Data Wrangling Project - Udacity Nanodegree

By Ricardo Rosas

As part of my Data Analyst Nano Degree (https://www.udacity.com/course/data-analyst-nanodegree--nd002) I am going to go through the different stages of the data wrangling process in this jupyter notebook.

From the project instructions

Introduction

Real-world data rarely comes clean. Using Python and its libraries, you will gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling. You will document your wrangling efforts in a Jupyter Notebook, plus showcase them through analyses and visualizations using Python (and its libraries) and/or SQL.

The dataset that you will be wrangling (and analyzing and visualizing) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10. The numerators, though? Almost always greater than 10. 11/10, 12/10, 13/10, etc. Why? Because "they're good dogs Brent." WeRateDogs has over 4 million followers and has received international media coverage.

WeRateDogs downloaded their Twitter archive and sent it to Udacity via email exclusively for you to use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017. More on this soon.

The stages of the Data Wrangling process

- Gathering Data
- Assessing Data
- Cleaning Data

Table of contents

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import requests
import tweepy
import json

%matplotlib inline
```


Gathering Data

In this section I will proceed to gather the data needed for our wrangling. The different data sources/files are:

> 1) WeRateDogs twitter archive (twitter-archive-enhanced.csv'). Name = twitter_archive
>
> 2) Image predictions of what breed of dog it is (image_predictions.tsv) name = predictions Need to access these data using requests from udacity's url (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv)
>
> 3) Each tweet's retweet count and favorite information from tweepy (http://www.tweepy.org/) (API for twitter) name = tweet_meta

Import WeRateDogs Twitter archive

These data were provided in csv format from Udacity

In [2]:

```
twitter_archive = pd.read_csv('twitter-archive-enhanced.csv')
twitter_archive.head(4)
```

Out[2]:

	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=' r...
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href=' r...
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	<a href=' r...
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	<a href=' r...

In [3]:

```
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
```

Import image predictions

These prediction were generated through a neural network that can classify breeds of dogs. The results: a table full of image predictions (the top three only) alongside each tweet ID, 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).

These predictions are hosted on Udacity's website (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv) and we will access them using `requests`

In [4]:

```
#Import using requests
url="https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-prediction
s/image-predictions.tsv"
res = requests.get(url,verify=False) #I had an issue without the verify=False
with open('image_predictions.tsv',mode='wb') as file:
    file.write(res.content)
```

```
C:\Udacity\lib\site-packages\urllib3\connectionpool.py:858: InsecureReques
tWarning: Unverified HTTPS request is being made. Adding certificate verif
ication is strongly advised. See: https://urllib3.readthedocs.io/en/lates
t/advanced-usage.html#ssl-warnings
  InsecureRequestWarning)
```

In [444]:

```
predictions = pd.read_csv('image_predictions.tsv',sep='\t') #I used sep = '\t' after re
ading in in a heloful post from Stackoverflow
#https://stackoverflow.com/questions/9652832/how-to-load-a-tsv-file-into-a-pandas-dataf
rame Thanks @huon
predictions.head()
```

Out[444]:

	tweet_id	jpg_url	img_num
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1

In [6]:

```
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
```

Import tweet's metadata using Tweepy

We want to access: (1) Favorite count and (2) Retweet count for all the ids in twitter_archive Warning from the project description:

> Tweet data is stored in JSON format by Twitter. Getting tweet JSON data via tweet ID using Tweepy is described well in this StackOverflow answer. Note that setting the tweet_mode parameter to 'extended' in the get_status call, i.e., api.get_status(tweet_id, tweet_mode='extended'), can be useful. Also, note that the tweets corresponding to a few tweet IDs in the archive may have been deleted. Try-except blocks may come in handy here

In [7]:

```
#Set up of Tweepy's API
consumer_key = 'here_your_own'
consumer_secret = 'here_your_own'
access_token = 'here_your_own-here_your_own'
access_secret = 'here_your_own'
```

In [8]:

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

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

In [9]:

```
#Test that it works
tweet = api.get_status(892420643555336193)
print(tweet.text)
```

```
This is Phineas. He's a mystical boy. Only ever appears in the hole of a d
onut. 13/10 https://t.co/MgUWQ76dJU
```

In [10]:

```
import time #to time code
```

In [11]:

```
#Testing creating a single json file with the tweet metadata
#Idea from https://stackabuse.com/reading-and-writing-json-to-a-file-in-python/
#Idea to add ._json https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-pyth
on-part-1/
t = time.process_time()
with open('test_individual.txt',mode='w') as outfile:
    tweet_id = 892420643555336193
    tweet = api.get_status(tweet_id, tweet_mode='extended')
    json.dump(tweet._json,outfile)
print(t)
```

```
5.4288348
```

In [12]:

```
#Miniature test
df_test = list(twitter_archive['tweet_id'].sample(10))
df_test
```

Out[12]:

```
[708853462201716736,
 667176164155375616,
 666691418707132416,
 689659372465688576,
 674790488185167872,
 799308762079035393,
 759793422261743616,
 711968124745228288,
 684097758874210310,
 787111942498508800]
```

In [13]:

```
t = time.process_time()
correct = []
incorrect = []
with open('tweet_meta.txt', mode='w') as outfile:
    for tweet_id in twitter_archive['tweet_id']:
        try:
            tweet = api.get_status(tweet_id, tweet_mode ='extended') #Use API to gather
 tweets metadata
            json.dump(tweet._json,outfile) #If we dont use ._json there is an error
            outfile.write('\n') #This is to have each tweet in a new line
            correct.append(tweet_id)
        except Exception as e:
            incorrect.append(tweet_id)
print(t)
```

```
Rate limit reached. Sleeping for: 616
```

```
5.5848358
```

In [20]:

```
len(incorrect)
```

Out[20]:

```
14
```

Nice! It looks like only 14 tweets could not be extracted. lets look at the proportion of tweets correctly extracted

In [21]:

```
len(correct)/twitter_archive.shape[0]
```

Out[21]:

```
0.9940577249575552
```

In [15]:

```
tweet_meta = pd.read_json('tweet_meta.txt',lines=True)
tweet_meta.sample(3)
```

Out[15]:

	contributors	coordinates	created_at	display_text_range	entities	
1813	NaN	NaN	2015-12-14 15:57:56	[0, 140]	{'hashtags': [], 'symbols': [], 'user_mentions...	{'me 676 'id_
1312	NaN	NaN	2016-03-05 16:24:01	[0, 120]	{'hashtags': [], 'symbols': [], 'user_mentions...	NaN
1097	NaN	NaN	2016-05-20 02:18:32	[0, 44]	{'hashtags': [], 'symbols': [], 'user_mentions...	{'me 733 'id_

3 rows × 32 columns

In [22]:

```
tweet_meta.shape
```

Out[22]:

```
(2342, 32)
```

In [23]:

```
twitter_archive.shape
```

Out[23]:

```
(2356, 17)
```

In [25]:

```
predictions.shape
```

Out[25]:

```
(2075, 12)
```

Our gathering phase is completed! We extracted all three sources of data

Assessing Data

In this section I will assess the data quality and tidiness visually and programmatically. The obsevations will be beneath

Data Quality

The four main data quality dimensions are:

- Completeness: do we have all of the records that we should? Do we have missing records or not? Are there specific rows, columns, or cells missing?
- Validity: we have the records, but they're not valid, i.e., they don't conform to a defined schema. A schema is a defined set of rules for data. These rules can be real-world constraints (e.g. negative height is impossible) and table-specific constraints (e.g. unique key constraints in tables).
- Accuracy: inaccurate data is wrong data that is valid. It adheres to the defined schema, but it is still incorrect. Example: a patient's weight that is 5 lbs too heavy because the scale was faulty.
- Consistency: inconsistent data is both valid and accurate, but there are multiple correct ways of referring to the same thing. Consistency, i.e., a standard format, in columns that represent the same data across tables and/or within tables is desired.

Data Tidiness

For a dataset to be considered tidy it needs to meet the following criteria:

- Each variable forms a column.
- Each observation forms a row.
- Each type of observational unit forms a table.

Programmatic and visual assessment

In [26]:

```
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
```

In [27]:

```
twitter_archive.sample(10)
```

Out[27]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
716	783821107061198850	NaN	NaN	2016-10-06 00:08:09 +0000	<a hr r..
2260	667550882905632768	NaN	NaN	2015-11-20 03:51:47 +0000	<a re
727	782305867769217024	NaN	NaN	2016-10-01 19:47:08 +0000	<a hr r..
1088	737826014890496000	NaN	NaN	2016-06-01 02:00:04 +0000	<a hr r..
2221	668480044826800133	NaN	NaN	2015-11-22 17:23:57 +0000	<a hr r..
120	869702957897576449	NaN	NaN	2017-05-30 23:51:58 +0000	<a hr r..
1119	731285275100512256	NaN	NaN	2016-05-14 00:49:30 +0000	<a hr r..
1676	682088079302213632	NaN	NaN	2015-12-30 06:37:25 +0000	<a re
83	876537666061221889	NaN	NaN	2017-06-18 20:30:39 +0000	<a hr r..
22	887517139158093824	NaN	NaN	2017-07-19 03:39:09 +0000	<a hr r..

In [30]:

```
#How many tweets are retweets?
twitter_archive[twitter_archive.text.str[:2] == "RT"].shape[0]
```

Out[30]:

183

In [31]:

```
twitter_archive.duplicated().sum()
```

Out[31]:

0

In [33]:

```
twitter_archive['name'].duplicated().sum()
```

Out[33]:

1399

In [34]:

```
twitter_archive['name'].value_counts()
```

Out[34]:

```
None              745
a                  55
Charlie            12
Cooper             11
Oliver             11
Lucy               11
Lola               10
Tucker             10
Penny              10
Bo                  9
Winston             9
Sadie               8
the                 8
an                  7
Toby                7
Bailey              7
Buddy               7
Daisy               7
Koda                6
Stanley             6
Bella               6
Dave                6
Jack                6
Scout               6
Oscar               6
Leo                 6
Jax                 6
Rusty               6
Milo                6
Louis               5
                  ...
Berb                1
Beya                1
Tedrick             1
Skye                1
Dunkin              1
Crawford            1
Monty               1
Vince               1
Gabby               1
Emanuel             1
Sweets              1
Murphy              1
Benny               1
Trevith             1
Enchilada           1
Iroh                1
Todo                1
Jaspers             1
Kial                1
Crouton             1
Lucia               1
Rodman              1
Reagan              1
Michelangelope      1
Kota                1
Horace              1
Travis              1
Sundance            1
```

```
Jareld              1
Theo                1
Name: name, Length: 957, dtype: int64
```

In [35]:

```
twitter_archive[twitter_archive.name == "a"]
```

Out[35]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp
56	881536004380872706	NaN	NaN	2017-07-02 15:32:16 +0000
649	792913359805018113	NaN	NaN	2016-10-31 02:17:31 +0000
801	772581559778025472	NaN	NaN	2016-09-04 23:46:12 +0000
1002	747885874273214464	NaN	NaN	2016-06-28 20:14:22 +0000
1004	747816857231626240	NaN	NaN	2016-06-28 15:40:07 +0000
1017	746872823977771008	NaN	NaN	2016-06-26 01:08:52 +0000
1049	743222593470234624	NaN	NaN	2016-06-15 23:24:09 +0000
1193	717537687239008257	NaN	NaN	2016-04-06 02:21:30 +0000
1207	715733265223708672	NaN	NaN	2016-04-01 02:51:22 +0000
1340	704859558691414016	NaN	NaN	2016-03-02 02:43:09 +0000

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp
1351	704054845121142784	NaN	NaN	2016-02-28 21:25:30 +0000
1361	703079050210877440	NaN	NaN	2016-02-26 04:48:02 +0000
1368	702539513671897089	NaN	NaN	2016-02-24 17:04:07 +0000
1382	700864154249383937	NaN	NaN	2016-02-20 02:06:50 +0000
1499	692187005137076224	NaN	NaN	2016-01-27 03:26:56 +0000
1737	679530280114372609	NaN	NaN	2015-12-23 05:13:38 +0000
1785	677644091929329666	NaN	NaN	2015-12-18 00:18:36 +0000
1853	675706639471788032	NaN	NaN	2015-12-12 15:59:51 +0000
1854	675534494439489536	NaN	NaN	2015-12-12 04:35:48 +0000
1877	675109292475830276	NaN	NaN	2015-12-11 00:26:12 +0000
1878	675047298674663426	NaN	NaN	2015-12-10 20:19:52 +0000

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp
1923	674082852460433408	NaN	NaN	2015-12-08 04:27:30 +0000
1941	673715861853720576	NaN	NaN	2015-12-07 04:09:13 +0000
1955	673636718965334016	NaN	NaN	2015-12-06 22:54:44 +0000
1994	672604026190569472	NaN	NaN	2015-12-04 02:31:10 +0000
2034	671743150407421952	NaN	NaN	2015-12-01 17:30:22 +0000
2066	671147085991960577	NaN	NaN	2015-11-30 02:01:49 +0000
2116	670427002554466305	NaN	NaN	2015-11-28 02:20:27 +0000
2125	670361874861563904	NaN	NaN	2015-11-27 22:01:40 +0000
2128	670303360680108032	NaN	NaN	2015-11-27 18:09:09 +0000
2146	669923323644657664	NaN	NaN	2015-11-26 16:59:01 +0000
2153	669661792646373376	NaN	NaN	2015-11-25 23:39:47 +0000
2161	669564461267722241	NaN	NaN	2015-11-25 17:13:02 +0000

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp
2191	668955713004314625	NaN	NaN	2015-11-24 00:54:05 +0000
2198	668815180734689280	NaN	NaN	2015-11-23 15:35:39 +0000
2211	668614819948453888	NaN	NaN	2015-11-23 02:19:29 +0000
2218	668507509523615744	NaN	NaN	2015-11-22 19:13:05 +0000
2222	668466899341221888	NaN	NaN	2015-11-22 16:31:42 +0000
2235	668171859951755264	NaN	NaN	2015-11-21 20:59:20 +0000
2249	667861340749471744	NaN	NaN	2015-11-21 00:25:26 +0000
2255	667773195014021121	NaN	NaN	2015-11-20 18:35:10 +0000
2264	667538891197542400	NaN	NaN	2015-11-20 03:04:08 +0000
2273	667470559035432960	NaN	NaN	2015-11-19 22:32:36 +0000
2287	667177989038297088	NaN	NaN	2015-11-19 03:10:02 +0000
2304	666983947667116034	NaN	NaN	2015-11-18 14:18:59 +0000
2311	666781792255496192	NaN	NaN	2015-11-18 00:55:42 +0000
2314	666701168228331520	NaN	NaN	2015-11-17 19:35:19 +0000

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp
2327	666407126856765440	NaN	NaN	2015-11-17 00:06:54 +0000
2334	666293911632134144	NaN	NaN	2015-11-16 16:37:02 +0000
2347	666057090499244032	NaN	NaN	2015-11-16 00:55:59 +0000
2348	666055525042405380	NaN	NaN	2015-11-16 00:49:46 +0000
2350	666050758794694657	NaN	NaN	2015-11-16 00:30:50 +0000
2352	666044226329800704	NaN	NaN	2015-11-16 00:04:52 +0000
2353	666033412701032449	NaN	NaN	2015-11-15 23:21:54 +0000

In [37]:

```
twitter_archive.describe()
```

Out[37]:

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

In [39]:

```
twitter_archive[twitter_archive.rating_denominator != 10].shape[0]
```

Out[39]:

23

In [48]:

```
twitter_archive.query('rating_numerator > 20')[['tweet_id','text','rating_numerator','r
ating_denominator']]
```

Out[48]:

	tweet_id	text	rating_numerator	rating_denomina
188	855862651834028034	@dhmontgomery We also gave snoop dogg a 420/10...	420	10
189	855860136149123072	@s8n You tried very hard to portray this good ...	666	10
290	838150277551247360	@markhoppus 182/10	182	10
313	835246439529840640	@jonnysun @Lin_Manuel ok jomny I know you're e...	960	0
340	832215909146226688	RT @dog_rates: This is Logan, the Chow who liv...	75	10
433	820690176645140481	The floofs have been released I repeat the flo...	84	70
516	810984652412424192	Meet Sam. She smiles 24/7 & secretly aspir...	24	7
695	786709082849828864	This is Logan, the Chow who lived. He solemnly...	75	10
763	778027034220126208	This is Sophie. She's a Jubilant Bush Pupper. ...	27	10
902	758467244762497024	Why does this never happen at my front door.....	165	150
979	749981277374128128	This is Atticus. He's quite simply America af....	1776	10
1120	731156023742988288	Say hello to this unbelievably well behaved sq...	204	170
1202	716439118184652801	This is Bluebert. He just saw that both #Final...	50	50

	tweet_id	text	rating_numerator	rating_denomina
1228	713900603437621249	Happy Saturday here's 9 puppers on a bench. 99...	99	90
1254	710658690886586372	Here's a brigade of puppers. All look very pre...	80	80
1274	709198395643068416	From left to right:\nCletus, Jerome, Alejandro...	45	50
1351	704054845121142784	Here is a whole flock of puppers. 60/50 I'll ...	60	50
1433	697463031882764288	Happy Wednesday here's a bucket of pups. 44/40...	44	40
1634	684225744407494656	Two sneaky puppers were not initially seen, mo...	143	130
1635	684222868335505415	Someone help the girl is being mugged. Several...	121	110
1712	680494726643068929	Here we have uncovered an entire battalion of ...	26	10
1779	677716515794329600	IT'S PUPPERGEDDON. Total of 144/120 ...I think...	144	120
1843	675853064436391936	Here we have an entire platoon of puppers. Tot...	88	80
2074	670842764863651840	After so many requests... here	420	10

In [55]:

```
twitter_archive[twitter_archive['text'].str.contains("@")][['tweet_id','text','rating_n
umerator','rating_denominator']]
#Did not deliver any particular useful information
```

Out[55]:

	tweet_id	text	rating_numerator
19	888202515573088257	RT @dog_rates: This is Canela. She attempted s...	13
30	886267009285017600	@NonWhiteHat @MayhewMayhem omg hello tanner yo...	12
32	886054160059072513	RT @Athletics: 12/10 #BATP https://t.co/WxwJmv...	12
36	885311592912609280	RT @dog_rates: This is Lilly. She just paralle...	13
55	881633300179243008	@roushfenway These are good dogs but 17/10 is ...	17
62	880095782870896641	Please don't send in photos without dogs in th...	11
64	879674319642796034	@RealKentMurphy 14/10 confirmed	14
68	879130579576475649	RT @dog_rates: This is Emmy. She was adopted t...	14
73	878404777348136964	RT @dog_rates: Meet Shadow. In an attempt to r...	13
74	878316110768087041	RT @dog_rates: Meet Terrance. He's being yelle...	11
78	877611172832227328	RT @rachel2195: @dog_rates the boyfriend and h...	14
91	874434818259525634	RT @dog_rates: This is Coco. At first I though...	12
95	873697596434513921	RT @dog_rates: This is Walter. He won't start ...	14
97	873337748698140672	RT @dog_rates: This is Sierra. She's one preci...	12
101	872668790621863937	RT @loganamnosis: Penelope here is doing me qu...	14
109	871166179821445120	RT @dog_rates: This is Dawn. She's just checki...	12
113	870726314365509632	@ComplicitOwl @ShopWeRateDogs >10/10 is res...	10
118	869988702071779329	RT @dog_rates: We only rate dogs. This is quit...	12
124	868639477480148993	RT @dog_rates: Say hello to Cooper. His expres...	12
130	867072653475098625	RT @rachaeleasler: these @dog_rates hats are 1...	13

	tweet_id	text	rating_numerator
132	866816280283807744	RT @dog_rates: This is Jamesy. He gives a kiss...	13
137	866094527597207552	RT @dog_rates: Here's a pupper before and afte...	12
145	863553081350529029	This is Neptune. He's a backpup vocalist for t...	13
146	863471782782697472	RT @dog_rates: Say hello to Quinn. She's quite...	13
148	863427515083354112	@Jack_Septic_Eye I'd need a few more pics to p...	12
155	861769973181624320	RT @dog_rates: "Good afternoon class today we'...	13
159	860981674716409858	RT @dog_rates: Meet Lorenzo. He's an avid nift...	13
160	860924035999428608	RT @tallylott: h*ckin adorable promposal. 13/1...	13
165	860177593139703809	RT @dog_rates: Ohboyohboyohboyohboyohboyohboyo...	10
171	858860390427611136	RT @dog_rates: Meet Winston. He knows he's a l...	12
...	...	...	...
1349	704134088924532736	This sneezy pupper is just adorable af. 12/10 ...	12
1364	702899151802126337	Say hello to Luna. Her tongue is malfunctionin...	12
1369	702332542343577600	This is Rudy. He's going to be a star. 13/10 t...	13
1416	698635131305795584	Here we are witnessing five Guatemalan Birch F...	12
1419	698342080612007937	This is Maximus. He's training for the tetherb...	11
1443	696744641916489729	This is Klevin. He doesn't want his family bra...	10
1459	695064344191721472	This may be the greatest video I've ever been ...	4
1461	694925794720792577	Please only send in dogs. This t-rex is very s...	5
1466	694342028726001664	It's okay pup. This happens every time I liste...	11
1471	693993230313091072	These lil fellas are the best of friends. 12/1...	12

	tweet_id	text	rating_numerator
1479	693582294167244802	Personally I'd give him an 11/10. Not sure why...	11
1480	693486665285931008	This is Lincoln. He doesn't understand his new...	11
1486	693109034023534592	"Thank you friend that was a swell petting" 11...	11
1502	692041934689402880	This is Teddy. His head is too heavy. 13/10 (v...	13
1523	690607260360429569	12/10 @LightningHoltt	12
1528	690348396616552449	This is Oddie. He's trying to communicate. 12/...	12
1534	689993469801164801	Here we are witnessing a rare High Stepping Al...	12
1549	689255633275777024	This is Ferg. He swallowed a chainsaw. 1 like ...	10
1562	688211956440801280	This is Derby. He's a superstar. 13/10 (vid by...	13
1566	687841446767013888	13/10 I can't stop watching this (vid by @k8ly...	13
1570	687732144991551489	This is Ember. That's the q-tip she owes money...	11
1577	687399393394311168	This is Barry. He's very fast. I hope he finds...	10
1586	686760001961103360	This pupper forgot how to walk. 12/10 happens ...	12
1596	686286779679375361	When bae calls your name from across the room....	12
1607	685663452032069632	Meet Brooks. He's confused by the almighty bal...	12
1884	674800520222154752	This is Tedders. He broke his leg saving babie...	11
1914	674330906434379776	13/10\n@ABC7	13
2189	668967877119254528	12/10 good shit Bubka\n@wane15	12
2259	667550904950915073	RT @dogratingrating: Exceptional talent. Origi...	12
2260	667550882905632768	RT @dogratingrating: Unoriginal idea. Blatant ...	5

267 rows × 4 columns

In [56]:

```
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 [57]:

```
predictions.shape
```

Out[57]:

```
(2075, 12)
```

In [64]:

```
#number of values missing compared to twitter_archive
twitter_archive.tweet_id.nunique() - predictions.tweet_id.nunique()
```

Out[64]:

```
281
```

In [156]:

```
predictions.describe()
```

Out[156]:

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

In [83]:

```
#Check all the categories togehter
p1, p2, p3 = predictions['p1'].copy(), predictions['p2'].copy(), predictions['p3'].copy
()
categories = pd.concat([p1,p2,p3])
categories.value_counts()
```

Out[83]:

```
golden_retriever                  290
Labrador_retriever                283
Chihuahua                         185
Pembroke                          143
Cardigan                          115
Pomeranian                        109
toy_poodle                        105
pug                                97
chow                               96
cocker_spaniel                     95
French_bulldog                     93
Chesapeake_Bay_retriever           91
Eskimo_dog                         83
beagle                             77
kuvasz                             76
Siberian_husky                     72
Samoyed                            70
Staffordshire_bullterrier          70
malamute                           69
Pekinese                           63
kelpie                             62
American_Staffordshire_terrier     58
miniature_pinscher                 57
Great_Pyrenees                     55
miniature_poodle                   54
collie                             51
Italian_greyhound                  49
German_shepherd                    49
seat_belt                          49
Shetland_sheepdog                  48
                                  ...
valley                              1
junco                               1
confectionery                       1
radio_telescope                     1
tiger_cat                           1
necklace                            1
alp                                 1
waffle_iron                         1
switch                              1
drake                               1
mashed_potato                       1
lacewing                            1
pop_bottle                          1
banded_gecko                        1
volcano                             1
apron                               1
mushroom                            1
bald_eagle                          1
pelican                             1
kimono                              1
breastplate                         1
syringe                             1
pretzel                             1
wolf_spider                         1
toaster                             1
sulphur_butterfly                   1
swimming_trunks                     1
pole                                1
```

```
dock                            1
military_uniform                1
Length: 634, dtype: int64
```

In [157]:

```
predictions.sample(15)
```

Out[157]:

	tweet_id	jpg_url	im
290	671166507850801152	https://pbs.twimg.com/media/CVB2TnWUYAA2pAU.jpg	1
981	707377100785885184	https://pbs.twimg.com/media/CdEbt0NXIAQH3Aa.jpg	1
187	669367896104181761	https://pbs.twimg.com/media/CUoSjTnWwAANNak.jpg	1
1338	758467244762497024	https://pbs.twimg.com/ext_tw_video_thumb/75846...	1
325	671882082306625538	https://pbs.twimg.com/media/CVMBL_LWUAAsvrL.jpg	1
252	670717338665226240	https://pbs.twimg.com/media/CU7d2vKUcAAFZyI.jpg	1
345	672272411274932228	https://pbs.twimg.com/media/CVRkLuJWUAAhhYp.jpg	2
1244	747461612269887489	https://pbs.twimg.com/media/Cl-EXHSWkAE2IN2.jpg	1
1452	776813020089548800	https://pbs.twimg.com/media/CsfLUDbXEAAu0VF.jpg	1
1089	719332531645071360	https://pbs.twimg.com/media/CfuVGl3WEAEKb16.jpg	1
13	666082916733198337	https://pbs.twimg.com/media/CT5m4VGWEAAtKc8.jpg	1
1171	736365877722001409	https://pbs.twimg.com/media/CjgYyuvWkAAHU8g.jpg	3
244	670465786746662913	https://pbs.twimg.com/media/CU35E7VWEAAKYBy.jpg	1
1702	817171292965273600	https://pbs.twimg.com/media/C1cs8uAWgAEwbXc.jpg	1
71	667200525029539841	https://pbs.twimg.com/media/CUJfVMPXIAAgbue.jpg	1

In [84]:

```
predictions[predictions.p1_dog == False]
```

Out[84]:

	tweet_id	jpg_url	i
6	666051853826850816	https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg	1
8	666057090499244032	https://pbs.twimg.com/media/CT5PY90WoAAQGLo.jpg	1
17	666104133288665088	https://pbs.twimg.com/media/CT56LSZWoAAlJj2.jpg	1
18	666268910803644416	https://pbs.twimg.com/media/CT8QCd1WEAADXws.jpg	1
21	666293911632134144	https://pbs.twimg.com/media/CT8mx7KW4AEQu8N.jpg	1
22	666337882303524864	https://pbs.twimg.com/media/CT9OwFlWEAMuRje.jpg	1
25	666362758909284353	https://pbs.twimg.com/media/CT9lXGsUcAAyUFt.jpg	1
29	666411507551481857	https://pbs.twimg.com/media/CT-RugiWIAELEaq.jpg	1
33	666430724426358785	https://pbs.twimg.com/media/CT-jNYqW4AAPi2M.jpg	1
43	666776908487630848	https://pbs.twimg.com/media/CUDeDoWUYAAD-EM.jpg	1
45	666786068205871104	https://pbs.twimg.com/media/CUDmZlkWcAAIPPe.jpg	1
50	666837028449972224	https://pbs.twimg.com/media/CUEUva1WsAA2jPb.jpg	1
51	666983947667116034	https://pbs.twimg.com/media/CUGaXDhW4AY9JUH.jpg	1
52	666996132027977728	https://pbs.twimg.com/media/CUGlb6iUwAITEbW.jpg	1
53	667012601033924608	https://pbs.twimg.com/media/CUG0bC0U8AAw2su.jpg	1
56	667065535570550784	https://pbs.twimg.com/media/CUHkkJpXIAA2w3n.jpg	1
69	667188689915760640	https://pbs.twimg.com/media/CUJUk2iWUAAVtOv.jpg	1
73	667369227918143488	https://pbs.twimg.com/media/CUL4xR9UkAEdlJ6.jpg	1
77	667437278097252352	https://pbs.twimg.com/media/CUM2qWaWoAUZ06L.jpg	1

	tweet_id	jpg_url	ir
78	667443425659232256	https://pbs.twimg.com/media/CUM8QZwW4AAVsBl.jpg	1
87	667524857454854144	https://pbs.twimg.com/media/CUOGUfJW4AA_eni.jpg	1
93	667549055577362432	https://pbs.twimg.com/media/CUOcVCwWsAERUKY.jpg	1
94	667550882905632768	https://pbs.twimg.com/media/CUObvUJVEAAnYPF.jpg	1
95	667550904950915073	https://pbs.twimg.com/media/CUOb_gUUkAACXdS.jpg	1
96	667724302356258817	https://pbs.twimg.com/media/CUQ7tv3W4AA3KlI.jpg	1
98	667766675769573376	https://pbs.twimg.com/media/CURiQMnUAAAPT2M.jpg	1
100	667782464991965184	https://pbs.twimg.com/media/CURwm3cUkAARcO6.jpg	1
103	667806454573760512	https://pbs.twimg.com/media/CUSGbXeVAAAgztZ.jpg	1
106	667866724293877760	https://pbs.twimg.com/media/CUS9PlUWwAANeAD.jpg	1
107	667873844930215936	https://pbs.twimg.com/media/CUTDtyGXIAARxus.jpg	1
...	...	...	..
1900	851464819735769094	https://pbs.twimg.com/media/C9ECujZXsAAPCSM.jpg	2
1902	851861385021730816	https://pbs.twimg.com/media/C8W6sY_W0AEmttW.jpg	1
1904	852189679701164033	https://pbs.twimg.com/media/C9OV99SXsAEmj1U.jpg	1
1905	852226086759018497	https://pbs.twimg.com/ext_tw_video_thumb/85222...	1
1906	852311364735569921	https://pbs.twimg.com/media/C9QEqZ7XYAIR7fS.jpg	1
1910	853299958564483072	https://pbs.twimg.com/media/C9eHyF7XgAAOxPM.jpg	1
1931	859074603037188101	https://pbs.twimg.com/media/C-wLyufW0AA546I.jpg	1
1932	859196978902773760	https://pbs.twimg.com/ext_tw_video_thumb/85919...	1
1936	860184849394610176	https://pbs.twimg.com/media/C-_9jWWUwAAnwkd.jpg	1

	tweet_id	jpg_url	ir
1937	860276583193509888	https://pbs.twimg.com/media/C_BQ_NlVwAAgYGD.jpg	1
1940	860924035999428608	https://pbs.twimg.com/media/C_KVJjDXsAEUCWn.jpg	2
1942	861288531465048066	https://pbs.twimg.com/ext_tw_video_thumb/86128...	1
1944	861769973181624320	https://pbs.twimg.com/media/CzG425nWgAAnP7P.jpg	2
1946	862457590147678208	https://pbs.twimg.com/media/C_gQmaTUMAAPYSS.jpg	1
1953	863907417377173506	https://pbs.twimg.com/media/C_03NPeUQAAgrMl.jpg	1
1956	864873206498414592	https://pbs.twimg.com/media/DAClmHkXcAA1kSv.jpg	2
1970	868880397819494401	https://pbs.twimg.com/media/DA7iHL5U0AA1OQo.jpg	1
1975	870063196459192321	https://pbs.twimg.com/media/DBMV3NnXUAAm0Pp.jpg	1
1979	870804317367881728	https://pbs.twimg.com/media/DBW35ZsVoAEWZUU.jpg	1
1984	872122724285648897	https://pbs.twimg.com/media/DBpm-5UXcAUeCru.jpg	1
1992	873697596434513921	https://pbs.twimg.com/media/DA7iHL5U0AA1OQo.jpg	1
2012	879050749262655488	https://pbs.twimg.com/media/DDMD_phXoAQ1qf0.jpg	1
2013	879376492567855104	https://pbs.twimg.com/media/DDQsQGFV0AAw6u9.jpg	1
2021	880935762899988482	https://pbs.twimg.com/media/DDm2Z5aXUAEDS2u.jpg	1
2022	881268444196462592	https://pbs.twimg.com/media/DDrk-f9WAAI-WQv.jpg	1
2026	882045870035918850	https://pbs.twimg.com/media/DD2oCl2WAAEl_4a.jpg	1
2046	886680336477933568	https://pbs.twimg.com/media/DE4fEDzWAAAyHMM.jpg	1
2052	887517139158093824	https://pbs.twimg.com/ext_tw_video_thumb/88751...	1
2071	891689557279858688	https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg	1

	tweet_id	jpg_url	i
2074	892420643555336193	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	1

543 rows × 12 columns

In [85]:

```
tweet_meta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2342 entries, 0 to 2341
Data columns (total 32 columns):
contributors                    0 non-null float64
coordinates                     0 non-null float64
created_at                      2342 non-null datetime64[ns]
display_text_range              2342 non-null object
entities                        2342 non-null object
extended_entities               2068 non-null object
favorite_count                  2342 non-null int64
favorited                       2342 non-null bool
full_text                       2342 non-null object
geo                             0 non-null float64
id                              2342 non-null int64
id_str                          2342 non-null int64
in_reply_to_screen_name         77 non-null object
in_reply_to_status_id           77 non-null float64
in_reply_to_status_id_str       77 non-null float64
in_reply_to_user_id             77 non-null float64
in_reply_to_user_id_str         77 non-null float64
is_quote_status                 2342 non-null bool
lang                            2342 non-null object
place                           1 non-null object
possibly_sensitive              2206 non-null float64
possibly_sensitive_appealable   2206 non-null float64
quoted_status                   24 non-null object
quoted_status_id                26 non-null float64
quoted_status_id_str            26 non-null float64
quoted_status_permalink         26 non-null object
retweet_count                   2342 non-null int64
retweeted                       2342 non-null bool
retweeted_status                168 non-null object
source                          2342 non-null object
truncated                       2342 non-null bool
user                            2342 non-null object
dtypes: bool(4), datetime64[ns](1), float64(11), int64(4), object(12)
memory usage: 521.5+ KB
```

In [91]:

```
tweet_meta_simple = tweet_meta[['id','favorite_count','retweet_count']]
tweet_meta_simple.sample(10)
```

Out[91]:

	id	favorite_count	retweet_count
2113	670319130621435904	3928	1261
306	835152434251116546	23632	3251
144	863079547188785154	8802	1117
1945	673576835670777856	1412	582
464	816091915477250048	9531	2364
304	835246439529840640	2196	78
27	886680336477933568	22073	4379
648	791406955684368384	14082	4536
1713	679877062409191424	2065	690
650	791026214425268224	0	4532

In [92]:

```
tweet_meta_simple.info()
```

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

In [93]:

```
tweet_meta_simple.describe()
```

Out[93]:

	id	favorite_count	retweet_count
count	2.342000e+03	2342.000000	2342.000000
mean	7.422212e+17	7981.822374	2941.923570
std	6.832408e+16	12354.766199	4947.875738
min	6.660209e+17	0.000000	0.000000
25%	6.783509e+17	1377.250000	591.000000
50%	7.186224e+17	3473.500000	1374.500000
75%	7.986971e+17	9780.250000	3430.500000
max	8.924206e+17	164599.000000	83869.000000

In [98]:

```
twitter_archive.shape[0] - tweet_meta_simple.shape[0]
```

Out[98]:

14

Observations

`tweet_meta` observations

Data Quality

- id is an int and not a string
- 14 missing values comparing twitter_archive and tweet_meta

Data tidiness

- All columns except id, favorite_count, and retweet_count are irrelevant for this analysis

`predictions` observations

Data quality

- 281 potentially missing values compared to twitter_archive
- tweet_id is an integer
- p1, p2, p3 are strings and not categories (#revision: actually, there are so many, best to let it as string)
- p1, p2, p3 consistency some lower case some capitlized
- p1_dog 543 predictions which are not dogs

Data tidiness

- p1, p2, p3 names are not really informative

`twitter_archive`

Data quality issues

- tweet_id is integer instead of string
- timestamp is not date format
- text: 183 are retweets
- Dog Stage: None instead of NaN
- Some columns (e.g. in_reply_to_status_id, retweeted_status_id) have only data for a small subset of population
- expanded_urls have 59 missing values (no media)
- type of dog does not have category data type (see also data tidiness)
- name: 745 dogs do not have name
- name: 55 dogs have name "a". Closer inspection shows that they don't have names
- name: duplicated names
- rating denominator: 23 tweets are not /10 (e.g. outlier 170)
- Whenever there's a '.', only the digits after the '.' are included in the numerator (e.g. tweet_id 832215909146226688)
- rating numerator: 24 are >20 / closer inspection shows that most are deviant from the usual rating

Data tidiness issues

- Breed of dog are not in only one column (violates 'each variable forms a column')

****

Cleaning Data

In this section I will clean some of the observations that I made during the assessment phase

The observations I will clean are:

- drop all unecessary columns for `twitter_archive`, `predictions`, `tweet_meta`
- change type of tweet_id from integer to strig for `twitter_archive`, `predictions`, `tweet_meta`
- `tweet_meta` : Change column name from id to tweet_id
- `predictions`: change column names
- `predictions`: drop rows where p1 is not a dog
- `predictions`: lower case p1, p2, p3
- `twitter_archive`: change breed of dog to a single column
- `twitter_archive`: dog stage change from none to NaN
- `twitter_archive`: drop / adapt rows where denominator is not /10
- merge the relevant columns fro the three datasets into one dataframe

In [497]:

```
predictions = pd.read_csv('image_predictions.tsv',sep='\t') #I used sep = '\t' after re
ading in in a heloful post from Stackoverflow
#https://stackoverflow.com/questions/9652832/how-to-load-a-tsv-file-into-a-pandas-dataf
rame Thanks @huon
predictions.sample()
```

Out[497]:

	tweet_id	jpg_url	im
1295	751937170840121344	https://pbs.twimg.com/media/Cm9q2d3XEAAqO2m.jpg	1

In [498]:

```
#Create _clean dataframes
twitter_archive_clean = twitter_archive.copy()
predictions_clean = predictions.copy()
meta_data_clean = tweet_meta.copy()
```

In [499]:

```
predictions_clean.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 [500]:

```
#drop all unecessary columns for `twitter_archive`, `predictions`, `tweet_meta`

twitter_archive_clean = twitter_archive_clean[['tweet_id','text','timestamp','rating_nu
merator','rating_denominator','name','doggo','floofer','pupper','puppo','expanded_urls'
]]
predictions_clean.drop('img_num',axis=1,inplace=True)
meta_data_clean = meta_data_clean[['id','favorite_count','retweet_count']]
```

In [503]:

```
# tweet_meta : Change column name from id to tweet_id
meta_data_clean.rename(columns={'id':'tweet_id'},inplace=True)
```

In [504]:

```
#change type of tweet_id from integer to strig for `twitter_archive`, `predictions`, `t
weet_meta`
twitter_archive_clean['tweet_id'] = twitter_archive_clean['tweet_id'].astype(str)
predictions_clean['tweet_id'] = predictions_clean['tweet_id'].astype(str)
meta_data_clean['tweet_id'] = meta_data_clean['tweet_id'].astype(str)
```

In [505]:

```
#`predictions`: change column names
predictions_clean.rename(columns={'p1':'prediction_1','p2':'prediction_2','p3':'predict
ion_3'},inplace=True)
```

In [506]:

```
#`predictions`: drop rows where p1 is not a dog
predictions_clean = predictions_clean[predictions_clean['p1_dog'] == True]
```

In [507]:

```
#`predictions`: lower case p1, p2, p3
predictions = ['prediction_1','prediction_2','prediction_3']
for prediction in predictions:
    predictions_clean[prediction] = predictions_clean[prediction].str.lower()
```

In [508]:

```
twitter_archive_clean.shape[0]
```

Out[508]:

2356

In [509]:

```
twitter_archive_clean.sample(3)
```

Out[509]:

	tweet_id	text	timestamp	rating_numerator	rating_deno
539	806576416489959424	Hooman catch successful. Massive hit by dog. F...	2016-12-07 19:09:37 +0000	13	10
1632	684460069371654144	This is Jeph. He's a Western Sagittarius Dookm...	2016-01-05 19:42:51 +0000	10	10
1949	673689733134946305	When you're having a blast and remember tomorr...	2015-12-07 02:25:23 +0000	11	10

In [519]:

```
doggo_n = twitter_archive_clean[twitter_archive_clean.doggo == 'doggo'].shape[0]
floofer_n = twitter_archive_clean[twitter_archive_clean.floofer == 'floofer'].shape[0]
pupper_n = twitter_archive_clean[twitter_archive_clean.pupper == 'pupper'].shape[0]
puppo_n = twitter_archive_clean[twitter_archive_clean.puppo == 'puppo'].shape[0]
sum_n = doggo_n + floofer_n + pupper_n + puppo_n
print('numbers of doggo: {}, floofer: {}, pupper: {}, puppo: {}, sum:{}'.format(doggo_n
,floofer_n,pupper_n,puppo_n,sum_n))
```

```
numbers of doggo: 97, floofer: 10, pupper: 257, puppo: 30, sum:394
```

In [520]:

```
#- `twitter_archive`: change breed of dog to a single column
twitter_archive_clean = pd.melt(twitter_archive_clean, id_vars = ['tweet_id','timestam
p','text','rating_numerator','rating_denominator','name','expanded_urls'],var_name='dog
_stage',value_vars=['doggo','floofer','pupper','puppo'])
```

In [534]:

```
twitter_archive_clean.shape[0]
```

Out[534]:

9424

In [533]:

```
twitter_archive_clean[twitter_archive_clean['tweet_id'] == '892420643555336193'] #This
 shows that the reason there are duplicates is because a new row is created for every c
ategory
```

Out[533]:

	tweet_id	timestamp	text	rating_numerator	rating_denominat
0	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	13	10
2356	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	13	10
4712	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	13	10
7068	892420643555336193	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	13	10

In [523]:

```
#Clean new column dog_stage
twitter_archive_clean.drop('dog_stage',axis=1,inplace=True)
twitter_archive_clean.rename(columns={'value':'dog_stage'},inplace=True)
```

In [524]:

```
twitter_archive_clean.dog_stage.value_counts()
```

Out[524]:

```
None       9030
pupper      257
doggo        97
puppo        30
floofer      10
Name: dog_stage, dtype: int64
```

In [529]:

```
#`twitter_archive`: dog stage change from none to NaN
twitter_archive_clean['dog_stage'].replace('None',np.NaN,inplace=True)
```

It seems we have more values that we should. Let's drop all the duplicated

In [550]:

```
twitter_archive_clean.drop_duplicates(inplace=True)
```

In [551]:

```
twitter_archive_clean.shape[0]
```

Out[551]:

```
2750
```

Still some duplicated.. why? Let's look at an example

In [556]:

```
twitter_archive_clean[twitter_archive_clean.tweet_id.duplicated()][:1]
```

Out[556]:

	tweet_id	timestamp	text	rating_numerator	rating_denor
2365	890240255349198849	2017-07-26 15:59:51 +0000	This is Cassie. She is a college pup. Studying...	14	10

In [557]:

```
twitter_archive_clean[twitter_archive_clean.tweet_id == '890240255349198849']
```

Out[557]:

	tweet_id	timestamp	text	rating_numerator	rating_denomin
9	890240255349198849	2017-07-26 15:59:51 +0000	This is Cassie. She is a college pup. Studying...	14	10
2365	890240255349198849	2017-07-26 15:59:51 +0000	This is Cassie. She is a college pup. Studying...	14	10

Ok, so we need to remove the NaNs whenever there is an available dog_stage

In [582]:

```
to_drop = twitter_archive_clean[twitter_archive_clean.dog_stage.notna()]['tweet_id'].to
list()
```

In [621]:

```
twitter_archive_clean[twitter_archive_clean['tweet_id'].isin(to_drop)].shape
```

Out[621]:

```
(774, 8)
```

In [635]:

```
twitter_archive_clean[twitter_archive_clean['dog_stage'].isna()].shape
```

Out[635]:

```
(2356, 8)
```

In [636]:

```
to_drop_index = twitter_archive_clean[(twitter_archive_clean['tweet_id'].isin(to_drop))
&(twitter_archive_clean['dog_stage'].isna())]
```

In [637]:

```
to_drop_index_list = to_drop_index.index.tolist()
len(to_drop_index_list)
```

Out[637]:

```
380
```

In [638]:

```
for dropv in to_drop_index_list:
    twitter_archive_clean = twitter_archive_clean[twitter_archive_clean.index != dropv]
```

In [651]:

```
twitter_archive_clean.shape
```

Out[651]:

```
(2370, 8)
```

In [650]:

```
twitter_archive_clean.dog_stage.value_counts()
```

Out[650]:

```
pupper     257
doggo       97
puppo       30
floofer     10
Name: dog_stage, dtype: int64
```

In [653]:

```
twitter_archive_clean[twitter_archive_clean.tweet_id == '817777686764523521']
```

Out[653]:

	tweet_id	timestamp	text	rating_numerator	rating_denominato
460	817777686764523521	2017-01-07 16:59:28 +0000	This is Dido. She's playing the lead role in "...	13	10
5172	817777686764523521	2017-01-07 16:59:28 +0000	This is Dido. She's playing the lead role in "...	13	10

In [643]:

```
twitter_archive_clean[twitter_archive_clean.tweet_id.duplicated()].shape
```

Out[643]:

```
(14, 8)
```

Seems there are still 14 with duplicates. Lets check those out

In [657]:

```
dupli = twitter_archive_clean[twitter_archive_clean.tweet_id.duplicated()]['tweet_id'].
tolist()
twitter_archive_clean[twitter_archive_clean.tweet_id.isin(dupli)].sort_values(by=['twee
t_id'])[['tweet_id','text','dog_stage']]
```

Out[657]:

	tweet_id	text	dog_stage
1113	733109485275860992	Like father (doggo), like son (pupper). Both 1...	doggo
5825	733109485275860992	Like father (doggo), like son (pupper). Both 1...	pupper
1063	741067306818797568	This is just downright precious af. 12/10 for ...	doggo
5775	741067306818797568	This is just downright precious af. 12/10 for ...	pupper
5668	751583847268179968	Please stop sending it pictures that don't eve...	pupper
956	751583847268179968	Please stop sending it pictures that don't eve...	doggo
889	759793422261743616	Meet Maggie & Lila. Maggie is the doggo, L...	doggo
5601	759793422261743616	Meet Maggie & Lila. Maggie is the doggo, L...	pupper
822	770093767776997377	RT @dog_rates: This is just downright precious...	doggo
5534	770093767776997377	RT @dog_rates: This is just downright precious...	pupper
778	775898661951791106	RT @dog_rates: Like father (doggo), like son (...	doggo
5490	775898661951791106	RT @dog_rates: Like father (doggo), like son (...	pupper
733	781308096455073793	Pupper butt 1, Doggo 0. Both 12/10 https://t.c...	doggo
5445	781308096455073793	Pupper butt 1, Doggo 0. Both 12/10 https://t.c...	pupper
705	785639753186217984	This is Pinot. He's a sophisticated doggo. You...	doggo
5417	785639753186217984	This is Pinot. He's a sophisticated doggo. You...	pupper
5287	801115127852503040	This is Bones. He's being haunted by another d...	pupper
575	801115127852503040	This is Bones. He's being haunted by another d...	doggo
565	802265048156610565	Like doggo, like pupper version 2. Both 11/10 ...	doggo
5277	802265048156610565	Like doggo, like pupper version 2. Both 11/10 ...	pupper
531	808106460588765185	Here we have Burke (pupper) and Dexter (doggo)...	doggo
5243	808106460588765185	Here we have Burke (pupper) and Dexter (doggo)...	pupper
5172	817776686764523521	This is Dido. She's playing the lead role in "...	pupper
460	817776686764523521	This is Dido. She's playing the lead role in "...	doggo
200	854010172552949760	At first I thought this was a shy doggo, but i...	doggo

	tweet_id	text	dog_stage
2556	854010172552949760	At first I thought this was a shy doggo, but i...	floofer
191	855851453814013952	Here's a puppo participating in the #ScienceMa...	doggo
7259	855851453814013952	Here's a puppo participating in the #ScienceMa...	puppo

Ok so it seems there are usually two dogs here... let's keep it that way

In [678]:

```
#Drop columns where denominator is not 10 in twitter_archive
twitter_archive_clean = twitter_archive_clean[twitter_archive_clean.rating_denominator
== 10]
```

In [679]:

```
twitter_archive_clean.sample()
```

Out[679]:

	tweet_id	timestamp	text	rating_numerato
1490	692901601640583168	2016-01-29 02:46:29	"Fuck the system" 10/10 https://t.co/N0OADmCnVV	10

In [680]:

```
#Change timestamp format to date time
twitter_archive_clean['timestamp'] = pd.to_datetime(twitter_archive_clean.timestamp)
```

In [681]:

```
#merge the relevant columns fro the three datasets into one dataframe
df_combined = pd.merge(twitter_archive_clean, predictions_clean,on='tweet_id',how='lef
t')
df_combined = pd.merge(df_combined, meta_data_clean,on='tweet_id',how='left')
```

In [682]:

```
#Remove duplicates
df_combined.drop_duplicates(inplace=True)
```

Test cleaning

In [683]:

```
df_combined.tail(4)
```

Out[683]:

	tweet_id	timestamp	text	rating_numerator	rating_denom
2343	751132876104687617	2016-07-07 19:16:47	This is Cooper. He's just so damn happy. 10/10...	10	10
2344	744995568523612160	2016-06-20 20:49:19	This is Abby. She got her face stuck in a glas...	9	10
2345	743253157753532416	2016-06-16 01:25:36	This is Kilo. He cannot reach the snackum. Nif...	10	10
2346	738537504001953792	2016-06-03 01:07:16	This is Bayley. She fell asleep trying to esca...	11	10

In [664]:

```
df_combined.shape
```

Out[664]:

```
(2347, 20)
```

In [665]:

```
df_combined.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2347 entries, 0 to 2346
Data columns (total 20 columns):
tweet_id              2347 non-null object
timestamp             2347 non-null datetime64[ns]
text                  2347 non-null object
rating_numerator      2347 non-null int64
rating_denominator    2347 non-null int64
name                  2347 non-null object
expanded_urls         2292 non-null object
dog_stage             394 non-null object
jpg_url               1529 non-null object
prediction_1          1529 non-null object
p1_conf               1529 non-null float64
p1_dog                1529 non-null object
prediction_2          1529 non-null object
p2_conf               1529 non-null float64
p2_dog                1529 non-null object
prediction_3          1529 non-null object
p3_conf               1529 non-null float64
p3_dog                1529 non-null object
favorite_count        2334 non-null float64
retweet_count         2334 non-null float64
dtypes: datetime64[ns](1), float64(5), int64(2), object(12)
memory usage: 385.1+ KB
```

In [666]:

```
twitter_archive.shape
```

Out[666]:

```
(2356, 17)
```

In [667]:

```
df_combined.duplicated().sum()
```

Out[667]:

```
0
```

In [668]:

```
twitter_archive_clean.head(3)
```

Out[668]:

	tweet_id	timestamp	text	rating_numerator	rating_denominator
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	13	10
1	892177421306343426	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you....	13	10
2	891815181378084864	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	12	10

In [669]:

```
twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2347 entries, 0 to 8151
Data columns (total 8 columns):
tweet_id              2347 non-null object
timestamp             2347 non-null datetime64[ns]
text                  2347 non-null object
rating_numerator      2347 non-null int64
rating_denominator    2347 non-null int64
name                  2347 non-null object
expanded_urls         2292 non-null object
dog_stage             394 non-null object
dtypes: datetime64[ns](1), int64(2), object(5)
memory usage: 165.0+ KB
```

In [670]:

```
predictions_clean.sample(3)
```

Out[670]:

	tweet_id	jpg_url	pred
507	676089483918516224	https://pbs.twimg.com/media/CWHzzFGXIAA0Y_H.jpg	bull_m
304	671518598289059840	https://pbs.twimg.com/media/CVG2l9jUYAAwg-w.jpg	lakelan
1667	812781120811126785	https://pbs.twimg.com/media/C0eUHfWUAAANEYr.jpg	bull_m

In [671]:

```
predictions_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1532 entries, 0 to 2073
Data columns (total 11 columns):
tweet_id        1532 non-null object
jpg_url         1532 non-null object
prediction_1    1532 non-null object
p1_conf         1532 non-null float64
p1_dog          1532 non-null bool
prediction_2    1532 non-null object
p2_conf         1532 non-null float64
p2_dog          1532 non-null bool
prediction_3    1532 non-null object
p3_conf         1532 non-null float64
p3_dog          1532 non-null bool
dtypes: bool(3), float64(3), object(5)
memory usage: 112.2+ KB
```

In [672]:

```
twitter_archive_clean.sample(3)
```

Out[672]:

	tweet_id	timestamp	text	rating_numerator	rating_denomi
2115	670428280563085312	2015-11-28 02:25:32	This is Willy. He's millennial af. 11/10 https...	11	10
457	818145370475810820	2017-01-08 17:20:31	This is Autumn. Her favorite toy is a cheesebu...	11	10
2064	671154572044468225	2015-11-30 02:31:34	Meet Holly. She's trying to teach small human-...	11	10

In [673]:

```
meta_data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2342 entries, 0 to 2341
Data columns (total 3 columns):
tweet_id          2342 non-null object
favorite_count    2342 non-null int64
retweet_count     2342 non-null int64
dtypes: int64(2), object(1)
memory usage: 55.0+ KB
```

In [674]:

```
meta_data_clean.sample(3)
```

Out[674]:

	tweet_id	favorite_count	retweet_count
342	831322785565769729	9709	1652
756	777641927919427584	0	4647
421	820837357901512704	0	7320

In [688]:

```
df_combined['retweet_count'].replace(np.NaN,0,inplace=True)
df_combined['favorite_count'].replace(np.NaN,0,inplace=True)
```


Storing, Analyzing and visualizing the wrangled data

Now that we have a combined data frame for all the data we could need for analysis we can proceed to do some visualizations as well as store the data on a convenient csv format

In [689]:

```
#Export to csv file
df_combined.to_csv('weratedogs_combined.csv')
meta_data_clean.to_csv('meta_data_clean.csv')
twitter_archive_clean.to_csv('twitter_archive_clean.csv')
predictions_clean.to_csv('predictions_clean.csv')
```

In [430]:

```
import seaborn as sns
```

In [481]:

```
df_combined.sample(3)
```

Out[481]:

	tweet_id	timestamp	text	rating_numerator	rating_denc
251	844979544864018432	2017-03-23 18:29:57	PUPDATE: I'm proud to announce that Toby is 23...	13	10
709	784183165795655680	2016-10-07 00:06:50	This is Reginald. He's one magical puppo. Aero...	12	10
2290	666776908487630848	2015-11-18 00:36:17	This is Josep. He is a Rye Manganese mix. Can ...	5	10

In [686]:

```
df_combined['retweet_count'].isnull().sum()
```

Out[686]:

13

In [687]:

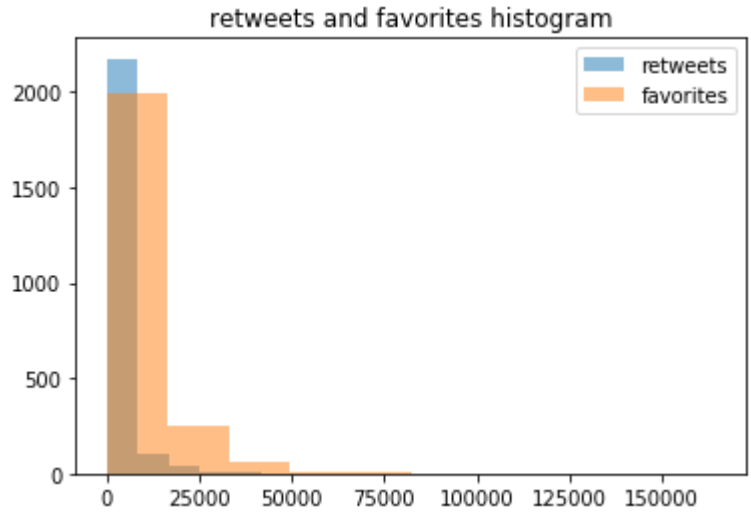
```
df_combined['favorite_count'].replace(np.NaN,0,inplace=True)
df_combined['retweet_count'].isnull().sum()
```

Out[687]:

13

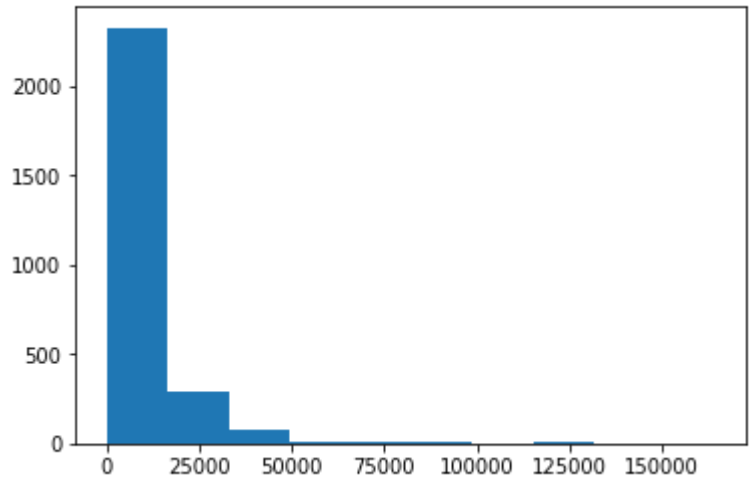
In [488]:

```
plt.hist(df_combined.retweet_count,alpha = 0.5, label='retweets')
plt.hist(df_combined.favorite_count,alpha = 0.5, label='favorites')
plt.title('retweets and favorites histogram')
plt.legend();
```



In [428]:

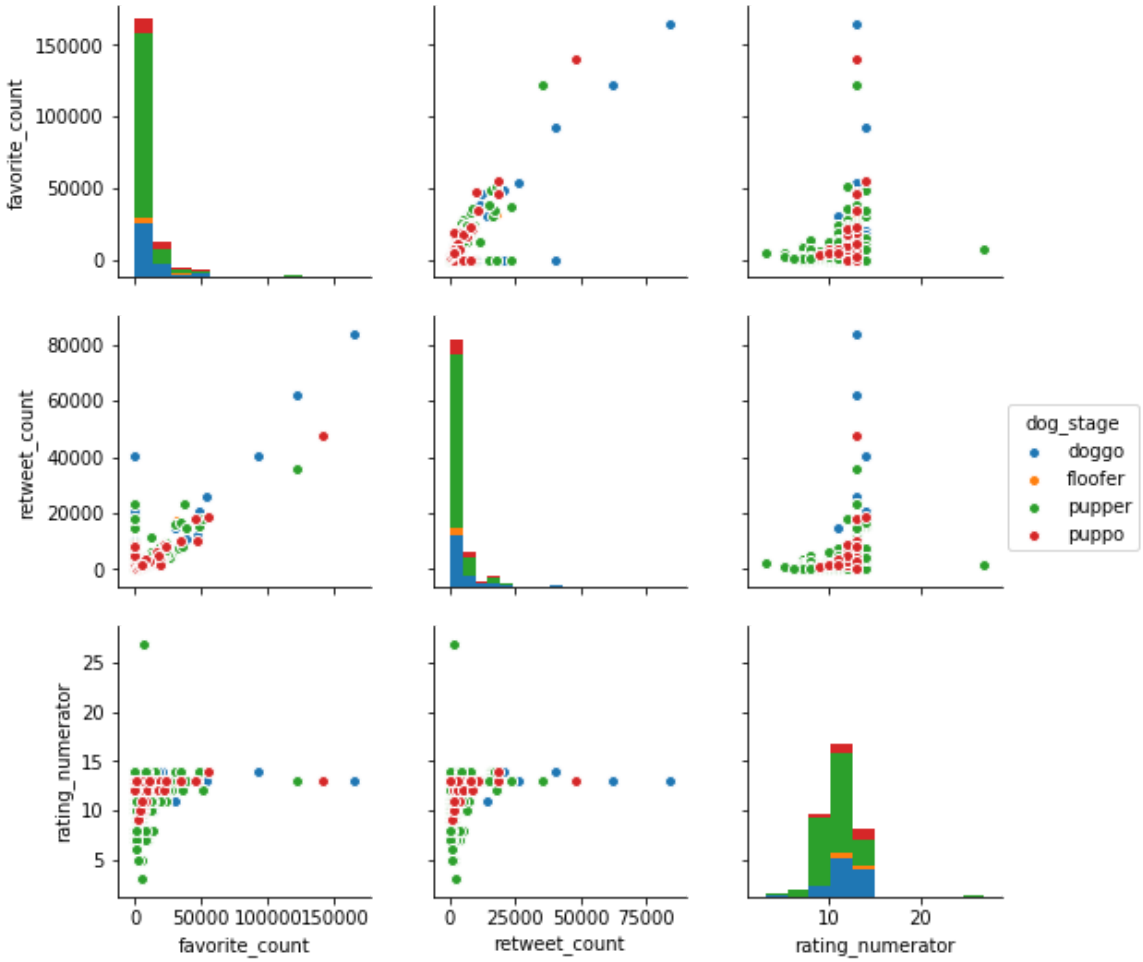
```
plt.hist(df_combined.favorite_count);
```



In [685]:

```
sns.pairplot(df_combined[['favorite_count','retweet_count','rating_numerator','dog_stage']],hue='dog_stage');
```

```
C:\Udacity\lib\site-packages\numpy\lib\function_base.py:780: RuntimeWarning: invalid value encountered in greater_equal
  keep = (tmp_a >= first_edge)
C:\Udacity\lib\site-packages\numpy\lib\function_base.py:781: RuntimeWarning: invalid value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```



Reporting on efforts and data analysis and visualizations

See other documents on github (rrosasl :))