Wrangling report

This report describes my wrangling process for the WeRateDogs Twitter account data. My process was divided in to 3 main phases; gathering, analyzing and cleaning. The process was not linear and I had to go back and forth between the different phases, so this report will describe the process in term of a timeline of my progress and not isolating the different phases.

1. After setting up the notebook and importing the required libraries, I started with gathering the data form the 3 different sources
    a. For the Twitter Archive it was a straight forward import from the provided csv file using the built-in pandas's method read_csv
    b. For the second dataset, it had to be downloaded first, for that I used the requests library to download the file and save it locally and then I imported it like the file above, the only difference was that this is a tsv file so I had to change the default sep parameter in the read_csv function
    c. For the last dataset, this was the one with some issues as I missed reading the Twitter API section, after figuring out the process of dealing with the Twitter APIs and creating a developer account, I had to wait for over a day to get an approval from Twitter and proceed with this part. After getting the approval, I created a Twitter app to get the key to authenticate with Twitter. As for using Tweepy library to get the data using the tweet_id column for the Twitter Archive table already created was easy. But in the analysis part I found that over 500 tweets were missing data, after analyzing the Exceptions fired I found that they are of 2 different types. One was an issue with the tweet_ids which is a data quality issue that I cleaned later and the other one was due to the exceeding Twitter's API rate limit, which I solved with the help of this post https://stackoverflow.com/questions/38775997/getting-this-error-when-using-tweepy. Finally, I stored the JSON data in a text file, read it into a dictionary and imported it into a DataFrame.
2. For the assessing part I used a few built-in functions to explore the data and came up with the 10 quality issues and 2 tidiness issues that I tackled in the cleaning part. I listed them by table and type at the bottom.
3. For the cleaning part I created copies for the DataFrames to work on them, usually I would start with the missing data, but in this case the missing data issue was more of removing data than getting missing data.
    a. In the Tidiness part I joined the tables together and combined the dog type columns
    b. In the quality part I used a combination of removals, changing data types, and fixing values, all the changes were done on the df DataFarme that I created in the tidiness section. Some of the quality issues were tackled together as is it was more efficient than fixing them separately.
    c. After fixing an issue or a group of issues some tests were made to confirm that it was solved
4. As a last step, the DataFrame was saved to a CSV file locally using the built-in to_csv function.

5. There is a lot more of things to do to clean up this data, like further analyzing the ratings or tidying up the dog breed columns for example.