wrangle_act

June 25, 2022

1 Project: Wrangling and Analyze Data

1.0.1 Importing Useful Libraries

```
In [1]: import pandas as pd
        import numpy as np
        import requests
        import tweepy
        import json
        import time
        from timeit import default_timer as timer
```

1.1 Data Gathering

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

1.1.1 Data Information

- data_1 = twitter_archive_enhanced.csv (which is already available for download)
- data_2 = image_predictions.tsv (queried from the udacity server)
- data_3 = twitter_data.json (scraped from twitter with twitter Api)

```
In [2]: # Reading the downloaded data to the notebook
        data_1 = pd.read_csv('twitter-archive-enhanced.csv')
```

```
In [ ]:
```

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

```
In [18]: # downloading the data_2 programmatically from udacity server
         url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predicti
         response = requests.get(url)
         # write the content of the response to an html file
         file_name = 'image-predictions.tsv'
         with open(file_name, mode='wb') as file:
             file.write(response.content)
```

```
In [3]: # Reading the downloaded tsv file
        data_2 = pd.read_csv('image-predictions.tsv', sep='\t')

In [ ]:
```

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

```
In [3]: # importing my keys
        import config

In [3]: tweepy.__version__

Out[3]: '4.6.0'

In [6]: # upgrading tweepy library to 4.6.0 to be able call Api v2 functions
        !pip install tweepy==4.6.0

Collecting tweepy==4.6.0
  Downloading https://files.pythonhosted.org/packages/02/cf/fab85d975d5da397bae3b855d9bccde712a8
    100% || 71kB 5.0MB/s ta 0:00:011
Collecting requests<3,>=2.27.0 (from tweepy==4.6.0)
  Downloading https://files.pythonhosted.org/packages/2d/61/08076519c80041bc0ffa1a8af0cbd3bf3e2b
    100% || 71kB 14.0MB/s ta 0:00:01
Collecting requests-oauthlib<2,>=1.2.0 (from tweepy==4.6.0)
  Downloading https://files.pythonhosted.org/packages/6f/bb/5deac77a9af870143c684ab46a7934038a53
Collecting oauthlib<4,>=3.2.0 (from tweepy==4.6.0)
  Downloading https://files.pythonhosted.org/packages/1d/46/5ee2475e1b46a26ca0fa10d3c1d479577fde
    100% || 153kB 11.8MB/s ta 0:00:01
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/lib/python3.6/site-packages (
Requirement already satisfied: idna<4,>=2.5; python_version >= "3" in /opt/conda/lib/python3.6/s
Collecting charset-normalizer~=2.0.0; python_version >= "3" (from requests<3,>=2.27.0->tweepy==4
  Downloading https://files.pythonhosted.org/packages/06/b3/24afc8868eba069a7f03650ac750a778862d
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.6/site-packages (fro
Installing collected packages: charset-normalizer, requests, oauthlib, requests-oauthlib, tweepy
  Found existing installation: requests 2.18.4
    Uninstalling requests-2.18.4:
      Successfully uninstalled requests-2.18.4
  Found existing installation: oauthlib 2.0.6
    Uninstalling oauthlib-2.0.6:
      Successfully uninstalled oauthlib-2.0.6
  Found existing installation: requests-oauthlib 0.8.0
    Uninstalling requests-oauthlib-0.8.0:
      Successfully uninstalled requests-oauthlib-0.8.0
  Found existing installation: tweepy 3.5.0
    Uninstalling tweepy-3.5.0:
      Successfully uninstalled tweepy-3.5.0
Successfully installed charset-normalizer-2.0.12 oauthlib-3.2.0 requests-2.27.1 requests-oauthli
```

```
In [4]: # authentication for API v1
        auth = tweepy.OAuthHandler(config.api_key, config.api_key_secret)
        auth.set_access_token(config.access_token, config.access_token_secret)
        api = tweepy.API(auth, wait_on_rate_limit=True)

In [5]: # Creating the id list to be used in the api query
        tweet_ids = data_1.tweet_id.values

In [6]: dog_tweets = []

In [7]: # parsing the tweet_ids to get data data for each id in the list
        start = timer()
        for ids in tweet_ids:

            try:
                tweet = api.get_status(ids, tweet_mode='extended')
                tweet_dict = {
                    "tweet_id":tweet.id,
                    "likes":tweet.favorite_count,
                    "retweet":tweet.retweet_count,
                    "timestamp":str(tweet.created_at)
                }
                dog_tweets.append(tweet_dict)
                print('Sucess')
            except Exception as e:
                print("Fail")
                pass
        end = timer()
        print(end - start)
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess

```
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Fail
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
```

Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Fail
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
```

```
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
Sucess
2085.703390442
```

```
In [9]: # saving json file to txt file with json dump
        file_name = "tweet_json.txt"
        with open(file_name, "w") as file:
            json.dump(dog_tweets, file)
```

```
In [4]: # loading the json text file to a variable in the notebook
        with open('tweet_json.txt') as f:
            tweet_data = json.load(f)
```

```
In [5]: # Converting the json text file to pandas dataframe
        data_3 = pd.DataFrame(tweet_data)
```

```
In [15]: data_3.to_csv('tweet_data.csv')
```

1.2 Assessing Data

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

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

- You only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets.
- Assessing and cleaning the entire dataset completely would require a lot of time, and is not necessary to practice and demonstrate your skills in data wrangling. Therefore, the requirements of this project are only to assess and clean at least 8 quality issues and at least 2 tidiness issues in this dataset.
- The fact that the rating numerators are greater than the denominators does not need to be cleaned. This unique rating system is a big part of the popularity of WeRateDogs.
- You do not need to gather the tweets beyond August 1st, 2017. You can, but note that you won't be able to gather the image predictions for these tweets since you don't have access to the algorithm used.

```
In [16]: # Checking the top five rows of data 1
         data_1.head()
```

Out[16]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id
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
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
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
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
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

```
   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 [14]:
```
# Assessing data programmatically using .info
data_1.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 [27]:
```
# Checking the null values in data_1
```

```
data_1.isnull().sum()
```

```
Out[27]: 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
```

```
In [25]: data_1.notnull().sum()
```

```
Out[25]: tweet_id                       2356
         in_reply_to_status_id            78
         in_reply_to_user_id              78
         timestamp                      2356
         source                         2356
         text                           2356
         retweeted_status_id             181
         retweeted_status_user_id        181
         retweeted_status_timestamp      181
         expanded_urls                  2297
         rating_numerator               2356
         rating_denominator             2356
         name                           2356
         doggo                          2356
         floofer                        2356
         pupper                         2356
         puppo                          2356
         dtype: int64
```

```
In [29]: data_1.dtypes
```

```
Out[29]: 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 [15]:
```
# Visual assessment for data 2
data_2.head()
```

Out[15]:

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True	collie	0.156665	True	Shetland_sheepdog	0.061428	True
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826	True	miniature_pinscher	0.074192	True	Rhodesian_ridgeback	0.072010	True
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.596461	True	malinois	0.138584	True	bloodhound	0.116197	True
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.408143	True	redbone	0.360687	True	miniature_pinscher	0.222752	True
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.560311	True	Rottweiler	0.243682	True	Doberman	0.154629	True

In [16]:
```
# Programmatic assessment for data 2
data_2.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 [17]: # Visual assessment for data 3
         data_3.head()
```

Out[17]:

	likes	retweet	timestamp	tweet_id
0	33820	7009	2017-08-01 16:23:56	892420643555336193
1	29337	5302	2017-08-01 00:17:27	892177421306343426
2	22061	3481	2017-07-31 00:18:03	891815181378084864
3	36947	7227	2017-07-30 15:58:51	891689557279858688
4	35315	7763	2017-07-29 16:00:24	891327558926688256

```
In [18]: # programmatic assessment for data 3
         data_3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2327 entries, 0 to 2326
Data columns (total 4 columns):
likes        2327 non-null int64
retweet      2327 non-null int64
timestamp    2327 non-null object
tweet_id     2327 non-null int64
dtypes: int64(3), object(1)
memory usage: 72.8+ KB
```

1.2.1 Quality issues

1. The timestamp values needs to be in standard date format(+0000 should be removed)
2. Data type issue for the timestamp, should be in date datatype
3. The likes column name in data_3 should be renamed to no_of_likes
4. The retweet column name should be retweet_count instead of retweet in data_3 to be more descriptive
5. Names in column name in data_1 are in lower,sentence and upper case(all should be in sentence case)
6. Too much NaNs drop all columns with too many NaNs(retweeted_status_user_id, in_reply_to_user_id, retweeted_status_user_id should be dropped they have very few values in the entire dataset)

7. Some dog type name is in lower case,sentence case and upper case in data_2 p1 column
8. Some dog type name is in lower case,sentence case and upper case in data_2 p2 column
9. Some dog type name is in lower case,sentence case and upper case in data_2 p3 column (should be consistent through out)

1.2.2 Tidiness issues

1. Create a new column and extract the tweet source from the html tag to the new column and drop the old source column
2. Data_3 needs restructuring (tweet_id should be the first column follow by the tweet text)
3. Data_1 and data_3 should be one data table (add the retweet_count and no_of_likes to data_1)
4. Merge data_2 to the new data

1.3 Cleaning Data

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

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

```
In [6]: # Make copies of original pieces of data
        data_1_copy = data_1.copy()
        data_2_copy = data_2.copy()
        data_3_copy = data_3.copy()
```

1.3.1 Issue #1: Tidiness Issue one

Create a new column and extract the tweet source from the html tag to the new column and drop the old source column

Define:Extract the tweet source values from the html tag to a new dataframe column called new_source and drop the source column using str.extract function and some regular expressions to criteria for extract

Code

```
In [7]: # Using the str.extract with re to extract any word characters,spaces and other characte
        # between character > and <
        data_1_copy['new_source'] = data_1['source'].str.extract('(?:.*>)([a-zA-Z-\s]+)(?:.*<)')
        data_1_copy.drop('source', axis=1, inplace=True)
```

Test

```
In [8]: data_1_copy.head()
```

```
Out[8]:              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

                                                    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          new_source
        0                  10   Phineas  None    None   None  None  Twitter for iPhone
        1                  10     Tilly  None    None   None  None  Twitter for iPhone
        2                  10    Archie  None    None   None  None  Twitter for iPhone
        3                  10     Darla  None    None   None  None  Twitter for iPhone
        4                  10  Franklin  None    None   None  None  Twitter for iPhone
```

1.3.2 Issue #2: Quality Issue 1

The timestamp values needs to be in standard date format(+0000 should be removed)

Define: Remove the +0000 in the timestamp values using string slicing method

Code

```
In [9]: data_1_copy['timestamp'] = data_1['timestamp'].str[0:-5]
```

Test

```
In [10]: data_1_copy['timestamp'].head()
```

```
Out[10]: 0    2017-08-01 16:23:56
         1    2017-08-01 00:17:27
         2    2017-07-31 00:18:03
         3    2017-07-30 15:58:51
         4    2017-07-29 16:00:24
         Name: timestamp, dtype: object
```

1.3.3 Issue #3: Quality Issue 2

Data type issue for the timestamp, should be in date datatype

Define: Change the datatype of timestamp to date time

Code

```
In [11]: data_1_copy['timestamp'] = pd.to_datetime(data_1['timestamp'])
```

Test

```
In [12]: data_1_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                    2356 non-null int64
in_reply_to_status_id       78 non-null float64
in_reply_to_user_id         78 non-null float64
timestamp                   2356 non-null datetime64[ns]
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
new_source                     2356 non-null object
dtypes: datetime64[ns](1), float64(4), int64(3), object(9)
memory usage: 313.0+ KB
```

1.3.4 Issue #4 and #5:Quality Issue 3 and 4

The likes and retweet column name in data_3 should be renamed to no_of_likes and retweet_count to be more descriptive

1.3.5 Define: Rename the column name for likes in data_3 to no_of_likes and retweet to retweet_count using .rename function

1.3.6 Code

```
In [13]: data_3_copy.rename({'likes':'no_of_likes', 'retweet':'retweet_count'}, axis=1, inplace=
```

1.3.7 Test

```
In [14]: data_3_copy.head()
```

```
Out[14]:    no_of_likes  retweet_count            timestamp            tweet_id
         0        33820           7009  2017-08-01 16:23:56  892420643555336193
         1        29337           5302  2017-08-01 00:17:27  892177421306343426
         2        22061           3481  2017-07-31 00:18:03  891815181378084864
         3        36947           7227  2017-07-30 15:58:51  891689557279858688
         4        35315           7763  2017-07-29 16:00:24  891327558926688256
```

1.3.8 Issue #6:Quality Issue 5

Names in column name in data_1 are in lower,sentence and upper case(all should be in sentence case)

1.3.9 Define :

Change names in column name in data_1 are in sentence case using the .tittle function

1.3.10 Code

```
In [15]: data_1_copy['name'] = data_1_copy['name'].str.title()
```

1.3.11 Test

```
In [16]: data_1_copy['name'].head()
```

```
Out[16]: 0     Phineas
         1       Tilly
         2      Archie
         3       Darla
```

```
4    Franklin
Name: name, dtype: object
```

1.3.12 Issue #7:Quality Issue 6

Too much NaNs drop all columns with too many NaNs(retweeted_status_user_id, in_reply_to_user_id, retweeted_status_user_id should be dropped they have very few values in the entire dataset)

1.3.13 Define:

Drop the columns with too many NaNs (retweeted_status_user_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_timestamp, in_reply_status_id) in data_1 using pandas drop function

1.3.14 Code

```
In [17]: data_1_copy.drop(columns = ['retweeted_status_user_id', 'in_reply_to_user_id', 'retweet
```

1.3.15 Test

```
In [18]: data_1_copy.head()
```

Out[18]:

	tweet_id	timestamp	text	expanded_urls	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo	new_source
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	https://twitter.com/dog_rates/status/892420643...	13	10	Phineas	None	None	None	None	Twitter for iPhone
1	892177421306343426	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you...	https://twitter.com/dog_rates/status/892177421...	13	10	Tilly	None	None	None	None	Twitter for iPhone
2	891815181378084864	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	https://twitter.com/dog_rates/status/891815181...	12	10	Archie	None	None	None	None	Twitter for iPhone
3	891689557279858688	2017-07-30 15:58:51	This is Darla. She commenced a snooze mid meal...	https://twitter.com/dog_rates/status/891689557...	13	10	Darla	None	None	None	None	Twitter for iPhone
4	891327558926688256	2017-07-29 16:00:24	This is Franklin. He would like you to stop ca...	https://twitter.com/dog_rates/status/891327558...	12	10	Franklin	None	None	None	None	Twitter for iPhone

```
3                 10     Darla  None    None   None  None  Twitter for iPhone
4                 10  Franklin  None    None   None  None  Twitter for iPhone
```

1.3.16 Issue #8, #9 and #10:Quality Issue 7, 8, 9

Some dog type name is in lower case,sentence case and upper case in data_2 p1,p2 and p3 column

1.3.17 Define:

Change all the dog type names case in data_2 to title case for all the predictions columns

1.3.18 Code

```
In [19]: data_2_copy['p1'] = data_2_copy['p1'].str.title()
         data_2_copy['p2'] = data_2_copy['p2'].str.title()
         data_2_copy['p3'] = data_2_copy['p3'].str.title()
```

1.3.19 Test

```
In [20]: data_2_copy.head()
```

```
Out[20]:              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
```

1.3.20 Issue #11:Tidiness Issue 2

Data_3 needs restructuring (tweet_id should be the first column follow by the tweet text)

1.3.21 Define:

Rearrange the columns indexes such that tweet_id will be in he first column using iloc

1.3.22 Code:

```
In [21]: data_3_copy = data_3_copy.iloc[:, [3, 2, 1, 0]]
```

1.3.23 Test:

```
In [22]: data_3_copy.head()
```

```
Out[22]:              tweet_id            timestamp  retweet_count  no_of_likes
         0  892420643555336193  2017-08-01 16:23:56           7009        33820
         1  892177421306343426  2017-08-01 00:17:27           5302        29337
         2  891815181378084864  2017-07-31 00:18:03           3481        22061
         3  891689557279858688  2017-07-30 15:58:51           7227        36947
         4  891327558926688256  2017-07-29 16:00:24           7763        35315
```

1.3.24 Issue #12:Tidiness Issue 3

Data_1 and data_2 should be one data table and merge with data_2 useful criterials (add the retweet_count and no_of_likes to data_1)

1.3.25 Define:

join the no_of_likes and retweet_count columns in data_2_copy to data_1_copy to make new cleaned data using pd.merge function

1.3.26 Code

```
In [23]: metrics = data_3_copy[['tweet_id', 'no_of_likes', 'retweet_count']]
         cleaned_archive_tweets = pd.merge(data_1_copy, metrics, on = 'tweet_id',  how='inner')
```

data_1 and data_2 copy was merged in the above code using inner join to avoid creating new nans and dirty the new cleaned datas because the rows of the two datasets are not equal

1.3.27 Test

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

```
Out[24]:              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

                                                 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  \
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

            new_source  no_of_likes  retweet_count
0  Twitter for iPhone        33820           7009
1  Twitter for iPhone        29337           5302
2  Twitter for iPhone        22061           3481
3  Twitter for iPhone        36947           7227
4  Twitter for iPhone        35315           7763
```

```
In [25]: cleaned_archive_tweets.shape
```

```
Out[25]: (2327, 14)
```

1.3.28 Issue #13:Tidiness Issue 3

subset the data_2_copy with only the first predictions, image url and tweet_id and merge it to the new cleaned archive data

1.3.29 Define:

Join column p1,tweet_id and img_url in table 2 to the new cleaned archive dataset using the inner join

1.3.30 Code

```
In [26]: pred = data_2_copy[['tweet_id', 'jpg_url', 'p1']]
         master_archive_cleaned = pd.merge(cleaned_archive_tweets, pred, on = 'tweet_id', how='i
```

1.3.31 Test

```
In [27]: master_archive_cleaned.head()
```

```
Out[27]:             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

                                                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  \
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

           new_source  no_of_likes  retweet_count  \
0  Twitter for iPhone        33820           7009
1  Twitter for iPhone        29337           5302
2  Twitter for iPhone        22061           3481
3  Twitter for iPhone        36947           7227
4  Twitter for iPhone        35315           7763

                                         jpg_url           p1
0  https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg       Orange
1  https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg    Chihuahua
2  https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg    Chihuahua
3  https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg  Paper_Towel
4  https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg       Basset
```

1.4 Storing Data

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

```
In [28]: master_archive_cleaned.to_csv('twitter_archive_master.csv')
```

```
In [3]: master_archive_cleaned = pd.read_csv('twitter_archive_master.csv')
```

1.5 Analyzing and Visualizing Data

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

1.5.1 Insights:

1. Which dog specie has the highest prediction count in prediction 1
2. Which dog tweet has the highest count of retweet
3. The most frequent source of dog's tweet

1.5.2 First Insight: Which dog specie has the highest prediction count in the first prediction

```
In [31]: master_archive_cleaned['p1'].describe()

Out[31]: count                 2057
         unique                 377
         top       Golden_Retriever
         freq                   150
         Name: p1, dtype: object
```

The result above shows the brief summary of prediction column in the cleaned dataset, which reveals that dog specie with the highest prediction count in prediction 1 is Golden_Retriever with the prediction count of 150

1.5.3 Second Insight: Which dog tweet has the highest count of retweet

```
In [32]: master_archive_cleaned['retweet_count'].max()

Out[32]: 70770

In [35]: max_tweet_retweeted = master_archive_cleaned[master_archive_cleaned['retweet_count'] ==

In [39]: max_tweet_retweeted['text']

Out[39]: 836    Here's a doggo realizing you can stand in a po...
         Name: text, dtype: object
```

The results above reveals the dog tweets with the highest retweet count between 2015 to 2017 tweet datas in the dataset

1.5.4 Third Insight: What is the most common source of dog's tweet between 2015 to 2017

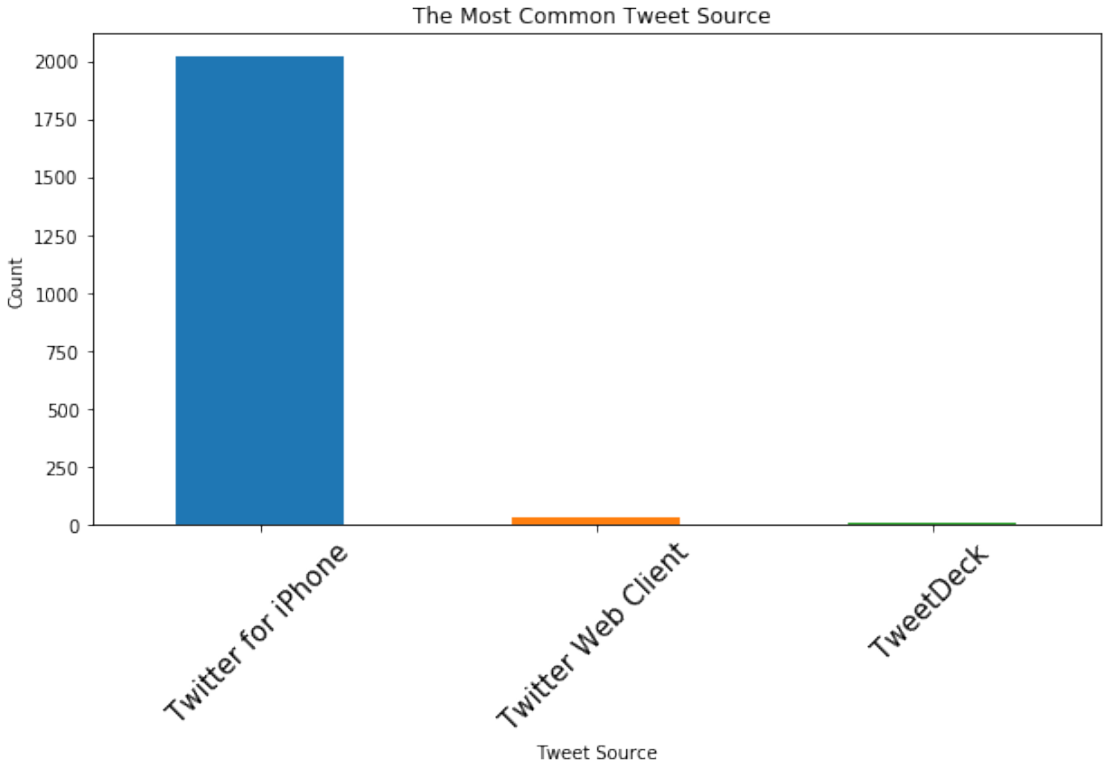
```
In [4]: master_archive_cleaned['new_source'].value_counts()

Out[4]: Twitter for iPhone    2017
        Twitter Web Client      30
        TweetDeck               10
        Name: new_source, dtype: int64
```

1.5.5 Visualization

```
In [2]: import matplotlib.pyplot as plt
```

```
In [5]: plt.figure(figsize=(10, 5))
        master_archive_cleaned['new_source'].value_counts().plot(kind='bar');
        plt.xlabel('Tweet Source')
        plt.ylabel('Count')
        plt.title('The Most Common Tweet Source ')
        plt.xticks(rotation=45, fontsize=15)
```

```
Out[5]: (array([0, 1, 2]), <a list of 3 Text xticklabel objects>)
```



The data visualization above shows the dogs tweets source by count

```
In [ ]:
```