Individual homework 2

Roles for this assignment:

Please push your assignment to GitHub by 10:00pm (Central) Sunday, Apr 7.

You are currently in the GitHub repository (repo) for `hw2-username`. The assignment prompt is shown below (i.e. in `README.Rmd`). You can view this online in your homework 2 GitHub repository as a Markdown file(`README.md`) or a pdf.

Please **use `hw2.Rmd` to complete this assignment**. Be sure to **knit your file to PDF before your final push to GitHub**.

Homework process

For help on the homework process, review

- Assignments in Stat 220 for content/formatting questions.
- GitHub Guide for Students in Stat 220 for Git and Github instructions.

When you are done with your homework, **don't forget to push your changes to ALL files back to GitHub!** This means you should commit and push all related files, not just your final PDF. Additionally, ensure you post the link to your **GitHub repository to Gradescope for the final submission and grading**. This step is crucial as it allows for a comprehensive review of both your code and the rendered output.

Assignment prompt

Problem 1: Spot the error

Explain why the following command does not color the data points blue, then write down the command that will turn the points blue.

```
library(ggplot2)
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = "blue"))
```

Problem 2: Penguins

Load the data `penguins` using the command

```
# install.packages("palmerpenguins")  # run once then DELETE FROM RMD
data(penguins, package = 'palmerpenguins')
```

Look at help file for for variable info: `?palmerpenguins::penguins`.

a.

Create a scatterplot of `body_mass_g` (y) against `bill_length_mm` (x) and separate the plot into facets by `species`.

b.

Add regression lines to the plot in (a) for each species with no standard error shading. See `?geom_smooth` to see how to add a regression (`lm`) line instead of a wigglely smoother.

c.

Instead of faceting by `species`, use `species` to color the points and regression lines. Use a color scheme for the points that is not the default coloring.

d.

Which visual, (b) or (c), makes it easiest to compare the **slopes** of the lines for each species? Explain.

Problem 3: Storm paths by year

Install the package `nasaweather`, if needed, and load the `storms` data.

```
# install.packages("nasaweather")
data(storms, package = "nasaweather")
```

a.

Use `geom_path()` to plot the path of each tropical storm in this data set where you

- use color to distinguish the storms from one another
- use faceting to plot each `year` on its own panel
- change your coordinate system to a map using `coord_map()`.

Hint: To use color to distinguish storms but not include the legend of storm names/colors you can add `scale_color_discrete(guide="none")`

b.

Add a map of the US, Mexico, Canada and the UK to your map in (a) so you can get a better idea where these storms are traveling, truncating the longitude as suggested in the hints.

Hints:

- Look at the help file for `?map_data`. To create map data with only the counties around the storm paths, use data from the the `world` map from `map_data` with the following `region`s specified: `usa`, `mexico`, `canada` and `uk`.
- Add the map data for the regions of interest using `geom_polygon`. (Hint: make sure your paths map doesn't have `color` as a global `aes`.)
- Modify your `coord_map` from part (a) to add the arguments `xlim` and `ylim` to set the min and max limits of your graph to the min/max latitude and longitude of the `storms`.

Problem 4: explain command (no R needed)

Consider the data set shown in the table below. Each row represents a class and `classType` of `S` denotes a statistics class and `C` denotes a CS class. The variable `m` counts the number of mac users in the class and `w` counts the number of windows users.

classType	m	w
C	10	4
C	3	1
C	7	3
S	2	7
S	7	10

What data set will be produced by the following commands? Describe (**in words**) how the original data set is being modified and show what it looks like using an R Markdown table (like the one used above) to display the new data set. Assume the original data set is named `mydata`. (No credit will be given creating the fake data frame and just running the code chunks.)

a.

```
mydata %>%
  filter(classType == "C") %>%
  select(m, w)
```

b.

```
mydata %>%
  mutate(ratioW = w/sum(w))
```

c.

```
mydata %>%
  group_by(classType) %>%
  mutate(ratioW = w/sum(w))
```

d.

```
mydata %>%
  group_by(classType) %>%
  summarize(Y = sum(w+m))
```

e.

```
mydata %>%
  group_by(classType) %>%
  mutate(X = w+m, Y = sum(w+m))
```
