wrangle_act

December 12, 2018

1 WeRateDogs Project- Wrangling & Analyzing Twitter Data

- Philipp Keupp
- Dezember 2018

1.1 Introduction

The goal of this project is to wrangle the WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The challenge lies in the fact that the Twitter archive is great, but it only contains very basic tweet information that comes in JSON format. For a successful project, I needed to gather, asses and clean the Twitter data for a worthy analysis and visualization.

In [2]:
```
# import main libraries

import pandas as pd
import numpy as np
import os
import requests as rq
import json
import time
import tweepy
```

1.2 Gathering Data

In [3]:
```
# load data
data_twitter = pd.read_csv('twitter-archive-enhanced.csv')
data_twitter.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 [4]: data_twitter.head(2)
```

Out[4]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweeted_status_user_id	retweeted_status_timestamp	expanded_urls	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892420643...	13	10	Phineas	None	None	None	None
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892177421...	13	10	Tilly	None	None	None	None

```
In [15]: # Download tsv file
         folder_name = 'images_prediction'
```

```
        # Make directory if it doesn't already exist
        if not os.path.exists(folder_name):
            os.makedirs(folder_name)

        url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predict
        r = rq.get(url)
        r

        with open(os.path.join(folder_name,
                               url.split('/')[-1]), mode='wb') as file:
            file.write(r.content)
```

In [5]:
```
        #open tsv file
        images = pd.read_table('images_prediction/image-predictions.tsv',
                               sep='\t')
```

In [ ]:
```
        #

        consumer_key = ''
        consumer_secret = ''
        access_token = ''
        access_secret = ''

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

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

In [ ]:
```
        import json
        tweet_ids = data_twitter.tweet_id.values

        with open('tweet_json.txt', 'a', encoding='utf8') as outfile:
            for tweet_id in tweet_ids:
                try:
                    tweet = api.get_status(tweet_id, tweet_mode='extended')
                    json.dump(tweet._json, outfile)
                    outfile.write('\n')
                except Exception as e:
                    print(tweet_id,e)
```

In [6]:
```
        data_list = []
        data_dict = {
                        'tweet_id': '',
                        'favorite_count': '',
                        'retweet_count': '',
                    }
        with open('tweet_json.txt') as json_file:
```

```
        for line in json_file:
            data = json.loads(line)
            data_dict['tweet_id'] = data['id']
            data_dict['favorite_count'] = data['favorite_count']
            data_dict['retweet_count'] = data['retweet_count']
            data_list.append(data_dict.copy())
    favorite_retweet_table = pd.DataFrame(data_list)
```

2 Assessing Data

```
In [7]: data_twitter.head(10)
```

Out[7]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	\
0	892420643555336193	NaN	NaN	
1	892177421306343426	NaN	NaN	
2	891815181378084864	NaN	NaN	
3	891689557279858688	NaN	NaN	
4	891327558926688256	NaN	NaN	
5	891087950875897856	NaN	NaN	
6	890971913173991426	NaN	NaN	
7	890729181411237888	NaN	NaN	
8	890609185150312448	NaN	NaN	
9	890240255349198849	NaN	NaN	

	timestamp	\
0	2017-08-01 16:23:56 +0000	
1	2017-08-01 00:17:27 +0000	
2	2017-07-31 00:18:03 +0000	
3	2017-07-30 15:58:51 +0000	
4	2017-07-29 16:00:24 +0000	
5	2017-07-29 00:08:17 +0000	
6	2017-07-28 16:27:12 +0000	
7	2017-07-28 00:22:40 +0000	
8	2017-07-27 16:25:51 +0000	
9	2017-07-26 15:59:51 +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...	
5	<a href="http://twitter.com/download/iphone" r...	
6	<a href="http://twitter.com/download/iphone" r...	
7	<a href="http://twitter.com/download/iphone" r...	
8	<a href="http://twitter.com/download/iphone" r...	
9	<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
5  Here we have a majestic great white breaching ...                   NaN
6  Meet Jax. He enjoys ice cream so much he gets ...                   NaN
7  When you watch your owner call another dog a g...                   NaN
8  This is Zoey. She doesn't want to be one of th...                   NaN
9  This is Cassie. She is a college pup. Studying...                   NaN

   retweeted_status_user_id retweeted_status_timestamp  \
0                       NaN                        NaN
1                       NaN                        NaN
2                       NaN                        NaN
3                       NaN                        NaN
4                       NaN                        NaN
5                       NaN                        NaN
6                       NaN                        NaN
7                       NaN                        NaN
8                       NaN                        NaN
9                       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
5  https://twitter.com/dog_rates/status/891087950...                13
6  https://gofundme.com/ydvmve-surgery-for-jax,ht...                13
7  https://twitter.com/dog_rates/status/890729181...                13
8  https://twitter.com/dog_rates/status/890609185...                13
9  https://twitter.com/dog_rates/status/890240255...                14

   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
5                  10      None   None    None   None  None
6                  10       Jax   None    None   None  None
7                  10      None   None    None   None  None
8                  10      Zoey   None    None   None  None
9                  10    Cassie  doggo    None   None  None
```

```
In [8]: data_twitter.tail(10)
```

Out[8]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id
2346	666058600524156928	NaN	NaN	2015-11-16 01:01:59 +0000	<a href="http://twitter.com/download/iphone" r...	Here is the Rand Paul of retrievers folks! He'...	NaN
2347	666057090499244032	NaN	NaN	2015-11-16 00:55:59 +0000	<a href="http://twitter.com/download/iphone" r...	My oh my. This is a rare blond Canadian terrie...	NaN
2348	666055525042405380	NaN	NaN	2015-11-16 00:49:46 +0000	<a href="http://twitter.com/download/iphone" r...	Here is a Siberian heavily armored polar bear ...	NaN
2349	666051853826850816	NaN	NaN	2015-11-16 00:35:11 +0000	<a href="http://twitter.com/download/iphone" r...	This is an odd dog. Hard on the outside but lo...	NaN
2350	666050758794694657	NaN	NaN	2015-11-16 00:30:50 +0000	<a href="http://twitter.com/download/iphone" r...	This is a truly beautiful English Wilson Staff...	NaN
2351	666049248165822465	NaN	NaN	2015-11-16 00:24:50 +0000	<a href="http://twitter.com/download/iphone" r...	Here we have a 1949 1st generation vulpix. Enj...	NaN
2352	666044226329800704	NaN	NaN	2015-11-16 00:04:52 +0000	<a href="http://twitter.com/download/iphone" r...	This is a purebred Piers Morgan. Loves to Netf...	NaN
2353	666033412701032449	NaN	NaN	2015-11-15 23:21:54 +0000	<a href="http://twitter.com/download/iphone" r...	Here is a very happy pup. Big fan of well-main...	NaN
2354	666029285002620928	NaN	NaN	2015-11-15 23:05:30 +0000	<a href="http://twitter.com/download/iphone" r...	This is a western brown Mitsubishi terrier. Up...	NaN
2355	666020888022790149	NaN	NaN	2015-11-15 22:32:08 +0000	<a href="http://twitter.com/download/iphone" r...		

```
2355  Here we have a Japanese Irish Setter. Lost eye...                  NaN

      retweeted_status_user_id retweeted_status_timestamp  \
2346                       NaN                         NaN
2347                       NaN                         NaN
2348                       NaN                         NaN
2349                       NaN                         NaN
2350                       NaN                         NaN
2351                       NaN                         NaN
2352                       NaN                         NaN
2353                       NaN                         NaN
2354                       NaN                         NaN
2355                       NaN                         NaN

                                              expanded_urls  rating_numerator  \
2346  https://twitter.com/dog_rates/status/666058600...                     8
2347  https://twitter.com/dog_rates/status/666057090...                     9
2348  https://twitter.com/dog_rates/status/666055525...                    10
2349  https://twitter.com/dog_rates/status/666051853...                     2
2350  https://twitter.com/dog_rates/status/666050758...                    10
2351  https://twitter.com/dog_rates/status/666049248...                     5
2352  https://twitter.com/dog_rates/status/666044226...                     6
2353  https://twitter.com/dog_rates/status/666033412...                     9
2354  https://twitter.com/dog_rates/status/666029285...                     7
2355  https://twitter.com/dog_rates/status/666020888...                     8

      rating_denominator  name doggo floofer pupper puppo
2346                  10   the  None    None   None  None
2347                  10     a  None    None   None  None
2348                  10     a  None    None   None  None
2349                  10    an  None    None   None  None
2350                  10     a  None    None   None  None
2351                  10  None  None    None   None  None
2352                  10     a  None    None   None  None
2353                  10     a  None    None   None  None
2354                  10     a  None    None   None  None
2355                  10  None  None    None   None  None
```

In [9]:
```
data_twitter.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 [10]: data_twitter.describe()
```

Out[10]:

	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+16	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 [11]: images.head(10)
```

Out[11]:

	tweet_id	jpg_url
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg
5	666050758794694657	https://pbs.twimg.com/media/CT5Jof1WUAEuVxN.jpg
6	666051853826850816	https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg
7	666055525042405380	https://pbs.twimg.com/media/CT5N9tpXIAAifs1.jpg
8	666057090499244032	https://pbs.twimg.com/media/CT5PY90WoAAQGLo.jpg
9	666058600524156928	https://pbs.twimg.com/media/CT5Qw94XAAA_2dP.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
5	1	Bernese_mountain_dog	0.651137	True	English_springer
6	1	box_turtle	0.933012	False	mud_turtle
7	1	chow	0.692517	True	Tibetan_mastiff
8	1	shopping_cart	0.962465	False	shopping_basket
9	1	miniature_poodle	0.201493	True	komondor

	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
5	0.263788	True	Greater_Swiss_Mountain_dog	0.016199	True
6	0.045885	False	terrapin	0.017885	False
7	0.058279	True	fur_coat	0.054449	False
8	0.014594	False	golden_retriever	0.007959	True
9	0.192305	True	soft-coated_wheaten_terrier	0.082086	True

In [12]:
```
images.tail(10)
```

Out[12]:

	tweet_id	jpg_url
2065	890240255349198849	https://pbs.twimg.com/media/DFrEyVuW0AAO3t9.jpg
2066	890609185150312448	https://pbs.twimg.com/media/DFwUU__XcAEpyXI.jpg
2067	890729181411237888	https://pbs.twimg.com/media/DFyBahAVwAAhUTd.jpg
2068	890971913173991426	https://pbs.twimg.com/media/DF1eOmZXUAALUcq.jpg
2069	891087950875897856	https://pbs.twimg.com/media/DF3HwyEWsAABqE6.jpg
2070	891327558926688256	https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg
2071	891689557279858688	https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg
2072	891815181378084864	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg
2073	892177421306343426	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg

```
2074  892420643555336193  https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg

      img_num                     p1   p1_conf  p1_dog                 p2  \
2065        1              Pembroke  0.511319    True          Cardigan
2066        1          Irish_terrier  0.487574    True      Irish_setter
2067        2           Pomeranian  0.566142    True        Eskimo_dog
2068        1           Appenzeller  0.341703    True     Border_collie
2069        1  Chesapeake_Bay_retriever  0.425595    True     Irish_terrier
2070        2                basset  0.555712    True  English_springer
2071        1           paper_towel  0.170278   False  Labrador_retriever
2072        1             Chihuahua  0.716012    True          malamute
2073        1             Chihuahua  0.323581    True          Pekinese
2074        1                orange  0.097049   False             bagel

       p2_conf  p2_dog                       p3   p3_conf  p3_dog
2065  0.451038    True                  Chihuahua  0.029248    True
2066  0.193054    True  Chesapeake_Bay_retriever  0.118184    True
2067  0.178406    True                 Pembroke  0.076507    True
2068  0.199287    True                ice_lolly  0.193548   False
2069  0.116317    True          Indian_elephant  0.076902   False
2070  0.225770    True  German_short-haired_pointer  0.175219    True
2071  0.168086    True                  spatula  0.040836   False
2072  0.078253    True                   kelpie  0.031379    True
2073  0.090647    True                 papillon  0.068957    True
2074  0.085851   False                   banana  0.076110   False
```

In [13]: images.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 [14]: images.describe()

Out[14]:

	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 [15]: images.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 [16]: sum(images.jpg_url.duplicated())
```

```
Out[16]: 66
```

```
In [17]: print(images.p1_dog.value_counts())
         print(images.p2_dog.value_counts())
         print(images.p3_dog.value_counts())
```

```
True     1532
False     543
Name: p1_dog, dtype: int64
True     1553
False     522
Name: p2_dog, dtype: int64
True     1499
False     576
Name: p3_dog, dtype: int64
```

```
In [18]: favorite_retweet_table.head(10)
```

Out[18]:

	favorite_count	retweet_count	tweet_id
0	38197	8366	892420643555336193
1	32748	6177	892177421306343426
2	24664	4089	891815181378084864
3	41522	8503	891689557279858688
4	39719	9207	891327558926688256
5	19936	3064	891087950875897856
6	11665	2029	890971913173991426
7	64457	18568	890729181411237888
8	27395	4205	890609185150312448
9	31434	7270	890240255349198849

```
In [19]: favorite_retweet_table.tail(10)
```

Out[19]:

	favorite_count	retweet_count	tweet_id
2332	111	57	666058600524156928
2333	294	141	666057090499244032
2334	431	244	666055525042405380
2335	1210	841	666051853826850816
2336	132	59	666050758794694657
2337	107	40	666049248165822465
2338	296	139	666044226329800704
2339	125	44	666033412701032449
2340	129	47	666029285002620928
2341	2541	508	666020888022790149

```
In [20]: favorite_retweet_table.info()
```

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

```
In [21]: favorite_retweet_table.describe()
```

Out[21]:

	favorite_count	retweet_count	tweet_id
count	2342.000000	2342.000000	2.342000e+03
mean	7998.083689	2949.687020	7.422212e+17
std	12379.215144	4960.763537	6.832408e+16
min	0.000000	0.000000	6.660209e+17
25%	1382.750000	592.500000	6.783509e+17
50%	3482.500000	1376.500000	7.186224e+17
75%	9803.500000	3441.250000	7.986971e+17
max	164902.000000	84054.000000	8.924206e+17

```
In [22]: data_twitter['rating_numerator'].value_counts()
```

```
Out[22]: 12      558
         11      464
         10      461
         13      351
         9       158
         8       102
         7        55
         14       54
         5        37
         6        32
         3        19
         4        17
         1         9
         2         9
         420       2
         0         2
         15        2
         75        2
         80        1
         20        1
         24        1
         26        1
         44        1
         50        1
         60        1
         165       1
         84        1
         88        1
         144       1
         182       1
         143       1
         666       1
         960       1
         1776      1
         17        1
         27        1
         45        1
         99        1
         121       1
         204       1
         Name: rating_numerator, dtype: int64
```

```
In [23]: print(data_twitter.loc[data_twitter.rating_numerator == 204, 'text'])
         print(data_twitter.loc[data_twitter.rating_numerator == 143, 'text'])
         print(data_twitter.loc[data_twitter.rating_numerator == 666, 'text'])
         print(data_twitter.loc[data_twitter.rating_numerator == 1176, 'text'])
         print(data_twitter.loc[data_twitter.rating_numerator == 144, 'text'])
```

```
1120    Say hello to this unbelievably well behaved sq...
Name: text, dtype: object
1634    Two sneaky puppers were not initially seen, mo...
Name: text, dtype: object
189    @s8n You tried very hard to portray this good ...
Name: text, dtype: object
Series([], Name: text, dtype: object)
1779    IT'S PUPPERGEDDON. Total of 144/120 ...I think...
Name: text, dtype: object
```

```
In [24]: #print whole text in order to verify numerators and denominators
         print(data_twitter['text'][1120]) #17 dogs
         print(data_twitter['text'][1634]) #13 dogs
         print(data_twitter['text'][313]) #just a tweet to explain actual ratings, this will b
         print(data_twitter['text'][189]) #no picture, this will be ignored when cleaning data
         print(data_twitter['text'][1779]) #12 dogs
```

```
Say hello to this unbelievably well behaved squad of doggos. 204/170 would try to pet all at or
Two sneaky puppers were not initially seen, moving the rating to 143/130. Please forgive us. TH
@jonnysun @Lin_Manuel ok jomny I know you're excited but 960/00 isn't a valid rating, 13/10 is
@s8n You tried very hard to portray this good boy as not so good, but you have ultimately faile
IT'S PUPPERGEDDON. Total of 144/120 ...I think https://t.co/ZanVtAtvIq
```

```
In [25]: data_twitter['rating_denominator'].value_counts()
```

```
Out[25]: 10     2333
         11        3
         50        3
         80        2
         20        2
         2         1
         16        1
         40        1
         70        1
         15        1
         90        1
         110       1
         120       1
         130       1
         150       1
         170       1
         7         1
         0         1
         Name: rating_denominator, dtype: int64
```

```
In [26]: print(data_twitter.loc[data_twitter.rating_denominator == 11, 'text'])
         print(data_twitter.loc[data_twitter.rating_denominator == 2, 'text'])
```

```
print(data_twitter.loc[data_twitter.rating_denominator == 16, 'text'])
print(data_twitter.loc[data_twitter.rating_denominator == 15, 'text'])
print(data_twitter.loc[data_twitter.rating_denominator == 7, 'text'])
```

```
784     RT @dog_rates: After so many requests, this is...
1068    After so many requests, this is Bretagne. She ...
1662    This is Darrel. He just robbed a 7/11 and is i...
Name: text, dtype: object
2335    This is an Albanian 3 1/2 legged  Episcopalian...
Name: text, dtype: object
1663    I'm aware that I could've said 20/16, but here...
Name: text, dtype: object
342    @docmisterio account started on 11/15/15
Name: text, dtype: object
516    Meet Sam. She smiles 24/7 &amp; secretly aspir...
Name: text, dtype: object
```

```
In [27]: print(data_twitter['text'][784]) #retweet - it will be deleted when delete all retwee
         print(data_twitter['text'][1068]) #actual rating 14/10 need to change manually
         print(data_twitter['text'][1662]) #actual rating 10/10 need to change manually
         print(data_twitter['text'][2335]) #actual rating 9/10 need to change manually
         print(data_twitter['text'][1663]) # tweet to explain rating
         print(data_twitter['text'][342]) #no rating - delete
         print(data_twitter['text'][516]) #no rating - delete
```

```
RT @dog_rates: After so many requests, this is Bretagne. She was the last surviving 9/11 searcl
After so many requests, this is Bretagne. She was the last surviving 9/11 search dog, and our s
This is Darrel. He just robbed a 7/11 and is in a high speed police chase. Was just spotted by
This is an Albanian 3 1/2 legged  Episcopalian. Loves well-polished hardwood flooring. Penis or
I'm aware that I could've said 20/16, but here at WeRateDogs we are very professional. An incor
@docmisterio account started on 11/15/15
Meet Sam. She smiles 24/7 &amp; secretly aspires to be a reindeer.
Keep Sam smiling by clicking and sharing this link:
https://t.co/98tB8y7y7t https://t.co/LouL5vdvxx
```

```
In [28]: data_twitter['name'].value_counts()
```

```
Out[28]: None             745
         a                 55
         Charlie           12
         Lucy              11
         Cooper            11
         Oliver            11
         Penny             10
         Tucker            10
         Lola              10
         Winston            9
```

```
Bo                9
the               8
Sadie             8
Toby              7
Daisy             7
an                7
Bailey            7
Buddy             7
Leo               6
Jack              6
Koda              6
Scout             6
Stanley           6
Oscar             6
Jax               6
Milo              6
Rusty             6
Bella             6
Dave              6
Phil              5
                ...
Cedrick           1
Asher             1
Pavlov            1
Molly             1
Wafer             1
Jersey            1
Rhino             1
Nico              1
Mark              1
Rufio             1
Brudge            1
this              1
Iggy              1
Ole               1
Pip               1
Trip              1
Glacier           1
Goliath           1
Lupe              1
Trevith           1
Leonidas          1
Koko              1
Lassie            1
General           1
Dylan             1
Rilo              1
Michelangelope    1
```

```
Clyde               1
Mojo                1
Georgie             1
Name: name, Length: 957, dtype: int64
```

```
In [29]: with pd.option_context('max_colwidth', 200):
             display(data_twitter[data_twitter['text'].str.contains(r"(\d+\.\d*\/\d+)")]
                     [['tweet_id', 'text', 'rating_numerator', 'rating_denominator']])
```

```
/Users/philipp/anaconda3/lib/python3.6/site-packages/ipykernel/__main__.py:2: UserWarning: This
  from ipykernel import kernelapp as app
```

```
                tweet_id  \
45    883482846933004288
340   832215909146226688
695   786709082849828864
763   778027034220126208
1689  681340665377193984
1712  680494726643068929


45                            This is Bella. She hopes her smile made you smile. If not, she
340                        RT @dog_rates: This is Logan, the Chow who lived. He solemnly swear
695                                  This is Logan, the Chow who lived. He solemnly swears h
763   This is Sophie. She's a Jubilant Bush Pupper. Super h*ckin rare. Appears at random just
1689                                                        I've been told there's a slight p
1712                                                        Here we have uncovered an entire

      rating_numerator  rating_denominator
45                   5                  10
340                 75                  10
695                 75                  10
763                 27                  10
1689                 5                  10
1712                26                  10
```

2.0.1 Quality

Completeness, Validity, Accuracy, Consistency => a.k.a content issues data_twitter - Keep original ratings (no retweets) that have images - Delete columns that won't be used for analysis - Erroneous datatypes (doggo, floofer, pupper and puppo columns) - Separate timestamp into day - month - year (3 columns) - Correct numerators with decimals - Correct denominators other than 10

images - Drop 66 jpg_url duplicated - Create 1 column for image prediction and 1 column for confidence level - Delete columns that won't be used for analysis

2.0.2 Tidiness

Untidy data => a.k.a structural issues

- Change tweet_id to type int64 in order to merge with the other 2 tables
- All tables should be part of one dataset

3 Cleaning Data

```
In [40]: data_twitter_clean = data_twitter.copy()
         images_clean = images.copy()
         json_clean = favorite_retweet_table.copy()
```

1.Quality Issue - data_twitter:

- Keep original ratings (no retweets) that have images

Based on info above, there are 181 values in retweeted_status_id and retweeted_status_user_id. Delete retweets. When I merge data_twitter with images, I will only take the ones with images.

```
In [41]: #CODE: Delete retweets by filtering the NaN of retweeted_status_user_id
         data_twitter_clean = data_twitter_clean[pd.isnull(data_twitter_clean['retweeted_status

         #TEST
         print(sum(data_twitter_clean.retweeted_status_user_id.value_counts()))
```

```
0
```

2.Quality Issue - data_twitter:

- Delete columns that won't be used for analysis

```
In [42]: #get the column names of data_twitter_clean
         print(list(data_twitter_clean))


         #CODE: Delete columns no needed
         data_twitter_clean = data_twitter_clean.drop(['source',
                                                       'in_reply_to_status_id',
                                                       'in_reply_to_user_id',
                                                       'retweeted_status_id',
                                                       'retweeted_status_user_id',
                                                       'retweeted_status_timestamp',
                                                       'expanded_urls'], 1)
```

```
['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id', 'timestamp', 'source', 'text', 're
```

```
In [43]: #TEST
         list(data_twitter_clean)

Out[43]: ['tweet_id',
          'timestamp',
          'text',
          'rating_numerator',
          'rating_denominator',
          'name',
          'doggo',
          'floofer',
          'pupper',
          'puppo']
```

3.Quality Issue - data_twitter:

- Erroneous datatypes (doggo, floofer, pupper and puppo columns)

Melt the doggo, floofer, pupper and puppo columns to dogs and dogs_stage column. Then drop dogs. Sort by dogs_stage in order to then drop duplicated based on tweet_id except for the last occurrence.

```
In [45]: # Check the values in those columns by excuting those columns
         print(data_twitter_clean.doggo.value_counts())
         print(data_twitter_clean.floofer.value_counts())
         print(data_twitter_clean.pupper.value_counts())
         print(data_twitter_clean.puppo.value_counts())

None     2088
doggo      87
Name: doggo, dtype: int64
None       2165
floofer      10
Name: floofer, dtype: int64
None      1941
pupper     234
Name: pupper, dtype: int64
None     2150
puppo      25
Name: puppo, dtype: int64


In [46]: #CODE
         # Select the columns to melt and to remain
         columns_to_melt = ['doggo', 'floofer', 'pupper', 'puppo']
         columns_to_stay = [x for x in data_twitter_clean.columns.tolist() if x not in columns

         # Melt the the columns into values
         data_twitter_clean = pd.melt(data_twitter_clean, id_vars = columns_to_stay, value_vars
```

```
                                    var_name = 'stages', value_name = 'dog_stage')

        # Delete column 'stages'
        data_twitter_clean = data_twitter_clean.drop('stages', 1)

        # Filter for unique values then remove duplicate values based on 'dog_stage' values

        #TEST 1
        print(data_twitter_clean.dog_stage.value_counts())

        data_twitter_clean = data_twitter_clean.sort_values('dog_stage').drop_duplicates('twe

        #TEST 2
        print(data_twitter_clean.dog_stage.value_counts())
        print(len(data_twitter_clean))
```

```
None       8344
pupper      234
doggo        87
puppo        25
floofer      10
Name: dog_stage, dtype: int64
None       1831
pupper      234
doggo        75
puppo        25
floofer      10
Name: dog_stage, dtype: int64
2175
```

4.Quality Issue - data_twitter:

- Separate timestamp into day - month - year (3 columns)

First convert timestamp to datetime. Then extract year, month and day to new columns. Finally drop timestamp column.

```
In [48]: #CODE: convert timestamp to datetime
         data_twitter_clean['timestamp'] = pd.to_datetime(data_twitter_clean['timestamp'])

         #extract year, month and day to new columns
         data_twitter_clean['year'] = data_twitter_clean['timestamp'].dt.year
         data_twitter_clean['month'] = data_twitter_clean['timestamp'].dt.month
         data_twitter_clean['day'] = data_twitter_clean['timestamp'].dt.day

         #Finally drop timestamp column
         data_twitter_clean = data_twitter_clean.drop('timestamp', 1)
```

```
In [50]: #TEST
         list(data_twitter_clean)

Out[50]: ['tweet_id',
          'text',
          'rating_numerator',
          'rating_denominator',
          'name',
          'dog_stage',
          'year',
          'month',
          'day']
```

5.Quality Issue - data_twitter:

- Correct numerators with decimals

```
In [51]: data_twitter_clean[['rating_numerator', 'rating_denominator']] = data_twitter_clean[[

         data_twitter_clean.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 2095 to 7298
Data columns (total 9 columns):
tweet_id              2175 non-null int64
text                  2175 non-null object
rating_numerator      2175 non-null float64
rating_denominator    2175 non-null float64
name                  2175 non-null object
dog_stage             2175 non-null object
year                  2175 non-null int64
month                 2175 non-null int64
day                   2175 non-null int64
dtypes: float64(2), int64(4), object(3)
memory usage: 169.9+ KB


In [52]: #CODE

         #First change numerator and denominators type int to float to allow decimals
         data_twitter_clean[['rating_numerator', 'rating_denominator']] = data_twitter_clean[[

         #Update numerators

         data_twitter_clean.loc[(data_twitter_clean.tweet_id == 883482846933004288), 'rating_nu
         data_twitter_clean.loc[(data_twitter_clean.tweet_id == 786709082849828864), 'rating_nu
         data_twitter_clean.loc[(data_twitter_clean.tweet_id == 778027034220126208), 'rating_nu
         data_twitter_clean.loc[(data_twitter_clean.tweet_id == 681340665377193984), 'rating_nu
         data_twitter_clean.loc[(data_twitter_clean.tweet_id == 680494726643068929), 'rating_nu
```

```
#TEST
with pd.option_context('max_colwidth', 200):
    display(data_twitter_clean[data_twitter_clean['text'].str.contains(r"(\d+\.\d*\/\d
            [['tweet_id', 'text', 'rating_numerator', 'rating_denominator']])
```

```
/Users/philipp/anaconda3/lib/python3.6/site-packages/ipykernel/__main__.py:16: UserWarning: Thi
```

```
                tweet_id  \
42    883482846933004288
3685  681340665377193984
3708  680494726643068929
2733  786709082849828864
4967  778027034220126208


42                               This is Bella. She hopes her smile made you smile. If not, she i
3685                                                        I've been told there's a slight po
3708                                                        Here we have uncovered an entire b
2733                                       This is Logan, the Chow who lived. He solemnly swears he
4967  This is Sophie. She's a Jubilant Bush Pupper. Super h*ckin rare. Appears at random just t

      rating_numerator  rating_denominator
42               13.50                10.0
3685              9.50                10.0
3708             11.26                10.0
2733              9.75                10.0
4967             11.27                10.0
```

6.Quality Issue - data_twitter:

- Correct denominators other than 10

Manually and programatically. Five tweets with denominator not equal to 10 for special circunstances. Update both numerators and denominators when necessary. Delete other five tweets because they do not have actual ratings. These tweets with denominator not equal to 10 are multiple dogs.

In [56]:
```
#CODE: Update both numerators and denominators
data_twitter_clean.loc[(data_twitter_clean.tweet_id == 740373189193256964), 'rating_nu
data_twitter_clean.loc[(data_twitter_clean.tweet_id == 740373189193256964), 'rating_de

data_twitter_clean.loc[(data_twitter_clean.tweet_id == 682962037429899265), 'rating_nu
data_twitter_clean.loc[(data_twitter_clean.tweet_id == 682962037429899265), 'rating_de

data_twitter_clean.loc[(data_twitter_clean.tweet_id == 666287406224695296), 'rating_nu
data_twitter_clean.loc[(data_twitter_clean.tweet_id == 666287406224695296), 'rating_de
```

```
data_twitter_clean.loc[(data_twitter_clean.tweet_id == 722974582966214656), 'rating_nu
data_twitter_clean.loc[(data_twitter_clean.tweet_id == 722974582966214656), 'rating_de

data_twitter_clean.loc[(data_twitter_clean.tweet_id == 716439118184652801), 'rating_nu
data_twitter_clean.loc[(data_twitter_clean.tweet_id == 716439118184652801), 'rating_de

#CODE: Delete five tweets with no actual ratings
data_twitter_clean = data_twitter_clean[data_twitter_clean['tweet_id'] != 832088576586
data_twitter_clean = data_twitter_clean[data_twitter_clean['tweet_id'] != 810984652412
data_twitter_clean = data_twitter_clean[data_twitter_clean['tweet_id'] != 682808988178
data_twitter_clean = data_twitter_clean[data_twitter_clean['tweet_id'] != 835246439529
data_twitter_clean = data_twitter_clean[data_twitter_clean['tweet_id'] != 686035780142

#TEST: Left only the group dogs for programatically clean
with pd.option_context('max_colwidth', 200):
    display(data_twitter_clean[data_twitter_clean['rating_denominator'] != 10][['twee
```

	tweet_id	\
3429	697463031882764288	
3631	684222868335505415	
3630	684225744407494656	
3250	710658690886586372	
3225	713900603437621249	
3270	709198395643068416	
3347	704054845121142784	
3775	677716515794329600	
3839	675853064436391936	
2538	820690176645140481	
2908	758467244762497024	
3117	731156023742988288	

3429	Happy Wednesday here's a bucket of
3631	Someone help the girl is being mugged. Several are distracting her while two steal
3630	Two sneaky puppers were not initially seen, moving the rating to 143/
3250	Here's a brigade of puppers. All look very prepared
3225	Happy Saturday here's 9 puppers on a
3270	From left to right:\nCletus, Jerome, Alejandro, Burp, & Titson\nNone know where camer
3347	Here is a whole flock of
3775	IT'S PUPPERGI
3839	Here we have an entire platoon of puppers. Total s
2538	The floofs have been released I repeat the
2908	Why does this never h
3117	Say hello to this unbelievably well behaved squad of doggos. 204

```
      rating_numerator  rating_denominator
3429              44.0                40.0
3631             121.0               110.0
3630             143.0               130.0
3250              80.0                80.0
3225              99.0                90.0
3270              45.0                50.0
3347              60.0                50.0
3775             144.0               120.0
3839              88.0                80.0
2538              84.0                70.0
2908             165.0               150.0
3117             204.0               170.0
```

In [57]:
```
#CODE: Create a new column with rating in float type to avoid converting all int colu
data_twitter_clean['rating'] = 10 * data_twitter_clean['rating_numerator'] / data_twi

#TEST
data_twitter_clean.sample(5)
```

Out[57]:
```
                tweet_id                                               text  \
355   821765923262631936  This is Duchess. She uses dark doggo forces to...
3727  679844490799091713  This is Willie. He's floating away and needs y...
5628  695095422348574720  This is just a beautiful pupper good shit evol...
2986  749064354620928000  Meet Winston. He's pupset because I forgot to ...
4098  670679630144274432  This is Pluto. He's holding little waddling do...

      rating_numerator  rating_denominator     name dog_stage  year  month  \
355               13.0                10.0  Duchess     doggo  2017      1
3727              10.0                10.0   Willie      None  2015     12
5628              12.0                10.0     just    pupper  2016      2
2986              11.0                10.0  Winston      None  2016      7
4098               8.0                10.0    Pluto      None  2015     11

      day  rating
355    18    13.0
3727   24    10.0
5628    4    12.0
2986    2    11.0
4098   28     8.0
```

7.Quality Issue - images:

- Drop 66 jpg_url duplicated

In [58]:
```
#CODE: Delete duplicated jpg_url
images_clean = images_clean.drop_duplicates(subset=['jpg_url'], keep='last')
```

```
In [58]: #TEST
         sum(images_clean['jpg_url'].duplicated())
```

```
Out[58]: 0
```

8.Quality Issue - images:

- Create 1 column for image prediction and 1 column for confidence level

```
In [61]: #CODE: the first true prediction (p1, p2 or p3) will be store in these lists
         dog_type = []
         confidence_list = []

         #create a function with nested if to capture the dog type and confidence level
         # from the first 'true' prediction
         def sort_image(image_prediction):
             if image_prediction['p1_dog'] == True:
                 dog_type.append(image_prediction['p1'])
                 confidence_list.append(image_prediction['p1_conf'])
             elif image_prediction['p2_dog'] == True:
                 dog_type.append(image_prediction['p2'])
                 confidence_list.append(image_prediction['p2_conf'])
             elif image_prediction['p3_dog'] == True:
                 dog_type.append(image_prediction['p3'])
                 confidence_list.append(image_prediction['p3_conf'])
             else:
                 dog_type.append('Error')
                 confidence_list.append('Error')

         #series objects having index the images_clean column.
         images_clean.apply(sort_image, axis=1)

         #create new columns
         images_clean['dog_type'] = dog_type
         images_clean['confidence_list'] = confidence_list

         #drop rows that has prediction_list 'error'
         images_clean = images_clean[images_clean['dog_type'] != 'Error']

         #TEST:
         images_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1691 entries, 0 to 2073
Data columns (total 14 columns):
tweet_id           1691 non-null int64
jpg_url            1691 non-null object
```

```
img_num             1691 non-null int64
p1                  1691 non-null object
p1_conf             1691 non-null float64
p1_dog              1691 non-null bool
p2                  1691 non-null object
p2_conf             1691 non-null float64
p2_dog              1691 non-null bool
p3                  1691 non-null object
p3_conf             1691 non-null float64
p3_dog              1691 non-null bool
dog_type            1691 non-null object
confidence_list     1691 non-null object
dtypes: bool(3), float64(3), int64(2), object(6)
memory usage: 163.5+ KB
```

9.Quality Issue - images:

- Delete columns that won't be used for analysis

```
In [62]: #CODE: print list of image_prediction columns
         print(list(images_clean))

         #Delete columns
         images_clean = images_clean.drop(['img_num', 'p1',
                                           'p1_conf', 'p1_dog',
                                           'p2', 'p2_conf',
                                           'p2_dog', 'p3',
                                           'p3_conf',
                                           'p3_dog'], 1)

         #TEST
         list(images_clean)
```

```
['tweet_id', 'jpg_url', 'img_num', 'p1', 'p1_conf', 'p1_dog', 'p2', 'p2_conf', 'p2_dog', 'p3',
```

```
Out[62]: ['tweet_id', 'jpg_url', 'dog_type', 'confidence_list']
```

1.Tidiness Issue - favorite_retweet_table:

- Change tweet_id to type int64 in order to merge with the other 2 tables

```
In [70]: #CODE: change tweet_id from str to int
         json_clean['tweet_id'] = json_clean['tweet_id'].astype(int)

         #TEST
         json_clean['tweet_id'].dtypes
```

```
Out[70]: dtype('int64')
```

2.Tidiness Issue:

- merge all tables to one dataset

```
In [72]: #CODE: create a new dataframe that merge data_twitter_clean and images_clean
         df_twitter = pd.merge(data_twitter_clean,
                               images_clean,
                               how = 'left', on = ['tweet_id'])

         #keep rows that have picture (jpg_url)
         df_twitter = df_twitter[df_twitter['jpg_url'].notnull()]

         #TEST
         df_twitter.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1625 entries, 1 to 2169
Data columns (total 13 columns):
tweet_id              1625 non-null int64
text                  1625 non-null object
rating_numerator      1625 non-null float64
rating_denominator    1625 non-null float64
name                  1625 non-null object
dog_stage             1625 non-null object
year                  1625 non-null int64
month                 1625 non-null int64
day                   1625 non-null int64
rating                1625 non-null float64
jpg_url               1625 non-null object
dog_type              1625 non-null object
confidence_list       1625 non-null object
dtypes: float64(3), int64(4), object(6)
memory usage: 177.7+ KB
```

```
In [74]: #CODE: create a new dataframe that merge df_twitter and json_clean
         df_twitter_master = pd.merge(df_twitter, json_clean,
                               how = 'left', on = ['tweet_id'])

         #TEST
         df_twitter_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1625 entries, 0 to 1624
Data columns (total 15 columns):
tweet_id              1625 non-null int64
text                  1625 non-null object
rating_numerator      1625 non-null float64
rating_denominator    1625 non-null float64
```

```
name                  1625 non-null object
dog_stage             1625 non-null object
year                  1625 non-null int64
month                 1625 non-null int64
day                   1625 non-null int64
rating                1625 non-null float64
jpg_url               1625 non-null object
dog_type              1625 non-null object
confidence_list       1625 non-null object
favorite_count        1624 non-null float64
retweet_count         1624 non-null float64
dtypes: float64(5), int64(4), object(6)
memory usage: 203.1+ KB
```

```
In [75]: df_twitter_master.to_csv('twitter_archive_master.csv', index=False)
```