Udacity’s DAND:
Data wrangling project (Twitter’s WeRateDogs @dog_rates analysis)
Student: Rodolfo Yoshii

The below report intends to describe the data wrangling process followed in the Jupyter Notebook file: `wrangle_act.ipynb (`The intended audience for this document is internal)

A separate file (`act_report.pdf`) will contain the interpretation of the insights of this project. The intended audience for this secondary file is the public at large)

Python libraries imported:

```
import pandas as pd
import numpy as np
import os
import requests
import tweepy
from bs4 import BeautifulSoup
from PIL import Image
from io import BytesIO
from tweepy import OAuthHandler
import json
from timeit import default_timer as timer
from sqlalchemy import create_engine
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Data wrangling process was divided in 3 stages: gathering, assessing and cleaning (note that the assessing and cleaning process was done by each quality or tidiness issue was encountered).

Gathering stage:
The objective was to gather data from 3 different sources:

- A provided file: twitter-archive-enhanced.csv (downloaded manually) and imported using the .read_csv method
- A tsv file: image-predictions.tsv (downloaded programmatically) and imported using the os and requests libraries
- Creating a json file and save it as a txt file (tweet_json.txt), downloaded programmatically using the tweepy api

The data from the 3 sources were saved as pandas dataframes with the names df_clean, df2_clean and df3_clean, respectively. From this stage I proceeded to assess and clean programmatically.

Note that I understand there are more quality issues remaining, but for the sake of this project the scope is to identify 8 quality issues:

1) Erroneous data type in the "timestamp"
2) Incorrect data entry in the "name"
3) Incorrect data entry in the “rating_numerator” and "rating_denominator"
4) The values under column "source"
5) Data type of "source" should be categorical
6) Non-descriptive column names in image-predictions.tsv
7) Incorrect data entry, values for doggo, floofer, pupper and puppo contain "None"
8) Unnecessary data (retweets) in data

In addition, I identified 2 tidiness issues (same as above, it’s a known issue that there are more tidiness issues):

a) Values in column names instead of row values in a single column
b) Each type of observational unit forms a table (json.txt file)

The assessment and definition of these issues were documented with docstring in the jupyter notebook. These docstring also explain the coding required to programmatically correct the quality and tidiness issues. Finally, each element was validated with a line of code to test if the correction is in place.

A brief description of the wrangle work is provided below for each of the identified quality issues:

1) Erroneous data type in the "timestamp"

Corrected by using the to_datetime method

2) Incorrect data entry in the "name"

Corrected by removing the incorrect values for names such as None, the, a, an

3) Incorrect data entry in the “rating_numerator” and "rating_denominator"

Used a regex to extract the correct numerator and denominator from the tweet text, then used those values to replace the columns’ values, changed to correct data types

4) The values under column "source"

Used a replace method and a regex to extract the source names

5) Data type of "source" should be categorical

Used the astype method to change the datatype

6) Non-descriptive column names in image-predictions.tsv

Used the rename method to change the specific columns that were not descriptive

7) Incorrect data entry, values for doggo, floofer, pupper and puppo contain "None"

Used the replace method in a loop to change the incorrect data entered from “None” to a NaN

8) Unnecessary data (retweets) in data

Used the isnull method to identify and remove the data from retweets

About the tidiness issues:

a) Values in column names instead of row values in a single column

Used a replace method and then created a column name dog_stage to place the values of the dog stages, this way we can end up with a single column for all dog stages.

b) Each type of observational unit forms a table (json.txt file)

I identified that the table from the json.txt file belong to the same table containing tweet information. This way we no longer need to have 3 dataset, only 2 (one with tweet information, the other with prediction information)

Once the data cleaning stage was completed, I included a section on how the clean data can be stored in both a master csv file (`twitter_archive_master.csv`) and a sqlite database (tweetanalysis.db).

At the end of the document there is a section with 3 insights and visualization, each of them with descriptive docstring and explanations of the code required to produce the visualiations.