wrangle_act

July 14, 2018

```
In [3]: import pandas as pd
        import numpy as np
        import requests
        import json
        import os
        import tweepy
```

1 Step 1: Data Gathering

```
In [7]: #Twitter Authentication
        #Masking Twitter Authenticcation for security Purpose.

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

        auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_token, access_secret)
        api = tweepy.API(auth,wait_on_rate_limit=True)
```

```
In [8]: #Importing DataFrame 1(twitter_archive from existing file)

        twitter_archive=pd.read_csv("twitter-archive-enhanced.csv")

        #Importing DataFrame 2(image_predictions from url link)
        url='https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions
        response=requests.get(url)

        pred_file_name='image_predictions.tsv'
        arch_file_name='twitter-archive-enhanced'

        with open (os.path.join(pred_file_name), mode='wb') as filee:
            filee.write(response.content)

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

```
In [16]: # Preparing copy of Dataframes (tw_copy,im_test)

         tw_copy=twitter_archive.copy()
         im_test=image_predictions.copy()
```

```
In [17]: #Creating 3rd Data Frame(tweetinfo) from twitter API (Containing Tweet Info like Retwee
         # Removing tweets which not found / Deleted

         tweetlist=[]
         notfoundlist=[]

         if os.path.exists('data.txt'):
             os.remove('data.txt')

         for id in tw_copy.tweet_id:
             try:
                 tweet = api.get_status(id)
                 with open('data.txt', 'a') as outfile:
                     json.dump(tweet._json, outfile)
                 retweet=tweet._json['retweet_count']
                 favorite=tweet._json['favorite_count']
                 tweetlist.append([id,retweet,favorite])
             except:
                 notfoundlist.append(id)
                 continue

         tweetinfo = pd.DataFrame(tweetlist, columns=('tweet_id', 'retweet_count', 'favorite_cou

         print(tweetinfo.shape[0])     #:2344
         print(len(notfoundlist))      #:12
```

```
2341
15
```

2 Step 2 :Access Data

```
In [21]: tweetinfo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2341 entries, 0 to 2340
Data columns (total 3 columns):
tweet_id          2341 non-null int64
retweet_count     2341 non-null int64
favorite_count    2341 non-null int64
dtypes: int64(3)
memory usage: 54.9 KB
```

```
In [22]: tw_copy.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 [23]: im_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 14 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
breed         2075 non-null object
breed_prob    2075 non-null float64
dtypes: bool(3), float64(4), int64(2), object(5)
memory usage: 184.5+ KB
```

```
In [24]: tweetinfo.head()

Out[24]:              tweet_id  retweet_count  favorite_count
         0  892420643555336193           8546           38646
         1  892177421306343426           6285           33124
         2  891815181378084864           4165           24932
         3  891689557279858688           8677           42024
         4  891327558926688256           9426           40180
```

```
In [25]: tw_copy.head()

Out[25]:              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

                           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

                                                       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 [26]: im_test.head()
```

```
Out[26]:              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

                            breed breed_prob
         0  Welsh_springer_spaniel   0.465074
         1                 redbone   0.506826
         2         German_shepherd   0.596461
         3     Rhodesian_ridgeback   0.408143
         4      miniature_pinscher   0.560311
```

3 Step 3 :Clean Data

3.1 Data Cleansing Process

Here we will be taking care of below Issues:

Quality Issues: 1.5.Remove All Retweets (Since we needs only Original Tweets)
2.Remove record whole url don't exist or Invalid (Tweet Deleted)
3.Ignore records having Out of bound ratings like: (1776/10 , 960/0)
4.Rationalize all rating Records out of 10
5.Segregated URL, Keep most appropriate One
6.Replace all Faulty Dog names to 'None'
7.Changing Data Type of 'breed' to Category
8.Changing type of retweet_count,favorite_count to int

Tidiness Issues: 1.Segregate Hour,Day,Date,Month,Year from DateTime column
2.Multiple Dog stages into 1 'Dog Stage'
3.Defining 'breed' , 'breed_probablity' columns considering p1_conf,p2_conf,p3_conf to remove excess columns

Handling Quality Issues

Define Consolidating probability of Individual dog to predict its Breed of Dog. #Defining 'breed' , 'breed_probablity' columns considering p1_conf,p2_conf,p3_conf

Code

```
In [27]: for i in range(im_test.shape[0]):
         #    print(i)
             if im_test.loc[i,'p1_dog']==True:
                 im_test.loc[i,'breed']=im_test.loc[i,'p1']
                 im_test.loc[i,'breed_prob']=im_test.loc[i,'p1_conf']
             elif im_test.loc[i,'p2_dog']==True:
                 im_test.loc[i,'breed']=im_test.loc[i,'p2']
                 im_test.loc[i,'breed_prob']=im_test.loc[i,'p2_conf']
             elif im_test.loc[i,'p3_dog']==True:
                 im_test.loc[i,'breed']=im_test.loc[i,'p3']
                 im_test.loc[i,'breed_prob']=im_test.loc[i,'p3_conf']
             else:
                 im_test.loc[i,'breed']='No Dog'
                 im_test.loc[i,'breed_prob']=0
```

```
In [28]: #Modifying DataFrame: im_test (Contain Only Dog's data)
         #Removing Extra Columns

         im_test=im_test[['tweet_id','jpg_url','img_num','breed','breed_prob']]
         #dim_final=dim_final[dim_final.breed != 'No Dog'].reset_index()
```

Test Testing the output:

```
In [29]: print(im_test.head())
```

```
             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                   breed  breed_prob
0        1  Welsh_springer_spaniel    0.465074
1        1                 redbone    0.506826
2        1         German_shepherd    0.596461
3        1     Rhodesian_ridgeback    0.408143
4        1      miniature_pinscher    0.560311
```

Define #Taking 97.5% confidenece Interval for dog probability (cutting off 2.5 % negative tail)
#changing Data Type of ‘breed’ to Category

Code

```
In [30]: Lower_limit=np.percentile(im_test.breed_prob, 2.5)
         Upper_limit=np.percentile(im_test.breed_prob, 100)

         im_test=im_test[im_test.breed_prob>=Lower_limit]
         im_test.breed=im_test.breed.astype('category')
```

Test

```
In [31]: print(im_test.breed.value_counts())
         print(Lower_limit,Upper_limit)
```

```
No Dog                            324
golden_retriever                  173
Labrador_retriever                113
Pembroke                           96
Chihuahua                          95
pug                                65
toy_poodle                         52
chow                               51
Samoyed                            46
Pomeranian                         42
malamute                           34
cocker_spaniel                     34
French_bulldog                     32
Chesapeake_Bay_retriever           31
miniature_pinscher                 26
Cardigan                           23
Eskimo_dog                         22
```

```
Staffordshire_bullterrier       22
German_shepherd                 21
beagle                          21
Siberian_husky                  20
Shih-Tzu                        20
Rottweiler                      19
kuvasz                          19
Lakeland_terrier                19
Shetland_sheepdog               19
Maltese_dog                     19
Italian_greyhound               17
basset                          17
West_Highland_white_terrier     16
                              ...
Rhodesian_ridgeback              4
Saluki                           4
giant_schnauzer                  4
Scottish_deerhound               4
Tibetan_terrier                  4
Weimaraner                       4
Welsh_springer_spaniel           4
Irish_water_spaniel              3
toy_terrier                      3
komondor                         3
curly-coated_retriever           3
Brabancon_griffon                3
cairn                            3
briard                           3
Leonberg                         3
Greater_Swiss_Mountain_dog       3
black-and-tan_coonhound          2
Sussex_spaniel                   2
groenendael                      2
Australian_terrier               2
Appenzeller                      2
wire-haired_fox_terrier          2
Irish_wolfhound                  1
clumber                          1
EntleBucher                      1
Bouvier_des_Flandres             1
Scotch_terrier                   1
Japanese_spaniel                 1
silky_terrier                    1
standard_schnauzer               1
Name: breed, Length: 114, dtype: int64
0.0 0.999956
```

Define #Merging im_test,tweetinfo into im_test #Changing type of retweet_count,favorite_count to int #Updating NaN retweet_count,favorite_count to '0' (indicating not found)

Code

```
In [32]: im_test=pd.merge(im_test,tweetinfo,on=['tweet_id'],how='left')

         im_test.retweet_count=im_test.retweet_count.astype(str).str[:-2]
         im_test.retweet_count=im_test.retweet_count.replace('n',0)
         im_test.favorite_count=im_test.favorite_count.astype(str).str[:-2]
         im_test.favorite_count=im_test.favorite_count.replace('n',0)
```

Test

```
In [33]: print(im_test.head(2))
```

```
             tweet_id                                          jpg_url  \
0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
1  666029285002620928  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg

   img_num                   breed  breed_prob retweet_count favorite_count
0        1  Welsh_springer_spaniel    0.465074           517           2564
1        1                 redbone    0.506826            47            130
```

Define #Remove All Retweet (Since we needs only Original Tweets)

Code

```
In [34]: print(tw_copy.shape[0])
         tw_copy=tw_copy[tw_copy.retweeted_status_id.isnull()]
```

```
2356
```

Test

```
In [35]: print(tw_copy.shape[0])
```

```
2175
```

Define #Segregated URL, Keep most appropriate One

Code

```
In [36]: tw_copy.expanded_urls=tw_copy.expanded_urls.fillna("No URL")

         def urlextract(dfurl):
             dfurl['url']=dfurl['expanded_urls'].split(',')[len(dfurl['expanded_urls'].split(','
             return dfurl['url']

         tw_copy['url']=tw_copy.apply(urlextract,axis=1)
```

Test

```
In [37]: print(tw_copy['url'].head(10))
```

```
0    https://twitter.com/dog_rates/status/892420643...
1    https://twitter.com/dog_rates/status/892177421...
2    https://twitter.com/dog_rates/status/891815181...
3    https://twitter.com/dog_rates/status/891689557...
4    https://twitter.com/dog_rates/status/891327558...
5    https://twitter.com/dog_rates/status/891087950...
6    https://twitter.com/dog_rates/status/890971913...
7    https://twitter.com/dog_rates/status/890729181...
8    https://twitter.com/dog_rates/status/890609185...
9    https://twitter.com/dog_rates/status/890240255...
Name: url, dtype: object
```

Define #Rationalize all rating Records out of 10 #Drop Numerator and Denominator retweeted_status_user_id,retweeted_status_timestamp columns

Code

```
In [38]: tw_copy['rating']=(tw_copy['rating_numerator']/tw_copy['rating_denominator'])*10
         tw_copy=tw_copy.drop(['rating_numerator','rating_denominator','retweeted_status_user_id
```

Test

```
In [39]: print(tw_copy.head())
```

```
             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

                   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

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

                                       expanded_urls      name doggo floofer  \
0  https://twitter.com/dog_rates/status/892420643...   Phineas  None    None
1  https://twitter.com/dog_rates/status/892177421...     Tilly  None    None
2  https://twitter.com/dog_rates/status/891815181...    Archie  None    None
3  https://twitter.com/dog_rates/status/891689557...     Darla  None    None
4  https://twitter.com/dog_rates/status/891327558...  Franklin  None    None

  pupper puppo                                               url  rating
0   None  None  https://twitter.com/dog_rates/status/892420643...    13.0
1   None  None  https://twitter.com/dog_rates/status/892177421...    13.0
2   None  None  https://twitter.com/dog_rates/status/891815181...    12.0
3   None  None  https://twitter.com/dog_rates/status/891689557...    13.0
4   None  None  https://twitter.com/dog_rates/status/891327558...    12.0
```

Manual Cleaning of few Obserserved records

Define #Records Cleasing as per manual observations #1.Rating for tweet_id:835246439529840640 should be 13/10 (Interpretted from Tweet's Text) #2.Rating for tweet:id:835152434251116546 should be 11/10 (Interpretted from Tweet's Image) #3.Rating for tweet: 786709082849828864: Valid Rating>9.75/10 #4.Remove record with tweet_id: 810984652412424192 (Invalid interpretation of dog rating) #5.Remove record with tweet_id: 855862651834028034,855860136149123072,838150277551247360 (Improper Ratings ratio, No URLs, Invalid Text)

```
In [40]: tw_copy.tweet_id=tw_copy.tweet_id.astype(str)
         im_test.tweet_id=im_test.tweet_id.astype(str)

         #print(tw_copy[tw_copy.tweet_id=='835246439529840640'])
```

```
#print(tw_copy[tw_copy.tweet_id=='835152434251116546'])
#print(tw_copy[tw_copy.tweet_id=='855862651834028034'])
#print(tw_copy[tw_copy.tweet_id=='838150277551247360'])


tw_copy.loc[tw_copy.tweet_id == '835246439529840640','rating']=13
tw_copy.loc[tw_copy.tweet_id == '835152434251116546','rating']=11
tw_copy.loc[tw_copy.tweet_id == '786709082849828864','rating']=9.75
tw_copy=tw_copy[~tw_copy.tweet_id.isin(['855862651834028034','855860136149123072','8381
```

Test

```
In [42]: print(tw_copy[tw_copy.tweet_id=='835246439529840640'])
         print(tw_copy[tw_copy.tweet_id=='835152434251116546'])
         print(tw_copy[tw_copy.tweet_id=='855862651834028034'])
         print(tw_copy[tw_copy.tweet_id=='838150277551247360'])
```

```
               tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
313  835246439529840640           8.352460e+17           26259576.0

                     timestamp  \
313  2017-02-24 21:54:03 +0000

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

                                                  text expanded_urls  name  \
313  @jonnysun @Lin_Manuel ok jomny I know you're e...        No URL  None

    doggo floofer pupper puppo     url  rating
313  None    None   None  None  No URL    13.0
               tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
315  835152434251116546                    NaN                  NaN

                     timestamp  \
315  2017-02-24 15:40:31 +0000

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

                                                  text  \
315  When you're so blinded by your systematic plag...

                                         expanded_urls  name doggo floofer  \
315  https://twitter.com/dog_rates/status/835152434...  None  None    None

    pupper puppo                                                url  rating
315   None  None  https://twitter.com/dog_rates/status/835152434...    11.0
```

```
Empty DataFrame
Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, text, expande
Index: []
Empty DataFrame
Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, text, expande
Index: []
```

Define #Replace all Faulty Dog names to 'None'

Invalid names Includes ('a','all','an','by','his','not','O','officially','old','one','quite','such','the','this','unaccept

Code

```
In [43]: invalid_names=['a','all','an','by','his','not','O','officially','old','one','quite','su

         for invalid in invalid_names:
             tw_copy.loc[tw_copy.name == invalid,'name']='None'
```

Test

```
In [44]: print(tw_copy.name.value_counts())
```

```
None          764
Lucy           11
Charlie        11
Cooper         10
Oliver         10
Tucker          9
Penny           9
Lola            8
Sadie           8
Winston         8
Toby            7
Daisy           7
Bo              6
Bailey          6
Stanley         6
Bella           6
Oscar           6
Koda            6
Jax             6
Buddy           5
Milo            5
Rusty           5
Bentley         5
Leo             5
Louis           5
Chester         5
```

```
Scout           5
Dave            5
Sophie          4
Bear            4
             ...
Skittle         1
Kendall         1
Obi             1
Linus           1
Dutch           1
Laela           1
Ralphus         1
Grizz           1
Augie           1
Dale            1
Cermet          1
Butter          1
Lassie          1
Bobb            1
Eve             1
Aldrick         1
Christoper      1
Shnuggles       1
Snicku          1
Tuck            1
Mojo            1
Marlee          1
Durg            1
Edd             1
Burt            1
Lucky           1
Bungalo         1
Blipson         1
Raphael         1
Colin           1
Name: name, Length: 941, dtype: int64
```

Define #Date Time Column Type conversion to DataTime

Code

```
In [45]: tw_copy.timestamp = pd.to_datetime(tw_copy.timestamp)
```

Test

```
In [46]: print(type(tw_copy.timestamp))
```

```
<class 'pandas.core.series.Series'>
```

Define #Extracting Hour,Day,Date,Month,Year from DateTime column, This would be helpful in doing analysis time based.

Code

```
In [47]: tw_copy['hour']=tw_copy['timestamp'].dt.hour.astype(int)
         tw_copy['weekday']=tw_copy['timestamp'].dt.dayofweek.astype(int)
         tw_copy['date']=tw_copy['timestamp'].dt.day.astype(int)
         tw_copy['month']=tw_copy['timestamp'].dt.month.astype(int)
         tw_copy['year']=tw_copy['timestamp'].dt.year.astype(int)
```

Test

```
In [48]: print(tw_copy.head(2))
```

```
             tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
0  892420643555336193                    NaN                  NaN
1  892177421306343426                    NaN                  NaN

            timestamp                                             source  \
0 2017-08-01 16:23:56  <a href="http://twitter.com/download/iphone" r...
1 2017-08-01 00:17:27  <a href="http://twitter.com/download/iphone" r...

                                                text  \
0  This is Phineas. He's a mystical boy. Only eve...
1  This is Tilly. She's just checking pup on you...

                                       expanded_urls     name doggo floofer  \
0  https://twitter.com/dog_rates/status/892420643...  Phineas  None    None
1  https://twitter.com/dog_rates/status/892177421...    Tilly  None    None

  pupper puppo                                                url  rating  \
0   None  None  https://twitter.com/dog_rates/status/892420643...    13.0
1   None  None  https://twitter.com/dog_rates/status/892177421...    13.0

   hour  weekday  date  month  year
0    16        1     1      8  2017
1     0        1     1      8  2017
```

Define #Multiple Dog stages into 1 'Dog Stages'.This would Save table space and Table would look simple. Defining multiple Stage with comma"," seperator. Like : doggo,floofer

Code

```
In [49]: tw_copy['dog_stage'] = 'None'

         def dogstagecon(dog):
             stage=[]
             if dog.doggo=='doggo':
                 stage.append('doggo')
             if dog.floofer=='floofer':
                 stage.append('floofer')
             if dog.pupper=='pupper':
                 stage.append('pupper')
             if dog.puppo=='puppo':
                 stage.append('puppo')
             if len(stage)<1:
                 stage.append('None')
             dog['dog_stage']=','.join(stage)
             return dog

         tw_copy=tw_copy.apply(dogstagecon,axis=1)

         tw_copy=tw_copy.drop(['doggo','floofer','pupper','puppo'],axis=1)
```

Test

```
In [50]: print(tw_copy.dog_stage.value_counts())
```

```
None                1827
pupper               224
doggo                 75
puppo                 24
doggo,pupper          10
floofer                9
doggo,floofer          1
doggo,puppo            1
Name: dog_stage, dtype: int64
```

Define #Drop all column which are not needed Columns like: timestamp’,‘in_reply_to_status_id’,’in_reply_to_user_id not needed any more, Hence Dropping it. #Count number of Replied Tweets

Code

```
In [51]: replied_tweet_count=tw_copy.in_reply_to_status_id.shape[0]-sum(tw_copy.in_reply_to_stat
         tw_copy=tw_copy.drop(['timestamp','in_reply_to_status_id','in_reply_to_user_id'],axis=1
```

Test

```
In [52]: print(replied_tweet_count)
         print(tw_copy.head(32))
```

```
75
              tweet_id                                             source  \
0   892420643555336193  <a href="http://twitter.com/download/iphone" r...
1   892177421306343426  <a href="http://twitter.com/download/iphone" r...
2   891815181378084864  <a href="http://twitter.com/download/iphone" r...
3   891689557279858688  <a href="http://twitter.com/download/iphone" r...
4   891327558926688256  <a href="http://twitter.com/download/iphone" r...
5   891087950875897856  <a href="http://twitter.com/download/iphone" r...
6   890971913173991426  <a href="http://twitter.com/download/iphone" r...
7   890729181411237888  <a href="http://twitter.com/download/iphone" r...
8   890609185150312448  <a href="http://twitter.com/download/iphone" r...
9   890240255349198849  <a href="http://twitter.com/download/iphone" r...
10  890006608113172480  <a href="http://twitter.com/download/iphone" r...
11  889880896479866881  <a href="http://twitter.com/download/iphone" r...
12  889665388333682689  <a href="http://twitter.com/download/iphone" r...
13  889638837579907072  <a href="http://twitter.com/download/iphone" r...
14  889531135344209921  <a href="http://twitter.com/download/iphone" r...
15  889278841981685760  <a href="http://twitter.com/download/iphone" r...
16  888917238123831296  <a href="http://twitter.com/download/iphone" r...
17  888804989199671297  <a href="http://twitter.com/download/iphone" r...
18  888554962724278272  <a href="http://twitter.com/download/iphone" r...
20  888078434458587136  <a href="http://twitter.com/download/iphone" r...
21  887705289381826560  <a href="http://twitter.com/download/iphone" r...
22  887517139158093824  <a href="http://twitter.com/download/iphone" r...
23  887473957103951883  <a href="http://twitter.com/download/iphone" r...
24  887343217045368832  <a href="http://twitter.com/download/iphone" r...
25  887101392804085760  <a href="http://twitter.com/download/iphone" r...
26  886983233522544640  <a href="http://twitter.com/download/iphone" r...
27  886736880519319552  <a href="http://twitter.com/download/iphone" r...
28  886680336477933568  <a href="http://twitter.com/download/iphone" r...
29  886366144734445568  <a href="http://twitter.com/download/iphone" r...
30  886267009285017600  <a href="http://twitter.com/download/iphone" r...
31  886258384151887873  <a href="http://twitter.com/download/iphone" r...
33  885984800019947520  <a href="http://twitter.com/download/iphone" r...

                                                 text  \
0   This is Phineas. He's a mystical boy. Only eve...
1   This is Tilly. She's just checking pup on you...
2   This is Archie. He is a rare Norwegian Pouncin...
3   This is Darla. She commenced a snooze mid meal...
4   This is Franklin. He would like you to stop ca...
5   Here we have a majestic great white breaching ...
6   Meet Jax. He enjoys ice cream so much he gets ...
7   When you watch your owner call another dog a g...
```

```
8   This is Zoey. She doesn't want to be one of th...
9   This is Cassie. She is a college pup. Studying...
10  This is Koda. He is a South Australian decksha...
11  This is Bruno. He is a service shark. Only get...
12  Here's a puppo that seems to be on the fence a...
13  This is Ted. He does his best. Sometimes that'...
14  This is Stuart. He's sporting his favorite fan...
15  This is Oliver. You're witnessing one of his m...
16  This is Jim. He found a fren. Taught him how t...
17  This is Zeke. He has a new stick. Very proud o...
18  This is Ralphus. He's powering up. Attempting ...
20  This is Gerald. He was just told he didn't get...
21  This is Jeffrey. He has a monopoly on the pool...
22  I've yet to rate a Venezuelan Hover Wiener. Th...
23  This is Canela. She attempted some fancy porch...
24  You may not have known you needed to see this ...
25  This... is a Jubilant Antarctic House Bear. We...
26  This is Maya. She's very shy. Rarely leaves he...
27  This is Mingus. He's a wonderful father to his...
28  This is Derek. He's late for a dog meeting. 13...
29  This is Roscoe. Another pupper fallen victim t...
30  @NonWhiteHat @MayhewMayhem omg hello tanner yo...
31  This is Waffles. His doggles are pupside down...
33  Viewer discretion advised. This is Jimbo. He w...

                                        expanded_urls      name  \
0   https://twitter.com/dog_rates/status/892420643...   Phineas
1   https://twitter.com/dog_rates/status/892177421...     Tilly
2   https://twitter.com/dog_rates/status/891815181...    Archie
3   https://twitter.com/dog_rates/status/891689557...     Darla
4   https://twitter.com/dog_rates/status/891327558...  Franklin
5   https://twitter.com/dog_rates/status/891087950...      None
6   https://gofundme.com/ydvmve-surgery-for-jax,ht...       Jax
7   https://twitter.com/dog_rates/status/890729181...      None
8   https://twitter.com/dog_rates/status/890609185...      Zoey
9   https://twitter.com/dog_rates/status/890240255...    Cassie
10  https://twitter.com/dog_rates/status/890006608...      Koda
11  https://twitter.com/dog_rates/status/889880896...     Bruno
12  https://twitter.com/dog_rates/status/889665388...      None
13  https://twitter.com/dog_rates/status/889638837...       Ted
14  https://twitter.com/dog_rates/status/889531135...    Stuart
15  https://twitter.com/dog_rates/status/889278841...    Oliver
16  https://twitter.com/dog_rates/status/888917238...       Jim
17  https://twitter.com/dog_rates/status/888804989...      Zeke
18  https://twitter.com/dog_rates/status/888554962...   Ralphus
20  https://twitter.com/dog_rates/status/888078434...    Gerald
21  https://twitter.com/dog_rates/status/887705289...   Jeffrey
22  https://twitter.com/dog_rates/status/887517139...      None
```

```
23  https://twitter.com/dog_rates/status/887473957...    Canela
24  https://twitter.com/dog_rates/status/887343217...      None
25  https://twitter.com/dog_rates/status/887101392...      None
26  https://twitter.com/dog_rates/status/886983233...      Maya
27  https://www.gofundme.com/mingusneedsus,https:/...    Mingus
28  https://twitter.com/dog_rates/status/886680336...     Derek
29  https://twitter.com/dog_rates/status/886366144...    Roscoe
30                                             No URL      None
31  https://twitter.com/dog_rates/status/886258384...   Waffles
33  https://twitter.com/dog_rates/status/885984800...     Jimbo

                                                  url  rating  hour  weekday  \
0   https://twitter.com/dog_rates/status/892420643...    13.0    16        1
1   https://twitter.com/dog_rates/status/892177421...    13.0     0        1
2   https://twitter.com/dog_rates/status/891815181...    12.0     0        0
3   https://twitter.com/dog_rates/status/891689557...    13.0    15        6
4   https://twitter.com/dog_rates/status/891327558...    12.0    16        5
5   https://twitter.com/dog_rates/status/891087950...    13.0     0        5
6   https://twitter.com/dog_rates/status/890971913...    13.0    16        4
7   https://twitter.com/dog_rates/status/890729181...    13.0     0        4
8   https://twitter.com/dog_rates/status/890609185...    13.0    16        3
9   https://twitter.com/dog_rates/status/890240255...    14.0    15        2
10  https://twitter.com/dog_rates/status/890006608...    13.0     0        2
11  https://twitter.com/dog_rates/status/889880896...    13.0    16        1
12  https://twitter.com/dog_rates/status/889665388...    13.0     1        1
13  https://twitter.com/dog_rates/status/889638837...    12.0     0        1
14  https://twitter.com/dog_rates/status/889531135...    13.0    17        0
15  https://twitter.com/dog_rates/status/889278841...    13.0     0        0
16  https://twitter.com/dog_rates/status/888917238...    12.0     0        6
17  https://twitter.com/dog_rates/status/888804989...    13.0    16        5
18  https://twitter.com/dog_rates/status/888554962...    13.0     0        5
20  https://twitter.com/dog_rates/status/888078434...    12.0    16        3
21  https://twitter.com/dog_rates/status/887705289...    13.0    16        2
22  https://twitter.com/dog_rates/status/887517139...    14.0     3        2
23  https://twitter.com/dog_rates/status/887473957...    13.0     0        2
24  https://twitter.com/dog_rates/status/887343217...    13.0    16        1
25  https://twitter.com/dog_rates/status/887101392...    12.0     0        1
26  https://twitter.com/dog_rates/status/886983233...    13.0    16        0
27  https://twitter.com/dog_rates/status/886736880...    13.0    23        6
28  https://twitter.com/dog_rates/status/886680336...    13.0    20        6
29  https://twitter.com/dog_rates/status/886366144...    12.0    23        5
30                                             No URL    12.0    16        5
31  https://twitter.com/dog_rates/status/886258384...    13.0    16        5
33  https://twitter.com/dog_rates/status/885984800...    12.0    22        4

    date  month  year dog_stage
0      1      8  2017      None
1      1      8  2017      None
```

```
2      31       7  2017       None
3      30       7  2017       None
4      29       7  2017       None
5      29       7  2017       None
6      28       7  2017       None
7      28       7  2017       None
8      27       7  2017       None
9      26       7  2017      doggo
10     26       7  2017       None
11     25       7  2017       None
12     25       7  2017      puppo
13     25       7  2017       None
14     24       7  2017      puppo
15     24       7  2017       None
16     23       7  2017       None
17     22       7  2017       None
18     22       7  2017       None
20     20       7  2017       None
21     19       7  2017       None
22     19       7  2017       None
23     19       7  2017       None
24     18       7  2017       None
25     18       7  2017       None
26     17       7  2017       None
27     16       7  2017       None
28     16       7  2017       None
29     15       7  2017     pupper
30     15       7  2017       None
31     15       7  2017       None
33     14       7  2017       None
```

Define #Merging im_test,tw_copy as : clean_df (Dimention Table) #Replacing Source as Appropriate Source and changing type to categorical variable As we found oout that Source Either code be one of below Only * Twitter for iPhone * Twitter Web Client * TweetDeck

Code

```
In [53]: clean_df=pd.DataFrame()
         clean_df=pd.merge(im_test,tw_copy,on=['tweet_id'],how='left')
         clean_df=clean_df.drop(['jpg_url','img_num','url','expanded_urls','text'],axis=1)
         clean_df.source=clean_df.source.astype('category')
         clean_df.source=clean_df.source.replace(['<a href="http://twitter.com/download/iphone"

         clean_df=clean_df[~clean_df.source.isnull()]
```

Test

```
In [54]: print(clean_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1993 entries, 0 to 2074
Data columns (total 14 columns):
tweet_id          1993 non-null object
breed             1993 non-null category
breed_prob        1993 non-null float64
retweet_count     1993 non-null object
favorite_count    1993 non-null object
source            1993 non-null object
name              1993 non-null object
rating            1993 non-null float64
hour              1993 non-null float64
weekday           1993 non-null float64
date              1993 non-null float64
month             1993 non-null float64
year              1993 non-null float64
dog_stage         1993 non-null object
dtypes: category(1), float64(7), object(6)
memory usage: 225.8+ KB
None
```

Define #Changing type of Year,day,month,year to Int and hence into corresponding textual name. Like 0>Monday,1>Tuesday . . . 6>Sunday 1>January,2>February . . . 12>December

Code

```
In [55]: clean_df['hour']=clean_df['hour'].astype(int)
         clean_df['weekday']=clean_df['weekday'].astype(int)
         clean_df['date']=clean_df['date'].astype(int)
         clean_df['month']=clean_df['month'].astype(int)
         clean_df['year']=clean_df['year'].astype(int)
         clean_df['retweet_count']=clean_df['retweet_count'].astype(int)
         clean_df['favorite_count']=clean_df['favorite_count'].astype(int)

         int_day=[0,1,2,3,4,5,6]
         str_day=['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']

         clean_df.weekday.replace(int_day, str_day, inplace=True)

         int_month=[1,2,3,4,5,6,7,8,9,10,11,12]
         str_month=['January','February','March','April','May','June','July','August','September

         clean_df.month.replace(int_month, str_month, inplace=True)
```

Test

```
In [56]: print(clean_df.tail(10))
         print(clean_df.info())
```

```
                tweet_id                     breed  breed_prob  retweet_count  \
2065  890240255349198849                  Pembroke    0.511319           7438
2066  890609185150312448             Irish_terrier    0.487574           4277
2067  890729181411237888                Pomeranian    0.566142          18941
2068  890971913173991426               Appenzeller    0.341703           2076
2069  891087950875897856  Chesapeake_Bay_retriever    0.425595           3122
2070  891327558926688256                    basset    0.555712           9426
2071  891689557279858688        Labrador_retriever    0.168086           8677
2072  891815181378084864                 Chihuahua    0.716012           4165
2073  892177421306343426                 Chihuahua    0.323581           6285
2074  892420643555336193                    No Dog    0.000000           8546

      favorite_count              source      name  rating  hour    weekday  \
2065           31837  Twitter for iPhone    Cassie    14.0    15  Wednesday
2066           27697  Twitter for iPhone      Zoey    13.0    16   Thursday
2067           65279  Twitter for iPhone      None    13.0     0     Friday
2068           11809  Twitter for iPhone       Jax    13.0    16     Friday
2069           20146  Twitter for iPhone      None    13.0     0   Saturday
2070           40180  Twitter for iPhone  Franklin    12.0    16   Saturday
2071           42024  Twitter for iPhone     Darla    13.0    15     Sunday
2072           24932  Twitter for iPhone    Archie    12.0     0     Monday
2073           33124  Twitter for iPhone     Tilly    13.0     0    Tuesday
2074           38646  Twitter for iPhone   Phineas    13.0    16    Tuesday

      date   month  year dog_stage
2065    26    July  2017     doggo
2066    27    July  2017      None
2067    28    July  2017      None
2068    28    July  2017      None
2069    29    July  2017      None
2070    29    July  2017      None
2071    30    July  2017      None
2072    31    July  2017      None
2073     1  August  2017      None
2074     1  August  2017      None
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1993 entries, 0 to 2074
Data columns (total 14 columns):
tweet_id          1993 non-null object
breed             1993 non-null category
breed_prob        1993 non-null float64
retweet_count     1993 non-null int32
favorite_count    1993 non-null int32
source            1993 non-null object
```

```
name                1993 non-null object
rating              1993 non-null float64
hour                1993 non-null int32
weekday             1993 non-null object
date                1993 non-null int32
month               1993 non-null object
year                1993 non-null int32
dog_stage           1993 non-null object
dtypes: category(1), float64(2), int32(5), object(6)
memory usage: 186.9+ KB
None
```

Define #Defining Datafram :only_dog_clean (Where Breed != 'No Dog') We are removing those records which confirms No Dog.

Code

```
In [57]: only_dog_clean=pd.DataFrame()
         only_dog_clean=clean_df.copy()
         only_dog_clean=only_dog_clean[only_dog_clean.breed!='No Dog']
```

Test

```
In [58]: print(only_dog_clean.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1685 entries, 0 to 2073
Data columns (total 14 columns):
tweet_id            1685 non-null object
breed               1685 non-null category
breed_prob          1685 non-null float64
retweet_count       1685 non-null int32
favorite_count      1685 non-null int32
source              1685 non-null object
name                1685 non-null object
rating              1685 non-null float64
hour                1685 non-null int32
weekday             1685 non-null object
date                1685 non-null int32
month               1685 non-null object
year                1685 non-null int32
dog_stage           1685 non-null object
dtypes: category(1), float64(2), int32(5), object(6)
memory usage: 158.9+ KB
None
```

Saving Clean Dataframe as "dogdf_clean.csv" file

```
In [59]: only_dog_clean.to_csv('dogdf_clean.csv',index=False)
```

4 Data Analysis and Visualization

Most Retweeted Tweet !!

```
In [60]: only_dog_clean.loc[only_dog_clean.retweet_count==only_dog_clean.retweet_count.max(),['t
```

```
Out[60]:                 tweet_id               breed  name  retweet_count
         1221  744234799360020481  Labrador_retriever  None          76969
```

A Labrador_retriever with an unknown name recieved 76986 Retweets,
which makes Tweet_id:744234799360020481 most retweeted Tweet.

Most Favorite Tweet !!

```
In [61]: only_dog_clean.loc[only_dog_clean.favorite_count==only_dog_clean.favorite_count.max(),[
```

```
Out[61]:                 tweet_id             breed  name  favorite_count
         1744  822872901745569793  Lakeland_terrier  None          142761
```

A Lakeland_terrier dog made Tweet_id:822872901745569793 most liked tweet,
It recieved 142803 likes !

Most Popular Breed ! Most common Breed as per Original twitter posts :

```
In [62]: only_dog_clean.breed.value_counts()[:5,]
```

```
Out[62]: golden_retriever      157
         Labrador_retriever    108
         Pembroke               95
         Chihuahua              91
         pug                    62
         Name: breed, dtype: int64
```

Observation Golden Retrievers are the most popular choice among dog lovers.
They are loyal, loving and intelligent breed, hence most popular.Next to Golden Retriever is Labrador Retreiver.
Which is the most common dog ratings ?
Top 4 ratings by WeRateDogs as:

```
In [63]: only_dog_clean.rating.value_counts()[:4,]
```

```
Out[63]: 12.0    426
         11.0    363
         10.0    362
         13.0    229
         Name: rating, dtype: int64
```

Observaton

Most Dog have recieved 12 Ratings out of 10 (All raings have been converted with common denominator 10)

Which was the Most popular Source of Tweeting !

```
In [64]: import matplotlib.pyplot as plt
         label=['Iphone','WebAppliaction','TweetDeck']
         only_dog_clean.source.value_counts().plot(kind='pie')
         print(only_dog_clean.source.value_counts()/ len(only_dog_clean) * 100)
```

```
Twitter for iPhone    98.160237
Twitter Web Client     1.305638
TweetDeck              0.534125
Name: source, dtype: float64
```

Observaton Twitter Handle's operator uses Iphone for tweeting purpose most of the times compared to 'Twitter Web client' and 'TweetDeck' Application.

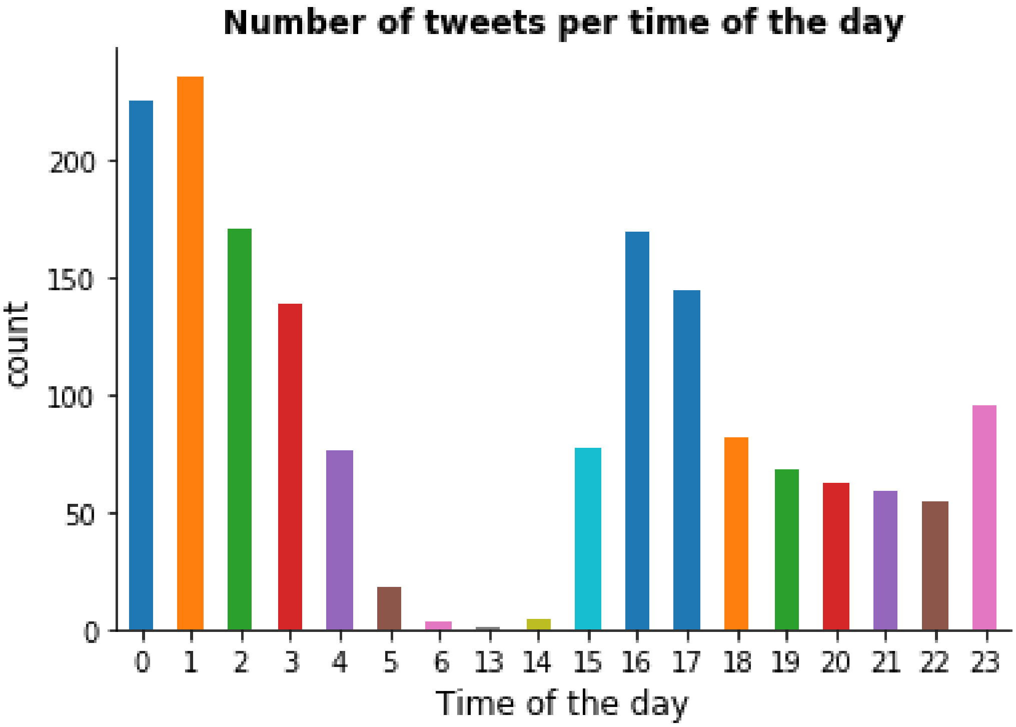
Few observed Tweets with Source as Vine were not Original Tweets i.e. Replied Tweets, Which we not included for analysis purpose.

Most Popular Hour

```
In [65]: only_dog_clean.hour.value_counts()
```

```
Out[65]: 1     236
         0     225
         2     171
         16    170
         17    144
         3     139
         23     95
         18     82
         15     77
         4      76
         19     68
         20     62
         21     59
         22     55
         5      18
         14      4
         6       3
         13      1
         Name: hour, dtype: int64
```

```
In [66]: fig, ax = plt.subplots()
         only_dog_clean['hour'].value_counts(sort=False).plot(kind='bar')
         ax.set_title('Number of tweets per time of the day', fontweight="bold")
ax.set_ylabel('count', fontsize=12)
ax.set_xlabel('Time of the day', fontsize=12)
plt.xticks(rotation='horizontal')
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
```



Observation: There were no tweets fron 7:00 to 13:00 and only few contable tweets from 5:00 to 7:00 (May be Sleeping Time)

Twitter Account consistently made higher number of tweets during midnight ie: from 23:00 to 03:00. Highest number of tweets on average were 236 at 1:00 to 2:00 Night.

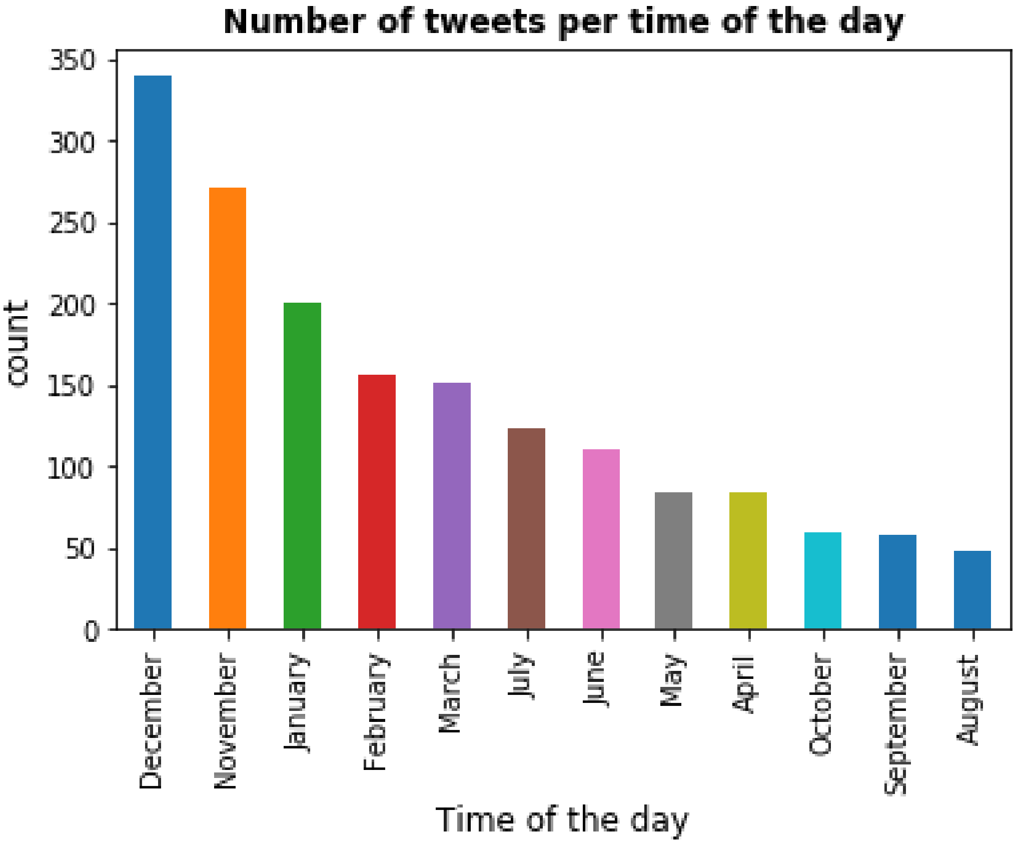
Most Active Month

```
In [67]: only_dog_clean.month.value_counts()

Out[67]: December     340
         November     271
         January      200
         February     156
         March        151
         July         124
         June         110
         May           84
```

```
April         84
October       60
September     57
August        48
Name: month, dtype: int64
```

```
In [68]: #Bar Chart

         fig, ax = plt.subplots()
         only_dog_clean['month'].value_counts().plot(kind='bar')

         ax.set_title('Number of tweets per time of the day', fontweight="bold")

         ax.set_ylabel('count', fontsize=12)
         ax.set_xlabel('Time of the day', fontsize=12)

         plt.xticks(rotation='vertical')
```

```
Out[68]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),
          <a list of 12 Text xticklabel objects>)
```



Observation This Twitter account was more active in Winters than Summer.

340 Tweets were made in December alone which is more than 10 tweets per day on average compared to 50 tweets in August on an average!

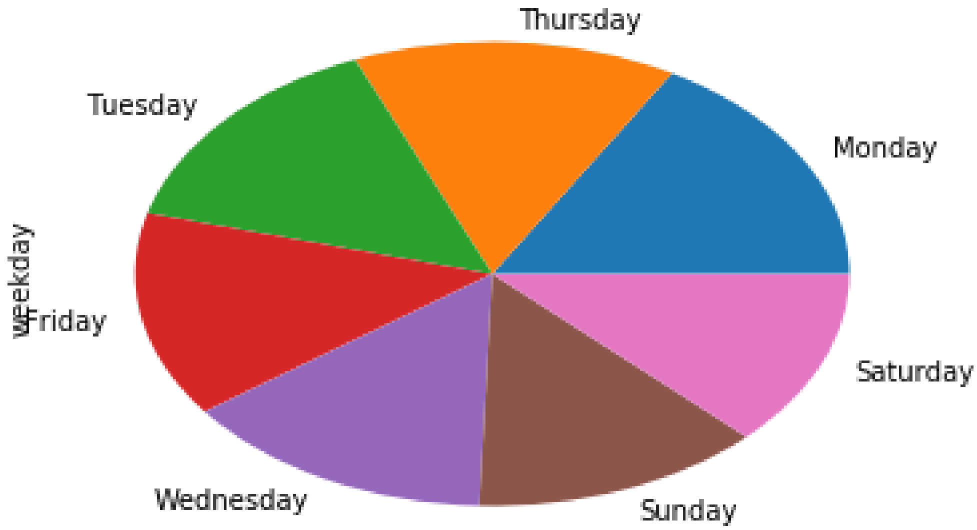
Most Active Day of Week

```
In [69]: only_dog_clean.weekday.value_counts()
```

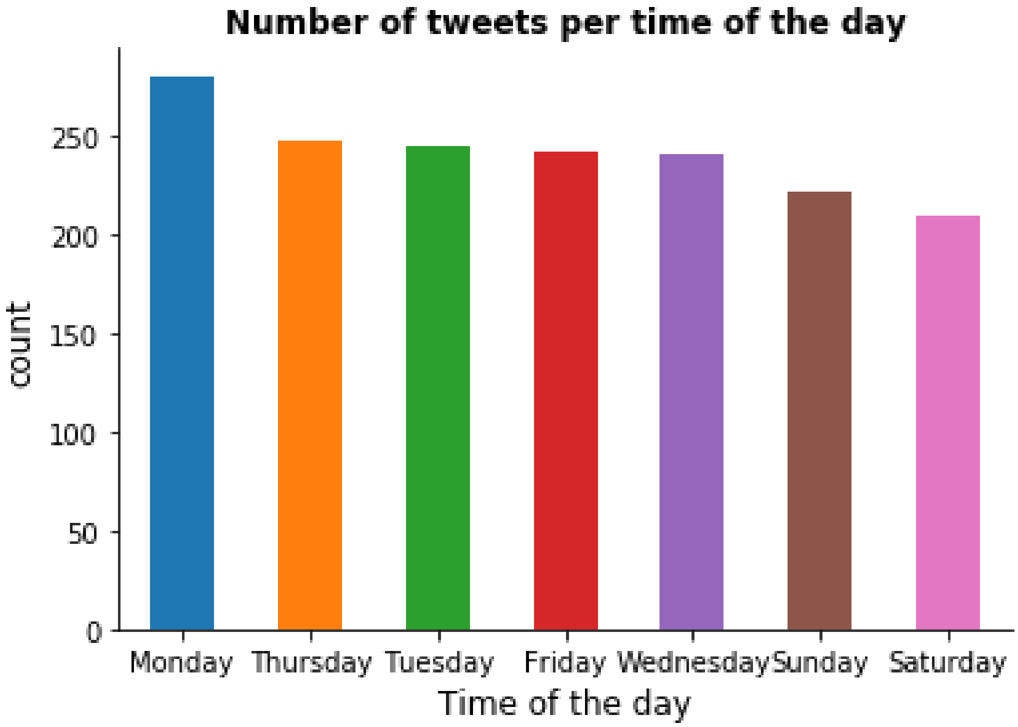
```
Out[69]: Monday       280
         Thursday     247
         Tuesday      244
         Friday       242
         Wednesday    241
         Sunday       222
         Saturday     209
         Name: weekday, dtype: int64
```

```
In [125]: #Pie Chart
          only_dog_clean.weekday.value_counts().plot(kind='pie')
```

```
Out[125]: <matplotlib.axes._subplots.AxesSubplot at 0x7f963d79b630>
```



```
In [70]: #Bar Chart
         fig, ax = plt.subplots()
         only_dog_clean['weekday'].value_counts(sort=True).plot(kind='bar')
         ax.set_title('Number of tweets per time of the day', fontweight="bold")
         ax.set_ylabel('count', fontsize=12)
ax.set_xlabel('Time of the day', fontsize=12)
plt.xticks(rotation='horizontal')
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
```



Observation: Most of the tweets were on Weekdays rather than Weekends.

There were 250 Tweets per day on average on Weekdays, wheres as 215 Tweets per day on Weekends. More Busy on Weekends !

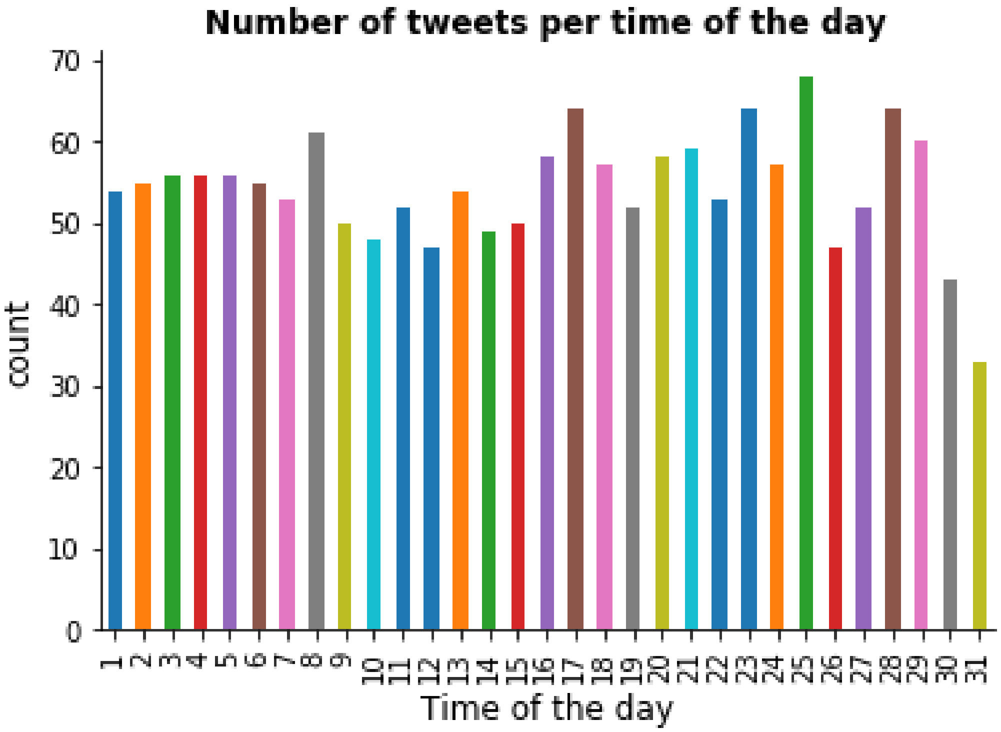
Most Active Day

```
In [71]: only_dog_clean.date.value_counts()
```

```
Out[71]: 25    68
         23    64
         17    64
         28    64
         8     61
         29    60
         21    59
         20    58
         16    58
         24    57
```

```
18    57
3     56
4     56
5     56
2     55
6     55
13    54
1     54
7     53
22    53
11    52
19    52
27    52
9     50
15    50
14    49
10    48
12    47
26    47
30    43
31    33
Name: date, dtype: int64
```

```
In [72]: fig, ax = plt.subplots()
         only_dog_clean['date'].value_counts(sort=False).plot(kind='bar')
         ax.set_title('Number of tweets per time of the day', fontweight="bold")
         ax.set_ylabel('count', fontsize=12)
         ax.set_xlabel('Time of the day', fontsize=12)
         plt.xticks(rotation='vertical')
         ax.spines['right'].set_visible(False)
         ax.spines['top'].set_visible(False)
```



Observation : There no common trend observed from the chart, Tweets were even distributed throught the days.

On an Average there are 68-33 Tweets per day.

Most Popular Dog Stage

```
In [73]: only_dog_clean.dog_stage.value_counts()/len(only_dog_clean)*100
```

```
Out[73]: None                84.569733
         pupper               9.970326
         doggo                3.204748
         puppo                1.246291
         doggo,pupper         0.474777
         floofer              0.415430
         doggo,floofer        0.059347
         doggo,puppo          0.059347
         Name: dog_stage, dtype: float64
```

Observation: Although we have 85 percent missing Data for Dog current Stages,

As per Available data 10 percent of Dogs are Pupper and seems popular choice.Next we have Doggo and Puppo in the List.We have few floofer in the list too.

Most Popular Dog Name

```
In [74]: only_dog_clean.name.value_counts()[1:6,]
```

```
Out[74]: Charlie    10
         Cooper     10
         Lucy       10
         Oliver      9
         Tucker      9
         Name: name, dtype: int64
```

Observation: Among available data and Excuding all those invalid Names, Cooper,Lucy,Charlie are among most popular choices for Dog names.Next most popular dog's names are Oliver , Tucker.

If you want your dog's name to be different from others Please dont name him Cooper !