To identify salient features of the rating dogs twitter, I have gathered, assessed, cleaned, and stored the data. To begin, for this project, I have used three different methods for gathering the appropriate data. I imported pandas to help work with the dataframes. I then uploaded a csv file using ‘read_csv’. I then used the requests library to get another file that was located on the network, making sure to make the appropriate accommodations as it was a tsv file. For the twitter API, I followed the instructions with produced a json file that was saved as a txt file and uploaded with read_json. I checked all three data frames with .head() to make sure they were all uploaded correctly.

Then, I assessed the data frames both programmatically and visually. Visually, I saw that these three data frames needed to be merged into one. I also saw that the columns labeled after dog stages included a lot of non-information, so they would be better when stored as one column. Programmatically, I saw with .info() that the json file included a lot of null values in specific columns that needed to be dropped. Also using .info, I saw that the timestamp was not typed as datetime but as an integer and needed to be changed. Also, the reply id and retweet id were both produced as floats instead of integers or objects. Using .value_counts(), I noticed that dog types and dog names both had issues, dog types with consistency and dog names with incorrect responses such as ‘a’ and ‘the’. When I looked more into specific ratings of the dogs, such as 0 and 1, I saw there were actually no dogs listed, and as such I felt we could lose that data from the set without affecting the outcome.

After listing all the different issues I had found, I cleaned the data using a variety of tools. For cleaning the timestamp, I removed the last 5 characters and changed the type to datetime. For the erroneous columns in the json file, I created a subset of the data that I wanted and carried it forward. For the source column and the name column I used the replace function to correct the values. To correct the dog stage columns, I used apply, drop, and replace to create a singular column with the dog stage in it. For the image file, if none of the predictions had a dog in it, I dropped those rows from the data. After cleaning, I then merged the three together and stored the file.