@WeRateDogs data wrangling project.

We started gathering all necessary information from three different sources.

- 'twitter-archive-enhanced.csv' stored as 'df1'
- 'image-predictions.tsv' stored as 'df2'
- 'tweet-json.txt' stored as 'df3'

We first took a look to some summary information about each dataset. Then some data wrangling is done in 'df1', this is going to be our future master data set, while 'df2' and 'df3' will be joined as additional information.

The first action is to assess and clean 'df1' creating 'df1mod' (keeping a copy of the original dataframe in df1). We drop all the retweeted status as asked by the project rubric. Then we proceed to examine 'rating_numerator' column. Lot's of values are represented in the column. After some google searching, we managed to find the range of scoring of WeRateDog, is from 0 to 20. So using .query method all values outside from that range are deleted.

Looking at the denominators, we found a few rows incompatible with the common 10 denominators, so we deleted them. Notice that instead of fixing the numbers we're opting to delete them. Our focus is to have a high quality dataset at a cost of having less observations in our final dataset. After that we examine the 'name' column and found almost 660 rows with 'None' as a name, we decide to dump all those rows too. From initial 2175 observations we now have 1486.

Then we continued to examine the dog's stage columns. These columns have very few rows with actual observations, while the majority have 'None' in each 3 columns. This data is changed to Boolean values (instead of strings) for tidiness. We proceed to extract 'favorite_count' and 'retweet_count' from 'df3' along with the 'full_text' column which is renamed to 'text' and then merged with 'df1mod' based on this column, giving that values are the same in both datasets.

Finally, image prediction dataset stored as 'df2' is merged based on shared 'tweet_id' column in both datasets. One final tidiness action is to modify the 'timestamp' column from string dtype to datetime64[ns] using pandas .to_datetime method.