Wrangling Report

I began my data wrangling project by first gathering the 3 needed datasources: twitter-archive-enhanced.csv, which I was able to download directly, image_predictions.tsv, which I downloaded programmatically, and tweet_json.txt, which I created by using Twitter's Tweepy API, by reading in each line to a .txt document. In performing the gathering step of the data wrangling process, I was able to practice what I had learned about gathering from a variety of sources in the gathering section, and I also got my first insights into how APIs work, and I had never worked with one in the past.

After loading each of the files into my Jupyter Notebook, I saved them each to a separate dataframe, and was then able to move forward with the assess portion of my wrangling. I began with a visual assessment of the 3 dataframes, and got to know what columns were there, and what information was present in each. I also was able to quickly spot some issues with the data straight away: lots of NaNs in several columns, values in the name column that did not appear to be proper names, 4 columns for dog stage, with lots of None values inside, and so on. I jotted these issues down and moved on to perform a programmatic assessment.

Here I saw that many data types were incorrect, that there were different numbers of rows and columns in each dataframe, that there were lots of columns with null values, and so on. I also jotted these issues down, and then assessed to determine which were messy and which were dirty data issues. After assessing these issues, I defined them, and then jumped into actually cleaning the data.

The messy data issues I worked on, were breaking down the doggo, puppo, pupper, floofer columns into 1 dog_stage column, and merging the 3 dataframes together in order to have the same tweet ids/amount of rows, and to have everything in one place for analysis.

The dirty data issues I worked on (which I wove in amongst the messy data issues as needed), were removing rows that were retweets in order to only assess original tweets, renaming the tweet_id column so that this column name matched each dataframe in order to merge them, changing tweet_id from integer to string type, dropping columns that were not necessary for the analysis, getting rid of words that were not names in the dog name column, correcting datatypes, replacing ratings in the denominator/numerator columns to ensure that all had denominators of 10, and removing the tweet url from the tweet text, in order to avoid having duplicate information.

Cleaning the data made my analysis substantially easier and quicker to complete, and I was able to create graphs with confidence, knowing that my data, while perhaps not perfectly cleaned (I did only fix 10 issues after all!) was pretty well cleaned up and organized, and that I was not working with extremely problematic, dirty data which could skew my analysis.