Data Wrangling Report:

The layout of this report is shown below:

- Gather
    - Twitter-archive-enhanced.csv
    - Image_predictions.tsv
    - Twitter API

- Assess & Clean
    - Quality issues
    - Tidiness issues

Gather:

1. Manually download twitter-archive-enhanced.csv from Udacity Project page and read the file to a tabular form in jupyter notebook using pd.read_csv and stored in a variable named archive.
2. I used the requests module and saved requests.get(url) contents into a tsv file. The file is then read in by using pd.read_csv('image_predictions.tsv', delimiter= '\t') and stored in image predictions dataframe
3. At last, I applied for a developer account on Twitter and got granted access to use Twitter API. Twitter API and twitter ids from #1's file is combined to get access to the post's information in json form, which is later written into a txt file called tweet_json.txt. The file is later read by json.loads and stored into a tabular form called counts using pd.DataFrame().

Assess:

** Each step is performed with checking

QUALITY:

1. Remove non-original Tweets from archive dataframe
    a. By API documentation's definition of retweeted status id, retweeted_status_user_id, retweeted_status_timestamp, having these means that the post is not an original post, therefore rows that contain any of these are removed.
    b. By definition if 'in_reply_to_status_id', 'in_reply_to_user_id' are not null, they are replies, which are not original posts, therefore discarded.

2. Re-extract rating numerators and denominators from text in archive dataframe
    a. There are cases where the numerators are extracted wrongly from tweet content. Therefore, regex expressions are used to extract the correct ones, 4 were found different. Then the extracted columns are casted as type float for numerator and int for denominator.

3. Modify rating denominators from archive df to make it 10 or remove row

a. Most of the dataset contains ratings with denominators of 10, however 17 of them don't. In order to make ratings consistent, indivisible denominators are filtered out and combined with their image from the image prediction dataset to check if the denominator can be modified without discarding the row.

   I found that in pictures where multiple dogs are present, the denominators are multiplied by the number of dogs. Also, there may be ratings with denominators indivisible by 10. Therefore, a function is written to divide denominators and numerators that are divisible by 10 with the number of dogs in the pictures; and the ones indivisible by 10 are removed.

4. Cast Retweet and Favorite counts to numerical data, integers in counts dataframe
5. Remove columns not necessary in the archive dataframe
    a. Since 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp', 'in_reply_to_status_id', 'in_reply_to_user_id'all contains null values, these columns are removed.
6. Tweet_id is changed to categorical in all 3 dataframes
7. Duplicated pictures are deleted from the image predictions
8. Dataframes of counts, image predictions, and archive are merged using inner join on the tweet_ids
9. Inconsistent alphabet cases in dog/item names in p1~p3 of image predictions are change to lowercase

TIDINESS

10. Age stages of dog are combined into one column
    a. One column is added first as None if the stage is not specified. Then by making them 1s and 0s, which is one-hot-encoding, we can reverse it by using idxmax(1) to get the index where the row is non-zero. After a new column of the result of idxmax(1) is added, originally 5 columns are dropped.

11. Reformat column names → lowercase_lowercase
    a. Reformat column names so they are more intuitive
12. create rating ratio
    a. Create rating ratio and drop the rating_denominator and rating_numerator

FINAL

13. Checkups
    a. Final clean up of dtypes, Image number is casted as integer instead of object
    b. checked no null values in each row
    c. timestamp doesn’t exceed 2017-08-01
14. Save cleaned dataframe to 'twitter_archive_master.csv'