WRANGLING EFFORTS

- The set of the 3 data set required high cleaning rate because especially the twitter enhanced csv was pretty large and, in some cases, visual assessment was not possible and somehow not applicable.

- Most of my wrangling analysis came from the programmatic assessments and I came up with the following quality and data issues;

Quality issues

Twitter enhanced csv: wrd_df

1.Rating denominator ranges, because only 2333 has it out of 10, they shoul al be similar

2.variable name has 745 None values and 55 "a" values

3.Many rows example in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp have so many missing data(NaN) that cannot be filled.

4.Many columns in this data set do not provide enough information for analysis amd all nedd to be dropped. Foe Example in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp that need to be removed.

5.Some of the names in the names columns are not consistent, ie Lower Case example, 'a', and 'an' which are also weird names.

6.The expanded urls column, the links are not working , maybe slicing to see the gadget source will have a more useful information.

7.420 is pretty high for a rate out of 10, so that is an outlier that needs to be sorted out.

Image prediction csv

8.Dropping the img num column, it has no use whatsoever.

Tweets api

9.The data frame has so many missing data in so many columns like contributors, coordinates, geo, place . Infact, these columns contain no values.

10.This data frame has so "useless" information that is inadequate for analysis. Example the in_reply_to_screen_name , in_reply_to_status_id,in_reply_to_status_id_str

,in_reply_to_user_id ,in_reply_to_user_id_str,is_quote_status columns. I saw I only needed the id, favourite count, and retweet count to be the only useful columns.

Tidiness issues

1.Columns doggo, floofer, pupper and puppo are in different columns

2.jpg_url variable should be in tw_arc table to satisfy tidiness definition

The first thing I dealt with is dealing with the numerator and denominator. After using the value_counts on these columns, the highest denominator was 420 and that was a bit extreme since it was out of 10, standard score.

So I did set all the values above 420 to be 14

```
wrd_clean.loc[wrd_clean['rating_numerator']>14, 'rating_numerator'] = 14
```

The denominator, I set all the values to be over 10, so that it is standard

```
wrd_clean["rating_denominator"]=10
```

Issue 2: The df_tweets data frame had a lot of unnecessary columns,

Example

So, I decided to use the groupby function to create a new data frame with the columns that I needed for analysis

```
tweets_api_df=df_tweets.groupby("id", as_index = False)[

                  ["favorite_count","retweet_count"]].mean().sort_values(

                                                            by="favorite_count",ascending=False)
```

Issue 3:Using the melt function to join all the columns with dog_types

Since the columns 'doggo', 'floofer', 'pupper', 'puppo, are all dog types and that means that the most of the values are NaN or None

Using the melt function like this to create a new column called dog_stage

```
dogs = pd.melt(dogs, id_vars =['tweet_id', 'name',
'rating_numerator','rating_denominator'],

                value_vars = ['doggo', 'floofer', 'pupper', 'puppo', 'unknown'],

                var_name = 'dog_stage',

               value_name = 'value')

df=pd.merge(dogs_df,tweets_api_df , how="left", on=["tweet_id"])
```

```
df.shape

master_df=pd.merge(df,wrd_clean , how="left", on=["tweet_id"])
```

3.Changing the timestamp column to a datetime format and extract the months that I used later in analysis

```
master_df['timestamp']=pd.to_datetime(master_df['timestamp'])

months=[dates.strftime('%B') for dates in master_df['timestamp']]
```

4.After joiing the data frames and inspecting it for any missing values and duplicates,

The favorite tweet column had 8 missing values and I imputated it with mean value.

```
mean_value=master_df['favorite_count'].mean()

master_df['favorite_count'].fillna(value=mean_value, inplace=True)
```

N.B; I eventually dropped the retweet column because it was not necessary in this project.