Homework 2

Important notes:

Submit via Canvas (Assignments tab).

Turn in a PDF writeup with a GitHub link in it. Your PDF writeup should contain your results/solutions. At the very top of this PDF, please include your name, your UT EID, and **a link to your GitHub repo containing your R code for this assignment.** Submissions without links to code on GitHub will be penalized 30% (i.e. your maximum possible score will be a 70).

Submit early and often. If you are trying to upload your solution for the first time 10 seconds before the deadline and your wifi breaks, we will be sympathetic to your situation, but we will not relax course policies regarding deadlines. This is to prevent obvious forms of abuse and to ensure a level playing field for all students. You can also submit multiple times before the deadline! If you intend to be working on a problem right up until the deadline, we suggest submitting an early draft of your homework to "lock in credit" for the problems you've already solved, and then replacing that submission later on Canvas, once you've made your finishing touches. Canvas will save your last submission, over-writing any previous submissions.

Do not include your raw R code in your PDF write-up unless we explicitly ask for it. Remember, your R script should be included as part of the GitHub repo that you link to, rather than pasted into the PDF writeup. If you use RMarkdown to create your writeup (recommended), learn about `echo=FALSE` to suppress printing R code for code chunks.

Problem 1: Beauty, or not, in the classroom

The University of Texas at Austin, like every major university in the country, asks students to evaluate their courses and professors. The `profs.csv` file contains data on course-instructor evaluation surveys from a sample of 463 UT Austin courses. These data represent evaluations from 25,547 students and most major academic departments. The data frame also includes information on characteristics of each course and facts about the professors such as seniority and demographics. Also included is a rating of each instructor's physical attractiveness, as judged by a panel of six students (3 males, 3 females) who were shown photos of the instructors. Key variables in the `profs.csv` data frame are:

- `eval`: the instructor's average teaching evaluation score, on a scale of 1 to 5, for courses in the sample
- `beauty`: the six panelists' average rating of the professor's physical attractiveness, shifted to have a mean of zero. For example, 2 is two points above average and -1 is one point below average.
- `minority`: is the professor from a non-white racial or ethnic minority?
- `age`: the professor's age in years
- `gender`: indicator of the professor's gender
- `credits`: indicator of whether the course is a single-credit elective ("single") or an academic course ("more")
- `division`: indicator of whether the course is a lower or upper division course
- `native`: indicator of whether the professor is a native English speaker
- `tenure`: indicator of whether the professor has tenure/is on the tenure track, or not
- `students`: the number of students who participated in the course evaluation survey
- `allstudents`: the number of students enrolled in the course
- `prof`: unique identifier variable for the professor

Use these data to address the following questions by creating plots and/or calculating summary statistics.

Format the plot professionally with clear labeling and consideration of best practices for effective plots.

Include in your write-up an image of each plot along with an informative caption below each plot. The caption may be typed in your write-up below the plot and does not have to be generated using ggplot's caption feature. The caption should consist of 1-2 sentences describing key features of the plot (if these are not already clear from the chart title and labels) and a short summary of key takeaways from your plot in its relevant context. Think of this caption as a walkthrough for your plot audience.

Part A. Create a histogram to display the overall data distribution of course evaluation scores.

Part B. Use side-by-side boxplots to show the distribution of course evaluation scores by whether or not the professor is a native English speaker.

Part C. Use a faceted histogram with two rows to compare the distribution of course evaluation scores for male and female instructors.

Part D. Create a scatterplot to visualize the extent to which there may be an association between the professor's physical attractiveness (x) and their course evaluations (y).

Problem 2: bike sharing

Bike-sharing systems are a new generation of traditional bike rentals where the whole process from rental to return is automatic. There are thousands of municipal bike-sharing systems around the world (e.g. Citi bikes in NYC or "Boris bikes" in London), and they have attracted a great deal of interest because of their important role in traffic, environmental, and health issues—especially in the wake of the Covid-19 pandemic, when ridership levels on public-transit systems have plummeted.

These bike-sharing systems also generate a tremendous amount of data, with time of travel, departure, and arrival position recorded for every trip. This feature turns bike sharing system into a virtual sensor network that can be used for sensing mobility patterns across a city.

Bike-sharing rental demand is highly correlated to environmental and seasonal variables like weather conditions, day of week, time of year, hour of the day, and so on. In this problem, you'll look at some of these demand-driving factors using the `bikeshare.csv` data from the course Canvas page. This data set contains a two-year historical log (2011 and 2012) from the Capital Bikeshare system in Washington D.C. The raw data is publicly available at http://capitalbikeshare.com/system-data. These data have been aggregated on an hourly and daily basis and then merged with weather and seasonal data.

The variables in this data set are as follows:

- instant: unique record identifier for each row
- dteday: date
- season: season (1:spring, 2:summer, 3:fall, 4:winter)
- yr: year (0: 2011, 1:2012)
- mnth: month (1 to 12)
- hr: hour (0 to 23)
- holiday: whether the day is holiday or not
- weekday: day of the week (1 = Sunday)
- workingday: if day is neither weekend nor holiday is 1, otherwise is 0.
- weathersit: a weather situation code with the following values
  - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
  - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
  - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
- temp: Normalized temperature in Celsius. The actual values are divided by 41 (max)
- total: count of total bike rentals that hour, including both casual and registered users

Your task in this problem is to prepare three figures. To make these figures, you will need to combine the ideas from our lesson on Plots with our lesson on Data wrangling. In other words, you won't be able to make these plots by calling `ggplot` on the raw data we've provided. First, you'll need to use some of our six key data verbs from the Data Wrangling review lesson to get the data into an appropriate form. Only then will you actually be able to create these plots.

- Plot A: a line graph showing average hourly bike rentals (`total`) across all hours of the day (`hr`).

- Plot B: a faceted line graph showing average bike rentals by hour of the day, faceted according to whether it is a working day (`workingday`).

- Plot C: a faceted bar plot showing average ridership (y) **during the 9 AM hour** by weather situation code (`weathersit`, x), faceted according to whether it is a working day or not. (Remember that you can focus on a specific subset of rows of a data set using `filter`.)

Your write-up should include each plot, together with an informative caption (i.e., written paragraph) below each plot. Think of this caption paragraph as a walkthrough for your plot audience – perhaps what you would say in a live presentation to incorporate the plot into your narrative. Your caption should clearly explain the plot itself (e.g., what the axes represent and what the panels show). Don't forget to specify variable units. The caption should also contain a one-sentence *take-home lesson* of what we have learned about ridership patterns from the plot.

Problem 3 - Capital Metro UT Ridership

The file `capmetro_UT.csv` contains data from Austin's own Capital Metro bus network, including shuttles to, from, and around the UT campus. These data track ridership on buses in the UT area. Ridership is measured by an optical scanner that counts how many people embark and alight the bus at each stop.

Each row in the data set corresponds to a 15-minute period between the hours of 6 AM and 10 PM, each and every day, from September through November 2018. The variables are:

- *timestamp*: the beginning of the 15-minute window for that row of data
- *boarding*: how many people got on board any Capital Metro bus on the UT campus in the specific 15 minute window
- *alighting*: how many people got off ("alit") any Capital Metro bus on the UT campus in the specific 15 minute window
- *day_of_week* and *weekend*: Monday, Tuesday, etc, as well as an indicator for whether it's a weekend.
- *temperature*: temperature at that time in degrees F
- *hour_of_day*: on 24-hour time, so 6 for 6 AM, 13 for 1 PM, 14 for 2 PM, etc.
- *month*: July through December

Your task in this problem is **to make two faceted plots** and to answer questions about them.

1. One faceted line graph that plots **average boardings** by hour of the day, day of week, and month. You should facet by day of week. Each facet should include three lines of average boardings (y) by hour of the day (x), one line for each month and distinguished by color. Give the figure an informative caption in which you explain what is shown in the figure and also address the following questions, citing evidence from the figure. Does the hour of peak boardings change from day to day, or is it broadly similar across days? Why do you think average boardings on Mondays in September look lower, compared to other days and months? Similarly, why do you think average boardings on Weds/Thurs/Fri in November look lower? (Hint: wrangle first, then plot.)

2. One faceted scatter plot showing boardings (y) vs. temperature (x), faceted by hour of the day, and with points colored in according to whether it is a weekday or weekend. Give the figure an informative caption in which you explain what is shown in the figure and also answer the following question, citing evidence from the figure. When we hold hour of day and weekend status constant, does temperature seem to have a noticeable effect on the number of UT students riding the bus?

These are exactly the kind of figures that Capital Metro planners might use to understand seasonal and intra-week variation in demand for UT bus service. These are also the kind of figures one would create in the process of building a model to predict ridership.

Notes:

First, a feature of R is that it orders categorical variables alphabetically by default. This doesn't make sense for something like the day of the week or the month of the year. To reorder the days of the week and months

in appropriate order, paste the following block of code into your R script at the top and execute it before you start further work on your plots for this problem:

```
# Recode the categorical variables in sensible, rather than alphabetical, order
capmetro_UT = mutate(capmetro_UT,
                day_of_week = factor(day_of_week,
                  levels=c("Mon", "Tue", "Wed","Thu", "Fri", "Sat", "Sun")),
                month = factor(month,
                  levels=c("Sep", "Oct","Nov")))
```

Second, please keep each figure + caption to a single page combined (i.e. two pages, one page for first figure + caption, a second page for second figure + caption).

Problem 4: Wrangling the Billboard Top 100

Return to the data in `billboard.csv` that we briefly analyzed in class, and that contains data on every song to appear on the weekly Billboard Top 100 chart since 1958. Each row of this data corresponds to a single song in a single week. For our purposes, the relevant columns here are:

- performer: who performed the song
- song: the title of the song
- year: year (1958 to 2021)
- week: chart week of that year (1, 2, etc)
- week_position: what position that song occupied that week on the Billboard top 100 chart.

Use your skills in data wrangling and plotting to answer the following three questions.

Part A: Make a table of the top 10 most popular songs since 1958, as measured by the *total number of weeks that a song spent on the Billboard Top 100.* Note that these data end in week 22 of 2021, so the most popular songs of 2021 onwards will not have up-to-the-minute data; please send our apologies to The Weeknd.

Your table should have **10 rows** and **3 columns**: `performer`, `song`, and `count`, where `count` represents the number of weeks that song appeared in the Billboard Top 100. Make sure the entries are sorted in descending order of the `count` variable, so that the more popular songs appear at the top of the table. Give your table a short caption describing what is shown in the table.

(*Note*: you'll want to use both `performer` and `song` in any `group_by` operations, to account for the fact that multiple unique songs can share the same title.)

Part B: Is the "musical diversity" of the Billboard Top 100 changing over time? Let's find out. We'll measure the musical diversity of given year as *the number of unique songs that appeared in the Billboard Top 100 that year.* Make a line graph that plots this measure of musical diversity over the years. The x axis should show the year, while the y axis should show the number of unique songs appearing at any position on the Billboard Top 100 chart in any week that year. For this part, please filter the data set so that it excludes the years 1958 and 2021, since we do not have complete data on either of those years. Give the figure an informative caption in which you explain what is shown in the figure and comment on any interesting trends you see.

There are number of ways to accomplish the data wrangling here. We offer you two hints on two possibilities:

1) You could use two distinct sets of data-wrangling steps. The first set of steps would get you a table that counts the number of times that a given song appears on the Top 100 in a given year. The second set of steps operate on the result of the first set of steps; it would count the number of unique songs that appeared on the Top 100 in each year, *irrespective of how many times* it had appeared.
2) You could use a single set of data-wrangling steps that combines the `length` and `unique` commands.

Part C: Let's define a "ten-week hit" as a single song that appeared on the Billboard Top 100 for at least ten weeks. There are 19 artists in U.S. musical history since 1958 who have had *at least 30 songs* that were "ten-week hits." Make a bar plot for these 19 artists, showing how many ten-week hits each one had in their musical career. Give the plot an informative caption in which you explain what is shown.

Notes:

1) You might find this easier to accomplish in two distinct sets of data wrangling steps.
2) Make sure that the individuals names of the artists are readable in your plot, and that they're not all jumbled together. If you find that your plot isn't readable with vertical bars, you can add a `coord_flip()` layer to your plot to make the bars (and labels) run horizontally instead.
3) By default a bar plot will order the artists in alphabetical order. This is acceptable to turn in. But if you'd like to order them according to some other variable, you can use the `fct_reorder` function, described in this blog post. This is optional.