wrangle_act

August 18, 2021

1 Project Data wrangling " WeRateDogs - 2021"

1.1 Table of Content

Introduction

Gathering data

Assessing data

Cleaning data

2 Introduction

The objective of this project is to verify three basic operations in dealing with data, which are:Gathering data , Assessing data and Cleaning data. This is to reach more accurate data and give realistic results and solutions that can be used. In this project, we discussed obtaining Twitter user @dog_rates, also known as WeRateDogs.

Here in this project, I used the data that Udacity presented in the project file, because of the delay of Twitter in responding to the request for data, the rooms and the goal are the same in both cases

2.1 What is Data Wrangling?

Data wrangling is the process of cleaning, structuring and enriching raw data into a desired format for better decision making in less time. Data wrangling is increasingly ubiquitous at today's top firms. Data has become more diverse and unstructured, demanding increased time spent culling, cleaning, and organizing data ahead of broader analysis. At the same time, with data informing just about every business decision, business users have less time to wait on technical resources for prepared data.

2.2 What is Gathering Data ?

Gathering Data is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes.

2.2.1 How to Gathering Data ?

We will collect this data

Twitter archive file: download this file manually by clicking the following link: twitter_archive_enhanced.csv

The tweet image predictions, i.e.This file (image_predictions.tsv) is hosted on Udacity's servers and should be downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

Twitter JSON

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import requests
import json
import datetime
import seaborn as sns
import tweepy
from PIL import Image
```

2.2.2 1- Twitter archive file

[2]:
```
twitter = pd.read_csv('twitter-archive-enhanced-2.csv')
```

[3]:
```
twitter.head()
```

[3]:
```
             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
```

[4]:
```
twitter.shape
```

[4]:
```
(2356, 17)
```

[5]:
```
twitter.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   tweet_id                    2356 non-null   int64
 1   in_reply_to_status_id       78 non-null     float64
 2   in_reply_to_user_id         78 non-null     float64
 3   timestamp                   2356 non-null   object
```

```
 4   source                      2356 non-null   object
 5   text                        2356 non-null   object
 6   retweeted_status_id         181 non-null    float64
 7   retweeted_status_user_id    181 non-null    float64
 8   retweeted_status_timestamp  181 non-null    object
 9   expanded_urls               2297 non-null   object
 10  rating_numerator            2356 non-null   int64
 11  rating_denominator          2356 non-null   int64
 12  name                        2356 non-null   object
 13  doggo                       2356 non-null   object
 14  floofer                     2356 non-null   object
 15  pupper                      2356 non-null   object
 16  puppo                       2356 non-null   object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

[6]:
```
twitter.dtypes
```

[6]:
```
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
```

2.2.3 2- Tweet image prediction

[7]:
```
url = "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/
 ↪599fd2ad_image-predictions/image-predictions.tsv"
response = requests.get(url)
with open('image-predictions-3.tsv', 'wb') as file:
    file.write(response.content)

image_predictions = pd.read_csv('image-predictions-3.tsv', sep='\t')
```

```
image_predictions.head(30)
```

[7]:

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826	True
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.596461	True
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.408143	True
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.560311	True
5	666050758794694657	https://pbs.twimg.com/media/CT5Jof1WUAEuVxN.jpg	1	Bernese_mountain_dog	0.651137	True
6	666051853826850816	https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg	1	box_turtle	0.933012	False
7	666055525042405380	https://pbs.twimg.com/media/CT5N9tpXIAAifs1.jpg	1	chow	0.692517	True
8	666057090499244032	https://pbs.twimg.com/media/CT5PY90WoAAQGLo.jpg	1	shopping_cart	0.962465	False
9	666058600524156928	https://pbs.twimg.com/media/CT5Qw94XAAA_2dP.jpg	1	miniature_poodle	0.201493	True
10	666063827256086533	https://pbs.twimg.com/media/CT5Vg_wXIAAXfnj.jpg	1	golden_retriever	0.775930	True
11	666071193221509120	https://pbs.twimg.com/media/CT5cN_3WEAAlOoZ.jpg	1	Gordon_setter	0.503672	True
12	666073100786774016	https://pbs.twimg.com/media/CT5d9DZXAAALcwe.jpg	1	Walker_hound	0.260857	True
13	666082916733198337	https://pbs.twimg.com/media/CT5m4VGWEAAtKc8.jpg	1	pug	0.489814	True
14	666094000022159362	https://pbs.twimg.com/media/CT5w9gUW4AAsBNN.jpg	1	bloodhound	0.195217	True
15	666099513787052032	https://pbs.twimg.com/media/CT51-JJUEAA6hV8.jpg	1	Lhasa	0.582842	True
16	666102155909144576	https://pbs.twimg.com/media/CT54YGiWUAEZnoK.jpg	1	English_springer	0.450567	True
17	666104133288665088	https://pbs.twimg.com/media/CT56LSZWoAAlJj2.jpg	1	German_shepherd	0.596461	True
18	666268910803644416	https://pbs.twimg.com/media/CT8QCd1WEAADXws.jpg	1	Siberian_husky	0.957979	True
19	666273097616637952	https://pbs.twimg.com/media/CT8T1mtUwAA3aqm.jpg	1	Rottweiler	0.243682	True
20	666287406224695296	https://pbs.twimg.com/media/CT8g3BpUEAAuFjg.jpg	1	basset	0.555712	True
21	666293911632134144	https://pbs.twimg.com/media/CT8mx7KW4AEQu8N.jpg	1	English_springer	0.425595	True
22	666337882303524864	https://pbs.twimg.com/media/CT9OwFIWEAMuRje.jpg	1	German_short-haired_pointer	0.875481	True
23	666345417576210432	https://pbs.twimg.com/media/CT9Vn7PWoAA_ZCM.jpg	1	Rottweiler	0.566142	True
24	666353288456101888	https://pbs.twimg.com/media/CT9cx0tUEAAhNN_.jpg	1	Maltese_dog	0.857531	True
25	666362758909284353	https://pbs.twimg.com/media/CT9lXGsUcAAyUFt.jpg	1	Eskimo_dog	0.362381	True
26	666373753744588802	https://pbs.twimg.com/media/CT9vZEYWUAAlZ05.jpg	1	Rottweiler	0.822502	True
27	666396247373291520	https://pbs.twimg.com/media/CT-D2ZHWIAA3gK1.jpg	1	Pembroke	0.943235	True
28	666407126856765440	https://pbs.twimg.com/media/CT-NvwmW4AAugGZ.jpg	1	basset	0.654761	True
29	666411507551481857	https://pbs.twimg.com/media/CT-RugiWIAELEaq.jpg	1	Italian_greyhound	0.176053	True

11	1	Gordon_setter	0.503672	True
12	1	Walker_hound	0.260857	True
13	1	pug	0.489814	True
14	1	bloodhound	0.195217	True
15	1	Lhasa	0.582330	True
16	1	English_setter	0.298617	True
17	1	hen	0.965932	False
18	1	desktop_computer	0.086502	False
19	1	Italian_greyhound	0.176053	True
20	1	Maltese_dog	0.857531	True
21	1	three-toed_sloth	0.914671	False
22	1	ox	0.416669	False
23	1	golden_retriever	0.858744	True
24	1	malamute	0.336874	True
25	1	guinea_pig	0.996496	False
26	1	soft-coated_wheaten_terrier	0.326467	True
27	1	Chihuahua	0.978108	True
28	1	black-and-tan_coonhound	0.529139	True
29	1	coho	0.404640	False

	p2	p2_conf	p2_dog	p3	\
0	collie	0.156665	True	Shetland_sheepdog	
1	miniature_pinscher	0.074192	True	Rhodesian_ridgeback	
2	malinois	0.138584	True	bloodhound	
3	redbone	0.360687	True	miniature_pinscher	
4	Rottweiler	0.243682	True	Doberman	
5	English_springer	0.263788	True	Greater_Swiss_Mountain_dog	
6	mud_turtle	0.045885	False	terrapin	
7	Tibetan_mastiff	0.058279	True	fur_coat	
8	shopping_basket	0.014594	False	golden_retriever	
9	komondor	0.192305	True	soft-coated_wheaten_terrier	
10	Tibetan_mastiff	0.093718	True	Labrador_retriever	
11	Yorkshire_terrier	0.174201	True	Pekinese	
12	English_foxhound	0.175382	True	Ibizan_hound	
13	bull_mastiff	0.404722	True	French_bulldog	
14	German_shepherd	0.078260	True	malinois	
15	Shih-Tzu	0.166192	True	Dandie_Dinmont	
16	Newfoundland	0.149842	True	borzoi	
17	cock	0.033919	False	partridge	
18	desk	0.085547	False	bookcase	
19	toy_terrier	0.111884	True	basenji	
20	toy_poodle	0.063064	True	miniature_poodle	
21	otter	0.015250	False	great_grey_owl	
22	Newfoundland	0.278407	True	groenendael	
23	Chesapeake_Bay_retriever	0.054787	True	Labrador_retriever	
24	Siberian_husky	0.147655	True	Eskimo_dog	
25	skunk	0.002402	False	hamster	

```
26              Afghan_hound  0.259551    True                       briard
27               toy_terrier  0.009397    True                     papillon
28                 bloodhound  0.244220    True       flat-coated_retriever
29                 barracouta  0.271485   False                          gar
```

	p3_conf	p3_dog
0	0.061428	True
1	0.072010	True
2	0.116197	True
3	0.222752	True
4	0.154629	True
5	0.016199	True
6	0.017885	False
7	0.054449	False
8	0.007959	True
9	0.082086	True
10	0.072427	True
11	0.109454	True
12	0.097471	True
13	0.048960	True
14	0.075628	True
15	0.089688	True
16	0.133649	True
17	0.000052	False
18	0.079480	False
19	0.111152	True
20	0.025581	True
21	0.013207	False
22	0.102643	True
23	0.014241	True
24	0.093412	True
25	0.000461	False
26	0.206803	True
27	0.004577	True
28	0.173810	True
29	0.189945	False

[8]:
```
image_predictions.shape
```

[8]:
```
(2075, 12)
```

[9]:
```
image_predictions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
 #   Column    Non-Null Count  Dtype
```

```
---  ------    --------------  -----
 0   tweet_id  2075 non-null   int64
 1   jpg_url   2075 non-null   object
 2   img_num   2075 non-null   int64
 3   p1        2075 non-null   object
 4   p1_conf   2075 non-null   float64
 5   p1_dog    2075 non-null   bool
 6   p2        2075 non-null   object
 7   p2_conf   2075 non-null   float64
 8   p2_dog    2075 non-null   bool
 9   p3        2075 non-null   object
 10  p3_conf   2075 non-null   float64
 11  p3_dog    2075 non-null   bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

2.2.4 3- Twitter JSON

[13]:
```
Json = pd.read_json('tweet-json copy', lines=True)

Json.head(5)
```

[13]:
```
                 created_at                   id              id_str  \
0 2017-08-01 16:23:56+00:00  892420643555336193  892420643555336192
1 2017-08-01 00:17:27+00:00  892177421306343426  892177421306343424
2 2017-07-31 00:18:03+00:00  891815181378084864  891815181378084864
3 2017-07-30 15:58:51+00:00  891689557279858688  891689557279858688
4 2017-07-29 16:00:24+00:00  891327558926688256  891327558926688256

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

  display_text_range                                              entities  \
0            [0, 85]  {'hashtags': [], 'symbols': [], 'user_mentions…
1           [0, 138]  {'hashtags': [], 'symbols': [], 'user_mentions…
2           [0, 121]  {'hashtags': [], 'symbols': [], 'user_mentions…
3            [0, 79]  {'hashtags': [], 'symbols': [], 'user_mentions…
4           [0, 138]  {'hashtags': [{'text': 'BarkWeek', 'indices': …

                                    extended_entities  \
0  {'media': [{'id': 892420639486877696, 'id_str'…
1  {'media': [{'id': 892177413194625024, 'id_str'…
```

```
2  {'media': [{'id': 891815175371796480, 'id_str'…
3  {'media': [{'id': 891689552724799489, 'id_str'…
4  {'media': [{'id': 891327551943041024, 'id_str'…

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

   …  favorite_count  favorited  retweeted possibly_sensitive  \
0  …           39467      False      False                0.0
1  …           33819      False      False                0.0
2  …           25461      False      False                0.0
3  …           42908      False      False                0.0
4  …           41048      False      False                0.0

  possibly_sensitive_appealable  lang  retweeted_status quoted_status_id  \
0                           0.0    en               NaN              NaN
1                           0.0    en               NaN              NaN
2                           0.0    en               NaN              NaN
3                           0.0    en               NaN              NaN
4                           0.0    en               NaN              NaN

   quoted_status_id_str  quoted_status
0                   NaN            NaN
1                   NaN            NaN
2                   NaN            NaN
3                   NaN            NaN
4                   NaN            NaN

[5 rows x 31 columns]
```

[14]:
```
Json.shape
```

[14]:
```
(2354, 31)
```

[15]:
```
Json.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 31 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   created_at                     2354 non-null   datetime64[ns, UTC]
 1   id                             2354 non-null   int64
```

```
 2   id_str                          2354 non-null   int64
 3   full_text                       2354 non-null   object
 4   truncated                       2354 non-null   bool
 5   display_text_range              2354 non-null   object
 6   entities                        2354 non-null   object
 7   extended_entities               2073 non-null   object
 8   source                          2354 non-null   object
 9   in_reply_to_status_id           78 non-null     float64
 10  in_reply_to_status_id_str       78 non-null     float64
 11  in_reply_to_user_id             78 non-null     float64
 12  in_reply_to_user_id_str         78 non-null     float64
 13  in_reply_to_screen_name         78 non-null     object
 14  user                            2354 non-null   object
 15  geo                             0 non-null      float64
 16  coordinates                     0 non-null      float64
 17  place                           1 non-null      object
 18  contributors                    0 non-null      float64
 19  is_quote_status                 2354 non-null   bool
 20  retweet_count                   2354 non-null   int64
 21  favorite_count                  2354 non-null   int64
 22  favorited                       2354 non-null   bool
 23  retweeted                       2354 non-null   bool
 24  possibly_sensitive              2211 non-null   float64
 25  possibly_sensitive_appealable   2211 non-null   float64
 26  lang                            2354 non-null   object
 27  retweeted_status                179 non-null    object
 28  quoted_status_id                29 non-null     float64
 29  quoted_status_id_str            29 non-null     float64
 30  quoted_status                   28 non-null     object
dtypes: bool(4), datetime64[ns, UTC](1), float64(11), int64(4), object(11)
memory usage: 505.9+ KB
```

[59]:
```
selected_attr = []
with open('tweet-json copy', 'r') as json_file:
    for line in json_file:
        json_data = json.loads(line)

        # create a dictionary with the JSON data, then add to a list tweet_id,␣
 ↪favorites, retweets from the JSON data
        selected_attr.append({'tweet_id': json_data['id'],
                              'favorites': json_data['favorite_count'],
                              'retweets': json_data['retweet_count']})
# convert the tweet JSON data dictionary list to a DataFrame
Json = pd.DataFrame(selected_attr, columns=['tweet_id','favorites','retweets'])
```

[60]:
```
## Test
Json.head()
```

[60]:

	tweet_id	favorites	retweets
0	892420643555336193	39467	8853
1	892177421306343426	33819	6514
2	891815181378084864	25461	4328
3	891689557279858688	42908	8964
4	891327558926688256	41048	9774

2.3 Assessing Data and Cleaning data

2.3.1 What is Assessing Data ?

Data Quality Assessment is a distinct phase within the data quality life-cycle that is used to verify the source, quantity and impact of any data items that breach pre-defined data quality rules. The Data Quality Assessment is a task typically executed by dedicated Data Quality Software

2.3.2 Quality And Tidiness Issues:

Quality Issues: Delete cells with a lot of missing values

Drop values that contain repeated information

Drop duplicates in jpg_url column

Remove html tags in source column

The columns (retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp) have only 181 values and 2175 missing values (Null values), and retweeted_status_timestamp Column has invalid format.

Create a backup to preserve the original files

The Column timestamp has invalid format, it should be a datetime type.

The Column tweet_id Change from an integer to string .

Tidiness Issues: Combine " 'doggo', 'floofer', 'pupper', 'puppo' "columns into a single column called " Dog_Stage "

Information about one type of observational unit (tweets) is spread across three different files/dataframes.

Drop columns with missing values: 'in_reply_to_status_id','in_reply_to_user_id','retweeted_status_id', 'retweeted_status_user_id','retweeted_status_timestamp','expanded_urls

The file has 2354 rows, the same of tweeter archive file, no issu Merge the columns (p1, p1_dog, p2, p2_dog, p3, p3_dog) in one column called: image prediction , and Merge the columns(p1_conf, p2_conf, p3_conf) in one column called confidence level.

Merge all three dataset and remove repetitive columns. Take both the twitter_clean and Json_clean tables and image_clean merge into one table using the join() method on the columns tweet_id.

2.3.3 What is Cleaning data ?

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled

1- Twitter archive file | twitter-archive-enhanced-2.csv

[16]:
```
twitter
```

[16]:
```
                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
…                    …                      …                    …
2351  666049248165822465                    NaN                  NaN
2352  666044226329800704                    NaN                  NaN
2353  666033412701032449                    NaN                  NaN
2354  666029285002620928                    NaN                  NaN
2355  666020888022790149                    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
…                           …
2351  2015-11-16 00:24:50 +0000
2352  2015-11-16 00:04:52 +0000
2353  2015-11-15 23:21:54 +0000
2354  2015-11-15 23:05:30 +0000
2355  2015-11-15 22:32:08 +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…
…                                                   …
2351  <a href="http://twitter.com/download/iphone" r…
```

```
2352  <a href="http://twitter.com/download/iphone" r…
2353  <a href="http://twitter.com/download/iphone" r…
2354  <a href="http://twitter.com/download/iphone" r…
2355  <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
…                                                   …                  …
2351  Here we have a 1949 1st generation vulpix. Enj…                  NaN
2352  This is a purebred Piers Morgan. Loves to Netf…                  NaN
2353  Here is a very happy pup. Big fan of well-main…                  NaN
2354  This is a western brown Mitsubishi terrier. Up…                  NaN
2355  Here we have a Japanese Irish Setter. Lost eye…                  NaN

      retweeted_status_user_id retweeted_status_timestamp  \
0                          NaN                         NaN
1                          NaN                         NaN
2                          NaN                         NaN
3                          NaN                         NaN
4                          NaN                         NaN
…                          …                           …
2351                       NaN                         NaN
2352                       NaN                         NaN
2353                       NaN                         NaN
2354                       NaN                         NaN
2355                       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
…                                                   …                 …
2351  https://twitter.com/dog_rates/status/666049248…                  5
2352  https://twitter.com/dog_rates/status/666044226…                  6
2353  https://twitter.com/dog_rates/status/666033412…                  9
2354  https://twitter.com/dog_rates/status/666029285…                  7
2355  https://twitter.com/dog_rates/status/666020888…                  8

      rating_denominator      name doggo floofer pupper puppo
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
…                   …        …    …       …     …    …
2351                 10      None  None    None    None  None
2352                 10         a  None    None    None  None
2353                 10         a  None    None    None  None
2354                 10         a  None    None    None  None
2355                 10      None  None    None    None  None

[2356 rows x 17 columns]
```

[17]:
```
twitter.nunique()
```

[17]:
```
tweet_id                    2356
in_reply_to_status_id         77
in_reply_to_user_id           31
timestamp                   2356
source                         4
text                        2356
retweeted_status_id          181
retweeted_status_user_id      25
retweeted_status_timestamp   181
expanded_urls               2218
rating_numerator              40
rating_denominator            18
name                         957
doggo                          2
floofer                        2
pupper                         2
puppo                          2
dtype: int64
```

[18]:
```
twitter.isnull().sum()
```

[18]:
```
tweet_id                       0
in_reply_to_status_id       2278
in_reply_to_user_id         2278
timestamp                      0
source                         0
text                           0
retweeted_status_id         2175
retweeted_status_user_id    2175
retweeted_status_timestamp  2175
expanded_urls                 59
rating_numerator               0
rating_denominator             0
```

```
name                            0
doggo                           0
floofer                         0
pupper                          0
puppo                           0
dtype: int64
```

[23]:
```
twitter.duplicated().sum()
```

[23]:
```
0
```

[19]:
```
sum(twitter['tweet_id'].duplicated())
```

[19]:
```
0
```

[26]:
```
twitter.describe()
```

[26]:

	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

2-The tweet image predictions, i.e.This file (image_predictions.tsv)

[20]:
```
image_predictions.nunique()
```

[20]:
```
tweet_id    2075
jpg_url     2009
img_num        4
p1           378
p1_conf     2006
p1_dog         2
p2           405
p2_conf     2004
p2_dog         2
p3           408
p3_conf     2006
p3_dog         2
dtype: int64
```

[21]:
```
image_predictions.isnull().sum()
```

[21]:
```
tweet_id    0
jpg_url     0
img_num     0
p1          0
p1_conf     0
p1_dog      0
p2          0
p2_conf     0
p2_dog      0
p3          0
p3_conf     0
p3_dog      0
dtype: int64
```

[22]:
```
image_predictions.duplicated().sum()
```

[22]:
```
0
```

[24]:
```
sum(image_predictions.jpg_url.duplicated())
```

[24]:
```
66
```

[25]:
```
image_predictions.describe()
```

[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
```

[30]:
```
print(image_predictions.p1_dog.value_counts())
print(image_predictions.p2_dog.value_counts())
print(image_predictions.p3_dog.value_counts())
```

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

3- Twitter JSON

[62]:
```
Json.isnull().sum()
```

[62]:
```
tweet_id     0
favorites    0
retweets     0
dtype: int64
```

[ ]:
```

```

2.3.4 Cleaning Data

[111]:
```
# Making a copy of all  dataframes before data cleaning keep orginals data

twitter_clean = twitter.copy()
image_clean = image_predictions.copy()
Json_clean = Json.copy()
```

Define Change format for timestamp and tweet_id column: The Column tweet_id change from an integer to string and timestamp has invalid format, it should be a datetime type.

Code

[94]:
```
twitter_clean['timestamp'] = pd.to_datetime(twitter_clean['timestamp'])
twitter_clean['tweet_id'] = twitter_clean['tweet_id'].astype(str)
image_clean['tweet_id'] = twitter_clean['tweet_id'].astype(str)
Json_clean['tweet_id'] = Json_clean['tweet_id'].astype(str)
```

Test

[95]:
```
twitter_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   tweet_id                    2356 non-null   object
 1   in_reply_to_status_id       78 non-null     float64
 2   in_reply_to_user_id         78 non-null     float64
 3   timestamp                   2356 non-null   datetime64[ns, UTC]
 4   source                      2356 non-null   object
 5   text                        2356 non-null   object
 6   retweeted_status_id         181 non-null    float64
 7   retweeted_status_user_id    181 non-null    float64
 8   retweeted_status_timestamp  181 non-null    object
 9   expanded_urls               2297 non-null   object
 10  rating_numerator            2356 non-null   int64
 11  rating_denominator          2356 non-null   int64
 12  name                        2356 non-null   object
 13  doggo                       2356 non-null   object
 14  floofer                     2356 non-null   object
 15  pupper                      2356 non-null   object
 16  puppo                       2356 non-null   object
dtypes: datetime64[ns, UTC](1), float64(4), int64(2), object(10)
memory usage: 313.0+ KB
```

[96]:
```
image_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   tweet_id  2075 non-null   object
 1   jpg_url   2075 non-null   object
 2   img_num   2075 non-null   int64
 3   p1        2075 non-null   object
 4   p1_conf   2075 non-null   float64
 5   p1_dog    2075 non-null   bool
```

```
 6   p2        2075 non-null   object
 7   p2_conf   2075 non-null   float64
 8   p2_dog    2075 non-null   bool
 9   p3        2075 non-null   object
 10  p3_conf   2075 non-null   float64
 11  p3_dog    2075 non-null   bool
dtypes: bool(3), float64(3), int64(1), object(5)
memory usage: 152.1+ KB
```

[97]:
```
Json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   tweet_id   2354 non-null   object
 1   favorites  2354 non-null   int64
 2   retweets   2354 non-null   int64
dtypes: int64(2), object(1)
memory usage: 55.3+ KB
```

Define Remove html tags in source column

Code

[105]:
```
twitter_clean.source = twitter_clean.source.str.replace(r'<(?:a\b[^>]*>|/a>)',␣
↪'')
twitter_clean.source = twitter_clean.source.astype('category')
```

```
<ipython-input-105-ea97778a6e9d>:1: FutureWarning: The default value of regex
will change from True to False in a future version.
  twitter_clean.source = twitter_clean.source.str.replace(r'<(?:a\b[^>]*>|/a>)',
'')
```

Test

[106]:
```
twitter_clean['source'].head(5)
```

[106]:
```
0    Twitter for iPhone
1    Twitter for iPhone
2    Twitter for iPhone
3    Twitter for iPhone
4    Twitter for iPhone
Name: source, dtype: category
Categories (4, object): ['TweetDeck', 'Twitter Web Client', 'Twitter for
iPhone', 'Vine - Make a Scene']
```

Define Remove duplicates of tweets in the row

Code

[107]:
```
twitter_clean.dropna(subset = ["rating_numerator"], inplace=True)
```

[108]:
```
twitter_clean.dropna(subset = ["retweeted_status_id"], inplace=True)
```

Test

[109]:
```
print (twitter_clean)
```

```
                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
19    888202515573088257                    NaN                  NaN
32    886054160059072513                    NaN                  NaN
36    885311592912609280                    NaN                  NaN
68    879130579576475649                    NaN                  NaN
73    878404777348136964                    NaN                  NaN
…                    …                      …                    …
1023  746521445350707200                    NaN                  NaN
1043  743835915802583040                    NaN                  NaN
1242  711998809858043904                    NaN                  NaN
2259  667550904950915073                    NaN                  NaN
2260  667550882905632768                    NaN                  NaN

                     timestamp              source  \
19    2017-07-21 01:02:36 +0000  Twitter for iPhone
32    2017-07-15 02:45:48 +0000  Twitter for iPhone
36    2017-07-13 01:35:06 +0000  Twitter for iPhone
68    2017-06-26 00:13:58 +0000  Twitter for iPhone
73    2017-06-24 00:09:53 +0000  Twitter for iPhone
…                           …                  …
1023  2016-06-25 01:52:36 +0000  Twitter for iPhone
1043  2016-06-17 16:01:16 +0000  Twitter for iPhone
1242  2016-03-21 19:31:59 +0000  Twitter for iPhone
2259  2015-11-20 03:51:52 +0000  Twitter Web Client
2260  2015-11-20 03:51:47 +0000  Twitter Web Client

                                                   text  retweeted_status_id  \
19    RT @dog_rates: This is Canela. She attempted s…         8.874740e+17
32    RT @Athletics: 12/10 #BATP https://t.co/WxwJmv…         8.860537e+17
36    RT @dog_rates: This is Lilly. She just paralle…         8.305833e+17
68    RT @dog_rates: This is Emmy. She was adopted t…         8.780576e+17
73    RT @dog_rates: Meet Shadow. In an attempt to r…         8.782815e+17
…                                                   …                    …
1023  RT @dog_rates: This is Shaggy. He knows exactl…         6.678667e+17
1043  RT @dog_rates: Extremely intelligent dog here…         6.671383e+17
1242  RT @twitter: @dog_rates Awesome Tweet! 12/10. …         7.119983e+17
```

```
2259  RT @dogratingrating: Exceptional talent. Origi…         6.675487e+17
2260  RT @dogratingrating: Unoriginal idea. Blatant …         6.675484e+17

      retweeted_status_user_id retweeted_status_timestamp  \
19                4.196984e+09  2017-07-19 00:47:34 +0000
32                1.960740e+07  2017-07-15 02:44:07 +0000
36                4.196984e+09  2017-02-12 01:04:29 +0000
68                4.196984e+09  2017-06-23 01:10:23 +0000
73                4.196984e+09  2017-06-23 16:00:04 +0000
…                          …                          …
1023              4.196984e+09  2015-11-21 00:46:50 +0000
1043              4.196984e+09  2015-11-19 00:32:12 +0000
1242              7.832140e+05  2016-03-21 19:29:52 +0000
2259              4.296832e+09  2015-11-20 03:43:06 +0000
2260              4.296832e+09  2015-11-20 03:41:59 +0000

                                          expanded_urls  rating_numerator  \
19    https://twitter.com/dog_rates/status/887473957…                 13
32    https://twitter.com/dog_rates/status/886053434…                 12
36    https://twitter.com/dog_rates/status/830583320…                 13
68    https://twitter.com/dog_rates/status/878057613…                 14
73    https://www.gofundme.com/3yd6y1c,https://twitt…                 13
…                                                   …                …
1023  https://twitter.com/dog_rates/status/667866724…                 10
1043  https://twitter.com/dog_rates/status/667138269…                 10
1242  https://twitter.com/twitter/status/71199827977…                 12
2259  https://twitter.com/dogratingrating/status/667…                 12
2260  https://twitter.com/dogratingrating/status/667…                  5

      rating_denominator    name doggo floofer pupper puppo
19                    10  Canela  None    None   None  None
32                    10    None  None    None   None  None
36                    10   Lilly  None    None   None  None
68                    10    Emmy  None    None   None  None
73                    10  Shadow  None    None   None  None
…                  …      …   …     …    …   …
1023                  10  Shaggy  None    None   None  None
1043                  10    None  None    None   None  None
1242                  10    None  None    None   None  None
2259                  10    None  None    None   None  None
2260                  10    None  None    None   None  None

[181 rows x 17 columns]
```

Define Drop columns with missing values: 'in_reply_to_status_id','in_reply_to_user_id','retweeted_status_
'retweeted_status_user_id','retweeted_status_timestamp','expanded_urls'

Code

[82]:
```
### Delete columns no needed
twitter_clean = twitter_clean.
 ↪drop(['in_reply_to_status_id','in_reply_to_user_id','retweeted_status_id',
'retweeted_status_user_id','retweeted_status_timestamp','expanded_urls'], 1)
```

Test

[35]:
```
list(twitter_clean)
```

[35]:
```
['tweet_id',
 'timestamp',
 'source',
 'text',
 'rating_numerator',
 'rating_denominator',
 'name',
 'doggo',
 'floofer',
 'pupper',
 'puppo']
```

Define Combine " 'doggo', 'floofer', 'pupper', 'puppo' "columns into a single column called " Dog_Stage "

Code

[39]:
```
twitter_clean[['doggo', 'floofer', 'pupper', 'puppo']].describe()
```

[39]:

	doggo	floofer	pupper	puppo
count	2356	2356	2356	2356
unique	2	2	2	2
top	None	None	None	None
freq	2259	2346	2099	2326

[40]:
```
twitter_clean['doggo'].replace('None', '', inplace=True)
twitter_clean['floofer'].replace('None', '', inplace=True)
twitter_clean['pupper'].replace('None', '', inplace=True)
twitter_clean['puppo'].replace('None', '', inplace=True)
```

[41]:
```
twitter_clean['Dog_Stage'] = twitter_clean['doggo'] + twitter_clean['floofer']␣
 ↪+twitter_clean['pupper'] + twitter_clean['puppo']
```

[42]:
```
twitter_clean.loc[twitter_clean.Dog_Stage == '', 'Dog_Stage'] = np.nan
```

[43]:
```
twitter_clean['Dog_Stage'].value_counts()
```

[43]:
```
pupper          245
doggo            83
puppo            29
doggopupper      12
floofer           9
doggopuppo        1
doggofloofer      1
Name: Dog_Stage, dtype: int64
```

[44]:
```
twitter_clean.loc[twitter_clean.Dog_Stage == 'doggopupper', 'Dog_Stage'] =␣
↪'doggo, pupper'
twitter_clean.loc[twitter_clean.Dog_Stage == 'doggofloofer', 'Dog_Stage'] =␣
↪'doggo, floofer'
twitter_clean.loc[twitter_clean.Dog_Stage == 'doggopuppo', 'Dog_Stage'] =␣
↪'doggo, puppo'
```

[45]:
```
twitter_clean['Dog_Stage'].value_counts()
```

[45]:
```
pupper            245
doggo              83
puppo              29
doggo, pupper      12
floofer             9
doggo, puppo        1
doggo, floofer      1
Name: Dog_Stage, dtype: int64
```

[46]:
```
twitter_clean['Dog_Stage'].isnull().value_counts()
```

[46]:
```
True     1976
False     380
Name: Dog_Stage, dtype: int64
```

[47]:
```
twitter_clean = twitter_clean.drop(['pupper','doggo','puppo','floofer'], axis =␣
↪1)
```

Test

[48]:
```
## Test

twitter_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
```

```
 0   tweet_id            2356 non-null   int64
 1   timestamp           2356 non-null   object
 2   source              2356 non-null   object
 3   text                2356 non-null   object
 4   rating_numerator    2356 non-null   int64
 5   rating_denominator  2356 non-null   int64
 6   name                2356 non-null   object
 7   Dog_Stage           380 non-null    object
dtypes: int64(3), object(5)
memory usage: 147.4+ KB
```

2- image predictions, i.e.This file (image_predictions.tsv)

Define Drop duplicates in jpg_url column

[50]:
```
sum(image_clean.jpg_url.duplicated())
```

[50]:
```
66
```

Code

[51]:
```
image_clean.drop_duplicates('jpg_url' , inplace = True)
```

Test

[52]:
```
sum(image_clean.jpg_url.duplicated())
```

[52]:
```
0
```

Define Merge the columns (p1, p1_dog, p2, p2_dog, p3, p3_dog) in one column called: image prediction , and merge the columns(p1_conf, p2_conf, p3_conf) in one column called confidence level.

Code

[53]:
```
dog_type = []
confidence_level = []

def image(image_clean):
    if image_clean['p1_dog'] == True:
        dog_type.append(image_clean['p1'])
        confidence_level.append(image_clean['p1_conf'])
    elif image_clean['p2_dog'] == True:
        dog_type.append(image_clean['p2'])
        confidence_level.append(image_clean['p2_conf'])
    elif image_clean['p3_dog'] == True:
```

```
        dog_type.append(image_clean['p3'])
        confidence_level.append(image_clean['p3_conf'])
    else:
        dog_type.append('Error')
        confidence_level.append('Error')

image_clean.apply(image, axis=1)

#create new columns
image_clean['dog_type'] = dog_type
image_clean['confidence_level'] = confidence_level
```

[54]:
```
image_clean = image_clean[image_clean['dog_type'] != 'Error']
```

[55]:
```
image_clean = image_clean.drop(['p1', 'p1_conf', 'p1_dog', 'p2', 'p2_conf',␣
↪'p2_dog', 'p3', 'p3_conf', 'p3_dog'], axis = 1)
```

Test

[56]:
```
## Test the execution
image_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1691 entries, 0 to 2073
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   tweet_id          1691 non-null   int64
 1   jpg_url           1691 non-null   object
 2   img_num           1691 non-null   int64
 3   dog_type          1691 non-null   object
 4   confidence_level  1691 non-null   object
dtypes: int64(2), object(3)
memory usage: 79.3+ KB
```

[ ]:
```

```

2.4 Merge Dataset

Define Merge all three dataset and remove repetitive columns.

Take both the twitter_clean and Json_clean tables and image_clean merge into one table using the join() method on the columns tweet_id.

Code

[66]:
```
## Merge file twitter and Json file with ID=Tweet_id
File_merge = twitter_clean.join(Json_clean.set_index('tweet_id'), on='tweet_id')
```

[67]:
```
Marge = File_merge.join(image_clean.set_index('tweet_id'), on='tweet_id')
```

Test

[68]:
```
Marge.head()
```

[68]:

	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	…	img_num
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r…	This is Phineas. He's a mystical boy. Only eve…	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892420643…	…	1.0
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r…	This is Tilly. She's just checking pup on you…	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892177421…	…	1.0
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	<a href="http://twitter.com/download/iphone" r…	This is Archie. He is a rare Norwegian Pouncin…	NaN	NaN	NaN			
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	<a href="http://twitter.com/download/iphone" r…	This is Darla. She commenced a snooze mid meal…	NaN	NaN	NaN			
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	<a href="http://twitter.com/download/iphone" r…	This is Franklin. He would like you to stop ca…	NaN	NaN	NaN			

```
2  https://twitter.com/dog_rates/status/891815181…  …      1.0
3  https://twitter.com/dog_rates/status/891689557…  …      1.0
4  https://twitter.com/dog_rates/status/891327558…  …      2.0

            p1   p1_conf p1_dog                  p2   p2_conf p2_dog  \
0       orange  0.097049  False               bagel  0.085851  False
1    Chihuahua  0.323581   True            Pekinese  0.090647   True
2    Chihuahua  0.716012   True            malamute  0.078253   True
3  paper_towel  0.170278  False  Labrador_retriever  0.168086   True
4       basset  0.555712   True    English_springer  0.225770   True

                           p3   p3_conf p3_dog
0                      banana  0.076110  False
1                    papillon  0.068957   True
2                      kelpie  0.031379   True
3                     spatula  0.040836  False
4  German_short-haired_pointer  0.175219   True

[5 rows x 30 columns]
```

[121]:
```
## Merage All Dataset into one file
Marge.to_csv('twitter_archive_master-2.csv', index=False, encoding = 'utf-8')
```

[122]:
```
## Take a copy form the Marge file

File_copy = Marge.copy()
```

[123]:
```
File_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 30 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   tweet_id                    2356 non-null   int64
 1   in_reply_to_status_id       78 non-null     float64
 2   in_reply_to_user_id         78 non-null     float64
 3   timestamp                   2356 non-null   object
 4   source                      2356 non-null   object
 5   text                        2356 non-null   object
 6   retweeted_status_id         181 non-null    float64
 7   retweeted_status_user_id    181 non-null    float64
 8   retweeted_status_timestamp  181 non-null    object
 9   expanded_urls               2297 non-null   object
 10  rating_numerator            2356 non-null   int64
 11  rating_denominator          2356 non-null   int64
 12  name                        2356 non-null   object
```

```
 13  doggo                       2356 non-null   object
 14  floofer                     2356 non-null   object
 15  pupper                      2356 non-null   object
 16  puppo                       2356 non-null   object
 17  favorites                   2354 non-null   float64
 18  retweets                    2354 non-null   float64
 19  jpg_url                     2075 non-null   object
 20  img_num                     2075 non-null   float64
 21  p1                          2075 non-null   object
 22  p1_conf                     2075 non-null   float64
 23  p1_dog                      2075 non-null   object
 24  p2                          2075 non-null   object
 25  p2_conf                     2075 non-null   float64
 26  p2_dog                      2075 non-null   object
 27  p3                          2075 non-null   object
 28  p3_conf                     2075 non-null   float64
 29  p3_dog                      2075 non-null   object
dtypes: float64(10), int64(3), object(17)
memory usage: 552.3+ KB
```

[124]:
```
File_copy.describe()
```

[124]:

	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	favorites	retweets	img_num
count	2356.000000	2354.000000	2354.000000	2075.000000
mean	10.455433	8080.968564	3164.797366	1.203855
std	6.745237	11814.771334	5284.770364	0.561875
min	0.000000	0.000000	0.000000	1.000000

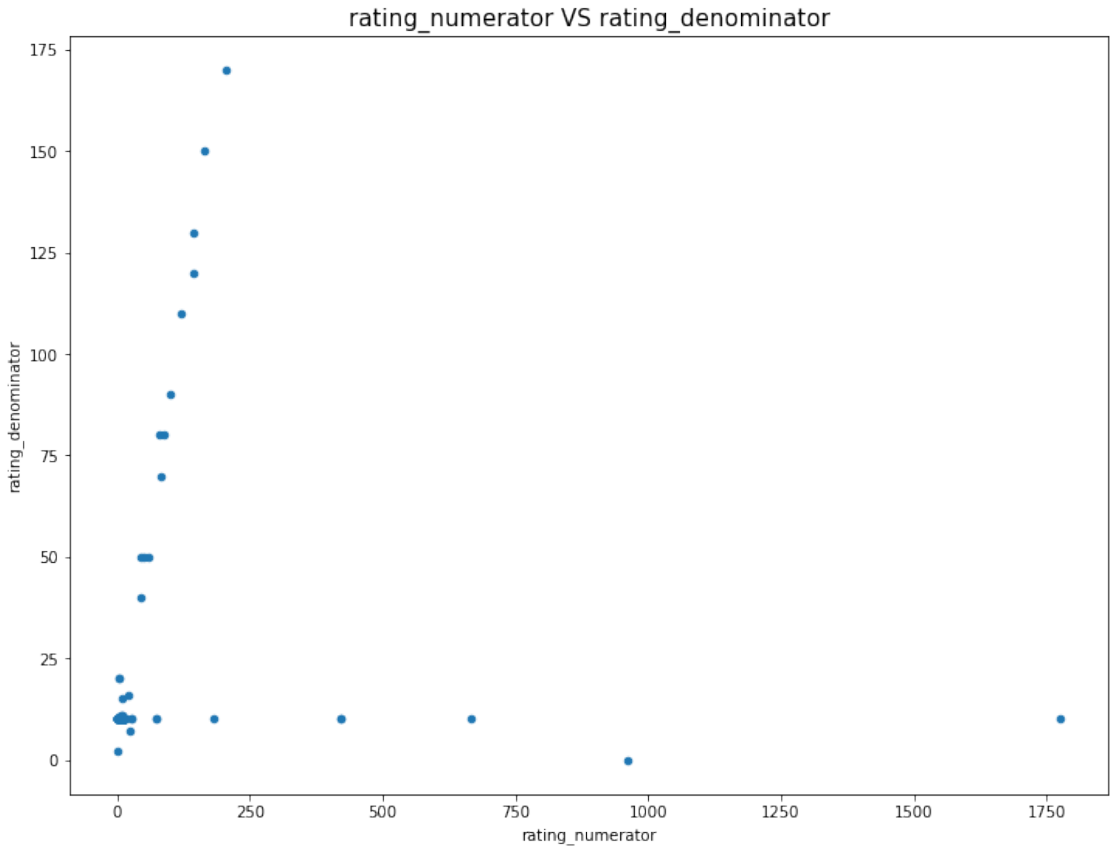
```
25%            10.000000    1415.000000     624.500000      1.000000
50%            10.000000    3603.500000    1473.500000      1.000000
75%            10.000000   10122.250000    3652.000000      1.000000
max           170.000000  132810.000000   79515.000000      4.000000
```

```
          p1_conf       p2_conf       p3_conf
count  2075.000000  2.075000e+03  2.075000e+03
mean      0.594548  1.345886e-01  6.032417e-02
std       0.271174  1.006657e-01  5.090593e-02
min       0.044333  1.011300e-08  1.740170e-10
25%       0.364412  5.388625e-02  1.622240e-02
50%       0.588230  1.181810e-01  4.944380e-02
75%       0.843855  1.955655e-01  9.180755e-02
max       1.000000  4.880140e-01  2.734190e-01
```

2.5 Visualizing Data

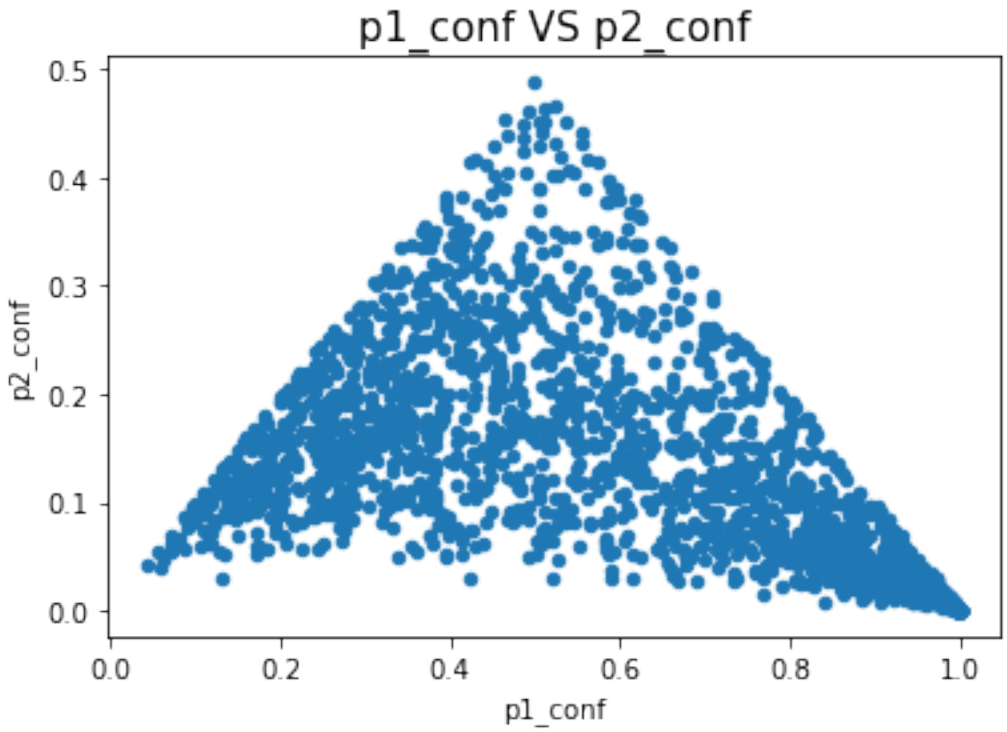
[113]:
```
twitter.plot(x='rating_numerator', y='rating_denominator', kind='scatter').
↪set_title("rating_numerator VS rating_denominator",size=15)
```

[113]:
```
Text(0.5, 1.0, 'rating_numerator VS rating_denominator')
```



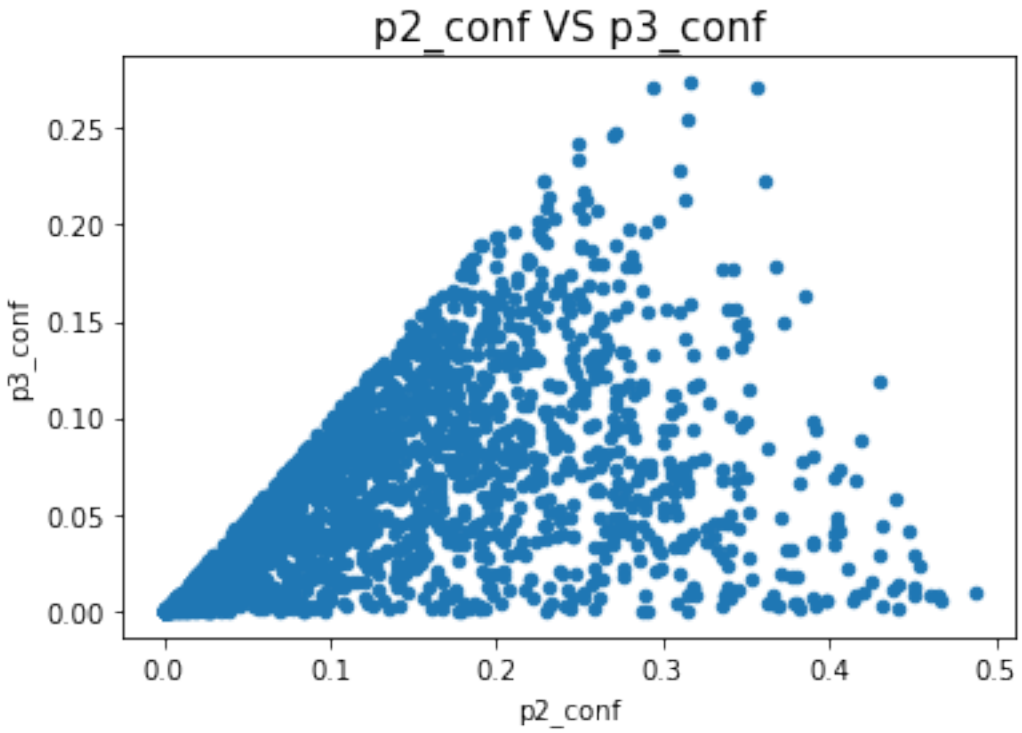
[72]:
```
image_predictions.plot(x='p1_conf', y='p2_conf', kind='scatter').
↪set_title("p1_conf VS p2_conf",size=15)
```

[72]:
```
Text(0.5, 1.0, 'p1_conf VS p2_conf')
```



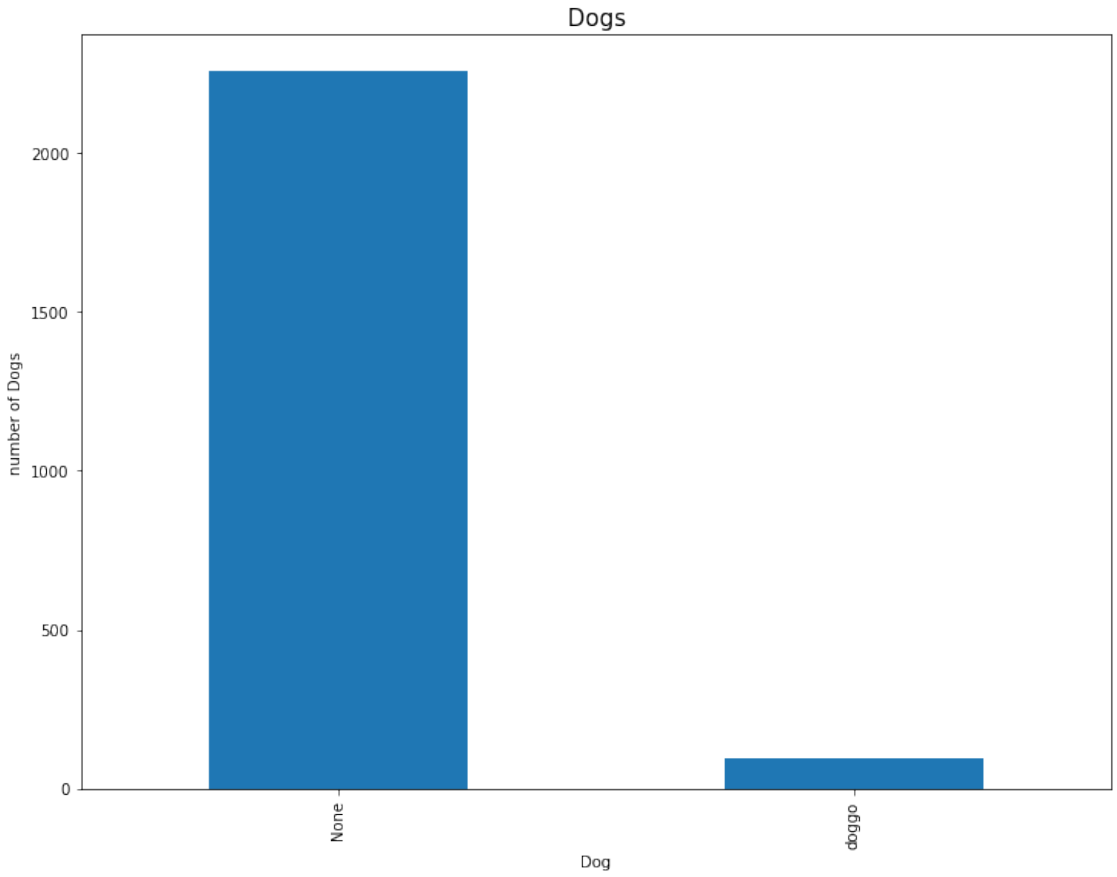
[73]:
```
image_predictions.plot(x='p2_conf', y='p3_conf', kind='scatter').
↪set_title("p2_conf VS p3_conf",size=15)
```

[73]:
```
Text(0.5, 1.0, 'p2_conf VS p3_conf')
```



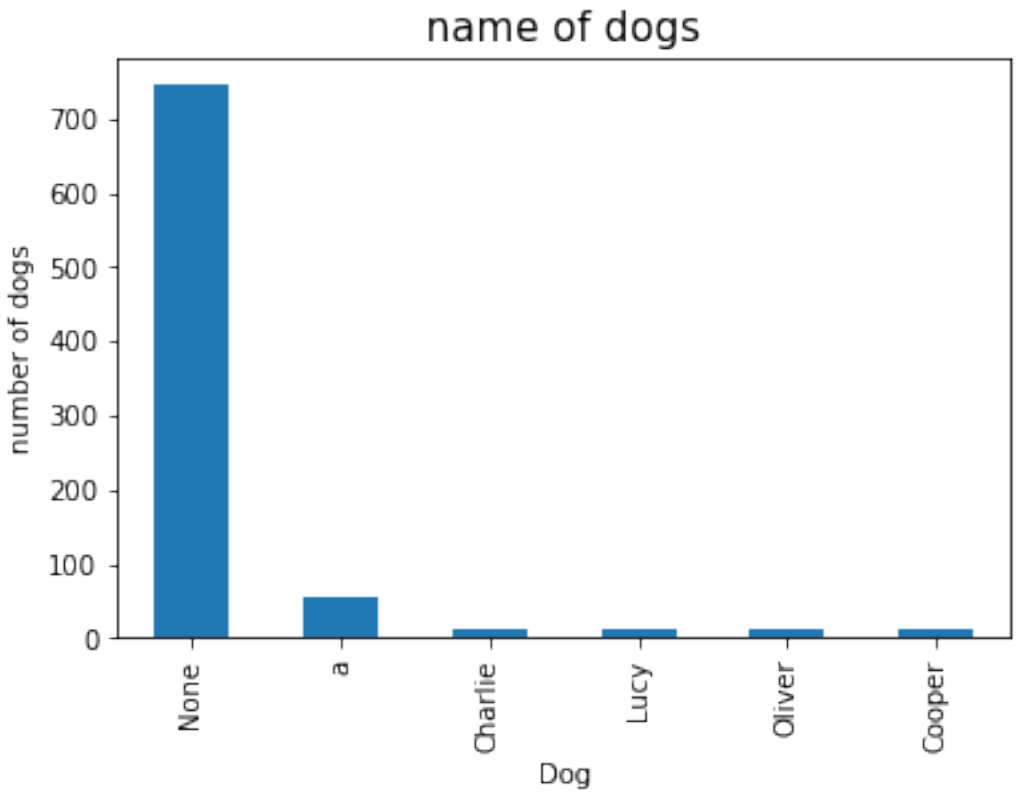
[114]:
```
twitter['doggo'].value_counts()[0:6].sort_values(ascending=False).plot(kind
 ↪='bar')
plt.xlabel('Dog')
plt.ylabel('number of Dogs')
plt.title(' Dogs ', size=15)
```

[114]:
```
Text(0.5, 1.0, ' Dogs ')
```



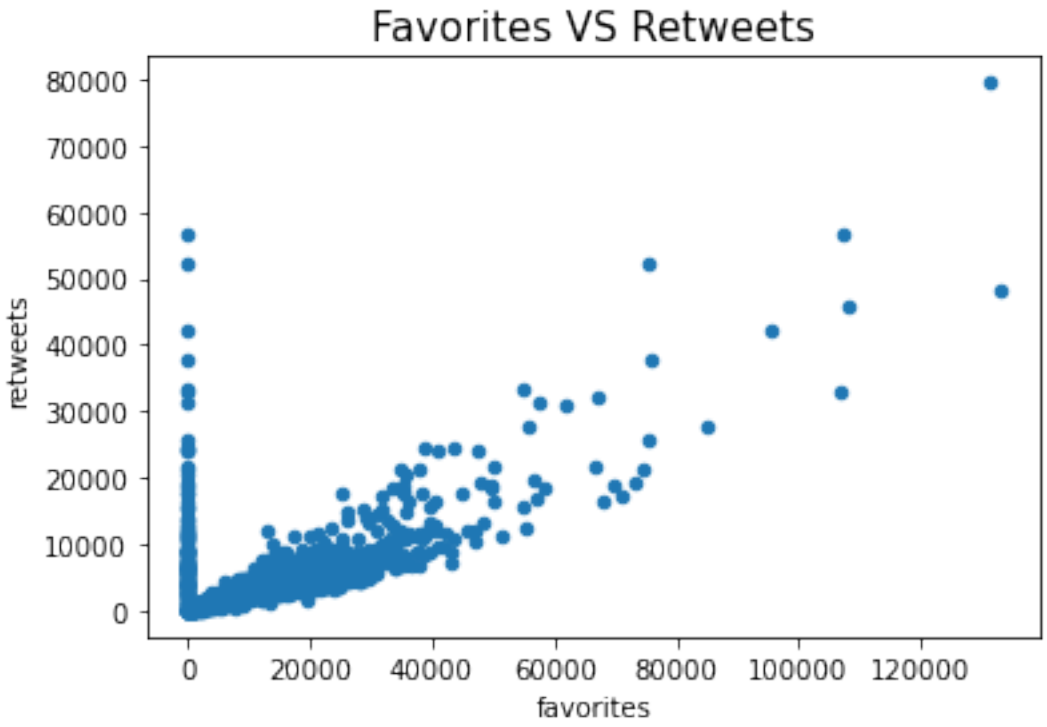
[75]:
```
twitter['name'].value_counts()[0:6].sort_values(ascending=False).plot(kind
↪='bar')
plt.xlabel('Dog')
plt.ylabel('number of dogs')
plt.title(' name of dogs ', size=15)
```

[75]: Text(0.5, 1.0, ' name of dogs ')



[90]:
```
File_copy.plot(x='favorites', y='retweets', kind='scatter').
↪set_title("Favorites VS Retweets",size=15)
```

[90]:
```
Text(0.5, 1.0, 'Favorites VS Retweets')
```



[ ]:

2.6 Summary

Download the data to be analyzed The files that must be viewed and made commensurate with each other, because they serve one account, and they are three files that were previously explained, namely (Twitter archive file: download this file manually by clicking the following link: twitter_archive_enhanced.csv The tweet image predictions, i.e.This file (image_predictions.tsv) is hosted on Udacity's servers and should be downloaded programmatically using the Requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv Twitter JSON) View the data and know its characteristics through the use of many Python programs in pandas Find and delete duplicate data Query for data that does not contain value Delete data that does not contain a value Describe data that contains individual values and make use of them Know the quality of the image data used in the image data Delete the columns that do not lead to a cognitive value in analyzing the data or that are not useful in the analysis. Rather, it is considered an obstacle in the analysis. If there are many columns, we must focus on the columns that lead to a result in their values and be used.

[ ]: