In [1]:

```
#mentioning all the libraries
import numpy as np
import pandas as pd
import requests
import tweepy
import json
import matplotlib.pyplot as plt
import warnings
```

## GATHER

In [2]:

```
#reading and loading a csv file
twitter=pd.read_csv('twitter_archive_enhanced.csv')
```

In [3]:

```
#loading a tsv file
url="https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_imag
predictions/image-predictions.tsv"
response = requests.get(url)

with open('image_predictions.tsv', 'wb') as file:
    file.write(response.content)

image = pd.read_csv('image_predictions.tsv', sep='\t')
```

In [4]:

```
#creating API object. I have removed original values of consumer_key, consu
mer_secret,OAUTH_TOKEN,OAUTH_TOKEN_SECRET
consumer_key = 'my consumer key'
consumer_secret = 'my consumer secret'
OAUTH_TOKEN = 'my oauth token'
OAUTH_TOKEN_SECRET = 'my token secret'
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
api = tweepy.API(auth, wait_on_rate_limit = True, wait_on_rate_limit_notify
= True)
```

In [5]:

```
#writing json data to tweet_json.txt file
with open('tweet_json.txt', 'a', encoding='utf8') as file:
    for tweet_id in twitter['tweet_id']:
        try:
            tweet = api.get_status(tweet_id, tweet_mode='extended')
            json.dump(tweet._json, file)
            file.write('\n')
        except:
            continue
```

```
Rate limit reached. Sleeping for: 722
Rate limit reached. Sleeping for: 724
```

In [6]:

```
#append each data to tweet_list line by line
tweet_list = []

file = open('tweet_json.txt', "r")

for line in file:
    try:
        data = json.loads(line)
        tweet_list.append(data)
    except:
        continue

file.close()
```

In [7]:

```
#creating dataframe for tweet_data
tweet_data = pd.DataFrame()
tweet_data['id'] = list(map(lambda tweet: tweet['id'], tweet_list))
tweet_data['retweet_count'] = list(map(lambda tweet: tweet['retweet_count']
, tweet_list))
tweet_data['favorite_count'] = list(map(lambda tweet:
tweet['favorite_count'], tweet_list))
```

**We have three dataframes named as :twitter , tweet_data , image**

## ASSESS

In [8]:

```
# first ten entries of twitter dataframe
twitter.head(10)
```

Out[8]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **0** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 +0000 | <a href="http://twitr... |
| **1** | 892177421306343426 | NaN | NaN | 2017-08-01 00:17:27 +0000 | <a href="http://twitr... |
| **2** | | | | 2017-07- | |

|   | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp |   |
|---|---|---|---|---|---|
|   | 891815181378084864 | NaN | NaN | 31 00:18:03 +0000 | <a href="http://twit r... |
| **3** | 891689557279858688 | NaN | NaN | 2017-07-30 15:58:51 +0000 | <a href="http://twit r... |
| **4** | 891327558926688256 | NaN | NaN | 2017-07-29 16:00:24 +0000 | <a href="http://twit r... |
| **5** | 891087950875897856 | NaN | NaN | 2017-07-29 00:08:17 +0000 | <a href="http://twit r... |
| **6** | 890971913173991426 | NaN | NaN | 2017-07-28 16:27:12 +0000 | <a href="http://twit r... |
| **7** | 890729181411237888 | NaN | NaN | 2017-07-28 00:22:40 +0000 | <a href="http://twit r... |
| **8** | 890609185150312448 | NaN | NaN | 2017-07-27 16:25:51 +0000 | <a href="http://twit r... |
| **9** | 890240255349198849 | NaN | NaN | 2017-07-26 15:59:51 +0000 | <a href="http://twit r... |

In [9]:

```
# first 10 entries of image dataframe
image.head(10)
```

Out[9]:

| | tweet_id | jpg_url | img_num | |
|---|---|---|---|---|
| **0** | 666020888022790149 | https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg | 1 | Welsh_ |
| **1** | 666029285002620928 | https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg | 1 | redbon |
| **2** | 666033412701032449 | https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg | 1 | Germa |
| **3** | 666044226329800704 | https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg | 1 | Rhodes |
| **4** | 666049248165822465 | https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg | 1 | miniatu |
| **5** | 666050758794694657 | https://pbs.twimg.com/media/CT5Jof1WUAEuVxN.jpg | 1 | Berne |
| **6** | 666051853826850816 | https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg | 1 | box_tur |
| **7** | 666055525042405380 | https://pbs.twimg.com/media/CT5N9tpXIAAifs1.jpg | 1 | chow |
| **8** | 666057090499244032 | https://pbs.twimg.com/media/CT5PY90WoAAQGLo.jpg | 1 | shoppi |
| **9** | 666058600524156928 | https://pbs.twimg.com/media/CT5Qw94XAAA_2dP.jpg | 1 | miniatu |

In [10]:

```
# first 10 entries of tweet_data dataframe
tweet_data.head(10)
```

Out[10]:

| | id | retweet_count | favorite_count |
|---|---|---|---|
| **0** | 892420643555336193 | 8580 | 38778 |
| **1** | 892177421306343426 | 6304 | 33207 |
| **2** | 891815181378084864 | 4186 | 25009 |
| **3** | 891689557279858688 | 8706 | 42146 |
| **4** | 891327558926688256 | 9468 | 40312 |
| **5** | 891087950875897856 | 3136 | 20207 |
| **6** | 890971913173991426 | 2087 | 11847 |
| **7** | 890729181411237888 | 19021 | 65476 |
| **8** | 890609185150312448 | 4292 | 27769 |
| **9** | 890240255349198849 | 7464 | 31923 |

In [11]:

```
twitter.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
```

```
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
doggo                         2356 non-null object
floofer                       2356 non-null object
pupper                        2356 non-null object
puppo                         2356 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

In [12]:

```
image.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null int64
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
p1_dog      2075 non-null bool
p2          2075 non-null object
p2_conf     2075 non-null float64
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

In [13]:

```
tweet_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23752 entries, 0 to 23751
Data columns (total 3 columns):
id                23752 non-null int64
retweet_count     23752 non-null int64
favorite_count    23752 non-null int64
dtypes: int64(3)
memory usage: 556.8 KB
```

In [14]:

```
# checking for null values in twitter dataframe for each column
twitter.isnull().sum()
```

Out[14]:

```
tweet_id                         0
in_reply_to_status_id         2278
in_reply_to_user_id           2278
timestamp                        0
source                           0
text                             0
```

```
retweeted_status_id           2175
retweeted_status_user_id      2175
retweeted_status_timestamp    2175
expanded_urls                   59
rating_numerator                 0
rating_denominator               0
name                             0
doggo                            0
floofer                          0
pupper                           0
puppo                            0
dtype: int64
```

In [15]:

```
# checking for null values in image dataframe for each column
image.isnull().sum()
```

Out[15]:

```
tweet_id    0
jpg_url     0
img_num     0
p1          0
p1_conf     0
p1_dog      0
p2          0
p2_conf     0
p2_dog      0
p3          0
p3_conf     0
p3_dog      0
dtype: int64
```

In [16]:

```
#checking for null values in tweet_data dataframe for each column
tweet_data.isnull().sum()
```

Out[16]:

```
id                0
retweet_count     0
favorite_count    0
dtype: int64
```

In [17]:

```
# inspecting the name column of twitter dataframe
twitter['name'].value_counts()
```

Out[17]:

```
None         745
a             55
Charlie       12
Lucy          11
Cooper        11
Oliver        11
Tucker        10
Penny         10
Lola          10
```

```
Bo            9
Winston       9
the           8
Sadie         8
Buddy         7
Daisy         7
Bailey        7
Toby          7
an            7
Koda          6
Oscar         6
Dave          6
Milo          6
Scout         6
Stanley       6
Leo           6
Jax           6
Bella         6
Rusty         6
Jack          6
Alfie         5
            ...
Moofasa       1
Glacier       1
Socks         1
Biden         1
Snickers      1
Chaz          1
Pumpkin       1
Eevee         1
Thor          1
Crumpet       1
Kane          1
Eazy          1
Julius        1
Charleson     1
Brownie       1
Rambo         1
Batdog        1
Tedrick       1
Rueben        1
Kulet         1
Nigel         1
Hanz          1
Bones         1
Diogi         1
Blakely       1
Sephie        1
Tug           1
Willow        1
Maks          1
Bubba         1
Name: name, Length: 957, dtype: int64
```

In [18]:

```
#checking those texts which contain any decimal rating as rating_numerator values can be wrong for those particular rows
twitter[twitter['text'].str.contains(r"(\d+\.\d*\/\d+)")]
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:2: UserWarning
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:1: UserWarning
: This pattern has match groups. To actually get the groups, use str.extrac
t.
```

Out[18]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **45** | 883482846933004288 | NaN | NaN | 2017-07-08 00:28:19 +0000 | <a href="http:/ r... |
| **340** | 832215909146226688 | NaN | NaN | 2017-02-16 13:11:49 +0000 | <a href="http:/ r... |
| **695** | 786709082849828864 | NaN | NaN | 2016-10-13 23:23:56 +0000 | <a href="http:/ r... |
| **763** | 778027034220126208 | NaN | NaN | 2016-09-20 00:24:34 +0000 | <a href="http:/ r... |
| **1689** | 681340665377193984 | 6.813394e+17 | 4.196984e+09 | 2015-12-28 05:07:27 +0000 | <a href="http:/ r... |
| **1712** | 680494726643068929 | NaN | NaN | 2015-12-25 21:06:00 +0000 | <a href="http:/ r... |

In [19]:

```
#Checking if any dog has 0 as its rating_denominator value as that value is
wrong
twitter[twitter['rating_denominator']==0]
```

Out[19]:

|  | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp |  |
|---|---|---|---|---|---|
| **313** | 835246439529840640 | 8.352460e+17 | 26259576.0 | 2017-02-24 21:54:03 +0000 | <a href="http://tw r... |

In [20]:

```
#checking for those rows in which rating denominator is not having value 10
twitter[twitter['rating_denominator']!=10]
```

Out[20]:

|  | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp |  |
|---|---|---|---|---|---|
| **313** | 835246439529840640 | 8.352460e+17 | 2.625958e+07 | 2017-02-24 21:54:03 +0000 | <a href="http:/ r... |
| **342** | 832088576586297345 | 8.320875e+17 | 3.058208e+07 | 2017-02-16 04:45:50 +0000 | <a href="http:/ r... |
| **433** | 820690176645140481 | NaN | NaN | 2017-01-15 17:52:40 +0000 | <a href="http:/ r... |
| **516** | 810984652412424192 | NaN | NaN | 2016-12-19 23:06:23 +0000 | <a href="http:/ r... |
| **784** | 775096608509886464 | NaN | NaN | 2016-09-11 22:20:06 +0000 | <a href="http:/ r... |
| **902** | 758467244762497024 | NaN | NaN | 2016-07-28 01:00:57 +0000 | <a href="http:/ r... |
| **1068** | 740373189193256964 | NaN | NaN | 2016-06-08 02:41:38 +0000 | <a href="http:/ r... |
| **1120** | 731156023742988288 | NaN | NaN | 2016-05-13 16:15:54 +0000 | <a href="http:/ r... |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **1165** | 722974582966214656 | NaN | NaN | 2016-04-21 02:25:47 +0000 | <a href="http:/ r... |
| **1202** | 716439118184652801 | NaN | NaN | 2016-04-03 01:36:11 +0000 | <a href="http:/ r... |
| **1228** | 713900603437621249 | NaN | NaN | 2016-03-27 01:29:02 +0000 | <a href="http:/ r... |
| **1254** | 710658690886586372 | NaN | NaN | 2016-03-18 02:46:49 +0000 | <a href="http:/ r... |
| **1274** | 709198395643068416 | NaN | NaN | 2016-03-14 02:04:08 +0000 | <a href="http:/ r... |
| **1351** | 704054845121142784 | NaN | NaN | 2016-02-28 21:25:30 +0000 | <a href="http:/ r... |
| **1433** | 697463031882764288 | NaN | NaN | 2016-02-10 16:51:59 +0000 | <a href="http:/ r... |
| **1598** | 686035780142297088 | 6.860340e+17 | 4.196984e+09 | 2016-01-10 04:04:10 +0000 | <a href="http:/ r... |
| **1634** | 684225744407494656 | 6.842229e+17 | 4.196984e+09 | 2016-01-05 04:11:44 +0000 | <a href="http:/ r... |
| **1635** | 684222868335505415 | NaN | NaN | 2016-01-05 04:00:18 +0000 | <a href="http:/ r... |
| **1662** | 682962037429899265 | NaN | NaN | 2016-01-01 16:30:13 +0000 | <a href="http:/ r... |
| **1663** | | | | 2016-01- | <a |

|  | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp |  |
|---|---|---|---|---|---|
|  | 682808988178739200 | 6.827884e+17 | 4.196984e+09 | 01 06:22:03 +0000 | href="http:/ r... |
| **1779** | 677716515794329600 | NaN | NaN | 2015-12-18 05:06:23 +0000 | <a href="http:/ r... |
| **1843** | 675853064436391936 | NaN | NaN | 2015-12-13 01:41:41 +0000 | <a href="http:/ r... |
| **2335** | 666287406224695296 | NaN | NaN | 2015-11-16 16:11:11 +0000 | <a href="http:/ r... |

In [21]:

```
image.head(3)
```

Out[21]:

|  | tweet_id | jpg_url | img_num |  |
|---|---|---|---|---|
| **0** | 666020888022790149 | https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg | 1 | Welsh_s |
| **1** | 666029285002620928 | https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg | 1 | redbone |
| **2** | 666033412701032449 | https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg | 1 | German |

In [22]:

```
#checking for duplicate values in tweet_data dataframe
tweet_data.duplicated().sum()
```

Out[22]:

```
14003
```

In [23]:

```
#checking for duplicate values in twitter dataframe
twitter.duplicated().sum()
```

Out[23]:

```
0
```

In [24]:

```
#checking for duplicate values in image dataframe
image.duplicated().sum()
```

Out[24]:

```
0
```

**QUALITY ISSUES WITH DATAFRAMES**

-> data type conflicts (time_stamp, in_reply_to_user_id, in_reply_to_status_id,tweet_id , etc...)

-> duplicacy of records( In tweet_data dataframe)

-> dataframe consists of retweets

-> Incorrect dog names in twitter dataframe

-> tweets with no images are present

-> record containing wrong value of rating_denominator as 0

-> some columns such as retweeted_status_id and retweeted_status_user_id are not required

-> the rating_numerator value of records having decimal rating in their text is not written properly

**TIDINESS ISSUES**

-> The distinct dog type columns ( can be resolved by combining dog type columns into one column)

-> merging all the three dataframes together to get a new dataframe

# CLEAN

In [25]:

```
# creating copies of dataframes for cleaning purpose
twitter_clean=twitter.copy()
image_clean=image.copy()
tweet_data_clean=tweet_data.copy()
```

**DEFINE**

Removing retweets from twitter_clean dataframe

**CODE**

In [26]:

```
# if retweet status id is null then there is no retweet
twitter_clean=twitter_clean[twitter_clean['retweeted_status_id'].isnull()]
```

In [27]:

```
twitter_clean
```

Out[27]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **0** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 +0000 | <a href="http:/ r... |
| **1** | 892177421306343426 | NaN | NaN | 2017-08-01 00:17:27 +0000 | <a href="http:/ r... |
| **2** | 891815181378084864 | NaN | NaN | 2017-07-31 00:18:03 +0000 | <a href="http:/ r... |
| **3** | 891689557279858688 | NaN | NaN | 2017-07-30 15:58:51 +0000 | <a href="http:/ r... |
| **4** | 891327558926688256 | NaN | NaN | 2017-07-29 16:00:24 +0000 | <a href="http:/ r... |
| **5** | 891087950875897856 | NaN | NaN | 2017-07-29 00:08:17 +0000 | <a href="http:/ r... |
| **6** | 890971913173991426 | NaN | NaN | 2017-07-28 16:27:12 +0000 | <a href="http:/ r... |
| **7** | 890729181411237888 | NaN | NaN | 2017-07-28 00:22:40 +0000 | <a href="http:/ r... |
| **8** | 890609185150312448 | NaN | NaN | 2017-07-27 16:25:51 +0000 | <a href="http:/ r... |
| **9** | 890240255349198849 | NaN | NaN | 2017-07-26 15:59:51 +0000 | <a href="http:/ r... |
| **10** | 890006608113172480 | NaN | NaN | 2017-07-26 00:31:25 | <a href="http:/ |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **11** | 889880896479866881 | NaN | NaN | 2017-07-25 16:11:53 +0000 | <a href="http:/ r... |
| **12** | 889665388333682689 | NaN | NaN | 2017-07-25 01:55:32 +0000 | <a href="http:/ r... |
| **13** | 889638837579907072 | NaN | NaN | 2017-07-25 00:10:02 +0000 | <a href="http:/ r... |
| **14** | 889531135344209921 | NaN | NaN | 2017-07-24 17:02:04 +0000 | <a href="http:/ r... |
| **15** | 889278841981685760 | NaN | NaN | 2017-07-24 00:19:32 +0000 | <a href="http:/ r... |
| **16** | 888917238123831296 | NaN | NaN | 2017-07-23 00:22:39 +0000 | <a href="http:/ r... |
| **17** | 888804989199671297 | NaN | NaN | 2017-07-22 16:56:37 +0000 | <a href="http:/ r... |
| **18** | 888554962724278272 | NaN | NaN | 2017-07-22 00:23:06 +0000 | <a href="http:/ r... |
| **20** | 888078434458587136 | NaN | NaN | 2017-07-20 16:49:33 +0000 | <a href="http:/ r... |
| **21** | 887705289381826560 | NaN | NaN | 2017-07-19 16:06:48 +0000 | <a href="http:/ r... |
| **22** | 887517139158093824 | NaN | NaN | 2017-07-19 03:39:09 +0000 | <a href="http:/ r... |
| **23** | | | | 2017-07- | |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| | 887473957103951883 | NaN | NaN | 19 00:47:34 +0000 | <a href="http:/ r... |
| **24** | 887343217045368832 | NaN | NaN | 2017-07-18 16:08:03 +0000 | <a href="http:/ r... |
| **25** | 887101392804085760 | NaN | NaN | 2017-07-18 00:07:08 +0000 | <a href="http:/ r... |
| **26** | 886983233522544640 | NaN | NaN | 2017-07-17 16:17:36 +0000 | <a href="http:/ r... |
| **27** | 886736880519319552 | NaN | NaN | 2017-07-16 23:58:41 +0000 | <a href="http:/ r... |
| **28** | 886680336477933568 | NaN | NaN | 2017-07-16 20:14:00 +0000 | <a href="http:/ r... |
| **29** | 886366144734445568 | NaN | NaN | 2017-07-15 23:25:31 +0000 | <a href="http:/ r... |
| **30** | 886267009285017600 | 8.862664e+17 | 2.281182e+09 | 2017-07-15 16:51:35 +0000 | <a href="http:/ r... |
| **...** | ... | ... | ... | ... | ... |
| **2326** | 666411507551481857 | NaN | NaN | 2015-11-17 00:24:19 +0000 | <a href="http:/ r... |
| **2327** | 666407126856765440 | NaN | NaN | 2015-11-17 00:06:54 +0000 | <a href="http:/ r... |
| **2328** | 666396247373291520 | NaN | NaN | 2015-11-16 23:23:41 +0000 | <a href="http:/ r... |
| **2329** | | | | 2015-11- | <a |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| | 666373753744588802 | NaN | NaN | 16 21:54:18 +0000 | href="http:/ r... |
| **2330** | 666362758909284353 | NaN | NaN | 2015-11-16 21:10:36 +0000 | <a href="http:/ r... |
| **2331** | 666353288456101888 | NaN | NaN | 2015-11-16 20:32:58 +0000 | <a href="http:/ r... |
| **2332** | 666345417576210432 | NaN | NaN | 2015-11-16 20:01:42 +0000 | <a href="http:/ r... |
| **2333** | 666337882303524864 | NaN | NaN | 2015-11-16 19:31:45 +0000 | <a href="http:/ r... |
| **2334** | 666293911632134144 | NaN | NaN | 2015-11-16 16:37:02 +0000 | <a href="http:/ r... |
| **2335** | 666287406224695296 | NaN | NaN | 2015-11-16 16:11:11 +0000 | <a href="http:/ r... |
| **2336** | 666273097616637952 | NaN | NaN | 2015-11-16 15:14:19 +0000 | <a href="http:/ r... |
| **2337** | 666268910803644416 | NaN | NaN | 2015-11-16 14:57:41 +0000 | <a href="http:/ r... |
| **2338** | 666104133288665088 | NaN | NaN | 2015-11-16 04:02:55 +0000 | <a href="http:/ r... |
| **2339** | 666102155909144576 | NaN | NaN | 2015-11-16 03:55:04 +0000 | <a href="http:/ r... |
| **2340** | 666099513787052032 | NaN | NaN | 2015-11-16 03:44:34 +0000 | <a href="http:/ r... |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| | | | | +0000 | |
| **2341** | 666094000022159362 | NaN | NaN | 2015-11-16 03:22:39 +0000 | <a href="http:/ r... |
| **2342** | 666082916733198337 | NaN | NaN | 2015-11-16 02:38:37 +0000 | <a href="http:/ r... |
| **2343** | 666073100786774016 | NaN | NaN | 2015-11-16 01:59:36 +0000 | <a href="http:/ r... |
| **2344** | 666071193221509120 | NaN | NaN | 2015-11-16 01:52:02 +0000 | <a href="http:/ r... |
| **2345** | 666063827256086533 | NaN | NaN | 2015-11-16 01:22:45 +0000 | <a href="http:/ r... |
| **2346** | 666058600524156928 | NaN | NaN | 2015-11-16 01:01:59 +0000 | <a href="http:/ r... |
| **2347** | 666057090499244032 | NaN | NaN | 2015-11-16 00:55:59 +0000 | <a href="http:/ r... |
| **2348** | 666055525042405380 | NaN | NaN | 2015-11-16 00:49:46 +0000 | <a href="http:/ r... |
| **2349** | 666051853826850816 | NaN | NaN | 2015-11-16 00:35:11 +0000 | <a href="http:/ r... |
| **2350** | 666050758794694657 | NaN | NaN | 2015-11-16 00:30:50 +0000 | <a href="http:/ r... |
| **2351** | 666049248165822465 | NaN | NaN | 2015-11-16 00:24:50 +0000 | <a href="http:/ r... |
| **2352** | | | | 2015-11- | <a |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| | 66604422632[illegible] | NaN | NaN | 16 00:04:52 +0000 | href="http:/ r... |
| **2353** | 666033412701032449 | NaN | NaN | 2015-11-15 23:21:54 +0000 | <a href="http:/ r... |
| **2354** | 666029285002620928 | NaN | NaN | 2015-11-15 23:05:30 +0000 | <a href="http:/ r... |
| **2355** | 666020888022790149 | NaN | NaN | 2015-11-15 22:32:08 +0000 | <a href="http:/ r... |

2175 rows × 17 columns

**TEST**

In [28]:

```
# Through info we can observe that there is no non null value present for r
etweeted_status_id
twitter_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2175 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2175 non-null object
source                        2175 non-null object
text                          2175 non-null object
retweeted_status_id           0 non-null float64
retweeted_status_user_id      0 non-null float64
retweeted_status_timestamp    0 non-null object
expanded_urls                 2117 non-null object
rating_numerator              2175 non-null int64
rating_denominator            2175 non-null int64
name                          2175 non-null object
doggo                         2175 non-null object
floofer                       2175 non-null object
pupper                        2175 non-null object
puppo                         2175 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 305.9+ KB
```

**DEFINE**

removing columns related to retweets from twitter_clean dataframe

**CODE**

In [29]:

```
columns=['retweeted_status_id','retweeted_status_user_id','retweeted_status_
timestamp']
twitter_clean.drop(columns,inplace=True,axis=1)
```

**TEST**

In [30]:

```
# to check if the columns related to retweets are deleted
twitter_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 14 columns):
tweet_id                 2175 non-null int64
in_reply_to_status_id    78 non-null float64
in_reply_to_user_id      78 non-null float64
timestamp                2175 non-null object
source                   2175 non-null object
text                     2175 non-null object
expanded_urls            2117 non-null object
rating_numerator         2175 non-null int64
rating_denominator       2175 non-null int64
name                     2175 non-null object
doggo                    2175 non-null object
floofer                  2175 non-null object
pupper                   2175 non-null object
puppo                    2175 non-null object
dtypes: float64(2), int64(3), object(9)
memory usage: 254.9+ KB
```

**DEFINE**

Removing the invalid names of dogs (We observed while assessing that the dog names having all lowercase letters are wrong names)

**CODE**

In [31]:

```
# replacing invalid names with None
for name in twitter_clean['name']:
    if name.islower()==True:
        twitter_clean.replace(name,'None',inplace=True)
```

**TEST**

In [32]:

```
# to check whether invalid dog names are removed and replaced with None
twitter_clean['name'].value_counts()
```

Out[32]:

```
None              784
Charlie            11
Lucy               11
Oliver             10
Cooper             10
Penny               9
Tucker              9
Winston             8
Sadie               8
Lola                8
Daisy               7
Toby                7
Bo                  6
Jax                 6
Bailey              6
Bella               6
Stanley             6
Oscar               6
Koda                6
Louis               5
Rusty               5
Scout               5
Buddy               5
Bentley             5
Dave                5
Milo                5
Leo                 5
Chester             5
Alfie               4
Brody               4
                 ...
Opie                1
Traviss             1
Kota                1
Rupert              1
Shooter             1
Bronte              1
Eve                 1
Blanket             1
Hercules            1
William             1
Apollo              1
Chompsky            1
Terrance            1
Ziva                1
Bobby               1
Rufio               1
Sandra              1
Cleopatricia        1
Hero                1
Eugene              1
Alejandro           1
Ozzie               1
Shikha              1
Jessiga             1
Fido                1
Cecil               1
Brat                1
```

```
Bell                 1
Danny                1
Chadrick             1
Name: name, Length: 931, dtype: int64
```

**DEFINE**

to replace invalid ratings with the decimal rating present in the text (if present)

**CODE**

In [33]:

```
#column width is adjusted to make the full text visible
pd.set_option('display.max_colwidth', -1)
twitter_clean[twitter_clean['text'].str.contains(r"(\d+\.\d*\/\d+)")]
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:3: UserWarning
: This pattern has match groups. To actually get the groups, use str.extrac
t.
  This is separate from the ipykernel package so we can avoid doing imports
until
```

Out[33]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **45** | 883482846933004288 | NaN | NaN | 2017-07-08 00:28:19 +0000 | <a href="http:/ rel="nofollo |
| **695** | 786709082849828864 | NaN | NaN | 2016-10-13 23:23:56 +0000 | <a href="http:/ rel="nofollo |
| **763** | 778027034220126208 | NaN | NaN | 2016-09-20 00:24:34 +0000 | <a href="http:/ rel="nofollo |
| **1689** | 681340665377193984 | 6.813394e+17 | 4.196984e+09 | 2015-12-28 05:07:27 +0000 | <a href="http:/ rel="nofollo |

| 1712 | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| | 680494726643068929 | NaN | NaN | 2015-12-25 21:06:00 +0000 | <a href="http:/ rel="nofollo |

In [34]:

```
# replacing the invalid ratings with the correct decimal ratings
twitter_clean.loc[twitter_clean['tweet_id']==883482846933004288,"rating_num
erator"]=13.5
twitter_clean.loc[twitter_clean['tweet_id']==786709082849828864,"rating_num
erator"]=9.75
twitter_clean.loc[twitter_clean['tweet_id']==778027034220126208,"rating_num
erator"]=11.27
twitter_clean.loc[twitter_clean['tweet_id']==681340665377193984,"rating_num
erator"]=9.5
twitter_clean.loc[twitter_clean['tweet_id']==680494726643068929,"rating_num
erator"]=11.26
```

**TEST**

In [35]:

```
# checking whether updation of rating_denominator and rating_numerator is d
one
twitter_clean[twitter_clean['text'].str.contains(r"(\d+\.\d*\/\d+)")]
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:2: UserWarning
: This pattern has match groups. To actually get the groups, use str.extrac
t.
```

Out[35]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| 45 | 883482846933004288 | NaN | NaN | 2017-07-08 00:28:19 +0000 | <a href="http:/ rel="nofollo |
| 695 | 786709082849828864 | NaN | NaN | 2016-10-13 23:23:56 +0000 | <a href="http:/ rel="nofollo |
| 763 | 778027034220126208 | NaN | NaN | 2016-09-20 | <a href="http:/ |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| | | | | | |
| **1689** | 681340665377193984 | 6.813394e+17 | 4.196984e+09 | 2015-12-28 05:07:27 +0000 | <a href="http:/ rel="nofollo |
| **1712** | 680494726643068929 | NaN | NaN | 2015-12-25 21:06:00 +0000 | <a href="http:/ rel="nofollo |

**DEFINE**

correcting the rating_numerator and rating_denominator values where the value of rating_denominator is 0

**CODE**

In [36]:

```
#checking for the rows having rating_denominator = 0
twitter_clean[twitter_clean['rating_denominator']==0]
```

Out[36]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **313** | 835246439529840640 | 8.352460e+17 | 26259576.0 | 2017-02-24 21:54:03 +0000 | <a href="http://t rel="nofollow |

In [37]:

```
#inspecting the text to check whether rating is available or not
twitter_clean.loc[twitter_clean['rating_denominator']==0].text
```

Out[37]:

```
313    @jonnysun @Lin_Manuel ok jomny I know you're excited but 960/00 isn'
t a valid rating, 13/10 is tho
Name: text, dtype: object
```

```
In [38]:
```

```
# changing the invalid rating values to valid values
twitter_clean.loc[twitter_clean['tweet_id']==835246439529840640,"rating_num
erator"]=13
twitter_clean.loc[twitter_clean['tweet_id']==835246439529840640,"rating_den
ominator"]=10
```

**TEST**

```
In [39]:
```

```
# checking whether changes are performed for the particular tweet_id
twitter_clean.loc[twitter_clean['tweet_id']==835246439529840640]
```

```
Out[39]:
```

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **313** | 835246439529840640 | 8.352460e+17 | 26259576.0 | 2017-02-24 21:54:03 +0000 | <a href="http://tw rel="nofollow |

**DEFINE**

Merging the columns of dog type into one column

**CODE**

```
In [40]:
```

```
#defining a new column dog_type and removing the columns which are not req
uired
twitter_clean['dog_type']=twitter_clean.text.str.extract('(puppo|pupper|floo
fer|doggo)',expand=True)
columns=['doggo','floofer','pupper','puppo']
twitter_clean=twitter_clean.drop(columns,axis=1)
```

**TEST**

```
In [41]:
```

```
#to check whether the columns are merged into one column dog_type
twitter_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 11 columns):
tweet_id                    2175 non-null int64
```

```
in_reply_to_status_id    78 non-null float64
in_reply_to_user_id      78 non-null float64
timestamp                2175 non-null object
source                   2175 non-null object
text                     2175 non-null object
expanded_urls            2117 non-null object
rating_numerator         2175 non-null float64
rating_denominator       2175 non-null int64
name                     2175 non-null object
dog_type                 364 non-null object
dtypes: float64(3), int64(2), object(6)
memory usage: 203.9+ KB
```

**DEFINE**

changing data types of columns of twitter_clean dataframe

**CODE**

In [42]:

```
#changing data type of rating_numerator and rating_denominator to float
twitter_clean['rating_numerator']=twitter_clean.rating_numerator.astype('fl
oat')
twitter_clean['rating_denominator']=twitter_clean.rating_denominator.astype
('float')
```

In [43]:

```
#changing data type of dog_type to category data type
#changing data type of timestamp to datetime data type
twitter_clean['dog_type']=twitter_clean['dog_type'].astype('category')
twitter_clean['timestamp']=pd.to_datetime(twitter_clean['timestamp'])
```

In [44]:

```
#changing datatype of mentioned columns to string
twitter_clean['in_reply_to_status_id'] =
twitter_clean['in_reply_to_status_id'].astype('str')
twitter_clean['in_reply_to_user_id'] = twitter_clean['in_reply_to_user_id']
.astype('str')
twitter_clean['tweet_id'] = twitter_clean['tweet_id'].astype('str')
```

**TEST**

In [45]:

```
twitter_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 11 columns):
tweet_id                 2175 non-null object
in_reply_to_status_id    2175 non-null object
in_reply_to_user_id      2175 non-null object
timestamp                2175 non-null datetime64[ns]
source                   2175 non-null object
```

```
text                      2175 non-null object
expanded_urls             2117 non-null object
rating_numerator          2175 non-null float64
rating_denominator        2175 non-null float64
name                      2175 non-null object
dog_type                  364 non-null category
dtypes: category(1), datetime64[ns](1), float64(2), object(7)
memory usage: 189.2+ KB
```

**DEFINE**

changing the rating_numerator and rating_denominator where rating denominator is not equal to 10

**CODE**

In [46]:

```
# checking for those records where rating_denominator is not equal to 10
twitter_clean[twitter_clean['rating_denominator']!=10]
```

Out[46]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **342** | 832088576586297345 | 8.3208754756e+17 | 30582082.0 | 2017-02-16 04:45:50 | <a href="http:/ rel="nofollo |
| **433** | 820690176645140481 | nan | nan | 2017-01-15 17:52:40 | <a href="http:/ rel="nofollo |
| **516** | 810984652412424192 | nan | nan | 2016-12-19 23:06:23 | <a href="http:/ rel="nofollo |
| **902** | 758467244762497024 | nan | nan | 2016-07-28 01:00:57 | <a href="http:/ rel="nofollo |
| **1068** | 740373189193256964 | nan | nan | 2016-06-08 02:41:38 | <a href="http:/ rel="nofollo |
| **1120** | 731156023742988288 | nan | nan | 2016-05-13 | <a href="http:/ |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | 16:15:54 timestamp | rel="nofollo |
|---|---|---|---|---|---|
| | | | | | |
| **1165** | 722974582966214656 | nan | nan | 2016-04-21 02:25:47 | <a href="http:/ rel="nofollo |
| **1202** | 716439118184652801 | nan | nan | 2016-04-03 01:36:11 | <a href="http:/ rel="nofollo |
| **1228** | 713900603437621249 | nan | nan | 2016-03-27 01:29:02 | <a href="http:/ rel="nofollo |
| **1254** | 710658690886586372 | nan | nan | 2016-03-18 02:46:49 | <a href="http:/ rel="nofollo |
| **1274** | 709198395643068416 | nan | nan | 2016-03-14 02:04:08 | <a href="http:/ rel="nofollo |
| **1351** | 704054845121142784 | nan | nan | 2016-02-28 21:25:30 | <a href="http:/ rel="nofollo |
| **1433** | 697463031882764288 | nan | nan | 2016-02-10 16:51:59 | <a href="http:/ rel="nofollo |
| **1598** | 686035780142297088 | 6.86034024801e+17 | 4196983835.0 | 2016-01-10 04:04:10 | <a href="http:/ rel="nofollo |
| **1634** | 684225744407494656 | 6.84222868336e+17 | 4196983835.0 | 2016-01-05 04:11:44 | <a href="http:/ rel="nofollo |
| **1635** | 684222868335505415 | nan | nan | 2016-01-05 | <a href="http:/ |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | 04:00:18 timestamp | rel="nofollo |
|---|---|---|---|---|---|
| | | | | | |
| **1662** | 682962037429899265 | nan | nan | 2016-01-01 16:30:13 | <a href="http:/ rel="nofollo |
| **1663** | 682808988178739200 | 6.82788441538e+17 | 4196983835.0 | 2016-01-01 06:22:03 | <a href="http:/ rel="nofollo |
| **1779** | 677716515794329600 | nan | nan | 2015-12-18 05:06:23 | <a href="http:/ rel="nofollo |
| **1843** | 675853064436391936 | nan | nan | 2015-12-13 01:41:41 | <a href="http:/ rel="nofollo |
| **2335** | 666287406224695296 | nan | nan | 2015-11-16 16:11:11 | <a href="http:/ rel="nofollo |

In [47]:

```
#changing incorrect values by inspecting the text in which the correct rati
ng is available
twitter_clean.loc[twitter_clean['tweet_id']=='722974582966214656',"rating_n
umerator"]=13
twitter_clean.loc[twitter_clean['tweet_id']=='722974582966214656',"rating_d
enominator"]=10
```

In [48]:

```
#one more record found whose rating was available in text
twitter_clean.loc[twitter_clean['tweet_id']=='716439118184652801',"rating_n
umerator"]=11
twitter_clean.loc[twitter_clean['tweet_id']=='716439118184652801',"rating_d
enominator"]=10
```

we observed that when ratings are done for group of dogs then rating_denominator is of other value than 10. We can ignore those records

**TEST**

In [49]:

```
#checking if values of rating_numerator and rating_denominator is changed
twitter_clean.loc[twitter_clean['tweet_id']=='716439118184652801']
```

Out[49]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **1202** | 716439118184652801 | nan | nan | 2016-04-03 01:36:11 | <a href="http:// rel="nofollo |

In [50]:

```
#checking that value of rating_numerator and rating_denominator is changed
for another tweet_id also
twitter_clean.loc[twitter_clean['tweet_id']=='722974582966214656']
```

Out[50]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **1165** | 722974582966214656 | nan | nan | 2016-04-21 02:25:47 | <a href="http:// rel="nofollo |

**DEFINE**

removing rows that does not contain expanded_urls for images

**code**

In [51]:

```
#removing rows not containing images
twitter_clean = twitter_clean.dropna(subset=['expanded_urls'])
```

**TEST**

In [52]:

```
#to check whether records not having image urls are removed from the datafr
ame
twitter_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2117 entries, 0 to 2355
```

```
Data columns (total 11 columns):
tweet_id                 2117 non-null object
in_reply_to_status_id    2117 non-null object
in_reply_to_user_id      2117 non-null object
timestamp                2117 non-null datetime64[ns]
source                   2117 non-null object
text                     2117 non-null object
expanded_urls            2117 non-null object
rating_numerator         2117 non-null float64
rating_denominator       2117 non-null float64
name                     2117 non-null object
dog_type                 356 non-null category
dtypes: category(1), datetime64[ns](1), float64(2), object(7)
memory usage: 184.2+ KB
```

**DEFINE**

removing duplicated records from tweet_data_clean dataframe

**CODE**

In [53]:

```
#checking duplicated records
tweet_data_clean.duplicated().sum()
```

Out[53]:

```
14003
```

In [54]:

```
#removing duplicated records
tweet_data_clean=tweet_data_clean.drop_duplicates()
```

**TEST**

In [55]:

```
#checking whether duplicated rows are deleted
tweet_data_clean.duplicated().sum()
```

Out[55]:

```
0
```

**DEFINE**

changing column name and data type of column named 'id' of tweet_data_clean dataframe

**CODE**

In [56]:

```
#renaming the id column of tweet_data_clean as tweet_id
tweet_data_clean=tweet_data_clean.rename(columns={'id':'tweet_id'})
```

In [57]:

```
# changing the data type
tweet_data_clean['tweet_id'] = tweet_data_clean['tweet_id'].astype('str')
```

**TEST**

In [58]:

```
#to check the change in data type and name of column
tweet_data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9749 entries, 0 to 23751
Data columns (total 3 columns):
tweet_id          9749 non-null object
retweet_count     9749 non-null int64
favorite_count    9749 non-null int64
dtypes: int64(2), object(1)
memory usage: 304.7+ KB
```

**DEFINE**

changing data type of columns of image_clean dataframe

**CODE**

In [59]:

```
#information about data types
image_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null int64
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
p1_dog      2075 non-null bool
p2          2075 non-null object
p2_conf     2075 non-null float64
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

In [60]:

```
#changing data type of tweet_id column to string
image_clean['tweet_id'] = image_clean['tweet_id'].astype('str')
```

**TEST**

```
In [61]:
```

```
image_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null object
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
p1_dog      2075 non-null bool
p2          2075 non-null object
p2_conf     2075 non-null float64
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(1), object(5)
memory usage: 152.1+ KB
```

**DEFINE**

merging dataframes tweet_data_clean, image_clean,twitter_clean

**CODE**

```
In [62]:
```

```
#merging dataframes on the basis of tweet_id
master_df = pd.merge(twitter_clean, tweet_data_clean,on='tweet_id', how='in
ner')
master_df= pd.merge(master_df, image_clean,on='tweet_id', how='inner')
```

**TEST**

```
In [63]:
```

```
#final dataframe is master_df
master_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8445 entries, 0 to 8444
Data columns (total 24 columns):
tweet_id                 8445 non-null object
in_reply_to_status_id    8445 non-null object
in_reply_to_user_id      8445 non-null object
timestamp                8445 non-null datetime64[ns]
source                   8445 non-null object
text                     8445 non-null object
expanded_urls            8445 non-null object
rating_numerator         8445 non-null float64
rating_denominator       8445 non-null float64
name                     8445 non-null object
dog_type                 1504 non-null category
retweet_count            8445 non-null int64
favorite_count           8445 non-null int64
```

```
jpg_url                 8445 non-null object
img_num                 8445 non-null int64
p1                      8445 non-null object
p1_conf                 8445 non-null float64
p1_dog                  8445 non-null bool
p2                      8445 non-null object
p2_conf                 8445 non-null float64
p2_dog                  8445 non-null bool
p3                      8445 non-null object
p3_conf                 8445 non-null float64
p3_dog                  8445 non-null bool
dtypes: bool(3), category(1), datetime64[ns](1), float64(5), int64(3), obje
ct(11)
memory usage: 1.4+ MB
```

In [64]:

```
master_df.head()
```

Out[64]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **0** | 892420643555336193 | nan | nan | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |
| **1** | 892420643555336193 | nan | nan | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |
| **2** | 892420643555336193 | nan | nan | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |
| **3** | 892420643555336193 | nan | nan | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |
| **4** | 892420643555336193 | nan | nan | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |

5 rows × 24 columns

## STORING DATA

In [84]:

```
#writing data to twitter_archive_master.csv
master_df.to_csv('twitter_archive_master.csv',index=False)
```

## ANALYZING

In [85]:

```
data=pd.read_csv('twitter_archive_master.csv')
```

In [86]:

```
data.head(3)
```

Out[86]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **0** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |
| **1** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |
| **2** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |

3 rows × 25 columns

In [87]:

```
#making a new column named rating ratio for further analysis

data['rating ratio']=data['rating_numerator']/data['rating_denominator']
```

In [77]:

```
#again displaying the dataframe
data.head()
```

Out[77]:

|  | Unnamed: 0 | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp |  |
|---|---|---|---|---|---|---|
| **0** | 0 | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a hre rel |
| **1** | 1 | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a hre rel |
| **2** | 2 | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a hre rel |
| **3** | 3 | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a hre rel |
| **4** | 4 | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a hre rel |

5 rows × 26 columns

In [88]:

```
#plotting rating ratio
%matplotlib inline
data['rating ratio'].value_counts()
```

Out[88]:

```
1.200000      2252
1.100000      1665
1.300000      1587
1.000000      1434
0.900000      431
0.800000      287
1.400000      233
0.700000      149
0.500000      95
0.600000      91
0.300000      57
0.400000      50
0.200000      32
0.100000      18
```

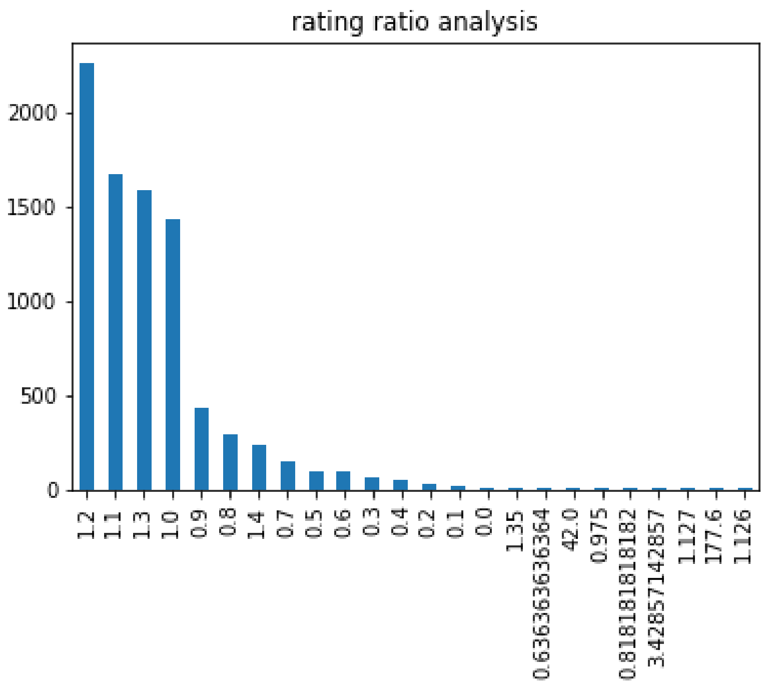
```
0.000000      10
1.350000       9
0.636364       8
42.000000      8
0.975000       8
0.818182       7
3.428571       4
1.127000       4
177.600000     3
1.126000       3
Name: rating ratio, dtype: int64
```

In [89]:

```
data['rating ratio'].value_counts().plot(kind='bar')
plt.title('rating ratio analysis')
```

Out[89]:

```
Text(0.5,1,'rating ratio analysis')
```



This graph gives the insight of how rating ratio differs and which rating ratio is the highest

In [90]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8445 entries, 0 to 8444
Data columns (total 25 columns):
tweet_id                 8445 non-null int64
in_reply_to_status_id    86 non-null float64
in_reply_to_user_id      86 non-null float64
timestamp                8445 non-null object
source                   8445 non-null object
text                     8445 non-null object
expanded_urls            8445 non-null object
rating_numerator         8445 non-null float64
rating_denominator       8445 non-null float64
name                     8445 non-null object
```

```
dog_type                  1504 non-null object
retweet_count             8445 non-null int64
favorite_count            8445 non-null int64
jpg_url                   8445 non-null object
img_num                   8445 non-null int64
p1                        8445 non-null object
p1_conf                   8445 non-null float64
p1_dog                    8445 non-null bool
p2                        8445 non-null object
p2_conf                   8445 non-null float64
p2_dog                    8445 non-null bool
p3                        8445 non-null object
p3_conf                   8445 non-null float64
p3_dog                    8445 non-null bool
rating ratio              8445 non-null float64
dtypes: bool(3), float64(8), int64(4), object(10)
memory usage: 1.4+ MB
```

In [91]:

```
data.dog_type.value_counts()
```

Out[91]:

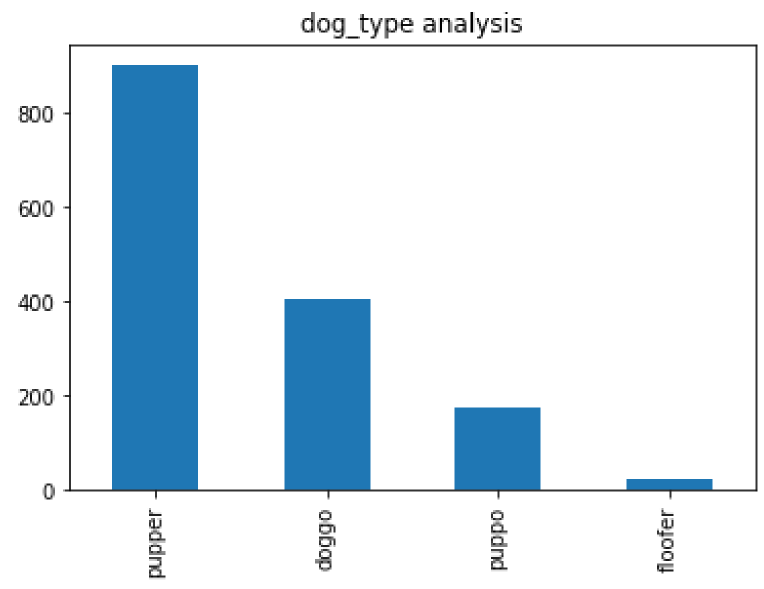
```
pupper     900
doggo      406
puppo      176
floofer     22
Name: dog_type, dtype: int64
```

In [92]:

```
data['dog_type'].value_counts().plot(kind='bar')
plt.title('dog_type analysis')
```

Out[92]:

```
Text(0.5,1,'dog_type analysis')
```



**We observe through this graph pupper is more in number that is pupper is more common dog type**

In [98]:

```
#extracting month
data['timestamp']=pd.to_datetime(data['timestamp'])
data['month'] = data['timestamp'].dt.month
```

In [101]:

```
data.head(3)
```

Out[101]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **0** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |
| **1** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |
| **2** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |

3 rows × 26 columns

In [102]:

```
#extracting year
data['year'] = data['timestamp'].dt.year
```

In [103]:

```
data.head(3)
```

Out[103]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **0** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |
| **1** | | | | 2017-08- | <a |

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| | 892420643555336193 | NaN | NaN | 01 16:23:56 | href="http://twit rel="nofollow"> |
| | | | | | |
| 2 | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 | <a href="http://twit rel="nofollow"> |

3 rows × 27 columns

In [114]:

```
plotting_detail = pd.DataFrame(data.groupby('month')['retweet_count'].count
())
```

In [115]:

```
plotting_detail
```

Out[115]:

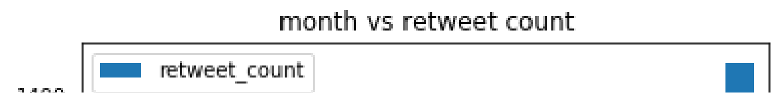
| | retweet_count |
|---|---|
| **month** | |
| **1** | 944 |
| **2** | 803 |
| **3** | 763 |
| **4** | 481 |
| **5** | 561 |
| **6** | 695 |
| **7** | 725 |
| **8** | 318 |
| **9** | 334 |
| **10** | 379 |
| **11** | 927 |
| **12** | 1515 |

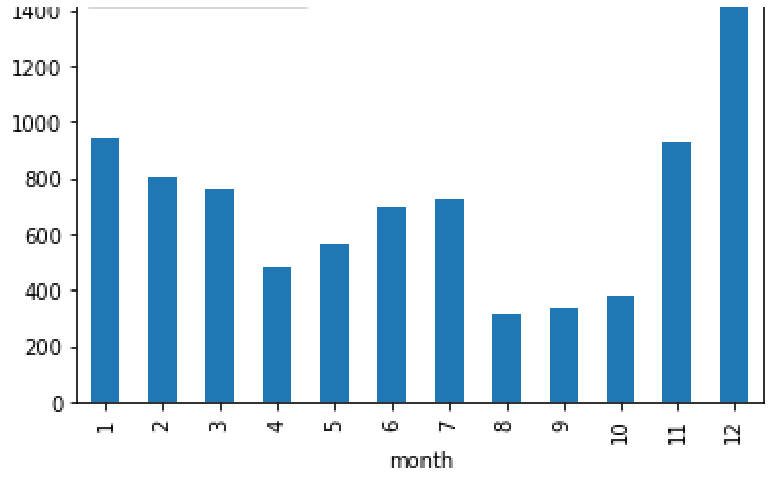
In [119]:

```
plotting_detail.plot(kind='bar',title='month vs retweet count')
```

Out[119]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9837256748>
```





we observed that in 12th month retweet count is maximum

```
In [138]:
```

```
plotting_detail1 = pd.DataFrame(data.groupby('year')['retweet_count'].count
())
```

```
In [139]:
```

```
plotting_detail1
```

```
Out[139]:
```

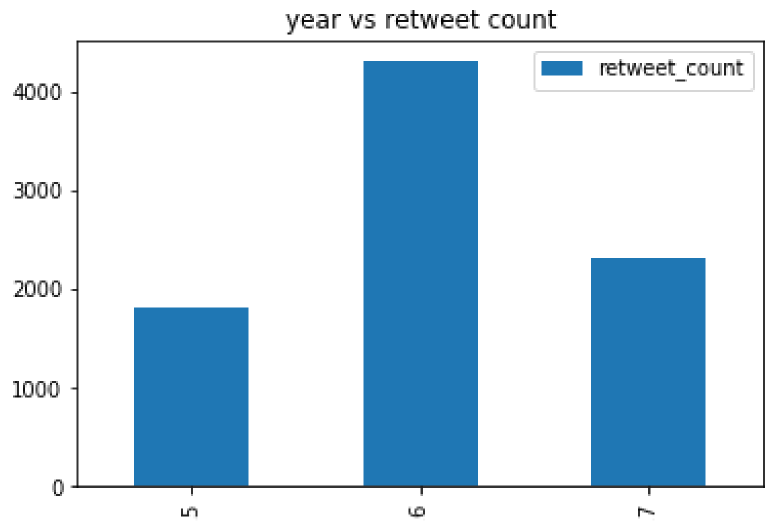
| | retweet_count |
|---|---|
| **year** | |
| **2015** | 1821 |
| **2016** | 4303 |
| **2017** | 2321 |

```
In [127]:
```

```
plotting_detail1.plot(kind='bar',title='year vs retweet count')
```

```
Out[127]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f983d38da58>
```



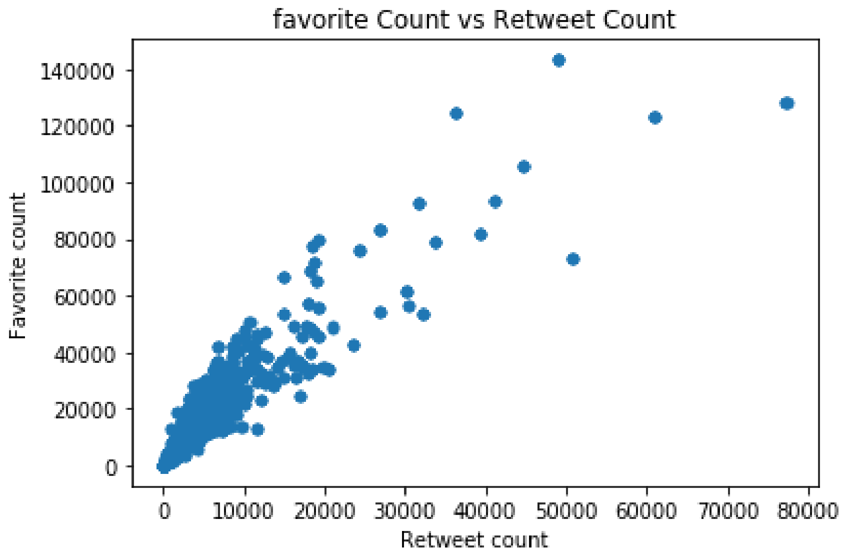
From above bar graph we noticed that 2016 year witnessed maximum retweet count

```
In [133]:
```

```
data.plot(x="retweet_count",y="favorite_count",kind="scatter")
plt.xlabel("Retweet count")
plt.ylabel("Favorite count")
plt.title("favorite Count vs Retweet Count")
plt.figure(figsize=(10,10))
```

```
Out[133]:
```

```
<matplotlib.figure.Figure at 0x7f983da63160>
```



```
<matplotlib.figure.Figure at 0x7f983da63160>
```

above graph shows strong correlation between favorite tweets and retweets

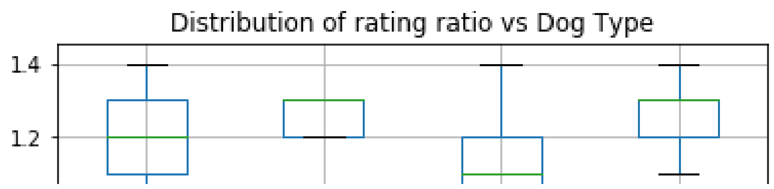
```
In [145]:
```

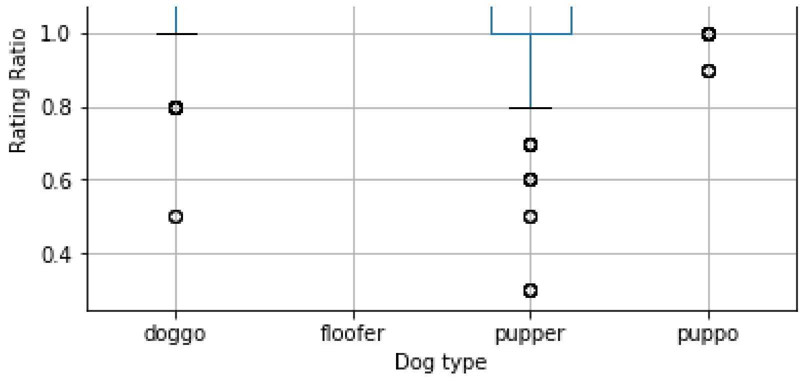
```
data.dog_type.value_counts()
data.boxplot(column='rating ratio', by='dog_type')
plt.xlabel("Dog type")
plt.ylabel("Rating Ratio")
plt.suptitle("")
plt.title("Distribution of rating ratio vs Dog Type")
```

```
/opt/conda/lib/python3.6/site-packages/numpy/core/fromnumeric.py:57: Future
Warning: reshape is deprecated and will raise in a subsequent release. Plea
se use .values.reshape(...) instead
  return getattr(obj, method)(*args, **kwds)
```

```
Out[145]:
```

```
Text(0.5,1,'Distribution of rating ratio vs Dog Type')
```





puppo dog type has highest median rating ratio and pupper has the lowest median rating ratio