Gather Data:

First, manually downloaded the "twitter_archive_enhanced.csv" file for this WeRateDogs Twitter account . Then used the Request library to download "image_prediction.tsv" file following a given URL. Next, I have set up my Twitter developer account and obtained consumer key and secret, as well as access token and secret. Using the tweet IDs in above Twitter archive, I have queried the Twitter API for each tweet's JSON data utilizing Python's Tweepy library and stored each tweet's entire set of JSON data in a file called "tweet_json.txt", separated by newline character. I have then read this .txt file line by line into a pandas DataFrame named "df_tweet_json" with tweet ID, retweet count and favorite count.

Assess Data:

Loaded the above three DataFrames to be "df_twitter_archive", "df_image_predictions" and " df_tweet_json", individually. I have then assessed them both visually and programmatically for quality and tidiness issues. Managed to detect and document 8 quality issues with table "df_twitter_archive", 1 quality issue with table "df_image_predictions" and 4 tidiness issues across three tables in my "wrangle_act.ipynb" Jupyter Notebook.

Clean Data:

The first problem I want to address is to remove all the retweets present in this Twitter archive table, keep only the original tweets. I have then changed the data type of column "tweet_id" on both df_twitter_archive" and "df_image_predictions" tables to be string type instead of integer. Next, change the data type of column "timestamp" at archive table to be timestamp object, in order to benefit later analysis.

Second, fix some important tidiness issues before working on more quality improvement. I dropped several less relevant columns from this archive table. Then merged it with the important records of retweet counts and favorite counts at 'df_tweet_json" table based on shared "tweet_id" to obtain the initial " twitter_archive_master" DataFrame. Notice now the number of tweet decreased from 2356 records to 2168, but with much better structure.

Then I have worked on the quality issues of this " twitter_archive_master" table, including the missing dog names and inaccurate ratings. Both problems can be addressed by identifying the patterns of dog names and ratings from text contents, then properly extract them using regular expressions from the column "text" (Details was documented in "wrangle_act.ipynb" ). Note: During the investigation, I have noticed that many tweets are warning about receiving pictures other than dogs, those records have been removed to improve the holistic quality. Some tweets simply don't contain dog names, I have therefore filled the "name" column of those records to

be "Not given". A small portion of ratings are based on larger numerical scales because of the presence of multiple dogs in those tweets.

Similarly, I have extracted the dog stage from the text content using a self-defined method and created an individual column "stage" to store the information. Again, many tweets simply don't offer dog stages, I have therefore filled the "stage" column of those records to be "not identified".

Finally, most of the quality issues have been resolved. Since we only want original tweets with pictures, I have merged the above "twitter_archive_master" table with only relevant column of the "df_image_predictions" table based on shared "tweet_id". Rename the p1 column to be " predicted_dog_breed" to convey clearer message. Now this final "twitter_archive_master" DataFrame have 1891 records with tidy structure for later analysis. I have then stored this clean table in a CSV file name " "twitter_archive_master.csv".