WeRateDogs Data Cleaning Project

Motivation
The goal of this project is to examine a dataset of tweets posted by the WeRateDogs account. The tweets include pictures of dogs submitted by other twitter users. WeRateDogs then rates the dogs and shares the original picture.

Data Sources
The data came from three sources: an archive of WeRateDogs tweets, Twitter API data containing retweet and favorite counts for the tweets, and a set of results from an image prediction neural network that produced predictions of dog breed or other image subjects for the tweet images.

Gathering
The tweet archive (*tw_arch)* was imported from a local file. The Twitter API data (*trf)* was also imported from a local file, since Twitter API access was not available to me at the time. The image prediction data (*img_pred)* was downloaded programmatically. Before assessing and cleaning each dataframe, I copied each dataframe, leaving the dataframes with the suffix 'prelim' untouched.

Assessing
I conducted visual and programmatic assessments of the data sources to identify data quality and tidiness issues.

Cleaning
- I converted 'tweet_id' to string format in each dataframe
- I removed '+0000' from each 'timestamp' value
- I converted 'timestamp' in tw_arch to datetime format
- I manually extracted rating numerator from 'text'
- I dropped all the rows without any dog image predictions
- I combined the rating numerator and denominator into a single float column
- I dropped rows with all null values
- I combined the columns 'doggo', 'floofer', 'pupper', and 'puppo' into a single column and filled rows without any results in the original four columns with 'none', and stored results with multiple stages as comma separated results
- I joined the three dataframes using an inner join, which resulted in a final cleaned dataframe with only rows that had values in all columns in the original dataframes

Testing
After running the code to address the data quality and tidiness issues and to combine the three datasets, I visually assessed the new dataframe, *df_clean*, and used the Pandas .info method to assess each column and ensure there were no null values.

Storing
After testing my data cleaning code, I saved the finalized dataframe to a .csv file entitled *twitter_archive_master.csv*. That file is included in the submitted .zip file.