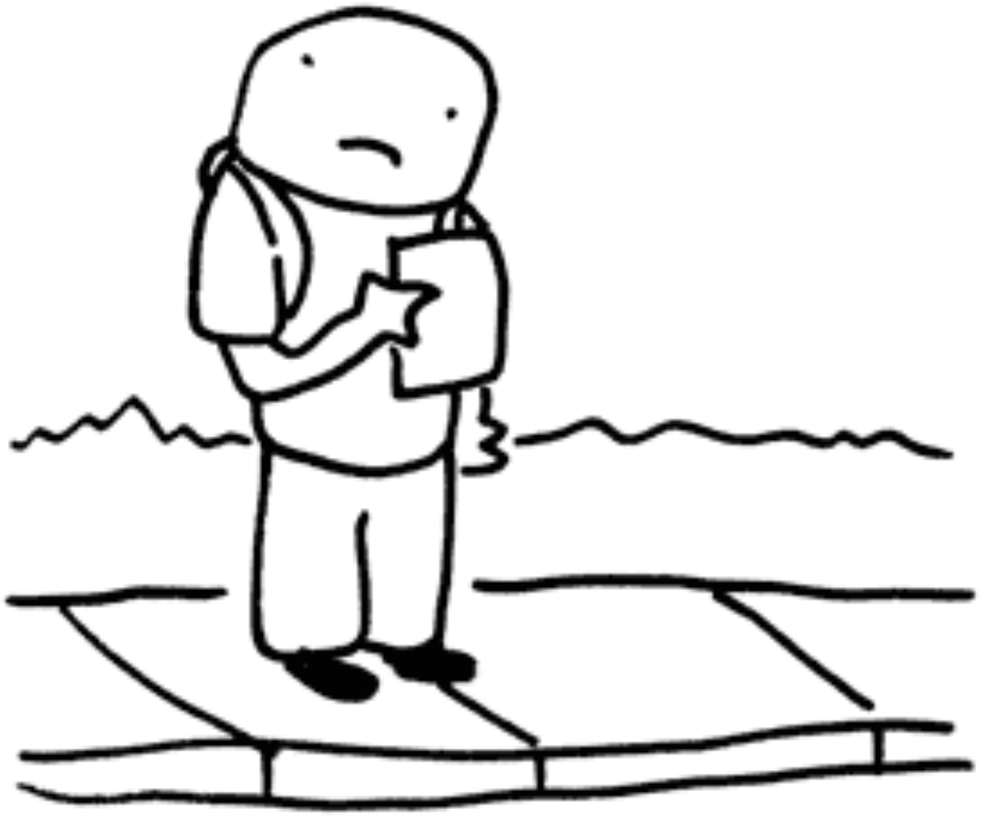
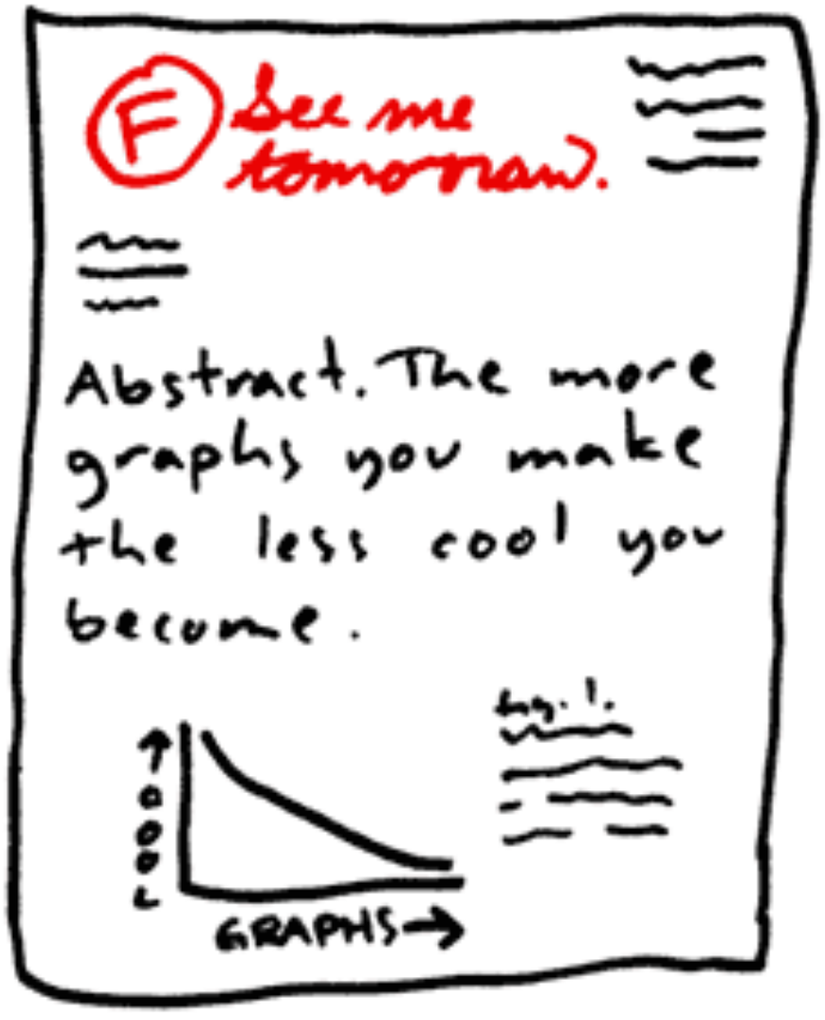
Please keep rows
3 and 6 free again!

... and get stickies

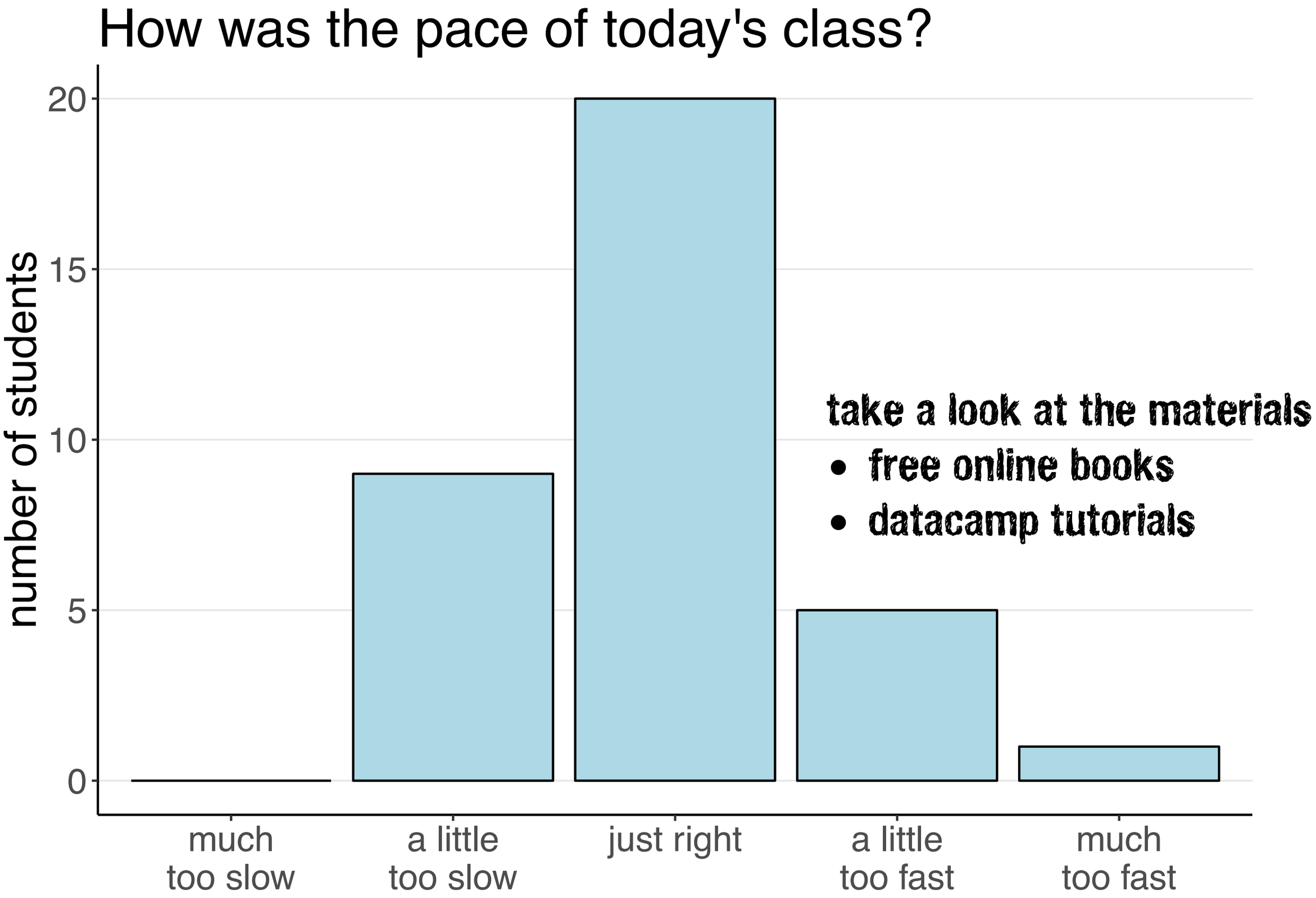
# Visualization 2



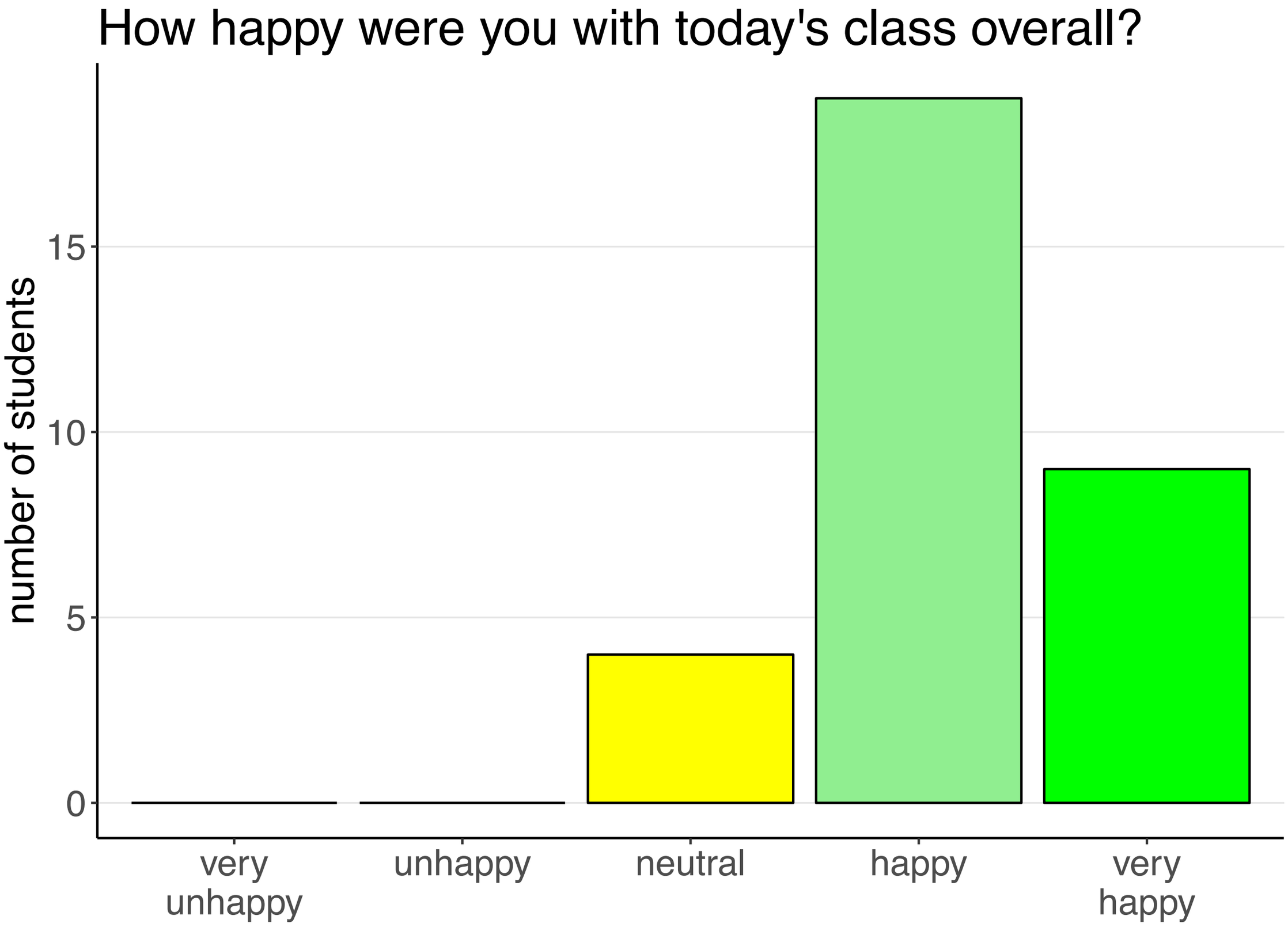


# Your feedback

# Your feedback



# Your feedback



# Your feedback

Good class! **Stress testing** the tech related stuff would be helpful (i.e. make sure figures are available for students). But I'm sure you were thinking that over

**we'll do more stress testing!**

# Your feedback

**Writing all the packages we need to install in the Rmd file somewhere so we can just copy and paste into console, instead of installing along the way**. I did appreciate it being in the email, but it's easier to have it in Rmd so that I don't have to have another window open

**good idea! we'll do that from now onwards**

**(might still miss a few here and there)**

# Your feedback

Maybe **telling everyone which line we are at now** would help.

Sometimes I lost track of where we were in the .Rmd file

**thanks! will try and do that better this time**

# Your feedback

I really liked the setup of going through the markdown file and doing examples together. **It made very easy to jump ahead when we finish early and made it feel like a very efficient use of time**

good to know!

# Your feedback

I think it may be more useful to have us to the practice exercises at home/outside of class and **use class time to go over ideas/ concepts** etc...

this will become more the focus as we go along

# Your feedback

**are students without programming experience able to keep up?**

I have experience but I felt I had to draw on it a lot in order to complete what was going on

**it'll be tough**
**but with the help of Datacamp it should be doable**

# Your feedback

The walkthrough of ggplot made things much clearer! I'm hoping to learn how to apply these skills **with actual psych data that's a little less straightforward** (like how to plot the output of a regression analysis, etc.).

**you'll get to deal with messy data soon :)**

# Logistics

# Introductory survey

**https://tinyurl.com/psych252survey**

# Final project

Final reports will be due on **Thursday, March 21st at 10pm**.

(same day that we'll have project presentations in the morning)

# Sections

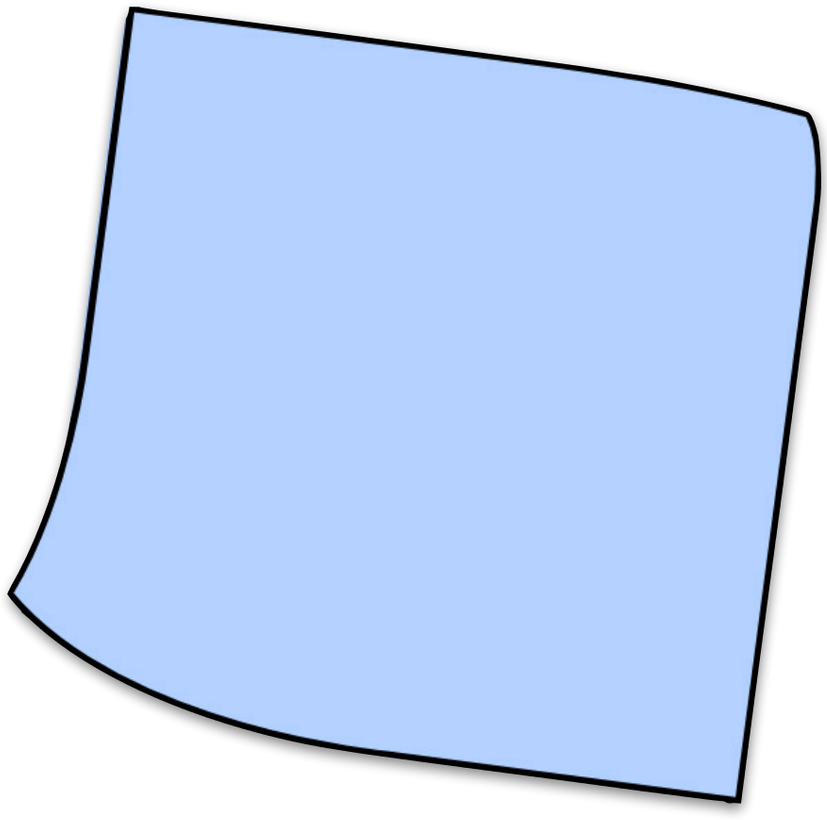
- sections aren't mandatory but strongly recommended
- learning a new language takes practice
- I can only offer you a glimpse in class
- TAs will be able to help you individually in sections

# Sections

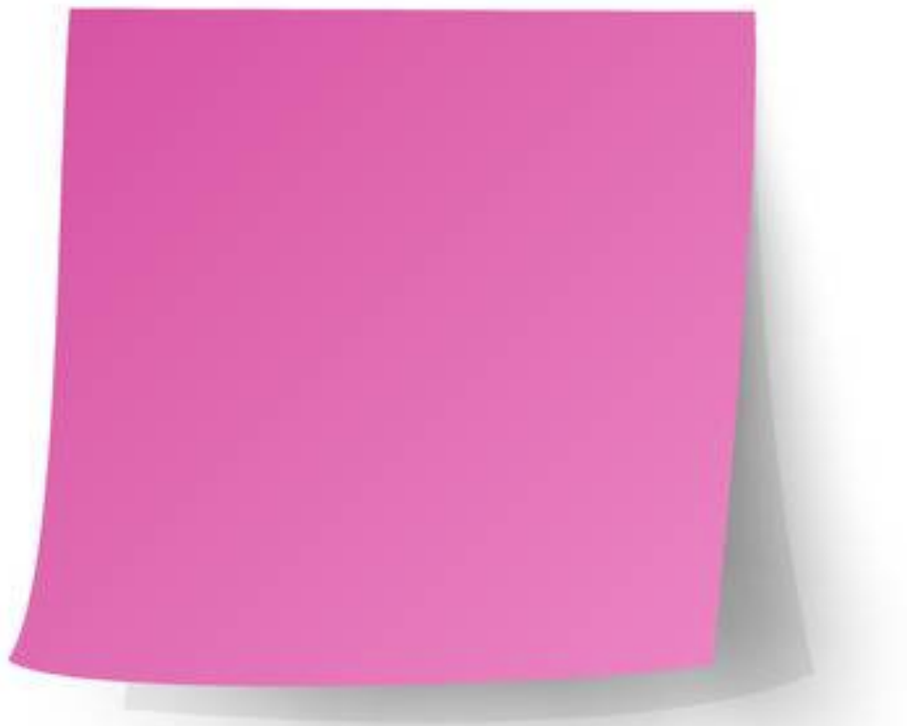
| | Andrew Lampinen | Shao-Fang (Pam) Wang |
|---|---|---|
| |  |  |
| Role: | Teaching assistant | Teaching assistant |
| Email: | lampinen@stanford.edu | shaofang@stanford.edu |
| Office: | 316 | 409 |
| Office hours: | Friday 12:30-1:30pm | Wednesday 1-2pm |
| Section: | Friday 1:30-2:20pm<br>in 160-314 | Wednesday 2:30-3:20pm<br>in 160-326 |

# Coding

Small point: Mike Frank used pink sticky notes for when you're having issues and blue sticky notes for when you're complete. This class uses the opposite system (pink = good, blue = bad), and that makes it slightly confusing.



**blue**



**pink**

# Coding


**blue**

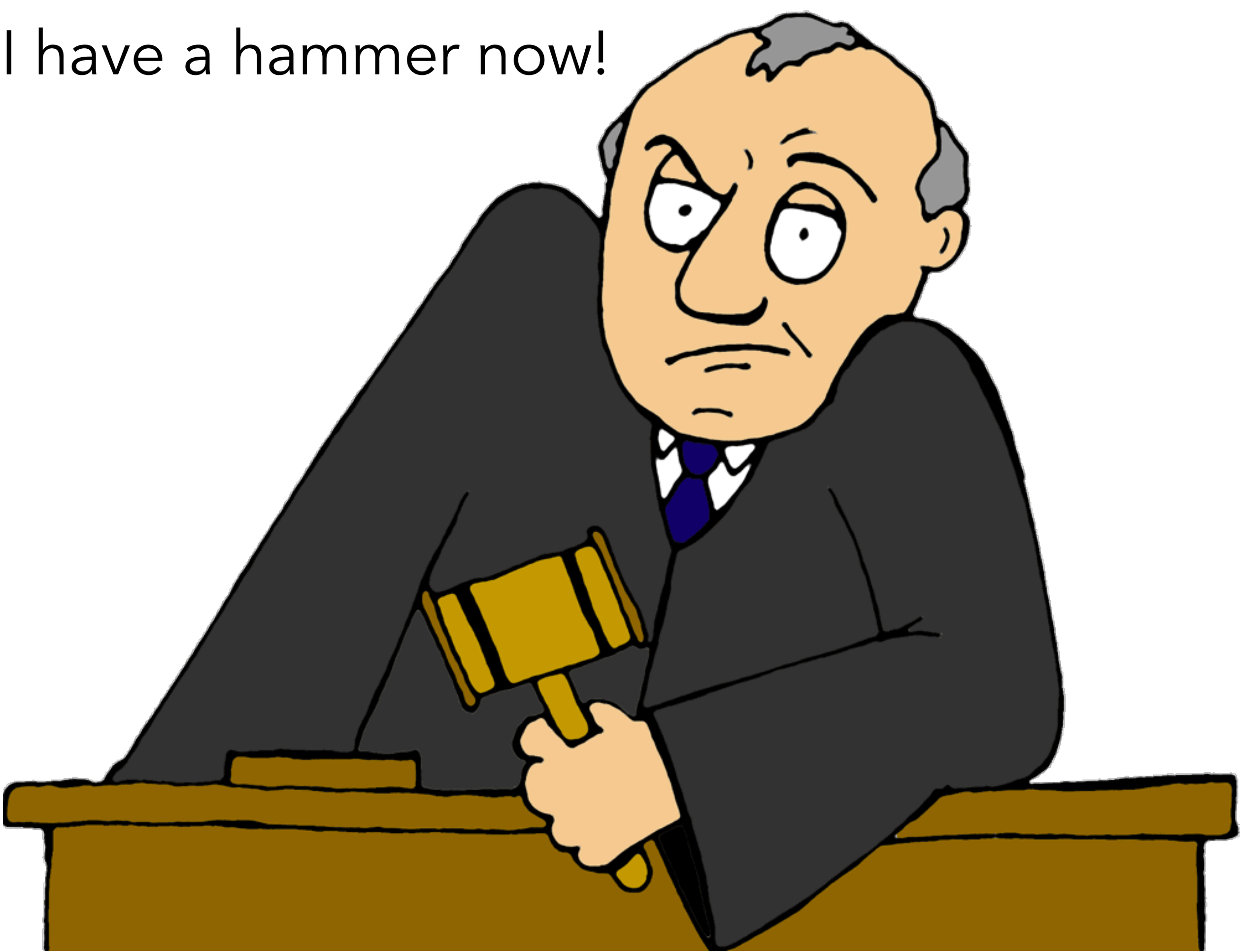

**pink**

# Coding


**blue**

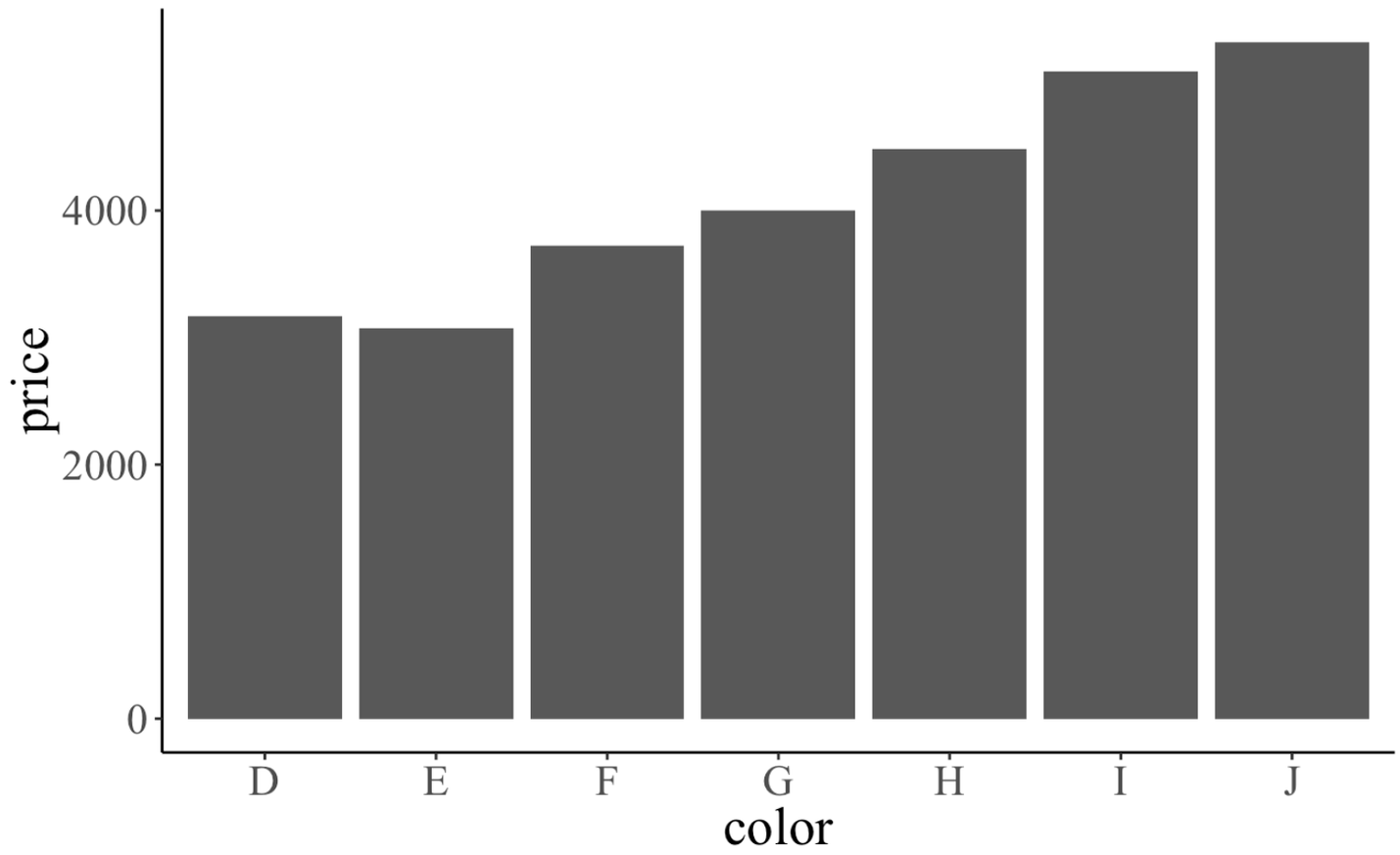

**pink**

I have a hammer now!



# Things that came up

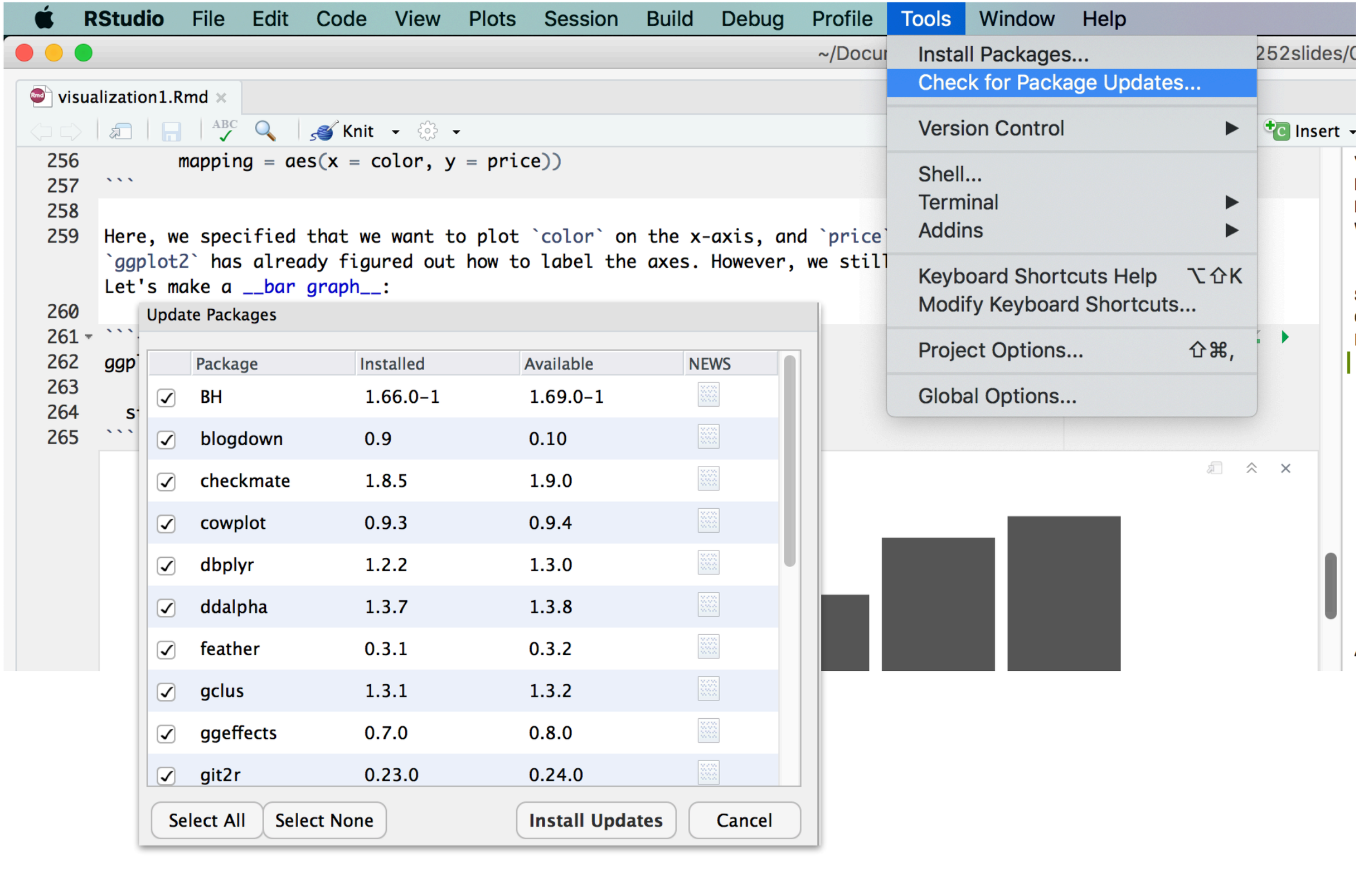
# Changing the font




```
1 ggplot(data = df.diamonds,
2          mapping = aes(x = color, y = price)) +
3   stat_summary(fun.y = "mean", geom = "bar") +
4   theme(text = element_text(family = "Times New Roman"))
```

lots of customization (fonts, legends, axes, space, ...)!

# Keeping your packages up to data



# Loading packages in R

```
1 library("knitr")      # for rendering the RMarkdown file
2 library("patchwork")  # for making figure panels
3 library("ggpol")      # for making fancy boxplots
4 library("ggridges")   # for making joyplots
5 library("gganimate")  # for making animations
6 library("gapminder")  # data available from Gapminder.org
7 library("tidyverse")  # for many cool things
```

- functions in later packages override functions of the same name from earlier packages
- always load `library("tidyverse")` last (since it's a collection of packages with many functions)
- if you want to be extra safe, specify a function with the package name in front `dplyr::select()`

# Reformatting code

```
ggplot(data = df.diamonds[1:150,], mapping = aes(x = color, y = price)) +
  # individual data points (jittered horizontally)
      geom_point(alpha = 0.2,
                 position = position_jitter(width = 0.1, height = 0),
             size = 2) +
  # error bars
  stat_summary(fun.data = "mean_cl_boot",
               geom = "linerange",
      color = "black",
                   size = 1) +
        # means
stat_summary(fun.y = "mean",
               geom = "point",
shape = 21,
               fill = "red",
               color = "black",
               size = 4)
```

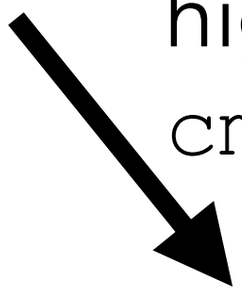
highlight code and press

`cmd/ctrl + i`

```
ggplot(data = df.diamonds[1:150,], mapping = aes(x = color, y = price)) +
  # individual data points (jittered horizontally)
  geom_point(alpha = 0.2,
             position = position_jitter(width = 0.1, height = 0),
             size = 2) +
  # error bars
  stat_summary(fun.data = "mean_cl_boot",
               geom = "linerange",
               color = "black",
               size = 1) +
  # means
  stat_summary(fun.y = "mean",
               geom = "point",
               shape = 21,
               fill = "red",
               color = "black",
               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/ctrl + 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/ctrl + 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")
```

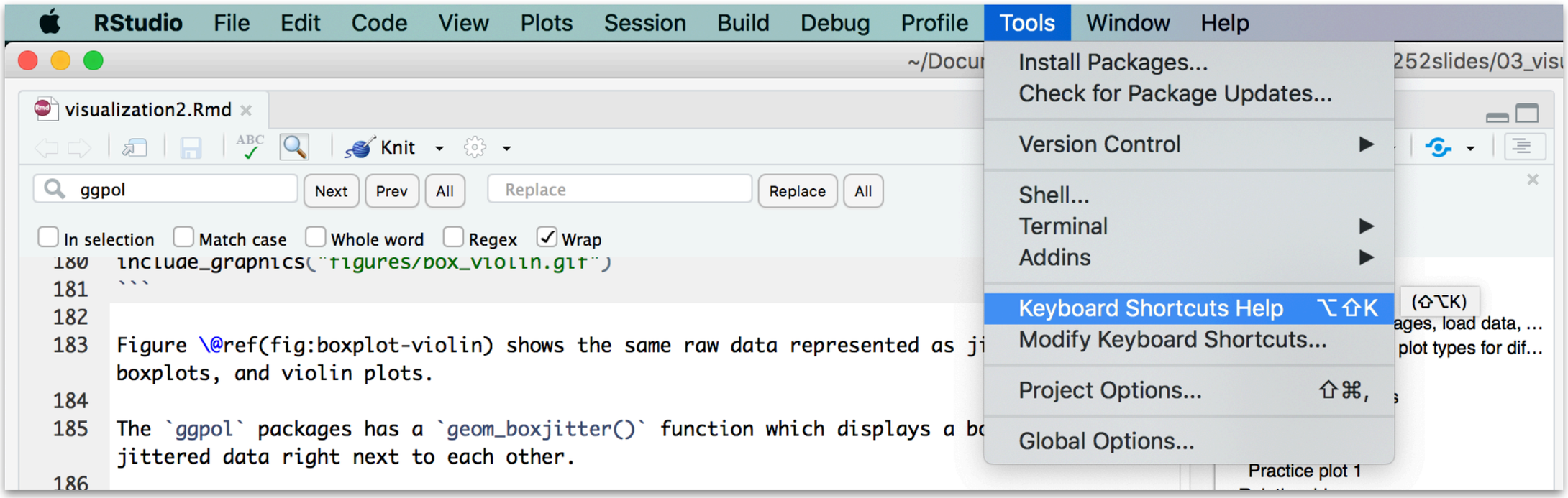
comment

`cmd/ctrl + shift + c`

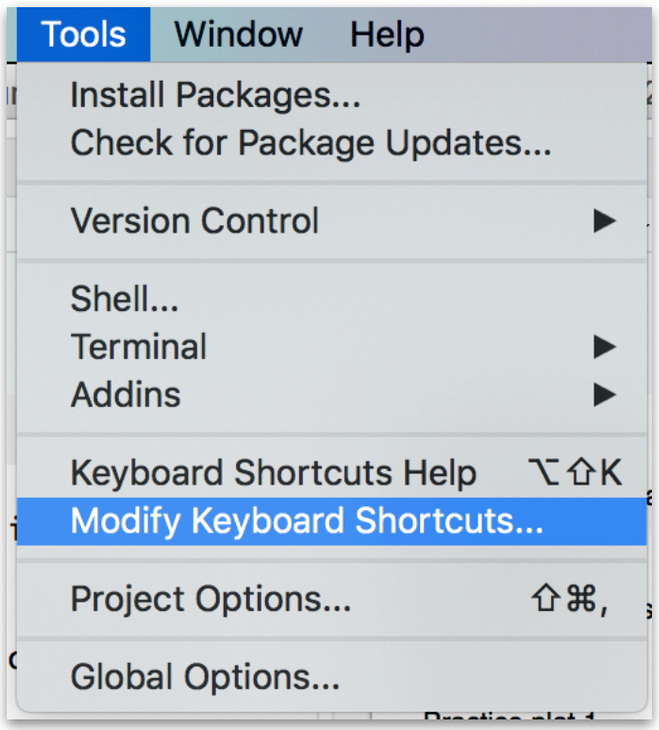
```
1 # ggplot(mapping = aes(x = color, y = price), data = df.diamonds) +
2 ggplot(mapping = aes(x = cut, y = price), data = df.diamonds) +
3   stat_summary(fun.y = "mean", geom = "point")
```

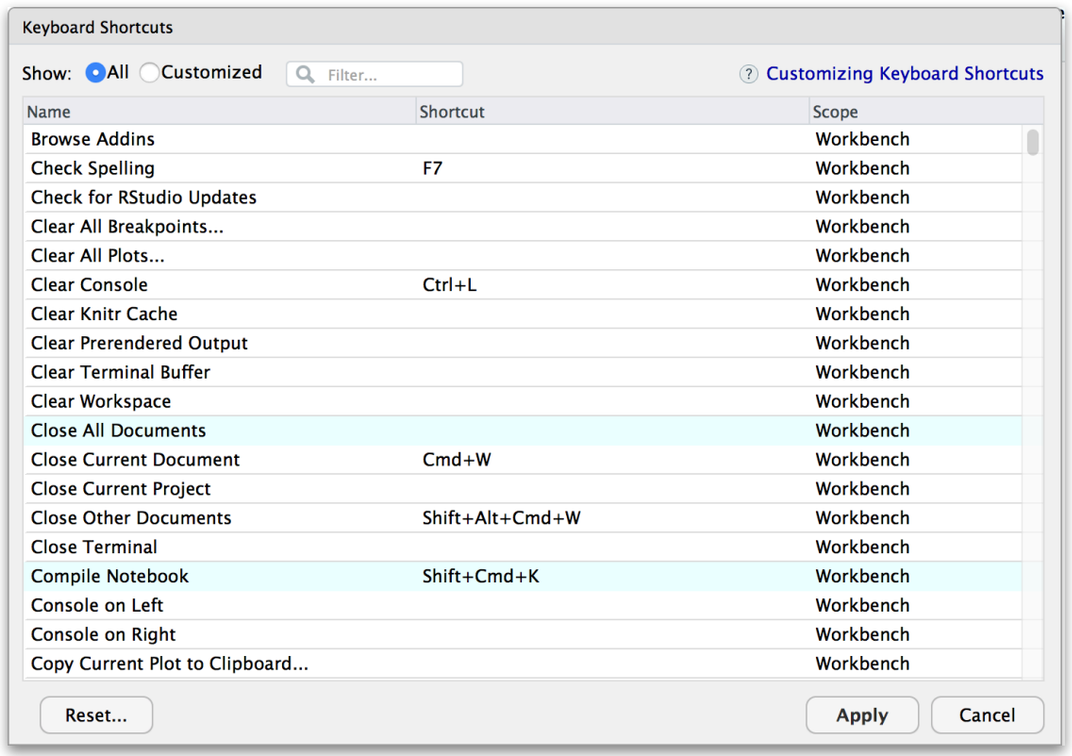
change

# Learn the keyboard shortcuts!




... and make your own



# RStudio & visualization time!

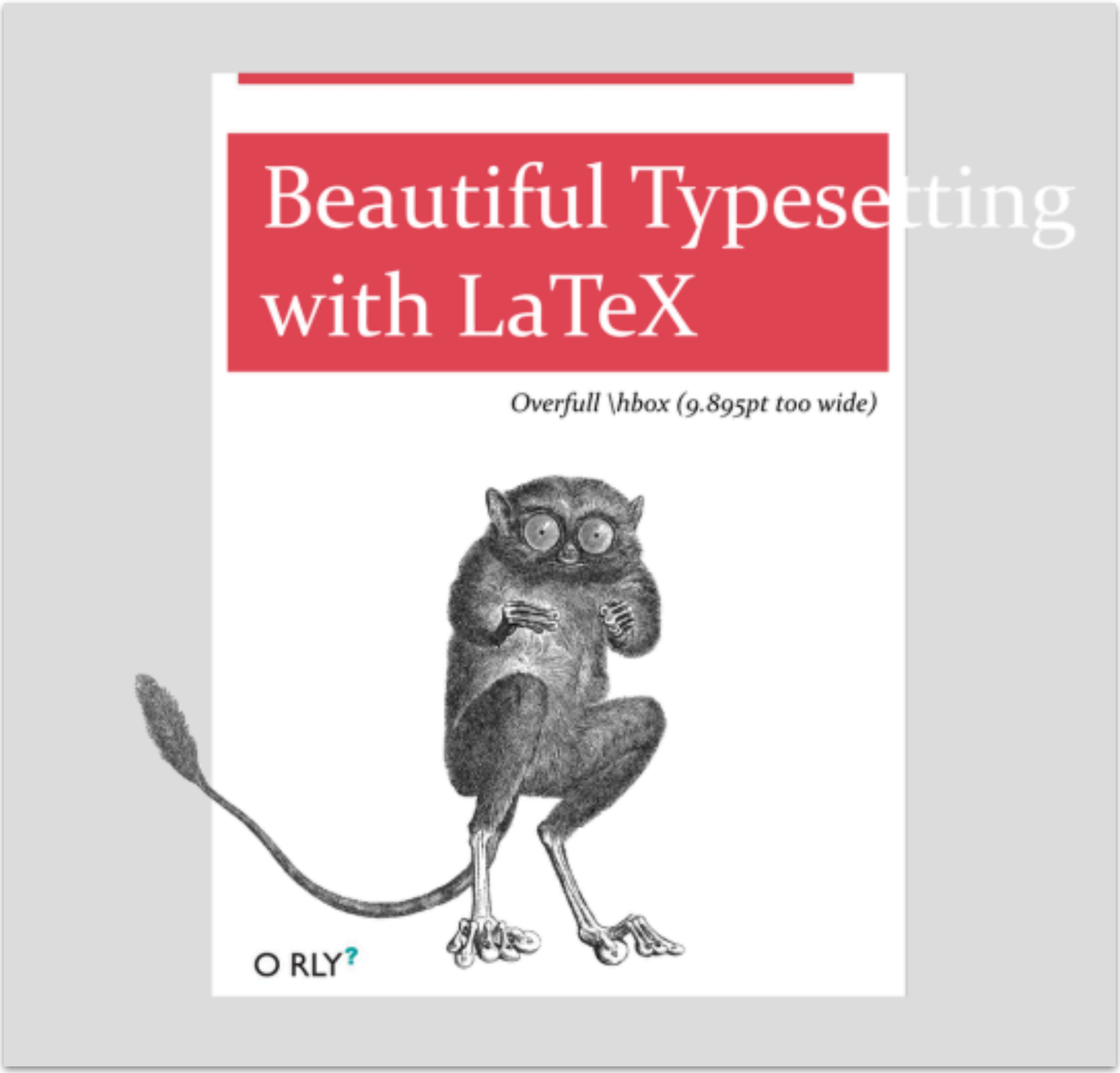
# Homework

# Homework

very long code without line break

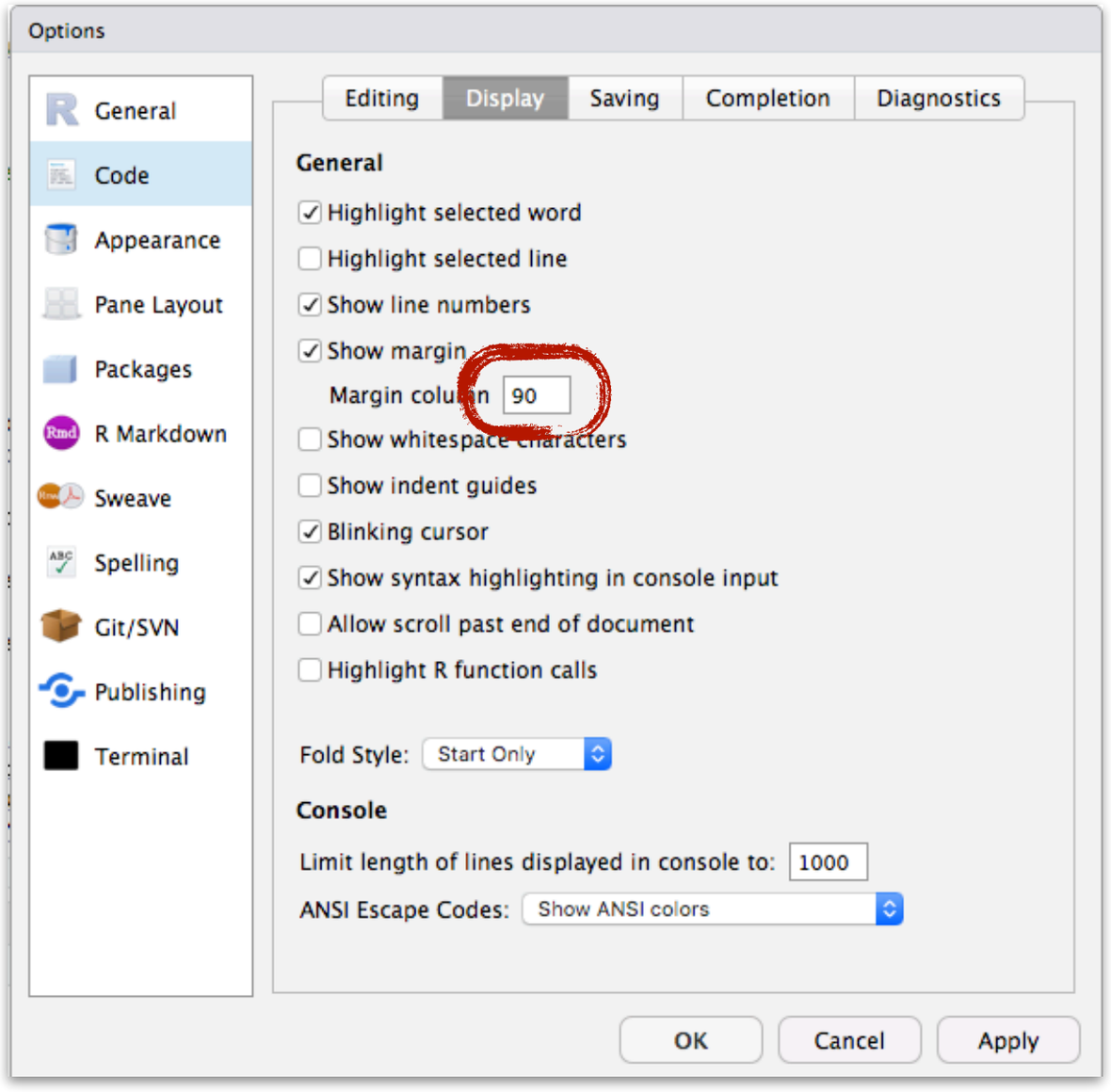
```
1 ggplot(data = df.diamonds, mapping = aes(y = price, x = color, fill = color, group = cut, shape = cut, ...)) +
2   stat_summary(fun.y = "mean", geom = "bar", color = "black") +
3   stat_summary(fun.data = "mean_cl_boot", geom = "linerange") +
4   facet_grid(rows = vars(cut), cols = vars(clarity))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

# Homework


Beautiful Typesetting
with LaTeX
Overfull \hbox (9.895pt too wide)
O RLY?

# Homework

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



# 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

only important in code chunks!

```
# 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
```

good!

# Homework

visualization2.Rmd

Knit | Insert | Run

```
---
title: "Class 3"
author: "Tobias Gerstenberg"
date: "January 11th, 2019"
output:
  bookdown::html_document2:
    toc: true
    toc_depth: 4
    theme: cosmo
    highlight: tango
---

```{r setup, include=FALSE}
# these options here change the formatting of how comments are rendered
knitr::opts_chunk$set(
  collapse = TRUE,
  comment = "#>")
```

# Visualization 2

In this lecture, we will lift our `ggplot2` skills to the next level!

## Learning objectives

- 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.
```

margin column



12:1 (Top Level) R Markdown

# Feedback