WeRateDogs Data Wrangling Process

During this data wrangling process I gathered, assessed, and cleaned data to produce a dataset that I could then use to make illustrations and explore data trends. In this report I will discuss my wrangling efforts in more detail.

I used the following commands to import the libraries I required for the data wrangling:

- import pandas as pd
- import requests
- from bs4 import BeautifulSoup
- import json
- import tweepy
- import numpy as np
- import re

1. Gathering the Data

For this project I had to gather data from 3 different sources:

1. a csv file containing the WeRateDogs twitter archive.
2. a link to tsv file called image_predictions.tsv, hosted on Udacity's servers. This tsv file contained the resulting breed prediction based on images from the WeRateDogs twitter account, according to a neural network.
3. querying twitter's API using python's tweepy library, and thereby getting the favourite and retweet count of each
   tweet in the twitter archive csv file.

First I read the csv file into a pandas dataframe and called it t_archive. I removed all retweets from the data by removing all rows in which “retweeted_status_id” is not Null. I renamed this dataframe t_archive_new. Next I programatically downloaded the image_predictions.tsv file using python's requests library and the provided url. I then read this tsv file into a pandas dataframe, naming it image_pred.
Finally,I used Python's Tweepy library to query the Twitter API for each tweet's JSON data. To do this I used the tweet IDs found in t_archive_new. I used the "_json" property of the tweepy status object to get JSON serializable response data and could then extract retweet count and favorite count from this data. I stored the retweet count and favorite count, and corresponding tweet id for each tweet from t_archive_new in a text file using the json library.

2. Assessing the Data

I assessed the 3 resulting dataframes from the gathering step by using the following pandas assessment methods:

- .head() was used to visually inspect the dataframes
- .info() was used to see the datatypes for each column and number of entries in the dataframe
- .type() was also used to see the datatype for a column
- .value_counts() was used to the different values present in a column
- .describe() was used to show the spread of the data in a column containing integers
- .duplicated was used to see if there were any duplicates in the dataframe
- .str.islower() was used in the name column of t_archive_new, to identify any non-names in this column

Based on this visual and programmatic assessment, I decided to address the following quality and tidiness issues:

Quality Issues:
1. Timestamp in t_archive_new is in string format not in timestamp object format
2. Entries in dog stage columns in t_archive_new include "None" where it should be NaN.
3. Entries in dog name columns in t_archive_new include "None" where it should be NaN.
4. Error in data - rating denominator of 170 in t_archive_new. Rating denominators should not be greater than 10.
5. There are lowercase non-name entries in the name column of t_archive_new.
6. Columns "retweeted_status", "retweeted_status_user_id", and "retweeted_status_timestamp" in t_archive_new should   not be in dataset.
7. Dog stages in t_archive_new should be of categorical datatype
8. There are duplicate rows in the t_archive_new dataset.
9. Some text entries in t_archive_new have more than one rating, where the first rating is not applicable to the dog.
10. Tweet ids should be strings not ints.
11. There are some fractional numerators that are being captured incorrectly., with only the number after the decimal point taken as the rating numerator.

Tidiness Issues:
1. Dog Stage column does not exist in t_archive_new
2. The image_pred and counts_data should be part of the a new t_archive_clean dateset

3. Cleaning the Data

Before starting the cleaning process and making any changes, I made a copy of each dataset.

I changed the "None" entries in the "floofer", "doggo", "pupper", and "puppo" columns to NaN and used the pandas melt function to create one column named "dog_stage".
After making these changes I used the .head(), .info(), and .value_counts functions to inspect these changes.

I then changed the dog_stage entries to be of categorical type, using the .info() method to check that this change was successful.
I dropped the duplicates from the t_archive dataset and merged the image_pred and counts_data to the t_archive_new dataset, based on the tweet id entries. I used the .columns method to make sure that all the column were now present in the t_archive_new dataset.
I removed the unnecessary "id", "retweeted_status", "retweeted_status_user_id", and "retweeted_status_timestamp" columns and once again used the .columns method to check that they have been removed successfully.
The pd.to_datetime function was used next to change the date time entries from string to timestamp objects.
I then removed entries in t_archive_new where the denominator was bigger than 10.

For all entries where names were in lowercase, I saved the corresponding text entries. I used these text entries and general expressions to obtain the correct names. If I couldn't find a name in the text using this method, I replaced the lowercase word with "None". To test my code, I checked that the names were changed correctly using the tweet ids of these tweets. I then changed all "None" entries in the name column to NaN. I also changed the datatype of the retweets from int to string, since no numerical operations would be carried out on them.

Some of the ratings in the text had fractional numerators. These numerators were incorrectly represented in the “rating_numerator” column - only the number after the decimal point was shown. I identified these ratings using regex and fixed them to show the true fractional numerator. Finally, I verified the ratings for all text entries containing more than one fraction in the text. I used general expressions to do this and then changed the ratings where required.

I saved the new dataset with the quality and tidiness changes, to a csv file called twitter_archive_master.csv.