The dataset I wrangled was from the tweet archive of this Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with humorous comment about the dogs.
The wrangling process involved:

1. Gathering data
2. Assessing data
3. Cleaning data
4. Storing data

1. Gathering data

The first file was a Twitter archive data (csv file), which I directly downloaded, uploaded and read into a pandas dataframe

For the second file, I used the Requests library to programmatically download the tweet image prediction (image_predictions.tsv) from a neural network hosted on Udacity's servers

Finally, for the last file, I used the Tweepy library to query additional data through the Twitter API for each tweet's JSON data and stored each tweet's entire set of JSON data in a file called tweet_json.txt file.

2. Accessing and inspection

After gathering all three pieces of data and reading them into different pandas dataframes, each of them were visually (use of Microsoft Excel) and programmatically (use of pandas methods like “df.info()” and so on) inspected for quality and tidiness issues. On inspection, some quality and tidiness issues were observed:

- Quality issues

1. Presence of 'None' for missing records, instead of pandas 'NaN'
2. From Twitter archive data: invalid data in the 'name' column like 'an', 'a', 'such', 'quite', and so on
3. Inconsistent data types in the 'retweeted_status_timestamp' column
4. 'tweet_id' column contains integer data type which might distort analysis
5. The 'text' column contains both original tweets and retweets, while we were required to work with just original tweets
6. Row with invalid denominator rating of zero (0)
7. Most of the column headers especially in the 'image_prediction' data set are not quite descriptive like 'p1_conf', 'p1_dog', 'p2', and the rest of them
8. Presence of retweet columns that won't be needed

- Tidiness issues

1. Combining the three(3) dataframes: Difference in column header names containing same data ('twitter_id' and 'id'), which will be an issue in merging the data sets
2. Separate columns for the dog stages

3. Cleaning data

I made sure to clean all the issues I identified in order to produce a high-quality and tidy master pandas DataFrame, but first I made a copy of the original data.
During cleaning, I used the define-code-test framework and clearly documented the process.
The Cleaning process involved:

- ✓ Renaming the "id" column in the tweet_json dataframe to "tweet_id" for uniformity with other dataframes; after which the dataframes were merged (2 at a time), “on” the tweet_id column
- ✓ Missing data that displayed the string “None”, were replaced with the pandas NaN for easy analysis
- ✓ The different dog stage columns were summed into one column
- ✓ I observed that the dog names always began with an uppercase letter, hence Regular expressions (with hint on case sensitive letters that occurred after a word or group of words) were used in the “.str.extract()” method to correctly extract the dog names from the 'text' column
- ✓ The datatype of the ‘retweeted_status_timestamp' column was converted to datetime
- ✓ Dropped row with invalid denominator rating
- ✓ The retweet texts usually contained 'RT', and this was used to drop all rows containing retweets using the”.str.contains()” method
- ✓ The ‘time_stamp' column was converted from object to datetime datatype
- ✓ Column headers like ‘p1_conf', 'p1_dog', 'p2', and the rest of them were renamed to something more descriptive (example: p1 was renamed to “prediction1”)
- ✓ Dropped retweet columns

4. Storing data

After cleaning was done, the master pandas dataframe was stored as a csv file (“twitter_archive_master.csv").

This concluded the wrangling process of this project.