Wrangle Report

The purpose of this report is to document and describe my wrangling efforts.

As stated in the Read Me file, we had to obtain data in 3 different ways.

1. We were given a ‘csv’ file which can resemble data given to us by a colleague or a company.
2. We had to download another file ‘image-predictions.tsv’ which can be compared to extracted a file from a website.
3. Finally, we had to extract the last piece of information from Twitter themselves. This was facilitated by the Tweepy wrapper available for python.

This is what the files contained:

-twitter-archive-enhanced.csv : a compiled list of approximately 2300 tweets, with their ID’s. This table contained dog ratings and data on dog stages.

-image_predictions.tsv : this is a file that was created by Udacity.com it contains the outputs after each image from X to Y date was ran through a neural network (image recognition algorithm) in an attempt to determine the dog’s breed.

-information gathered through Twitter’s API: When I ran the script for this section, I obtained JSON objects with 310 keys for every tweet.

Wrangling and cleaning the data:

The goal here was to combine all 3 datasets and create a cleaner table. I had to spend a lot of time investigating the datasets to identify issues. This meant printing out specific results and analyzing specific columns and the corresponding tweet. Below I will explain the algorithms used for some of the tougher issues.

Problem 3: Some of the ratings in twitter-archive-enhanced.csv were not accurate.

I solved this issue by analyzing the text content of the tweets. I separated the text using the .split('/10)[0] string method. This separates the string in portions depending on where the '/10' (the denominator) is found. Since we are only interested in the numerator here, I only considered the first part with the [0] argument. The final step was extracting all the numbers using the 'Regular expression' library which I imported with 'import re'. Since the result can output multiple numbers, I made sure to select the last element in the list with [-1]. This element is the numerator we are looking for because the string section ends right before the '/10' for the denominator. The .apply() function lets us work on columns row by row in a quick and efficient manner. The set function lets me work on a specific set of indexes. I made sure not to include the ratings solved in 'Problem 2' as those included ratings like 84/70 and they did not contain '/10'

```
df_clean.loc[set(df_clean.index) - set(ratings_with_denom_not_10),'rating_numerator'] =
df_clean.loc[set(df_clean.index) -set(ratings_with_denom_not_10)].split_text.apply(lambda x:
re.findall(r"\d*\.\d+|\d+", x)).apply(lambda x: ''.join(x[-1])).apply(lambda x: float(x))
```

Problem 5: The dog_stages in 'twitter-archive-enhanced.csv' contained many errors:

I decided here to re-calculate everything by re-extracting the dog stages from the text column.

```
floofer_rows = df_clean[df_clean['text'].str.contains("floof", case=False)].index
```

This was the code used for floofer. The reason I chose floof was because I wanted to consider 'floof', 'floofer' 'floofs' or any other combination of the word 'floof'. Once the index was found I set the value in the column with the .loc command. I repeated this for every dog stage.

```
df_clean.loc[floofer_rows,'floofer'] = 'floofer'
```

Problem 8: In certain cases, the algorithm used to predict the dog breed did not predict a dog at all.

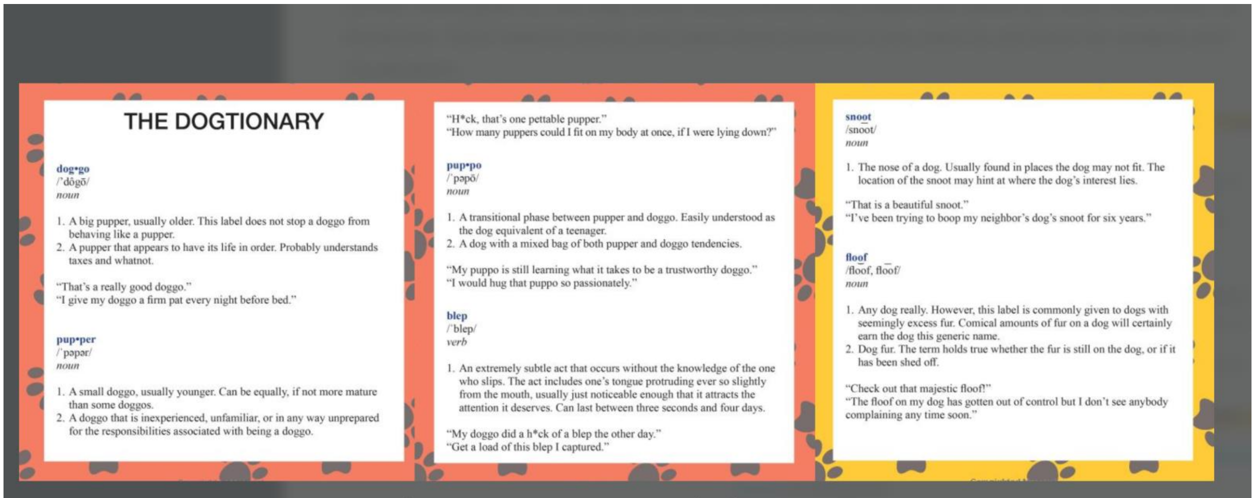
I took the decision here to remove these rows (324 in total) from the table. This was a quality issue and not doing so could have given us erroneous results since we are only interested in ratings associated with dogs. I discovered this was a problem with the following command:

```
image_predictions_clean.query('p1_dog ==0 and p2_dog==0 and p3_dog==0').tweet_id.count()
```

Problem 10: Combining all the dataframes and creating a master dataframe:

I dedicated a small section in the code for this part. I trimmed out the columns which I didn't need anymore to obtain and smaller and cleaner dataframe. I used inner merges to make sure the final database only contained dogs as confirmed by the image_prediction algorithm. Also, a few tweets have since been deleted so I didn't want to include those for the retweet and favorite/like portion of the work.

See appendix for dog stages:



THE DOGTIONARY

dog•go
/ˈdôgō/
noun

1. A big pupper, usually older. This label does not stop a doggo from behaving like a pupper.
2. A pupper that appears to have its life in order. Probably understands taxes and whatnot.

"That's a really good doggo."
"I give my doggo a firm pat every night before bed."

pup•per
/ˈpəpər/
noun

1. A small doggo, usually younger. Can be equally, if not more mature than some doggos.
2. A doggo that is inexperienced, unfamiliar, or in any way unprepared for the responsibilities associated with being a doggo.

"H*ck, that's one pettable pupper."
"How many puppers could I fit on my body at once, if I were lying down?"

pup•po
/ˈpəpō/
noun

1. A transitional phase between pupper and doggo. Easily understood as the dog equivalent of a teenager.
2. A dog with a mixed bag of both pupper and doggo tendencies.

"My puppo is still learning what it takes to be a trustworthy doggo."
"I would hug that puppo so passionately."

blep
/ˈblep/
verb

1. An extremely subtle act that occurs without the knowledge of the one who slips. The act includes one's tongue protruding ever so slightly from the mouth, usually just noticeable enough that it attracts the attention it deserves. Can last between three seconds and four days.

"My doggo did a h*ck of a blep the other day."
"Get a load of this blep I captured."

snoot
/sno͞ot/
noun

1. The nose of a dog. Usually found in places the dog may not fit. The location of the snoot may hint at where the dog's interest lies.

"That is a beautiful snoot."
"I've been trying to boop my neighbor's dog's snoot for six years."

floof
/flo͝of, flo͞of/
noun

1. Any dog really. However, this label is commonly given to dogs with seemingly excess fur. Comical amounts of fur on a dog will certainly earn the dog this generic name.
2. Dog fur. The term holds true whether the fur is still on the dog, or if it has been shed off.

"Check out that majestic floof!"
"The floof on my dog has gotten out of control but I don't see anybody complaining any time soon."

The Dogtionary explains the various stages of dog: doggo, pupper, puppo, and floof(er) (via the #WeRateDogs book on Amazon)

Referenced book:

https://www.amazon.com/WeRateDogs-Most-Hilarious-Adorable-Youve/dp/1510717145