Wrangle Report

Gathering Data

For the first two files, the .csv and the .tsv, I followed the instructions that were listed and downloaded the first one while programmatically downloaded the .tsv file.

For the final dataset, I managed to create a Twitter development account on there and got the required keys to download the .json file from the Twitter API. I used the code that was provided, *twitter_api.py*, to save the data as *tweet_json.txt*. As I have little knowledge of extracting data from .json files, I looked up the Knowledge forum and it described how Myles C was helping another student create the dataframe from the .json file. I read that code and applied it to my own changing the variable names, but understood how that code work to read the .json file line-by-line and save it to a dataframe, which I named *df_twitter*

Assessing Data

The first thing I did to assess the data was to do a visual assessment of the *df_csv* dataframe. The most noticeable thing I saw was a lot of null values and columns I didn't think were relevant. Using the .info() method on the *df_csv* dataframe. I decided I wanted to remove these columns; so I put this under one of the tidiness issues as it changes the structure of the dataframe and will eliminate a lot of the null values. The second thing I noticed was the structure of the categorical columns, doggo, pupper, puppo, and floofer. To acquaint myself to what these terms meant, I looked at the definitions that were provided by the website. I felt that doggo, pupper, and puppo were terms used to describe the age of the dog while floofer described if the dog was fluffy or not. I decided I wanted to combine the doggo, pupper, and puppo columns while using a simple yes or no to describe if the dog is a floofer or not. The next thing I noticed in this dataframe was the values in the source column were in html tags. I felt I wanted to make it more readable. I did a *value_counts* on that column and it showed only four types of sources. I figured it was worth cleaning. In addition, I wanted to change the name to one of the columns to *rating_out_of_10* as I felt the *rating_demoninator* column is unnecessary. The rest of the items in the df_csv dataframe are datatypes I wanted to convert to a more appropriate datatype.

The df_tsv dataframe had a few quality issues I wanted to deal with mainly in the p1, p2, and p3 columns by capitalizing the names in those columns as well as changing the underscores to spaces. The only datatype conversion I wanted to do in this dataframe

was to convert the tweet_id column to string format so I can eventually merge all three dataframes.

The df_twitter dataframe looked clean as it is.

Cleaning Data

I decided to tackle the tidiness issues first as that is related to the structural part of the df_csv dataframe. I made a new classification column that copied the puppo column. I then did a for loop that checked each value of the doggo, pupper, and puppo column and assigned that respective value to the classification column. If none of those three values were present in a particular row, then 'unknown' was assigned.

The second tidiness issue I tackled was eliminating the unnecessary columns. I just had the clean version of the df_csv equal the relevant columns that I wanted.

I then tackled the quality issues that were described above in both the df_csv and the df_tsv dataframes and then merged all three dataframes together as one. Finally, I saved the cleaned dataframe as *twitter_archive_master.csv*.