Wrangle Report

Gathering Data

To gather data I pulled in data from 3 different data sources.

1. Link to a CSV
2. Link to TSV
3. Twitter API (tweepy)

I pulled in the CSV and TSV files using the request library paired with the IO library and was able to decode both using UTF-8.

The twitter API pulled in json format data which needed looping tweet by tweet to pull in the data to a pandas dataframe. I chose to pull in favorite count, retweet count, lang, source and followers count (ended up not using followers count). Because of tweepy rate limits I had to pass wait parameters to run the api successfully in one go.

Assess

Dogs DataFrame

Quality

these fields do not have complete data, they are also not original tweets (replies/retweets) and should be removed

> *in_reply_to_status_id*
> *in_reply_to_user_id*
> *retweeted_status_id*
> *retweeted_status_user_id*
> *retweeted_status_timestamp*

- tweetid fields need to be converted to strings
- Timestamp is not in the correct datetime format
- Source field contains html tag format and can be pulled from twitter API dataframe
- Name contains errorneous values (all lowercase values)
- investigate duplicate urls, but unique tweetids
- expanded_urls appears to have missing data

Tidiness

- Dog stages should be a single column rather than four; one of the requirements for tidy data is that each variable forms a column.

Prediction Dataframe

Quality

- investigate duplicate images, unique tweet_ids

Tidiness

- These prediction can be appended to the dogs df

API DataFrame

Quality

- Dedupe
- tweet likes and retweet counts need to be converted into integers

Tidiness

- These values could be appended to the Dogs df

Clean

I was able to find more than the minimum cleaning requirements.

The code to perform the cleaning will be available in my Jupyter notebook. This document will outline the reasoning and thought process of why I chose to clean a particular issue.

Quality 1

The fields “in_reply” and “retweeted_status” come to a total of five unique fields. These fields indicate if the tweet in question is a reply or a retweet. Because we only want to analyze top level tweets these fields should not contain any data. I did find 78 replies and 181 retweet records. I removed these records and then eventually removed the fields completely in a tidy step later on.

Quality 2

I found 55 dog names that contained just the letter “a”. Looking into it further I found that the erroneous names were all in lowercase. I cleaned these records up by setting these names to empty strings. While “a” or “the” can be a legitimate dog name, higher volumes of these odd occurrences appears to be too common and is likely a data issue.

Quality 3

The timestamp field was not in datetime format. In order to properly use this data in datetime order I needed to update the datatype to datetime. The datetime library was used.

Quality 4

There were a handful of instances where the “Expanded URLs” fields were missing. Since we want to be able to have clean data that can be looked up and referenced, I removed these records.

Quality 5

The Source field contained html tags. I identified that there were only four different sources and hardcoded the update. If there were more than a few sources I could have stripped the html and found a way to programmatically derive a user-friendly way to represent the source.

Below are the different source fields that I ended up needing to reduce.

```
<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>     1962
<a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>                          91
<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>                       30
<a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>      11
Name: source, dtype: int64
```

Reduced to

```
iPhone        1962
Vine            91
Web Client      30
TweetDeck       11
Name: source, dtype: int64
```

Quality 6

I found duplicate URLs in the dataset. There was only one occurrence but I used the duplicated() method to help remove the duplicates.

Duplicate URL:

```
2212    https://vine.co/v/ea0OwvPTx9l
Name: expanded_urls, dtype: object
```

Quality 7

I updated the tweet ids of all three datasets to be string. Tweet ids are not to be used with arithmetic functions and may contain leading zeros. It's best practice to have these fields as string.

Quality 8

There were numerators that contained decimals but the data did not extract these scores correctly. I used the text field and pulled the numerator and denominator using regular expressions to update the numerator with the proper float values.

```
# this will pull ratings from the text that look like ##.## / ##
ratings = df_dogs_clean.text.str.extract('((?:\d+\.)?\d+)\/(\d+)', expand=True)
```

Tidy 1

The dog stages were broken up into multiple columns (doggo, floofler, pupper, puppo). I proceeded to condense the four separate statuses into a single field named "dog_stage". In the event that there was no dog stage prediction I set the dog stage to an empty string (there were a high percentage of None).

Note that there were instances where a tweet had multiple stages

	doggo	floofer	pupper	puppo	count
0					1758
1				puppo	23
2			pupper		220
3		floofer			9
4	doggo				72
5	doggo			puppo	1
6	doggo		pupper		9
7	doggo	floofer			1

To handle these cases I merged all stages together and separated each stage with commas if multiple existed. This was done manually. I could have used the join(‘,’) method if this was a more complex dataset.

```
df_dogs_clean['stage'] = df_dogs_clean.doggo + df_dogs_clean.floofer + df_dogs_clean.pupper + df_dogs
df_dogs_clean.loc[df_dogs_clean.stage == 'doggopupper', 'stage'] = 'doggo, pupper'
df_dogs_clean.loc[df_dogs_clean.stage == 'doggopuppo', 'stage'] = 'doggo, puppo'
df_dogs_clean.loc[df_dogs_clean.stage == 'doggofloofer', 'stage'] = 'doggo, floofer'
```

Tidy 2

The score was broken out into two fields (numerator and denominator). I condensed the score into a percentage for reference. The numerator and denominator are still relevant but visually and analytically it will be easier to aggregate on a single score field.

Tidy 3

I merged these fields into a single DF (final_df)

Final DF

Because we are working with such a small dataset (less than 5000). I decided to merge all datafiles into a single master file. In the event that we continue and expand this analysis for future use it may make sense to keep the prediction data in one file and the tweet data (api and csv) in another.