WeRateDogs Data Wrangling

For this Data Wrangling project, I gathered data from the three different sources below.

-Source #1: This data was provided by Udacity in a file called 'twitter-archive-enhanced.csv'. I downloaded the file and used the pandas read_csv function to read the data into the 'twitter_archive' dataframe.

-Source #2: This data was This data was provided by Udacity in a file called 'image_predictions-3.tsv'. I downloaded the file and used the library requests function and used pandas read_csv function to read the data into the 'image_predicitons' dataframe

-Source 3#: This data is from Twitter and retrieved using Twitter's API (tweepy) and gathered data about retweet counts and favorite counts and stored it in the tweet_data dataframe.

Next I assessed the data both visual and programmatically. I then listed eight quality and two tidiness issues that I discovered while assessing the data. I took the below steps in order the clean the data and fix the issues I noted in my assessment.

Quality:

1) I corrected the datatype in the 'timestamp' column to datetime and the 'tweet_id' column to string in the twitter_archive dataframe
2) I removed columns that did not provide relevant information about our data. I dropped the 'in_reply_to_status_id' and the 'in_reply_to_user_id' columns
3) I noticed errors in the names such as 'None' and 'a'. I then created a list of all the lower case names (which were incorrect) and replaced them with NaN values. I also replaced all 'None' with NaN values
4) I removed retweets by dropping entries that had a retweed_status_id
5) I corrected the rating_denominator to 10. Then I fixed the incorrect numerators by changing the datatype to float instead of integer, looking through the tweet's text to find the correct numerator. In some instances the rating was a multiplied due to there being more than 1 dog in the image. To standardize the ratings I divided the multiplied rating by its respective factor. And changed the rating_numerator to the new rating. I then dropped some outliers in the data that were given intentionally high ratings.
6) I replaced the 'none' values found in the dog_stage data with NaN values.
7) I corrected the datatype in the 'timestamp' column to datetime and the 'tweet_id' column to string image_predictions dataframe
8) I corrected the datatype in the 'timestamp' column to datetime and the 'tweet_id' column to string tweet_data dataframe

Tidy:

1) For tidier data, I decided to merge the doggo, pupper, puppo, and floofer columns into a new 'dog_stage' column. In doing this step, I created a new variable "multiple" for the entries that had multiple dog stages assigned. I then dropped the original doggo, pupperm puppo, and floofer columns.
2) Lastly, I merged all the tables together into one dataframe (joined on tweet_id), so that we can better perform our analysis and understand the data.