Gather, Assess, Clean WeRateDogs Data

Introduction

The purpose of this project is to practice and demonstrate data wrangling proficiency in Python by gathering, assessing, and cleaning a dataset. The dataset is a collection of tweets from WeRateDogs. The data was provided in several different formats, with varying degrees of tidiness and quality. The process of wrangling is outlined below.

Gather

The first step in the data wrangling process is gathering data. The data for this project was provided in three pieces. First, there was file on hand, twitter_archive_enhanced.csv. I read this file into a Pandas DataFrame using read_csv(). Next, a file was made available on Udacity's servers, image_predictions.tsv. This file was obtained using requests.get(), saved to a file, and read to DataFrame again using read_csv(). Finally, the project required gathering some extra information about the tweets directly from Twitter, using the Tweepy library. This information was gathered into a Python dictionary, written to a JSON file, and read into a Pandas DataFrame.

Assess

The next step in the process was to assess the data for tidiness and quality. I inspected each individual DataFrame using the info() function to check data types and count null values. I also inspected each DataFrame using the head() function as well as copies of each DataFrame saved to a csv file and viewed in Excel.

According to the rules of clean data at https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html, I identified three data tidiness issues. One, there were unnecessary columns and duplicated rows. Two, there were four columns related to one variable, dog stage. And three, all three DataFrames addressed the same "observational unit" of a tweet; they could be combined into one.

There were also many data quality issues, mostly – but not always – related to data types:

- tweet_id is an integer, but should really be treated as a string.
- timestamp is a string but should be a datetime object.
- The project specifications state that we do not need tweets with no images; tweets with no images can be dropped.
- Some of the dog names are not names but words like a, an, the, etc. Replace these words with a more useful string like 'None'.
- Retweets and favorite counts are counted as floats, but I don't understand what a partial favorite or retweet is; count them as ints.
- dog_class is a string but can be considered a categorical variable.
- img_num is a float, but can be considered a categorical variable.
- Rating numerators and denominators are considered ints, but because of their wildly inconsistent style it might be easier to make calculations from them if they are floats.
- p1, p2, and p3 are coded as strings, but they refer to the truth or falsity of predictions, so they can be coded as Booleans.

Clean

Before cleaning the data, I made copies of each DataFrame to work from.

Clean for Tidy Data

I then tidied up the data by dropping unnecessary columns, combining the four dog_stage columns into one, and joining the DataFrames together using the merge() function.

Clean for Quality Data

Finally, I cleaned up the quality issues. Using the astype() function, I adjusted the data types for each variable. I also used todatetime() function to use the timestamp variable as a datetime object, dropna() to remove rows with no images, and I used fillna() to replace null values of numbered variables with 0 or 0.0 and null strings with the word 'None'.

Conclusion

In the end, a clean(er), tidy(er) version of the dataset was saved to twitter_master_archive.csv. The cleaning work is not entirely complete; however there is much less unnecessary data, and each variable is recorded as the most appropriate type.