wrangle_act

November 8, 2022

1 Project: Wrangling and Analyze Data By Ashim Sharma

1.0.1 Table of Contents

1.1 1. Introduction

With the help of actual data, this project aims to hone data wrangling techniques. Three steps make up the data wrangling process: collect, evaluate, and clean. The tweet history of Twitter user @dog rates, commonly known as WeRateDogs, serves as the project's dataset. Twitter user WeRateDogs provides ratings and humorous comments for user's dogs.I'll start by gathering information from many sources in various formats. After that, I'll evaluate the data programmatically and visually to spot any data quality or organization problems. Then I'll perform programmatic cleaning to address all the problems. After that, I'll examine the cleaned dataset and display the findings.

1.2 2. Data Gathering

Assemble the three bits of information from various sources, one for each:

- Twitter archive from WeRateDogs: supplied to the project twitter archive enhanced.csv
- Using the Requests library and the URL https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599 image-predictions/image-predictions.tsv, image predictions.tsv was downloaded programmatically.
- Twitter API for like and retweet counts: Using the Requests library and the URL https://video.udacity-data.com/topher/2018/November/5be5fb7d tweet-json/tweet-json.txt, tweet json.txt was downloaded programmatically.
- Let's import our libraries to get going.

```
In [82]: # Importing the packages required for the project
         import pandas as pd
         import numpy as np
         import requests
```

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

In [83]:
```
# Import Manually Downloaded Data
df_twitter_archieve = pd.read_csv("twitter-archive-enhanced.csv")
df_twitter_archieve.head()
```

Out[83]:

	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
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	Twitter for iPhone	This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 13/1	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892420643555336193/photo/1
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	Twitter for iPhone	This is Tilly. She's just checking pup on you. Hopes you're doing ok. If not, she's	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892177421306343426/photo/1
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	Twitter for iPhone	This is Archie. He is a rare Norwegian Pouncing Corgo. Lives in the tall grass. You	NaN	NaN	NaN	https://twitter.com/dog_rates/status/891815181378084864/photo/1
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	Twitter for iPhone	This is Darla. She commenced a snooze mid meal. 13/10 happens to the best of us http	NaN	NaN	NaN	https://twitter.com/dog_rates/status/891689557279858688/photo/1
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	Twitter for iPhone	This is Franklin. He would like you to stop calling him "cute." He is a very fierce	NaN	NaN	NaN	https://twitter.com/dog_rates/status/891327558926688256/photo/1

```
4  https://twitter.com/dog_rates/status/891327558926688256/photo/1,https://twitter.com/

   rating_numerator  rating_denominator      name doggo floofer pupper puppo
0  13                10                   Phineas   None  None    None   None
1  13                10                   Tilly     None  None    None   None
2  12                10                   Archie    None  None    None   None
3  13                10                   Darla     None  None    None   None
4  12                10                   Franklin  None  None    None   None
```

1.3 2. Use the Requests library to download the tweet image prediction (image_predictions.tsv)

```
In [84]: #I downloaded the file using the specified URL and the Requests library.
         url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predicti
         response = requests.get(url)
         print(response)
         # Saving this file
         with open('image-predictions.tsv', mode = 'wb') as file:
             file.write(response.content)

         # Reading the TSV file
         image_prediction = pd.read_csv('image-predictions.tsv', sep = '\t')

         image_prediction.head()
```

```
<Response [200]>
```

```
Out[84]:              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

            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

             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
```

1.4 3. Use the Tweepy library to query additional data via the Twitter API (tweet_json.txt)

```
In [85]: # Download file using Requests library via URL provided
         url = 'https://video.udacity-data.com/topher/2018/November/5be5fb7d_tweet-json/tweet-js
         response = requests.get(url)

         # Saving the file
         with open('tweet-json.txt', mode = 'wb') as file:
             file.write(response.content)
```

```
In [86]: # Read downloaded txt file line by line into a pandas DataFrame
         df_list = []
         with open('tweet-json.txt', 'r') as file:
             lines = file.readlines()
             for line in lines:
                 parsed_json = json.loads(line)
                 df_list.append({'tweet_id': parsed_json['id'],
                                 'retweet_count': parsed_json['retweet_count'],
                                 'favorite_count': parsed_json['favorite_count']})

         tweet_json = pd.DataFrame(df_list, columns = ['tweet_id', 'retweet_count', 'favorite_co

         tweet_json.head()
```

Out[86]:

	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

1.5 4. Assessing Data

In order to find any difficulties with data tidiness (structural issues) or data quality (content), I will evaluate the data both visually and programmatically.

Dimensions of data quality:

- Completeness
- Validity
- Accuracy
- Consistency.

Needs for tidy data:

- Each parameter creates a column.
- Each note creates a row.
- A table is formed by each variety of observational unit.

1.6 Twitter Archieve Table

In [87]: df_twitter_archieve

Out[87]:

	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

                                                                                sourc
0     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
1     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
3     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
4     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
5     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
6     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
7     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
8     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
9     <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
10    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
11    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
12    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
13    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
```

```
14    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
15    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
16    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
17    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
18    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
19    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
20    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
21    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
22    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
23    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
24    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
25    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
26    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
27    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
28    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
29    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
...                                                                                 ..
2326  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2327  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2328  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2329  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2330  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2331  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2332  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2333  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2334  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2335  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2336  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2337  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2338  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2339  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2340  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2341  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2342  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2343  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2344  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2345  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2346  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2347  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2348  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2349  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2350  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2351  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2352  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2353  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2354  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
2355  <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
```

```
0     This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 1
1     This is Tilly. She's just checking pup on you. Hopes you're doing ok. If not, she
2     This is Archie. He is a rare Norwegian Pouncing Corgo. Lives in the tall grass. Y
3     This is Darla. She commenced a snooze mid meal. 13/10 happens to the best of us h
4     This is Franklin. He would like you to stop calling him "cute." He is a very fier
5     Here we have a majestic great white breaching off South Africa's coast. Absolutel
6     Meet Jax. He enjoys ice cream so much he gets nervous around it. 13/10 help Jax e
7     When you watch your owner call another dog a good boy but then they turn back to
8     This is Zoey. She doesn't want to be one of the scary sharks. Just wants to be a
9     This is Cassie. She is a college pup. Studying international doggo communication
10    This is Koda. He is a South Australian deckshark. Deceptively deadly. Frightening
11    This is Bruno. He is a service shark. Only gets out of the water to assist you. 1
12    Here's a puppo that seems to be on the fence about something haha no but seriousl
13    This is Ted. He does his best. Sometimes that's not enough. But it's ok. 12/10 wo
14    This is Stuart. He's sporting his favorite fanny pack. Secretly filled with bones
15    This is Oliver. You're witnessing one of his many brutal attacks. Seems to be pla
16    This is Jim. He found a fren. Taught him how to sit like the good boys. 12/10 for
17    This is Zeke. He has a new stick. Very proud of it. Would like you to throw it fo
18    This is Ralphus. He's powering up. Attempting maximum borkdrive. 13/10 inspiratio
19    RT @dog_rates: This is Canela. She attempted some fancy porch pics. They were uns
20    This is Gerald. He was just told he didn't get the job he interviewed for. A h*ck
21    This is Jeffrey. He has a monopoly on the pool noodles. Currently running a 'boop
22    I've yet to rate a Venezuelan Hover Wiener. This is such an honor. 14/10 paw-insp
23    This is Canela. She attempted some fancy porch pics. They were unsuccessful. 13/1
24    You may not have known you needed to see this today. 13/10 please enjoy (IG: emmy
25    This... is a Jubilant Antarctic House Bear. We only rate dogs. Please only send d
26    This is Maya. She's very shy. Rarely leaves her cup. 13/10 would find her an envi
27    This is Mingus. He's a wonderful father to his smol pup. Confirmed 13/10, but he
28    This is Derek. He's late for a dog meeting. 13/10 pet...al to the metal https://t
29    This is Roscoe. Another pupper fallen victim to spontaneous tongue ejections. Get
...
2326  This is quite the dog. Gets really excited when not in water. Not very soft tho.
2327  This is a southern Vesuvius bumblegruff. Can drive a truck (wow). Made friends wi
2328  Oh goodness. A super rare northeast Qdoba kangaroo mix. Massive feet. No pouch (d
2329  Those are sunglasses and a jean jacket. 11/10 dog cool af https://t.co/uHXrPkUEyl
2330  Unique dog here. Very small. Lives in container of Frosted Flakes (?). Short legs
2331  Here we have a mixed Asiago from the Galápagos Islands. Only one ear working. Big
2332  Look at this jokester thinking seat belt laws don't apply to him. Great tongue th
2333  This is an extremely rare horned Parthenon. Not amused. Wears shoes. Overall very
2334  This is a funny dog. Weird toes. Won't come down. Loves branch. Refuses to eat hi
2335  This is an Albanian 3 1/2 legged  Episcopalian. Loves well-polished hardwood floo
2336  Can take selfies 11/10 https://t.co/ws2AMaNwPW
2337  Very concerned about fellow dog trapped in computer. 10/10 https://t.co/0yxApIikp
2338  Not familiar with this breed. No tail (weird). Only 2 legs. Doesn't bark. Surpris
2339  Oh my. Here you are seeing an Adobe Setter giving birth to twins!!! The world is
2340  Can stand on stump for what seems like a while. Built that birdhouse? Impressive.
2341  This appears to be a Mongolian Presbyterian mix. Very tired. Tongue slip confirme
```

```
2342  Here we have a well-established sunblockerspaniel. Lost his other flip-flop. 6/10
2343  Let's hope this flight isn't Malaysian (lol). What a dog! Almost completely camou
2344  Here we have a northern speckled Rhododendron. Much sass. Gives 0 fucks. Good ton
2345  This is the happiest dog you will ever see. Very committed owner. Nice couch. 10/
2346  Here is the Rand Paul of retrievers folks! He's probably good at poker. Can drink
2347  My oh my. This is a rare blond Canadian terrier on wheels. Only $8.98. Rather doc
2348  Here is a Siberian heavily armored polar bear mix. Strong owner. 10/10 I would do
2349  This is an odd dog. Hard on the outside but loving on the inside. Petting still f
2350  This is a truly beautiful English Wilson Staff retriever. Has a nice phone. Privi
2351  Here we have a 1949 1st generation vulpix. Enjoys sweat tea and Fox News. Cannot
2352  This is a purebred Piers Morgan. Loves to Netflix and chill. Always looks like he
2353  Here is a very happy pup. Big fan of well-maintained decks. Just look at that ton
2354  This is a western brown Mitsubishi terrier. Upset about leaf. Actually 2 dogs her
2355  Here we have a Japanese Irish Setter. Lost eye in Vietnam (?). Big fan of relaxin

      retweeted_status_id  retweeted_status_user_id  \
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     8.874740e+17         4.196984e+09
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

     retweeted_status_timestamp  \
0     NaN
1     NaN
2     NaN
3     NaN
4     NaN
5     NaN
6     NaN
7     NaN
8     NaN
9     NaN
10    NaN
11    NaN
12    NaN
13    NaN
14    NaN
15    NaN
16    NaN
```

```
17    NaN
18    NaN
19    2017-07-19 00:47:34 +0000
20    NaN
21    NaN
22    NaN
23    NaN
24    NaN
25    NaN
26    NaN
27    NaN
28    NaN
29    NaN
...   ...
2326  NaN
2327  NaN
2328  NaN
2329  NaN
2330  NaN
2331  NaN
2332  NaN
2333  NaN
2334  NaN
2335  NaN
2336  NaN
2337  NaN
2338  NaN
2339  NaN
2340  NaN
2341  NaN
2342  NaN
2343  NaN
2344  NaN
2345  NaN
2346  NaN
2347  NaN
2348  NaN
2349  NaN
2350  NaN
2351  NaN
2352  NaN
2353  NaN
2354  NaN
2355  NaN


0     https://twitter.com/dog_rates/status/892420643555336193/photo/1
1     https://twitter.com/dog_rates/status/892177421306343426/photo/1
```

2	https://twitter.com/dog_rates/status/891815181378084864/photo/1
3	https://twitter.com/dog_rates/status/891689557279858688/photo/1
4	https://twitter.com/dog_rates/status/891327558926688256/photo/1,https://twitter.c
5	https://twitter.com/dog_rates/status/891087950875897856/photo/1
6	https://gofundme.com/ydvmve-surgery-for-jax,https://twitter.com/dog_rates/status/
7	https://twitter.com/dog_rates/status/890729181411237888/photo/1,https://twitter.c
8	https://twitter.com/dog_rates/status/890609185150312448/photo/1
9	https://twitter.com/dog_rates/status/890240255349198849/photo/1
10	https://twitter.com/dog_rates/status/890006608113172480/photo/1,https://twitter.c
11	https://twitter.com/dog_rates/status/889880896479866881/photo/1
12	https://twitter.com/dog_rates/status/889665388333682689/photo/1
13	https://twitter.com/dog_rates/status/889638837579907072/photo/1,https://twitter.c
14	https://twitter.com/dog_rates/status/889531135344209921/photo/1
15	https://twitter.com/dog_rates/status/889278841981685760/video/1
16	https://twitter.com/dog_rates/status/888917238123831296/photo/1
17	https://twitter.com/dog_rates/status/888804989199671297/photo/1,https://twitter.c
18	https://twitter.com/dog_rates/status/888554962724278272/photo/1,https://twitter.c
19	https://twitter.com/dog_rates/status/887473957103951883/photo/1,https://twitter.c
20	https://twitter.com/dog_rates/status/888078434458587136/photo/1,https://twitter.c
21	https://twitter.com/dog_rates/status/887705289381826560/photo/1
22	https://twitter.com/dog_rates/status/887517139158093824/video/1
23	https://twitter.com/dog_rates/status/887473957103951883/photo/1,https://twitter.c
24	https://twitter.com/dog_rates/status/887343217045368832/video/1
25	https://twitter.com/dog_rates/status/887101392804085760/photo/1
26	https://twitter.com/dog_rates/status/886983233522544640/photo/1,https://twitter.c
27	https://www.gofundme.com/mingusneedsus,https://twitter.com/dog_rates/status/88673
28	https://twitter.com/dog_rates/status/886680336477933568/photo/1
29	https://twitter.com/dog_rates/status/886366144734445568/photo/1,https://twitter.c
...	
2326	https://twitter.com/dog_rates/status/666411507551481857/photo/1
2327	https://twitter.com/dog_rates/status/666407126856765440/photo/1
2328	https://twitter.com/dog_rates/status/666396247373291520/photo/1
2329	https://twitter.com/dog_rates/status/666373753744588802/photo/1
2330	https://twitter.com/dog_rates/status/666362758909284353/photo/1
2331	https://twitter.com/dog_rates/status/666353288456101888/photo/1
2332	https://twitter.com/dog_rates/status/666345417576210432/photo/1
2333	https://twitter.com/dog_rates/status/666337882303524864/photo/1
2334	https://twitter.com/dog_rates/status/666293911632134144/photo/1
2335	https://twitter.com/dog_rates/status/666287406224695296/photo/1
2336	https://twitter.com/dog_rates/status/666273097616637952/photo/1
2337	https://twitter.com/dog_rates/status/666268910803644416/photo/1
2338	https://twitter.com/dog_rates/status/666104133288665088/photo/1
2339	https://twitter.com/dog_rates/status/666102155909144576/photo/1
2340	https://twitter.com/dog_rates/status/666099513787052032/photo/1
2341	https://twitter.com/dog_rates/status/666094000022159362/photo/1
2342	https://twitter.com/dog_rates/status/666082916733198337/photo/1
2343	https://twitter.com/dog_rates/status/666073100786774016/photo/1
2344	https://twitter.com/dog_rates/status/666071193221509120/photo/1

```
2345  https://twitter.com/dog_rates/status/666063827256086533/photo/1
2346  https://twitter.com/dog_rates/status/666058600524156928/photo/1
2347  https://twitter.com/dog_rates/status/666057090499244032/photo/1
2348  https://twitter.com/dog_rates/status/666055525042405380/photo/1
2349  https://twitter.com/dog_rates/status/666051853826850816/photo/1
2350  https://twitter.com/dog_rates/status/666050758794694657/photo/1
2351  https://twitter.com/dog_rates/status/666049248165822465/photo/1
2352  https://twitter.com/dog_rates/status/666044226329800704/photo/1
2353  https://twitter.com/dog_rates/status/666033412701032449/photo/1
2354  https://twitter.com/dog_rates/status/666029285002620928/photo/1
2355  https://twitter.com/dog_rates/status/666020888022790149/photo/1

      rating_numerator  rating_denominator      name  doggo floofer  pupper  \
0     13                10                    Phineas   None   None    None
1     13                10                    Tilly     None   None    None
2     12                10                    Archie    None   None    None
3     13                10                    Darla     None   None    None
4     12                10                    Franklin  None   None    None
5     13                10                    None      None   None    None
6     13                10                    Jax       None   None    None
7     13                10                    None      None   None    None
8     13                10                    Zoey      None   None    None
9     14                10                    Cassie    doggo  None    None
10    13                10                    Koda      None   None    None
11    13                10                    Bruno     None   None    None
12    13                10                    None      None   None    None
13    12                10                    Ted       None   None    None
14    13                10                    Stuart    None   None    None
15    13                10                    Oliver    None   None    None
16    12                10                    Jim       None   None    None
17    13                10                    Zeke      None   None    None
18    13                10                    Ralphus   None   None    None
19    13                10                    Canela    None   None    None
20    12                10                    Gerald    None   None    None
21    13                10                    Jeffrey   None   None    None
22    14                10                    such      None   None    None
23    13                10                    Canela    None   None    None
24    13                10                    None      None   None    None
25    12                10                    None      None   None    None
26    13                10                    Maya      None   None    None
27    13                10                    Mingus    None   None    None
28    13                10                    Derek     None   None    None
29    12                10                    Roscoe    None   None    pupper
...   ..                ..                       ...     ...    ...       ...
2326  2                 10                    quite     None   None    None
2327  7                 10                    a         None   None    None
2328  9                 10                    None      None   None    None
2329  11                10                    None      None   None    None
```

2330	6	10	None	None	None	None
2331	8	10	None	None	None	None
2332	10	10	None	None	None	None
2333	9	10	an	None	None	None
2334	3	10	a	None	None	None
2335	1	2	an	None	None	None
2336	11	10	None	None	None	None
2337	10	10	None	None	None	None
2338	1	10	None	None	None	None
2339	11	10	None	None	None	None
2340	8	10	None	None	None	None
2341	9	10	None	None	None	None
2342	6	10	None	None	None	None
2343	10	10	None	None	None	None
2344	9	10	None	None	None	None
2345	10	10	the	None	None	None
2346	8	10	the	None	None	None
2347	9	10	a	None	None	None
2348	10	10	a	None	None	None
2349	2	10	an	None	None	None
2350	10	10	a	None	None	None
2351	5	10	None	None	None	None
2352	6	10	a	None	None	None
2353	9	10	a	None	None	None
2354	7	10	a	None	None	None
2355	8	10	None	None	None	None

	puppo
0	None
1	None
2	None
3	None
4	None
5	None
6	None
7	None
8	None
9	None
10	None
11	None
12	puppo
13	None
14	puppo
15	None
16	None
17	None
18	None
19	None

```
20    None
21    None
22    None
23    None
24    None
25    None
26    None
27    None
28    None
29    None
...    ...
2326  None
2327  None
2328  None
2329  None
2330  None
2331  None
2332  None
2333  None
2334  None
2335  None
2336  None
2337  None
2338  None
2339  None
2340  None
2341  None
2342  None
2343  None
2344  None
2345  None
2346  None
2347  None
2348  None
2349  None
2350  None
2351  None
2352  None
2353  None
2354  None
2355  None

[2356 rows x 17 columns]
```

```
In [88]: df_twitter_archieve.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 [89]: df_twitter_archieve.describe()
```

Out[89]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	\
count	2.356000e+03	7.800000e+01	7.800000e+01	
mean	7.427716e+17	7.455079e+17	2.014171e+16	
std	6.856705e+16	7.582492e+16	1.252797e+17	
min	6.660209e+17	6.658147e+17	1.185634e+07	
25%	6.783989e+17	6.757419e+17	3.086374e+08	
50%	7.196279e+17	7.038708e+17	4.196984e+09	
75%	7.993373e+17	8.257804e+17	4.196984e+09	
max	8.924206e+17	8.862664e+17	8.405479e+17	

	retweeted_status_id	retweeted_status_user_id	rating_numerator	\
count	1.810000e+02	1.810000e+02	2356.000000	
mean	7.720400e+17	1.241698e+16	13.126486	
std	6.236928e+16	9.599254e+16	45.876648	
min	6.661041e+17	7.832140e+05	0.000000	
25%	7.186315e+17	4.196984e+09	10.000000	
50%	7.804657e+17	4.196984e+09	11.000000	
75%	8.203146e+17	4.196984e+09	12.000000	
max	8.874740e+17	7.874618e+17	1776.000000	

	rating_denominator
count	2356.000000
mean	10.455433
std	6.745237

```
min     0.000000
25%    10.000000
50%    10.000000
75%    10.000000
max    170.000000
```

```
In [90]: df_twitter_archieve.duplicated().sum()
```

```
Out[90]: 0
```

```
In [91]: df_twitter_archieve.tweet_id.duplicated().sum()
```

```
Out[91]: 0
```

```
In [92]: df_twitter_archieve.rating_numerator.value_counts()
```

```
Out[92]: 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 [93]: df_twitter_archieve.rating_denominator.value_counts()
```

```
Out[93]: 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 [94]: df_twitter_archieve.doggo.value_counts()
```

```
Out[94]: None     2259
         doggo    97
         Name: doggo, dtype: int64
```

```
In [95]: df_twitter_archieve.floofer.value_counts()
```

```
Out[95]: None       2346
         floofer    10
         Name: floofer, dtype: int64
```

```
In [96]: df_twitter_archieve.pupper.value_counts()
```

```
Out[96]: None      2099
         pupper    257
         Name: pupper, dtype: int64
```

```
In [97]: df_twitter_archieve.puppo.value_counts()
```

```
Out[97]: None     2326
         puppo    30
         Name: puppo, dtype: int64

In [98]: df_twitter_archieve.source.value_counts()

Out[98]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
         <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
         <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
         <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
         Name: source, dtype: int64
```

1.7 Findings:

- Retweeted status id, retweeted status user id, in reply to status id, and retweeted status id are float values that should all be ints. Retweets and replies entries should also be eliminated, as should related columns. Later, the image portion will be fixed.
- Remove +0000 from the timestamp and change it to datetime if timestamp is str.
- All canine stages, such as doggo, floofer, pupper, and puppo, should be in one column with aberrant numbers in the rating denominator, such as 170, 150, 130, etc. Almost always, the rating denominator is 10 out of the rating numerator's usual values, such as 1776, 960, 666, 204, 165, etc., make no sense.Source data redundant and difficult to read

1.8 Image Prediction Table Operations

```
In [99]: image_prediction

Out[99]:                 tweet_id  \
         0     666020888022790149
         1     666029285002620928
         2     666033412701032449
         3     666044226329800704
         4     666049248165822465
         5     666050758794694657
         6     666051853826850816
         7     666055525042405380
         8     666057090499244032
         9     666058600524156928
         10    666063827256086533
         11    666071193221509120
         12    666073100786774016
         13    666082916733198337
         14    666094000022159362
         15    666099513787052032
         16    666102155909144576
         17    666104133288665088
         18    666268910803644416
         19    666273097616637952
         20    666287406224695296
```

```
21    666293911632134144
22    666337882303524864
23    666345417576210432
24    666353288456101888
25    666362758909284353
26    666373753744588802
27    666396247373291520
28    666407126856765440
29    666411507551481857
...                   ...
2045  886366144734445568
2046  886680336477933568
2047  886736880519319552
2048  886983233522544640
2049  887101392804085760
2050  887343217045368832
2051  887473957103951883
2052  887517139158093824
2053  887705289381826560
2054  888078434458587136
2055  888202515573088257
2056  888554962724278272
2057  888804989199671297
2058  888917238123831296
2059  889278841981685760
2060  889531135344209921
2061  889638837579907072
2062  889665388333682689
2063  889880896479866881
2064  890006608113172480
2065  890240255349198849
2066  890609185150312448
2067  890729181411237888
2068  890971913173991426
2069  891087950875897856
2070  891327558926688256
2071  891689557279858688
2072  891815181378084864
2073  892177421306343426
2074  892420643555336193
```

```
0     https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
1     https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
2     https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
3     https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
4     https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg
5     https://pbs.twimg.com/media/CT5Jof1WUAEuVxN.jpg
```

6	https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg
7	https://pbs.twimg.com/media/CT5N9tpXIAAifs1.jpg
8	https://pbs.twimg.com/media/CT5PY9OWoAAQGLo.jpg
9	https://pbs.twimg.com/media/CT5Qw94XAAA_2dP.jpg
10	https://pbs.twimg.com/media/CT5Vg_wXIAAXfnj.jpg
11	https://pbs.twimg.com/media/CT5cN_3WEAAlOoZ.jpg
12	https://pbs.twimg.com/media/CT5d9DZXAAALcwe.jpg
13	https://pbs.twimg.com/media/CT5m4VGWEAAtKc8.jpg
14	https://pbs.twimg.com/media/CT5w9gUW4AAsBNN.jpg
15	https://pbs.twimg.com/media/CT51-JJUEAA6hV8.jpg
16	https://pbs.twimg.com/media/CT54YGiWUAEZnoK.jpg
17	https://pbs.twimg.com/media/CT56LSZWoAAlJj2.jpg
18	https://pbs.twimg.com/media/CT8QCd1WEAADXws.jpg
19	https://pbs.twimg.com/media/CT8T1mtUwAA3aqm.jpg
20	https://pbs.twimg.com/media/CT8g3BpUEAAuFjg.jpg
21	https://pbs.twimg.com/media/CT8mx7KW4AEQu8N.jpg
22	https://pbs.twimg.com/media/CT9OwFIWEAMuRje.jpg
23	https://pbs.twimg.com/media/CT9Vn7PWoAA_ZCM.jpg
24	https://pbs.twimg.com/media/CT9cxOtUEAAhNN_.jpg
25	https://pbs.twimg.com/media/CT9lXGsUcAAyUFt.jpg
26	https://pbs.twimg.com/media/CT9vZEYWUAAlZO5.jpg
27	https://pbs.twimg.com/media/CT-D2ZHWIAA3gK1.jpg
28	https://pbs.twimg.com/media/CT-NvwmW4AAugGZ.jpg
29	https://pbs.twimg.com/media/CT-RugiWIAELEaq.jpg
...	...
2045	https://pbs.twimg.com/media/DEOBTnQUwAApKEH.jpg
2046	https://pbs.twimg.com/media/DE4fEDzWAAAyHMM.jpg
2047	https://pbs.twimg.com/media/DE5Se8FXcAAJFx4.jpg
2048	https://pbs.twimg.com/media/DE8yicJWOAAAvBJ.jpg
2049	https://pbs.twimg.com/media/DE-eAq6UwAA-jaE.jpg
2050	https://pbs.twimg.com/ext_tw_video_thumb/887343120832229379/pu/img/6HSuFrW1lzI_9M
2051	https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg
2052	https://pbs.twimg.com/ext_tw_video_thumb/887517108413886465/pu/img/WanJKwssZj4VJv
2053	https://pbs.twimg.com/media/DFHDQBbXgAEqY7t.jpg
2054	https://pbs.twimg.com/media/DFMWn56WsAAkA7B.jpg
2055	https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg
2056	https://pbs.twimg.com/media/DFTH_O-UQAACu2O.jpg
2057	https://pbs.twimg.com/media/DFWra-3VYAA2piG.jpg
2058	https://pbs.twimg.com/media/DFYRgsOUQAARGhO.jpg
2059	https://pbs.twimg.com/ext_tw_video_thumb/889278779352338437/pu/img/VlbFB3v8H8VwzV
2060	https://pbs.twimg.com/media/DFg_2PVWOAEHN3p.jpg
2061	https://pbs.twimg.com/media/DFihzFfXsAYGDPR.jpg
2062	https://pbs.twimg.com/media/DFi579UWsAAatzw.jpg
2063	https://pbs.twimg.com/media/DFl99B1WsAITKsg.jpg
2064	https://pbs.twimg.com/media/DFnwSY4WAAAMliS.jpg
2065	https://pbs.twimg.com/media/DFrEyVuWOAAO3t9.jpg
2066	https://pbs.twimg.com/media/DFwUU__XcAEpyXI.jpg
2067	https://pbs.twimg.com/media/DFyBahAVwAAhUTd.jpg

```
2068  https://pbs.twimg.com/media/DF1eOmZXUAALUcq.jpg
2069  https://pbs.twimg.com/media/DF3HwyEWsAABqE6.jpg
2070  https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg
2071  https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg
2072  https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg
2073  https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg
2074  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 [100]: image_prediction.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 [101]: image_prediction.describe()
```

Out[101]:

	tweet_id	img_num	p1_conf	p2_conf	p3_conf
count	2.075000e+03	2075.000000	2075.000000	2.075000e+03	2.075000e+03
mean	7.384514e+17	1.203855	0.594548	1.345886e-01	6.032417e-02
std	6.785203e+16	0.561875	0.271174	1.006657e-01	5.090593e-02
min	6.660209e+17	1.000000	0.044333	1.011300e-08	1.740170e-10
25%	6.764835e+17	1.000000	0.364412	5.388625e-02	1.622240e-02
50%	7.119988e+17	1.000000	0.588230	1.181810e-01	4.944380e-02
75%	7.932034e+17	1.000000	0.843855	1.955655e-01	9.180755e-02
max	8.924206e+17	4.000000	1.000000	4.880140e-01	2.734190e-01

```
In [102]: image_prediction.tweet_id.duplicated().sum()
```

```
Out[102]: 0
```

```
In [103]: image_prediction.jpg_url.duplicated().sum()
```

```
Out[103]: 66
```

```
In [104]: image_prediction.p1.value_counts()
```

Out[104]:

golden_retriever	150
Labrador_retriever	100
Pembroke	89
Chihuahua	83
pug	57
chow	44
Samoyed	43
toy_poodle	39
Pomeranian	38
malamute	30
cocker_spaniel	30
French_bulldog	26
Chesapeake_Bay_retriever	23
miniature_pinscher	23
seat_belt	22
Siberian_husky	20
Staffordshire_bullterrier	20
German_shepherd	20
Cardigan	19
web_site	19
Eskimo_dog	18
Maltese_dog	18
teddy	18
beagle	18
Shetland_sheepdog	18
Lakeland_terrier	17
Shih-Tzu	17
Rottweiler	17
kuvasz	16
Italian_greyhound	16
	..
hand_blower	1
cuirass	1
lorikeet	1
convertible	1
timber_wolf	1
polecat	1
mortarboard	1
minibus	1
traffic_light	1
military_uniform	1
sliding_door	1
silky_terrier	1
clumber	1
lynx	1
bow	1
peacock	1
tiger_shark	1

```
water_buffalo               1
conch                       1
coffee_mug                  1
mud_turtle                  1
bald_eagle                  1
microphone                  1
china_cabinet               1
stove                       1
crash_helmet                1
snowmobile                  1
pencil_box                  1
microwave                   1
cowboy_boot                 1
Name: p1, Length: 378, dtype: int64
```

```
In [105]: image_prediction.p2.value_counts()
```

```
Out[105]: Labrador_retriever               104
          golden_retriever                 92
          Cardigan                         73
          Chihuahua                        44
          Pomeranian                       42
          French_bulldog                   41
          Chesapeake_Bay_retriever         41
          toy_poodle                       37
          cocker_spaniel                   34
          Siberian_husky                   33
          miniature_poodle                 33
          beagle                           28
          collie                           27
          Eskimo_dog                       27
          Pembroke                         27
          kuvasz                           26
          Italian_greyhound                22
          Pekinese                         21
          American_Staffordshire_terrier   21
          malinois                         20
          miniature_pinscher               20
          Samoyed                          20
          chow                             20
          toy_terrier                      20
          Boston_bull                      19
          Norwegian_elkhound               19
          Staffordshire_bullterrier        18
          Irish_terrier                    17
          pug                              17
          kelpie                           16
                                           ..
```

```
dugong                          1
space_heater                    1
mashed_potato                   1
wallaby                         1
armadillo                       1
coral_reef                      1
home_theater                    1
cowboy_boot                     1
white_wolf                      1
menu                            1
coral_fungus                    1
cornet                          1
toucan                          1
bow                             1
banded_gecko                    1
Japanese_spaniel                1
dining_table                    1
water_bottle                    1
snail                           1
dock                            1
pier                            1
rule                            1
hotdog                          1
tiger                           1
promontory                      1
hair_slide                      1
shower_curtain                  1
timber_wolf                     1
killer_whale                    1
medicine_chest                  1
Name: p2, Length: 405, dtype: int64
```

```
In [106]: image_prediction.p3.value_counts()
```

```
Out[106]: Labrador_retriever               79
          Chihuahua                        58
          golden_retriever                 48
          Eskimo_dog                       38
          kelpie                           35
          kuvasz                           34
          chow                             32
          Staffordshire_bullterrier        32
          beagle                           31
          cocker_spaniel                   31
          Pomeranian                       29
          Pekinese                         29
          toy_poodle                       29
          Chesapeake_Bay_retriever         27
```

```
Pembroke                         27
Great_Pyrenees                   27
French_bulldog                   26
malamute                         26
American_Staffordshire_terrier   24
pug                              23
Cardigan                         23
basenji                          21
bull_mastiff                     20
toy_terrier                      20
Siberian_husky                   19
Shetland_sheepdog                17
Boston_bull                      17
doormat                          16
boxer                            16
Lakeland_terrier                 16
                                 ..
drumstick                        1
hatchet                          1
space_shuttle                    1
mushroom                         1
rhinoceros_beetle                1
cuirass                          1
sea_cucumber                     1
stinkhorn                        1
pot                              1
beach_wagon                      1
bow                              1
maze                             1
pajama                           1
shovel                           1
passenger_car                    1
great_grey_owl                   1
pier                             1
screw                            1
plunger                          1
wing                             1
padlock                          1
nipple                           1
American_black_bear              1
restaurant                       1
cowboy_boot                      1
consomme                         1
neck_brace                       1
kimono                           1
chimpanzee                       1
standard_schnauzer               1
Name: p3, Length: 408, dtype: int64
```

1.9 Evaluation of the Image Prediction Table:

- Mismatched capitalization in columns p1, p2, and p3 and duplicated jpg URL's were found.
- Many elements in the twitter archive table should be excluded because they are not dogs, such as the jaguar, mailbox, peacock, cloak, etc. For this study, the most certain dog breed prediction is all that is required.

1.10 Tweet json table Operations:

```
In [107]: tweet_json.head()
```

```
Out[107]:              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
```

```
In [108]: tweet_json.duplicated().sum()
```

```
Out[108]: 0
```

2 Quality Issues:

2.1 Findings and Summaries:

Twitter Archieve Table:

- Only original ratings with photos are required; retweets and responses entries should be eliminated, along with related columns. The following columns in the twitter archive table should all be str: in reply to status id, in reply to user id, retweeted status id, and retweeted status user id. Later, we'll fix the picture component.
- Remove +0000 from timestamp aberrant numbers in rating denominator, such as 170, 150, 130, etc., and timestamp is str, should be datetime. Most frequently, the rating denominator There are 10 aberrant values in the rating numerator that are illogical, such as 1776, 960, 666, 204, 165, etc.

2.1.1 Image Prediction table:

- data sources redundant and challenging to read table with predicted images p1, p2, and p3 columns have erroneous capitalisation and duplicate jpg urls
- There are numerous entries that aren't dogs, such as a jaguar, postbox, peacock, cloak, etc.
- For this investigation, only the most certain dog breed predictions will do.

2.1.2 Tweet json Table:

- missing data in twitter archive perhaps as a result of retweets

2.1.3 Uniformity and Tidiness of data:

- Twitter archive organization: doggo, floofer, pupper, and puppo are all dog stage names, and they should all be in one column.
- According to the standards for clean data, the three tables should be consolidated into one because they are all connected to the same kind of observational unit.

2.2 5. Cleaning Data

We will use the programmatic way to clean the data. We will use the inline steps to implement:

- Describe: transform our evaluations into clearly defined cleaning chores.
- Coding: Transform those definitions into code and execute it.
- Testing: Test the dataset to ensure that the cleaning processes were successful, either visually or through coding.

```
In [109]: # Make copies of original pieces of data
          df_twitter_archive_clean = df_twitter_archieve.copy()
          df_image_prediction_clean = image_prediction.copy()
          df_tweet_json_clean = tweet_json.copy()
```

2.3 Issue #1:

2.3.1 df_twitter_archieve:Retweets and replies are not desired; only original ratings are desired.

2.3.2 Define:

Only keep rows where the retweeted status id field contains NaN by using the isnull() filter. The same procedure is used for in reply to status id.

Code

```
In [110]: # Eliminate retweets
          df_twitter_archive_clean = df_twitter_archive_clean[df_twitter_archive_clean.retweeted

          # Eliminate replies
          df_twitter_archive_clean = df_twitter_archive_clean[df_twitter_archive_clean.in_reply_
```

Test

```
In [111]: df_twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                    2097 non-null int64
in_reply_to_status_id       0 non-null float64
in_reply_to_user_id         0 non-null float64
timestamp                   2097 non-null object
```

```
source                        2097 non-null object
text                          2097 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                 2094 non-null object
rating_numerator              2097 non-null int64
rating_denominator            2097 non-null int64
name                          2097 non-null object
doggo                         2097 non-null object
floofer                       2097 non-null object
pupper                        2097 non-null object
puppo                         2097 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 294.9+ KB
```

2.4 Issue #2:

2.4.1 df_twitter_archieve:Remove the retweets and replies-related columns. The datatype issue with those columns will be resolved after they are dropped.

Define: Retweeted status id, Retweeted status user id, and Retweeted status timestamp columns can all be deleted with df.drop.

Code

```
In [112]: df_twitter_archive_clean = df_twitter_archive_clean.drop(['in_reply_to_status_id',
                                        'in_reply_to_user_id',
                                        'retweeted_status_id',
                                        'retweeted_status_user_id',
                                        'retweeted_status_timestamp'],axis=1)
```

Test

```
In [113]: df_twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2355
Data columns (total 12 columns):
tweet_id              2097 non-null int64
timestamp             2097 non-null object
source                2097 non-null object
text                  2097 non-null object
expanded_urls         2094 non-null object
rating_numerator      2097 non-null int64
rating_denominator    2097 non-null int64
name                  2097 non-null object
doggo                 2097 non-null object
```

```
floofer               2097 non-null object
pupper                2097 non-null object
puppo                 2097 non-null object
dtypes: int64(3), object(9)
memory usage: 213.0+ KB
```

2.5 Issue #3:

2.5.1 df_twitter_archieve:timestamp datatype should be datetime; delete +0000

Define: To convert a timestamp from str to datetime, remove +0000 and use pd.to datetime to do so.

Code

```
In [114]: # Remove +0000
          df_twitter_archive_clean.timestamp = df_twitter_archive_clean.timestamp.str[:-6]

          # Convert to datetime
          df_twitter_archive_clean.timestamp = pd.to_datetime(df_twitter_archive_clean.timestamp
```

Test

```
In [115]: df_twitter_archive_clean.timestamp.head()

Out[115]: 0   2017-08-01 16:23:56
          1   2017-08-01 00:17:27
          2   2017-07-31 00:18:03
          3   2017-07-30 15:58:51
          4   2017-07-29 16:00:24
          Name: timestamp, dtype: datetime64[ns]
```

2.6 Issue #4:

2.6.1 df_twitter_archieve:Duplicate source information, substitute shorter category names for the lengthy url

Define: Use replace to substitute short category names for the url.

Code

```
In [116]: df_twitter_archive_clean.source.value_counts()

Out[116]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
          <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
          <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
          <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
          Name: source, dtype: int64
```

```
In [117]: df_twitter_archive_clean.source = df_twitter_archive_clean.source.replace({'Twitter fo
                                                                                     'Vine - Make a Sc
                                                                                     'Twitter Web Clie
                                                                                     'TweetDeck': 'Twe
```

Test

```
In [118]: df_twitter_archive_clean.source.value_counts()
```

```
Out[118]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
          <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
          <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
          <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
          Name: source, dtype: int64
```

2.7 Issue #5:

2.7.1 Image Prediction Table:Delete any dog-related entries. The same is true of tweet json and twitter archive. It would be simpler to address concerns with abnormal rating in rating denominator and rating numerator since many anomalous rating values would be gone.

Define: Use isin function to filter and remove rows from all three dataframes that have p1 dog, p2 dog, and p3 dog columns that are False.

Code

```
In [119]: # Rows in the data before cleaning
          print(df_image_prediction_clean.shape)
          print(df_twitter_archive_clean.shape)
          print(df_tweet_json_clean.shape)
```

```
(2075, 12)
(2097, 12)
(2354, 3)
```

```
In [120]: #Put tweet ids from postings that are not about dogs into a drop list after filtering
          df_image_prediction_clean.query('p1_dog == False and p2_dog == False and p3_dog == Fal
          drop_list = df_image_prediction_clean.query('p1_dog == False and p2_dog == False and p

          # Drop the rows with tweet_id in the drop_list in all dataframes
          df_image_prediction_clean = df_image_prediction_clean[~df_image_prediction_clean.tweet
          df_twitter_archive_clean = df_twitter_archive_clean[~df_twitter_archive_clean.tweet_id
          df_tweet_json_clean = df_tweet_json_clean[~df_tweet_json_clean.tweet_id.isin(drop_list
```

Test

```
In [121]: # Number of rows remaining after data cleaning
          print(df_image_prediction_clean.shape)
          print(df_twitter_archive_clean.shape)
          print(df_tweet_json_clean.shape)
```

```
(1751, 12)
(1792, 12)
(2031, 3)
```

2.8 Issue #6:

2.8.1 Twitter Archieve Table:incorrect values in the rating denominator. The project overview states that the ratings typically have a denominator of 10. After deleting the ratings that weren't for dogs, a lot of the aberrant rating values were gone, which made fixing the anomalous rating easy. Further research revealed that tweets with a denominator less than 10 were typically numerous dogs.

Define: The columns tweet id, text, rating numerator, and rating denominator should be added to a new dataframe. To correct these ratings, filter for rating denominator not equal to 10 and then read the text.

Code

```
In [122]: # Creating new dataframe with selected columns
          df_abnor_rating = df_twitter_archive_clean[['tweet_id', 'text', 'rating_numerator', 'r

          # Filter rating_denominator not equal to 10
          df_abnor_denominator = df_abnor_rating.query('rating_denominator != 10')

          # Display full text
          pd.set_option('display.max_colwidth', -1)

          df_abnor_denominator
```

```
Out[122]:                tweet_id  \
          433   820690176645140481
          516   810984652412424192
          902   758467244762497024
          1068  740373189193256964
          1165  722974582966214656
          1202  716439118184652801
          1228  713900603437621249
          1254  710658690886586372
          1274  709198395643068416
          1351  704054845121142784
          1433  697463031882764288
```

```
1635  684222868335505415
1662  682962037429899265
1779  677716515794329600
1843  675853064436391936
2335  666287406224695296


433   The floofs have been released I repeat the floofs have been released. 84/70 http
516   Meet Sam. She smiles 24/7 &amp;amp; secretly aspires to be a reindeer. \nKeep Sam sm
902   Why does this never happen at my front door... 165/150 https://t.co/HmwrdfEfUE
1068  After so many requests, this is Bretagne. She was the last surviving 9/11 search
1165  Happy 4/20 from the squad! 13/10 for all https://t.co/eV1diwds8a
1202  This is Bluebert. He just saw that both #FinalFur match ups are split 50/50. Ama
1228  Happy Saturday here's 9 puppers on a bench. 99/90 good work everybody https://t.
1254  Here's a brigade of puppers. All look very prepared for whatever happens next. 8
1274  From left to right:\nCletus, Jerome, Alejandro, Burp, &amp;amp; Titson\nNone know wh
1351  Here is a whole flock of puppers.  60/50 I'll take the lot https://t.co/9dpcw6Md
1433  Happy Wednesday here's a bucket of pups. 44/40 would pet all at once https://t.c
1635  Someone help the girl is being mugged. Several are distracting her while two ste
1662  This is Darrel. He just robbed a 7/11 and is in a high speed police chase. Was j
1779  IT'S PUPPERGEDDON. Total of 144/120 ...I think https://t.co/ZanVtAtvIq
1843  Here we have an entire platoon of puppers. Total score: 88/80 would pet all at c
2335  This is an Albanian 3 1/2 legged  Episcopalian. Loves well-polished hardwood flo

      rating_numerator  rating_denominator
433   84                70
516   24                7
902   165               150
1068  9                 11
1165  4                 20
1202  50                50
1228  99                90
1254  80                80
1274  45                50
1351  60                50
1433  44                40
1635  121               110
1662  7                 11
1779  144               120
1843  88                80
2335  1                 2
```

In [123]:
```
# Correction of ratings by reading through the text, most of the abnormal ratings are
# tweet_id: 666287406224695296
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 666287406224695296,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 666287406224695296,
# tweet_id: 697463031882764288 ---> Several Dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 697463031882764288,
```

```
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 697463031882764288,
# tweet_id: 684222868335505415 ---> several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 684222868335505415,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 684222868335505415,
# tweet_id: 682962037429899265
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 682962037429899265,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 682962037429899265,
# tweet_id: 710658690886586372 --- several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 710658690886586372,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 710658690886586372,
# tweet_id: 713900603437621249 --- several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 713900603437621249,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 713900603437621249,
# tweet_id: 709198395643068416 --- several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 709198395643068416,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 709198395643068416,
# tweet_id: 722974582966214656
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 722974582966214656,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 722974582966214656,
# tweet_id: 716439118184652801
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 716439118184652801,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 716439118184652801,
# tweet_id: 704054845121142784 --- several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 704054845121142784,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 704054845121142784,
# tweet_id: 677716515794329600 --- several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 677716515794329600,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 677716515794329600,
# tweet_id: 675853064436391936 --- several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 675853064436391936,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 675853064436391936,
# tweet_id: 810984652412424192  rating missing
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 810984652412424192,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 810984652412424192,
# tweet_id: 820690176645140481 --- several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 820690176645140481,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 820690176645140481,
# tweet_id: 731156023742988288 --- several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 731156023742988288,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 731156023742988288,
# tweet_id: 758467244762497024 --- several dogs
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 758467244762497024,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 758467244762497024,
# tweet_id: 740373189193256964
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 740373189193256964,
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 740373189193256964,
```

Test

```
In [124]: df_twitter_archive_clean.rating_denominator.value_counts()
```

```
Out[124]: 10    1792
          Name: rating_denominator, dtype: int64
```

2.9 Issue #7:

2.9.1 Twitter Archieve Table:Irregular values in the rating numerator. After deleting the ratings that weren't for dogs, several anomalous rating levels disappeared.

Define:

- To identify unusual values, use value counts; to fix the ratings, verify the text.
- To eliminate entires that are not dogs, use isin and~.

Code

```
In [125]: df_twitter_archive_clean.rating_numerator.value_counts()
```

```
Out[125]: 12    464
          10    380
          11    379
          13    256
          9     136
          8     71
          7     31
          14    27
          6     16
          5     15
          4     6
          3     5
          2     2
          75    1
          27    1
          26    1
          0     1
          Name: rating_numerator, dtype: int64
```

```
In [126]: df_abnor_rating.query('rating_numerator == 75 or rating_numerator == 26 or rating_nume
```

```
Out[126]:                 tweet_id  \
          315   835152434251116546
          695   786709082849828864
          763   778027034220126208
          1712  680494726643068929


          315   When you're so blinded by your systematic plagiarism that you forget what day it
```

```
695   This is Logan, the Chow who lived. He solemnly swears he's up to lots of good. H
763   This is Sophie. She's a Jubilant Bush Pupper. Super h*ckin rare. Appears at rand
1712  Here we have uncovered an entire battalion of holiday puppers. Average of 11.26/

      rating_numerator  rating_denominator
315   0                 10
695   75                10
763   27                10
1712  26                10
```

In [127]:
```
# Correcting the ratings
# tweet_id: 786709082849828864, rating _numberator should be 9.75 according to the tea
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 786709082849828864,
# tweet_id: 680494726643068929, rating _numberator should be 11.26 according to the te
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 680494726643068929,
# tweet_id: 778027034220126208, rating _numberator should be 11.27 according to the te
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 778027034220126208,
# tweet_id: 835152434251116546
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 835152434251116546,
# tweet_id: 883482846933004288
df_twitter_archive_clean.loc[df_twitter_archive_clean.tweet_id == 883482846933004288,
```

In [128]:
```
# rating_numerator 3 and 4
df_abnor_rating.query('rating_numerator == 3 or rating_numerator == 4')
```

Out[128]:
```
                tweet_id  \
765   777885040357281792
1004  747816857231626240
1165  722974582966214656
1189  718246886998687744
1249  711306686208872448
1303  707420581654872064
1701  680940246314430465
1938  673906403526995968
2070  671122204919246848
2183  668989615043424256
2288  667176164155375616
2316  666649482315059201


765   This is Wesley. He's clearly trespassing. Seems rather h*ckin violent too. Weapo
1004  Viewer discretion is advised. This is a terrible attack in progress. Not even in
1165  Happy 4/20 from the squad! 13/10 for all https://t.co/eV1diwds8a
1189  This is Alexanderson. He's got a weird ass birth mark. Dreadful at fetch. Won't
1249  What hooligan sent in pictures w/out a dog in them? Churlish af. 3/10 just bc th
1303  This is Keurig. He's a rare dog. Laughs like an idiot tho. Head is basically a w
1701  This is Alice. She's an idiot. 4/10 https://t.co/VQXdwJfkyS
1938  Guys I'm getting real tired of this. We only rate dogs. Please don't send in oth
```

```
2070  Two miniature golden retrievers here. Webbed paws. Don't walk very efficiently.
2183  This is Bernie. He's taking his Halloween costume very seriously. Wants to be ba
2288  These are strange dogs. All have toupees. Long neck for dogs. In a shed of sorts
2316  Cool dog. Enjoys couch. Low monotone bark. Very nice kicks. Pisses milk (must be

      rating_numerator  rating_denominator
765   3                 10
1004  4                 10
1165  4                 20
1189  3                 10
1249  3                 10
1303  4                 10
1701  4                 10
1938  3                 10
2070  4                 10
2183  3                 10
2288  4                 10
2316  4                 10
```

rating_numerator = 3,is not a dog,So we can delete

- 778885040357281792:
- 718246886998687744:
- 673906403526995968: #### rating_numerator = 4,is not a dog,So we can delete
- 707420581654872064
- 680940246314430465
- 671122204919246848
- 667176164155375616
- 666649482315059201

```
In [129]: # removing the I for the data which are not Dogs
          id_list = [778885040357281792, 718246886998687744, 673906403526995968, 707420581654872
          df_twitter_archive_clean = df_twitter_archive_clean[~df_twitter_archive_clean.tweet_id
```

Test

```
In [130]: df_twitter_archive_clean.rating_numerator.value_counts()

Out[130]: 12.00    464
          10.00    380
          11.00    380
          13.00    256
          9.00     136
          8.00     71
```

```
7.00     31
14.00    27
6.00     16
5.00     14
3.00     2
2.00     2
9.75     1
11.26    1
4.00     1
13.50    1
11.27    1
Name: rating_numerator, dtype: int64
```

2.9.2 Constraint: There are still some rating problems. For instance, some ratings still don't apply to dogs even after many items from the image prediction table have been eliminated. It would be impractical to read them all.

2.10 Issue #8:

2.10.1 Image Prediction Table:We simply need to use the image forecast that is the most certain.

Define:

- Add two new columns for breed and confidence level.
- Create a function to sort through the predictions and identify the one that is most certain to be a breed of dog. The most certain prediction is p1, which is followed by p2 and p3.
- Removing any unnecessary columns.

Code

```
In [131]: # Create a dog_breed column and a confidence_level column
          dog_breed = []
          confidence_level = []

          # Create a function to find the most confidence prediction that is a dog_breed
          # p1 is the most confidence prediction, followed by p2 and p3
          def image_pred(df_image_prediction_clean):
              if df_image_prediction_clean.p1_dog == True:
                  dog_breed.append(df_image_prediction_clean.p1)
                  confidence_level.append(df_image_prediction_clean.p1_conf)
              elif df_image_prediction_clean.p2_dog == True:
                  dog_breed.append(df_image_prediction_clean.p2)
                  confidence_level.append(df_image_prediction_clean.p2_conf)
              elif df_image_prediction_clean.p3_dog == True:
                  dog_breed.append(df_image_prediction_clean.p3)
                  confidence_level.append(df_image_prediction_clean.p3_conf)
              else:
                  dog_breed.append('Unknown_dog_breed')
```

```
            confidence_level.append(0)

    # Apply the function by column
    df_image_prediction_clean.apply(image_pred, axis=1)

    # Add the dog_breed and confidence_level column to df_image_prediction_clean
    df_image_prediction_clean['dog_breed'] = dog_breed
    df_image_prediction_clean['confidence_level'] = confidence_level

    # Drop columns no longer needed
    df_image_prediction_clean = df_image_prediction_clean.drop(['img_num',
                                                                'p1','p1_conf','p1_dog',
                                                                'p2','p2_conf','p2_dog',
                                                                'p3','p3_conf','p3_dog'],axis=1)
```

Test

```
In [132]: df_image_prediction_clean.head()
```

```
Out[132]:              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

                       dog_breed  confidence_level
          0  Welsh_springer_spaniel          0.465074
          1                redbone          0.506826
          2        German_shepherd          0.596461
          3    Rhodesian_ridgeback          0.408143
          4      miniature_pinscher          0.560311
```

2.11 Issue #9:

2.11.1 Image Prediction Table:Irregular capitalization in the p1 column

Define: To capitalize the first letter, use str.capitalize function fronm the pandas library.

Code

```
In [133]: df_image_prediction_clean.dog_breed = df_image_prediction_clean.dog_breed.str.capitali
```

```
In [134]: df_image_prediction_clean.head(15)
```

```
Out[134]:              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
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_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

                dog_breed  confidence_level
0   Welsh_springer_spaniel  0.465074
1   Redbone                 0.506826
2   German_shepherd         0.596461
3   Rhodesian_ridgeback     0.408143
4   Miniature_pinscher      0.560311
5   Bernese_mountain_dog    0.651137
7   Chow                    0.692517
8   Golden_retriever        0.007959
9   Miniature_poodle        0.201493
10  Golden_retriever        0.775930
11  Gordon_setter           0.503672
12  Walker_hound            0.260857
13  Pug                     0.489814
14  Bloodhound              0.195217
15  Lhasa                   0.582330
```

3 6. Tidiness

3.0.1 1. df_twitter_archive data: All canine stages, including doggo, floofer, pupper, and puppo, should be in a single column.

Define: The doggo, floofer, pupper, and puppo columns can be melted into a type and dog stage column using pd.melt. Remove the middle column.

Code

```
In [135]: # Melt the doggo, floofer, pupper and puppo columns to type and dogs_phase column
          df_twitter_archive_clean = pd.melt(df_twitter_archive_clean,
                                             id_vars = ['tweet_id', 'timestamp', 'source', 'text',
                                             value_vars = ['doggo', 'floofer', 'pupper', 'puppo'],
                                             var_name = 'type',
                                             value_name = 'dog_phase')

          # Drop type column
```

```
df_twitter_archive_clean.drop('type', 1, inplace = True)

# Sort by dog_phase and drop duplicates
df_twitter_archive_clean = df_twitter_archive_clean.sort_values('dog_phase').drop_dupl
```

Test

```
In [136]: df_twitter_archive_clean.dog_phase.value_counts()
```

```
Out[136]: None       1494
          pupper     194
          doggo      63
          puppo      23
          floofer    10
          Name: dog_phase, dtype: int64
```

3.0.2 2. df_twitter_archive data:The twitter archive table should contain the tweet json file.

Define:Join the retweet count and favorite count columns to the tweet id field in the twitter archive database.

Code

```
In [137]: df_twitter_archive_clean = pd.merge(df_twitter_archive_clean, df_tweet_json_clean,
                                    on = ['tweet_id'], how = 'left')
```

Test

```
In [138]: df_twitter_archive_clean.head()
```

```
Out[138]:              tweet_id           timestamp  \
          0  667470559035432960 2015-11-19 22:32:36
          1  667491009379606528 2015-11-19 23:53:52
          2  667495797102141441 2015-11-20 00:12:54
          3  667502640335572993 2015-11-20 00:40:05
          4  667509364010450944 2015-11-20 01:06:48

                                                                source  \
          0  <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
          1  <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
          2  <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
          3  <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
          4  <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>


          0  This is a northern Wahoo named Kohl. He runs this town. Chases tumbleweeds. Draws g
          1  Two dogs in this one. Both are rare Jujitsu Pythagoreans. One slightly whiter than
          2  This is Philippe from Soviet Russia. Commanding leader. Misplaced other boot. Hung
          3  Say hello to Hall and Oates. Oates is winking and Hall is contemplating the artisti
```

```
4  This a Norwegian Pewterschmidt named Tickles. Ears for days. 12/10 I care deeply fo

                                                    expanded_urls  \
0  https://twitter.com/dog_rates/status/667470559035432960/photo/1
1  https://twitter.com/dog_rates/status/667491009379606528/photo/1
2  https://twitter.com/dog_rates/status/667495797102141441/photo/1
3  https://twitter.com/dog_rates/status/667502640335572993/photo/1
4  https://twitter.com/dog_rates/status/667509364010450944/photo/1

   rating_numerator  rating_denominator      name dog_phase  retweet_count  \
0  11.0              10                   a         None      102
1  7.0               10                   None      None      242
2  9.0               10                   Philippe  None      294
3  11.0              10                   Hall      None      231
4  12.0              10                   None      None      2272

   favorite_count
0  273
1  559
2  565
3  563
4  7148
```

3.0.3 3. The twitter archive table needs to include image prediction. Rows with images should only be retained because we only want original ratings that include them.

Define:

- Using the tweet id as a joining factor, use merge to combine the image prediction table and the twitter archive table.
- Only nonnull rows should be retained after using the notnull filter.

Code

```
In [139]: # Joining tables using left join
          df_twitter_archive_clean = pd.merge(df_twitter_archive_clean, df_image_prediction_clea
                                      on = ['tweet_id'], how = 'left')

          # number of null values before cleaning
          df_twitter_archive_clean.jpg_url.isnull().sum()
```

```
Out[139]: 126
```

```
In [140]: df_twitter_archive_clean = (df_twitter_archive_clean[df_twitter_archive_clean.jpg_url.
```

Test

```
In [141]: # number of null values after cleaning
          df_twitter_archive_clean.jpg_url.isnull().sum()
```

```
Out[141]: 0
```

3.1 Storing Data

Save gathered, assessed, and cleaned master dataset to a CSV file named "twitter_archive_master.csv".

```
In [142]: df_twitter_archive_clean.head()  ## Viewing the data
```

Out[142]:

	tweet_id	timestamp	source	text	expanded_urls	rating_numerator	rating_denominator	name	dog_phase	retweet_count	favorite_count	jpg_url
0	667470559035432960	2015-11-19 22:32:36	Twitter Web Client	This is a northern Wahoo named Kohl. He runs this town. Chases tumbleweeds. Draws g	https://twitter.com/dog_rates/status/667470559035432960/photo/1	11.0	10	a	None	102	273	https://pbs.twimg.com/media/CUNU78YWEAECmpB.jpg
1	667491009379606528	2015-11-19 23:53:52	Twitter Web Client	Two dogs in this one. Both are rare Jujitsu Pythagoreans. One slightly whiter than	https://twitter.com/dog_rates/status/667491009379606528/photo/1	7.0	10	None	None	242	559	https://pbs.twimg.com/media/CUNniSlUYAEj1Jl.jpg
2	667495797102141441	2015-11-20 00:12:54	Twitter Web Client	This is Philippe from Soviet Russia. Commanding leader. Misplaced other boot. Hung	https://twitter.com/dog_rates/status/667495797102141441/photo/1	9.0	10	Philippe	None	294	565	https://pbs.twimg.com/media/CUNr4-7UwAAg2lq.jpg
3	667502640335572993	2015-11-20 00:40:05	Twitter Web Client	Say hello to Hall and Oates. Oates is winking and Hall is contemplating the artisti	https://twitter.com/dog_rates/status/667502640335572993/photo/1	11.0	10	Hall	None	231	563	https://pbs.twimg.com/media/CUNyHTMUYAAQVch.jpg
4	667509364010450944	2015-11-20 01:06:48	Twitter Web Client	This a Norwegian Pewterschmidt named Tickles. Ears for days. 12/10 I care deeply fo	https://twitter.com/dog_rates/status/667509364010450944/photo/1	12.0	10	None	None	2272	7148	https://pbs.twimg.com/media/CUN4Or5UAAAa5K4.jpg

```
                dog_breed  confidence_level
0  Toy_poodle          0.304175
1  Borzoi              0.852088
2  Chihuahua           0.143957
3  Labrador_retriever  0.996709
4  Beagle              0.636169
```

```
In [143]: df_twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1658 entries, 0 to 1783
Data columns (total 14 columns):
tweet_id              1658 non-null int64
timestamp             1658 non-null datetime64[ns]
source                1658 non-null object
text                  1658 non-null object
expanded_urls         1658 non-null object
rating_numerator      1658 non-null float64
rating_denominator    1658 non-null int64
name                  1658 non-null object
dog_phase             1658 non-null object
retweet_count         1658 non-null int64
favorite_count        1658 non-null int64
jpg_url               1658 non-null object
dog_breed             1658 non-null object
confidence_level      1658 non-null float64
dtypes: datetime64[ns](1), float64(2), int64(4), object(7)
memory usage: 194.3+ KB
```

```
In [144]: # Store the clean dataframe in a CSV file named twitter_archive_master.csv
          df_twitter_archive_clean.to_csv('twitter_archive_master.csv')

          # load data to a dataframe
          df = pd.read_csv('twitter_archive_master.csv')
```

```
In [ ]:
```

3.2 Analyzing and Visualizing Data

3.2.1 Popular dog breeds determined on the basis of the following:

- the quantity of unique tweets
- total number of retweets
- total number of favorites

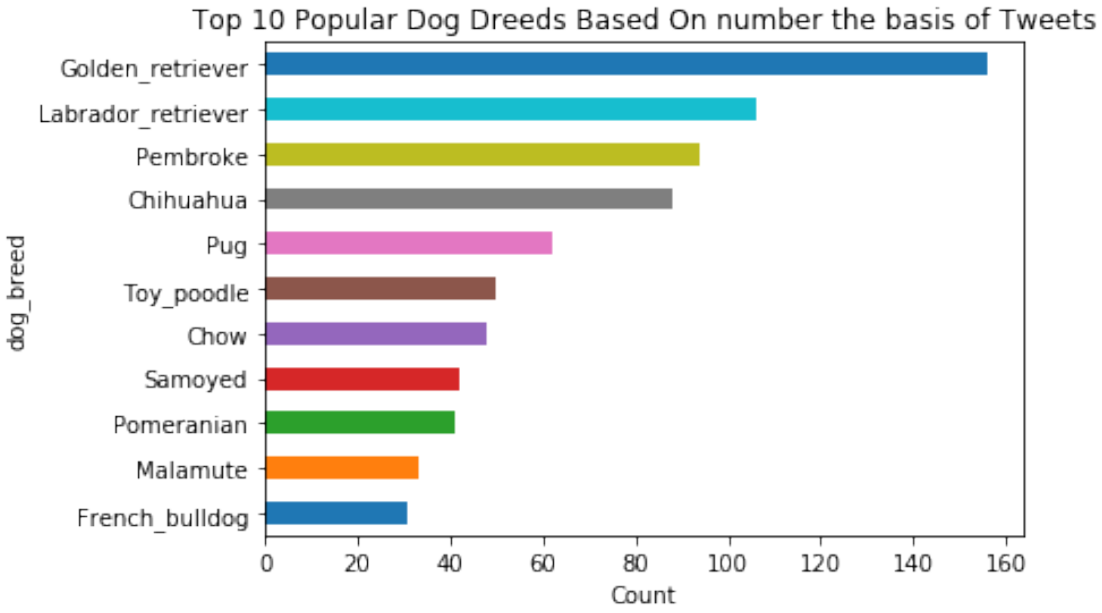
```
In [145]: df.dog_breed.value_counts()
```

```
Out[145]: Golden_retriever                  156
          Labrador_retriever                106
```

```
Pembroke                        94
Chihuahua                       88
Pug                             62
Toy_poodle                      50
Chow                            48
Samoyed                         42
Pomeranian                      41
Malamute                        33
French_bulldog                  31
Chesapeake_bay_retriever        31
Cocker_spaniel                  30
Miniature_pinscher              24
Eskimo_dog                      22
Cardigan                        21
German_shepherd                 21
Shih-tzu                        20
Beagle                          20
Siberian_husky                  20
Staffordshire_bullterrier       20
Maltese_dog                     19
Rottweiler                      18
Shetland_sheepdog               18
Italian_greyhound               17
Lakeland_terrier                17
Basset                          17
Kuvasz                          16
American_staffordshire_terrier  16
West_highland_white_terrier     16
                                ..
Tibetan_mastiff                 4
Scottish_deerhound              4
Bluetick                        4
Weimaraner                      4
Gordon_setter                   4
Komondor                        3
Cairn                           3
Giant_schnauzer                 3
Curly-coated_retriever          3
Briard                          3
Toy_terrier                     3
Irish_water_spaniel             3
Brabancon_griffon               3
Greater_swiss_mountain_dog      3
Leonberg                        3
Wire-haired_fox_terrier         2
Appenzeller                     2
Groenendael                     2
Sussex_spaniel                  2
```

```
Afghan_hound                    2
Black-and-tan_coonhound         2
Australian_terrier              2
Silky_terrier                   1
Irish_wolfhound                 1
Entlebucher                     1
Bouvier_des_flandres            1
Clumber                         1
Standard_schnauzer              1
Scotch_terrier                  1
Japanese_spaniel                1
Name: dog_breed, Length: 113, dtype: int64
```

In [146]:
```
# Create a bar graph showing the top 10 dog breeds based on the number of posts.
df.dog_breed.value_counts()[10::-1].plot(kind = 'barh', title = 'Top 10 Popular Dog Dr
plt.xlabel('Count')
plt.ylabel('dog_breed');
```

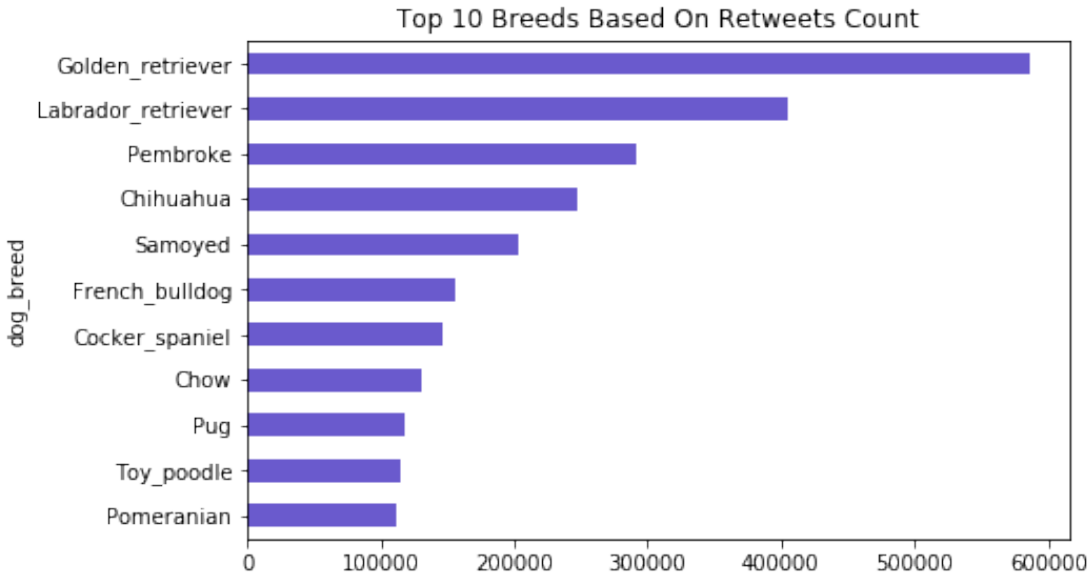


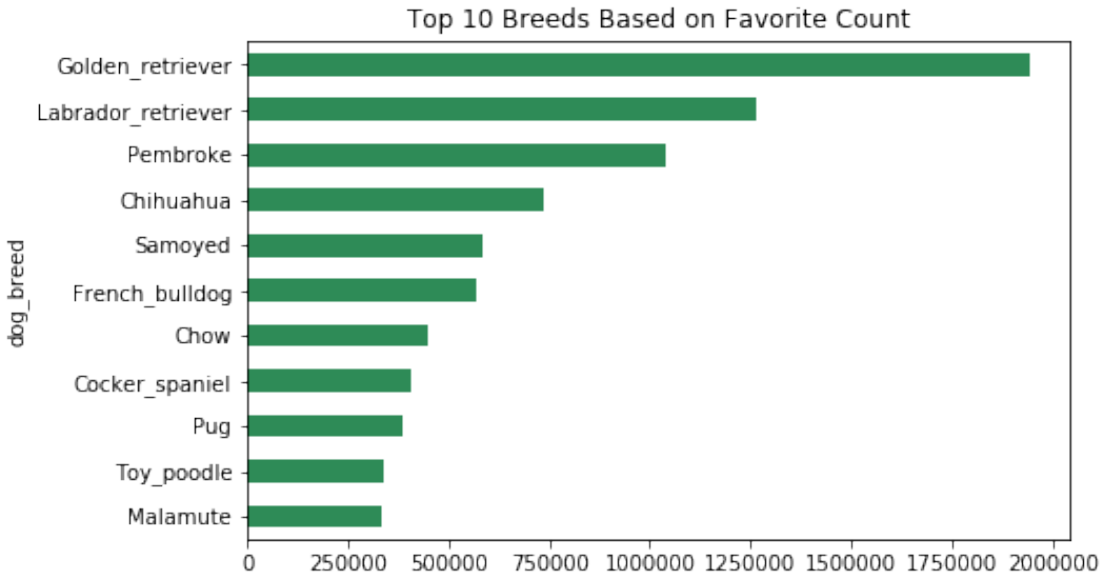
3.3 Insights from the above Graph:

To keep tweets about dogs solely, we sanitized the image prediction data. The Golden Retriever is the most popular dog breed, according to the graph, which shows 156 tweets about it. 106 tweets on the Labrador retriever make it the second most popular breed. Pembroke (94), Chihuahua (88), and Pug (62), followed in decreasing order by other breeds, are the following three breeds. Then, using data from favorites and retweets, we'll plot the most popular dog breeds.

In [147]:
```
# Data to plot
columns = ['dog_breed', 'retweet_count', 'favorite_count']
```

```
df_dog_breed = df[columns]

dog_breed_retweet = df_dog_breed.groupby('dog_breed')['retweet_count'].agg('sum').sort
dog_breed_favorite = df_dog_breed.groupby('dog_breed')['favorite_count'].agg('sum').so

# Plot horizontal bar chart
fig, (ax1, ax2) = plt.subplots(2, 1)

# Top 10 breeds based on number of retweets
dog_breed_retweet.plot.barh(ax=ax1, figsize=(7,10), color='#6A5ACD')
ax1.set_title("Top 10 Breeds Based On Retweets Count")

# Top 10 breeds based on number of favorite
dog_breed_favorite.plot.barh(ax=ax2, color='#2E8B57')
ax2.set_title("Top 10 Breeds Based on Favorite Count")

fig.subplots_adjust(hspace=0.3)
```





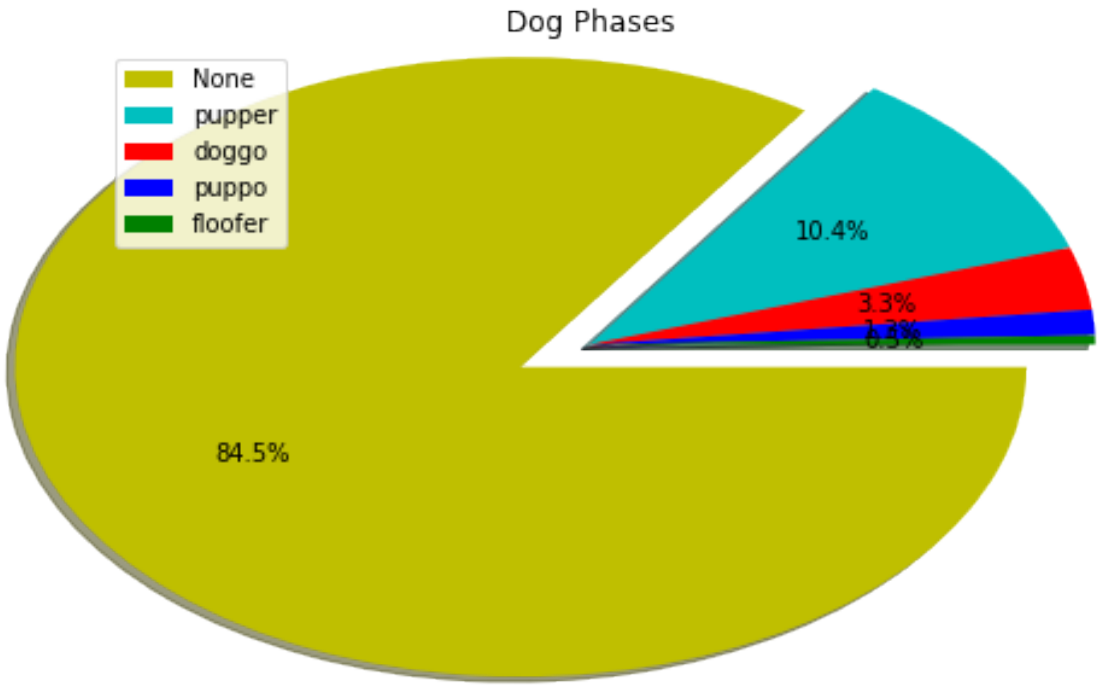
3.4 Insights From the above Bar graphs:

Golden retrievers, Labrador retrievers, Pembroke, Chihuahua, Pug, Toy poodle, Chow, and Samoyed are the top 10 most popular breeds according to the number of tweets. In decreasing order, Pomeranian, Malamute, and Chesapeake Bay retriever. Golden retrievers, Labrador retrievers, Pembroke, Chihuahua, Samoyed, French bulldog, Cocker spaniels, Chow, Pug, Toy poodles, and Pomeranians are the top 10 breeds according to the number of retweets. Top 10 breeds based on favorite count are very similar to retweet count (with the exception of the 5th-ranked French bulldog, 6th-ranked Chow, 7th-ranked Cocker spaniel, and 10th-ranked Malamute). This is likely because there may be a correlation between favorites and retweets since people who retweet are

more likely to click favorites. The three graphs show the same pattern, showing that the top four dog breeds in popularity are the golden retriever, labrador retriever, pembroke, and chihuahua.

3.5 2. What is the most common Dog Phase?

```
In [148]: df.dog_phase.value_counts(normalize=True)

Out[148]: None       0.844994
          pupper     0.104343
          doggo      0.032569
          puppo      0.013269
          floofer    0.004825
          Name: dog_phase, dtype: float64

In [149]: # Plot pie chart
          labels = ['None', 'pupper', 'doggo', 'puppo', 'floofer']
          values = df.dog_phase.value_counts(normalize=True)
          colors = ['y', 'c', 'r' ,'b', 'g']
          explode = (0.2, 0, 0, 0, 0)

          plt.pie(values, colors=colors, explode=explode, autopct='%1.1f%%', radius = 1.3, shado
          plt.legend(labels, loc=0)
          plt.title('Dog Phases')
          plt.tight_layout()
```



3.6 Insights from Pie Chart

According to the pie chart, more than 80% of tweets do not include information about the dog phase in the article. Pupper is the most frequent phase among all those tweets for individuals who have the phase information.

3.7 3. Rating the numerator

We'll concentrate on rating numerator here because the denominator of a rating is almost always 10, and because the dataset has been cleaned.

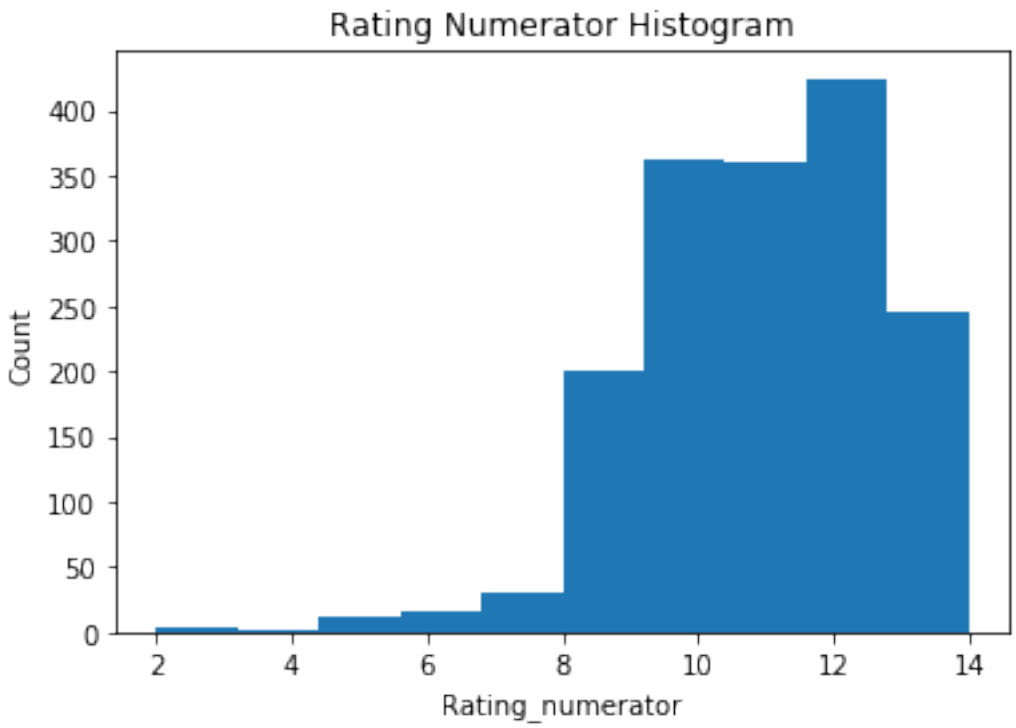
```
In [150]: df.rating_numerator.value_counts()
```

```
Out[150]: 12.00    424
          10.00    361
          11.00    359
          13.00    222
          9.00     133
          8.00     68
          7.00     31
          14.00    22
          6.00     16
          5.00     13
          2.00     2
          3.00     2
          11.26    1
          13.50    1
          9.75     1
          4.00     1
          11.27    1
          Name: rating_numerator, dtype: int64
```

```
In [151]: df.rating_numerator.describe()
```

```
Out[151]: count    1658.000000
          mean     10.868384
          std      1.683681
          min      2.000000
          25%      10.000000
          50%      11.000000
          75%      12.000000
          max      14.000000
          Name: rating_numerator, dtype: float64
```

```
In [152]: plt.hist(df.rating_numerator)
          plt.title('Rating Numerator Histogram')
          plt.xlabel('Rating_numerator')
          plt.ylabel('Count');
```



As we can see, the most popular ratings are 12 with 424 tweets, followed by 10.00, 11.00, 13 (222 tweets), and 9.00. (133 tweets). The rating is 10.87 on average.

3.7.1 4. Based on the number of tweets, popular dog breeds receive average ratings

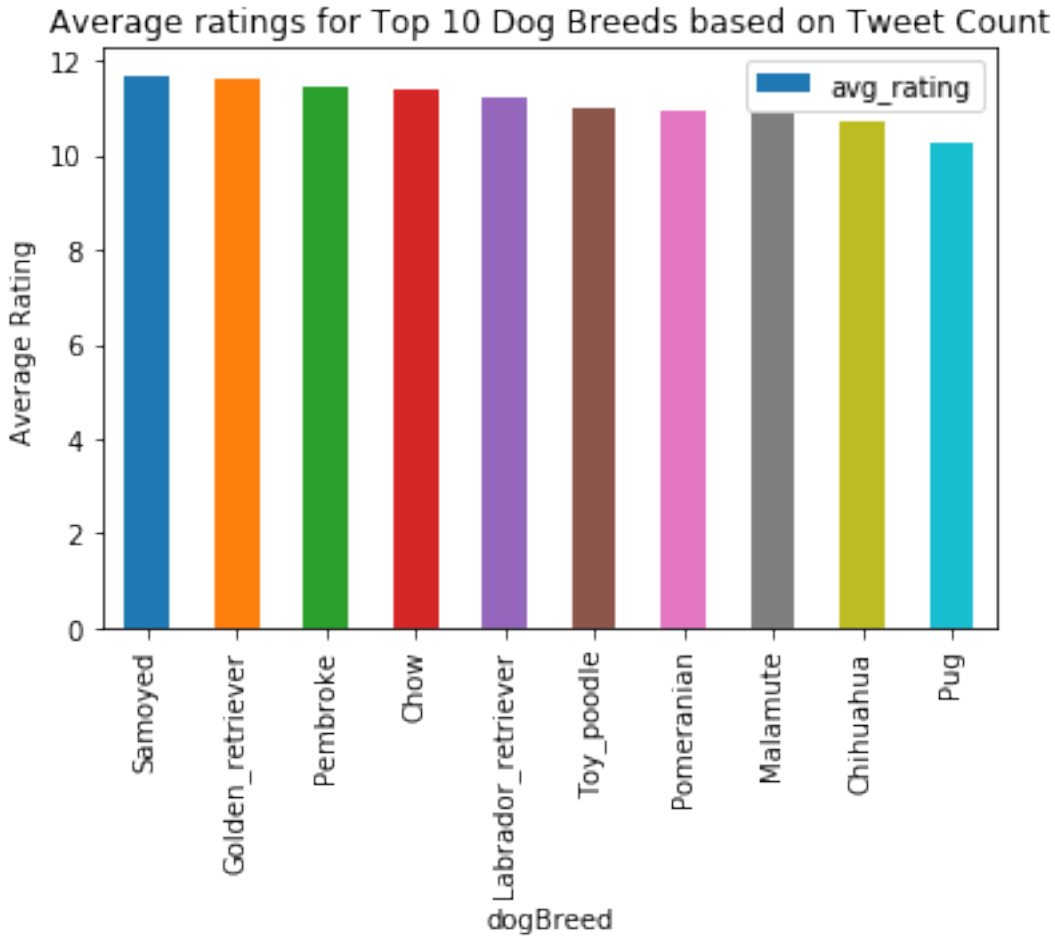
```
In [153]: # Make a list of top popular dog breeds based on number of tweets
          top_tweet_count = df.dog_breed.value_counts().sort_values(ascending=False).nlargest(10
          dog_breed_list = top_tweet_count.dog_breed.tolist()

          # Average rating for top breeds based on number of tweets
          avg_rating = df.groupby('dog_breed').rating_numerator.mean().sort_values(ascending=Fal
          dog_breed_avg_rating = avg_rating[avg_rating['dog_breed'].isin(dog_breed_list)]

          dog_breed_avg_rating
```

Out[153]:

	dog_breed	avg_rating
10	Samoyed	11.690476
11	Golden_retriever	11.612179
15	Pembroke	11.425532
16	Chow	11.416667
28	Labrador_retriever	11.198113
37	Toy_poodle	11.000000
47	Pomeranian	10.945122

```
51  Malamute            10.878788
61  Chihuahua           10.693182
78  Pug                 10.241935
```

```
In [154]: # Plot chart
          dog_breed_avg_rating.plot(kind='bar', x='dog_breed', y='avg_rating')
          plt.title('Average ratings for Top 10 Dog Breeds based on Tweet Count')
          plt.ylabel("Average Rating")
          plt.xlabel("dogBreed");
```



3.7.2 It is clear that popular dogs are rated similarly to one another.

3.8 Citation:

```
In [155]: df.source.value_counts()
```

```
Out[155]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
          <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
```

```
<a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
Name: source, dtype: int64
```

```
In [156]: df.source.value_counts(normalize=True)
```

```
Out[156]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
          <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
          <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
          Name: source, dtype: float64
```

3.9 We can see that the vast majority of users (98%) use iPhones.

```
In [157]: from subprocess import call
          call(['python', '-m', 'nbconvert', 'wrangle_act.ipynb'])
```

```
Out[157]: 0
```

```
In [ ]:
```

```
In [ ]:
```