wrangle_act

February 13, 2019

0.1 Gathering Data

In [1]:
```
# Packages

import tweepy
import pandas as pd
import numpy as np
import requests
import json
import matplotlib.pyplot as plt
import seaborn as sns
% matplotlib inline
```

In [2]:
```
#Gather data from csv file


archive = pd.read_csv('twitter-archive-enhanced.csv')

archive.head()
```

Out[2]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	<a href="http://twitter.com/download/iphone" r...
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	

```
3  <a href="http://twitter.com/download/iphone" r...
4  <a href="http://twitter.com/download/iphone" r...

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

   retweeted_status_user_id retweeted_status_timestamp  \
0                       NaN                        NaN
1                       NaN                        NaN
2                       NaN                        NaN
3                       NaN                        NaN
4                       NaN                        NaN

                                       expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/892420643...                13
1  https://twitter.com/dog_rates/status/892177421...                13
2  https://twitter.com/dog_rates/status/891815181...                12
3  https://twitter.com/dog_rates/status/891689557...                13
4  https://twitter.com/dog_rates/status/891327558...                12

   rating_denominator      name doggo floofer pupper puppo
0                  10   Phineas  None    None   None  None
1                  10     Tilly  None    None   None  None
2                  10    Archie  None    None   None  None
3                  10     Darla  None    None   None  None
4                  10  Franklin  None    None   None  None
```

In [3]:
```
# Download file from URL

URL= 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-prediction

with open('image.tsv' , 'wb') as f:
    image_f = requests.get(URL)
    f.write(image_f.content)

image = pd.read_csv('image.tsv', sep='\t')

image.head()
```

Out[3]:
```
             tweet_id                                          jpg_url  \
0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
1  666029285002620928  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
2  666033412701032449  https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
3  666044226329800704  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
```

```
4  666049248165822465  https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg
```

```
   img_num                     p1   p1_conf  p1_dog                 p2  \
0        1  Welsh_springer_spaniel  0.465074    True             collie
1        1                redbone  0.506826    True  miniature_pinscher
2        1        German_shepherd  0.596461    True            malinois
3        1    Rhodesian_ridgeback  0.408143    True             redbone
4        1     miniature_pinscher  0.560311    True          Rottweiler

    p2_conf  p2_dog                   p3   p3_conf  p3_dog
0  0.156665    True    Shetland_sheepdog  0.061428    True
1  0.074192    True  Rhodesian_ridgeback  0.072010    True
2  0.138584    True           bloodhound  0.116197    True
3  0.360687    True   miniature_pinscher  0.222752    True
4  0.243682    True             Doberman  0.154629    True
```

In [4]:
```
# Gather and store data from twitter API

consumer_key = '******'
consumer_secret = '**********'
access_token = '382817036-********'
access_secret = '*******'

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

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

#public_tweets = api.home_timeline()
#for tweet in public_tweets:
#        print(tweet.text)
#tweet = api.get_status(id_of_tweet)
#print(tweet.text)
```

In [5]:
```
#tweets= []
#deleted_tweets= []

#with open ('tweet_json.txt', 'w') as file:
#    for tweet_id in archive['tweet_id']:
#        try:
#            tweets.append(api.get_status(tweet_id, tweet_mode = 'extended')._json)
#        except Exception as e:
#            deleted_tweets.append(tweet_id)
#    file.write(json.dumps(tweets))


# Creating CSV file which have tweets thats not in api
```

```
        #deleted_tweets = pd.DataFrame(deleted_tweets)
        #deleted_tweets.to_csv('deleted_tweets.csv', sep = ',')
```

In [6]:
```
        #with open('tweet_json.txt') as jf:
        #    tweets_info = pd.DataFrame(columns = ['tweet_id',
        #                                          'favorites',
        #                                          'retweets'])
        #    for line in jf:
        #        tweet = json.loads(line)
        #        tweets_info = tweets_info.append({
        #            'tweet_id': tweet['id'],
        #            'favorites': tweet['favorite_count'],
        #            'retweets': tweet['retweet_count']
        #        }, ignore_index=True)
        #
        #tweets_info


        # Read Json file

        with open('tweet_json.txt','r') as json_file:
            tweets = json.loads(json_file.read())

        json_tweets = pd.DataFrame(tweets)
```

In [ ]:

In [ ]:

0.2 Assess

In this section I will explore the data to improve evaluation of the data.

In [7]:
```
archive.head()
```

Out[7]:

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

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

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

   retweeted_status_user_id retweeted_status_timestamp  \
0                       NaN                        NaN
1                       NaN                        NaN
2                       NaN                        NaN
3                       NaN                        NaN
4                       NaN                        NaN

                                       expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/892420643...                13
1  https://twitter.com/dog_rates/status/892177421...                13
2  https://twitter.com/dog_rates/status/891815181...                12
3  https://twitter.com/dog_rates/status/891689557...                13
4  https://twitter.com/dog_rates/status/891327558...                12

   rating_denominator      name doggo floofer pupper puppo
0                  10   Phineas  None    None   None  None
1                  10     Tilly  None    None   None  None
2                  10    Archie  None    None   None  None
3                  10     Darla  None    None   None  None
4                  10  Franklin  None    None   None  None
```

In [8]:
```
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 [9]: archive.describe()
```

Out[9]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	\
count	2.356000e+03	7.800000e+01	7.800000e+01	
mean	7.427716e+17	7.455079e+17	2.014171e+16	
std	6.856705e+16	7.582492e+17	1.252797e+17	
min	6.660209e+17	6.658147e+17	1.185634e+07	
25%	6.783989e+17	6.757419e+17	3.086374e+08	
50%	7.196279e+17	7.038708e+17	4.196984e+09	
75%	7.993373e+17	8.257804e+17	4.196984e+09	
max	8.924206e+17	8.862664e+17	8.405479e+17	

	retweeted_status_id	retweeted_status_user_id	rating_numerator	\
count	1.810000e+02	1.810000e+02	2356.000000	
mean	7.720400e+17	1.241698e+16	13.126486	
std	6.236928e+16	9.599254e+16	45.876648	
min	6.661041e+17	7.832140e+05	0.000000	
25%	7.186315e+17	4.196984e+09	10.000000	
50%	7.804657e+17	4.196984e+09	11.000000	
75%	8.203146e+17	4.196984e+09	12.000000	
max	8.874740e+17	7.874618e+17	1776.000000	

	rating_denominator
count	2356.000000
mean	10.455433
std	6.745237
min	0.000000
25%	10.000000
50%	10.000000
75%	10.000000
max	170.000000

```
In [10]: image.head()
```

Out[10]:

	tweet_id	jpg_url	\
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	

```
1  666029285002620928  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
2  666033412701032449  https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
3  666044226329800704  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
4  666049248165822465  https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg

   img_num                     p1   p1_conf  p1_dog                 p2  \
0        1  Welsh_springer_spaniel  0.465074    True             collie
1        1                redbone  0.506826    True  miniature_pinscher
2        1         German_shepherd  0.596461    True            malinois
3        1     Rhodesian_ridgeback  0.408143    True             redbone
4        1      miniature_pinscher  0.560311    True          Rottweiler

    p2_conf  p2_dog                    p3   p3_conf  p3_dog
0  0.156665    True    Shetland_sheepdog  0.061428    True
1  0.074192    True  Rhodesian_ridgeback  0.072010    True
2  0.138584    True           bloodhound  0.116197    True
3  0.360687    True   miniature_pinscher  0.222752    True
4  0.243682    True              Doberman  0.154629    True
```

In [11]:
```
image.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 [12]:
```
image.describe()
```

Out[12]:

	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 [13]:
```
json_tweets.head()
#pd.set_option('display.max_colwidth', -1)
```

Out[13]:

```
  contributors coordinates                       created_at display_text_range  \
0         None        None  Tue Aug 01 16:23:56 +0000 2017            [0, 85]
1         None        None  Tue Aug 01 00:17:27 +0000 2017           [0, 138]
2         None        None  Mon Jul 31 00:18:03 +0000 2017           [0, 121]
3         None        None  Sun Jul 30 15:58:51 +0000 2017            [0, 79]
4         None        None  Sat Jul 29 16:00:24 +0000 2017           [0, 138]

                                            entities  \
0  {'hashtags': [], 'symbols': [], 'user_mentions...
1  {'hashtags': [], 'symbols': [], 'user_mentions...
2  {'hashtags': [], 'symbols': [], 'user_mentions...
3  {'hashtags': [], 'symbols': [], 'user_mentions...
4  {'hashtags': [{'text': 'BarkWeek', 'indices': ...

                                     extended_entities  favorite_count  \
0  {'media': [{'id': 892420639486877696, 'id_str'...           37957
1  {'media': [{'id': 892177413194625024, 'id_str'...           32594
2  {'media': [{'id': 891815175371796480, 'id_str'...           24540
3  {'media': [{'id': 891689552724799489, 'id_str'...           41294
4  {'media': [{'id': 891327551943041024, 'id_str'...           39482

   favorited                                          full_text   geo  \
0      False  This is Phineas. He's a mystical boy. Only eve...  None
1      False  This is Tilly. She's just checking pup on you...  None
2      False  This is Archie. He is a rare Norwegian Pouncin...  None
3      False  This is Darla. She commenced a snooze mid meal...  None
4      False  This is Franklin. He would like you to stop ca...  None

                          ...                          quoted_status  \
0                         ...                                    NaN
1                         ...                                    NaN
2                         ...                                    NaN
3                         ...                                    NaN
4                         ...                                    NaN

  quoted_status_id quoted_status_id_str  quoted_status_permalink  \
0              NaN                  NaN                      NaN
1              NaN                  NaN                      NaN
2              NaN                  NaN                      NaN
3              NaN                  NaN                      NaN
4              NaN                  NaN                      NaN
```

```
   retweet_count  retweeted retweeted_status  \
0           8291      False              NaN
1           6123      False              NaN
2           4054      False              NaN
3           8430      False              NaN
4           9130      False              NaN

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

                                                user
0  {'id': 4196983835, 'id_str': '4196983835', 'na...
1  {'id': 4196983835, 'id_str': '4196983835', 'na...
2  {'id': 4196983835, 'id_str': '4196983835', 'na...
3  {'id': 4196983835, 'id_str': '4196983835', 'na...
4  {'id': 4196983835, 'id_str': '4196983835', 'na...

[5 rows x 32 columns]
```

```
In [14]: json_tweets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2340 entries, 0 to 2339
Data columns (total 32 columns):
contributors                    0 non-null object
coordinates                     0 non-null object
created_at                      2340 non-null object
display_text_range              2340 non-null object
entities                        2340 non-null object
extended_entities               2067 non-null object
favorite_count                  2340 non-null int64
favorited                       2340 non-null bool
full_text                       2340 non-null object
geo                             0 non-null object
id                              2340 non-null int64
id_str                          2340 non-null object
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 object
in_reply_to_user_id             77 non-null float64
in_reply_to_user_id_str         77 non-null object
is_quote_status                 2340 non-null bool
lang                            2340 non-null object
place                           1 non-null object
```

```
possibly_sensitive               2205 non-null object
possibly_sensitive_appealable    2205 non-null object
quoted_status                    24 non-null object
quoted_status_id                 26 non-null float64
quoted_status_id_str             26 non-null object
quoted_status_permalink          26 non-null object
retweet_count                    2340 non-null int64
retweeted                        2340 non-null bool
retweeted_status                 167 non-null object
source                           2340 non-null object
truncated                        2340 non-null bool
user                             2340 non-null object
dtypes: bool(4), float64(3), int64(3), object(22)
memory usage: 521.1+ KB
```

```
In [15]: json_tweets.describe()
```

Out[15]:

	favorite_count	id	in_reply_to_status_id	\
count	2340.000000	2.340000e+03	7.700000e+01	
mean	7943.589744	7.422176e+17	7.440692e+17	
std	12304.566122	6.832564e+16	7.524295e+16	
min	0.000000	6.660209e+17	6.658147e+17	
25%	1370.500000	6.783394e+17	6.757073e+17	
50%	3454.500000	7.186224e+17	7.032559e+17	
75%	9719.500000	7.986954e+17	8.233264e+17	
max	163903.000000	8.924206e+17	8.862664e+17	

	in_reply_to_user_id	quoted_status_id	retweet_count
count	7.700000e+01	2.600000e+01	2340.000000
mean	2.040329e+16	8.113972e+17	2919.271368
std	1.260797e+17	6.295843e+16	4918.221041
min	1.185634e+07	6.721083e+17	0.000000
25%	3.589728e+08	7.761338e+17	584.750000
50%	4.196984e+09	8.281173e+17	1362.000000
75%	4.196984e+09	8.637581e+17	3399.750000
max	8.405479e+17	8.860534e+17	83356.000000

```
In [16]: archive['name'].value_counts()
```

Out[16]:

None	745
a	55
Charlie	12
Lucy	11
Oliver	11
Cooper	11
Lola	10
Tucker	10
Penny	10

```
Winston       9
Bo            9
Sadie         8
the           8
Buddy         7
Bailey        7
Toby          7
an            7
Daisy         7
Jack          6
Oscar         6
Rusty         6
Milo          6
Koda          6
Jax           6
Bella         6
Dave          6
Scout         6
Leo           6
Stanley       6
Bentley       5
              ...
Vince         1
Mojo          1
Ralphy        1
Deacon        1
Jersey        1
Rumpole       1
Gin           1
old           1
Binky         1
Snoopy        1
Boston        1
Filup         1
Rizzo         1
Amber         1
Mona          1
Walker        1
Dixie         1
Stella        1
Striker       1
Doobert       1
Winifred      1
Jed           1
Bloo          1
Kloey         1
Keet          1
Newt          1
```

```
Gustaf        1
Siba          1
Hermione      1
Champ         1
Name: name, Length: 957, dtype: int64
```

```
In [17]: image['jpg_url'].value_counts()
         #image[image['jpg_url'] == 'https://pbs.twimg.com/media/CYLDikFWEAAIy1y.jpg']
         #test= archive.query('tweet_id == "761750502866649088"')
         #test
```

```
Out[17]: https://pbs.twimg.com/media/CkjMx99UoAM2B1a.jpg
         https://pbs.twimg.com/media/CVgdFjNWEAAxmbq.jpg
         https://pbs.twimg.com/media/ChK1tdBWwAQ1flD.jpg
         https://pbs.twimg.com/media/CsrjryzWgAAZY00.jpg
         https://pbs.twimg.com/media/Cs_DYr1XEAA54Pu.jpg
         https://pbs.twimg.com/media/CU3mITUWIAAfyQS.jpg
         https://pbs.twimg.com/media/CvyVxQRWEAAdSZS.jpg
         https://pbs.twimg.com/media/CdHwZd0VIAA4792.jpg
         https://pbs.twimg.com/media/CsGnz64WYAEIDHJ.jpg
         https://pbs.twimg.com/media/CU1zsMSUAAAS0qW.jpg
         https://pbs.twimg.com/media/CYLDikFWEAAIy1y.jpg
         https://pbs.twimg.com/media/CvoBPWRWgAA4het.jpg
         https://pbs.twimg.com/media/CwJR1okWIAA6XMp.jpg
         https://pbs.twimg.com/media/CvJCabcWgAIoUxW.jpg
         https://pbs.twimg.com/media/Ct72q9jWcAAhlnw.jpg
         https://pbs.twimg.com/media/Ct2qO5PXEAE6eB0.jpg
         https://pbs.twimg.com/media/Cbs3DOAXIAAp3Bd.jpg
         https://pbs.twimg.com/media/C2oRbOuWEAAbVSl.jpg
         https://pbs.twimg.com/media/CvaYgDOWgAEfjls.jpg
         https://pbs.twimg.com/ext_tw_video_thumb/807106774843039744/pu/img/8XZg1xW35Xp2J6JW.jpg
         https://pbs.twimg.com/ext_tw_video_thumb/817423809049493505/pu/img/5OFW0yueFu9oTUiQ.jpg
         https://pbs.twimg.com/media/CWza7kpWcAAdYLc.jpg
         https://pbs.twimg.com/media/C12whDoVEAALRxa.jpg
         https://pbs.twimg.com/media/Co-hmcYXYAASkiG.jpg
         https://pbs.twimg.com/media/CwiuEJmW8AAZnit.jpg
         https://pbs.twimg.com/media/CiyHLocU4AI2pJu.jpg
         https://pbs.twimg.com/media/CkNjahBXAAQ2kWo.jpg
         https://pbs.twimg.com/media/CxqsX-8XUAAEvjD.jpg
         https://pbs.twimg.com/media/CtzKC7zXEAALfSo.jpg
         https://pbs.twimg.com/media/Cwx99rpW8AMk_Ie.jpg

         https://pbs.twimg.com/media/Ch0LVPdW0AEdHgU.jpg
         https://pbs.twimg.com/media/ChKDKmIWIAIJP_e.jpg
         https://pbs.twimg.com/media/CU3FbQgVAAACdCQ.jpg
         https://pbs.twimg.com/media/CZBUO2UWsAAKehS.jpg
         https://pbs.twimg.com/media/CoeWSJcUIAAv3Bq.jpg
         https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg
```

```
https://pbs.twimg.com/media/Cu7dg2RXYAIaGXE.jpg
https://pbs.twimg.com/ext_tw_video_thumb/887517108413886465/pu/img/WanJKwssZj4VJvL9.jpg
https://pbs.twimg.com/media/CW7bkW6WQAAksgB.jpg
https://pbs.twimg.com/media/CdJnJ1dUEAARNcf.jpg
https://pbs.twimg.com/media/CdecUSzUIAAHCvg.jpg
https://pbs.twimg.com/media/CZWugJsWYAIzVzJ.jpg
https://pbs.twimg.com/media/CbSqEOrVIAEOPE4.jpg
https://pbs.twimg.com/media/CcFRCfRW4AA5a72.jpg
https://pbs.twimg.com/media/CfKYfeBXIAAopp2.jpg
https://pbs.twimg.com/media/CVCE9uYXIAEtSzR.jpg
https://pbs.twimg.com/media/CtkFS72WcAAiUrs.jpg
https://pbs.twimg.com/media/C4RCiIHWYAAwgJM.jpg
https://pbs.twimg.com/media/Cate3eLUcAEIuph.jpg
https://pbs.twimg.com/media/CuA-iRHXYAAWP8e.jpg
https://pbs.twimg.com/media/DAJfxqGVoAAnvQt.jpg
https://pbs.twimg.com/media/DFWra-3VYAA2piG.jpg
https://pbs.twimg.com/media/CZ1riVOWwAATfGf.jpg
https://pbs.twimg.com/media/CqWcgcqWcAI43jm.jpg
https://pbs.twimg.com/media/CX7EkuHWkAESLZk.jpg
https://pbs.twimg.com/media/CiWWhVNUYAAab_r.jpg
https://pbs.twimg.com/media/Cbn4OqKWwAADGWt.jpg
https://pbs.twimg.com/media/CUroc7QW4AATIff.jpg
https://pbs.twimg.com/media/CmieRQRXgAA8MV3.jpg
https://pbs.twimg.com/media/CyEg2AXUsAA1Qpf.jpg
Name: jpg_url, Length: 2009, dtype: int64
```

In [18]:
```
archive.tweet_id.duplicated().sum()
```

Out[18]:
```
0
```

In [19]:
```
image.tweet_id.duplicated().sum()
```

Out[19]:
```
0
```

In [20]:
```
json_tweets.id.duplicated().sum()
```

Out[20]:
```
0
```

In [21]:
```
#archive.text.value_counts()
tweet_text= archive[archive.text.str.contains('&amp;')]
tweet_text.text.value_counts()

#source for this point is here:https://github.com/kdow/WeRateDogs

#twitter_archive_clean['text'] = twitter_archive_clean['text'].str.replace('&amp;', '&'
```

Out[21]:
```
Meet Holly. She's trying to teach small human-like pup about blocks but he's not paying
Say hello to Bobb. Bobb is a Golden High Fescue &amp; a proud father of 8. Bobb sleeps
RT @dog_rates: This is Pipsy. He is a fluffball. Enjoys traveling the sea &amp; getting
```

Meet Jennifur. She's supposed to be navigating. Not even buckled up. Insubordinate &amp
Say hello to Penny & Gizmo. They are practicing their caroling. The ambition in the
Say hello to Kallie. There was a tornado in the area & the news guy said everyone s
Meet Travis and Flurp. Travis is pretty chill but Flurp can't lie down properly. 10/10
Meet Oliviér. He takes killer selfies. Has a dog of his own. It leaps at random & c
Meet Chester (bottom) & Harold (top). They are different dogs not only in appearanc
This is Timofy. He's a pilot for Southwest. It's Christmas morning & everyone has g
This is the best thing I've ever seen so spread it like wildfire & maybe we'll find
Meet Chesney. On the outside he stays calm & collected. On the inside he's having a
This is Lilli Bee & Honey Bear. Unfortunately, they were both born with no eyes. So
These are Peruvian Feldspars. Their names are Cupit and Prencer. Both resemble Rand Pau
Two gorgeous dogs here. Little waddling dog is a rebel. Refuses to look at camera. Must
& this is Yoshi. Another world record contender 11/10 (what the hell is happening w
This is Godzilla pupper. He had a ruff childhood & now deflects that pain outward b
Meet Jaycob. He got scared of the vacuum. Hide & seek champ. Almost better than Kon
Meet Roosevelt. He's preparing for takeoff. Make sure tray tables are in their full pup
This is Pipsy. He is a fluffball. Enjoys traveling the sea & getting tangled in lea
Meet Trooper & Maya. Trooper protects Maya from bad things like dognappers and Comc
Meet Jax & Jil. Jil is yelling the pledge of allegiance. If u cant take the freedom
Say hello to Gin & Tonic. They're having a staring contest. Very very intense. 9/10
This is Tedrick. He lives on the edge. Needs someone to hit the gas tho. Other than tha
RT @dog_rates: Meet Beau & Wilbur. Wilbur stole Beau's bed from him. Wilbur now has
This is Dook & Milo. Dook is struggling to find who he really is and Milo is terrif
Meet Maggie & Lila. Maggie is the doggo, Lila is the pupper. They are sisters. Both
When you try to recreate the scene from Lady & The Tramp but then remember you don'
Great picture here. Dog on the right panicked & forgot about his tongue. Middle gre
This is Lolo. She's America af. Behind in science & math but can say whatever she w
When bae says they can't go out but you see them with someone else that same night. 5/1
This is Sadie and her 2 pups Shebang & Ruffalo. Sadie says single parenting is chal
Meet Tassy & Bee. Tassy is pretty chill, but Bee is convinced the Ruffles are haunt
Say hello to Andy. He can balance on one foot, obliterate u in checkers, & transfor
From left to right:\nCletus, Jerome, Alejandro, Burp, & Titson\nNone know where cam
Meet Rambo & Kiwi. Rambo's the pup with the sharp toes & rad mohawk. One stays
These two dogs are Bo & Smittens. Smittens is trying out a new deodorant and wanted
Here we have Pancho and Peaches. Pancho is a Condoleezza Gryffindor, and Peaches is jus
Here we see a faulty pupper. Might need to replace batteries. Try turning off & bac
Meet Buckley. His family & some neighbors came over to watch him perform but he's n
Meet Daisy. She has no eyes & her face has been blurry since birth. Quite the troop
Meet Fynn & Taco. Fynn is an all-powerful leaf lord and Taco is in the wrong place
Say hello to Eugene & Patti Melt. No matter how dysfunctional they get, they will n
Meet Bruiser & Charlie. They are the best of pals. Been through it all together. Bo
Meet Beau & Wilbur. Wilbur stole Beau's bed from him. Wilbur now has so much room f
Meet Rufio. He is unaware of the pink legless pupper wrapped around him. Might want to
Touching scene here. Really stirs up the emotions. The bond between father & son. S
Meet Indie. She's not a fan of baths but she's definitely a fan of hide & seek. 12/
Say hello to Crimson. He's a Speckled Winnebago. Main passions are air hockey & par
Meet Sam. She smiles 24/7 & secretly aspires to be a reindeer. \nKeep Sam smiling b
This is Ben & Carson. It's impossible for them to tilt their heads in the same dire

```
Meet Jeb &amp;amp; Bush. Jeb is somehow stuck in that fence and Bush won't stop whispering
This is Spark. He's nervous. Other dog hasn't moved in a while. Won't come when called.
Meet Sid &amp;amp; Murphy. Murphy floats alongside Sid and whispers motivational quotes in
Name: text, dtype: int64
```

0.2.1 Quality

1- Missing values from images dataset which are 2075 rows and the archive are 2356.
2- Retweets need to be removed
3- Timestamp should be datetime instead of object (string)
4- There are same jpg_url for more than one tweet_ids
5- Incorrect dog names, The most popular name is 'a' which is not corrected name.
6- Drop some columns which is not usefull for analysis.
7- Some columns need to rename it to understandable name.
8- Nulls represented as 'None' in columns 'name', 'doggo', 'floofer', 'pupper','puppo'.
9- Some tweets have additional characters in the text which is not meaningful.
10- The numerator and denominator columns have invalid values "this is mentioned in Project Motivation but I cannot clean it because it will take time and I need to submit this project as soon as possible"

0.2.2 Tidiness

1- Dog stage variable in different columns: doggo, floofer, pupper, puppo
2- Marge 'json_tweets' and 'image' to 'archive' dataset

0.3 Cleaning

In [22]:
```
# make copies of datasets

archive_clean = archive.copy()
image_clean = image.copy()
json_tweets_clean = json_tweets.copy()

#json_tweets_clean.head(2)
```

Issue:

```
Some of tweets haven same jpg_url images
```

Define:

```
Delete the duplicated jpg_url images
```

In [23]:
```
#image_clean['jpg_url'].value_counts()


image_clean['jpg_url'] = image_clean['jpg_url'].drop_duplicates()

image_clean = image_clean[pd.notnull(image_clean['jpg_url'])]
```

```
In [24]: # test

         image_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2009 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2009 non-null int64
jpg_url     2009 non-null object
img_num     2009 non-null int64
p1          2009 non-null object
p1_conf     2009 non-null float64
p1_dog      2009 non-null bool
p2          2009 non-null object
p2_conf     2009 non-null float64
p2_dog      2009 non-null bool
p3          2009 non-null object
p3_conf     2009 non-null float64
p3_dog      2009 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 162.8+ KB
```

Issue:

Some of these tweets haven't images, we want only the tweets with image

Define:

Delete the tweets which is haven't image from the dataset

```
In [25]: # Delete tweets without Image


         #test= archive_clean.query('tweet_id == "685325112850124800"')
         #test

         image_id=image_clean[['tweet_id']]

         #image_id

         archive_clean=pd.merge(archive_clean,image_id,on='tweet_id')
```

```
In [26]: # For test

         archive_clean.info()
         #archive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2009 entries, 0 to 2008
Data columns (total 17 columns):
tweet_id                      2009 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     2009 non-null object
source                        2009 non-null object
text                          2009 non-null object
retweeted_status_id           15 non-null float64
retweeted_status_user_id      15 non-null float64
retweeted_status_timestamp    15 non-null object
expanded_urls                 2009 non-null object
rating_numerator              2009 non-null int64
rating_denominator            2009 non-null int64
name                          2009 non-null object
doggo                         2009 non-null object
floofer                       2009 non-null object
pupper                        2009 non-null object
puppo                         2009 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 282.5+ KB
```

Issue:

```
we want only the origen tweets not retweets
```

Define:

```
Delete  retweets from the dataset
```

```
In [27]: #Delete retweets

         archive_clean = archive_clean[pd.isnull(archive_clean['retweeted_status_id'])]
```

```
In [28]: # test

         archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1994 entries, 0 to 2008
Data columns (total 17 columns):
tweet_id                      1994 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     1994 non-null object
source                        1994 non-null object
text                          1994 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                 1994 non-null object
rating_numerator              1994 non-null int64
rating_denominator            1994 non-null int64
name                          1994 non-null object
doggo                         1994 non-null object
floofer                       1994 non-null object
pupper                        1994 non-null object
puppo                         1994 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 280.4+ KB
```

Issue:

```
Datatype of timestamp
```

Define:

```
Convert timestamp to datetime data type.
```

```
In [29]: #Convert timestamp

         archive_clean['timestamp'] =pd.to_datetime(archive_clean['timestamp'])
```

```
In [30]: # For test

         archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1994 entries, 0 to 2008
Data columns (total 17 columns):
tweet_id                      1994 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     1994 non-null datetime64[ns]
source                        1994 non-null object
text                          1994 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                 1994 non-null object
rating_numerator              1994 non-null int64
rating_denominator            1994 non-null int64
name                          1994 non-null object
doggo                         1994 non-null object
floofer                       1994 non-null object
```

```
pupper                        1994 non-null object
puppo                         1994 non-null object
dtypes: datetime64[ns](1), float64(4), int64(3), object(9)
memory usage: 280.4+ KB
```

Issue:

Some columns not usefull in our datasets.

Define:

Delete these columns which is not usefull for analysis.

```
In [31]: # Delete some columns

         #json_tweets_clean.info()
         json_tweets_clean = json_tweets_clean.drop(json_tweets_clean.columns[[0, 1,2, 3,4,5,7,8
```

```
In [32]: # For test

         json_tweets_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2340 entries, 0 to 2339
Data columns (total 3 columns):
favorite_count    2340 non-null int64
id                2340 non-null int64
retweet_count     2340 non-null int64
dtypes: int64(3)
memory usage: 54.9 KB
```

Issue:

Some columns have difficult understandable name.

Define:

rename the columns to understandable name.

```
In [33]: # rename the columns

         json_tweets_clean.rename(columns={'favorite_count': 'favorite', 'id': 'tweet_id', 'retw

         image_clean.rename(columns={'p1': 'prediction', 'p1_conf': 'confidence', 'p1_dog': 'Dog
                                     'p2': 'prediction2', 'p2_conf': 'confidence2', 'p2_dog': 'p
                                     'p3': 'prediction3', 'p3_conf': 'confidence3', 'p3_dog': 'p
```

```
In [34]: list(image_clean)
         list(json_tweets_clean)
```

```
Out[34]: ['favorite', 'tweet_id', 'retweet']

In [35]: # Delete some columns

         #archive_clean.info()
         archive_clean = archive_clean.drop(archive_clean.columns[[6,7,8]], axis=1)

In [36]: # For test

         archive_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1994 entries, 0 to 2008
Data columns (total 14 columns):
tweet_id                 1994 non-null int64
in_reply_to_status_id    23 non-null float64
in_reply_to_user_id      23 non-null float64
timestamp                1994 non-null datetime64[ns]
source                   1994 non-null object
text                     1994 non-null object
expanded_urls            1994 non-null object
rating_numerator         1994 non-null int64
rating_denominator       1994 non-null int64
name                     1994 non-null object
doggo                    1994 non-null object
floofer                  1994 non-null object
pupper                   1994 non-null object
puppo                    1994 non-null object
dtypes: datetime64[ns](1), float64(2), int64(3), object(8)
memory usage: 233.7+ KB
```

Define:

Marge the cleaned datasets together

```
In [37]: tweet_master = pd.merge(archive_clean, image_clean, how = 'left', on = ['tweet_id'] )

         tweets_master = pd.merge(tweet_master, json_tweets_clean, how = 'left', on = ['tweet_id

         #tweets_master.info()

         tweets_master.to_csv('tweets_master.csv')

In [38]: #test

         tweets_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1994 entries, 0 to 1993
Data columns (total 27 columns):
tweet_id                1994 non-null int64
in_reply_to_status_id   23 non-null float64
in_reply_to_user_id     23 non-null float64
timestamp               1994 non-null datetime64[ns]
source                  1994 non-null object
text                    1994 non-null object
expanded_urls           1994 non-null object
rating_numerator        1994 non-null int64
rating_denominator      1994 non-null int64
name                    1994 non-null object
doggo                   1994 non-null object
floofer                 1994 non-null object
pupper                  1994 non-null object
puppo                   1994 non-null object
jpg_url                 1994 non-null object
img_num                 1994 non-null int64
prediction              1994 non-null object
confidence              1994 non-null float64
Dog?                    1994 non-null bool
prediction2             1994 non-null object
confidence2             1994 non-null float64
p2_Dog?                 1994 non-null bool
prediction3             1994 non-null object
confidence3             1994 non-null float64
p3_Dog?                 1994 non-null bool
favorite                1992 non-null float64
retweet                 1992 non-null float64
dtypes: bool(3), datetime64[ns](1), float64(7), int64(4), object(12)
memory usage: 395.3+ KB
```

Define:

Melt columnes ['doggo', 'floofer', 'pupper', 'puppo'] under stage column.

```
In [39]: idv = [x for x in list(tweets_master.columns) if x not in ['doggo', 'floofer', 'pupper'

         tweets_master = pd.melt(tweets_master, id_vars = idv , value_vars = ['doggo', 'floofer'
                                 var_name='stages', value_name = 'stage')

         tweets_master = tweets_master.drop('stages', 1)

         tweets_master = tweets_master.sort_values('stage').drop_duplicates('tweet_id', keep = '
```

```
In [40]: # for test
```

```
#tweets_master.info()

tweets_master.stage.value_counts()
```

```
Out[40]: None       1688
         pupper      212
         doggo        63
         puppo        23
         floofer       8
         Name: stage, dtype: int64
```

```
In [41]: tweets_master.info()
         tweets_master.head(2)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1994 entries, 1918 to 6764
Data columns (total 24 columns):
tweet_id                 1994 non-null int64
in_reply_to_status_id    23 non-null float64
in_reply_to_user_id      23 non-null float64
timestamp                1994 non-null datetime64[ns]
source                   1994 non-null object
text                     1994 non-null object
expanded_urls            1994 non-null object
rating_numerator         1994 non-null int64
rating_denominator       1994 non-null int64
name                     1994 non-null object
jpg_url                  1994 non-null object
img_num                  1994 non-null int64
prediction               1994 non-null object
confidence               1994 non-null float64
Dog?                     1994 non-null bool
prediction2              1994 non-null object
confidence2              1994 non-null float64
p2_Dog?                  1994 non-null bool
prediction3              1994 non-null object
confidence3              1994 non-null float64
p3_Dog?                  1994 non-null bool
favorite                 1992 non-null float64
retweet                  1992 non-null float64
stage                    1994 non-null object
dtypes: bool(3), datetime64[ns](1), float64(7), int64(4), object(9)
memory usage: 348.6+ KB
```

```
Out[41]:                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         1918  667405339315146752                    NaN                  NaN
         1917  667435689202614272                    NaN                  NaN
```

```
                 timestamp                                              source  \
1918 2015-11-19 18:13:27  <a href="http://twitter.com/download/iphone" r...
1917 2015-11-19 20:14:03  <a href="http://twitter.com/download/iphone" r...

                                                  text  \
1918  This is Biden. Biden just tripped... 7/10 http...
1917            Ermergerd 12/10 https://t.co/PQni2sjPsm

                                          expanded_urls  rating_numerator  \
1918  https://twitter.com/dog_rates/status/667405339...                 7
1917  https://twitter.com/dog_rates/status/667435689...                12

      rating_denominator   name  ...    Dog?         prediction2 confidence2  \
1918                  10  Biden  ...    True            Leonberg    0.127998
1917                  10   None  ...    True  miniature_pinscher    0.000450

      p2_Dog?              prediction3 confidence3  p3_Dog?  favorite retweet  \
1918     True         golden_retriever    0.069357     True     467.0   221.0
1917     True  black-and-tan_coonhound    0.000157     True     305.0    84.0

      stage
1918   None
1917   None

[2 rows x 24 columns]
```

Issue:

Some dogs have incorrect name.

Define:

replace incorrect name to Nan.

```
In [42]: #Replace incorrect name to Nan.

         #tweets_master['name'].value_counts()

         tweets_master['name']= tweets_master['name'].replace(['a', 'an', 'the'], np.nan)
```

```
In [43]: #test

         #tweets_master['name'].value_counts()

         testname= tweets_master.query('name == "a"')
         testname
```

```
Out[43]: Empty DataFrame
         Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, text
```

```
Index: []

[0 rows x 24 columns]
```

Issue:

Nulls represented as 'None'.

Define:

replace None to Nan in the datasets.

```
In [44]: # replace None

         tweets_master['name']= tweets_master['name'].replace(['None'], np.nan)
         tweets_master['stage']= tweets_master['stage'].replace(['None'], np.nan)

In [45]: # test
         #tweets_master['name'].value_counts()


         #t1 = tweets_master.query('name == "None"')
         #t2 = tweets_master.query('stage == "None"')
         tweets_master.stage.value_counts()


         #print (t1, t2)

Out[45]: pupper     212
         doggo       63
         puppo       23
         floofer      8
         Name: stage, dtype: int64
```

Issue:

Additional characters in tweet text .

Define:

Remove the Additional characters in tweet text.

```
In [46]: tweets_master['text'] = tweets_master['text'].str.replace('&amp;', '&')

In [47]: #FOr test


         tweet_text= tweets_master[tweets_master.text.str.contains('&amp;')]
         tweet_text.text.value_counts()
```

```
Out[47]: Series([], Name: text, dtype: int64)
```

Define:

Reorder the columns and drop the additional columns to make the dataset easy to read.

```
In [48]: #Reorder the columns

         #list(tweets_master.columns.values)


         tweets_master = tweets_master[['tweet_id','text','retweet','favorite','name','stage','r
                                        'timestamp','prediction','confidence','Dog?','jpg_url','
                                        'img_num','prediction2','confidence2','p2_Dog?','predict
```

```
In [49]: #drop the additional columns


         tweets_master = tweets_master.drop(tweets_master.columns[[14,15,17,18,19,20,21,22,23]],
```

```
In [50]: #for test

         tweets_master.tail(2)
```

```
Out[50]:                tweet_id                                               text  \
         6279  825535076884762624  Here's a very loving and accepting puppo. Appe...
         6764  743253157753532416  This is Kilo. He cannot reach the snackum. Nif...

               retweet  favorite  name  stage  rating_numerator  rating_denominator  \
         6279  18593.0   55008.0   NaN  puppo                14                  10
         6764   1299.0    4421.0  Kilo  puppo                10                  10

                        timestamp  prediction  confidence  Dog?  \
         6279 2017-01-29 02:44:34  Rottweiler    0.681495  True
         6764 2016-06-16 01:25:36    malamute    0.442612  True

                                                    jpg_url  \
         6279  https://pbs.twimg.com/media/C3TjvitXAAAI-QH.jpg
         6764  https://pbs.twimg.com/media/ClCQzFUUYAA5vAu.jpg

                                                      source  \
         6279  <a href="http://twitter.com/download/iphone" r...
         6764  <a href="http://twitter.com/download/iphone" r...

                                                 expanded_urls
         6279  https://twitter.com/dog_rates/status/825535076...
         6764  https://twitter.com/dog_rates/status/743253157...
```

0.3.1 Storing

Storing cleaned master dataset:

```
In [51]: #Storing


         tweets_master.to_csv('twitter_archive_master.csv')
```

0.3.2 Analyzing and Visualizing Data

```
In [52]: # make copy for analyzing

         tweets_data= pd.read_csv('twitter_archive_master.csv')

         tweets_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1994 entries, 0 to 1993
Data columns (total 16 columns):
Unnamed: 0            1994 non-null int64
tweet_id              1994 non-null int64
text                  1994 non-null object
retweet               1992 non-null float64
favorite              1992 non-null float64
name                  1380 non-null object
stage                 306 non-null object
rating_numerator      1994 non-null int64
rating_denominator    1994 non-null int64
timestamp             1994 non-null object
prediction            1994 non-null object
confidence            1994 non-null float64
Dog?                  1994 non-null bool
jpg_url               1994 non-null object
source                1994 non-null object
expanded_urls         1994 non-null object
dtypes: bool(1), float64(3), int64(4), object(8)
memory usage: 235.7+ KB
```

0.3.3 Famous and lovely Dogs at WeRateDogs

- Let's see who are the famous dogs in WeRateDogs which have a top retweet.
- Also we will see who are the lovely dogs that the pepole like it by display dogs favorite.

Top 10 of Famous Dogs:

In [59]:
```
#top_dog_retweets= tweets_data.groupby('prediction')["retweet"].sum().sort_values(ascen

#top_dog_retweets.head()

top_dog_retweets = tweets_data.sort_values(by='retweet',ascending =False ).head(10)

top_dog_retweets[['name','stage','retweet','text','jpg_url']]
#pd.reset_option('display.max_rows')

#pd.set_option('display.max_colwidth', -1)
```

Out[59]:

	name	stage	retweet	text	jpg_url
1718	NaN	doggo	83356.0	Here's a doggo realizing you can stand in a pool. 13/10 enlightened af (vid by Ti	https://pbs.twimg.com/ext_tw_video_thumb/744234667679821824/pu/img/1GaWmtJtdqzZV7
1728	NaN	doggo	61721.0	Here's a doggo blowing bubbles. It's downright legendary. 13/10 would watch on re	https://pbs.twimg.com/ext_tw_video_thumb/739238016737267712/pu/img/-tLpyiuIzD5zR1
1037	Stephan	NaN	60783.0	This is Stephan. He just wants to help. 13/10 such a good boy https://t.co/DkBYaC	https://pbs.twimg.com/ext_tw_video_thumb/807106774843039744/pu/img/8XZg1xW35Xp2J6
1975	NaN	puppo	47539.0	Here's a super supportive puppo participating in the Toronto #WomensMarch today.	https://pbs.twimg.com/media/C2tugXLXgAArJO4.jpg
166	Duddles	NaN	43308.0	This is Duddles. He did an attempt. 13/10 someone help him (vid by Georgia Felici	https://pbs.twimg.com/ext_tw_video_thumb/879415784908390401/pu/img/cX7XI1TnUsseGE
1717	Bo	doggo	39937.0	This is Bo. He was a very good First Doggo. 14/10 would be an absolute honor to p	https://pbs.twimg.com/media/C12whDoVEAALRxa.jpg
1036	NaN	NaN	38112.0	"Good afternoon class today we're going to learn what makes a good boy so good" 1	https://pbs.twimg.com/media/CzG425nWgAAnP7P.jpg
1867	Jamesy	pupper	35323.0	This is Jamesy. He gives a kiss to every other pupper he sees on his walk. 13/10	https://pbs.twimg.com/media/DAZAUfBXcAAG_Nn.jpg
917	NaN	NaN	33667.0	This made my day. 12/10 please enjoy https://t.co/VRTbo3aAcm	https://pbs.twimg.com/ext_tw_video_thumb/678399528077250560/pu/img/BOjUNHRsYLeSoC
868	Kenneth	NaN	32499.0	This is Kenneth. He's stuck in a bubble. 10/10 hang in there Kenneth https://t.co	https://pbs.twimg.com/media/CWJqN9iWwAAg86R.jpg

In [55]:
```
from PIL import Image
import requests
```

```
from io import BytesIO


response = requests.get('https://pbs.twimg.com/ext_tw_video_thumb/744234667679821824/pu
img = Image.open(BytesIO(response.content))

img


#sources "https://stackoverflow.com/questions/7391945/how-do-i-read-image-data-from-a-u
```

Out[55]:

Hi there, I am the top one Famous Dog in WeRateDogs, I have more than 83K retweet, my stage is doggo, I will not tall you my name right now

Top 10 of Lovely Dogs:

```
In [62]: top_dog_favorite = tweets_data.sort_values(by='favorite',ascending =False ).head(10)

         top_dog_favorite[['name','stage','retweet','favorite','text','jpg_url']]
```

Out[62]:

	name	stage	retweet	favorite	\
1718	NaN	doggo	83356.0	163903.0	
1975	NaN	puppo	47539.0	140063.0	
1037	Stephan	NaN	60783.0	126742.0	
1867	Jamesy	pupper	35323.0	121681.0	
1728	NaN	doggo	61721.0	121088.0	
166	Duddles	NaN	43308.0	103786.0	
1717	Bo	doggo	39937.0	91971.0	
1309	quite	NaN	30663.0	90490.0	
917	NaN	NaN	33667.0	82235.0	
1294	Zoey	NaN	26014.0	81769.0	

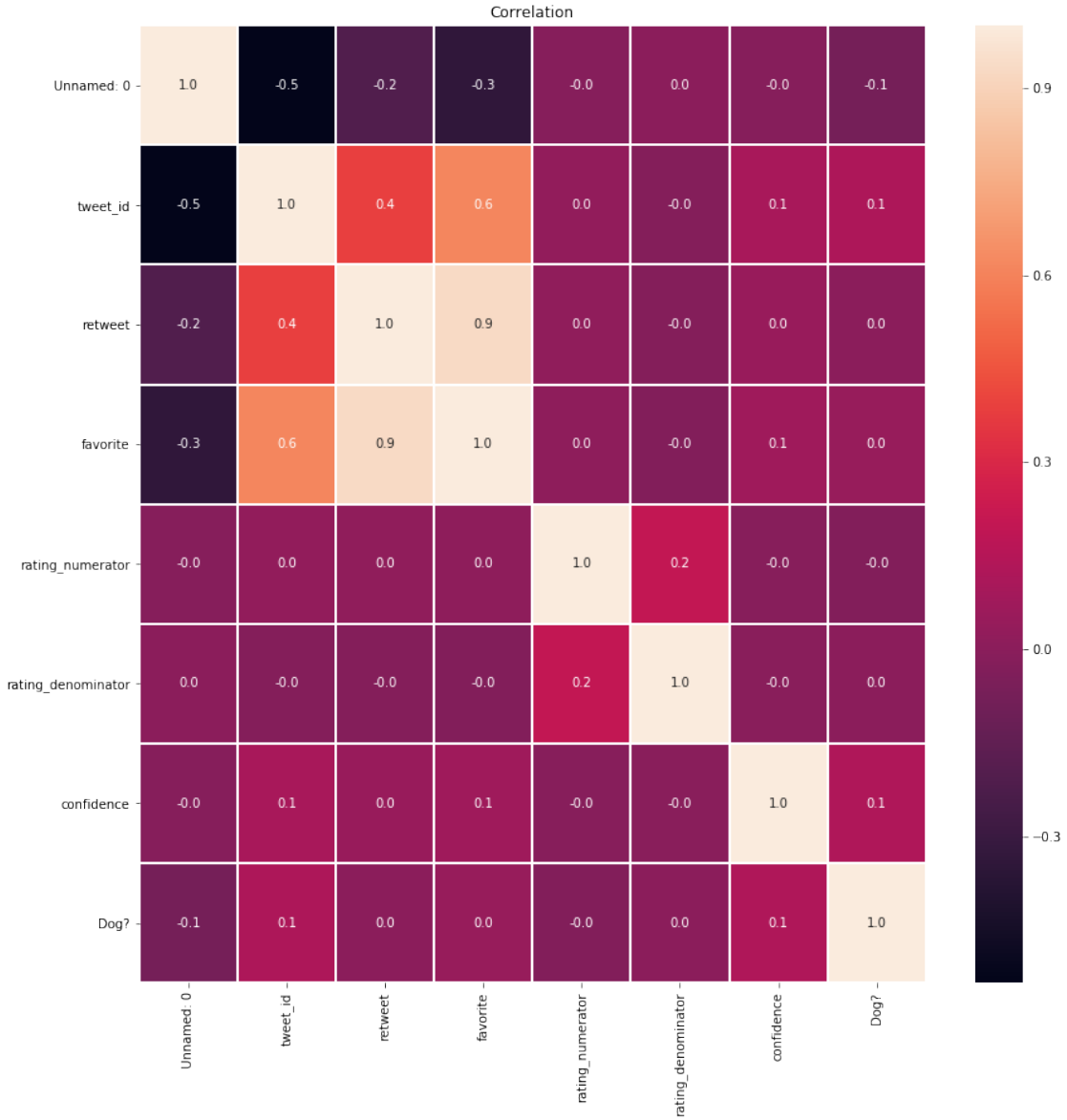
1718	Here's a doggo realizing you can stand in a pool. 13/10 enlightened af (vid by Ti
1975	Here's a super supportive puppo participating in the Toronto #WomensMarch today.
1037	This is Stephan. He just wants to help. 13/10 such a good boy https://t.co/DkBYaC
1867	This is Jamesy. He gives a kiss to every other pupper he sees on his walk. 13/10
1728	Here's a doggo blowing bubbles. It's downright legendary. 13/10 would watch on re
166	This is Duddles. He did an attempt. 13/10 someone help him (vid by Georgia Felici
1717	This is Bo. He was a very good First Doggo. 14/10 would be an absolute honor to p
1309	We only rate dogs. This is quite clearly a smol broken polar bear. We'd appreciat
917	This made my day. 12/10 please enjoy https://t.co/VRTbo3aAcm
1294	This is Zoey. She really likes the planet. Would hate to see willful ignorance an

1718	https://pbs.twimg.com/ext_tw_video_thumb/744234667679821824/pu/img/1GaWmtJtdqzZV7
1975	https://pbs.twimg.com/media/C2tugXLXgAArJO4.jpg
1037	https://pbs.twimg.com/ext_tw_video_thumb/807106774843039744/pu/img/8XZg1xW35Xp2J6
1867	https://pbs.twimg.com/media/DAZAUfBXcAAG_Nn.jpg
1728	https://pbs.twimg.com/ext_tw_video_thumb/739238016737267712/pu/img/-tLpyiuIzD5zR1
166	https://pbs.twimg.com/ext_tw_video_thumb/879415784908390401/pu/img/cX7XI1TnUsseGE
1717	https://pbs.twimg.com/media/C12whDoVEAALRxa.jpg
1309	https://pbs.twimg.com/ext_tw_video_thumb/859196962498805762/pu/img/-yBpr4-o4GJZEC
917	https://pbs.twimg.com/ext_tw_video_thumb/678399528077250560/pu/img/BOjUNHRsYLeSoC
1294	https://pbs.twimg.com/media/DBQwlFCXkAACSkI.jpg

As we see from the list above, The top one retweet dog have top favorite also, that's mean there is correlation between retweets and favorites.

The relationship between the variables

```
In [67]: #Correlation map
         f,ax = plt.subplots(figsize=(14, 14))
         sns.heatmap(tweets_data.corr(), annot=True, linewidths=1, fmt= '.1f',ax=ax)
         plt.title('Correlation')
         plt.show();
```



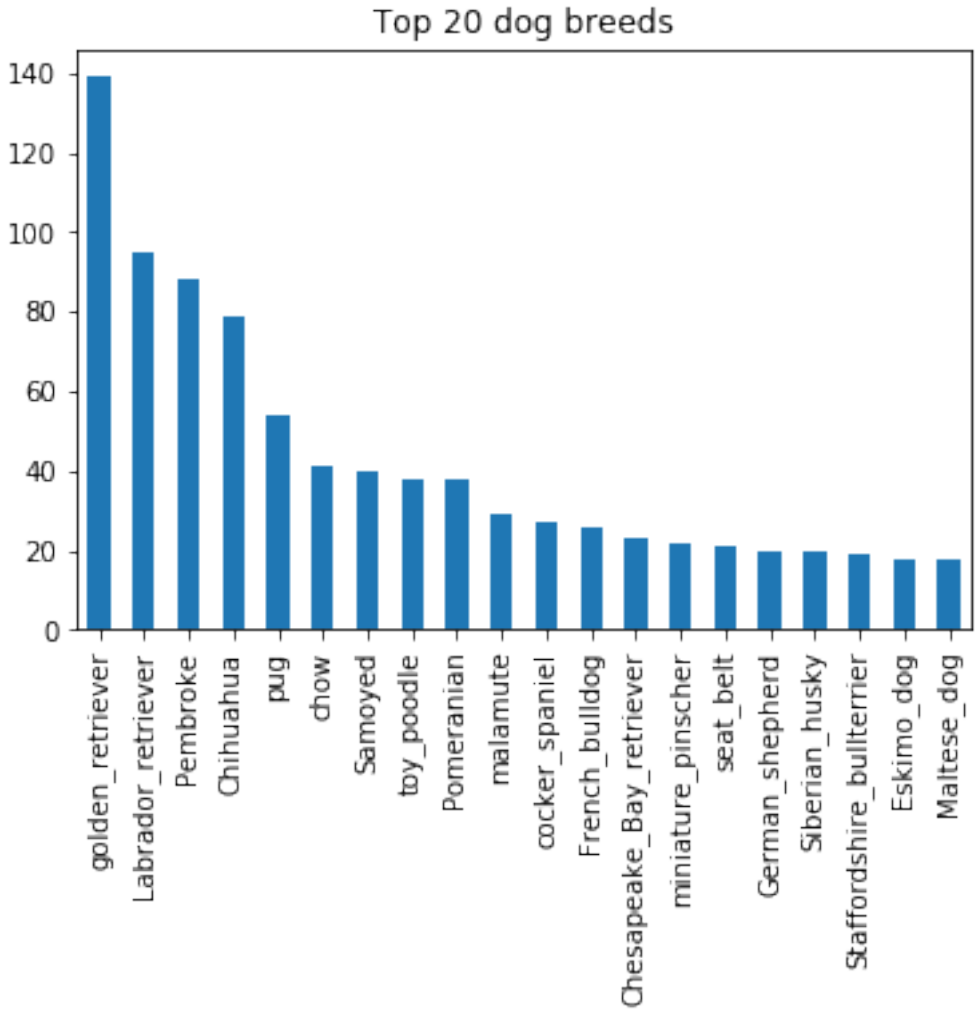
From the correlation map there is a strong correlation between favorites and retweets.

0.3.4 Top dog breeds in the tweets based on prediction data:

```
In [79]: #Top dog breeds in the tweets based on prediction data
```

```
top_dog_breeds= tweets_data['prediction'].value_counts().head(20)

top_dog_breeds.plot(kind='bar', title='Top 20 dog breeds')
plt.show();
```



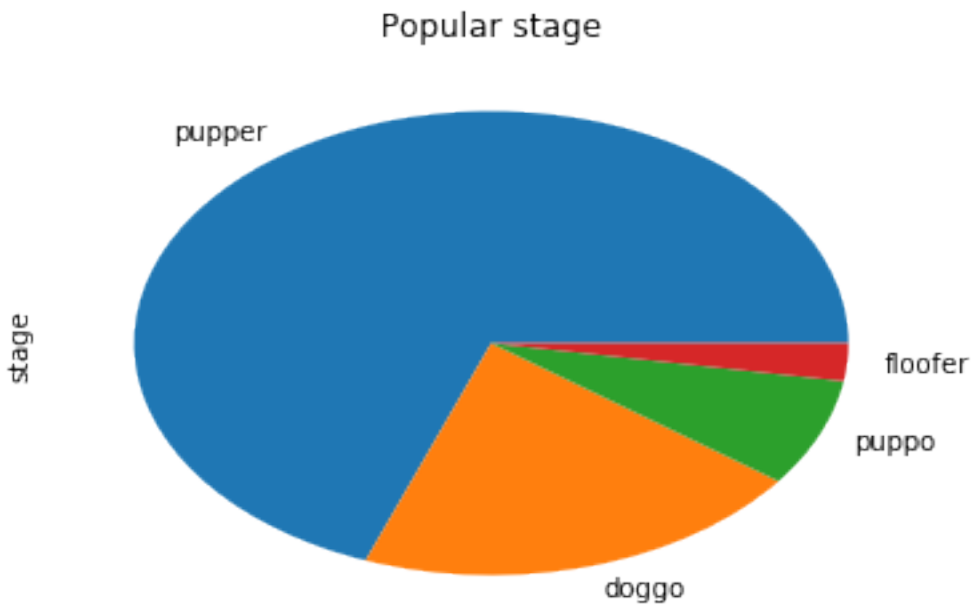
The top one on Dog breeds is golden_retriever then Labrador_retriever

0.3.5 Popular stage on the tweets

In [81]:
```
#Popular stage

dog_stage= tweets_data['stage'].value_counts()

dog_stage.plot(kind='pie', title='Popular stage')
plt.show();
```



Pupper represent the big number of the pie

```
In [82]: from subprocess import call
         call(['python','-m','nbconvert','wrangle_act.ipynb'])

Out[82]: 0

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```