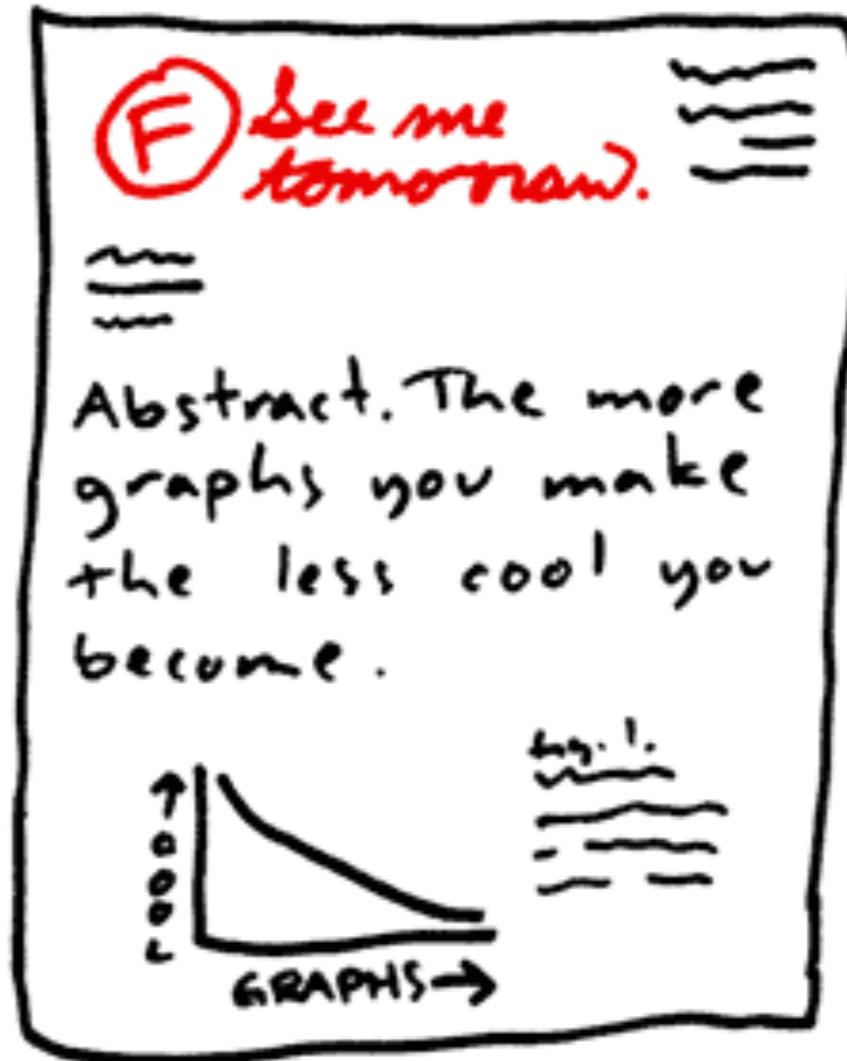


Visualization 2



Things that came up ...

Struggling with 3D plots

plea for open science: 30 years later, some poor sap might end up needing to estimate your raw data by inverting difference scores read from a 3d bar plot.

EXPERIMENT 2

The chart displays the net error proportion for two conditions: CONCRETE (solid black bars) and ABSTRACT (hatched bars). The y-axis ranges from -0.1 to 0.2. The x-axis shows PLURAL and SINGULAR categories. For PLURAL, the CONCRETE bar is at approximately 0.14 and the ABSTRACT bar is at approximately 0.05. For SINGULAR, the CONCRETE bar is at approximately -0.03 and the ABSTRACT bar is at approximately 0.02. Error bars are present on all bars.

Number of Local Noun	Concrete (Net Error Proportion)	Abstract (Net Error Proportion)
PLURAL	~0.14	~0.05
SINGULAR	~-0.03	~0.02

LOCAL NOUN:

- CONCRETE
- ABSTRACT

NET ERROR PROPORTION

PLURAL SINGULAR

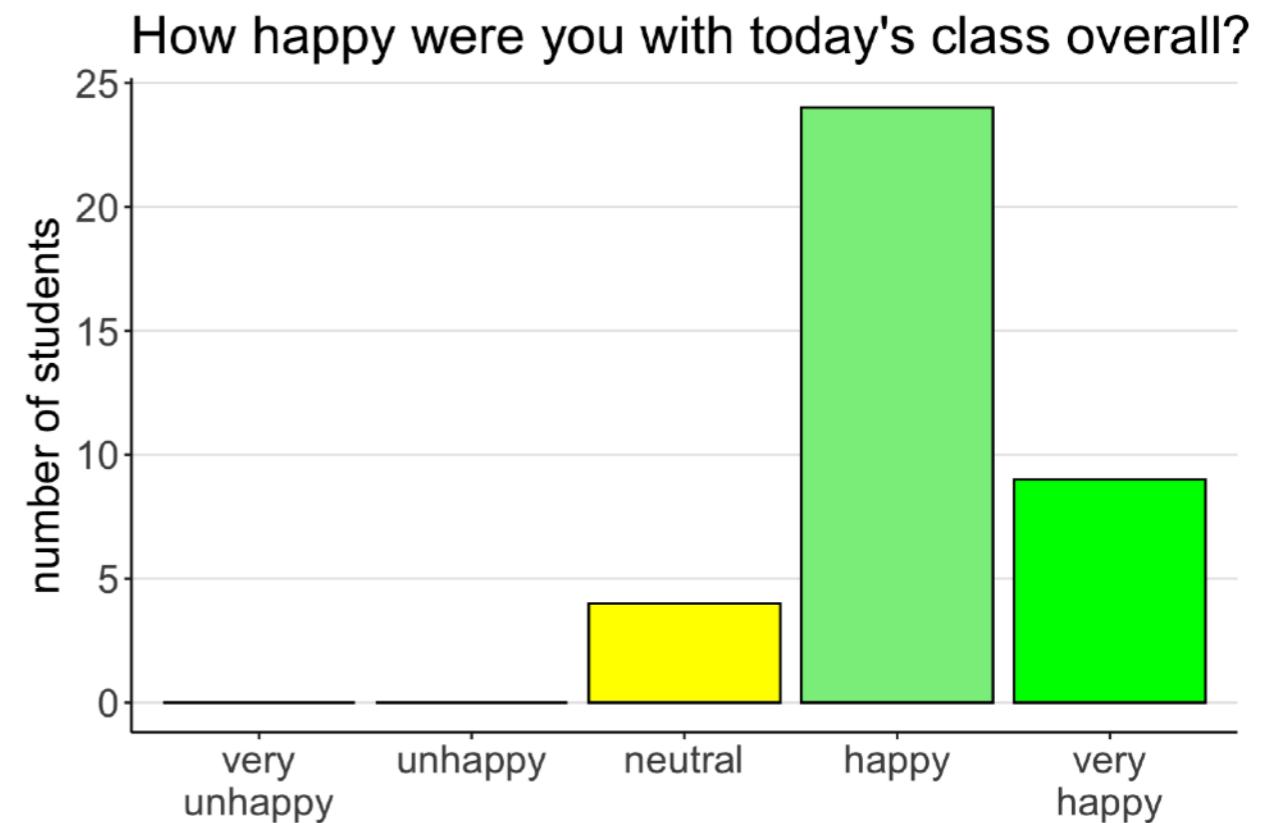
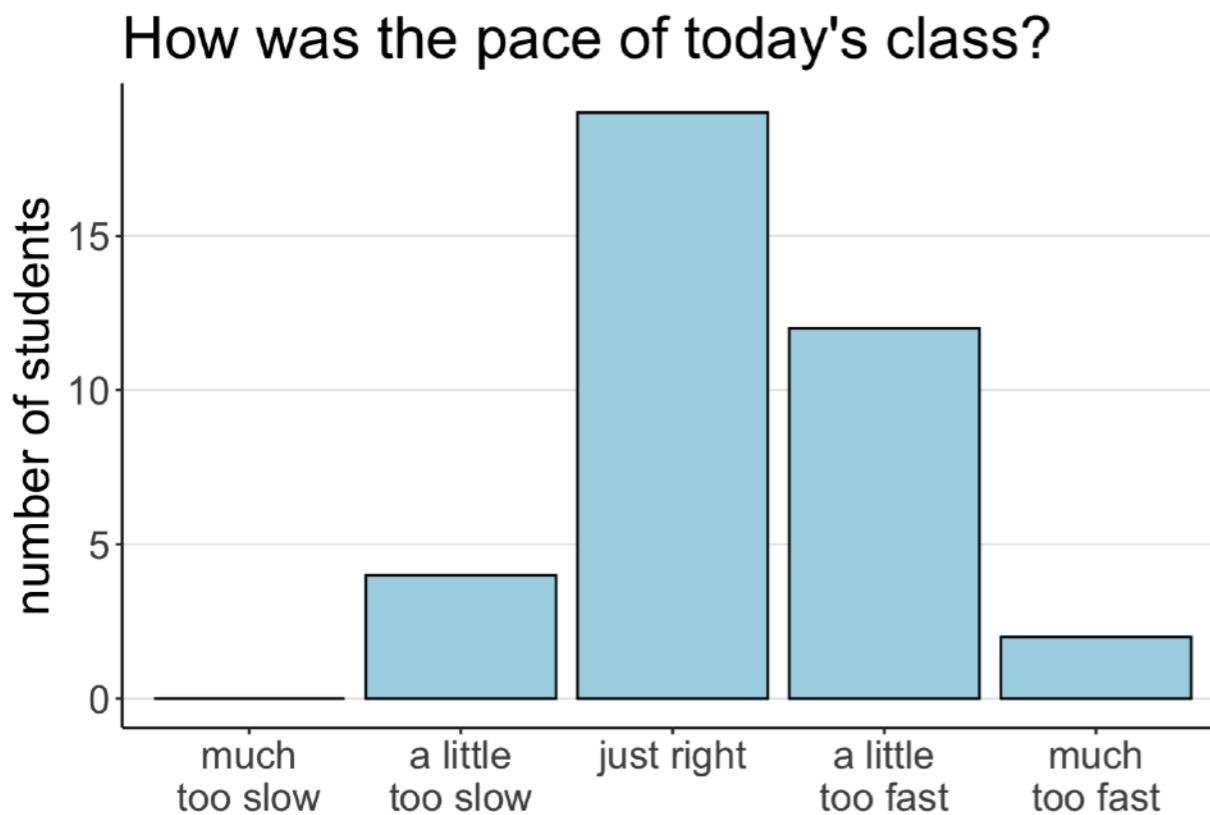
NUMBER OF LOCAL NOUN

1:50 PM · Jan 8, 2020 · Twitter Web App

20 Likes

Your feedback

Your feedback



take a look at the datacamp tutorials
and other resources mentioned in
the RMarkdown scripts



Your feedback

can you give us the playlist

check it out here: <https://tinyurl.com/psych252playlist20>

Your feedback

I enjoyed the detailed explanations for the reasons for errors in R. I also thought the practice sets were a good way to consolidate the newly learned concepts. **Please I would like the notes to always be available a day before the class.**

**I'll try to make them available as early as possible
(might still tweak small things the day of class)**

Your feedback

I really liked the parts where we did “hands-on” exercises and wrote code ourselves. **I wish we could have spent a little more time (as a group) going over the “correct” code**, because I felt that the answer part went a little too quickly for those of us who made mistakes.

I'll spend more time on going over the solutions together

Your feedback

It may be nice to actually have more individual/group exercises. I usually find that this kind of material is hard to learn until I have the opportunity to apply it.

come to homework section and application section!

Your feedback

I was surprised by how much overlap there was between the course material and the course readings posted on the website (i.e. word-for-word). **Should I assume moving forward that we don't need to complete the readings before class, but rather that they are most helpful for reviewing anything we missed / need clarity on after a class?**

yes, that's right -- sorry for the confusion!

Friday, January 10th: Visualization II

Content:

- Deciding what plot is appropriate for what kind of data.
- Customizing plots: Take a sad plot and make it better.
- Saving plots.
- Making figure panels.
- Debugging.
- Making animations.
- Defining snippets.

Resources:

- [Cheatsheet shiny](#)

Datacamp:

- [ggplot part 3](#)
- [Shiny 1](#)
- [Shiny 2](#)

Reading:

- [Course notes: Visualization 2](#)
- [Data visualization \(#4\)](#)
- [Data visualization \(#8\)](#)
- [R for Data Science \(#27\)](#)

Your feedback

Just a heads up that I noticed today before class, two of the data camp courses listed on the course website under the first day (RStudio IDE 1, RStudio IDE 2) are archived on data camp and are no longer available.

thanks! I've removed the broken links from the materials

Your feedback

I still didn't figure out how to do an R Project properly. Am I supposed to save the file from Canvas into the R project I created? What if I create a new file? Still not quite sure how to use R Project properly.

**the files on Canvas contain an .Rproj file
you can open up that file, and then navigate to the
RMarkdown file within RStudio's Files browser**

**the idea is to have one .Rproj that
can contain many .r or .rmd files**

Final projects

Final projects

W20-PSYCH-252-01 > Files > final_project

Search for files		Q	0 items selected			+ Folder	Upload
Name	Date Created	Date Modified	Modified By	Size	Actions		
final_report	8:42pm			--	<input checked="" type="checkbox"/>		
proposal	8:42pm			--	<input checked="" type="checkbox"/>		

Statistical Methods for Psychology
final_project
final_report
proposal
homework
slides

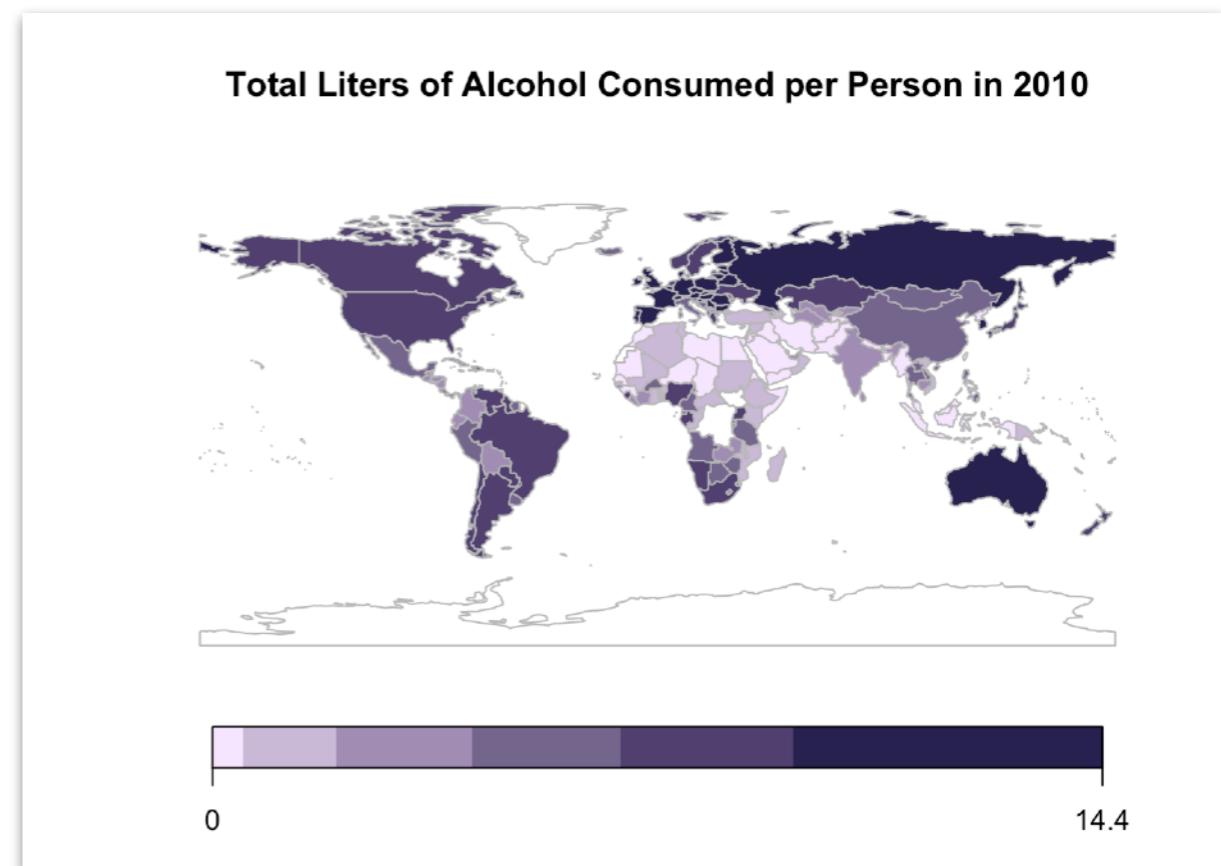
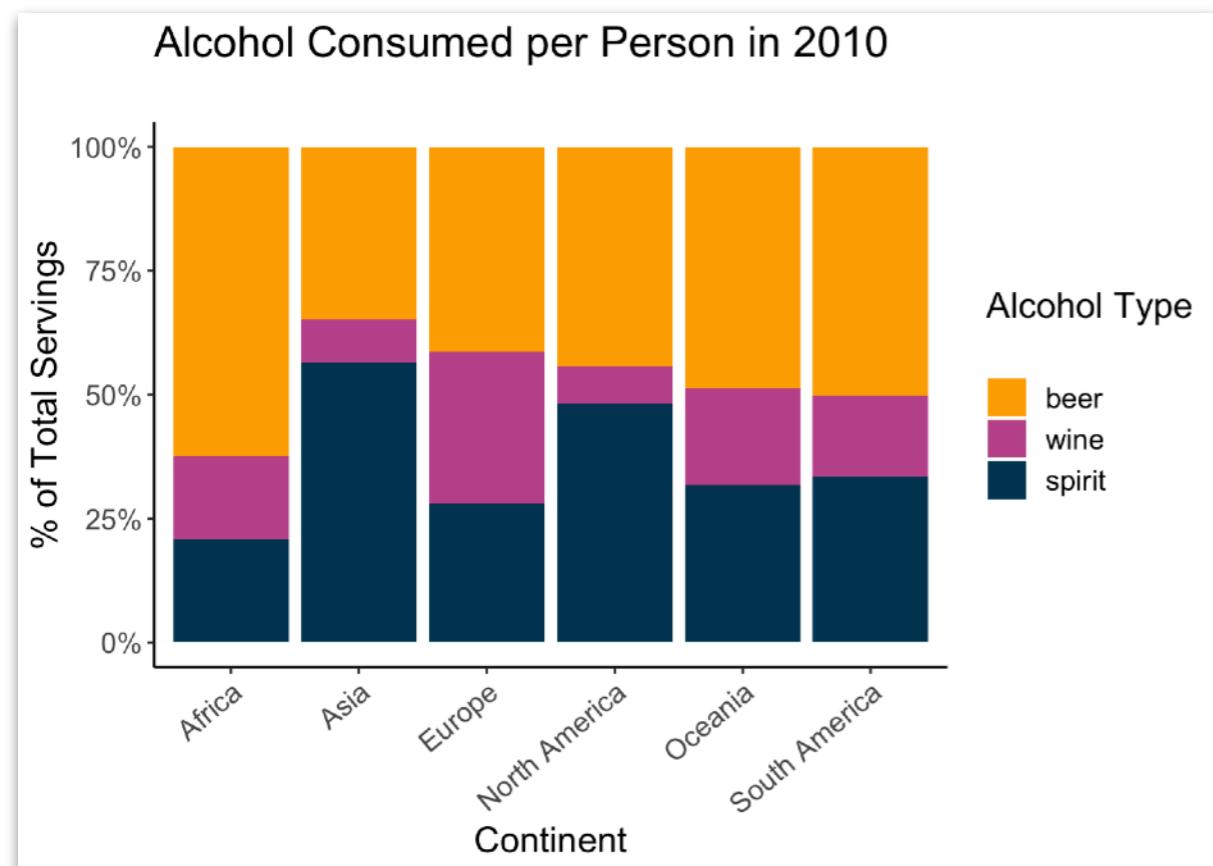
check out examples of proposals and final reports on Canvas

Homework

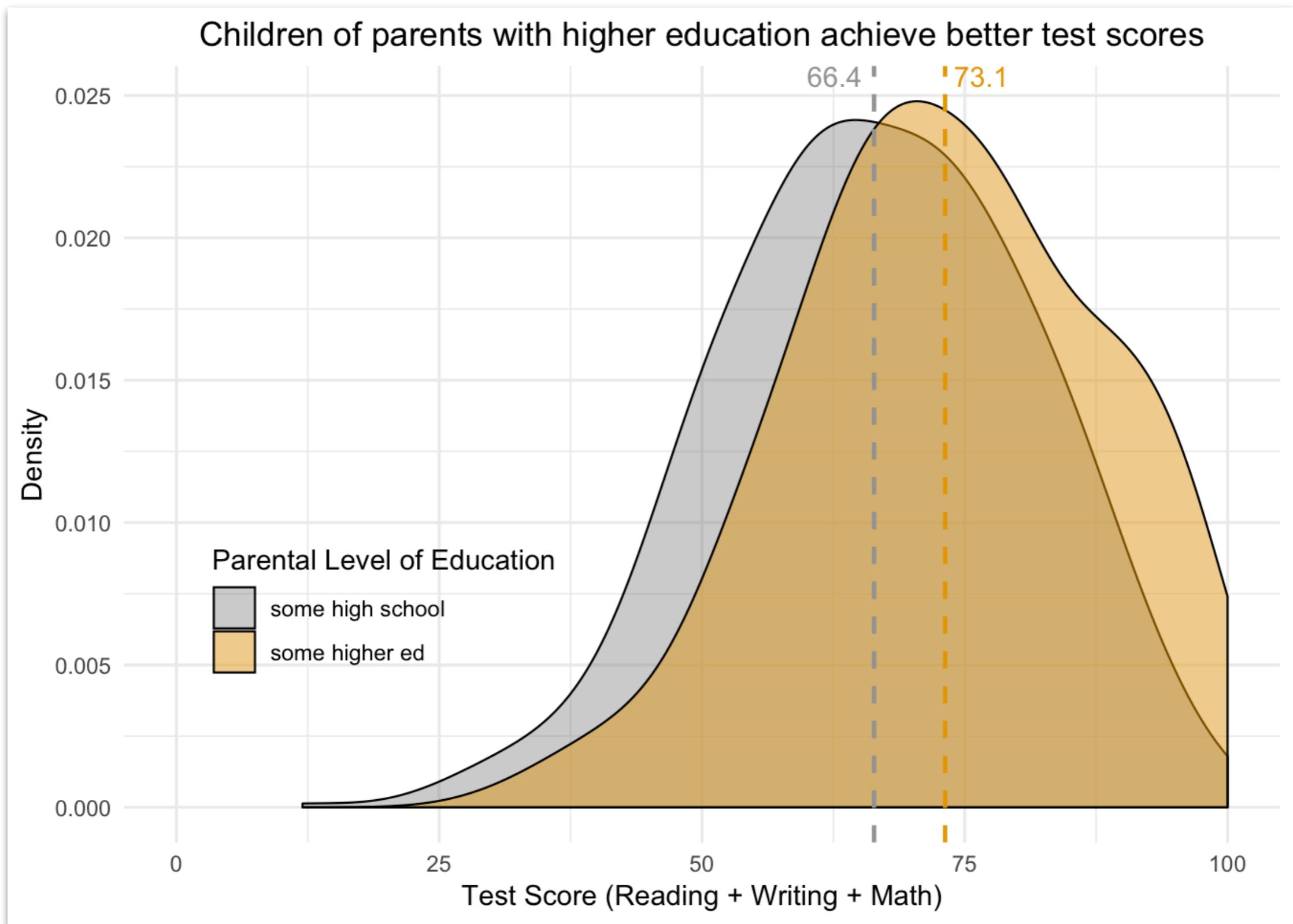
Homework

In this homework, **you'll write a short blog post** about a data set. Your goal is to tell us something interesting using a well-crafted, thoughtfully-prepared data graphic.

Homework

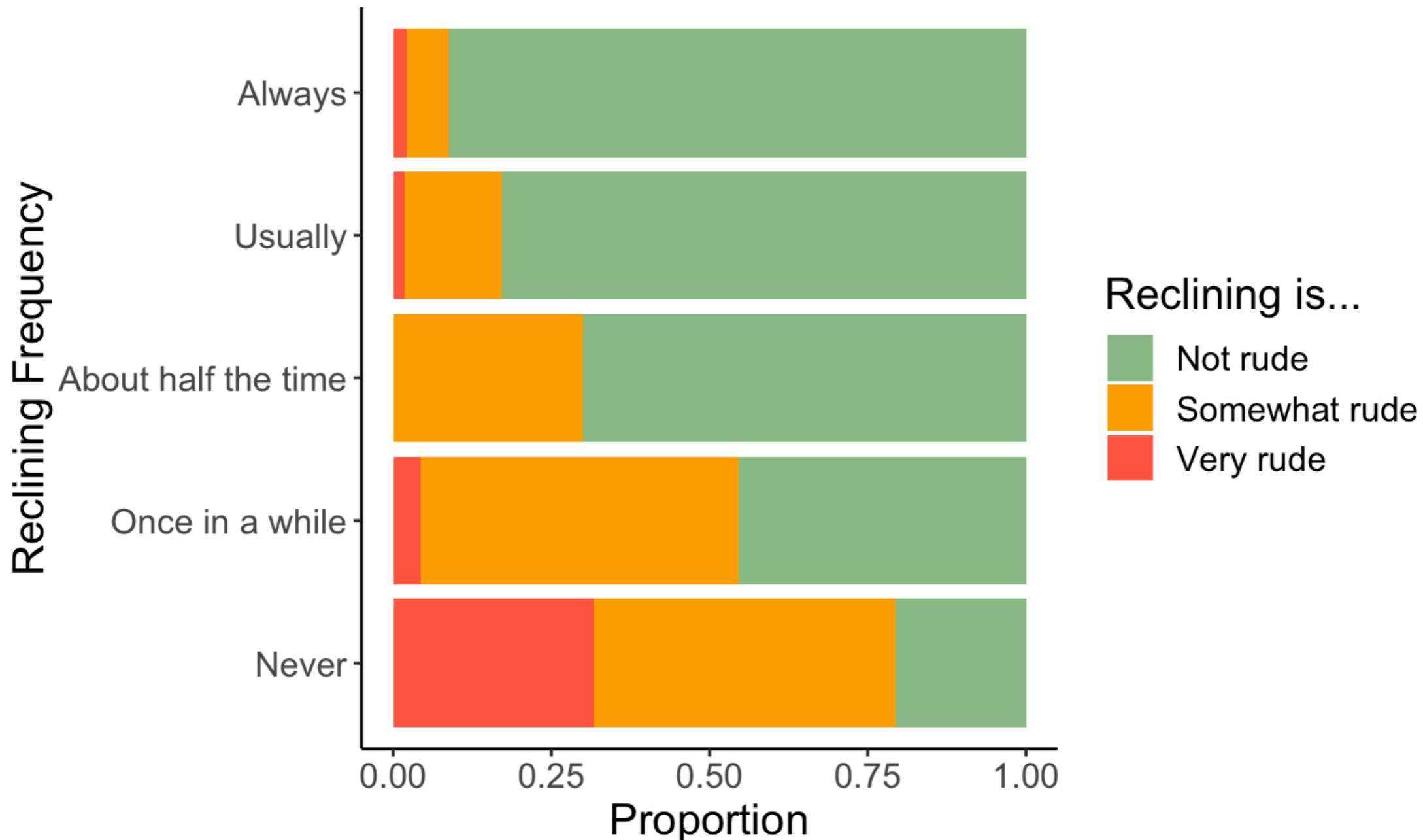


Homework



Homework

Perceived Rudeness of Reclining Airplane Seat
by Frequency of Own
Seat Reclining



Homework

Homework is due by **Thursday 16th, 8pm**

Remember the 0 points for late submissions ...

Name	Date Created	Date Modified	Modified By	Size	Actions
1_visualization_homework.html	8:38pm	8:38pm	Tobias Gerstenberg	696 KB	
1_visualization_homework.pdf	8:38pm	8:38pm	Tobias Gerstenberg	223 KB	
1_visualization_homework.Rmd	8:38pm	8:38pm	Tobias Gerstenberg	6 KB	
1_visualization.Rproj	8:38pm	8:38pm	Tobias Gerstenberg	205 bytes	

Submit **one pdf file** (knitted with RMarkdown) that contains the code as well as the figure.

Homework

1_visualization_homework.pdf (page 3 of 4)

{Your blog post title goes here ...}

Load packages

Add the package with the data set that you'd like to load below.

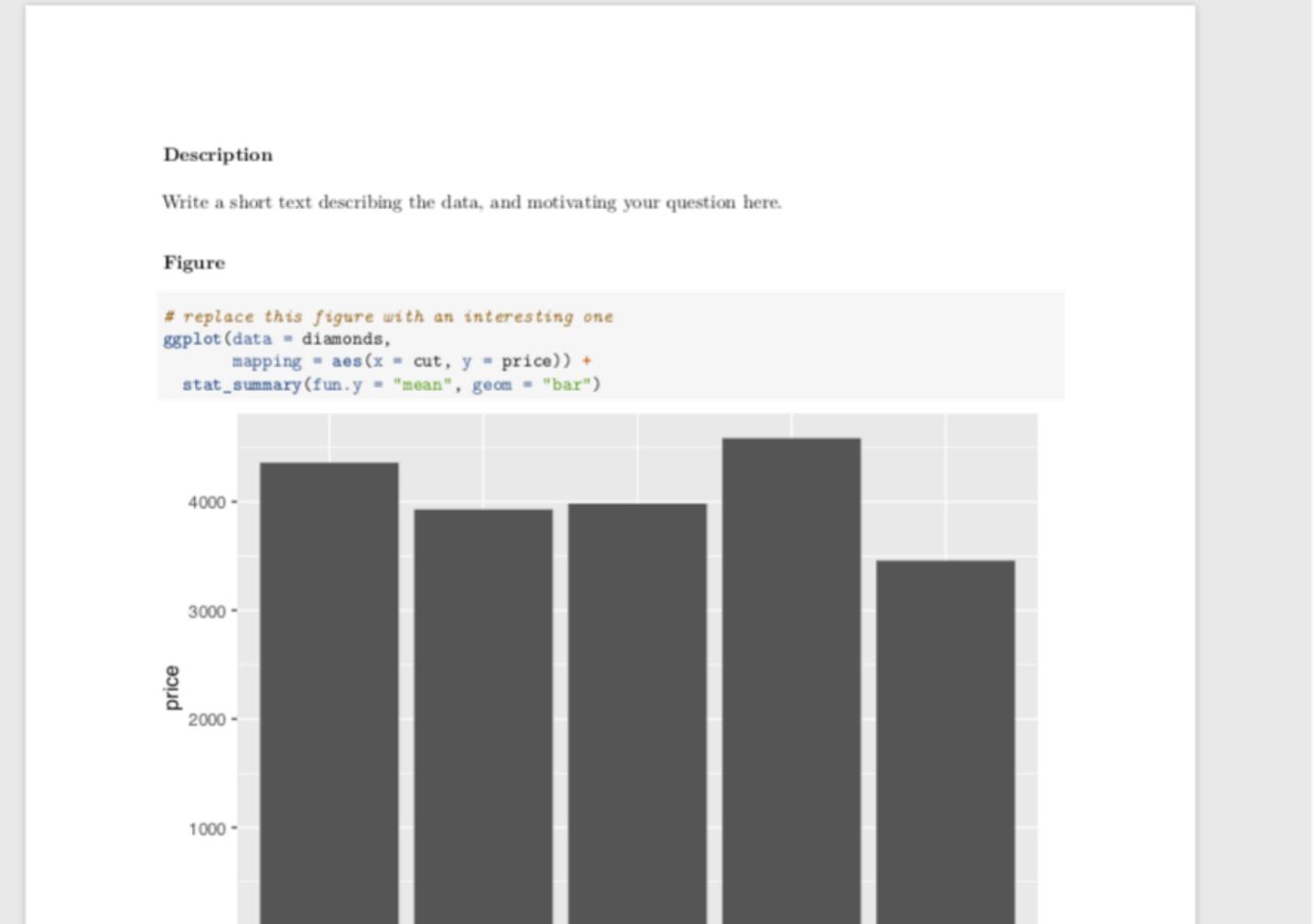
```
library("knitr")
library("tidyverse")
```

Load the data set

```
# load the data set here
```

2

should look sort of
like this ...



Homework

- install tinytex (<https://yihui.name/tinytex/r/>)
 - open 1-visualization.Rproj
 - open 1-visualization_homework.Rmd within RStudio

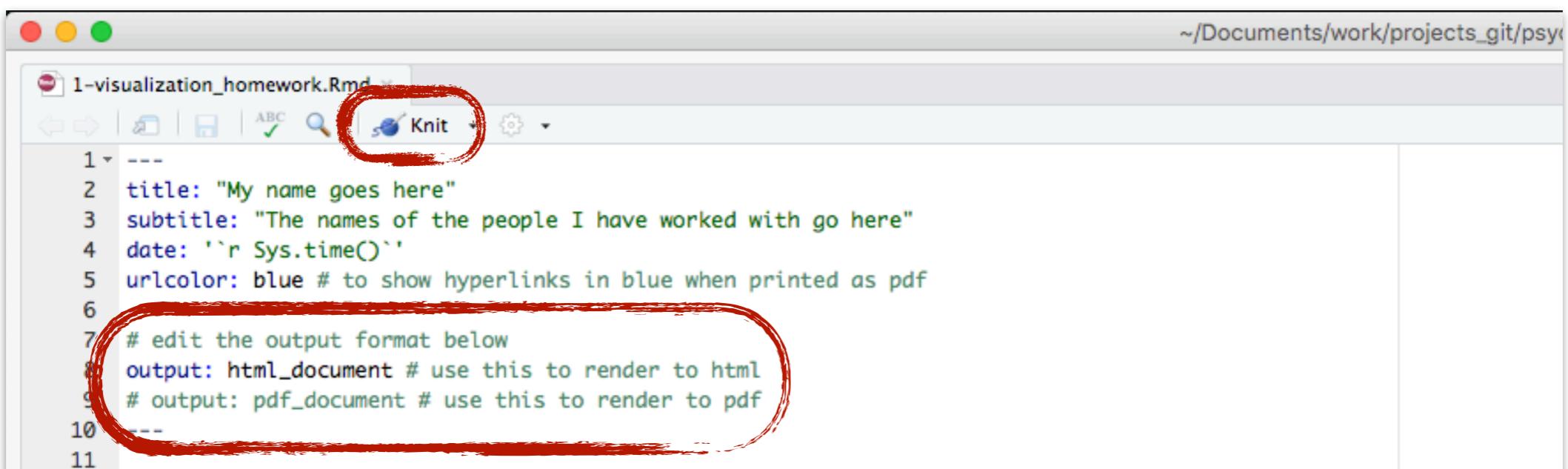
```
30 ## Install tinytex
31
32 In order to knit an RMarkdown document to a pdf file, you have to install LaTeX on your computer. The easiest way of doing so is via the `tinytex` package. Run the code in the following code chunk to do so:
33
34 ```{r eval=FALSE}
35 install.packages("tinytex")
36 tinytex::install_tinytex()
37
38 # If you experience an error like the following when trying to knit to pdf:
39 # !LaTeX Error: File 'xcolor.sty' not found.
40 # then run the following command: tinytex::tlmgr_install("xcolor")
41 # and try to knit again.
42 ```
43
44 You can find out more about the `tinytex` package \[here\](https://yihui.org/tinytex/).
```

run this code

post on Piazza if you have any trouble getting this to work

Homework

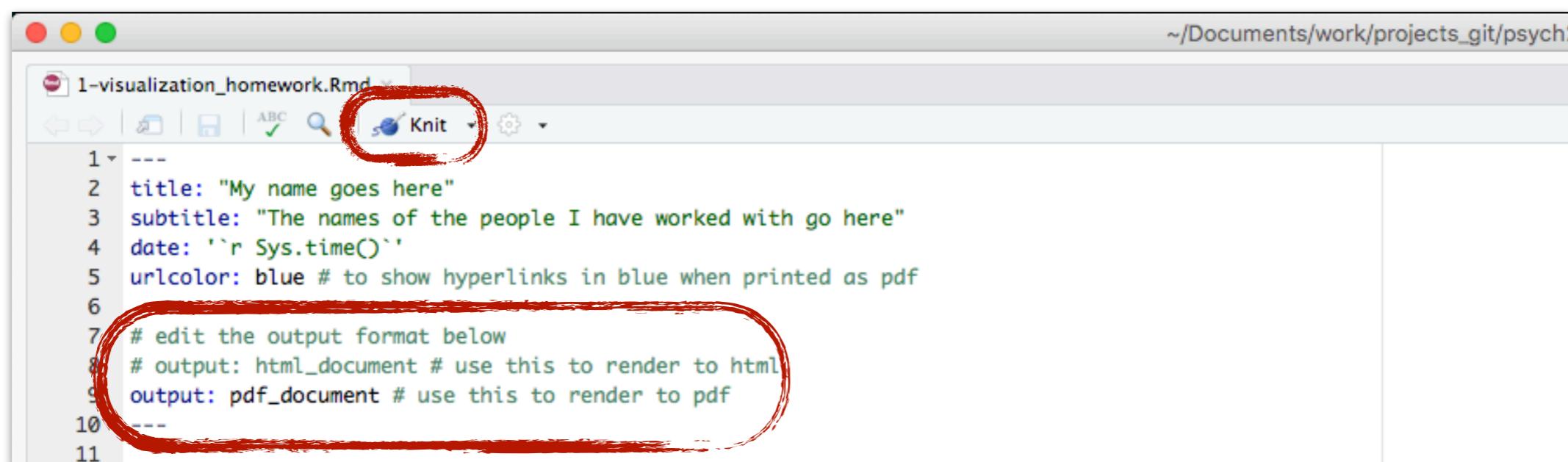
- you can change the output format from html to pdf like so ...



A screenshot of the RStudio interface showing an R Markdown file named "1-visualization_homework.Rmd". The code in the editor is:

```
1 ---  
2 title: "My name goes here"  
3 subtitle: "The names of the people I have worked with go here"  
4 date: ``r Sys.time()``  
5 urlcolor: blue # to show hyperlinks in blue when printed as pdf  
6  
7 # edit the output format below  
8 output: html_document # use this to render to html  
9 # output: pdf_document # use this to render to pdf  
10 ---  
11
```

The "Knit" button in the toolbar is circled in red. A large red circle highlights the entire section of code starting with "# edit the output format below" and ending with "# output: pdf_document # use this to render to pdf".



A screenshot of the RStudio interface showing the same R Markdown file "1-visualization_homework.Rmd". The code in the editor is identical to the previous screenshot:

```
1 ---  
2 title: "My name goes here"  
3 subtitle: "The names of the people I have worked with go here"  
4 date: ``r Sys.time()``  
5 urlcolor: blue # to show hyperlinks in blue when printed as pdf  
6  
7 # edit the output format below  
8 # output: html_document # use this to render to html  
9 output: pdf_document # use this to render to pdf  
10 ---  
11
```

The "Knit" button in the toolbar is circled in red. A large red circle highlights the entire section of code starting with "# edit the output format below" and ending with "# output: pdf_document # use this to render to pdf".

Homework

very long code without line break



```
1 ggplot(data = df.diamonds, mapping = aes(y = price, x = color, fill = color, group = cut, shape = cut, ...)) +  
2   stat_summary(fun.y = "mean", geom = "bar", color = "black") +  
3   stat_summary(fun.data = "mean_cl_boot", geom = "linerange") +  
4   facet_grid(rows = vars(cut), cols = vars(clarity))
```

Homework

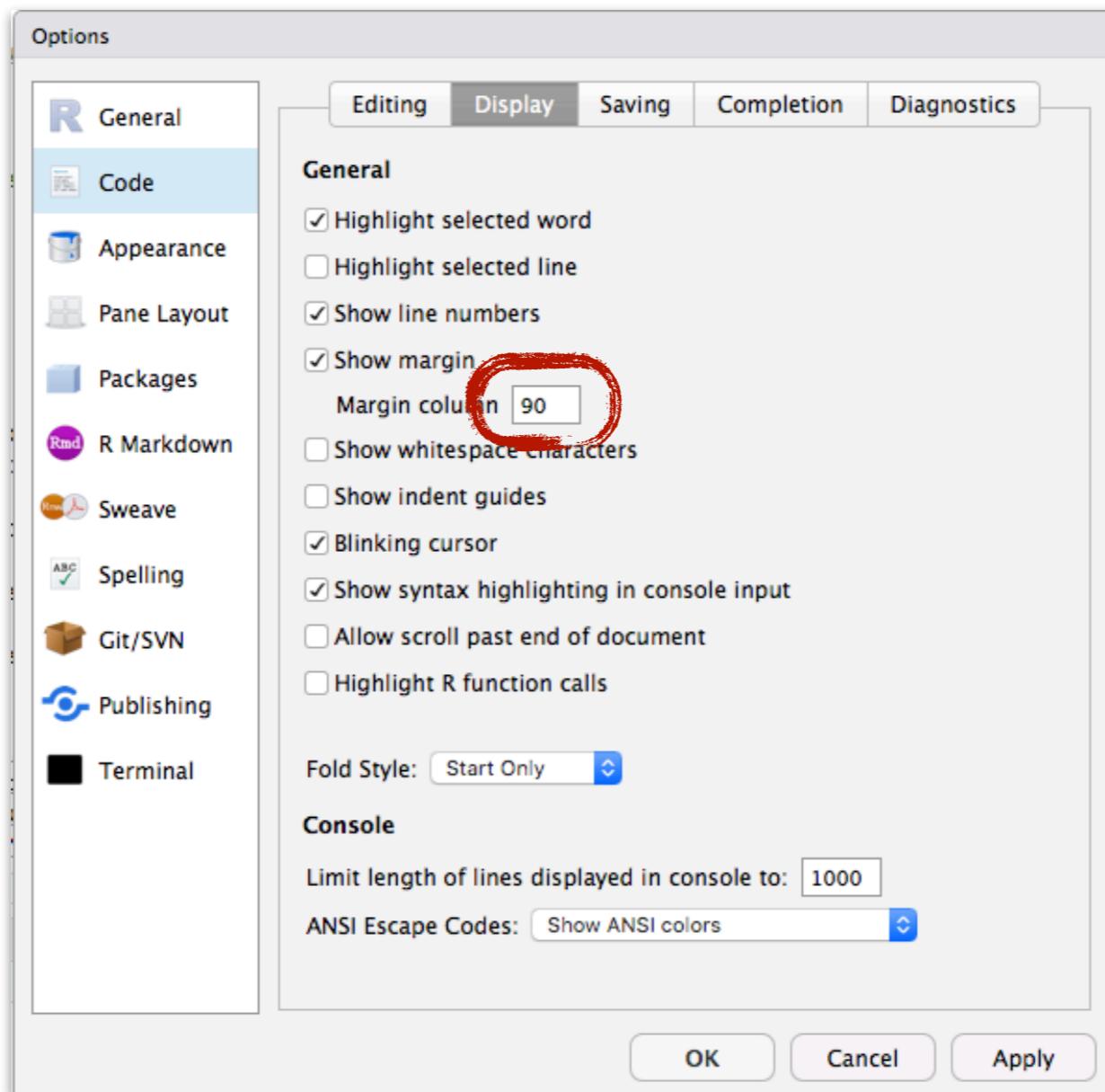
Beautiful Typesetting with LaTeX

Overfull \hbox (9.895pt too wide)

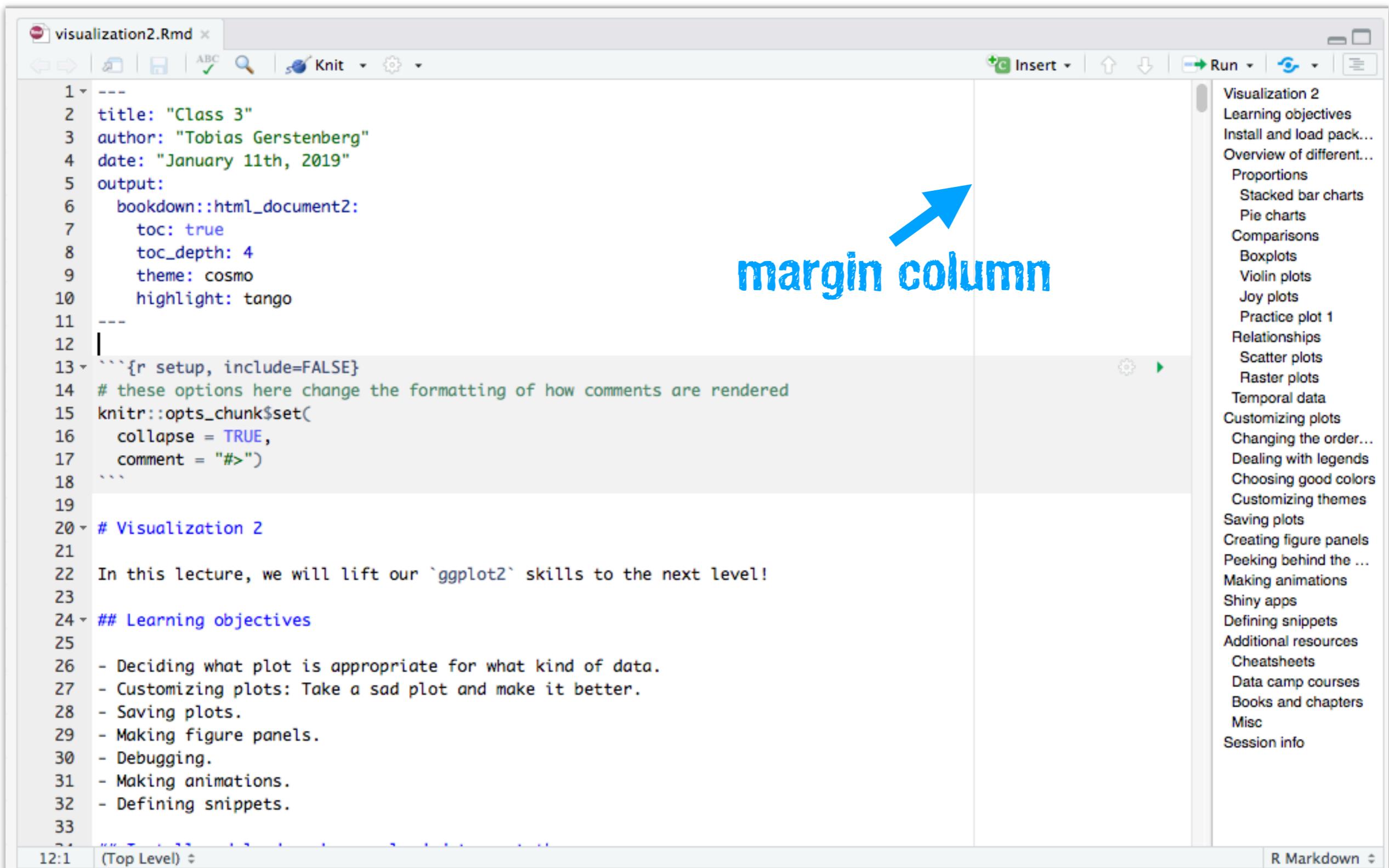


Homework

- set the margin to 90 (and make sure not to go over that margin in code blocks)
- Preferences... > Code > Display



Homework



A screenshot of the RStudio interface showing an R Markdown file named "visualization2.Rmd". The code editor pane contains the following R Markdown code:

```
1 ---  
2 title: "Class 3"  
3 author: "Tobias Gerstenberg"  
4 date: "January 11th, 2019"  
5 output:  
6   bookdown::html_document2:  
7     toc: true  
8     toc_depth: 4  
9     theme: cosmo  
10    highlight: tango  
11 ---  
12 |  
13 `r setup, include=FALSE}  
14 # these options here change the formatting of how comments are rendered  
15 knitr::opts_chunk$set(  
16   collapse = TRUE,  
17   comment = "#>")  
18 `r  
19  
20 # Visualization 2  
21  
22 In this lecture, we will lift our `ggplot2` skills to the next level!  
23  
24 ## Learning objectives  
25  
26 - Deciding what plot is appropriate for what kind of data.  
27 - Customizing plots: Take a sad plot and make it better.  
28 - Saving plots.  
29 - Making figure panels.  
30 - Debugging.  
31 - Making animations.  
32 - Defining snippets.  
33
```

The right sidebar shows a navigation tree for "Visualization 2". A large blue arrow points from the text "margin column" to the vertical line separating the code editor from the sidebar.

margin column

- Visualization 2
- Learning objectives
- Install and load pack...
- Overview of different...
- Proportions
- Stacked bar charts
- Pie charts
- Comparisons
- Boxplots
- Violin plots
- Joy plots
- Practice plot 1
- Relationships
- Scatter plots
- Raster plots
- Temporal data
- Customizing plots
- Changing the order...
- Dealing with legends
- Choosing good colors
- Customizing themes
- Saving plots
- Creating figure panels
- Peeking behind the ...
- Making animations
- Shiny apps
- Defining snippets
- Additional resources
- Cheatsheets
- Data camp courses
- Books and chapters
- Misc
- Session info

Homework

- set the margin to 90 (and make sure not to go over that margin in code blocks)
- Preferences... > Code > Display

```
# take a look at the data sets that come with the package
data(package = "fivethirtyeight")

# take a look at the help file to get more information about the different data sets (not all packages
help("fivethirtyeight")

# the "fivethirtyeight" provides a detailed overview over the different data sets with this command
vignette("fivethirtyeight", package = "fivethirtyeight")

# to load a particular data set (e.g. US_births_2000_2014, replace with the name of the data set you'd
df.data = US_births_2000_2014
```

not good

```
# take a look at the data sets that come with the package
data(package = "fivethirtyeight")

# take a look at the help file to get more information about the different data sets (not
# all packages have help files)
help("fivethirtyeight")

# the "fivethirtyeight" provides a detailed overview over the different data sets with
# this command
vignette("fivethirtyeight", package = "fivethirtyeight")

# to load a particular data set (e.g. US_births_2000_2014, replace with the name of the
# data set you'd liked to load) into your environment, run the following
df.data = US_births_2000_2014
```

only important in
code chunks!

good!

Some tips and tricks

Piazza

Practice Plot 3

Actions ▾

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")
```



this is great!

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!

rstudio

- 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

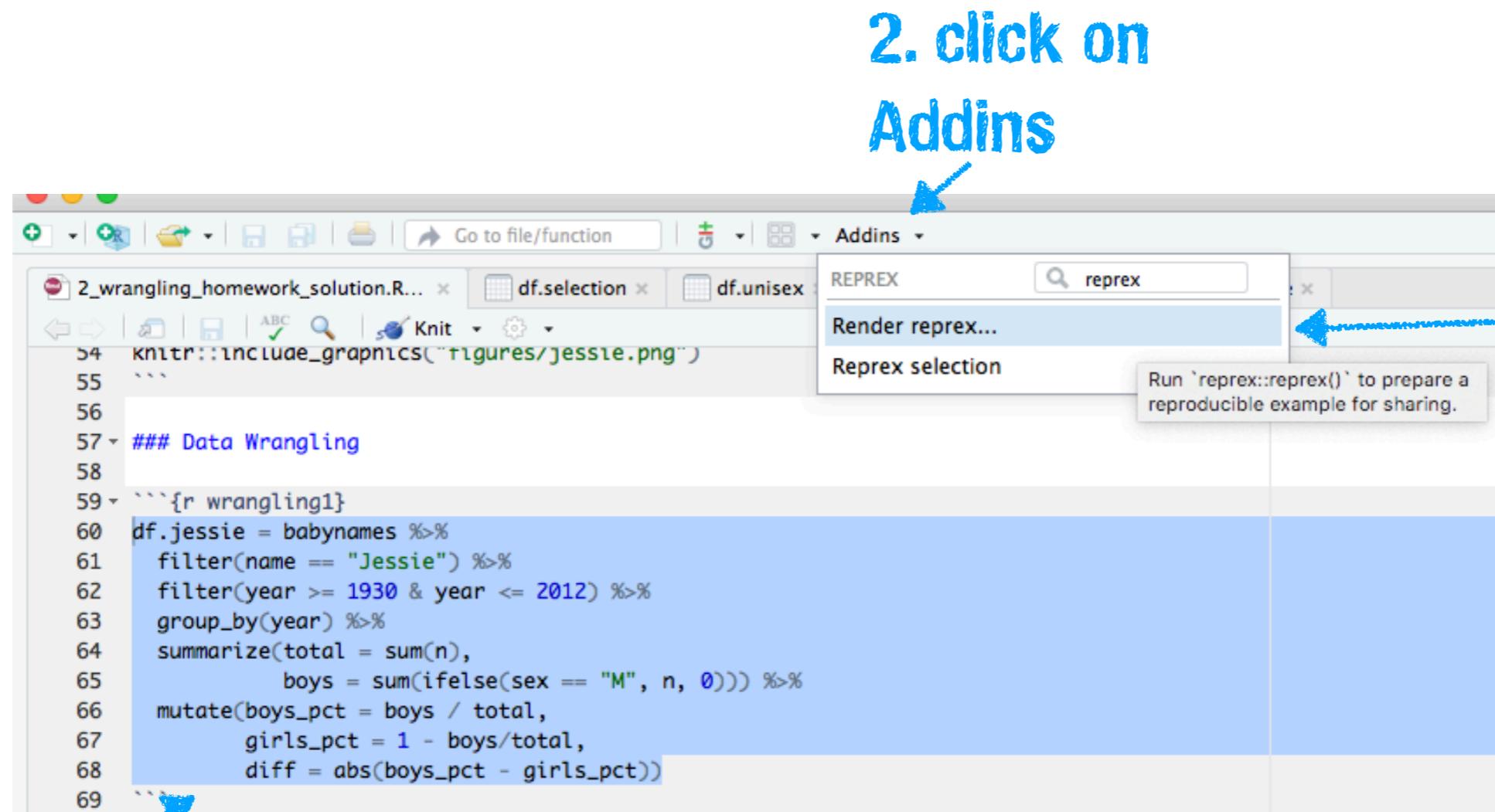
- Markdown, formatted for target venue, e.g. `gh` or `so`, or as
- R code, augmented with commented output.

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")
```

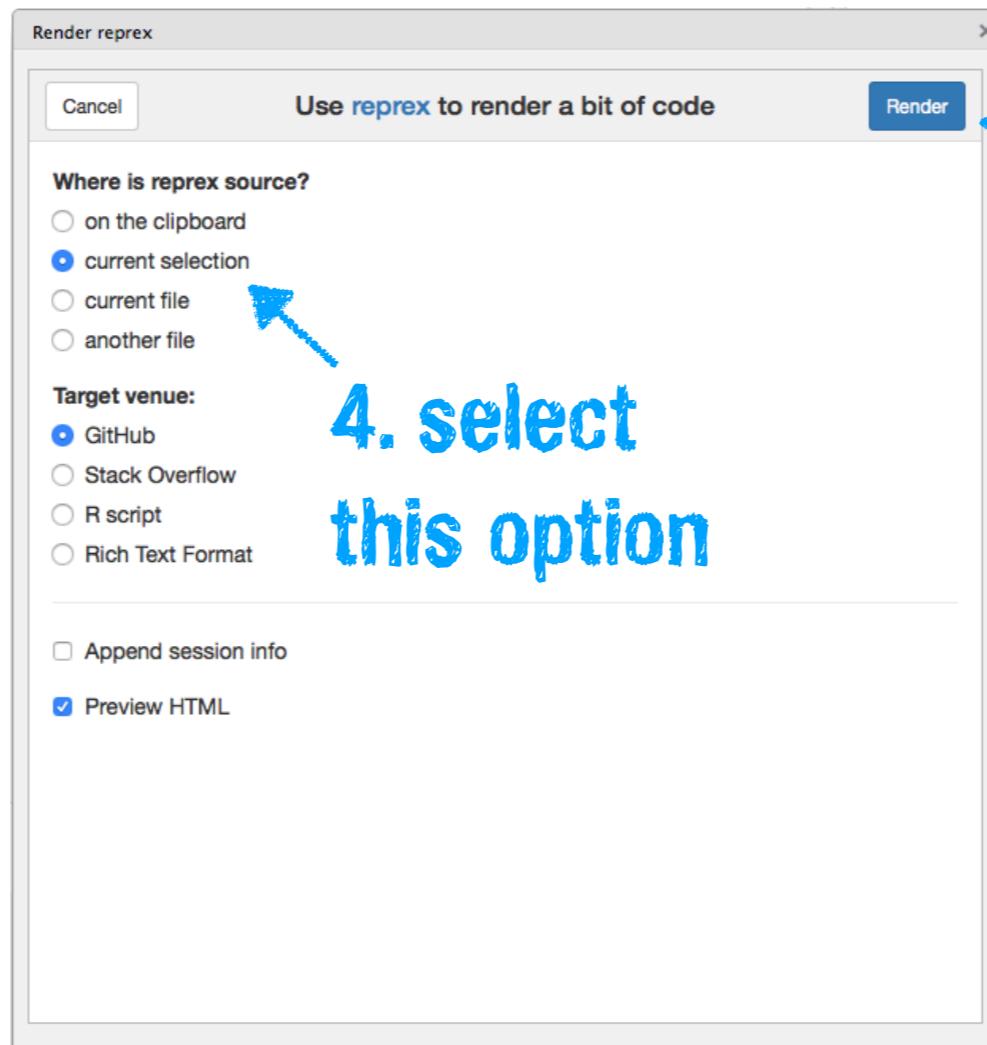


1. select
the text

2. click on
Addins

3. Render
reprex

Piazza



5. click
render

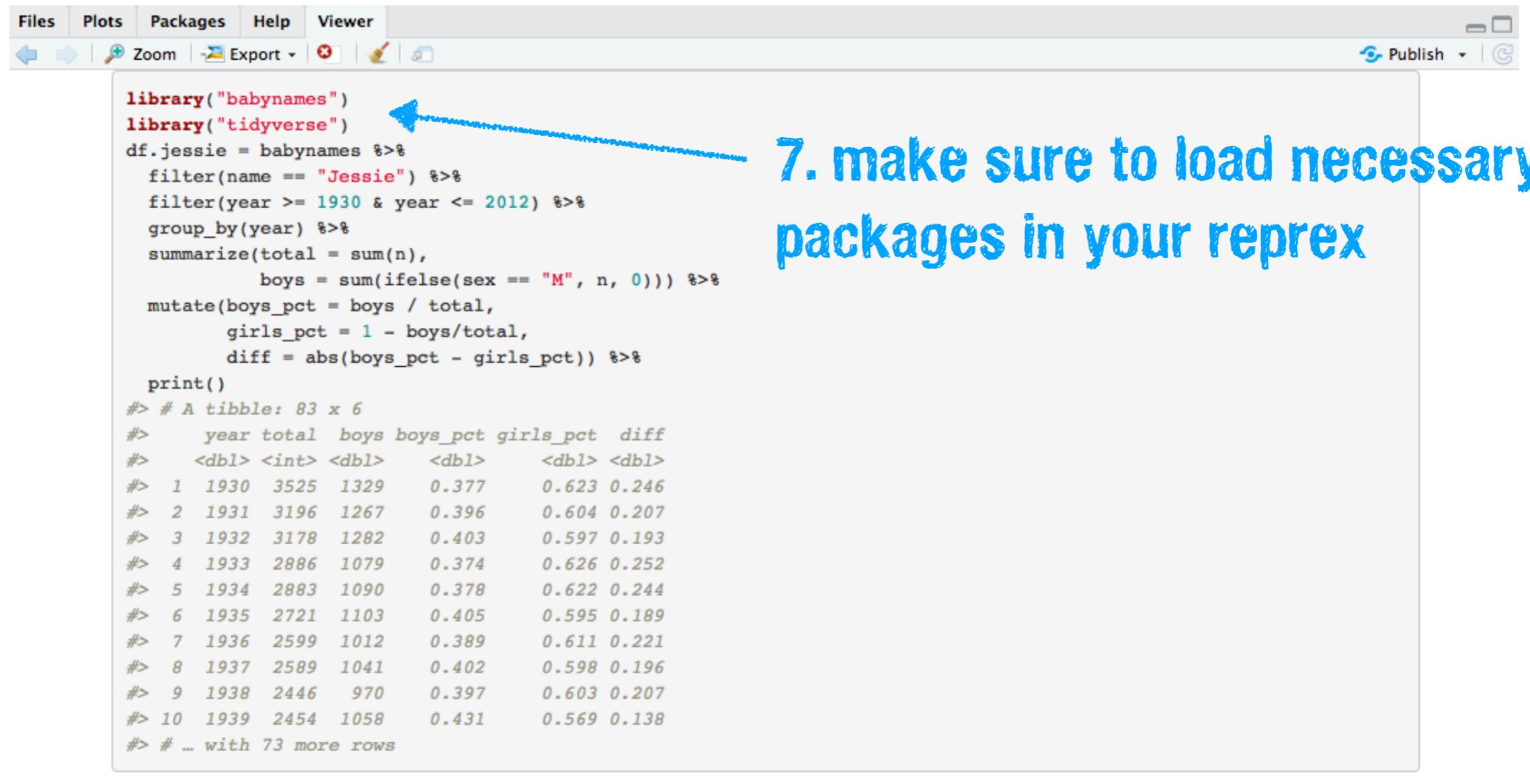
4. select
this option

6. copy and paste from the viewer

The screenshot shows the RStudio interface with the 'Viewer' tab selected in the menu bar. The main pane displays the following R code:

```
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

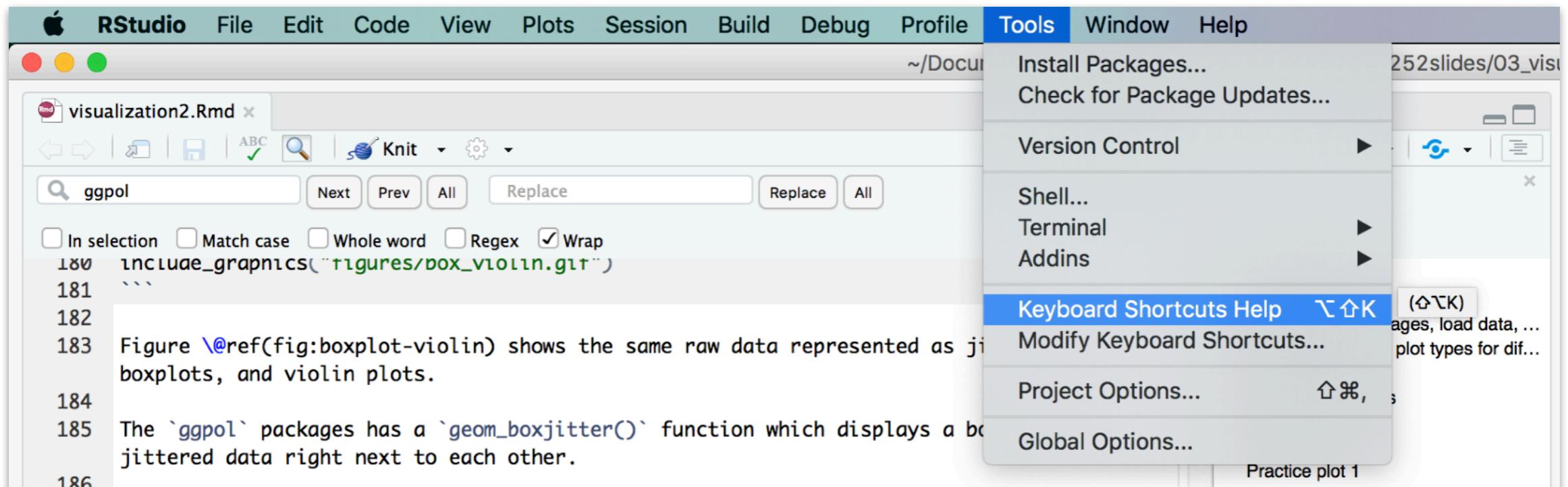


```
library("babynames")
library("tidyverse")
df.jessie = babynames %>%
  filter(name == "Jessie") %>%
  filter(year >= 1930 & year <= 2012) %>%
  group_by(year) %>%
  summarize(total = sum(n),
            boys = sum(ifelse(sex == "M", n, 0))) %>%
  mutate(boys_pct = boys / total,
        girls_pct = 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
```

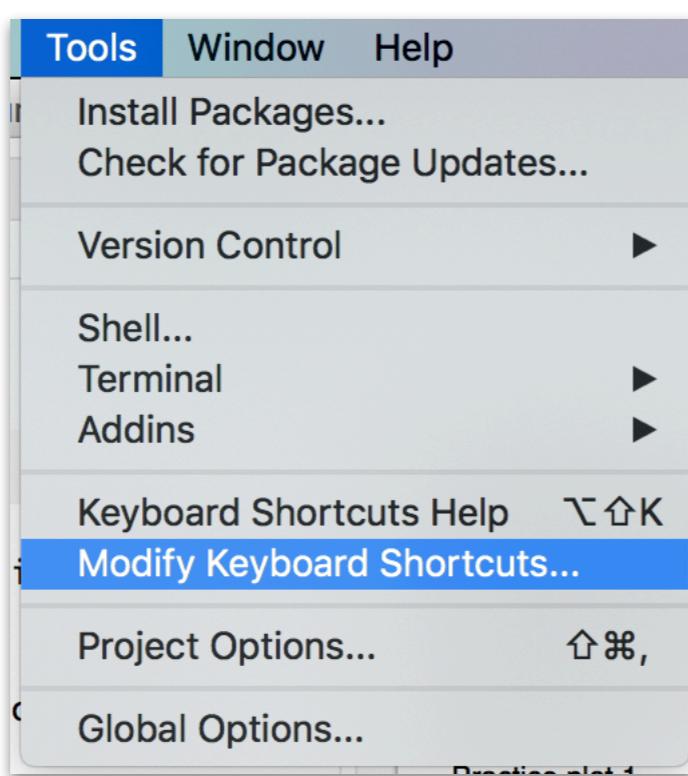
Created on 2019-01-24 by the [reprex package](#) (v0.2.1)

7. make sure to load necessary packages in your reprex

Learn the keyboard shortcuts!



... and make
your own



Name	Shortcut	Scope
Browse Addins		Workbench
Check Spelling	F7	Workbench
Check for RStudio Updates		Workbench
Clear All Breakpoints...		Workbench
Clear All Plots...		Workbench
Clear Console	Ctrl+L	Workbench
Clear Knitr Cache		Workbench
Clear Prerendered Output		Workbench
Clear Terminal Buffer		Workbench
Clear Workspace		Workbench
Close All Documents		Workbench
Close Current Document	Cmd+W	Workbench
Close Current Project		Workbench
Close Other Documents	Shift+Alt+Cmd+W	Workbench
Close Terminal		Workbench
Compile Notebook	Shift+Cmd+K	Workbench
Console on Left		Workbench
Console on Right		Workbench
Copy Current Plot to Clipboard...		Workbench

Reformatting code

```
1 ggplot(data = df.diamonds[1:150,], mapping = aes(x = color, y = price)) +  
2   # individual data points (jittered horizontally)  
3   geom_point(alpha = 0.2,  
4             position = position_jitter(width = 0.1, height = 0),  
5             size = 2) +  
6   # error bars  
7   stat_summary(fun.data = "mean_cl_boot",  
8                 geom = "linerange",  
9                 color = "black",  
10                size = 1) +  
11   # means  
12  stat_summary(fun.y = "mean",  
13                 geom = "point",  
14                 shape = 21,  
15                 fill = "red",  
16                 color = "black",  
17                 size = 4)
```

highlight code and press
cmd + i

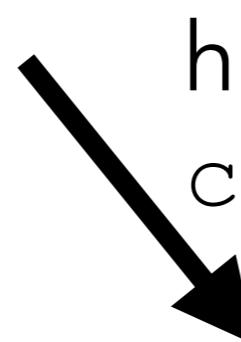


```
1 ggplot(data = df.diamonds[1:150,], mapping = aes(x = color, y = price)) +  
2   # individual data points (jittered horizontally)  
3   geom_point(alpha = 0.2,  
4             position = position_jitter(width = 0.1, height = 0),  
5             size = 2) +  
6   # error bars  
7   stat_summary(fun.data = "mean_cl_boot",  
8                 geom = "linerange",  
9                 color = "black",  
10                size = 1) +  
11   # means  
12  stat_summary(fun.y = "mean",  
13                 geom = "point",  
14                 shape = 21,  
15                 fill = "red",  
16                 color = "black",  
17                 size = 4)
```

Commenting code

```
1 ggplot(data = df.diamonds,  
2         mapping = aes(x = color, y = price)) +  
3     stat_summary(fun.y = "mean", geom = "bar")
```

highlight code and press
cmd + shift + c



```
1 # ggplot(data = df.diamonds,  
2 #           mapping = aes(x = color, y = price)) +  
3 #     stat_summary(fun.y = "mean", geom = "bar")
```

Quickly copying code

```
1 ggplot(mapping = aes(x = color, y = price), data = df.diamonds) +  
2   stat_summary(fun.y = "mean", geom = "point")
```



put cursor anywhere in line 1
cmd + shift + d

```
1 ggplot(mapping = aes(x = color, y = price), data = df.diamonds) +  
2 ggplot(mapping = aes(x = color, y = price), data = df.diamonds) +  
3   stat_summary(fun.y = "mean", geom = "point")
```

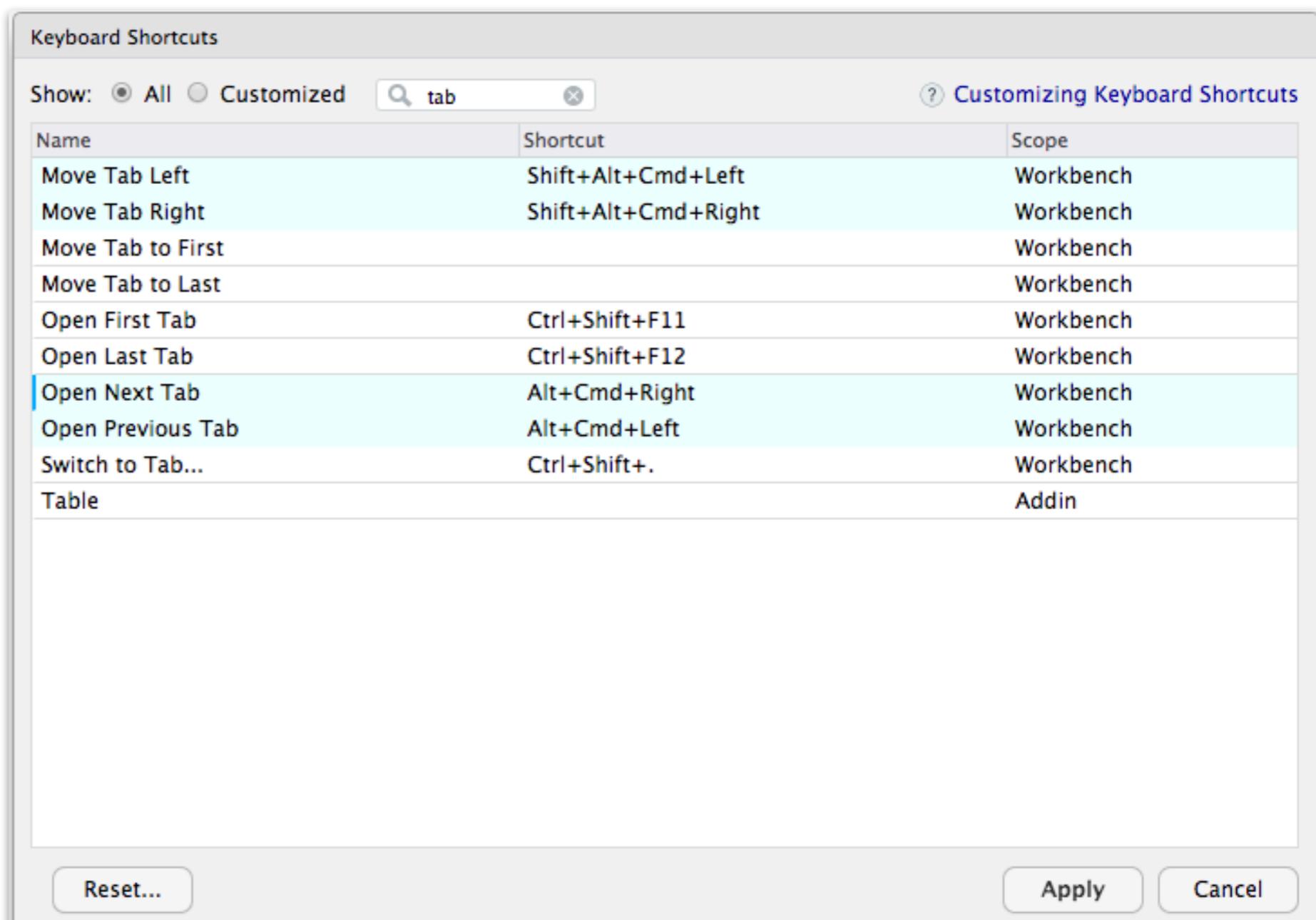
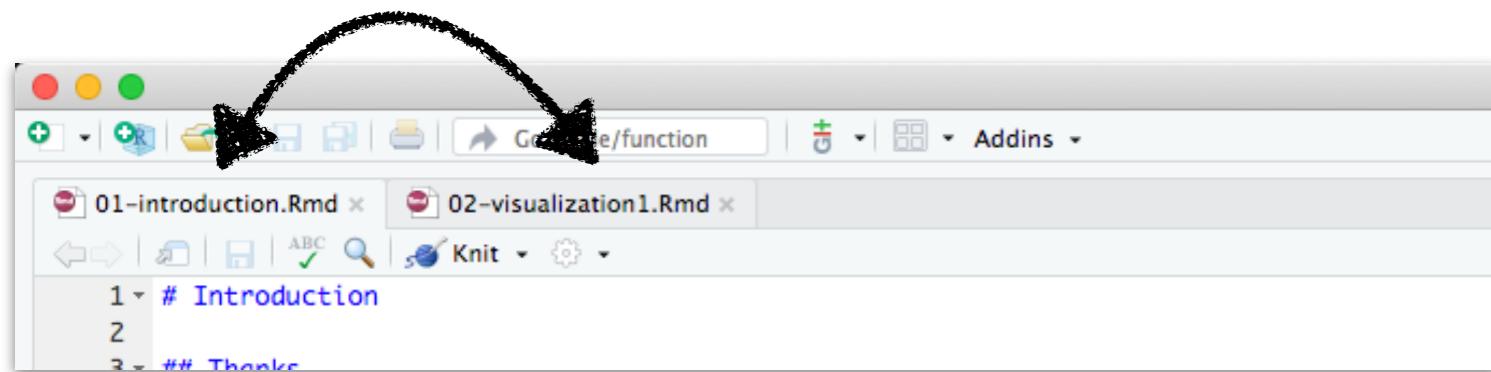


comment
cmd + shift + c

```
1 # ggplot(mapping = aes(x = color, y = price), data = df.diamonds) +  
2 ggplot(mapping = aes(x = cut, y = price), data = df.diamonds) +  
3   stat_summary(fun.y = "mean", geom = "point")
```

change

Navigating between tabs



Jumping between code chunks

```
190 Here is the help file for the `print()` function:  
191  
192 ```{r visualization1-17, echo=FALSE, fig.cap="Help file for the print()  
function.", out.width="95%"}  
193 include_graphics("figures/help_print.png")  
194 ```  
195  
196 ## Data visualization using `ggplot2`  
197  
198 We will use the `ggplot2` package to visualize data. By the end of next class,  
you'll be able to make a figure like this:  
199  
200 ```{r visualization1-18, echo=FALSE, fig.cap="What a nice figure!",  
out.width="95%"}  
201 include_graphics("figures/combined_plot.png")  
202 ```  
...
```



Keyboard Shortcuts		
Show:	All	Customized
<input type="text"/> chunk		Customizing Keyboard Shortcuts
Name	Shortcut	Scope
Restart R Session and Clear Chunk Output		Workbench
Restart R Session and Run All Chunks		Workbench
Go to Next Chunk	Alt+Cmd+Down	Editor
Go to Previous Chunk	Alt+Cmd+Up	Editor
Insert Chunk	Alt+Cmd+I	Editor

RStudio & visualization time!



I'm done.

blue

Please help.

pink

Anatomy of a nice ggplot

```
1 # ggplot call with global aesthetics
2 ggplot(data = data,
3         mapping = aes(x = cause,
4                         y = effect)) +
5     # add geometric objects (geoms)
6     geom_point() +
7     stat_summary(fun.y = "mean", geom = "point") +
8     ...
9     # add text objects
10    geom_text() +
11    annotate() +
12    # adjust axes and coordinates
13    scale_x_continuous() +
14    scale_y_continuous() +
15    coord_cartesian() +
16    # define plot title, and axis titles
17    labs(title = "Title",
18          x = "Cause",
19          y = "Effect") +
20    # change global aspects of the plot
21    theme(text = element_text(size = 20),
22           plot.margin = margin(t = 1, b = 1, l = 0.5, r = 0.5, unit = "cm")) +
23    # save the plot
24    ggsave(filename = "super_nice_plot.pdf",
25           width = 8,
26           height = 6)
```

The diagram illustrates the components of a ggplot command with handwritten annotations:

- A blue arrow points from the word "what?" to the first line of the mapping section: `mapping = aes(x = cause, y = effect)) +`.
- A blue arrow points from the word "how?" to the second line of the geoms section: `geom_point() +`.
- A blue arrow points from the text "add some text?" to the first line of the text objects section: `geom_text() +`.
- A blue arrow points from the text "\"local\" adjustments" to the last three lines of the plot area: `scale_x_continuous() +`, `scale_y_continuous() +`, and `coord_cartesian() +`.
- A blue arrow points from the text "\"global\" adjustments" to the line: `theme(text = element_text(size = 20),`.
- A blue arrow points from the text "save the beauty!" to the final line: `ggsave(filename = "super_nice_plot.pdf", width = 8, height = 6)`.

Feedback

How was the pace of today's class?

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

How happy were you with today's class overall?



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

Thank you!