CptS 475/575: Data Science, Fall 2020

Assignment 4: Joins and Visualization

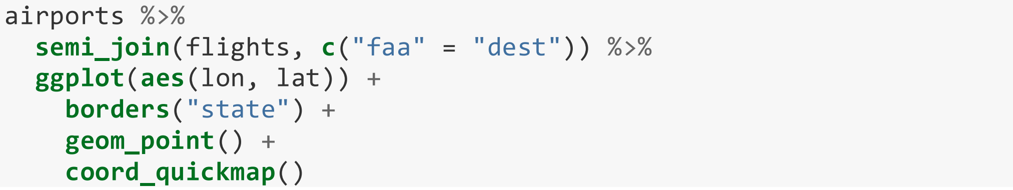
Release Date: September 30, 2020 **Due Date:** October 7, 2020 (11:59 pm)

General instruction: This assignment has **three problems.** The first problem is on Joins (relational data from the data wrangling series of topics), and the second and third problems are on visualization.

Your solution will be submitted as a **PDF file**. You are encouraged to use R Markdown or a similar tool (like Jupyter) to prepare your file.

Problem 1 (50 pts). This problem will involve the nycflights13 dataset (including tables airlines, airports, planes and weather), which we saw in class. It is available in both R and Python, however R is recommended for at least the visualization portion of the question. Start by installing and importing the dataset to your chosen platform. We will first use joins to search and manipulate the dataset, then we will produce a flightpath visualization.

a. (10/9 pts) Filter the dataset (using a **left join**) to display the tail number, year, month, day, hour, origin, and humidity for all flights heading to Tampa International Airport (TPA) on the afternoon of November 1, 2013.
b. (10/9 pts) What is the difference between the following two joins?
```
anti_join(flights, airports, by = c("dest" = "faa"))
anti_join(airports, flights, by = c("faa" = "dest"))
```
c. (10/9 pts) Select the origin and destination airports and their latitude and longitude for all fights in the dataset (using one or more **inner joins**). *Hint: There should be 329,174 flights if you've done this correctly.*
d. (10/9 pts) Use groupby and count to get the number of flights to each unique origin/destination combination *Hint: There should be 217 of these total*.
e. (10/9 pts) Produce a map that colors each destination airport by the average air time of its incoming flights. Here is a code snippet to draw a map of all flight destinations, which you can use as a starting point. You may need to install the maps packages if you have not already. Adjust the title, axis labels and aesthetics to make this visualization as clear as possible. *Hint: You may find it useful to use a different type of join in your solution than the one in the snippet.*



```
airports %>%
  semi_join(flights, c("faa" = "dest")) %>%
  ggplot(aes(lon, lat)) +
    borders("state") +
    geom_point() +
    coord_quickmap()
```

Problem 2 (30 pts). You may recall on the lecture on Friday, Sep 25 (when we had Dr. Ofer Amram as a guest speaker), the warm-up question that day was to type in the city and state (or city and country) where you grew up in. The result of that warm-up question is summarized in a file that is posted under Lecture/Sep 25. The task you have for this problem is to visualize that list on a world map, indicating in some way the cities. You could use the same mark to denote the cities. Try to make your visualization as nice looking as possible.

You are free to choose any mapping tool you wish to produce this visualization. The first thing you need to do is to get the coordinates for each place. Research how this can be done and use what you find. The dataplusscience.com website I mentioned in the lecture on Monday September 28 (in the slide that has the website analytics dashboard) has some blogs about mapping that you may find useful. After you have coordinates you can use different methods for mapping. The simplest is probably through https://batchgeo.com/features/map-coordinates/ However, you can also use d3 to map the locations, if you want to learn something that you could use for other projects later.

Problem 3 (20 pts). Create a word cloud for an interesting (relatively short, say a couple pages) document of your own choice. Examples of suitable documents include: summary of a recent project you are working or have worked on; your own recent Statement of Purpose or Research Statement or some other similar document.

You can create the word clouds in R using the package called *wordcloud* or you can use another tool outside of R such as *Wordle*. If you do this in R, you will first need to install *wordcloud* (using install.packages("wordcloud")) and then load it (using library(wordcloud)). Then look up the documentation for the function called *wordcloud* in the package with the same name to create your cloud. Note that this function takes many arguments, but you would be mostly fine with the default settings. Only providing the text of your words may suffice for a minimalist purpose. You are welcome (and encouraged) to take the generated word cloud and manipulate it using another software to enhance its aesthetic. If you have used Wordle instead of R, Wordle gives you functionalities to play with the look of the word cloud you get. Experiment till you get something you like most.

Your submission for this would include the figure (cloud) and a brief caption that describes the text for the cloud. For example, it could be something like ``Jenneth Joe's Essay on Life During Pandemic, written in October 2020."