Report on Data Wrangling Efforts for the WeRateDogs Twitter Account:

When I began working on this project it was clear I would need to gather data from multiple sources. This data would need to be cleaned significantly as well. The first effort was to ingest data from a CSV that was provided to me. This was relatively simple and only required that I used the pd.read_csv from pandas, and assign the resulting data to a data frame variable. After confirming the data had been correctly added to the frame, I next moved on to the image prediction file. This data is stored in a TSV at a known URL. I would first need to import the requests library, once this is done, I can use requests.get to submit a request for the needed file. After this I created a new file called 'image_predictions.tsv' I then wrote the data from the request to the file, and lastly, I assigned the data in the file to a data frame. The next place I had to ingest data from was Twitter directly. Using a developer account I authenticate with Twitter using tweepy I then request all tweets that have a tweet_ID that matches one from the previously imported CSV. All these tweets are then stored as JSON in a text file. This file is then loaded into a list, which is then converted into a data frame.

With all three sources loaded into data frames, we can start to get a handle on what were working with. After going through and assessing the data 7 actions are needed to take to properly clean the data. First, several names in the 'name' column are not actually names. The easiest way to clear this out is to set all the names that aren't capitalized to null, as well as any name that says 'None'. Next, there are 78 reply tweets, because reply tweets are likely not to fit the standard, we rate dogs format these will be dropped. The same process applies to retweets. After that, I have to address the doggo, floofer, pupper, and puppo columns. Just like with the 'name' column, the data uses 'None" and not null, this makes the data look like there is no null values, to fix this all 'None" record will be set to null. Next is the missing expanded_urls, since this data is not retrievable, and it only affects 3 rows these rows will be dropped. After that I noticed that the timestamp column has the wrong data type. To fix this I simply corrected the data type simply using pd.to_datetime function. The last two issues to fix are related to the same problem and have the same solution. After deleting all of the retweets and reply tweets the rows in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, and retweeted_status_timestamp are all null, additionally, they are all also unneeded for our analysis. Because of this these columns are dropped.

To tidy the data up I took two more steps. First, I used pd.merge to combine all of the data into one single data frame, matching on tweet_id. After that, I wanted the doggo, floofer, pupper, and puppo columns in just one column. So, I concatenated the data together into the type column. This created the combination of doggo-floofer, doggo-pupper, and doggo-puppo. With the dash being added for readability. After this the data, while not perfect, was ready to examine.