Data Wrangling is a serious business for Data Analysts, without the application of some wrangling efforts, it is impossible to get good analysis results.

In the data wrangling project, I focused on three main points of wrangling:

1.Gathering

2. Assessing

3. Cleaning

4. Saving the data, getting insights, visualizations

Gathering

I gathered three datasets for the wrangling project, using three different gathering methods:

- Manual download of data via a provided link: I manually downloaded my data from the provided link and then uploaded the data to my google drive which is accessible from my working environment which in my case is Google Colab. In order to easily access my data, I mounted the google drive onto colab and was able to get the file path through which I read in the data to a dataframe.
- Programmatic download of data using Requests library via a provided link: Owing to the fact that web link contents can always change, the best way to download from a link is to download programmatically, no matter the change in the website contents, as long as we run the download link, we can always get updated contents of the website.
- Querying the Twitter API for each tweet's JSON data using Python's Tweepy library: First to be able to query twitter data, I had to get a developer's account which gave me access to the access key and token and then the consumer key and token whichl was used in querying the data through tweepy, thereafter, I stored each tweet's entire set of JSON data in a file called tweet_json.txt file using glob and json library.

Assessing

After a visual assessment of the data in both my working environment and excel, I found these:

Quality Issues

1. "None" has been used to represent empty records in "name", "doggo", "fluffer", "pupper", and "puppo" column which makes it seem as if the records were never empty even when it was accessed programmatically.
2. Some urls under “extendedurls” column contain two urls separated by ',', this error is making the urls lead to a page that doesn't exist.
3. There are missing records in the "name" column of df_1 and some dogs have been wrongly named as "a", "an", "the".

Tidiness Issues

1. The "floofer", "doggo", "pups" columns are meant to all be in just one column, they are all dog stages but they've been spread across multiple columns.
2. There is a link attached to the end of all the texts in each row of the 'text' column in df_1. The link needs to be extracted out and turned into an individual column on its own.

Programmatic assessment via methods/functions like “.isnull().sum()”, “.duplicated()”, “.info()”, “.describe()” helped me take a closer look at the data to get other issues that couldn’t have been noticed via visual assessment.

Quality Issues

4. There are several columns with missing data in df_1, such as in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp,expanded_urls.

5. Incorrect data types for columns tweet_id, in_reply_to_status_id, in_reply_to_user_id,timestamp,retweeted_status_id,retweeted_status_user_id,retweeted_status_timestamp in df_1, then tweet_id in df_2 and id in df_3.

6. Some dogs are in groups and have therefore been rated more than over 10, there may need to be an extra column to indicate these dogs.

7. Non-Uniform name for "pups" in the df_1 table, "pups" is represented as "pupper" and "puppo", hence the two columns may need to be merged and all the values replaced with "pups".

8. There are missing records in the "doggo", "floofer", "puppo" and "pupper" columns of df_1.

Tidiness issues

3. Information about the same set of tweets is spread across three dataframes, the dataframes need to be merged for the data to be considered tidy.

Cleaning

Owing to the nature of the dataset, I had to do a combination of both manual (cleaning via excel formulas and methods) and then programmatic cleaning. Most of the cleaning methods used can be found in my code script. Cleaning took the majority of time but thereafter, it was easy merging the three data gathered together.

Saving the data, getting insights, visualizations

In order to get insight from the cleaned data, I merged the three datasets tgether using .merge() with the argument “how”, then I saved the merged dataframes into a csv file using “.csv”. Thereafter, I was able to read the csv file into my working environment and then perform some analysis to get insights and visualizations.