Data Wrangling Report

In this project I was able to put to use what I have learned in the data wrangling section of the course. The data wrangled consists of the tweets of Twitter user @dog_rates, known as WeRateDogs which rates people's dogs on a scale from 1 to 10. While completing the project I was able to retrieve various sets of data from the twitter archive and clean and analyze the data collected. This report will briefly describe my wrangling process.

Project Tasks

The tasks to complete the project are as follows:

- Gathering Data
- Assessing Data
- Cleaning Data
- Gathering insights and visualizing Data

Gathering Data

The completion of this project required the use of three different data tables:

- **Twitter_archive_enhanced.csv**: this file was provided by Udacity and downloaded manualy.
- **Image-predictions.tsv**: This table is hosted on the Udacity website and was downloaded programmatically using the url information and request library. This data gives the dog breed present in each tweet.
- **Tweet_json.txt**: I used the Tweepy library to query the Twitter api using the tweet id's in the twitter archive for the json data of each tweet and saved each one to a .txt file. Then I created a dataframe with the data that was queried.

Assessing The Data

With the three tables on hand I then assessed the data the following way:

- First, I imported the tables to a jupyter notebook, then I created a dataframe for each table and printed them so that I could visually inspect the contents of the dataframe.

- Then to get a more in-depth look at the database I used commands like value_counts, duplicated, groupby and sample. By doing this I was able to find some issues with the data regarding the quality and tidiness of the information collected.

Cleaning the data

After assessing the data for any quality and tidiness issues I preceded to clean the data. There were several quality issues with the data sets such as erroneous data types, different denominators in the rating system, missing dog names, duplicates, etc.

First, I created copies of the three dataframes so that I could work from them without changing the original in case if I make a mistake. After that I dropped the duplicated rows and removed columns that would not be useful. Subsequently I then Melt the doggo, floofer, pupper and puppo columns to dogs and dogs_stage column together and then I dropped the dog's column. The next step was to change the date and time format, to do this I used the to_datetime function to accomplish this. Next in the image_pre dataframe I used the drop function to drop the 66-duplicate image URL's. and in the Tweet_json removed any tweet that was not an original tweet such as retweets. After cleaning the datasets I combined them into one table.