wrangle_act

May 4, 2019

1 Importing Packages

```
In [1]: # Importing all necessary packages
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import tweepy
        import json
        import requests

        # to allow plots appear in the jupyter notebook
        % matplotlib inline
```

2 Data Gathering

Twitter Archive Enhanced File

```
In [2]: #Reading a csv file containing we rate dogs account tweets in data frame

        archive_df=pd.read_csv('twitter-archive-enhanced.csv')
        archive_df.head()
```

```
Out[2]:             tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
        0  892420643555336193                    NaN                  NaN
        1  892177421306343426                    NaN                  NaN
        2  891815181378084864                    NaN                  NaN
        3  891689557279858688                    NaN                  NaN
        4  891327558926688256                    NaN                  NaN

                          timestamp  \
        0  2017-08-01 16:23:56 +0000
        1  2017-08-01 00:17:27 +0000
        2  2017-07-31 00:18:03 +0000
        3  2017-07-30 15:58:51 +0000
        4  2017-07-29 16:00:24 +0000
```

```
                                                source  \
0  <a href="http://twitter.com/download/iphone" r...
1  <a href="http://twitter.com/download/iphone" r...
2  <a href="http://twitter.com/download/iphone" r...
3  <a href="http://twitter.com/download/iphone" r...
4  <a href="http://twitter.com/download/iphone" r...

                                                text  retweeted_status_id  \
0  This is Phineas. He's a mystical boy. Only eve...                  NaN
1  This is Tilly. She's just checking pup on you...                  NaN
2  This is Archie. He is a rare Norwegian Pouncin...                  NaN
3  This is Darla. She commenced a snooze mid meal...                  NaN
4  This is Franklin. He would like you to stop ca...                  NaN

   retweeted_status_user_id retweeted_status_timestamp  \
0                       NaN                        NaN
1                       NaN                        NaN
2                       NaN                        NaN
3                       NaN                        NaN
4                       NaN                        NaN

                                       expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/892420643...                13
1  https://twitter.com/dog_rates/status/892177421...                13
2  https://twitter.com/dog_rates/status/891815181...                12
3  https://twitter.com/dog_rates/status/891689557...                13
4  https://twitter.com/dog_rates/status/891327558...                12

   rating_denominator      name doggo floofer pupper puppo
0                  10   Phineas  None    None   None  None
1                  10     Tilly  None    None   None  None
2                  10    Archie  None    None   None  None
3                  10     Darla  None    None   None  None
4                  10  Franklin  None    None   None  None
```

3 Image Prediction File

In [3]:
```
#we'are going to scrap the tsv file containing image predictions and then print out the

link = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predicti

get_response= requests.get(link)

with open('image-predictions.tsv', mode='wb') as f:
    f.write(get_response.content)
```

```
image_prediction_df=pd.read_csv('image-predictions.tsv', sep='\t')

image_prediction_df.head(10)

#Refererence:
#https://realpython.com/python-requests/
```

Out[3]:

	tweet_id	jpg_url
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg
5	666050758794694657	https://pbs.twimg.com/media/CT5Jof1WUAEuVxN.jpg
6	666051853826850816	https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg
7	666055525042405380	https://pbs.twimg.com/media/CT5N9tpXIAAifs1.jpg
8	666057090499244032	https://pbs.twimg.com/media/CT5PY90WoAAQGLo.jpg
9	666058600524156928	https://pbs.twimg.com/media/CT5Qw94XAAA_2dP.jpg

	img_num	p1	p1_conf	p1_dog	p2
0	1	Welsh_springer_spaniel	0.465074	True	collie
1	1	redbone	0.506826	True	miniature_pinscher
2	1	German_shepherd	0.596461	True	malinois
3	1	Rhodesian_ridgeback	0.408143	True	redbone
4	1	miniature_pinscher	0.560311	True	Rottweiler
5	1	Bernese_mountain_dog	0.651137	True	English_springer
6	1	box_turtle	0.933012	False	mud_turtle
7	1	chow	0.692517	True	Tibetan_mastiff
8	1	shopping_cart	0.962465	False	shopping_basket
9	1	miniature_poodle	0.201493	True	komondor

	p2_conf	p2_dog	p3	p3_conf	p3_dog
0	0.156665	True	Shetland_sheepdog	0.061428	True
1	0.074192	True	Rhodesian_ridgeback	0.072010	True
2	0.138584	True	bloodhound	0.116197	True
3	0.360687	True	miniature_pinscher	0.222752	True
4	0.243682	True	Doberman	0.154629	True
5	0.263788	True	Greater_Swiss_Mountain_dog	0.016199	True
6	0.045885	False	terrapin	0.017885	False
7	0.058279	True	fur_coat	0.054449	False
8	0.014594	False	golden_retriever	0.007959	True
9	0.192305	True	soft-coated_wheaten_terrier	0.082086	True

3.1 JSON File

```
In [4]: #reading tweet_json file and create a data frame with id, favourite and retweet counts
        #https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_json.html
```

```
df= pd.read_json('tweet-json.txt', orient='records', lines=True)

tweet_counts=df[['favorite_count', 'retweet_count', 'id']]

tweet_counts.tail(10)
```

Out[4]:

	favorite_count	retweet_count	id
2344	115	61	666058600524156928
2345	304	146	666057090499244032
2346	448	261	666055525042405380
2347	1253	879	666051853826850816
2348	136	60	666050758794694657
2349	111	41	666049248165822465
2350	311	147	666044226329800704
2351	128	47	666033412701032449
2352	132	48	666029285002620928
2353	2535	532	666020888022790149

4 Assessing Data

5 Part 1

- First of all, we will assess data **visually** by printing out the dataset and uncover any issues in archive_df, image_prediction_df, tweet_counts

In [5]:
```
archive_df
```

Out[5]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	\
0	892420643555336193	NaN	NaN	
1	892177421306343426	NaN	NaN	
2	891815181378084864	NaN	NaN	
3	891689557279858688	NaN	NaN	
4	891327558926688256	NaN	NaN	
5	891087950875897856	NaN	NaN	
6	890971913173991426	NaN	NaN	
7	890729181411237888	NaN	NaN	
8	890609185150312448	NaN	NaN	
9	890240255349198849	NaN	NaN	
10	890006608113172480	NaN	NaN	
11	889880896479866881	NaN	NaN	
12	889665388333682689	NaN	NaN	
13	889638837579907072	NaN	NaN	
14	889531135344209921	NaN	NaN	
15	889278841981685760	NaN	NaN	
16	888917238123831296	NaN	NaN	
17	888804989199671297	NaN	NaN	
18	888554962724278272	NaN	NaN	
19	888202515573088257	NaN	NaN	

20	888078434458587136	NaN	NaN
21	887705289381826560	NaN	NaN
22	887517139158093824	NaN	NaN
23	887473957103951883	NaN	NaN
24	887343217045368832	NaN	NaN
25	887101392804085760	NaN	NaN
26	886983233522544640	NaN	NaN
27	886736880519319552	NaN	NaN
28	886680336477933568	NaN	NaN
29	886366144734445568	NaN	NaN
...	...	...	...
2326	666411507551481857	NaN	NaN
2327	666407126856765440	NaN	NaN
2328	666396247373291520	NaN	NaN
2329	666373753744588802	NaN	NaN
2330	666362758909284353	NaN	NaN
2331	666353288456101888	NaN	NaN
2332	666345417576210432	NaN	NaN
2333	666337882303524864	NaN	NaN
2334	666293911632134144	NaN	NaN
2335	666287406224695296	NaN	NaN
2336	666273097616637952	NaN	NaN
2337	666268910803644416	NaN	NaN
2338	666104133288665088	NaN	NaN
2339	666102155909144576	NaN	NaN
2340	666099513787052032	NaN	NaN
2341	666094000022159362	NaN	NaN
2342	666082916733198337	NaN	NaN
2343	666073100786774016	NaN	NaN
2344	666071193221509120	NaN	NaN
2345	666063827256086533	NaN	NaN
2346	666058600524156928	NaN	NaN
2347	666057090499244032	NaN	NaN
2348	666055525042405380	NaN	NaN
2349	666051853826850816	NaN	NaN
2350	666050758794694657	NaN	NaN
2351	666049248165822465	NaN	NaN
2352	666044226329800704	NaN	NaN
2353	666033412701032449	NaN	NaN
2354	666029285002620928	NaN	NaN
2355	666020888022790149	NaN	NaN

	timestamp	\
0	2017-08-01 16:23:56 +0000	
1	2017-08-01 00:17:27 +0000	
2	2017-07-31 00:18:03 +0000	
3	2017-07-30 15:58:51 +0000	
4	2017-07-29 16:00:24 +0000	

```
5     2017-07-29 00:08:17 +0000
6     2017-07-28 16:27:12 +0000
7     2017-07-28 00:22:40 +0000
8     2017-07-27 16:25:51 +0000
9     2017-07-26 15:59:51 +0000
10    2017-07-26 00:31:25 +0000
11    2017-07-25 16:11:53 +0000
12    2017-07-25 01:55:32 +0000
13    2017-07-25 00:10:02 +0000
14    2017-07-24 17:02:04 +0000
15    2017-07-24 00:19:32 +0000
16    2017-07-23 00:22:39 +0000
17    2017-07-22 16:56:37 +0000
18    2017-07-22 00:23:06 +0000
19    2017-07-21 01:02:36 +0000
20    2017-07-20 16:49:33 +0000
21    2017-07-19 16:06:48 +0000
22    2017-07-19 03:39:09 +0000
23    2017-07-19 00:47:34 +0000
24    2017-07-18 16:08:03 +0000
25    2017-07-18 00:07:08 +0000
26    2017-07-17 16:17:36 +0000
27    2017-07-16 23:58:41 +0000
28    2017-07-16 20:14:00 +0000
29    2017-07-15 23:25:31 +0000
...                          ...
2326  2015-11-17 00:24:19 +0000
2327  2015-11-17 00:06:54 +0000
2328  2015-11-16 23:23:41 +0000
2329  2015-11-16 21:54:18 +0000
2330  2015-11-16 21:10:36 +0000
2331  2015-11-16 20:32:58 +0000
2332  2015-11-16 20:01:42 +0000
2333  2015-11-16 19:31:45 +0000
2334  2015-11-16 16:37:02 +0000
2335  2015-11-16 16:11:11 +0000
2336  2015-11-16 15:14:19 +0000
2337  2015-11-16 14:57:41 +0000
2338  2015-11-16 04:02:55 +0000
2339  2015-11-16 03:55:04 +0000
2340  2015-11-16 03:44:34 +0000
2341  2015-11-16 03:22:39 +0000
2342  2015-11-16 02:38:37 +0000
2343  2015-11-16 01:59:36 +0000
2344  2015-11-16 01:52:02 +0000
2345  2015-11-16 01:22:45 +0000
2346  2015-11-16 01:01:59 +0000
2347  2015-11-16 00:55:59 +0000
```

```
2348  2015-11-16 00:49:46 +0000
2349  2015-11-16 00:35:11 +0000
2350  2015-11-16 00:30:50 +0000
2351  2015-11-16 00:24:50 +0000
2352  2015-11-16 00:04:52 +0000
2353  2015-11-15 23:21:54 +0000
2354  2015-11-15 23:05:30 +0000
2355  2015-11-15 22:32:08 +0000

                                                 source  \
0     <a href="http://twitter.com/download/iphone" r...
1     <a href="http://twitter.com/download/iphone" r...
2     <a href="http://twitter.com/download/iphone" r...
3     <a href="http://twitter.com/download/iphone" r...
4     <a href="http://twitter.com/download/iphone" r...
5     <a href="http://twitter.com/download/iphone" r...
6     <a href="http://twitter.com/download/iphone" r...
7     <a href="http://twitter.com/download/iphone" r...
8     <a href="http://twitter.com/download/iphone" r...
9     <a href="http://twitter.com/download/iphone" r...
10    <a href="http://twitter.com/download/iphone" r...
11    <a href="http://twitter.com/download/iphone" r...
12    <a href="http://twitter.com/download/iphone" r...
13    <a href="http://twitter.com/download/iphone" r...
14    <a href="http://twitter.com/download/iphone" r...
15    <a href="http://twitter.com/download/iphone" r...
16    <a href="http://twitter.com/download/iphone" r...
17    <a href="http://twitter.com/download/iphone" r...
18    <a href="http://twitter.com/download/iphone" r...
19    <a href="http://twitter.com/download/iphone" r...
20    <a href="http://twitter.com/download/iphone" r...
21    <a href="http://twitter.com/download/iphone" r...
22    <a href="http://twitter.com/download/iphone" r...
23    <a href="http://twitter.com/download/iphone" r...
24    <a href="http://twitter.com/download/iphone" r...
25    <a href="http://twitter.com/download/iphone" r...
26    <a href="http://twitter.com/download/iphone" r...
27    <a href="http://twitter.com/download/iphone" r...
28    <a href="http://twitter.com/download/iphone" r...
29    <a href="http://twitter.com/download/iphone" r...
...                                                 ...
2326  <a href="http://twitter.com/download/iphone" r...
2327  <a href="http://twitter.com/download/iphone" r...
2328  <a href="http://twitter.com/download/iphone" r...
2329  <a href="http://twitter.com/download/iphone" r...
2330  <a href="http://twitter.com/download/iphone" r...
2331  <a href="http://twitter.com/download/iphone" r...
2332  <a href="http://twitter.com/download/iphone" r...
```

```
2333  <a href="http://twitter.com/download/iphone" r...
2334  <a href="http://twitter.com/download/iphone" r...
2335  <a href="http://twitter.com/download/iphone" r...
2336  <a href="http://twitter.com/download/iphone" r...
2337  <a href="http://twitter.com/download/iphone" r...
2338  <a href="http://twitter.com/download/iphone" r...
2339  <a href="http://twitter.com/download/iphone" r...
2340  <a href="http://twitter.com/download/iphone" r...
2341  <a href="http://twitter.com/download/iphone" r...
2342  <a href="http://twitter.com/download/iphone" r...
2343  <a href="http://twitter.com/download/iphone" r...
2344  <a href="http://twitter.com/download/iphone" r...
2345  <a href="http://twitter.com/download/iphone" r...
2346  <a href="http://twitter.com/download/iphone" r...
2347  <a href="http://twitter.com/download/iphone" r...
2348  <a href="http://twitter.com/download/iphone" r...
2349  <a href="http://twitter.com/download/iphone" r...
2350  <a href="http://twitter.com/download/iphone" r...
2351  <a href="http://twitter.com/download/iphone" r...
2352  <a href="http://twitter.com/download/iphone" r...
2353  <a href="http://twitter.com/download/iphone" r...
2354  <a href="http://twitter.com/download/iphone" r...
2355  <a href="http://twitter.com/download/iphone" r...

                                                 text  retweeted_status_id  \
0     This is Phineas. He's a mystical boy. Only eve...                  NaN
1     This is Tilly. She's just checking pup on you...                  NaN
2     This is Archie. He is a rare Norwegian Pouncin...                  NaN
3     This is Darla. She commenced a snooze mid meal...                  NaN
4     This is Franklin. He would like you to stop ca...                  NaN
5     Here we have a majestic great white breaching ...                  NaN
6     Meet Jax. He enjoys ice cream so much he gets ...                  NaN
7     When you watch your owner call another dog a g...                  NaN
8     This is Zoey. She doesn't want to be one of th...                  NaN
9     This is Cassie. She is a college pup. Studying...                  NaN
10    This is Koda. He is a South Australian decksha...                  NaN
11    This is Bruno. He is a service shark. Only get...                  NaN
12    Here's a puppo that seems to be on the fence a...                  NaN
13    This is Ted. He does his best. Sometimes that'...                  NaN
14    This is Stuart. He's sporting his favorite fan...                  NaN
15    This is Oliver. You're witnessing one of his m...                  NaN
16    This is Jim. He found a fren. Taught him how t...                  NaN
17    This is Zeke. He has a new stick. Very proud o...                  NaN
18    This is Ralphus. He's powering up. Attempting ...                  NaN
19    RT @dog_rates: This is Canela. She attempted s...         8.874740e+17
20    This is Gerald. He was just told he didn't get...                  NaN
21    This is Jeffrey. He has a monopoly on the pool...                  NaN
22    I've yet to rate a Venezuelan Hover Wiener. Th...                  NaN
```

```
23    This is Canela. She attempted some fancy porch...                  NaN
24    You may not have known you needed to see this ...                  NaN
25    This... is a Jubilant Antarctic House Bear. We...                  NaN
26    This is Maya. She's very shy. Rarely leaves he...                  NaN
27    This is Mingus. He's a wonderful father to his...                  NaN
28    This is Derek. He's late for a dog meeting. 13...                  NaN
29    This is Roscoe. Another pupper fallen victim t...                  NaN
...                                                 ...                  ...
2326  This is quite the dog. Gets really excited whe...                  NaN
2327  This is a southern Vesuvius bumblegruff. Can d...                  NaN
2328  Oh goodness. A super rare northeast Qdoba kang...                  NaN
2329  Those are sunglasses and a jean jacket. 11/10 ...                  NaN
2330  Unique dog here. Very small. Lives in containe...                  NaN
2331  Here we have a mixed Asiago from the Galápagos...                  NaN
2332  Look at this jokester thinking seat belt laws ...                  NaN
2333  This is an extremely rare horned Parthenon. No...                  NaN
2334  This is a funny dog. Weird toes. Won't come do...                  NaN
2335  This is an Albanian 3 1/2 legged  Episcopalian...                  NaN
2336     Can take selfies 11/10 https://t.co/ws2AMaNwPW                  NaN
2337  Very concerned about fellow dog trapped in com...                  NaN
2338  Not familiar with this breed. No tail (weird)...                  NaN
2339  Oh my. Here you are seeing an Adobe Setter giv...                  NaN
2340  Can stand on stump for what seems like a while...                  NaN
2341  This appears to be a Mongolian Presbyterian mi...                  NaN
2342  Here we have a well-established sunblockerspan...                  NaN
2343  Let's hope this flight isn't Malaysian (lol). ...                  NaN
2344  Here we have a northern speckled Rhododendron...                   NaN
2345  This is the happiest dog you will ever see. Ve...                  NaN
2346  Here is the Rand Paul of retrievers folks! He'...                  NaN
2347  My oh my. This is a rare blond Canadian terrie...                  NaN
2348  Here is a Siberian heavily armored polar bear ...                  NaN
2349  This is an odd dog. Hard on the outside but lo...                  NaN
2350  This is a truly beautiful English Wilson Staff...                  NaN
2351  Here we have a 1949 1st generation vulpix. Enj...                  NaN
2352  This is a purebred Piers Morgan. Loves to Netf...                  NaN
2353  Here is a very happy pup. Big fan of well-main...                  NaN
2354  This is a western brown Mitsubishi terrier. Up...                  NaN
2355  Here we have a Japanese Irish Setter. Lost eye...                  NaN

      retweeted_status_user_id retweeted_status_timestamp  \
0                          NaN                         NaN
1                          NaN                         NaN
2                          NaN                         NaN
3                          NaN                         NaN
4                          NaN                         NaN
5                          NaN                         NaN
6                          NaN                         NaN
7                          NaN                         NaN
```

8	NaN	NaN
9	NaN	NaN
10	NaN	NaN
11	NaN	NaN
12	NaN	NaN
13	NaN	NaN
14	NaN	NaN
15	NaN	NaN
16	NaN	NaN
17	NaN	NaN
18	NaN	NaN
19	4.196984e+09	2017-07-19 00:47:34 +0000
20	NaN	NaN
21	NaN	NaN
22	NaN	NaN
23	NaN	NaN
24	NaN	NaN
25	NaN	NaN
26	NaN	NaN
27	NaN	NaN
28	NaN	NaN
29	NaN	NaN
...	...	...
2326	NaN	NaN
2327	NaN	NaN
2328	NaN	NaN
2329	NaN	NaN
2330	NaN	NaN
2331	NaN	NaN
2332	NaN	NaN
2333	NaN	NaN
2334	NaN	NaN
2335	NaN	NaN
2336	NaN	NaN
2337	NaN	NaN
2338	NaN	NaN
2339	NaN	NaN
2340	NaN	NaN
2341	NaN	NaN
2342	NaN	NaN
2343	NaN	NaN
2344	NaN	NaN
2345	NaN	NaN
2346	NaN	NaN
2347	NaN	NaN
2348	NaN	NaN
2349	NaN	NaN
2350	NaN	NaN

```
2351                    NaN                    NaN
2352                    NaN                    NaN
2353                    NaN                    NaN
2354                    NaN                    NaN
2355                    NaN                    NaN

                                          expanded_urls  rating_numerator  \
0     https://twitter.com/dog_rates/status/892420643...                13
1     https://twitter.com/dog_rates/status/892177421...                13
2     https://twitter.com/dog_rates/status/891815181...                12
3     https://twitter.com/dog_rates/status/891689557...                13
4     https://twitter.com/dog_rates/status/891327558...                12
5     https://twitter.com/dog_rates/status/891087950...                13
6     https://gofundme.com/ydvmve-surgery-for-jax,ht...                13
7     https://twitter.com/dog_rates/status/890729181...                13
8     https://twitter.com/dog_rates/status/890609185...                13
9     https://twitter.com/dog_rates/status/890240255...                14
10    https://twitter.com/dog_rates/status/890006608...                13
11    https://twitter.com/dog_rates/status/889880896...                13
12    https://twitter.com/dog_rates/status/889665388...                13
13    https://twitter.com/dog_rates/status/889638837...                12
14    https://twitter.com/dog_rates/status/889531135...                13
15    https://twitter.com/dog_rates/status/889278841...                13
16    https://twitter.com/dog_rates/status/888917238...                12
17    https://twitter.com/dog_rates/status/888804989...                13
18    https://twitter.com/dog_rates/status/888554962...                13
19    https://twitter.com/dog_rates/status/887473957...                13
20    https://twitter.com/dog_rates/status/888078434...                12
21    https://twitter.com/dog_rates/status/887705289...                13
22    https://twitter.com/dog_rates/status/887517139...                14
23    https://twitter.com/dog_rates/status/887473957...                13
24    https://twitter.com/dog_rates/status/887343217...                13
25    https://twitter.com/dog_rates/status/887101392...                12
26    https://twitter.com/dog_rates/status/886983233...                13
27    https://www.gofundme.com/mingusneedsus,https:/...                13
28    https://twitter.com/dog_rates/status/886680336...                13
29    https://twitter.com/dog_rates/status/886366144...                12
...                                                 ...               ...
2326  https://twitter.com/dog_rates/status/666411507...                 2
2327  https://twitter.com/dog_rates/status/666407126...                 7
2328  https://twitter.com/dog_rates/status/666396247...                 9
2329  https://twitter.com/dog_rates/status/666373753...                11
2330  https://twitter.com/dog_rates/status/666362758...                 6
2331  https://twitter.com/dog_rates/status/666353288...                 8
2332  https://twitter.com/dog_rates/status/666345417...                10
2333  https://twitter.com/dog_rates/status/666337882...                 9
2334  https://twitter.com/dog_rates/status/666293911...                 3
2335  https://twitter.com/dog_rates/status/666287406...                 1
```

```
2336  https://twitter.com/dog_rates/status/666273097...                11
2337  https://twitter.com/dog_rates/status/666268910...                10
2338  https://twitter.com/dog_rates/status/666104133...                 1
2339  https://twitter.com/dog_rates/status/666102155...                11
2340  https://twitter.com/dog_rates/status/666099513...                 8
2341  https://twitter.com/dog_rates/status/666094000...                 9
2342  https://twitter.com/dog_rates/status/666082916...                 6
2343  https://twitter.com/dog_rates/status/666073100...                10
2344  https://twitter.com/dog_rates/status/666071193...                 9
2345  https://twitter.com/dog_rates/status/666063827...                10
2346  https://twitter.com/dog_rates/status/666058600...                 8
2347  https://twitter.com/dog_rates/status/666057090...                 9
2348  https://twitter.com/dog_rates/status/666055525...                10
2349  https://twitter.com/dog_rates/status/666051853...                 2
2350  https://twitter.com/dog_rates/status/666050758...                10
2351  https://twitter.com/dog_rates/status/666049248...                 5
2352  https://twitter.com/dog_rates/status/666044226...                 6
2353  https://twitter.com/dog_rates/status/666033412...                 9
2354  https://twitter.com/dog_rates/status/666029285...                 7
2355  https://twitter.com/dog_rates/status/666020888...                 8

      rating_denominator      name  doggo floofer  pupper  puppo
0                     10   Phineas   None    None    None   None
1                     10     Tilly   None    None    None   None
2                     10    Archie   None    None    None   None
3                     10     Darla   None    None    None   None
4                     10  Franklin   None    None    None   None
5                     10      None   None    None    None   None
6                     10       Jax   None    None    None   None
7                     10      None   None    None    None   None
8                     10      Zoey   None    None    None   None
9                     10    Cassie  doggo    None    None   None
10                    10      Koda   None    None    None   None
11                    10     Bruno   None    None    None   None
12                    10      None   None    None    None  puppo
13                    10       Ted   None    None    None   None
14                    10    Stuart   None    None    None  puppo
15                    10    Oliver   None    None    None   None
16                    10       Jim   None    None    None   None
17                    10      Zeke   None    None    None   None
18                    10   Ralphus   None    None    None   None
19                    10    Canela   None    None    None   None
20                    10    Gerald   None    None    None   None
21                    10   Jeffrey   None    None    None   None
22                    10      such   None    None    None   None
23                    10    Canela   None    None    None   None
24                    10      None   None    None    None   None
25                    10      None   None    None    None   None
```

```
26                   10      Maya   None    None    None   None
27                   10    Mingus   None    None    None   None
28                   10     Derek   None    None    None   None
29                   10    Roscoe   None    None  pupper   None
...                 ...       ...    ...     ...     ...    ...
2326                 10     quite   None    None    None   None
2327                 10         a   None    None    None   None
2328                 10      None   None    None    None   None
2329                 10      None   None    None    None   None
2330                 10      None   None    None    None   None
2331                 10      None   None    None    None   None
2332                 10      None   None    None    None   None
2333                 10        an   None    None    None   None
2334                 10         a   None    None    None   None
2335                  2        an   None    None    None   None
2336                 10      None   None    None    None   None
2337                 10      None   None    None    None   None
2338                 10      None   None    None    None   None
2339                 10      None   None    None    None   None
2340                 10      None   None    None    None   None
2341                 10      None   None    None    None   None
2342                 10      None   None    None    None   None
2343                 10      None   None    None    None   None
2344                 10      None   None    None    None   None
2345                 10       the   None    None    None   None
2346                 10       the   None    None    None   None
2347                 10         a   None    None    None   None
2348                 10         a   None    None    None   None
2349                 10        an   None    None    None   None
2350                 10         a   None    None    None   None
2351                 10      None   None    None    None   None
2352                 10         a   None    None    None   None
2353                 10         a   None    None    None   None
2354                 10         a   None    None    None   None
2355                 10      None   None    None    None   None

[2356 rows x 17 columns]
```

```
In [6]:  image_prediction_df
```

```
Out[6]:                tweet_id                                          jpg_url  \
0     666020888022790149    https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
1     666029285002620928    https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
2     666033412701032449    https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
3     666044226329800704    https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
4     666049248165822465    https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg
5     666050758794694657    https://pbs.twimg.com/media/CT5Jof1WUAEuVxN.jpg
6     666051853826850816    https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg
```

7	666055525042405380	https://pbs.twimg.com/media/CT5N9tpXIAAifs1.jpg
8	666057090499244032	https://pbs.twimg.com/media/CT5PY9OWoAAQGLo.jpg
9	666058600524156928	https://pbs.twimg.com/media/CT5Qw94XAAA_2dP.jpg
10	666063827256086533	https://pbs.twimg.com/media/CT5Vg_wXIAAXfnj.jpg
11	666071193221509120	https://pbs.twimg.com/media/CT5cN_3WEAAlOoZ.jpg
12	666073100786774016	https://pbs.twimg.com/media/CT5d9DZXAAALcwe.jpg
13	666082916733198337	https://pbs.twimg.com/media/CT5m4VGWEAAtKc8.jpg
14	666094000022159362	https://pbs.twimg.com/media/CT5w9gUW4AAsBNN.jpg
15	666099513787052032	https://pbs.twimg.com/media/CT51-JJUEAA6hV8.jpg
16	666102155909144576	https://pbs.twimg.com/media/CT54YGiWUAEZnoK.jpg
17	666104133288665088	https://pbs.twimg.com/media/CT56LSZWoAAlJj2.jpg
18	666268910803644416	https://pbs.twimg.com/media/CT8QCd1WEAADXws.jpg
19	666273097616637952	https://pbs.twimg.com/media/CT8T1mtUwAA3aqm.jpg
20	666287406224695296	https://pbs.twimg.com/media/CT8g3BpUEAAuFjg.jpg
21	666293911632134144	https://pbs.twimg.com/media/CT8mx7KW4AEQu8N.jpg
22	666337882303524864	https://pbs.twimg.com/media/CT9OwFIWEAMuRje.jpg
23	666345417576210432	https://pbs.twimg.com/media/CT9Vn7PWoAA_ZCM.jpg
24	666353288456101888	https://pbs.twimg.com/media/CT9cx0tUEAAhNN_.jpg
25	666362758909284353	https://pbs.twimg.com/media/CT9lXGsUcAAyUFt.jpg
26	666373753744588802	https://pbs.twimg.com/media/CT9vZEYWUAAlZO5.jpg
27	666396247373291520	https://pbs.twimg.com/media/CT-D2ZHWIAA3gK1.jpg
28	666407126856765440	https://pbs.twimg.com/media/CT-NvwmW4AAugGZ.jpg
29	666411507551481857	https://pbs.twimg.com/media/CT-RugiWIAELEaq.jpg
...	...	...
2045	886366144734445568	https://pbs.twimg.com/media/DE0BTnQUwAApKEH.jpg
2046	886680336477933568	https://pbs.twimg.com/media/DE4fEDzWAAAyHMM.jpg
2047	886736880519319552	https://pbs.twimg.com/media/DE5Se8FXcAAJFx4.jpg
2048	886983233522544640	https://pbs.twimg.com/media/DE8yicJW0AAAvBJ.jpg
2049	887101392804085760	https://pbs.twimg.com/media/DE-eAq6UwAA-jaE.jpg
2050	887343217045368832	https://pbs.twimg.com/ext_tw_video_thumb/88734...
2051	887473957103951883	https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg
2052	887517139158093824	https://pbs.twimg.com/ext_tw_video_thumb/88751...
2053	887705289381826560	https://pbs.twimg.com/media/DFHDQBbXgAEqY7t.jpg
2054	888078434458587136	https://pbs.twimg.com/media/DFMWn56WsAAkA7B.jpg
2055	888202515573088257	https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg
2056	888554962724278272	https://pbs.twimg.com/media/DFTH_O-UQAACu2O.jpg
2057	888804989199671297	https://pbs.twimg.com/media/DFWra-3VYAA2piG.jpg
2058	888917238123831296	https://pbs.twimg.com/media/DFYRgsOUQAARGhO.jpg
2059	889278841981685760	https://pbs.twimg.com/ext_tw_video_thumb/88927...
2060	889531135344209921	https://pbs.twimg.com/media/DFg_2PVW0AEHN3p.jpg
2061	889638837579907072	https://pbs.twimg.com/media/DFihzFfXsAYGDPR.jpg
2062	889665388333682689	https://pbs.twimg.com/media/DFi579UWsAAatzw.jpg
2063	889880896479866881	https://pbs.twimg.com/media/DFl99B1WsAITKsg.jpg
2064	890006608113172480	https://pbs.twimg.com/media/DFnwSY4WAAAMliS.jpg
2065	890240255349198849	https://pbs.twimg.com/media/DFrEyVuW0AAO3t9.jpg
2066	890609185150312448	https://pbs.twimg.com/media/DFwUU__XcAEpyXI.jpg
2067	890729181411237888	https://pbs.twimg.com/media/DFyBahAVwAAhUTd.jpg
2068	890971913173991426	https://pbs.twimg.com/media/DF1eOmZXUAALUcq.jpg

```
2069  891087950875897856    https://pbs.twimg.com/media/DF3HwyEWsAABqE6.jpg
2070  891327558926688256    https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg
2071  891689557279858688    https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg
2072  891815181378084864    https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg
2073  892177421306343426    https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg
2074  892420643555336193    https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg

      img_num                          p1   p1_conf  p1_dog  \
0           1       Welsh_springer_spaniel  0.465074    True
1           1                     redbone  0.506826    True
2           1             German_shepherd  0.596461    True
3           1          Rhodesian_ridgeback  0.408143    True
4           1           miniature_pinscher  0.560311    True
5           1         Bernese_mountain_dog  0.651137    True
6           1                   box_turtle  0.933012   False
7           1                         chow  0.692517    True
8           1                shopping_cart  0.962465   False
9           1             miniature_poodle  0.201493    True
10          1             golden_retriever  0.775930    True
11          1                Gordon_setter  0.503672    True
12          1                 Walker_hound  0.260857    True
13          1                          pug  0.489814    True
14          1                   bloodhound  0.195217    True
15          1                        Lhasa  0.582330    True
16          1               English_setter  0.298617    True
17          1                          hen  0.965932   False
18          1             desktop_computer  0.086502   False
19          1            Italian_greyhound  0.176053    True
20          1                 Maltese_dog  0.857531    True
21          1             three-toed_sloth  0.914671   False
22          1                           ox  0.416669   False
23          1             golden_retriever  0.858744    True
24          1                     malamute  0.336874    True
25          1                   guinea_pig  0.996496   False
26          1  soft-coated_wheaten_terrier  0.326467    True
27          1                    Chihuahua  0.978108    True
28          1      black-and-tan_coonhound  0.529139    True
29          1                         coho  0.404640   False
...       ...                          ...       ...     ...
2045        1               French_bulldog  0.999201    True
2046        1                  convertible  0.738995   False
2047        1                       kuvasz  0.309706    True
2048        2                    Chihuahua  0.793469    True
2049        1                      Samoyed  0.733942    True
2050        1             Mexican_hairless  0.330741    True
2051        2                     Pembroke  0.809197    True
2052        1                    limousine  0.130432   False
2053        1                       basset  0.821664    True
```

2054	1	French_bulldog	0.995026	True
2055	2	Pembroke	0.809197	True
2056	3	Siberian_husky	0.700377	True
2057	1	golden_retriever	0.469760	True
2058	1	golden_retriever	0.714719	True
2059	1	whippet	0.626152	True
2060	1	golden_retriever	0.953442	True
2061	1	French_bulldog	0.991650	True
2062	1	Pembroke	0.966327	True
2063	1	French_bulldog	0.377417	True
2064	1	Samoyed	0.957979	True
2065	1	Pembroke	0.511319	True
2066	1	Irish_terrier	0.487574	True
2067	2	Pomeranian	0.566142	True
2068	1	Appenzeller	0.341703	True
2069	1	Chesapeake_Bay_retriever	0.425595	True
2070	2	basset	0.555712	True
2071	1	paper_towel	0.170278	False
2072	1	Chihuahua	0.716012	True
2073	1	Chihuahua	0.323581	True
2074	1	orange	0.097049	False

	p2	p2_conf	p2_dog	p3	\
0	collie	0.156665	True	Shetland_sheepdog	
1	miniature_pinscher	0.074192	True	Rhodesian_ridgeback	
2	malinois	0.138584	True	bloodhound	
3	redbone	0.360687	True	miniature_pinscher	
4	Rottweiler	0.243682	True	Doberman	
5	English_springer	0.263788	True	Greater_Swiss_Mountain_dog	
6	mud_turtle	0.045885	False	terrapin	
7	Tibetan_mastiff	0.058279	True	fur_coat	
8	shopping_basket	0.014594	False	golden_retriever	
9	komondor	0.192305	True	soft-coated_wheaten_terrier	
10	Tibetan_mastiff	0.093718	True	Labrador_retriever	
11	Yorkshire_terrier	0.174201	True	Pekinese	
12	English_foxhound	0.175382	True	Ibizan_hound	
13	bull_mastiff	0.404722	True	French_bulldog	
14	German_shepherd	0.078260	True	malinois	
15	Shih-Tzu	0.166192	True	Dandie_Dinmont	
16	Newfoundland	0.149842	True	borzoi	
17	cock	0.033919	False	partridge	
18	desk	0.085547	False	bookcase	
19	toy_terrier	0.111884	True	basenji	
20	toy_poodle	0.063064	True	miniature_poodle	
21	otter	0.015250	False	great_grey_owl	
22	Newfoundland	0.278407	True	groenendael	
23	Chesapeake_Bay_retriever	0.054787	True	Labrador_retriever	
24	Siberian_husky	0.147655	True	Eskimo_dog	

25	skunk	0.002402	False	hamster
26	Afghan_hound	0.259551	True	briard
27	toy_terrier	0.009397	True	papillon
28	bloodhound	0.244220	True	flat-coated_retriever
29	barracouta	0.271485	False	gar
...	...	...	...	...
2045	Chihuahua	0.000361	True	Boston_bull
2046	sports_car	0.139952	False	car_wheel
2047	Great_Pyrenees	0.186136	True	Dandie_Dinmont
2048	toy_terrier	0.143528	True	can_opener
2049	Eskimo_dog	0.035029	True	Staffordshire_bullterrier
2050	sea_lion	0.275645	False	Weimaraner
2051	Rhodesian_ridgeback	0.054950	True	beagle
2052	tow_truck	0.029175	False	shopping_cart
2053	redbone	0.087582	True	Weimaraner
2054	pug	0.000932	True	bull_mastiff
2055	Rhodesian_ridgeback	0.054950	True	beagle
2056	Eskimo_dog	0.166511	True	malamute
2057	Labrador_retriever	0.184172	True	English_setter
2058	Tibetan_mastiff	0.120184	True	Labrador_retriever
2059	borzoi	0.194742	True	Saluki
2060	Labrador_retriever	0.013834	True	redbone
2061	boxer	0.002129	True	Staffordshire_bullterrier
2062	Cardigan	0.027356	True	basenji
2063	Labrador_retriever	0.151317	True	muzzle
2064	Pomeranian	0.013884	True	chow
2065	Cardigan	0.451038	True	Chihuahua
2066	Irish_setter	0.193054	True	Chesapeake_Bay_retriever
2067	Eskimo_dog	0.178406	True	Pembroke
2068	Border_collie	0.199287	True	ice_lolly
2069	Irish_terrier	0.116317	True	Indian_elephant
2070	English_springer	0.225770	True	German_short-haired_pointer
2071	Labrador_retriever	0.168086	True	spatula
2072	malamute	0.078253	True	kelpie
2073	Pekinese	0.090647	True	papillon
2074	bagel	0.085851	False	banana

	p3_conf	p3_dog
0	0.061428	True
1	0.072010	True
2	0.116197	True
3	0.222752	True
4	0.154629	True
5	0.016199	True
6	0.017885	False
7	0.054449	False
8	0.007959	True
9	0.082086	True

```
10    0.072427    True
11    0.109454    True
12    0.097471    True
13    0.048960    True
14    0.075628    True
15    0.089688    True
16    0.133649    True
17    0.000052   False
18    0.079480   False
19    0.111152    True
20    0.025581    True
21    0.013207   False
22    0.102643    True
23    0.014241    True
24    0.093412    True
25    0.000461   False
26    0.206803    True
27    0.004577    True
28    0.173810    True
29    0.189945   False
...        ...     ...
2045  0.000076    True
2046  0.044173   False
2047  0.086346    True
2048  0.032253   False
2049  0.029705    True
2050  0.134203    True
2051  0.038915    True
2052  0.026321   False
2053  0.026236    True
2054  0.000903    True
2055  0.038915    True
2056  0.111411    True
2057  0.073482    True
2058  0.105506    True
2059  0.027351    True
2060  0.007958    True
2061  0.001498    True
2062  0.004633    True
2063  0.082981   False
2064  0.008167    True
2065  0.029248    True
2066  0.118184    True
2067  0.076507    True
2068  0.193548   False
2069  0.076902   False
2070  0.175219    True
2071  0.040836   False
```

```
2072  0.031379    True
2073  0.068957    True
2074  0.076110   False

[2075 rows x 12 columns]
```

In [7]:
```
tweet_counts
```

Out[7]:

	favorite_count	retweet_count	id
0	39467	8853	892420643555336193
1	33819	6514	892177421306343426
2	25461	4328	891815181378084864
3	42908	8964	891689557279858688
4	41048	9774	891327558926688256
5	20562	3261	891087950875897856
6	12041	2158	890971913173991426
7	56848	16716	890729181411237888
8	28226	4429	890609185150312448
9	32467	7711	890240255349198849
10	31166	7624	890006608113172480
11	28268	5156	889880896479866881
12	38818	8538	889665388333682689
13	27672	4735	889638837579907072
14	15359	2321	889531135344209921
15	25652	5637	889278841981685760
16	29611	4709	888917238123831296
17	26080	4559	888804989199671297
18	20290	3732	888554962724278272
19	22201	3653	888078434458587136
20	30779	5609	887705289381826560
21	46959	12082	887517139158093824
22	69871	18781	887473957103951883
23	34222	10737	887343217045368832
24	31061	6167	887101392804085760
25	35859	8084	886983233522544640
26	12306	3443	886736880519319552
27	22798	4610	886680336477933568
28	21524	3316	886366144734445568
29	117	4	886267009285017600
...	...	...	...
2324	459	339	666411507551481857
2325	113	44	666407126856765440
2326	172	92	666396247373291520
2327	194	100	666373753744588802
2328	804	595	666362758909284353
2329	229	77	666353288456101888
2330	307	146	666345417576210432
2331	204	96	666337882303524864

```
2332             522            368  666293911632134144
2333             152             71  666287406224695296
2334             184             82  666273097616637952
2335             108             37  666268910803644416
2336           14765           6871  666104133288665088
2337              81             16  666102155909144576
2338             164             73  666099513787052032
2339             169             79  666094000022159362
2340             121             47  666082916733198337
2341             335            174  666073100786774016
2342             154             67  666071193221509120
2343             496            232  666063827256086533
2344             115             61  666058600524156928
2345             304            146  666057090499244032
2346             448            261  666055525042405380
2347            1253            879  666051853826850816
2348             136             60  666050758794694657
2349             111             41  666049248165822465
2350             311            147  666044226329800704
2351             128             47  666033412701032449
2352             132             48  666029285002620928
2353            2535            532  666020888022790149

[2354 rows x 3 columns]
```

6 Assessing Part 2

- after we made a visual assessment on the dataset, it's time to make a programmatic assessment
- I'll Start the programmatic assesment with ***1. archive_df***

```
In [8]: archive_df.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 [9]: archive_df.name.value_counts()
```

```
Out[9]: None            745
        a                55
        Charlie          12
        Oliver           11
        Cooper           11
        Lucy             11
        Lola             10
        Tucker           10
        Penny            10
        Bo                9
        Winston           9
        the               8
        Sadie             8
        Daisy             7
        Buddy             7
        Bailey            7
        Toby              7
        an                7
        Dave              6
        Koda              6
        Jax               6
        Milo              6
        Bella             6
        Rusty             6
        Stanley           6
        Jack              6
        Oscar             6
        Scout             6
        Leo               6
        Sunny             5
                        ...
        Ozzie             1
        Sunshine          1
        Striker           1
        Harvey            1
        Arnold            1
        Rupert            1
```

```
Harnold           1
Lenox             1
Ike               1
Kollin            1
Kayla             1
Goose             1
Chadrick          1
Kuyu              1
Poppy             1
infuriating       1
Severus           1
Corey             1
Dotsy             1
Cilantro          1
Jebberson         1
Gòrdón            1
Pete              1
unacceptable      1
Tove              1
Zara              1
incredibly        1
Biden             1
Deacon            1
Emma              1
Name: name, Length: 957, dtype: int64
```

```
In [10]: archive_df.tweet_id.duplicated().sum()
```

```
Out[10]: 0
```

```
In [11]: archive_df.isnull().sum()
```

```
Out[11]: 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 [12]: #Not Retweeted Tweets (Original Tweets)
         archive_df.retweeted_status_id.isnull().sum()
```

```
Out[12]: 2175
```

```
In [13]: archive_df.rating_numerator.value_counts()
```

```
Out[13]: 12      558
         11      464
         10      461
         13      351
         9       158
         8       102
         7        55
         14       54
         5        37
         6        32
         3        19
         4        17
         1         9
         2         9
         420       2
         0         2
         15        2
         75        2
         80        1
         20        1
         24        1
         26        1
         44        1
         50        1
         60        1
         165       1
         84        1
         88        1
         144       1
         182       1
         143       1
         666       1
         960       1
         1776      1
         17        1
         27        1
         45        1
         99        1
         121       1
```

```
204       1
Name: rating_numerator, dtype: int64
```

```
In [14]: archive_df.rating_denominator.value_counts()
```

```
Out[14]: 10     2333
         11        3
         50        3
         80        2
         20        2
         2         1
         16        1
         40        1
         70        1
         15        1
         90        1
         110       1
         120       1
         130       1
         150       1
         170       1
         7         1
         0         1
         Name: rating_denominator, dtype: int64
```

```
In [15]: (archive_df.rating_denominator != 10).sum()
```

```
Out[15]: 23
```

```
In [16]: (archive_df.rating_numerator > 20).sum()
```

```
Out[16]: 24
```

- most of ratings are >= 10 and <= 20
- I choosen the limit to be 20 and every value exceeds 20 can be considered as a typo or someone is biased towards a certain type of dog so he gave him an extremly high rating or simply an "error".

```
In [17]: archive_df.source.describe()
```

```
Out[17]: count                                                   2356
         unique                                                     4
         top       <a href="http://twitter.com/download/iphone" r...
         freq                                                    2221
         Name: source, dtype: object
```

```
In [18]: archive_df.expanded_urls.describe()
```

```
Out[18]: count                                                  2297
         unique                                                 2218
         top       https://twitter.com/dog_rates/status/698195409...
         freq                                                      2
         Name: expanded_urls, dtype: object

In [19]: archive_df.doggo.value_counts()

Out[19]: None     2259
         doggo      97
         Name: doggo, dtype: int64

In [20]: archive_df.floofer.value_counts()

Out[20]: None       2346
         floofer      10
         Name: floofer, dtype: int64

In [21]: archive_df.puppo.value_counts()

Out[21]: None     2326
         puppo      30
         Name: puppo, dtype: int64

In [22]: archive_df.pupper.value_counts()

Out[22]: None      2099
         pupper     257
         Name: pupper, dtype: int64

In [23]: #checking Tweet text if it may contain decimal points inside the text itself

         archive_df[archive_df.text.str.contains(r"(\d+\.\d*\/\d+)")]



         #Helpful Resources:

         #https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.Series.str.conta
         #https://stackoverflow.com/questions/21923361/how-to-check-a-string-contains-only-digit
         #https://stackoverflow.com/questions/4999064/regex-for-string-contains
         #https://cs.lmu.edu/~ray/notes/regex/
         #https://stackoverflow.com/questions/14017134/what-is-d-d-in-regex

/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:3: UserWarning: This pattern has ma
  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[23]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweeted_status_user_id	retweeted_status_timestamp	expanded_urls	rating_numerator
45	883482846933004288	NaN	NaN	2017-07-08 00:28:19 +0000	<a href="http://twitter.com/download/iphone" r...	This is Bella. She hopes her smile made you sm...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/883482846...	5
340	832215909146226688	NaN	NaN	2017-02-16 13:11:49 +0000	<a href="http://twitter.com/download/iphone" r...	RT @dog_rates: This is Logan, the Chow who liv...	7.867091e+17	4.196984e+09	2016-10-13 23:23:56 +0000	https://twitter.com/dog_rates/status/786709082...	75
695	786709082849828864	NaN	NaN	2016-10-13 23:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Logan, the Chow who lived. He solemnly...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/786709082...	75
763	778027034220126208	NaN	NaN	2016-09-20 00:24:34 +0000	<a href="http://twitter.com/download/iphone" r...	This is Sophie. She's a Jubilant Bush Pupper. ...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/778027034...	27
1689	681340665377193984	6.813394e+17	4.196984e+09	2015-12-28 05:07:27 +0000	<a href="http://twitter.com/download/iphone" r...	I've been told there's a slight possibility he...	NaN	NaN	NaN	NaN	5
1712	680494726643068929	NaN	NaN	2015-12-25 21:06:00 +0000	<a href="http://twitter.com/download/iphone" r...	Here we have uncovered an entire battalion of ...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/680494726...	26

```
      rating_denominator    name doggo floofer  pupper puppo
45                    10   Bella  None    None    None  None
340                   10   Logan  None    None    None  None
695                   10   Logan  None    None    None  None
763                   10  Sophie  None    None  pupper  None
1689                  10    None  None    None    None  None
1712                  10    None  None    None    None  None
```

2. Assessing image_predictions_df

```
In [24]: image_prediction_df.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 [25]: image_prediction_df.tweet_id.duplicated().sum()
```

```
Out[25]: 0
```

```
In [26]: image_prediction_df.jpg_url.duplicated().sum()
```

```
Out[26]: 66
```

```
In [27]: image_prediction_df.p1_dog.value_counts()
```

```
Out[27]: True     1532
         False     543
         Name: p1_dog, dtype: int64
```

```
In [28]: image_prediction_df.p2_dog.value_counts()
```

```
Out[28]: True     1553
         False     522
         Name: p2_dog, dtype: int64
```

```
In [29]: image_prediction_df.p3_dog.value_counts()

Out[29]: True     1499
         False     576
         Name: p3_dog, dtype: int64
```

3. Assessing "tweet_counts" Data Frame

```
In [30]: tweet_counts.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
favorite_count    2354 non-null int64
retweet_count     2354 non-null int64
id                2354 non-null int64
dtypes: int64(3)
memory usage: 55.2 KB


In [31]: tweet_counts.id.duplicated().sum()

Out[31]: 0

In [32]: tweet_counts.favorite_count.value_counts()

Out[32]: 0        179
         610        3
         345        3
         2918       3
         1691       3
         2176       3
         2768       3
         1339       3
         2706       3
         522        2
         3134       2
         1618       2
         250        2
         2250       2
         2660       2
         2262       2
         2305       2
         1111       2
         784        2
         4878       2
         346        2
         14685      2
         780        2
```

```
6923       2
6515       2
2433       2
3603       2
13518      2
3593       2
1536       2
          ...
4681       1
523        1
559        1
802        1
527        1
27154      1
6676       1
535        1
537        1
6682       1
8731       1
23074      1
21029      1
667        1
6696       1
2608       1
35400      1
21041      1
4659       1
10804      1
4099       1
68152      1
10812      1
573        1
6718       1
33345      1
814        1
23108      1
2630       1
8143       1
Name: favorite_count, Length: 2007, dtype: int64
```

```
In [33]: tweet_counts.retweet_count.value_counts()
```

```
Out[33]: 1972     5
         3652     5
         83       5
         146      4
         61       4
         748      4
```

```
2243     4
336      4
183      4
179      4
1207     4
265      4
115      4
71       4
1124     4
542      4
819      4
577      4
516      4
397      3
619      3
661      3
2511     3
261      3
431      3
482      3
403      3
557      3
572      3
576      3
         ..
2088     1
1271     1
2030     1
43       1
5365     1
4143     1
3316     1
1263     1
16439    1
2104     1
4125     1
27       1
4121     1
4119     1
4079     1
1285     1
10226    1
8183     1
2042     1
11524    1
6148     1
7        1
1281     1
```

```
2060     1
1825     1
8209     1
19       1
2068     1
30742    1
0        1
Name: retweet_count, Length: 1724, dtype: int64
```

7 Quality Issues:

7.0.1 archive_df :

- Remove Data contains retweets and leave the original content
- change column timestamp data type to appropriate data type
- change the form of source column to contain only the needed text that describes the source of navigating. in other words, Remove the HTML tags that are not needed
- change data type of tweet_id into object
- some inaccurate names like: 'a', 'an', 'the', 'very', 'by', 'al'
- Replace name values which equal to None
- unnecessary columns can be removed
- change the rating_denominator and numerator datatype into float

7.0.2 image_predictions_df :

- Remove jpg_url duplicated items
- Remove unnecessary columns
- tweet_id is int

7.0.3 tweet_counts :

- Remove retweets and keep original tweets

8 Tidiness Issues:

- The data in 3 tables could be merged into 1 dataset, because they describe one thing
- We have dog stages doggo, floofer, pupper and puppo in 4 different columns instead of one column in **archive_df**

9 Data Cleaning:

In [34]:
```
#First of all, I'll Make a Copy of our original Datasets before starting the process of

archive_df_copy= archive_df.copy()
image_prediction_df_copy=image_prediction_df.copy()
tweet_counts_copy=tweet_counts.copy()
```

```
        #Refrence:
        #https://pandas.pydata.org/pandas-docs/version/0.22/generated/pandas.DataFrame.copy.htm
```

1st Tidiness Issue

```
In [35]: #Define: All 3 Tables will be merged into one dataframe since they'are all containing t
         # to make cleaning process easier we will use merge function in pandas


         #Code:

         #merging on tweet_id column like in SQL Inner Join PK & FK
         archive_df_copy = pd.merge(left=archive_df_copy,
                                    right=tweet_counts_copy, left_on='tweet_id', right_on=

         archive_df_copy = archive_df_copy.merge(image_prediction_df_copy, on='tweet_id', how='i

         #Test:
         archive_df_copy.info()

         #Reference:
         #https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.merge.html
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2073 entries, 0 to 2072
Data columns (total 31 columns):
tweet_id                      2073 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     2073 non-null object
source                        2073 non-null object
text                          2073 non-null object
retweeted_status_id           79 non-null float64
retweeted_status_user_id      79 non-null float64
retweeted_status_timestamp    79 non-null object
expanded_urls                 2073 non-null object
rating_numerator              2073 non-null int64
rating_denominator            2073 non-null int64
name                          2073 non-null object
doggo                         2073 non-null object
floofer                       2073 non-null object
pupper                        2073 non-null object
puppo                         2073 non-null object
favorite_count                2073 non-null int64
retweet_count                 2073 non-null int64
id                            2073 non-null int64
jpg_url                       2073 non-null object
```

```
img_num                       2073 non-null int64
p1                            2073 non-null object
p1_conf                       2073 non-null float64
p1_dog                        2073 non-null bool
p2                            2073 non-null object
p2_conf                       2073 non-null float64
p2_dog                        2073 non-null bool
p3                            2073 non-null object
p3_conf                       2073 non-null float64
p3_dog                        2073 non-null bool
dtypes: bool(3), float64(7), int64(7), object(14)
memory usage: 475.7+ KB
```

2nd Tidiness Issue

```
In [36]: #Define:
         # Dog stages are in 4 seperate columns, we can put the stage of every dog in a single c

         #Code:
         #Extracting the 4 stages of dogs from the context using extract function and then creat
         archive_df_copy['stage']=archive_df_copy['text'].str.extract('(doggo|floofer|pupper|pup

         archive_df_copy= archive_df_copy.drop(['doggo', 'floofer', 'pupper', 'puppo'], axis=1)

         #Test:
         archive_df_copy.info()
         archive_df_copy['stage'].value_counts()

         #Refrences:
         #https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.str.extract.h
         #https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2073 entries, 0 to 2072
Data columns (total 28 columns):
tweet_id                      2073 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     2073 non-null object
source                        2073 non-null object
text                          2073 non-null object
retweeted_status_id           79 non-null float64
retweeted_status_user_id      79 non-null float64
retweeted_status_timestamp    79 non-null object
expanded_urls                 2073 non-null object
rating_numerator              2073 non-null int64
rating_denominator            2073 non-null int64
```

```
name                        2073 non-null object
favorite_count              2073 non-null int64
retweet_count               2073 non-null int64
id                          2073 non-null int64
jpg_url                     2073 non-null object
img_num                     2073 non-null int64
p1                          2073 non-null object
p1_conf                     2073 non-null float64
p1_dog                      2073 non-null bool
p2                          2073 non-null object
p2_conf                     2073 non-null float64
p2_dog                      2073 non-null bool
p3                          2073 non-null object
p3_conf                     2073 non-null float64
p3_dog                      2073 non-null bool
stage                       337 non-null object
dtypes: bool(3), float64(7), int64(7), object(11)
memory usage: 427.2+ KB
```

```
Out[36]: pupper     230
         doggo       75
         puppo       29
         floofer      3
         Name: stage, dtype: int64
```

9.0.1 Quality Issues in Archive_df

1st Quality Issue

```
In [37]: #Define:
         # Remove any data contains retweets from columns: retweeted_status_id, retweeted_status
         # and keep the null values a.k.a original tweets

         #Code:
         archive_df_copy=archive_df_copy.drop(['retweeted_status_id','retweeted_status_user_id',

         #Test:
         archive_df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2073 entries, 0 to 2072
Data columns (total 25 columns):
tweet_id                 2073 non-null int64
in_reply_to_status_id    23 non-null float64
in_reply_to_user_id      23 non-null float64
timestamp                2073 non-null object
source                   2073 non-null object
text                     2073 non-null object
```

```
expanded_urls            2073 non-null object
rating_numerator         2073 non-null int64
rating_denominator       2073 non-null int64
name                     2073 non-null object
favorite_count           2073 non-null int64
retweet_count            2073 non-null int64
id                       2073 non-null int64
jpg_url                  2073 non-null object
img_num                  2073 non-null int64
p1                       2073 non-null object
p1_conf                  2073 non-null float64
p1_dog                   2073 non-null bool
p2                       2073 non-null object
p2_conf                  2073 non-null float64
p2_dog                   2073 non-null bool
p3                       2073 non-null object
p3_conf                  2073 non-null float64
p3_dog                   2073 non-null bool
stage                    337 non-null object
dtypes: bool(3), float64(5), int64(7), object(10)
memory usage: 378.6+ KB
```

2nd Quality Issue

```
In [38]: #Define: change datatype of timestamp column from object to datetime


         #code:
         archive_df_copy.timestamp= pd.to_datetime(archive_df_copy.timestamp)

         #Test
         archive_df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2073 entries, 0 to 2072
Data columns (total 25 columns):
tweet_id                 2073 non-null int64
in_reply_to_status_id    23 non-null float64
in_reply_to_user_id      23 non-null float64
timestamp                2073 non-null datetime64[ns]
source                   2073 non-null object
text                     2073 non-null object
expanded_urls            2073 non-null object
rating_numerator         2073 non-null int64
rating_denominator       2073 non-null int64
name                     2073 non-null object
favorite_count           2073 non-null int64
```

```
retweet_count            2073 non-null int64
id                       2073 non-null int64
jpg_url                  2073 non-null object
img_num                  2073 non-null int64
p1                       2073 non-null object
p1_conf                  2073 non-null float64
p1_dog                   2073 non-null bool
p2                       2073 non-null object
p2_conf                  2073 non-null float64
p2_dog                   2073 non-null bool
p3                       2073 non-null object
p3_conf                  2073 non-null float64
p3_dog                   2073 non-null bool
stage                    337 non-null object
dtypes: bool(3), datetime64[ns](1), float64(5), int64(7), object(9)
memory usage: 378.6+ KB
```

3rd Quality Issue

```
In [39]: #Define: Split the markup tags around the text to make it more human readable and more
         # and then use it

         #Code:
         archive_df_copy.source=archive_df_copy.source.str.split('>').str[1].str.split('<').str[

         #Test
         archive_df_copy.source.value_counts()

         #Reference:
         #https://pandas.pydata.org/pandas-docs/version/0.23/generated/pandas.Series.str.split.h
```

```
Out[39]: Twitter for iPhone    2032
         Twitter Web Client      30
         TweetDeck               11
         Name: source, dtype: int64
```

4th Quality Issue

```
In [40]: #Define: The Datatype of tweet_id should be object since it carries a list of strings n
         # this might an error in data entry from spreadsheets like google or ms excel

         #Code:
         archive_df_copy['tweet_id']= archive_df_copy['tweet_id'].astype(str)

         #Test
         archive_df_copy.info()
```

```
        #Reference:
        #https://stackoverflow.com/questions/17950374/converting-a-column-within-pandas-datafra
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2073 entries, 0 to 2072
Data columns (total 25 columns):
tweet_id                 2073 non-null object
in_reply_to_status_id    23 non-null float64
in_reply_to_user_id      23 non-null float64
timestamp                2073 non-null datetime64[ns]
source                   2073 non-null object
text                     2073 non-null object
expanded_urls            2073 non-null object
rating_numerator         2073 non-null int64
rating_denominator       2073 non-null int64
name                     2073 non-null object
favorite_count           2073 non-null int64
retweet_count            2073 non-null int64
id                       2073 non-null int64
jpg_url                  2073 non-null object
img_num                  2073 non-null int64
p1                       2073 non-null object
p1_conf                  2073 non-null float64
p1_dog                   2073 non-null bool
p2                       2073 non-null object
p2_conf                  2073 non-null float64
p2_dog                   2073 non-null bool
p3                       2073 non-null object
p3_conf                  2073 non-null float64
p3_dog                   2073 non-null bool
stage                    337 non-null object
dtypes: bool(3), datetime64[ns](1), float64(5), int64(6), object(10)
memory usage: 378.6+ KB
```

5th Quality Issue

```
In [41]: #Define: Remove any unusual names which are most of times are typos, we'll replace it u
```

```
        #Code:
        archive_df_copy['name']= archive_df_copy['name'].replace('a', np.nan)
        archive_df_copy['name']= archive_df_copy['name'].replace('an', np.nan)
        archive_df_copy['name']= archive_df_copy['name'].replace('al', np.nan)
        archive_df_copy['name']= archive_df_copy['name'].replace('a', np.nan)
        archive_df_copy['name']= archive_df_copy['name'].replace('the', np.nan)
        archive_df_copy['name']= archive_df_copy['name'].replace('very', np.nan)
```

```
archive_df_copy['name']= archive_df_copy['name'].replace('by', np.nan)
archive_df_copy['name']= archive_df_copy['name'].replace('O', np.nan)

#Test:
archive_df_copy['name'].value_counts()

#Reference:
#https://www.programcreek.com/python/example/6575/numpy.nan
```

```
Out[41]: None          577
         Charlie        11
         Cooper         10
         Lucy           10
         Oliver         10
         Penny          10
         Tucker         10
         Sadie           8
         Bo              8
         Lola            8
         Winston         8
         Toby            7
         Daisy           7
         Bella           6
         Koda            6
         Bailey          6
         Stanley         6
         Dave            6
         Jax             6
         Scout           6
         Rusty           6
         Milo            6
         Louis           5
         Larry           5
         Alfie           5
         Leo             5
         Chester         5
         Buddy           5
         Oscar           5
         Bear            4
                        ...
         Goose           1
         Flurpson        1
         Kollin          1
         Wishes          1
         Glacier         1
         Jaspers         1
         Boston          1
         Clifford        1
```

```
Severus         1
Philbert        1
Emma            1
Kuyu            1
Mingus          1
Binky           1
Andy            1
Ashleigh        1
Millie          1
Divine          1
Holly           1
Remus           1
Harrison        1
Winifred        1
Brownie         1
Christoper      1
Jameson         1
Caryl           1
Snoopy          1
Ralf            1
Striker         1
Corey           1
Name: name, Length: 930, dtype: int64
```

6th Quality Issue

```
In [42]: #Define: Replace name values which equal to None with NaN Values


         #Code:
         archive_df_copy['name']= archive_df_copy['name'].replace('None', np.nan)

         #Test

         archive_df_copy['name'].value_counts

Out[42]: <bound method IndexOpsMixin.value_counts of 0        Phineas
         1          Tilly
         2         Archie
         3          Darla
         4       Franklin
         5            NaN
         6            Jax
         7            NaN
         8           Zoey
         9         Cassie
         10          Koda
         11         Bruno
```

```
12           NaN
13           Ted
14        Stuart
15        Oliver
16           Jim
17          Zeke
18       Ralphus
19        Gerald
20       Jeffrey
21          such
22        Canela
23           NaN
24           NaN
25          Maya
26        Mingus
27         Derek
28        Roscoe
29       Waffles
          ...
2043       quite
2044         NaN
2045         NaN
2046         NaN
2047         NaN
2048         NaN
2049         NaN
2050         NaN
2051         NaN
2052         NaN
2053         NaN
2054         NaN
2055         NaN
2056         NaN
2057         NaN
2058         NaN
2059         NaN
2060         NaN
2061         NaN
2062         NaN
2063         NaN
2064         NaN
2065         NaN
2066         NaN
2067         NaN
2068         NaN
2069         NaN
2070         NaN
2071         NaN
```

```
2072         NaN
Name: name, Length: 2073, dtype: object>
```

7th Quality Issue

```
In [43]: #Define: Remove all unnecessary columns in our analyses

         #Code:

         #Remove unneeded columns and save it in the same place
         archive_df_copy.drop(['in_reply_to_status_id','in_reply_to_user_id','expanded_urls','im

         #Test:
         archive_df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2073 entries, 0 to 2072
Data columns (total 20 columns):
tweet_id              2073 non-null object
timestamp             2073 non-null datetime64[ns]
source                2073 non-null object
rating_numerator      2073 non-null int64
rating_denominator    2073 non-null int64
name                  1421 non-null object
favorite_count        2073 non-null int64
retweet_count         2073 non-null int64
id                    2073 non-null int64
jpg_url               2073 non-null object
p1                    2073 non-null object
p1_conf               2073 non-null float64
p1_dog                2073 non-null bool
p2                    2073 non-null object
p2_conf               2073 non-null float64
p2_dog                2073 non-null bool
p3                    2073 non-null object
p3_conf               2073 non-null float64
p3_dog                2073 non-null bool
stage                 337 non-null object
dtypes: bool(3), datetime64[ns](1), float64(3), int64(5), object(8)
memory usage: 297.6+ KB
```

9.0.2 Originally from Image_prediction_df dataset before merging it with the other two datasets

8th Quality Issue

```
In [44]: #Define: Remove duplicates in jpg_url
```

```
#Code:
archive_df_copy= archive_df_copy.drop_duplicates(subset=['jpg_url'], keep='first')


#Test:
archive_df_copy.info()

#Reference:
#https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop_dupli
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2008 entries, 0 to 2072
Data columns (total 20 columns):
tweet_id              2008 non-null object
timestamp             2008 non-null datetime64[ns]
source                2008 non-null object
rating_numerator      2008 non-null int64
rating_denominator    2008 non-null int64
name                  1374 non-null object
favorite_count        2008 non-null int64
retweet_count         2008 non-null int64
id                    2008 non-null int64
jpg_url               2008 non-null object
p1                    2008 non-null object
p1_conf               2008 non-null float64
p1_dog                2008 non-null bool
p2                    2008 non-null object
p2_conf               2008 non-null float64
p2_dog                2008 non-null bool
p3                    2008 non-null object
p3_conf               2008 non-null float64
p3_dog                2008 non-null bool
stage                 328 non-null object
dtypes: bool(3), datetime64[ns](1), float64(3), int64(5), object(8)
memory usage: 288.3+ KB
```

9th Quality Issue

```
In [45]: #Define: Remove another unnecessary columns originally from image_prediction_df dataset

         #Code:

         archive_df_copy = archive_df_copy.drop([ 'p1', 'p1_conf', 'p1_dog', 'p2', 'p2_conf', 'p

         #Test:
         archive_df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2008 entries, 0 to 2072
Data columns (total 11 columns):
tweet_id              2008 non-null object
timestamp             2008 non-null datetime64[ns]
source                2008 non-null object
rating_numerator      2008 non-null int64
rating_denominator    2008 non-null int64
name                  1374 non-null object
favorite_count        2008 non-null int64
retweet_count         2008 non-null int64
id                    2008 non-null int64
jpg_url               2008 non-null object
stage                 328 non-null object
dtypes: datetime64[ns](1), int64(5), object(5)
memory usage: 188.2+ KB
```

10th Quality Issue

```
In [46]: #Define: change the rating_denominator and rating_numerator datatype into float


         #Code:
         archive_df_copy.rating_denominator=archive_df_copy.rating_denominator.astype('float')
         archive_df_copy.rating_numerator=archive_df_copy.rating_numerator.astype('float')

         #Test
         archive_df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2008 entries, 0 to 2072
Data columns (total 11 columns):
tweet_id              2008 non-null object
timestamp             2008 non-null datetime64[ns]
source                2008 non-null object
rating_numerator      2008 non-null float64
rating_denominator    2008 non-null float64
name                  1374 non-null object
favorite_count        2008 non-null int64
retweet_count         2008 non-null int64
id                    2008 non-null int64
jpg_url               2008 non-null object
stage                 328 non-null object
dtypes: datetime64[ns](1), float64(2), int64(3), object(5)
memory usage: 188.2+ KB
```

9.1 Storing Data Frame

After finishing our data wrangling process we have to save our work in a master csv file

```
In [55]: archive_df_copy.to_csv('twitter_archive_master.csv')
```

10 Data Analysis and Visualizations

```
In [58]: plots_df=pd.read_csv('twitter_archive_master.csv')

         plots_df.head()
```

Out[58]:

	Unnamed: 0	tweet_id	timestamp	source
0	0	892420643555336193	2017-08-01 16:23:56	Twitter for iPhone
1	1	892177421306343426	2017-08-01 00:17:27	Twitter for iPhone
2	2	891815181378084864	2017-07-31 00:18:03	Twitter for iPhone
3	3	891689557279858688	2017-07-30 15:58:51	Twitter for iPhone
4	4	891327558926688256	2017-07-29 16:00:24	Twitter for iPhone

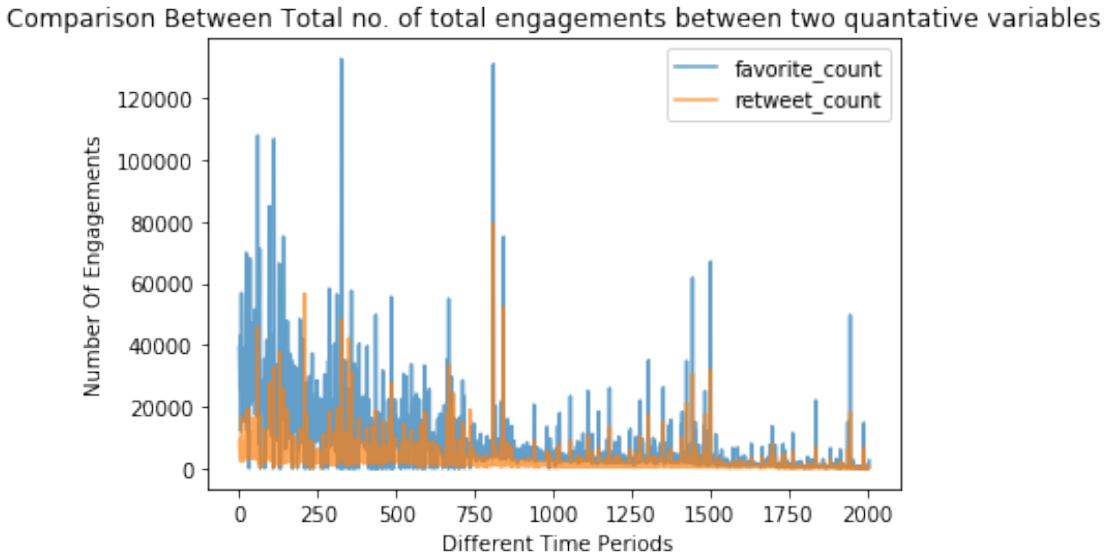
	rating_numerator	rating_denominator	name	favorite_count
0	13.0	10.0	Phineas	39467
1	13.0	10.0	Tilly	33819
2	12.0	10.0	Archie	25461
3	13.0	10.0	Darla	42908
4	12.0	10.0	Franklin	41048

	retweet_count	id
0	8853	892420643555336193
1	6514	892177421306343426
2	4328	891815181378084864
3	8964	891689557279858688
4	9774	891327558926688256

	jpg_url	stage
0	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	NaN
1	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg	NaN
2	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg	NaN
3	https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg	NaN
4	https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg	NaN

10.1 Retweet count and favorite count over the time

```
In [83]: plots_df[['favorite_count','retweet_count']].plot(alpha=0.7);
         plt.xlabel(' Different Time Periods');
         plt.ylabel('Number Of Engagements');
         plt.title('Comparison Between Total no. of total engagements between two quantative var
```

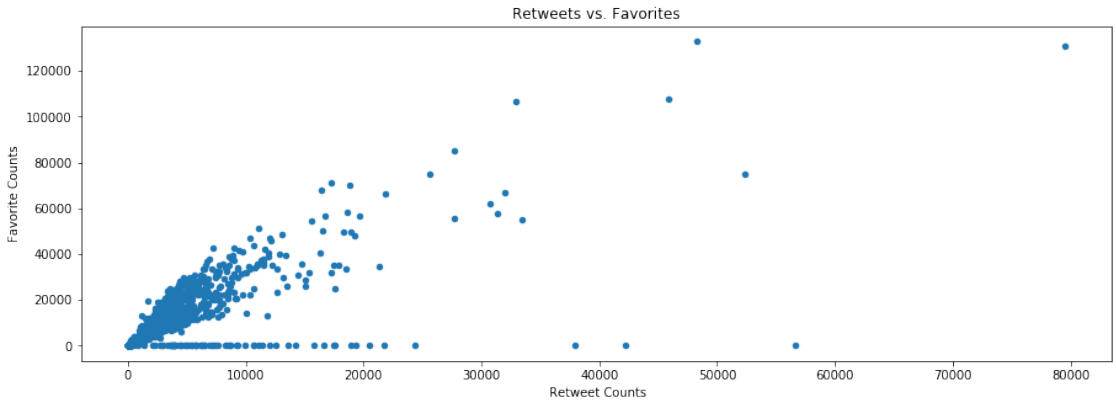


Insight:

- I see over the plot here, people tend to interact by using favorite button more than interacting with retweets
- the interaction with retweets was more than favorites only one time in the period before value 250

10.1.1 Retweets And Favorites correlation

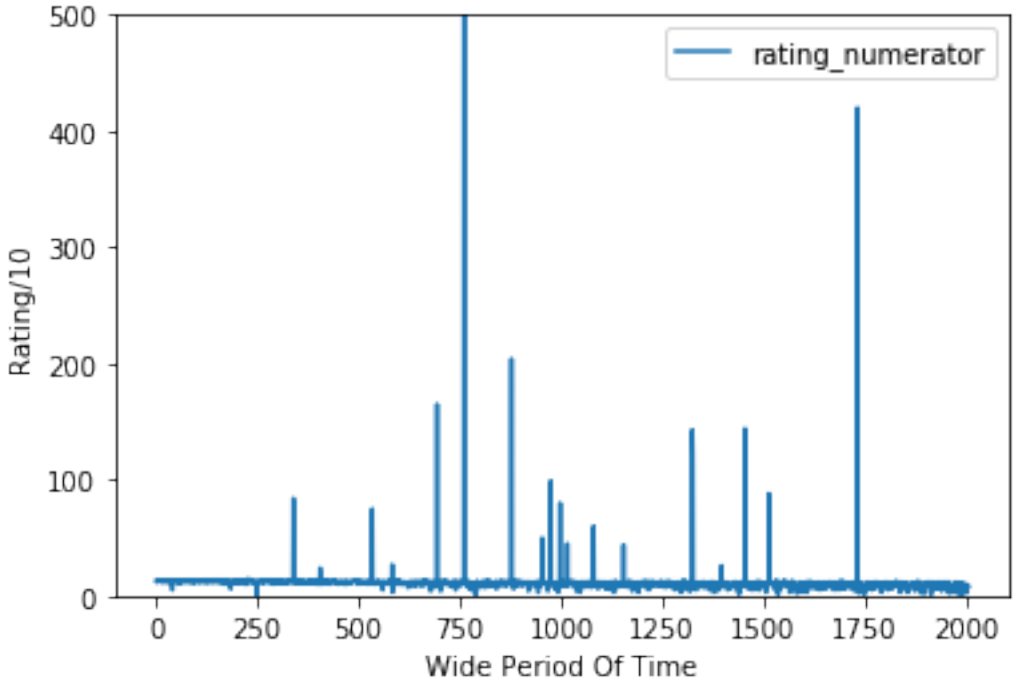
```
In [84]: plots_df.plot(x='retweet_count', y='favorite_count', kind='scatter', figsize=(15,5));
         plt.xlabel(' Retweet Counts');
         plt.ylabel('Favorite Counts');
         plt.title('Retweets vs. Favorites');
```



Insight

- though, the favorites engagements is always higher than retweets, there's a strong correlation between them as we see here in the period from 0 to 1000 on x-axis and from 0 to 20,000 on y-axis, also there's a small outliers in the other regions in the plot

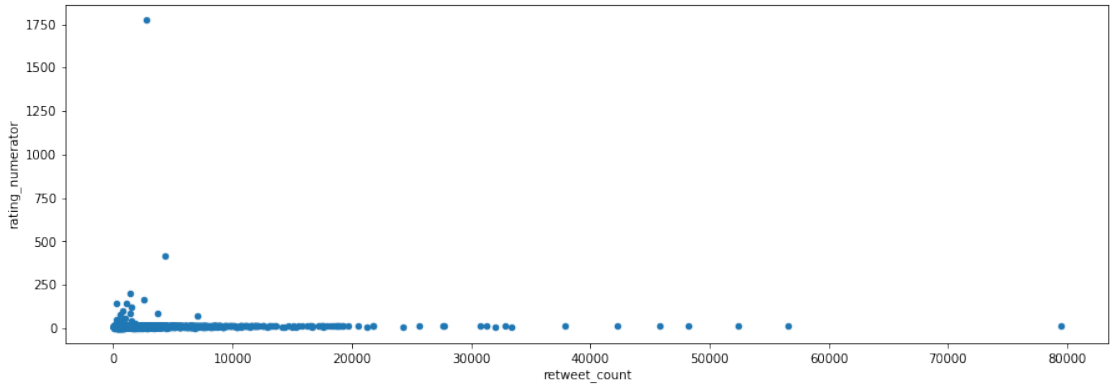
```
In [112]: plots_df.plot(y='rating_numerator',ylim=[0,500])
          plt.xlabel('Wide Period Of Time')
          plt.ylabel('Rating/10');
```



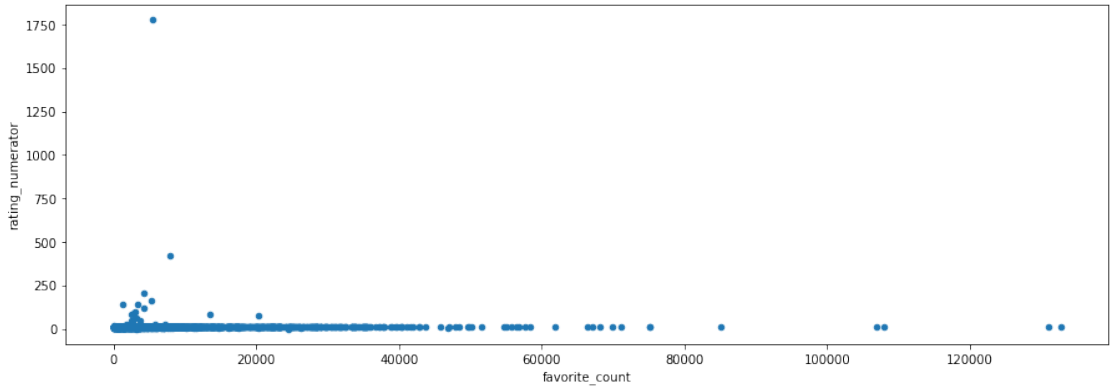
Insight

- We see here, that the people's engagements on WeRateDogs account was very small in the beginnigs and started to grow and it's clearly that was the time that account created,
- there's a few outliers as the standarized rating is preferable not to exceed extremly high values
- extremely high ratings could be a sign of biasing towards certain type of dog or could be a sign that the people are in love with content of the account

```
In [113]: plots_df.plot(x='retweet_count', y='rating_numerator', kind='scatter', figsize=[15,5])
```




```
In [106]: plots_df.plot(x='favorite_count', y='rating_numerator', kind='scatter', figsize=[15,5]
```



10.1.2 Insight:

- we see that the higher ratings tends to have a smaller retweets engagements than that of favorites engagements