Problem Set 1

Insert Name

Stat 108, Week 2

Collaborators

I collaborated with… (list names of collaborators here).

```
# Put all necessary libraries here
# I got you started!
library(tidyverse)
library(viridis)
library(lubridate)
```

Due: Wednesday, February 8th at 5:00pm

Goals of this lab

1. Practice creating and refining graphs with `ggplot2`.
2. Consider the strengths and weaknesses of various `geom`s and `aes`thetics for telling a data story.
3. Create and share a reproducible example with `reprex`.

Notes

- When creating your graphs, consider context (i.e. axis labels, title, annotation)!
- If I provide partially completed code, I will put `eval = FALSE` at the top of the chunk. Make sure to change that to `eval = TRUE` once you have completed the code in the chunk.
- Be prepared to ask for help! We scratched the surface of `ggplot2` in class but we encourage you to really dig in and make your graphs your own (i.e. don't rely on defaults).

Problems

Problem 0

Reminder: **By this Saturday (Feb 4th)**, please fill out this short form so that we can add you to the Stat 108 GitHub organization and so we know how many paper copies of the slides to bring to lecture.

Problem 1

Let's decompose a graph from the wild. Henrik Lindberg and Amber Thomas of The Pudding wrote an article entitled Table for One. Navigate to that article and scroll down to the graph called "Percent of Meals Spent with Companions by Age" and answer the following questions about that graph.

a. Identify the geom(s).

Geoms:

b. Identify the variables.

Variables:

c. Explain how the variables are mapped to the geom(s). When important, address the scale used.

Mapping:

d. What additional context is provided?

e. Reflecting on the data viz considerations we discussed on Day 2, what does this graph do well?

f. Reflecting on the data viz considerations we discussed on Day 2, how could this graph be improved?

Problem 2: Visualizing Locations

We will cover mapping later in the course but right now we can still make simple graphs of spatial data. For Problems 2 - 4, we will use data on the car crashes in Cambridge where a pedestrian or cyclist was hit, covering crashes from 2021 to now. MassDOT provides some information on the variables here (though they've abbreviated some of the variable names in the `.csv` file). Feel free to use the class Slack to ask each other questions about the data.

```
# Read in the data
crash_data <- read_csv("https://raw.githubusercontent.com/harvard-stat108s23/materials/main/psets/data/
glimpse(crash_data)
```

```
## Rows: 1,311
## Columns: 24
## $ crash_numb             <dbl> 4175194, 4181538, 4181543, 4181575, 4181601, 41~
## $ city_town_name         <chr> "CAMBRIDGE", "CAMBRIDGE", "CAMBRIDGE", "CAMBRID~
## $ crash_date             <chr> "01/11/2016", "02/17/2016", "02/17/2016", "02/1~
## $ crash_severity_descr   <chr> "Not Reported", "Non-fatal injury", "Non-fatal ~
## $ crash_status           <chr> "Closed", "Closed", "Closed", "Closed", "Closed~
## $ crash_time_2           <time> 18:42:00, 11:30:00, 10:00:00, 09:00:00, 10:53:~
## $ year                   <dbl> 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016,~
## $ max_injr_svrty_cl      <chr> "Not Applicable", "Non-fatal injury - Non-incap~
## $ numb_vehc              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,~
## $ polc_agncy_type_descr  <chr> "Local police", "Local police", "Local police",~
## $ sptroop                <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ age_drvr_oldest        <chr> "21-24", NA, "25-34", "45-54", "18-20", "55-64"~
## $ crash_hour             <chr> "06:00PM to 06:59PM", "11:00AM to 11:59AM", "10~
## $ first_hrmf_event_descr <chr> "Not reported", "Collision with pedalcycle (bic~
## $ ambnt_light_descr      <chr> "Dusk", "Daylight", "Daylight", "Daylight", "Da~
## $ manr_coll_descr        <chr> "Angle", "Sideswipe, same direction", "Angle", ~
## $ road_surf_cond_descr   <chr> "Dry", "Dry", "Dry", "Dry", "Dry", "Dry", "Dry"~
## $ numb_fatal_injr        <dbl> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ weath_cond_descr       <chr> "Cloudy", "Clear", "Clear", "Clear", "Clear", "~
## $ hit_run_descr          <chr> NA, "No hit and run", NA, NA, NA, NA, NA, NA, N~
## $ most_hrmfl_evt_cl      <chr> "V1:(Collision with pedestrian)", "V1:(Collisio~
## $ speed_limit            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 30, NA, NA, NA,~
## $ lat                    <dbl> 42.37011, 42.37057, 42.39463, 42.36016, 42.3885~
## $ lon                    <dbl> -71.11294, -71.11356, -71.14245, -71.09485, -71~
```

a. Create a scatterplot of longitude and latitude. Make sure to map the variables to the appropriate axes and deal with overplotting in a sensible way.

b. Change the color of the points in your scatterplot to be your favorite color.

c. Create a heatmap of longitude and latitude and pick a color scale that is different from the default color scale.

d. Let's add some context. Recreate your scatterplot or heatmap but this time mark where the Harvard Science Center is located.

e. Using your scatterplot and heatmap, reflect on the distribution of these crashes. Where are the crashes clumping? Which plot gives us a better sense of where the crashes are occurring? (We recommend you look at a map of Cambridge.)

f. Map another variable from the dataset to one of the aesthetics of the points in your scatterplot. What additional information does this provide?

Problem 3

In this problem, we want you to explore the crash data by creating several data visualizations.

a. From the dataset, pick 3 - 4 variables you want to explore. Provide the variable names here; i.e., the name of the variables as they appear in the dataframe.

b. Create 4 graphs. A few things to consider:

   - Each graph does not need to contain all the variables you selected.
   - You can use the same `geom` more than once, but do not use the same `geom` for all four groups.
   - Think carefully about `geoms`, `aesthetics` and `scales`.
   - Feel free to subset or wrangle the dataset if you want to but that isn't a requirement for this problem.
   - Some of the variables have MANY categories. If you don't have prior data wrangling experience, we'd recommend steering clear of these variables for now.

c. Pick your favorite graph from part b) and change at least four things about it by changing (at least) four arguments in `theme()`.

d. Discuss the pros/cons of your 4 graphs.

e. What useful information do your graphs provide to pedestrians and cyclists in Cambridge?

Problem 4

For this problem, we want you to use the following wrangled data to explore how the pandemic may have impacted the monthly number of crashes involving cyclists and pedestrians in Cambridge. And, in particular, we want you to focus on ways to make your graphs more accessible.

```r
# Wrangled data
crash_data <- mutate(crash_data,
                     month = month(mdy(crash_date), label = TRUE)) %>%
  filter(year %in% 2019:2021)
monthly_counts <- count(crash_data, month, year)
```

a. Create a line graph of the monthly counts where month is mapped to the x location, frequency is mapped to the y location and color is mapped to the year.

Hints:

- You might also need to map a variable to `group`.
- If you want `r` to treat a quantitative variable as a categorical variable, you can wrap that variable name in `factor()`.

b. Adjust your line graph so that the colors of your lines are robust to color blindness and that the background color and colors of each line has a color ratio of at least 4.5. You can check the robustness to color blindness by putting your plot through a simulator like that given by the `colorBlindness` package.

c. We say a variable is "double encoded" if it is mapped to two aesthetics of a geom. This is another great way to make your graph more accessible. Recreate the line graph but this time also map year to another aesthetic.

d. Legends require the viewer to connect the scale information in the legend to the different geoms in the graph. A more accessible way of providing this scale information is by directly labeling the geoms (and not including a legend at all). Use `geom_text()` or `geom_text_repel()` to directly label the lines in your graph. Hint: We recommend creating a new dataset first that has the labels and other pertinent information.

e. Providing alt(ernative) text, a written description of the graph, is very helpful for users of screen readers. In this case, when the screen reader gets to a graph, it reads out loud the provided alt text so that the user can still understand the story the graph is trying to convey. Using the advice given in Amy Cesal's article on how to write alt text for data visualizations, write alt text for your plot from part d).

```
# Insert your ggplot
# Add the labs layer
labs(alt = "alt text goes here")
```

Problem 5

They say that "imitation is the sincerest form of flattery". For this problem, we want you to try to recreate a FiveThirtyEight.com graphic. Awesomely, they share their data with the world here.

Notes:

- You don't need to recreate all their branding/background color scheme.
- Some of their graphs require an extensive amount of data wrangling. For this problem, we'd recommend you steer toward a data visualization that can be made with little to no data wrangling.

a. Take a screenshot of the graph you are going to recreate. Upload the screenshot to the same folder where you have saved your p-set, and insert the file name below. Then change the `eval = FALSE` to `eval = TRUE`.

b. Load the data and recreate the graph as best as you can. The teaching team can help if you are having trouble loading the data!

c. Now make the graph better somehow.

d. Justify why your rendition of this FiveThirtyEight.com graph is more effective at telling the data story than the original.

Problem 6

We've only scratched the surface of what you can do with `ggplot2`. Create a `reprex` (reproducible example) of a `ggplot2` feature that we didn't cover in class. (This could be a new `geom`, a `theme()` option, a different `scale` layer, etc.)

a. Put the code of your `reprex` in the chunk below (and leave `eval = FALSE` so the chunk is not evaluated when you knit the document.)

b. Go to the `#q-and-a` channel of the Stat 108 Slack workspace and reply to the **P-Set 1 Reprex** thread (not in the main channel). Provide an example of the new feature you explored and a brief explanation of this feature. (Also include your explanation below for the graders.)