For this data wrangling effort I used a Twitter archive from the “WeRateDogs” twitter account. This is an account known for rating dogs above 10/10 and for assigning a the dog’s type to a cute category. For example, the four named categories present in this data set are “puppo”, “doggo”, “fluffer”, and “pupper”. There is also the uncategorised dogs that are in the “None” category. I also used file that contained predictions for what dog breed the dog was, based on the picture of the dog in the tweet.

I started out this project with gathering the three files needed for this data set. The first file was the twitter archive that I retrieved via the requests library. The second file could be gathered by applying for an api key from twitter (in my case I was unable to obtain an api key from twitter so I used the Udacity provided alternative), then using the api to retrieve a json file with detailed tweet data and then using python to read the data into a DataFrame line by line. The third file was provided already in the udacity project space.

I moved on to doing a brief inspection of the three DataFrames and it immediately became obvious that there were far more cleaning issues than I expected. However, in the interest of time I selected 8 quality issues and 3 tidiness issues. I mostly focused my efforts on cleaning the data that mostly focused on the dogs, which I titled the “dog_data” DataFrame. However I did fix some things in the other 2 DataFrames as well.

First I converted the timestamp columns in the “tweet_data” and “dog_data” frames from string type to datetime type. Next I renamed the “id” column in the “tweet_data” frame to “tweet_id” so that I could join the frames together later using the “tweet_id” column as a primary key. I then spent some time removing those rows that had very low favorite counts and retweet counts, due to the fact that this is such a popular twitter account, it is very unlikely that there would be any tweets with 0 favorites. Since I had no way to recover the actual data regarding the favorite amount for these tweets and I will be using them in a later analysis I had no choice but to drop them. I then went into the “image_predictions” frame and standardized the capitalization for all the dog breeds. Since I only want to use tweets that are from this specific account, I went ahead and dropped tweets that were actually retweets, or were replies, rather than original tweets. I then dropped some extra columns that I had in the “dog_data” frame.

Eventually I was able to get the “dog_data” frame clean enough that I could melt the 4 dog type columns into one column called ‘dog_type’. This left me with a ton of duplicates so I sorted the frame by the ‘dog_types’ value count and dropped the duplicate tweet_ids. I then dropped the extra column from the melt and converted the ‘dog_type’ column to categorical data type from string type.

Finally I dropped the very few numerators that were way above 20 and the ones that were 0 or negative. I did this because I want to do analysis based on average rating and these outliers would affect the results. I then merged the DataFrame’s together based on the ‘tweet_id’ column as a primary key. I did this so that I could do analysis and visualisations more easily.