WRANGLE REPORT.

Data wrangling can be quite intense depending on the quality of the dataset one wants to analyze and the number of the datasets to be analyzed.Doing so with the datasets given was not different.

The Process.

I started the data wrangling process by first gathering the datasets from different sources and after that, I started assessing them. The dataframes I was analyzing were, df, df_1 and df_2.

There are two main ways of assessing the dataframes that are visually and programmatically. Visual assessment is when one goes through a dataframe to see if there are any quality or tidiness issues.
Through visual analysis in the first dataframe, df,several observations were made. The name column had names of dogs that are nouns but there were other names like very, a to name a few that were not names for the dogs.They needed to be replaced with NaN to show that those names were not given.

Other observations made were that the doggo, floofer, pupper and puppo were supposed to be in the same column.

1. The first column, tweet_id is in integer form, instead of being a string.

To change the column form, I used the values.astype() function to convert the values from integer form to string.

```
df_clean['tweet_id'] = df_clean['tweet_id'].values.astype(str)
```

2. The timestamp is in object form, it should be changed to timestamp form.

To convert the data type given to timestamp, I used the .to_datetime function and the code was as this,

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

3. The source column is incomplete because only the truncated data can be viewed.

Using the set.option('display.max_coldwidth',None) to display the whole text content in the source column this is because there was an indication of incomplete text by this(...)

4. Extracting the source name from the source column.

After str.stripping(), I used the str.replace() to replace the '</a' with space ' ' and then I identified the unique values which were the sources of the tweets using the .unique() function.

5. Change all the names in the name column that start with a lowercase letter to NaN.

Identified all the names in the lowercase, using islower() function. All names that are not nouns were displayed and then they had to be replaced with NaN.

The second dataframe, df_1 had the following quality issues:

1. The tweet_id is in integer form whereas it's supposed to be a string.

To change the column form, I used the values.astype() function to convert the values from integer form to string.

2. Change the titles p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_conf and p3_dog to be meaningful titles.

The .rename() function was used to convert the p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_conf and p3_dog to be meaningful titles. P1 was renamed to prediction 1, p1_conf to prediction 1_confidence, p1_dog to prediction 1_dog, p2 to prediction 2, p2_conf to prediction 2_confidence, p2_dog to prediction 2_dog, p3 to prediction 3, p3_conf to prediction 3_confidence and p3_dog to prediction 3_dog.

3. The names in column p1 are separated by a hyphen and should be capitalized.

Dropped the hyphen between the names by using the .replace() function with space, and capitalizing the names using the .str.capitalize() function.

4. The names in column p2 are separated by a hyphen and should be capitalized.

Dropped the hyphen between the names by using the .replace() function with space, and capitalizing the names using the .str.capitalize() function.

5. The names in column p3 are separated by a hyphen and should be capitalized.

Dropped the hyphen between the names by using the .replace() function with space, and capitalizing the names using the .str.capitalize() function.

The third dataframe, df_2, had the following quality issues:

1. Id_column is an integer and it's supposed to be a string.

To change the column form, I used the values.astype() function to convert the values from integer form to string.

2. Id column to be renamed tweet_id.

Using the .rename() function, I renamed the id column to tweet_id.

The following were the tidiness issues identified:

1. Add a column called rates and compute rates by taking numerical value/ denominator value.

By adding another column called rates, I had to take the numerical value/ denominator value. This was in order to get the rating of each individual dog.

```
df_clean['rates'] = (df_clean['rating_numerator']/df_clean['rating_denominator'])
```

2. In_reply_status_id, in_reply_retweet_id, retweet_start_id, retweeted_status_user_id, retweeted_status_timestamp columns should be dropped.

All these columns had to be dropped since they were not going to be used for the analysis and their values were NaN. This was executed using the .drop() function.

3. Doggo, floofer, pupper and puppo are one variable and should be in one column.

Since doggo, floofer, pupper and puppo are affection names, they fall under one variable and should be in the same column.
This was done using the .melt() function to melt the columns to one column called, slang and their values column called name_2.

I started by working on the quality issues first using the define, code and test procedure and then after working on each data frame individually rectified the tidiness issues.
Thereafter, analyzing and visualization was done.