Data Wrangle Report

The code bundled with this report utilizes python with the following packages: Tweepy, Pandas, Numpy, Requests, IO, matplotlib. Its purpose is gather data from 3 different sources, then condition the data into a clean dataframe to use for analysis. The first source of data, dog archive is already on hand. We load this via pandas read_csv() function into the workspace. The second source of data is a dataframe, dog predictions, stored on Udacity servers. We use the requests package to collect the data from the server. We then write it to a csv file to create our own copy within the project. Finally, we need to scrape a few pieces of information from twitter. The tweepy package is used to create an API from twitter. We pull the needed information from the API and write it to a dataframe within the project. Now that we have all the dataframes setup, we assess them for quality and tidiness issues. We identify the following issues:

- Dog predictions should be in the dog archive file
- Dog archive missing retweets and favorites
- Some entries are replies to other tweets, irrelevant
- Not all entries in dog archive have pictures
- Some entries are retweets
- Some images don't have dogs
- Column time stamp should be a date object
- Columns doggo, puppo, pupper, floofer should be categorical
- All dog types need to be capitalized
- Null values in doggo, puppo, pupper, floofer stored as objects

Before we address these issues, we create copies of all 3 dataframes. Then, we perform a right join between dog archive and dog predictions. This combines both dataframes, while filtering out tweets without an image. We perform another left join with dog archive and the twitter API dataframe to append retweets and favorites. Next, we filter out all tweets that are replies to other tweets by removing non-Null values in the in_reply_to_status_id column. We also remove non-Null values in the retweeted_status_id to filter out retweets. To fix the pictures without dogs, we remove all observations that don't have a true prediction in at least 1 of these 3 columns: p1_dog, p2_dog, p3_dog. Next, we convert the timestamp column to a datetime object via pandas to_datetime method. This allows us to pull data from it to create a graph later in the project. We capitalize all the dog breeds in columns p1, p2, and p3 with the str.capitalize method. We convert the 'None' values into Null values in the doggo, puppo, pupper, and floofer columns. Finally, we cast those 4 columns as categorical variables using the astype method. This was done after removing the 'None' values as they would become a separate category otherwise. We write this cleaned dataframe to a master csv file to subsequently be analyzed.