Lab 9 - Math 58 / 58b: wrangling & graphing quantitative data

done during lab April 1 or 3, 2020

not due

Lab Goals

- revisit `tidyverse` commands for wrangling quantitative data
- revisit `ggplot2` for graphing quantitative data

Getting Started

Load packages & data

For the lab, we'll use functions from the `tidyverse` (which includes all the `ggplot2` functions). The data come from the datasets provided by OpenIntro. These observations are Google stock data from 2006 to early 2014. Data from the first day of each month was collected (unless the first day of the month was a weekend or holiday, in which case the data is from the first day of the month when the stock price was available). https://www.openintro.org/data/index.php?data=goog

Note that the stock prices are in dollars per share. `volume` is the number of shares traded on that day. The Google stock split right after the last date in the dataset, so the prices in the rest of 2014 (and beyond) aren't comparable to the data contained here.

```
library(tidyverse)
library(lubridate)


goog <- read_csv("https://www.openintro.org/data/csv/goog.csv")
goog <- goog %>%
  select(-adj_close) %>%
  mutate(year = lubridate::year(date), month = lubridate::month(date))
```

Structure of the lab

For the first half of the lab, we'll try a variety of data summarizing techniques. Note that we can also wrangle the data (e.g., filter, sort, etc.). Make sure you understand how each of the summary values is calculated – mean, median, standard deviation, 25 quartile, 75 quartile, interquartile range, range.

> Recall, there is a data wrangling cheat sheet at: https://github.com/rstudio/cheatsheets/raw/master/data-transformation.pdf

For the second half of the lab, we'll try different visualizations to see how the data can be represented graphically. Feel free to play around with all the many different types of graphs that can be created!

> Recall, there is a `ggplot2` cheat sheet at: https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf

Let's Go!

Wrangling

1. First explore the dataset (called `goog`). How many observations? How many variables? What types of variables (quantitative or categorical)?

2. Find the min, max, mean, and standard deviation for the high and low stock price variables. Convince yourself that you understand each of the numbers. Why is the `sd` lower than the `mean`?

Note that we can use the function `summarize_at` or `summarise_all` !! https://dplyr.tidyverse.org/reference/summarise_all.html

3. Repeat the above analysis, but first `group_by` year. Tell yourself some things you notice about the trends that you see below. For example, there seems to be different trends with respect to the `mean` versus the `sd`. Why?
4. Find the same summaries, but this time, filter for only the fall months (Oct, Nov, Dec) before grouping and the summarizing. Can you tell whether or not your code worked? How would you know? (Hint: see the next part on plotting.)

Plotting

Always really fun to plot data!!

5. Create a boxplot of the stock price `open` broken down by year. What can you see about the trend in the average price of the stock? What about the variability in the price of the stock?
6. Create a line plot with `month` on the x-axis and `open` on the y-axis. Color and group the lines by year. Use `as.factor(year)` so that the years are plotted as distinct and not continuous. After looking at the line graph, go back and compare your numerical summaries from 3 and 4 above. Does it make sense that the numbers in 4 are higher? Why?
7. Create a scatterplot with `open` on the x-axis and `close` on the y-axis. Make the size of the points related to the `volume` of stocks traded that day. Color the points by `as.factor(year)`.
8. (a) Add the line y=x. Use `geom_abline`.

(b) Add another line which is the "best fit" line. Use `geom_smooth` with `method = "lm"` and `se = FALSE`.