Data Wrangling Report Gathering:

1. The WeRateDogs Twitter archive was downloaded manually from Udacity (twitter_archive_enhanced.csv) and saved as a data frame (archive).
2. The tweet image predictions were downloaded programmatically using the requests library, using the URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv and saved to image_predictions.tsv and loaded to a data frame (predict).
3. Additional data from the Twitter API was queried by Twitter API and each tweet's JSON data using the Tweepy library and stored as tweets' entire set of JSON data in a file called tweet_json.txt file. A data frame (twitter_api) was created including only ['tweet_id', 'retweet_count', 'favorite_count'].
4. All three data frames were merged into one data frame called df_final for assessment.

Assessing:

- Using pd.head and pd.tail, some issues were spotted, such as columns with missing values, categorical data spread in more than one column, capitalized and lower case present in string columns, columns with no interest in the subject.
- Using pd.info, some issues were spotted, such as columns with the wrong datatype and different shapes.
- Using pd.describe on the object columns, we can spot a lot of missing dog stages, names, and images.
- Using pd.isnull, 10124 missing values were spotted. These values are spread in:
- Using pd.value_counts on rating_numerator and rating_denominator, values greater than 10 were observed, but WeRateDogs says the rating system is fine.
- Using pd.duplicated, no duplicates were found.
- Using pd.Series.value_counts on 'doggo', 'floofer', 'pupper', and 'puppo' to get a clear view of how stage values are spread and if there are NaN values.
- Using pd.loc to get a random sample of dog's naming, showing single letter names and “None” values.

Cleaning:

- Using pd.copy is good practice before cleaning, and a structure was built ("Define", "Code", "Test").
- Using a lambda expression to fix dog stages.
- Using pd.to_datetime to fix the 'timestamp' type.
- Using a function called Choose_breed to fix dog breed and confidence levels.
- Using pd.str.lower for the fix of the name of dog breeds.
- Using pd.str.replace to fix the name of dog breeds.
- Using np.clip to limit 'rating_numerator'.
- Using pd.sample for the final view.
- Using pd.to_csv to save the cleaned data frame to 'twitter_archive_master.csv'.
- 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp'.