Report: act_report

Project: Wrangling of We Rate Dogs Twitter Data

Table of Contents

Introduction to Wrangling

WeRateDogs is a twitter account that posts dogs photos with a rating of the dog. The data used in this project is a download of the tweets posted by WeRateDogs account. The data contains information about the dog e.g. a photo of the dog, name, breed or 'age group' and rating of the dog. Secondary data also obtained from the data are retweet counts, favorite count. The goal of this report is to gather all the data partaining the WeRateDogs account (there are three different datasets), assesses the data, noting all the issues persent, cleaning the noted issues and perform vaious analysis.

The following are some of the research questions that are to be answered:

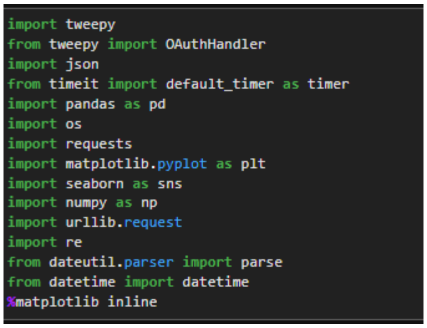
- What device is the most used to for tweeting the content on WeRateDogs account?
- What is the relationship between favorite counts and retweet counts?
- What is the relationship between the numerator rating and the favorite counts?

Data Gathering

Three datasets were provided for this project:

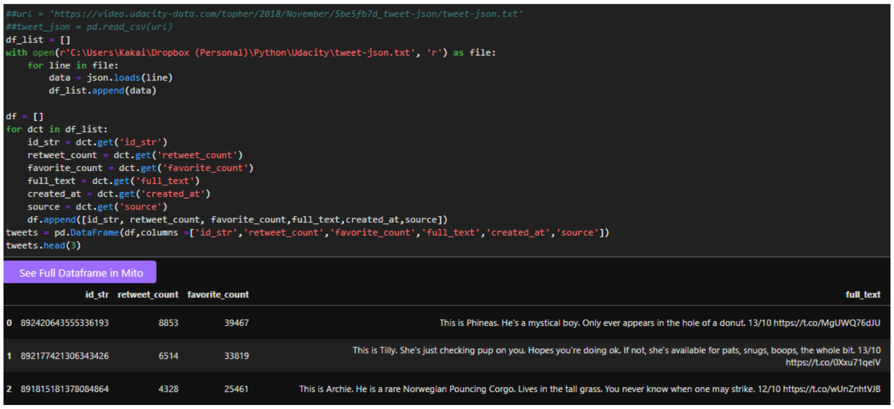
- Json file containing raw tweets data extracted using tweepy
- tweet image predictions file containing predictions of all the dog images through a machine learning model
- WeRateDogs Twitter archive file of partially extracted tweets data

First the following packages were import to aid with the data wrangling and visualization:



```python
import tweepy
from tweepy import OAuthHandler
import json
from timeit import default_timer as timer
import pandas as pd
import os
import requests
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import urllib.request
import re
from dateutil.parser import parse
from datetime import datetime
%matplotlib inline
```

1. JSON File >

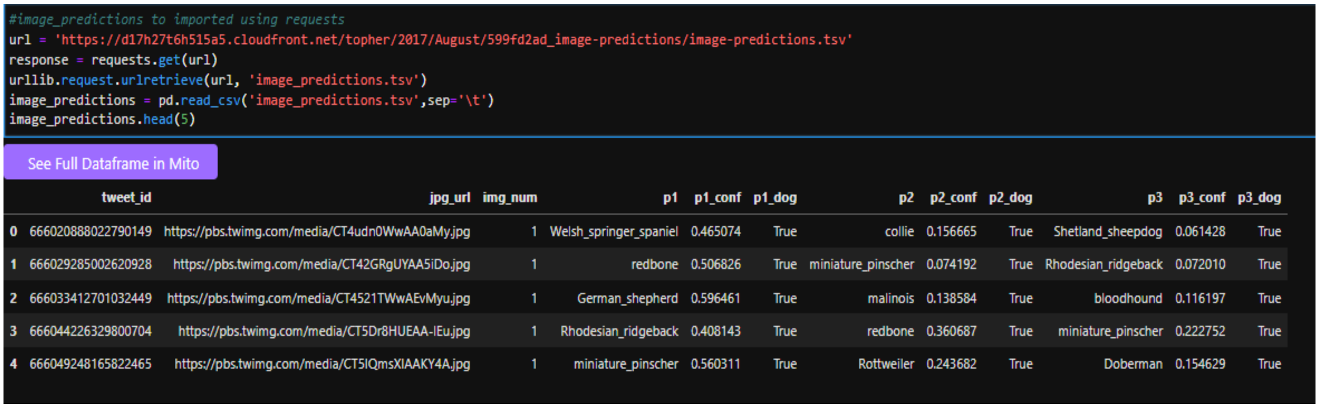


```python
##url = 'https://video.udacity-data.com/topher/2018/November/5be5fb7d_tweet-json/tweet-json.txt'
##tweet_json = pd.read_csv(url)
df_list = []
with open(r'C:\Users\Kakai\Dropbox (Personal)\Python\Udacity\tweet-json.txt', 'r') as file:
    for line in file:
        data = json.loads(line)
        df_list.append(data)

df = []
for dct in df_list:
    id_str = dct.get('id_str')
    retweet_count = dct.get('retweet_count')
    favorite_count = dct.get('favorite_count')
    full_text = dct.get('full_text')
    created_at = dct.get('created_at')
    source = dct.get('source')
    df.append([id_str, retweet_count, favorite_count,full_text,created_at,source])
tweets = pd.DataFrame(df,columns =['id_str','retweet_count','favorite_count','full_text','created_at','source'])
tweets.head(3)
```

See Full Dataframe in Mito

	id_str	retweet_count	favorite_count	full_text
0	892420643555336193	8853	39467	This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76dJU
1	892177421306343426	6514	33819	This is Tilly. She's just checking pup on you. Hopes you're doing ok. If not, she's available for pats, snugs, boops, the whole bit. 13/10 https://t.co/0Xxu71qeIV
2	891815181378084864	4328	25461	This is Archie. He is a rare Norwegian Pouncing Corgo. Lives in the tall grass. You never know when one may strike. 12/10 https://t.co/wUnZnhtVJB

2. Tweet Image Predictions >

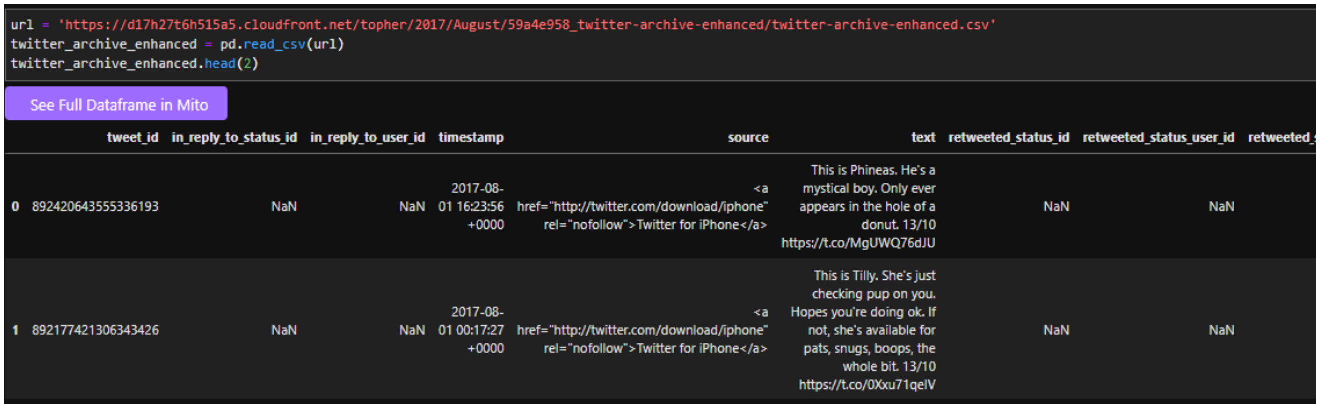


```python
#image_predictions to imported using requests
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv'
response = requests.get(url)
urllib.request.urlretrieve(url, 'image_predictions.tsv')
image_predictions = pd.read_csv('image_predictions.tsv',sep='\t')
image_predictions.head(5)
```

See Full Dataframe in Mito

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True	collie	0.156665	True	Shetland_sheepdog	0.061428	True
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826	True	miniature_pinscher	0.074192	True	Rhodesian_ridgeback	0.072010	True
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.596461	True	malinois	0.138584	True	bloodhound	0.116197	True
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.408143	True	redbone	0.360687	True	miniature_pinscher	0.222752	True
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.560311	True	Rottweiler	0.243682	True	Doberman	0.154629	True

3. WeRateDogs Twitter Archive File >



```python
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv'
twitter_archive_enhanced = pd.read_csv(url)
twitter_archive_enhanced.head(2)
```

See Full Dataframe in Mito

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweeted_status_user_id	retweeted_
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76dJU	NaN	NaN	
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Tilly. She's just checking pup on you. Hopes you're doing ok. If not, she's available for pats, snugs, boops, the whole bit. 13/10 https://t.co/0Xxu71qeIV	NaN	NaN	

Data Accessement

After accessing the data both visually and programatically a number of data quality and tidiness issues were noticed and recorded as follows:

Quality issues

`twitter_archive` `json` files

- Some numerator values were wrongly extracted as per the text field e.g tweet with id #680494726643068929 was extracted as 26/10 yet it should be 11.26/10
- Timestamp is not in the correct format, to be converted to a timedate format
- Source column is not clean, contains some html left over special charaters
- Tweet id extracted from json file has data type object and not integer
- in_reply_to_user_id/in_reply_to_status_id columns should have int as their data types and not float
- Some denominator and numerator values need to be transformed to be in the same format as other tweets e.g #677716515794329600 144/120 which should be 12/10
- rating numerator column has outliers, it has values values as big as 1776
- rating denominator column has outliers, it should a 10 across all tweets, has values both less than and more than 10

Tidiness issues

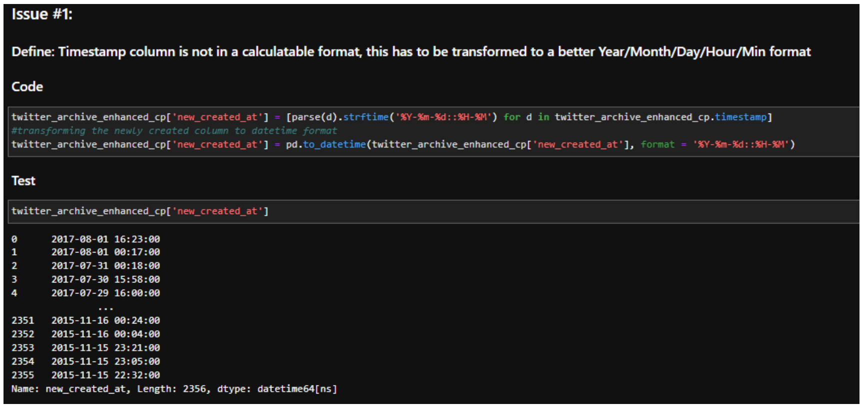
`twitter_archive` file

- Source column can be split into two different columns with device and url column being created
- Retweeted tweets are part of the main dataframe, these needs dropping

Data Cleaning

All the captured issues were cleaned by first redefining the issues, coding and lastly testing to check if the issues had been fixed. Below are three examples of the issues noted above having been fixed:

- Issue One: Format of the timestap column was not clean enough to be used for analysis, it had a string format instead of datetime:



Issue #1:

Define: Timestamp column is not in a calculatable format, this has to be transformed to a better Year/Month/Day/Hour/Min format

Code

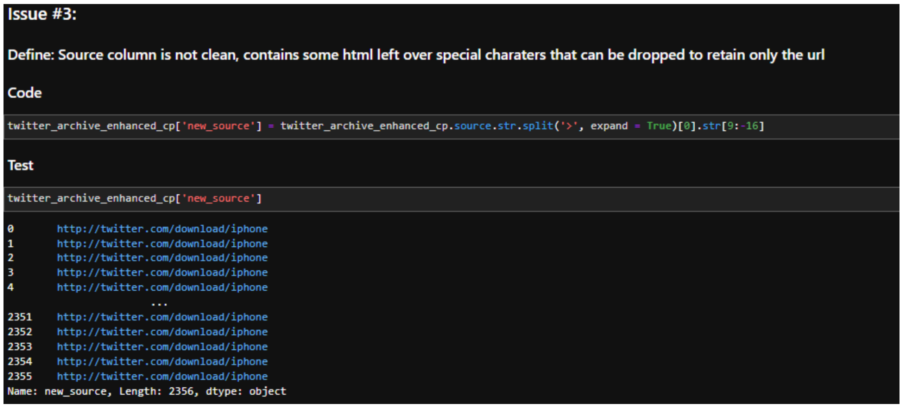
```
twitter_archive_enhanced_cp['new_created_at'] = [parse(d).strftime('%Y-%m-%d::%H-%M') for d in twitter_archive_enhanced_cp.timestamp]
#transforming the newly created column to datetime format
twitter_archive_enhanced_cp['new_created_at'] = pd.to_datetime(twitter_archive_enhanced_cp['new_created_at'], format = '%Y-%m-%d::%H-%M')
```

Test

```
twitter_archive_enhanced_cp['new_created_at']
```

```
0       2017-08-01 16:23:00
1       2017-08-01 00:17:00
2       2017-07-31 00:18:00
3       2017-07-30 15:58:00
4       2017-07-29 16:00:00
                ...
2351    2015-11-16 00:24:00
2352    2015-11-16 00:04:00
2353    2015-11-15 23:21:00
2354    2015-11-15 23:05:00
2355    2015-11-15 22:32:00
Name: new_created_at, Length: 2356, dtype: datetime64[ns]
```

- Issue Three: Source column was split into device and url columns as it was not tidy enough:



Issue #3:

Define: Source column is not clean, contains some html left over special charaters that can be dropped to retain only the url

Code

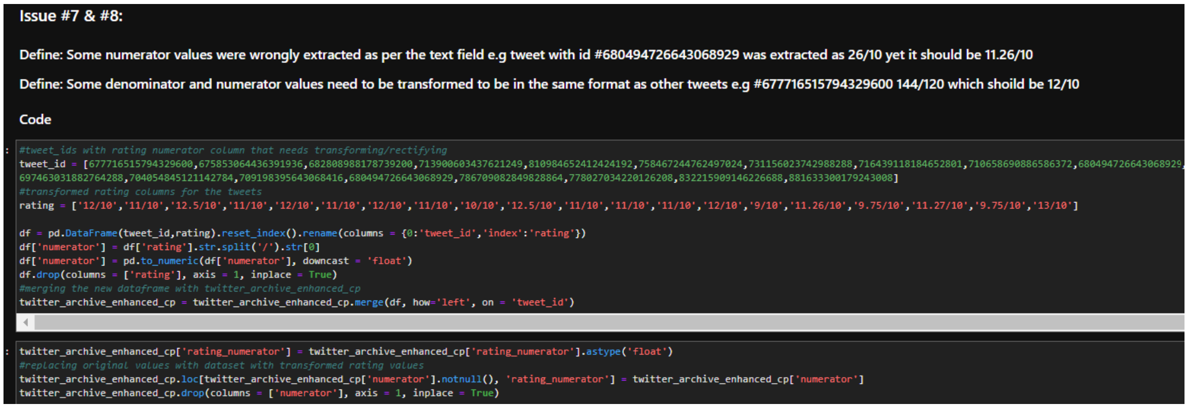
```
twitter_archive_enhanced_cp['new_source'] = twitter_archive_enhanced_cp.source.str.split('>', expand = True)[0].str[9:-16]
```

Test

```
twitter_archive_enhanced_cp['new_source']
```

```
0       http://twitter.com/download/iphone
1       http://twitter.com/download/iphone
2       http://twitter.com/download/iphone
3       http://twitter.com/download/iphone
4       http://twitter.com/download/iphone
                       ...
2351    http://twitter.com/download/iphone
2352    http://twitter.com/download/iphone
2353    http://twitter.com/download/iphone
2354    http://twitter.com/download/iphone
2355    http://twitter.com/download/iphone
Name: new_source, Length: 2356, dtype: object
```

- Issue Seven and Eight: Here some of the values in rating_numerator column were wrongly extracted while some need transformation to fit the scale:



Issue #7 & #8:

Define: Some numerator values were wrongly extracted as per the text field e.g tweet with id #680494726643068929 was extracted as 26/10 yet it should be 11.26/10

Define: Some denominator and numerator values need to be transformed to be in the same format as other tweets e.g #677716515794329600 144/120 which shoild be 12/10

Code

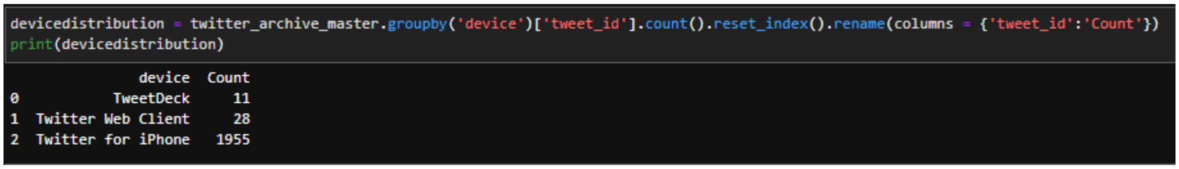
```
#tweet_ids with rating numerator column that needs transforming/rectifying
tweet_id = [677716515794329600,675853064436391936,682808988178739200,713900603437621249,810984652412424192,758467244762497024,731156023742988288,716439118184652801,710658690886586372,680494726643068929,
697463031882764288,704054845121142784,709198395643068416,680494726643068929,786709082849828864,778027034220126208,832215909146226688,881633300179243008]
#transformed rating columns for the tweets
rating = ['12/10','11/10','12.5/10','11/10','12/10','11/10','12/10','11/10','10/10','12.5/10','11/10','11/10','11/10','12/10','9/10','11.26/10','9.75/10','11.27/10','9.75/10','13/10']

df = pd.DataFrame(tweet_id,rating).reset_index().rename(columns = {0:'tweet_id','index':'rating'})
df['numerator'] = df['rating'].str.split('/').str[0]
df['numerator'] = pd.to_numeric(df['numerator'], downcast = 'float')
df.drop(columns = ['rating'], axis = 1, inplace = True)
#merging the new dataframe with twitter_archive_enhanced_cp
twitter_archive_enhanced_cp = twitter_archive_enhanced_cp.merge(df, how='left', on = 'tweet_id')
```

```
twitter_archive_enhanced_cp['rating_numerator'] = twitter_archive_enhanced_cp['rating_numerator'].astype('float')
#replacing original values with dataset with transformed rating values
twitter_archive_enhanced_cp.loc[twitter_archive_enhanced_cp['numerator'].notnull(), 'rating_numerator'] = twitter_archive_enhanced_cp['numerator']
twitter_archive_enhanced_cp.drop(columns = ['numerator'], axis = 1, inplace = True)
```

Analyzing & Visualizing Data

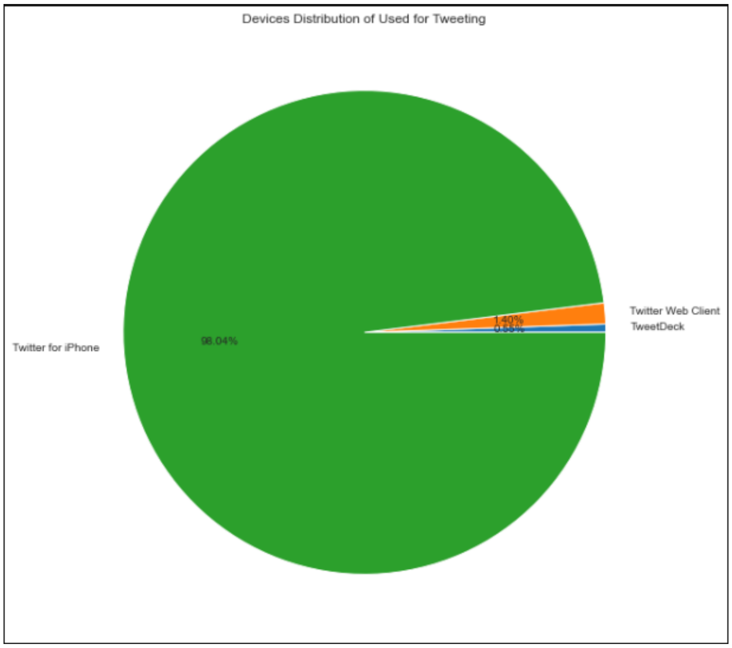
Research Questions

- What device is the most used to for tweeting the content on WeRateDogs account?

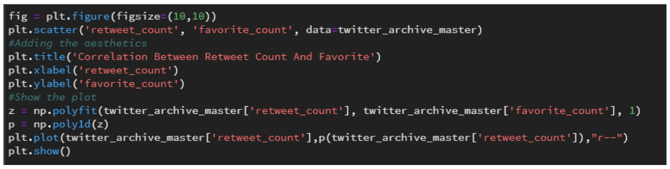


```
devicedistribution = twitter_archive_master.groupby('device')['tweet_id'].count().reset_index().rename(columns = {'tweet_id':'Count'})
print(devicedistribution)
```

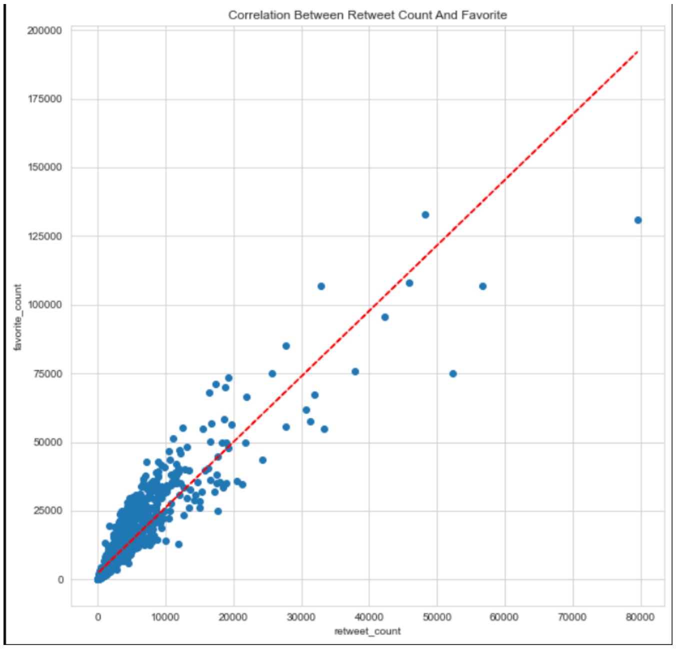
```
               device  Count
0           TweetDeck     11
1  Twitter Web Client     28
2  Twitter for iPhone   1955
```



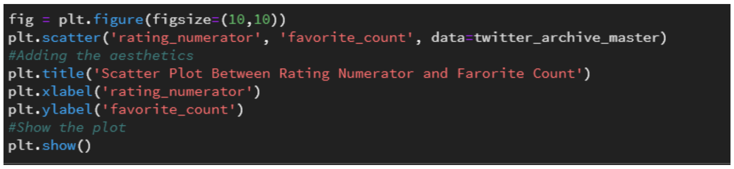
* What is the relationship between favorite counts and retweet counts?



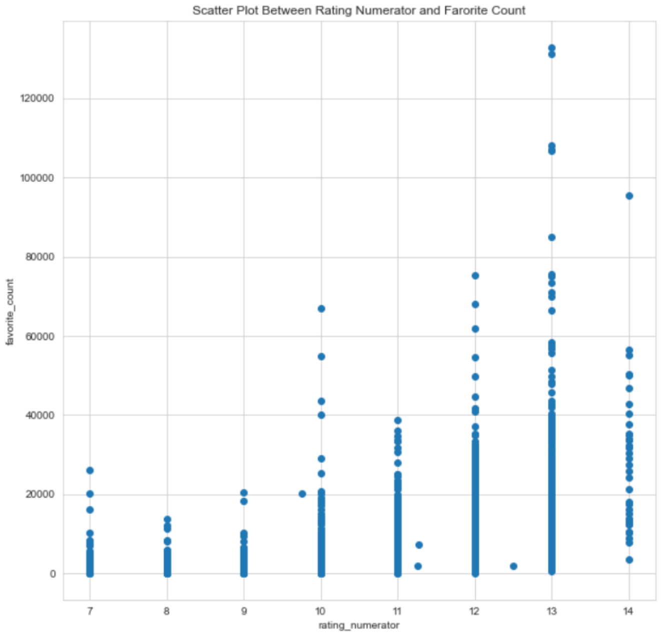
```
fig = plt.figure(figsize=(10,10))
plt.scatter('retweet_count', 'favorite_count', data=twitter_archive_master)
#Adding the aesthetics
plt.title('Correlation Between Retweet Count And Favorite')
plt.xlabel('retweet_count')
plt.ylabel('favorite_count')
#Show the plot
z = np.polyfit(twitter_archive_master['retweet_count'], twitter_archive_master['favorite_count'], 1)
p = np.poly1d(z)
plt.plot(twitter_archive_master['retweet_count'],p(twitter_archive_master['retweet_count']),"r--")
plt.show()
```



* What is the relationship between the numerator rating and the favorite counts?



```
fig = plt.figure(figsize=(10,10))
plt.scatter('rating_numerator', 'favorite_count', data=twitter_archive_master)
#Adding the aesthetics
plt.title('Scatter Plot Between Rating Numerator and Farorite Count')
plt.xlabel('rating_numerator')
plt.ylabel('favorite_count')
#Show the plot
plt.show()
```



Conclusions

Majority of the tweets from WeRateDogs accounts were posted using an Iphone. 98.04% of the tweets have Twitter for Iphone as the device used with the remaining less than two percentage having been posted by TweetDeck (1.40%) and Twitter Web Client (0.56%)

There is a positive correlation between retweet count and favorite count, an increase in retweet count would mean also there would an increase in the favourites counts

Best rated dogs are between ratings 13 and 14. Likewise these dogs received more favorite counts as compared to least rate dogs at 7 which similarly received low counts of retweets and favorite