DATA WRANGLING REPORT

Data wrangling has three steps called gathering, assessing and cleaning data.

The main objectives of this project was to wrangle, store the cleaned file then analyze and give insights and visualizations.

Gathering

Three pieces of data were used for this project and collected in different ways, they were then loaded on a notebook. The data collected were:

- WeRateDogs Twitter archive data (twitter_archive_enhanced.csv) which was directly provided to download.
- Tweet image prediction file (image_predictions.tsv), a url was provided and the data was downloaded using the requests.get ( ) function.
- Tweepy library was used to query additional data via the Twitter API (tweet_json.txt), which was downloaded programmatically using the tweepy API, in order to use it a twitter developer's account is needed in order to access the consumer keys and the authentication tokens. The json data was then written on its own line.

Assessing and cleaning data

Quality

Twitter archive dataset

Since we want data without any retweets, we need to remove the tweets with a retweet status id thus remaining with only the ones with null status IDs.

The timestamp data type was converted to date time from object using to_datetime ()

The source column had href tags, this was removed using regular expressions.

In the name column, the data wasn't valid since some of the names started with a lowercase letter instead of starting with a capital letter.

The text has working urls for the twitter pages, these are extracted using the regex functions.

Columns that aren't necessary for analysis of this data were then dropped.

Image prediction dataset

The names of the dog predictions are inconsistent in some are start in caps while others don't, in order to achieve consistency the names are all converted to lowercase.

Since p1 has the most accurate prediction, it means that if the prediction of whether it's a dog is false affects the other two predictions bringing about inaccuracy.

Tidiness

Twitter archive dataset: Doggo, floofer, pupper and puppo should all be in one column, none values were all converted to NANs, this could not be melted since some columns have more than one dog stage. Thus .stack () was used to drop null values then the rest were grouped together.

Since there are too many data sets, this brings about untidiness. Each observational unit forms a table, only two tables were left, the twitter archive dataset and the json dataset were combined.

Below is the .info () of the dataset before and after cleaning. Before cleaning the dataset had 17 columns and after cleaning and merging only 12 necessary columns are left:

Before assessing and cleaning	
tweet_id	2356 non null int64
in_reply_to_status_id	78 non null float64
in_reply_to_user_id	78 non null float64
timestamp	2356 non null object
source	2356 non null object
text	2356 non null object
retweeted_status_id	181 non null float64
retweeted_status_user_id	181 non null float64
retweeted_status_timestamp	181 non null object
expanded_urls	2297 non null object
rating_numerator	2356 non null int64
rating_denominator	2356 non null int64
name	2356 non null object
doggo	2356 non-null object
floofer	2356 non-null object
pupper	2356 non-null object
puppo	2356 non-null object

After assessing and cleaning	
tweet_id	2167 non null int64
timestamp	2167 non null datetime64[ns]
source	2167 non null object
text	2167 non null object
expanded_urls	2167 non null object
rating_numerator	2167 non null int64
rating_denominator	2167 non null int64
name	2167 non null object
direct_links	2109 non-null object
stage	2167 non-null object
retweet_count	2167 non-null int64
favorite_count	2167 non-null int64