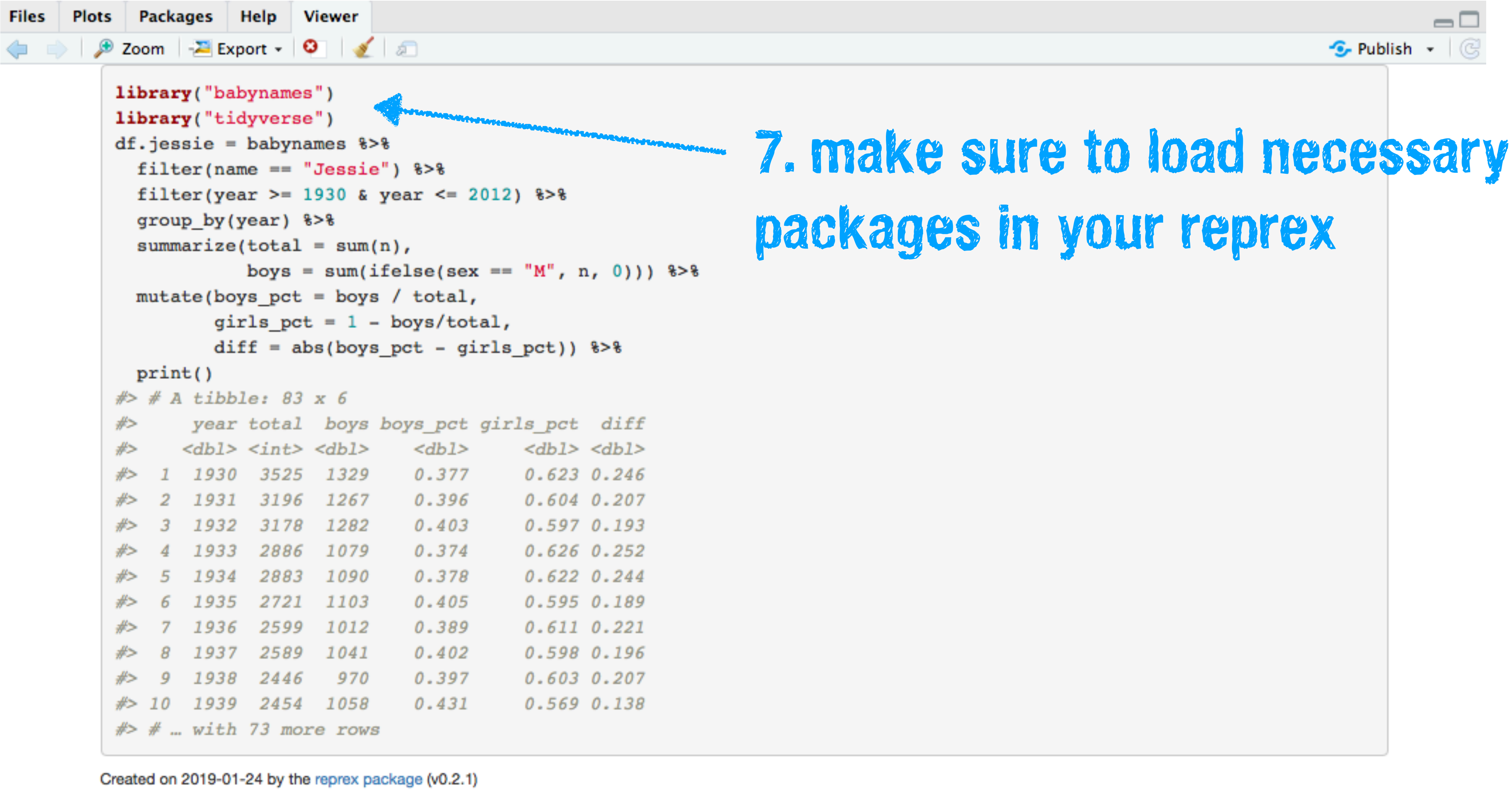
5. click render

6. copy and paste from the viewer



```
df.jessie = babynames %>%
  filter(name == "Jessie") %>%
  filter(year >= 1930 & year <= 2012) %>%
  group_by(year) %>%
  summarize(total = sum(n),
            boys = sum(ifelse(sex == "M", n, 0))) %>%
  mutate(boys_pct = boys / total,
         girls_pct = 1 - boys/total,
         diff = abs(boys_pct - girls_pct))
#> Error in babynames %>% filter(name == "Jessie") %>% filter(year >= 1930 & : could not find function "%>%"
```

Piazza



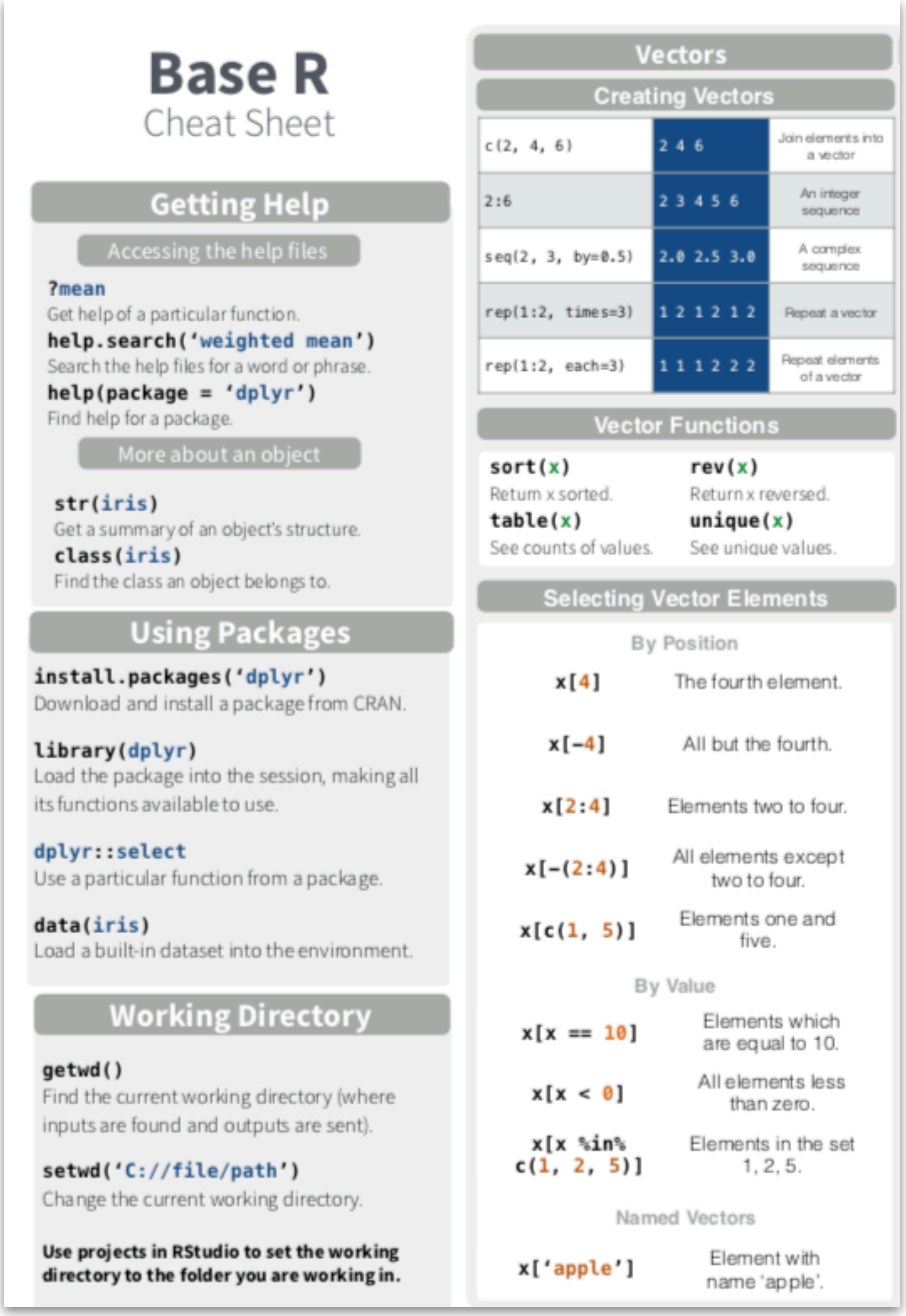
```
library("babynames")
library("tidyverse")
df.jessie = babynames %>%
  filter(name == "Jessie") %>%
  filter(year >= 1930 & year <= 2012) %>%
  group_by(year) %>%
  summarize(total = sum(n),
            boys = sum(ifelse(sex == "M", n, 0))) %>%
  mutate(boys_pct = boys / total,
         girls_pct = 1 - boys/total,
         diff = abs(boys_pct - girls_pct)) %>%
  print()
#> # A tibble: 83 x 6
#>     year total  boys boys_pct girls_pct  diff
#>    <dbl> <int> <dbl>    <dbl>     <dbl> <dbl>
#>  1  1930  3525  1329    0.377     0.623 0.246
#>  2  1931  3196  1267    0.396     0.604 0.207
#>  3  1932  3178  1282    0.403     0.597 0.193
#>  4  1933  2886  1079    0.374     0.626 0.252
#>  5  1934  2883  1090    0.378     0.622 0.244
#>  6  1935  2721  1103    0.405     0.595 0.189
#>  7  1936  2599  1012    0.389     0.611 0.221
#>  8  1937  2589  1041    0.402     0.598 0.196
#>  9  1938  2446   970    0.397     0.603 0.207
#> 10  1939  2454  1058    0.431     0.569 0.138
#> # … with 73 more rows
```

7. make sure to load necessary packages in your reprex

Created on 2019-01-24 by the reprex package (v0.2.1)

Data wrangling 1

Two styles of coding in R



Base R
Cheat Sheet

Getting Help

Accessing the help files

`?mean`
Get help of a particular function.

`help.search('weighted mean')`
Search the help files for a word or phrase.

`help(package = 'dplyr')`
Find help for a package.

More about an object

`str(iris)`
Get a summary of an object's structure.

`class(iris)`
Find the class an object belongs to.

Using Packages

`install.packages('dplyr')`
Download and install a package from CRAN.

`library(dplyr)`
Load the package into the session, making all its functions available to use.

`dplyr::select`
Use a particular function from a package.

`data(iris)`
Load a built-in dataset into the environment.

Working Directory

`getwd()`
Find the current working directory (where inputs are found and outputs are sent).

`setwd('C://file/path')`
Change the current working directory.

Use projects in RStudio to set the working directory to the folder you are working in.

Vectors

Creating Vectors

`c(2, 4, 6)`	2 4 6	Join elements into a vector
`2:6`	2 3 4 5 6	An integer sequence
`seq(2, 3, by=0.5)`	2.0 2.5 3.0	A complex sequence
`rep(1:2, times=3)`	1 2 1 2 1 2	Repeat a vector
`rep(1:2, each=3)`	1 1 1 2 2 2	Repeat elements of a vector

Vector Functions

`sort(x)`
Return x sorted.

`table(x)`
See counts of values.

`rev(x)`
Return x reversed.

`unique(x)`
See unique values.

Selecting Vector Elements

By Position

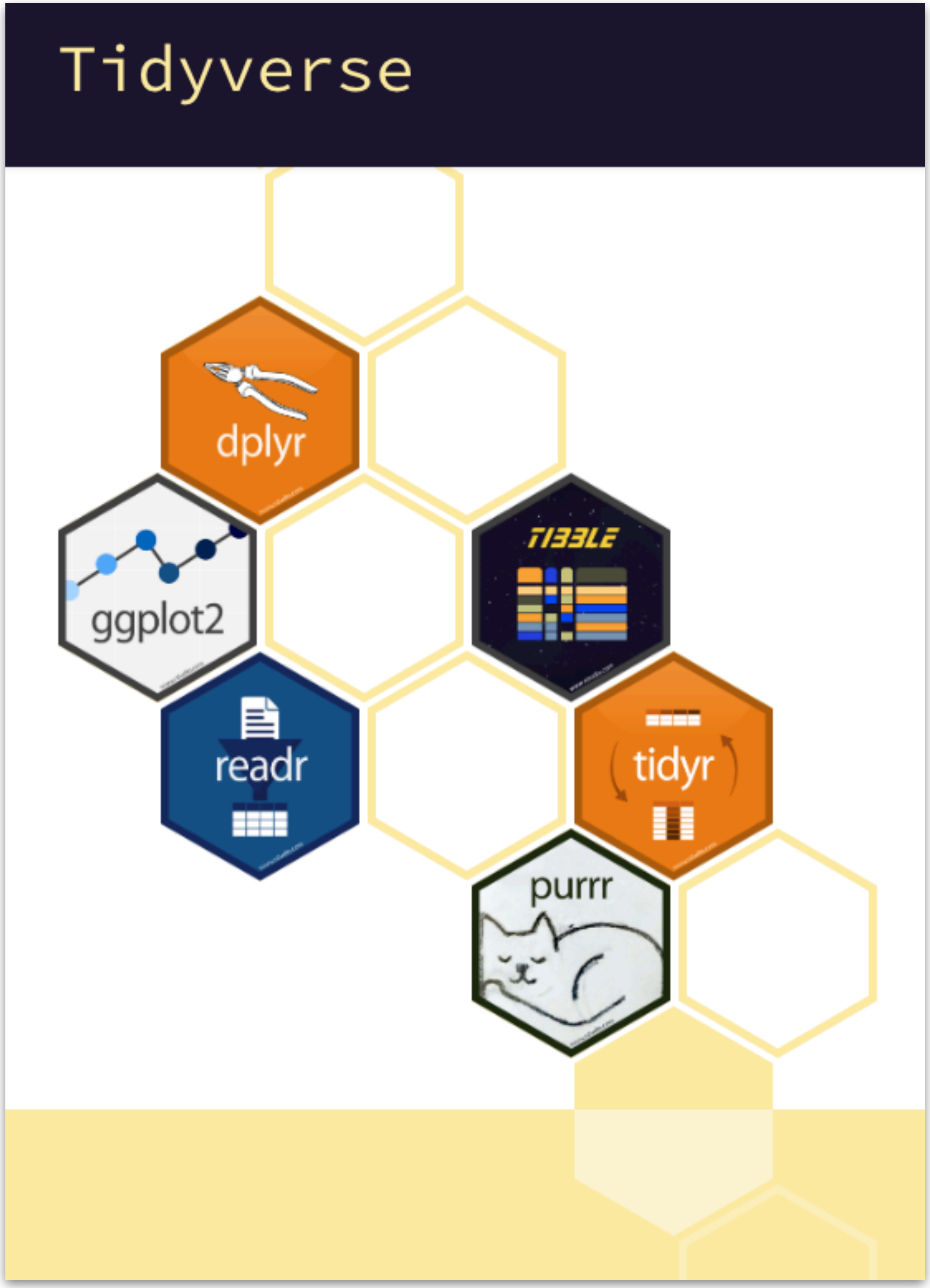
`x[4]`	The fourth element.
`x[-4]`	All but the fourth.
`x[2:4]`	Elements two to four.
`x[-(2:4)]`	All elements except two to four.
`x[c(1, 5)]`	Elements one and five.

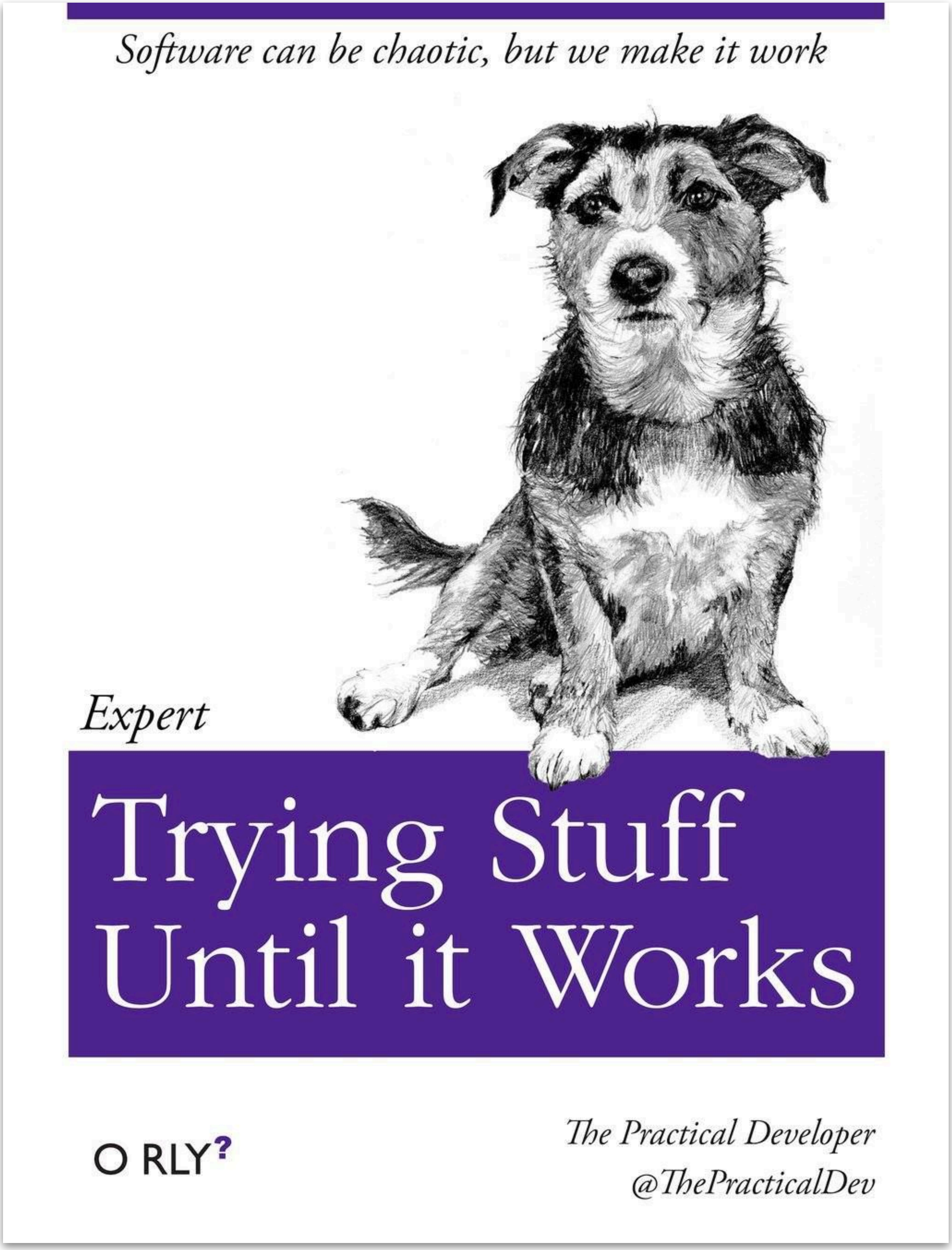
By Value

`x[x == 10]`	Elements which are equal to 10.
`x[x < 0]`	All elements less than zero.
`x[x %in% c(1, 2, 5)]`	Elements in the set 1, 2, 5.

Named Vectors

`x['apple']`	Element with name 'apple'.





Software can be chaotic, but we make it work

Expert

Trying Stuff Until it Works

O RLY?

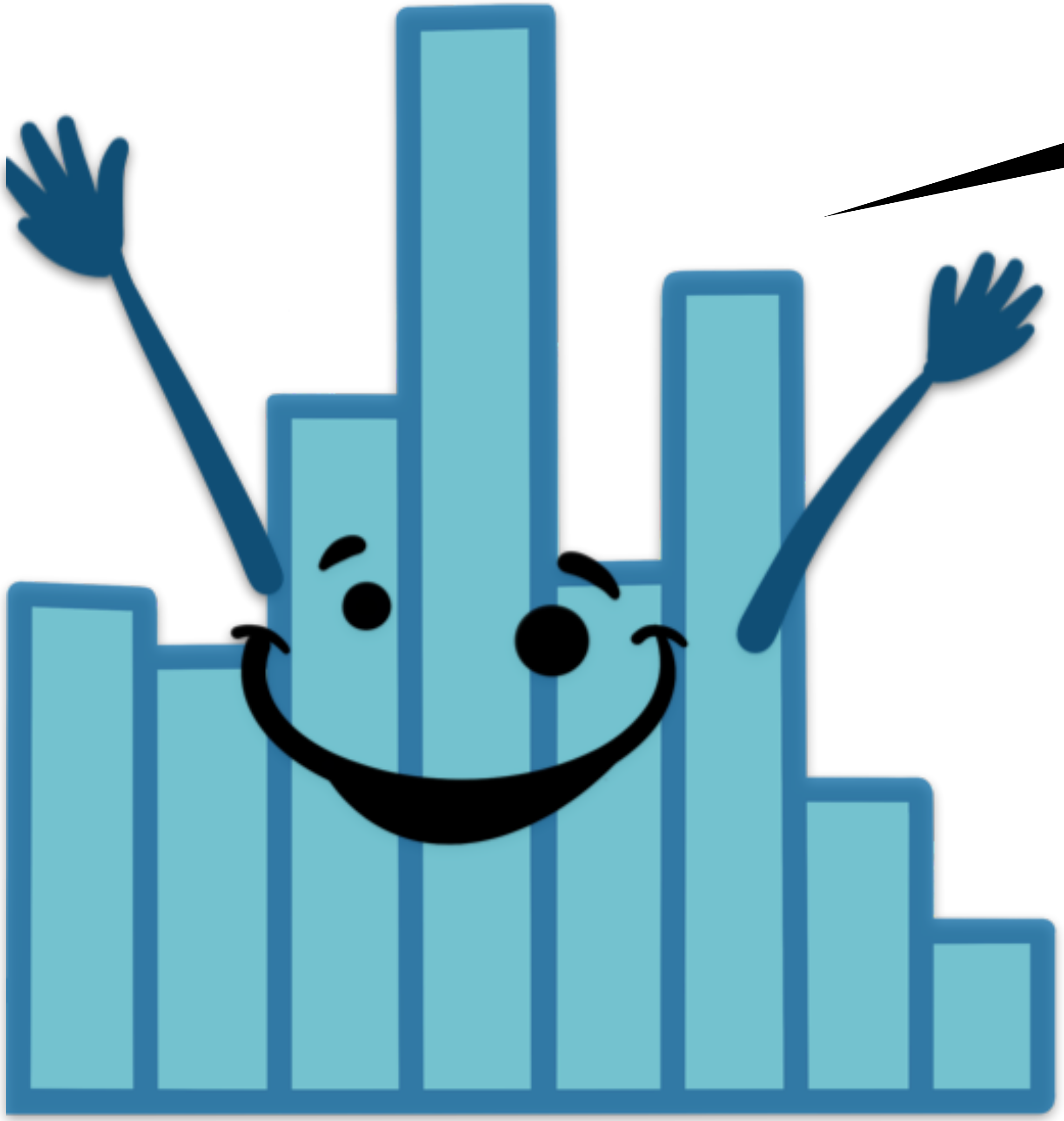
The Practical Developer
@ThePracticalDev

RStudio time

STAR WARS

01:00

stretch break!



Feedback

How was the pace of today's class?

much too slow
a little too slow
just right
a little too fast
much too fast

How happy were you with today's class overall?

What did you like about today's class? What could be improved next time?

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**