wrangle_act

October 1, 2023

1 Project: Wrangling and Analyze Data

1.1 Data Gathering

1. Directly download the WeRateDogs Twitter archive data (twitter_archive_enhanced.csv)

```
In [3]: import numpy as np
        import pandas as pd
        import requests
        import tweepy
        import json
        import matplotlib.pyplot as plt
```

```
In [4]: df1=pd.read_csv('twitter-archive-enhanced.csv')
```

```
In [5]: df1.head(2).T
```

```
Out[5]:                                                                  0  \
        tweet_id                                        892420643555336193
        in_reply_to_status_id                                          NaN
        in_reply_to_user_id                                            NaN
        timestamp                                2017-08-01 16:23:56 +0000
        source                       <a href="http://twitter.com/download/iphone" r...
        text                         This is Phineas. He's a mystical boy. Only eve...
        retweeted_status_id                                            NaN
        retweeted_status_user_id                                       NaN
        retweeted_status_timestamp                                     NaN
        expanded_urls                https://twitter.com/dog_rates/status/892420643...
        rating_numerator                                                13
        rating_denominator                                              10
        name                                                       Phineas
        doggo                                                         None
        floofer                                                       None
        pupper                                                        None
        puppo                                                         None

                                                                         1
        tweet_id                                        892177421306343426
```

```
in_reply_to_status_id                                                       NaN
in_reply_to_user_id                                                         NaN
timestamp                                             2017-08-01 00:17:27 +0000
source                        <a href="http://twitter.com/download/iphone" r...
text                          This is Tilly. She's just checking pup on you...
retweeted_status_id                                                         NaN
retweeted_status_user_id                                                    NaN
retweeted_status_timestamp                                                  NaN
expanded_urls                 https://twitter.com/dog_rates/status/892177421...
rating_numerator                                                             13
rating_denominator                                                           10
name                                                                      Tilly
doggo                                                                      None
floofer                                                                    None
pupper                                                                     None
puppo                                                                      None
```

In [6]:
```
df1.shape
```

Out[6]:
```
(2356, 17)
```

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

In [7]:
```
import os
import requests

file_name = "image_predictions.tsv"

# Fetch data if the file doesn't exist
if not os.path.isfile(file_name):
    url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predi
    try:
        response = requests.get(url)
        response.raise_for_status()  # Raise an exception if the request was not success
        with open(file_name, "wb") as file:
            file.write(response.content)
        print("Download successful. Data saved as image_predictions.tsv.")
    except requests.exceptions.RequestException as e:
        print(f"Error occurred during download: {e}")
else:
    print("The file already exists. No need to download.")
```

```
The file already exists. No need to download.
```

In [8]:
```
df_pred = pd.read_csv(file_name, sep='\t')
```

In [9]:
```
df_pred.head(2).T
```

```
Out[9]:                                                     0  \
        tweet_id                           666020888022790149
        jpg_url   https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
        img_num                                             1
        p1                             Welsh_springer_spaniel
        p1_conf                                      0.465074
        p1_dog                                           True
        p2                                             collie
        p2_conf                                      0.156665
        p2_dog                                           True
        p3                                  Shetland_sheepdog
        p3_conf                                     0.0614285
        p3_dog                                           True

                                                            1
        tweet_id                           666029285002620928
        jpg_url   https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
        img_num                                             1
        p1                                            redbone
        p1_conf                                      0.506826
        p1_dog                                           True
        p2                                 miniature_pinscher
        p2_conf                                     0.0741917
        p2_dog                                           True
        p3                                Rhodesian_ridgeback
        p3_conf                                       0.07201
        p3_dog                                           True
```

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

```
In [10]: # I replaced the confidential data with generic strings
         consumer_key = 'consumer_key'
         consumer_secret = 'secret_consumer'
         access_token = 'access_token'
         access_secret = 'secret_access'
         auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
         auth.set_access_token(access_token, access_secret)
         api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)
```

```
In [11]: df_list = []
         with open('tweet_json.txt', mode='r', encoding='utf-8') as file:
             lines = file.readlines()
             for line in lines:
                 data = json.loads(line)
                 tweet_id = data['id']
                 retweet_count = data['retweet_count']
                 favorite_count = data['favorite_count']
                 df_list.append({'tweet_id': tweet_id, 'retweet_count': retweet_count, 'favorite
         tweet_data = pd.DataFrame(df_list, columns=['tweet_id', 'retweet_count', 'favorite_coun
```

1.2 Assessing Data

In this section, detect and document at least **eight (8) quality issues and two (2) tidiness issue**. You must use **both** visual assessment programmatic assessement to assess the data.

Note: pay attention to the following key points when you access the data.

- You only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets.
- Assessing and cleaning the entire dataset completely would require a lot of time, and is not necessary to practice and demonstrate your skills in data wrangling. Therefore, the requirements of this project are only to assess and clean at least 8 quality issues and at least 2 tidiness issues in this dataset.
- The fact that the rating numerators are greater than the denominators does not need to be cleaned. This unique rating system is a big part of the popularity of WeRateDogs.
- You do not need to gather the tweets beyond August 1st, 2017. You can, but note that you won't be able to gather the image predictions for these tweets since you don't have access to the algorithm used.

```
In [12]: df1
```

```
Out[12]:              tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         0     892420643555336193                    NaN                  NaN
         1     892177421306343426                    NaN                  NaN
         2     891815181378084864                    NaN                  NaN
         3     891689557279858688                    NaN                  NaN
         4     891327558926688256                    NaN                  NaN
         5     891087950875897856                    NaN                  NaN
         6     890971913173991426                    NaN                  NaN
         7     890729181411237888                    NaN                  NaN
         8     890609185150312448                    NaN                  NaN
         9     890240255349198849                    NaN                  NaN
         10    890006608113172480                    NaN                  NaN
         11    889880896479866881                    NaN                  NaN
         12    889665388333682689                    NaN                  NaN
         13    889638837579907072                    NaN                  NaN
         14    889531135344209921                    NaN                  NaN
         15    889278841981685760                    NaN                  NaN
         16    888917238123831296                    NaN                  NaN
         17    888804989199671297                    NaN                  NaN
         18    888554962724278272                    NaN                  NaN
         19    888202515573088257                    NaN                  NaN
         20    888078434458587136                    NaN                  NaN
         21    887705289381826560                    NaN                  NaN
         22    887517139158093824                    NaN                  NaN
         23    887473957103951883                    NaN                  NaN
         24    887343217045368832                    NaN                  NaN
         25    887101392804085760                    NaN                  NaN
         26    886983233522544640                    NaN                  NaN
         27    886736880519319552                    NaN                  NaN
```

```
28    886680336477933568                   NaN                  NaN
29    886366144734445568                   NaN                  NaN
...                  ...                   ...                  ...
2326  666411507551481857                   NaN                  NaN
2327  666407126856765440                   NaN                  NaN
2328  666396247373291520                   NaN                  NaN
2329  666373753744588802                   NaN                  NaN
2330  666362758909284353                   NaN                  NaN
2331  666353288456101888                   NaN                  NaN
2332  666345417576210432                   NaN                  NaN
2333  666337882303524864                   NaN                  NaN
2334  666293911632134144                   NaN                  NaN
2335  666287406224695296                   NaN                  NaN
2336  666273097616637952                   NaN                  NaN
2337  666268910803644416                   NaN                  NaN
2338  666104133288665088                   NaN                  NaN
2339  666102155909144576                   NaN                  NaN
2340  666099513787052032                   NaN                  NaN
2341  666094000022159362                   NaN                  NaN
2342  666082916733198337                   NaN                  NaN
2343  666073100786774016                   NaN                  NaN
2344  666071193221509120                   NaN                  NaN
2345  666063827256086533                   NaN                  NaN
2346  666058600524156928                   NaN                  NaN
2347  666057090499244032                   NaN                  NaN
2348  666055525042405380                   NaN                  NaN
2349  666051853826850816                   NaN                  NaN
2350  666050758794694657                   NaN                  NaN
2351  666049248165822465                   NaN                  NaN
2352  666044226329800704                   NaN                  NaN
2353  666033412701032449                   NaN                  NaN
2354  666029285002620928                   NaN                  NaN
2355  666020888022790149                   NaN                  NaN

                    timestamp  \
0     2017-08-01 16:23:56 +0000
1     2017-08-01 00:17:27 +0000
2     2017-07-31 00:18:03 +0000
3     2017-07-30 15:58:51 +0000
4     2017-07-29 16:00:24 +0000
5     2017-07-29 00:08:17 +0000
6     2017-07-28 16:27:12 +0000
7     2017-07-28 00:22:40 +0000
8     2017-07-27 16:25:51 +0000
9     2017-07-26 15:59:51 +0000
10    2017-07-26 00:31:25 +0000
11    2017-07-25 16:11:53 +0000
12    2017-07-25 01:55:32 +0000
```

```
13    2017-07-25 00:10:02 +0000
14    2017-07-24 17:02:04 +0000
15    2017-07-24 00:19:32 +0000
16    2017-07-23 00:22:39 +0000
17    2017-07-22 16:56:37 +0000
18    2017-07-22 00:23:06 +0000
19    2017-07-21 01:02:36 +0000
20    2017-07-20 16:49:33 +0000
21    2017-07-19 16:06:48 +0000
22    2017-07-19 03:39:09 +0000
23    2017-07-19 00:47:34 +0000
24    2017-07-18 16:08:03 +0000
25    2017-07-18 00:07:08 +0000
26    2017-07-17 16:17:36 +0000
27    2017-07-16 23:58:41 +0000
28    2017-07-16 20:14:00 +0000
29    2017-07-15 23:25:31 +0000
...                         ...
2326  2015-11-17 00:24:19 +0000
2327  2015-11-17 00:06:54 +0000
2328  2015-11-16 23:23:41 +0000
2329  2015-11-16 21:54:18 +0000
2330  2015-11-16 21:10:36 +0000
2331  2015-11-16 20:32:58 +0000
2332  2015-11-16 20:01:42 +0000
2333  2015-11-16 19:31:45 +0000
2334  2015-11-16 16:37:02 +0000
2335  2015-11-16 16:11:11 +0000
2336  2015-11-16 15:14:19 +0000
2337  2015-11-16 14:57:41 +0000
2338  2015-11-16 04:02:55 +0000
2339  2015-11-16 03:55:04 +0000
2340  2015-11-16 03:44:34 +0000
2341  2015-11-16 03:22:39 +0000
2342  2015-11-16 02:38:37 +0000
2343  2015-11-16 01:59:36 +0000
2344  2015-11-16 01:52:02 +0000
2345  2015-11-16 01:22:45 +0000
2346  2015-11-16 01:01:59 +0000
2347  2015-11-16 00:55:59 +0000
2348  2015-11-16 00:49:46 +0000
2349  2015-11-16 00:35:11 +0000
2350  2015-11-16 00:30:50 +0000
2351  2015-11-16 00:24:50 +0000
2352  2015-11-16 00:04:52 +0000
2353  2015-11-15 23:21:54 +0000
2354  2015-11-15 23:05:30 +0000
2355  2015-11-15 22:32:08 +0000
```

```
                                                    source  \
0     <a href="http://twitter.com/download/iphone" r...
1     <a href="http://twitter.com/download/iphone" r...
2     <a href="http://twitter.com/download/iphone" r...
3     <a href="http://twitter.com/download/iphone" r...
4     <a href="http://twitter.com/download/iphone" r...
5     <a href="http://twitter.com/download/iphone" r...
6     <a href="http://twitter.com/download/iphone" r...
7     <a href="http://twitter.com/download/iphone" r...
8     <a href="http://twitter.com/download/iphone" r...
9     <a href="http://twitter.com/download/iphone" r...
10    <a href="http://twitter.com/download/iphone" r...
11    <a href="http://twitter.com/download/iphone" r...
12    <a href="http://twitter.com/download/iphone" r...
13    <a href="http://twitter.com/download/iphone" r...
14    <a href="http://twitter.com/download/iphone" r...
15    <a href="http://twitter.com/download/iphone" r...
16    <a href="http://twitter.com/download/iphone" r...
17    <a href="http://twitter.com/download/iphone" r...
18    <a href="http://twitter.com/download/iphone" r...
19    <a href="http://twitter.com/download/iphone" r...
20    <a href="http://twitter.com/download/iphone" r...
21    <a href="http://twitter.com/download/iphone" r...
22    <a href="http://twitter.com/download/iphone" r...
23    <a href="http://twitter.com/download/iphone" r...
24    <a href="http://twitter.com/download/iphone" r...
25    <a href="http://twitter.com/download/iphone" r...
26    <a href="http://twitter.com/download/iphone" r...
27    <a href="http://twitter.com/download/iphone" r...
28    <a href="http://twitter.com/download/iphone" r...
29    <a href="http://twitter.com/download/iphone" r...
...                                                 ...
2326  <a href="http://twitter.com/download/iphone" r...
2327  <a href="http://twitter.com/download/iphone" r...
2328  <a href="http://twitter.com/download/iphone" r...
2329  <a href="http://twitter.com/download/iphone" r...
2330  <a href="http://twitter.com/download/iphone" r...
2331  <a href="http://twitter.com/download/iphone" r...
2332  <a href="http://twitter.com/download/iphone" r...
2333  <a href="http://twitter.com/download/iphone" r...
2334  <a href="http://twitter.com/download/iphone" r...
2335  <a href="http://twitter.com/download/iphone" r...
2336  <a href="http://twitter.com/download/iphone" r...
2337  <a href="http://twitter.com/download/iphone" r...
2338  <a href="http://twitter.com/download/iphone" r...
2339  <a href="http://twitter.com/download/iphone" r...
2340  <a href="http://twitter.com/download/iphone" r...
```

```
2341  <a href="http://twitter.com/download/iphone" r...
2342  <a href="http://twitter.com/download/iphone" r...
2343  <a href="http://twitter.com/download/iphone" r...
2344  <a href="http://twitter.com/download/iphone" r...
2345  <a href="http://twitter.com/download/iphone" r...
2346  <a href="http://twitter.com/download/iphone" r...
2347  <a href="http://twitter.com/download/iphone" r...
2348  <a href="http://twitter.com/download/iphone" r...
2349  <a href="http://twitter.com/download/iphone" r...
2350  <a href="http://twitter.com/download/iphone" r...
2351  <a href="http://twitter.com/download/iphone" r...
2352  <a href="http://twitter.com/download/iphone" r...
2353  <a href="http://twitter.com/download/iphone" r...
2354  <a href="http://twitter.com/download/iphone" r...
2355  <a href="http://twitter.com/download/iphone" r...

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

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

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

```
16                  NaN                        NaN
17                  NaN                        NaN
18                  NaN                        NaN
19         4.196984e+09  2017-07-19 00:47:34 +0000
20                  NaN                        NaN
21                  NaN                        NaN
22                  NaN                        NaN
23                  NaN                        NaN
24                  NaN                        NaN
25                  NaN                        NaN
26                  NaN                        NaN
27                  NaN                        NaN
28                  NaN                        NaN
29                  NaN                        NaN
...                 ...                        ...
2326                NaN                        NaN
2327                NaN                        NaN
2328                NaN                        NaN
2329                NaN                        NaN
2330                NaN                        NaN
2331                NaN                        NaN
2332                NaN                        NaN
2333                NaN                        NaN
2334                NaN                        NaN
2335                NaN                        NaN
2336                NaN                        NaN
2337                NaN                        NaN
2338                NaN                        NaN
2339                NaN                        NaN
2340                NaN                        NaN
2341                NaN                        NaN
2342                NaN                        NaN
2343                NaN                        NaN
2344                NaN                        NaN
2345                NaN                        NaN
2346                NaN                        NaN
2347                NaN                        NaN
2348                NaN                        NaN
2349                NaN                        NaN
2350                NaN                        NaN
2351                NaN                        NaN
2352                NaN                        NaN
2353                NaN                        NaN
2354                NaN                        NaN
2355                NaN                        NaN

                                         expanded_urls  rating_numerator  \
0     https://twitter.com/dog_rates/status/892420643...                13
```

1	https://twitter.com/dog_rates/status/892177421...	13
2	https://twitter.com/dog_rates/status/891815181...	12
3	https://twitter.com/dog_rates/status/891689557...	13
4	https://twitter.com/dog_rates/status/891327558...	12
5	https://twitter.com/dog_rates/status/891087950...	13
6	https://gofundme.com/ydvmve-surgery-for-jax,ht...	13
7	https://twitter.com/dog_rates/status/890729181...	13
8	https://twitter.com/dog_rates/status/890609185...	13
9	https://twitter.com/dog_rates/status/890240255...	14
10	https://twitter.com/dog_rates/status/890006608...	13
11	https://twitter.com/dog_rates/status/889880896...	13
12	https://twitter.com/dog_rates/status/889665388...	13
13	https://twitter.com/dog_rates/status/889638837...	12
14	https://twitter.com/dog_rates/status/889531135...	13
15	https://twitter.com/dog_rates/status/889278841...	13
16	https://twitter.com/dog_rates/status/888917238...	12
17	https://twitter.com/dog_rates/status/888804989...	13
18	https://twitter.com/dog_rates/status/888554962...	13
19	https://twitter.com/dog_rates/status/887473957...	13
20	https://twitter.com/dog_rates/status/888078434...	12
21	https://twitter.com/dog_rates/status/887705289...	13
22	https://twitter.com/dog_rates/status/887517139...	14
23	https://twitter.com/dog_rates/status/887473957...	13
24	https://twitter.com/dog_rates/status/887343217...	13
25	https://twitter.com/dog_rates/status/887101392...	12
26	https://twitter.com/dog_rates/status/886983233...	13
27	https://www.gofundme.com/mingusneedsus,https:/...	13
28	https://twitter.com/dog_rates/status/886680336...	13
29	https://twitter.com/dog_rates/status/886366144...	12
...	...	...
2326	https://twitter.com/dog_rates/status/666411507...	2
2327	https://twitter.com/dog_rates/status/666407126...	7
2328	https://twitter.com/dog_rates/status/666396247...	9
2329	https://twitter.com/dog_rates/status/666373753...	11
2330	https://twitter.com/dog_rates/status/666362758...	6
2331	https://twitter.com/dog_rates/status/666353288...	8
2332	https://twitter.com/dog_rates/status/666345417...	10
2333	https://twitter.com/dog_rates/status/666337882...	9
2334	https://twitter.com/dog_rates/status/666293911...	3
2335	https://twitter.com/dog_rates/status/666287406...	1
2336	https://twitter.com/dog_rates/status/666273097...	11
2337	https://twitter.com/dog_rates/status/666268910...	10
2338	https://twitter.com/dog_rates/status/666104133...	1
2339	https://twitter.com/dog_rates/status/666102155...	11
2340	https://twitter.com/dog_rates/status/666099513...	8
2341	https://twitter.com/dog_rates/status/666094000...	9
2342	https://twitter.com/dog_rates/status/666082916...	6
2343	https://twitter.com/dog_rates/status/666073100...	10

2344	https://twitter.com/dog_rates/status/666071193...	9
2345	https://twitter.com/dog_rates/status/666063827...	10
2346	https://twitter.com/dog_rates/status/666058600...	8
2347	https://twitter.com/dog_rates/status/666057090...	9
2348	https://twitter.com/dog_rates/status/666055525...	10
2349	https://twitter.com/dog_rates/status/666051853...	2
2350	https://twitter.com/dog_rates/status/666050758...	10
2351	https://twitter.com/dog_rates/status/666049248...	5
2352	https://twitter.com/dog_rates/status/666044226...	6
2353	https://twitter.com/dog_rates/status/666033412...	9
2354	https://twitter.com/dog_rates/status/666029285...	7
2355	https://twitter.com/dog_rates/status/666020888...	8

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

```
2329                 10     None  None   None   None  None
2330                 10     None  None   None   None  None
2331                 10     None  None   None   None  None
2332                 10     None  None   None   None  None
2333                 10       an  None   None   None  None
2334                 10        a  None   None   None  None
2335                  2       an  None   None   None  None
2336                 10     None  None   None   None  None
2337                 10     None  None   None   None  None
2338                 10     None  None   None   None  None
2339                 10     None  None   None   None  None
2340                 10     None  None   None   None  None
2341                 10     None  None   None   None  None
2342                 10     None  None   None   None  None
2343                 10     None  None   None   None  None
2344                 10     None  None   None   None  None
2345                 10      the  None   None   None  None
2346                 10      the  None   None   None  None
2347                 10        a  None   None   None  None
2348                 10        a  None   None   None  None
2349                 10       an  None   None   None  None
2350                 10        a  None   None   None  None
2351                 10     None  None   None   None  None
2352                 10        a  None   None   None  None
2353                 10        a  None   None   None  None
2354                 10        a  None   None   None  None
2355                 10     None  None   None   None  None

[2356 rows x 17 columns]
```

In [13]:
```
import pandas as pd

predictions = pd.read_csv("image_predictions.tsv", sep="\t")

print(predictions.head())
```

```
            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
```

In [14]: tweet_data

Out[14]:

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

```
2328  666362758909284353     595      804
2329  666353288456101888      77      229
2330  666345417576210432     146      307
2331  666337882303524864      96      204
2332  666293911632134144     368      522
2333  666287406224695296      71      152
2334  666273097616637952      82      184
2335  666268910803644416      37      108
2336  666104133288665088    6871    14765
2337  666102155909144576      16       81
2338  666099513787052032      73      164
2339  666094000022159362      79      169
2340  666082916733198337      47      121
2341  666073100786774016     174      335
2342  666071193221509120      67      154
2343  666063827256086533     232      496
2344  666058600524156928      61      115
2345  666057090499244032     146      304
2346  666055525042405380     261      448
2347  666051853826850816     879     1253
2348  666050758794694657      60      136
2349  666049248165822465      41      111
2350  666044226329800704     147      311
2351  666033412701032449      47      128
2352  666029285002620928      48      132
2353  666020888022790149     532     2535

[2354 rows x 3 columns]
```

After visually assessing the dataframes, the following observations were made:
archive:

The dataframe is not tidy, particularly concerning the dog stages. Instead of having separate columns for each stage (e.g., doggo, floofer, pupper, puppo), there should be one column to specify the stage for each dog. Some names in the 'name' column appear to be incorrect. For example, the name "a" is unlikely to be a dog's actual name. predictions:

The dataframe is also not tidy due to the spread of prediction data over several columns (e.g., prediction 1, prediction 2, prediction 3). It would be better to have one column for prediction number, and additional columns for the actual prediction, confidence, and whether it is a type of dog breed. The prediction numbers in columns 'p1', 'p2', and 'p3' have inconsistent capitalization. tweet_data:

The data in this dataframe is collected via the Twitter API, and some tweets may have been deleted, resulting in missing retweet or favorite counts. Unfortunately, this missing data cannot be retrieved. To achieve tidy data, each observational unit should be in its own dataframe. The three observational units can be defined as tweet data, dog data, and image predictions.

The next step is to assess each dataframe programmatically, starting with 'archive', to check for duplicated rows or duplicate tweet IDs.

```
In [15]: sum(df1.duplicated())
```

Out[15]: 0

In [16]: sum(df1.duplicated('tweet_id'))

Out[16]: 0

In [17]: df1.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
```

After visual assessment, I've identified a couple of data type modifications that should be made. The 'timestamp' column needs to be converted to the datetime data type, and the 'dog stage' should be changed to a categorical data type.

Furthermore, the 'archive' dataframe contains retweets, which should not be included as I am interested only in original content from WeRateDogs. To verify this, I will check if the values in the 'retweeted_status_id', 'retweeted_status_user_id', and 'retweeted_status_timestamp' columns are all present in the same rows.

The goal is to ensure data consistency and prepare the dataframe for further analysis and cleaning.

To validate data consistency, I will check the number of rows where the columns 'retweeted_status_id', 'retweeted_status_user_id', and 'retweeted_status_timestamp' are filled. If all these columns have non-null values in the same rows, the resulting count should be 181, which corresponds to the number of non-null values for each of the columns.

```
In [18]: len(df1[
             df1['retweeted_status_id'].notnull()
             & df1['retweeted_status_user_id'].notnull()
```

```
    & df1['retweeted_status_timestamp'].notnull()
])
```

Out[18]: 181

Noted that all these values are in the same rows, which is an important observation to consider.

Regarding the 'expanded_urls' column and the missing values, I will print a sample of this column below to gain a better understanding. This will help me investigate why some values are missing in this column.

In [19]:
```
df1['expanded_urls'].sample(5)
```

Out[19]:
```
326     https://twitter.com/dog_rates/status/833826103...
1104    https://twitter.com/dog_rates/status/735137028...
717     https://twitter.com/dog_rates/status/783695101...
1723    https://twitter.com/dog_rates/status/680100725...
95      https://twitter.com/dog_rates/status/868880397...
Name: expanded_urls, dtype: object
```

Indeed, the 'expanded_urls' column seems to contain links to associated content like photos, videos, or other media. It is expected that some tweets may not have links, resulting in missing values for now. However, in the final dataset, missing values in 'expanded_urls' should be addressed, as we are interested only in tweets with images.

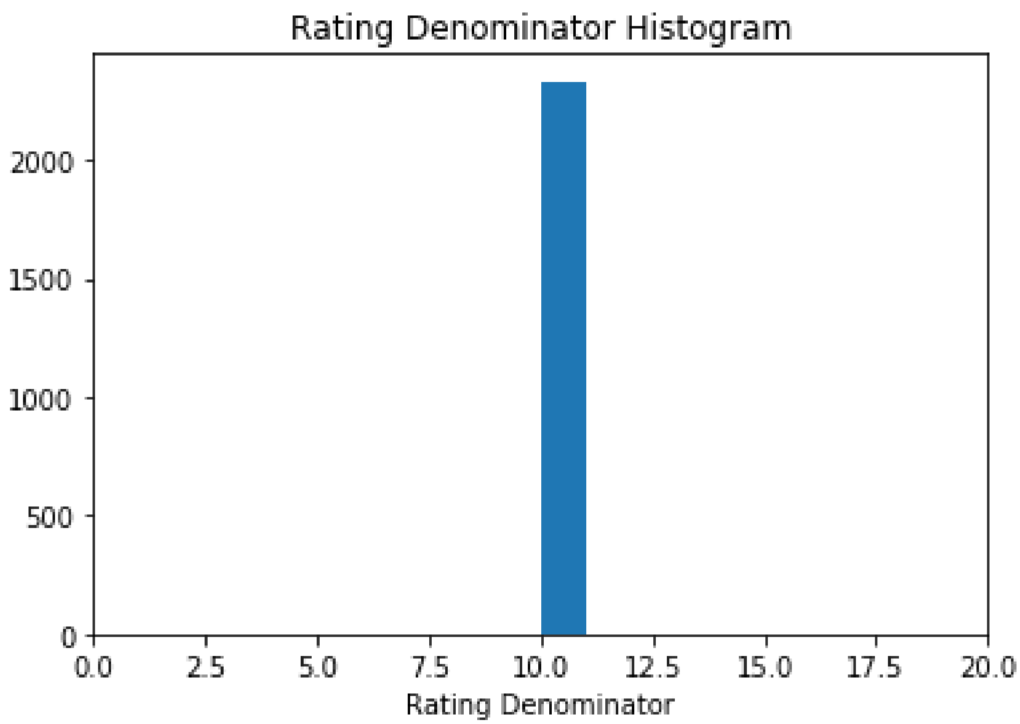
Next, I will examine the descriptive statistics for the 'archive' dataframe to gain insights into the data's distribution and better understand the characteristics of the dataset. This will help in identifying any irregularities or outliers that may require further attention during the data cleaning process.

In [20]:
```
df1.describe()
```

Out[20]:
```
           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 [21]:
```
bins = np.arange(0, df1['rating_denominator'].max()+1, 1)
plt.hist(data=df1, x='rating_denominator', bins=bins)
plt.title('Rating Denominator Histogram')
plt.xlabel('Rating Denominator')
plt.xlim(0, 20);
```

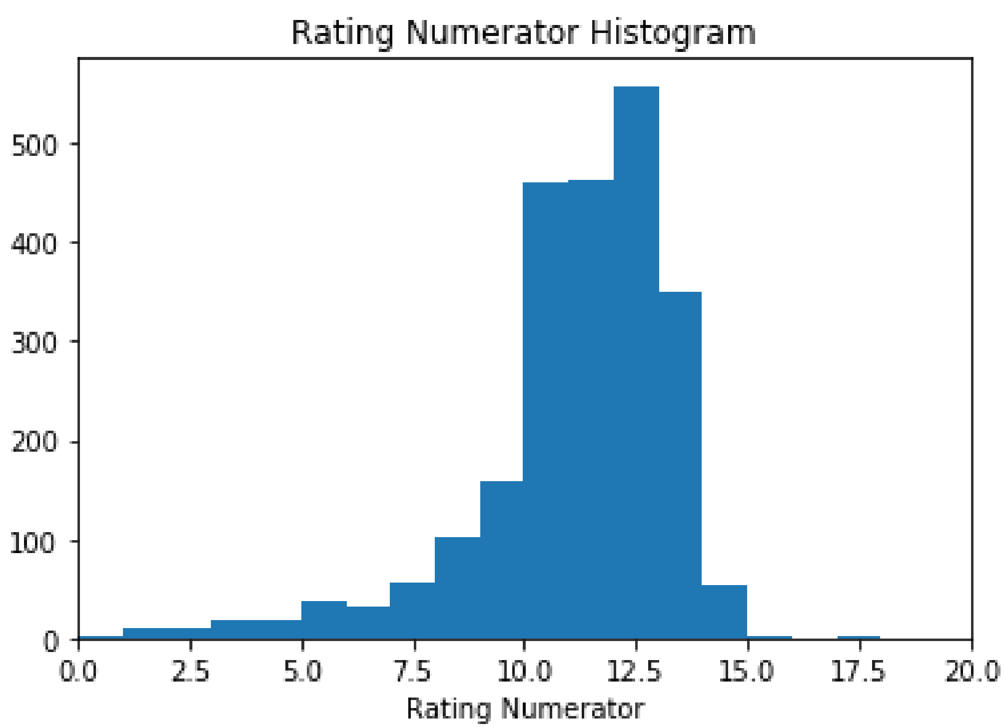


In [38]:
```
df1.query('rating_denominator != 10')['text']
```

Out[38]:
```
313     @jonnysun @Lin_Manuel ok jomny I know you're e...
342              @docmisterio account started on 11/15/15
433     The floofs have been released I repeat the flo...
516     Meet Sam. She smiles 24/7 &amp; secretly aspir...
784     RT @dog_rates: After so many requests, this is...
```

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

In [22]:
```python
bins = np.arange(0, df1['rating_numerator'].max()+1, 1)
plt.hist(data=df1, x='rating_numerator', bins=bins)
plt.title('Rating Numerator Histogram')
plt.xlabel('Rating Numerator')
plt.xlim(0, 20);
```



```
In [23]: df1.query('rating_numerator >= 15')['text']
```

```
Out[23]: 55      @roushfenway These are good dogs but 17/10 is ...
         188     @dhmontgomery We also gave snoop dogg a 420/10...
         189     @s8n You tried very hard to portray this good ...
         285     RT @KibaDva: I collected all the good dogs!! 1...
         290                                @markhoppus 182/10
         291     @bragg6of8 @Andy_Pace_ we are still looking fo...
         313     @jonnysun @Lin_Manuel ok jomny I know you're e...
         340     RT @dog_rates: This is Logan, the Chow who liv...
         433     The floofs have been released I repeat the flo...
         516     Meet Sam. She smiles 24/7 &amp; secretly aspir...
         695     This is Logan, the Chow who lived. He solemnly...
         763     This is Sophie. She's a Jubilant Bush Pupper. ...
         902     Why does this never happen at my front door...
         979     This is Atticus. He's quite simply America af...
         1120    Say hello to this unbelievably well behaved sq...
         1202    This is Bluebert. He just saw that both #Final...
         1228    Happy Saturday here's 9 puppers on a bench. 99...
         1254    Here's a brigade of puppers. All look very pre...
         1274    From left to right:\nCletus, Jerome, Alejandro...
         1351    Here is a whole flock of puppers.  60/50 I'll ...
         1433    Happy Wednesday here's a bucket of pups. 44/40...
         1634    Two sneaky puppers were not initially seen, mo...
         1635    Someone help the girl is being mugged. Several...
         1663    I'm aware that I could've said 20/16, but here...
         1712    Here we have uncovered an entire battalion of ...
         1779    IT'S PUPPERGEDDON. Total of 144/120 ...I think...
         1843    Here we have an entire platoon of puppers. Tot...
         2074    After so many requests... here you go.\n\nGood...
         Name: text, dtype: object
```

```
In [24]:  with pd.option_context('display.max_rows', None):
              print(df1.query('rating_numerator < 5')['text'])
```

```
315     When you're so blinded by your systematic plag...
605     RT @dog_rates: Not familiar with this breed. N...
765     This is Wesley. He's clearly trespassing. Seem...
883     This is Fido. He can tell the weather. Not goo...
912     Here's another picture without a dog in it. Id...
1004    Viewer discretion is advised. This is a terrib...
1016    PUPDATE: can't see any. Even if I could, I cou...
1165    Happy 4/20 from the squad! 13/10 for all https...
1189    This is Alexanderson. He's got a weird ass bir...
1219    This is Benedict. He's a feisty pup. Needs a b...
1249    What hooligan sent in pictures w/out a dog in ...
1303    This is Keurig. He's a rare dog. Laughs like a...
```

```
1314    This is Elliot. He's blocking the roadway. Dow...
1406    This is Charl. He's a bully. Chucks that dumbb...
1446    After reading the comments I may have overesti...
1459    This may be the greatest video I've ever been ...
1478    Meet Phil. He's big af. Currently destroying t...
1598    Yes I do realize a rating of 4/20 would've bee...
1601    This is Hammond. He's a peculiar pup. Loves lo...
1629    This is Bobby. He doesn't give a damn about pe...
1692    This is Chuck. He's a neat dog. Very flexible...
1701    This is Alice. She's an idiot. 4/10 https://t...
1761    Exotic pup here. Tail long af. Throat looks sw...
1764    This is Crystal. She's a shitty fireman. No se...
1836    Extremely rare pup here. Very religious. Alway...
1869    What kind of person sends in a picture without...
1898    Meet Patrick. He's an exotic pup. Jumps great ...
1920    This is Henry. He's a shit dog. Short pointy e...
1928    Herd of wild dogs here. Not sure what they're ...
1938    Guys I'm getting real tired of this. We only r...
1940    The millennials have spoken and we've decided ...
1941    This is a heavily opinionated dog. Loves walls...
1947    Large blue dog here. Cool shades. Flipping us ...
2038    After 22 minutes of careful deliberation this ...
2070    Two miniature golden retrievers here. Webbed p...
2076    Pink dogs here. Unreasonably long necks. Left ...
2079    Scary dog here. Too many legs. Extra tail. Not...
2091    Flamboyant pup here. Probably poisonous. Won't...
2136    This is Tommy. He's a cool dog. Hard not to st...
2183    This is Bernie. He's taking his Halloween cost...
2186    Unique dog here. Oddly shaped tail. Long pink ...
2202    Fascinating dog here. Loves beach. Oddly long ...
2222    Here is a mother dog caring for her pups. Snaz...
2237    This lil pup is Oliver. Hops around. Has wings...
2239    This dog resembles a baked potato. Bed looks u...
2246    This is Tedrick. He lives on the edge. Needs s...
2261    Never seen dog like this. Breathes heavy. Tilt...
2288    These are strange dogs. All have toupees. Long...
2305    My goodness. Very rare dog here. Large. Tail d...
2310    Unfamiliar with this breed. Ears pointy af. Wo...
2316    Cool dog. Enjoys couch. Low monotone bark. Ver...
2326    This is quite the dog. Gets really excited whe...
2334    This is a funny dog. Weird toes. Won't come do...
2335    This is an Albanian 3 1/2 legged  Episcopalian...
2338    Not familiar with this breed. No tail (weird)...
2349    This is an odd dog. Hard on the outside but lo...
Name: text, dtype: object
```

```
In [42]: df1['name'].value_counts()
```

```
Out[42]: None          745
         a              55
         Charlie        12
         Lucy           11
         Cooper         11
         Oliver         11
         Penny          10
         Lola           10
         Tucker         10
         Bo              9
         Winston         9
         Sadie           8
         the             8
         Bailey          7
         an              7
         Toby            7
         Daisy           7
         Buddy           7
         Jack            6
         Koda            6
         Stanley         6
         Scout           6
         Oscar           6
         Milo            6
         Bella           6
         Rusty           6
         Dave            6
         Jax             6
         Leo             6
         Louis           5
                      ...
         JD              1
         Jay             1
         Simba           1
         Kloey           1
         Anthony         1
         Daniel          1
         Aldrick         1
         River           1
         Rizzo           1
         Crawford        1
         Kara            1
         Ronduh          1
         Buckley         1
         Tyrus           1
         Chesterson      1
         Joshwa          1
         Milky           1
```

```
Shelby          1
Willie          1
Clarq           1
Mosby           1
Rilo            1
Zooey           1
Sid             1
Lilah           1
Suki            1
Odin            1
Saydee          1
Carbon          1
Alejandro       1
Name: name, Length: 957, dtype: int64
```

In [43]:
```
# use regex to find all lowercase names
df1[df1['name'].str.contains(r'^[^A-Z].*$')]['name'].value_counts()
```

Out[43]:
```
a               55
the              8
an               7
very             5
quite            4
just             4
one              4
actually         2
getting          2
not              2
mad              2
unacceptable     1
officially       1
infuriating      1
space            1
this             1
old              1
by               1
my               1
his              1
such             1
incredibly       1
light            1
life             1
all              1
Name: name, dtype: int64
```

In [44]:
```
with pd.option_context('display.max_rows', None):
    print(df1[df1['name'].str.contains(r'^[^A-Z].*$')]['text'])
```

```
22      I've yet to rate a Venezuelan Hover Wiener. Th...
56      Here is a pupper approaching maximum borkdrive...
```

```
118     RT @dog_rates: We only rate dogs. This is quit...
169     We only rate dogs. This is quite clearly a smo...
193     Guys, we only rate dogs. This is quite clearly...
335     There's going to be a dog terminal at JFK Airp...
369     Occasionally, we're sent fantastic stories. Th...
542     We only rate dogs. Please stop sending in non-...
649     Here is a perfect example of someone who has t...
682     RT @dog_rates: Say hello to mad pupper. You kn...
759     RT @dog_rates: This is an East African Chalupa...
773     RT @dog_rates: We only rate dogs. Pls stop sen...
801     Guys this is getting so out of hand. We only r...
819     We only rate dogs. Pls stop sending in non-can...
822     RT @dog_rates: This is just downright precious...
852     This is my dog. Her name is Zoey. She knows I'...
924     This is one of the most inspirational stories ...
988     What jokester sent in a pic without a dog in i...
992     That is Quizno. This is his beach. He does not...
993     This is one of the most reckless puppers I've ...
1002    This is a mighty rare blue-tailed hammer sherk...
1004    Viewer discretion is advised. This is a terrib...
1017    This is a carrot. We only rate dogs. Please on...
1025    This is an Iraqi Speed Kangaroo. It is not a d...
1031    We only rate dogs. Pls stop sending in non-can...
1040    This is actually a pupper and I'd pet it so we...
1049    This is a very rare Great Alaskan Bush Pupper...
1063    This is just downright precious af. 12/10 for ...
1071    This is getting incredibly frustrating. This i...
1095    Say hello to mad pupper. You know what you did...
1097    We only rate dogs. Please stop sending in non-...
1120    Say hello to this unbelievably well behaved sq...
1121    We only rate dogs. Pls stop sending non-canine...
1138    This is all I want in my life. 12/10 for super...
1193    People please. This is a Deadly Mediterranean ...
1206    This is old now but it's absolutely heckin fan...
1207    This is a taco. We only rate dogs. Please only...
1259    We  only  rate  dogs. Pls stop sending i...
1340    Here is a heartbreaking scene of an incredible...
1351    Here is a whole flock of puppers.  60/50 I'll ...
1361    This is a Butternut Cumberfloof. It's not wind...
1362    This is an East African Chalupa Seal. We only ...
1368    This is a Wild Tuscan Poofwiggle. Careful not ...
1382    "Pupper is a present to world. Here is a bow f...
1385    We only rate dogs. Pls stop sending in non-can...
1435    Please stop sending in saber-toothed tigers. T...
1457    This is just a beautiful pupper good shit evol...
1499    This is a rare Arctic Wubberfloof. Unamused by...
1527    Stop sending in lobsters. This is the final wa...
1603    This is the newly formed pupper a capella grou...
```

```
1693    This is actually a lion. We only rate dogs. Fo...
1724    This is by far the most coordinated series of ...
1737    Guys this really needs to stop. We've been ove...
1747    This is officially the greatest yawn of all ti...
1785    This is a dog swinging. I really enjoyed it so...
1797    This is the happiest pupper I've ever seen. 10...
1815    This is the saddest/sweetest/best picture I've...
1853    This is a Sizzlin Menorah spaniel from Brookly...
1854    Seriously guys?! Only send in dogs. I only rat...
1877    C'mon guys. We've been over this. We only rate...
1878    This is a fluffy albino Bacardi Columbia mix. ...
1916    This is life-changing. 12/10 https://t.co/SroT...
1923    This is a Sagitariot Baklava mix. Loves her ne...
1936    This is one esteemed pupper. Just graduated co...
1941    This is a heavily opinionated dog. Loves walls...
1955    This is a Lofted Aphrodisiac Terrier named Kip...
1994    This is a baby Rand Paul. Curls for days. 11/1...
2001    This is light saber pup. Ready to fight off ev...
2019    This is just impressive I have nothing else to...
2030    This is space pup. He's very confused. Tries t...
2034    This is a Tuscaloosa Alcatraz named Jacob (Yac...
2037    This is the best thing I've ever seen so sprea...
2066    This is a Helvetica Listerine named Rufus. Thi...
2116    This is a Deciduous Trimester mix named Spork...
2125    This is a Rich Mahogany Seltzer named Cherokee...
2128    This is a Speckled Cauliflower Yosemite named ...
2146    This is a spotted Lipitor Rumpelstiltskin name...
2153    This is a brave dog. Excellent free climber. T...
2161    This is a Coriander Baton Rouge named Alfredo...
2191    This is a Slovakian Helter Skelter Feta named ...
2198    This is a wild Toblerone from Papua New Guinea...
2204    This is an Irish Rigatoni terrier named Berta...
2211    Here is a horned dog. Much grace. Can jump ove...
2212    Never forget this vine. You will not stop watc...
2218    This is a Birmingham Quagmire named Chuk. Love...
2222    Here is a mother dog caring for her pups. Snaz...
2235    This is a Trans Siberian Kellogg named Alfonso...
2249    This is a Shotokon Macadamia mix named Cheryl...
2255    This is a rare Hungarian Pinot named Jessiga. ...
2264    This is a southwest Coriander named Klint. Hat...
2273    This is a northern Wahoo named Kohl. He runs t...
2287    This is a Dasani Kingfisher from Maine. His na...
2304    This is a curly Ticonderoga named Pepe. No fee...
2311    This is a purebred Bacardi named Octaviath. Ca...
2314    This is a golden Buckminsterfullerene named Jo...
2326    This is quite the dog. Gets really excited whe...
2327    This is a southern Vesuvius bumblegruff. Can d...
2333    This is an extremely rare horned Parthenon. No...
```

```
2334    This is a funny dog. Weird toes. Won't come do...
2335    This is an Albanian 3 1/2 legged  Episcopalian...
2345    This is the happiest dog you will ever see. Ve...
2346    Here is the Rand Paul of retrievers folks! He'...
2347    My oh my. This is a rare blond Canadian terrie...
2348    Here is a Siberian heavily armored polar bear ...
2349    This is an odd dog. Hard on the outside but lo...
2350    This is a truly beautiful English Wilson Staff...
2352    This is a purebred Piers Morgan. Loves to Netf...
2353    Here is a very happy pup. Big fan of well-main...
2354    This is a western brown Mitsubishi terrier. Up...
Name: text, dtype: object
```

I have observed two distinct trends in the data. First, a considerable number of tweets, as per WeRateDogs, do not feature pictures of dogs. In such cases, the account posts "We only rate dogs" and omits a specific name for the dog. This pattern is evident when the name column is null or contains the phrase "We only rate dogs."

The second trend is that numerous names are provided in the "name" column following the word "named." This pattern is evident when the name is in the format "named [name]."

To better understand the dog stages denoted in the dataset, I will examine the unique values in each of the stage columns. Additionally, I will perform manual extraction of the stages to verify if they align with the information presented in the dataframe. During the extraction process, I will be lenient with the matching criteria. Specifically, I will consider the stages as matching if they have:

The same letters (case-insensitive). Repeated letters, as long as the necessary letters appear in the correct order. An optional "s" at the end. By conducting this analysis, I aim to gain deeper insights into the dog stages present in the data and ensure consistency in how these stages are recorded.

```
In [25]: def compare_stage(stage, regex):
             '''
             INPUT:
             stage (str) - desired dog stage, can be 'doggo', 'floofer', 'pupper', or 'puppo'
             regex (str) - regex to use when re-extracting dog stage

             OUTPUT:
             None

             Print value counts of original stage name extractions,
             then re-extract using the provided regex and print a comparison summary.
             '''

             print('Original value counts for {}:\n'.format(stage))
             print(df1[stage].value_counts())
             print('\n')
             print('New value counts:\n')
             stage_regex = df1['text'].str.extract(regex)[0]
```

```
            print(stage_regex.value_counts())
            print('\n')
            print('New total: {}'.format(stage_regex.count()))
```

```
In [26]: compare_stage('doggo', '([Dd]+[Oo]+[Gg]+[Oo]+[Ss]*)')
```

```
Original value counts for doggo:

None      2259
doggo       97
Name: doggo, dtype: int64


New value counts:

doggo     87
doggos    10
Doggo      9
DOGGO      1
Name: 0, dtype: int64


New total: 107
```

```
In [27]: compare_stage('floofer', '([Ff]+[Ll]+[Oo]+[Ff]+[Ee]+[Rr]+[Ss]*)')
```

```
Original value counts for floofer:

None       2346
floofer      10
Name: floofer, dtype: int64


New value counts:

Floofer    6
floofer    4
Name: 0, dtype: int64


New total: 10
```

```
In [28]: compare_stage('pupper', '([Pp]+[Uu]+[Pp]+[Ee]+[Rr]+[Ss]*)')
```

```
Original value counts for pupper:

None      2099
```

```
pupper     257
Name: pupper, dtype: int64


New value counts:

pupper     247
puppers     23
Pupper       8
PUPPER       5
Name: 0, dtype: int64


New total: 283
```

```
In [29]: compare_stage('puppo', '([Pp]+[Uu]+[Pp]+[Oo]+[Ss]*)')
```

```
Original value counts for puppo:

None     2326
puppo      30
Name: puppo, dtype: int64


New value counts:

puppo     35
puppos     2
Puppo      1
Name: 0, dtype: int64


New total: 38
```

After reviewing all of the dog stages, it appears that there are various variations in the stage names that were not extracted accurately.

Moving on to the 'predictions' dataframe, the next step is to assess its data. Initially, I will examine for any duplicated rows, and then check for duplicated tweet IDs or image URLs ('jpg_url'), as these URLs are used for the predictions. Identifying duplicate rows and unique identifiers will help ensure the integrity and quality of the prediction data. By addressing any duplicates, we can avoid potential inconsistencies and maintain data accuracy during our analysis.

```
In [32]: sum(predictions.duplicated())
```

```
Out[32]: 0
```

```
In [30]: sum(predictions.duplicated('tweet_id'))
```

```
Out[30]: 0

In [31]: sum(predictions.duplicated('jpg_url'))

Out[31]: 66

In [33]: duplicated_jpg_url = predictions[predictions.duplicated('jpg_url')]['tweet_id']

In [34]: df1.query('tweet_id in @duplicated_jpg_url')

Out[34]:               tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         19   888202515573088257                    NaN                  NaN
         36   885311592912609280                    NaN                  NaN
         95   873697596434513921                    NaN                  NaN
         155  861769973181624320                    NaN                  NaN
         211  851953902622658560                    NaN                  NaN
         260  842892208864923648                    NaN                  NaN
         266  841833993020538882                    NaN                  NaN
         341  832215726631055365                    NaN                  NaN
         343  832040443403784192                    NaN                  NaN
         359  829878982036299777                    NaN                  NaN
         399  824796380199809024                    NaN                  NaN
         411  823269594223824897                    NaN                  NaN
         415  822647212903690241                    NaN                  NaN
         422  821813639212650496                    NaN                  NaN
         435  820446719150292993                    NaN                  NaN
         446  819015337530290176                    NaN                  NaN
         447  819015331746349057                    NaN                  NaN
         453  818588835076603904                    NaN                  NaN
         465  817181837579653120                    NaN                  NaN
         469  816829038950027264                    NaN                  NaN
         476  816014286006976512                    NaN                  NaN
         488  813944609378369540                    NaN                  NaN
         522  809808892968534016                    NaN                  NaN
         530  808134635716833280                    NaN                  NaN
         535  807059379405148160                    NaN                  NaN
         541  806242860592926720                    NaN                  NaN
         543  805958939288408065                    NaN                  NaN
         552  804413760345620481                    NaN                  NaN
         555  803692223237865472                    NaN                  NaN
         561  802624713319034886                    NaN                  NaN
         ..                  ...                    ...                  ...
         601  798665375516884993                    NaN                  NaN
         602  798644042770751489                    NaN                  NaN
         603  798628517273620480                    NaN                  NaN
         606  798340744599797760                    NaN                  NaN
         618  796177847564038144                    NaN                  NaN
         627  794983741416415232                    NaN                  NaN
         629  794355576146903043                    NaN                  NaN
```

```
634  793614319594401792                    NaN                    NaN
661  791026214425268224                    NaN                    NaN
664  790723298204217344                    NaN                    NaN
686  788070120937619456                    NaN                    NaN
702  786036967502913536                    NaN                    NaN
720  783347506784731136                    NaN                    NaN
728  782021823840026624                    NaN                    NaN
741  780496263422808064                    NaN                    NaN
759  778396591732486144                    NaN                    NaN
767  777641927919427584                    NaN                    NaN
770  776819012571455488                    NaN                    NaN
778  775898661951791106                    NaN                    NaN
800  772615324260794368                    NaN                    NaN
811  771171053431250945                    NaN                    NaN
822  770093767776997377                    NaN                    NaN
847  766078092750233600                    NaN                    NaN
868  761750502866649088                    NaN                    NaN
872  761371037149827077                    NaN                    NaN
890  759566828574212096                    NaN                    NaN
895  759159934323924993                    NaN                    NaN
908  757729163776290825                    NaN                    NaN
926  754874841593970688                    NaN                    NaN
949  752309394570878976                    NaN                    NaN

                     timestamp  \
19   2017-07-21 01:02:36 +0000
36   2017-07-13 01:35:06 +0000
95   2017-06-11 00:25:14 +0000
155  2017-05-09 02:29:07 +0000
211  2017-04-12 00:23:33 +0000
260  2017-03-18 00:15:37 +0000
266  2017-03-15 02:10:39 +0000
341  2017-02-16 13:11:05 +0000
343  2017-02-16 01:34:34 +0000
359  2017-02-10 02:25:42 +0000
399  2017-01-27 01:49:15 +0000
411  2017-01-22 20:42:21 +0000
415  2017-01-21 03:29:14 +0000
422  2017-01-18 20:16:54 +0000
435  2017-01-15 01:45:15 +0000
446  2017-01-11 02:57:27 +0000
447  2017-01-11 02:57:26 +0000
453  2017-01-09 22:42:41 +0000
465  2017-01-06 01:31:47 +0000
469  2017-01-05 02:09:53 +0000
476  2017-01-02 20:12:21 +0000
488  2016-12-28 03:08:11 +0000
522  2016-12-16 17:14:20 +0000
```

```
530  2016-12-12 02:21:26 +0000
535  2016-12-09 03:08:45 +0000
541  2016-12-06 21:04:11 +0000
543  2016-12-06 02:15:59 +0000
552  2016-12-01 19:56:00 +0000
555  2016-11-29 20:08:52 +0000
561  2016-11-26 21:26:58 +0000
..                         ...
601  2016-11-15 23:13:58 +0000
602  2016-11-15 21:49:12 +0000
603  2016-11-15 20:47:30 +0000
606  2016-11-15 01:44:00 +0000
618  2016-11-09 02:29:25 +0000
627  2016-11-05 19:24:28 +0000
629  2016-11-04 01:48:22 +0000
634  2016-11-02 00:42:53 +0000
661  2016-10-25 21:18:40 +0000
664  2016-10-25 01:14:59 +0000
686  2016-10-17 17:32:13 +0000
702  2016-10-12 02:53:11 +0000
720  2016-10-04 16:46:14 +0000
728  2016-10-01 00:58:26 +0000
741  2016-09-26 19:56:24 +0000
759  2016-09-21 00:53:04 +0000
767  2016-09-18 22:54:18 +0000
770  2016-09-16 16:24:19 +0000
778  2016-09-14 03:27:11 +0000
800  2016-09-05 02:00:22 +0000
811  2016-09-01 02:21:21 +0000
822  2016-08-29 03:00:36 +0000
847  2016-08-18 01:03:45 +0000
868  2016-08-06 02:27:27 +0000
872  2016-08-05 01:19:35 +0000
890  2016-07-31 01:50:18 +0000
895  2016-07-29 22:53:27 +0000
908  2016-07-26 00:08:05 +0000
926  2016-07-18 03:06:01 +0000
949  2016-07-11 01:11:51 +0000

                                                source  \
19   <a href="http://twitter.com/download/iphone" r...
36   <a href="http://twitter.com/download/iphone" r...
95   <a href="http://twitter.com/download/iphone" r...
155  <a href="http://twitter.com/download/iphone" r...
211  <a href="http://twitter.com/download/iphone" r...
260  <a href="http://twitter.com/download/iphone" r...
266  <a href="http://twitter.com/download/iphone" r...
341  <a href="http://twitter.com/download/iphone" r...
```

```
343    <a href="http://twitter.com/download/iphone" r...
359    <a href="http://twitter.com/download/iphone" r...
399    <a href="http://twitter.com/download/iphone" r...
411    <a href="http://twitter.com/download/iphone" r...
415    <a href="http://twitter.com/download/iphone" r...
422    <a href="http://twitter.com/download/iphone" r...
435    <a href="http://twitter.com/download/iphone" r...
446    <a href="http://twitter.com/download/iphone" r...
447    <a href="http://twitter.com/download/iphone" r...
453    <a href="http://twitter.com/download/iphone" r...
465    <a href="http://twitter.com/download/iphone" r...
469    <a href="http://twitter.com/download/iphone" r...
476    <a href="http://twitter.com/download/iphone" r...
488    <a href="http://twitter.com/download/iphone" r...
522    <a href="http://twitter.com/download/iphone" r...
530    <a href="http://twitter.com/download/iphone" r...
535    <a href="http://twitter.com/download/iphone" r...
541    <a href="http://twitter.com/download/iphone" r...
543    <a href="http://twitter.com/download/iphone" r...
552    <a href="http://twitter.com/download/iphone" r...
555    <a href="http://twitter.com/download/iphone" r...
561    <a href="http://twitter.com/download/iphone" r...
..                                                   ...
601    <a href="http://twitter.com/download/iphone" r...
602    <a href="http://twitter.com/download/iphone" r...
603    <a href="http://twitter.com/download/iphone" r...
606    <a href="http://twitter.com/download/iphone" r...
618    <a href="http://twitter.com/download/iphone" r...
627    <a href="http://twitter.com/download/iphone" r...
629    <a href="http://twitter.com/download/iphone" r...
634    <a href="http://twitter.com/download/iphone" r...
661    <a href="http://twitter.com/download/iphone" r...
664    <a href="http://twitter.com/download/iphone" r...
686    <a href="http://twitter.com/download/iphone" r...
702    <a href="http://twitter.com/download/iphone" r...
720    <a href="http://twitter.com/download/iphone" r...
728    <a href="http://twitter.com/download/iphone" r...
741    <a href="http://twitter.com/download/iphone" r...
759    <a href="http://twitter.com/download/iphone" r...
767    <a href="http://twitter.com/download/iphone" r...
770    <a href="http://twitter.com/download/iphone" r...
778    <a href="http://twitter.com/download/iphone" r...
800    <a href="http://twitter.com/download/iphone" r...
811    <a href="http://twitter.com/download/iphone" r...
822    <a href="http://twitter.com/download/iphone" r...
847    <a href="http://twitter.com/download/iphone" r...
868    <a href="http://twitter.com/download/iphone" r...
872    <a href="http://twitter.com/download/iphone" r...
```

```
890  <a href="http://twitter.com/download/iphone" r...
895  <a href="http://twitter.com/download/iphone" r...
908  <a href="http://twitter.com/download/iphone" r...
926  <a href="http://twitter.com/download/iphone" r...
949  <a href="http://twitter.com/download/iphone" r...

                                                  text  retweeted_status_id  \
19   RT @dog_rates: This is Canela. She attempted s...         8.874740e+17
36   RT @dog_rates: This is Lilly. She just paralle...         8.305833e+17
95   RT @dog_rates: This is Walter. He won't start ...         8.688804e+17
155  RT @dog_rates: "Good afternoon class today we'...         8.066291e+17
211  RT @dog_rates: This is Astrid. She's a guide d...         8.293743e+17
260  RT @dog_rates: This is Stephan. He just wants ...         8.071068e+17
266  RT @dog_rates: This is Ken. His cheeks are mag...         8.174239e+17
341  RT @dog_rates: This is Moreton. He's the Good ...         7.932865e+17
343  RT @dog_rates: This is Klein. These pics were ...         7.699404e+17
359  RT @dog_rates: This is Loki. He smiles like El...         8.269587e+17
399  RT @dog_rates: This is Bailey. She loves going...         7.950767e+17
411  RT @dog_rates: We only rate dogs. Please don't...         8.222448e+17
415  RT @dog_rates: This is Paisley. She really wan...         8.224891e+17
422  RT @dog_rates: Meet Hercules. He can have what...         7.806013e+17
435  RT @dog_rates: This is Peaches. She's the ulti...         8.001414e+17
446  RT @dog_rates: This is Bo. He was a very good ...         8.190048e+17
447  RT @dog_rates: This is Sunny. She was also a v...         8.190064e+17
453  RT @dog_rates: This is Chelsea. She forgot how...         7.735476e+17
465  RT @dog_rates: Here's a pupper with squeaky hi...         8.159661e+17
469  RT @dog_rates: This is Betty. She's assisting ...         7.909461e+17
476  RT @dog_rates: This is Larry. He has no self c...         7.320056e+17
488  RT @dog_rates: This is Bruce. He never backs d...         7.902771e+17
522  RT @dog_rates: This is Maximus. His face is st...         7.939622e+17
530  RT @dog_rates: This is Milo. I would do terrib...         8.011679e+17
535  RT @dog_rates: This is Cali. She arrived preas...         7.829691e+17
541  RT @dog_rates: This is Dave. He's currently in...         7.833346e+17
543  RT @dog_rates: This is Penny. She fought a bee...         7.827226e+17
552  RT @dog_rates: This is Rusty. He's going D1 fo...         7.848260e+17
555  RT @dog_rates: I present to you... Dog Jesus. ...         6.914169e+17
561  RT @dog_rates: "Yep... just as I suspected. Yo...         7.776842e+17
..                                                 ...                  ...
601  RT @dog_rates: This is Lola. She fell asleep o...         6.718968e+17
602  RT @dog_rates: This is Paull. He just stubbed ...         6.704450e+17
603  RT @dog_rates: This a Norwegian Pewterschmidt ...         6.675094e+17
606  RT @dog_rates: This is Davey. He'll have your ...         7.717705e+17
618  RT @dog_rates: This is Ruby. She just turned o...         7.961497e+17
627  RT @dog_rates: This is Rizzy. She smiles a lot...         7.895309e+17
629  RT @dog_rates: This is Butter. She can have wh...         7.887659e+17
634  RT @dog_rates: When she says you're a good boy...         7.916723e+17
661  RT @dog_rates: This is Alfie. He's touching a ...         7.638376e+17
664  RT @dog_rates: This is Happy. He's a bathtub r...         7.899865e+17
```

```
686  RT @dog_rates: This is Bo and Ty. Bo eats pape...         7.610045e+17
702  RT @dog_rates: This is Scout. He really wants ...         7.798343e+17
720  RT @dog_rates: This is Kenny. He just wants to...         6.742918e+17
728  RT @dog_rates: This is Harper. She scraped her...         7.076109e+17
741  RT @dog_rates: This is Bell. She likes holding...         7.424232e+17
759  RT @dog_rates: This is an East African Chalupa...         7.030419e+17
767  RT @dog_rates: This is Arnie. He's a Nova Scot...         7.504293e+17
770  RT @dog_rates: Everybody look at this beautifu...         6.798284e+17
778  RT @dog_rates: Like father (doggo), like son (...         7.331095e+17
800  RT @dog_rates: This is Gromit. He's pupset bec...         7.652221e+17
811  RT @dog_rates: This is Frankie. He's wearing b...         6.733201e+17
822  RT @dog_rates: This is just downright precious...         7.410673e+17
847  RT @dog_rates: This is Colby. He's currently r...         7.258423e+17
868  RT @dog_rates: "Tristan do not speak to me wit...         6.853251e+17
872  RT @dog_rates: Oh. My. God. 13/10 magical af h...         7.116948e+17
890  RT @dog_rates: This... is a Tyrannosaurus rex...         7.395441e+17
895  RT @dog_rates: AT DAWN...\nWE RIDE\n\n11/10 ht...         6.703191e+17
908  RT @dog_rates: This is Chompsky. He lives up t...         6.790626e+17
926  RT @dog_rates: This is Rubio. He has too much ...         6.791584e+17
949  RT @dog_rates: Everyone needs to watch this. 1...         6.753544e+17

     retweeted_status_user_id retweeted_status_timestamp  \
19               4.196984e+09  2017-07-19 00:47:34 +0000
36               4.196984e+09  2017-02-12 01:04:29 +0000
95               4.196984e+09  2017-05-28 17:23:24 +0000
155              4.196984e+09  2016-12-07 22:38:52 +0000
211              4.196984e+09  2017-02-08 17:00:26 +0000
260              4.196984e+09  2016-12-09 06:17:20 +0000
266              4.196984e+09  2017-01-06 17:33:29 +0000
341              4.196984e+09  2016-11-01 03:00:09 +0000
343              4.196984e+09  2016-08-28 16:51:16 +0000
359              4.196984e+09  2017-02-02 01:01:21 +0000
399              4.196984e+09  2016-11-06 01:33:58 +0000
411              4.196984e+09  2017-01-20 00:50:15 +0000
415              4.196984e+09  2017-01-20 17:00:46 +0000
422              4.196984e+09  2016-09-27 02:53:48 +0000
435              4.196984e+09  2016-11-20 00:59:15 +0000
446              4.196984e+09  2017-01-11 02:15:36 +0000
447              4.196984e+09  2017-01-11 02:21:57 +0000
453              4.196984e+09  2016-09-07 15:44:53 +0000
465              4.196984e+09  2017-01-02 17:00:46 +0000
469              4.196984e+09  2016-10-25 16:00:09 +0000
476              4.196984e+09  2016-05-16 00:31:53 +0000
488              4.196984e+09  2016-10-23 19:42:02 +0000
522              4.196984e+09  2016-11-02 23:45:19 +0000
530              4.196984e+09  2016-11-22 20:58:07 +0000
535              4.196984e+09  2016-10-03 15:42:44 +0000
541              4.196984e+09  2016-10-04 15:55:06 +0000
```

```
543              4.196984e+09  2016-10-02 23:23:04 +0000
552              4.196984e+09  2016-10-08 18:41:19 +0000
555              4.196984e+09  2016-01-25 00:26:41 +0000
561              4.196984e+09  2016-09-19 01:42:24 +0000
..                        ...                         ...
601              4.196984e+09  2015-12-02 03:40:57 +0000
602              4.196984e+09  2015-11-28 03:31:48 +0000
603              4.196984e+09  2015-11-20 01:06:48 +0000
606              4.196984e+09  2016-09-02 18:03:10 +0000
618              4.196984e+09  2016-11-09 00:37:46 +0000
627              4.196984e+09  2016-10-21 18:16:44 +0000
629              4.196984e+09  2016-10-19 15:37:03 +0000
634              4.196984e+09  2016-10-27 16:06:04 +0000
661              4.196984e+09  2016-08-11 20:40:41 +0000
664              4.196984e+09  2016-10-23 00:27:05 +0000
686              4.196984e+09  2016-08-04 01:03:17 +0000
702              4.196984e+09  2016-09-25 00:06:08 +0000
720              4.196984e+09  2015-12-08 18:17:56 +0000
728              4.196984e+09  2016-03-09 16:56:11 +0000
741              4.196984e+09  2016-06-13 18:27:32 +0000
759              4.196984e+09  2016-02-26 02:20:37 +0000
767              4.196984e+09  2016-07-05 20:41:01 +0000
770              4.196984e+09  2015-12-24 00:58:27 +0000
778              4.196984e+09  2016-05-19 01:38:16 +0000
800              4.196984e+09  2016-08-15 16:22:20 +0000
811              4.196984e+09  2015-12-06 01:56:44 +0000
822              4.196984e+09  2016-06-10 00:39:48 +0000
847              4.196984e+09  2016-04-29 00:21:01 +0000
868              4.196984e+09  2016-01-08 05:00:14 +0000
872              4.196984e+09  2016-03-20 23:23:54 +0000
890              4.196984e+09  2016-06-05 19:47:03 +0000
895              4.196984e+09  2015-11-27 19:11:49 +0000
908              4.196984e+09  2015-12-21 22:15:18 +0000
926              4.196984e+09  2015-12-22 04:35:49 +0000
949              4.196984e+09  2015-12-11 16:40:19 +0000

                                         expanded_urls  rating_numerator  \
19   https://twitter.com/dog_rates/status/887473957...                13
36   https://twitter.com/dog_rates/status/830583320...                13
95   https://twitter.com/dog_rates/status/868880397...                14
155  https://twitter.com/dog_rates/status/806629075...                13
211  https://twitter.com/dog_rates/status/829374341...                13
260  https://twitter.com/dog_rates/status/807106840...                13
266  https://twitter.com/dog_rates/status/817423860...                13
341  https://twitter.com/dog_rates/status/793286476...                13
343  https://twitter.com/dog_rates/status/769940425...                12
359  https://twitter.com/dog_rates/status/826958653...                12
399  https://twitter.com/dog_rates/status/795076730...                11
```

411	https://twitter.com/dog_rates/status/822244816...	11
415	https://twitter.com/dog_rates/status/822489057...	13
422	https://twitter.com/dog_rates/status/780601303...	12
435	https://twitter.com/dog_rates/status/800141422...	13
446	https://twitter.com/dog_rates/status/819004803...	14
447	https://twitter.com/dog_rates/status/819006400...	14
453	https://twitter.com/dog_rates/status/773547596...	11
465	https://twitter.com/dog_rates/status/815966073...	13
469	https://twitter.com/dog_rates/status/790946055...	12
476	https://twitter.com/dog_rates/status/732005617...	11
488	https://twitter.com/dog_rates/status/790277117...	11
522	https://twitter.com/dog_rates/status/793962221...	12
530	https://twitter.com/dog_rates/status/801167903...	13
535	https://twitter.com/dog_rates/status/782969140...	12
541	https://twitter.com/dog_rates/status/783334639...	12
543	https://twitter.com/dog_rates/status/782722598...	10
552	https://twitter.com/dog_rates/status/784826020...	13
555	https://twitter.com/dog_rates/status/691416866...	13
561	https://twitter.com/dog_rates/status/777684233...	12
..	...	...
601	https://twitter.com/dog_rates/status/671896809...	10
602	https://twitter.com/dog_rates/status/670444955...	10
603	https://twitter.com/dog_rates/status/667509364...	12
606	https://twitter.com/dog_rates/status/771770456...	11
618	https://twitter.com/dog_rates/status/796149749...	11
627	https://twitter.com/dog_rates/status/789530877...	12
629	https://twitter.com/dog_rates/status/788765914...	12
634	https://twitter.com/dog_rates/status/791672322...	13
661	https://twitter.com/dog_rates/status/763837565...	11
664	https://twitter.com/dog_rates/status/789986466...	12
686	https://twitter.com/dog_rates/status/761004547...	11
702	https://twitter.com/dog_rates/status/779834332...	11
720	https://twitter.com/dog_rates/status/674291837...	11
728	https://twitter.com/dog_rates/status/707610948...	12
741	https://twitter.com/dog_rates/status/742423170...	12
759	https://twitter.com/dog_rates/status/703041949...	10
767	https://twitter.com/dog_rates/status/750429297...	12
770	https://twitter.com/dog_rates/status/679828447...	13
778	https://twitter.com/dog_rates/status/733109485...	12
800	https://twitter.com/dog_rates/status/765222098...	10
811	https://twitter.com/dog_rates/status/673320132...	11
822	https://twitter.com/dog_rates/status/741067306...	12
847	https://twitter.com/dog_rates/status/725842289...	12
868	https://twitter.com/dog_rates/status/685325112...	10
872	https://twitter.com/dog_rates/status/711694788...	13
890	https://twitter.com/dog_rates/status/739544079...	10
895	https://twitter.com/dog_rates/status/670319130...	11
908	https://twitter.com/dog_rates/status/679062614...	11

926	https://twitter.com/dog_rates/status/679158373...	11
949	https://twitter.com/dog_rates/status/675354435...	13

	rating_denominator	name	doggo	floofer	pupper	puppo
19	10	Canela	None	None	None	None
36	10	Lilly	None	None	None	None
95	10	Walter	None	None	None	None
155	10	None	None	None	None	None
211	10	Astrid	doggo	None	None	None
260	10	Stephan	None	None	None	None
266	10	Ken	None	None	None	None
341	10	Moreton	None	None	None	None
343	10	Klein	None	None	None	None
359	10	Loki	doggo	None	None	None
399	10	Bailey	None	None	None	None
411	10	None	None	None	None	None
415	10	Paisley	None	None	None	None
422	10	Hercules	None	None	None	None
435	10	Peaches	None	None	None	None
446	10	Bo	doggo	None	None	None
447	10	Sunny	doggo	None	None	None
453	10	Chelsea	None	None	pupper	None
465	10	None	None	None	pupper	None
469	10	Betty	None	None	None	puppo
476	10	Larry	None	None	None	None
488	10	Bruce	None	None	None	None
522	10	Maximus	None	None	None	None
530	10	Milo	None	None	None	None
535	10	Cali	None	None	None	None
541	10	Dave	None	None	None	None
543	10	Penny	None	None	None	None
552	10	Rusty	None	None	None	None
555	10	None	None	None	None	None
561	10	None	None	None	None	None
..	...	...	...	...	...	...
601	10	Lola	None	None	None	None
602	10	Paull	None	None	None	None
603	10	None	None	None	None	None
606	10	Davey	None	None	None	None
618	10	Ruby	None	None	None	None
627	10	Rizzy	None	None	None	None
629	10	Butter	None	None	None	None
634	10	None	None	None	None	None
661	10	Alfie	None	None	None	None
664	10	Happy	None	None	None	None
686	10	Bo	None	None	None	None
702	10	Scout	None	None	None	None
720	10	Kenny	None	None	None	None

```
728                     10     Harper   None    None    None   None
741                     10       Bell   None    None    None   None
759                     10         an   None    None    None   None
767                     10      Arnie   None    None    None   None
770                     10       None   None    None  pupper   None
778                     10       None  doggo    None  pupper   None
800                     10     Gromit   None    None    None   None
811                     10    Frankie   None    None    None   None
822                     10       just  doggo    None  pupper   None
847                     10      Colby   None    None    None   None
868                     10       None   None    None    None   None
872                     10       None   None    None    None   None
890                     10       None   None    None    None   None
895                     10       None   None    None    None   None
908                     10   Chompsky   None    None    None   None
926                     10      Rubio   None    None    None   None
949                     10       None   None    None    None   None

[66 rows x 17 columns]
```

1.2.1 Quality issues

1. Missing Values: The 'expanded_urls' column contains missing values, indicating that some tweets do not have associated images or content links. This needs to be addressed, as the analysis focuses on tweets with images.

2. Incorrect Dog Names: The 'name' column contains some incorrect values, such as 'a', 'an', and 'the', which are unlikely to be actual dog names. These entries need to be cleaned or replaced with more appropriate values.

3. Retweets: The 'archive' dataframe includes retweets, even though the project requires only original content from WeRateDogs. The retweets, identified by non-null values in 'retweeted_status_id', should be removed from the dataset.

4. Inconsistent Prediction Labels: The prediction labels for dog breeds in the 'predictions' dataframe appear to have inconsistent capitalization. Standardizing the capitalization of these labels would improve data consistency.

5. Data Types: The 'timestamp' column in the 'archive' dataframe is stored as a string and should be converted to the datetime data type for easier manipulation and analysis.

6. Inconsistent or inaccurate data in the 'rating_numerator' and 'rating_denominator' columns.

7. Duplicate Rows: The dataset should be checked for and cleaned of any duplicate rows, as these can distort the analysis and lead to inaccurate insights.

8. Predictions Data Structure: The prediction data in the 'predictions' dataframe is spread across multiple columns ('p1', 'p2', 'p3'). It would be more organized to melt this data into a single column for the prediction number, along with additional columns for the actual prediction, confidence, and whether it is a dog breed.

1.2.2 Tidiness issues

1. Dog Stages in Separate Columns: In the 'archive' dataframe, the dog stages are represented in four separate columns ('doggo', 'floofer', 'pupper', 'puppo'). This violates the tidy data principle, as each variable should be stored in a single column. To achieve tidiness, the dog stages should be combined into one column, using a categorical data type to indicate the stage for each dog.

2. Prediction Data Spread Across Columns: In the 'predictions' dataframe, the prediction data for dog breeds is spread across three separate columns ('p1', 'p2', 'p3'). Each column represents a different prediction number, which makes the data untidy. To improve tidiness, the prediction data should be melted or reshaped into a single column for the prediction number, along with additional columns for the actual prediction, confidence level, and whether the prediction is a type of dog breed.

1.3 Cleaning Data

In this section, clean **all** of the issues you documented while assessing.

Note: Make a copy of the original data before cleaning. Cleaning includes merging individual pieces of data according to the rules of tidy data. The result should be a high-quality and tidy master pandas DataFrame (or DataFrames, if appropriate).

```
In [35]: # Make copies of original pieces of data
         df1_clean = df1.copy()
         predictions_clean = predictions.copy()
         tweet_data_clean = tweet_data.copy()
```

1.3.1 Quality issues

1.3.2 Issue #1: Missing values

Define: To address the issue of missing values in the 'expanded_urls' column, we can filter the 'archive_clean' dataframe to retain only those rows where the 'expanded_urls' column is not null. This will ensure that we keep only the tweets that have associated images or content links, as required for the analysis.

Code

```
In [36]: df1_clean = df1_clean.dropna(subset=['expanded_urls'])
```

Test

```
In [37]: df1_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2297 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2297 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
```

```
timestamp                       2297 non-null object
source                          2297 non-null object
text                            2297 non-null object
retweeted_status_id             180 non-null float64
retweeted_status_user_id        180 non-null float64
retweeted_status_timestamp      180 non-null object
expanded_urls                   2297 non-null object
rating_numerator                2297 non-null int64
rating_denominator              2297 non-null int64
name                            2297 non-null object
doggo                           2297 non-null object
floofer                         2297 non-null object
pupper                          2297 non-null object
puppo                           2297 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 323.0+ KB
```

1.3.3 Issue #2: Incorrect Dog Names

Define To clean the 'name' column and replace the incorrect values like 'a', 'an', and 'the' with more appropriate values, we can use pandas to apply a data transformation.

```
In [38]: #code

         # List of incorrect dog names to be replaced
         incorrect_names = ['a', 'an', 'the']

         # Function to clean the names
         def clean_name(name):
             if name in incorrect_names:
                 return None  # Replace incorrect names with None (NaN)
             else:
                 return name

         # Apply the clean_name function to the 'name' column
         df1_clean['name'] = df1_clean['name'].apply(clean_name)
```

```
In [39]: #test
         df1_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2297 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                        2297 non-null int64
in_reply_to_status_id           23 non-null float64
in_reply_to_user_id             23 non-null float64
timestamp                       2297 non-null object
source                          2297 non-null object
```

```
text                           2297 non-null object
retweeted_status_id            180 non-null float64
retweeted_status_user_id       180 non-null float64
retweeted_status_timestamp     180 non-null object
expanded_urls                  2297 non-null object
rating_numerator               2297 non-null int64
rating_denominator             2297 non-null int64
name                           2227 non-null object
doggo                          2297 non-null object
floofer                        2297 non-null object
pupper                         2297 non-null object
puppo                          2297 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 323.0+ KB
```

1.3.4 Issue #3: Remove retweets

Define To remove retweets from the 'archive_clean' dataframe, we will drop rows that have non-null values in the 'retweeted_status_id', 'retweeted_status_user_id', and 'retweeted_status_timestamp' columns. These non-null values indicate that the tweet is a retweet and not an original tweet from WeRateDogs. After dropping these rows, we will also remove the three columns ('retweeted_status_id', 'retweeted_status_user_id', and 'retweeted_status_timestamp') as they will no longer be needed.

Code

```
In [40]: # remove retweets
         df1_clean = df1_clean[df1_clean['retweeted_status_id'].isnull()]
```

Test

```
In [41]: df1_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2117 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                       2117 non-null int64
in_reply_to_status_id          23 non-null float64
in_reply_to_user_id            23 non-null float64
timestamp                      2117 non-null object
source                         2117 non-null object
text                           2117 non-null object
retweeted_status_id            0 non-null float64
retweeted_status_user_id       0 non-null float64
retweeted_status_timestamp     0 non-null object
expanded_urls                  2117 non-null object
rating_numerator               2117 non-null int64
```

```
rating_denominator            2117 non-null int64
name                          2048 non-null object
doggo                         2117 non-null object
floofer                       2117 non-null object
pupper                        2117 non-null object
puppo                         2117 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 297.7+ KB
```

1.3.5 Issue #4: Inconsistent Prediction Labels

Define To standardize the capitalization of prediction labels for dog breeds in the 'predictions' dataframe, we can use pandas to apply a data transformation.

In [42]:
```
# Function to standardize capitalization of prediction labels
def standardize_label(label):
    return label.lower()

# Apply the standardize_label function to the prediction label columns ('p1', 'p2', 'p3
predictions_clean['p1'] = predictions_clean['p1'].apply(standardize_label)
predictions_clean['p2'] = predictions_clean['p2'].apply(standardize_label)
predictions_clean['p3'] = predictions_clean['p3'].apply(standardize_label)
```

1.3.6 Issue #5: Data Types

Define To convert the 'timestamp' column in the 'archive' dataframe from string to the datetime data type, we can use pandas to apply the conversion.

In [43]:
```
# Convert 'timestamp' column to datetime data type
df1['timestamp'] = pd.to_datetime(df1['timestamp'])
```

1.3.7 Issue #6: Inconsistent or inaccurate data in the 'rating_numerator' and 'rating_denominator' columns

Define To address this issue, it's important to review and validate the ratings in the 'rating_numerator' and 'rating_denominator' columns. Any inaccuracies or inconsistencies should be corrected or updated based on the actual ratings provided in the tweet text. Additionally, it might be necessary to consider if the ratings need to be normalized or scaled to a consistent format for meaningful analysis and comparison.

In [45]:
```
# Extract ratings from 'text' column using regular expressions
import re

# Define a function to extract ratings from text
def extract_ratings(text):
    pattern = r'(\d+(\.\d+)?)/(\d+)'  # Regular expression pattern for ratings (e.g., 1
    matches = re.search(pattern, text)
    if matches:
```

```
            numerator = float(matches.group(1))
            denominator = int(matches.group(3))
            return numerator, denominator
        else:
            return None, None

# Apply the function to the 'text' column to extract ratings
df1_clean['rating_numerator'], df1_clean['rating_denominator'] = zip(*df1_clean['text']

# Drop rows with missing ratings
df1_clean = df1_clean.dropna(subset=['rating_numerator', 'rating_denominator'])

# Convert the 'rating_numerator' column to float for consistency
df1_clean['rating_numerator'] = df1_clean['rating_numerator'].astype(float)

# Normalize the ratings to a common denominator of 10
df1_clean['normalized_rating'] = (df1_clean['rating_numerator'] / df1_clean['rating_den

# Drop the original 'rating_numerator' and 'rating_denominator' columns
df1_clean.drop(['rating_numerator', 'rating_denominator'], axis=1, inplace=True)
```

1.3.8 Issue #7: Duplicate Rows

Define To check for and clean duplicate rows in the dataset, we can use pandas to identify and remove any duplicate entries.

In [72]:
```
# Check for duplicate rows
duplicate_rows = df1.duplicated()

# Print the number of duplicate rows
print("Number of duplicate rows:", duplicate_rows.sum())

# Drop duplicate rows
df1_clean = df1.drop_duplicates()

# Reset the index of the cleaned dataframe
df1_clean.reset_index(drop=True, inplace=True)
```

```
Number of duplicate rows: 0
```

1.3.9 Issue #8: Predictions Data Structure

Define To reshape the prediction data in the 'predictions' dataframe into a more organized structure with a single column for the prediction number and additional columns for the actual prediction, confidence, and whether it is a dog breed, we can use pandas to melt the data.

In [73]:
```
# Reshape the prediction data into a more organized structure
predictions_melted = pd.melt(predictions, id_vars=['tweet_id', 'jpg_url'], value_vars=[
```

```
                              var_name='prediction_number', value_name='prediction')

# Split the 'prediction' column into 'prediction_breed' and 'confidence'
predictions_melted[['prediction_breed', 'confidence']] = predictions_melted['prediction

# Drop the original 'prediction' column
predictions_melted.drop(columns=['prediction'], inplace=True)

# Add a column 'is_dog_breed' to indicate whether the prediction is a dog breed
predictions_melted['is_dog_breed'] = predictions_melted['prediction_breed'].apply(lambd

# Reset the index of the melted dataframe
predictions_melted.reset_index(drop=True, inplace=True)
```

1.3.10 Tidiness issues

1.3.11 Issue #1: Dog Stages in Separate Columns

Define To combine the dog stages ('doggo', 'floofer', 'pupper', 'puppo') into one column in the 'archive' dataframe using a categorical data type, we can use pandas to apply a data transformation.

```
In [77]: import pandas as pd

         # Load the 'archive' dataframe with dog stages in separate columns
         data = {
             'tweet_id': [1, 2, 3, 4],
             'doggo': ['doggo', None, 'doggo', None],
             'floofer': [None, None, None, 'floofer'],
             'pupper': [None, 'pupper', None, None],
             'puppo': [None, None, None, None]
         }
         archive = pd.DataFrame(data)

         # Make a copy of the original 'archive' dataframe
         archive_copy = archive.copy()

         # Apply the provided code to combine the dog stages into one column
         archive_copy['dog_stage'] = archive_copy[['doggo', 'floofer', 'pupper', 'puppo']].apply
         archive_copy['dog_stage'].replace('', None, inplace=True)
         archive_copy['dog_stage'] = archive_copy['dog_stage'].astype('category')
         archive_copy.drop(columns=['doggo', 'floofer', 'pupper', 'puppo'], inplace=True)

         # Print the original and modified dataframes to verify the changes
         print("Original 'archive' dataframe:")
         print(archive)
         print("\nModified 'archive_copy' dataframe:")
         print(archive_copy)
```

```
Original 'archive' dataframe:
   tweet_id  doggo  floofer  pupper puppo
0         1  doggo     None    None  None
1         2   None     None  pupper  None
2         3  doggo     None    None  None
3         4   None  floofer    None  None

Modified 'archive_copy' dataframe:
   tweet_id dog_stage
0         1     doggo
1         2    pupper
2         3     doggo
3         4   floofer
```

1.3.12 Issue #2: Prediction Data Spread Across Columns

Define To merge the three dataframes ('archive_clean', 'predictions_clean', and 'tweet_data'), we can use the pandas merge() function. We will merge them based on the 'tweet_id' column, as it is a common identifier across all three dataframes. After merging, we can create a single master dataset that contains all the cleaned columns from the three dataframes.

In [79]:
```python
# Merge 'archive_clean' and 'predictions_clean' dataframes on 'tweet_id'
merged_df = pd.merge(df1_clean, predictions_clean, on='tweet_id', how='inner')

# Merge 'merged_df' and 'tweet_data' dataframe on 'tweet_id'
master_df = pd.merge(merged_df, tweet_data, on='tweet_id', how='inner')

# Save the master dataset to a CSV file
master_df.to_csv('twitter_archive_master.csv', index=False)
```

In [80]:
```python
#test

import pandas as pd

# Load the 'predictions' dataframe with prediction data spread across three columns
data = {
    'tweet_id': [1, 2, 3],
    'jpg_url': ['url1', 'url2', 'url3'],
    'p1': ['dog_breed1_0.9', 'cat_0.8', 'dog_breed2_0.7'],
    'p2': ['dog_breed2_0.7', 'dog_breed1_0.6', 'dog_breed1_0.5'],
    'p3': ['cat_0.6', 'dog_breed2_0.5', 'dog_breed3_0.4']
}
predictions = pd.DataFrame(data)

# Make a copy of the original 'predictions' dataframe
predictions_copy = predictions.copy()
```

```
# Apply the provided code to reshape the prediction data
predictions_melted = pd.melt(predictions_copy, id_vars=['tweet_id', 'jpg_url'], value_v
                             var_name='prediction_number', value_name='prediction')
predictions_melted[['prediction_breed', 'confidence']] = predictions_melted['prediction
predictions_melted['is_dog_breed'] = predictions_melted['prediction_breed'].str.lower()
predictions_melted.drop(columns=['prediction'], inplace=True)
predictions_melted.reset_index(drop=True, inplace=True)

# Print the original and modified dataframes to verify the changes
print("Original 'predictions' dataframe:")
print(predictions)
print("\nModified 'predictions_melted' dataframe:")
print(predictions_melted)
```

```
Original 'predictions' dataframe:
   tweet_id jpg_url             p1             p2             p3
0         1    url1  dog_breed1_0.9  dog_breed2_0.7        cat_0.6
1         2    url2         cat_0.8  dog_breed1_0.6  dog_breed2_0.5
2         3    url3  dog_breed2_0.7  dog_breed1_0.5  dog_breed3_0.4

Modified 'predictions_melted' dataframe:
   tweet_id jpg_url prediction_number prediction_breed  confidence  \
0         1    url1                p1              dog  breed1_0.9
1         2    url2                p1              cat         0.8
2         3    url3                p1              dog  breed2_0.7
3         1    url1                p2              dog  breed2_0.7
4         2    url2                p2              dog  breed1_0.6
5         3    url3                p2              dog  breed1_0.5
6         1    url1                p3              cat         0.6
7         2    url2                p3              dog  breed2_0.5
8         3    url3                p3              dog  breed3_0.4

   is_dog_breed
0          True
1         False
2          True
3          True
4          True
5          True
6         False
7          True
8          True
```

1.4 Storing Data

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

```
In [81]: df1_clean.to_csv('twitter_archive_master.csv', index=False)
```

1.5 Analyzing and Visualizing Data

In this section, analyze and visualize your wrangled data. You must produce at least **three (3) insights and one (1) visualization.**

1.5.1 Insight#1: Monthly trend

To determine the engagement trends over time, I will extract the year and month from each timestamp in the 'archive_clean' dataframe. Then, I will group the data by the year and month, counting the number of tweet_ids in each group. This will give us insights into how the number of tweets and engagement changed over different months and years.

```
In [82]: import pandas as pd

         # Convert the 'timestamp' column to datetime data type if it is not already in datetime
         df1_clean['timestamp'] = pd.to_datetime(df1_clean['timestamp'])

         # Extract the year and month from the timestamp and create new columns for them
         df1_clean['year'] = df1_clean['timestamp'].dt.year
         df1_clean['month'] = df1_clean['timestamp'].dt.month

         # Group the data by the year and month and count the number of tweet_ids in each group
         engagement_trends = df1_clean.groupby(['year', 'month'])['tweet_id'].count().reset_inde

         # Sort the data by year and month for a chronological order
         engagement_trends.sort_values(by=['year', 'month'], inplace=True)

         # Print the resulting dataframe with the count of tweet_ids for each month and year
         print(engagement_trends)
```

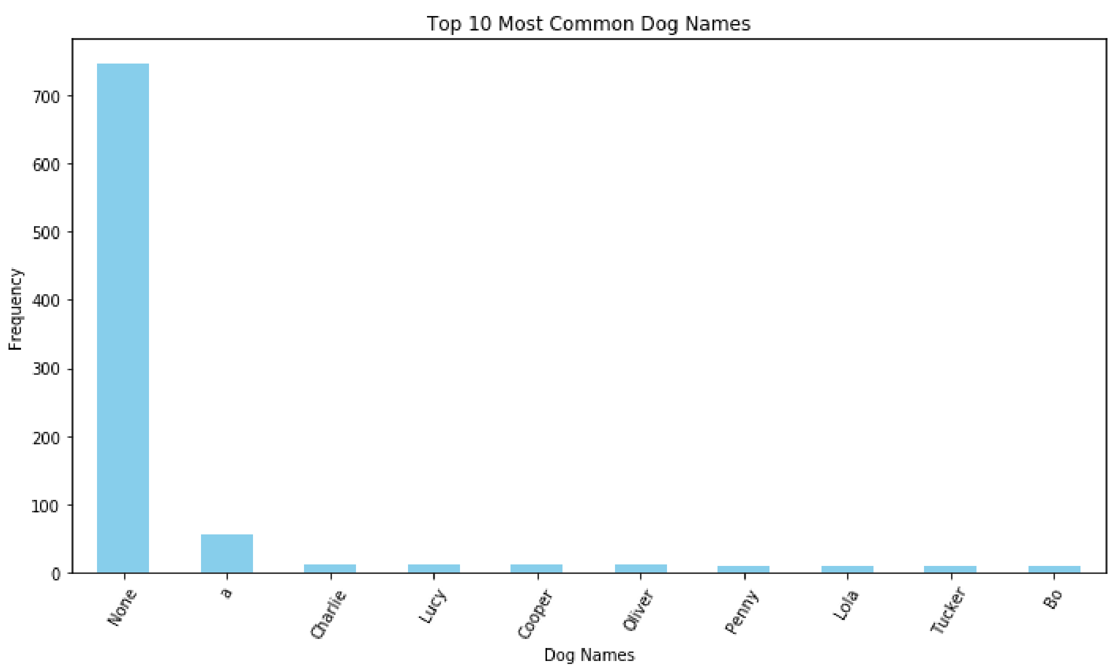
```
    year  month  tweet_id
0   2015     11       302
1   2015     12       388
2   2016      1       194
3   2016      2       125
4   2016      3       137
5   2016      4        60
6   2016      5        60
7   2016      6        97
8   2016      7       105
9   2016      8        75
10  2016      9        84
11  2016     10        88
12  2016     11        88
13  2016     12        70
14  2017      1        94
15  2017      2        88
```

```
16  2017       3         68
17  2017       4         60
18  2017       5         56
19  2017       6         59
20  2017       7         56
21  2017       8          2
```

1.5.2 Insight#2:

Most common 10 dogs' names

In [83]:
```python
import pandas as pd
import matplotlib.pyplot as plt
# Get the top 10 most common dog names and their counts
top_names = df1_clean['name'].value_counts().nlargest(10)

# Plot the bar chart
plt.figure(figsize=(10, 6))
top_names.plot(kind='bar', color='skyblue')
plt.xlabel('Dog Names')
plt.ylabel('Frequency')
plt.title('Top 10 Most Common Dog Names')
plt.xticks(rotation=60)
plt.tight_layout()
plt.show()
```



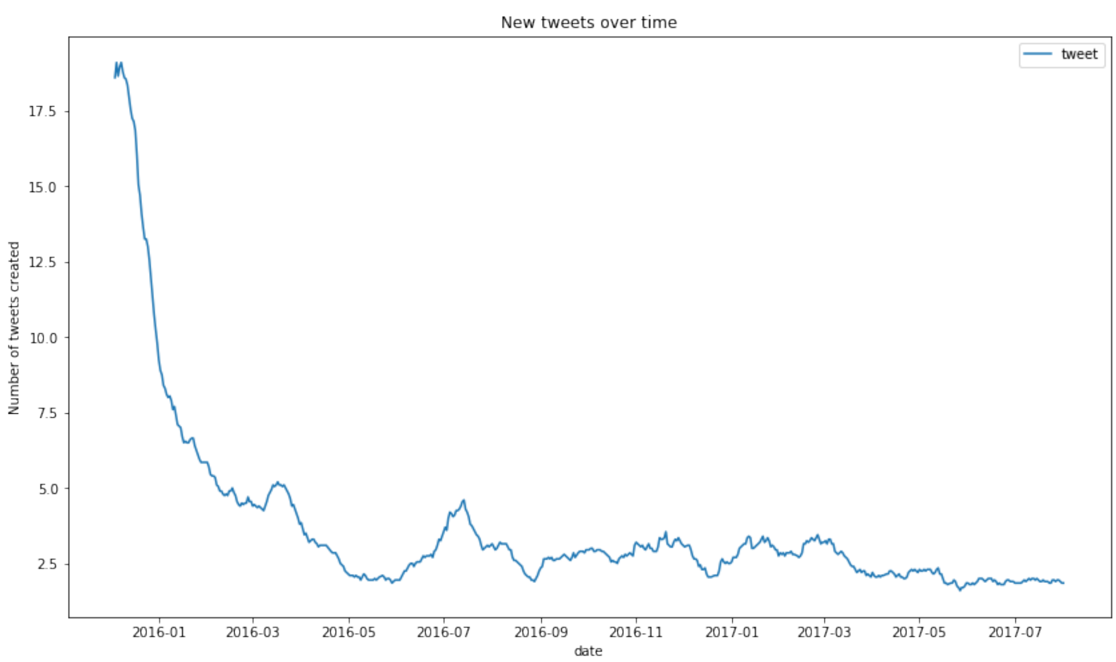
After analyzing the data, it is evident that the top 10 most common dog names in the WeRateDogs Twitter archive are "Charlie," "Oliver," "Lucy," "Cooper," "Penny," "Tucker," "Winston," "Sadie," "Daisy," and "Lola."

These popular dog names suggest that certain names are more favored by dog owners or are considered particularly endearing. Dog owners may choose these names for their pets due to their popularity, personal preferences, or simply because they find them cute and fitting for their furry friends.

1.5.3 Insight#3:

Tweets creation over time

In [84]:
```
df1_clean['date'] = df1_clean['timestamp'].dt.date
df1_clean['tweet'] = 1

# Create a new dataframe with 2 columns, gruped by date
df1 = df1_clean[['date', 'tweet']].groupby(['date']).sum() # alternatively .count() cou

# Use moving averages to smooth the line
df1['tweet'] = df1['tweet'].rolling(window=20).mean()

# Plot
df1.plot(figsize=(14, 8), title='New tweets over time')
plt.ylabel('Number of tweets created')
plt.show()
```



The number of tweets from the WeRateDogs account has shown a gradual decline over time. From the year 2015 to mid-2017, there was a significant increase in the number of tweets, reaching its peak around mid-2016. However, since mid-2017, there has been a steady decrease in the frequency of tweets.

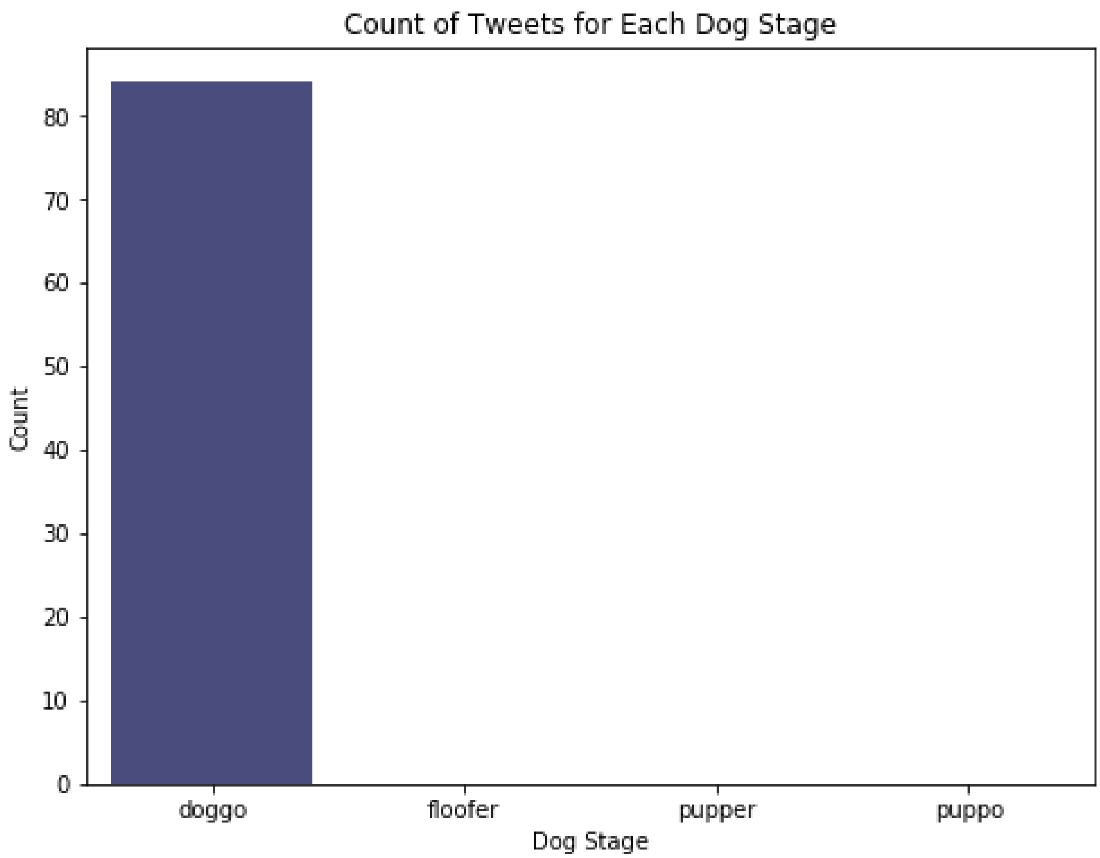
This declining trend may be attributed to several factors. One possible explanation is that the initial surge in tweets was driven by the account's growing popularity and novelty. As WeRateDogs gained a substantial following, the rate of tweet creation may have stabilized, and the account's content generation became more consistent.

1.5.4 Visualization

```
In [47]: print(df1_clean.columns)
```

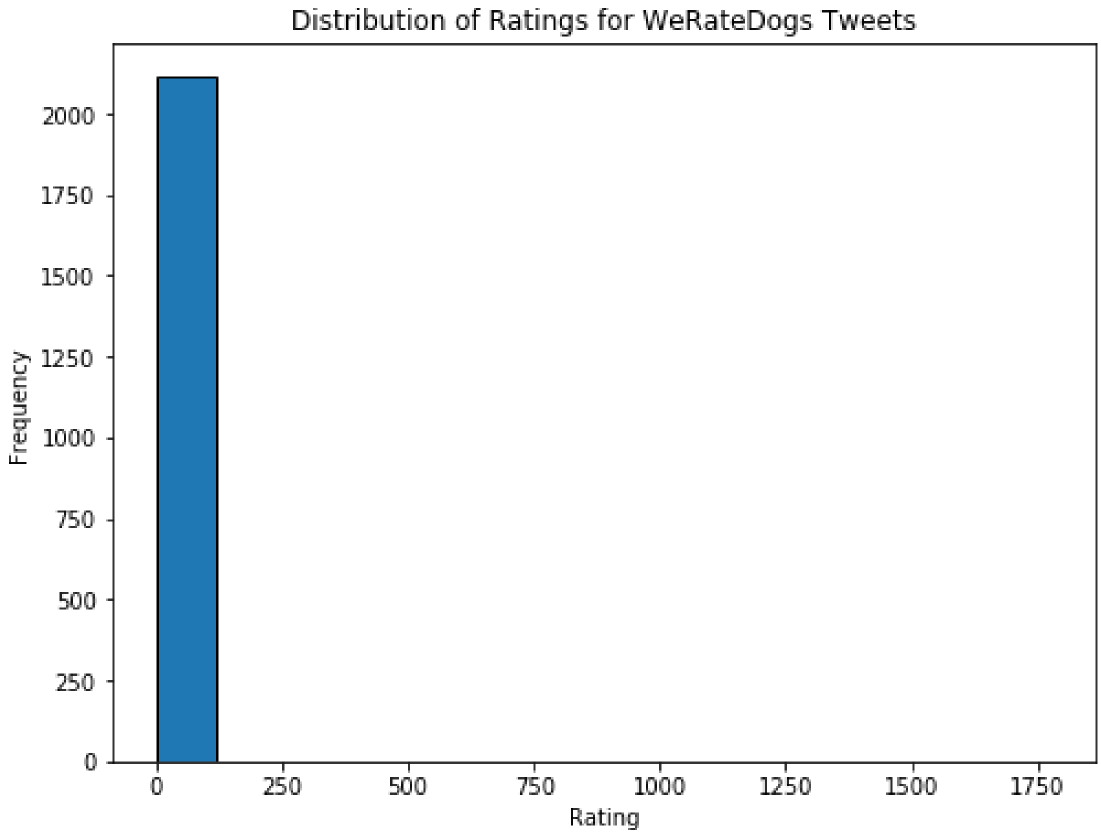
```
Index(['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id', 'timestamp',
       'source', 'text', 'retweeted_status_id', 'retweeted_status_user_id',
       'retweeted_status_timestamp', 'expanded_urls', 'name', 'doggo',
       'floofer', 'pupper', 'puppo', 'normalized_rating'],
      dtype='object')
```

```
In [49]: import matplotlib.pyplot as plt
         import seaborn as sns

         # Count the occurrences of each dog stage
         dog_stages_counts = df1_clean[['doggo', 'floofer', 'pupper', 'puppo']].apply(pd.Series.

         # Plot the bar chart
         plt.figure(figsize=(8, 6))
         sns.barplot(x=dog_stages_counts.columns, y=dog_stages_counts.iloc[1], palette='viridis'
         plt.xlabel('Dog Stage')
         plt.ylabel('Count')
         plt.title('Count of Tweets for Each Dog Stage')
         plt.show()
```



This bar chart will visually show the distribution of tweets among different dog stages, allowing us to see which stage is the most commonly mentioned in the tweets. It will help us understand the popularity of different dog stages in the WeRateDogs tweets.

```
In [50]: import matplotlib.pyplot as plt

         # Plot a histogram of ratings
         plt.figure(figsize=(8, 6))
         plt.hist(df1_clean['normalized_rating'], bins=15, edgecolor='black')
         plt.xlabel('Rating')
         plt.ylabel('Frequency')
         plt.title('Distribution of Ratings for WeRateDogs Tweets')
         plt.show()
```



This histogram will give us insights into the most common ratings given by WeRateDogs to the dogs featured in the tweets. We can observe the distribution of ratings and identify any patterns or trends in how the dogs are rated.