wrangle_act

December 18, 2020

1 Wrangle Act

1.1 Table of Contents

1. Gather data
2. Assess
3. Clean
4. Tidiness
5. Store
6. Analyze
7. Report

2 GATHER

```
In [129]: # I used the Mentor Help section for the .txt Twitter file.

          import pandas as pd
          import numpy as np
          import json
          from timeit import default_timer as timer
          import tweepy
          from tweepy import OAuthHandler


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

```
In [130]: # Check import of 'twitter-archive-enhanced'
          df.head()
```

```
Out[130]:             tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
          0  892420643555336193                    NaN                  NaN
          1  892177421306343426                    NaN                  NaN
          2  891815181378084864                    NaN                  NaN
          3  891689557279858688                    NaN                  NaN
          4  891327558926688256                    NaN                  NaN

                             timestamp  \
0  2017-08-01 16:23:56 +0000
1  2017-08-01 00:17:27 +0000
2  2017-07-31 00:18:03 +0000
3  2017-07-30 15:58:51 +0000
4  2017-07-29 16:00:24 +0000

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

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

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

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

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

```
In [131]: df.describe()
```

```
Out[131]:
             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 [132]: # Show duplicated tweet id's in 'twitter-archive-enhanced'
          df_dup_rows = df[df.duplicated(['tweet_id'])]
          df_dup_rows

Out[132]: Empty DataFrame
          Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, tex
          Index: []
```

```
In [133]: df.query('rating_numerator').tweet_id.max()

Out[133]: 892420643555336193
```

```
In [134]: df['in_reply_to_status_id'].sort_values()

Out[134]: 1914    6.658147e+17
          2298    6.670655e+17
          1339    6.671522e+17
          149     6.671522e+17
          2169    6.678065e+17
          2189    6.689207e+17
          2149    6.693544e+17
          1464    6.706684e+17
          2038    6.715449e+17
          2036    6.715610e+17
```

```
1885    6.717299e+17
1940    6.737159e+17
1905    6.744689e+17
1895    6.747400e+17
1892    6.747522e+17
1882    6.747934e+17
1866    6.749998e+17
1452    6.753494e+17
1852    6.754971e+17
1842    6.757073e+17
1844    6.758457e+17
1819    6.765883e+17
1774    6.780211e+17
1689    6.813394e+17
1663    6.827884e+17
1634    6.842229e+17
1630    6.844811e+17
1618    6.849598e+17
1605    6.855479e+17
1598    6.860340e+17
            ...
2326             NaN
2327             NaN
2328             NaN
2329             NaN
2330             NaN
2331             NaN
2332             NaN
2333             NaN
2334             NaN
2335             NaN
2336             NaN
2337             NaN
2338             NaN
2339             NaN
2340             NaN
2341             NaN
2342             NaN
2343             NaN
2344             NaN
2345             NaN
2346             NaN
2347             NaN
2348             NaN
2349             NaN
2350             NaN
2351             NaN
2352             NaN
```

```
          2353             NaN
          2354             NaN
          2355             NaN
          Name: in_reply_to_status_id, Length: 2356, dtype: float64
```

```
In [135]: df.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 [136]: # import tsv file (I used this video or assistance: https://www.youtube.com/watch?v=ch

          url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predict
          url_df = pd.read_table(url, sep='\t')
          url_df.head()

          # Save html to file #(helper = https://cmdlinetips.com/2020/03/save-a-pandas-data-fram
          url_df.to_csv('weratedogs-image-predictions.tsv', sep='\t')
```

```
In [137]: df.dtypes
```

```
Out[137]: tweet_id                         int64
          in_reply_to_status_id          float64
          in_reply_to_user_id            float64
          timestamp                       object
          source                          object
          text                            object
          retweeted_status_id            float64
```

```
retweeted_status_user_id      float64
retweeted_status_timestamp     object
expanded_urls                  object
rating_numerator                int64
rating_denominator              int64
name                           object
doggo                          object
floofer                        object
pupper                         object
puppo                          object
dtype: object
```

```
In [138]: url_df.head()
```

```
Out[138]:              tweet_id                                          jpg_url  \
          0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
          1  666029285002620928  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg
          2  666033412701032449  https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg
          3  666044226329800704  https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg
          4  666049248165822465  https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg

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

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

```
In [139]: url_df.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 [140]: # Twitter denied me developer access. I used the Udacity Mentor Help section to comple
          # File was corrupted / unusable upon creation, ultimately downloaded from Udacity

          #consumer_key='@@@@'
          #consumer_secret='@@@@'
          #access_token='@@@@'
          #access_secret='@@@@'

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

          #api = tweepy.API(auth_handler=auth, parser = tweepy.parsers.JSONParser(), wait_on_rat
```

```
In [141]: # Opening the file after running
          # Save the retweet and favorite counts for each tweet ID in a new pandas DataFrame
          from pprint import pprint

          tweets_list = []
          with open('tweet-json.txt') as f:
              for line in f:
                  temp_dict = {}
                  status  = json.loads(line)
                  temp_dict["tweet_id"] = status['id']
                  temp_dict["retweet_count"] = status['retweet_count']
                  temp_dict["favorite_count"] = status['favorite_count']
                  tweets_list.append(temp_dict)

          df_tweets = pd.DataFrame(tweets_list)
          df_tweets
```

```
Out[141]:       favorite_count  retweet_count            tweet_id
          0              39467           8853  892420643555336193
          1              33819           6514  892177421306343426
          2              25461           4328  891815181378084864
          3              42908           8964  891689557279858688
          4              41048           9774  891327558926688256
          5              20562           3261  891087950875897856
          6              12041           2158  890971913173991426
          7              56848          16716  890729181411237888
          8              28226           4429  890609185150312448
          9              32467           7711  890240255349198849
```

10	31166	7624	890006608113172480
11	28268	5156	889880896479866881
12	38818	8538	889665388333682689
13	27672	4735	889638837579907072
14	15359	2321	889531135344209921
15	25652	5637	889278841981685760
16	29611	4709	888917238123831296
17	26080	4559	888804989199671297
18	20290	3732	888554962724278272
19	22201	3653	888078434458587136
20	30779	5609	887705289381826560
21	46959	12082	887517139158093824
22	69871	18781	887473957103951883
23	34222	10737	887343217045368832
24	31061	6167	887101392804085760
25	35859	8084	886983233522544640
26	12306	3443	886736880519319552
27	22798	4610	886680336477933568
28	21524	3316	886366144734445568
29	117	4	886267009285017600
...	...	...	...
2324	459	339	666411507551481857
2325	113	44	666407126856765440
2326	172	92	666396247373291520
2327	194	100	666373753744588802
2328	804	595	666362758909284353
2329	229	77	666353288456101888
2330	307	146	666345417576210432
2331	204	96	666337882303524864
2332	522	368	666293911632134144
2333	152	71	666287406224695296
2334	184	82	666273097616637952
2335	108	37	666268910803644416
2336	14765	6871	666104133288665088
2337	81	16	666102155909144576
2338	164	73	666099513787052032
2339	169	79	666094000022159362
2340	121	47	666082916733198337
2341	335	174	666073100786774016
2342	154	67	666071193221509120
2343	496	232	666063827256086533
2344	115	61	666058600524156928
2345	304	146	666057090499244032
2346	448	261	666055525042405380
2347	1253	879	666051853826850816
2348	136	60	666050758794694657
2349	111	41	666049248165822465
2350	311	147	666044226329800704

```
          2351             128             47  666033412701032449
          2352             132             48  666029285002620928
          2353            2535            532  666020888022790149

          [2354 rows x 3 columns]
```

```
In [142]: df_tweets.info()
```

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

2.1 Assess

2.1.1 Quality Issues

List at least 8 quality issues with the three data files:

File 1 - Twitter-archive-enhanced.csv - 1.1 - Several columns missing data (’in_reply_to_status_id’, ’in_reply_to_user_id’, ’retweeted_status_id’, ’retweeted_status_user_id’, ’retweeted_status_timestamp’, ’expanded_urls’) - 1.2 - ’timestamp’ and ’retweeted_status_timestamp’ columns are objects, should be datetime - 1.3 - Investigate rating outliers by creating a rating column where the numerator is divided by the denominator - 1.4 - Remove rows listed as replying to an original tweet as not being an original tweet and therefore should not be included in comparing ratings - 1.5 - Drop the rows containing all zeros

File 2 - weratedogs-image-predictions.tsv

File 3 - json_tweets.txt

2.1.2 Tidiness Issues

List at least 2 tidiness issues with the three data files:

File 1 - Twitter-archive-enhanced.csv - ’Source’ column needs to be stripped down to one distinct variable in a column to be called ’device’; drop ’source’ column and keep new ’device’ column - Combine dog image columns doggo, floofer, pupper and puppo into one column so there is one variable for the stage of the dog

File 2 - weratedogs-image-predictions.tsv - Add image data to ’Twitter-archive-enhanced’ to add attributes for analysis

File 3 - json_tweets.txt - Join json_tweets with ’Twitter-archive-enhanced’ to add attributes for analysis

2.2 Clean

```
In [143]: #Create a copy of each file
          df_archive_clean = df.copy()
          url_img_df_clean = url_df.copy()
          df_tweets_clean = df_tweets.copy()
```

```
In [144]: # check copy for data
          df_archive_clean.head(3)
```

```
Out[144]:              tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
          0  892420643555336193                    NaN                  NaN
          1  892177421306343426                    NaN                  NaN
          2  891815181378084864                    NaN                  NaN

                             timestamp  \
          0  2017-08-01 16:23:56 +0000
          1  2017-08-01 00:17:27 +0000
          2  2017-07-31 00:18:03 +0000

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

                                                          text  retweeted_status_id  \
          0  This is Phineas. He's a mystical boy. Only eve...                  NaN
          1  This is Tilly. She's just checking pup on you...                  NaN
          2  This is Archie. He is a rare Norwegian Pouncin...                  NaN

             retweeted_status_user_id retweeted_status_timestamp  \
          0                       NaN                        NaN
          1                       NaN                        NaN
          2                       NaN                        NaN

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

             rating_denominator     name doggo floofer pupper puppo
          0                  10  Phineas  None    None   None  None
          1                  10    Tilly  None    None   None  None
          2                  10   Archie  None    None   None  None
```

```
In [145]: # Check copy for data
          url_img_df_clean.head(3)
```

```
Out[145]:              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
```

```
   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

    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
```

```
In [146]: # Check copy for data

          df_tweets_clean.head(3)
```

```
Out[146]:    favorite_count  retweet_count            tweet_id
          0           39467           8853  892420643555336193
          1           33819           6514  892177421306343426
          2           25461           4328  891815181378084864
```

```
In [148]: # 1.1 - I need these as integers for this part of my project
          df_archive_clean['in_reply_to_status_id'] = df_archive_clean['in_reply_to_status_id'].
          df_archive_clean['in_reply_to_status_id'] = df_archive_clean['in_reply_to_status_id'].

          df_archive_clean['in_reply_to_user_id'] = df_archive_clean['in_reply_to_user_id'].fill
          df_archive_clean['in_reply_to_user_id'] = df_archive_clean['in_reply_to_user_id'].asty

          df_archive_clean['retweeted_status_id'] = df_archive_clean['retweeted_status_id'].fill
          df_archive_clean['retweeted_status_id'] = df_archive_clean['retweeted_status_id'].asty

          df_archive_clean['retweeted_status_user_id'] = df_archive_clean['retweeted_status_user
          df_archive_clean['retweeted_status_user_id'] = df_archive_clean['retweeted_status_user
```

```
In [149]: # 1.1 cont. -  Taking a look at the values in the 'in_reply_to_status_id' columns
          sorted(df_archive_clean['in_reply_to_status_id'])
```

```
Out[149]: [0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           0,
           ...]
```

```
In [150]: # 1.2 - Convert 'timestamp' and 'retweeted_status_timestamp' to datetime format https:
          # Define the code to clean the files
          from datetime import datetime

          df_archive_clean['timestamp'] = pd.to_datetime(df_archive_clean['timestamp'])
          df_archive_clean['retweeted_status_timestamp'] = pd.to_datetime(df_archive_clean['retw


          # Programmatically clean the file - test the code to see if it was successful
          df_archive_clean.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         2356 non-null int64
in_reply_to_user_id           2356 non-null int64
timestamp                     2356 non-null datetime64[ns]
source                        2356 non-null object
text                          2356 non-null object
```

```
retweeted_status_id           2356 non-null int64
retweeted_status_user_id      2356 non-null int64
retweeted_status_timestamp    181 non-null datetime64[ns]
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](2), int64(7), object(8)
memory usage: 313.0+ KB
```

```
In [151]: # 1.2. - Test to see if the code was successful
          df_archive_clean.head()
```

```
Out[151]:              tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
          0  892420643555336193                      0                    0
          1  892177421306343426                      0                    0
          2  891815181378084864                      0                    0
          3  891689557279858688                      0                    0
          4  891327558926688256                      0                    0

                      timestamp                                             source  \
          0 2017-08-01 16:23:56  <a href="http://twitter.com/download/iphone" r...
          1 2017-08-01 00:17:27  <a href="http://twitter.com/download/iphone" r...
          2 2017-07-31 00:18:03  <a href="http://twitter.com/download/iphone" r...
          3 2017-07-30 15:58:51  <a href="http://twitter.com/download/iphone" r...
          4 2017-07-29 16:00:24  <a href="http://twitter.com/download/iphone" r...

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

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

                                                expanded_urls  rating_numerator  \
          0  https://twitter.com/dog_rates/status/892420643...                13
```

```
1  https://twitter.com/dog_rates/status/892177421...                13
2  https://twitter.com/dog_rates/status/891815181...                12
3  https://twitter.com/dog_rates/status/891689557...                13
4  https://twitter.com/dog_rates/status/891327558...                12

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

In [152]:
```
# 1.3 - Investigate rating outliers by creating a new rating column where the numerato
#Define the code to clean the files
df_archive_clean['new_rating'] = df_archive_clean['rating_numerator'] / df_archive_cle

# Programmatically clean the file - test the code to see if it was successful

df_archive_clean.head(3)
```

Out[152]:
```
             tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
0  892420643555336193                      0                    0
1  892177421306343426                      0                    0
2  891815181378084864                      0                    0

            timestamp                                              source  \
0 2017-08-01 16:23:56  <a href="http://twitter.com/download/iphone" r...
1 2017-08-01 00:17:27  <a href="http://twitter.com/download/iphone" r...
2 2017-07-31 00:18:03  <a href="http://twitter.com/download/iphone" r...

                                                text  retweeted_status_id  \
0  This is Phineas. He's a mystical boy. Only eve...                    0
1  This is Tilly. She's just checking pup on you...                    0
2  This is Archie. He is a rare Norwegian Pouncin...                    0

   retweeted_status_user_id retweeted_status_timestamp  \
0                         0                        NaT
1                         0                        NaT
2                         0                        NaT

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

   rating_denominator     name doggo floofer pupper puppo  new_rating
0                   10  Phineas  None    None   None  None         1.3
1                   10    Tilly  None    None   None  None         1.3
2                   10   Archie  None    None   None  None         1.2
```

```
In [153]: # 1.4 - Remove rows listed as replying to an original tweet as not being an original t
          # 1.4.1 - Remove retweet rows for column 'in_reply_to_status_id'
          #Define the code to clean the files
          df_archive_clean = df_archive_clean[df_archive_clean.in_reply_to_status_id == 0]

          # Programmatically clean the file - test the code to see if it was successful
          df_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2278 entries, 0 to 2355
Data columns (total 18 columns):
tweet_id                      2278 non-null int64
in_reply_to_status_id         2278 non-null int64
in_reply_to_user_id           2278 non-null int64
timestamp                     2278 non-null datetime64[ns]
source                        2278 non-null object
text                          2278 non-null object
retweeted_status_id           2278 non-null int64
retweeted_status_user_id      2278 non-null int64
retweeted_status_timestamp    181 non-null datetime64[ns]
expanded_urls                 2274 non-null object
rating_numerator              2278 non-null int64
rating_denominator            2278 non-null int64
name                          2278 non-null object
doggo                         2278 non-null object
floofer                       2278 non-null object
pupper                        2278 non-null object
puppo                         2278 non-null object
new_rating                    2278 non-null float64
dtypes: datetime64[ns](2), float64(1), int64(7), object(8)
memory usage: 338.1+ KB
```

```
In [154]: # 1.4.1 - Sum to ensure the column total is zero
          df_archive_clean['in_reply_to_status_id'].sum()
```

```
Out[154]: 0
```

```
In [155]: # 1.4 - Remove rows listed as replying to an original tweet as not being an original t
          # 1.4.2 - Remove retweet rows for column 'in_reply_to_user_id'
          #Define the code to clean the files
          df_archive_clean = df_archive_clean[df_archive_clean.in_reply_to_user_id == 0]

          # Programmatically clean the file - test the code to see if it was successful
          df_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2278 entries, 0 to 2355
Data columns (total 18 columns):
```

```
tweet_id                      2278 non-null int64
in_reply_to_status_id         2278 non-null int64
in_reply_to_user_id           2278 non-null int64
timestamp                     2278 non-null datetime64[ns]
source                        2278 non-null object
text                          2278 non-null object
retweeted_status_id           2278 non-null int64
retweeted_status_user_id      2278 non-null int64
retweeted_status_timestamp    181 non-null datetime64[ns]
expanded_urls                 2274 non-null object
rating_numerator              2278 non-null int64
rating_denominator            2278 non-null int64
name                          2278 non-null object
doggo                         2278 non-null object
floofer                       2278 non-null object
pupper                        2278 non-null object
puppo                         2278 non-null object
new_rating                    2278 non-null float64
dtypes: datetime64[ns](2), float64(1), int64(7), object(8)
memory usage: 338.1+ KB
```

```
In [156]: # 1.4.2 - Sum to ensure the column total is zero
          len(df_archive_clean['in_reply_to_user_id'])
```

```
Out[156]: 2278
```

```
In [157]: # 1.4 - Remove rows listed as replying to an original tweet as not being an original t
          # 1.4.3 - Remove retweet rows for column 'retweeted_status_id'
          #Define the code to clean the files
          df_archive_clean = df_archive_clean[df_archive_clean.retweeted_status_id == 0]

          # Programmatically clean the file - test the code to see if it was successful
          df_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2355
Data columns (total 18 columns):
tweet_id                      2097 non-null int64
in_reply_to_status_id         2097 non-null int64
in_reply_to_user_id           2097 non-null int64
timestamp                     2097 non-null datetime64[ns]
source                        2097 non-null object
text                          2097 non-null object
retweeted_status_id           2097 non-null int64
retweeted_status_user_id      2097 non-null int64
retweeted_status_timestamp    0 non-null datetime64[ns]
expanded_urls                 2094 non-null object
rating_numerator              2097 non-null int64
```

```
rating_denominator            2097 non-null int64
name                          2097 non-null object
doggo                         2097 non-null object
floofer                       2097 non-null object
pupper                        2097 non-null object
puppo                         2097 non-null object
new_rating                    2097 non-null float64
dtypes: datetime64[ns](2), float64(1), int64(7), object(8)
memory usage: 311.3+ KB
```

```
In [158]: # 1.4.3 - Ensure the column length is zero
          len(df_archive_clean['retweeted_status_id'])

Out[158]: 2097

In [159]: # Check to make sure 'retweeted_status_user_id' length is zero
          len(df_archive_clean['retweeted_status_user_id'])

Out[159]: 2097

In [160]: # 1.5 - Drop columns containing zeros and are unrelated to what we plan to analyze
          # 1.5.1 - Remove columns 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted
          #Define the code to clean the files
          df_archive_clean = df_archive_clean.drop(['retweeted_status_user_id'], axis = 1)
          df_archive_clean = df_archive_clean.drop(['retweeted_status_id'], axis = 1)
          df_archive_clean = df_archive_clean.drop(['retweeted_status_timestamp'], axis = 1)

In [161]: # Programmatically clean the file - test the code to see if it was successful

          df_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2355
Data columns (total 15 columns):
tweet_id                 2097 non-null int64
in_reply_to_status_id    2097 non-null int64
in_reply_to_user_id      2097 non-null int64
timestamp                2097 non-null datetime64[ns]
source                   2097 non-null object
text                     2097 non-null object
expanded_urls            2094 non-null object
rating_numerator         2097 non-null int64
rating_denominator       2097 non-null int64
name                     2097 non-null object
doggo                    2097 non-null object
floofer                  2097 non-null object
pupper                   2097 non-null object
puppo                    2097 non-null object
```

```
new_rating               2097 non-null float64
dtypes: datetime64[ns](1), float64(1), int64(5), object(8)
memory usage: 262.1+ KB
```

```
In [162]: # 1.5 - Drop columns containing zeros and are unrelated to what we plan to analyze
          # 1.5.2 - Remove columns 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted
          #Define the code to clean the files
          df_archive_clean = df_archive_clean.drop(['in_reply_to_status_id', 'in_reply_to_user_i
```

```
In [163]: new_df_archive = df_archive_clean.copy()
          new_df_archive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2355
Data columns (total 13 columns):
tweet_id              2097 non-null int64
timestamp             2097 non-null datetime64[ns]
source                2097 non-null object
text                  2097 non-null object
expanded_urls         2094 non-null object
rating_numerator      2097 non-null int64
rating_denominator    2097 non-null int64
name                  2097 non-null object
doggo                 2097 non-null object
floofer               2097 non-null object
pupper                2097 non-null object
puppo                 2097 non-null object
new_rating            2097 non-null float64
dtypes: datetime64[ns](1), float64(1), int64(3), object(8)
memory usage: 229.4+ KB
```

2.3 Tidiness

```
In [164]: # I am going to join the data and then strip the source column down to one variable
```

```
In [165]: # Combine the stage of the dog columns doggo, floofer, pupper and puppo into one colum
          # Replace the 'None' in the dog stage columns with "", code provided by Udacity projec
          new_df_archive.doggo.replace('None', "", inplace=True)
          new_df_archive.floofer.replace('None', "", inplace=True)
          new_df_archive.pupper.replace('None', "", inplace=True)
          new_df_archive.puppo.replace('None', "", inplace=True)

          # Test the code
          new_df_archive.head(3)
```

```
Out[165]:              tweet_id           timestamp  \
          0  892420643555336193 2017-08-01 16:23:56
```

```
1  892177421306343426 2017-08-01 00:17:27
2  891815181378084864 2017-07-31 00:18:03

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

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

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

   rating_denominator     name doggo floofer pupper puppo  new_rating
0                  10  Phineas                                    1.3
1                  10    Tilly                                    1.3
2                  10   Archie                                    1.2
```

```
In [166]: # Combine the stage columns
          # This code was provided by the Udacity reviewer
          new_df_archive['stage'] = new_df_archive.doggo + new_df_archive.floofer + new_df_archi

          # Test the code

          new_df_archive['stage'].count()

Out[166]: 2097
```

```
In [167]: # Combine the stage columns
          # This code was provided by the Udacity reviewer
          new_df_archive.loc[new_df_archive.stage=='doggopupper', 'stage']='doggo, pupper'
          new_df_archive.loc[new_df_archive.stage=='doggopuppo', 'stage']='doggo, puppo'
          new_df_archive.loc[new_df_archive.stage=='doggofloofer', 'stage']='doggo, floofer'

          #Test the code
          new_df_archive.head()

Out[167]:              tweet_id           timestamp  \
          0  892420643555336193 2017-08-01 16:23:56
          1  892177421306343426 2017-08-01 00:17:27
          2  891815181378084864 2017-07-31 00:18:03
          3  891689557279858688 2017-07-30 15:58:51
          4  891327558926688256 2017-07-29 16:00:24
```

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

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

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

   rating_denominator      name doggo floofer pupper puppo  new_rating stage
0                  10   Phineas                                    1.3
1                  10     Tilly                                    1.3
2                  10    Archie                                    1.2
3                  10     Darla                                    1.3
4                  10  Franklin                                    1.2
```

```
In [168]: # 3.1 - Merge datasets 'df_archive_clean' and 'df_tweets_clean' on 'tweet_id'
          # Define the code
          tweets_merged = pd.merge(left=new_df_archive, right=df_tweets_clean)
```

```
In [169]: # 3.1 - Test the code
          tweets_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2096
Data columns (total 16 columns):
tweet_id              2097 non-null int64
timestamp             2097 non-null datetime64[ns]
source                2097 non-null object
text                  2097 non-null object
expanded_urls         2094 non-null object
rating_numerator      2097 non-null int64
rating_denominator    2097 non-null int64
name                  2097 non-null object
doggo                 2097 non-null object
floofer               2097 non-null object
```

```
pupper                  2097 non-null object
puppo                   2097 non-null object
new_rating              2097 non-null float64
stage                   2097 non-null object
favorite_count          2097 non-null int64
retweet_count           2097 non-null int64
dtypes: datetime64[ns](1), float64(1), int64(5), object(9)
memory usage: 278.5+ KB
```

```
In [170]: # 3.2 - Merge 'url_im_df_clean' with the new 'tweets_merged' dataframe
          # Define the code
          df_tweets_merged = pd.merge(left=tweets_merged, right=url_img_df_clean)
```

```
In [171]: # 3.2 - Test the code
          df_tweets_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1971 entries, 0 to 1970
Data columns (total 27 columns):
tweet_id                1971 non-null int64
timestamp               1971 non-null datetime64[ns]
source                  1971 non-null object
text                    1971 non-null object
expanded_urls           1971 non-null object
rating_numerator        1971 non-null int64
rating_denominator      1971 non-null int64
name                    1971 non-null object
doggo                   1971 non-null object
floofer                 1971 non-null object
pupper                  1971 non-null object
puppo                   1971 non-null object
new_rating              1971 non-null float64
stage                   1971 non-null object
favorite_count          1971 non-null int64
retweet_count           1971 non-null int64
jpg_url                 1971 non-null object
img_num                 1971 non-null int64
p1                      1971 non-null object
p1_conf                 1971 non-null float64
p1_dog                  1971 non-null bool
p2                      1971 non-null object
p2_conf                 1971 non-null float64
p2_dog                  1971 non-null bool
p3                      1971 non-null object
p3_conf                 1971 non-null float64
p3_dog                  1971 non-null bool
dtypes: bool(3), datetime64[ns](1), float64(4), int64(6), object(13)
```

```
memory usage: 390.7+ KB
```

```
In [172]: df_tweets_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1971 entries, 0 to 1970
Data columns (total 27 columns):
tweet_id              1971 non-null int64
timestamp             1971 non-null datetime64[ns]
source                1971 non-null object
text                  1971 non-null object
expanded_urls         1971 non-null object
rating_numerator      1971 non-null int64
rating_denominator    1971 non-null int64
name                  1971 non-null object
doggo                 1971 non-null object
floofer               1971 non-null object
pupper                1971 non-null object
puppo                 1971 non-null object
new_rating            1971 non-null float64
stage                 1971 non-null object
favorite_count        1971 non-null int64
retweet_count         1971 non-null int64
jpg_url               1971 non-null object
img_num               1971 non-null int64
p1                    1971 non-null object
p1_conf               1971 non-null float64
p1_dog                1971 non-null bool
p2                    1971 non-null object
p2_conf               1971 non-null float64
p2_dog                1971 non-null bool
p3                    1971 non-null object
p3_conf               1971 non-null float64
p3_dog                1971 non-null bool
dtypes: bool(3), datetime64[ns](1), float64(4), int64(6), object(13)
memory usage: 390.7+ KB
```

```
In [173]: # 3.3 - Strip 'source' column to one variable, https://stackoverflow.com/questions/259
          # Define code, first get the devices into their own column
          df_tweets_merged['device'] = df_tweets_merged['source'].str.split().str[-1]

In [174]: # 3.3 - Strip '</a>' out of the new 'device' column, https://stackoverflow.com/questio
          df_tweets_merged['device'] = df_tweets_merged['device'].str.replace('</a>','')

In [175]: # Check to see if code extracted device
          df_tweets_merged.head()
```

```
Out[175]:
             tweet_id           timestamp  \
0  892420643555336193 2017-08-01 16:23:56
1  892177421306343426 2017-08-01 00:17:27
2  891815181378084864 2017-07-31 00:18:03
3  891689557279858688 2017-07-30 15:58:51
4  891327558926688256 2017-07-29 16:00:24

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

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

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

   rating_denominator      name doggo floofer   ...             p1   p1_conf  \
0                  10   Phineas                 ...         orange  0.097049
1                  10     Tilly                 ...      Chihuahua  0.323581
2                  10    Archie                 ...      Chihuahua  0.716012
3                  10     Darla                 ...    paper_towel  0.170278
4                  10  Franklin                 ...         basset  0.555712

   p1_dog                  p2   p2_conf  p2_dog                           p3  \
0   False               bagel  0.085851   False                       banana
1    True            Pekinese  0.090647    True                     papillon
2    True            malamute  0.078253    True                       kelpie
3   False  Labrador_retriever  0.168086    True                      spatula
4    True    English_springer  0.225770    True  German_short-haired_pointer

    p3_conf p3_dog  device
0  0.076110  False  iPhone
1  0.068957   True  iPhone
2  0.031379   True  iPhone
3  0.040836  False  iPhone
4  0.175219   True  iPhone
```

```
          [5 rows x 28 columns]

In [176]: # 3.3 cont. - drop original source column
          # Define the code
          new_df_tweets_merged = df_tweets_merged.drop(['source'], axis =1)

          # Test Code

          new_df_tweets_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1971 entries, 0 to 1970
Data columns (total 27 columns):
tweet_id              1971 non-null int64
timestamp             1971 non-null datetime64[ns]
text                  1971 non-null object
expanded_urls         1971 non-null object
rating_numerator      1971 non-null int64
rating_denominator    1971 non-null int64
name                  1971 non-null object
doggo                 1971 non-null object
floofer               1971 non-null object
pupper                1971 non-null object
puppo                 1971 non-null object
new_rating            1971 non-null float64
stage                 1971 non-null object
favorite_count        1971 non-null int64
retweet_count         1971 non-null int64
jpg_url               1971 non-null object
img_num               1971 non-null int64
p1                    1971 non-null object
p1_conf               1971 non-null float64
p1_dog                1971 non-null bool
p2                    1971 non-null object
p2_conf               1971 non-null float64
p2_dog                1971 non-null bool
p3                    1971 non-null object
p3_conf               1971 non-null float64
p3_dog                1971 non-null bool
device                1971 non-null object
dtypes: bool(3), datetime64[ns](1), float64(4), int64(6), object(13)
memory usage: 390.7+ KB
```

```
In [177]: all_tweets_df = new_df_tweets_merged.copy()

In [178]: # Per Udacity reviewer, all 'id' columns should be strings https://stackoverflow.com/q
          # However I need the columns as they are for what I'm doing
# new_df_tweets_merged['tweet_id'] = new_df_tweets_merged['tweet_id'].astype(str)

# Test the code

#new_df_tweets_merged.info()
```

2.4 Store, Analyze and Visualize Data Wrangling

2.4.1 Store

```
In [179]: #Store the file and download to os
          all_tweets_df.to_csv('twitter-archive-master.csv', index=False)
```

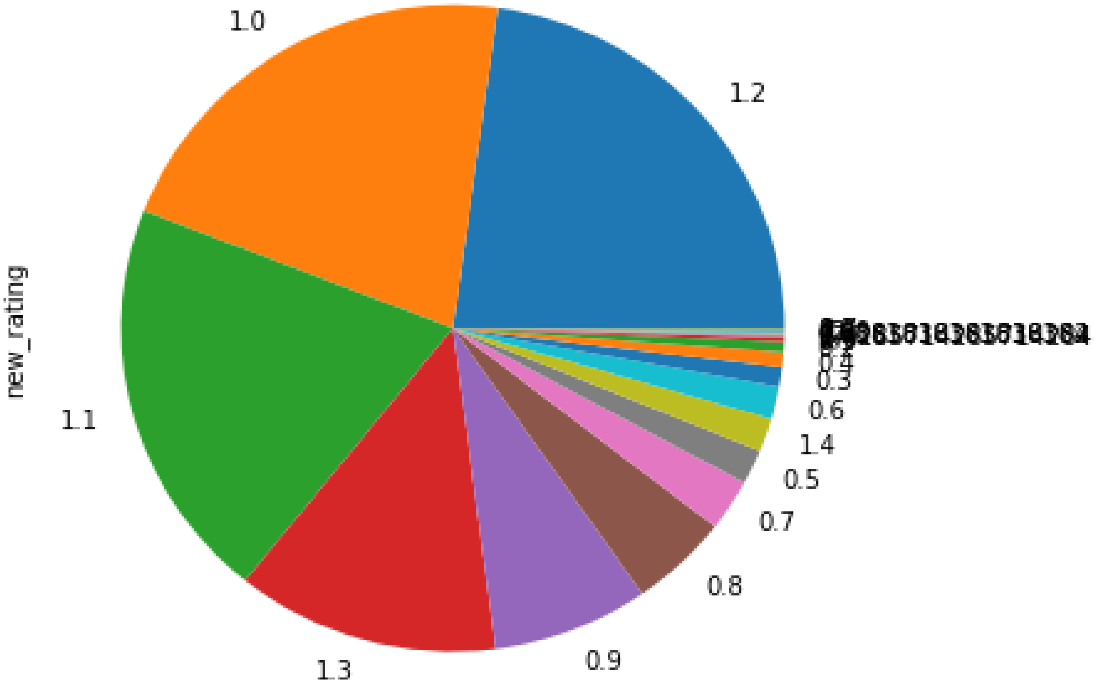
2.4.2 Analyze

```
In [180]: # Investigate ratings using groupby

          ratings_df = all_tweets_df.groupby('new_rating').tweet_id.count()
          ratings_df
```

```
Out[180]: new_rating
          0.000000        1
          0.100000        4
          0.200000       10
          0.300000       19
          0.400000       15
          0.500000       34
          0.600000       32
          0.636364        1
          0.700000       51
          0.800000       95
          0.818182        1
          0.900000      150
          1.000000      419
          1.100000      397
          1.200000      450
          1.300000      253
          1.400000       33
          2.600000        1
          2.700000        1
          3.428571        1
          7.500000        1
          42.000000       1
          177.600000      1
          Name: tweet_id, dtype: int64
```

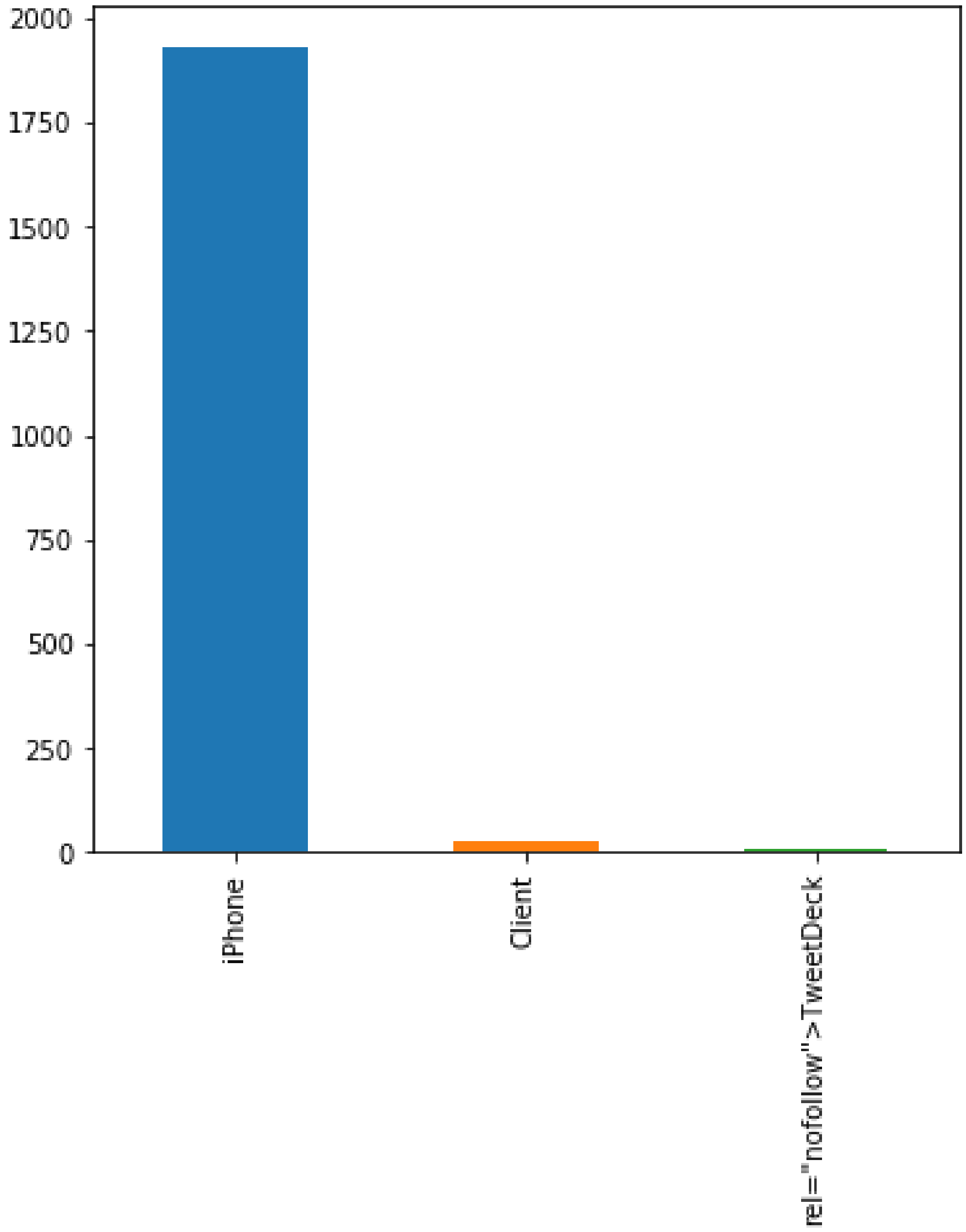
```
In [181]: # Here we can see that most ratings fall between 1.0 and 1.3.
          # Maybe we should consider exploring and removing the outliers? Are these mistakes?
          import matplotlib.pyplot as plt
```

```
        % matplotlib inline

        all_tweets_df['new_rating'].value_counts().plot(kind='pie', figsize=(6,6))

Out[181]: <matplotlib.axes._subplots.AxesSubplot at 0x7f554619eeb8>
```



```
In [182]: # Create a DataFrame to explore the relationship between ratings and favorite count
          most_ratings = all_tweets_df[all_tweets_df['new_rating']> .71]
          most_ratings['new_rating'].count()

Out[182]: 1804

In [183]: all_ratings = all_tweets_df['new_rating'].count()
          all_ratings

Out[183]: 1971

In [184]: most_ratings['new_rating'].count()/all_ratings

Out[184]: 0.91527143581938097
```

```
In [185]: #  # Investigate devices using groupby
          device_df = all_tweets_df.groupby('device').tweet_id.count()
          device_df

Out[185]: device
          Client                        28
          iPhone                      1932
          rel="nofollow">TweetDeck      11
          Name: tweet_id, dtype: int64

In [186]: # Chart for devices
          all_tweets_df['device'].value_counts().plot(kind='bar', figsize=(6,6))

Out[186]: <matplotlib.axes._subplots.AxesSubplot at 0x7f55440f3a90>
```



```
In [187]: all_tweets_df['text']

Out[187]: 0       This is Phineas. He's a mystical boy. Only eve...
          1       This is Tilly. She's just checking pup on you...
          2       This is Archie. He is a rare Norwegian Pouncin...
          3       This is Darla. She commenced a snooze mid meal...
          4       This is Franklin. He would like you to stop ca...
          5       Here we have a majestic great white breaching ...
```

```
6       Meet Jax. He enjoys ice cream so much he gets ...
7       When you watch your owner call another dog a g...
8       This is Zoey. She doesn't want to be one of th...
9       This is Cassie. She is a college pup. Studying...
10      This is Koda. He is a South Australian decksha...
11      This is Bruno. He is a service shark. Only get...
12      Here's a puppo that seems to be on the fence a...
13      This is Ted. He does his best. Sometimes that'...
14      This is Stuart. He's sporting his favorite fan...
15      This is Oliver. You're witnessing one of his m...
16      This is Jim. He found a fren. Taught him how t...
17      This is Zeke. He has a new stick. Very proud o...
18      This is Ralphus. He's powering up. Attempting ...
19      This is Gerald. He was just told he didn't get...
20      This is Jeffrey. He has a monopoly on the pool...
21      I've yet to rate a Venezuelan Hover Wiener. Th...
22      This is Canela. She attempted some fancy porch...
23      You may not have known you needed to see this ...
24      This... is a Jubilant Antarctic House Bear. We...
25      This is Maya. She's very shy. Rarely leaves he...
26      This is Mingus. He's a wonderful father to his...
27      This is Derek. He's late for a dog meeting. 13...
28      This is Roscoe. Another pupper fallen victim t...
29      This is Waffles. His doggles are pupside down...
                                ...
1941    This is quite the dog. Gets really excited whe...
1942    This is a southern Vesuvius bumblegruff. Can d...
1943    Oh goodness. A super rare northeast Qdoba kang...
1944    Those are sunglasses and a jean jacket. 11/10 ...
1945    Unique dog here. Very small. Lives in containe...
1946    Here we have a mixed Asiago from the Galápagos...
1947    Look at this jokester thinking seat belt laws ...
1948    This is an extremely rare horned Parthenon. No...
1949    This is a funny dog. Weird toes. Won't come do...
1950    This is an Albanian 3 1/2 legged  Episcopalian...
1951       Can take selfies 11/10 https://t.co/ws2AMaNwPW
1952    Very concerned about fellow dog trapped in com...
1953    Not familiar with this breed. No tail (weird)...
1954    Oh my. Here you are seeing an Adobe Setter giv...
1955    Can stand on stump for what seems like a while...
1956    This appears to be a Mongolian Presbyterian mi...
1957    Here we have a well-established sunblockerspan...
1958    Let's hope this flight isn't Malaysian (lol). ...
1959    Here we have a northern speckled Rhododendron...
1960    This is the happiest dog you will ever see. Ve...
1961    Here is the Rand Paul of retrievers folks! He'...
1962    My oh my. This is a rare blond Canadian terrie...
1963    Here is a Siberian heavily armored polar bear ...
```

```
1964    This is an odd dog. Hard on the outside but lo...
1965    This is a truly beautiful English Wilson Staff...
1966    Here we have a 1949 1st generation vulpix. Enj...
1967    This is a purebred Piers Morgan. Loves to Netf...
1968    Here is a very happy pup. Big fan of well-main...
1969    This is a western brown Mitsubishi terrier. Up...
1970    Here we have a Japanese Irish Setter. Lost eye...
Name: text, Length: 1971, dtype: object
```

2.4.3 Report

Report on the data findings

The analysis shows that most tweets came from an iphone, with a count total of 1,932. There were twenty-eight devices classified as a "client" and eleven as "tweet deck". I looked at devices because I was wondering if the size of the viewing screen impacted the ratings, but it doesn't appear there was enough variety in device types to have an impact on ratings.

The bulk of the ratings, 91.52%, were above .71. I devided the numerator column by the denominator column, even though it had a unique structure where the numerators were larger than the denominators. I would trim the rows with outlier ratings off of the rating data before using it in dog type comparisons.

The data could be used to determine which types of dogs appeared to have higher ratings, or were higher ratings correlated with retweet_count or favorite_count. Dog types could be compared to favorite_count, and then again to retweet_count. Also, there is a lot of missing doogo, floofer, pupper and puppo data. How could I fill this in? Where could I get the missing data.

Finally, you could see if dates, times or hashtags correlated with ratings, favorite_count, and retweet count. The analysis could be used to determine the best time for WeRateDogs to tweet promotional tweets, in months, days, or times. What is #BarkWeek? When is #BarkWeek? Is this prime promotion time for WeRateDogs?

```
In [ ]:
```

```
In [ ]:
```