Gather:

I gather data frame from three different sources. First, I downloaded the twitter_archive_enhanced.csv from Udacity. Then I downed the image_predictions.tsv from Udacity. Last, I used Python's Tweepy library to store each tweet's entire set of JSON data in tweet_json.txt.

Assess:

Firstly I loaded these tables and used df.sample(5) to randomly check the columns and rows to see whether there are some mistakes in the table. Then I used df.info() and df.describe() to programmatically check whether there are some mistakes in the table.

Both tidiness and quality issues were found.

Tidiness:

1.tweet_json table has too many columns. 2.twitter table has too many columns. 3.Three tables have different number of rows. 4.twitter_json and twitter table should be combined to one table.
Quality:

tweet_json: 1.Wrong data type(Created_at). 2.Column with missing values(too many to label here). 3.favourite_count has some outliers(max is 150944 while 75% of the counts are lower than 8960). 4.retweet_count has some outliers(max is 74577 while 75% is lower than 3010).
twitter: 1.Wrong data type(Created_at). 2.Column with missing values(in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, expanded_urls). 3.rating_numerator has outliers(this number should be 10). 4.rating_denominator has outliers(max is 1776 while 75% is lower than 12).
image: column p1,p2 and p3 have different format of elements(some dog breeds are capitalized while others are not).

Clean:

Tidiness:

Tweet_json and twitter tables were copied as tweet_clean and twitter_clean. I used df.drop() to delete irrelated columns in tweet_json and twitter data frame. These three tables were combined as one using df.merge(). Tweet_id is the column I used to join these tables.

Quality:

I used df.astype() to correct all the wrong data type in the newly combined table. Df.dropna() was used to delete the rows with missing values. To remove outliers in the table, I introduced stats from spicy and put a filter on the data frame such that we select all rows where the values are within 1.5 standard deviations from the mean. Finally str.lower() was used to uncapitalized all the words in the p1, p2 and p3 columns. The cleaned data frame was saved as a .csv file.