Introduction

Real-world data rarely come clean. Using Python and its libraries, I gathered data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling.

The dataset that I wrangle (and analyzing and visualizing) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. WeRateDogs has over 4 million followers and has received international media coverage.

The first step of data wrangling is Gathering data:

I gathered each of the three pieces of data as described below in a Jupyter Notebook titled wrangle_act.ipynb:

1.The WeRateDogs Twitter archive. I Download this file manually (twitter_archive_enhanced.csv)
2. The tweet image predictions, i.e., what breed of dog (or another object, animal, etc.) is present in each tweet according to a neural network. This file (image_predictions.tsv) is hosted on Udacity's servers and I downloaded programmatically using the Requests library and the following URL:
https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

3. I use the tweet_json.txt: This is the resulting data from twitter_api.py.
(Using the tweet IDs in the WeRateDogs Twitter archive, query the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file. Each tweet's JSON data should be written to its own line.)

The second step of data wrangling is Assessing data:

I can assess data for:
Quality: issues with content. Low quality data is also known as dirty data.
Tidiness: issues with structure that prevent easy analysis. Untidy data is also known as messy data. Tidy data requirements:
Each variable forms a column.
Each observation forms a row.
Each type of observational unit forms a table.

using two types of assessment:
Visual assessment.
Programmatic assessment.

I found 10 Quality Issues:

1.timestamp column has type int instead of DateTime type in twitter_archive.

2.tweet_id column has type int instead of String type in twitter_archive and image_predictions and tweet_data.

3.Wrong entries in the name column as 'a', 'an', 'the'… in twitter_archive.

4.Missing value name in twitter_archive recorded as 'None' string instead of NaN.

5.source column has some prefix and suffix in twitter_archive.

6.the rating_denominators are inconsistent in twitter_archive.

7.Unnecessary entries where p1_dog, p2_dog, and p3_dog are all "False" in image_predictions.

8. Incorrectly entered the name as 'O' in twitter_archive instead of O'Malley.

9. Tweets in twitter_archive some are retweets.

10. Columns pertaining to retweets and expanded URLs are unnecessary in twitter_archive.

And two Tidiness Issues:
1.Doggo, floofer, pupper, puppo are one variable spread in different columns in twitter_archive.

2. Data Frames are separated, but they contain the same observations.

The third step of data wrangling is Cleaning data:

I cleaned each issue in assessing step

1.convert timestamp column type to DateTime type, using pd.to_datetime().

2.convert tweet_id column type to String type, using DataFrame.astype(str).

4.convert 'a', 'an', 'the' in column name to ‘None’.

3.convert None in column name to NaN, using replace('None',np.nan).

5.remove prefix and suffix from source column, using .replace(,).

6.remove entries where rating_denominator does not equal 10.

7.remove entries where p1_dog, p2_dog, and p3_dog are all "False".

8.replace 'O' with "O'Malley" in twitter_archive.name[775].

9.remove entries that are retweets in twitter_archive.

10.remove columns ('retweeted_status_id','retweeted_status_user_id', 'retweeted_status_timestamp', 'expanded_urls') using DataFrame.drop()

Tidiness Issues:
1.create dog_stage column that contains the contents of columns doggo, floofer, pupper, puppo.

2.Joining the 3 data frames in one master data frame on the 'tweet_id' .