I firstly gathered the data as advised, manually and programmatically. Unfortunately, I was not able to get permission on time by twitter as their team had a lot of follow up questions and it was taking a lot of time. I therefore accessed the Twitter data by reading the tweet_json.txt file line by line into a new dataframe.

After gathering the data, I accessed it both visually and programmatically. I came up with 2 tidiness issues and 9 quality issues:

Tidiness:

twitter_archive_enchanced:

1. dog stage values form columns instead of having one column with these values as variables
2. twitter_archive_enchanced dataset should be merged with json_file datframe

Quality:

twitter_archive_enchanced:

1. timestamp column's datatype is object
2. replies and retweets have many null values which is natural, however we will most definitely not need the non-null rows
3. rating is expressed in two columns (numerator, denominator) instead of only one rating
4. rating numerator values' range is between 0 and 1776
5. similarly rating numerator values' range is between 0 and
6. inaccurate dog names

image prediciton:

7. column names are not descriptive
8. inconsistency in dog races, values have both lower and upper case values
9. img_url has 66 duplicates

I merged the twitter_archive_enchanced dataframe with the json_file dataframe, I created one column for the dog stages, I filtered the rows of the retweets and replies out of the main dataframe, and then checked for denominators not equal to 10. I visually checked the entries in the dataframe as well as the .csv file and found out that most of the denominators not equal to 10 are aggregates for more than one dogs. Because the plan was to create a new column named rating (rating numerator/rating denominator) these entries were good to stay.

Then I needed to extract the correct values of the rows with denominators not equal to 10. The way I thought was to use regular expressions and search for matches of any numerical value including decimal ones, followed by the valid denominator of 10 (/10). In order for the aggregates not to be left out of the matches I created a list with those rows and then iterated all the rows not equal to them, saved them in a new list and then applied the regular expressions there. After this process the datatype was changed to object so I had to convert it to float (there were decimal values too), and lastly I created the new column “rating” by dividing the numerator column by the denominator one. I also filled the rows with no ratings with the mean rating value.

I then converted the datatype of “timestamp” to datetime ad tried to extract as many correct names as possible, as there were inaccuracies in the name column. I created a function with regular expression which I applied to the dataset and saved the new correct names to a new column “names” and replaced the rows where no name was found with “None”

After that, I change the columns of the image_predictions dataset to more descriptive ones, capitalized all values of the prediction columns in order to be consistent, I dropped the duplicated img_url rows and lastly I dropped all the columns that were not needed from the master dataset.

In the (almost final) master dataframe I double checked for duplicated values and found some in the expanded_urls as well as some null values. After making sure that the duplicated values were dead links and there were no images of those tweets in the prediction dataframe I filtered them out of the dataset along with the null values.

Lastly, I stored the two final dataframes in .csv files and analysed the wrangled data.