Project: Wrangle and Analyze WeRateDogs Data

Table of Contents

Introduction

About the data: three data sources are used in this project:

> **1**. The tweet achive of Twitter user @ dog_rates, also known as WeRatedogs. This archive contains basic tweet data (tweet ID, timestamp,text,etc.) for all 5000+ of their tweets until Aug 1,2017.
>
> **2**. Additional Data such as likes and retweets extracted via the Twitter API
>
> **3**. Predictions of dog breeds based on their images. This is done by running every image through a neural network classifier built by Udacity.

Task:

> **1**. Wrangle and analyze the WeRateDogs datasets
>
> **2**. Build at least three insights and create visualizations
>
> **3**. Write two reports. One as an internal document which describes the wrangling maneuvor. The other as a magazine post for external use, which communicates the findings.

Gather Data

In [2]:
```
#Import python libs for file downloads and data wrangling and analysis
import requests
import pandas as pd
import os
import tweepy
import numpy as np
import json
from timeit import default_timer as timer
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
#set pandas view option to see the entire text
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('display.max_colwidth', -1)
```

In [4]:
```
#Download the image prediction file from the web and save it to the folder
img_folder='image_pred'
if not os.path.exists(img_folder):
    os.makedirs(img_folder)

url='https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv'
r=requests.get(url)
with open(os.path.join(img_folder,url.split('/')[-1]),mode='wb') as f:
    f.write(r.content)
```

In [5]:
```
#Load personal API keys
consumer_key = ''
consumer_secret = ''
access_token = ''
access_secret = ''

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

api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True )
```

In [6]:
```
#Load the twitter archive file into the data frame
twitter_arch=pd.read_csv('twitter-archive-enhanced.csv')
```

In [7]:
```
#Load image_predictions file into the dataframe
image_pred=pd.read_csv('image-predictions.tsv',sep="\t")
```

In [ ]:
```
#Add each tweet to a new line of tweet_json.text
fails={}
start_time=timer()
count=0
with open('tweet_json.txt', 'w', encoding='utf8') as f:
    for tweet_id in twitter_arch['tweet_id']:
        count+=1
        print(str(count)+" : "+str(tweet_id))
        try:
            tweet = api.get_status(tweet_id, tweet_mode='extended')
            json.dump(tweet._json, f)
            f.write('\n')
        except tweepy.TweepError as e:
            fails[tweet_id]=e
            print('Fail'+str(tweet_id))
end_time=timer()
print(count)
print(end_time-start_time)
print(fails)
```

In [5]:
```
#Load tweets data into dataframe
tweets=pd.read_json('tweet_json.txt',lines='true')
```

Assess Data

image_pred dataset:

In [6]:
```
#Check columns,datatypes and null values
image_pred.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 [7]:
```
#Sample data from image_pred
image_pred.sample(10)
```

Out[7]:

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	
416	674019345211760640	https://pbs.twimg.com/media/CVqZBO8WUAAd931.jpg	1	collie	0.992732	True	borzoi	0.005043	True	Shetlan
1525	788412144018661376	https://pbs.twimg.com/media/CvEAqQoWgAADj5K.jpg	1	golden_retriever	0.805238	True	Labrador_retriever	0.113798	True	Brit
1892	849336543269576704	https://pbs.twimg.com/media/C8lzFC4XcAAQxB4.jpg	1	patio	0.521788	False	prison	0.149544	False	
1076	717421804990701568	https://pbs.twimg.com/media/CfTLUYWXEAEkyES.jpg	2	miniature_pinscher	0.286479	True	Italian_greyhound	0.084134	True	
1617	802323869084381190	https://pbs.twimg.com/media/CyJtSmDUAAA2F9x.jpg	4	home_theater	0.765069	False	television	0.203578	False	entertain
785	690248561355657216	https://pbs.twimg.com/media/CZRBZ9mWkAAWblt.jpg	1	motor_scooter	0.382690	False	moped	0.318017	False	
59	667119796878725120	https://pbs.twimg.com/media/CUIV6F7XIAA1tAM.jpg	1	Pembroke	0.741563	True	Chihuahua	0.057866	True	
1965	867421006826221569	https://pbs.twimg.com/media/DAmyy8FXYAIH8Ty.jpg	1	Eskimo_dog	0.616457	True	Siberian_husky	0.381330	True	
1696	816450570814898180	https://pbs.twimg.com/media/C1SddosXUAQcVR1.jpg	1	web_site	0.352857	False	envelope	0.060107	False	
533	676897532954456065	https://pbs.twimg.com/media/CWTSt0UW4AALMNB.jpg	1	hamster	0.628255	False	guinea_pig	0.318646	False	

In [8]:
```
image_pred.p1.value_counts()[:10]
```

Out[8]:
```
golden_retriever      150
Labrador_retriever    100
Pembroke              89
Chihuahua             83
pug                   57
chow                  44
Samoyed               43
toy_poodle            39
Pomeranian            38
malamute              30
Name: p1, dtype: int64
```

In [9]:

```
image_pred.p2.value_counts()[:10]
```

Out[9]:

```
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
Name: p2, dtype: int64
```

In [10]:

```
image_pred.p3.value_counts()[:10]
```

Out[10]:

```
Labrador_retriever            79
Chihuahua                     58
golden_retriever              48
Eskimo_dog                    38
kelpie                        35
kuvasz                        34
Staffordshire_bullterrier     32
chow                          32
beagle                        31
cocker_spaniel                31
Name: p3, dtype: int64
```

Quality issue 1: some breeds with the first letter capitalized, others not.

Quality issue 2: twitterID should be string not integer

Quality issue 3: column names are not informative.

In [11]:

```
#How many predictions are not dogs?
image_pred.p1_dog.value_counts()
```

Out[11]:

```
True     1532
False     543
Name: p1_dog, dtype: int64
```

In [12]:

```
image_pred.p2_dog.value_counts()
```

Out[12]:

```
True     1553
False     522
Name: p2_dog, dtype: int64
```

In [13]:

```
image_pred.p3_dog.value_counts()
```

Out[13]:

```
True     1499
False     576
Name: p3_dog, dtype: int64
```

`twitter_arch` dataset:

In [14]:

```
#Check columns,datatypes and null values
twitter_arch.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
```

Quality issue 4:Timestamp and Retweet timestamp should be datetime, not object. All ids should be string not float

In [15]:

```
#Sample the data and get some feeling about the quality and tidiness
twitter_arch.sample(10)
```

Out[15]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id
1490	692901601640583168	NaN	NaN	2016-01-29 02:46:29 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	"Fuck the system" 10/10 https://t.co/N0OADmCnVV	NaN
1686	681610798867845120	NaN	NaN	2015-12-28 23:00:52 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	*collapses* 12/10 https://t.co/C7M8mnzHIK	NaN
171	858860390427611136	NaN	NaN	2017-05-01 01:47:28 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	RT @dog_rates: Meet Winston. He knows he's a little too big for the swing, but he doesn't care. Kindly requests a push. 12/10 would happily…	8.395493e+17
985	749075273010798592	NaN	NaN	2016-07-02 03:00:36 +0000	<a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>	This is Boomer. He's self-baptizing. Other doggo not ready to renounce sins. 11/10 spiritually awakened af https://t.co/cRTJiQQk9o	NaN
903	758405701903519748	NaN	NaN	2016-07-27 20:56:24 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Odie. He falls asleep wherever he wants. Must be nice. 10/10 https://t.co/M9BXCSDVjh	NaN
477	815990720817401858	NaN	NaN	2017-01-02 18:38:42 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	Meet Jack. He's one of the rare doggos that doesn't mind baths. 11/10 click the link to see how you can help Jack!\n\nhttps://t.co/r4W111FzAq https://t.co/fQpYuMKG3p	NaN
1308	707297311098011648	NaN	NaN	2016-03-08 20:09:54 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Cassie. She can go from sweet to scary af in a matter of seconds. 10/10 points deducted for cats on pajamas https://t.co/B6dmZmJBdK	NaN
1612	685321586178670592	NaN	NaN	2016-01-08 04:46:13 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Rocky. He sleeps like a psychopath. 10/10 quality tongue slip https://t.co/MbgG95mUdu	NaN
980	749774190421639168	NaN	NaN	2016-07-04 01:17:51 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Lucy. She's a Benebop Cumberplop. 12/10 would hold against my face https://t.co/4yXa801fgl	NaN
544	805932879469572096	NaN	NaN	2016-12-06 00:32:26 +0000	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Major. He put on a tie for his first real walk. Only a little crooked. Can also drool upwards. H*ckin talented. 12/10 https://t.co/Zcwr8LgoO8	NaN

Tidiness issue 1: Dog stages should be one column with 5 possible outcomes (4 stages and None). Data type for the new column "stage" should be categorical

Quality issue 5: There are retweets which we don't need

Quality issue 6: There are many columns irrelevant to the analysis

In [16]:

```
#Check dog names
twitter_arch.name.unique()
```

```
Out[16]: array(['Phineas', 'Tilly', 'Archie', 'Darla', 'Franklin', 'None', 'Jax',
                'Zoey', 'Cassie', 'Koda', 'Bruno', 'Ted', 'Stuart', 'Oliver',
                'Jim', 'Zeke', 'Ralphus', 'Canela', 'Gerald', 'Jeffrey', 'such',
                'Maya', 'Mingus', 'Derek', 'Roscoe', 'Waffles', 'Jimbo', 'Maisey',
                'Lilly', 'Earl', 'Lola', 'Kevin', 'Yogi', 'Noah', 'Bella',
                'Grizzwald', 'Rusty', 'Gus', 'Stanley', 'Alfy', 'Koko', 'Rey',
                'Gary', 'a', 'Elliot', 'Louis', 'Jesse', 'Romeo', 'Bailey',
                'Duddles', 'Jack', 'Emmy', 'Steven', 'Beau', 'Snoopy', 'Shadow',
                'Terrance', 'Aja', 'Penny', 'Dante', 'Nelly', 'Ginger', 'Benedict',
                'Venti', 'Goose', 'Nugget', 'Cash', 'Coco', 'Jed', 'Sebastian',
                'Walter', 'Sierra', 'Monkey', 'Harry', 'Kody', 'Lassie', 'Rover',
                'Napolean', 'Dawn', 'Boomer', 'Cody', 'Rumble', 'Clifford',
                'quite', 'Dewey', 'Scout', 'Gizmo', 'Cooper', 'Harold', 'Shikha',
                'Jamesy', 'Lili', 'Sammy', 'Meatball', 'Paisley', 'Albus',
                'Neptune', 'Quinn', 'Belle', 'Zooey', 'Dave', 'Jersey', 'Hobbes',
                'Burt', 'Lorenzo', 'Carl', 'Jordy', 'Milky', 'Trooper', 'Winston',
                'Sophie', 'Wyatt', 'Rosie', 'Thor', 'Oscar', 'Luna', 'Callie',
                'Cermet', 'George', 'Marlee', 'Arya', 'Einstein', 'Alice',
                'Rumpole', 'Benny', 'Aspen', 'Jarod', 'Wiggles', 'General',
                'Sailor', 'Astrid', 'Iggy', 'Snoop', 'Kyle', 'Leo', 'Riley',
                'Gidget', 'Noosh', 'Odin', 'Jerry', 'Charlie', 'Georgie', 'Rontu',
                'Cannon', 'Furzey', 'Daisy', 'Tuck', 'Barney', 'Vixen', 'Jarvis',
                'Mimosa', 'Pickles', 'Bungalo', 'Brady', 'Margo', 'Sadie', 'Hank',
                'Tycho', 'Stephan', 'Indie', 'Winnie', 'Bentley', 'Ken', 'Max',
                'Maddie', 'Pipsy', 'Monty', 'Sojourner', 'Odie', 'Arlo', 'Sunny',
                'Vincent', 'Lucy', 'Clark', 'Mookie', 'Meera', 'Buddy', 'Ava',
                'Rory', 'Eli', 'Ash', 'Tucker', 'Tobi', 'Chester', 'Wilson',
                'Sunshine', 'Lipton', 'Gabby', 'Bronte', 'Poppy', 'Rhino',
                'Willow', 'not', 'Orion', 'Eevee', 'Smiley', 'Logan', 'Moreton',
                'Klein', 'Miguel', 'Emanuel', 'Kuyu', 'Dutch', 'Pete', 'Scooter',
                'Reggie', 'Kyro', 'Samson', 'Loki', 'Mia', 'Malcolm', 'Dexter',
                'Alfie', 'Fiona', 'one', 'Mutt', 'Bear', 'Doobert', 'Beebop',
                'Alexander', 'Sailer', 'Brutus', 'Kona', 'Boots', 'Ralphie',
                'Phil', 'Cupid', 'Pawnd', 'Pilot', 'Ike', 'Mo', 'Toby', 'Sweet',
                'Pablo', 'Nala', 'Balto', 'Crawford', 'Gabe', 'Mattie', 'Jimison',
                'Hercules', 'Duchess', 'Harlso', 'Sampson', 'Sundance', 'Luca',
                'Flash', 'Finn', 'Peaches', 'Howie', 'Jazzy', 'Anna', 'Bo',
                'Seamus', 'Wafer', 'Chelsea', 'Tom', 'Moose', 'Florence', 'Autumn',
                'Dido', 'Eugene', 'Herschel', 'Strudel', 'Tebow', 'Chloe', 'Betty',
                'Timber', 'Binky', 'Dudley', 'Comet', 'Larry', 'Levi', 'Akumi',
                'Titan', 'Olivia', 'Alf', 'Oshie', 'Bruce', 'Chubbs', 'Sky',
                'Atlas', 'Eleanor', 'Layla', 'Rocky', 'Baron', 'Tyr', 'Bauer',
                'Swagger', 'Brandi', 'Mary', 'Moe', 'Halo', 'Augie', 'Craig',
                'Sam', 'Hunter', 'Pavlov', 'Maximus', 'Wallace', 'Ito', 'Milo',
                'Ollie', 'Cali', 'Lennon', 'incredibly', 'Major', 'Duke',
                'Reginald', 'Sansa', 'Shooter', 'Django', 'Diogi', 'Sonny',
                'Philbert', 'Marley', 'Severus', 'Ronnie', 'Anakin', 'Bones',
                'Mauve', 'Chef', 'Doc', 'Sobe', 'Longfellow', 'Mister', 'Iroh',
                'Baloo', 'Stubert', 'Paull', 'Timison', 'Davey', 'Pancake',
                'Tyrone', 'Snicku', 'Ruby', 'Brody', 'Rizzy', 'Mack', 'Butter',
                'Nimbus', 'Laika', 'Dobby', 'Juno', 'Maude', 'Lily', 'Newt',
                'Benji', 'Nida', 'Robin', 'Monster', 'BeBe', 'Remus', 'Mabel',
                'Misty', 'Happy', 'Mosby', 'Maggie', 'Leela', 'Ralphy', 'Brownie',
                'Meyer', 'Stella', 'mad', 'Frank', 'Tonks', 'Lincoln', 'Oakley',
                'Dale', 'Rizzo', 'Arnie', 'Pinot', 'Dallas', 'Hero', 'Frankie',
                'Stormy', 'Mairi', 'Loomis', 'Godi', 'Kenny', 'Deacon', 'Timmy',
                'Harper', 'Chipson', 'Combo', 'Dash', 'Bell', 'Hurley', 'Jay',
                'Mya', 'Strider', 'an', 'Wesley', 'Solomon', 'Huck', 'very', 'O',
                'Blue', 'Finley', 'Sprinkles', 'Heinrich', 'Shakespeare', 'Fizz',
                'Chip', 'Grey', 'Roosevelt', 'Gromit', 'Willem', 'Dakota', 'Dixie',
                'Al', 'Jackson', 'just', 'Carbon', 'DonDon', 'Kirby', 'Lou',
                'Nollie', 'Chevy', 'Tito', 'Louie', 'Rupert', 'Rufus', 'Brudge',
                'Shadoe', 'Colby', 'Angel', 'Brat', 'Tove', 'my', 'Aubie', 'Kota',
                'Eve', 'Glenn', 'Shelby', 'Sephie', 'Bonaparte', 'Albert',
                'Wishes', 'Rose', 'Theo', 'Rocco', 'Fido', 'Emma', 'Spencer',
                'Lilli', 'Boston', 'Brandonald', 'Corey', 'Leonard', 'Chompsky',
                'Beckham', 'Devón', 'Gert', 'Watson', 'Rubio', 'Keith', 'Dex',
                'Carly', 'Ace', 'Tayzie', 'Grizzie', 'Fred', 'Gilbert', 'Zoe',
                'Stewie', 'Calvin', 'Lilah', 'Spanky', 'Jameson', 'Piper',
                'Atticus', 'Blu', 'Dietrich', 'Divine', 'Tripp', 'his', 'Cora',
                'Huxley', 'Keurig', 'Bookstore', 'Linus', 'Abby', 'Shaggy',
                'Shiloh', 'Gustav', 'Arlen', 'Percy', 'Lenox', 'Sugar', 'Harvey',
                'Blanket', 'actually', 'Geno', 'Stark', 'Beya', 'Kilo', 'Kayla',
                'Maxaroni', 'Doug', 'Edmund', 'Aqua', 'Theodore', 'Chase',
                'getting', 'Rorie', 'Simba', 'Charles', 'Bayley', 'Axel',
                'Storkson', 'Remy', 'Chadrick', 'Kellogg', 'Buckley', 'Livvie',
                'Terry', 'Hermione', 'Ralpher', 'Aldrick', 'this', 'unacceptable',
                'Rooney', 'Crystal', 'Ziva', 'Stefan', 'Pupcasso', 'Puff',
                'Flurpson', 'Coleman', 'Enchilada', 'Raymond', 'all', 'Rueben',
                'Cilantro', 'Karll', 'Sprout', 'Blitz', 'Bloop', 'Lillie',
                'Ashleigh', 'Kreggory', 'Sarge', 'Luther', 'Ivar', 'Jangle',
                'Schnitzel', 'Panda', 'Berkeley', 'Ralphé', 'Charleson', 'Clyde',
                'Harnold', 'Sid', 'Pippa', 'Otis', 'Carper', 'Bowie',
                'Alexanderson', 'Suki', 'Barclay', 'Skittle', 'Ebby', 'Flávio',
                'Smokey', 'Link', 'Jennifur', 'Ozzy', 'Bluebert', 'Stephanus',
                'Bubbles', 'old', 'Zeus', 'Bertson', 'Nico', 'Michelangelope',
                'Siba', 'Calbert', 'Curtis', 'Travis', 'Thumas', 'Kanu', 'Lance',
                'Opie', 'Kane', 'Olive', 'Chuckles', 'Staniel', 'Sora', 'Beemo',
                'Gunner', 'infuriating', 'Lacy', 'Tater', 'Olaf', 'Cecil', 'Vince',
                'Karma', 'Billy', 'Walker', 'Rodney', 'Klevin', 'Malikai',
                'Bobble', 'River', 'Jebberson', 'Remington', 'Farfle', 'Jiminus',
                'Clarkus', 'Finnegus', 'Cupcake', 'Kathmandu', 'Ellie', 'Katie',
                'Kara', 'Adele', 'Zara', 'Ambrose', 'Jimothy', 'Bode', 'Terrenth',
                'Reese', 'Chesterson', 'Lucia', 'Bisquick', 'Ralphson', 'Socks',
                'Rambo', 'Rudy', 'Fiji', 'Rilo', 'Bilbo', 'Coopson', 'Yoda',
                'Millie', 'Chet', 'Crouton', 'Daniel', 'Kaia', 'Murphy', 'Dotsy',
                'Eazy', 'Coops', 'Fillup', 'Miley', 'Charl', 'Reagan', 'Yukon',
                'CeCe', 'Cuddles', 'Claude', 'Jessiga', 'Carter', 'Ole', 'Pherb',
                'Blipson', 'Reptar', 'Trevith', 'Berb', 'Bob', 'Colin', 'Brian',
                'Oliviér', 'Grady', 'Kobe', 'Freddery', 'Bodie', 'Dunkin', 'Wally',
                'Tupawc', 'Amber', 'Edgar', 'Teddy', 'Kingsley', 'Brockly',
                'Richie', 'Molly', 'Vinscent', 'Cedrick', 'Hazel', 'Lolo', 'Eriq',
       'Phred', 'the', 'Oddie', 'Maxwell', 'Geoff', 'Covach', 'Durg',
       'Fynn', 'Ricky', 'Herald', 'Lucky', 'Ferg', 'Trip', 'Clarence',
       'Hamrick', 'Brad', 'Pubert', 'Frönq', 'Derby', 'Lizzie', 'Ember',
       'Blakely', 'Opal', 'Marq', 'Kramer', 'Barry', 'Gordon', 'Baxter',
       'Mona', 'Horace', 'Crimson', 'Birf', 'Hammond', 'Lorelei', 'Marty',
       'Brooks', 'Petrick', 'Hubertson', 'Gerbald', 'Oreo', 'Bruiser',
       'Perry', 'Bobby', 'Jeph', 'Obi', 'Tino', 'Kulet', 'Sweets', 'Lupe',
       'Tiger', 'Jiminy', 'Griffin', 'Banjo', 'Brandy', 'Lulu', 'Darrel',
       'Taco', 'Joey', 'Patrick', 'Kreg', 'Todo', 'Tess', 'Ulysses',
       'Toffee', 'Apollo', 'Asher', 'Glacier', 'Chuck', 'Champ', 'Ozzie',
       'Griswold', 'Cheesy', 'Moofasa', 'Hector', 'Goliath', 'Kawhi',
       'by', 'Emmie', 'Penelope', 'Willie', 'Rinna', 'Mike', 'William',
       'Dwight', 'Evy', 'officially', 'Rascal', 'Linda', 'Tug', 'Tango',
       'Grizz', 'Jerome', 'Crumpet', 'Jessifer', 'Izzy', 'Ralph', 'Sandy',
       'Humphrey', 'Tassy', 'Juckson', 'Chuq', 'Tyrus', 'Karl',
       'Godzilla', 'Vinnie', 'Kenneth', 'Herm', 'Bert', 'Striker',
       'Donny', 'Pepper', 'Bernie', 'Buddah', 'Lenny', 'Arnold', 'Zuzu',
       'Mollie', 'Laela', 'Tedders', 'Superpup', 'Rufio', 'Jeb', 'Rodman',
       'Jonah', 'Chesney', 'life', 'Henry', 'Bobbay', 'Mitch', 'Kaiya',
       'Acro', 'Aiden', 'Obie', 'Dot', 'Shnuggles', 'Kendall', 'Jeffri',
       'Steve', 'Mac', 'Fletcher', 'Kenzie', 'Pumpkin', 'Schnozz',
       'Gustaf', 'Cheryl', 'Ed', 'Leonidas', 'Norman', 'Caryl', 'Scott',
       'Taz', 'Darby', 'Jackie', 'light', 'Jazz', 'Franq', 'Pippin',
       'Rolf', 'Snickers', 'Ridley', 'Cal', 'Bradley', 'Bubba', 'Tuco',
       'Patch', 'Mojo', 'Batdog', 'Dylan', 'space', 'Mark', 'JD',
       'Alejandro', 'Scruffers', 'Pip', 'Julius', 'Tanner', 'Sparky',
       'Anthony', 'Holly', 'Jett', 'Amy', 'Sage', 'Andy', 'Mason',
       'Trigger', 'Antony', 'Creg', 'Traviss', 'Gin', 'Jeffrie', 'Danny',
       'Ester', 'Pluto', 'Bloo', 'Edd', 'Willy', 'Herb', 'Damon',
       'Peanut', 'Nigel', 'Butters', 'Sandra', 'Fabio', 'Randall', 'Liam',
       'Tommy', 'Ben', 'Raphael', 'Julio', 'Andru', 'Kloey', 'Shawwn',
       'Skye', 'Kollin', 'Ronduh', 'Billl', 'Saydee', 'Dug', 'Tessa',
       'Sully', 'Kirk', 'Ralf', 'Clarq', 'Jaspers', 'Samsom', 'Harrison',
       'Chaz', 'Jeremy', 'Jaycob', 'Lambeau', 'Ruffles', 'Amélie', 'Bobb',
       'Banditt', 'Kevon', 'Winifred', 'Hanz', 'Churlie', 'Zeek',
       'Timofy', 'Maks', 'Jomathan', 'Kallie', 'Marvin', 'Spark',
       'Gòrdón', 'Jo', 'DayZ', 'Jareld', 'Torque', 'Ron', 'Skittles',
       'Cleopatricia', 'Erik', 'Stu', 'Tedrick', 'Filup', 'Kial',
       'Naphaniel', 'Dook', 'Hall', 'Philippe', 'Biden', 'Fwed',
       'Genevieve', 'Joshwa', 'Bradlay', 'Clybe', 'Keet', 'Carll',
       'Jockson', 'Josep', 'Lugan', 'Christoper'], dtype=object)
```

Quality issue 7: Some names are captured wrong. Get a list of names without the first letter capitalized. They don't look like dog names.

In [17]:

```
nl=list(twitter_arch.name.unique())
name_error=[]
for n in nl:
    if n[0].islower():
        name_error.append(n)
        print(n)
```

```
such
a
quite
not
one
incredibly
mad
an
very
just
my
his
actually
getting
this
unacceptable
all
old
infuriating
the
by
officially
life
light
space
```

In [18]:

```
#Double check why the name is wrong. These are tweets without a name specified. They should be None instead
twitter_arch[twitter_arch.name.isin(name_error)][['name','text']]
```

Out[18]:

	name	text
22	such	I've yet to rate a Venezuelan Hover Wiener. This is such an honor. 14/10 paw-inspiring af (IG: roxy.thedoxy) https://t.co/20VrLAA8ba
56	a	Here is a pupper approaching maximum borkdrive. Zooming at never before seen speeds. 14/10 paw-inspiring af \n(IG: puffie_the_chow) https://t.co/ghXBIIeQZF
118	quite	RT @dog_rates: We only rate dogs. This is quite clearly a smol broken polar bear. We'd appreciate if you only send dogs. Thank you... 12/10…
169	quite	We only rate dogs. This is quite clearly a smol broken polar bear. We'd appreciate if you only send dogs. Thank you... 12/10 https://t.co/g2nSyGenG9
193	quite	Guys, we only rate dogs. This is quite clearly a bulbasaur. Please only send dogs. Thank you... 12/10 human used pet, it's super effective https://t.co/Xc7uj1C64x
335	not	There's going to be a dog terminal at JFK Airport. This is not a drill. 10/10 \nhttps://t.co/dp5h9bCwU7
369	one	Occasionally, we're sent fantastic stories. This is one of them. 14/10 for Grace https://t.co/bZ4axuH6OK
542	incredibly	We only rate dogs. Please stop sending in non-canines like this Freudian Poof Lion. This is incredibly frustrating... 11/10 https://t.co/IZidSrBvhi
649	a	Here is a perfect example of someone who has their priorities in order. 13/10 for both owner and Forrest https://t.co/LRyMrU7Wfq
682	mad	RT @dog_rates: Say hello to mad pupper. You know what you did. 13/10 would pet until no longer furustrated https://t.co/u1ulQ5heLX
759	an	RT @dog_rates: This is an East African Chalupa Seal. We only rate dogs. Please only send in dogs. Thank you... 10/10 https://t.co/iHe6liLwWR
773	very	RT @dog_rates: We only rate dogs. Pls stop sending in non-canines like this Mongolian grass snake. This is very frustrating. 11/10 https://…
801	a	Guys this is getting so out of hand. We only rate dogs. This is a Galapagos Speed Panda. Pls only send dogs... 10/10 https://t.co/8lpAGaZRFn
819	very	We only rate dogs. Pls stop sending in non-canines like this Arctic Floof Kangaroo. This is very frustrating. 11/10 https://t.co/qlUDuPoE3d
822	just	RT @dog_rates: This is just downright precious af. 12/10 for both pupper and doggo https://t.co/o5J479bZUC
852	my	This is my dog. Her name is Zoey. She knows I've been rating other dogs. She's not happy. 13/10 no bias at all https://t.co/ep1NkYoiwB
924	one	This is one of the most inspirational stories I've ever come across. I have no words. 14/10 for both doggo and owner https://t.co/I5ld3eKD5k
988	not	What jokester sent in a pic without a dog in it? This is not @rock_rates. This is @dog_rates. Thank you ...10/10 https://t.co/nDPaYHrtNX
992	his	That is Quizno. This is his beach. He does not tolerate human shenanigans on his beach. 10/10 reclaim ur land doggo https://t.co/vdr7DaRSa7
993	one	This is one of the most reckless puppers I've ever seen. How she got a license in the first place is beyond me. 6/10 https://t.co/z5bAdtn9kd
1002	a	This is a mighty rare blue-tailed hammer sherk. Human almost lost a limb trying to take these. Be careful guys. 8/10 https://t.co/TGenMeXreW
1004	a	Viewer discretion is advised. This is a terrible attack in progress. Not even in water (tragic af). 4/10 bad sherk https://t.co/L3U0j14N5R
1017	a	This is a carrot. We only rate dogs. Please only send in dogs. You all really should know this by now ...11/10 https://t.co/9e48aPrBm2
1025	an	This is an Iraqi Speed Kangaroo. It is not a dog. Please only send in dogs. I'm very angry with all of you ...9/10 https://t.co/5qpBTTpgUt
1031	very	We only rate dogs. Pls stop sending in non-canines like this Jamaican Flop Seal. This is very very frustrating. 9/10 https://t.co/nc53zEN0hZ
1040	actually	This is actually a pupper and I'd pet it so well. 12/10\nhttps://t.co/RNqS7C4Y4N
1049	a	This is a very rare Great Alaskan Bush Pupper. Hard to stumble upon without spooking. 12/10 would pet passionately https://t.co/xOBKCdpzaa
1063	just	This is just downright precious af. 12/10 for both pupper and doggo https://t.co/o5J479bZUC
1071	getting	This is getting incredibly frustrating. This is a Mexican Golden Beaver. We only rate dogs. Only send dogs ...10/10 https://t.co/0yolOOyD3X
1095	mad	Say hello to mad pupper. You know what you did. 13/10 would pet until no longer furustrated https://t.co/u1ulQ5heLX
1097	very	We only rate dogs. Please stop sending in non-canines like this Alaskan Flop Turtle. This is very frustrating. 10/10 https://t.co/qXteK6Atxc
1120	this	Say hello to this unbelievably well behaved squad of doggos. 204/170 would try to pet all at once https://t.co/yGQI3He3xv
1121	unacceptable	We only rate dogs. Pls stop sending non-canines like this Bulgarian Eyeless Porch Bear. This is unacceptable... 9/10 https://t.co/2yctWAUZ3Z
1138	all	This is all I want in my life. 12/10 for super sleepy pupper https://t.co/4RlLA5ObMh
1193	a	People please. This is a Deadly Mediterranean Plop T-Rex. We only rate dogs. Only send in dogs. Thanks you... 11/10 https://t.co/2ATDsgHD4n
1206	old	This is old now but it's absolutely heckin fantastic and I can't not share it with you all. 13/10 https://t.co/wJX74TSgzP
1207	a	This is a taco. We only rate dogs. Please only send in dogs. Dogs are what we rate. Not tacos. Thank you... 10/10 https://t.co/cxl6xGY8B9
1259	infuriating	We 👏🏻 only 👏🏻 rate 👏🏻 dogs. Pls stop sending in non-canines like this Dutch Panda Worm. This is infuriating. 11/10 https://t.co/odfLzBonG2
1340	a	Here is a heartbreaking scene of an incredible pupper being laid to rest. 10/10 RIP pupper https://t.co/81mvJ0rGRu
1351	a	Here is a whole flock of puppers. 60/50 I'll take the lot https://t.co/9dpcw6MdWa
1361	a	This is a Butternut Cumberfloof. It's not windy they just look like that. 11/10 back at it again with the red socks https://t.co/hMjzhdUHaW
1362	an	This is an East African Chalupa Seal. We only rate dogs. Please only send in dogs. Thank you... 10/10 https://t.co/iHe6liLwWR
1368	a	This is a Wild Tuscan Poofwiggle. Careful not to startle. Rare tongue slip. One eye magical. 12/10 would def pet https://t.co/4EnShAQjv6
1382	a	"Pupper is a present to world. Here is a bow for pupper." 12/10 precious as hell https://t.co/ItSsE92gCW
1385	very	We only rate dogs. Pls stop sending in non-canines like this Mongolian grass snake. This is very frustrating. 11/10 https://t.co/22x9SbCYCU
1435	getting	Please stop sending in saber-toothed tigers. This is getting ridiculous. We only rate dogs.\n...8/10 https://t.co/iAeQNueou8
1457	just	This is just a beautiful pupper good shit evolution. 12/10 https://t.co/2L8pI0Z2Ib
1499	a	This is a rare Arctic Wubberfloof. Unamused by the happenings. No longer has the appetites. 12/10 would totally hug https://t.co/krvbacIX0N
1527	the	Stop sending in lobsters. This is the final warning. We only rate dogs. Thank you... 9/10 https://t.co/B9ZXXKJYNx
1603	the	This is the newly formed pupper a capella group. They're just starting out but I see tons of potential. 8/10 for all https://t.co/wbAcvFoNtn
1693	actually	This is actually a lion. We only rate dogs. For the last time please only send dogs. Thank u.\n12/10 would still pet https://t.co/Pp26dMQxap
1724	by	This is by far the most coordinated series of pictures I was sent. Downright impressive in every way. 12/10 for all https://t.co/etzLo3sdZE
1737	a	Guys this really needs to stop. We've been over this way too many times. This is a giraffe. We only rate dogs.. 7/10 https://t.co/yavgkHYPOC
1747	officially	This is officially the greatest yawn of all time. 12/10 https://t.co/4R0Cc0sLVE
1785	a	This is a dog swinging. I really enjoyed it so I hope you all do as well. 11/10 https://t.co/Ozo9KHTRND
1797	the	This is the happiest pupper I've ever seen. 10/10 would trade lives with https://t.co/ep8ATEJwRb
1815	the	This is the saddest/sweetest/best picture I've been sent. 12/10 😢🐶 https://t.co/vQ2Lw1BLBF
1853	a	This is a Sizzlin Menorah spaniel from Brooklyn named Wylie. Lovable eyes. Chiller as hell. 10/10 and I'm out.. poof https://t.co/7E0AiJXPmI
1854	a	Seriously guys?! Only send in dogs. I only rate dogs. This is a baby black bear... 11/10 https://t.co/H7kpabTfLj
1877	a	C'mon guys. We've been over this. We only rate dogs. This is a cow. Please only submit dogs. Thank you...... 9/10 https://t.co/WjcELNEqN2
1878	a	This is a fluffy albino Bacardi Columbia mix. Excellent at the tweets. 11/10 would hug gently https://t.co/diboDRUuEI
1916	life	This is life-changing. 12/10 https://t.co/SroTpI6psB
1923	a	This is a Sagitariot Baklava mix. Loves her new hat. 11/10 radiant pup https://t.co/Bko5kFJYUU
1936	one	This is one esteemed pupper. Just graduated college. 10/10 what a champ https://t.co/nyReCVRiyd

	name	text
1941	a	This is a heavily opinionated dog. Loves walls. Nobody knows how the hair works. Always ready for a kiss. 4/10 https://t.co/dFiaKZ9cDl
1955	a	This is a Lofted Aphrodisiac Terrier named Kip. Big fan of bed n breakfasts. Fits perfectly. 10/10 would pet firmly https://t.co/gKlLpNzIl3
1994	a	This is a baby Rand Paul. Curls for days. 11/10 would cuddle the hell out of https://t.co/xHXNaPAYRe
2001	light	This is light saber pup. Ready to fight off evil with light saber. 10/10 true hero https://t.co/LPPa3btIIt
2019	just	This is just impressive I have nothing else to say. 11/10 https://t.co/LquQZiZjJP
2030	space	This is space pup. He's very confused. Tries to moonwalk at one point. Super spiffy uniform. 13/10 I love space pup https://t.co/SfPQ2KeLdq
2034	a	This is a Tuscaloosa Alcatraz named Jacob (Yacōb). Loves to sit in swing. Stellar tongue. 11/10 look at his feet https://t.co/2IslQ8ZSc7
2037	the	This is the best thing I've ever seen so spread it like wildfire &amp; maybe we'll find the genius who created it. 13/10 https://t.co/q6RsuOVYwU
2066	a	This is a Helvetica Listerine named Rufus. This time Rufus will be ready for the UPS guy. He'll never expect it 9/10 https://t.co/34OhVhMkVr
2116	a	This is a Deciduous Trimester mix named Spork. Only 1 ear works. No seat belt. Incredibly reckless. 9/10 still cute https://t.co/CtuJoLHiDo
2125	a	This is a Rich Mahogany Seltzer named Cherokee. Just got destroyed by a snowball. Isn't very happy about it. 9/10 https://t.co/98ZBi6o4dj
2128	a	This is a Speckled Cauliflower Yosemite named Hemry. He's terrified of intruder dog. Not one bit comfortable. 9/10 https://t.co/yV3Qgjh8iN
2146	a	This is a spotted Lipitor Rumpelstiltskin named Alphred. He can't wait for the Turkey. 10/10 would pet really well https://t.co/6GUGO7azNX
2153	a	This is a brave dog. Excellent free climber. Trying to get closer to God. Not very loyal though. Doesn't bark. 5/10 https://t.co/ODnILTr4QM
2161	a	This is a Coriander Baton Rouge named Alfredo. Loves to cuddle with smaller well-dressed dog. 10/10 would hug lots https://t.co/eCRdwouKCl
2191	a	This is a Slovakian Helter Skelter Feta named Leroi. Likes to skip on roofs. Good traction. Much balance. 10/10 wow! https://t.co/Dmy2mY2Qj5
2198	a	This is a wild Toblerone from Papua New Guinea. Mouth always open. Addicted to hay. Acts blind. 7/10 handsome dog https://t.co/IGmVbz07tZ
2204	an	This is an Irish Rigatoni terrier named Berta. Completely made of rope. No eyes. Quite large. Loves to dance. 10/10 https://t.co/EM5fDykrJg
2211	a	Here is a horned dog. Much grace. Can jump over moons (dam!). Paws not soft. Bad at barking. 7/10 can still pet tho https://t.co/2Su7gmsnZm
2212	the	Never forget this vine. You will not stop watching for at least 15 minutes. This is the second coveted.. 13/10 https://t.co/roqIxCvEB3
2218	a	This is a Birmingham Quagmire named Chuk. Loves to relax and watch the game while sippin on that iced mocha. 10/10 https://t.co/HvNg9JWxFt
2222	a	Here is a mother dog caring for her pups. Snazzy red mohawk. Doesn't wag tail. Pups look confused. Overall 4/10 https://t.co/YOHe6lf09m
2235	a	This is a Trans Siberian Kellogg named Alfonso. Huge ass eyeballs. Actually Dobby from Harry Potter. 7/10 https://t.co/XpseHBlAAb
2249	a	This is a Shotokon Macadamia mix named Cheryl. Sophisticated af. Looks like a disappointed librarian. Shh (lol) 9/10 https://t.co/J4GnJ5Swba
2255	a	This is a rare Hungarian Pinot named Jessiga. She is either mid-stroke or got stuck in the washing machine. 8/10 https://t.co/ZU0i0KJyqD
2264	a	This is a southwest Coriander named Klint. Hat looks expensive. Still on house arrest :(\n9/10 https://t.co/IQTOMqDUIe
2273	a	This is a northern Wahoo named Kohl. He runs this town. Chases tumbleweeds. Draws gun wicked fast. 11/10 legendary https://t.co/J4vn2rOYFk
2287	a	This is a Dasani Kingfisher from Maine. His name is Daryl. Daryl doesn't like being swallowed by a panda. 8/10 https://t.co/jpaeu6LNmW
2304	a	This is a curly Ticonderoga named Pepe. No feet. Loves to jet ski. 11/10 would hug until forever https://t.co/cyDfaK8NBc
2311	a	This is a purebred Bacardi named Octaviath. Can shoot spaghetti out of mouth. 10/10 https://t.co/uEvsGLOFHa
2314	a	This is a golden Buckminsterfullerene named Johm. Drives trucks. Lumberjack (?). Enjoys wall. 8/10 would hug softly https://t.co/uQbZJM2DQB
2326	quite	This is quite the dog. Gets really excited when not in water. Not very soft tho. Bad at fetch. Can't do tricks. 2/10 https://t.co/aMCTNWO94t
2327	a	This is a southern Vesuvius bumblegruff. Can drive a truck (wow). Made friends with 5 other nifty dogs (neat). 7/10 https://t.co/LopTBkKa8h
2333	an	This is an extremely rare horned Parthenon. Not amused. Wears shoes. Overall very nice. 9/10 would pet aggressively https://t.co/QpRjllzWAL
2334	a	This is a funny dog. Weird toes. Won't come down. Loves branch. Refuses to eat his food. Hard to cuddle with. 3/10 https://t.co/IIXis0zta0
2335	an	This is an Albanian 3 1/2 legged Episcopalian. Loves well-polished hardwood flooring. Penis on the collar. 9/10 https://t.co/d9NcXFKwLv
2345	the	This is the happiest dog you will ever see. Very committed owner. Nice couch. 10/10 https://t.co/RhUEAloehK
2346	the	Here is the Rand Paul of retrievers folks! He's probably good at poker. Can drink beer (lol rad). 8/10 good dog https://t.co/pYAJkAe76p
2347	a	My oh my. This is a rare blond Canadian terrier on wheels. Only $8.98. Rather docile. 9/10 very rare https://t.co/yWBqbrzy8O
2348	a	Here is a Siberian heavily armored polar bear mix. Strong owner. 10/10 I would do unspeakable things to pet this dog https://t.co/rdivxLiqEt
2349	an	This is an odd dog. Hard on the outside but loving on the inside. Petting still fun. Doesn't play catch well. 2/10 https://t.co/v5A4vzSDdc
2350	a	This is a truly beautiful English Wilson Staff retriever. Has a nice phone. Privileged. 10/10 would trade lives with https://t.co/fvIbQfHjIe
2352	a	This is a purebred Piers Morgan. Loves to Netflix and chill. Always looks like he forgot to unplug the iron. 6/10 https://t.co/DWnyCjf2mx
2353	a	Here is a very happy pup. Big fan of well-maintained decks. Just look at that tongue. 9/10 would cuddle af https://t.co/y671yMhoiR
2354	a	This is a western brown Mitsubishi terrier. Upset about leaf. Actually 2 dogs here. 7/10 would walk the shit out of https://t.co/r7mOb2m0UI

```
In [19]: #Check the distribution of dog ratings
         twitter_arch.rating_numerator.value_counts()
```

```
Out[19]: 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
```

Quality Issue 8: There are errors in numerator column.

```
In [20]: twitter_arch.rating_denominator.value_counts()
```

```
Out[20]: 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
```

Quality Issue 9: There are errors in denominator column

```
In [21]: #check duplicated data
         twitter_arch.duplicated().sum()
```

```
Out[21]: 0
```

```
In [22]: #How many dogs with no stage records?
         len(twitter_arch[(twitter_arch.doggo=='None') & (twitter_arch.floofer=='None')
                      & (twitter_arch.pupper=='None') &(twitter_arch.puppo=='None')])
```

```
Out[22]: 1976
```

```
In [23]: #source is illegible. Check what information we can find there.
         twitter_arch.source.value_counts()
```

```
Out[23]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>     2221
         <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>                        91
         <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>                     33
         <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>    11
         Name: source, dtype: int64
```

Quality issue 10: source information needs to be cleaned up. It should show which application/device people used to access twitter

`tweets` dataset:

In [24]:
```
#Check columns,datatypes and null values
tweets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2330 entries, 0 to 2329
Data columns (total 32 columns):
created_at                       2330 non-null datetime64[ns, UTC]
id                               2330 non-null int64
id_str                           2330 non-null int64
full_text                        2330 non-null object
truncated                        2330 non-null bool
display_text_range               2330 non-null object
entities                         2330 non-null object
extended_entities                2058 non-null object
source                           2330 non-null object
in_reply_to_status_id            77 non-null float64
in_reply_to_status_id_str        77 non-null float64
in_reply_to_user_id              77 non-null float64
in_reply_to_user_id_str          77 non-null float64
in_reply_to_screen_name          77 non-null object
user                             2330 non-null object
geo                              0 non-null float64
coordinates                      0 non-null float64
place                            1 non-null object
contributors                     0 non-null float64
is_quote_status                  2330 non-null bool
retweet_count                    2330 non-null int64
favorite_count                   2330 non-null int64
favorited                        2330 non-null bool
retweeted                        2330 non-null bool
possibly_sensitive               2196 non-null float64
possibly_sensitive_appealable    2196 non-null float64
lang                             2330 non-null object
retweeted_status                 163 non-null object
quoted_status_id                 26 non-null float64
quoted_status_id_str             26 non-null float64
quoted_status_permalink          26 non-null object
quoted_status                    24 non-null object
dtypes: bool(4), datetime64[ns, UTC](1), float64(11), int64(4), object(12)
memory usage: 518.9+ KB
```

In [25]:
```
tweets.sample(3)
```

Out[25]:

	created_at	id	id_str	full_text	truncated	display_text_range	
1537	2016-01-16 15:40:14+00:00	688385280030670848	688385280030670848	This is Louis. He's takes top-notch selfies. 12/10 would snapchat with https://t.co/vz2DukO0th	False	[0, 94]	{'hashtags': [], 'symbols': [], 'user_mentions': [], 'urls': [] [{'id': 688385255020036098, 'id_str': '68838525502 'indices': [71, 94], 'm 'http://pbs.twimg.com/media/CY2ivgIWAAI 'media_ 'https://pbs.twimg.com/media/CY2ivgIWAAIPQzF. 'https://t.co/vz2DukO0th', 'display_url': 'pic.twitter.com/vz2 'expar 'https://twitter.com/dog_rates/status/688385280030670848/ 'type': 'photo', 'sizes': {'thumb': {'w': 150, 'h': 150, 'resize 'large': {'w': 882, 'h': 1024, 'resize': 'fit'}, 'medium': {'w' 1024, 'resize': 'fit'}, 'small': {'w': 586, 'h': 680, 'resiz
1406	2016-02-11 00:18:49+00:00	697575480820686848	697575480820686848	This is Ole. He's not sure how to gravity. 8/10 https://t.co/PsqqotpBBQ	False	[0, 71]	{'hashtags': [], 'symbols': [], 'user_mentions': [], 'urls': [] [{'id': 697575475464417280, 'id_str': '69757547546 'indices': [48, 71], 'm 'http://pbs.twimg.com/media/Ca5JMvMUsAA 'media_ 'https://pbs.twimg.com/media/Ca5JMvMUsAAGMlI. 'https://t.co/PsqqotpBBQ', 'dis 'pic.twitter.com/PsqqotpBBQ', 'expar 'https://twitter.com/dog_rates/status/697575480820686848/ 'type': 'photo', 'sizes': {'thumb': {'w': 150, 'h': 150, 'resize 'small': {'w': 510, 'h': 680, 'resize': 'fit'}, 'large': {'w': 768, 'resize': 'fit'}, 'medium': {'w': 768, 'h': 1024, 'resiz
2002	2015-12-02 01:39:53+00:00	671866342182637568	671866342182637568	Meet Dylan. He can use a fork but clearly can't put on a sweatshirt correctly. Looks like a disgruntled teen. 10/10 https://t.co/FWJQ1zQLiI	False	[0, 139]	{'hashtags': [], 'symbols': [], 'user_mentions': [], 'urls': [] [{'id': 671866334851014656, 'id_str': '671866334851 'indices': [116, 139], 'm 'http://pbs.twimg.com/media/CVLy3zFWoAA 'media_ 'https://pbs.twimg.com/media/CVLy3zFWoAA93qJ. 'https://t.co/FWJQ1zQLiI', 'dis 'pic.twitter.com/FWJQ1zQLiI', 'expar 'https://twitter.com/dog_rates/status/671866342182637568/ 'type': 'photo', 'sizes': {'small': {'w': 382, 'h': 680, 'res 'thumb': {'w': 150, 'h': 150, 'resize': 'crop'}, 'large': {'w' 1024, 'resize': 'fit'}, 'medium': {'w': 575, 'h': 1024, 'resiz

In [26]:
```
#Look at the statistics for quantitative variables
tweets.describe()
```

Out[26]:

	id	id_str	in_reply_to_status_id	in_reply_to_status_id_str	in_reply_to_user_id	in_reply_to_user_id_str	geo	coordinates	contributors	retweet_count
count	2.330000e+03	2.330000e+03	7.700000e+01	7.700000e+01	7.700000e+01	7.700000e+01	0.0	0.0	0.0	2330.000000
mean	7.419336e+17	7.419336e+17	7.440692e+17	7.440692e+17	2.040329e+16	2.040329e+16	NaN	NaN	NaN	2724.365236
std	6.823506e+16	6.823506e+16	7.524295e+16	7.524295e+16	1.260797e+17	1.260797e+17	NaN	NaN	NaN	4608.763163
min	6.660209e+17	6.660209e+17	6.658147e+17	6.658147e+17	1.185634e+07	1.185634e+07	NaN	NaN	NaN	1.000000
25%	6.782612e+17	6.782612e+17	6.757073e+17	6.757073e+17	3.589728e+08	3.589728e+08	NaN	NaN	NaN	549.500000
50%	7.183508e+17	7.183508e+17	7.032559e+17	7.032559e+17	4.196984e+09	4.196984e+09	NaN	NaN	NaN	1278.000000
75%	7.986712e+17	7.986712e+17	8.233264e+17	8.233264e+17	4.196984e+09	4.196984e+09	NaN	NaN	NaN	3168.250000
max	8.924206e+17	8.924206e+17	8.862664e+17	8.862664e+17	8.405479e+17	8.405479e+17	NaN	NaN	NaN	78432.000000

In [27]:
```
#Most of the tweets are in english so there is probably not many insights we can get from this column
tweets.lang.value_counts()
```

Out[27]:
```
en     2312
und       7
nl        3
in        3
ro        1
eu        1
es        1
tl        1
et        1
Name: lang, dtype: int64
```

Quality Issue 11: TwitterId should be string not integer

Tidiness issue 2: `favorite_count` and `retweet_count` should be part of the `Twitter_arch` table.

Summary:

Quality

`twitter_arch`

- Timestamp and Retweet timestamp should be datetime, not object. All ids should be string not float.
- Remove the retweets.
- Remove the columns irrelevant to the analysis.
- Correct the wrongly captured names and remove those that cannot be fixed.
- Correct the wrongly captured numerator and denominator ratings and remove those that cannot be fixed.
- Source information needs to be cleaned up. It should show which application/device people used to access twitter.
- Dog stages are wrongly captured.

`image_pred`

- Some breeds with the first letter capitalized, others not. Change all breed names to lowercase and remove the "-" dash in between.
- tweet_id should be string not integer.
- Column names 'p1','p2','p3','p1-dog','p2-dog',and 'p3-dog' are not informative. Rename them.

`tweets`

- Ids should be string not integer.

Tidiness

`twitter_arch`

- Dog stages should be one column with 5 possible outcomes (4 stages and None). Data type for the new column "stage" should be categorical.
- All three datasets should be merged into one on twitter_id

Clean Data

In [28]:
```
#make a copy for each dataset:
twitter_arch_clean=twitter_arch.copy()
image_pred_clean=image_pred.copy()
tweets_clean=tweets.copy()
```

`twitter_arch_clean`

Define:

Remove the retweets

Code:

In [29]:
```
twitter_arch_clean=twitter_arch_clean[twitter_arch_clean.in_reply_to_status_id.isnull()]
twitter_arch_clean=twitter_arch_clean[~twitter_arch_clean.text.str.contains('RT @')]
```

Test:

```
In [30]: twitter_arch_clean.in_reply_to_status_id.notnull().sum()
```

```
Out[30]: 0
```

```
In [31]: len(twitter_arch_clean[twitter_arch_clean.text.str.contains('RT @')])
```

```
Out[31]: 0
```

Define:

Drop the irrelevant columns

Code:

```
In [32]: #Drop the columns irrelevant to the analysis
         twitter_arch_clean.drop(['in_reply_to_status_id','in_reply_to_user_id','retweeted_status_id','retweeted_status_user_id','retweeted_status_timestamp','expanded_urls'],axis=1,inplace=True)
```

Test:

```
In [33]: #double check the columns
         list(twitter_arch_clean)
```

```
Out[33]: ['tweet_id',
          'timestamp',
          'source',
          'text',
          'rating_numerator',
          'rating_denominator',
          'name',
          'doggo',
          'floofer',
          'pupper',
          'puppo']
```

Define:

Change tweet_id datatype to string

Code:

```
In [34]: #Correct the datatypes
         twitter_arch_clean.tweet_id=twitter_arch_clean.tweet_id.astype(str)
```

Test:

```
In [35]: twitter_arch_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2355
Data columns (total 11 columns):
tweet_id              2097 non-null object
timestamp             2097 non-null object
source                2097 non-null object
text                  2097 non-null object
rating_numerator      2097 non-null int64
rating_denominator    2097 non-null int64
name                  2097 non-null object
doggo                 2097 non-null object
floofer               2097 non-null object
pupper                2097 non-null object
puppo                 2097 non-null object
dtypes: int64(2), object(9)
memory usage: 196.6+ KB
```

Define:

Remove the +0000 in timestamp column and change the datatype to datetime

Code:

```
In [36]: #Remove the +0000 in timestamp
         twitter_arch_clean.timestamp=twitter_arch_clean.timestamp.apply(lambda x: x[:-5])
```

```
In [37]: #Change the datatype to datetime
         twitter_arch_clean.timestamp=pd.to_datetime(twitter_arch_clean.timestamp,format='%Y/%m/%d %H:%M:%S')
```

Test:

In [38]:
```
twitter_arch_clean.timestamp.sample(5)
```

Out[38]:
```
1045   2016-06-17 00:05:25
995    2016-06-30 02:45:28
2096   2015-11-29 00:06:39
410    2017-01-23 00:13:17
1924   2015-12-08 03:57:26
Name: timestamp, dtype: datetime64[ns]
```

Define:

Correct the wrong names. Change those in the error list to 'None'

Code:

In [39]:
```
#Correct the wrongly catpured names: change those in the error list to None
twitter_arch_clean.name=twitter_arch_clean.name.apply(lambda x: 'None' if x in (name_error) else x)
```

Test:

In [40]:
```
#Double check whether erroneous names are changed to None

len(twitter_arch_clean[twitter_arch_clean.name.isin(name_error)])
```

Out[40]:
```
0
```

Define:

Correct the errors in numerators and denominators. Set all denominators to 10 and scale the numerators accordingly.Name the clean rating column as 'rating'.

Code:

In [41]:
```
#First look at the non-standardized denominators. i.e. those not equal to 10
twitter_arch_clean.query('rating_denominator!=10')[['text','rating_numerator',
                                                      'rating_denominator']]
```

Out[41]:

	text	rating_numerator	rating_denominator
433	The floofs have been released I repeat the floofs have been released. 84/70 https://t.co/NIYC820tmd	84	70
516	Meet Sam. She smiles 24/7 &amp; secretly aspires to be a reindeer. \nKeep Sam smiling by clicking and sharing this link:\nhttps://t.co/98tB8y7y7t https://t.co/LouL5vdvxx	24	7
902	Why does this never happen at my front door... 165/150 https://t.co/HmwrdfEfUE	165	150
1068	After so many requests, this is Bretagne. She was the last surviving 9/11 search dog, and our second ever 14/10. RIP https://t.co/XAVDNDaVgQ	9	11
1120	Say hello to this unbelievably well behaved squad of doggos. 204/170 would try to pet all at once https://t.co/yGQI3He3xv	204	170
1165	Happy 4/20 from the squad! 13/10 for all https://t.co/eV1diwds8a	4	20
1202	This is Bluebert. He just saw that both #FinalFur match ups are split 50/50. Amazed af. 11/10 https://t.co/Kky1DPG4iq	50	50
1228	Happy Saturday here's 9 puppers on a bench. 99/90 good work everybody https://t.co/mpvaVxKmc1	99	90
1254	Here's a brigade of puppers. All look very prepared for whatever happens next. 80/80 https://t.co/0eb7R1Om12	80	80
1274	From left to right:\nCletus, Jerome, Alejandro, Burp, &amp; Titson\nNone know where camera is. 45/50 would hug all at once https://t.co/sedre1ivTK	45	50
1351	Here is a whole flock of puppers. 60/50 I'll take the lot https://t.co/9dpcw6MdWa	60	50
1433	Happy Wednesday here's a bucket of pups. 44/40 would pet all at once https://t.co/HppvrYuamZ	44	40
1635	Someone help the girl is being mugged. Several are distracting her while two steal her shoes. Clever puppers 121/110 https://t.co/1zfnTJLt55	121	110
1662	This is Darrel. He just robbed a 7/11 and is in a high speed police chase. Was just spotted by the helicopter 10/10 https://t.co/7EsP8LmSp5	7	11
1779	IT'S PUPPERGEDDON. Total of 144/120 ...I think https://t.co/ZanVtAtvIq	144	120
1843	Here we have an entire platoon of puppers. Total score: 88/80 would pet all at once https://t.co/y93p6FLvVw	88	80
2335	This is an Albanian 3 1/2 legged Episcopalian. Loves well-polished hardwood flooring. Penis on the collar. 9/10 https://t.co/d9NcXFKwLv	1	2

In [42]:
```
#Correct the wrongly captured scores and
twitter_arch_clean.loc[1202,['rating_numerator','rating_denominator']]=(11,10)
twitter_arch_clean.loc[1662,['rating_numerator','rating_denominator']]=(10,10)
twitter_arch_clean.loc[2335,['rating_numerator','rating_denominator']]=(9,10)
twitter_arch_clean.loc[1068,['rating_numerator','rating_denominator']]=(14,10)
twitter_arch_clean.loc[1165,['rating_numerator','rating_denominator']]=(13,10)
```

In [43]:
```
#Remove the entry without rating
twitter_arch_clean=twitter_arch_clean[(twitter_arch_clean.rating_numerator!=24)]
```

In [44]:

```
#Double check before the standardization
twitter_arch_clean.query('rating_denominator!=10')[['text','rating_numerator',
                                                     'rating_denominator']]
```

Out[44]:

	text	rating_numerator	rating_denominator
433	The floofs have been released I repeat the floofs have been released. 84/70 https://t.co/NIYC820tmd	84	70
902	Why does this never happen at my front door... 165/150 https://t.co/HmwrdfEfUE	165	150
1120	Say hello to this unbelievably well behaved squad of doggos. 204/170 would try to pet all at once https://t.co/yGQI3He3xv	204	170
1228	Happy Saturday here's 9 puppers on a bench. 99/90 good work everybody https://t.co/mpvaVxKmc1	99	90
1254	Here's a brigade of puppers. All look very prepared for whatever happens next. 80/80 https://t.co/0eb7R1Om12	80	80
1274	From left to right:\nCletus, Jerome, Alejandro, Burp, &amp; Titson\nNone know where camera is. 45/50 would hug all at once https://t.co/sedre1ivTK	45	50
1351	Here is a whole flock of puppers. 60/50 I'll take the lot https://t.co/9dpcw6MdWa	60	50
1433	Happy Wednesday here's a bucket of pups. 44/40 would pet all at once https://t.co/HppvrYuamZ	44	40
1635	Someone help the girl is being mugged. Several are distracting her while two steal her shoes. Clever puppers 121/110 https://t.co/1zfnTJLt55	121	110
1779	IT'S PUPPERGEDDON. Total of 144/120 ...I think https://t.co/ZanVtAtvIq	144	120
1843	Here we have an entire platoon of puppers. Total score: 88/80 would pet all at once https://t.co/y93p6FLvVw	88	80

In [45]:

```
#Standardize the denominator by setting it to 10 and scaling the numerator
twitter_arch_clean['rating']=twitter_arch_clean.apply(lambda x: int(10*x.rating_numerator/x.rating_denominator) if (x.rating_denominator!=10) else
                                                             x.rating_numerator,axis=1)
```

In [46]:

```
#The standardization looks good
twitter_arch_clean.query('rating_denominator!=10')[['rating','rating_numerator','rating_denominator']]
```

Out[46]:

	rating	rating_numerator	rating_denominator
433	12	84	70
902	11	165	150
1120	12	204	170
1228	11	99	90
1254	10	80	80
1274	9	45	50
1351	12	60	50
1433	11	44	40
1635	11	121	110
1779	12	144	120
1843	11	88	80

In [47]:

```
#Second, look at rating outliers after standardization:
twitter_arch_clean.query('rating>15 |rating<6')[['text','rating']]
```

Out[47]:

	text	rating
45	This is Bella. She hopes her smile made you smile. If not, she is also offering you her favorite monkey. 13.5/10 https://t.co/qjrljjt948	5
315	When you're so blinded by your systematic plagiarism that you forget what day it is. 0/10 https://t.co/YbEJPkg4Ag	0
695	This is Logan, the Chow who lived. He solemnly swears he's up to lots of good. H*ckin magical af 9.75/10 https://t.co/yBO5wuqaPS	75
730	Who keeps sending in pictures without dogs in them? This needs to stop. 5/10 for the mediocre road https://t.co/ELqelxWMrC	5
763	This is Sophie. She's a Jubilant Bush Pupper. Super h*ckin rare. Appears at random just to smile at the locals. 11.27/10 would smile back https://t.co/QFaUiIHxHq	27
765	This is Wesley. He's clearly trespassing. Seems rather h*ckin violent too. Weaponized forehead. 3/10 wouldn't let in https://t.co/pL7wbMRW7M	3
883	This is Fido. He can tell the weather. Not good at fetch tho. Never comes when called. 4/10 would probably still pet https://t.co/4gOv2Q3iKP	4
912	Here's another picture without a dog in it. Idk why you guys keep sending these. 4/10 just because that's a neat rug https://t.co/mOmnL19Wsl	4
956	Please stop sending it pictures that don't even have a doggo or pupper in them. Churlish af. 5/10 neat couch tho https://t.co/u2c9c7qSg8	5
979	This is Atticus. He's quite simply America af. 1776/10 https://t.co/GRXwMxLBkh	1776
1004	Viewer discretion is advised. This is a terrible attack in progress. Not even in water (tragic af). 4/10 bad sherk https://t.co/L3U0j14N5R	4
1189	This is Alexanderson. He's got a weird ass birth mark. Dreadful at fetch. Won't eat kibble. 3/10 wtf @Target https://t.co/FmxOpf2Sgl	3
1219	This is Benedict. He's a feisty pup. Needs a brushing. Portable af. Looks very angry actually. 4/10 might not pet https://t.co/3oeFfHjv0Z	4
1249	What hooligan sent in pictures w/out a dog in them? Churlish af. 3/10 just bc that's a neat fluffy bean bag chair https://t.co/wcwoGOkZvz	3
1303	This is Keurig. He's a rare dog. Laughs like an idiot tho. Head is basically a weapon. Poorly maintained goatee 4/10 https://t.co/xOrUyj7K30	4
1314	This is Elliot. He's blocking the roadway. Downright rude as hell. Doesn't care that you're already late. 3/10 https://t.co/FMUxir5pYu	3
1399	This is Dave. He's a tropical pup. Short lil legs (dachshund mix?) Excels underwater, but refuses to eat kibble 5/10 https://t.co/ZJnCxlIf62	5
1406	This is Charl. He's a bully. Chucks that dumbbell around like its nothing. Sharp neck. Exceptionally unfluffy. 3/10 https://t.co/VfLoDZecJ7	3
1459	This may be the greatest video I've ever been sent. 4/10 for Charles the puppy, 13/10 overall. (Vid by @stevenxx_) https://t.co/uaJmNgXR2P	4
1461	Please only send in dogs. This t-rex is very scary. 5/10 ...might still pet (vid by @helizabethmicha) https://t.co/Vn6w5w8TO2	5
1478	Meet Phil. He's big af. Currently destroying this nice family home. Completely uncalled for. 3/10 not a good pupper https://t.co/fShNNhBWYx	3
1508	When bae says they can't go out but you see them with someone else that same night. 5/10 &amp; 10/10 for heartbroken pup https://t.co/aenk0KpoWM	5
1583	Army of water dogs here. None of them know where they're going. Have no real purpose. Aggressive barks. 5/10 for all https://t.co/A88x73TwMN	5
1601	This is Hammond. He's a peculiar pup. Loves long walks. Bark barely audible. Too many legs. 3/10 must be rare https://t.co/NOIiRWr5Jf	3
1619	This is Jerry. He's a neat dog. No legs (tragic). Has more horns than a dog usually does. Bark is unique af. 5/10 https://t.co/85q7xlplsJ	5
1624	Here we have a basking dino pupper. Looks powerful. Occasionally shits eggs. Doesn't want the holidays to end. 5/10 https://t.co/DnNweb5eTO	5
1629	This is Bobby. He doesn't give a damn about personal space. Convinced he called shotgun first. 4/10 not the best dog https://t.co/b8XW69gSaU	4
1645	This is Jiminy. He's not the brightest dog. Needs to lay off the kibble. 5/10 still petable https://t.co/omln4LOy1x	5
1680	Unique dog here. Wrinkly as hell. Weird segmented neck. Finger on fire. Doesn't seem to notice. 5/10 might still pet https://t.co/Hy9La4xNX3	5
1692	This is Chuck. He's a neat dog. Very flexible. Trapped in a glass case of emotion. Devastatingly unfluffy 3/10 https://t.co/YqbU9xHV3p	3
1701	This is Alice. She's an idiot. 4/10 https://t.co/VQXdwJfkyS	4
1712	Here we have uncovered an entire battalion of holiday puppers. Average of 11.26/10 https://t.co/eNm2S6p9BD	26
1727	Meet Penelope. She's a bacon frise. Total babe (lol get it like the movie). Doesn't bark tho. 5/10 very average dog https://t.co/SDcQYg0HSZ	5
1761	Exotic pup here. Tail long af. Throat looks swollen. Might breathe fire. Exceptionally unfluffy 2/10 would still pet https://t.co/a8SqCaSo2r	2
1764	This is Crystal. She's a shitty fireman. No sense of urgency. People could be dying Crystal. 2/10 just irresponsible https://t.co/rtMtjSl9pz	2
1796	This is Juckson. He's totally on his way to a nascar race. 5/10 for Juckson https://t.co/IoLRvF0Kak	5
1808	Exotic handheld dog here. Appears unathletic. Feet look deadly. Can be thrown a great distance. 5/10 might pet idk https://t.co/Avq4awulqk	5
1820	This is Bubbles. He kinda resembles a fish. Always makes eye contact with u no matter what. Sneaky tongue slip. 5/10 https://t.co/Nrhvc5tLFT	5
1836	Extremely rare pup here. Very religious. Always praying. Too many legs. Not overwhelmingly fluffy. Won't bark. 3/10 https://t.co/REyE5YKVBb	3
1861	Rare shielded battle dog here. Very happy about abundance of lettuce. Painfully slow fetcher. Still petable. 5/10 https://t.co/C3tlKVq7eO	5
1869	What kind of person sends in a picture without a dog in it? 1/10 just because that's a nice table https://t.co/RDXCfk8hK0	1
1874	This is Steven. He got locked outside. Damn it Steven. 5/10 nice grill tho https://t.co/zf7Sxxjfp3	5
1898	Meet Patrick. He's an exotic pup. Jumps great distances for a dog. Always gets injured when I toss him a ball. 3/10 https://t.co/Unz1uNrOzo	3
1901	Two gorgeous dogs here. Little waddling dog is a rebel. Refuses to look at camera. Must be a preteen. 5/10 &amp; 8/10 https://t.co/YPfw7oahbD	5
1904	Rare submerged pup here. Holds breath for a long time. Frowning because that spoon ignores him. 5/10 would still pet https://t.co/EJzzNHE8bE	5
1920	This is Henry. He's a shit dog. Short pointy ears. Leaves trail of pee. Not fluffy. Doesn't come when called. 2/10 https://t.co/Pu9RhfHDEQ	2
1925	This is Earl. Earl is lost. Someone help Earl. He has no tags. Just trying to get home. 5/10 hang in there Earl https://t.co/1ZbfqAVDg6	5
1928	Herd of wild dogs here. Not sure what they're trying to do. No real goals in life. 3/10 find your purpose puppers https://t.co/t5ih0VrK02	3
1938	Guys I'm getting real tired of this. We only rate dogs. Please don't send in other things like this Bulbasaur. 3/10 https://t.co/t5rQHl6W8M	3
1941	This is a heavily opinionated dog. Loves walls. Nobody knows how the hair works. Always ready for a kiss. 4/10 https://t.co/dFiaKZ9cDl	4
1947	Large blue dog here. Cool shades. Flipping us off w both hands. Obviously a preteen. 3/10 for rude blue preteen pup https://t.co/mcPd5AFfhA	3
1979	Extraordinary dog here. Looks large. Just a head. No body. Rather intrusive. 5/10 would still pet https://t.co/ufHWUFA9Pu	5
2013	Exotic underwater dog here. Very shy. Wont return tennis balls I toss him. Never been petted. 5/10 I bet he's soft https://t.co/WH7Nzc5IBA	5
2026	This is Brad. He's a chubby lil pup. Doesn't really need the food he's trying to reach. 5/10 you've had enough Brad https://t.co/vPXKSaNsbE	5
2063	This is Anthony. He just finished up his masters at Harvard. Unprofessional tattoos. Always looks perturbed. 5/10 https://t.co/iHLo9rGay1	5
2070	Two miniature golden retrievers here. Webbed paws. Don't walk very efficiently. Can't catch a tennis ball. 4/10s https://t.co/WzVLdSHJU7	4
2074	After so many requests... here you go.\n\nGood dogg. 420/10 https://t.co/yfAAo1gdeY	420
2076	Pink dogs here. Unreasonably long necks. Left guy has only 1 leg. Quite nimble. Don't bark tho 4/10s would still pet https://t.co/QY5uvMmmQk	4
2079	Scary dog here. Too many legs. Extra tail. Not soft, let alone fluffy. Won't bark. Moves sideways. Has weapon. 2/10 https://t.co/XOPXCSXiUT	2
2091	Flamboyant pup here. Probably poisonous. Won't eat kibble. Doesn't bark. Slow af. Petting doesn't look fun. 1/10 https://t.co/jxukeh2BeO	1
2092	This dude slaps your girl's ass what do you do?\n5/10 https://t.co/6dioUL6gcP	5
2109	Vibrant dog here. Fabulous tail. Only 2 legs tho. Has wings but can barely fly (lame). Rather elusive. 5/10 okay pup https://t.co/cixC0M3P1e	5
2134	This is Randall. He's from Chernobyl. Built playground himself. Has been stuck up there quite a while. 5/10 good dog https://t.co/pzrvc7wKGd	5
2136	This is Tommy. He's a cool dog. Hard not to step on. Won't let go of seashell. Not fast by any means. 3/10 https://t.co/0gY6XTOpn3	3
2139	Awesome dog here. Not sure where it is tho. Spectacular camouflage. Enjoys leaves. Not very soft. 5/10 still petable https://t.co/rOTOteKx4q	5

	text	rating
2153	This is a brave dog. Excellent free climber. Trying to get closer to God. Not very loyal though. Doesn't bark. 5/10 https://t.co/ODnILTr4QM	5
2181	Two gorgeous pups here. Both have cute fake horns(adorable). Barn in the back looks on fire. 5/10 would pet rly well https://t.co/w5oYFXi0uh	5
2183	This is Bernie. He's taking his Halloween costume very seriously. Wants to be baked. 3/10 not a good idea Bernie smh https://t.co/1zBp1moFlX	3
2186	Unique dog here. Oddly shaped tail. Long pink front legs. I don't think dogs breath underwater sos. 4/10 bad owner https://t.co/0EJXxE9UxW	4
2202	Fascinating dog here. Loves beach. Oddly long nose for dog. Massive ass paws. Hard to cuddle w. 3/10 would still pet https://t.co/IiSdmhkC5N	3
2206	Meet Zeek. He is a grey Cumulonimbus. Zeek is hungry. Someone should feed Zeek asap. 5/10 absolutely terrifying https://t.co/fvVNScw8VH	5
2222	Here is a mother dog caring for her pups. Snazzy red mohawk. Doesn't wag tail. Pups look confused. Overall 4/10 https://t.co/YOHe6lf09m	4
2237	This lil pup is Oliver. Hops around. Has wings but doesn't fly (lame). Annoying chirp. Won't catch tennis balls 2/10 https://t.co/DnhUw0aBM2	2
2239	This dog resembles a baked potato. Bed looks uncomfortable. No tail. Comes with butter tho. 3/10 petting still fun https://t.co/x89NSCEZCq	3
2242	Wow. Armored dog here. Ready for battle. Face looks dangerous. Not very loyal. Lil dog on back havin a blast. 5/10 https://t.co/SyMoWrp368	5
2246	This is Tedrick. He lives on the edge. Needs someone to hit the gas tho. Other than that he's a baller. 10&amp;2/10 https://t.co/LvP1TTYSCN	2
2261	Never seen dog like this. Breathes heavy. Tilts head in a pattern. No bark. Shitty at fetch. Not even cordless. 1/10 https://t.co/i9iSGNn3fx	1
2288	These are strange dogs. All have toupees. Long neck for dogs. In a shed of sorts? Work in groups? 4/10 still petable https://t.co/PZxSarAfSN	4
2305	My goodness. Very rare dog here. Large. Tail dangerous. Kinda fat. Only eats leaves. Doesn't come when called 3/10 https://t.co/xYGdBrMS9h	3
2310	Unfamiliar with this breed. Ears pointy af. Won't let go of seashell. Won't eat kibble. Not very fast. Bad dog 2/10 https://t.co/EIn5kElY1S	2
2312	This is Josep. He is a Rye Manganese mix. Can drive w eyes closed. Very irresponsible. Menace on the roadways. 5/10 https://t.co/XNGeDwrtYH	5
2316	Cool dog. Enjoys couch. Low monotone bark. Very nice kicks. Pisses milk (must be rare). Can't go down stairs. 4/10 https://t.co/vXMKrJC81s	4
2326	This is quite the dog. Gets really excited when not in water. Not very soft tho. Bad at fetch. Can't do tricks. 2/10 https://t.co/aMCTNWO94t	2
2334	This is a funny dog. Weird toes. Won't come down. Loves branch. Refuses to eat his food. Hard to cuddle with. 3/10 https://t.co/IIXis0zta0	3
2338	Not familiar with this breed. No tail (weird). Only 2 legs. Doesn't bark. Surprisingly quick. Shits eggs. 1/10 https://t.co/Asgdc6kuLX	1
2349	This is an odd dog. Hard on the outside but loving on the inside. Petting still fun. Doesn't play catch well. 2/10 https://t.co/v5A4vzSDdc	2
2351	Here we have a 1949 1st generation vulpix. Enjoys sweat tea and Fox News. Cannot be phased. 5/10 https://t.co/4B7cOc1EDq	5

In [48]:

```
#Correct the wrongly captured rating and round them to the nearest integer:
twitter_arch_clean.loc[45,'rating']=14
twitter_arch_clean.loc[763,'rating']=11
twitter_arch_clean.loc[1712,'rating']=11
twitter_arch_clean.loc[695,'rating']=10
```

In [49]:

```
#Remove the symbolic 1776 and the entry without rating.
twitter_arch_clean.drop(index=979,inplace=True)
```

Test:

In [50]:

```
#Check the rating distribution again
twitter_arch_clean.rating.value_counts()
```

Out[50]:

```
12     490
10     439
11     421
13     288
9      154
8       98
7       51
14      40
5       33
6       32
3       19
4       15
2        9
1        4
420      1
0        1
Name: rating, dtype: int64
```

Define:

Drop the redundant ratingcolumns and just keep the clean 'rating' column

Code:

In [51]:

```
#Drop the original numerator and denominator columns as rating column alone suffices now:
twitter_arch_clean.drop(['rating_numerator','rating_denominator'],axis=1,inplace=True)
```

Test:

In [52]:
```
#Double check
twitter_arch_clean.sample(5)
```

Out[52]:

	tweet_id	timestamp	source	text	name	doggo	floofer	pupper	puppo	rating
168	859607811541651456	2017-05-03 03:17:27	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	Sorry for the lack of posts today. I came home from school and had to spend quality time with my puppo. Her name is Zoey and she's 13/10 https://t.co/BArWupFAn0	None	None	None	None	puppo	13
459	817827839487737858	2017-01-07 20:18:46	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Buddy. He ran into a glass door once. Now he's h*ckin skeptical. 13/10 empowering af (vid by Brittany Gaunt) https://t.co/q2BgNIi3OA	Buddy	None	None	None	None	13
496	813157409116065792	2016-12-25 23:00:08	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Layla. It is her first Christmas. She got to be one of the presents. 12/10 I wish my presents would bark https://t.co/hwhCbhCjnV	Layla	None	None	None	None	12
2069	671134062904504320	2015-11-30 01:10:04	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	Say hello to Clarence. He's a western Alkaline Pita. Very proud of himself for dismembering his stuffed dog pal 8/10 https://t.co/BHxr9O7wJY	Clarence	None	None	None	None	8
1878	675047298674663426	2015-12-10 20:19:52	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is a fluffy albino Bacardi Columbia mix. Excellent at the tweets. 11/10 would hug gently https://t.co/diboDRUuEl	None	None	None	None	None	11

Define:

Clean up the source column by extracting the relevant text.

Code:

In [53]:
```
#Extract relevant information from the source column and assign it to the source column again
twitter_arch_clean.source=twitter_arch_clean.source.str.extract(r'\>(.*?)\<')
```

Test:

In [54]:
```
#Double check the extraction
twitter_arch_clean.source.value_counts()
```

Out[54]:
```
Twitter for iPhone      1963
Vine - Make a Scene       91
Twitter Web Client        31
TweetDeck                 10
Name: source, dtype: int64
```

Define:

Create a single column for dog stages

Code:

In [55]:
```
#Create a single column for dog stage
doggo=twitter_arch_clean.doggo.replace('None','')
floofer=twitter_arch_clean.floofer.replace('None','')
pupper=twitter_arch_clean.pupper.replace('None','')
puppo=twitter_arch_clean.puppo.replace('None','')
twitter_arch_clean['stage']=doggo+floofer+pupper+puppo
```

In [56]:
```
#Check the newly created column 'stage'
twitter_arch_clean['stage'].value_counts()
```

Out[56]:
```
                1759
pupper           221
doggo             72
puppo             23
floofer            9
doggopupper        9
doggopuppo         1
doggofloofer       1
Name: stage, dtype: int64
```

In [57]:

```
#Double check with two stages. They probably have stages wrongly captured
twitter_arch_clean.query('stage ==["doggopupper","doggofloofer","doggopuppo"]')
```

Out[57]:

	tweet_id	timestamp	source	text	name	doggo	floofer	pupper	puppo	rating	stage
191	855851453814013952	2017-04-22 18:31:02	Twitter for iPhone	Here's a puppo participating in the #ScienceMarch. Cleverly disguising her own doggo agenda. 13/10 would keep the planet habitable for https://t.co/cMhq16isel	None	doggo	None	None	puppo	13	doggopuppo
200	854010172552949760	2017-04-17 16:34:26	Twitter for iPhone	At first I thought this was a shy doggo, but it's actually a Rare Canadian Floofer Owl. Amateurs would confuse the two. 11/10 only send dogs https://t.co/TXdT3tmuYk	None	doggo	floofer	None	None	11	doggofloofer
460	817777686764523521	2017-01-07 16:59:28	Twitter for iPhone	This is Dido. She's playing the lead role in "Pupper Stops to Catch Snow Before Resuming Shadow Box with Dried Apple." 13/10 (IG: didodoggo) https://t.co/m7isZrOBX7	Dido	doggo	None	pupper	None	13	doggopupper
531	808106460588765185	2016-12-12 00:29:28	Twitter for iPhone	Here we have Burke (pupper) and Dexter (doggo). Pupper wants to be exactly like doggo. Both 12/10 would pet at same time https://t.co/ANBpEYHaho	None	doggo	None	pupper	None	12	doggopupper
575	801115127852503040	2016-11-22 17:28:25	Twitter for iPhone	This is Bones. He's being haunted by another doggo of roughly the same size. 12/10 deep breaths pupper everything's fine https://t.co/55Dqe0SJNj	Bones	doggo	None	pupper	None	12	doggopupper
705	785639753186217984	2016-10-11 00:34:48	Twitter for iPhone	This is Pinot. He's a sophisticated doggo. You can tell by the hat. Also pointier than your average pupper. Still 10/10 would pet cautiously https://t.co/f2wmLZTPHd	Pinot	doggo	None	pupper	None	10	doggopupper
733	781308096455073793	2016-09-29 01:42:20	Vine - Make a Scene	Pupper butt 1, Doggo 0. Both 12/10 https://t.co/WQvcPEpH2u	None	doggo	None	pupper	None	12	doggopupper
889	759793422261743616	2016-07-31 16:50:42	Twitter for iPhone	Meet Maggie &amp; Lila. Maggie is the doggo, Lila is the pupper. They are sisters. Both 12/10 would pet at the same time https://t.co/MYwR4DQKll	Maggie	doggo	None	pupper	None	12	doggopupper
956	751583847268179968	2016-07-09 01:08:47	Twitter for iPhone	Please stop sending it pictures that don't even have a doggo or pupper in them. Churlish af. 5/10 neat couch tho https://t.co/u2c9c7qSg8	None	doggo	None	pupper	None	5	doggopupper
1063	741067306818797568	2016-06-10 00:39:48	Twitter for iPhone	This is just downright precious af. 12/10 for both pupper and doggo https://t.co/o5J479bZUC	None	doggo	None	pupper	None	12	doggopupper
1113	733109485275860992	2016-05-19 01:38:16	Twitter for iPhone	Like father (doggo), like son (pupper). Both 12/10 https://t.co/pG2inLaOda	None	doggo	None	pupper	None	12	doggopupper

In [58]:

```
#Fix the wrong stages
twitter_arch_clean.loc[191,'stage']='puppo'
twitter_arch_clean.loc[200,'stage']='doggo'
twitter_arch_clean.loc[460,'stage']='pupper'
twitter_arch_clean.loc[575,'stage']='pupper'
twitter_arch_clean.loc[705,'stage']='doggo'
```

In [59]:

```
#Drop the rest which contain more than one dog or don't provide any info
twitter_arch_clean.drop(twitter_arch_clean.query('stage ==["doggopupper","doggofloofer","doggopuppo"]').index,inplace=True)
```

In [60]:

```
#change the whitespace back to "none"
twitter_arch_clean.stage.replace('','none',inplace=True);
```

Test:

In [61]:

```
#double check the stage column
twitter_arch_clean.stage.value_counts()
```

Out[61]:

```
none       1759
pupper      223
doggo        74
puppo        24
floofer       9
Name: stage, dtype: int64
```

Define:

Change the stage column to categorical and drop the redundant columns

Code:

In [62]:

```
#Change the stage column datatype to categorical
twitter_arch_clean.stage=twitter_arch_clean.stage.astype('category');
```

In [63]:

```
#Drop the old dog stage columns
twitter_arch_clean.drop(['doggo','floofer','pupper','puppo'],axis=1,inplace=True)
```

Test:

In [64]:

```
#Check the result
twitter_arch_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2089 entries, 0 to 2355
Data columns (total 7 columns):
tweet_id     2089 non-null object
timestamp    2089 non-null datetime64[ns]
source       2089 non-null object
text         2089 non-null object
name         2089 non-null object
rating       2089 non-null int64
stage        2089 non-null category
dtypes: category(1), datetime64[ns](1), int64(1), object(4)
memory usage: 116.5+ KB
```

In [65]:

```
twitter_arch_clean.sample(5)
```

Out[65]:

	tweet_id	timestamp	source	text	name	rating	stage
1744	679158373988876288	2015-12-22 04:35:49	Twitter for iPhone	This is Rubio. He has too much skin. 11/10 https://t.co/NLOHmlENag	Rubio	11	none
871	761599872357261312	2016-08-05 16:28:54	Twitter for iPhone	This is Sephie. According to this picture, she can read. Fantastic at following directions. 11/10 such a good girl https://t.co/7HY9RvCudo	Sephie	11	none
2202	668643542311546881	2015-11-23 04:13:37	Twitter for iPhone	Fascinating dog here. Loves beach. Oddly long nose for dog. Massive ass paws. Hard to cuddle w. 3/10 would still pet https://t.co/IiSdmhkC5N	None	3	none
2301	667044094246576128	2015-11-18 18:17:59	Twitter for iPhone	12/10 gimme now https://t.co/QZAnwgnOMB	None	12	none
1022	746542875601690625	2016-06-25 03:17:46	Vine - Make a Scene	Here's a golden floofer helping with the groceries. Bed got in way. Still 11/10 helpful af (vid by @categoen) https://t.co/6ZRoZUWFmd	None	11	floofer

image_pred_clean

In [66]:

```
image_pred_clean=image_pred.copy()
```

Define: Standardize the dog breed names by removing the dash and changing all to lowercase

Code:

In [67]:

```
#Standardize the dog breed names by removing the dash and changing all to lowercase
image_pred_clean[['p1','p2','p3']]=image_pred_clean[['p1','p2',
                                    'p3']].apply(lambda x:x.str.lower().str.replace("_"," "),axis=1)
```

Test:

In [68]:

```
#double check
image_pred_clean[['p1','p2','p3']].sample(10)
```

Out[68]:

	p1	p2	p3
317	ice bear	ram	arctic fox
1331	bluetick	beagle	walker hound
937	pembroke	basenji	cardigan
856	weasel	toy poodle	scottish deerhound
1622	bookcase	entertainment center	file
1376	miniature poodle	toy poodle	teddy
424	ostrich	bearskin	swab
800	lakeland terrier	irish terrier	airedale
1837	american staffordshire terrier	staffordshire bullterrier	dalmatian
278	redbone	beagle	rhodesian ridgeback

Define:

Change the tweet_id datatype to str

Code:

In [69]:

```
#change the tweet_id datatype to str
image_pred_clean.tweet_id=image_pred_clean.tweet_id.astype(str)
```

Test:

In [70]:

```
image_pred_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null object
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(1), object(5)
memory usage: 152.1+ KB
```

Define:

Create a new column 'dog_or_not' which synthesizes the information from 3 predictions

> If 3 out of 3 predictions point to dog- Yes, it's a dog
>
> If The one or two out of 3 predictions point(s) to dog- Maybe, it's a dog
>
> If None of the predictions points to dog- No, it's not a dog

Drop the irrelevant columns and rename the newly created one

Code:

In [71]:

```
predict={0:'No',1:'Maybe',2:'Maybe',3:'Yes'}
image_pred_clean['predictions']=image_pred_clean.p1_dog*1 + image_pred_clean.p2_dog*1 + image_pred_clean.p3_dog*1
```

In [72]:

```
#Mapping from bools to answers
image_pred_clean['predictions']=image_pred_clean['predictions'].apply(lambda x: predict[x])
```

In [73]:

```
#drop columns about the 2nd and the 3rd predictions and also p1_dog
image_pred_clean.drop(['p1_dog','p2','p2_conf','p2_dog','p3','p3_conf','p3_dog'],axis=1,inplace=True)
```

In [74]:

```
#Rename columns:
image_pred_clean.rename(columns={'p1':'pred_breed','p1_conf':'pred_confidence','predictions':'dog_or_not'},inplace=True);
```

Test:

In [75]:

```
image_pred_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 6 columns):
tweet_id           2075 non-null object
jpg_url            2075 non-null object
img_num            2075 non-null int64
pred_breed         2075 non-null object
pred_confidence    2075 non-null float64
dog_or_not         2075 non-null object
dtypes: float64(1), int64(1), object(4)
memory usage: 97.4+ KB
```

`tweets_clean`

Define:

Create a new dataframe with just twitter id, retweet_count and favorite_count. Rename the id column and change the datatype to string.

Code:

In [80]:

```
#Create a new dataframe with just twitter id, retweet_count and favorite_count
df=tweets_clean[['id','retweet_count','favorite_count']]
```

In [81]:

```
#Change datatype and column name for twitter_id
df.id=df.id.astype(str)
df.rename(columns={'id':'tweet_id'},inplace=True)
```

Text:

In [82]:
```
#double check
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2330 entries, 0 to 2329
Data columns (total 3 columns):
tweet_id          2330 non-null object
retweet_count     2330 non-null int64
favorite_count    2330 non-null int64
dtypes: int64(2), object(1)
memory usage: 54.7+ KB
```

Merge the three dataframes

Define:

Join the three tables on tweet_id

Code:

In [82]:
```
# Merge the three dataframes
dog_rating=pd.merge((pd.merge(twitter_arch_clean,image_pred_clean,on='tweet_id',how='left')),df,on='tweet_id',how='left')
```

Test:

In [83]:
```
#Check nulls and datatypes
dog_rating.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2089 entries, 0 to 2088
Data columns (total 14 columns):
tweet_id           2089 non-null object
timestamp          2089 non-null datetime64[ns]
source             2089 non-null object
text               2089 non-null object
name               2089 non-null object
rating             2089 non-null int64
stage              2089 non-null category
jpg_url            1964 non-null object
img_num            1964 non-null float64
pred_breed         1964 non-null object
pred_confidence    1964 non-null float64
dog_or_not         1964 non-null object
retweet_count      2081 non-null float64
favorite_count     2081 non-null float64
dtypes: category(1), datetime64[ns](1), float64(4), int64(1), object(7)
memory usage: 230.7+ KB
```

Define:

Drop the missing values and reset the index

Code:

In [84]:
```
#Drop nulls
dog_rating.dropna(inplace=True)
```

In [85]:
```
# Reset the index
dog_rating.reset_index();
```

Test:

In [86]:
```
#A final check!
dog_rating.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1956 entries, 0 to 2088
Data columns (total 14 columns):
tweet_id           1956 non-null object
timestamp          1956 non-null datetime64[ns]
source             1956 non-null object
text               1956 non-null object
name               1956 non-null object
rating             1956 non-null int64
stage              1956 non-null category
jpg_url            1956 non-null object
img_num            1956 non-null float64
pred_breed         1956 non-null object
pred_confidence    1956 non-null float64
dog_or_not         1956 non-null object
retweet_count      1956 non-null float64
favorite_count     1956 non-null float64
dtypes: category(1), datetime64[ns](1), float64(4), int64(1), object(7)
memory usage: 216.0+ KB
```

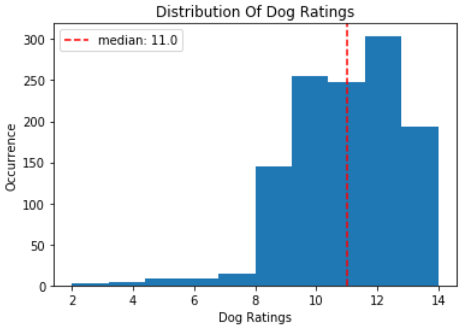
In [87]:
```
#Save the clean data to csv
dog_rating.to_csv('twitter_archive_master.csv',index=False)
```

Analyze and Visualize Data

1. What is the distribution of the rankings for dogs recognized by the image predictor?

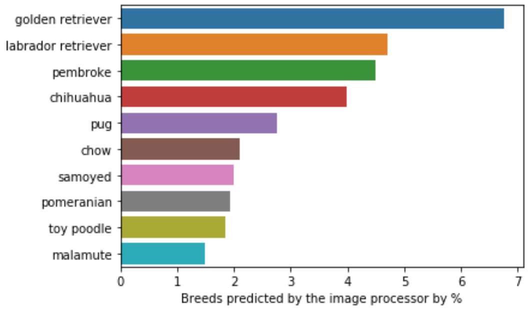
```
In [89]: #Select photos which pass the prediction as dogs
         dog_r=dog_rating.query('dog_or_not=="Yes"')
```

```
In [90]: plt.hist(dog_r['rating'])
         plt.axvline(dog_r['rating'].median(),color='r',linestyle='--',label='median: '+str(dog_r['rating'].median()))
         plt.xlabel('Dog Ratings')
         plt.ylabel('Occurrence')
         plt.title('Distribution Of Dog Ratings')
         plt.legend()
         plt.show();
```



2. What are the top 10 breeds based on the image predictor?

```
In [91]: top5_breed=dog_rating.pred_breed.value_counts()[:10]
         top5_breed=top5_breed/len(dog_rating)*100
         sns.barplot(x=top5_breed,y=top5_breed.index)
         plt.xlabel('Breeds predicted by the image processor by %');
```



98%+ of users accessed via Twitter for Iphone app, only less than 2% used other apps

3. What are the stages of these dogs?

```
In [92]: df1=dog_rating.query('stage==["puppo","doggo","pupper","floofer"]')
```

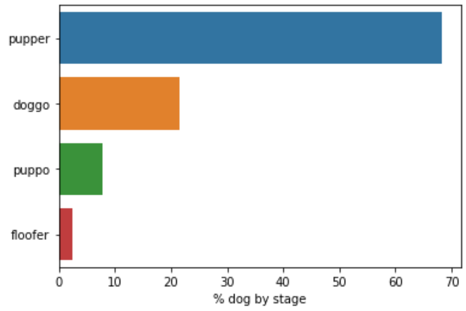
```
In [93]: stage=(df1.stage.value_counts()[:4])/len(df1)*100
         stage.sort_values(ascending=False,inplace=True)
```

```
In [94]: stage
```

```
Out[94]: pupper     68.350168
         doggo      21.548822
         puppo       7.744108
         floofer     2.356902
         Name: stage, dtype: float64
```

In [95]:

```
sns.barplot(x=stage,y=list(stage.index))
plt.xlabel('% dog by stage')
plt.show();
```

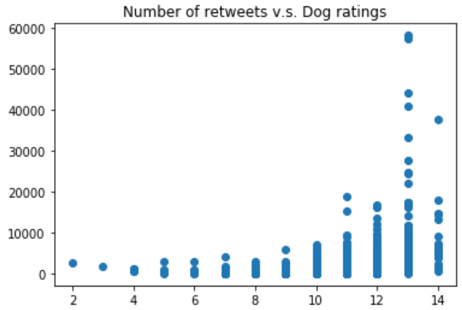


Among tweets which contain stage information, the majority (68%) concern puppers.

4. What is the relationship between rating and the number of retweets or rating and the number of favorites,with only the dog photos counted?

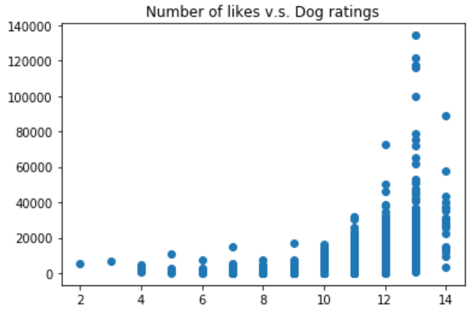
In [96]:

```
plt.scatter('rating','retweet_count',data=dog_rating.query('dog_or_not=="Yes"'))
plt.title('Number of retweets v.s. Dog ratings')
plt.show();
```



In [97]:

```
plt.scatter('rating','favorite_count',data=dog_rating.query('dog_or_not=="Yes"'))
plt.title('Number of likes v.s. Dog ratings')
plt.show();
```



5. How do users access their twitter accounts?

In [98]:

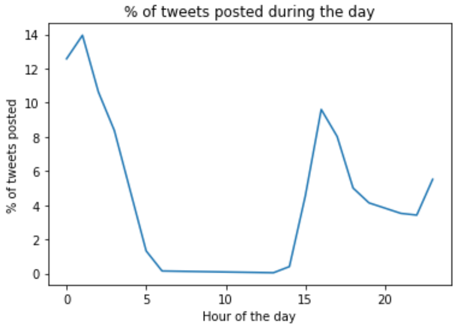
```
source=dog_rating.groupby(['source']).count()['tweet_id']/len(dog_rating)*100
source
```

Out[98]:

```
source
TweetDeck              0.460123
Twitter Web Client     1.431493
Twitter for iPhone    98.108384
Name: tweet_id, dtype: float64
```

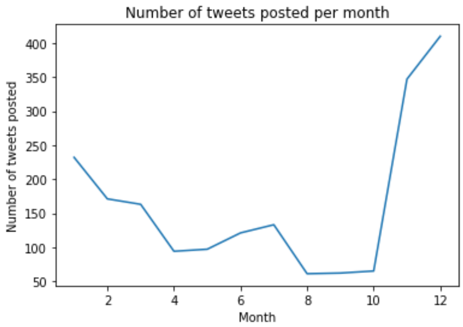
6. When were these tweets posted during the day?

```
In [101]: t=(pd.DatetimeIndex(dog_rating.timestamp).hour.value_counts().sort_index())/len(dog_rating)*100
          plt.plot(t.index,t)
          plt.xlabel('Hour of the day')
          plt.ylabel('% of tweets posted')
          plt.title('% of tweets posted during the day');
```



7. What are the number of tweets per month?

```
In [102]: m=pd.DatetimeIndex(dog_rating.timestamp).month.value_counts().sort_index()
          plt.plot(m.index,m)
          plt.xlabel('Month')
          plt.ylabel('Number of tweets posted')
          plt.title('Number of tweets posted per month');
```



8. What are the most popular dog names?

```
In [105]: dog_rating.query('name!="None"').name.value_counts()[:10]
```

```
Out[105]: Oliver      10
          Cooper      10
          Charlie     10
          Penny        9
          Lucy         9
          Tucker       9
          Winston      8
          Sadie        8
          Lola         7
          Daisy        7
          Name: name, dtype: int64
```

Create a 300-600 word written report called wrangle_report.pdf or wrangle_report.html that briefly describes your wrangling efforts. This is to be framed as an internal document.

Create a 250-word-minimum written report called act_report.pdf or act_report.html that communicates the insights and displays the visualization(s) produced from your wrangled data. This is to be framed as an external document, like a blog post or magazine article, for example.

Summary

I have performed data wrangling on three datesets which contain messy and untidy WeRateDogs data in this project. The clean data is saved as `twitter_archive_master.csv`. Please refer to 'act_report 'for analysis and insights into final data and wrangle_report for data wrangling steps.

```
In [ ]:
```