wrangle_act2

January 26, 2018

1 WeRateDogs Data Wrangling

The dataset that we have been wrangling consists of three sources:

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

Additional Data via the Twitter API

Back to the basic-ness of Twitter archives: retweet count and favorite count are two of the notable column omissions. Fortunately, this additional data can be gathered by anyone from Twitter's API. Well, "anyone" who has access to data for the 3000 most recent tweets, at least. But you, because you have the WeRateDogs Twitter archive and specifically the tweet IDs within it, can gather this data for all 5000+. And guess what? You're going to query Twitter's API to gather this valuable data.

The current Tweet data of those archived tweets has been extracted from the life twitter API

-

1.1 Gather

```
In [2]: import requests

        #deal with data
        import numpy as np
        import pandas as pd

        #deal with datetime
        import datetime as dt
        import pytz

        #deal with visualization
        import seaborn as sns
        %matplotlib inline
        import matplotlib.pyplot as plt

        #use padasql for SQL-query on dataframe
        #http://blog.yhat.com/posts/pandasql-intro.html
        from pandasql import sqldf
```

1.1.1 gather twitter-archive-enhanced.csv file

```
In [2]: #read locally stored file
        df_twarchive = pd.read_csv('twitter-archive-enhanced.csv')
```

1.1.2 gather image-predictions.tsv file

```
In [3]: tgt_filename = 'image-predictions.tsv'

In [15]: #download file from the cloud
         url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predicti
         r = requests.get(url)
         with open(tgt_filename, 'wb') as file:
             file.write(r.content)

In [4]: #read dataframe from file
        df_image = pd.read_csv(tgt_filename, sep = '\t')
        df_image.tweet_id.count()

Out[4]: 2075
```

1.1.3 gather tweed_json.txt

Each tweet's retweet count and favorite ("like") count at minimum, and any additional data you find interesting. Using the **tweet IDs in the WeRateDogs Twitter archive**, query the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file. Each tweet's JSON data should be written to its own line. Then read this .txt file line by line into a pandas DataFrame with (at minimum) - tweet ID - retweet count - favorite count

```
In [5]: #refer to twitter application https://apps.twitter.com/app/14714948/keys
        #import authorization variables for security reasons
        import access.my_secret as xs
        import tweepy

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

Extract tweet in json format from twitter api: def get_tweet(tweet_id): tweet = api.get_status(tweet_id, tweet_mode='extended', wait_on_rate_limit=True) return tweet._json

```
In [20]: # for testing
         # print(get_tweet(666082916733198337))
```

```
In [5]: #get the timer set up
        from timeit import default_timer as timer
        import datetime
        import json

        json_filename = 'tweet_json.txt'

In [22]: #get tweet
         #write json file line by line and log progress
         with open(json_filename, 'w') as file, open('log.txt','w') as log:
             for tweet_id in df_twarchive.tweet_id:
                 result = ''
                 start = timer()
                 try:
                     content = get_tweet(tweet_id)
                     #add newline is helpful to read the file line by line later
                     file.write(json.dumps(content) + '\n')
                     result = 'ok'
                 except tweepy.TweepError:
                     result = 'TweepError'
                 end = timer()
                 log.write(('%s\t%s\t%s\t%s\n') % (str(datetime.datetime.now()), result, str(twe


In [6]: #count number of tweets
        i = 0
        with open(json_filename, 'r') as file:
            for row in file:
                i += 1
        print("row count:", i)

row count: 2345
```

1.2 Assess

After gathering each of the above pieces of data, assess them visually and programmatically for quality and tidiness issues. Detect and document at least **eight (8) quality issues and two (2) tidiness issues** in your wrangle_act.ipynb Jupyter Notebook. To meet specifications, the **issues that satisfy the Project Motivation must be assessed.**

Of the 5000+ tweets, I have filtered for tweets with ratings only (there are 2356).

1.2.1 assess twitter-archive-enhanced.csv file

quality issues

1. derive clear categories for source(iPhone App, vine.co, Twitter Web Client, TweetDeck)
2. make this source column categorical
3. deal with a few invalid ratings where rating_numerator is 0 or rating_denominator is 0

    - 835246439529840640 rating has been changed to from 960/0 to 13/10 according to current twitter post
    - 835152434251116546 rating has been changed to from 0/10 to 11/10 according to current twitter post
    - 746906459439529985 no rating according to current twitter post -> filter out

4. take out ratings with a rating_denominator other than 10 as these are invalid ratings or rate multiple dogs at once.
5. take out ratings with rating_numerator > 15 they seem to be invalid
6. take out retweeds You only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets. (retweeted_status_id is NaN)
7. convert wrong dog names to Null: "a", "the", "an", "n", "None"

tidyness issues

1. four columns for the dog_stage can be put into one categorial variable (doggo,floofer,pupper,puppo) and turn the default value "None" into a Null or "unknown"

```
In [7]: df_twarchive.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 [8]: df_twarchive.sample(10)
```

```
Out[8]:                 tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
        1473  693647888581312512                    NaN                  NaN
```

```
2320  666437273139982337                    NaN                   NaN
1690  681339448655802368                    NaN                   NaN
1100  735648611367784448                    NaN                   NaN
28    886680336477933568                    NaN                   NaN
1326  706153300320784384                    NaN                   NaN
559   803276597545603072                    NaN                   NaN
2071  671115716440031232                    NaN                   NaN
802   772193107915964416                    NaN                   NaN
2125  670361874861563904                    NaN                   NaN

                      timestamp  \
1473  2016-01-31 04:11:58 +0000
2320  2015-11-17 02:06:42 +0000
1690  2015-12-28 05:02:37 +0000
1100  2016-05-26 01:47:51 +0000
28    2017-07-16 20:14:00 +0000
1326  2016-03-05 16:24:01 +0000
559   2016-11-28 16:37:19 +0000
2071  2015-11-29 23:57:10 +0000
802   2016-09-03 22:02:38 +0000
2125  2015-11-27 22:01:40 +0000

                                                 source  \
1473  <a href="http://twitter.com/download/iphone" r...
2320  <a href="http://twitter.com/download/iphone" r...
1690  <a href="http://twitter.com/download/iphone" r...
1100  <a href="http://twitter.com/download/iphone" r...
28    <a href="http://twitter.com/download/iphone" r...
1326  <a href="http://vine.co" rel="nofollow">Vine -...
559   <a href="http://twitter.com/download/iphone" r...
2071  <a href="http://twitter.com/download/iphone" r...
802   <a href="http://twitter.com/download/iphone" r...
2125  <a href="http://twitter.com/download/iphone" r...

                                                   text  retweeted_status_id  \
1473  What kind of person sends in a pic without a d...                  NaN
2320  Here we see a lone northeastern Cumberbatch. H...                  NaN
1690  This is Asher. He's not wearing a seatbelt or ...                  NaN
1100  *faints* 12/10 perfection in pupper form https...                  NaN
28    This is Derek. He's late for a dog meeting. 13...                  NaN
1326  This is Layla. She's giving you a standing ova...                  NaN
559   This is Winston. His selfie game is legendary...                 NaN
2071  Meet Phred. He isn't steering, looking at the ...                  NaN
802   This is Willem. He's a Penn State pupper. Thin...                  NaN
2125  This is a Rich Mahogany Seltzer named Cherokee...                  NaN

      retweeted_status_user_id retweeted_status_timestamp  \
1473                       NaN                        NaN
```

```
2320                         NaN                           NaN
1690                         NaN                           NaN
1100                         NaN                           NaN
28                           NaN                           NaN
1326                         NaN                           NaN
559                          NaN                           NaN
2071                         NaN                           NaN
802                          NaN                           NaN
2125                         NaN                           NaN

                                            expanded_urls  rating_numerator  \
1473  https://twitter.com/dog_rates/status/693647888...                 7
2320  https://twitter.com/dog_rates/status/666437273...                 7
1690  https://twitter.com/dog_rates/status/681339448...                 9
1100  https://twitter.com/dog_rates/status/735648611...                12
28    https://twitter.com/dog_rates/status/886680336...                13
1326                     https://vine.co/v/iXidJXBJ3P9                13
559   https://twitter.com/dog_rates/status/803276597...                11
2071  https://twitter.com/dog_rates/status/671115716...                 6
802   https://twitter.com/dog_rates/status/772193107...                12
2125  https://twitter.com/dog_rates/status/670361874...                 9

      rating_denominator     name doggo floofer  pupper puppo
1473                  10     None  None    None    None  None
2320                  10     None  None    None    None  None
1690                  10    Asher  None    None    None  None
1100                  10     None  None    None  pupper  None
28                    10    Derek  None    None    None  None
1326                  10    Layla  None    None    None  None
559                   10  Winston  None    None    None  None
2071                  10    Phred  None    None    None  None
802                   10   Willem  None    None  pupper  None
2125                  10        a  None    None    None  None
```

```
In [9]: df_twarchive.source.value_counts()
```

```
Out[9]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>     2
        <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
        <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
        <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
        Name: source, dtype: int64
```

```
In [10]: #check the latest tweet
         print(df_twarchive.timestamp.min())
         print(df_twarchive.timestamp.max())
```

```
2015-11-15 22:32:08 +0000
2017-08-01 16:23:56 +0000
```

```
In [11]: df_twarchive.describe()

Out[11]:            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 [12]:  df_twarchive.query('rating_numerator == 0 | rating_denominator == 0')
         #835246439529840640 rating has been changed to from 960/0 to 13/10 according to current
         #835152434251116546 rating has been changed to from 0/10 to 11/10 according to current
         #746906459439529985 no rating according to current twitter post -> filter out

Out[12]:                  tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         313   835246439529840640           8.352460e+17         2.625958e+07
         315   835152434251116546                    NaN                  NaN
         1016  746906459439529985           7.468859e+17         4.196984e+09

                               timestamp  \
         313   2017-02-24 21:54:03 +0000
         315   2017-02-24 15:40:31 +0000
         1016  2016-06-26 03:22:31 +0000

                                                                   source  \
313   <a href="http://twitter.com/download/iphone" r...
315   <a href="http://twitter.com/download/iphone" r...
1016  <a href="http://twitter.com/download/iphone" r...

                                                   text  retweeted_status_id  \
313   @jonnysun @Lin_Manuel ok jomny I know you're e...                  NaN
315   When you're so blinded by your systematic plag...                  NaN
1016  PUPDATE: can't see any. Even if I could, I cou...                  NaN

      retweeted_status_user_id retweeted_status_timestamp  \
313                        NaN                        NaN
315                        NaN                        NaN
1016                       NaN                        NaN

                                          expanded_urls  rating_numerator  \
313                                                 NaN               960
315   https://twitter.com/dog_rates/status/835152434...                 0
1016  https://twitter.com/dog_rates/status/746906459...                 0

      rating_denominator  name doggo floofer pupper puppo
313                    0  None  None    None   None  None
315                   10  None  None    None   None  None
1016                  10  None  None    None   None  None
```

```
In [13]: df_twarchive[['tweet_id','rating_numerator','rating_denominator']].query('rating_denomi
```

```
Out[13]:                 tweet_id  rating_numerator  rating_denominator
         313   835246439529840640               960                   0
         342   832088576586297345                11                  15
         433   820690176645140481                84                  70
         516   810984652412424192                24                   7
         784   775096608509886464                 9                  11
         902   758467244762497024               165                 150
         1068  740373189193256964                 9                  11
         1120  731156023742988288               204                 170
         1165  722974582966214656                 4                  20
         1202  716439118184652801                50                  50
         1228  713900603437621249                99                  90
         1254  710658690886586372                80                  80
         1274  709198395643068416                45                  50
         1351  704054845121142784                60                  50
         1433  697463031882764288                44                  40
         1598  686035780142297088                 4                  20
         1634  684225744407494656               143                 130
         1635  684222868335505415               121                 110
         1662  682962037429899265                 7                  11
         1663  682808988178739200                20                  16
         1779  677716515794329600               144                 120
```

```
        1843  675853064436391936                 88                  80
        2335  666287406224695296                  1                   2
```

```
In [14]: df = df_twarchive[['tweet_id','rating_numerator','rating_denominator']].query('rating_d
         df.rating_numerator.value_counts()
         df.query('rating_numerator > 15')
```

```
Out[14]:                 tweet_id  rating_numerator  rating_denominator
         55    881633300179243008                17                  10
         188   855862651834028034               420                  10
         189   855860136149123072               666                  10
         290   838150277551247360               182                  10
         340   832215909146226688                75                  10
         695   786709082849828864                75                  10
         763   778027034220126208                27                  10
         979   749981277374128128              1776                  10
         1712  680494726643068929                26                  10
         2074  670842764863651840               420                  10
```

```
In [15]: #check if categories are clean
         categories = ['doggo', 'floofer', 'pupper', 'puppo']
         for category in categories:
             print(df_twarchive[category].value_counts())
             print()
```

```
None     2259
doggo      97
Name: doggo, dtype: int64

None       2346
floofer      10
Name: floofer, dtype: int64

None      2099
pupper     257
Name: pupper, dtype: int64

None     2326
puppo      30
Name: puppo, dtype: int64
```

```
In [20]: df_twarchive.name.value_counts().head(20)
         #convert to Null: a, the, an
```

```
Out[20]: None       745
         a           55
         Charlie     12
```

```
Cooper      11
Oliver      11
Lucy        11
Tucker      10
Lola        10
Penny       10
Winston      9
Bo           9
the          8
Sadie        8
Bailey       7
an           7
Daisy        7
Toby         7
Buddy        7
Oscar        6
Koda         6
Name: name, dtype: int64
```

The tweet image predictions, i.e., what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network.

Example for the data set: - tweet_id is the last part of the tweet URL after "status/" https://twitter.com/dog_rates/status/889531135344209921 - p1 is the algorithm's #1 prediction for the image in the tweet golden retriever - p1_conf is how confident the algorithm is in its #1 prediction 95% - p1_dog is whether or not the #1 prediction is a breed of dog TRUE - p2 is the algorithm's second most likely prediction Labrador retriever - p2_conf is how confident the algorithm is in its #2 prediction 1% - p2_dog is whether or not the #2 prediction is a breed of dog TRUE - etc.

quality issues

- we have 66 duplicates and have to make sure they don't exists in the final data set

tidiyness issues

2. get the best true dog prediction in one column
3. make the dog prediction categorial

```
In [23]: df_image.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null int64
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
p1_dog      2075 non-null bool
```

```
p2          2075 non-null object
p2_conf     2075 non-null float64
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

```
In [24]: df_image.sample(5)
```

```
Out[24]:                 tweet_id                                            jpg_url  \
         250   670691627984359425  https://pbs.twimg.com/media/CU7GehOUYAA9nn-.jpg
         1128  727685679342333952  https://pbs.twimg.com/media/ChlCQg-VIAQ_8g4.jpg
         1335  758041019896193024  https://pbs.twimg.com/media/CoUaSKEXYAAYsAl.jpg
         280   671115716440031232  https://pbs.twimg.com/media/CVBILUgVAAA1ZUr.jpg
         858   696886256886657024  https://pbs.twimg.com/media/CavWWdFWAAArflW.jpg

               img_num                  p1   p1_conf  p1_dog               p2  \
         250         1  Shetland_sheepdog  0.071124    True     home_theater
         1128        1      Border_collie  0.462408    True           collie
         1335        1           bookshop  0.794272   False         Cardigan
         280         1           malinois  0.406341    True           kelpie
         858         1             kuvasz  0.383941    True  golden_retriever

                p2_conf  p2_dog                              p3   p3_conf  p3_dog
         250   0.068398   False  American_Staffordshire_terrier  0.066964    True
         1128  0.214556    True                      Eskimo_dog  0.035604    True
         1335  0.051265    True            Bernese_mountain_dog  0.026596    True
         280   0.143366    True                           dingo  0.129802   False
         858   0.289085    True                           dingo  0.056548   False
```

```
In [25]: df_image.describe()
```

```
Out[25]:           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 [50]: df_image[df_image.duplicated(subset=['jpg_url'], keep='first')].head(5)
```

```
        ---------------------------------------------------------------------------
        NameError                                 Traceback (most recent call last)

        <ipython-input-50-6f41e28b1ff4> in <module>()
    ----> 1 df_image[df_image.duplicated(subset=['jpg_url'], keep='first')].head(5)


        NameError: name 'df_image' is not defined
```

1.2.2 assess tweet_json.txt

To assess the JSON object I refer to - https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object or - http://support.gnip.com/sources/twitter/data_format.html with more detail on - http://support.gnip.com/doing-more-with-140.html

tweet ID, retweet count, and favorite count

```
In [27]: #read json file line by line
         df_list=[]
         with open(json_filename, 'r') as file:
             for row in file:
                 data = json.loads(row)
                 df_list.append({"tweet_id":data["id"],
                                 "retweet_count":data["retweet_count"],
                                 "favorite_count":data["favorite_count"]
                                 })
         #df_tweet = pd.DataFrame.from_dict(df_list)
         df_tweet = pd.DataFrame(df_list, columns=["tweet_id", "retweet_count", "favorite_count"
```

```
In [28]: df_tweet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2345 entries, 0 to 2344
Data columns (total 3 columns):
tweet_id          2345 non-null int64
retweet_count     2345 non-null int64
favorite_count    2345 non-null int64
dtypes: int64(3)
memory usage: 55.0 KB
```

```
In [29]: df_tweet.describe()
```

```
Out[29]:            tweet_id  retweet_count  favorite_count
         count  2.345000e+03    2345.000000     2345.000000
         mean   7.423760e+17    3078.081876     8114.238380
         std    6.836820e+16    5105.627191    12189.503552
         min    6.660209e+17       0.000000        0.000000
         25%    6.783802e+17     614.000000     1410.000000
```

```
50%    7.189719e+17    1437.000000     3574.000000
75%    7.987057e+17    3585.000000    10068.000000
max    8.924206e+17   78450.000000   142572.000000
```

```
In [30]: df_tweet[df_tweet.retweet_count>70000]
```

```
Out[30]:                 tweet_id  retweet_count  favorite_count
         1029  744234799360020481          78450          129628
```

```
In [32]: df_twarchive[df_twarchive.tweet_id == 744234799360020481]
```

```
Out[32]:                 tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         1039  744234799360020481                    NaN                  NaN

                              timestamp  \
         1039  2016-06-18 18:26:18 +0000

                                                    source  \
         1039  <a href="http://twitter.com/download/iphone" r...

                                                      text  retweeted_status_id  \
         1039  Here's a doggo realizing you can stand in a po...                  NaN

               retweeted_status_user_id retweeted_status_timestamp  \
         1039                       NaN                        NaN

                                                 expanded_urls  rating_numerator  \
         1039  https://twitter.com/dog_rates/status/744234799...                13

               rating_denominator  name  doggo floofer pupper puppo
         1039                  10  None  doggo    None   None  None
```

1.3 Assess new file twitter_archive_master.csv

tidyness issues

- merge all three dataset into one
- drop columns in the merged data set which are not required

1.4 Clean

Clean each of the issues you documented while assessing. Perform this cleaning in wrangle_act.ipynb as well. The result should be a high quality and tidy master pandas DataFrame (or DataFrames, if appropriate). Again, **the issues that satisfy the Project Motivation must be cleaned**.

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

```
In [34]: #preserve the raw data
         dfc_twarchive = df_twarchive.copy()
```

Define

- generate mapping dictionary to map to a short description with clear clear categories iPhone App, vine.co, Twitter Web Client, TweetDeck
- create function to change source
- call function
- change datatype to categorial

Code

```
In [35]: dfc_twarchive.source.value_counts()
```

```
Out[35]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
         <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
         <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
         <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
         Name: source, dtype: int64
```

```
In [36]: ref_source = {'<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for
                       '<a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>': 'vine.
                       '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>': 'Tw
                       '<a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">Tw

         def clean_source(tweet):
             if tweet['source'] in ref_source.keys():
                 #return clean description
                 return ref_source[tweet['source']]
             else:
                 return dfc_twarchive.source

         dfc_twarchive.source = dfc_twarchive.apply(clean_source, axis=1)

         dfc_twarchive.source = dfc_twarchive.source.astype('category')
```

Test

```
In [37]: dfc_twarchive.source.value_counts()
```

```
Out[37]: iPhone App            2221
         vine.co                 91
         Twitter Web Client      33
         TweetDeck               11
         Name: source, dtype: int64
```

```
In [38]: dfc_twarchive.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 category
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: category(1), float64(4), int64(3), object(9)
memory usage: 297.1+ KB
```

Define

- deal with a few invalid ratings where rating_numerator is 0 or rating_denominator is 0
  - 835246439529840640 rating has been changed to from 960/0 to 13/10 according to current twitter post
  - 835152434251116546 rating has been changed to from 0/10 to 11/10 according to current twitter post
  - 746906459439529985 no rating according to current twitter post -> filter out

Code

```
In [39]: #update df cell
         #see https://stackoverflow.com/questions/12307099/modifying-a-subset-of-rows-in-a-panda
         dfc_twarchive.loc[dfc_twarchive.tweet_id == 835246439529840640, 'rating_numerator'] = 1
         dfc_twarchive.loc[dfc_twarchive.tweet_id == 835246439529840640, 'rating_denominator'] =

         dfc_twarchive.loc[dfc_twarchive.tweet_id == 835152434251116546, 'rating_numerator'] = 1

         dfc_twarchive = dfc_twarchive[dfc_twarchive.tweet_id != 746906459439529985]
```

Test

```
In [40]: dfc_twarchive.loc[dfc_twarchive.tweet_id.isin([835246439529840640, 835152434251116546,

Out[40]:                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         313  835246439529840640           8.352460e+17           26259576.0
         315  835152434251116546                    NaN                  NaN
```

```
                     timestamp      source  \
313  2017-02-24 21:54:03 +0000  iPhone App
315  2017-02-24 15:40:31 +0000  iPhone App

                                                  text  retweeted_status_id  \
313  @jonnysun @Lin_Manuel ok jomny I know you're e...                  NaN
315  When you're so blinded by your systematic plag...                  NaN

     retweeted_status_user_id retweeted_status_timestamp  \
313                       NaN                        NaN
315                       NaN                        NaN

                                         expanded_urls  rating_numerator  \
313                                                NaN                13
315  https://twitter.com/dog_rates/status/835152434...                11

     rating_denominator  name doggo floofer pupper puppo
313                  10  None  None    None   None  None
315                  10  None  None    None   None  None
```

Define

- take out ratings with a rating_denominator other than 10 as these are invalid ratings or rate multiple dogs at once.
- take out ratings with rating_numerator > 15 they seem to be invalid
- take out retweets You only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets. So **keep** all tweets with retweeted_status_id is NaN

Code

```
In [41]: #delete retweets
         dfc_twarchive = dfc_twarchive[dfc_twarchive.retweeted_status_id.isnull()]

In [42]: dfc_twarchive = dfc_twarchive[dfc_twarchive.rating_denominator == 10]
         dfc_twarchive = dfc_twarchive[dfc_twarchive.rating_numerator <= 15]
```

Test

```
In [43]: dfc_twarchive.describe()

Out[43]:            tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         count  2.144000e+03           6.900000e+01         6.900000e+01
         mean   7.370728e+17           7.403786e+17         2.276889e+16
         std    6.754280e+16           7.383460e+16         1.330848e+17
         min    6.660209e+17           6.658147e+17         1.185634e+07
         25%    6.766166e+17           6.753494e+17         1.198989e+09
         50%    7.095381e+17           7.030419e+17         4.196984e+09
```

```
75%    7.895480e+17           8.131273e+17         4.196984e+09
max    8.924206e+17           8.862664e+17         8.405479e+17

       retweeted_status_id  retweeted_status_user_id  rating_numerator  \
count                  0.0                       0.0       2144.000000
mean                   NaN                       NaN         10.630131
std                    NaN                       NaN          2.171990
min                    NaN                       NaN          1.000000
25%                    NaN                       NaN         10.000000
50%                    NaN                       NaN         11.000000
75%                    NaN                       NaN         12.000000
max                    NaN                       NaN         15.000000

       rating_denominator
count              2144.0
mean                 10.0
std                   0.0
min                  10.0
25%                  10.0
50%                  10.0
75%                  10.0
max                  10.0
```

Define

- convert wrong dog names to Null: "a", "the", "an", "n"

Code

```
In [44]: # Deal with value=None
         # see https://stackoverflow.com/questions/17097236/how-to-replace-values-with-none-in-p
         dfc_twarchive.name = dfc_twarchive.name.replace(to_replace=['^(a|an|the|n|None)$'], val
```

Test

```
In [45]: dfc_twarchive.loc[dfc_twarchive.name.isin(['a', 'an', 'the', 'n', 'None'])].name.count(

Out[45]: 0
```

Define

tidyness issues

- four columns for the dog_stage can be put into one categorial variable (doggo,floofer,pupper,puppo) and turn the default value "None" into a Null or "unknown"

Code

```
In [46]: dfc_twarchive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2144 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2144 non-null int64
in_reply_to_status_id         69 non-null float64
in_reply_to_user_id           69 non-null float64
timestamp                     2144 non-null object
source                        2144 non-null category
text                          2144 non-null object
retweeted_status_id           0 non-null float64
retweeted_status_user_id      0 non-null float64
retweeted_status_timestamp    0 non-null object
expanded_urls                 2093 non-null object
rating_numerator              2144 non-null int64
rating_denominator            2144 non-null int64
name                          1419 non-null object
doggo                         2144 non-null object
floofer                       2144 non-null object
pupper                        2144 non-null object
puppo                         2144 non-null object
dtypes: category(1), float64(4), int64(3), object(9)
memory usage: 287.0+ KB
```

```
In [47]: #concat the stage columns doggo,floofer,pupper,puppo while replacing the None wordw wit
         dfc_twarchive['dog_stage'] = dfc_twarchive.doggo.replace('None','') + dfc_twarchive.flo
         dfc_twarchive['dog_stage'] = dfc_twarchive['dog_stage'].replace(to_replace=[''], value=
```

```
In [48]: dfc_twarchive['dog_stage'].value_counts()
```

```
Out[48]: pupper          223
         doggo            75
         puppo            24
         doggopupper      10
         floofer           9
         doggopuppo        1
         doggofloofer      1
         Name: dog_stage, dtype: int64
```

```
In [49]: # get rid of dog stages which are not defined
         # https://stackoverflow.com/questions/23330654/update-a-dataframe-in-pandas-while-itera
         for row in dfc_twarchive.itertuples():
             if dfc_twarchive.loc[row.Index, 'dog_stage'] in ['doggo','floofer','pupper','puppo'
                 next
             else:
                 dfc_twarchive.loc[row.Index, 'dog_stage'] = None
         dfc_twarchive['dog_stage'].value_counts()
```

```
Out[49]: pupper     223
         doggo       75
         puppo       24
         floofer      9
         Name: dog_stage, dtype: int64
```

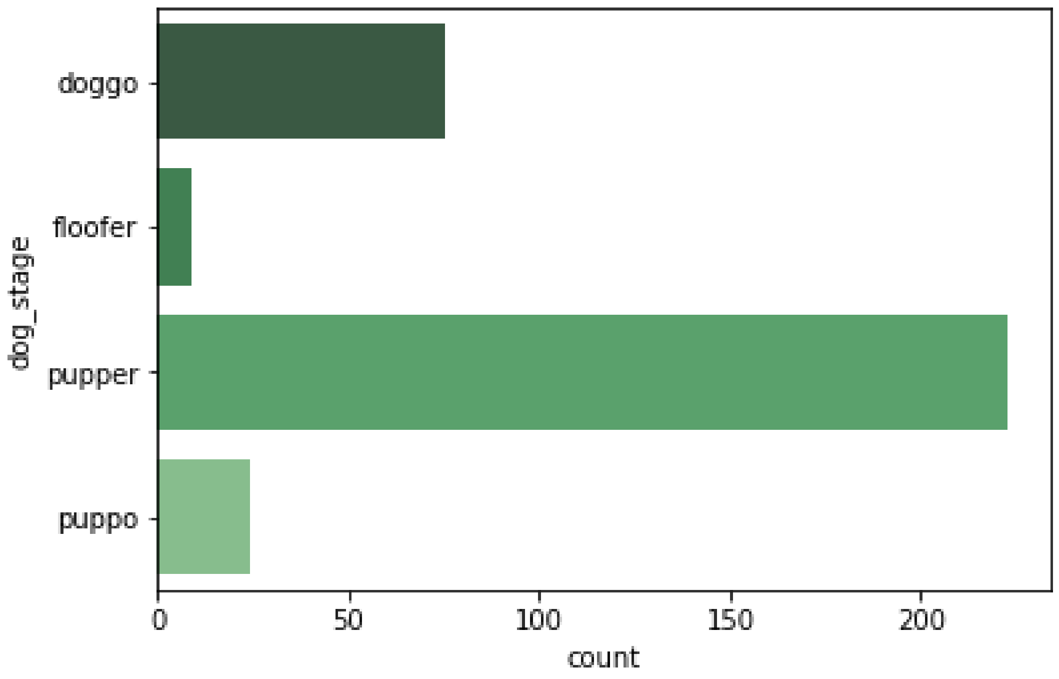
```
In [50]: dfc_twarchive['dog_stage'] = dfc_twarchive['dog_stage'].astype('category')
```

Test

```
In [51]: dfc_twarchive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2144 entries, 0 to 2355
Data columns (total 18 columns):
tweet_id                      2144 non-null int64
in_reply_to_status_id         69 non-null float64
in_reply_to_user_id           69 non-null float64
timestamp                     2144 non-null object
source                        2144 non-null category
text                          2144 non-null object
retweeted_status_id           0 non-null float64
retweeted_status_user_id      0 non-null float64
retweeted_status_timestamp    0 non-null object
expanded_urls                 2093 non-null object
rating_numerator              2144 non-null int64
rating_denominator            2144 non-null int64
name                          1419 non-null object
doggo                         2144 non-null object
floofer                       2144 non-null object
pupper                        2144 non-null object
puppo                         2144 non-null object
dog_stage                     331 non-null category
dtypes: category(2), float64(4), int64(3), object(9)
memory usage: 369.3+ KB
```

```
In [52]: sns.countplot(y="dog_stage", data=dfc_twarchive, palette="Greens_d");
```



Define drop all irrelevant columns

Code

```
In [53]: dfc_twarchive.drop(['in_reply_to_status_id',
                             'in_reply_to_user_id',
                             'retweeted_status_id',
                             'retweeted_status_user_id',
                             'retweeted_status_timestamp',
                             'doggo',
                             'floofer',
                             'pupper',
                             'puppo'], axis=1, inplace=True)
```

Test

```
In [54]: dfc_twarchive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2144 entries, 0 to 2355
Data columns (total 9 columns):
tweet_id              2144 non-null int64
timestamp             2144 non-null object
source                2144 non-null category
```

```
text                    2144 non-null object
expanded_urls           2093 non-null object
rating_numerator        2144 non-null int64
rating_denominator      2144 non-null int64
name                    1419 non-null object
dog_stage               331 non-null category
dtypes: category(2), int64(3), object(4)
memory usage: 218.6+ KB
```

1.4.1 clean image-predictions.tsv file

```
In [55]: dfc_image = df_image.copy()
```

Define

- Go through each line and look for true dog predictions from column 1 to column 3.
- Stop at the first occurence of a dog prediction
- Write that info into a separate field
- make that the dog prediction column categorial

Code

```
In [56]: dfc_image.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null int64
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
p1_dog      2075 non-null bool
p2          2075 non-null object
p2_conf     2075 non-null float64
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

```
In [57]: #make sure the priority is correct in the columns
         dfc_image.query('p1_conf < p2_conf | p2_conf < p3_conf | p1_conf < p3_conf')

Out[57]: Empty DataFrame
         Columns: [tweet_id, jpg_url, img_num, p1, p1_conf, p1_dog, p2, p2_conf, p2_dog, p3, p3_
         Index: []
```

```
In [58]: #merge thre dog columns into one according to priority
         df_p1 = dfc_image[dfc_image.p1_dog == True].filter(items=['tweet_id','jpg_url','p1','p1
         df_p1 = df_p1.rename(columns={'p1': 'dog', 'p1_conf': 'conf'})
         df_p2 = dfc_image.query('p1_dog == False & p2_dog == True').filter(items=['tweet_id','j
         df_p2 = df_p2.rename(columns={'p2': 'dog', 'p2_conf': 'conf'})
         df_p3 = dfc_image.query('p1_dog == False & p2_dog == False & p3_dog == True').filter(it
         df_p3 = df_p3.rename(columns={'p3': 'dog', 'p3_conf': 'conf'})
         dfc_image = pd.concat([df_p1, df_p2, df_p3])
```

```
In [59]: dfc_image.dog = dfc_image.dog.astype('category')
```

Test

```
In [60]: dfc_image.head(5)

Out[60]:              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

                                dog      conf
         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
```

```
In [61]: dfc_image.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1751 entries, 0 to 2026
Data columns (total 4 columns):
tweet_id    1751 non-null int64
jpg_url     1751 non-null object
dog         1751 non-null category
conf        1751 non-null float64
dtypes: category(1), float64(1), int64(1), object(1)
memory usage: 62.3+ KB
```

```
In [62]: df_image.query('tweet_id == 666044226329800704')

Out[62]:              tweet_id                                          jpg_url  \
         3  666044226329800704  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
```

```
   img_num                   p1   p1_conf  p1_dog       p2   p2_conf  p2_dog  \
3        1  Rhodesian_ridgeback  0.408143    True  redbone  0.360687    True

                  p3   p3_conf  p3_dog
3  miniature_pinscher  0.222752    True
```

Define join together dfc_twarchive x df_tweet add image info if it exists from dfc_image

Code

```
In [63]: df_tweet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2345 entries, 0 to 2344
Data columns (total 3 columns):
tweet_id          2345 non-null int64
retweet_count     2345 non-null int64
favorite_count    2345 non-null int64
dtypes: int64(3)
memory usage: 55.0 KB
```

```
In [64]: j1 = pd.merge(dfc_twarchive, df_tweet, on='tweet_id')
         #filter out dogs where we don't know the type
         df_master = pd.merge(j1, dfc_image, on='tweet_id')
```

```
In [66]: df_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1664 entries, 0 to 1663
Data columns (total 14 columns):
tweet_id              1664 non-null int64
timestamp             1664 non-null object
source                1664 non-null category
text                  1664 non-null object
expanded_urls         1664 non-null object
rating_numerator      1664 non-null int64
rating_denominator    1664 non-null int64
name                  1203 non-null object
dog_stage             249 non-null category
retweet_count         1664 non-null int64
favorite_count        1664 non-null int64
jpg_url               1664 non-null object
dog                   1664 non-null category
conf                  1664 non-null float64
dtypes: category(3), float64(1), int64(5), object(5)
memory usage: 167.1+ KB
```

```
In [67]: df_master.sample(1)

Out[67]:                 tweet_id                  timestamp      source  \
         1468  670782429121134593  2015-11-29 01:52:48 +0000  iPhone App

                                                           text  \
         1468  This dude slaps your girl's ass what do you do...

                                                  expanded_urls  rating_numerator  \
         1468  https://twitter.com/dog_rates/status/670782429...                 5

               rating_denominator  name dog_stage  retweet_count  favorite_count  \
         1468                  10  None       NaN            847            1667

                                                   jpg_url        dog      conf
         1468  https://pbs.twimg.com/media/CU8ZDu9WwAADg3N.jpg  Chihuahua  0.952963

In [68]: df_master[df_master.jpg_url.isnull()].count()

Out[68]: tweet_id              0
         timestamp             0
         source                0
         text                  0
         expanded_urls         0
         rating_numerator      0
         rating_denominator    0
         name                  0
         dog_stage             0
         retweet_count         0
         favorite_count        0
         jpg_url               0
         dog                   0
         conf                  0
         dtype: int64
```

Test

```
In [70]: #proove that there is no duplicate jpg anymore.
         df_master[df_master.duplicated(subset=['jpg_url'], keep='first')]

Out[70]: Empty DataFrame
         Columns: [tweet_id, timestamp, source, text, expanded_urls, rating_numerator, rating_de
         Index: []
```

Define set timezone to EST according to wikipedia that's where the twitter account owner studies at university

Code

```
In [74]: #convert object to datetime64
         #https://pandas.pydata.org/pandas-docs/stable/timeseries.html
         #localize to UTC and convert to EST
         #https://stackoverflow.com/questions/20689288/converting-pandas-columns-to-datetime64-i
         #df_master['timestamp'] =
         df_master.timestamp = pd.to_datetime(df_master.timestamp).dt.tz_localize('UTC').dt.tz_c
         #extract local hour
         df_master.create_HH24 = df_master.timestamp.dt.hour


        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-74-6c77b85dbeaf> in <module>()
          4 #https://stackoverflow.com/questions/20689288/converting-pandas-columns-to-datetime6
          5 #df_master['timestamp'] =
    ----> 6 df_master.timestamp = pd.to_datetime(df_master.timestamp).dt.tz_localize('UTC').dt.t
          7 #extract local hour
          8 df_master.create_HH24 = df_master.timestamp.dt.hour


        ~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\core\base.py in f(self, *ar
        209
        210             def f(self, *args, **kwargs):
    --> 211                 return self._delegate_method(name, *args, **kwargs)
        212
        213             f.__name__ = name


        ~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\core\indexes\accessors.py i
        131
        132         method = getattr(self.values, name)
    --> 133         result = method(*args, **kwargs)
        134
        135         if not is_list_like(result):


        ~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrap
         89                 else:
         90                     kwargs[new_arg_name] = new_arg_value
    ---> 91             return func(*args, **kwargs)
         92         return wrapper
         93     return _deprecate_kwarg


        ~\AppData\Local\Continuum\anaconda3\lib\site-packages\pandas\core\indexes\datetimes.py i
```

```
   1835                 new_dates = libts.tz_convert(self.asi8, 'UTC', self.tz)
   1836             else:
-> 1837                 raise TypeError("Already tz-aware, use tz_convert to convert.")
   1838         else:
   1839             tz = libts.maybe_get_tz(tz)


    TypeError: Already tz-aware, use tz_convert to convert.
```

```
In [86]: df_master['create_HH24'] = df_master.timestamp.dt.hour.astype('category')
```

Test

```
In [88]: df_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1664 entries, 0 to 1663
Data columns (total 15 columns):
tweet_id              1664 non-null int64
timestamp             1664 non-null datetime64[ns, EST]
source                1664 non-null category
text                  1664 non-null object
expanded_urls         1664 non-null object
rating_numerator      1664 non-null int64
rating_denominator    1664 non-null int64
name                  1203 non-null object
dog_stage             249 non-null category
retweet_count         1664 non-null int64
favorite_count        1664 non-null int64
jpg_url               1664 non-null object
dog                   1664 non-null category
conf                  1664 non-null float64
create_HH24           1664 non-null category
dtypes: category(4), datetime64[ns, EST](1), float64(1), int64(5), object(4)
memory usage: 249.5+ KB
```

1.5 Storing, Analyzing, and Visualizing Data for this Project

Store the clean DataFrame(s) in a CSV file with the main one named **twitter_archive_master.csv**. If additional files exist because multiple tables are required for tidiness, name these files appropriately. Additionally, you may **store the cleaned data in a SQLite database** (which is to be submitted as well if you do).

Analyze and visualize your wrangled data in your wrangle_act.ipynb Jupyter Notebook. At least **three (3) insights and one (1) visualization** must be produced.

```
In [89]: df_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1664 entries, 0 to 1663
Data columns (total 15 columns):
tweet_id              1664 non-null int64
timestamp             1664 non-null datetime64[ns, EST]
source                1664 non-null category
text                  1664 non-null object
expanded_urls         1664 non-null object
rating_numerator      1664 non-null int64
rating_denominator    1664 non-null int64
name                  1203 non-null object
dog_stage             249 non-null category
retweet_count         1664 non-null int64
favorite_count        1664 non-null int64
jpg_url               1664 non-null object
dog                   1664 non-null category
conf                  1664 non-null float64
create_HH24           1664 non-null category
dtypes: category(4), datetime64[ns, EST](1), float64(1), int64(5), object(4)
memory usage: 249.5+ KB
```

```
In [91]: df_master.to_csv('twitter_archive_master.csv',index=False, encoding='utf-8')
```

```
In [5]: df_master = pd.read_csv('twitter_archive_master.csv')
```

```
In [6]: df_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1664 entries, 0 to 1663
Data columns (total 15 columns):
tweet_id              1664 non-null int64
timestamp             1664 non-null object
source                1664 non-null object
text                  1664 non-null object
expanded_urls         1664 non-null object
rating_numerator      1664 non-null int64
rating_denominator    1664 non-null int64
name                  1203 non-null object
dog_stage             249 non-null object
retweet_count         1664 non-null int64
favorite_count        1664 non-null int64
jpg_url               1664 non-null object
dog                   1664 non-null object
conf                  1664 non-null float64
create_HH24           1664 non-null int64
dtypes: float64(1), int64(6), object(8)
memory usage: 195.1+ KB
```

```
In [7]: df_master.source = df_master.source.astype('category')
        df_master.dog_stage = df_master.dog_stage.astype('category')
        df_master.dog = df_master.dog.astype('category')
        df_master.create_HH24 = df_master.create_HH24.astype('category')
        df_master.info()
```

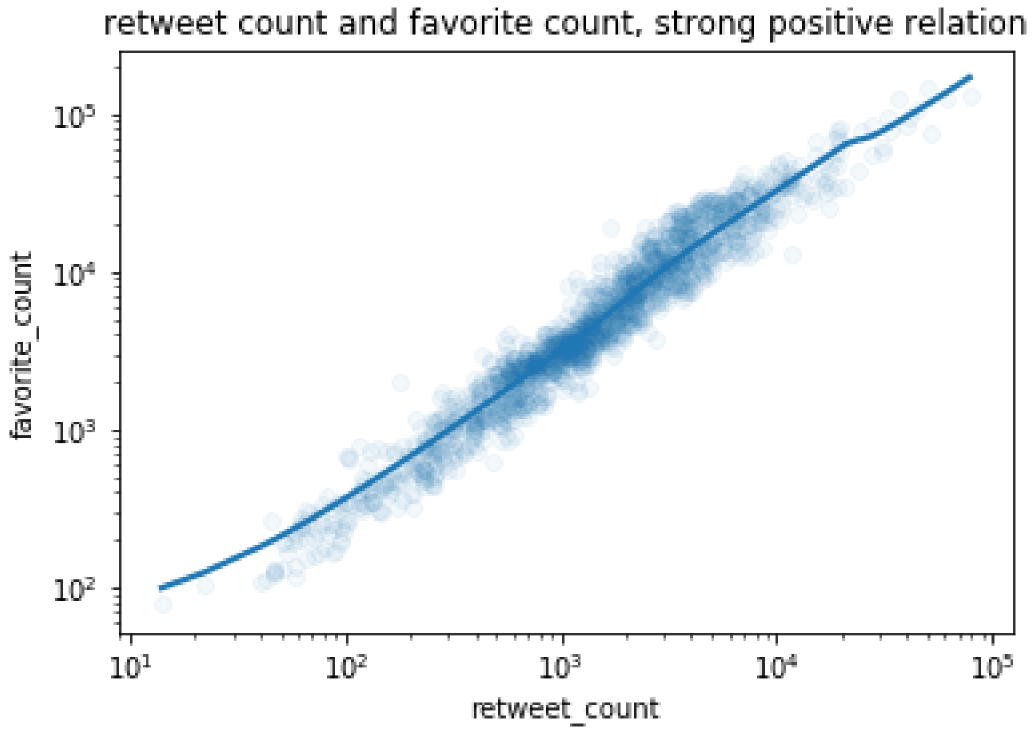
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1664 entries, 0 to 1663
Data columns (total 15 columns):
tweet_id              1664 non-null int64
timestamp             1664 non-null object
source                1664 non-null category
text                  1664 non-null object
expanded_urls         1664 non-null object
rating_numerator      1664 non-null int64
rating_denominator    1664 non-null int64
name                  1203 non-null object
dog_stage             249 non-null category
retweet_count         1664 non-null int64
favorite_count        1664 non-null int64
jpg_url               1664 non-null object
dog                   1664 non-null category
conf                  1664 non-null float64
create_HH24           1664 non-null category
dtypes: category(4), float64(1), int64(5), object(5)
memory usage: 156.5+ KB
```

```
In [8]: df_master.corr(method='pearson')
```

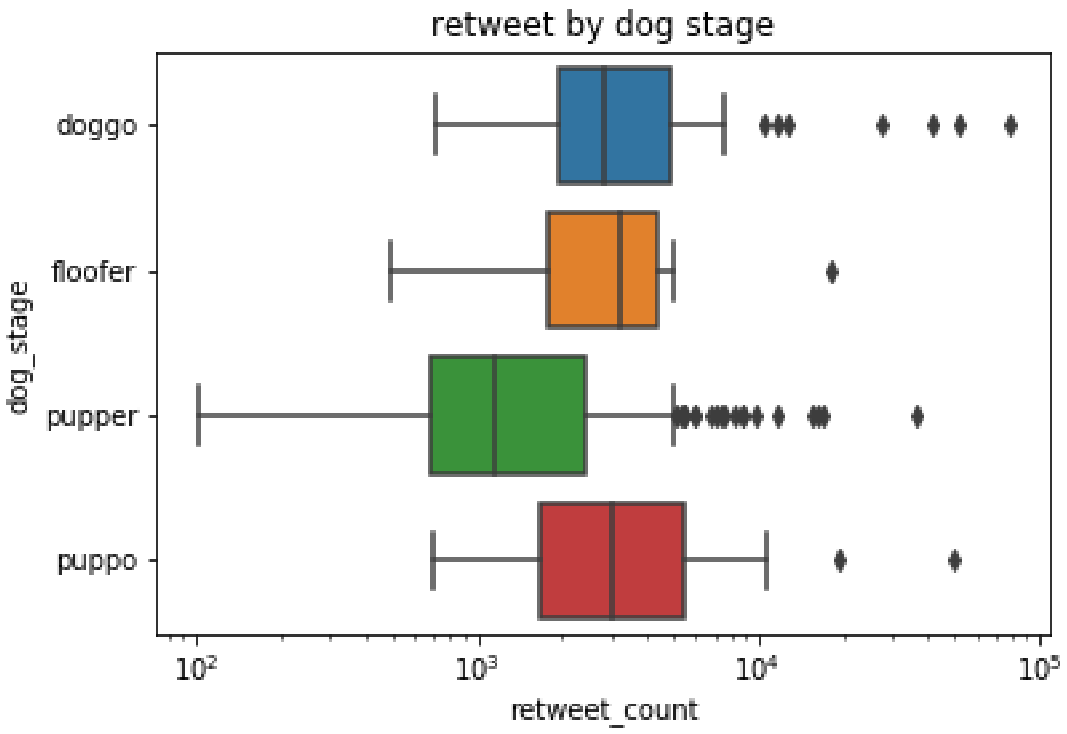
```
Out[8]:                     tweet_id  rating_numerator  rating_denominator  \
        tweet_id            1.000000          0.550155                 NaN
        rating_numerator    0.550155          1.000000                 NaN
        rating_denominator       NaN               NaN                 NaN
        retweet_count       0.392921          0.317499                 NaN
        favorite_count      0.630534          0.420476                 NaN
        conf                0.103490          0.142090                 NaN

                            retweet_count  favorite_count      conf
        tweet_id                 0.392921        0.630534  0.103490
        rating_numerator         0.317499        0.420476  0.142090
        rating_denominator            NaN             NaN       NaN
        retweet_count            1.000000        0.917411  0.027693
        favorite_count           0.917411        1.000000  0.059848
        conf                     0.027693        0.059848  1.000000
```

```
In [9]: # Initialize figure and ax
        fig, ax = plt.subplots()
```

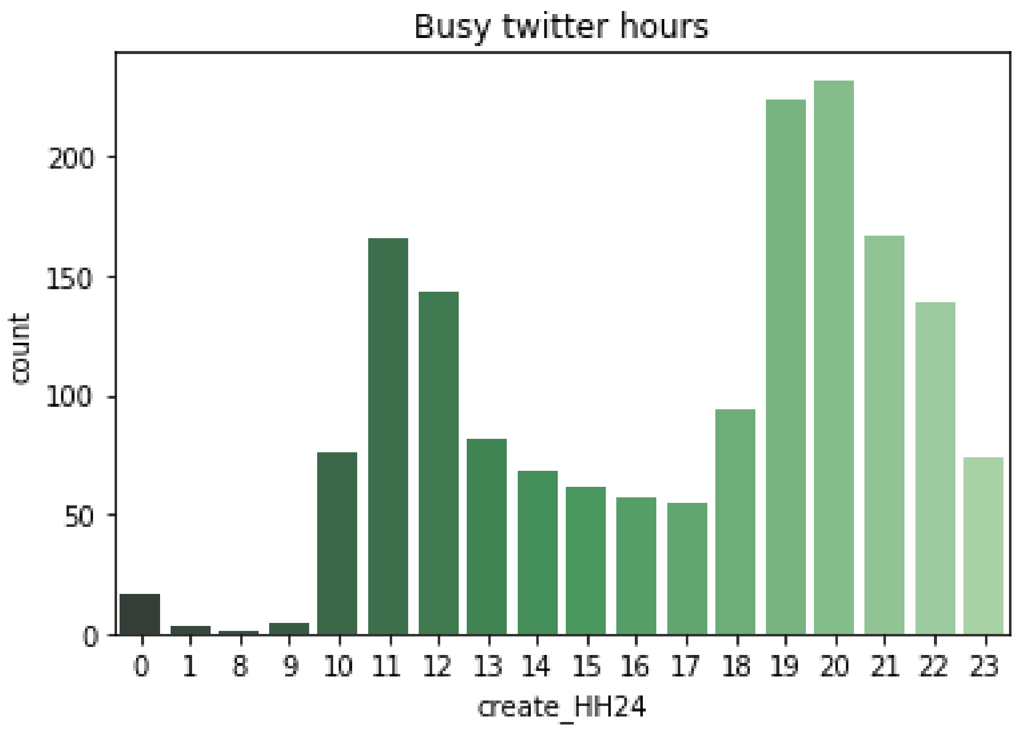
```
# Set the scale of the x-and y-axes
ax.set(xscale="log", yscale="log")
sns.regplot(x='retweet_count', y='favorite_count', data=df_master, ax=ax, scatter_kws={'
plt.title('retweet count and favorite count, strong positive relation')
plt.show()
```



```
In [10]: # Create the boxplot
         ax = sns.boxplot(x="retweet_count", y="dog_stage", data=df_master)
         #ax = sns.boxplot(x="retweet_count", y="dog_stage", data=df_master)
         #ax = sns.boxplot(x="rating_numerator", y="dog_stage", data=df_master)
         # Initialize figure and ax
         #fig, ax = plt.subplots()

         # Set the `xlim`
         ax.set(xscale="log")

         # Set title
         ax.set_title("retweet by dog stage")

         # Show the plot
         plt.show()
```



```
In [40]: #students work hard, ... after 10 in the morning.
         ax = sns.countplot(x="create_HH24", data=df_master, palette="Greens_d")
         ax.set_title('Busy twitter hours')
         plt.show()
```



```
In [41]: pysqldf = lambda q: sqldf(q, globals())

         q = """
         SELECT
          dog
         ,count(*) as cnt
         ,avg(rating_numerator) as avg_rating_numerator
         ,avg(retweet_count) as avg_retweet_count
         ,avg(favorite_count) as avg_favorite_count
         FROM df_master
         GROUP BY 1
         having cnt>10
         ORDER BY avg_favorite_count desc
         ;
         """

In [42]: df_dog = pysqldf(q)
         df_dog.head(10)

Out[42]:                   dog  cnt  avg_rating_numerator  avg_retweet_count  \
         0      French_bulldog   31             11.193548        5179.774194
         1             whippet   11             10.181818        4995.909091
         2    standard_poodle    11             10.727273        5425.818182
         3            Samoyed   42             11.690476        4763.714286
```