# Wrangle Report

Firstly, I had to gather the data I needed from three separate sources. The ‘ Archive Enhanced Data’ was the most straightforward to obtain, as all it required was for me to save the file into the directory containing my Jupyter Notebook, and then use the basic ‘pd.read_csv’ to open it.

The image predictions data require use of the Python’s request library to extract the desired data using the URL provided by Udacity. Once the file had successfully been obtained, I was again able to use the ‘pd.read_csv’ function to open it.

Lastly, I tried used the twitter API method to extract the enhanced api data, though I ran into problems and after consulting on the Udacity forum, proceeded instead to use the JSON.txt file provided by Udacity, and read it into a pandas dataframe line by line.

At this point it was time for the Assessment step. I looked through each of the dataframes in Jupyter Notebook and spotted most issues within the notebook without needing Excel. Once I had made the assessments, I tallied up the 8 quality and 3 tidiness issues and saved copies of each of my dataframes at this point so that I could use them in the Cleaning stage without affecting the data.

The next step was to clean the issues that I had found. I decided to start with the tidiness issues, initially combining the dataframes into one using the merge function, as this made it easier to execute the following cleaning steps and it removed tweets from consideration that the datasets didn’t have in common.  I followed this with another tidying step to combine the dog stages into one column, firstly using the groupby function to see all combinations of the dog stages in the dataframes rows, then creating a stage column as a sum of the individual dog stage columns, and assigning new values to these based on the values I initially observed in the groupby table.

The first change I made to the messy data was to remove the replies by restricting the dataset to only those values which are not replies. I then repeated the process in order to remove the retweets. After that it was time to change the datatypes, so I changed the timestamp column to a datetime object using pd.todatetime, and I also changed the source column to a category. In order to correct the ratings, I had to split the process into two steps:

1) To capture decimals, I extracted the entire value from the text column into numerator and denominator columns, then converted the extracted column to floats and replaced the original dataframe’s columns using these newly extracted values.
2) Removed invalid ratings by searching for the few rows with a denominator not equal to ten, then read the text to deem whether the rating was valid or not before replacing.

The next messy data cleaning step involved cleaning up the source column, using the replace function to re-write the three options in a simpler form, then I used the replace function once more for the next step, replacing all of the ‘none’ values with NAN for consistency. I replaced the three image prediction columns with a single column with with only the best dog prediction (p1 being bettere than p2, p2 better than p3 for example) by first defining the row then using if/elif to select the first row that actually contained a dog.  Lastly, I dropped all the unnecessary columns that I wouldn’t need for my visualization.