Reporting: wrangle_report

- This wrangle report is a document that briefly describes my wrangling efforts.

First, I imported the important libraries. Libraries such as **pandas** and **numpy** for analysis, **requests** for programmatically downloading files, **tweepy** for querying Twitter's API, **matplotlib** and **seaborn** for visualization amongst others.

Data Gathering

In this section, I gathered data from three different sources or in three different ways. And also handled files in different formats such as JSON, txt, tsv and csv.

The first file (a csv file) named **'twitter_archive_enhanced.csv'** was provided by Udacity, so I directly downloaded the data using pandas to read the csv file into a dataframe and named it **df1**.

The second file (a tsv file) named **'image_predictions.tsv'** was located on a site. I had to use the request library to programmatically download the file. After which I used pandas to read the file into a dataframe, and named it **df2**.

The third file was gotten by querying Twitter's API using Tweepy (a twitter library). The file returned was in JSON format, but for scalability and reproducability I had to write each line of the JSON file into a txt file named **'tweet_json.txt'**. This txt file is then read line by line into a list, and likewise converted into a dataframe named **df3**.

The three dataframes (**df1, df2, df3**) gotten so far are all related, making it a relational data. Therefore, I merged them into one dataframe, and did a join on the tweet_id column, which they all had in common. I had to import **reduce** and use the **lamda** function for this merging operation.

Assessing Data

In this section, I detected and documented **eight (8) quality issues and two (2) tidiness issue**. Which I assessed **both** visually and programmatically.

The eight quality issues are as follows:

1. The timestamp column should be in datetime format not object format.

2. Retweet values in row dropped.

3. The 'tweet_id' column should be in string format not integer format.

4. The 'floofer' column should be renamed to 'floof'.

5. The 'source' column looks too messy and clusters the table, will be dropped.

6. The type of dogs in p1, p2 and p3 column contains both lower and uppercase letters, needs to be handled.

7. The 'in_reply_to_status_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp', 'in_reply_to_user_id' column has too many null values, and won't be relevant to this analysis, it will be dropped.

8. Incorrect ratings in columns fixed.

The two tidiness issues are:

1. The 'doggo', 'floofer', 'pupper', 'puppo' column should not be a column because it is an observation.

2. The three datasets will be merged into one, because it is relational data.

Cleaning Data

In this section, I cleaned **all** of the issues I documented while assessing.

First, I created a copy of the merged dataset. This was done to preserve the original copy and so it can be easy to revert to the original dataset, incase of issues with cleaning this copied data set.

In handling the first quality issue, I used pandas **to_datetime** function to convert the 'timestamp' column from object to datetime format. For the second quality issue, I kept only the retweet rows that were nulls, this led to the loss of the non-null retweet rows. For the third quality issue, I convert the 'tweet_id' column from integer to string format using **.astype**. The fourth quality issue, was handled by changing the floofer column to floof (this should be the ideal according to the dog dictionary) using **.rename**. For the fifth quality issue, I dropped the 'source' column as it was too messy, using the **drop** function. I handled the sixth quality issue by using **.str.lower** to make all the enteries in the different dog type columns uniform (all lowercase). The seventh quality issue, I used the **drop** function to drop the 'in_reply_to_status_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp', and 'in_reply_to_user_id' columns. For the eighth quality issue, I fixed some of the ratings columns that was incorrectly recorded.

For the first tidiness issue, a new column was created called 'dog_stages' from the values of 'doggo', 'floof', 'pupper', 'puppo' columns. And the 'doggo', 'floof', 'pupper' and 'puppo' columns were dropped. For the last tidiness issue, all the three dataframes (df1,df2,df3) were merged to form one dataframe, although this operation happened before the cleaning operation began.

Finally, I saved the gathered, assessed, and cleaned master dataset to a CSV file named "twitter_archive_master.csv, using the **to_csv** function.