wrangle_act

December 9, 2022

1 Project: Wrangling and Analyze Data

This project is aimed to practice the skills of data wrangling using real-world data. Real-world data rarely comes clean, Data wrangling process consists of three parts: Gather, Assess and Clean. Real-world data rarely comes clean, so using Python and its libraries, we will gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. The dataset for this project is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. We will document our wrangling efforts in a Jupyter Notebook, plus showcase them through analyses and visualizations using Python (and its libraries).

1.1 Table of Contents:

- Data Gathering

Link to the Data Gathering'

- Assessing

Link to the Assessing Data'

- Cleaning

Link to the Cleaning Data'

- Storing

Link to the Storing Data'

- Analysis and Visualization

Link to the Analyzing and Visualizing Data'

Data Gathering In the cell below, gather **all** three pieces of data for this project and load them in the notebook. **Note:** the methods required to gather each data are different. 1. Directly download the WeRateDogs Twitter archive data (twitter_archive_enhanced.csv)

In [1]:
```
#Import statements for all packages
import pandas as pd
import numpy as np
import seaborn as sns
import requests
import tweepy
import json
from IPython.display import Image
import os
import re
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
matplotlib.style.use('ggplot')
import datetime
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
# script to upgrade certificate on this Workspace
!pip install --upgrade certifi
```

```
Requirement already up-to-date: certifi in /opt/conda/lib/python3.6/site-packages (2022.9.24)
```

In [3]:
```
# Load Data
twitter_archives = pd.read_csv('twitter-archive-enhanced.csv')

# Checking the firstfew values of the dataset
twitter_archives.head(2)
```

Out[3]:

	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
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	NaN	NaN	NaN
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you...	NaN	NaN	NaN

```
                                            expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/892420643...                13
1  https://twitter.com/dog_rates/status/892177421...                13

   rating_denominator     name doggo floofer pupper puppo
0                  10  Phineas  None    None   None  None
1                  10    Tilly  None    None   None  None
```

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

Image Predictions

```
In [4]: # Let's dowload the Image Predictions file programmatically:
        url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictic
        image_request = requests.get(url, allow_redirects=True)

        # Let's save the file
        open('image_predictions.tsv', 'wb').write(image_request.content)
```

```
Out[4]: 335079
```

```
In [5]: # Read TSV file
        image_predictions = pd.read_csv('image_predictions.tsv', sep = '\t')

        # Checking the firstfew values of the dataset
        image_predictions.head()
```

```
Out[5]:              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
```

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

Twitter API

```
In [6]: # Let's download file using Requests library via URL provided to Udacity Students
        url = 'https://video.udacity-data.com/topher/2018/November/5be5fb7d_tweet-json/tweet-jso
        response = requests.get(url)

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

```
In [7]: # Now, we can read downloaded txt file line by line into our Pandas DataFrame
        df_list = []
        with open('tweet-json.txt', 'r') as file:
            lines = file.readlines()
            for line in lines:
                parsed_json = json.loads(line)
                df_list.append({'tweet_id': parsed_json['id'],
                                'retweet_count': parsed_json['retweet_count'],
                                'favorite_count': parsed_json['favorite_count']})

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

        # Checking the firstfew values of the dataset
        tweet_json.head()
```

Out[7]:

	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

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.

1.1.1 Twitter

In [8]:
```
twitter_archives.head()
```

Out[8]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweeted_status_user_id	retweeted_status_timestamp	expanded_urls	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892420643...	13	10	Phineas	None	None	None	None
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892177421...	13	10	Tilly	None	None	None	None
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/891815181...	12	10	Archie	None	None	None	None
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/891689557...	13	10	Darla	None	None	None	None
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/891327558...	12	10	Franklin	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
```

```
In [9]: twitter_archives.source.value_counts()
```

```
Out[9]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>     2
        <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
        <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
        <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
        Name: source, dtype: int64
```

Quality:

Some columns have NaN values

Source info can not be read easily.

Tidiness:

Dog type is in four columns (doggo, floofer, pupper, puppo), which suppose to be in one column.

```
In [10]: # Get general info
         twitter_archives.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
doggo                         2356 non-null object
floofer                       2356 non-null object
pupper                        2356 non-null object
puppo                         2356 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

In [11]:
```
#filter out all the rows where retweeted_status_id is null
twitter_archives[- twitter_archives.retweeted_status_id.isnull()]
```

Out[11]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	\
19	888202515573088257	NaN	NaN	
32	886054160059072513	NaN	NaN	
36	885311592912609280	NaN	NaN	
68	879130579576475649	NaN	NaN	
73	878404777348136964	NaN	NaN	
74	878316110768087041	NaN	NaN	
78	877611172832227328	NaN	NaN	
91	874434818259525634	NaN	NaN	
95	873697596434513921	NaN	NaN	
97	873337748698140672	NaN	NaN	
101	872668790621863937	NaN	NaN	
109	871166179821445120	NaN	NaN	
118	869988702071779329	NaN	NaN	
124	868639477480148993	NaN	NaN	
130	867072653475098625	NaN	NaN	
132	866816280283807744	NaN	NaN	
137	866094527597207552	NaN	NaN	
146	863471782782697472	NaN	NaN	
155	861769973181624320	NaN	NaN	
159	860981674716409858	NaN	NaN	
160	860924035999428608	NaN	NaN	
165	860177593139703809	NaN	NaN	
171	858860390427611136	NaN	NaN	
180	857062103051644929	NaN	NaN	
182	856602993587888130	NaN	NaN	
185	856330835276025856	NaN	NaN	
194	855245323840757760	NaN	NaN	
195	855138241867124737	NaN	NaN	
204	852936405516943360	NaN	NaN	
211	851953902622658560	NaN	NaN	
...	...	...	...	
784	775096608509886464	NaN	NaN	
794	773336787167145985	NaN	NaN	
800	772615324260794368	NaN	NaN	
811	771171053431250945	NaN	NaN	
815	771004394259247104	NaN	NaN	
818	770743923962707968	NaN	NaN	
822	770093767776997377	NaN	NaN	
826	769335591808995329	NaN	NaN	
829	768909767477751808	NaN	NaN	
833	768554158521745409	NaN	NaN	
841	766864461642756096	NaN	NaN	
847	766078092750233600	NaN	NaN	
860	763167063695355904	NaN	NaN	

868	761750502866649088	NaN	NaN
872	761371037149827077	NaN	NaN
885	760153949710192640	NaN	NaN
890	759566828574212096	NaN	NaN
895	759159934323924993	NaN	NaN
908	757729163776290825	NaN	NaN
911	757597904299253760	NaN	NaN
926	754874841593970688	NaN	NaN
937	753298634498793472	NaN	NaN
943	752701944171524096	NaN	NaN
949	752309394570878976	NaN	NaN
1012	747242308580548608	NaN	NaN
1023	746521445350707200	NaN	NaN
1043	743835915802583040	NaN	NaN
1242	711998809858043904	NaN	NaN
2259	667550904950915073	NaN	NaN
2260	667550882905632768	NaN	NaN

	timestamp
19	2017-07-21 01:02:36 +0000
32	2017-07-15 02:45:48 +0000
36	2017-07-13 01:35:06 +0000
68	2017-06-26 00:13:58 +0000
73	2017-06-24 00:09:53 +0000
74	2017-06-23 18:17:33 +0000
78	2017-06-21 19:36:23 +0000
91	2017-06-13 01:14:41 +0000
95	2017-06-11 00:25:14 +0000
97	2017-06-10 00:35:19 +0000
101	2017-06-08 04:17:07 +0000
109	2017-06-04 00:46:17 +0000
118	2017-05-31 18:47:24 +0000
124	2017-05-28 01:26:04 +0000
130	2017-05-23 17:40:04 +0000
132	2017-05-23 00:41:20 +0000
137	2017-05-21 00:53:21 +0000
146	2017-05-13 19:11:30 +0000
155	2017-05-09 02:29:07 +0000
159	2017-05-06 22:16:42 +0000
160	2017-05-06 18:27:40 +0000
165	2017-05-04 17:01:34 +0000
171	2017-05-01 01:47:28 +0000
180	2017-04-26 02:41:43 +0000
182	2017-04-24 20:17:23 +0000
185	2017-04-24 02:15:55 +0000
194	2017-04-21 02:22:29 +0000
195	2017-04-20 19:16:59 +0000
204	2017-04-14 17:27:40 +0000

```
211   2017-04-12 00:23:33 +0000
...                         ...
784   2016-09-11 22:20:06 +0000
794   2016-09-07 01:47:12 +0000
800   2016-09-05 02:00:22 +0000
811   2016-09-01 02:21:21 +0000
815   2016-08-31 15:19:06 +0000
818   2016-08-30 22:04:05 +0000
822   2016-08-29 03:00:36 +0000
826   2016-08-27 00:47:53 +0000
829   2016-08-25 20:35:48 +0000
833   2016-08-24 21:02:45 +0000
841   2016-08-20 05:08:29 +0000
847   2016-08-18 01:03:45 +0000
860   2016-08-10 00:16:21 +0000
868   2016-08-06 02:27:27 +0000
872   2016-08-05 01:19:35 +0000
885   2016-08-01 16:43:19 +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
911   2016-07-25 15:26:30 +0000
926   2016-07-18 03:06:01 +0000
937   2016-07-13 18:42:44 +0000
943   2016-07-12 03:11:42 +0000
949   2016-07-11 01:11:51 +0000
1012  2016-06-27 01:37:04 +0000
1023  2016-06-25 01:52:36 +0000
1043  2016-06-17 16:01:16 +0000
1242  2016-03-21 19:31:59 +0000
2259  2015-11-20 03:51:52 +0000
2260  2015-11-20 03:51:47 +0000

                                                  source  \
19    <a href="http://twitter.com/download/iphone" r...
32    <a href="http://twitter.com/download/iphone" r...
36    <a href="http://twitter.com/download/iphone" r...
68    <a href="http://twitter.com/download/iphone" r...
73    <a href="http://twitter.com/download/iphone" r...
74    <a href="http://twitter.com/download/iphone" r...
78    <a href="http://twitter.com/download/iphone" r...
91    <a href="http://twitter.com/download/iphone" r...
95    <a href="http://twitter.com/download/iphone" r...
97    <a href="http://twitter.com/download/iphone" r...
101   <a href="http://twitter.com/download/iphone" r...
109   <a href="http://twitter.com/download/iphone" r...
118   <a href="http://twitter.com/download/iphone" r...
124   <a href="http://twitter.com/download/iphone" r...
```

```
130   <a href="http://twitter.com/download/iphone" r...
132   <a href="http://twitter.com/download/iphone" r...
137   <a href="http://twitter.com/download/iphone" r...
146   <a href="http://twitter.com/download/iphone" r...
155   <a href="http://twitter.com/download/iphone" r...
159   <a href="http://twitter.com/download/iphone" r...
160   <a href="http://twitter.com/download/iphone" r...
165   <a href="http://twitter.com/download/iphone" r...
171   <a href="http://twitter.com/download/iphone" r...
180   <a href="http://twitter.com/download/iphone" r...
182   <a href="http://twitter.com/download/iphone" r...
185   <a href="http://twitter.com/download/iphone" r...
194   <a href="http://twitter.com/download/iphone" r...
195   <a href="http://twitter.com/download/iphone" r...
204   <a href="http://twitter.com/download/iphone" r...
211   <a href="http://twitter.com/download/iphone" r...
...                                                  ...
784   <a href="http://twitter.com/download/iphone" r...
794   <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...
815   <a href="http://twitter.com/download/iphone" r...
818   <a href="http://twitter.com/download/iphone" r...
822   <a href="http://twitter.com/download/iphone" r...
826   <a href="http://twitter.com/download/iphone" r...
829   <a href="http://twitter.com/download/iphone" r...
833   <a href="http://twitter.com/download/iphone" r...
841   <a href="http://twitter.com/download/iphone" r...
847   <a href="http://twitter.com/download/iphone" r...
860   <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...
885   <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...
911   <a href="http://twitter.com/download/iphone" r...
926   <a href="http://twitter.com/download/iphone" r...
937   <a href="http://twitter.com/download/iphone" r...
943   <a href="http://twitter.com/download/iphone" r...
949   <a href="http://twitter.com/download/iphone" r...
1012  <a href="http://twitter.com/download/iphone" r...
1023  <a href="http://twitter.com/download/iphone" r...
1043  <a href="http://twitter.com/download/iphone" r...
1242  <a href="http://twitter.com/download/iphone" r...
2259  <a href="http://twitter.com" rel="nofollow">Tw...
2260  <a href="http://twitter.com" rel="nofollow">Tw...
```

	text	retweeted_status_id	\
19	RT @dog_rates: This is Canela. She attempted s...	8.874740e+17	
32	RT @Athletics: 12/10 #BATP https://t.co/WxwJmv...	8.860537e+17	
36	RT @dog_rates: This is Lilly. She just paralle...	8.305833e+17	
68	RT @dog_rates: This is Emmy. She was adopted t...	8.780576e+17	
73	RT @dog_rates: Meet Shadow. In an attempt to r...	8.782815e+17	
74	RT @dog_rates: Meet Terrance. He's being yelle...	6.690004e+17	
78	RT @rachel2195: @dog_rates the boyfriend and h...	8.768508e+17	
91	RT @dog_rates: This is Coco. At first I though...	8.663350e+17	
95	RT @dog_rates: This is Walter. He won't start ...	8.688804e+17	
97	RT @dog_rates: This is Sierra. She's one preci...	8.732138e+17	
101	RT @loganamnosis: Penelope here is doing me qu...	8.726576e+17	
109	RT @dog_rates: This is Dawn. She's just checki...	8.410770e+17	
118	RT @dog_rates: We only rate dogs. This is quit...	8.591970e+17	
124	RT @dog_rates: Say hello to Cooper. His expres...	8.685523e+17	
130	RT @rachaeleasler: these @dog_rates hats are 1...	8.650134e+17	
132	RT @dog_rates: This is Jamesy. He gives a kiss...	8.664507e+17	
137	RT @dog_rates: Here's a pupper before and afte...	8.378202e+17	
146	RT @dog_rates: Say hello to Quinn. She's quite...	8.630625e+17	
155	RT @dog_rates: "Good afternoon class today we'...	8.066291e+17	
159	RT @dog_rates: Meet Lorenzo. He's an avid nift...	8.605638e+17	
160	RT @tallylott: h*ckin adorable promposal. 13/1...	8.609145e+17	
165	RT @dog_rates: Ohboyohboyohboyohboyohboyohboyo...	7.616730e+17	
171	RT @dog_rates: Meet Winston. He knows he's a l...	8.395493e+17	
180	RT @AaronChewning: First time wearing my @dog_...	8.570611e+17	
182	RT @dog_rates: This is Luna. It's her first ti...	8.447048e+17	
185	RT @Jenna_Marbles: @dog_rates Thanks for ratin...	8.563302e+17	
194	RT @dog_rates: Meet George. He looks slightly ...	8.421635e+17	
195	RT @frasercampbell_: oh my... what's that... b...	8.551225e+17	
204	RT @dog_rates: I usually only share these on F...	8.316501e+17	
211	RT @dog_rates: This is Astrid. She's a guide d...	8.293743e+17	
...	...	...	
784	RT @dog_rates: After so many requests, this is...	7.403732e+17	
794	RT @dog_rates: Meet Fizz. She thinks love is a...	7.713808e+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	
815	RT @katieornah: @dog_rates learning a lot at c...	7.710021e+17	
818	RT @dog_rates: Here's a doggo blowing bubbles...	7.392382e+17	
822	RT @dog_rates: This is just downright precious...	7.410673e+17	
826	RT @dog_rates: Ever seen a dog pet another dog...	7.069045e+17	
829	RT @dog_rates: When it's Janet from accounting...	7.001438e+17	
833	RT @dog_rates: This is Nollie. She's waving at...	7.399792e+17	
841	RT @dog_rates: We only rate dogs... this is a ...	7.599238e+17	
847	RT @dog_rates: This is Colby. He's currently r...	7.258423e+17	
860	RT @dog_rates: Meet Eve. She's a raging alcoho...	6.732953e+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	
885	RT @hownottodraw: The story/person behind @dog...	7.601538e+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
911   RT @jon_hill987: @dog_rates There is a cunning...        7.575971e+17
926   RT @dog_rates: This is Rubio. He has too much ...        6.791584e+17
937   RT @dog_rates: This is Carly. She's actually 2...        6.815232e+17
943   RT @dog_rates: HEY PUP WHAT'S THE PART OF THE ...        6.835159e+17
949   RT @dog_rates: Everyone needs to watch this. 1...        6.753544e+17
1012  RT @dog_rates: This pupper killed this great w...        7.047611e+17
1023  RT @dog_rates: This is Shaggy. He knows exactl...        6.678667e+17
1043  RT @dog_rates: Extremely intelligent dog here...         6.671383e+17
1242  RT @twitter: @dog_rates Awesome Tweet! 12/10. ...        7.119983e+17
2259  RT @dogratingrating: Exceptional talent. Origi...        6.675487e+17
2260  RT @dogratingrating: Unoriginal idea. Blatant ...        6.675484e+17

      retweeted_status_user_id retweeted_status_timestamp  \
19                4.196984e+09  2017-07-19 00:47:34 +0000
32                1.960740e+07  2017-07-15 02:44:07 +0000
36                4.196984e+09  2017-02-12 01:04:29 +0000
68                4.196984e+09  2017-06-23 01:10:23 +0000
73                4.196984e+09  2017-06-23 16:00:04 +0000
74                4.196984e+09  2015-11-24 03:51:38 +0000
78                5.128045e+08  2017-06-19 17:14:49 +0000
91                4.196984e+09  2017-05-21 16:48:45 +0000
95                4.196984e+09  2017-05-28 17:23:24 +0000
97                4.196984e+09  2017-06-09 16:22:42 +0000
101               1.547674e+08  2017-06-08 03:32:35 +0000
109               4.196984e+09  2017-03-13 00:02:39 +0000
118               4.196984e+09  2017-05-02 00:04:57 +0000
124               4.196984e+09  2017-05-27 19:39:34 +0000
130               7.874618e+17  2017-05-18 01:17:25 +0000
132               4.196984e+09  2017-05-22 00:28:40 +0000
137               4.196984e+09  2017-03-04 00:21:08 +0000
146               4.196984e+09  2017-05-12 16:05:02 +0000
155               4.196984e+09  2016-12-07 22:38:52 +0000
159               4.196984e+09  2017-05-05 18:36:06 +0000
160               3.638908e+08  2017-05-06 17:49:42 +0000
165               4.196984e+09  2016-08-05 21:19:27 +0000
171               4.196984e+09  2017-03-08 18:52:12 +0000
180               5.870972e+07  2017-04-26 02:37:47 +0000
182               4.196984e+09  2017-03-23 00:18:10 +0000
185               6.669901e+07  2017-04-24 02:13:14 +0000
194               4.196984e+09  2017-03-16 00:00:07 +0000
195               7.475543e+17  2017-04-20 18:14:33 +0000
204               4.196984e+09  2017-02-14 23:43:18 +0000
211               4.196984e+09  2017-02-08 17:00:26 +0000
...                        ...                        ...
784               4.196984e+09  2016-06-08 02:41:38 +0000
```

```
794              4.196984e+09  2016-09-01 16:14:48 +0000
800              4.196984e+09  2016-08-15 16:22:20 +0000
811              4.196984e+09  2015-12-06 01:56:44 +0000
815              1.732729e+09  2016-08-31 15:10:07 +0000
818              4.196984e+09  2016-06-04 23:31:25 +0000
822              4.196984e+09  2016-06-10 00:39:48 +0000
826              4.196984e+09  2016-03-07 18:09:06 +0000
829              4.196984e+09  2016-02-18 02:24:13 +0000
833              4.196984e+09  2016-06-07 00:36:02 +0000
841              4.196984e+09  2016-08-01 01:28:46 +0000
847              4.196984e+09  2016-04-29 00:21:01 +0000
860              4.196984e+09  2015-12-06 00:17:55 +0000
868              4.196984e+09  2016-01-08 05:00:14 +0000
872              4.196984e+09  2016-03-20 23:23:54 +0000
885              1.950368e+08  2016-08-01 16:42:51 +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
911              2.804798e+08  2016-07-25 15:23:28 +0000
926              4.196984e+09  2015-12-22 04:35:49 +0000
937              4.196984e+09  2015-12-28 17:12:42 +0000
943              4.196984e+09  2016-01-03 05:11:12 +0000
949              4.196984e+09  2015-12-11 16:40:19 +0000
1012             4.196984e+09  2016-03-01 20:11:59 +0000
1023             4.196984e+09  2015-11-21 00:46:50 +0000
1043             4.196984e+09  2015-11-19 00:32:12 +0000
1242             7.832140e+05  2016-03-21 19:29:52 +0000
2259             4.296832e+09  2015-11-20 03:43:06 +0000
2260             4.296832e+09  2015-11-20 03:41:59 +0000

                                          expanded_urls  rating_numerator  \
19    https://twitter.com/dog_rates/status/887473957...                13
32    https://twitter.com/dog_rates/status/886053434...                12
36    https://twitter.com/dog_rates/status/830583320...                13
68    https://twitter.com/dog_rates/status/878057613...                14
73    https://www.gofundme.com/3yd6y1c,https://twitt...                13
74    https://twitter.com/dog_rates/status/669000397...                11
78    https://twitter.com/rachel2195/status/87685077...                14
91    https://twitter.com/dog_rates/status/866334964...                12
95    https://twitter.com/dog_rates/status/868880397...                14
97    https://www.gofundme.com/help-my-baby-sierra-g...                12
101   https://twitter.com/loganamnosis/status/872657...                14
109   https://twitter.com/dog_rates/status/841077006...                12
118   https://twitter.com/dog_rates/status/859196978...                12
124   https://www.gofundme.com/3ti3nps,https://twitt...                12
130   https://twitter.com/rachaeleasler/status/86501...                13
132   https://twitter.com/dog_rates/status/866450705...                13
137   https://twitter.com/dog_rates/status/837820167...                12
```

```
146   https://www.gofundme.com/helpquinny,https://tw...                13
155   https://twitter.com/dog_rates/status/806629075...                13
159   https://www.gofundme.com/help-lorenzo-beat-can...                13
160   https://twitter.com/tallylott/status/860914485...                13
165   https://twitter.com/dog_rates/status/761672994...                10
171   https://twitter.com/dog_rates/status/839549326...                12
180   https://twitter.com/AaronChewning/status/85706...                13
182   https://twitter.com/dog_rates/status/844704788...                13
185                                                 NaN                14
194   https://twitter.com/dog_rates/status/842163532...                12
195   https://twitter.com/frasercampbell_/status/855...                14
204   http://www.gofundme.com/bluethewhitehusky,http...                13
211   https://twitter.com/dog_rates/status/829374341...                13
...                                                 ...               ...
784   https://twitter.com/dog_rates/status/740373189...                 9
794   https://twitter.com/dog_rates/status/771380798...                11
800   https://twitter.com/dog_rates/status/765222098...                10
811   https://twitter.com/dog_rates/status/673320132...                11
815   https://twitter.com/katieornah/status/77100213...                12
818   https://twitter.com/dog_rates/status/739238157...                13
822   https://twitter.com/dog_rates/status/741067306...                12
826   https://vine.co/v/iXQAm5Lrgrh,https://vine.co/...                13
829   https://twitter.com/dog_rates/status/700143752...                10
833   https://twitter.com/dog_rates/status/739979191...                12
841   https://twitter.com/dog_rates/status/759923798...                10
847   https://twitter.com/dog_rates/status/725842289...                12
860   https://twitter.com/dog_rates/status/673295268...                 8
868   https://twitter.com/dog_rates/status/685325112...                10
872   https://twitter.com/dog_rates/status/711694788...                13
885   https://weratedogs.com/pages/about-us,https://...                11
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
911   https://twitter.com/jon_hill987/status/7575971...                11
926   https://twitter.com/dog_rates/status/679158373...                11
937   https://twitter.com/dog_rates/status/681523177...                12
943   https://vine.co/v/ibvnzrauFuV,https://vine.co/...                11
949   https://twitter.com/dog_rates/status/675354435...                13
1012  https://twitter.com/dog_rates/status/704761120...                13
1023  https://twitter.com/dog_rates/status/667866724...                10
1043  https://twitter.com/dog_rates/status/667138269...                10
1242  https://twitter.com/twitter/status/71199827977...                12
2259  https://twitter.com/dogratingrating/status/667...                12
2260  https://twitter.com/dogratingrating/status/667...                 5

      rating_denominator      name  doggo floofer  pupper puppo
19                    10    Canela   None    None    None  None
32                    10      None   None    None    None  None
```

36	10	Lilly	None	None	None	None
68	10	Emmy	None	None	None	None
73	10	Shadow	None	None	None	None
74	10	Terrance	None	None	None	None
78	10	None	None	None	pupper	None
91	10	Coco	None	None	None	None
95	10	Walter	None	None	None	None
97	10	Sierra	None	None	pupper	None
101	10	None	None	None	None	None
109	10	Dawn	None	None	None	None
118	10	quite	None	None	None	None
124	10	Cooper	None	None	None	None
130	10	None	None	None	None	None
132	10	Jamesy	None	None	pupper	None
137	10	None	None	None	pupper	None
146	10	Quinn	None	None	None	None
155	10	None	None	None	None	None
159	10	Lorenzo	None	None	None	None
160	10	None	None	None	None	None
165	10	None	None	None	None	None
171	10	Winston	None	None	None	None
180	10	None	None	None	None	None
182	10	Luna	None	None	None	None
185	10	None	None	None	None	None
194	10	George	None	None	None	None
195	10	None	None	None	None	None
204	10	None	None	None	None	None
211	10	Astrid	doggo	None	None	None
...	...	...	...	...	...	...
784	11	None	None	None	None	None
794	10	Fizz	None	None	None	None
800	10	Gromit	None	None	None	None
811	10	Frankie	None	None	None	None
815	10	None	None	None	pupper	None
818	10	None	doggo	None	None	None
822	10	just	doggo	None	pupper	None
826	10	None	None	None	None	None
829	10	None	None	None	pupper	None
833	10	Nollie	None	None	None	None
841	10	None	None	None	None	None
847	10	Colby	None	None	None	None
860	10	Eve	None	None	pupper	None
868	10	None	None	None	None	None
872	10	None	None	None	None	None
885	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

```
911                  10      None    None    None  pupper  None
926                  10     Rubio    None    None    None  None
937                  10     Carly    None    None    None  None
943                  10      None    None    None    None  None
949                  10      None    None    None    None  None
1012                 10      None    None    None  pupper  None
1023                 10    Shaggy    None    None    None  None
1043                 10      None    None    None    None  None
1242                 10      None    None    None    None  None
2259                 10      None    None    None    None  None
2260                 10      None    None    None    None  None

[181 rows x 17 columns]
```

Quality:

- These aren't really valid as there are duplicated tweets and we don't want retweets.
- in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id should be integers instead of float.
- retweeted_status_timestamp, timestamp should be datetime instead of object which is string.
- We may change this columns type: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id and tweet_id to string because We don't want any operations on them.

```
In [12]: # Check for duplicates
         twitter_archives.duplicated().sum()

Out[12]: 0

In [13]: # Check the column labels of the DataFrame
         twitter_archives.columns

Out[13]: 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', 'rating_numerator',
                'rating_denominator', 'name', 'doggo', 'floofer', 'pupper', 'puppo'],
               dtype='object')

In [14]: # Return a Series containing counts of Name rows in the DataFrame.
         twitter_archives['name'].value_counts()

Out[14]: None       745
         a           55
         Charlie     12
         Cooper      11
         Oliver      11
```

```
Lucy        11
Lola        10
Tucker      10
Penny       10
Bo           9
Winston      9
the          8
Sadie        8
Toby         7
an           7
Bailey       7
Daisy        7
Buddy        7
Bella        6
Koda         6
Oscar        6
Stanley      6
Rusty        6
Dave         6
Jack         6
Milo         6
Leo          6
Jax          6
Scout        6
Chester      5
            ...
Iggy         1
Snicku       1
Moofasa      1
Carper       1
Mona         1
Willow       1
Remy         1
Deacon       1
Grizzie      1
Glacier      1
Teddy        1
Sailer       1
Laela        1
Corey        1
Sailor       1
Caryl        1
Jordy        1
Wiggles      1
Chloe        1
Clarkus      1
Lilah        1
Meera        1
```

```
Timmy        1
Monkey       1
Durg         1
Coopson      1
Tebow        1
Katie        1
Tuco         1
Brudge       1
Name: name, Length: 957, dtype: int64
```

Quality:

There are invalid names like a, an, etc., which is less than three characters.

There are duplicated names

```
In [15]: # Return a Series containing counts of rating_numerator rows in the DataFrame.
         twitter_archives.rating_numerator.value_counts()
```

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

```
182       1
143       1
666       1
960       1
1776      1
17        1
27        1
45        1
99        1
121       1
204       1
Name: rating_numerator, dtype: int64
```

Observation Abnormal values in rating_numerator, e.g., 1776, 960, 666, 204, 165, etc., is inappropriate

```
In [16]: # Return a Series containing counts of rating_denominator rows in the DataFrame.

         twitter_archives.rating_denominator.value_counts()
```

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

Observation Abnormal values in rating_denominator, e.g., 170, 150, 130, etc, because these ratings almost always have a denominator of 10.

1.1.2 Image Predictions

```
In [17]: image_predictions
```

```
Out[17]:              tweet_id                                          jpg_url  \
         0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
```

1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg
5	666050758794694657	https://pbs.twimg.com/media/CT5Jof1WUAEuVxN.jpg
6	666051853826850816	https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg
7	666055525042405380	https://pbs.twimg.com/media/CT5N9tpXIAAifs1.jpg
8	666057090499244032	https://pbs.twimg.com/media/CT5PY9OWoAAQGLo.jpg
9	666058600524156928	https://pbs.twimg.com/media/CT5Qw94XAAA_2dP.jpg
10	666063827256086533	https://pbs.twimg.com/media/CT5Vg_wXIAAXfnj.jpg
11	666071193221509120	https://pbs.twimg.com/media/CT5cN_3WEAAlOoZ.jpg
12	666073100786774016	https://pbs.twimg.com/media/CT5d9DZXAAALcwe.jpg
13	666082916733198337	https://pbs.twimg.com/media/CT5m4VGWEAAtKc8.jpg
14	666094000022159362	https://pbs.twimg.com/media/CT5w9gUW4AAsBNN.jpg
15	666099513787052032	https://pbs.twimg.com/media/CT51-JJUEAA6hV8.jpg
16	666102155909144576	https://pbs.twimg.com/media/CT54YGiWUAEZnoK.jpg
17	666104133288665088	https://pbs.twimg.com/media/CT56LSZWoAAlJj2.jpg
18	666268910803644416	https://pbs.twimg.com/media/CT8QCd1WEAADXws.jpg
19	666273097616637952	https://pbs.twimg.com/media/CT8T1mtUwAA3aqm.jpg
20	666287406224695296	https://pbs.twimg.com/media/CT8g3BpUEAAuFjg.jpg
21	666293911632134144	https://pbs.twimg.com/media/CT8mx7KW4AEQu8N.jpg
22	666337882303524864	https://pbs.twimg.com/media/CT9OwFIWEAMuRje.jpg
23	666345417576210432	https://pbs.twimg.com/media/CT9Vn7PWoAA_ZCM.jpg
24	666353288456101888	https://pbs.twimg.com/media/CT9cx0tUEAAhNN_.jpg
25	666362758909284353	https://pbs.twimg.com/media/CT9lXGsUcAAyUFt.jpg
26	666373753744588802	https://pbs.twimg.com/media/CT9vZEYWUAAlZO5.jpg
27	666396247373291520	https://pbs.twimg.com/media/CT-D2ZHWIAA3gK1.jpg
28	666407126856765440	https://pbs.twimg.com/media/CT-NvwmW4AAugGZ.jpg
29	666411507551481857	https://pbs.twimg.com/media/CT-RugiWIAELEaq.jpg
...	...	...
2045	886366144734445568	https://pbs.twimg.com/media/DEOBTnQUwAApKEH.jpg
2046	886680336477933568	https://pbs.twimg.com/media/DE4fEDzWAAAyHMM.jpg
2047	886736880519319552	https://pbs.twimg.com/media/DE5Se8FXcAAJFx4.jpg
2048	886983233522544640	https://pbs.twimg.com/media/DE8yicJWOAAAvBJ.jpg
2049	887101392804085760	https://pbs.twimg.com/media/DE-eAq6UwAA-jaE.jpg
2050	887343217045368832	https://pbs.twimg.com/ext_tw_video_thumb/88734...
2051	887473957103951883	https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg
2052	887517139158093824	https://pbs.twimg.com/ext_tw_video_thumb/88751...
2053	887705289381826560	https://pbs.twimg.com/media/DFHDQBbXgAEqY7t.jpg
2054	888078434458587136	https://pbs.twimg.com/media/DFMWn56WsAAkA7B.jpg
2055	888202515573088257	https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg
2056	888554962724278272	https://pbs.twimg.com/media/DFTH_O-UQAACu2O.jpg
2057	888804989199671297	https://pbs.twimg.com/media/DFWra-3VYAA2piG.jpg
2058	888917238123831296	https://pbs.twimg.com/media/DFYRgsOUQAARGhO.jpg
2059	889278841981685760	https://pbs.twimg.com/ext_tw_video_thumb/88927...
2060	889531135344209921	https://pbs.twimg.com/media/DFg_2PVWOAEHN3p.jpg
2061	889638837579907072	https://pbs.twimg.com/media/DFihzFfXsAYGDPR.jpg
2062	889665388333682689	https://pbs.twimg.com/media/DFi579UWsAAatzw.jpg

```
2063  889880896479866881    https://pbs.twimg.com/media/DFl99B1WsAITKsg.jpg
2064  890066608113172480    https://pbs.twimg.com/media/DFnwSY4WAAAMliS.jpg
2065  890240255349198849    https://pbs.twimg.com/media/DFrEyVuW0AAO3t9.jpg
2066  890609185150312448    https://pbs.twimg.com/media/DFwUU__XcAEpyXI.jpg
2067  890729181411237888    https://pbs.twimg.com/media/DFyBahAVwAAhUTd.jpg
2068  890971913173991426    https://pbs.twimg.com/media/DF1eOmZXUAALUcq.jpg
2069  891087950875897856    https://pbs.twimg.com/media/DF3HwyEWsAABqE6.jpg
2070  891327558926688256    https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg
2071  891689557279858688    https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg
2072  891815181378084864    https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg
2073  892177421306343426    https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg
2074  892420643555336193    https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg

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

```
2048        2                   Chihuahua  0.793469    True
2049        1                     Samoyed  0.733942    True
2050        1             Mexican_hairless  0.330741    True
2051        2                    Pembroke  0.809197    True
2052        1                   limousine  0.130432   False
2053        1                      basset  0.821664    True
2054        1               French_bulldog  0.995026    True
2055        2                    Pembroke  0.809197    True
2056        3               Siberian_husky  0.700377    True
2057        1             golden_retriever  0.469760    True
2058        1             golden_retriever  0.714719    True
2059        1                     whippet  0.626152    True
2060        1             golden_retriever  0.953442    True
2061        1               French_bulldog  0.991650    True
2062        1                    Pembroke  0.966327    True
2063        1               French_bulldog  0.377417    True
2064        1                     Samoyed  0.957979    True
2065        1                    Pembroke  0.511319    True
2066        1                Irish_terrier  0.487574    True
2067        2                  Pomeranian  0.566142    True
2068        1                 Appenzeller  0.341703    True
2069        1     Chesapeake_Bay_retriever  0.425595    True
2070        2                      basset  0.555712    True
2071        1                  paper_towel  0.170278   False
2072        1                    Chihuahua  0.716012    True
2073        1                    Chihuahua  0.323581    True
2074        1                       orange  0.097049   False

                         p2   p2_conf  p2_dog                          p3  \
0                   collie  0.156665    True           Shetland_sheepdog
1        miniature_pinscher  0.074192    True         Rhodesian_ridgeback
2                 malinois  0.138584    True                   bloodhound
3                  redbone  0.360687    True          miniature_pinscher
4               Rottweiler  0.243682    True                    Doberman
5          English_springer  0.263788    True  Greater_Swiss_Mountain_dog
6                mud_turtle  0.045885   False                     terrapin
7           Tibetan_mastiff  0.058279    True                     fur_coat
8           shopping_basket  0.014594   False           golden_retriever
9                  komondor  0.192305    True  soft-coated_wheaten_terrier
10          Tibetan_mastiff  0.093718    True          Labrador_retriever
11        Yorkshire_terrier  0.174201    True                    Pekinese
12         English_foxhound  0.175382    True                Ibizan_hound
13             bull_mastiff  0.404722    True               French_bulldog
14         German_shepherd  0.078260    True                     malinois
15                 Shih-Tzu  0.166192    True               Dandie_Dinmont
16            Newfoundland  0.149842    True                       borzoi
17                    cock  0.033919   False                    partridge
18                    desk  0.085547   False                     bookcase
```

```
19                toy_terrier  0.111884    True                    basenji
20                toy_poodle  0.063064    True            miniature_poodle
21                     otter  0.015250   False              great_grey_owl
22               Newfoundland  0.278407    True                 groenendael
23    Chesapeake_Bay_retriever  0.054787    True          Labrador_retriever
24              Siberian_husky  0.147655    True                  Eskimo_dog
25                       skunk  0.002402   False                     hamster
26               Afghan_hound  0.259551    True                      briard
27                toy_terrier  0.009397    True                    papillon
28                 bloodhound  0.244220    True       flat-coated_retriever
29                 barracouta  0.271485   False                         gar
...                       ...       ...     ...                         ...
2045                Chihuahua  0.000361    True                Boston_bull
2046               sports_car  0.139952   False                  car_wheel
2047           Great_Pyrenees  0.186136    True             Dandie_Dinmont
2048              toy_terrier  0.143528    True                  can_opener
2049               Eskimo_dog  0.035029    True   Staffordshire_bullterrier
2050                 sea_lion  0.275645   False                  Weimaraner
2051      Rhodesian_ridgeback  0.054950    True                      beagle
2052                tow_truck  0.029175   False               shopping_cart
2053                  redbone  0.087582    True                  Weimaraner
2054                      pug  0.000932    True                bull_mastiff
2055      Rhodesian_ridgeback  0.054950    True                      beagle
2056                Eskimo_dog  0.166511    True                   malamute
2057       Labrador_retriever  0.184172    True             English_setter
2058          Tibetan_mastiff  0.120184    True          Labrador_retriever
2059                   borzoi  0.194742    True                      Saluki
2060       Labrador_retriever  0.013834    True                     redbone
2061                    boxer  0.002129    True   Staffordshire_bullterrier
2062                 Cardigan  0.027356    True                     basenji
2063       Labrador_retriever  0.151317    True                      muzzle
2064                Pomeranian  0.013884    True                        chow
2065                 Cardigan  0.451038    True                   Chihuahua
2066             Irish_setter  0.193054    True    Chesapeake_Bay_retriever
2067               Eskimo_dog  0.178406    True                    Pembroke
2068            Border_collie  0.199287    True                   ice_lolly
2069            Irish_terrier  0.116317    True             Indian_elephant
2070         English_springer  0.225770    True  German_short-haired_pointer
2071       Labrador_retriever  0.168086    True                      spatula
2072                 malamute  0.078253    True                       kelpie
2073                 Pekinese  0.090647    True                    papillon
2074                    bagel  0.085851   False                       banana
```

```
      p3_conf  p3_dog
0    0.061428    True
1    0.072010    True
2    0.116197    True
3    0.222752    True
```

```
4     0.154629    True
5     0.016199    True
6     0.017885   False
7     0.054449   False
8     0.007959    True
9     0.082086    True
10    0.072427    True
11    0.109454    True
12    0.097471    True
13    0.048960    True
14    0.075628    True
15    0.089688    True
16    0.133649    True
17    0.000052   False
18    0.079480   False
19    0.111152    True
20    0.025581    True
21    0.013207   False
22    0.102643    True
23    0.014241    True
24    0.093412    True
25    0.000461   False
26    0.206803    True
27    0.004577    True
28    0.173810    True
29    0.189945   False
...        ...     ...
2045  0.000076    True
2046  0.044173   False
2047  0.086346    True
2048  0.032253   False
2049  0.029705    True
2050  0.134203    True
2051  0.038915    True
2052  0.026321   False
2053  0.026236    True
2054  0.000903    True
2055  0.038915    True
2056  0.111411    True
2057  0.073482    True
2058  0.105506    True
2059  0.027351    True
2060  0.007958    True
2061  0.001498    True
2062  0.004633    True
2063  0.082981   False
2064  0.008167    True
2065  0.029248    True
```

```
2066  0.118184    True
2067  0.076507    True
2068  0.193548   False
2069  0.076902   False
2070  0.175219    True
2071  0.040836   False
2072  0.031379    True
2073  0.068957    True
2074  0.076110   False

[2075 rows x 12 columns]
```

In [18]:
```
# Get general info
image_predictions.info()
```

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

In [19]:
```
# Return a random sample of items in Image Predictions Table
image_predictions.sample(5)
```

Out[19]:
```
                tweet_id                                          jpg_url  \
1210  742465774154047488  https://pbs.twimg.com/media/Ck3EribXEAAPhZn.jpg
745   687494652870668288  https://pbs.twimg.com/media/CYp4vFrVAAEs9AX.jpg
947   704761120771465216  https://pbs.twimg.com/media/CcfQgHVWoAAxauy.jpg
76    667435689202614272  https://pbs.twimg.com/media/CUM1OHCW4AEgGSi.jpg
818   692901601640583168  https://pbs.twimg.com/media/CZ2uU37UcAANzmK.jpg

      img_num                        p1   p1_conf  p1_dog  \
1210        1                   web_site  0.997154   False
745         1                 Rottweiler  0.391471    True
947         1                Siamese_cat  0.202294   False
76          1                 Rottweiler  0.999091    True
```

```
818         1  soft-coated_wheaten_terrier  0.403496    True

                        p2   p2_conf  p2_dog                       p3   p3_conf  \
1210          comic_book  0.000439   False         desktop_computer  0.000268
745   miniature_pinscher  0.273595    True          Tibetan_mastiff  0.041692
947            Chihuahua  0.100418    True                   basenji  0.072097
76    miniature_pinscher  0.000450    True  black-and-tan_coonhound  0.000157
818       cocker_spaniel  0.135164    True         golden_retriever  0.088719

      p3_dog
1210   False
745     True
947     True
76      True
818     True
```

In [20]:
```
# Testing Image
# This is an image for tweet_id: 770093767776997377
Image(url = 'https://pbs.twimg.com/media/CkjMx99UoAM2B1a.jpg')
```

Out[20]:
```
<IPython.core.display.Image object>
```

In [21]:
```
# Overall descriptive statistics
image_predictions.describe()
```

Out[21]:

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

In [22]:
```
# Check for duplicates
image_predictions.tweet_id.duplicated().sum()
```

Out[22]:
```
0
```

In [23]:
```
# Return a Series containing counts of p1 rows in the DataFrame.

image_predictions.p1.value_counts()
```

Out[23]:
```
golden_retriever             150
Labrador_retriever           100
Pembroke                      89
Chihuahua                     83
pug                           57
```

chow	44
Samoyed	43
toy_poodle	39
Pomeranian	38
malamute	30
cocker_spaniel	30
French_bulldog	26
miniature_pinscher	23
Chesapeake_Bay_retriever	23
seat_belt	22
German_shepherd	20
Siberian_husky	20
Staffordshire_bullterrier	20
Cardigan	19
web_site	19
Maltese_dog	18
teddy	18
beagle	18
Shetland_sheepdog	18
Eskimo_dog	18
Rottweiler	17
Shih-Tzu	17
Lakeland_terrier	17
Italian_greyhound	16
kuvasz	16
...	
restaurant	1
espresso	1
killer_whale	1
pillow	1
carton	1
wooden_spoon	1
pot	1
American_black_bear	1
guenon	1
hummingbird	1
conch	1
cougar	1
beaver	1
dhole	1
bow	1
jersey	1
electric_fan	1
shopping_basket	1
lawn_mower	1
snowmobile	1
Scotch_terrier	1
platypus	1

```
crane                          1
desktop_computer               1
bookshop                       1
grey_fox                       1
earthstar                      1
remote_control                 1
bison                          1
peacock                        1
Name: p1, Length: 378, dtype: int64
```

In [24]:
```
# Return a Series containing counts of p2 rows in the DataFrame.

image_predictions.p2.value_counts()
```

Out[24]:
```
Labrador_retriever               104
golden_retriever                  92
Cardigan                          73
Chihuahua                         44
Pomeranian                        42
French_bulldog                    41
Chesapeake_Bay_retriever          41
toy_poodle                        37
cocker_spaniel                    34
miniature_poodle                  33
Siberian_husky                    33
beagle                            28
collie                            27
Pembroke                          27
Eskimo_dog                        27
kuvasz                            26
Italian_greyhound                 22
Pekinese                          21
American_Staffordshire_terrier    21
malinois                          20
chow                              20
miniature_pinscher                20
Samoyed                           20
toy_terrier                       20
Boston_bull                       19
Norwegian_elkhound                19
Staffordshire_bullterrier         18
Irish_terrier                     17
pug                               17
kelpie                            16
                                 ...
dam                                1
torch                              1
EntleBucher                        1
```

```
cloak                              1
hotdog                             1
water_buffalo                      1
coffee_mug                         1
white_wolf                         1
bagel                              1
bathing_cap                        1
folding_chair                      1
African_hunting_dog                1
cannon                             1
Windsor_tie                        1
brown_bear                         1
lifeboat                           1
medicine_chest                     1
wood_rabbit                        1
rule                               1
barbershop                         1
hyena                              1
printer                            1
shoji                              1
lawn_mower                         1
mashed_potato                      1
tree_frog                          1
birdhouse                          1
shower_cap                         1
crate                              1
tiger                              1
Name: p2, Length: 405, dtype: int64
```

In [25]:
```
# Return a Series containing counts of p3 rows in the DataFrame.

image_predictions.p3.value_counts()
```

Out[25]:
```
Labrador_retriever                79
Chihuahua                         58
golden_retriever                  48
Eskimo_dog                        38
kelpie                            35
kuvasz                            34
chow                              32
Staffordshire_bullterrier         32
beagle                            31
cocker_spaniel                    31
Pomeranian                        29
toy_poodle                        29
Pekinese                          29
Chesapeake_Bay_retriever          27
Great_Pyrenees                    27
```

```
Pembroke                         27
malamute                         26
French_bulldog                   26
American_Staffordshire_terrier   24
pug                              23
Cardigan                         23
basenji                          21
toy_terrier                      20
bull_mastiff                     20
Siberian_husky                   19
Boston_bull                      17
Shetland_sheepdog                17
doormat                          16
Lakeland_terrier                 16
boxer                            16
                                 ..
pop_bottle                        1
buckeye                           1
wolf_spider                       1
chest                             1
barrow                            1
screen                            1
paintbrush                        1
guillotine                        1
panpipe                           1
tiger_cat                         1
restaurant                        1
European_fire_salamander          1
beach_wagon                       1
standard_schnauzer                1
space_shuttle                     1
bonnet                            1
African_chameleon                 1
pajama                            1
crossword_puzzle                  1
Windsor_tie                       1
brown_bear                        1
drumstick                         1
quill                             1
hare                              1
triceratops                       1
loupe                             1
valley                            1
shower_cap                        1
shoji                             1
power_drill                       1
Name: p3, Length: 408, dtype: int64
```

Observation:

- Inconsistent capitalization in p1, p2 and p3 columns
- Many entries are not dogs, e.g., soccer_ball, cardigan, stove, pot, mailbox, shovel, banana, etc.

1.1.3 Tweet Json

In [26]: `tweet_json`

Out[26]:

	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]
```

Observation:

Missing data probably due to retweets in twitter_archive

```
In [27]: # Get general info
         tweet_json.info()
```

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

```
In [28]: # Overall Descriptive statistics
         tweet_json.describe()
```

```
Out[28]:            tweet_id  retweet_count  favorite_count
         count  2.354000e+03    2354.000000     2354.000000
```

```
mean    7.426978e+17    3164.797366     8080.968564
std     6.852812e+16    5284.770364    11814.771334
min     6.660209e+17       0.000000        0.000000
25%     6.783975e+17     624.500000     1415.000000
50%     7.194596e+17    1473.500000     3603.500000
75%     7.993058e+17    3652.000000    10122.250000
max     8.924206e+17   79515.000000   132810.000000
```

```
In [29]: # Check for duplicates
         tweet_json.duplicated().sum()
```

```
Out[29]: 0
```

1.1.4 Quality

1. Some missing values in the column. Remove retweets: It is only original tweets we need.
2. Name column contain false names like "a".
3. Underscores in p1, p2 and p3 column
4. Inconsistent capitalization in p1, p2 and p3 columns
5. False predictions: predictions contain many entries that are not dogs , e.g., soccer_ball, cardigan, stove, pot, mailbox, shovel, banana, etc.
6. The proportions in p1_conf, p2_conf and p3_conf columns should be percentages
7. The numerator and denominator columns have invalid values and rating containing decimal numbers in numerator
8. Most predicted breed for each prediction level should be created.
9. Some column headers are not descriptive e.g jpg_url
10. Sources format are not readable.
11. timestamp is in object instead of string and datetime and date and time should be separated
12. Remove duplicated columns
13. Some columns are in inappropriate data type e.g tweet_id should be a string not an integer

1.1.5 Tidiness

1. Dog type(doggo, floofer, pupper, puppo) should be in one column
2. The three tables(twitter_archive, image_predictions and tweet_json) should be merged into one since they're all related to the same type of observational unit according to tidy data requirements

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 [30]: # Make copies of original pieces of data
         twitter_archives_clean = twitter_archives.copy()
         image_predictions_clean = image_predictions.copy()
         tweet_json_clean = tweet_json.copy()
```

Merge the clean versions of twitter_archives, image_predictions, and tweet_json dataframes

```
In [31]: twitter_dogs_data = pd.concat([twitter_archives_clean, image_predictions_clean, tweet_j

In [32]: twitter_dogs_data.head()
```

Out[32]:

	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	...	p1_dog
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892420643...	...	True
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you...	NaN	NaN	NaN			
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	NaN	NaN	NaN			
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	NaN	NaN	NaN			
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	NaN	NaN	NaN			

```
1  https://twitter.com/dog_rates/status/892177421...       ...          True
2  https://twitter.com/dog_rates/status/891815181...       ...          True
3  https://twitter.com/dog_rates/status/891689557...       ...          True
4  https://twitter.com/dog_rates/status/891327558...       ...          True

                   p2   p2_conf p2_dog                   p3   p3_conf p3_dog  \
0              collie  0.156665   True    Shetland_sheepdog  0.061428   True
1  miniature_pinscher  0.074192   True  Rhodesian_ridgeback  0.072010   True
2           malinois  0.138584   True            bloodhound  0.116197   True
3             redbone  0.360687   True   miniature_pinscher  0.222752   True
4          Rottweiler  0.243682   True              Doberman  0.154629   True

       tweet_id retweet_count  favorite_count
0  8.924206e+17        8853.0         39467.0
1  8.921774e+17        6514.0         33819.0
2  8.918152e+17        4328.0         25461.0
3  8.916896e+17        8964.0         42908.0
4  8.913276e+17        9774.0         41048.0

[5 rows x 32 columns]
```

1.1.6 Issue #1:

Removing the missing values

Define: There are some missing values in the column, so we need to remove them.

Code

```
In [33]: # checking for missing values in the dataframe
         twitter_dogs_data.isna().sum()

Out[33]: tweet_id                       0
         in_reply_to_status_id       2278
         in_reply_to_user_id         2278
         timestamp                      0
         source                         0
         text                           0
         retweeted_status_id         2175
         retweeted_status_user_id    2175
         retweeted_status_timestamp  2175
         expanded_urls                 59
         rating_numerator               0
         rating_denominator             0
         name                           0
         doggo                          0
         floofer                        0
         pupper                         0
```

```
puppo                         0
tweet_id                    281
jpg_url                     281
img_num                     281
p1                          281
p1_conf                     281
p1_dog                      281
p2                          281
p2_conf                     281
p2_dog                      281
p3                          281
p3_conf                     281
p3_dog                      281
tweet_id                      2
retweet_count                 2
favorite_count                2
dtype: int64
```

In [34]:
```
# This gives you a percentage value of missing values
(twitter_dogs_data.isna().sum()/twitter_dogs_data.shape[0])*100

# when there is too much percentage of missing values you drop the columns
```

Out[34]:
```
tweet_id                     0.000000
in_reply_to_status_id       96.689304
in_reply_to_user_id         96.689304
timestamp                    0.000000
source                       0.000000
text                         0.000000
retweeted_status_id         92.317487
retweeted_status_user_id    92.317487
retweeted_status_timestamp  92.317487
expanded_urls                2.504244
rating_numerator             0.000000
rating_denominator           0.000000
name                         0.000000
doggo                        0.000000
floofer                      0.000000
pupper                       0.000000
puppo                        0.000000
tweet_id                    11.926995
jpg_url                     11.926995
img_num                     11.926995
p1                          11.926995
p1_conf                     11.926995
p1_dog                      11.926995
p2                          11.926995
p2_conf                     11.926995
```

```
p2_dog                      11.926995
p3                          11.926995
p3_conf                     11.926995
p3_dog                      11.926995
tweet_id                     0.084890
retweet_count                0.084890
favorite_count               0.084890
dtype: float64
```

```
In [35]: # Let's drop missing values
         twitter_dogs_data.dropna(axis=1).isna().sum()
```

```
Out[35]: tweet_id             0
         timestamp            0
         source               0
         text                 0
         rating_numerator     0
         rating_denominator   0
         name                 0
         doggo                0
         floofer              0
         pupper               0
         puppo                0
         dtype: int64
```

Test

```
In [36]: twitter_dogs_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 32 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
tweet_id                       2075 non-null float64
jpg_url                        2075 non-null object
img_num                        2075 non-null float64
p1                             2075 non-null object
p1_conf                        2075 non-null float64
p1_dog                         2075 non-null object
p2                             2075 non-null object
p2_conf                        2075 non-null float64
p2_dog                         2075 non-null object
p3                             2075 non-null object
p3_conf                        2075 non-null float64
p3_dog                         2075 non-null object
tweet_id                       2354 non-null float64
retweet_count                  2354 non-null float64
favorite_count                 2354 non-null float64
dtypes: float64(12), int64(3), object(17)
memory usage: 589.1+ KB
```

Removing retweets

Define: We only need original tweets

Code

```
In [37]: twitter_dogs_data = twitter_dogs_data[twitter_dogs_data.retweeted_status_id.isnull()]
         twitter_dogs_data = twitter_dogs_data[twitter_dogs_data.retweeted_status_user_id.isnull
         twitter_dogs_data = twitter_dogs_data[twitter_dogs_data.retweeted_status_timestamp.isnu
```

Test

```
In [38]: twitter_dogs_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 32 columns):
tweet_id                       2175 non-null int64
in_reply_to_status_id          78 non-null float64
in_reply_to_user_id            78 non-null float64
timestamp                      2175 non-null object
source                         2175 non-null object
text                           2175 non-null object
retweeted_status_id            0 non-null float64
retweeted_status_user_id       0 non-null float64
retweeted_status_timestamp     0 non-null object
expanded_urls                  2117 non-null object
rating_numerator               2175 non-null int64
```

```
rating_denominator            2175 non-null int64
name                          2175 non-null object
doggo                         2175 non-null object
floofer                       2175 non-null object
pupper                        2175 non-null object
puppo                         2175 non-null object
tweet_id                      1896 non-null float64
jpg_url                       1896 non-null object
img_num                       1896 non-null float64
p1                            1896 non-null object
p1_conf                       1896 non-null float64
p1_dog                        1896 non-null object
p2                            1896 non-null object
p2_conf                       1896 non-null float64
p2_dog                        1896 non-null object
p3                            1896 non-null object
p3_conf                       1896 non-null float64
p3_dog                        1896 non-null object
tweet_id                      2173 non-null float64
retweet_count                 2173 non-null float64
favorite_count                2173 non-null float64
dtypes: float64(12), int64(3), object(17)
memory usage: 560.7+ KB
```

Let's drop columns not needed

Code

```
In [39]: # We will remove retweet and a column we won't use
         twitter_dogs_data = twitter_dogs_data.drop(['in_reply_to_status_id', 'in_reply_to_user_
                                                     'retweeted_status_id', 'retweeted
                                                     'retweeted_status_timestamp', 'ex
```

Test

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 26 columns):
tweet_id              2175 non-null int64
timestamp             2175 non-null object
source                2175 non-null object
text                  2175 non-null object
rating_numerator      2175 non-null int64
rating_denominator    2175 non-null int64
name                  2175 non-null object
```

```
doggo                2175 non-null object
floofer              2175 non-null object
pupper               2175 non-null object
puppo                2175 non-null object
tweet_id             1896 non-null float64
jpg_url              1896 non-null object
img_num              1896 non-null float64
p1                   1896 non-null object
p1_conf              1896 non-null float64
p1_dog               1896 non-null object
p2                   1896 non-null object
p2_conf              1896 non-null float64
p2_dog               1896 non-null object
p3                   1896 non-null object
p3_conf              1896 non-null float64
p3_dog               1896 non-null object
tweet_id             2173 non-null float64
retweet_count        2173 non-null float64
favorite_count       2173 non-null float64
dtypes: float64(8), int64(3), object(15)
memory usage: 458.8+ KB
```

1.1.7 Issue #2:

Here, we will fix the name column. It contains false names like "a"

Define: Change lowercase names to None as they are wrong.

Code

```
In [41]: #filtering out rows where there is the word 'named' in the text and the name is in lowe
         #the names are incorrect and then creating a list out of their corresponding indices
         index_list = twitter_dogs_data.loc[twitter_dogs_data.text.str.contains('named') &
                                                    twitter_dogs_data.name.str.islower()].ind
         all_indices = twitter_dogs_data.index.tolist()
         for e in all_indices:
             if e in index_list:
                 for ele in list(range(len(index_list))):
                     #creating a list out of all such text values that contain the word 'named'
                     #is in lowercase
                     text_list = (twitter_dogs_data.loc[twitter_dogs_data.text.str.contains('nam
                                                        twitter_dogs_data.name.str.islower()].
                     #finding the index position in every text value in the list where 'named' o
                     num = text_list[ele].find('named')
                     #using this index position to extract a particular pattern of dog name out
                     #corresponding name value
                     x = twitter_dogs_data.loc[twitter_dogs_data.index == e, 'text'].str[num+6:]
```

```
                    twitter_dogs_data.loc[twitter_dogs_data.index == e, 'name'] = x
                    break
```

In [42]:
```
#replacing the remaining 'None's in the name column by NaN values
twitter_dogs_data.name = twitter_dogs_data.name.replace('None', np.nan)
twitter_dogs_data.name.head()
```

Out[42]:
```
0     Phineas
1       Tilly
2      Archie
3       Darla
4    Franklin
Name: name, dtype: object
```

Test

In [43]:
```
twitter_dogs_data.loc[twitter_dogs_data.text.str.contains('named')]
```

Out[43]:

	tweet_id	timestamp	\
1853	675706639471788032	2015-12-12 15:59:51 +0000	
1955	673636718965334016	2015-12-06 22:54:44 +0000	
2034	671743150407421952	2015-12-01 17:30:22 +0000	
2066	671147085991960577	2015-11-30 02:01:49 +0000	
2116	670427002554466305	2015-11-28 02:20:27 +0000	
2125	670361874861563904	2015-11-27 22:01:40 +0000	
2128	670303360680108032	2015-11-27 18:09:09 +0000	
2146	669923323644657664	2015-11-26 16:59:01 +0000	
2161	669564461267722241	2015-11-25 17:13:02 +0000	
2166	669363888236994561	2015-11-25 03:56:01 +0000	
2191	668955713004314625	2015-11-24 00:54:05 +0000	
2204	668636665813057536	2015-11-23 03:46:18 +0000	
2218	668507509523615744	2015-11-22 19:13:05 +0000	
2227	668268907921326080	2015-11-22 03:24:58 +0000	
2235	668171859951755264	2015-11-21 20:59:20 +0000	
2249	667861340749471744	2015-11-21 00:25:26 +0000	
2255	667773195014021121	2015-11-20 18:35:10 +0000	
2264	667538891197542400	2015-11-20 03:04:08 +0000	
2269	667509364010450944	2015-11-20 01:06:48 +0000	
2273	667470559035432960	2015-11-19 22:32:36 +0000	
2304	666983947667116034	2015-11-18 14:18:59 +0000	
2311	666781792255496192	2015-11-18 00:55:42 +0000	
2314	666701168228331520	2015-11-17 19:35:19 +0000	

	source	\
1853	<a href="http://twitter.com/download/iphone" r...	
1955	<a href="http://twitter.com/download/iphone" r...	
2034	<a href="http://twitter.com/download/iphone" r...	

```
2066  <a href="http://twitter.com/download/iphone" r...
2116  <a href="http://twitter.com/download/iphone" r...
2125  <a href="http://twitter.com/download/iphone" r...
2128  <a href="http://twitter.com/download/iphone" r...
2146  <a href="http://twitter.com/download/iphone" r...
2161  <a href="http://twitter.com/download/iphone" r...
2166  <a href="http://twitter.com/download/iphone" r...
2191  <a href="http://twitter.com/download/iphone" r...
2204  <a href="http://twitter.com/download/iphone" r...
2218  <a href="http://twitter.com/download/iphone" r...
2227  <a href="http://twitter.com/download/iphone" r...
2235  <a href="http://twitter.com/download/iphone" r...
2249  <a href="http://twitter.com/download/iphone" r...
2255  <a href="http://twitter.com" rel="nofollow">Tw...
2264  <a href="http://twitter.com" rel="nofollow">Tw...
2269  <a href="http://twitter.com" rel="nofollow">Tw...
2273  <a href="http://twitter.com" rel="nofollow">Tw...
2304  <a href="http://twitter.com/download/iphone" r...
2311  <a href="http://twitter.com/download/iphone" r...
2314  <a href="http://twitter.com/download/iphone" r...

                                                 text  rating_numerator  \
1853  This is a Sizzlin Menorah spaniel from Brookly...                10
1955  This is a Lofted Aphrodisiac Terrier named Kip...                10
2034  This is a Tuscaloosa Alcatraz named Jacob (Yac...                11
2066  This is a Helvetica Listerine named Rufus. Thi...                 9
2116  This is a Deciduous Trimester mix named Spork...                 9
2125  This is a Rich Mahogany Seltzer named Cherokee...                 9
2128  This is a Speckled Cauliflower Yosemite named ...                 9
2146  This is a spotted Lipitor Rumpelstiltskin name...                10
2161  This is a Coriander Baton Rouge named Alfredo...                10
2166  Here we have a Gingivitis Pumpernickel named Z...                10
2191  This is a Slovakian Helter Skelter Feta named ...                10
2204  This is an Irish Rigatoni terrier named Berta...                10
2218  This is a Birmingham Quagmire named Chuk. Love...                10
2227  Here we have an Azerbaijani Buttermilk named G...                10
2235  This is a Trans Siberian Kellogg named Alfonso...                 7
2249  This is a Shotokon Macadamia mix named Cheryl...                 9
2255  This is a rare Hungarian Pinot named Jessiga. ...                 8
2264  This is a southwest Coriander named Klint. Hat...                 9
2269  This a Norwegian Pewterschmidt named Tickles. ...                12
2273  This is a northern Wahoo named Kohl. He runs t...                11
2304  This is a curly Ticonderoga named Pepe. No fee...                11
2311  This is a purebred Bacardi named Octaviath. Ca...                10
2314  This is a golden Buckminsterfullerene named Jo...                 8

      rating_denominator       name doggo floofer pupper      ...        \
1853                  10      Wylie  None    None   None      ...
```

```
1955                10        Kip  None    None   None      ...
2034                10      Jacob  None    None   None      ...
2066                10      Rufus  None    None   None      ...
2116                10      Spork  None    None   None      ...
2125                10   Cherokee  None    None   None      ...
2128                10      Hemry  None    None   None      ...
2146                10    Alphred  None    None   None      ...
2161                10    Alfredo  None    None   None      ...
2166                10        NaN  None    None   None      ...
2191                10      Leroi  None    None   None      ...
2204                10      Berta  None    None   None      ...
2218                10       Chuk  None    None   None      ...
2227                10        NaN  None    None   None      ...
2235                10    Alfonso  None    None   None      ...
2249                10     Cheryl  None    None   None      ...
2255                10    Jessiga  None    None   None      ...
2264                10      Klint  None    None   None      ...
2269                10        NaN  None    None   None      ...
2273                10       Kohl  None    None   None      ...
2304                10       Pepe  None    None   None      ...
2311                10  Octaviath  None    None   None      ...
2314                10       Johm  None    None   None      ...

     p1_dog                 p2   p2_conf  p2_dog                       p3  \
1853  False               rule  0.007087   False                 envelope
1955   True      French_bulldog  0.180562    True  Staffordshire_bullterrier
2034   True  miniature_pinscher  0.299603    True                   kelpie
2066   True        Irish_setter  0.193054    True  Chesapeake_Bay_retriever
2116    NaN                 NaN       NaN     NaN                       NaN
2125    NaN                 NaN       NaN     NaN                       NaN
2128    NaN                 NaN       NaN     NaN                       NaN
2146    NaN                 NaN       NaN     NaN                       NaN
2161    NaN                 NaN       NaN     NaN                       NaN
2166    NaN                 NaN       NaN     NaN                       NaN
2191    NaN                 NaN       NaN     NaN                       NaN
2204    NaN                 NaN       NaN     NaN                       NaN
2218    NaN                 NaN       NaN     NaN                       NaN
2227    NaN                 NaN       NaN     NaN                       NaN
2235    NaN                 NaN       NaN     NaN                       NaN
2249    NaN                 NaN       NaN     NaN                       NaN
2255    NaN                 NaN       NaN     NaN                       NaN
2264    NaN                 NaN       NaN     NaN                       NaN
2269    NaN                 NaN       NaN     NaN                       NaN
2273    NaN                 NaN       NaN     NaN                       NaN
2304    NaN                 NaN       NaN     NaN                       NaN
2311    NaN                 NaN       NaN     NaN                       NaN
2314    NaN                 NaN       NaN     NaN                       NaN
```

```
      p3_conf p3_dog      tweet_id  retweet_count favorite_count
1853  0.006820  False  6.755315e+17          428.0          1276.0
1955  0.052237   True  6.735831e+17          403.0          1273.0
2034  0.063020   True  6.717299e+17         4795.0          9119.0
2066  0.118184   True  6.711387e+17          448.0           996.0
2116       NaN    NaN  6.704206e+17          342.0           668.0
2125       NaN    NaN  6.703191e+17         1359.0          4110.0
2128       NaN    NaN  6.700939e+17          365.0          1106.0
2146       NaN    NaN  6.697494e+17           71.0           289.0
2161       NaN    NaN  6.693757e+17          792.0          1425.0
2166       NaN    NaN  6.693544e+17         1390.0          2889.0
2191       NaN    NaN  6.689030e+17          107.0           338.0
2204       NaN    NaN  6.686314e+17          349.0           763.0
2218       NaN    NaN  6.684842e+17          253.0           453.0
2227       NaN    NaN  6.682485e+17          523.0          1056.0
2235       NaN    NaN  6.681423e+17          306.0           592.0
2249       NaN    NaN  6.678065e+17          535.0          1111.0
2255       NaN    NaN  6.677282e+17          162.0           398.0
2264       NaN    NaN  6.675309e+17          264.0           501.0
2269       NaN    NaN  6.674958e+17          294.0           565.0
2273       NaN    NaN  6.674530e+17           96.0           327.0
2304       NaN    NaN  6.668350e+17           83.0           222.0
2311       NaN    NaN  6.667393e+17           71.0           244.0
2314       NaN    NaN  6.666495e+17          608.0           923.0

[23 rows x 26 columns]
```

```
In [44]: twitter_dogs_data[twitter_dogs_data.name == 'None'] # This is for checking for none val
         twitter_dogs_data[twitter_dogs_data.name == 'Nan'] # Also checked for missing values an
```

```
Out[44]: Empty DataFrame
         Columns: [tweet_id, timestamp, source, text, rating_numerator, rating_denominator, name
         Index: []

         [0 rows x 26 columns]
```

1.1.8 Issue #3:

Replacing the Underscores

Define: Replace the underscores in the p1, p2 and p3 columns by spaces

Code

```
In [45]: # Replacing using the replace function
         twitter_dogs_data.p1 = twitter_dogs_data.p1.str.replace('_',' ')
         twitter_dogs_data.p2 = twitter_dogs_data.p2.str.replace('_',' ')
         twitter_dogs_data.p3 = twitter_dogs_data.p3.str.replace('_',' ')
```

Test

```
In [46]: twitter_dogs_data[['p1', 'p2', 'p3']].head()
```

```
Out[46]:                        p1                 p2                   p3
         0  Welsh springer spaniel              collie    Shetland sheepdog
         1                 redbone  miniature pinscher  Rhodesian ridgeback
         2        German shepherd            malinois           bloodhound
         3    Rhodesian ridgeback             redbone   miniature pinscher
         4     miniature pinscher          Rottweiler             Doberman
```

1.1.9 Issue #4:

Inconsistent capitalization in p1, p2 and p3 columns

Define: Fixing inconsistent capitalization in p1, p2 and p3 columns

Code

```
In [47]: # Fix capitalization by using the str.title function
         twitter_dogs_data.p1 = twitter_dogs_data.p1.str.title()
         twitter_dogs_data.p2 = twitter_dogs_data.p2.str.title()
         twitter_dogs_data.p3 = twitter_dogs_data.p3.str.title()
```

Test

```
In [48]: twitter_dogs_data[['p1', 'p2', 'p3']]
```

```
Out[48]:                            p1                     p2  \
         0          Welsh Springer Spaniel                 Collie
         1                        Redbone     Miniature Pinscher
         2                German Shepherd                Malinois
         3             Rhodesian Ridgeback                Redbone
         4              Miniature Pinscher              Rottweiler
         5            Bernese Mountain Dog       English Springer
         6                       Box Turtle             Mud Turtle
         7                             Chow        Tibetan Mastiff
         8                  Shopping Cart        Shopping Basket
         9                Miniature Poodle               Komondor
         10               Golden Retriever        Tibetan Mastiff
         11                  Gordon Setter      Yorkshire Terrier
         12                  Walker Hound       English Foxhound
         13                           Pug           Bull Mastiff
         14                    Bloodhound        German Shepherd
         15                         Lhasa               Shih-Tzu
         16                English Setter            Newfoundland
         17                           Hen                   Cock
         18              Desktop Computer                   Desk
         20                   Maltese Dog            Toy Poodle
```

21	Three-Toed Sloth	Otter
22	Ox	Newfoundland
23	Golden Retriever	Chesapeake Bay Retriever
24	Malamute	Siberian Husky
25	Guinea Pig	Skunk
26	Soft-Coated Wheaten Terrier	Afghan Hound
27	Chihuahua	Toy Terrier
28	Black-And-Tan Coonhound	Bloodhound
29	Coho	Barracouta
30	Toy Terrier	Papillon
...	...	...
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

	p3
0	Shetland Sheepdog
1	Rhodesian Ridgeback
2	Bloodhound
3	Miniature Pinscher
4	Doberman

5	Greater Swiss Mountain Dog
6	Terrapin
7	Fur Coat
8	Golden Retriever
9	Soft-Coated Wheaten Terrier
10	Labrador Retriever
11	Pekinese
12	Ibizan Hound
13	French Bulldog
14	Malinois
15	Dandie Dinmont
16	Borzoi
17	Partridge
18	Bookcase
20	Miniature Poodle
21	Great Grey Owl
22	Groenendael
23	Labrador Retriever
24	Eskimo Dog
25	Hamster
26	Briard
27	Papillon
28	Flat-Coated Retriever
29	Gar
30	Chihuahua
...	...
2326	NaN
2327	NaN
2328	NaN
2329	NaN
2330	NaN
2331	NaN
2332	NaN
2333	NaN
2334	NaN
2335	NaN
2336	NaN
2337	NaN
2338	NaN
2339	NaN
2340	NaN
2341	NaN
2342	NaN
2343	NaN
2344	NaN
2345	NaN
2346	NaN
2347	NaN

```
2348                          NaN
2349                          NaN
2350                          NaN
2351                          NaN
2352                          NaN
2353                          NaN
2354                          NaN
2355                          NaN

[2175 rows x 3 columns]
```

1.1.10 Let's remove the NaNs in p1, p2, p3 columns

Code

```
In [49]: dog_breed = twitter_dogs_data[['p1', 'p2', 'p3']]
         dog_breed.head()
```

Out[49]:

	p1	p2	p3
0	Welsh Springer Spaniel	Collie	Shetland Sheepdog
1	Redbone	Miniature Pinscher	Rhodesian Ridgeback
2	German Shepherd	Malinois	Bloodhound
3	Rhodesian Ridgeback	Redbone	Miniature Pinscher
4	Miniature Pinscher	Rottweiler	Doberman

```
In [50]: dog_breed.shape

Out[50]: (2175, 3)

In [51]: dog_breed.isna().sum()

Out[51]: p1    279
         p2    279
         p3    279
         dtype: int64

In [52]: dog_breed.dropna(inplace=True)
```

Test

```
In [53]: dog_breed.isna().any()

Out[53]: p1    False
         p2    False
         p3    False
         dtype: bool

In [54]: dog_breed.sample(10)
```

Out[54]:

	p1	p2	p3
824	Vizsla	Chesapeake Bay Retriever	Rhodesian Ridgeback
1603	Golden Retriever	Kuvasz	Labrador Retriever
1565	Old English Sheepdog	Bedlington Terrier	Kerry Blue Terrier
1175	Samoyed	Pomeranian	Maltese Dog
2054	French Bulldog	Pug	Bull Mastiff
1781	Hippopotamus	Mexican Hairless	Ice Lolly
378	Airedale	Brown Bear	Chesapeake Bay Retriever
350	Doberman	Rottweiler	Appenzeller
697	Chihuahua	Siamese Cat	Macaque
1796	Old English Sheepdog	Tibetan Terrier	Guinea Pig

In [55]: dog_breed.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1896 entries, 0 to 2074
Data columns (total 3 columns):
p1    1896 non-null object
p2    1896 non-null object
p3    1896 non-null object
dtypes: object(3)
memory usage: 59.2+ KB
```

In [56]: twitter_dogs_data.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 26 columns):
tweet_id              2175 non-null int64
timestamp             2175 non-null object
source                2175 non-null object
text                  2175 non-null object
rating_numerator      2175 non-null int64
rating_denominator    2175 non-null int64
name                  1495 non-null object
doggo                 2175 non-null object
floofer               2175 non-null object
pupper                2175 non-null object
puppo                 2175 non-null object
tweet_id              1896 non-null float64
jpg_url               1896 non-null object
img_num               1896 non-null float64
p1                    1896 non-null object
p1_conf               1896 non-null float64
p1_dog                1896 non-null object
p2                    1896 non-null object
p2_conf               1896 non-null float64
p2_dog                1896 non-null object
```

```
p3                    1896 non-null object
p3_conf               1896 non-null float64
p3_dog                1896 non-null object
tweet_id              2173 non-null float64
retweet_count         2173 non-null float64
favorite_count        2173 non-null float64
dtypes: float64(8), int64(3), object(15)
memory usage: 458.8+ KB
```

1.1.11 Issue #5:

False predictions: predictions contain many entries that are not dogs

Define: Drop each of the false row prediction

Code

```
In [57]: false_predictions = ~((twitter_dogs_data.p1_dog) | (twitter_dogs_data.p2_dog) | (twitte
         false_predictions_dog = twitter_dogs_data[false_predictions].index.tolist()

In [58]: len(twitter_dogs_data[false_predictions])

Out[58]: 564

In [59]: twitter_dogs_data.drop(false_predictions_dog, inplace = True)

In [60]: twitter_dogs_data = twitter_dogs_data.reset_index(drop = True)
```

Test

```
In [61]: false_predictions = ~((twitter_dogs_data.p1_dog) | (twitter_dogs_data.p2_dog) | (twitte

         len(twitter_dogs_data[false_predictions])

Out[61]: 0
```

1.1.12 Issue #6:

The proportions in p1_conf, p2_conf and p3_conf columns should be percentages

Define: Convert the proportions in the p1_conf, p2_conf and p3_conf columns into percentages

```
In [62]: #using apply, multiplying 100 to each column value in each row
         twitter_dogs_data.p1_conf = twitter_dogs_data.p1_conf.apply(lambda x: round(x*100, 2))
         twitter_dogs_data.p2_conf = twitter_dogs_data.p2_conf.apply(lambda x: round(x*100, 2))
         twitter_dogs_data.p3_conf = twitter_dogs_data.p3_conf.apply(lambda x: round(x*100, 2))

In [63]: twitter_dogs_data.head()
```

Out[63]:

```
             tweet_id                  timestamp  \
0  892420643555336193  2017-08-01 16:23:56 +0000
1  892177421306343426  2017-08-01 00:17:27 +0000
2  891815181378084864  2017-07-31 00:18:03 +0000
3  891689557279858688  2017-07-30 15:58:51 +0000
4  891327558926688256  2017-07-29 16:00:24 +0000

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

                                                text  rating_numerator  \
0  This is Phineas. He's a mystical boy. Only eve...                 13
1  This is Tilly. She's just checking pup on you...                 13
2  This is Archie. He is a rare Norwegian Pouncin...                 12
3  This is Darla. She commenced a snooze mid meal...                 13
4  This is Franklin. He would like you to stop ca...                 12

   rating_denominator      name doggo floofer pupper      ...       p1_dog  \
0                  10   Phineas  None    None   None      ...         True
1                  10     Tilly  None    None   None      ...         True
2                  10    Archie  None    None   None      ...         True
3                  10     Darla  None    None   None      ...         True
4                  10  Franklin  None    None   None      ...         True

                    p2 p2_conf  p2_dog                   p3  p3_conf p3_dog  \
0              Collie   15.67    True    Shetland Sheepdog     6.14   True
1  Miniature Pinscher    7.42    True  Rhodesian Ridgeback     7.20   True
2           Malinois   13.86    True            Bloodhound    11.62   True
3             Redbone   36.07    True   Miniature Pinscher    22.28   True
4          Rottweiler   24.37    True              Doberman    15.46   True

       tweet_id  retweet_count favorite_count
0  8.924206e+17         8853.0        39467.0
1  8.921774e+17         6514.0        33819.0
2  8.918152e+17         4328.0        25461.0
3  8.916896e+17         8964.0        42908.0
4  8.913276e+17         9774.0        41048.0

[5 rows x 26 columns]
```

1.1.13 Issue #7:

The numerator and denominator columns have invalid values.

Define: Fix rating numerator and denominators that are not ratings

Code

```
In [64]: tmp_rating = twitter_dogs_data[twitter_dogs_data.text.str.contains( r"(\d+\.?\d*\/\d+\.

         for i in tmp_rating:
             x = twitter_dogs_data.text == i
             column_1 = 'rating_numerator'
             column_2 = 'rating_denominator'
             twitter_dogs_data.loc[x, column_1] = re.findall(r"\d+\.?\d*\/\d+\.?\d*\D+(\d+\.?\d*
             twitter_dogs_data.loc[x, column_2] = 10
```

Test

```
In [65]: twitter_dogs_data[twitter_dogs_data.text.isin(tmp_rating)]
```

```
Out[65]:                 tweet_id                  timestamp  \
         42    881633300179243008  2017-07-02 21:58:53 +0000
         200   835246439529840640  2017-02-24 21:54:03 +0000
         486   777684233540206592  2016-09-19 01:42:24 +0000
         673   746006769478692864  2016-06-28 01:21:27 +0000
         723   740373189193256964  2016-06-08 02:41:38 +0000
         842   716439118184652801  2016-04-03 01:36:11 +0000
         857   714258258790387713  2016-03-28 01:10:13 +0000
         975   703356393781329922  2016-02-26 23:10:06 +0000
         1061  695064344191721472  2016-02-04 02:00:27 +0000
         1066  694352839993344000  2016-02-02 02:53:12 +0000
         1104  691483041324204033  2016-01-25 04:49:38 +0000
         1120  690400367696297985  2016-01-22 05:07:29 +0000
         1133  689835978131935233  2016-01-20 15:44:48 +0000
         1244  682962037429899265  2016-01-01 16:30:13 +0000
         1366  677314812125323265  2015-12-17 02:30:09 +0000
         1453  674737130913071104  2015-12-09 23:47:22 +0000
         1456  674646392044941312  2015-12-09 17:46:48 +0000
         1514  673295268553605120  2015-12-06 00:17:55 +0000
         1552  672248013293752320  2015-12-03 02:56:30 +0000
         1601  671154572044468225  2015-11-30 02:31:34 +0000

                                                      source  \
         42    <a href="http://twitter.com/download/iphone" r...
         200   <a href="http://twitter.com/download/iphone" r...
         486   <a href="http://twitter.com/download/iphone" r...
         673   <a href="http://twitter.com/download/iphone" r...
         723   <a href="http://twitter.com/download/iphone" r...
         842   <a href="http://twitter.com/download/iphone" r...
         857   <a href="http://twitter.com/download/iphone" r...
         975   <a href="http://twitter.com/download/iphone" r...
```

```
1061  <a href="http://twitter.com/download/iphone" r...
1066  <a href="http://twitter.com/download/iphone" r...
1104  <a href="http://twitter.com/download/iphone" r...
1120  <a href="http://twitter.com/download/iphone" r...
1133  <a href="http://twitter.com/download/iphone" r...
1244  <a href="http://twitter.com/download/iphone" r...
1366  <a href="http://twitter.com/download/iphone" r...
1453  <a href="http://twitter.com/download/iphone" r...
1456  <a href="http://twitter.com/download/iphone" r...
1514  <a href="http://twitter.com/download/iphone" r...
1552  <a href="http://twitter.com/download/iphone" r...
1601  <a href="http://twitter.com/download/iphone" r...

                                                 text rating_numerator  \
42    @roushfenway These are good dogs but 17/10 is ...               13
200   @jonnysun @Lin_Manuel ok jomny I know you're e...               13
486   "Yep... just as I suspected. You're not flossi...               11
673   This is Bookstore and Seaweed. Bookstore is ti...                7
723   After so many requests, this is Bretagne. She ...               14
842   This is Bluebert. He just saw that both #Final...               11
857   Meet Travis and Flurp. Travis is pretty chill ...                8
975   This is Socks. That water pup w the super legs...                2
1061  This may be the greatest video I've ever been ...               13
1066  Meet Oliviér. He takes killer selfies. Has a d...                5
1104  When bae says they can't go out but you see th...               10
1120  This is Eriq. His friend just reminded him of ...                6
1133  Meet Fynn &amp;amp; Taco. Fynn is an all-powerful ...            10
1244  This is Darrel. He just robbed a 7/11 and is i...               10
1366  Meet Tassy &amp;amp; Bee. Tassy is pretty chill, b...            11
1453  Meet Rufio. He is unaware of the pink legless ...                4
1456  Two gorgeous dogs here. Little waddling dog is...                8
1514  Meet Eve. She's a raging alcoholic 8/10 (would...               11
1552  10/10 for dog. 7/10 for cat. 12/10 for human. ...                7
1601  Meet Holly. She's trying to teach small human-...                8

      rating_denominator       name doggo floofer  pupper      ...        \
42                    10        NaN  None    None    None      ...
200                   10        NaN  None    None    None      ...
486                   10        NaN  None    None    None      ...
673                   10  Bookstore  None    None    None      ...
723                   10        NaN  None    None    None      ...
842                   10   Bluebert  None    None    None      ...
857                   10     Travis  None    None    None      ...
975                   10      Socks  None    None    None      ...
1061                  10        NaN  None    None    None      ...
1066                  10    Oliviér  None    None    None      ...
1104                  10        NaN  None    None    None      ...
1120                  10       Eriq  None    None    None      ...
```

```
1133                 10       Fynn  None    None    None      ...
1244                 10     Darrel  None    None    None      ...
1366                 10      Tassy  None    None    None      ...
1453                 10      Rufio  None    None  pupper      ...
1456                 10        NaN  None    None    None      ...
1514                 10        Eve  None    None  pupper      ...
1552                 10        NaN  None    None    None      ...
1601                 10      Holly  None    None    None      ...

     p1_dog                          p2 p2_conf  p2_dog  \
42     True                      Vizsla    9.10    True
200    True      Black-And-Tan Coonhound   27.09    True
486    True             Golden Retriever    5.41    True
673    True      Wire-Haired Fox Terrier    8.35    True
723    True               French Bulldog    1.97    True
842    True                          Pug    1.49    True
857    True                   Pomeranian    1.22    True
975    True               Cocker Spaniel   10.58    True
1061   True               Cocker Spaniel    1.59    True
1066   True                Irish Terrier   12.15    True
1104   True             Golden Retriever   19.84    True
1120   True           Labrador Retriever   11.38    True
1133  False                      Bathtub   32.51   False
1244   True               French Bulldog   15.99    True
1366   True  Soft-Coated Wheaten Terrier    0.91    True
1453   True                  Maltese Dog   16.64    True
1456   True                    Chihuahua    7.73    True
1514  False               French Bulldog   10.00    True
1552   True                      Whippet   15.05    True
1601   True                   Pomeranian    1.39    True

                        p3  p3_conf p3_dog      tweet_id  retweet_count  \
42                  Kelpie     2.30   True  8.815360e+17        16570.0
200             Rottweiler    15.32   True  8.351728e+17         6516.0
486               Airedale     3.06   True  7.776419e+17         4947.0
673         English Setter     8.32   True  7.475127e+17         1803.0
723           Bull Mastiff     0.23   True  7.403590e+17          967.0
842               Pekinese     1.28   True  7.160809e+17         1935.0
857             White Wolf     0.48  False  7.142141e+17          990.0
975                 Borzoi     7.39   True  7.030791e+17         3494.0
1061                Lhasa     0.65   True  6.949258e+17         1043.0
1066      Lakeland Terrier     1.46   True  6.943297e+17          569.0
1104              Pekinese    14.33   True  6.914449e+17          955.0
1120      Brittany Spaniel     3.86   True  6.903604e+17         1006.0
1133      Golden Retriever     7.85   True  6.896594e+17         4412.0
1244               Doormat     5.87  False  6.827884e+17         1258.0
1366  Bouvier Des Flandres     0.47   True  6.772693e+17          790.0
1453              Shih-Tzu    14.23   True  6.746706e+17          729.0
```

```
1456        French Bulldog     7.66   True  6.746386e+17          650.0
1514               Printer     7.71  False  6.732408e+17          798.0
1552          Ibizan Hound     3.97   True  6.722393e+17          347.0
1601                 Chow      0.82   True  6.711471e+17          254.0

     favorite_count
42          50199.0
200         28552.0
486             0.0
673          6110.0
723          3610.0
842          5272.0
857          2480.0
975          8064.0
1061         2965.0
1066         2203.0
1104         2890.0
1120         2925.0
1133        11394.0
1244         2706.0
1366         2164.0
1453         1751.0
1456         1806.0
1514         1510.0
1552          953.0
1601          713.0

[20 rows x 26 columns]
```

1.1.14 Rating containing decimal numbers in numerator.

Define: Clean decimal values in rating numerators.

Code

```
In [66]: twitter_dogs_data[twitter_dogs_data.text.str.contains(r"(\d+\.\d*\/\d+)")]
```

```
Out[66]:                tweet_id                   timestamp  \
438   786709082849828864  2016-10-13 23:23:56 +0000
1269  681340665377193984  2015-12-28 05:07:27 +0000
1291  680494726643068929  2015-12-25 21:06:00 +0000

                                                source  \
438   <a href="http://twitter.com/download/iphone" r...
1269  <a href="http://twitter.com/download/iphone" r...
1291  <a href="http://twitter.com/download/iphone" r...

                                                 text rating_numerator  \
```

```
438   This is Logan, the Chow who lived. He solemnly...               75
1269  I've been told there's a slight possibility he...                5
1291  Here we have uncovered an entire battalion of ...               26

      rating_denominator   name doggo floofer pupper      ...       p1_dog  \
438                   10  Logan  None    None   None      ...         True
1269                  10    NaN  None    None   None      ...         True
1291                  10    NaN  None    None   None      ...         True

                           p2 p2_conf  p2_dog  \
438   Chesapeake Bay Retriever   19.49    True
1269       Norwegian Elkhound   14.00    True
1291  Chesapeake Bay Retriever   13.72    True

                                 p3  p3_conf p3_dog      tweet_id  \
438   American Staffordshire Terrier     5.95   True  7.866650e+17
1269                        Malinois     2.48   True  6.813202e+17
1291                        Malamute     7.14   True  6.804404e+17

      retweet_count favorite_count
438          2996.0        11957.0
1269          863.0         2918.0
1291          564.0         1583.0

[3 rows x 26 columns]
```

In [67]:
```
ratings = twitter_dogs_data.text.str.extract('((?:\d+\.)?\d+)\/(\d+)', expand=True)
ratings
```

Out[67]:

	0	1
0	13	10
1	13	10
2	12	10
3	13	10
4	12	10
5	13	10
6	13	10
7	13	10
8	14	10
9	13	10
10	13	10
11	13	10
12	12	10
13	13	10
14	13	10
15	12	10
16	12	10
17	14	10

```
18    13  10
19    13  10
20    13  10
21    13  10
22    13  10
23    12  10
24    13  10
25    12  10
26    13  10
27    14  10
28    13  10
29    12  10
...   ..  ..
1581   8  10
1582   9  10
1583   6  10
1584  10  10
1585   9  10
1586   8  10
1587  11  10
1588  11  10
1589   8  10
1590  11  10
1591   7  10
1592  11  10
1593   9  10
1594  10  10
1595   9  10
1596  10  10
1597  12  10
1598  10  10
1599   8  10
1600   5  10
1601  11  10
1602  12  10
1603   9  10
1604   9  10
1605  10  10
1606   8  10
1607   4  10
1608   6  10
1609   8  10
1610  12  10

[1611 rows x 2 columns]
```

In [68]:
```
#Convert the null values to None type
twitter_dogs_data['rating_numerator'] = ratings[0]
```

Test

```
In [69]: twitter_dogs_data[twitter_dogs_data.text.str.contains(r"(\d+\.\d*\/\d+)")]
```

```
Out[69]:                 tweet_id                   timestamp  \
         438   786709082849828864  2016-10-13 23:23:56 +0000
         1269  681340665377193984  2015-12-28 05:07:27 +0000
         1291  680494726643068929  2015-12-25 21:06:00 +0000

                                                    source  \
         438   <a href="http://twitter.com/download/iphone" r...
         1269  <a href="http://twitter.com/download/iphone" r...
         1291  <a href="http://twitter.com/download/iphone" r...

                                                    text rating_numerator  \
         438   This is Logan, the Chow who lived. He solemnly...             9.75
         1269  I've been told there's a slight possibility he...              9.5
         1291  Here we have uncovered an entire battalion of ...            11.26

              rating_denominator   name doggo floofer pupper      ...       p1_dog  \
         438                  10  Logan  None    None   None      ...         True
         1269                 10    NaN  None    None   None      ...         True
         1291                 10    NaN  None    None   None      ...         True

                                    p2 p2_conf  p2_dog  \
         438   Chesapeake Bay Retriever   19.49    True
         1269       Norwegian Elkhound   14.00    True
         1291  Chesapeake Bay Retriever   13.72    True

                                       p3  p3_conf p3_dog      tweet_id  \
         438   American Staffordshire Terrier     5.95   True  7.866650e+17
         1269                      Malinois     2.48   True  6.813202e+17
         1291                      Malamute     7.14   True  6.804404e+17

              retweet_count favorite_count
         438          2996.0        11957.0
         1269          863.0         2918.0
         1291          564.0         1583.0

         [3 rows x 26 columns]
```

1.1.15 Issue #8:

Most predicted breed for each prediction level should be created.

Define: Create top accurate predicted dog breed in a column and drop p1,p2,p3 related columns

Code

```
In [70]: columns = ['p1_dog','p2_dog','p3_dog']
         pred = []
         preds = []

         for index in range(twitter_dogs_data.shape[0]):
             for col in columns:
                 if twitter_dogs_data.loc[index,col] == True:
                     pred.append(col)
             preds.append(pred)
             pred = []
```

```
In [71]: corresponding_columns = {'p1_dog': ['p1','p1_conf'], 'p2_dog': ['p2','p2_conf'], 'p3_do
         prediction1 = []
         conf1 = []

         for index in range(twitter_dogs_data.shape[0]):
             prediction1.append(twitter_dogs_data.loc[index,corresponding_columns[preds[index][0
             conf1.append(twitter_dogs_data.loc[index,corresponding_columns[preds[index][0]][1]]

         prediction1 = pd.Series(prediction1)
         conf1 = pd.Series(conf1)
```

```
In [72]: twitter_dogs_data = pd.concat([twitter_dogs_data, prediction1,conf1], axis=1, join = 'o
```

```
In [73]: twitter_dogs_data.rename(columns={0: "dog_breed_prediction", 1: "confidence_percentage"
```

```
In [74]: twitter_dogs_data.drop(columns=['p1', 'p2','p3','p1_conf','p1_conf','p2_conf','p3_conf'
```

Test

```
In [75]: twitter_dogs_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1611 entries, 0 to 1610
Data columns (total 19 columns):
tweet_id                1611 non-null int64
timestamp               1611 non-null object
source                  1611 non-null object
text                    1611 non-null object
rating_numerator        1611 non-null object
rating_denominator      1611 non-null int64
name                    1111 non-null object
doggo                   1611 non-null object
floofer                 1611 non-null object
pupper                  1611 non-null object
puppo                   1611 non-null object
tweet_id                1611 non-null float64
jpg_url                 1611 non-null object
img_num                 1611 non-null float64
```

```
tweet_id                 1611 non-null float64
retweet_count            1611 non-null float64
favorite_count           1611 non-null float64
dog_breed_prediction     1611 non-null object
confidence_percentage    1611 non-null float64
dtypes: float64(6), int64(2), object(11)
memory usage: 239.2+ KB
```

```
In [76]: twitter_dogs_data.head()
```

Out[76]:

	tweet_id	timestamp	source	text	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo	tweet_id	jpg_url	img_num	tweet_id	retweet_count	favorite_count	dog_breed_prediction
0	892420643555336193	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	13	10	Phineas	None	None	None	None	6.660209e+17	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1.0	8.924206e+17	8853.0	39467.0	Welsh Springer Spaniel
1	892177421306343426	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you...	13	10	Tilly	None	None	None	None	6.660293e+17	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1.0	8.921774e+17			
2	891815181378084864	2017-07-31 00:18:03 +0000	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	12	10	Archie	None	None	None	None	6.660334e+17	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1.0	8.918152e+17			
3	891689557279858688	2017-07-30 15:58:51 +0000	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	13	10	Darla	None	None	None	None	6.660442e+17	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1.0	8.916896e+17			
4	891327558926688256	2017-07-29 16:00:24 +0000	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	12	10	Franklin	None	None	None	None	6.660492e+17	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1.0	8.913276e+17			

```
1         6514.0         33819.0                 Redbone
2         4328.0         25461.0         German Shepherd
3         8964.0         42908.0     Rhodesian Ridgeback
4         9774.0         41048.0      Miniature Pinscher

   confidence_percentage
0                  46.51
1                  50.68
2                  59.65
3                  40.81
4                  56.03
```

1.1.16 Issue #9:

Some column headers are not descriptive

Define: Change column headers to be more readable column names e.g, the column name of jpg_url for better viewing.

Code

```
In [77]: twitter_dogs_data.rename(columns={'jpg_url': 'image_url', 'name': "dog_name"}, inplace
```

Test

```
In [78]: twitter_dogs_data.head()
```

```
Out[78]:              tweet_id                  timestamp  \
         0  892420643555336193  2017-08-01 16:23:56 +0000
         1  892177421306343426  2017-08-01 00:17:27 +0000
         2  891815181378084864  2017-07-31 00:18:03 +0000
         3  891689557279858688  2017-07-30 15:58:51 +0000
         4  891327558926688256  2017-07-29 16:00:24 +0000

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

                                                  text rating_numerator  \
         0  This is Phineas. He's a mystical boy. Only eve...                13
         1  This is Tilly. She's just checking pup on you...                13
         2  This is Archie. He is a rare Norwegian Pouncin...                12
         3  This is Darla. She commenced a snooze mid meal...                13
         4  This is Franklin. He would like you to stop ca...                12
```

```
   rating_denominator  dog_name doggo floofer pupper puppo      tweet_id  \
0                  10   Phineas  None    None   None  None  6.660209e+17
1                  10     Tilly  None    None   None  None  6.660293e+17
2                  10    Archie  None    None   None  None  6.660334e+17
3                  10     Darla  None    None   None  None  6.660442e+17
4                  10  Franklin  None    None   None  None  6.660492e+17

                                       image_url  img_num      tweet_id  \
0  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg      1.0  8.924206e+17
1  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg      1.0  8.921774e+17
2  https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg      1.0  8.918152e+17
3  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg      1.0  8.916896e+17
4  https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg      1.0  8.913276e+17

   retweet_count  favorite_count    dog_breed_prediction  \
0         8853.0         39467.0  Welsh Springer Spaniel
1         6514.0         33819.0                Redbone
2         4328.0         25461.0         German Shepherd
3         8964.0         42908.0     Rhodesian Ridgeback
4         9774.0         41048.0      Miniature Pinscher

   confidence_percentage
0                  46.51
1                  50.68
2                  59.65
3                  40.81
4                  56.03
```

1.1.17 Issue #10:

Sources format are not readable.

Define: Make 'source' column clean and readable.

Code

```
In [79]: twitter_dogs_data['source'] = twitter_dogs_data['source'].apply(lambda x: re.findall(r'
```

Test

```
In [80]: twitter_dogs_data.head()
```

```
Out[80]:              tweet_id                  timestamp              source  \
         0  892420643555336193  2017-08-01 16:23:56 +0000  Twitter for iPhone
         1  892177421306343426  2017-08-01 00:17:27 +0000  Twitter for iPhone
         2  891815181378084864  2017-07-31 00:18:03 +0000  Twitter for iPhone
         3  891689557279858688  2017-07-30 15:58:51 +0000  Twitter for iPhone
         4  891327558926688256  2017-07-29 16:00:24 +0000  Twitter for iPhone
```

```
                                                  text rating_numerator  \
0  This is Phineas. He's a mystical boy. Only eve...                 13
1  This is Tilly. She's just checking pup on you...                 13
2  This is Archie. He is a rare Norwegian Pouncin...                 12
3  This is Darla. She commenced a snooze mid meal...                 13
4  This is Franklin. He would like you to stop ca...                 12

   rating_denominator  dog_name doggo floofer pupper puppo      tweet_id  \
0                  10   Phineas  None    None   None  None  6.660209e+17
1                  10     Tilly  None    None   None  None  6.660293e+17
2                  10    Archie  None    None   None  None  6.660334e+17
3                  10     Darla  None    None   None  None  6.660442e+17
4                  10  Franklin  None    None   None  None  6.660492e+17

                                       image_url  img_num      tweet_id  \
0  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg      1.0  8.924206e+17
1  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg      1.0  8.921774e+17
2  https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg      1.0  8.918152e+17
3  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg      1.0  8.916896e+17
4  https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg      1.0  8.913276e+17

   retweet_count  favorite_count    dog_breed_prediction  \
0         8853.0         39467.0  Welsh Springer Spaniel
1         6514.0         33819.0                Redbone
2         4328.0         25461.0         German Shepherd
3         8964.0         42908.0     Rhodesian Ridgeback
4         9774.0         41048.0      Miniature Pinscher

   confidence_percentage
0                  46.51
1                  50.68
2                  59.65
3                  40.81
4                  56.03
```

1.1.18 Issue #11:

timestamp is in object instead of string and datetime

Define: timestamp should be in datetime format and date and time should be separated

Code

```
In [81]: twitter_dogs_data.timestamp = pd.to_datetime(twitter_dogs_data.timestamp, yearfirst = T

In [82]: #using the apply function, applying the strftime function to each value of the timestam
         twitter_dogs_data['date'] = twitter_dogs_data['timestamp'].apply(lambda x: x.strftime('
```

```
        twitter_dogs_data['time'] = twitter_dogs_data['timestamp'].apply(lambda x: x.strftime('

        #changing datatype of the date column to datetime
        twitter_dogs_data.date = pd.to_datetime(twitter_dogs_data.date, dayfirst = True)
```

```
In [83]: #Now, let's drop the timestamp column
         twitter_dogs_data = twitter_dogs_data.drop('timestamp', axis = 1)
```

Test

```
In [84]: twitter_dogs_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1611 entries, 0 to 1610
Data columns (total 20 columns):
tweet_id                1611 non-null int64
source                  1611 non-null object
text                    1611 non-null object
rating_numerator        1611 non-null object
rating_denominator      1611 non-null int64
dog_name                1111 non-null object
doggo                   1611 non-null object
floofer                 1611 non-null object
pupper                  1611 non-null object
puppo                   1611 non-null object
tweet_id                1611 non-null float64
image_url               1611 non-null object
img_num                 1611 non-null float64
tweet_id                1611 non-null float64
retweet_count           1611 non-null float64
favorite_count          1611 non-null float64
dog_breed_prediction    1611 non-null object
confidence_percentage   1611 non-null float64
date                    1611 non-null datetime64[ns]
time                    1611 non-null object
dtypes: datetime64[ns](1), float64(6), int64(2), object(11)
memory usage: 251.8+ KB
```

1.1.19 Issue #12:

Duplicated Columns

Define: Remove duplicated columns

Code

```
In [85]: twitter_dogs_data = twitter_dogs_data.loc[:,~twitter_dogs_data.columns.duplicated()]
```

Test

```
In [86]: twitter_dogs_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1611 entries, 0 to 1610
Data columns (total 18 columns):
tweet_id                 1611 non-null int64
source                   1611 non-null object
text                     1611 non-null object
rating_numerator         1611 non-null object
rating_denominator       1611 non-null int64
dog_name                 1111 non-null object
doggo                    1611 non-null object
floofer                  1611 non-null object
pupper                   1611 non-null object
puppo                    1611 non-null object
image_url                1611 non-null object
img_num                  1611 non-null float64
retweet_count            1611 non-null float64
favorite_count           1611 non-null float64
dog_breed_prediction     1611 non-null object
confidence_percentage    1611 non-null float64
date                     1611 non-null datetime64[ns]
time                     1611 non-null object
dtypes: datetime64[ns](1), float64(4), int64(2), object(11)
memory usage: 226.6+ KB
```

1.1.20 Dog Type

Define: Dog type(doggo, floofer, pupper, puppo) should be in one column

Code

```
In [87]: # Checking for multiple stages
         twitter_dogs_data[['doggo','floofer','pupper','puppo']].sum(axis=1).unique()
```

```
Out[87]: array(['NoneNoneNoneNone', 'doggoNoneNoneNone', 'NoneNoneNonepuppo',
                'NoneflooferNoneNone', 'NoneNonepupperNone', 'doggoNoneNonepuppo',
                'doggoflooferNoneNone', 'doggoNonepupperNone'], dtype=object)
```

```
In [88]: twitter_dogs_data['multiple_stages'] = twitter_dogs_data[['doggo','floofer','pupper','p
         multiple_stages = twitter_dogs_data.query('multiple_stages == True')[['text', 'doggo','
         multiple_stages.shape
```

```
Out[88]: (0, 5)
```

```
In [89]: twitter_dogs_data['dog_type'] = twitter_dogs_data[
             ['doggo', 'floofer', 'pupper', 'puppo']].apply(lambda x: ', '.join(x), axis=1)
```

```
In [90]: twitter_dogs_data.drop(columns = ['doggo', 'floofer','pupper','puppo'], inplace = True)

In [91]: twitter_dogs_data = twitter_dogs_data.replace(regex=r'(None,? ?)', value='').replace(re

In [92]: twitter_dogs_data = twitter_dogs_data.replace(regex=r'', value= np.nan)
```

Test

```
In [93]: twitter_dogs_data.dog_type.value_counts()

Out[93]: pupper            191
         doggo              63
         puppo              20
         floofer             9
         doggo, pupper       8
         doggo, puppo        1
         doggo, floofer      1
         Name: dog_type, dtype: int64

In [94]: twitter_dogs_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1611 entries, 0 to 1610
Data columns (total 16 columns):
tweet_id                 1611 non-null int64
source                   1611 non-null object
text                     1611 non-null object
rating_numerator         1611 non-null object
rating_denominator       1611 non-null int64
dog_name                 1111 non-null object
image_url                1611 non-null object
img_num                  1611 non-null float64
retweet_count            1611 non-null float64
favorite_count           1611 non-null float64
dog_breed_prediction     1611 non-null object
confidence_percentage    1611 non-null float64
date                     1611 non-null datetime64[ns]
time                     1611 non-null object
multiple_stages          1611 non-null bool
dog_type                 293 non-null object
dtypes: bool(1), datetime64[ns](1), float64(4), int64(2), object(8)
memory usage: 190.4+ KB


In [95]: twitter_dogs_data.head()

Out[95]:              tweet_id              source  \
         0  892420643555336193  Twitter for iPhone
         1  892177421306343426  Twitter for iPhone
```

```
2  891815181378084864  Twitter for iPhone
3  891689557279858688  Twitter for iPhone
4  891327558926688256  Twitter for iPhone

                                                text rating_numerator  \
0  This is Phineas. He's a mystical boy. Only eve...               13
1  This is Tilly. She's just checking pup on you...               13
2  This is Archie. He is a rare Norwegian Pouncin...               12
3  This is Darla. She commenced a snooze mid meal...               13
4  This is Franklin. He would like you to stop ca...               12

   rating_denominator  dog_name  \
0                  10   Phineas
1                  10     Tilly
2                  10    Archie
3                  10     Darla
4                  10  Franklin

                                       image_url  img_num  retweet_count  \
0  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg      1.0         8853.0
1  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg      1.0         6514.0
2  https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg      1.0         4328.0
3  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg      1.0         8964.0
4  https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg      1.0         9774.0

   favorite_count    dog_breed_prediction  confidence_percentage       date  \
0         39467.0  Welsh Springer Spaniel                  46.51 2017-08-01
1         33819.0                Redbone                  50.68 2017-08-01
2         25461.0         German Shepherd                  59.65 2017-07-31
3         42908.0     Rhodesian Ridgeback                  40.81 2017-07-30
4         41048.0      Miniature Pinscher                  56.03 2017-07-29

       time  multiple_stages dog_type
0  16:23:56            False      NaN
1  00:17:27            False      NaN
2  00:18:03            False      NaN
3  15:58:51            False      NaN
4  16:00:24            False      NaN
```

1.1.21 Issue #13:

Data Types

Define: Change datatypes

Code

```
In [96]: twitter_dogs_data['tweet_id'] = twitter_dogs_data['tweet_id'].astype(str)

         twitter_dogs_data['dog_type'] = twitter_dogs_data['dog_type'].astype('category')

         twitter_dogs_data['source'] = twitter_dogs_data['source'].astype('category')

         twitter_dogs_data['rating_numerator'] = twitter_dogs_data['rating_numerator'].astype(fl

         twitter_dogs_data['rating_denominator'] = twitter_dogs_data['rating_denominator'].astyp
```

Test

```
In [97]: twitter_dogs_data.dtypes

Out[97]: tweet_id                       object
         source                       category
         text                           object
         rating_numerator              float64
         rating_denominator            float64
         dog_name                       object
         image_url                      object
         img_num                       float64
         retweet_count                 float64
         favorite_count                float64
         dog_breed_prediction           object
         confidence_percentage         float64
         date                   datetime64[ns]
         time                           object
         multiple_stages                  bool
         dog_type                     category
         dtype: object
```

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

```
In [98]: twitter_dogs_data.to_csv('twitter_archive_master.csv', encoding = 'utf-8', index=False)
```

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.**

```
In [99]: df = pd.read_csv('twitter_archive_master.csv')

In [100]: # Get general info
          df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1611 entries, 0 to 1610
Data columns (total 16 columns):
tweet_id                1611 non-null int64
```

```
source                  1611 non-null object
text                    1611 non-null object
rating_numerator        1611 non-null float64
rating_denominator      1611 non-null float64
dog_name                1111 non-null object
image_url               1611 non-null object
img_num                 1611 non-null float64
retweet_count           1611 non-null float64
favorite_count          1611 non-null float64
dog_breed_prediction    1611 non-null object
confidence_percentage   1611 non-null float64
date                    1611 non-null object
time                    1611 non-null object
multiple_stages         1611 non-null bool
dog_type                293 non-null object
dtypes: bool(1), float64(6), int64(1), object(8)
memory usage: 190.4+ KB
```

```
In [101]: # Convert columns to their appropriate types and set the timestamp as an index
          df['tweet_id'] = df['tweet_id'].astype(object)
          df['source'] = df['source'].astype('category')
          df['multiple_stages'] = df['multiple_stages'].astype('category')
          df['dog_type'] = df['dog_type'].astype('category')
```

```
In [102]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1611 entries, 0 to 1610
Data columns (total 16 columns):
tweet_id                1611 non-null object
source                  1611 non-null category
text                    1611 non-null object
rating_numerator        1611 non-null float64
rating_denominator      1611 non-null float64
dog_name                1111 non-null object
image_url               1611 non-null object
img_num                 1611 non-null float64
retweet_count           1611 non-null float64
favorite_count          1611 non-null float64
dog_breed_prediction    1611 non-null object
confidence_percentage   1611 non-null float64
date                    1611 non-null object
time                    1611 non-null object
multiple_stages         1611 non-null category
dog_type                293 non-null category
dtypes: category(3), float64(6), object(7)
memory usage: 169.1+ KB
```

```
In [103]: df['multiple_stages'].value_counts()

Out[103]: False    1611
          Name: multiple_stages, dtype: int64

In [104]: df.drop(columns = ['multiple_stages'], inplace = True)

In [105]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1611 entries, 0 to 1610
Data columns (total 15 columns):
tweet_id                 1611 non-null object
source                   1611 non-null category
text                     1611 non-null object
rating_numerator         1611 non-null float64
rating_denominator       1611 non-null float64
dog_name                 1111 non-null object
image_url                1611 non-null object
img_num                  1611 non-null float64
retweet_count            1611 non-null float64
favorite_count           1611 non-null float64
dog_breed_prediction     1611 non-null object
confidence_percentage    1611 non-null float64
date                     1611 non-null object
time                     1611 non-null object
dog_type                 293 non-null category
dtypes: category(2), float64(6), object(7)
memory usage: 167.4+ KB
```

Insights and Visualization:

1. The most popular dog type:

- By rating ratio
- By favorite count
- By retweet count

2. The most popular dog breed:

- By rating ratio
- By favorite count
- By retweet count

3. The relationship between retweets and favorites

1.1.22 The most popular dog type

```
In [106]: df['rating_ratio'] = df['rating_numerator'] / df['rating_denominator']

In [107]: df['dog_type'].value_counts()

Out[107]: pupper            191
          doggo              63
          puppo              20
          floofer             9
          doggo, pupper       8
          doggo, puppo        1
          doggo, floofer      1
          Name: dog_type, dtype: int64

In [108]: # Remove missing stages, the stages that only have one sample size, and outliers
          stage = df.query('dog_type !="" & dog_type != "doggo, puppo" & dog_type != "doggo, flo
          stage['dog_type'].value_counts()

Out[108]: pupper            191
          doggo              63
          puppo              20
          floofer             9
          doggo, pupper       8
          doggo, puppo        0
          doggo, floofer      0
          Name: dog_type, dtype: int64
```

1.1.23 By Rating Ratio

```
In [109]: stage.groupby('dog_type')['rating_ratio'].describe()

Out[109]:                count      mean       std  min    25%  50%  75%  max
          dog_type
          doggo           63.0  1.182540  0.137418  0.8  1.100  1.2  1.3  1.4
          doggo, floofer   0.0       NaN       NaN  NaN    NaN  NaN  NaN  NaN
          doggo, pupper    8.0  1.087500  0.253194  0.5  1.075  1.2  1.2  1.3
          doggo, puppo     0.0       NaN       NaN  NaN    NaN  NaN  NaN  NaN
          floofer          9.0  1.188889  0.105409  1.0  1.100  1.2  1.3  1.3
          pupper         191.0  1.067016  0.171713  0.3  1.000  1.1  1.2  1.4
          puppo           20.0  1.185000  0.130888  0.9  1.100  1.2  1.3  1.4
```

1.1.24 By Favorite Count

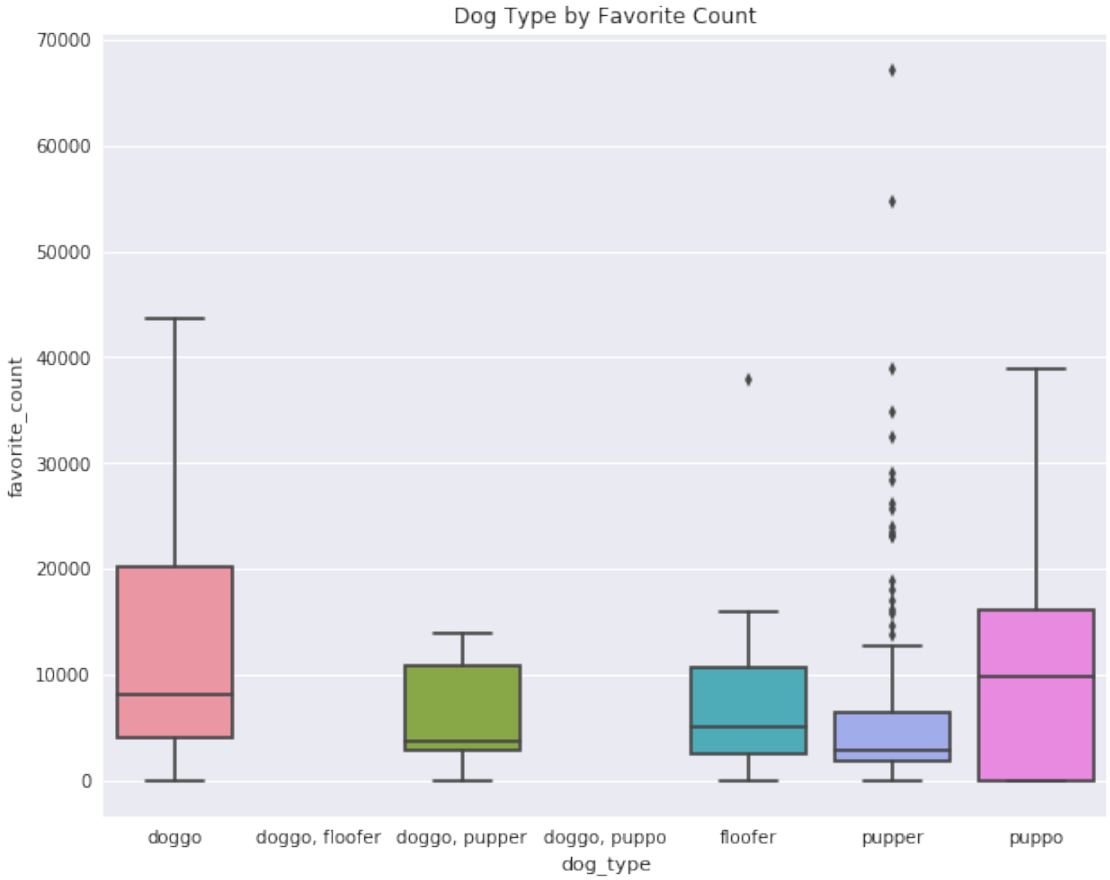
```
In [110]: stage.groupby('dog_type')['favorite_count'].describe()

Out[110]:                count          mean           std  min     25%     50%  \
          dog_type
          doggo           63.0  12175.317460  11419.241075  0.0  4037.5  8157.0
          doggo, floofer   0.0           NaN           NaN  NaN     NaN     NaN
```

```
doggo, pupper      8.0   6148.500000   5492.964097  0.0  2831.5  3795.5
doggo, puppo       0.0           NaN           NaN  NaN     NaN     NaN
floofer            9.0   9425.333333  11714.938796  0.0  2651.0  5094.0
pupper           191.0   6038.198953   8846.010750  0.0  1932.5  2890.0
puppo             20.0  11773.000000  11930.784180  0.0     0.0  9930.0

                     75%      max
dog_type
doggo            20181.00  43694.0
doggo, floofer        NaN      NaN
doggo, pupper    10913.50  14015.0
doggo, puppo          NaN      NaN
floofer          10681.00  37914.0
pupper            6429.50  67100.0
puppo            16072.25  38818.0
```

In [111]:
```
#Now, let's plot a boxplot
plt.figure(figsize=(10,8))
sns.set(style="darkgrid")
sns.boxplot(x='dog_type', y="favorite_count", data=stage).set_title('Dog Type by Favor
plt.savefig('stage_fav.png');
```



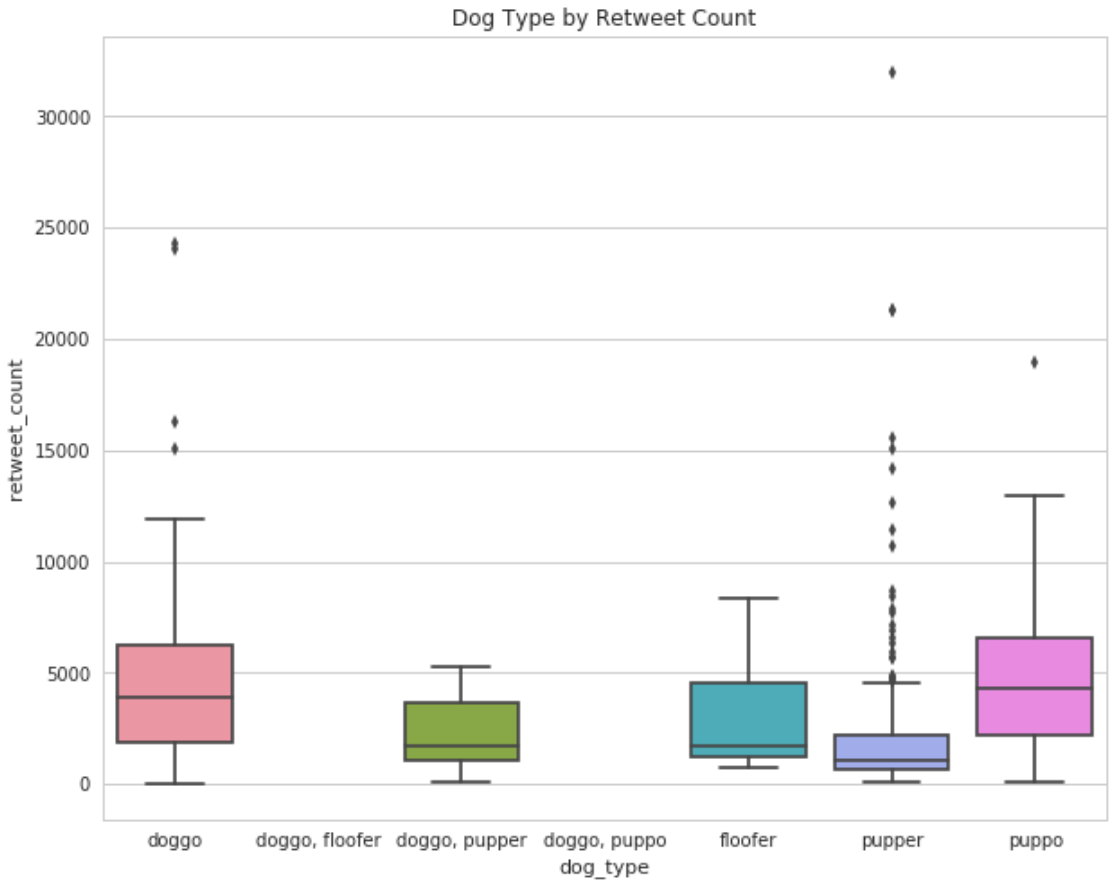
1.1.25 By Retweet Counts

In [112]:
```
stage.groupby('dog_type')['retweet_count'].describe()
```

Out[112]:

dog_type	count	mean	std	min	25%	50%	75%	max
doggo	63.0	4956.000000	4853.681343	10.0	1899.00	3884.0	6229.50	24319.0
doggo, floofer	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
doggo, pupper	8.0	2311.125000	1802.520254	113.0	1047.25	1730.5	3653.25	5247.0
doggo, puppo	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
floofer	9.0	3191.111111	2812.196448	764.0	1226.00	1714.0	4581.00	8329.0
pupper	191.0	2413.895288	3953.312856	97.0	640.50	1045.0	2241.00	31989.0
puppo	20.0	5207.800000	4608.195190	135.0	2175.25	4272.0	6611.50	18963.0

In [113]:
```
#Now, let's plot a boxplot
plt.figure(figsize=(10,8))
sns.set(style="whitegrid")
sns.boxplot(x='dog_type', y="retweet_count", data=stage).set_title('Dog Type by Retwee
plt.savefig('stag_fav.png');
```



As you can see, the most popular dog type is Pupper

1.1.26 The most popular dog breed

```
In [114]: df.dog_breed_prediction.value_counts().head(5)
```

```
Out[114]: Golden Retriever      165
          Labrador Retriever    102
          Chihuahua              88
          Pembroke               85
          Pug                    63
          Name: dog_breed_prediction, dtype: int64
```

```
In [115]: # Find out the top 5 breed that has the most images
          array = ['Golden_Retriever','Labrador_Retriever','Chihuahua','Pembroke','Pug']
          new_df = df.loc[df['dog_breed_prediction'].isin(array)]
```

1.1.27 By Rating Ratio

```
In [116]: new_df.groupby('dog_breed_prediction').rating_ratio.describe()
```

```
Out[116]:                       count      mean       std  min  25%  50%  75%  max
          dog_breed_prediction
          Chihuahua              88.0  1.170455  0.467854  0.2  1.1  1.2  1.3  5.0
          Pembroke               85.0  1.052941  0.215798  0.3  1.0  1.1  1.2  1.4
          Pug                    63.0  1.096825  0.194246  0.3  1.0  1.1  1.2  1.4

In [117]: new_df.groupby('dog_breed_prediction')['dog_type'].value_counts()

Out[117]: dog_breed_prediction  dog_type
          Chihuahua             pupper           7
                                doggo            3
                                puppo            2
                                doggo, pupper    1
          Pembroke              pupper           6
                                doggo            5
          Pug                   doggo            3
                                pupper           3
                                puppo            2
                                floofer          1
          Name: dog_type, dtype: int64
```

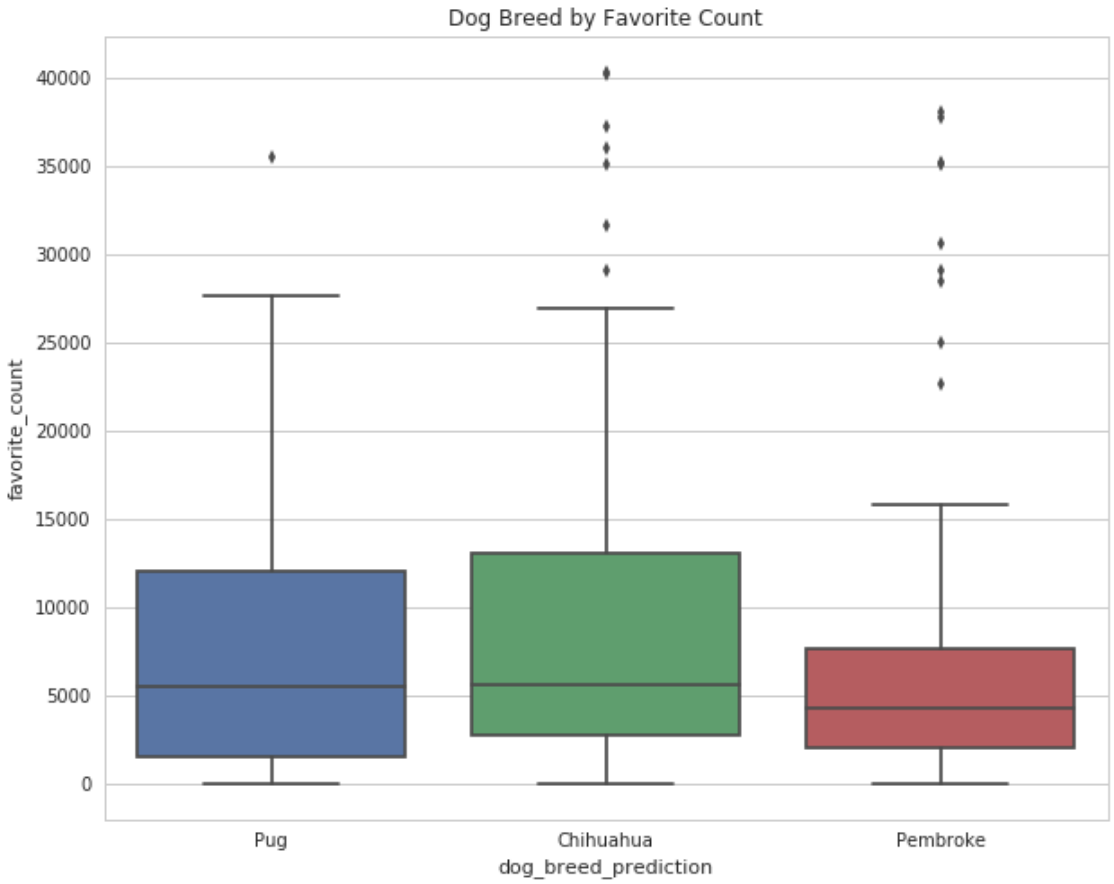
1.1.28 By Favorite Count

```
In [118]: new_df.groupby('dog_breed_prediction')['favorite_count'].describe()

Out[118]:                       count         mean           std  min      25%     50%  \
          dog_breed_prediction
          Chihuahua              88.0  9646.522727  10126.887452  0.0  2742.25  5665.5
          Pembroke               85.0  7362.988235   9112.917834  0.0  2056.00  4249.0
          Pug                    63.0  7529.539683   7676.342547  0.0  1580.00  5565.0

                                    75%      max
          dog_breed_prediction
          Chihuahua             13032.0  40325.0
          Pembroke               7620.0  38074.0
          Pug                   12065.5  35553.0

In [119]: plt.figure(figsize=(10,8))
          sns.boxplot(x='dog_breed_prediction', y="favorite_count", data=new_df).set_title('Dog
          plt.savefig('breed_fav.png');
```



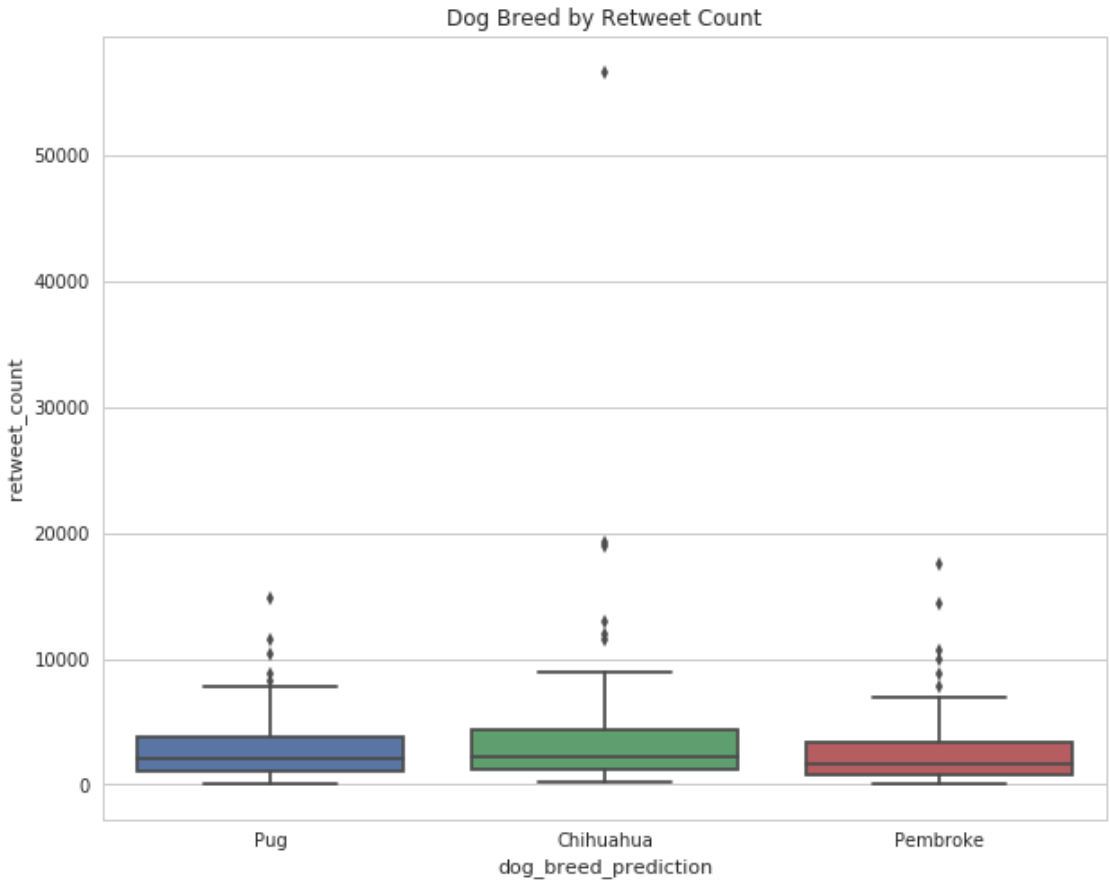
1.1.29 By Retweet Counts

```
In [120]: new_df.groupby('dog_breed_prediction')['retweet_count'].describe()
```

```
Out[120]:                        count         mean          std    min     25%     50%  \
          dog_breed_prediction
          Chihuahua               88.0  4019.977273  6750.211106  213.0  1158.5  2252.5
          Pembroke                85.0  2661.929412  3061.029751  119.0   786.0  1561.0
          Pug                     63.0  2960.174603  2923.306290   39.0   991.0  2013.0

                                    75%      max
          dog_breed_prediction
          Chihuahua              4294.0  56625.0
          Pembroke               3282.0  17465.0
          Pug                    3806.5  14740.0
```

```
In [121]: plt.figure(figsize=(10,8))
          sns.boxplot(x='dog_breed_prediction', y="retweet_count", data=new_df).set_title('Dog B
          plt.savefig('breed_retw.png');
```



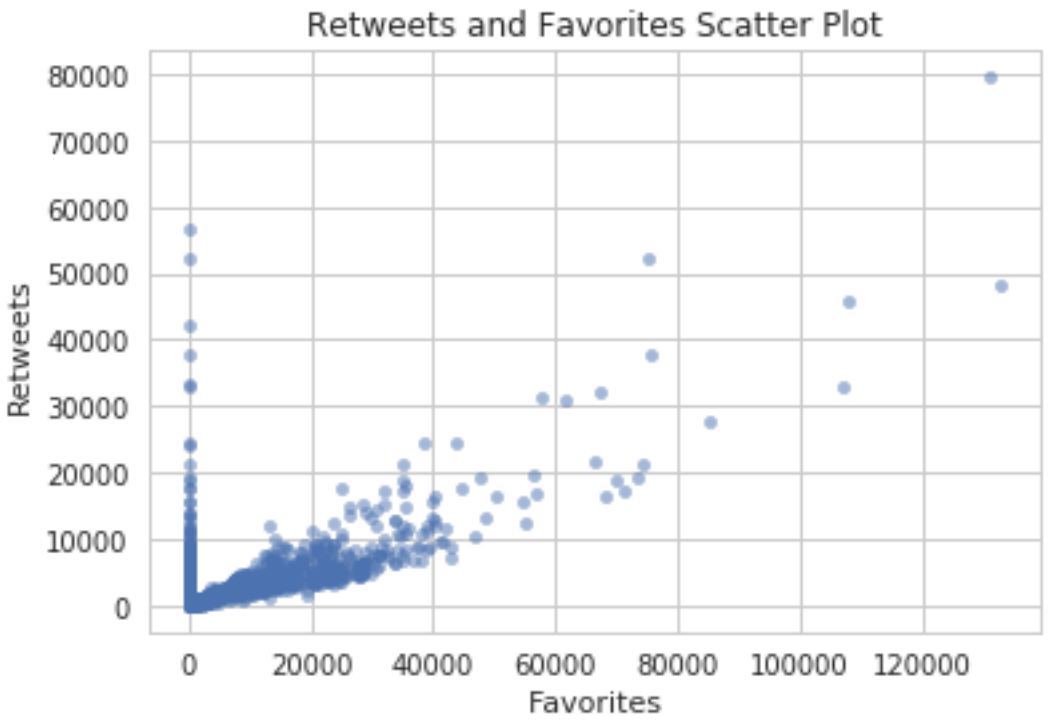
Now, you see that the most popular dog breed is Chihuahua

1.1.30 The relationship between retweets and favorites

In [122]:
```
#Retweets vs. Favorites

df.plot(kind='scatter',x='favorite_count',y='retweet_count', alpha = 0.5)
plt.xlabel('Favorites')
plt.ylabel('Retweets')
plt.title('Retweets and Favorites Scatter Plot')

plt.savefig('Retweets_vs_Favorites.png', bbox_inches='tight')
```



Here, we can see that Retweets are positively correlated with Favorites.

1.2 References:

- https://stackoverflow.com/questions/44327999/python-pandas-merge-multiple-dataframes/44338256
- https://stackoverflow.com/questions/14984119/python-pandas-remove-duplicate-columns
- Data Analysis Nanodegree/Data Wrangling/Lesson 3: Assessing Data/Concepts 4-18
- https://video.udacity-data.com/topher/2018/November/5be5fb7d_tweet-json/tweet-json.txt
- https://stackoverflow.com/questions/44594945/pandas-str-replace-regex-application
- https://pandas.pydata.org
- https://www.w3resource.com/pandas/series/series-str-islower.php
- https://docs.python.org
- https://thepythonguru.com
- https://ipython.org

- https://sebastianraschka.com/Articles/2014_ipython_internal_links.html#bottom
- https://medium.com/@sambozek/ipython-er-jupyter-table-of-contents-69bb72cf39d3