Brad DeFauw
badefauw@gmail.com

The contents of this report are centered around my wrangling efforts that ultimately led to insights and visualizations in the twitter_archive_master dataframe. This data was messy, and although I cleaned it in a variety of different ways, I'm sure there is a lot more to this data that I did not clean. The report will be structured as follows.

- Issue that needed to be resolved
    - How I resolved it

Before referencing the code, it should be noted that the cleaning of this data started by making a copy of each of the three dataframes that were imported - so, df became df_clean, image_predictions_df became image_predictions_df_clean, and tweet_likes_df became tweet_likes_df_clean using .copy(). I obtained the corresponding .csv, .tsv and .txt files by clicking on the links provided to us in the Project Details and Twitter API sections of the project.

First, in our initial 'twitter_archive_advanced.csv' file, which I imported as `df`.

- Remove retweeted tweets from df
    - Using df.info(), we saw that there were 181 rows that were retweets in our original dataframe. To eliminate them, I used .isnull() with the retweeted_status_id column, although using it for retweeted_status_user_id or retweeted_status_timestamp would have been just as effective.
- Doggo/Floofer/Pupper/Puppo Columns should be merged into one dog_stage column
    - First, I concatenated the doggo, floofer, pupper and puppo columns into one 'good_dog_stage' column. From there, I replaced the concatenated names with correct names - for example, NoneNonepupperNone was replaced with simply puuper.
- unneeded columns need to be dropped (in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp)
    - Now that we have removed rows that were retweets, we can remove the rows listed above by using .drop().
- Convert timestamp to DateTime
    - I used pd.to_datetime to convert timestamp to the proper data type.
- Missing name columns (can't clean)
    - We are unable to clean this because we were not provided a database with the missing names, although it should be noted that this is an issue that may need addressing in the future.
- All denominators need to be 10
    - I needed to isolate only denominators with a value of 10 to ensure consistency within the dataframe.
- Remove outliers from numerators
    - Likewise, I needed the numerators to avoid outliers, so I isolated rows with numerators greater than 20 and removed those rows.

Brad DeFauw
badefauw@gmail.com

Next, in the 'image_predictions_df' table.

- Dog names should be written with spaces instead of underscores
    - I used .replace() within the same line of code to make these changes.
- Drop unneeded columns from dataset (jpg_url, img_num)
    - Similar to 'df', .drop() was useful in eliminating the unnecessary columns from this dataset.

Finally, in the 'tweet_likes_df' table.

- rename id column so that tables can later be merged
    - This was necessary so that we could merge all three tables together at the end of this project. I used .rename() to change 'id' to 'tweet_id'

There were also two tidiness issues that were cleaned.

- Confidence interval should be written as a percent instead of a decimal in `image_predictions_df`
    - I multipled p1_conf, p2_conf and p3_conf by 100 to make our confidence intervals appear as percentages instead of decimals.
- Image_predictions and tweet_likes should be part of df
    - Here, I used .merge() twice, merging on 'tweet_id'.

That summarizes my wrangling efforts - gathering, assessing and cleaning - for this project.