For this project the first step in data wrangling was gathering the data from three different sources. First, I loaded the file we were given (twitter_archive_enhanced.csv) into a dataframe. After that I did an initial assessment using functions like .info and .shape. Next I programatically downloaded the second file (image_predictions.tsv) using a provided URL and the Requests library. I saved this data as “df2”, using read_csv to convert the file format. The last step in gathering data was using Twitter’s API, Tweeter, to retrieve retweet and favorite counts. To use Twitter’s API, you need to apply for permission. When granted, you can create a consumer key, consumer secret, access token, and access token secret. These allow the user to access to the database.

After gathering the data, it is time to organize it. First I created two lists, one to write our data to and one to write the errors from our data into. Next I created a loop that queries Tweepy and stores the favorite count and retweet count for each tweet id. Tweepy does have a rate limit for the number of times you can query the database in 15 minute time frames. I changed the settings for wait_on_rate_limit to keep the code from throwing an error every time it hits the maximum. After that I created a second error list to add another buffer before I started working directly on the files.

Next I converted the main list (“df_list”) to a dataframe (“df3”). I merged df3 and df2, (the image predictions file) and labeled it df_master. I merged df_master with the original dataframe, creating my full data set (“main2”). This fixed quality issue #2, the data was spread out.

Now it was time to view and assess my data. I used exploratory functions like .describe and .value_counts(). My first quality error was that the tweet_id column should be a string, not an int64. I got rid of duplicated IDs as well as retweets or replies. I changed ‘none’ in the name column to NaN. Next I queried the data for the abnormal numerators to retrieve their indices and then I dropped those rows. For the denominators, I filtered out ones other than 10 and used the indices to drop them. For my last quality errors I removed “the”, “a”, and “an” from the name column entries and edited the appearance of the breed prediction column.

The first tidiness issue is the four columns for dog stage (‘doggo’, ‘pupper’, etc.). I created a new column, ‘stage’ and created a loop that wrote in their stage if one is given. Then I dropped columns that cluttered my chart and were not being used. Finally, I saved the dataframe into a csv file (twitter_archive_master.csv).