wrangle_act

January 26, 2022

1 Project: Wrangling and Analyze Data

1.1 Data Gathering

In the cell below, gather **all** three pieces of data for this project and load them in the notebook. **Note:** the methods required to gather each data are different. 1. Directly download the WeRate-Dogs Twitter archive data (twitter_archive_enhanced.csv)

In [2]:
```
import numpy as np
import pandas as pd
import requests
import tweepy
from tweepy import OAuthHandler
import json
from timeit import default_timer as timer
import math
```

In [3]:
```
twitter_archive= pd.read_csv('twitter-archive-enhanced.csv')
twitter_archive.sample()
```

Out[3]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweeted_status_user_id	retweeted_status_timestamp	expanded_urls	rating_numerator
1989	672828000000000000	NaN	NaN	2015-12-04 17:23:04 +0000	<a href="http://twitter.com/download/iphone" r...	This is Jerry. He's a Timbuk Slytherin. Eats h...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/672828477...	9

```
      rating_denominator   name doggo floofer pupper puppo
1989                  10  Jerry  None    None   None  None
```

```
In [4]: df_archive = twitter_archive.copy()
```

2. Use the Requests library to download the tweet image prediction (image_predictions.tsv)

```
In [5]: #Downloading Tweet image predictions:
        predictions_url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_ima
        image_request = requests.get(predictions_url, allow_redirects=True)

        open('image_predictions.tsv', 'wb').write(image_request.content)
```

```
Out[5]: 335079
```

```
In [6]: #Diplaying data in the image predictions :
        df_image_predictions = pd.read_csv('image_predictions.tsv', sep = '\t')
        df_image_predictions.head()
        df_image=df_image_predictions.copy()
        df_image.head()
```

```
Out[6]:              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
```

3. Use the Tweepy library to query additional data via the Twitter API (tweet_json.txt)

```
In [7]: consumer_key = 'API key'
        consumer_secret = 'API key secret'
        access_token = 'Access token'
        access_secret = 'Access token secret'
```

```
        # Queried each tweets re-tweet:
        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]: tweet_ids = twitter_archive.tweet_id.values
        len(tweet_ids)
```

```
Out[8]: 2356
```

```
In [9]: df_list = []
        with open('tweet_json.txt', 'r') as json_f:
            tweets_info = pd.DataFrame(columns = ['tweet_id','favorites','retweets'])

            for info in json_f:
                tweets = json.loads(info)


                data = {'tweet_id': tweets['id'],
                        'favorites': tweets['favorite_count'],
                        'retweets': tweets['retweet_count']}
                ser = pd.Series(data)
                tweets_info = tweets_info.append(data,ignore_index=True)
        tweets_info.head()
        df_tweets=tweets_info.copy()
        df_tweets.head()
```

```
Out[9]:              tweet_id favorites retweets
        0  892420643555336193     39467     8853
        1  892177421306343426     33819     6514
        2  891815181378084864     25461     4328
        3  891689557279858688     42908     8964
        4  891327558926688256     41048     9774
```

1.2 Assessing Data

In this section, detect and document at least **eight (8) quality issues and two (2) tidiness issue**. You must use **both** visual assessment programmatic assessement to assess the data.

Note: pay attention to the following key points when you access the data.

- 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.
- Assessing and cleaning the entire dataset completely would require a lot of time, and is not necessary to practice and demonstrate your skills in data wrangling. Therefore, the requirements of this project are only to assess and clean at least 8 quality issues and at least 2 tidiness issues in this dataset.
- The fact that the rating numerators are greater than the denominators does not need to be cleaned. This unique rating system is a big part of the popularity of WeRateDogs.

- You do not need to gather the tweets beyond August 1st, 2017. You can, but note that you won't be able to gather the image predictions for these tweets since you don't have access to the algorithm used.

```
In [10]: df_archive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                    2356 non-null int64
in_reply_to_status_id       78 non-null float64
in_reply_to_user_id         78 non-null float64
timestamp                   2356 non-null object
source                      2356 non-null object
text                        2356 non-null object
retweeted_status_id         181 non-null float64
retweeted_status_user_id    181 non-null float64
retweeted_status_timestamp  181 non-null object
expanded_urls               2297 non-null object
rating_numerator            2356 non-null int64
rating_denominator          2356 non-null int64
name                        2356 non-null object
doggo                       2356 non-null object
floofer                     2356 non-null object
pupper                      2356 non-null object
puppo                       2356 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

```
In [11]: df_archive.sample(5)
```

```
Out[11]:              tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         1867  675334000000000000                    NaN                  NaN
         905   758100000000000000                    NaN                  NaN
         303   836398000000000000                    NaN                  NaN
         933   753656000000000000                    NaN                  NaN
         540   806542000000000000                    NaN                  NaN

                           timestamp  \
         1867  2015-12-11 15:19:21 +0000
         905   2016-07-27 00:40:12 +0000
         303   2017-02-28 02:09:08 +0000
         933   2016-07-14 18:22:23 +0000
         540   2016-12-07 16:53:43 +0000

                                                    source  \
         1867  <a href="http://twitter.com/download/iphone" r...
         905   <a href="http://vine.co" rel="nofollow">Vine -...
```

```
303   <a href="http://twitter.com/download/iphone" r...
933   <a href="http://twitter.com/download/iphone" r...
540   <a href="http://twitter.com/download/iphone" r...

                                                  text  retweeted_status_id  \
1867  Good morning here's a grass pupper. 12/10 http...                  NaN
905   In case you haven't seen the most dramatic sne...                  NaN
303   RT @dog_rates: This is Buddy. He ran into a gl...         8.180000e+17
933   "The dogtor is in hahahaha no but seriously I'...                  NaN
540   This is Waffles. He's concerned that the dandr...                  NaN

      retweeted_status_user_id retweeted_status_timestamp  \
1867                       NaN                        NaN
905                        NaN                        NaN
303               4.196984e+09  2017-01-07 20:18:46 +0000
933                        NaN                        NaN
540                        NaN                        NaN

                                          expanded_urls  rating_numerator  \
1867  https://twitter.com/dog_rates/status/675334060...                12
905                       https://vine.co/v/hQJbaj1VpIz                13
303   https://twitter.com/dog_rates/status/817827839...                13
933   https://twitter.com/dog_rates/status/753655901...                10
540   https://twitter.com/dog_rates/status/806542213...                11

      rating_denominator     name doggo floofer  pupper puppo
1867                  10     None  None    None  pupper  None
905                   10     None  None    None    None  None
303                   10    Buddy  None    None    None  None
933                   10     None  None    None    None  None
540                   10  Waffles  None    None    None  None
```

In [12]:
```
df_archive.shape
```

Out[12]:
```
(2356, 17)
```

In [13]:
```
sum(df_archive.duplicated())
```

Out[13]:
```
0
```

In [14]:
```
df_archive.describe()
```

Out[14]:
```
           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           7.455128e+17         2.015385e+16
std    6.856706e+16           7.583419e+16         1.253546e+17
min    6.660210e+17           6.660000e+17         1.185634e+07
25%    6.783992e+17           6.760000e+17         3.086374e+08
50%    7.196275e+17           7.035000e+17         4.196984e+09
```

```
75%    7.993375e+17           8.260000e+17         4.196984e+09
max    8.924210e+17           8.860000e+17         8.410000e+17

       retweeted_status_id  retweeted_status_user_id  rating_numerator  \
count         1.810000e+02              1.810000e+02       2356.000000
mean          7.720221e+17              1.241437e+16         13.126486
std           6.236131e+16              9.597227e+16         45.876648
min           6.660000e+17              7.832140e+05          0.000000
25%           7.190000e+17              4.196984e+09         10.000000
50%           7.800000e+17              4.196984e+09         11.000000
75%           8.200000e+17              4.196984e+09         12.000000
max           8.870000e+17              7.870000e+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 [15]:
```
#sorting by names:
df_archive.name.value_counts().sort_index(ascending=False)
```

Out[15]:
```
very             5
unacceptable     1
this             1
the              8
such             1
space            1
quite            4
one              4
old              1
officially       1
not              2
my               1
mad              2
light            1
life             1
just             4
infuriating      1
incredibly       1
his              1
getting          2
by               1
an               7
```

```
all              1
actually         2
a               55
Zuzu             1
Zooey            1
Zoey             3
Zoe              1
Ziva             1
                ..
Apollo           1
Antony           1
Anthony          1
Anna             1
Angel            1
Andy             1
Andru            1
Anakin           2
Amélie           1
Amy              1
Ambrose          1
Amber            1
Alice            2
Alfy             1
Alfie            5
Alf              1
Alexanderson     1
Alexander        1
Alejandro        1
Aldrick          1
Albus            2
Albert           2
Al               1
Akumi            1
Aja              1
Aiden            1
Adele            1
Acro             1
Ace              1
Abby             2
Name: name, Length: 957, dtype: int64
```

In [16]:
```
#Sorting by rating numerator values:
df_archive.rating_numerator.value_counts().sort_index()
```

Out[16]:
```
0         2
1         9
2         9
3        19
```

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

```
In [17]: #Sorting by rating denominator values:
         df_archive.rating_denominator.value_counts().sort_index()
```

```
Out[17]: 0          1
         2          1
         7          1
         10      2333
         11         3
         15         1
         16         1
```

```
20        2
40        1
50        3
70        1
80        2
90        1
110       1
120       1
130       1
150       1
170       1
Name: rating_denominator, dtype: int64
```

In [18]:
```
df_archive.query('rating_denominator<10')
```

Out[18]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweeted_status_user_id	retweeted_status_timestamp	expanded_urls	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo
313	835246000000000000	8.350000e+17	26259576.0	2017-02-24 21:54:03 +0000	<a href="http://twitter.com/download/iphone" r...	@jonnysun @Lin_Manuel ok jomny I know you're e...	NaN	NaN	NaN	NaN	960	0	None	None	None	None	None
516	810985000000000000	NaN	NaN	2016-12-19 23:06:23 +0000	<a href="http://twitter.com/download/iphone" r...	Meet Sam. She smiles 24/7 & secretly aspir...	NaN	NaN	NaN	https://www.gofundme.com/sams-smile,https://tw...	24	7	Sam	None	None	None	None
2335	666287000000000000	NaN	NaN	2015-11-16 16:11:11 +0000	<a href="http://twitter.com/download/iphone" r...	This is an Albanian 3 1/2 legged Episcopalian...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/666287406...	1	2	an	None	None	None	None

```
In [19]: #Checking for num of retweets :
         len(df_archive[df_archive.retweeted_status_id.isnull() == False])
```

```
Out[19]: 181
```

```
In [20]: #Checkin for duplicate tweet_ids:
         df_archive.tweet_id.duplicated().sum()
```

```
Out[20]: 7
```

```
In [21]: 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 [22]: df_image.sample(10)
```

```
Out[22]:                tweet_id                                          jpg_url  \
         1465  778624900596654080  https://pbs.twimg.com/media/Cs47N3eWcAEmgiW.jpg
         255   670755717859713024  https://pbs.twimg.com/media/CU8AwZ_UsAA-Lbu.jpg
         821   693095443459342336  https://pbs.twimg.com/media/CZ5entwWYAAocEg.jpg
         1873  845306882940190720  https://pbs.twimg.com/media/C7siH5DXkAACnDT.jpg
         1276  750071704093859840  https://pbs.twimg.com/media/CmjKOzVWcAAQN6w.jpg
         2027  882268110199369728  https://pbs.twimg.com/media/DD5yKdPW0AArzX8.jpg
         589   679132435750195208  https://pbs.twimg.com/media/CWzDWOkXAAAP0k7.jpg
         1504  785170936622350336  https://pbs.twimg.com/media/CuV8yfxXEAAUlye.jpg
         296   671362598324076544  https://pbs.twimg.com/media/CVEouDRXAAEe8mt.jpg
         66    667176164155375616  https://pbs.twimg.com/media/CUJJLtWWsAE-go5.jpg

               img_num                      p1   p1_conf  p1_dog  \
         1465        2                Airedale  0.786089    True
         255         1                keeshond  0.994065    True
         821         1               ice_lolly  0.660099   False
```

```
1873        1          Irish_water_spaniel  0.567475    True
1276        2                      redbone  0.382113    True
2027        1            golden_retriever  0.762211    True
589         1          Scottish_deerhound  0.194610    True
1504        2                   seat_belt  0.891193   False
296         1                         tub  0.393616   False
66          1  soft-coated_wheaten_terrier  0.318981    True

                     p2   p2_conf  p2_dog                    p3   p3_conf  \
1465      Irish_terrier  0.121488    True       Lakeland_terrier  0.014603
255   Norwegian_elkhound  0.001827    True                 cairn  0.001821
821          neck_brace  0.039563   False      Yorkshire_terrier  0.033488
1873  Labrador_retriever  0.169496    True  curly-coated_retriever  0.101518
1276           malinois  0.249943    True     miniature_pinscher  0.070926
2027  Labrador_retriever  0.098985    True         cocker_spaniel  0.017199
589     Irish_wolfhound  0.162855    True        giant_schnauzer  0.159837
1504          Eskimo_dog  0.027494    True                Samoyed  0.019530
296              bathtub  0.383522   False        swimming_trunks  0.077301
66      Lakeland_terrier  0.215218    True             toy_poodle  0.106014

      p3_dog
1465    True
255     True
821     True
1873    True
1276    True
2027    True
589     True
1504    True
296    False
66      True
```

In [23]:
```
df_image.shape
```

Out[23]:
```
(2075, 12)
```

In [24]:
```
sum(df_image.duplicated())
```

Out[24]:
```
0
```

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 [26]: #Checkin for duplicate tweet_ids:
         df_image.tweet_id.duplicated().sum()
         df_image['tweet_id'].fillna(value="None", inplace=True)
```

```
In [27]: df_tweets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
tweet_id     2354 non-null object
favorites    2354 non-null object
retweets     2354 non-null object
dtypes: object(3)
memory usage: 55.2+ KB
```

```
In [28]: df_tweets.sample(10)
```

```
Out[28]:                tweet_id favorites retweets
         1685  681579835668455424      3893     1489
         2044  671520732782923777      1499      582
         405   823719002937630720         0    12953
         59    880465832366813184     29075     6546
         1781  677673981332312066      3603     1677
         68    879050749262655488     23022     4941
         839   766864461642756096         0     6521
         1955  673583129559498752      1273      403
         963   750429297815552001     14569     4947
         2086  670792680469889025       889      298
```

```
In [29]: df_tweets.shape
```

```
Out[29]: (2354, 3)
```

```
In [30]: df_tweets.describe()
```

```
Out[30]:                  tweet_id  favorites  retweets
         count                2354       2354      2354
         unique               2354       2007      1724
         top     667495797102141441          0      3652
         freq                    1        179         5
```

```
In [31]: sum(df_tweets.duplicated())
```

```
Out[31]: 0
```

```
In [32]: #Checkin for duplicate tweet_ids:
         df_tweets.tweet_id.duplicated().sum()
         df_tweets['tweet_id'].fillna(value="None", inplace=True)
```

1.2.1 Quality issues

df_archive(Twitter Archive table):
1.Timestamp should be converted to datetime datatype.
2.Dog names are not correct(starting with lowercase like 'a','by','such','not',etc..)
3.There are 181 retweets these can be duplicate values or null values.
4.Columns like in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id,retweeted_status_user_id,,re
can be dropped.
5.Rating numerator column has very large numbers(like 1176).
6.Rating denominators have values less than 10.
7.Data types for tweet-ids must be changed from float to string.
df_image(Image predictions table):
8.Datatypes has to be changed for tweet_id(change to string)
df_tweets(Tweets_info table):
9.Data types for favorites and tweets must be changed.(to integer)

1.2.2 Tidiness issues

1.We have to merge the 3 tables to make it one.
2.We need to address the dog columns as one stage column.

1.3 Cleaning Data

In this section, clean **all** of the issues you documented while assessing.

Note: Make a copy of the original data before cleaning. Cleaning includes merging individual pieces of data according to the rules of tidy data. The result should be a high-quality and tidy master pandas DataFrame (or DataFrames, if appropriate).

```
In [33]: # Make copies of original pieces of data
         df_archive_copy = df_archive.copy()
         df_image_copy = df_image.copy()
         df_tweets_copy = df_tweets.copy()
```

1.3.1 Issue #1:

Define: a.Convert tweet_id data type in df_twitter archive table to string: b.Convert tweet_id dat type in df_image prediction table to string:

Code

```
In [34]: #Convert tweet_id data type in df_twitter archive table to string:
         df_archive_copy.tweet_id = df_archive_copy.tweet_id .astype(str)
```

Test

```
In [35]: df_archive_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null object
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(2), object(11)
memory usage: 313.0+ KB
```

```
In [36]: #Convert tweet_id dat type in df_image prediction table to string:
         df_image_copy.tweet_id = df_image_copy.tweet_id.astype(str)
```

Test:

```
In [37]: df_image_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null object
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(1), object(5)
memory usage: 152.1+ KB
```

1.3.2 Issue #2:

Define: Change the timestamp column.

Code

```
In [38]: #Convert timestamp to datetime:
         df_archive_copy['timestamp'] = pd.to_datetime(df_archive_copy['timestamp'],errors='igno
```

Test

```
In [39]: df_archive_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null object
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null datetime64[ns]
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: datetime64[ns](1), float64(4), int64(2), object(10)
memory usage: 313.0+ KB
```

1.3.3 Issue #3:

Define: Convert all the names with a, by,not,etc to None:

Code

```
In [40]: df_archive_copy['name'].sample(10)
```

```
Out[40]: 737        Dash
         511         Ted
         609      Cassie
         1749       None
```

```
452        Bear
1314     Elliot
50      Stanley
334        None
2235          a
618        Ruby
Name: name, dtype: object
```

```
In [41]: #Converting all the names with a,by,etc to None values:
         df_archive_copy.name.replace(['such', 'an', 'the', 'just', 'by', 'a', 'mad', 'old', 'sp
                     'quite', 'actually', 'infuriating', 'all', 'officially', 'my', 'unacceptab
                      'not', 'O', 'life', 'one', 'his', 'very'],np.NaN, inplace =True)
```

```
In [42]: df_archive_copy['name'].fillna(value="None", inplace=True)
```

Test:

```
In [43]: df_archive_copy.name.value_counts()
```

```
Out[43]: None        850
         Charlie      12
         Cooper       11
         Oliver       11
         Lucy         11
         Tucker       10
         Penny        10
         Lola         10
         Bo            9
         Winston       9
         Sadie         8
         Toby          7
         Bailey        7
         Daisy         7
         Buddy         7
         Bella         6
         Stanley       6
         Koda          6
         Milo          6
         Dave          6
         Jack          6
         Rusty         6
         Leo           6
         Jax           6
         Scout         6
         Oscar         6
         Sammy         5
         Alfie         5
         Larry         5
```

```
Chester       5
              ...
Stark         1
Snicku        1
Tripp         1
Mary          1
Dallas        1
Crumpet       1
Ed            1
Carper        1
Strudel       1
Pluto         1
Berb          1
Tove          1
Andy          1
Rambo         1
Lorelei       1
William       1
Sprout        1
Chesney       1
Rodman        1
Rodney        1
Milky         1
Kody          1
Ralphson      1
Crawford      1
Lacy          1
Wafer         1
Marty         1
Boston        1
Gin           1
Fido          1
Name: name, Length: 935, dtype: int64
```

1.3.4 Issue #5:

Define: Keeping original retweets in df_archive and remove the retweets in retweeted_status_id column thats null or duplicates.

Code:

```
In [44]: df_archive_copy=df_archive_copy[df_archive_copy.retweeted_status_id.isnull()]
```

Test:

```
In [45]: len(df_archive_copy[df_archive_copy.retweeted_status_id.isnull()==False])

Out[45]: 0
```

1.3.5 Issue #6:

Define: Dropping uneccessary columns in df_archive_copy table(retweeted_status_id,retweeted_user_id,retweeted_status_timestamp)

Code:

```
In [46]: df_archive_copy.drop(['retweeted_status_id','retweeted_status_user_id','retweeted_statu
```

Test:

```
In [47]: df_archive_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 14 columns):
tweet_id                 2175 non-null object
in_reply_to_status_id    78 non-null float64
in_reply_to_user_id      78 non-null float64
timestamp                2175 non-null datetime64[ns]
source                   2175 non-null object
text                     2175 non-null object
expanded_urls            2117 non-null object
rating_numerator         2175 non-null int64
rating_denominator       2175 non-null int64
name                     2175 non-null object
doggo                    2175 non-null object
floofer                  2175 non-null object
pupper                   2175 non-null object
puppo                    2175 non-null object
dtypes: datetime64[ns](1), float64(2), int64(2), object(9)
memory usage: 254.9+ KB
```

1.3.6 Issue #7:

Define: Working with Numerator and Denominator:

Code:

```
In [48]: #Using describe function for numerator:
         df_archive_copy[['rating_numerator']].describe()
```

```
Out[48]:        rating_numerator
         count       2175.000000
         mean          13.215172
         std           47.725696
         min            0.000000
         25%           10.000000
```

```
50%           11.000000
75%           12.000000
max         1776.000000
```

```
In [49]: # Finding unique values in rating_numerator:
         df_archive_copy.rating_numerator.unique()
```

```
Out[49]: array([  13,   12,   14,    5,   17,   11,   10,  420,  666,    6,  182,
                  15,  960,    0,    7,   84,   24,   75,   27,    3,    8,    9,
                   4,  165, 1776,  204,   50,   99,   80,   45,   60,   44,    1,
                 143,  121,   20,   26,    2,  144,   88])
```

```
In [50]: #Using describe function for denominator:
         df_archive_copy[['rating_denominator']].describe()
```

```
Out[50]:        rating_denominator
         count         2175.000000
         mean            10.492874
         std              7.019084
         min              0.000000
         25%             10.000000
         50%             10.000000
         75%             10.000000
         max            170.000000
```

```
In [51]: # Finding unique values in rating_denominators:
         df_archive_copy.rating_denominator.unique()
```

```
Out[51]: array([ 10,   0,  15,  70,   7, 150,  11, 170,  20,  50,  90,  80,  40,
                130, 110,  16, 120,   2])
```

```
In [52]: #Finding numerator for 1176:
         df_archive_copy.query('rating_numerator ==1776')
```

```
Out[52]:                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         979  749981000000000000                    NaN                  NaN

                      timestamp                                             source  \
         979 2016-07-04 15:00:45  <a href="https://about.twitter.com/products/tw...

                                                        text  \
         979  This is Atticus. He's quite simply America af...

                                                 expanded_urls  rating_numerator  \
         979  https://twitter.com/dog_rates/status/749981277...              1776

              rating_denominator     name doggo floofer pupper puppo
         979                  10  Atticus  None    None   None  None
```

Rating is inconsistent, but the rating numerators are greater than the denominators and does not need to be cleaned.

```
In [53]: #Finding denominators for <10:
         df_archive_copy.query('rating_denominator<10')
```

Out[53]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	expanded_urls	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo
313	835246000000000000	8.350000e+17	26259576.0	2017-02-24 21:54:03	<a href="http://twitter.com/download/iphone" r...	@jonnysun @Lin_Manuel ok jomny I know you're e...	NaN	960	0	None	None	None	None	None
516	810985000000000000	NaN	NaN	2016-12-19 23:06:23	<a href="http://twitter.com/download/iphone" r...	Meet Sam. She smiles 24/7 & secretly aspir...	https://www.gofundme.com/sams-smile,https://tw...	24	7	Sam	None	None	None	None
2335	666287000000000000	NaN	NaN	2015-11-16 16:11:11	<a href="http://twitter.com/download/iphone" r...	This is an Albanian 3 1/2 legged Episcopalian...	https://twitter.com/dog_rates/status/666287406...	1	2	None	None	None	None	None

```
In [54]: #Dropping the unwanted rows:
         df_archive_copy.drop([313], inplace=True)
         df_archive_copy.drop([516], inplace=True)
         df_archive_copy.drop([2335], inplace=True)
```

Test:

```
In [55]: df_archive_copy[df_archive_copy['rating_denominator']==7.0]
         df_archive_copy[df_archive_copy['rating_denominator']==2.0]
         df_archive_copy[df_archive_copy['rating_denominator']==0.0]
```

```
Out[55]: Empty DataFrame
         Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, text
         Index: []
```

1.3.7 Issue #8:

Define: Working with dog_names and converting the dog_names to one column as dog_stage:

Code:

```
In [56]: #Creating a new dataframe df1_archive_copy:
         df1_archive_copy = pd.DataFrame(df_archive_copy)
         #Replace all NaN and None dog_stage to an empty string:
         df_archive_copy.doggo.replace('None', ' ', inplace=True)
         df_archive_copy.doggo.replace(np.NaN, ' ', inplace=True)
         df_archive_copy.floofer.replace('None', ' ', inplace=True)
         df_archive_copy.floofer.replace(np.NaN, ' ', inplace=True)
         df_archive_copy.pupper.replace('None', ' ', inplace=True)
         df_archive_copy.pupper.replace(np.NaN , ' ', inplace=True)
         df_archive_copy.puppo.replace('None', ' ', inplace=True)
         df_archive_copy.puppo.replace(np.NaN, ' ', inplace=True)

In [57]: #Now get the columns combined :
         df_archive_copy['dog_stages'] = df_archive_copy.text.str.extract('(doggo|floofer|pupper

In [58]: #There are some dogs have multiple stages:
         df_archive_copy['dog_stages'] = df1_archive_copy.doggo + df1_archive_copy.floofer + df1
         df_archive_copy.loc[df1_archive_copy.dog_stages == 'doggopupper', 'dog_stages'] = 'dogg
         df_archive_copy.loc[df1_archive_copy.dog_stages == 'doggopuppo', 'dog_stages'] = 'doggo
         df_archive_copy.loc[df1_archive_copy.dog_stages == 'doggofloofer', 'dog_stages'] = 'dog

In [59]: #Now delete useless columns :
         df_archive_copy.drop(['doggo','floofer','pupper','puppo'],axis=1,inplace=True)
```

Test:

```
In [60]: df_archive_copy.dog_stages.value_counts()

Out[60]:                   1828
           pupper           224
         doggo               75
            puppo            24
         doggo pupper        10
          floofer             9
         doggofloofer         1
         doggo  puppo         1
         Name: dog_stages, dtype: int64

In [61]: df_archive_copy.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2172 entries, 0 to 2355
Data columns (total 11 columns):
tweet_id                2172 non-null object
in_reply_to_status_id    77 non-null float64
in_reply_to_user_id      77 non-null float64
timestamp               2172 non-null datetime64[ns]
```

```
source                  2172 non-null object
text                    2172 non-null object
expanded_urls           2115 non-null object
rating_numerator        2172 non-null int64
rating_denominator      2172 non-null int64
name                    2172 non-null object
dog_stages              2172 non-null object
dtypes: datetime64[ns](1), float64(2), int64(2), object(6)
memory usage: 203.6+ KB
```

```
In [62]: #Drop columns in_reply_to_status_id and in_reply_to_user_id:
         df_archive_copy.drop(['in_reply_to_status_id','in_reply_to_user_id'],axis=1,inplace=Tru
```

```
In [63]: #Test:
         df_archive_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2172 entries, 0 to 2355
Data columns (total 9 columns):
tweet_id              2172 non-null object
timestamp             2172 non-null datetime64[ns]
source                2172 non-null object
text                  2172 non-null object
expanded_urls         2115 non-null object
rating_numerator      2172 non-null int64
rating_denominator    2172 non-null int64
name                  2172 non-null object
dog_stages            2172 non-null object
dtypes: datetime64[ns](1), int64(2), object(6)
memory usage: 169.7+ KB
```

```
In [64]: #Drop unwanted columns in df_image predictions table:
         df_image_copy.drop(['img_num','p1','p1_conf','p1_dog','p2','p2_conf','p2_dog','p3','p3_
```

```
In [65]: #Testing:
         df_image_copy.head()
```

```
Out[65]:              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
```

1.3.8 Issue #9:

Define: Convert favourites and retweets to integer in df_tweets table:

```
In [66]: df_tweets_copy['favorites'] = df_tweets_copy['favorites'].apply(pd.to_numeric, errors='
         df_tweets_copy['retweets'] = df_tweets_copy['retweets'].apply(pd.to_numeric, errors='co
```

Test:

```
In [67]: df_tweets_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
tweet_id     2354 non-null int64
favorites    2354 non-null int64
retweets     2354 non-null int64
dtypes: int64(3)
memory usage: 55.2 KB
```

```
In [68]: #converting tweet_id in all 3 dataset tables to string:
         df_archive_copy.tweet_id = df_archive_copy.tweet_id.astype(str)
         df_tweets_copy.tweet_id = df_tweets_copy.tweet_id .astype(str)
         df_image_copy.tweet_id = df_image_copy.tweet_id .astype(str)
         df_tweets_copy.info()
         df_archive_copy.info()
         df_image_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
tweet_id     2354 non-null object
favorites    2354 non-null int64
retweets     2354 non-null int64
dtypes: int64(2), object(1)
memory usage: 55.2+ KB
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2172 entries, 0 to 2355
Data columns (total 9 columns):
tweet_id              2172 non-null object
timestamp             2172 non-null datetime64[ns]
source                2172 non-null object
text                  2172 non-null object
expanded_urls         2115 non-null object
rating_numerator      2172 non-null int64
rating_denominator    2172 non-null int64
name                  2172 non-null object
dog_stages            2172 non-null object
dtypes: datetime64[ns](1), int64(2), object(6)
memory usage: 169.7+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 2 columns):
tweet_id    2075 non-null object
jpg_url     2075 non-null object
```

```
dtypes: object(2)
memory usage: 32.5+ KB
```

2 merge the 3 tables df_twitter,df_image,df_archive to one table twitter_archive_master:

df_twitter_archive = pd.merge(df_archive_copy, df_tweets_copy,on = 'tweet_id',how = 'outer')
df_twitter_archive = pd.merge(df_twitter_archive, df_image_copy, on= 'tweet_id',how = 'outer')
df_twitter_archive.head()

```
In [73]: df_twitter_clean=df_image_copy.merge(df_archive_copy,on='tweet_id',how = 'right')
         df_twitter_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2172 entries, 0 to 2171
Data columns (total 10 columns):
tweet_id              2172 non-null object
jpg_url               0 non-null object
timestamp             2172 non-null datetime64[ns]
source                2172 non-null object
text                  2172 non-null object
expanded_urls         2115 non-null object
rating_numerator      2172 non-null int64
rating_denominator    2172 non-null int64
name                  2172 non-null object
dog_stages            2172 non-null object
dtypes: datetime64[ns](1), int64(2), object(7)
memory usage: 186.7+ KB
```

```
In [ ]:
```

```
In [ ]:
```

2.1 Storing Data

Save gathered, assessed, and cleaned master dataset to a CSV file named "twitter_archive_master.csv".

```
In [ ]:
```

2.2 Analyzing and Visualizing Data

In this section, analyze and visualize your wrangled data. You must produce at least **three (3) insights and one (1) visualization.**

```
In [ ]:
```

2.2.1 Insights:

1.

2.

3.

2.2.2 Visualization

In [ ]:

In [ ]:

In [ ]:

In [ ]: