ACT REPORT

The Act Report is the summary of the project from the data wrangling process. I gathered three dataset and for the purposed of this project. The dataset include:

- Twitter_archive_enhanced.csv(tweetarchive): this dataset was provided for the project and had over 2357 tweets when downloaded.
- Image_prediction.tsv(imgpred): This dataset was programmatically downloaded with over 2075 predictions of dog breeds.
- Tweet_json_text(tweety): I scrapped the twitter API using python tweepy's library and has 2327 tweets.

While access the datasets, the quality issues in each dataset, and the tidiness issues were discovered and cleaned using some python pandas to get them clean using the three steps cleaning method which are to define, code and test.

Also, the three datasets were merged and called twitter_archive_master.csv. Some visuals were gotten after loading into the pandas dataframe.

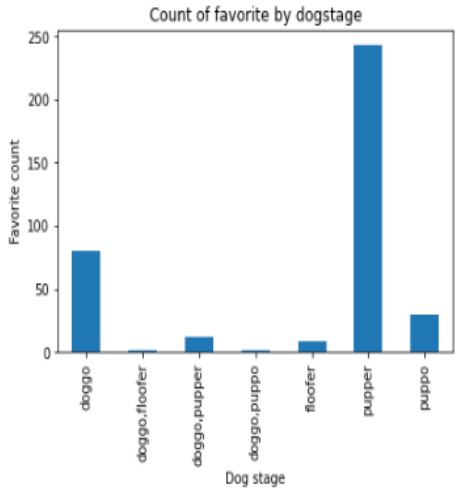
```
In [99]: Data.describe()
```

Out[99]:

	tweet_id	rating_numerator	rating_denominator	favorite_count	followers_count	friends_count
count	2.356000e+03	2356.000000	2356.000000	2327.000000	2.327000e+03	2327.0
mean	7.427716e+17	13.126486	10.455433	7021.832402	9.364621e+06	21.0
std	6.856705e+16	45.876648	6.745237	10910.459118	1.985462e+02	0.0
min	6.660209e+17	0.000000	0.000000	0.000000	9.364488e+06	21.0
25%	6.783989e+17	10.000000	10.000000	1218.500000	9.364492e+06	21.0
50%	7.196279e+17	11.000000	10.000000	3035.000000	9.364517e+06	21.0
75%	7.993373e+17	12.000000	10.000000	8561.500000	9.364949e+06	21.0
max	8.924206e+17	1776.000000	170.000000	144058.000000	9.364973e+06	21.0

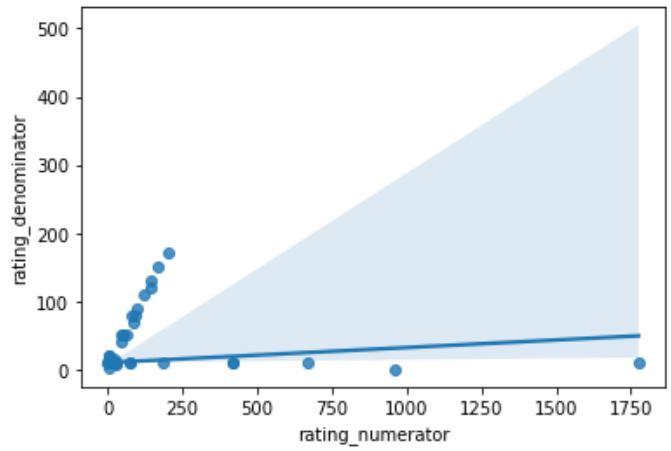
The dataset above shows the descriptive statistics for numerical dataset.

```
In [105]: Data.groupby('dogstage')['favorite_count'].count().plot(kind = 'bar')
          plt.title('Count of favorite by dogstage')
          plt.xlabel('Dog stage')
          plt.ylabel('Favorite count')

Out[105]: Text(0, 0.5, 'Favorite count')
```



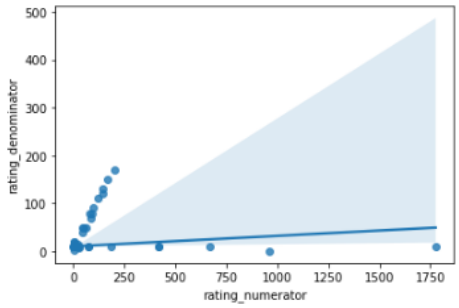
The bar charts above show most rated dog according to favorite count. The most rated dog is pupper, followed by doggo then puppo.

```
In [81]: sns.regplot(x=Data['rating_numerator'],y=Data['rating_denominator'])

Out[81]: <AxesSubplot:xlabel='rating_numerator', ylabel='rating_denominator'>
```



The diagram above here shows a positive linear relationship between the rated numerator and rated denominator.

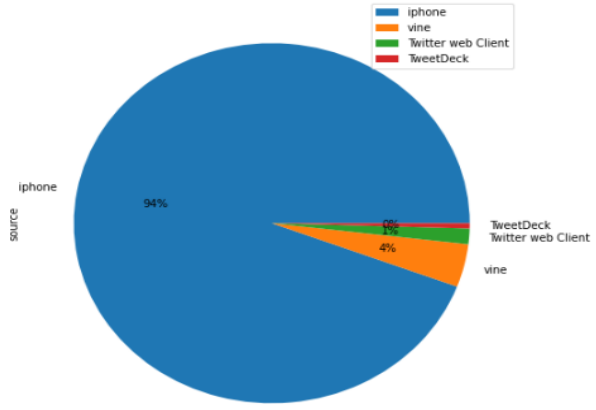
```
In [108]: sns.regplot(x=Data['rating_numerator'],y=Data['rating_denominator'])
Out[108]: <AxesSubplot:xlabel='rating_numerator', ylabel='rating_denominator'>
```



The above diagram shows a positive linear relationship between the rated numerator and rated denominator. It also shows the outlier.

Finally, I also check which of the source was mostly used.

```
In [109]: Data.source.unique()
Out[109]: array(['<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>',
                 '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>',
                 '<a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>',
                 '<a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>'],
                dtype=object)

In [110]: Data.source.value_counts()
Out[110]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>     2221
          <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>                          91
          <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>                       33
          <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>      11
          Name: source, dtype: int64

In [111]: def pie_plot(x):
              have = ['iphone', 'vine', 'Twitter web Client', 'TweetDeck']
              Data[x].value_counts().plot.pie(labels=have,autopct='%1.0f%%', figsize=(8,8));

In [112]: pie_plot('source')
          plt.legend();
```



Activate Wi
Go to Settings t

From the pie chart above, the iPhone is the highest source with 94%, followed by those using vine