Wrangle Report

Introduction:

Real-world data rarely comes clean. Using Python and its libraries, I have gathered data from a variety of sources and in a variety of formats, assessed its quality and tidiness, then cleaned it. This is called data wrangling.

It is a process divided into 3 main steps:
• Gathering • Assessing • Cleaning

Gathering:
Data was gathered from 3 different sources:

1) *From WeRateDogs Twitter archive given by Udacity in csv format:* Using panda's method 'read_csv', I managed to read the data stored in the file 'twitter-archive-enhanced.csv'. I stored it in a DataFrame called 'twitter_archive'. The data had issues for cleaning and resolving.
2) *Image prediction file downloaded programmatically using Requests library and the URL provided by Udacity in tsv format:* Using Requests library and 'get' method, data was downloaded in a file 'image_predictions.tsv'. Then, the content was stored in a DataFrame called 'image_predictions' using pandas' method 'read_csv'.
3) *Data retrieved by querying Twitter's APIs and using Tweepy library.* Using the list of tweet_id's in dataframe 'twitter_archive', I made a loop through each tweet and query Twitter's APIs with the tweet ID to get each tweet's JSON data. Then, I retrieved the required data ('favorite_count', 'retweet_count', 'followers_count', 'favourites_count', 'created_at') and stored it in a list called 'df_list'. There were some errors, and the tweet_id of each error was stored in a list called 'error_list'. Finally, I created a DataFrame called 'tweet_data' using the list.

   Result:
   We reached the limit of Twitter APIs 2 times.
   We got _ tweet_id correctly and _ errors.
   The total time was about _ seconds = 32 min.

Assessing:

After gathering the data and storing them in DataFrames, the following step was assessing the data for quality and tidiness. Data were assessed programmatically and visually.

Quality: Issues with content. Low quality data is also known as dirty data. Identified quality issues are:

Quality:

- In several columns null objects are non-null (None to NaN).
- There are invalid names (a, an and less than 3 characters).- We only want original ratings tweets, not retweets.
- Some tweet_ids have the same jpg_url
- Some tweets have 2 different tweet_id, that are retweets.
- in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id should be integers/strings instead of float.
- retweeted_status_timestamp, timestamp should be datetime instead of object (string).
- The numerator and denominator columns have invalid values.
- We might change the type of columns: (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id and tweet_id) to string since we aren't doing any actions on them.
- Sources are not readable.
- Missing values from images dataset (2075 rows instead of 2356)

Tidiness: Issues with structure that prevent easy analysis. Untidy data is also known as messy data.

Tidiness:

- Dog stage is in 4 columns (doggo, floofer, pupper, puppo), this was not necessary.
- Merge 'tweet_info' and 'image_predictions' into 'twitter_archive'.

Cleaning:
It is the process of fixing and resolving issues identified in the Cleaning process. The (define, code, and test) steps were used in the cleaning process. First, copies of the DataFrames were created before cleaning. Then, the steps of cleaning were applied iteratively on all issues.

Storing:
The final DataFrame called 'twitter_archive_clean' contains 1990 rows and 15 columns with the correct data types. The dataset is then stored in a csv file called 'twitter_archive_master.csv'. At this point, the data was successfully wrangled and therefore ready for analysis and visualization.

Analysis & Visualization:
Analysis & Visualization reinforces correct and accurate insights without performing data wrangling first. Visualizations and insights are provided in ‘act_report.pdf’.