Introduction

WeRateDogs: hilarious Twitter feed where dogs are numerically objectified

The dataset I will be wrangling, analyzing, and visualizing is the tweet archive of WeRateDogs (Twitter user @dog_rate). WeRateDogs is a Twitter account that rates people's dogs along with a humorous comment. These ratings almost always have a denominator of 10, and a numerator that is usually greater than 10. The higher the numerator rating, the 'better' the dog. WeRateDogs has over 4 million followers and has received international media coverage.

The Data Wrangling Process: Gather, Assess, Clean

I will be gathering data from three sources:

- The 'enhanced' Twitter archive WeRateDogs, a csv file provided by Udacity. This archive contains very basic tweet data for all 5000+ of their tweets, but not everything. The archive contains each tweet's text, which Udacity used to enhance by extracting rating, dog name, and dog 'stage' (doggo, floofer, pupper, and puppo). Of the 5000+ tweets, this archive is filtered for tweets with ratings only (there are 2356).
- An 'image prediction' file, or what breed of dog is in each tweet, according to a neural network. This shows the top three breed predictions alongside each tweet ID, the image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images). I will download the image predictions file programmatically from Udcity's servers using the Requests library.
- I will query Twitter's API to gather additional data, retweet count and favorite count, two of the notable column omissions in the Twitter archive. Using the tweet IDs in the WeRateDogs archive, I will query the API for each tweet's JSON data using Python's Tweepy library.

After gathering, I will assess the datasets by inspecting for quality (content) issues and tidiness (structural) issues. Finally, I will clean the data addressing each assessment with the Define-Code-Test method.

We are interested in original ratings only (no retweets) that have images. Though there are 5000+ tweets in the dataset, but not all are dog ratings and some are retweets. The requirements of this project are to assess and clean at least 8 quality and 2 tidiness issues in this dataset. Assessing and cleaning the entire dataset completely would require much more time is not necessary to practice and demonstrate data wrangling skills.

Gather

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import requests
import tweepy
import json
import re
import datetime as dt
```

In [2]:
```
# Read in csv file as pandas dataframe and quick check to view structure
twitter_archive = pd.read_csv('twitter-archive-enhanced.csv')
twitter_archive.sample(3)
```

Out[2]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
797	773191612633579521	NaN	NaN	2016-09-06 16:10:20 +0000	<a href="h r...
2021	672082170312290304	NaN	NaN	2015-12-02 15:57:30 +0000	<a href="h r...
1416	698635131305795584	NaN	NaN	2016-02-13 22:29:29 +0000	<a href="h r...

In [3]:
```
# Use Requests library to programmatically download tsv file from a webs
ite
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad
_image-predictions/image-predictions.tsv'
response = requests.get(url)

# Save tsv to file
with open('image_predictions.tsv', mode='wb') as file:
    file.write(response.content)

# Read in tsv file in pandas dataframe and quick check to view structure
image_predictions = pd.read_csv('image_predictions.tsv', sep='\t')
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 [4]:
```
# Authentication Details: load personal API keys (replaced with placehol
ders)
consumer_key = 'MY CONSUMER KEY'
consumer_secret = "MY CONSUMER SECRET KEY"
access_token = 'MY ACCESS TOKEN'
access_secret = 'MY ACCESS SECRET'

# variables for Twitter API connection
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)
api = tweepy.API(auth, wait_on_rate_limit = True)
```

In [5]:
```
# Add each tweet to a new line of tweet_json.txt
with open('tweet_json.txt', 'w', encoding='utf8') as f:
    for tweet_id in twitter_archive['tweet_id']:
        try:
            tweet = api.get_status(tweet_id, tweet_mode='extended')
            json.dump(tweet._json, f)
            f.write('\n')
        except:
            continue
```

In [6]:
```
# Append each tweet into a list
tweets_data = []
tweet_file = open('tweet_json.txt', 'r')

for line in tweet_file:
    try:
        tweet = json.loads(line)
        tweets_data.append(tweet)
    except:
        continue

tweet_file.close()
```

In [7]:
```
# Create dataframe for tweet information
tweet_info = pd.DataFrame()

# Add variables to df: tweet ID, retweet count, favorite count
tweet_info['tweet_id'] = list(map(lambda tweet: tweet['id'], tweets_data
))
tweet_info['retweet_count'] = list(map(lambda tweet: tweet['retweet_coun
t'], tweets_data))
tweet_info['favorite_count'] = list(map(lambda tweet: tweet['favorite_co
unt'], tweets_data))

# Quick check to view df structure
tweet_info.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2342 entries, 0 to 2341
Data columns (total 3 columns):
tweet_id          2342 non-null int64
retweet_count     2342 non-null int64
favorite_count    2342 non-null int64
dtypes: int64(3)
memory usage: 55.0 KB
```

Assess

Dataframe summary:

- twitter_archive
- image_predictions
- tweet_info

Assess each dataframe for quality and tidiness and describe each column variable. Assess both visually and programmatically, and keep key metrics in mind.

Twitter Archive

In [8]:
```
twitter_archive.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
```

There are 2356 entries. There are some missing values: 'in_reply_to_status_id' and 'in_reply_to_user_id' only have 78 entries. There are only 181 retweets ('retweeted_status_x'), but this is ok since we are interested in original tweets only and will remove these. Not all tweets have URLs; we don't have access to additional URLs, however we have another dataframe with photos. twitter_archive columns:

- tweet_id: unique tweet identifier
- in_reply_to_status_id:
- in_reply_to_user_id:
- timestamp: time of the tweet
- source: where the tweet originated (Twitter iPhone, Vine, Twitter web, TweetDeck)
- text: humerous dog caption
- retweeted_status_id: status identifier for retweets
- retweeted_status_user_id: user identifier for retweets
- retweeted_status_timestamp: time of retweet
- expanded_urls: the url where the tweet is housed
- rating_numerator: rated on a scale of 1-10, but most have ratings above the max of 10
- rating_denominator: usually 10 (original maximum)
- name: given name of the dog
- doggo: dog stage (adult)
- floofer: dog stage (fluffy)
- pupper: dog stage (young)
- puppo: dog stage (transitioning from young to adult)

In [9]:
```
twitter_archive.rating_denominator.value_counts()
```

Out[9]:
```
10     2333
11        3
50        3
80        2
20        2
2         1
16        1
40        1
70        1
15        1
90        1
110       1
120       1
130       1
150       1
170       1
7         1
0         1
Name: rating_denominator, dtype: int64
```

In [12]:
```
# Rating is an integer, should it be a float?
twitter_archive['rating_numerator'] = twitter_archive['rating_numerator'
].astype(float)
twitter_archive['rating_numerator'].value_counts()
# Although all these values show '.0' there might be future ratings that
 are decimals, will keep as float
```

Out[12]:
```
12.0      558
11.0      464
10.0      461
13.0      351
9.0       158
8.0       102
7.0        55
14.0       54
5.0        37
6.0        32
3.0        19
4.0        17
1.0         9
2.0         9
75.0        2
15.0        2
420.0       2
0.0         2
144.0       1
666.0       1
121.0       1
182.0       1
165.0       1
17.0        1
45.0        1
204.0       1
960.0       1
1776.0      1
84.0        1
24.0        1
27.0        1
88.0        1
99.0        1
50.0        1
80.0        1
60.0        1
44.0        1
20.0        1
26.0        1
143.0       1
Name: rating_numerator, dtype: int64
```

In [13]:
```
twitter_archive.sample(5)
```

Out[13]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
1653	683462770029932544	NaN	NaN	2016-01-03 01:39:57 +0000	<a href="h r...
965	750429297815552001	NaN	NaN	2016-07-05 20:41:01 +0000	<a href="h r...
204	852936405516943360	NaN	NaN	2017-04-14 17:27:40 +0000	<a href="h r...
1361	703079050210877440	NaN	NaN	2016-02-26 04:48:02 +0000	<a href="h r...
2045	671528761649688577	NaN	NaN	2015-12-01 03:18:27 +0000	<a href="h r...

In [14]:
```
# View entire 'text' string to see if the URL is different from 'expande
d_urls'
pd.set_option('display.max_colwidth', -1)
twitter_archive.head(3)
```

Out[14]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http:/ rel="nofollc
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http:/ rel="nofollc
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	<a href="http:/ rel="nofollc

Expanded_url's are the same, so no need to extract 'https//x' from text.

In [15]:
```
twitter_archive['name'].value_counts()
```

Out[15]:

None	745
a	55
Charlie	12
Cooper	11
Oliver	11
Lucy	11
Penny	10
Lola	10
Tucker	10
Winston	9
Bo	9
Sadie	8
the	8
Daisy	7
Buddy	7
Toby	7
Bailey	7
an	7
Koda	6
Jax	6
Rusty	6
Milo	6
Leo	6
Dave	6
Jack	6
Scout	6
Stanley	6
Bella	6
Oscar	6
very	5
..	..
Wesley	1
Iggy	1
Brockly	1
Gordon	1
Rey	1
Kota	1
Trevith	1
Bobble	1
Kellogg	1
Pancake	1
Lolo	1
Shooter	1
Alexanderson	1
Spark	1
Dallas	1
Strudel	1
Laela	1
by	1
Ralphie	1
Livvie	1
Grady	1
Blipson	1
Jazzy	1
his	1
Harvey	1
Chadrick	1

```
Jeffrie         1
Chaz            1
Miguel          1
Mark            1
Name: name, Length: 957, dtype: int64
```

Not all entries appear to be the correct names nor are they in title case, will clean after merging the datasets.

In [16]:
```
# View number of values in the source column
twitter_archive['source'].value_counts()
```

Out[16]:
```
<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">T
weetDeck</a>    11
Name: source, dtype: int64
```

In [17]:
```
# Make sure all id's are unique, no duplicates
twitter_archive[twitter_archive.tweet_id.duplicated()]
```

Out[17]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_

In [18]:
```
# View numerical descriptions
twitter_archive.describe()
```

Out[18]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	retweeted_status_id	retw
count	2.356000e+03	7.800000e+01	7.800000e+01	1.810000e+02	1.81
mean	7.427716e+17	7.455079e+17	2.014171e+16	7.720400e+17	1.24
std	6.856705e+16	7.582492e+16	1.252797e+17	6.236928e+16	9.59
min	6.660209e+17	6.658147e+17	1.185634e+07	6.661041e+17	7.83
25%	6.783989e+17	6.757419e+17	3.086374e+08	7.186315e+17	4.19
50%	7.196279e+17	7.038708e+17	4.196984e+09	7.804657e+17	4.19
75%	7.993373e+17	8.257804e+17	4.196984e+09	8.203146e+17	4.19
max	8.924206e+17	8.862664e+17	8.405479e+17	8.874740e+17	7.87

Note: The interquartile range of the numerator rating is between 10 and 12. Since the max is 1776 there are likely outliers.

In [19]:
```
twitter_archive['rating_numerator'].sort_values(ascending=False).head(25
)
```

Out[19]:
```
979     1776.0
313     960.0
189     666.0
188     420.0
2074    420.0
1120    204.0
290     182.0
902     165.0
1779    144.0
1634    143.0
1635    121.0
1228    99.0
1843    88.0
433     84.0
1254    80.0
695     75.0
340     75.0
1351    60.0
1202    50.0
1274    45.0
1433    44.0
763     27.0
1712    26.0
516     24.0
1663    20.0
Name: rating_numerator, dtype: float64
```

In [20]:
```
# How many dogs have a rating greater than 10
twitter_archive[twitter_archive.rating_numerator > 20].shape
```

Out[20]:
```
(24, 17)
```

Only 24 entries out of 2356 have a rating above 20.

Image Predictions

In [21]:
```
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
```

There are 2075 entries and no missing values. image_predictions columns:

- tweet_id: unique tweet identifier
- jpg_url: image of the dog
- img_num: number out of 4 possible images (most are 1)
- p1: the algorithm's first prediction for the image in the tweet
- p1_conf: is how confident the algorithm is in its first prediction
- p1_dog: is whether or not the first prediction is a breed of dog
- p2: the algorithm's second most likely prediction
- p3: the algorithm's third most likely prediction

In [22]:
```
image_predictions.sample(5)
```

Out[22]:

	tweet_id	jpg_url	img_n
629	680913438424612864	https://pbs.twimg.com/media/CXMXKKHUMAA1QN3.jpg	1
481	675362609739206656	https://pbs.twimg.com/media/CV9etctWUAAl5Hp.jpg	1
693	684225744407494656	https://pbs.twimg.com/media/CX7br3HWsAAQ9L1.jpg	2
2029	882762694511734784	https://pbs.twimg.com/media/DEAz_HHXsAA-p_z.jpg	1
1914	854120357044912130	https://pbs.twimg.com/media/C9px7jyVwAAnmwN.jpg	4

In [23]:
```
# How many first predictions are actually dogs
image_predictions['p1_dog'].value_counts()
```

Out[23]:
```
True     1532
False    543
Name: p1_dog, dtype: int64
```

In [24]:
```
# How many second predictions are not dogs
(image_predictions.p2_dog == False).sum()
```

Out[24]: 522

In [25]:
```
# How many third predictions are not dogs
(image_predictions.p3_dog == False).sum()
```

Out[25]: 576

In [26]:
```
# Find rows where p1, p2, p3 are all false (first line of code is the co
unt)
# image_predictions[(image_predictions['p1_dog']==False) & (image_predic
tions['p2_dog']==False) & (image_predictions['p3_dog']==False)].count()
image_predictions[(image_predictions['p1_dog']==False) & (image_predicti
ons['p2_dog']==False) & (image_predictions['p3_dog']==False)].sample(5)
```

Out[26]:

	tweet_id	jpg_url	img_nu
512	676215927814406144	https://pbs.twimg.com/media/CWJmzNsWUAE706Z.jpg	1
166	668981893510119424	https://pbs.twimg.com/media/CUize-0WEAAerAK.jpg	1
117	668142349051129856	https://pbs.twimg.com/media/CUW37BzWsAAlJIN.jpg	1
296	671362598324076544	https://pbs.twimg.com/media/CVEouDRXAAEe8mt.jpg	1
1134	728653952833728512	https://pbs.twimg.com/media/Chyy5lQWUAEzxSL.jpg	2

There are 324 entries that not dogs (p1, p2, p3 are all false)

In [27]:
```
# What kind of dogs are in the first prediction?
image_predictions['p1'].value_counts()
```

Out[27]:

golden_retriever	150
Labrador_retriever	100
Pembroke	89
Chihuahua	83
pug	57
chow	44
Samoyed	43
toy_poodle	39
Pomeranian	38
cocker_spaniel	30
malamute	30
French_bulldog	26
Chesapeake_Bay_retriever	23
miniature_pinscher	23
seat_belt	22
Siberian_husky	20
German_shepherd	20
Staffordshire_bullterrier	20
Cardigan	19
web_site	19
teddy	18
Maltese_dog	18
beagle	18
Eskimo_dog	18
Shetland_sheepdog	18
Rottweiler	17
Shih-Tzu	17
Lakeland_terrier	17
kuvasz	16
Italian_greyhound	16
	..
starfish	1
cheetah	1
bison	1
beach_wagon	1
canoe	1
teapot	1
walking_stick	1
school_bus	1
beaver	1
pedestal	1
peacock	1
restaurant	1
bighorn	1
harp	1
handkerchief	1
limousine	1
conch	1
lynx	1
lion	1
hummingbird	1
mud_turtle	1
panpipe	1
lacewing	1
marmot	1
pool_table	1
king_penguin	1

```
black-footed_ferret          1
Egyptian_cat                 1
groenendael                  1
leaf_beetle                  1
Name: p1, Length: 378, dtype: int64
```

In [28]:
```
# Check for tweet duplicates in image_predictions
image_predictions[image_predictions.tweet_id.duplicated()]
```

Out[28]:

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_

Tweet Additional Information

In [29]:
```
tweet_info.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2342 entries, 0 to 2341
Data columns (total 3 columns):
tweet_id          2342 non-null int64
retweet_count     2342 non-null int64
favorite_count    2342 non-null int64
dtypes: int64(3)
memory usage: 55.0 KB
```

There are 2342 entries and no missing values. tweet_info columns:

- tweet_id: unique tweet identifier
- retweet_count: number of retweets a tweet received
- favorite_count: number of favorites a tweets received

In [30]:
```
tweet_info.sample(5)
```

Out[30]:

	tweet_id	retweet_count	favorite_count
1838	675707330206547968	740	2077
1367	700890391244103680	621	2361
667	789280767834746880	5526	0
599	797545162159308800	5424	15726
2088	670679630144274432	302	767

In [33]:
```
# Make sure all id's are unique - no duplicates
print(sum(tweet_info.groupby('tweet_id')['tweet_id'].nunique())) # sum o
f unique values
print(sum(tweet_info.tweet_id.duplicated())) # Sum of duplicates
```

```
2342
0
```

Assessment Observations

Low quality, also known as dirty, data has content issues such as missing, invalid, inaccurate, and inconsistent data. Untidy, also known as messy, data has structural issues: each variable should form a column, each observation should form a row, and each observational unit a table. Assessment observations are not action items; actions items will be defined when cleaning.

Quality

- Remove unnecessary columns
- twitter_archive, remove rows that are retweets (181 rows where retweeted_x have values)
- twitter_archive, convert timestamp to datetime object
- twitter_archive, update source column from url to text
- twitter_archive, fix rating_numerator that are not extracted properly (those that have decimals)
- twitter_archive, make all values in ratings_denominator '10' for consistency (or remove col)
- twitter_archive, convert non-dog names to 'None' then make title case
- twitter_archive, make names Title case
- twitter_archive, check rating_numerator outliers - there are only 24 values over 15 (review manually)
- twitter_archive, tweet_id datatype is an integer, convert to string (object)
- image_predictions, Remove non-dogs, the 324 rows where p1, p2, and p3 are false
- image_predictions, Update p1, p2, p3 to title text and remove underscores

Tidiness

- twitter_archive, gather dog stages (doggo, puppo, pupper, floofer) into one column 'dog_stage'
- twitter_archive, parse timestamp into separate columns: year, month, day, time (not necessary, keep timestamp column, but also want to view most popular days and months of tweets)
- image_predictions, create a 'prediction' column (Dog, Maybe Dog, Not Dog)
- Join tweet_info, twitter_archive, and image_predictions into one master dataset on 'tweet_id'

Clean

I'll create copies of the dataframes to use for cleaning and keep the originals intact for future reference.

- **archive_clean** (original df: **twitter_archive**)
- **image_clean** (original df: **image_predictions**)
- **tweet_clean** (original df: **tweet_info**)

Finally, I'll join all 3 datasets into one master: **twitter_archive_master**

Step one in cleaning is to address missing data. Next I'll address tidiness issues and finally quality issues.

In [767]:
```
# Create copies of original dataframes
archive_clean = twitter_archive.copy()
image_clean = image_predictions.copy()
tweet_clean = tweet_info.copy()
```

In [768]:
```
archive_clean.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 float64
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(5), int64(2), object(10)
memory usage: 313.0+ KB
```

Define: Fix missing data in archive_clean. Remove rows with 'retweeted_status_x' since we are interested in original tweets only. Check to make sure the values are decreased by 181, the number of retweets (2356 -> 2175) then drop those columns. Drop the 'in_reply_to_x' columns as these are unneccessary.

Code:

In [769]:
```
archive_clean.drop(archive_clean[archive_clean.retweeted_status_id.notnu
ll()].index, inplace=True)
archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 17 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 float64
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
dtypes: float64(5), int64(2), object(10)
memory usage: 305.9+ KB
```

In [770]:
```
archive_clean.drop(['retweeted_status_id',
                    'retweeted_status_user_id',
                    'retweeted_status_timestamp',
                    'in_reply_to_status_id',
                    'in_reply_to_user_id'], axis=1, inplace=True)
```

Test

In [771]:
```
archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 12 columns):
tweet_id              2175 non-null int64
timestamp             2175 non-null object
source                2175 non-null object
text                  2175 non-null object
expanded_urls         2117 non-null object
rating_numerator      2175 non-null float64
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
dtypes: float64(1), int64(2), object(9)
memory usage: 220.9+ KB
```

Define: Combine dog stage columns (doggo, floofer, pupper, puppo) into one 'dog_stage' column. Delete the separate dog stage categories after visually inspecting a random sample to ensure the combination worked accurately. Convert entries in this column title case.

Code

In [772]:
```
# Replace empty entries with a blank
archive_clean.doggo.replace('None', '', inplace=True)
archive_clean.floofer.replace('None', '', inplace=True)
archive_clean.pupper.replace('None', '', inplace=True)
archive_clean.puppo.replace('None', '', inplace=True)

# Create a new column for dog_stage
archive_clean['dog_stage'] = archive_clean.doggo + archive_clean.floofer
 + archive_clean.pupper + archive_clean.puppo
archive_clean.dog_stage.value_counts()
```

Out[772]:
```
                1831
pupper           224
doggo             75
puppo             24
doggopupper       10
floofer            9
doggofloofer       1
doggopuppo         1
Name: dog_stage, dtype: int64
```

In [773]:
```
# Quick test to make sure dog_stage is accurate (compare to old columns)
archive_clean[['doggo', 'floofer', 'pupper', 'puppo', 'dog_stage']].samp
le(10)
```

Out[773]:

	doggo	floofer	pupper	puppo	dog_stage
296					
1681					
239					
452					
718					
1547					
54					
1321			pupper		pupper
482					
2143					

In [774]:
```
# Rename values in dog_stage column
archive_clean.loc[archive_clean.dog_stage == 'pupper', 'dog_stage'] = 'P
upper'
archive_clean.loc[archive_clean.dog_stage == 'doggo', 'dog_stage'] = 'Do
ggo'
archive_clean.loc[archive_clean.dog_stage == 'puppo', 'dog_stage'] = 'Pu
ppo'
archive_clean.loc[archive_clean.dog_stage == 'doggopupper', 'dog_stage']
 = 'Doggo, Pupper'
archive_clean.loc[archive_clean.dog_stage == 'floofer', 'dog_stage'] =
'Floofer'
archive_clean.loc[archive_clean.dog_stage == 'doggopuppo', 'dog_stage']
= 'Doggo, Puppo'
archive_clean.loc[archive_clean.dog_stage == 'doggofloofer', 'dog_stage'
] = 'Doggo, Floofer'

# Replace blank cells with NaNs
archive_clean.loc[archive_clean.dog_stage == '', 'dog_stage'] = np.nan

# Replace NaNs with text so we have non-null values
archive_clean.dog_stage = archive_clean.dog_stage.fillna('Unknown')

archive_clean.dog_stage.value_counts()
```

Out[774]:
```
Unknown           1831
Pupper             224
Doggo               75
Puppo               24
Doggo, Pupper       10
Floofer              9
Doggo, Puppo         1
Doggo, Floofer       1
Name: dog_stage, dtype: int64
```

In [775]:
```
# Drop unnecessary columns
archive_clean.drop(['doggo', 'floofer', 'pupper', 'puppo'], axis=1, inpl
ace=True)
```

Test

In [776]:
```
archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 9 columns):
tweet_id              2175 non-null int64
timestamp             2175 non-null object
source                2175 non-null object
text                  2175 non-null object
expanded_urls         2117 non-null object
rating_numerator      2175 non-null float64
rating_denominator    2175 non-null int64
name                  2175 non-null object
dog_stage             2175 non-null object
dtypes: float64(1), int64(2), object(6)
memory usage: 169.9+ KB
```

In [777]:
```
archive_clean.dog_stage.sample(5)
```

Out[777]:
```
2144    Unknown
1378    Unknown
1834    Unknown
1553    Unknown
2184    Unknown
Name: dog_stage, dtype: object
```

Define: Replace 4 source links with text string defining the link.

In [778]:
```
archive_clean.source.value_counts()
```

Out[778]:
```
<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for
iPhone</a>     2042
<a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
                91
<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
                31
<a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">T
weetDeck</a>    11
Name: source, dtype: int64
```

Code

In [779]:
```
# Text replacements
source_txt = {'<a href="http://twitter.com/download/iphone" rel="nofollo
w">Twitter for iPhone</a>': 'Twitter for iPhone',
               '<a href="http://vine.co" rel="nofollow">Vine - Make a S
cene</a>': 'Vine - Make a Scene',
               '<a href="http://twitter.com" rel="nofollow">Twitter Web
 Client</a>': 'Twitter Web Client',
               '<a href="https://about.twitter.com/products/tweetdeck"
 rel="nofollow">TweetDeck</a>': 'TweetDeck'}

# Apply function
def text_source(archive_clean):
    if archive_clean['source'] in source_txt.keys():
        abbrev = source_txt[archive_clean['source']]
        return abbrev
    else:
        return archive_clean['source']

archive_clean['source'] = archive_clean.apply(text_source, axis=1)
```

Test

In [780]:
```
archive_clean.source.value_counts()
```

Out[780]:
```
Twitter for iPhone     2042
Vine - Make a Scene    91
Twitter Web Client     31
TweetDeck              11
Name: source, dtype: int64
```

Define: Combine tweet_clean and archive_clean, via inner join (default) on 'tweet_id'.

In [781]:
```
archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 9 columns):
tweet_id              2175 non-null int64
timestamp             2175 non-null object
source                2175 non-null object
text                  2175 non-null object
expanded_urls         2117 non-null object
rating_numerator      2175 non-null float64
rating_denominator    2175 non-null int64
name                  2175 non-null object
dog_stage             2175 non-null object
dtypes: float64(1), int64(2), object(6)
memory usage: 169.9+ KB
```

In [782]:
```
tweet_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2342 entries, 0 to 2341
Data columns (total 3 columns):
tweet_id          2342 non-null int64
retweet_count     2342 non-null int64
favorite_count    2342 non-null int64
dtypes: int64(3)
memory usage: 55.0 KB
```

Code

In [783]:
```
twitter_archive_master = pd.merge(archive_clean, tweet_clean, on='tweet_
id', how = 'inner')
```

Test

In [784]:
```
# ensure that the new master includes columnsn from both archive_clean a
nd tweet_clean
twitter_archive_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2174 entries, 0 to 2173
Data columns (total 11 columns):
tweet_id              2174 non-null int64
timestamp             2174 non-null object
source                2174 non-null object
text                  2174 non-null object
expanded_urls         2116 non-null object
rating_numerator      2174 non-null float64
rating_denominator    2174 non-null int64
name                  2174 non-null object
dog_stage             2174 non-null object
retweet_count         2174 non-null int64
favorite_count        2174 non-null int64
dtypes: float64(1), int64(4), object(6)
memory usage: 203.8+ KB
```

Define: Convert timestamp to datetime and spread into 4 columns for year, month, day, and time. Keep the timestamp (datetime) column for visualizations.

In [785]:
```
from datetime import date

# Convert timestamp to datetime
twitter_archive_master['timestamp'] = pd.to_datetime(twitter_archive_mas
ter['timestamp'])

# Extract datetime to new year, month, day, time columns
twitter_archive_master['year'] = twitter_archive_master['timestamp'].dt.
year # separate year, month, day, time
twitter_archive_master['month'] = twitter_archive_master['timestamp'].dt
.month
twitter_archive_master['day'] = twitter_archive_master['timestamp'].dt.d
ay
twitter_archive_master['time'] = twitter_archive_master['timestamp'].dt.
time

# Create day of week column
twitter_archive_master['weekday'] = twitter_archive_master['timestamp'].
dt.dayofweek
days = {0:'Mon',1:'Tues',2:'Weds',3:'Thurs',4:'Fri',5:'Sat',6:'Sun'}
twitter_archive_master['weekday'] = twitter_archive_master['weekday'].ap
ply(lambda x: days[x])

twitter_archive_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2174 entries, 0 to 2173
Data columns (total 16 columns):
tweet_id              2174 non-null int64
timestamp             2174 non-null datetime64[ns]
source                2174 non-null object
text                  2174 non-null object
expanded_urls         2116 non-null object
rating_numerator      2174 non-null float64
rating_denominator    2174 non-null int64
name                  2174 non-null object
dog_stage             2174 non-null object
retweet_count         2174 non-null int64
favorite_count        2174 non-null int64
year                  2174 non-null int64
month                 2174 non-null int64
day                   2174 non-null int64
time                  2174 non-null object
weekday               2174 non-null object
dtypes: datetime64[ns](1), float64(1), int64(7), object(7)
memory usage: 288.7+ KB
```

In [786]:
```
twitter_archive_master['weekday'].value_counts()
```

Out[786]:
```
Mon      357
Tues     326
Weds     322
Thurs    305
Fri      304
Sat      284
Sun      276
Name: weekday, dtype: int64
```

Define: Create a new column for the dog prediction summary in image_prediction:

- When all three predictions are true, insert text 'Dog'
- When all three predictions are false, insert text 'Not Dog'
- When 1 or 2 precictions are true, insert text 'Maybe Dog'

Code

In [787]:
```
# convert p1_dog, p2_dog, p3_dog to an integer (True=1, False=0)
prediction_summary = ['p1_dog', 'p2_dog', 'p3_dog']

for p in prediction_summary:
    image_clean[p] = image_clean[p].astype(int)

# Create a new column that adds the total number of True and False for t
he 3 predictions
image_clean['prediction'] = image_clean.p1_dog + image_clean.p2_dog + im
age_clean.p3_dog

# Replace the number with a defining text string
image_clean['prediction'] = image_clean['prediction'].replace(3, 'Dog')
image_clean['prediction'] = image_clean['prediction'].replace(2, 'Maybe
 Dog')
image_clean['prediction'] = image_clean['prediction'].replace(1, 'Maybe
 Dog')
image_clean['prediction'] = image_clean['prediction'].replace(0, 'Not Do
g')
```

Test

In [788]:
```
image_clean[['p1_dog', 'p2_dog', 'p3_dog','prediction']].sample(10)
```

Out[788]:

	p1_dog	p2_dog	p3_dog	prediction
1409	0	0	1	Maybe Dog
0	1	1	1	Dog
1351	1	1	1	Dog
1941	1	1	0	Maybe Dog
699	1	1	1	Dog
652	1	1	1	Dog
1633	1	1	1	Dog
1286	1	1	0	Maybe Dog
2031	1	1	1	Dog
965	1	1	1	Dog

In [789]:
```
image_clean.prediction.value_counts()
```

Out[789]:
```
Dog          1243
Maybe Dog    508
Not Dog      324
Name: prediction, dtype: int64
```

Define: We see above that the image prediction column worked accurately. Now we can drop the extraneous p1_dog, p2_dog, and p3_dog columns for simplicity, along with image_num.

Code

In [790]:
```
image_clean.drop(['p1_dog', 'p2_dog', 'p3_dog', 'img_num'], axis=1, inpl
ace=True)
```

Test

In [791]:
```
image_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 9 columns):
tweet_id      2075 non-null int64
jpg_url       2075 non-null object
p1            2075 non-null object
p1_conf       2075 non-null float64
p2            2075 non-null object
p2_conf       2075 non-null float64
p3            2075 non-null object
p3_conf       2075 non-null float64
prediction    2075 non-null object
dtypes: float64(3), int64(1), object(5)
memory usage: 146.0+ KB
```

Define: Join the image_clean df to the twitter_archive_master df (default = inner join)

Code

In [792]:
```
twitter_archive_master = pd.merge(twitter_archive_master, image_clean, o
n='tweet_id')
```

Test

In [793]:
```
twitter_archive_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1993 entries, 0 to 1992
Data columns (total 24 columns):
tweet_id              1993 non-null int64
timestamp             1993 non-null datetime64[ns]
source                1993 non-null object
text                  1993 non-null object
expanded_urls         1993 non-null object
rating_numerator      1993 non-null float64
rating_denominator    1993 non-null int64
name                  1993 non-null object
dog_stage             1993 non-null object
retweet_count         1993 non-null int64
favorite_count        1993 non-null int64
year                  1993 non-null int64
month                 1993 non-null int64
day                   1993 non-null int64
time                  1993 non-null object
weekday               1993 non-null object
jpg_url               1993 non-null object
p1                    1993 non-null object
p1_conf               1993 non-null float64
p2                    1993 non-null object
p2_conf               1993 non-null float64
p3                    1993 non-null object
p3_conf               1993 non-null float64
prediction            1993 non-null object
dtypes: datetime64[ns](1), float64(4), int64(7), object(12)
memory usage: 389.3+ KB
```

In [794]:
```
twitter_archive_master.prediction.value_counts()
```

Out[794]:
```
Dog          1202
Maybe Dog    483
Not Dog      308
Name: prediction, dtype: int64
```

We now have one combined dataset with 1993 entries and a full set of values.

In [795]:
```
twitter_archive_master.sample(3)
```

Out[795]:

	tweet_id	timestamp	source	text	
1860	668484198282485761	2015-11-22 17:40:27	Twitter for iPhone	Good teamwork between these dogs. One is on lookout while other eats. Long necks. Nice big house. 9/10s good pups https://t.co/uXgmECGYEB	https://twitter.cc
1651	672264251789176834	2015-12-03 04:01:02	Twitter for iPhone	This is Kreg. He has the eyes of a tyrannical dictator. Will not rest until household is his. 10/10 https://t.co/TUeuaOmunV	https://twitter.cc
32	885167619883638784	2017-07-12 16:03:00	Twitter for iPhone	Here we have a corgi undercover as a malamute. Pawbably doing important investigative work. Zero control over tongue happenings. 13/10 https://t.co/44ltaMubBf	https://twitter.cc

3 rows × 24 columns

Note: we lost some source information when joining the 3 datasets. For visualizations, might want to use archive_clean for a more complete view of sources.

In [796]:
```
archive_clean.source.value_counts()
```

Out[796]:
```
Twitter for iPhone     2042
Vine - Make a Scene    91
Twitter Web Client     31
TweetDeck              11
Name: source, dtype: int64
```

Define: We could change the denominator rating to 10 for all entries, but it makes more sense to drop this column and rename the rating_numerator to rating for simplicity.

Code

In [797]:
```
# twitter_archive_master['rating_denominator'] = 10
twitter_archive_master.drop(['rating_denominator'], axis=1, inplace=True
)
twitter_archive_master.rename(columns={'rating_numerator': 'rating'}, in
place=True)
```

Test

In [798]:
```
twitter_archive_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1993 entries, 0 to 1992
Data columns (total 23 columns):
tweet_id          1993 non-null int64
timestamp         1993 non-null datetime64[ns]
source            1993 non-null object
text              1993 non-null object
expanded_urls     1993 non-null object
rating            1993 non-null float64
name              1993 non-null object
dog_stage         1993 non-null object
retweet_count     1993 non-null int64
favorite_count    1993 non-null int64
year              1993 non-null int64
month             1993 non-null int64
day               1993 non-null int64
time              1993 non-null object
weekday           1993 non-null object
jpg_url           1993 non-null object
p1                1993 non-null object
p1_conf           1993 non-null float64
p2                1993 non-null object
p2_conf           1993 non-null float64
p3                1993 non-null object
p3_conf           1993 non-null float64
prediction        1993 non-null object
dtypes: datetime64[ns](1), float64(4), int64(6), object(12)
memory usage: 373.7+ KB
```

Define: Clean 'name' column. Convert non-names to 'None'.

In [799]:
```
#twitter_archive_master['name'].value_counts()
twitter_archive_master['name'].str.lower()
```

```
Out[799]: 0       phineas
          1       tilly
          2       archie
          3       darla
          4       franklin
          5       none
          6       jax
          7       none
          8       zoey
          9       cassie
          10      koda
          11      bruno
          12      none
          13      ted
          14      stuart
          15      oliver
          16      jim
          17      zeke
          18      ralphus
          19      gerald
          20      jeffrey
          21      such
          22      canela
          23      none
          24      none
          25      maya
          26      mingus
          27      derek
          28      roscoe
          29      waffles
                   ...
          1963    quite
          1964    a
          1965    none
          1966    none
          1967    none
          1968    none
          1969    none
          1970    an
          1971    a
          1972    an
          1973    none
          1974    none
          1975    none
          1976    none
          1977    none
          1978    none
          1979    none
          1980    none
          1981    none
          1982    the
          1983    the
          1984    a
          1985    a
          1986    an
          1987    a
          1988    none
```

```
1989    a
1990    a
1991    a
1992    none
Name: name, Length: 1993, dtype: object
```

In [800]:
```
wrong_name = twitter_archive_master.name.str.islower()
twitter_archive_master.loc[wrong_name, 'name'] = 'None'

# Convert names to title case
twitter_archive_master.name = twitter_archive_master.name.str.title()
```

Test

In [801]:
```
twitter_archive_master['name'].value_counts()
```

Out[801]:

None	644
Charlie	10
Cooper	10
Lucy	10
Oliver	10
Tucker	9
Penny	9
Winston	8
Sadie	8
Lola	7
Toby	7
Daisy	7
Jax	6
Koda	6
Bella	6
Stanley	6
Bo	6
Rusty	5
Oscar	5
Buddy	5
Scout	5
Louis	5
Chester	5
Milo	5
Bailey	5
Leo	5
Dave	5
Gus	4
Reggie	4
Derek	4
..	..
Willem	1
Brandi	1
Pippin	1
Eugene	1
Rodman	1
Kellogg	1
Quinn	1
Laela	1
Kobe	1
Fizz	1
Bloo	1
Arya	1
Fillup	1
Naphaniel	1
Robin	1
Genevieve	1
Akumi	1
Samsom	1
Florence	1
Strudel	1
Grizzwald	1
Tuco	1
Bobble	1
Stubert	1
Lolo	1
Shooter	1

```
Alexanderson    1
Spark           1
Dallas          1
Mark            1
Name: name, Length: 914, dtype: int64
```

Define: For all predictions (p1, p2, p3), remove underscores and make title case.

Code

In [802]:
```
predictions = ['p1', 'p2', 'p3']

for p in predictions:
    twitter_archive_master[p] = twitter_archive_master[p].str.title().st
r.replace('_', " ")
```

Test

In [803]:
```
twitter_archive_master[['p1', 'p2', 'p3']].sample(10)
```

Out[803]:

	p1	p2	p3
304	Schipperke	Curly-Coated Retriever	Labrador Retriever
430	Golden Retriever	Kuvasz	Labrador Retriever
1067	Bloodhound	Sussex Spaniel	Clumber
1274	Papillon	Toy Terrier	Cardigan
95	Comic Book	Envelope	Book Jacket
1706	Pitcher	Sunglasses	Mask
613	Toy Poodle	Miniature Poodle	Irish Terrier
834	Golden Retriever	Chow	Labrador Retriever
1345	Chihuahua	Doormat	Toy Terrier
552	German Shepherd	Malinois	Norwegian Elkhound

Define: Convert confidence levels to a percentage by multiplying by 100, converting the float to an integer, and displaying only 2 numbers.

Code

In [804]:
```
confidence = ['p1_conf', 'p2_conf', 'p3_conf']

for c in confidence:
    twitter_archive_master[c] = round(twitter_archive_master[c]*100).ast
ype(int)
```

Test

In [805]:
```
twitter_archive_master[['p1_conf', 'p2_conf', 'p3_conf']].sample(5)
```

Out[805]:

	p1_conf	p2_conf	p3_conf
1460	55	8	5
799	81	10	2
1359	41	35	15
456	86	4	2
258	53	18	10

In [806]:
```
twitter_archive_master.prediction.value_counts()
```

Out[806]:
```
Dog          1202
Maybe Dog    483
Not Dog      308
Name: prediction, dtype: int64
```

In [807]:
```
twitter_archive_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1993 entries, 0 to 1992
Data columns (total 23 columns):
tweet_id          1993 non-null int64
timestamp         1993 non-null datetime64[ns]
source            1993 non-null object
text              1993 non-null object
expanded_urls     1993 non-null object
rating            1993 non-null float64
name              1993 non-null object
dog_stage         1993 non-null object
retweet_count     1993 non-null int64
favorite_count    1993 non-null int64
year              1993 non-null int64
month             1993 non-null int64
day               1993 non-null int64
time              1993 non-null object
weekday           1993 non-null object
jpg_url           1993 non-null object
p1                1993 non-null object
p1_conf           1993 non-null int64
p2                1993 non-null object
p2_conf           1993 non-null int64
p3                1993 non-null object
p3_conf           1993 non-null int64
prediction        1993 non-null object
dtypes: datetime64[ns](1), float64(1), int64(9), object(12)
memory usage: 453.7+ KB
```

Define: Remove non-dogs (308 entries) from master dataset.

Code

In [808]:
```
twitter_archive_master = twitter_archive_master[twitter_archive_master[
'prediction'] != "Not Dog"]
```

Test

In [809]:
```
twitter_archive_master.prediction.value_counts()
```

Out[809]:
```
Dog          1202
Maybe Dog    483
Name: prediction, dtype: int64
```

In [810]:
```
twitter_archive_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1685 entries, 1 to 1992
Data columns (total 23 columns):
tweet_id          1685 non-null int64
timestamp         1685 non-null datetime64[ns]
source            1685 non-null object
text              1685 non-null object
expanded_urls     1685 non-null object
rating            1685 non-null float64
name              1685 non-null object
dog_stage         1685 non-null object
retweet_count     1685 non-null int64
favorite_count    1685 non-null int64
year              1685 non-null int64
month             1685 non-null int64
day               1685 non-null int64
time              1685 non-null object
weekday           1685 non-null object
jpg_url           1685 non-null object
p1                1685 non-null object
p1_conf           1685 non-null int64
p2                1685 non-null object
p2_conf           1685 non-null int64
p3                1685 non-null object
p3_conf           1685 non-null int64
prediction        1685 non-null object
dtypes: datetime64[ns](1), float64(1), int64(9), object(12)
memory usage: 315.9+ KB
```

In [811]:
```
twitter_archive_master.sample(3)
```

Out[811]:

	tweet_id	timestamp	source	text	
1769	670338931251150849	2015-11-27 20:30:30	Twitter for iPhone	This is Butters. He's not ready for Thanksgiving to be over. 10/10 poor Butters https://t.co/iTc578yDmY	https://twitter.com
1624	672975131468300288	2015-12-05 03:05:49	Twitter for iPhone	This is Chuckles. He is one skeptical pupper. 10/10 stay woke Chuckles https://t.co/ZlcF0TIRW1	https://twitter.com
926	716080869887381504	2016-04-02 01:52:38	Twitter for iPhone	Here's a super majestic doggo and a sunset 11/10 https://t.co/UACnoyi8zu	https://twitter.com

3 rows × 23 columns

Define: Rename p1, p2, p2 with more obvious names (prediction_x).

Code

In [812]:
```
twitter_archive_master.rename(columns={'p1': 'prediction_1',
                                       'p2': 'prediction_2',
                                       'p3': 'prediction_3'}, inplace=True)
```

Test

In [813]:
```
twitter_archive_master[['prediction_1', 'prediction_2', 'prediction_3']]
.sample(10)
```

Out[813]:

	prediction_1	prediction_2	prediction_3
547	German Shepherd	Malinois	Kelpie
759	Golden Retriever	Labrador Retriever	Seat Belt
96	Pembroke	Cardigan	Malamute
1754	Sussex Spaniel	Otterhound	Irish Terrier
1094	Golden Retriever	Irish Setter	Labrador Retriever
1499	Golden Retriever	Kuvasz	Saluki
1274	Papillon	Toy Terrier	Cardigan
375	Golden Retriever	Labrador Retriever	Kuvasz
87	Shetland Sheepdog	Collie	Pomeranian
187	Golden Retriever	Labrador Retriever	Tibetan Mastiff

Define: Convert tweet_id from an integer to a string (object) since we are not intending to perform math calculations with this data.

Code

In [814]:
```
twitter_archive_master['tweet_id'] = twitter_archive_master['tweet_id'].
astype(str)
```

In [815]:
```
twitter_archive_master['tweet_id'].describe()
```

Out[815]:
```
count     1685
unique    1685
top       850145622816686080
freq      1
Name: tweet_id, dtype: object
```

Define: Investigate numerators that might be inaccurate by extracting decimals.

In [816]:
```
# twitter_archive_master['rating'].astype(float)
twitter_archive_master['rating'] = twitter_archive_master.text.str.extra
ct('(\d[.,]?\d+)')
twitter_archive_master['rating'].unique()
```

```
/Users/karenbevis/anaconda3/lib/python3.6/site-packages/ipykernel/__mai
n__.py:2: FutureWarning: currently extract(expand=None) means expand=Fa
lse (return Index/Series/DataFrame) but in a future version of pandas t
his will be changed to expand=True (return DataFrame)
  from ipykernel import kernelapp as app
```

Out[816]:
```
array(['13', '12', '14', '11', '10', '236', '60', '84', '24', '98',
       '9.75', '46', '100', '165', '50', '17', '2002', '2.0', '47', '2
0',
       '99', '80', '45', '400', '44', '31', '33', '97', '61', '143',
       '121', '260', '2015', '92', '144', '88', '85', '8.98', '1949'],
      dtype=object)
```

In [817]:
```
# locate tweed_ids for the suspect numerators
twitter_archive_master[(twitter_archive_master['rating'] == '9.75') |
                       (twitter_archive_master['rating'] == '2.0') |
                       (twitter_archive_master['rating'] == '8.98') |
                       (twitter_archive_master['rating'] == '165') |
                       (twitter_archive_master['rating'] == '260') |
                       (twitter_archive_master['rating'] == '2015') |
                       (twitter_archive_master['rating'] == '2002') |
                       (twitter_archive_master['rating'] == '1949')]
```

Out[817]:

	tweet_id	timestamp	source	text	
503	786709082849828864	2016-10-13 23:23:56	Twitter for iPhone	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	https://twitte
662	758467244762497024	2016-07-28 01:00:57	Twitter for iPhone	Why does this never happen at my front door... 165/150 https://t.co/HmwrdfEfUE	https://twitte
718	750086836815486976	2016-07-04 22:00:12	TweetDeck	This is Spanky. He was a member of the 2002 USA Winter Olympic speed skating team. Accomplished af. 12/10 https://t.co/7tlZPrePXd	https://twitte
757	746818907684614144	2016-06-25 21:34:37	Twitter for iPhone	Guys... Dog Jesus 2.0\n13/10 buoyant af https://t.co/CuNA7OwfKQ	https://twitte
1314	683773439333797890	2016-01-03 22:14:26	Twitter for iPhone	This is Buddy. He's gaining strength. Currently an F4 tornado with wind speeds up to 260mph. Very devastating. 9/10 https://t.co/qipZbshNsR	https://twitte
1326	683030066213818368	2016-01-01 21:00:32	Twitter for iPhone	This is Lulu. She's contemplating all her unreached 2015 goals and daydreaming of a more efficient tomorrow. 10/10 https://t.co/h3ScYuz77J	https://twitte
1331	682662431982772225	2015-12-31 20:39:41	Twitter for iPhone	Meet Joey and Izzy. Joey only has one ear that works and Izzy wants 2015 to be over already. Both great pups. 11/10s https://t.co/WgQTlQ93BB	https://twitte

	tweet_id	timestamp	source	text	
1984	666057090499244032	2015-11-16 00:55:59	Twitter for iPhone	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	https://twitte
1988	666049248165822465	2015-11-16 00:24:50	Twitter for iPhone	Here we have a 1949 1st generation vulpix. Enjoys sweat tea and Fox News. Cannot be phased. 5/10 https://t.co/4B7cOc1EDq	https://twitte

9 rows × 23 columns

Code

In [818]:
```
twitter_archive_master.loc[twitter_archive_master.tweet_id == '786709082
849828864', 'rating'] = 10 # replace 9.75, round up
twitter_archive_master.loc[twitter_archive_master.tweet_id == '746818907
684614144', 'rating'] = 13 # replace 2.0
twitter_archive_master.loc[twitter_archive_master.tweet_id == '666057090
499244032', 'rating'] = 9 # replace 8.98
twitter_archive_master.loc[twitter_archive_master.tweet_id == '683773439
333797890', 'rating'] = 9 # replace 260
twitter_archive_master.loc[twitter_archive_master.tweet_id == '683030066
213818368', 'rating'] = 10 # replace 2015
twitter_archive_master.loc[twitter_archive_master.tweet_id == '682662431
982772225', 'rating'] = 11 # replace 2015
twitter_archive_master.loc[twitter_archive_master.tweet_id == '666049248
165822465', 'rating'] = 5 # replace 1949
twitter_archive_master.loc[twitter_archive_master.tweet_id == '750086836
815486976', 'rating'] = 12 # replace 2002

# replace 165 (165/150 = 11/10)
twitter_archive_master.loc[twitter_archive_master.tweet_id == '758467244
762497024', 'rating'] = 11
```

Test

In [819]:
```
twitter_archive_master[(twitter_archive_master['tweet_id'] == '786709082
849828864') |
                       (twitter_archive_master['tweet_id'] == '746818907
684614144') |
                       (twitter_archive_master['tweet_id'] == '666057090
499244032') |
                       (twitter_archive_master['tweet_id'] == '683773439
333797890') |
                       (twitter_archive_master['tweet_id'] == '683030066
213818368') |
                       (twitter_archive_master['tweet_id'] == '682662431
982772225') |
                       (twitter_archive_master['tweet_id'] == '666049248
165822465') |
                       (twitter_archive_master['tweet_id'] == '758467244
762497024') |
                       (twitter_archive_master['tweet_id'] == '750086836
815486976')]
```

Out[819]:

	tweet_id	timestamp	source	text	
503	786709082849828864	2016-10-13 23:23:56	Twitter for iPhone	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	https://twitte
662	758467244762497024	2016-07-28 01:00:57	Twitter for iPhone	Why does this never happen at my front door... 165/150 https://t.co/HmwrdfEfUE	https://twitte
718	750086836815486976	2016-07-04 22:00:12	TweetDeck	This is Spanky. He was a member of the 2002 USA Winter Olympic speed skating team. Accomplished af. 12/10 https://t.co/7tlZPrePXd	https://twitte
757	746818907684614144	2016-06-25 21:34:37	Twitter for iPhone	Guys... Dog Jesus 2.0\n13/10 buoyant af https://t.co/CuNA7OwfKQ	https://twitte
1314	683773439333797890	2016-01-03 22:14:26	Twitter for iPhone	This is Buddy. He's gaining strength. Currently an F4 tornado with wind speeds up to 260mph. Very devastating. 9/10 https://t.co/qipZbshNsR	https://twitte
1326	683030066213818368	2016-01-01 21:00:32	Twitter for iPhone	This is Lulu. She's contemplating all her unreached 2015 goals and daydreaming of a more efficient tomorrow. 10/10 https://t.co/h3ScYuz77J	https://twitte
1331	682662431982772225	2015-12-31 20:39:41	Twitter for iPhone	Meet Joey and Izzy. Joey only has one ear that works and Izzy wants 2015 to be over already. Both great pups. 11/10s https://t.co/WgQTlQ93BB	https://twitte

	tweet_id	timestamp	source	text	
1984	666057090499244032	2015-11-16 00:55:59	Twitter for iPhone	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	https://twitte
1988	666049248165822465	2015-11-16 00:24:50	Twitter for iPhone	Here we have a 1949 1st generation vulpix. Enjoys sweat tea and Fox News. Cannot be phased. 5/10 https://t.co/4B7cOc1EDq	https://twitte

9 rows × 23 columns

In [820]:
```
# convert rating to float
twitter_archive_master['rating'] = twitter_archive_master['rating'].asty
pe(float)
```

Define: Remove outliers by investigating all ratings that are above 14.

In [822]:
```
twitter_archive_master['rating'].sort_values(ascending=False).head(15)
```

Out[822]:
```
996     400.0
192     236.0
1434    144.0
1301    143.0
1302    121.0
614     100.0
1821    100.0
945      99.0
448      98.0
1269     97.0
1425     92.0
1493     88.0
1564     85.0
323      84.0
1182     80.0
Name: rating, dtype: float64
```

In [823]:
```
# inspect other suspect ratings
twitter_archive_master[(twitter_archive_master['rating'] == 400.0) |
                       (twitter_archive_master['rating'] == 236.0) |
                       (twitter_archive_master['rating'] == 144.0) |
                       (twitter_archive_master['rating'] == 143.0) |
                       (twitter_archive_master['rating'] == 121.0) |
                       (twitter_archive_master['rating'] == 100.0) |
                       (twitter_archive_master['rating'] == 99.0) |
                       (twitter_archive_master['rating'] == 98.0) |
                       (twitter_archive_master['rating'] == 97.0)]
```

Out[823]:

	tweet_id	timestamp	source	text	
192	844979544864018432	2017-03-23 18:29:57	Twitter for iPhone	PUPDATE: I'm proud to announce that Toby is 236 days sober. Pupgraded to a 13/10. We're all very proud of you, Toby https://t.co/a5OaJeRl9B	https://twitter.cor
448	796080075804475393	2016-11-08 20:00:55	Twitter for iPhone	This is Yogi. He's 98% floof. Snuggable af. 12/10 https://t.co/opoXKxmfFm	https://twitter.cor
614	766793450729734144	2016-08-20 00:26:19	Twitter for iPhone	This is Rufus. He just missed out on the 100m final at Rio. Already training hard for Tokyo. 10/10 never give pup https://t.co/exrRjjJqeO	https://twitter.cor
945	713900603437621249	2016-03-27 01:29:02	Twitter for iPhone	Happy Saturday here's 9 puppers on a bench. 99/90 good work everybody https://t.co/mpvaVxKmc1	https://twitter.cor
996	708469915515297792	2016-03-12 01:49:25	Twitter for iPhone	This is Bobble. He's a Croatian Galifianakis. Hears everything within 400 miles. 11/10 would snug diligently https://t.co/VwDc6PTDzk	https://twitter.cor
1269	686050296934563840	2016-01-10 05:01:51	Twitter for iPhone	This is Flávio. He's a Macedonian Poppycock. 97% floof. Jubilant af. 11/10 personally I'd pet the hell out of https://t.co/BUyX7isHRg	https://twitter.cor
1301	684225744407494656	2016-01-05 04:11:44	Twitter for iPhone	Two sneaky puppers were not initially seen, moving the rating to 143/130. Please forgive us. Thank you https://t.co/kRK51Y5ac3	https://twitter.cor

	tweet_id	timestamp	source	text	
1302	684222868335505415	2016-01-05 04:00:18	Twitter for iPhone	Someone help the girl is being mugged. Several are distracting her while two steal her shoes. Clever puppers 121/110 https://t.co/1zfnTJLt55	https://twitter.cor
1434	677716515794329600	2015-12-18 05:06:23	Twitter for iPhone	IT'S PUPPERGEDDON. Total of 144/120 ...I think https://t.co/ZanVtAtvIq	https://twitter.cor
1821	669006782128353280	2015-11-24 04:17:01	Twitter for iPhone	This is Tucker. He is 100% ready for the sports. 12/10 I would watch anything with him https://t.co/k0ddVUWTcu	https://twitter.cor

10 rows × 23 columns

Code

In [825]:
```
# fix
twitter_archive_master.loc[twitter_archive_master.tweet_id == '844979544
864018432', 'rating'] = 13 # replace 236.0
twitter_archive_master.loc[twitter_archive_master.tweet_id == '844979544
864018432', 'rating'] = 12 # replace 98%
twitter_archive_master.loc[twitter_archive_master.tweet_id == '766793450
729734144', 'rating'] = 10 # replace 100.0
twitter_archive_master.loc[twitter_archive_master.tweet_id == '713900603
437621249', 'rating'] = 11 # replace 99/90
twitter_archive_master.loc[twitter_archive_master.tweet_id == '708469915
515297792', 'rating'] = 11 # replace 400
twitter_archive_master.loc[twitter_archive_master.tweet_id == '686050296
934563840', 'rating'] = 11 # replace 97%
twitter_archive_master.loc[twitter_archive_master.tweet_id == '684225744
407494656', 'rating'] = 11 # replace 143/130
twitter_archive_master.loc[twitter_archive_master.tweet_id == '684222868
335505415', 'rating'] = 11 # replace 121/110
twitter_archive_master.loc[twitter_archive_master.tweet_id == '677716515
794329600', 'rating'] = 12 # replace 144/120
twitter_archive_master.loc[twitter_archive_master.tweet_id == '669006782
128353280', 'rating'] = 12 # replace 100%
```

In [826]:
```
twitter_archive_master['rating'].sort_values(ascending=False).head(5)
```

Out[826]:
```
448     98.0
1425    92.0
1493    88.0
1564    85.0
323     84.0
Name: rating, dtype: float64
```

In [827]:
```
# inspect additional 5 at a time
twitter_archive_master[(twitter_archive_master['rating'] == 98.0) |
                       (twitter_archive_master['rating'] == 92.0) |
                       (twitter_archive_master['rating'] == 88.0) |
                       (twitter_archive_master['rating'] == 85.0) |
                       (twitter_archive_master['rating'] == 84.0)]
```

Out[827]:

	tweet_id	timestamp	source	text	
323	820690176645140481	2017-01-15 17:52:40	Twitter for iPhone	The floofs have been released I repeat the floofs have been released. 84/70 https://t.co/NIYC820tmd	https://twitter.cor
448	796080075804475393	2016-11-08 20:00:55	Twitter for iPhone	This is Yogi. He's 98% floof. Snuggable af. 12/10 https://t.co/opoXKxmfFm	https://twitter.cor
1425	678389028614488064	2015-12-20 01:38:42	Twitter for iPhone	This is Bella. She just learned that her final grade in chem was a 92.49 \npoor pupper 11/10 https://t.co/auOoKuoveM	https://twitter.cor
1493	675853064436391936	2015-12-13 01:41:41	Twitter for iPhone	Here we have an entire platoon of puppers. Total score: 88/80 would pet all at once https://t.co/y93p6FLvVw	https://twitter.cor
1564	674269164442398721	2015-12-08 16:47:50	Twitter for iPhone	This is Bob. He's a Juniper Fitzsimmons. His body is 2, but his face is 85. Always looks miserable. Nice stool. 8/10 https://t.co/vYe9RlVz2N	https://twitter.cor

5 rows × 23 columns

In [828]:
```
# fix
twitter_archive_master.loc[twitter_archive_master.tweet_id == '820690176
645140481', 'rating'] = 12 # replace 84/70
twitter_archive_master.loc[twitter_archive_master.tweet_id == '796080075
804475393', 'rating'] = 12 # replace 98%
twitter_archive_master.loc[twitter_archive_master.tweet_id == '678389028
614488064', 'rating'] = 11 # replace 92.0
twitter_archive_master.loc[twitter_archive_master.tweet_id == '675853064
436391936', 'rating'] = 11 # replace 88/80
twitter_archive_master.loc[twitter_archive_master.tweet_id == '674269164
442398721', 'rating'] = 8 # replace 85.0
```

In [829]:
```
twitter_archive_master['rating'].sort_values(ascending=False).head(5)
```

Out[829]:
```
969     80.0
1182    80.0
1278    61.0
199     60.0
1445    60.0
Name: rating, dtype: float64
```

In [830]:
```
# inspect additional 5 at a time
twitter_archive_master[(twitter_archive_master['rating'] == 80.0) |
                       (twitter_archive_master['rating'] == 80.0) |
                       (twitter_archive_master['rating'] == 61.0) |
                       (twitter_archive_master['rating'] == 60.0) |
                       (twitter_archive_master['rating'] == 60.0)]
```

Out[830]:

	tweet_id	timestamp	source	text	
199	843235543001513987	2017-03-18 22:59:54	Twitter for iPhone	This is Tycho. She just had new wheels installed. About to do a zoom. 0-60 in 2.4 seconds. 13/10 inspirational as h*ck https://t.co/DKwp2ByMsL	https://twitter.co
969	710658690886586372	2016-03-18 02:46:49	Twitter for iPhone	Here's a brigade of puppers. All look very prepared for whatever happens next. 80/80 https://t.co/0eb7R1Om12	https://twitter.co
1053	704054845121142784	2016-02-28 21:25:30	Twitter for iPhone	Here is a whole flock of puppers. 60/50 I'll take the lot https://t.co/9dpcw6MdWa	https://twitter.co
1182	692530551048294401	2016-01-28 02:12:04	Twitter for iPhone	Say hello to Cody. He's been to like 80 countries and is way more cultured than you. He wanted me to say that. 10/10 https://t.co/Iv3flDTpXu	https://twitter.co
1278	685641971164143616	2016-01-09 01:59:19	Twitter for iPhone	This is Otis. He just passed a cop while going 61 in a 45. Very nervous pupper. 7/10 https://t.co/jJS8qQeuNO	https://twitter.co
1445	677530072887205888	2015-12-17 16:45:31	Twitter for iPhone	Say hello to Axel. He's a Black Chevy Pinot on wheels. 0 to 60 in 5.7 seconds (if downhill). 9/10 I call shotgun https://t.co/DKe9DBnnHE	https://twitter.co

6 rows × 23 columns

In [831]:
```
# fix
twitter_archive_master.loc[twitter_archive_master.tweet_id == '843235543
001513987', 'rating'] = 13
twitter_archive_master.loc[twitter_archive_master.tweet_id == '710658690
886586372', 'rating'] = 10
twitter_archive_master.loc[twitter_archive_master.tweet_id == '704054845
121142784', 'rating'] = 12
twitter_archive_master.loc[twitter_archive_master.tweet_id == '692530551
048294401', 'rating'] = 10
twitter_archive_master.loc[twitter_archive_master.tweet_id == '685641971
164143616', 'rating'] = 7
twitter_archive_master.loc[twitter_archive_master.tweet_id == '677530072
887205888', 'rating'] = 9
```

In [832]:
```
twitter_archive_master['rating'].sort_values(ascending=False).head(5)
```

Out[832]:
```
665    50.0
924    50.0
835    47.0
532    46.0
987    45.0
Name: rating, dtype: float64
```

In [833]:
```
twitter_archive_master[(twitter_archive_master['rating'] == 50.0) |
                       (twitter_archive_master['rating'] == 50.0) |
                       (twitter_archive_master['rating'] == 47.0) |
                       (twitter_archive_master['rating'] == 46.0) |
                       (twitter_archive_master['rating'] == 45.0)]
```

Out[833]:

	tweet_id	timestamp	source	text	
532	781251288990355457	2016-09-28 21:56:36	Twitter for iPhone	This is Oakley. He just got yelled at for going 46 in a 45. Churlish af. 11/10 would still pet so well https://t.co/xIYsa6LPA4	https://twitter.com
665	758041019896193024	2016-07-26 20:47:17	Twitter for iPhone	Teagan reads entire books in store so they're free. Loved 50 Shades of Grey (how dare I make that joke so late) 9/10 https://t.co/l46jwv5WYv	https://twitter.com
835	734776360183431168	2016-05-23 16:01:50	Twitter for iPhone	This is Livvie. Someone should tell her it's been 47 years since Woodstock. Magical eyes tho 11/10 would stare into https://t.co/qw07vhVHuO	https://twitter.com
924	716439118184652801	2016-04-03 01:36:11	Twitter for iPhone	This is Bluebert. He just saw that both #FinalFur match ups are split 50/50. Amazed af. 11/10 https://t.co/Kky1DPG4iq	https://twitter.com
987	709198395643068416	2016-03-14 02:04:08	Twitter for iPhone	From left to right:\nCletus, Jerome, Alejandro, Burp, & Titson\nNone know where camera is. 45/50 would hug all at once https://t.co/sedre1ivTK	https://twitter.com

5 rows × 23 columns

In [834]:
```
# fix
twitter_archive_master.loc[twitter_archive_master.tweet_id == '781251288
990355457', 'rating'] = 11
twitter_archive_master.loc[twitter_archive_master.tweet_id == '758041019
896193024', 'rating'] = 9
twitter_archive_master.loc[twitter_archive_master.tweet_id == '734776360
183431168', 'rating'] = 11
twitter_archive_master.loc[twitter_archive_master.tweet_id == '716439118
184652801', 'rating'] = 11
twitter_archive_master.loc[twitter_archive_master.tweet_id == '709198395
643068416', 'rating'] = 9
```

In [835]:
```
twitter_archive_master['rating'].sort_values(ascending=False).head(5)
```

Out[835]:
```
1129    44.0
1221    33.0
1141    31.0
385     24.0
890     20.0
Name: rating, dtype: float64
```

In [836]:
```
# Find tweet_ids
twitter_archive_master[(twitter_archive_master['rating'] == 44.0) |
                       (twitter_archive_master['rating'] == 33.0) |
                       (twitter_archive_master['rating'] == 31.0) |
                       (twitter_archive_master['rating'] == 24.0) |
                       (twitter_archive_master['rating'] == 20.0)]
```

Out[836]:

	tweet_id	timestamp	source	text	
385	810984652412424192	2016-12-19 23:06:23	Twitter for iPhone	Meet Sam. She smiles 24/7 & secretly aspires to be a reindeer. \nKeep Sam smiling by clicking and sharing this link:\nhttps://t.co/98tB8y7y7t https://t.co/LouL5vdvxx	https://www.g smile,https://t
890	722974582966214656	2016-04-21 02:25:47	Twitter for iPhone	Happy 4/20 from the squad! 13/10 for all https://t.co/eV1diwds8a	https://twitter
1129	697463031882764288	2016-02-10 16:51:59	Twitter for iPhone	Happy Wednesday here's a bucket of pups. 44/40 would pet all at once https://t.co/HppvrYuamZ	https://twitter
1141	696405997980676096	2016-02-07 18:51:43	Twitter for iPhone	This is Berb. He just found out that they have made 31 Kidz Bop CD's. Downright terrifying. 7/10 hang in there Berb https://t.co/CIFLjiTFwZ	https://twitter
1221	689599056876867584	2016-01-20 00:03:21	Twitter for iPhone	Here we see 33 dogs posing for a picture. All get 11/10 for superb cooperation https://t.co/TRAri5iHzd	https://twitter

5 rows × 23 columns

In [837]:
```
# drop row 385 (tweet_id = 810984652412424192) has no rating but says 24
twitter_archive_master = twitter_archive_master[twitter_archive_master[
'tweet_id'] != '810984652412424192']
twitter_archive_master[(twitter_archive_master['rating'] == 24.0)]
```

Out[837]:

	tweet_id	timestamp	source	text	expanded_urls	rating	name	dog_stage	retweet_co

0 rows × 23 columns

In [838]:
```
# fix
twitter_archive_master.loc[twitter_archive_master.tweet_id == '722974582
966214656', 'rating'] = 13
twitter_archive_master.loc[twitter_archive_master.tweet_id == '697463031
882764288', 'rating'] = 11
twitter_archive_master.loc[twitter_archive_master.tweet_id == '696405997
980676096', 'rating'] = 7
twitter_archive_master.loc[twitter_archive_master.tweet_id == '689599056
876867584', 'rating'] = 11
```

In [839]:
```
twitter_archive_master['rating'].sort_values(ascending=False).head(5)
```

Out[839]:
```
687    17.0
313    14.0
297    14.0
49     14.0
100    14.0
Name: rating, dtype: float64
```

In [840]:
```
# Find tweet_ids
twitter_archive_master[(twitter_archive_master['rating'] == 17.0) |
                       (twitter_archive_master['rating'] == 14.0)]
```

Out[840]:

	tweet_id	timestamp	source	text	
9	890240255349198849	2017-07-26 15:59:51	Twitter for iPhone	This is Cassie. She is a college pup. Studying international doggo communication and stick theory. 14/10 so elegant much sophisticate https://t.co/t1bfwz5S2A	https://tw
36	884441805382717440	2017-07-10 15:58:53	Twitter for iPhone	I present to you, Pup in Hat. Pup in Hat is great for all occasions. Extremely versatile. Compact as h*ck. 14/10 (IG: itselizabethgales) https://t.co/vvBOcC2VdC	https://tw
49	881536004380872706	2017-07-02 15:32:16	Twitter for iPhone	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	https://tw
64	878057613040115712	2017-06-23 01:10:23	Twitter for iPhone	This is Emmy. She was adopted today. Massive round of pupplause for Emmy and her new family. 14/10 for all involved https://t.co/cwtWnHMVpe	https://tw
100	868880397819494401	2017-05-28 17:23:24	Twitter for iPhone	This is Walter. He won't start hydrotherapy without his favorite floatie. 14/10 keep it pup Walter https://t.co/r28jFx9uyF	https://tw
119	863079547188785154	2017-05-12 17:12:53	Twitter for iPhone	Ladies and gentlemen... I found Pipsy. He may have changed his name to Pablo, but he never changed his love for the sea. Pupgraded to 14/10 https://t.co/lVU5GyNFen	https://tw
146	856526610513747968	2017-04-24 15:13:52	Twitter for iPhone	THIS IS CHARLIE, MARK. HE DID JUST WANT TO SAY HI AFTER ALL. PUPGRADED TO A 14/10. WOULD BE AN HONOR TO FLY WITH https://t.co/p1hBHCmWnA	https://tw
147	856282028240666624	2017-04-23 23:01:59	Twitter for iPhone	This is Cermet, Paesh, and Morple. They are absolute h*ckin superstars. Watered every day so they can grow. 14/10 for all https://t.co/GUefqUmZv8	https://tw

	tweet_id	timestamp	source	text	
153	854120357044912130	2017-04-17 23:52:16	Twitter for iPhone	Sometimes you guys remind me just how impactful a pupper can be. Cooper will be remembered as a good boy by so many. 14/10 rest easy friend https://t.co/oBL7LEJEzR	https://tw
206	841439858740625411	2017-03-14 00:04:30	Twitter for iPhone	Here we have some incredible doggos for #K9VeteransDay. All brave as h*ck. Salute your dog in solidarity. 14/10 for all https://t.co/SVNMdFqKDL	https://tw
253	832273440279240704	2017-02-16 17:00:25	Twitter for iPhone	Say hello to Smiley. He's a blind therapy doggo having a h*ckin blast high steppin around in the snow. 14/10 would follow anywhere https://t.co/SHAb1wHjMz	https://tw
275	828650029636317184	2017-02-06 17:02:17	Twitter for iPhone	Occasionally, we're sent fantastic stories. This is one of them. 14/10 for Grace https://t.co/bZ4axuH6OK	https://tw
278	828381636999917570	2017-02-05 23:15:47	Twitter for iPhone	Meet Doobert. He's a deaf doggo. Didn't stop him on the field tho. Absolute legend today. 14/10 would pat head approvingly https://t.co/iCk7zstRA9	https://tw
297	825535076884762624	2017-01-29 02:44:34	Twitter for iPhone	Here's a very loving and accepting puppo. Appears to have read her Constitution well. 14/10 would pat head approvingly https://t.co/6ao80wIpV1	https://tw
313	822462944365645825	2017-01-20 15:17:01	Twitter for iPhone	This is Gabe. He was the unequivocal embodiment of a dream meme, but also one h*ck of a pupper. You will be missed by so many. 14/10 RIP https://t.co/M3hZGadUuO	https://tw
318	821407182352777218	2017-01-17 17:21:47	Twitter for iPhone	This is Sundance. He's a doggo drummer. Even sings a bit on the side. 14/10 entertained af (vid by @sweetsundance) https://t.co/Xn5AQtiqzG	https://tw

	tweet_id	timestamp	source	text	
324	820314633777061888	2017-01-14 17:00:24	Twitter for iPhone	We are proud to support @LoveYourMelon on their mission to put a hat on every kid battling cancer. They are 14/10\n\nhttps://t.co/XQlmPTLHPl https://t.co/ZNIkkHgtYE	https://w
333	819004803107983360	2017-01-11 02:15:36	Twitter for iPhone	This is Bo. He was a very good First Doggo. 14/10 would be an absolute honor to pet https://t.co/AdPKrI8BZ1	https://tw
362	813812741911748608	2016-12-27 18:24:12	Twitter for iPhone	Meet Gary, Carrie Fisher's dog. Idk what I can say about Gary that reflects the inspirational awesomeness that was Carrie Fisher. 14/10 RIP https://t.co/uBnQTNEeGg	https://tw
399	807621403335917568	2016-12-10 16:22:02	Twitter for iPhone	This is Ollie Vue. He was a 3 legged pupper on a mission to overcome everything. This is very hard to write. 14/10 we will miss you Ollie https://t.co/qTRY2qX9y4	https://tw
549	778408200802557953	2016-09-21 01:39:11	Twitter for iPhone	RIP Loki. Thank you for the good times. You will be missed by many. 14/10 https://t.co/gJKD9pst5A	https://tw
571	774314403806253056	2016-09-09 18:31:54	Twitter for iPhone	I WAS SENT THE ACTUAL DOG IN THE PROFILE PIC BY HIS OWNER THIS IS SO WILD. 14/10 ULTIMATE LEGEND STATUS https://t.co/7oQ1wpfxIH	https://tw
687	754120377874386944	2016-07-16 01:08:03	Twitter for iPhone	When you hear your owner say they need to hatch another egg, but you've already been on 17 walks today. 10/10 https://t.co/lFEoGqZ4oA	https://tw

23 rows × 23 columns

In [841]:
```
# fix (all of the 14's are accurate)
twitter_archive_master.loc[twitter_archive_master.tweet_id == '754120377
874386944', 'rating'] = 10
```

Test

In [843]:
```
# Max value should be 14.0
twitter_archive_master['rating'].value_counts()
```

Out[843]:
```
10.0    630
12.0    426
11.0    367
13.0    230
14.0    22
9.0     5
7.0     2
5.0     1
8.0     1
Name: rating, dtype: int64
```

In [845]:
```
# convert back to integer for simplicity as there are no longer decimals

twitter_archive_master['rating'].astype(int)
twitter_archive_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1684 entries, 1 to 1992
Data columns (total 23 columns):
tweet_id          1684 non-null object
timestamp         1684 non-null datetime64[ns]
source            1684 non-null object
text              1684 non-null object
expanded_urls     1684 non-null object
rating            1684 non-null float64
name              1684 non-null object
dog_stage         1684 non-null object
retweet_count     1684 non-null int64
favorite_count    1684 non-null int64
year              1684 non-null int64
month             1684 non-null int64
day               1684 non-null int64
time              1684 non-null object
weekday           1684 non-null object
jpg_url           1684 non-null object
prediction_1      1684 non-null object
p1_conf           1684 non-null int64
prediction_2      1684 non-null object
p2_conf           1684 non-null int64
prediction_3      1684 non-null object
p3_conf           1684 non-null int64
prediction        1684 non-null object
dtypes: datetime64[ns](1), float64(1), int64(8), object(13)
memory usage: 315.8+ KB
```

In [846]:
```
twitter_archive_master['rating'].describe()
```

Out[846]:
```
count    1684.000000
mean       11.175178
std         1.148334
min         5.000000
25%        10.000000
50%        11.000000
75%        12.000000
max        14.000000
Name: rating, dtype: float64
```

Define: Create a copy of the master dataset for known dogs only (all 3 predictions True). Include only key columns and reorder for legibility.

Code

In [851]:
```
twitter_archive_dogs = twitter_archive_master[twitter_archive_master['pr
ediction'] == "Dog"].copy()
twitter_archive_dogs.prediction.value_counts()
twitter_archive_dogs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1201 entries, 1 to 1992
Data columns (total 23 columns):
tweet_id          1201 non-null object
timestamp         1201 non-null datetime64[ns]
source            1201 non-null object
text              1201 non-null object
expanded_urls     1201 non-null object
rating            1201 non-null float64
name              1201 non-null object
dog_stage         1201 non-null object
retweet_count     1201 non-null int64
favorite_count    1201 non-null int64
year              1201 non-null int64
month             1201 non-null int64
day               1201 non-null int64
time              1201 non-null object
weekday           1201 non-null object
jpg_url           1201 non-null object
prediction_1      1201 non-null object
p1_conf           1201 non-null int64
prediction_2      1201 non-null object
p2_conf           1201 non-null int64
prediction_3      1201 non-null object
p3_conf           1201 non-null int64
prediction        1201 non-null object
dtypes: datetime64[ns](1), float64(1), int64(8), object(13)
memory usage: 225.2+ KB
```

In [852]:
```
twitter_archive_dogs.drop(['year', 'month', 'day', 'time', 'prediction',
 'source',
                           'p1_conf', 'prediction_2', 'p2_conf', 'predict
ion_3', 'p3_conf',
                           'expanded_urls'], axis=1, inplace=True)
twitter_archive_dogs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1201 entries, 1 to 1992
Data columns (total 11 columns):
tweet_id          1201 non-null object
timestamp         1201 non-null datetime64[ns]
text              1201 non-null object
rating            1201 non-null float64
name              1201 non-null object
dog_stage         1201 non-null object
retweet_count     1201 non-null int64
favorite_count    1201 non-null int64
weekday           1201 non-null object
jpg_url           1201 non-null object
prediction_1      1201 non-null object
dtypes: datetime64[ns](1), float64(1), int64(2), object(7)
memory usage: 112.6+ KB
```

In [854]:
```
twitter_archive_dogs.reindex(['tweet_id', 'prediction_1', 'rating',
                              'favorite_count', 'retweet_count', 'dog_st
age',
                              'name', 'text', 'timestamp', 'jpg_url'], a
xis=1).sample(3)
```

Out[854]:

	tweet_id	prediction_1	rating	favorite_count	retweet_count	dog_stag
1243	688064179421470721	Eskimo Dog	11.0	1828	389	Unknown
1249	687494652870668288	Rottweiler	10.0	2035	625	Unknown
1425	678389028614488064	Miniature Pinscher	11.0	1972	454	Pupper

In [855]:
```
twitter_archive_master.to_csv('twitter_archive_master.csv', index=False)
twitter_archive_dogs.to_csv('twitter_archive_dogs.csv', index=False)
```

Analysis & Visualizations

In sifting through this dataset, I'm interested in which breeds are most popoular, which have the highest ratings, which are most favorited, which are most retweeted? What's the most common dog stage? What are the most popular dog names? Have tweets increased or decreased over the years? Which dogs are outliers, rating so much higher than the others? What is the most popular platform for originating tweets?

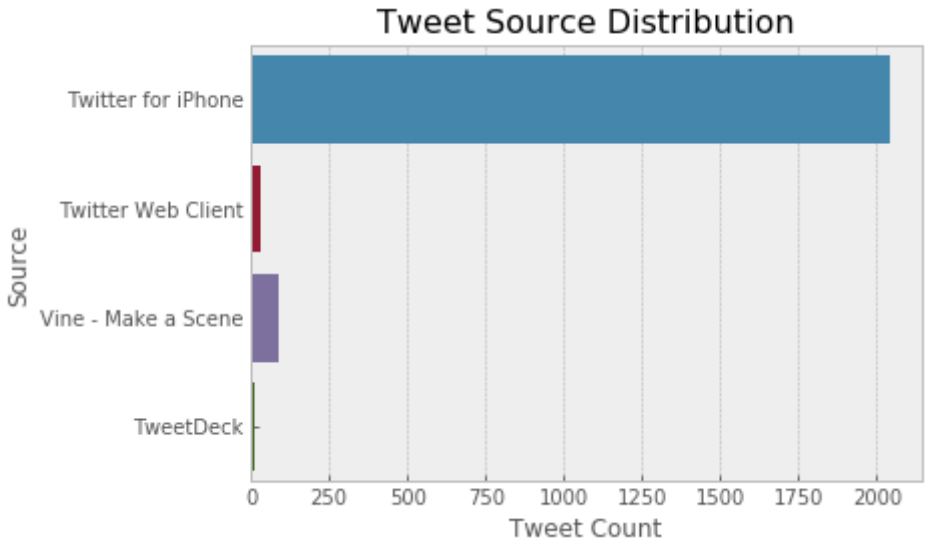
In [856]:
```
plt.style.use('ggplot')
print(plt.style.available)
```

```
['seaborn-dark', 'seaborn-darkgrid', 'seaborn-ticks', 'fivethirtyeigh
t', 'seaborn-whitegrid', 'classic', '_classic_test', 'fast', 'seaborn-t
alk', 'seaborn-dark-palette', 'seaborn-bright', 'seaborn-pastel', 'gray
scale', 'seaborn-notebook', 'ggplot', 'seaborn-colorblind', 'seaborn-mu
ted', 'seaborn', 'Solarize_Light2', 'seaborn-paper', 'bmh', 'tableau-co
lorblind10', 'seaborn-white', 'dark_background', 'seaborn-poster', 'sea
born-deep']
```

In [857]:
```
# There are 5 preset seaborn themes: darkgrid, whitegrid, dark, white, a
nd ticks.
#sns.set_style("whitegrid")

plt.style.use('bmh')

sns.countplot(data = archive_clean, y = 'source')
plt.title('Tweet Source Distribution', fontsize=16)
plt.xlabel('Tweet Count', fontsize=12)
plt.ylabel('Source', fontsize=12)
plt.savefig('tweet-source.png');
```



The Twitter app is the most widely used platform; 94% of twitter users use the mobile iPhone app to originate tweets. The other 6% use the Twitter web client (mobile and desktop), Vine, and TweetDeck.

In [858]:
```
archive_clean.source.value_counts()
```

Out[858]:
```
Twitter for iPhone     2042
Vine - Make a Scene    91
Twitter Web Client     31
TweetDeck              11
Name: source, dtype: int64
```

In [859]:
```
twitter_archive_dogs['rating'].value_counts()
```

Out[859]:
```
10.0    433
12.0    309
11.0    253
13.0    183
14.0    17
9.0     3
5.0     1
8.0     1
7.0     1
Name: rating, dtype: int64
```

In [860]:
```
# There are 17 top rated dogs
# twitter_archive_dogs.sort_values(by=['rating'], ascending=False).head
(17)
top_rated = twitter_archive_dogs.query('rating == 14')
top_rated.sort_values(by=['retweet_count'], ascending=False)
#top_rated.sort_values(by=['favorite_count'], ascending=False)
```

Out[860]:

	tweet_id	timestamp	text	rating	name
333	819004803107983360	2017-01-11 02:15:36	This is Bo. He was a very good First Doggo. 14/10 would be an absolute honor to pet https://t.co/AdPKrI8BZ1	14.0	Bo
297	825535076884762624	2017-01-29 02:44:34	Here's a very loving and accepting puppo. Appears to have read her Constitution well. 14/10 would pat head approvingly https://t.co/6ao80wIpV1	14.0	None
313	822462944365645825	2017-01-20 15:17:01	This is Gabe. He was the unequivocal embodiment of a dream meme, but also one h*ck of a pupper. You will be missed by so many. 14/10 RIP https://t.co/M3hZGadUuO	14.0	Gabe
362	813812741911748608	2016-12-27 18:24:12	Meet Gary, Carrie Fisher's dog. Idk what I can say about Gary that reflects the inspirational awesomeness that was Carrie Fisher. 14/10 RIP https://t.co/uBnQTNEeGg	14.0	Gary
153	854120357044912130	2017-04-17 23:52:16	Sometimes you guys remind me just how impactful a pupper can be. Cooper will be remembered as a good boy by so many. 14/10 rest easy friend https://t.co/oBL7LEJEzR	14.0	None
9	890240255349198849	2017-07-26 15:59:51	This is Cassie. She is a college pup. Studying international doggo communication and stick theory. 14/10 so elegant much sophisticate https://t.co/t1bfwz5S2A	14.0	Cassie
64	878057613040115712	2017-06-23 01:10:23	This is Emmy. She was adopted today. Massive round of pupplause for Emmy and her new family. 14/10 for all involved https://t.co/cwtWnHMVpe	14.0	Emmy
147	856282028240666624	2017-04-23 23:01:59	This is Cermet, Paesh, and Morple. They are absolute h*ckin superstars. Watered every day so they can grow. 14/10 for all https://t.co/GUefqUmZv8	14.0	Cermet

	tweet_id	timestamp	text	rating	name
571	774314403806253056	2016-09-09 18:31:54	I WAS SENT THE ACTUAL DOG IN THE PROFILE PIC BY HIS OWNER THIS IS SO WILD. 14/10 ULTIMATE LEGEND STATUS https://t.co/7oQ1wpfxIH	14.0	None
36	884441805382717440	2017-07-10 15:58:53	I present to you, Pup in Hat. Pup in Hat is great for all occasions. Extremely versatile. Compact as h*ck. 14/10 (IG: itselizabethgales) https://t.co/vvBOcC2VdC	14.0	None
549	778408200802557953	2016-09-21 01:39:11	RIP Loki. Thank you for the good times. You will be missed by many. 14/10 https://t.co/gJKD9pst5A	14.0	None
318	821407182352777218	2017-01-17 17:21:47	This is Sundance. He's a doggo drummer. Even sings a bit on the side. 14/10 entertained af (vid by @sweetsundance) https://t.co/Xn5AQtiqzG	14.0	Sundance
399	807621403335917568	2016-12-10 16:22:02	This is Ollie Vue. He was a 3 legged pupper on a mission to overcome everything. This is very hard to write. 14/10 we will miss you Ollie https://t.co/qTRY2qX9y4	14.0	Ollie
278	828381636999917570	2017-02-05 23:15:47	Meet Doobert. He's a deaf doggo. Didn't stop him on the field tho. Absolute legend today. 14/10 would pat head approvingly https://t.co/iCk7zstRA9	14.0	Doobert
146	856526610513747968	2017-04-24 15:13:52	THIS IS CHARLIE, MARK. HE DID JUST WANT TO SAY HI AFTER ALL. PUPGRADED TO A 14/10. WOULD BE AN HONOR TO FLY WITH https://t.co/p1hBHCmWnA	14.0	None
275	828650029636317184	2017-02-06 17:02:17	Occasionally, we're sent fantastic stories. This is one of them. 14/10 for Grace https://t.co/bZ4axuH6OK	14.0	None

	tweet_id	timestamp	text	rating	name
324	820314633777061888	2017-01-14 17:00:24	We are proud to support @LoveYourMelon on their mission to put a hat on every kid battling cancer. They are 14/10\n\nhttps://t.co/XQlmPTLHPl https://t.co/ZNIkkHgtYE	14.0	None

The Winner

Of the 17 top rated dogs, **Bo**, a Standard Poodle is clearly the overall winner with a combined rating of 14, retweet count of 40,641 and favorite count of 92,985. The retweet count and favorite counts are both the highest in this group of ratings.

In [861]:
```
# Top favorited dogs
twitter_archive_dogs.sort_values(by=['favorite_count'], ascending=False)
.head()
```

Out[861]:

	tweet_id	timestamp	text	rating	name	dog_stag
309	822872901745569793	2017-01-21 18:26:02	Here's a super supportive puppo participating in the Toronto #WomensMarch today. 13/10 https://t.co/nTz3FtorBc	13.0	None	Puppo
108	866450705531457537	2017-05-22 00:28:40	This is Jamesy. He gives a kiss to every other pupper he sees on his walk. 13/10 such passion, much tender https://t.co/wk7TfysWHr	13.0	Jamesy	Pupper
400	807106840509214720	2016-12-09 06:17:20	This is Stephan. He just wants to help. 13/10 such a good boy https://t.co/DkBYaCAg2d	13.0	Stephan	Unknown
809	739238157791694849	2016-06-04 23:31:25	Here's a doggo blowing bubbles. It's downright legendary. 13/10 would watch on repeat forever (vid by Kent Duryee) https://t.co/YcXgHfp1EC	13.0	None	Doggo
58	879415818425184262	2017-06-26 19:07:24	This is Duddles. He did an attempt. 13/10 someone help him (vid by Georgia Felici) https://t.co/UDT7ZkcTgY	13.0	Duddles	Unknown

The top favorited dog has is a super supportive Lakeland Terrior who marches for women.

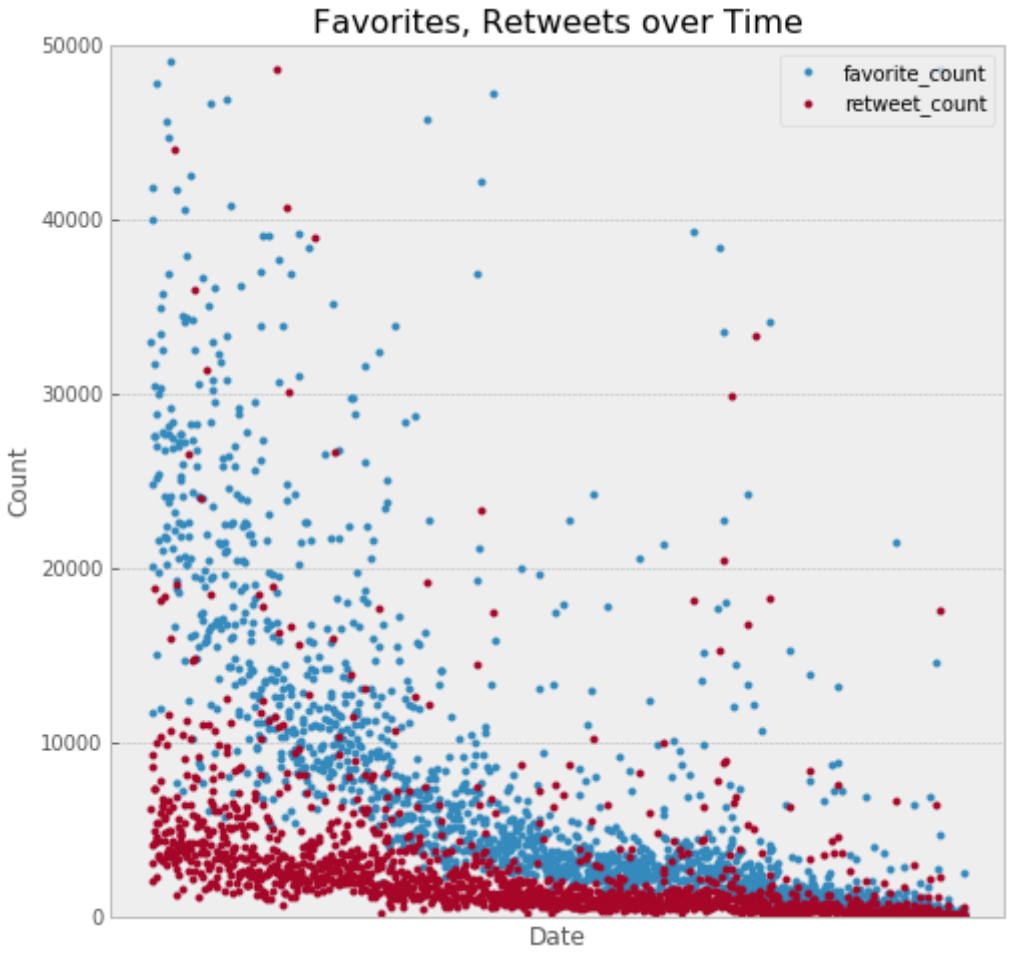
In [862]:
```
# Top retweeted dogs
twitter_archive_dogs.sort_values(by=['retweet_count'], ascending=False).
head(5)
```

Out[862]:

	tweet_id	timestamp	text	rating	name	dog_stag
809	739238157791694849	2016-06-04 23:31:25	Here's a doggo blowing bubbles. It's downright legendary. 13/10 would watch on repeat forever (vid by Kent Duryee) https://t.co/YcXgHfp1EC	13.0	None	Doggo
400	807106840509214720	2016-12-09 06:17:20	This is Stephan. He just wants to help. 13/10 such a good boy https://t.co/DkBYaCAg2d	13.0	Stephan	Unknown
309	822872901745569793	2017-01-21 18:26:02	Here's a super supportive puppo participating in the Toronto #WomensMarch today. 13/10 https://t.co/nTz3FtorBc	13.0	None	Puppo
58	879415818425184262	2017-06-26 19:07:24	This is Duddles. He did an attempt. 13/10 someone help him (vid by Georgia Felici) https://t.co/UDT7ZkcTgY	13.0	Duddles	Unknown
333	819004803107983360	2017-01-11 02:15:36	This is Bo. He was a very good First Doggo. 14/10 would be an absolute honor to pet https://t.co/AdPKrI8BZ1	14.0	Bo	Doggo

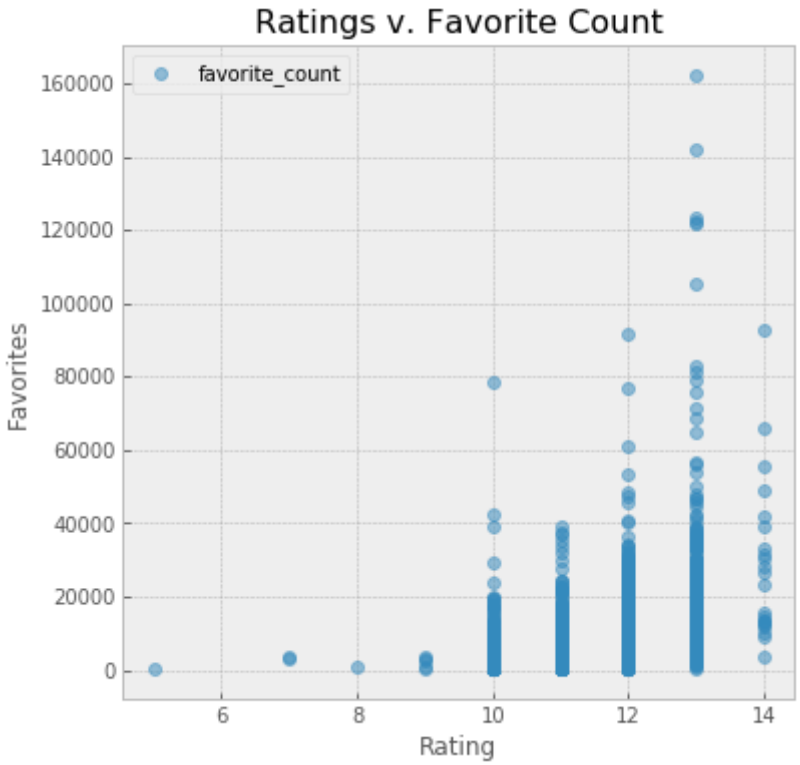
The most retweeted dog is a Siberian Husky with bubble blowing skills.

In [863]:
```
twitter_archive_master[['favorite_count', 'retweet_count']].plot(style =
 '.', ylim=[0, 50000], figsize=(8,8))
plt.title('Favorites, Retweets over Time', size=16)
plt.xlabel('Date', size=12)
plt.xticks([], [])
plt.ylabel('Count', size=12)
plt.legend(ncol=1, loc='upper right')
plt.savefig('retweets-favorites-time.png');
```



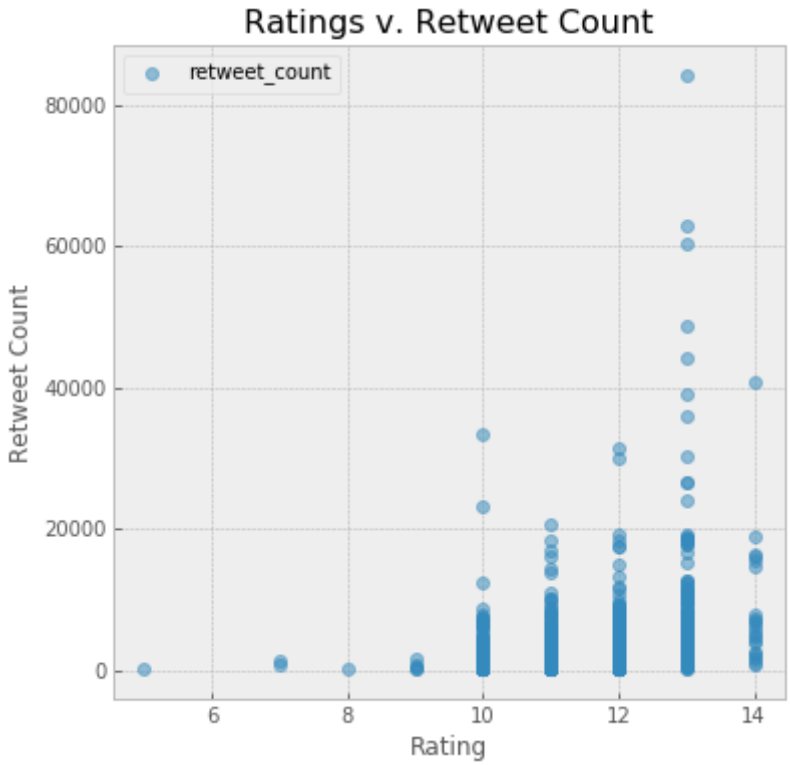
Favorites are more popular than retweets. Both decreasing over time, and retweets even more so.

In [864]:
```
twitter_archive_master.plot(x = 'rating', y = 'favorite_count', style =
'o', figsize=(6,6), alpha=.5)
plt.title('Ratings v. Favorite Count', size=16)
plt.xlabel('Rating', size=12)
plt.ylabel('Favorites', size=12);
```



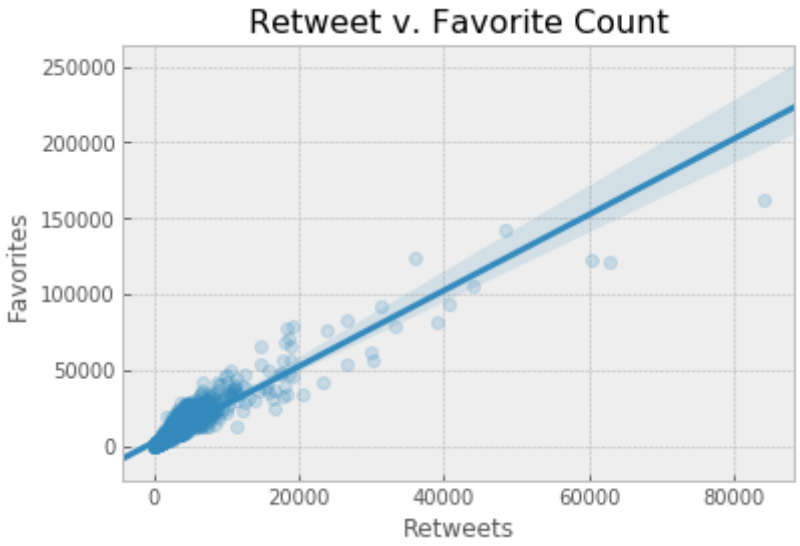
Higher rated dogs get more favorites.

In [865]:
```
twitter_archive_master.plot(x = 'rating', y = 'retweet_count', style =
'o', alpha=.5, figsize=(6,6));
plt.title('Ratings v. Retweet Count', size=16)
plt.xlabel('Rating', size=12)
plt.ylabel('Retweet Count', size=12);
```



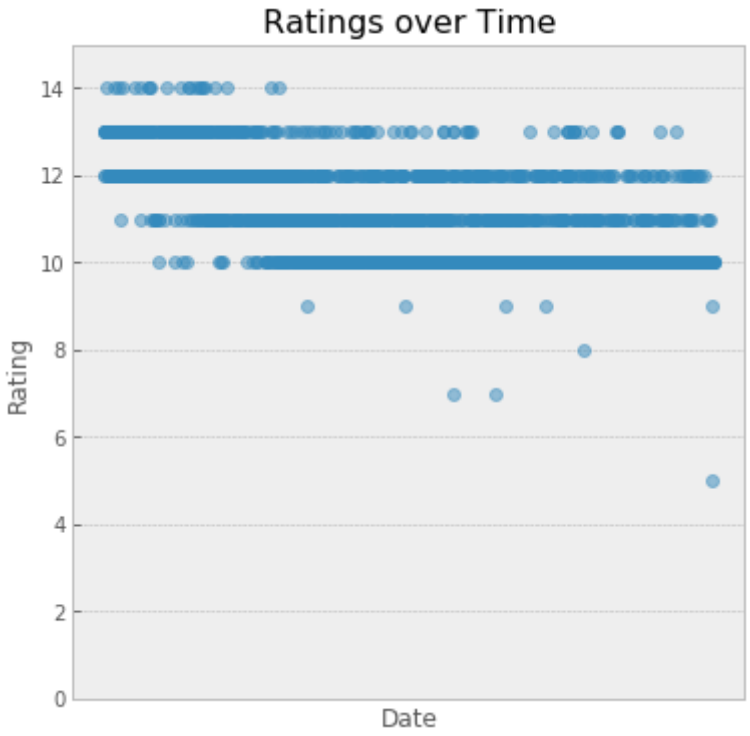
Higher rated dogs have more retweets.

In [866]:
```
sns.regplot(x="retweet_count", y="favorite_count", data=twitter_archive_
master, scatter_kws={'alpha':0.2})
plt.title('Retweet v. Favorite Count', size=16)
plt.xlabel('Retweets', size=12)
plt.ylabel('Favorites', size=12)
plt.savefig('retweet-favorite.png');
```



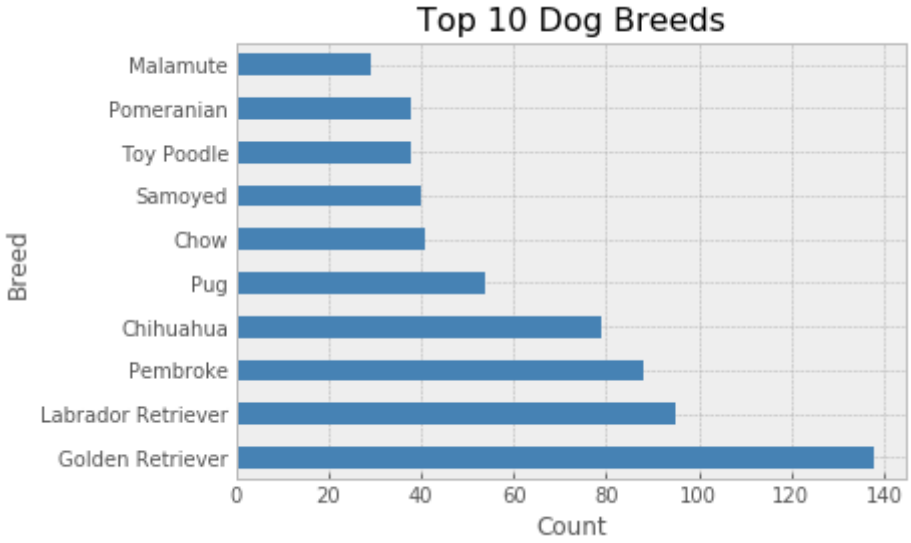
There is a strong positive correlation between number of retweets and favorite count. It does seem reasonable that the more a post is retweeted, the more eyes view the post, the more favorites the post receives.

In [867]:
```
twitter_archive_master['rating'].plot(style = 'o', alpha=.5, figsize=(6,
6), ylim=[0, 15])
plt.title('Ratings over Time', size=16)
plt.xlabel('Date')
plt.xticks([], [])
plt.ylabel('Rating');
```



Ratings have decreased over time.

In [868]:
```
top_breeds = twitter_archive_master.prediction_1.value_counts()[0:10].so
rt_values(axis=0, ascending=False)
top_breeds.plot(kind = 'barh', color=['steelblue'])
plt.title('Top 10 Dog Breeds', size=16)
plt.xlabel('Count', size=12)
plt.ylabel('Breed', size=12)
plt.savefig('top-breeds.png');
```



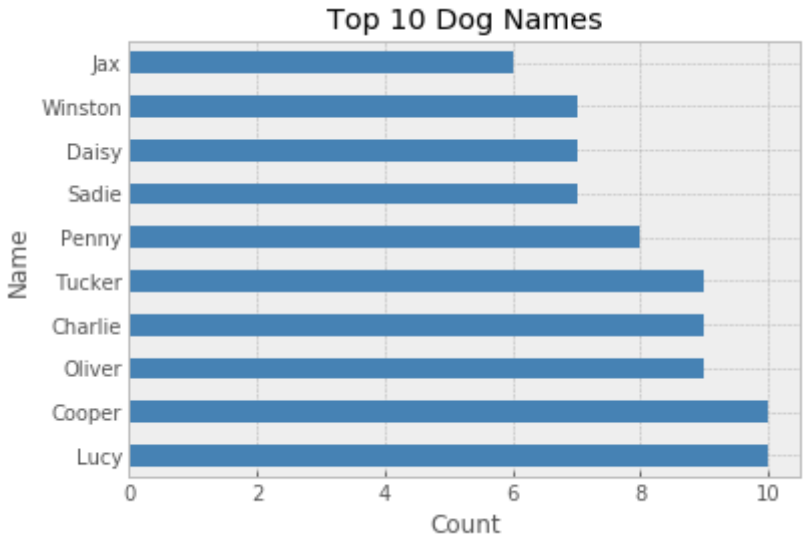
In [887]:
```
twitter_archive_master.prediction_1.value_counts()[0:10].sort_values(axi
s=0, ascending=False)
```

Out[887]:
```
Golden Retriever      138
Labrador Retriever     95
Pembroke               88
Chihuahua              79
Pug                    54
Chow                   41
Samoyed                40
Toy Poodle             38
Pomeranian             38
Malamute               29
Name: prediction_1, dtype: int64
```

There are more golden Retrievers than any other dog in the dataset. Labrador Retrievers are the second most common.

In [869]:
```
top_names = twitter_archive_master.name.value_counts()[1:11].sort_values
(axis=0, ascending=False)
top_names.plot(kind = 'barh', color='steelblue')

plt.title('Top 10 Dog Names')
plt.xlabel('Count')
plt.ylabel('Name')
plt.savefig('top-names.png');
```



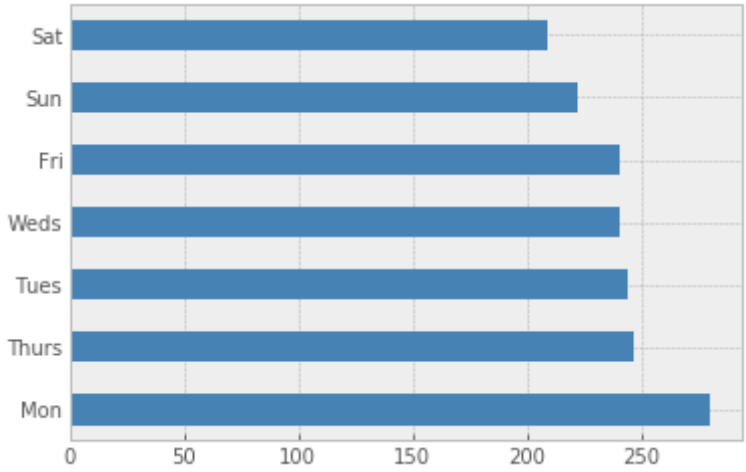
Lucy and Cooper are the most popular dog names. Tucker, Oliver, and Charlie follow close behind.

In [870]:
```
top_rated_breeds = twitter_archive_dogs.sort_values(by=['rating'], ascen
ding=False).head(10)
top_rated_breeds.groupby('prediction_1')['rating'].describe()
```

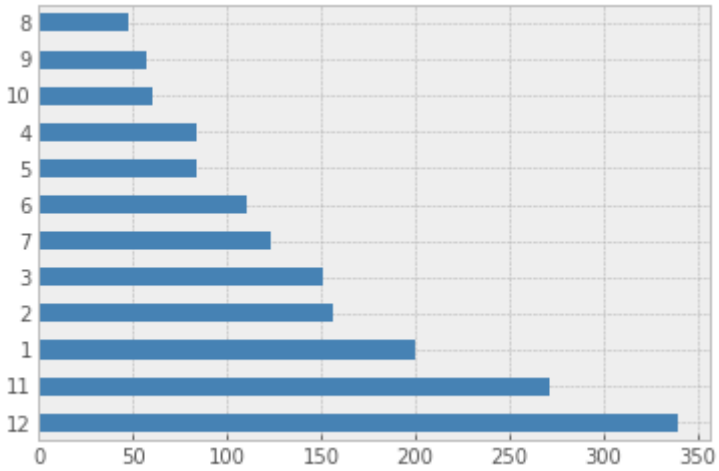
Out[870]:

	count	mean	std	min	25%	50%	75%	max
prediction_1								
Bedlington Terrier	1.0	14.0	NaN	14.0	14.0	14.0	14.0	14.0
Black-And-Tan Coonhound	1.0	14.0	NaN	14.0	14.0	14.0	14.0	14.0
Chihuahua	1.0	14.0	NaN	14.0	14.0	14.0	14.0	14.0
Eskimo Dog	1.0	14.0	NaN	14.0	14.0	14.0	14.0	14.0
French Bulldog	2.0	14.0	0.0	14.0	14.0	14.0	14.0	14.0
Golden Retriever	1.0	14.0	NaN	14.0	14.0	14.0	14.0	14.0
Old English Sheepdog	1.0	14.0	NaN	14.0	14.0	14.0	14.0	14.0
Pembroke	1.0	14.0	NaN	14.0	14.0	14.0	14.0	14.0
Rottweiler	1.0	14.0	NaN	14.0	14.0	14.0	14.0	14.0

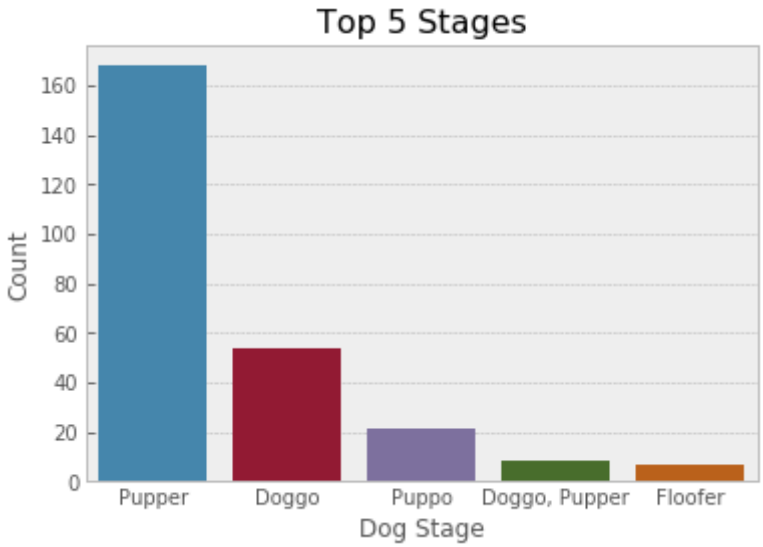
In [875]:
```
tweets = twitter_archive_master['weekday'].value_counts()
tweets.plot(kind = 'barh', color='steelblue')
plt.savefig('weekdays.png');
```



In [876]:
```
tweets_month = tweets = twitter_archive_master['month'].value_counts()
tweets_month.plot(kind = 'barh', color='steelblue')
plt.savefig('month-tweets.png');
```



In [877]:
```
sorted_stage = twitter_archive_master['dog_stage'].value_counts()[1:6].i
ndex
sns.countplot(data = twitter_archive_master, x = 'dog_stage', order = so
rted_stage, orient = 'h')
plt.xlabel('Dog Stage', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.title('Top 5 Stages',fontsize=16)
plt.savefig('top-stages.png');
```

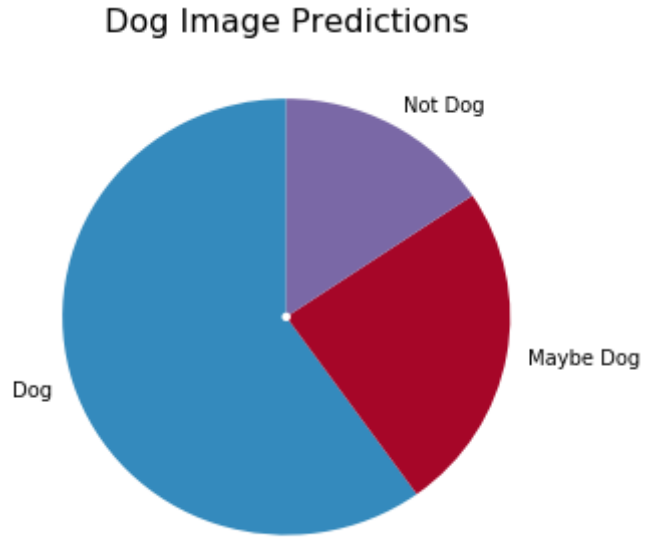


In [886]:
```
twitter_archive_master['dog_stage'].value_counts()
```

Out[886]:
```
Unknown          1424
Pupper            168
Doggo              54
Puppo              21
Doggo, Pupper       8
Floofer             7
Doggo, Puppo        1
Doggo, Floofer      1
Name: dog_stage, dtype: int64
```

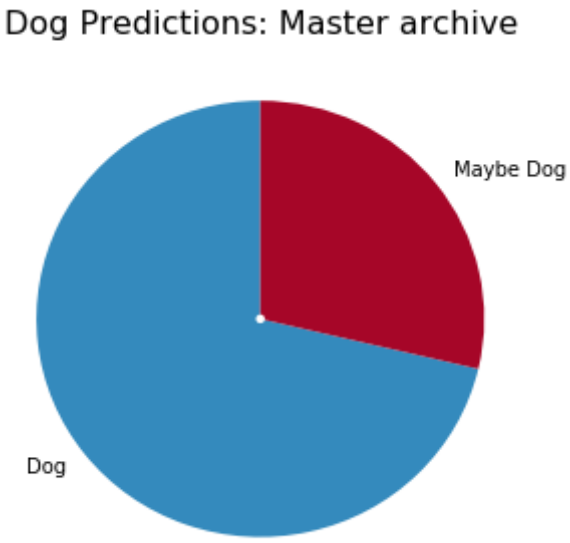
Most dogs are classified in the 'Pupper' stage: "A pupper is a small doggo, usually younger. Can be equally, if not more mature, than most doggos. A doggo that is inexperienced, unfamiliar, or in any way unprepared for the responsibilities associated with being a doggo."

In [878]:
```
image_clean['prediction'].value_counts().plot(kind='pie', figsize=(5,5),
 startangle = 90, wedgeprops = {'width': 0.98})
plt.title('Dog Image Predictions',fontsize=16);
plt.ylabel('');
```



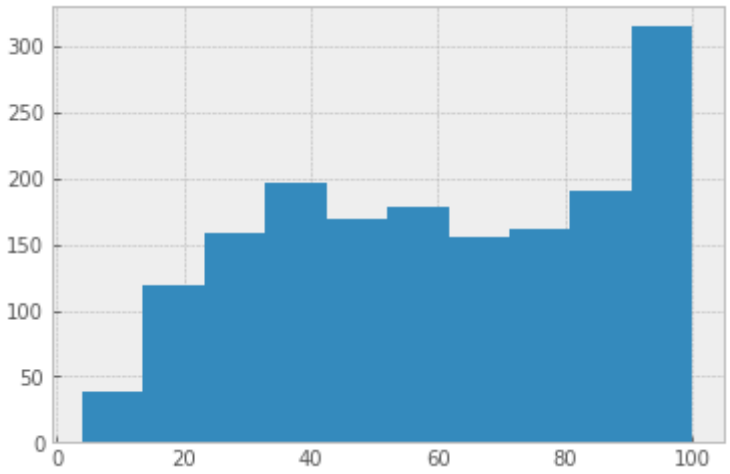
Just over half of the 'dogs' in our image datatbase might actually be dogs.

In [879]:
```
twitter_archive_master['prediction'].value_counts().plot(kind='pie', fig
size=(5,5), startangle = 90, wedgeprops = {'width': 0.98})
plt.title('Dog Predictions: Master archive',fontsize=16);
plt.ylabel('');
```

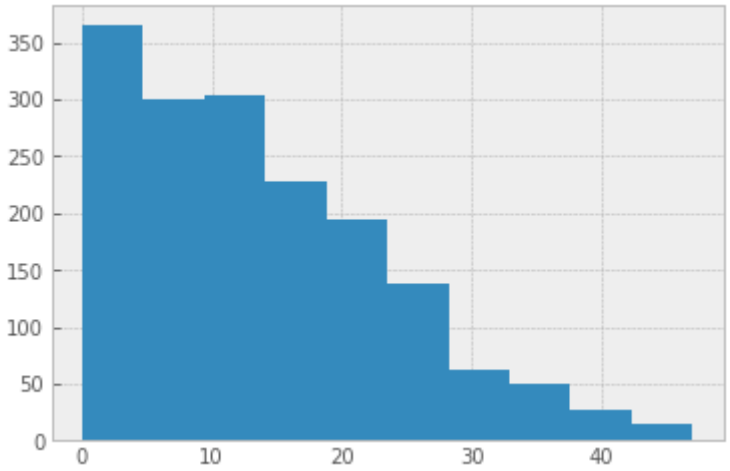


Nearly three quarters of the 'dogs' in our master archive datatbase are actually dogs.

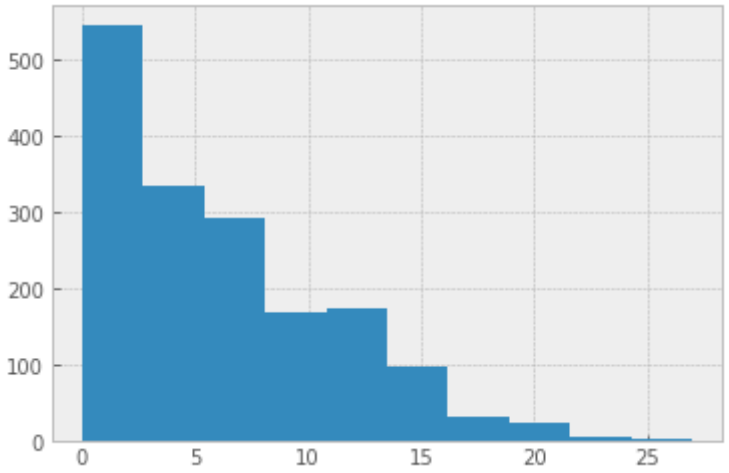
In [885]:
```
twitter_archive_master['p1_conf'].hist();
```



In [884]:
```
# ax = sns.distplot(twitter_archive_master['p2_conf'])
twitter_archive_master['p2_conf'].hist();
```



In [883]:
```
twitter_archive_master['p3_conf'].hist();
```



References:

Tidy Data Rules: https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html (https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html)
WeRateDogs Twitter: https://twitter.com/dog_rates?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor (https://twitter.com/dog_rates?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor)
The Dogtionary: https://www.amazon.com/WeRateDogs-Most-Hilarious-Adorable-Youve/dp/1510717145 (https://www.amazon.com/WeRateDogs-Most-Hilarious-Adorable-Youve/dp/1510717145)
Tweepy Library: http://www.tweepy.org/ (http://www.tweepy.org/)