wrangle_act

June 24, 2019

1 Wrangle and Analyze Data

wrangle WeRateDogs Twitter data to analyze and visualize it.

```
WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog.
almost always have a denominator of 10. The numerators, though? Almost always greater than 10.
```

By Maram Mahmoud

1.0.1 Gathering Data

I will gather data from three sources. - The WeRateDogs Twitter archive. This file is locally available. - The tweet image predictions, what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network. This file is hosted on Udacity's servers. - Each tweet's retweet count and favorite count. Using the tweet IDs in the WeRateDogs Twitter archive query the Twitter API for each tweet's JSON data using Python's Tweepy library. I will import each source's data in separate pandas DataFrame.

```
In [1]: import numpy as np
        import pandas as pd
        import requests
        import tweepy
        import json

        import matplotlib.pyplot as plt
        %matplotlib inline
```

Getting WeRateDogs Twitter archive

```
In [2]: archive_df = pd.read_csv('twitter-archive-enhanced-2.csv')
```

Getting tweet image predictions

```
In [3]: url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictio
        response = requests.get(url)

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

```
#Read TSV file
image_df = pd.read_csv('image-predictions.tsv', sep='\t' )
```

Getting more tweet information setting my Twitter API

```
In [4]: '''
consumer_key = '00'
consumer_secret = '00'
access_token = '0-00'
access_secret = '00'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth)
''';
```

downloading Tweepy status object based on Tweet ID from archive and store in list 'tweets_status'

```
In [5]: '''
tweets =[]
for tweet_id in archive_df['tweet_id']:
    try:
        tweet = api.get_status(id = tweet_id)
        tweets.append(tweet)
    except:
        pass
''';
```

Isolating the json part of each tweepy status and storing it in txt file

```
In [6]: '''
tweets_json = []
for tweet_json in tweets_status:
    tweets_json.append(tweet_json._json)

with open('tweet_json.txt', 'w') as outfile:
    json.dump(tweets_json, outfile)
outfile.close()
''';
```

importing data in a dataframe

```
In [7]: df_list =[]
with open('tweet_json.txt', 'r') as file:
    for j in file:
        data = json.loads(j)
```

```
            df_list.append({'tweet_id':data['id'],
                            'retweet_count':data['retweet_count'],
                            'favorite_count':data['favorite_count']})
    json_df = pd.DataFrame(df_list,columns = ['tweet_id','retweet_count','favorite_count'])
```

Please note that I didn't use what I actually got from twitter API because it was way less than what in the archive.

1.1 Assessing data

Visual assessment by looking through the data

```
In [8]: archive_df
```

Out[8]:

	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	887978434458587136	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 [9]:
```
image_df
```

Out[9]:

	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
10	666063827256086533	https://pbs.twimg.com/media/CT5Vg_wXIAAXfnj.jpg
11	666071193221509120	https://pbs.twimg.com/media/CT5cN_3WEAAl0oZ.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_0-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 [10]:
```
json_df
```

Out[10]:

	tweet_id	retweet_count	favorite_count
0	892420643555336193	8853	39467
1	892177421306343426	6514	33819
2	891815181378084864	4328	25461
3	891689557279858688	8964	42908
4	891327558926688256	9774	41048

5	891087950875897856	3261	20562
6	890971913173991426	2158	12041
7	890729181411237888	16716	56848
8	890609185150312448	4429	28226
9	890240255349198849	7711	32467
10	890006608113172480	7624	31166
11	889880896479866881	5156	28268
12	889665388333682689	8538	38818
13	889638837579907072	4735	27672
14	889531135344209921	2321	15359
15	889278841981685760	5637	25652
16	888917238123831296	4709	29611
17	888804989199671297	4559	26080
18	888554962724278272	3732	20290
19	888078434458587136	3653	22201
20	887705289381826560	5609	30779
21	887517139158093824	12082	46959
22	887473957103951883	18781	69871
23	887343217045368832	10737	34222
24	887101392804085760	6167	31061
25	886983233522544640	8084	35859
26	886736880519319552	3443	12306
27	886680336477933568	4610	22798
28	886366144734445568	3316	21524
29	886267009285017600	4	117
...	...	...	...
2324	666411507551481857	339	459
2325	666407126856765440	44	113
2326	666396247373291520	92	172
2327	666373753744588802	100	194
2328	666362758909284353	595	804
2329	666353288456101888	77	229
2330	666345417576210432	146	307
2331	666337882303524864	96	204
2332	666293911632134144	368	522
2333	666287406224695296	71	152
2334	666273097616637952	82	184
2335	666268910803644416	37	108
2336	666104133288665088	6871	14765
2337	666102155909144576	16	81
2338	666099513787052032	73	164
2339	666094000022159362	79	169
2340	666082916733198337	47	121
2341	666073100786774016	174	335
2342	666071193221509120	67	154
2343	666063827256086533	232	496
2344	666058600524156928	61	115
2345	666057090499244032	146	304

```
2346  666055525042405380            261            448
2347  666051853826850816            879           1253
2348  666050758794694657             60            136
2349  666049248165822465             41            111
2350  666044226329800704            147            311
2351  666033412701032449             47            128
2352  666029285002620928             48            132
2353  666020888022790149            532           2535

[2354 rows x 3 columns]
```

I noticed that each variable forms a column in dog stage.

And each type of observational unit forms a table (archive_df, image_df, data_df).

I should make one column for image prediction and one column for confidence to summarize the prediction result.

Programmatic assessment by using pandas helpful functions (e.g. info(), value_count(), duplicated(), etc)

```
In [11]: 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 [12]: image_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 [13]: json_df.info()
```

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

The timestamp should be type date.

There are 181 values in retweeted_status_id and retweeted_status_user_id. not original tweets

```
In [14]: print('number of tweets in archive =',len(archive_df.tweet_id))
         print('number of tweets from image predictions =',len(image_df.tweet_id))
         print('number of tweets from twitter API =',len(json_df.tweet_id))
```

```
number of tweets in archive = 2356
number of tweets from image predictions = 2075
number of tweets from twitter API = 2354
```

The images dataframe contains the smallest subset of tweets. and because I dont have the algorithm to complete images information,I will subset the other dataframes to just the tweets that are contained in images. The tweet that is in archive but not fond in twitter database is likely a deleted tweet.

```
In [15]: print('- whole set of tweets from image predictions contained in the set of archive twe
               str(image_df.tweet_id.isin(archive_df.tweet_id).all()))

         print('- whole set of tweets from image predictions contained in the set of tweets from
               str(image_df.tweet_id.isin(json_df.tweet_id).all()))
```

```
- whole set of tweets from image predictions contained in the set of archive tweets?  True
- whole set of tweets from image predictions contained in the set of tweets from twitter API?  F
```

I don't have data (from API) for all tweet collected from image predictions

```
In [16]: sum(archive_df.tweet_id.duplicated())

Out[16]: 0

In [17]: sum(image_df.tweet_id.duplicated())

Out[17]: 0

In [18]: sum(json_df.tweet_id.duplicated())

Out[18]: 0
```

good thing there are no duplicated tweets

```
In [19]: print('Number of denominators not equal to 10 is' ,len(archive_df[archive_df.rating_den

Number of denominators not equal to 10 is 23


In [20]: archive_df[archive_df.rating_denominator != 10].text

Out[20]: 313     @jonnysun @Lin_Manuel ok jomny I know you're e...
         342               @docmisterio account started on 11/15/15
         433     The floofs have been released I repeat the flo...
         516     Meet Sam. She smiles 24/7 &amp;amp; secretly aspir...
         784     RT @dog_rates: After so many requests, this is...
         902     Why does this never happen at my front door...
         1068    After so many requests, this is Bretagne. She ...
         1120    Say hello to this unbelievably well behaved sq...
         1165    Happy 4/20 from the squad! 13/10 for all https...
         1202    This is Bluebert. He just saw that both #Final...
         1228    Happy Saturday here's 9 puppers on a bench. 99...
         1254    Here's a brigade of puppers. All look very pre...
         1274    From left to right:\nCletus, Jerome, Alejandro...
         1351    Here is a whole flock of puppers.  60/50 I'll ...
         1433    Happy Wednesday here's a bucket of pups. 44/40...
         1598    Yes I do realize a rating of 4/20 would've bee...
         1634    Two sneaky puppers were not initially seen, mo...
         1635    Someone help the girl is being mugged. Several...
         1662    This is Darrel. He just robbed a 7/11 and is i...
         1663    I'm aware that I could've said 20/16, but here...
         1779    IT'S PUPPERGEDDON. Total of 144/120 ...I think...
         1843    Here we have an entire platoon of puppers. Tot...
         2335    This is an Albanian 3 1/2 legged  Episcopalian...
         Name: text, dtype: object
```

From visual assessment earlier I noticed most dogs rated out of 10 but 23 one is not.

It is either tweets that contain more than one dog or wrong identificating the percentage from the tweet text.

```
In [21]: archive_df.rating_numerator.value_counts()
```

```
Out[21]: 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 [22]: #last tweet showed in denominator != 10 was indexd 2335 lets see the full text of it.
```

```
print('tweet : ',archive_df['text'][2335])
print('numerator = ',archive_df['rating_numerator'][2335])
print('denominator = ',archive_df['rating_denominator'][2335])
archive_df['text'][2335]
```

```
tweet :  This is an Albanian 3 1/2 legged  Episcopalian. Loves well-polished hardwood flooring.
numerator =  1
denominator =  2
```

Out[22]: 'This is an Albanian 3 1/2 legged Episcopalian. Loves well-polished hardwood flooring.

A lot of ratings goes above 10/10 and there are unvalid rating numerators

In [23]:
```
archive_df.name.value_counts()
```

Out[23]:
```
None        745
a            55
Charlie      12
Cooper       11
Oliver       11
Lucy         11
Tucker       10
Lola         10
Penny        10
Bo            9
Winston       9
the           8
Sadie         8
an            7
Buddy         7
Bailey        7
Daisy         7
Toby          7
Stanley       6
Leo           6
Koda          6
Jack          6
Oscar         6
Rusty         6
Milo          6
Dave          6
Jax           6
Bella         6
Scout         6
Gus           5
             ...
Tyrus         1
Clifford      1
```

```
Jeremy         1
Edmund         1
Ralph          1
Taco           1
DonDon         1
Ralf           1
Berkeley       1
Nugget         1
Cedrick        1
Bloo           1
Godi           1
Tedders        1
Gilbert        1
Todo           1
Eleanor        1
Andy           1
Willie         1
Kobe           1
Brutus         1
Baron          1
Lili           1
Pippin         1
Ambrose        1
Pilot          1
Ozzie          1
Mya            1
Howie          1
Major          1
Name: name, Length: 957, dtype: int64
```

```
In [24]: archive_df.loc[ archive_df.name.str.islower() ].name.value_counts()
```

```
Out[24]: a              55
         the             8
         an              7
         very            5
         one             4
         quite           4
         just            4
         actually        2
         getting         2
         not             2
         mad             2
         infuriating     1
         my              1
         incredibly      1
         by              1
         unacceptable    1
```

```
his              1
light            1
this             1
officially       1
life             1
old              1
such             1
all              1
space            1
Name: name, dtype: int64
```

Data contains unvalid names like a, the and an

```
In [25]: sum(image_df['jpg_url'].duplicated())
```

```
Out[25]: 66
```

There are 66 duplicat images

```
In [26]: image_df.loc[(image_df.p1_dog==False) & (image_df.p2_dog==False) & (image_df.p3_dog==Fa
```

```
Out[26]: 6          https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg
         17         https://pbs.twimg.com/media/CT56LSZWoAAlJj2.jpg
         18         https://pbs.twimg.com/media/CT8QCd1WEAADXws.jpg
         21         https://pbs.twimg.com/media/CT8mx7KW4AEQu8N.jpg
         25         https://pbs.twimg.com/media/CT9lXGsUcAAyUFt.jpg
         29         https://pbs.twimg.com/media/CT-RugiWIAELEaq.jpg
         45         https://pbs.twimg.com/media/CUDmZIkWcAAIPPe.jpg
         50         https://pbs.twimg.com/media/CUEUva1WsAA2jPb.jpg
         51         https://pbs.twimg.com/media/CUGaXDhW4AY9JUH.jpg
         53         https://pbs.twimg.com/media/CUGObCOU8AAw2su.jpg
         56         https://pbs.twimg.com/media/CUHkkJpXIAA2w3n.jpg
         69         https://pbs.twimg.com/media/CUJUk2iWUAAVtOv.jpg
         73         https://pbs.twimg.com/media/CUL4xR9UkAEdlJ6.jpg
         77         https://pbs.twimg.com/media/CUM2qWaWoAUZO6L.jpg
         78         https://pbs.twimg.com/media/CUM8QZwW4AAVsBl.jpg
         93         https://pbs.twimg.com/media/CUOcVCwWsAERUKY.jpg
         94         https://pbs.twimg.com/media/CUObvUJVEAAnYPF.jpg
         96         https://pbs.twimg.com/media/CUQ7tv3W4AA3KlI.jpg
         98         https://pbs.twimg.com/media/CURiQMnUAAAPT2M.jpg
         100        https://pbs.twimg.com/media/CURwm3cUkAARcO6.jpg
         106        https://pbs.twimg.com/media/CUS9PlUWwAANeAD.jpg
         107        https://pbs.twimg.com/media/CUTDtyGXIAARxus.jpg
         112        https://pbs.twimg.com/media/CUTl5m1WUAAabZG.jpg
         115        https://pbs.twimg.com/media/CUT9PuQWwAABQv7.jpg
         117        https://pbs.twimg.com/media/CUW37BzWsAAlJlN.jpg
         118        https://pbs.twimg.com/media/CUXDGR2WcAAUQKz.jpg
         123        https://pbs.twimg.com/media/CUYEFlQXAAUkPGm.jpg
         130        https://pbs.twimg.com/media/CUZABzGW4AE5FOk.jpg
```

```
132       https://pbs.twimg.com/media/CUbfGbbWoAApZth.jpg
140       https://pbs.twimg.com/media/CUcl5jeWsAA6ufS.jpg
                                ...
1839      https://pbs.twimg.com/media/C59VqMUXEAAzldG.jpg
1844      https://pbs.twimg.com/media/C6RkiQZUsAAM4R4.jpg
1847      https://pbs.twimg.com/media/C6XBt9XXEAEEW9U.jpg
1851      https://pbs.twimg.com/media/C6mYrKOUwAANhep.jpg
1853      https://pbs.twimg.com/media/C6rBLenUOAAr8MN.jpg
1869      https://pbs.twimg.com/media/C7iNfq1WOAAcbsR.jpg
1886      https://pbs.twimg.com/media/C8SRpHNUIAARB3j.jpg
1887      https://pbs.twimg.com/media/C8SZH1EWAAAIRRF.jpg
1891      https://pbs.twimg.com/media/C8hwNxbXYAAwyVG.jpg
1892      https://pbs.twimg.com/media/C8lzFC4XcAAQxB4.jpg
1900      https://pbs.twimg.com/media/C9ECujZXsAAPCSM.jpg
1902      https://pbs.twimg.com/media/C8W6sY_WOAEmttW.jpg
1905    https://pbs.twimg.com/ext_tw_video_thumb/85222...
1906      https://pbs.twimg.com/media/C9QEqZ7XYAIR7fS.jpg
1910      https://pbs.twimg.com/media/C9eHyF7XgAAOxPM.jpg
1931      https://pbs.twimg.com/media/C-wLyufWOAA546I.jpg
1936      https://pbs.twimg.com/media/C-_9jWWUwAAnwkd.jpg
1937      https://pbs.twimg.com/media/C_BQ_NlVwAAgYGD.jpg
1940      https://pbs.twimg.com/media/C_KVJjDXsAEUCWn.jpg
1946      https://pbs.twimg.com/media/C_gQmaTUMAAPYSS.jpg
1953      https://pbs.twimg.com/media/C_O3NPeUQAAgrMl.jpg
1956      https://pbs.twimg.com/media/DAClmHkXcAA1kSv.jpg
1975      https://pbs.twimg.com/media/DBMV3NnXUAAmOPp.jpg
1979      https://pbs.twimg.com/media/DBW35ZsVoAEWZUU.jpg
2012      https://pbs.twimg.com/media/DDMD_phXoAQ1qf0.jpg
2021      https://pbs.twimg.com/media/DDm2Z5aXUAEDS2u.jpg
2022      https://pbs.twimg.com/media/DDrk-f9WAAI-WQv.jpg
2046      https://pbs.twimg.com/media/DE4fEDzWAAAyHMM.jpg
2052    https://pbs.twimg.com/ext_tw_video_thumb/88751...
2074      https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg
Name: jpg_url, Length: 324, dtype: object
```

Not all the image are for doges as the algorithm predicted.

1.1.1 Issues

Quality issues - Completeness: - *Missing data in images file - Missing data in JSON file* - Validity: - *Invalid timestamp type - Invalid dog names - Invalid rating denomnator - Invalid rating numerator* - Accuracy: - *Duplicated dog images - The data contain retweets that aren't from original user - Inaccurate images that are not for dogs*

- Consistency:
  - *Inconsistent dog rating*

Tidiness issues - Each variable forms a column in dog stages. - Delete columns that won't be used for analysis. - Each type of observational unit forms a table (archive_df, image_df, data_df).

1.2 Cleaning Data

```
In [27]: archive_clean = archive_df.copy()
         images_clean = image_df.copy()
         json_clean = json_df.copy()
```

Define

Missing data in **images_df** and **json_df** files. Keep the intersect (greatest common subset) of all files.

Code

```
In [28]: #assign the ids to keep to be as the smallest set (images)
         tweets_to_keep = set(images_clean.tweet_id)

         #assign json file to have the intersect tweet of images set and json set
         json_clean = json_clean[json_clean['tweet_id'].isin(tweets_to_keep)]

         #assign that ids to be the ids to keep
         tweets_to_keep = set(json_clean.tweet_id)

         #assign archive file to have the intersect tweet of tweet to keep
         archive_clean = archive_clean[archive_clean['tweet_id'].isin(tweets_to_keep)]

         #assign images file to have the intersect tweet of tweet to keep
         images_clean = images_clean[images_clean['tweet_id'].isin(tweets_to_keep)]
```

Test

```
In [29]: print('archive_clean tweet count = ' + str(len(archive_clean)))
         print('images_clean tweet count  = ' + str(len(images_clean)))
         print('json_clean tweet count    = ' + str(len(json_clean)),'\n')

         print('- All image set is in the archive set?  ' +
               str(images_clean.tweet_id.isin(archive_clean.tweet_id).all()))

         print('- All image set is in the json set? ' +
               str(images_clean.tweet_id.isin(json_clean.tweet_id).all()))
```

```
archive_clean tweet count = 2073
images_clean tweet count  = 2073
json_clean tweet count    = 2073

- All image set is in the archive set?  True
- All image set is in the json set? True
```

Define

Invalid timestamp type. change it to date.

Code

```
In [30]: archive_clean.timestamp = pd.to_datetime(archive_clean.timestamp)
```

Test

```
In [31]: archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2073 entries, 0 to 2355
Data columns (total 17 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
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
dtypes: datetime64[ns](1), float64(4), int64(3), object(9)
memory usage: 291.5+ KB
```

Define
Invalid dog names. change the wrong names to None.
Code

```
In [32]: archive_clean.set_value(archive_clean.name.str.islower() ,'name','None');
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:1: FutureWarning: set_value is depr
  """Entry point for launching an IPython kernel.
```

Test

```
In [33]: archive_clean.loc[ archive_clean.name.str.islower() ].name.value_counts()
```

```
Out[33]: Series([], Name: name, dtype: int64)
```

Define

- Invalid rating denomnator.
- Invalid rating numerator.

- Inconsistent dog rating.

remove tweets that do not contain the string '/10' in the tweet text. then manually correct the tweets which were wrongly identified.

Code

```
In [34]: # Inconsistent tweets
         tweets_to_remove = set(archive_clean[~archive_clean.text.str.contains('/10')].tweet_id)

         json_clean = json_clean[~json_clean['tweet_id'].isin(tweets_to_remove)]
         archive_clean = archive_clean[~archive_clean['tweet_id'].isin(tweets_to_remove)]
         images_clean = images_clean[~images_clean['tweet_id'].isin(tweets_to_remove)]

         # the wrong tweets
         tweets_to_correct = set(archive_clean[archive_clean.rating_denominator!=10].tweet_id)

In [35]: archive_clean.set_value(archive_clean.rating_denominator!=10 ,'rating_denominator',10);
         for ID in tweets_to_correct :
             tweet = archive_clean[archive_clean.tweet_id==ID].text
             numerator = tweet.str.extract(pat = '(..)/10')[0]
             archive_clean.set_value(archive_clean.tweet_id==ID,'rating_numerator', numerator);
         archive_clean.rating_numerator = archive_clean.rating_numerator.astype(int)

/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:1: FutureWarning: set_value is depr
  """Entry point for launching an IPython kernel.
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:5: FutureWarning: set_value is depr
  """
```

Test

```
In [36]: print('number of rating_denominator not equal 10 is ', len(archive_clean[archive_clean.
         set = archive_clean[archive_clean['tweet_id'].isin(tweets_to_correct)]
         print(list(set.text) ,'\n', set.rating_numerator )

number of rating_denominator not equal 10 is  0
['After so many requests, this is Bretagne. She was the last surviving 9/11 search dog, and our
 1068    14
1165    13
1202    11
1662    10
2335     9
Name: rating_numerator, dtype: int64
```

Define

Duplicated dog images. delete these tweets and keep the first one.

Code

```
In [37]: images_clean = images_clean.drop_duplicates(subset='jpg_url', keep='first')
```

Test

```
In [38]: sum(images_clean['jpg_url'].duplicated())
```

```
Out[38]: 0
```

Define
The data contain retweets that aren't from original user. delete them.
Code

```
In [39]: # Retweets
         tweets_to_remove =archive_clean[~archive_clean['retweeted_status_id'].isnull()].tweet_i

         json_clean = json_clean[~json_clean['tweet_id'].isin(tweets_to_remove)]
         archive_clean = archive_clean[~archive_clean['tweet_id'].isin(tweets_to_remove)]
         images_clean = images_clean[~images_clean['tweet_id'].isin(tweets_to_remove)]
```

Test

```
In [40]: archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1981 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                    1981 non-null int64
in_reply_to_status_id       22 non-null float64
in_reply_to_user_id         22 non-null float64
timestamp                   1981 non-null datetime64[ns]
source                      1981 non-null object
text                        1981 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               1981 non-null object
rating_numerator            1981 non-null int64
rating_denominator          1981 non-null int64
name                        1981 non-null object
doggo                       1981 non-null object
floofer                     1981 non-null object
pupper                      1981 non-null object
puppo                       1981 non-null object
dtypes: datetime64[ns](1), float64(4), int64(3), object(9)
memory usage: 278.6+ KB
```

Define
Inaccurate images that are not for dogs. delete it.
Code

```
In [41]: dog_type = []
         confidence_list = []

         def image(images_clean):
             if images_clean['p1_dog'] == True:
                 dog_type.append(images_clean['p1'])
                 confidence_list.append(images_clean['p1_conf'])
             elif images_clean['p2_dog'] == True:
                 dog_type.append(images_clean['p2'])
                 confidence_list.append(images_clean['p2_conf'])
             elif images_clean['p3_dog'] == True:
                 dog_type.append(images_clean['p3'])
                 confidence_list.append(images_clean['p3_conf'])
             else:
                 dog_type.append('Error')
                 confidence_list.append('Error')

         images_clean.apply(image, axis=1)

         #create new columns
         images_clean['dog_type'] = dog_type
         images_clean['confidence_list'] = confidence_list

         tweets_to_remove = images_clean[images_clean['dog_type'] == 'Error'].tweet_id

         json_clean = json_clean[~json_clean['tweet_id'].isin(tweets_to_remove)]
         archive_clean = archive_clean[~archive_clean['tweet_id'].isin(tweets_to_remove)]
         images_clean = images_clean[~images_clean['tweet_id'].isin(tweets_to_remove)]
```

Test

```
In [42]: images_clean.loc[(images_clean.p1_dog==False) & (images_clean.p2_dog==False) & (images_

Out[42]: Series([], Name: jpg_url, dtype: object)
```

Define

Each variable forms a column in dog stages. Convert doggo, floofer, pupper, puppo, and none to categories in a single column. drop the tweets that have more than one stage.

Code

```
In [43]: archive_clean.doggo = archive_clean.doggo.replace('None', 0)
         archive_clean.doggo = archive_clean.doggo.replace('doggo', 1)
         archive_clean.floofer = archive_clean.floofer.replace('None', 0)
         archive_clean.floofer = archive_clean.floofer.replace('floofer', 1)
         archive_clean.pupper = archive_clean.pupper.replace('None', 0)
         archive_clean.pupper = archive_clean.pupper.replace('pupper', 1)
         archive_clean.puppo = archive_clean.puppo.replace('None', 0)
         archive_clean.puppo = archive_clean.puppo.replace('puppo', 1)
```

```
        archive_clean['none'] = 1 - (archive_clean.doggo + archive_clean.floofer + archive_clea
        print('Duplicate stage categories = ' + str(archive_clean[archive_clean.none == -1].twe
```

```
Duplicate stage categories = 10
```

```
In [44]: tweets_to_remove = archive_clean[archive_clean.none == -1].tweet_id

        json_clean = json_clean[~json_clean['tweet_id'].isin(tweets_to_remove)]
        archive_clean = archive_clean[~archive_clean['tweet_id'].isin(tweets_to_remove)]
        images_clean = images_clean[~images_clean['tweet_id'].isin(tweets_to_remove)]
```

```
In [45]: values = ['doggo', 'floofer', 'pupper', 'puppo', 'none']
        ids = [x for x in list(archive_clean.columns) if x not in values]
        archive_clean = pd.melt(archive_clean, id_vars = ids, value_vars = values, var_name='st
```

```
In [46]: archive_clean = archive_clean[archive_clean.value == 1]
        archive_clean.drop('value', axis=1, inplace=True)
        archive_clean.reset_index(drop=True, inplace=True);
        archive_clean.stage = archive_clean.stage.astype('category')
```

Test

```
In [47]: archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1664 entries, 0 to 1663
Data columns (total 14 columns):
tweet_id                      1664 non-null int64
in_reply_to_status_id         18 non-null float64
in_reply_to_user_id           18 non-null float64
timestamp                     1664 non-null datetime64[ns]
source                        1664 non-null object
text                          1664 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                 1664 non-null object
rating_numerator              1664 non-null int64
rating_denominator            1664 non-null int64
name                          1664 non-null object
stage                         1664 non-null category
dtypes: category(1), datetime64[ns](1), float64(4), int64(3), object(5)
memory usage: 170.9+ KB
```

```
In [48]: archive_clean.stage.value_counts()
```

```
Out[48]: none       1414
         pupper      168
         doggo        54
         puppo        21
         floofer       7
         Name: stage, dtype: int64
```

Define
Delete columns that won't be used for analysis.
Code

```
In [49]: archive_clean.columns.tolist()
```

```
Out[49]: ['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',
          'rating_denominator',
          'name',
          'stage']
```

```
In [50]: #Delete columns
         archive_clean = archive_clean.drop(['in_reply_to_status_id',
                                             'in_reply_to_user_id',
                                             'source',
                                             'retweeted_status_id',
                                             'retweeted_status_user_id',
                                             'retweeted_status_timestamp',
                                             'expanded_urls',
                                             'rating_denominator' ], 1)
```

```
In [51]: json_clean.columns.tolist()
```

```
Out[51]: ['tweet_id', 'retweet_count', 'favorite_count']
```

```
In [52]: images_clean.columns.tolist()
```

```
Out[52]: ['tweet_id',
          'jpg_url',
          'img_num',
          'p1',
          'p1_conf',
```

```
          'p1_dog',
          'p2',
          'p2_conf',
          'p2_dog',
          'p3',
          'p3_conf',
          'p3_dog',
          'dog_type',
          'confidence_list']
```

```
In [53]: images_clean = images_clean.drop(['img_num',
                                           'p1',
                                           'p1_conf',
                                           'p1_dog',
                                           'p2',
                                           'p2_conf',
                                           'p2_dog',
                                           'p3',
                                           'p3_conf',
                                           'p3_dog'],1)
```

Test

```
In [54]: archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1664 entries, 0 to 1663
Data columns (total 6 columns):
tweet_id            1664 non-null int64
timestamp           1664 non-null datetime64[ns]
text                1664 non-null object
rating_numerator    1664 non-null int64
name                1664 non-null object
stage               1664 non-null category
dtypes: category(1), datetime64[ns](1), int64(2), object(2)
memory usage: 66.9+ KB
```

```
In [55]: archive_clean.rating_numerator
```

```
Out[55]: 0       14
         1       12
         2       12
         3       12
         4       12
         5       13
         6       13
         7       13
         8       12
```

```
9       13
10      13
11      14
12      12
13      12
14      12
15      11
16      13
17      14
18      12
19      13
20      12
21      13
22      13
23      14
24      11
25      14
26      11
27      11
28      11
29      11
        ..
1634     7
1635    10
1636     6
1637     7
1638    12
1639    10
1640     7
1641     9
1642    11
1643     8
1644    10
1645     9
1646     9
1647    11
1648    11
1649     8
1650     9
1651     6
1652    10
1653     9
1654    10
1655     8
1656     9
1657    10
1658    10
1659     5
```

```
1660     6
1661     9
1662     7
1663     8
Name: rating_numerator, Length: 1664, dtype: int64
```

```
In [56]: archive_clean.columns.tolist()
```

```
Out[56]: ['tweet_id', 'timestamp', 'text', 'rating_numerator', 'name', 'stage']
```

```
In [57]: json_clean.columns.tolist()
```

```
Out[57]: ['tweet_id', 'retweet_count', 'favorite_count']
```

```
In [58]: images_clean.columns.tolist()
```

```
Out[58]: ['tweet_id', 'jpg_url', 'dog_type', 'confidence_list']
```

Define

Each type of observational unit forms a table (archive_df, image_df, data_df). Consolidate archive and json and images into a single table.

Code

```
In [59]: print(archive_clean.shape)
         print(json_clean.shape)
         print(images_clean.shape)
         df_twitter = pd.merge(archive_clean,
                               images_clean,
                               how = 'left', on = ['tweet_id'])
         df_twitter = pd.merge(df_twitter, json_clean,
                               how = 'left', on = ['tweet_id'])
```

```
(1664, 6)
(1664, 3)
(1664, 4)
```

Test

All tables should be part of one dataset

```
In [60]: df_twitter.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1664 entries, 0 to 1663
Data columns (total 11 columns):
tweet_id            1664 non-null int64
timestamp           1664 non-null datetime64[ns]
text                1664 non-null object
rating_numerator    1664 non-null int64
name                1664 non-null object
```

```
stage               1664 non-null category
jpg_url             1664 non-null object
dog_type            1664 non-null object
confidence_list     1664 non-null object
retweet_count       1664 non-null int64
favorite_count      1664 non-null int64
dtypes: category(1), datetime64[ns](1), int64(4), object(5)
memory usage: 144.8+ KB
```

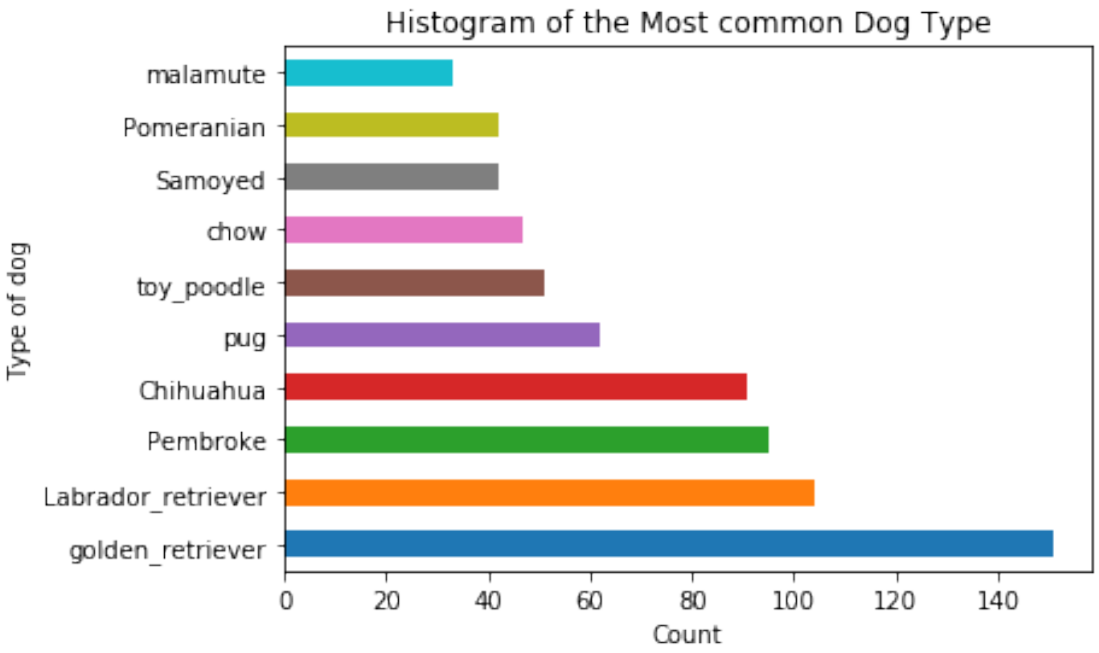
1.2.1 Store

```
In [61]: df_twitter.to_csv('twitter_archive_master.csv')
```

1.3 Analyzing and Visualizing

1.3.1 What is the most common dog type?

```
In [62]: df_dog_type = df_twitter.groupby('dog_type').filter(lambda x: len(x) >= 33)
         df_dog_type['dog_type'].value_counts().plot(kind = 'barh')

         plt.title('Histogram of the Most common Dog Type')
         plt.xlabel('Count')
         plt.ylabel('Type of dog')

         plt.show()
```



Golden retriever is the most common dog in this dataset then Labrador retriever then Pembroke.

1.3.2 What is the lowest rated dog type?

```
In [63]: df_twitter.groupby('dog_type').mean().rating_numerator.sort_values()
```

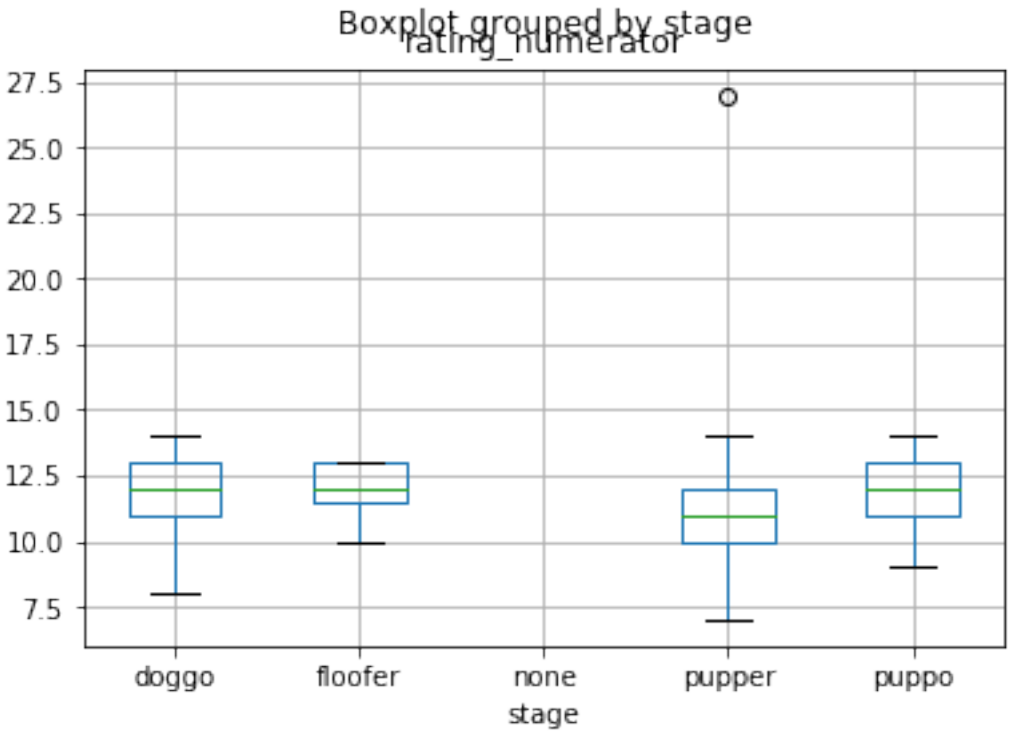
```
Out[63]: dog_type
         Japanese_spaniel                5.000000
         soft-coated_wheaten_terrier     8.538462
         Scotch_terrier                  9.000000
         curly-coated_retriever          9.000000
         Walker_hound                    9.000000
         Tibetan_terrier                 9.250000
         Boston_bull                     9.416667
         dalmatian                       9.500000
         Welsh_springer_spaniel          9.500000
         Dandie_Dinmont                  9.571429
         miniature_schnauzer             9.600000
         Norwich_terrier                 9.600000
         redbone                         9.666667
         Afghan_hound                    9.666667
         Maltese_dog                     9.736842
         Rhodesian_ridgeback             9.750000
         Scottish_deerhound              9.750000
         Airedale                        9.833333
         Newfoundland                    9.857143
         Mexican_hairless                9.857143
         Saint_Bernard                   9.857143
         English_setter                  9.875000
         miniature_poodle                9.875000
         Brabancon_griffon              10.000000
         Italian_greyhound              10.000000
         groenendael                    10.000000
         miniature_pinscher             10.000000
         papillon                       10.000000
         Irish_terrier                  10.000000
         Ibizan_hound                   10.000000
                                           ...
         Bernese_mountain_dog           11.272727
         Siberian_husky                 11.300000
         kelpie                         11.307692
         Greater_Swiss_Mountain_dog     11.333333
         Irish_water_spaniel            11.333333
         Leonberg                       11.333333
         Doberman                       11.333333
         cocker_spaniel                 11.333333
         chow                           11.404255
```

```
Eskimo_dog                    11.409091
Pembroke                      11.410526
Great_Pyrenees                11.428571
Norfolk_terrier               11.428571
wire-haired_fox_terrier       11.500000
giant_schnauzer               11.500000
Australian_terrier            11.500000
golden_retriever              11.556291
kuvasz                        11.611111
Samoyed                       11.690476
Irish_setter                  11.750000
Gordon_setter                 11.750000
silky_terrier                 12.000000
standard_schnauzer            12.000000
Border_terrier                12.142857
Tibetan_mastiff               12.250000
briard                        12.333333
Pomeranian                    12.476190
Saluki                        12.500000
Bouvier_des_Flandres          13.000000
clumber                       27.000000
Name: rating_numerator, Length: 113, dtype: float64
```

Japanese spaniel is the lowest rated dog in this dataset.

1.3.3 Which stage gets a higher rating?

```
In [64]: subset = df_twitter
         subset = subset.drop(subset[subset['stage'] == 'none'].index)
         subset.boxplot(column='rating_numerator', by='stage');
```



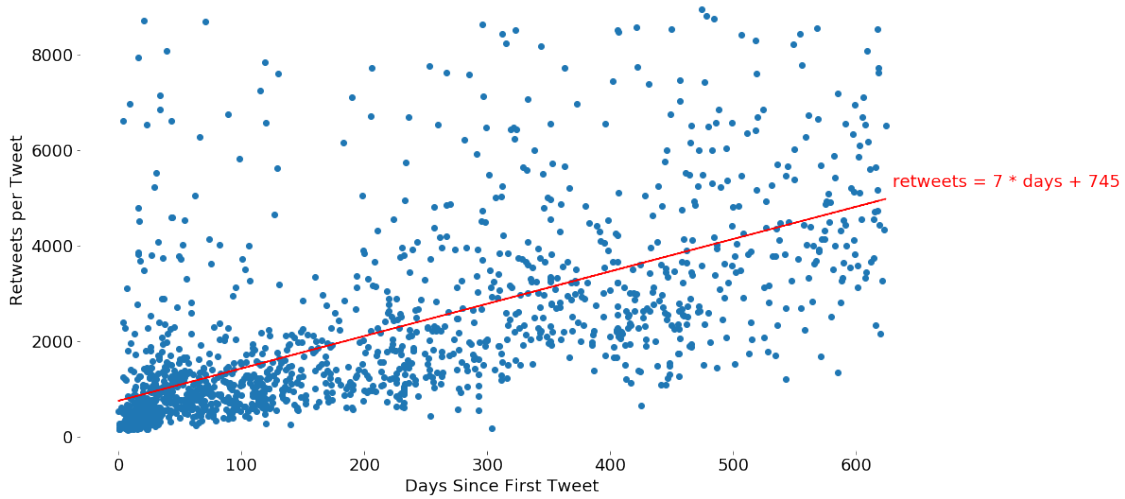
Floofer consistently gets high rating.

1.3.4 Is retweets on the @dog_rates tweets has increased over the time?

```
In [65]: q = df_twitter.retweet_count.quantile([0.05, 0.95])

         subset = df_twitter
         subset = subset.drop(df_twitter[df_twitter['retweet_count'] < q[0.05]].index)
         subset = subset.drop(df_twitter[df_twitter['retweet_count'] > q[0.95]].index)

         fig, ax = plt.subplots(1, 1, figsize=(16, 9))

         ax.spines['top'].set_visible(False)
         ax.spines['bottom'].set_visible(False)
         ax.spines['right'].set_visible(False)
         ax.spines['left'].set_visible(False)

         x = subset['timestamp'].dt.dayofyear + \
                 (subset['timestamp'].dt.year-2015)*365-319
         y = subset['retweet_count']

         plt.scatter(x, y);

         z = np.polyfit(x, y, 1)
```

```
p = np.poly1d(z)
plt.plot(x,p(x),"r--")

label = "retweets = %.0f * days + %.0f"%(z[0],z[1])

plt.text(630, 5220, label, fontsize=18, color='red')
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
plt.xlabel('Days Since First Tweet', fontsize=18)
plt.ylabel('Retweets per Tweet', fontsize=18)
```

Out[65]: Text(0,0.5,'Retweets per Tweet')



The retweets count doubled to sevenfold per day

In [ ]: