Wrangle Report

Here are my wrangling efforts for the Wrangle_Act notebook:

Gather Data

- Load the appropriate CSV and JSON files into a Jupyter Notebook using Pandas and Request functions.
- Installation of the Tweepy function to access the Twitter API specifically.
- Query each tweet's retweet count and favorite count using the Tweepy Library and stored the data in 'tweet_json.txt'.
- Read the 'tweet_json.txt' line by line onto Pandas DataFrame with id, retweet count, and favorite count. Change the column name from ‘id’ to ‘tweet_id’ for ease of merging tables later on.

Assessing Data

- **Quality Issues**
  - Several columns with NaN values
  - Some names in the ‘name’ column were not names at all
  - Several entries with incorrect ratings
  - Some entries were missing relevant ‘dog type’ data
  - ‘Timestamp’ column is an object instead of a datetime
  - Some ratings contain decimal values
  - Not all dog breed entries are capitalized
  - Some of the data in the ‘dog predictions’ columns were incorrectly predicted as inanimate objects

- **Tidiness Issues**
  - Data is found on three separate data sets. These should be joined.
  - ‘Text’ column contains Twitter URL
  - Having separate ‘dog type’ columns creates a lot of NaN values

Cleaning Data

First step was to make clean copies of the data using .copy().

Next step was to address the quality issues.

1. **Several columns with NaN values** - Remove rows with retweet data by converting null values in the 'retweet_status_id' column to 0 with .fillna(), then sort out all of the id's greater than 0, leaving us with only columns with no retweet status.
2. **Some names in the ‘name’ column were not names at all** - Replace incorrect name values in name column with 'none' using .replace() and regex functions.
3. **Several entries with incorrect ratings** - Replace incorrect rating at index 42 (for tweet_id 883482846933004288) using the .at function.
4. **Several entries with incorrect ratings** - Replace incorrect rating numerator and denominator at index 2154 using .at function.
5. **Some entries were missing relevant ‘dog type’ data** - Include 'floofer' qualification for tweet_id 796080075804475393 using .at function.
6. **‘Timestamp’ column is an object instead of a datetime** - change 'timestamp' to a datetime object using the pd.to_datetime function.
7. **Some ratings contain decimal values** - Create a 'ratings' series in order to properly extract the rating from the 'text' column using .str.extract and regex functions. Convert the ratings to floats with .astype function in order to capture decimal values. Upload these corrected ratings into df_merge.
8. **Not all dog breed entries are capitalized** - Standardize capitalization of dog breeds in the p1, p2, and p3 columns using str.upper() function.

   <u>Additional:</u> Some of the data in the ‘dog predictions’ columns were incorrectly predicted as inanimate objects - Remove columns that predict dogs as anything other than dogs using a boolean condition. This was more of a tidiness issue (see tidiness step 3)

After cleaning the quality issues, I moved on to the tidiness issues.

1. **Data is found on three separate data sets. These should be joined -** Merge dataframes for ease of access using .merge(). *## note that this step was done between quality issues 1 and 2*
2. **Having separate ‘dog type’ columns creates a lot of NaN values** - combine all dog type columns into one column. Use .fillna() to de-clutter and .map() to clean whitespace.
3. *See ‘additional’ quality cleaning step.*
4. **‘Text’ column contains Twitter URL** - Remove the URL from the text block and add to its own column using .str.split with regular expression.

5. **columns have been shifted during data cleaning** - rearrange columns for readability using .loc. Drop redundant columns.

Once finished with the cleaning process, I created a cleaned dataframe called df_clean, then downloaded it as a csv file titled ‘twitter_archive_master.csv’.