twitter_data

May 6, 2022

[ ]:
```
import pandas as pd
import snscrape.modules.twitter as sntwitter
from datetime import datetime
from datetime import timedelta
import glob
import os
import numpy
import matplotlib.pyplot as plt
import re
import seaborn as sns
plt.rcParams.update({'figure.figsize':(7,5), 'figure.dpi':75})
plt.rcParams["figure.autolayout"] = True

%matplotlib inline
```

1 Data Upload

Formatting companies file

[ ]:
```
company = pd.read_csv('Superbowl_Companies.csv')
company.head(5)
```

	Product Type	Company	2015	2016	2017	2018	2019	2020	2021	2022
0	Beer	Budweiser	Yes	no	yes	yes	yes	no	no	yes
1	Beer	Bud Light	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
2	candy	Mar Inc.	Yes	Yes	Yes	Yes	Yes	No	Yes	No
3	car	Jeep	Yes	Yes	No	Yes	No	Yes	Yes	No
4	car	Toyato	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes

[ ]:
```
company = company.replace(to_replace=['Yes', 'yes'], value=1)
company = company.replace(to_replace=['No', 'no'], value=0)
company = company.replace(to_replace=['No ', 'no'], value=0)
company_dict = company.drop(['Product Type'], axis = 1).set_index(['Company']).
↪T.to_dict('list')
```

[ ]:
```
import numpy as np
list2 = [2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]
dict_keys = []
```

```
new_dict = []

for i, j in company_dict.items():
    j = np.multiply(j, list2).tolist()
    dict_keys.append(j)

    for m in dict_keys:
        c = m.count(0)

        for n in range(c):
            m.remove(0)

    for list in dict_keys:
            j = list

    company_dict[i] = j


print(company_dict)
```

```
{'Budweiser': [2015, 2017, 2018, 2019, 2022], 'Bud Light': [2015, 2016, 2017,
2018, 2019, 2021, 2022], 'Mar Inc.': [2015, 2016, 2017, 2018, 2019, 2021],
'Jeep': [2015, 2016, 2018, 2020, 2021], 'Toyato': [2015, 2016, 2018, 2019, 2020,
2021, 2022], 'Doritos': [2015, 2016, 2018, 2019, 2020, 2022], 'Avocado from
Mexico ': [2015, 2016, 2017, 2018, 2019, 2020, 2022], 'pringles': [2018, 2019],
'Pepsi/ Mountain Dew': [2015, 2016, 2018, 2019, 2020, 2022], 'Coca-Cola': [2015,
2016, 2017, 2018, 2019, 2020], 'T-Mobile': [2015, 2016, 2017, 2018, 2019, 2021,
2022], 'sprint': [2015, 2017, 2018], 'Tide': [2017, 2018, 2020, 2021], 'weather
tech': [2015, 2016, 2018, 2021, 2022], 'turbotax': [2015, 2017, 2018, 2022],
'wix.com': [2015, 2017, 2018], 'squarespace': [2015, 2017, 2018, 2021, 2022],
'amazon alexa': [2018, 2019], 'uber east': [2021, 2022]}
```

1.0.1 Dowloading from Twitter Data

[ ]:
```
superbowl_dates = ['2015-02-01', '2016-02-07', '2017-02-05', '2018-02-04',␣
↪'2019-02-03', '2020-02-02', '2021-02-07', '2022-02-13']
dates_list = [datetime.strptime(date, "%Y-%m-%d").date() for date in␣
↪superbowl_dates]#turning dates to datetime format
before_date = [str(date-timedelta(days=2)) for date in dates_list] #calculating␣
↪dates that is 2 days before superbowl
after_date = [str(date+timedelta(days=2)) for date in dates_list]  #calculating␣
↪dates that is 2 days after superbowl
```

[ ]:
```
dates_list
```

[ ]:
```
[datetime.date(2015, 2, 1),
 datetime.date(2016, 2, 7),
```

```
 datetime.date(2017, 2, 5),
 datetime.date(2018, 2, 4),
 datetime.date(2019, 2, 3),
 datetime.date(2020, 2, 2),
 datetime.date(2021, 2, 7),
 datetime.date(2022, 2, 13)]
```

[ ]:
```
before_date
```

[ ]:
```
['2015-01-30',
 '2016-02-05',
 '2017-02-03',
 '2018-02-02',
 '2019-02-01',
 '2020-01-31',
 '2021-02-05',
 '2022-02-11']
```

[ ]:
```
after_date
```

[ ]:
```
['2015-02-03',
 '2016-02-09',
 '2017-02-07',
 '2018-02-06',
 '2019-02-05',
 '2020-02-04',
 '2021-02-09',
 '2022-02-15']
```

[ ]:
```
def get_brand_tweets_in_year_before(brand, year):
    """
    Function to download data for days before superbowl for each brand during␣
 ↪different years
    Data is downloaded in before folder
    """


    tweet: sntwitter.Tweet
    # Creating list to append tweet data to
    tweets_list2 = []

    # Using TwitterSearchScraper to scrape data and append tweets to list

    #calculating date from
    year_match_since = [s for s in before_date if str(year) in s]
    since = (','.join(year_match_since))
```

```
    #calculating date to
    year_match_after = [s for s in superbowl_dates if str(year) in s]
    until = (','.join(year_match_after))


    for i,tweet in enumerate(sntwitter.TwitterSearchScraper(f'{brand} since:
↪{since} until:{until} lang:en filter:has_engagement').get_items()):
        #filtering for specific time range, tweets are in english and tweets␣
↪have engagements

        if tweet.retweetCount > 0:
            tweets_list2.append([tweet.url, tweet.date, tweet.id, tweet.content.
↪replace('\n', ' '), tweet.retweetCount, tweet.replyCount,
                                 tweet.likeCount, tweet.user.username, tweet.user.
↪displayname, tweet.user.followersCount,
                                 tweet.user.friendsCount, tweet.renderedContent])
        if len(tweets_list2) == 750: #pulling 750 tweets max for each brand for␣
↪each year
            break

    # Creating a dataframe from the tweets list above
    tweets_df2 = pd.DataFrame(tweets_list2, columns=['url','Datetime', 'Tweet␣
↪Id', 'Text', 'Retweet Count', 'Reply Count',
                                                     'Like Count',␣
↪'Username', 'Display Name','Followers Count', 'Friends Count',
                                                     'Rendered Content'])

    tweets_df2['Datetime'] = pd.to_datetime(tweets_df2['Datetime'])
    tweets_df2['Day'] = tweets_df2['Datetime'].dt.day
    tweets_df2['Month'] = tweets_df2['Datetime'].dt.month
    tweets_df2['Year'] = tweets_df2['Datetime'].dt.year


    # Write df to a csv file
    tweets_df2.to_csv(f'data/before/{year}_{brand}.csv', index=False,␣
↪header=True)
```

[ ]:
```
def get_brand_tweets_in_year_after(brand, year):
    """
    Function to download data for days after superbowl for each brand during␣
↪different years
    Data is downloaded in after folder
    """


    tweet: sntwitter.Tweet
```

```
    # Creating list to append tweet data to
    tweets_list2 = []



    year_match_since = [s for s in superbowl_dates if str(year) in s]
    since = (','.join(year_match_since))

    year_match_after = [s for s in after_date if str(year) in s]
    until = (','.join(year_match_after))



    for i,tweet in enumerate(sntwitter.TwitterSearchScraper(f'{brand} since:
↪{since} until:{until} lang:en filter:has_engagement').get_items()):
        if tweet.retweetCount > 0:
            tweets_list2.append([tweet.url, tweet.date, tweet.id, tweet.content.
↪replace('\n', ' '), tweet.retweetCount, tweet.replyCount,
                                 tweet.likeCount, tweet.user.username, tweet.user.
↪displayname, tweet.user.followersCount,
                                 tweet.user.friendsCount, tweet.renderedContent])

        if len(tweets_list2) == 750:
            break

    # Creating a dataframe from the tweets list above
    tweets_df2 = pd.DataFrame(tweets_list2, columns=['url','Datetime', 'Tweet␣
↪Id', 'Text', 'Retweet Count', 'Reply Count',
                                                     'Like Count',␣
↪'Username', 'Display Name','Followers Count', 'Friends Count',
                                                     'Rendered Content'])

    tweets_df2['Datetime'] = pd.to_datetime(tweets_df2['Datetime'])
    tweets_df2['Day'] = tweets_df2['Datetime'].dt.day
    tweets_df2['Month'] = tweets_df2['Datetime'].dt.month
    tweets_df2['Year'] = tweets_df2['Datetime'].dt.year


    # Write df to a csv file
    tweets_df2.to_csv(f'data/after/{year}_{brand}.csv', index=False,␣
↪header=True)
```

[ ]:
```
def get_brand_tweets_in_year_after_sb(brand, year):
    """
    Function to download data for days after superbowl for each brand during␣
↪different years and tweets that also contains "Superbowl" in them
```

```
    Data is downloaded in after/superbowl folder
    """


    tweet: sntwitter.Tweet
    # Creating list to append tweet data to
    tweets_list2 = []

    # Using TwitterSearchScraper to scrape data and append tweets to list

    year_match_since = [s for s in superbowl_dates if str(year) in s]
    since = (','.join(year_match_since))

    year_match_after = [s for s in after_date if str(year) in s]
    until = (','.join(year_match_after))

    for i,tweet in enumerate(sntwitter.TwitterSearchScraper(f'({brand} AND␣
↪Superbowl) since:{since} until:{until} lang:en filter:has_engagement').
↪get_items()):
        if tweet.retweetCount > 0:
            tweets_list2.append([tweet.url, tweet.date, tweet.id, tweet.content.
↪replace('\n', ' '), tweet.retweetCount, tweet.replyCount,
                                 tweet.likeCount, tweet.user.username, tweet.user.
↪displayname, tweet.user.followersCount,
                                 tweet.user.friendsCount, tweet.renderedContent])
        if len(tweets_list2) == 750:
            break

    # Creating a dataframe from the tweets list above
    tweets_df2 = pd.DataFrame(tweets_list2, columns=['url','Datetime', 'Tweet␣
↪Id', 'Text', 'Retweet Count', 'Reply Count',
                                                     'Like Count',␣
↪'Username', 'Display Name','Followers Count', 'Friends Count',
                                                     'Rendered Content'])

    tweets_df2['Datetime'] = pd.to_datetime(tweets_df2['Datetime'])
    tweets_df2['Day'] = tweets_df2['Datetime'].dt.day
    tweets_df2['Month'] = tweets_df2['Datetime'].dt.month
    tweets_df2['Year'] = tweets_df2['Datetime'].dt.year


    # Write df to a csv file
    tweets_df2.to_csv(f'data/after/superbowl/{year}_{brand}.csv', index=False,␣
↪header=True)
```

```
[ ]: #dictionary for each brand and the years when they had a superbowl ad
brands = {
  'Budweiser': [2015, 2017, 2018, 2019, 2022],
  'Budlight': [2015, 2016, 2017, 2018, 2019, 2021, 2022],
  'Mars': [2015, 2016, 2017, 2018, 2019, 2021],
  'Jeep': [2015, 2016, 2018, 2020, 2021],
  'Toyota': [2015, 2016, 2018, 2019, 2020, 2021, 2022],
  'Doritos': [2015, 2016, 2018, 2019, 2020, 2022],
  'Avocado from Mexico': [2015, 2016, 2017, 2018, 2019, 2020, 2022],
  'pringles': [2018, 2019],
  'Pepsi': [2015, 2016, 2018, 2019, 2020, 2022],
  'Coca-Cola': [2015, 2016, 2017, 2018, 2019, 2020],
  'T-Mobile': [2015, 2016, 2017, 2018, 2019, 2021, 2022],
  'sprint': [2015, 2017, 2018],
  'Tide': [2017, 2018, 2020, 2021],
  'weather tech': [2015, 2016, 2018, 2021, 2022],
  'turbotax': [2015, 2017, 2018, 2022],
  'wix': [2015, 2017, 2018],
  'squarespace': [2015, 2017, 2018, 2021, 2022],
  'amazon alexa': [2018, 2019],
  'uber eats': [2021, 2022]
  }
```

```
[ ]: brands= {'wix': [2015, 2017, 2018],
          'Mars': [2015, 2016, 2017, 2018, 2019, 2021],
          'uber eats': [2021, 2022]}
```

```
[ ]: #get data for before superbowl. Date for 2 days before.
for brand, years in brands.items():
    for year in years:
        get_brand_tweets_in_year_before(brand, year)
```

```
[ ]: #get data for after superbowl. Date for 2 days after.
for brand, years in brands.items():
    for year in years:
        get_brand_tweets_in_year_after(brand, year)
```

```
[ ]: #get data for after superbowl and contains "superbowl". Date for 2 days after.
for brand, years in brands.items():
    for year in years:
        get_brand_tweets_in_year_after_sb(brand, year)
```

Data Exploration

```
[ ]: companies = ["T-Mobile",
                 "sprint",
                 "Tide",
```

```
                "weather tech",
                "turbotax",
                "wix",
                "squarespace",
                "amazon alexa",
                "uber eats",
                'Budweiser',
                'Budlight',
                'Mars',
                'Jeep',
                'Toyota',
                'Doritos',
                'Avocado from Mexico',
                'pringles',
                'Pepsi',
                'Coca-Cola']
```

[ ]:
```
print(glob.glob('data/before/*sprint.csv'))#print all sprint files with before␣
 ↪folder
print(glob.glob('data/after/*sprint.csv'))#print all sprint files with after␣
 ↪folder
print(glob.glob('data/after/superbowl/*sprint.csv'))#print all sprint files␣
 ↪with after/superbowl folder
```

```
['data/before/2015_sprint.csv', 'data/before/2018_sprint.csv',
'data/before/2017_sprint.csv']
['data/after/2015_sprint.csv', 'data/after/2018_sprint.csv',
'data/after/2017_sprint.csv']
['data/after/superbowl/2015_sprint.csv', 'data/after/superbowl/2018_sprint.csv',
'data/after/superbowl/2017_sprint.csv']
```

[ ]:
```
#print all files under before folder
print(sorted(glob.glob(f'data/before/*{company}.csv')for company in companies))
```

```
[['data/before/2015_Jeep.csv', 'data/before/2020_Jeep.csv',
'data/before/2021_Jeep.csv', 'data/before/2018_Jeep.csv',
'data/before/2016_Jeep.csv'], ['data/before/2015_Mars.csv',
'data/before/2021_Mars.csv', 'data/before/2018_Mars.csv',
'data/before/2019_Mars.csv', 'data/before/2016_Mars.csv',
'data/before/2017_Mars.csv'], ['data/before/2015_sprint.csv',
'data/before/2018_sprint.csv', 'data/before/2017_sprint.csv'],
['data/before/2016_Pepsi.csv', 'data/before/2022_Pepsi.csv',
'data/before/2019_Pepsi.csv', 'data/before/2020_Pepsi.csv',
'data/before/2015_Pepsi.csv', 'data/before/2018_Pepsi.csv'],
['data/before/2016_T-Mobile.csv', 'data/before/2017_T-Mobile.csv',
'data/before/2021_T-Mobile.csv', 'data/before/2015_T-Mobile.csv',
'data/before/2022_T-Mobile.csv', 'data/before/2019_T-Mobile.csv',
```

```
'data/before/2018_T-Mobile.csv'], ['data/before/2016_weather tech.csv',
'data/before/2022_weather tech.csv', 'data/before/2018_weather tech.csv',
'data/before/2021_weather tech.csv', 'data/before/2015_weather tech.csv'],
['data/before/2017_Avocado from Mexico.csv', 'data/before/2018_Avocado from
Mexico.csv', 'data/before/2022_Avocado from Mexico.csv',
'data/before/2019_Avocado from Mexico.csv', 'data/before/2016_Avocado from
Mexico.csv', 'data/before/2015_Avocado from Mexico.csv',
'data/before/2020_Avocado from Mexico.csv'], ['data/before/2017_Budweiser.csv',
'data/before/2019_Budweiser.csv', 'data/before/2018_Budweiser.csv',
'data/before/2022_Budweiser.csv', 'data/before/2015_Budweiser.csv'],
['data/before/2017_Tide.csv', 'data/before/2021_Tide.csv',
'data/before/2020_Tide.csv', 'data/before/2018_Tide.csv'],
['data/before/2017_squarespace.csv', 'data/before/2018_squarespace.csv',
'data/before/2022_squarespace.csv', 'data/before/2015_squarespace.csv',
'data/before/2021_squarespace.csv'], ['data/before/2017_turbotax.csv',
'data/before/2015_turbotax.csv', 'data/before/2022_turbotax.csv',
'data/before/2018_turbotax.csv'], ['data/before/2018_Budlight.csv',
'data/before/2019_Budlight.csv', 'data/before/2022_Budlight.csv',
'data/before/2015_Budlight.csv', 'data/before/2017_Budlight.csv',
'data/before/2021_Budlight.csv', 'data/before/2016_Budlight.csv'],
['data/before/2018_Coca-Cola.csv', 'data/before/2016_Coca-Cola.csv',
'data/before/2015_Coca-Cola.csv', 'data/before/2017_Coca-Cola.csv',
'data/before/2019_Coca-Cola.csv', 'data/before/2020_Coca-Cola.csv'],
['data/before/2018_Doritos.csv', 'data/before/2020_Doritos.csv',
'data/before/2015_Doritos.csv', 'data/before/2022_Doritos.csv',
'data/before/2019_Doritos.csv', 'data/before/2016_Doritos.csv'],
['data/before/2018_Toyota.csv', 'data/before/2022_Toyota.csv',
'data/before/2015_Toyota.csv', 'data/before/2020_Toyota.csv',
'data/before/2019_Toyota.csv', 'data/before/2016_Toyota.csv',
'data/before/2021_Toyota.csv'], ['data/before/2018_wix.csv',
'data/before/2015_wix.csv', 'data/before/2017_wix.csv'],
['data/before/2019_amazon alexa.csv', 'data/before/2018_amazon alexa.csv'],
['data/before/2019_pringles.csv', 'data/before/2018_pringles.csv'],
['data/before/2022_uber eats.csv', 'data/before/2021_uber eats.csv']]
```

[ ]:
```python
def print_shape(filenames):
    """
    Function to print the shape of all files in each folder
    """
    for company in filenames:
        for filename in company:
            print(filename)
            data = pd.read_csv(filename, lineterminator='\n')
            print(data.shape)

less_than_rows = 100 #number of rows that is considered low. If row is less␣
 ↪than 100 rows them select the file
```

```
def print_low_rows(filenames):
    """
    Function to print files that had low number of rows
    """
    for company in filenames:
        for filename in company:
            data = pd.read_csv(filename, lineterminator='\n')
            if data.shape[0]<less_than_rows:
                print(f'files with low number of rows: "{filename}" and they␣
 ↪have the shape {data.shape}')
```

[ ]:
```
#shape of files within before folder
filenames_before = sorted(glob.glob(f'data/before/*{company}.csv') for company␣
 ↪in companies)
print_shape(filenames_before)
```

```
data/before/2015_Jeep.csv
(750, 15)
data/before/2020_Jeep.csv
(324, 15)
data/before/2021_Jeep.csv
(322, 15)
data/before/2018_Jeep.csv
(349, 15)
data/before/2016_Jeep.csv
(659, 15)
data/before/2015_Mars.csv
(750, 15)
data/before/2021_Mars.csv
(750, 15)
data/before/2018_Mars.csv
(750, 15)
data/before/2019_Mars.csv
(750, 15)
data/before/2016_Mars.csv
(750, 15)
data/before/2017_Mars.csv
(750, 15)
data/before/2015_sprint.csv
(750, 15)
data/before/2018_sprint.csv
(750, 15)
data/before/2017_sprint.csv
(750, 15)
data/before/2016_Pepsi.csv
(518, 15)
data/before/2022_Pepsi.csv
```

```
(454, 15)
data/before/2019_Pepsi.csv
(505, 15)
data/before/2020_Pepsi.csv
(750, 15)
data/before/2015_Pepsi.csv
(750, 15)
data/before/2018_Pepsi.csv
(601, 15)
data/before/2016_T-Mobile.csv
(278, 15)
data/before/2017_T-Mobile.csv
(627, 15)
data/before/2021_T-Mobile.csv
(394, 15)
data/before/2015_T-Mobile.csv
(338, 15)
data/before/2022_T-Mobile.csv
(250, 15)
data/before/2019_T-Mobile.csv
(209, 15)
data/before/2018_T-Mobile.csv
(242, 15)
data/before/2016_weather tech.csv
(30, 15)
data/before/2022_weather tech.csv
(19, 15)
data/before/2018_weather tech.csv
(18, 15)
data/before/2021_weather tech.csv
(11, 15)
data/before/2015_weather tech.csv
(18, 15)
data/before/2017_Avocado from Mexico.csv
(37, 15)
data/before/2018_Avocado from Mexico.csv
(7, 15)
data/before/2022_Avocado from Mexico.csv
(5, 15)
data/before/2019_Avocado from Mexico.csv
(51, 15)
data/before/2016_Avocado from Mexico.csv
(17, 15)
data/before/2015_Avocado from Mexico.csv
(20, 15)
data/before/2020_Avocado from Mexico.csv
(63, 15)
data/before/2017_Budweiser.csv
```

```
(611, 15)
data/before/2019_Budweiser.csv
(166, 15)
data/before/2018_Budweiser.csv
(142, 15)
data/before/2022_Budweiser.csv
(96, 15)
data/before/2015_Budweiser.csv
(750, 15)
data/before/2017_Tide.csv
(717, 15)
data/before/2021_Tide.csv
(750, 15)
data/before/2020_Tide.csv
(750, 15)
data/before/2018_Tide.csv
(750, 15)
data/before/2017_squarespace.csv
(30, 15)
data/before/2018_squarespace.csv
(45, 15)
data/before/2022_squarespace.csv
(40, 15)
data/before/2015_squarespace.csv
(56, 15)
data/before/2021_squarespace.csv
(45, 15)
data/before/2017_turbotax.csv
(205, 15)
data/before/2015_turbotax.csv
(70, 15)
data/before/2022_turbotax.csv
(61, 15)
data/before/2018_turbotax.csv
(43, 15)
data/before/2018_Budlight.csv
(232, 15)
data/before/2019_Budlight.csv
(199, 15)
data/before/2022_Budlight.csv
(247, 15)
data/before/2015_Budlight.csv
(385, 15)
data/before/2017_Budlight.csv
(250, 15)
data/before/2021_Budlight.csv
(128, 15)
data/before/2016_Budlight.csv
```

```
(216, 15)
data/before/2018_Coca-Cola.csv
(433, 15)
data/before/2016_Coca-Cola.csv
(404, 15)
data/before/2015_Coca-Cola.csv
(728, 15)
data/before/2017_Coca-Cola.csv
(232, 15)
data/before/2019_Coca-Cola.csv
(417, 15)
data/before/2020_Coca-Cola.csv
(444, 15)
data/before/2018_Doritos.csv
(152, 15)
data/before/2020_Doritos.csv
(174, 15)
data/before/2015_Doritos.csv
(323, 15)
data/before/2022_Doritos.csv
(151, 15)
data/before/2019_Doritos.csv
(198, 15)
data/before/2016_Doritos.csv
(292, 15)
data/before/2018_Toyota.csv
(543, 15)
data/before/2022_Toyota.csv
(750, 15)
data/before/2015_Toyota.csv
(750, 15)
data/before/2020_Toyota.csv
(566, 15)
data/before/2019_Toyota.csv
(512, 15)
data/before/2016_Toyota.csv
(689, 15)
data/before/2021_Toyota.csv
(571, 15)
data/before/2018_wix.csv
(45, 15)
data/before/2015_wix.csv
(67, 15)
data/before/2017_wix.csv
(48, 15)
data/before/2019_amazon alexa.csv
(161, 15)
data/before/2018_amazon alexa.csv
```

```
(331, 15)
data/before/2019_pringles.csv
(190, 15)
data/before/2018_pringles.csv
(77, 15)
data/before/2022_uber eats.csv
(322, 15)
data/before/2021_uber eats.csv
(509, 15)
```

[ ]:
```
less_than_rows = 100
print_low_rows(filenames_before)
#wix, weather tech, square space, turbotax, uber eat, Avocado from Mexico have␣
 ↪low data
```

```
files with low number of rows: "data/before/2016_weather tech.csv" and they have
the shape (30, 15)
files with low number of rows: "data/before/2022_weather tech.csv" and they have
the shape (19, 15)
files with low number of rows: "data/before/2018_weather tech.csv" and they have
the shape (18, 15)
files with low number of rows: "data/before/2021_weather tech.csv" and they have
the shape (11, 15)
files with low number of rows: "data/before/2015_weather tech.csv" and they have
the shape (18, 15)
files with low number of rows: "data/before/2017_Avocado from Mexico.csv" and
they have the shape (37, 15)
files with low number of rows: "data/before/2018_Avocado from Mexico.csv" and
they have the shape (7, 15)
files with low number of rows: "data/before/2022_Avocado from Mexico.csv" and
they have the shape (5, 15)
files with low number of rows: "data/before/2019_Avocado from Mexico.csv" and
they have the shape (51, 15)
files with low number of rows: "data/before/2016_Avocado from Mexico.csv" and
they have the shape (17, 15)
files with low number of rows: "data/before/2015_Avocado from Mexico.csv" and
they have the shape (20, 15)
files with low number of rows: "data/before/2020_Avocado from Mexico.csv" and
they have the shape (63, 15)
files with low number of rows: "data/before/2022_Budweiser.csv" and they have
the shape (96, 15)
files with low number of rows: "data/before/2017_squarespace.csv" and they have
the shape (30, 15)
files with low number of rows: "data/before/2018_squarespace.csv" and they have
the shape (45, 15)
files with low number of rows: "data/before/2022_squarespace.csv" and they have
the shape (40, 15)
files with low number of rows: "data/before/2015_squarespace.csv" and they have
```

```
the shape (56, 15)
files with low number of rows: "data/before/2021_squarespace.csv" and they have
the shape (45, 15)
files with low number of rows: "data/before/2015_turbotax.csv" and they have the
shape (70, 15)
files with low number of rows: "data/before/2022_turbotax.csv" and they have the
shape (61, 15)
files with low number of rows: "data/before/2018_turbotax.csv" and they have the
shape (43, 15)
files with low number of rows: "data/before/2018_wix.csv" and they have the
shape (45, 15)
files with low number of rows: "data/before/2015_wix.csv" and they have the
shape (67, 15)
files with low number of rows: "data/before/2017_wix.csv" and they have the
shape (48, 15)
files with low number of rows: "data/before/2018_pringles.csv" and they have the
shape (77, 15)
```

[ ]:
```
#printing shape of files in after folder
filenames_after = sorted(glob.glob(f'data/after/*{company}.csv') for company in␣
 ↪companies)
print_shape(filenames_after)
#similar trend
```

```
data/after/2015_Jeep.csv
(750, 15)
data/after/2020_Jeep.csv
(750, 15)
data/after/2021_Jeep.csv
(750, 15)
data/after/2018_Jeep.csv
(750, 15)
data/after/2016_Jeep.csv
(750, 15)
data/after/2015_Mars.csv
(750, 15)
data/after/2021_Mars.csv
(750, 15)
data/after/2018_Mars.csv
(750, 15)
data/after/2019_Mars.csv
(750, 15)
data/after/2016_Mars.csv
(750, 15)
data/after/2017_Mars.csv
(750, 15)
data/after/2015_sprint.csv
(750, 15)
```

```
data/after/2018_sprint.csv
(750, 15)
data/after/2017_sprint.csv
(750, 15)
data/after/2016_Pepsi.csv
(750, 15)
data/after/2022_Pepsi.csv
(750, 15)
data/after/2019_Pepsi.csv
(750, 15)
data/after/2020_Pepsi.csv
(750, 15)
data/after/2015_Pepsi.csv
(750, 15)
data/after/2018_Pepsi.csv
(750, 15)
data/after/2016_T-Mobile.csv
(750, 15)
data/after/2017_T-Mobile.csv
(750, 15)
data/after/2021_T-Mobile.csv
(635, 15)
data/after/2015_T-Mobile.csv
(750, 15)
data/after/2022_T-Mobile.csv
(456, 15)
data/after/2019_T-Mobile.csv
(750, 15)
data/after/2018_T-Mobile.csv
(750, 15)
data/after/2016_weather tech.csv
(136, 15)
data/after/2022_weather tech.csv
(43, 15)
data/after/2018_weather tech.csv
(178, 15)
data/after/2021_weather tech.csv
(76, 15)
data/after/2015_weather tech.csv
(102, 15)
data/after/2017_Avocado from Mexico.csv
(434, 15)
data/after/2018_Avocado from Mexico.csv
(62, 15)
data/after/2022_Avocado from Mexico.csv
(247, 15)
data/after/2019_Avocado from Mexico.csv
(112, 15)
```

```
data/after/2016_Avocado from Mexico.csv
(96, 15)
data/after/2015_Avocado from Mexico.csv
(295, 15)
data/after/2020_Avocado from Mexico.csv
(88, 15)
data/after/2017_Budweiser.csv
(750, 15)
data/after/2019_Budweiser.csv
(653, 15)
data/after/2018_Budweiser.csv
(750, 15)
data/after/2022_Budweiser.csv
(277, 15)
data/after/2015_Budweiser.csv
(750, 15)
data/after/2017_Tide.csv
(750, 15)
data/after/2021_Tide.csv
(750, 15)
data/after/2020_Tide.csv
(750, 15)
data/after/2018_Tide.csv
(750, 15)
data/after/2017_squarespace.csv
(158, 15)
data/after/2018_squarespace.csv
(174, 15)
data/after/2022_squarespace.csv
(144, 15)
data/after/2015_squarespace.csv
(446, 15)
data/after/2021_squarespace.csv
(144, 15)
data/after/2017_turbotax.csv
(220, 15)
data/after/2015_turbotax.csv
(467, 15)
data/after/2022_turbotax.csv
(233, 15)
data/after/2018_turbotax.csv
(169, 15)
data/after/2018_Budlight.csv
(750, 15)
data/after/2019_Budlight.csv
(750, 15)
data/after/2022_Budlight.csv
(565, 15)
```

```
data/after/2015_Budlight.csv
(750, 15)
data/after/2017_Budlight.csv
(643, 15)
data/after/2021_Budlight.csv
(750, 15)
data/after/2016_Budlight.csv
(750, 15)
data/after/2018_Coca-Cola.csv
(750, 15)
data/after/2016_Coca-Cola.csv
(750, 15)
data/after/2015_Coca-Cola.csv
(750, 15)
data/after/2017_Coca-Cola.csv
(750, 15)
data/after/2019_Coca-Cola.csv
(750, 15)
data/after/2020_Coca-Cola.csv
(440, 15)
data/after/2018_Doritos.csv
(750, 15)
data/after/2020_Doritos.csv
(750, 15)
data/after/2015_Doritos.csv
(750, 15)
data/after/2022_Doritos.csv
(673, 15)
data/after/2019_Doritos.csv
(750, 15)
data/after/2016_Doritos.csv
(750, 15)
data/after/2018_Toyota.csv
(750, 15)
data/after/2022_Toyota.csv
(750, 15)
data/after/2015_Toyota.csv
(750, 15)
data/after/2020_Toyota.csv
(608, 15)
data/after/2019_Toyota.csv
(750, 15)
data/after/2016_Toyota.csv
(750, 15)
data/after/2021_Toyota.csv
(750, 15)
data/after/2018_wix.csv
(117, 15)
```

```
data/after/2015_wix.csv
(241, 15)
data/after/2017_wix.csv
(144, 15)
data/after/2019_amazon alexa.csv
(252, 15)
data/after/2018_amazon alexa.csv
(544, 15)
data/after/2019_pringles.csv
(264, 15)
data/after/2018_pringles.csv
(285, 15)
data/after/2022_uber eats.csv
(696, 15)
data/after/2021_uber eats.csv
(750, 15)
```

[ ]:
```
less_than_rows = 100
print_low_rows(filenames_after)
#weather tech, avocado from mexico, have low numbers
#there are less number of files that has less than 100 tweets after superbowl
```

```
files with low number of rows: "data/after/2022_weather tech.csv" and they have
the shape (43, 15)
files with low number of rows: "data/after/2021_weather tech.csv" and they have
the shape (76, 15)
files with low number of rows: "data/after/2018_Avocado from Mexico.csv" and
they have the shape (62, 15)
files with low number of rows: "data/after/2016_Avocado from Mexico.csv" and
they have the shape (96, 15)
files with low number of rows: "data/after/2020_Avocado from Mexico.csv" and
they have the shape (88, 15)
```

[ ]:
```
#after/superbowl folder file sizes
filenames_after_superbowl = sorted(glob.glob(f'data/after/superbowl/*{company}.
 ↪csv') for company in companies)
print_shape(filenames_after_superbowl)
```

```
data/after/superbowl/2015_Jeep.csv
(173, 15)
data/after/superbowl/2020_Jeep.csv
(219, 15)
data/after/superbowl/2021_Jeep.csv
(316, 15)
data/after/superbowl/2018_Jeep.csv
(136, 15)
data/after/superbowl/2016_Jeep.csv
(87, 15)
```

```
data/after/superbowl/2015_Mars.csv
(136, 15)
data/after/superbowl/2021_Mars.csv
(226, 15)
data/after/superbowl/2018_Mars.csv
(233, 15)
data/after/superbowl/2019_Mars.csv
(123, 15)
data/after/superbowl/2016_Mars.csv
(729, 15)
data/after/superbowl/2017_Mars.csv
(115, 15)
data/after/superbowl/2015_sprint.csv
(61, 15)
data/after/superbowl/2018_sprint.csv
(74, 15)
data/after/superbowl/2017_sprint.csv
(76, 15)
data/after/superbowl/2016_Pepsi.csv
(148, 15)
data/after/superbowl/2022_Pepsi.csv
(228, 15)
data/after/superbowl/2019_Pepsi.csv
(340, 15)
data/after/superbowl/2020_Pepsi.csv
(195, 15)
data/after/superbowl/2015_Pepsi.csv
(232, 15)
data/after/superbowl/2018_Pepsi.csv
(254, 15)
data/after/superbowl/2016_T-Mobile.csv
(39, 15)
data/after/superbowl/2017_T-Mobile.csv
(112, 15)
data/after/superbowl/2021_T-Mobile.csv
(26, 15)
data/after/superbowl/2015_T-Mobile.csv
(53, 15)
data/after/superbowl/2022_T-Mobile.csv
(29, 15)
data/after/superbowl/2019_T-Mobile.csv
(36, 15)
data/after/superbowl/2018_T-Mobile.csv
(53, 15)
data/after/superbowl/2016_weather tech.csv
(1, 15)
data/after/superbowl/2022_weather tech.csv
(0, 15)
```

```
data/after/superbowl/2018_weather tech.csv
(1, 15)
data/after/superbowl/2021_weather tech.csv
(0, 15)
data/after/superbowl/2015_weather tech.csv
(2, 15)
data/after/superbowl/2017_Avocado from Mexico.csv
(91, 15)
data/after/superbowl/2018_Avocado from Mexico.csv
(17, 15)
data/after/superbowl/2022_Avocado from Mexico.csv
(27, 15)
data/after/superbowl/2019_Avocado from Mexico.csv
(12, 15)
data/after/superbowl/2016_Avocado from Mexico.csv
(8, 15)
data/after/superbowl/2015_Avocado from Mexico.csv
(24, 15)
data/after/superbowl/2020_Avocado from Mexico.csv
(15, 15)
data/after/superbowl/2017_Budweiser.csv
(678, 15)
data/after/superbowl/2019_Budweiser.csv
(125, 15)
data/after/superbowl/2018_Budweiser.csv
(156, 15)
data/after/superbowl/2022_Budweiser.csv
(52, 15)
data/after/superbowl/2015_Budweiser.csv
(750, 15)
data/after/superbowl/2017_Tide.csv
(169, 15)
data/after/superbowl/2021_Tide.csv
(69, 15)
data/after/superbowl/2020_Tide.csv
(120, 15)
data/after/superbowl/2018_Tide.csv
(750, 15)
data/after/superbowl/2017_squarespace.csv
(31, 15)
data/after/superbowl/2018_squarespace.csv
(44, 15)
data/after/superbowl/2022_squarespace.csv
(42, 15)
data/after/superbowl/2015_squarespace.csv
(61, 15)
data/after/superbowl/2021_squarespace.csv
(27, 15)
```

```
data/after/superbowl/2017_turbotax.csv
(48, 15)
data/after/superbowl/2015_turbotax.csv
(45, 15)
data/after/superbowl/2022_turbotax.csv
(68, 15)
data/after/superbowl/2018_turbotax.csv
(21, 15)
data/after/superbowl/2018_Budlight.csv
(141, 15)
data/after/superbowl/2019_Budlight.csv
(471, 15)
data/after/superbowl/2022_Budlight.csv
(80, 15)
data/after/superbowl/2015_Budlight.csv
(187, 15)
data/after/superbowl/2017_Budlight.csv
(125, 15)
data/after/superbowl/2021_Budlight.csv
(145, 15)
data/after/superbowl/2016_Budlight.csv
(103, 15)
data/after/superbowl/2018_Coca-Cola.csv
(53, 15)
data/after/superbowl/2016_Coca-Cola.csv
(46, 15)
data/after/superbowl/2015_Coca-Cola.csv
(245, 15)
data/after/superbowl/2017_Coca-Cola.csv
(577, 15)
data/after/superbowl/2019_Coca-Cola.csv
(82, 15)
data/after/superbowl/2020_Coca-Cola.csv
(24, 15)
data/after/superbowl/2018_Doritos.csv
(356, 15)
data/after/superbowl/2020_Doritos.csv
(190, 15)
data/after/superbowl/2015_Doritos.csv
(269, 15)
data/after/superbowl/2022_Doritos.csv
(112, 15)
data/after/superbowl/2019_Doritos.csv
(142, 15)
data/after/superbowl/2016_Doritos.csv
(512, 15)
data/after/superbowl/2018_Toyota.csv
(251, 15)
```

```
data/after/superbowl/2022_Toyota.csv
(90, 15)
data/after/superbowl/2015_Toyota.csv
(149, 15)
data/after/superbowl/2020_Toyota.csv
(17, 15)
data/after/superbowl/2019_Toyota.csv
(88, 15)
data/after/superbowl/2016_Toyota.csv
(54, 15)
data/after/superbowl/2021_Toyota.csv
(135, 15)
data/after/superbowl/2018_wix.csv
(17, 15)
data/after/superbowl/2015_wix.csv
(33, 15)
data/after/superbowl/2017_wix.csv
(48, 15)
data/after/superbowl/2019_amazon alexa.csv
(48, 15)
data/after/superbowl/2018_amazon alexa.csv
(111, 15)
data/after/superbowl/2019_pringles.csv
(35, 15)
data/after/superbowl/2018_pringles.csv
(56, 15)
data/after/superbowl/2022_uber eats.csv
(95, 15)
data/after/superbowl/2021_uber eats.csv
(117, 15)
```

[ ]:
```
less_than_rows = 10
print_low_rows(filenames_after_superbowl)
#weather tech have less numer of rows
```

```
files with low number of rows: "data/after/superbowl/2016_weather tech.csv" and
they have the shape (1, 15)
files with low number of rows: "data/after/superbowl/2022_weather tech.csv" and
they have the shape (0, 15)
files with low number of rows: "data/after/superbowl/2018_weather tech.csv" and
they have the shape (1, 15)
files with low number of rows: "data/after/superbowl/2021_weather tech.csv" and
they have the shape (0, 15)
files with low number of rows: "data/after/superbowl/2015_weather tech.csv" and
they have the shape (2, 15)
files with low number of rows: "data/after/superbowl/2016_Avocado from
Mexico.csv" and they have the shape (8, 15)
```

Joining Files

[ ]:
```
def prep_concat(filenames):
    for files in filenames:
        for file in files:
            df = pd.read_csv(file, lineterminator='\n')
            directory = file.partition("/")[2].partition("/")[0]
            company_name = file.partition("_")[2].partition(".")[0]
            df['Company_Name'] = company_name
            df['Directory'] = directory
            df = df.reset_index(drop=True)
            yield df
```

[ ]:
```
before_df = pd.concat((prep_concat(filenames_before))).reset_index(drop=True)
before_df.head()
```

[ ]:
```
                                                 url  \
0  https://twitter.com/UWnt2/status/5616740628090…
1  https://twitter.com/ShowDreamCar/status/561671…
2  https://twitter.com/ShowDreamCar/status/561671…
3  https://twitter.com/_VictoriaManson/status/561…
4  https://twitter.com/Jeepings/status/5616670153…

                   Datetime            Tweet Id  \
0  2015-01-31 23:55:06+00:00  561674062809022465
1  2015-01-31 23:46:28+00:00  561671888288493568
2  2015-01-31 23:43:45+00:00  561671203966840834
3  2015-01-31 23:28:26+00:00  561667350819520512
4  2015-01-31 23:27:06+00:00  561667015312961536

                                                Text  Retweet Count  \
0  Amazing weekend #off-road &amp;amp; #camping at Th…              1
1  List of things better than perfect Jeep weathe…              1
2  #tbtuesday 1 yr ago when I first started worki…              1
3  Okay, that jeep is beautiful! Loving everythin…              2
4  They call my house The Jeep Garage ;) #jeeplif…              3

   Reply Count  Like Count         Username      Display Name  \
0            0           1            UWnt2        YOUNTO.com
1            0           2     ShowDreamCar      Car Pictures
2            0           1     ShowDreamCar      Car Pictures
3            0           0  _VictoriaManson             Vicki
4            0          11         Jeepings  Northern Jeeper

   Followers Count  Friends Count  \
0              433            711
1            12775             19
2            12775             19
```

```
3             4432           4950
4             1619            738

                                    Rendered Content  Day  Month  Year  \
0  Amazing weekend #off-road &amp; #camping at Th…   31      1  2015
1  List of things better than perfect Jeep weathe…   31      1  2015
2  #tbtuesday 1 yr ago when I first started worki…   31      1  2015
3  Okay, that jeep is beautiful! Loving everythin…   31      1  2015
4  They call my house The Jeep Garage ;) #jeeplif…   31      1  2015

  Company_Name Directory
0         Jeep    before
1         Jeep    before
2         Jeep    before
3         Jeep    before
4         Jeep    before
```

[ ]:
```
before_df.shape
```

[ ]:
```
(32429, 17)
```

[ ]:
```
after_df = pd.concat((prep_concat(filenames_after))).reset_index(drop=True)
after_df.head()
```

[ ]:
```
                                                 url  \
0  https://twitter.com/billyg67/status/5623991633…
1  https://twitter.com/thefox1019/status/56239886…
2  https://twitter.com/ChadHaase/status/562397634…
3  https://twitter.com/CiscaPr/status/56239722094…
4  https://twitter.com/FlexinJC/status/5623969134…

                    Datetime            Tweet Id  \
0  2015-02-02 23:56:24+00:00  562399163350925312
1  2015-02-02 23:55:12+00:00  562398865127530498
2  2015-02-02 23:50:19+00:00  562397634976301056
3  2015-02-02 23:48:40+00:00  562397220943589378
4  2015-02-02 23:47:27+00:00  562396913417617408

                                                Text  Retweet Count  \
0  @SolarTJChick I would love for all jeepgirls t…              4
1  We're paying a visit to Bundoora RIGHT NOW! Ge…              1
2  Jeep stuck in mud in France 1944. http://t.co/…             11
3  Good morning sexy people! Have a sexy Tuesday!…              2
4  @YellowJeepJewel Wow. You don't need people li…              1

   Reply Count  Like Count    Username          Display Name  Followers Count  \
0            3           9    billyg67    Porsche & schmoopy             4435
```

```
1            0           2  thefox1019         The Fox 101.9            42613
2            1          14   ChadHaase            Chad Haase             1086
3            1           7     CiscaPr  Francisca Prandayani             6077
4            1           2    FlexinJC          James Miller             1156

   Friends Count                                   Rendered Content  Day  \
0           4885  @SolarTJChick I would love for all jeepgirls t…    2
1            296  We're paying a visit to Bundoora RIGHT NOW! Ge…    2
2            673  Jeep stuck in mud in France 1944. http://t.co/…    2
3           1992  Good morning sexy people! Have a sexy Tuesday!…    2
4           2090  @YellowJeepJewel Wow. You don't need people li…    2

   Month  Year Company_Name Directory
0      2  2015         Jeep     after
1      2  2015         Jeep     after
2      2  2015         Jeep     after
3      2  2015         Jeep     after
4      2  2015         Jeep     after
```

[ ]:
```
after_df.shape
```

[ ]:
```
(52017, 17)
```

[ ]:
```
after_superbowl_df = pd.concat((prep_concat(filenames_after_superbowl))).
↪reset_index(drop=True)
after_superbowl_df.head()
```

[ ]:
```
                                                 url  \
0  https://twitter.com/NihadAwad/status/562388095…
1  https://twitter.com/i4unews/status/56238505431…
2  https://twitter.com/LoveInshAllah/status/56237…
3  https://twitter.com/MirDAliZ/status/5623749644…
4  https://twitter.com/MaryWbn/status/56236490636…

                   Datetime            Tweet Id  \
0  2015-02-02 23:12:25+00:00  562388095522193410
1  2015-02-02 23:00:20+00:00  562385054312464384
2  2015-02-02 22:26:35+00:00  562376561643372544
3  2015-02-02 22:20:14+00:00  562374964473372672
4  2015-02-02 21:40:16+00:00  562364906365005826

                                                Text Retweet Count  \
0  We should all commend @Jeep for being inclusiv…              2
1  Jeep #SuperBowl 2015 Ad - I4U News http://t.co…              2
2  Make sure to thank @Jeep! "Racists whine abt i…              1
3  Peace is the message that was being conveyed b…              2
4  Embarrassing "@Adweek: Jeep's #SuperBowl ad is…              1
```

```
  Reply Count Like Count       Username      Display Name Followers Count  \
0           0          5      NihadAwad        Nihad Awad           35774
1           0          0        i4unews          I4U News           12101
2           0          2  LoveInshAllah   Love, InshAllah            6774
3           0          1       MirDAliZ    MIR DAWOOD ALI              15
4           0          0        MaryWbn  Mary Weatherburn            1378

  Friends Count                                   Rendered Content Day Month  \
0          1849  We should all commend @Jeep for being inclusiv…   2     2
1          4650  Jeep #SuperBowl 2015 Ad - I4U News buff.ly/1HI…   2     2
2          1842  Make sure to thank @Jeep! "Racists whine abt i…   2     2
3            89  Peace is the message that was being conveyed b…   2     2
4          1779  Embarrassing "@Adweek: Jeep's #SuperBowl ad is…   2     2

   Year Company_Name Directory
0  2015         Jeep     after
1  2015         Jeep     after
2  2015         Jeep     after
3  2015         Jeep     after
4  2015         Jeep     after
```

[ ]:
```
after_superbowl_df.shape
```

[ ]:
```
(13107, 17)
```

1.0.2 Exploring Concated Files

1.0.3 Before files

[ ]:
```
before_df.head()
```

[ ]:
```
                                                 url  \
0  https://twitter.com/UWnt2/status/5616740628090…
1  https://twitter.com/ShowDreamCar/status/561671…
2  https://twitter.com/ShowDreamCar/status/561671…
3  https://twitter.com/_VictoriaManson/status/561…
4  https://twitter.com/Jeepings/status/5616670153…

                   Datetime            Tweet Id  \
0  2015-01-31 23:55:06+00:00  561674062809022465
1  2015-01-31 23:46:28+00:00  561671888288493568
2  2015-01-31 23:43:45+00:00  561671203966840834
3  2015-01-31 23:28:26+00:00  561667350819520512
4  2015-01-31 23:27:06+00:00  561667015312961536

                                                Text  Retweet Count  \
```

```
0  Amazing weekend #off-road &amp;amp; #camping at Th…              1
1  List of things better than perfect Jeep weathe…              1
2  #tbtuesday 1 yr ago when I first started worki…              1
3  Okay, that jeep is beautiful! Loving everythin…              2
4  They call my house The Jeep Garage ;) #jeeplif…              3

   Reply Count  Like Count         Username      Display Name  \
0            0           1            UWnt2        YOUNTO.com
1            0           2      ShowDreamCar      Car Pictures
2            0           1      ShowDreamCar      Car Pictures
3            0           0  _VictoriaManson             Vicki
4            0          11          Jeepings  Northern Jeeper

   Followers Count  Friends Count  \
0              433            711
1            12775             19
2            12775             19
3             4432           4950
4             1619            738

                                  Rendered Content  Day  Month  Year  \
0  Amazing weekend #off-road &amp;amp; #camping at Th…   31      1  2015
1  List of things better than perfect Jeep weathe…   31      1  2015
2  #tbtuesday 1 yr ago when I first started worki…   31      1  2015
3  Okay, that jeep is beautiful! Loving everythin…   31      1  2015
4  They call my house The Jeep Garage ;) #jeeplif…   31      1  2015

  Company_Name Directory
0         Jeep    before
1         Jeep    before
2         Jeep    before
3         Jeep    before
4         Jeep    before
```

[ ]:
```
before_df.shape
```

[ ]:
```
(32429, 17)
```

[ ]:
```
before_df.dtypes
```

[ ]:
```
url                object
Datetime           object
Tweet Id            int64
Text               object
Retweet Count       int64
Reply Count         int64
Like Count          int64
```

```
Username            object
Display Name        object
Followers Count      int64
Friends Count        int64
Rendered Content    object
Day                  int64
Month                int64
Year                 int64
Company_Name        object
Directory           object
dtype: object
```

[ ]:
```
before_df.columns
```

[ ]:
```
Index(['url', 'Datetime', 'Tweet Id', 'Text', 'Retweet Count', 'Reply Count',
       'Like Count', 'Username', 'Display Name', 'Followers Count',
       'Friends Count', 'Rendered Content', 'Day', 'Month', 'Year',
       'Company_Name', 'Directory'],
      dtype='object')
```

[ ]:
```
before_df.describe()
```

[ ]:
```
           Tweet Id  Retweet Count   Reply Count     Like Count  \
count  3.242900e+04   32429.000000  32429.000000   32429.000000
mean   9.517949e+17      11.610380      2.506306      52.711246
std    2.978051e+17     157.906887     46.216103    1297.789061
min    5.609507e+17       1.000000      0.000000       0.000000
25%    6.957416e+17       1.000000      0.000000       1.000000
50%    9.594939e+17       1.000000      0.000000       3.000000
75%    1.223308e+18       3.000000      1.000000      10.000000
max    1.492649e+18   12668.000000   7047.000000  152794.000000

       Followers Count  Friends Count           Day         Month  \
count     3.242900e+04   3.242900e+04  32429.000000  32429.000000
mean      2.302539e+05   4.248884e+03     10.621450      1.758334
std       2.054389e+06   2.266479e+04     11.556959      0.428100
min       0.000000e+00   0.000000e+00      1.000000      1.000000
25%       6.300000e+02   3.000000e+02      3.000000      2.000000
50%       2.729000e+03   7.800000e+02      5.000000      2.000000
75%       1.880600e+04   2.154000e+03     12.000000      2.000000
max       9.108857e+07   2.201899e+06     31.000000      2.000000

              Year
count  32429.000000
mean    2017.939622
std        2.244587
min     2015.000000
```

```
25%     2016.000000
50%     2018.000000
75%     2020.000000
max     2022.000000
```

[ ]:
```
before_df.groupby('Company_Name').mean().filter(regex='Count$',axis=1)
```

[ ]:
```
                    Retweet Count  Reply Count  Like Count  Followers Count  \
Company_Name
Avocado from Mexico       7.245000     2.620000    6.650000    175989.195000
Budlight                 13.476765     3.681955   65.061557    361102.984309
Budweiser                 9.104816     2.132011   24.156941    316883.579037
Coca-Cola                12.992099     1.700903   31.886757    287951.140331
Doritos                  14.318605     2.049612   98.542636    220235.940310
Jeep                      6.201747     1.341930   24.570300     46794.937188
Mars                     14.345556     2.261556   86.350222    213877.997778
Pepsi                    15.380939     5.153717   59.462549    303302.615428
T-Mobile                 17.715569     3.726689   86.783576    402266.858426
Tide                      8.780923     1.524098   35.724638     89269.651500
Toyota                   10.728829     2.193563   53.391235    201923.424104
amazon alexa              4.341463     0.670732    9.835366    297795.983740
pringles                  8.546816     3.179775   42.898876    660119.464419
sprint                    4.962222     0.663111   15.204889     94718.465333
squarespace               5.490741     0.805556   37.513889    145110.837963
turbotax                  8.675462     5.184697   15.817942    165868.335092
uber eats                16.252708     4.699158   87.330927    370199.460890
weather tech              3.541667     1.791667   30.552083    102307.979167
wix                       4.900000     0.931250   15.093750    167993.325000

                    Friends Count
Company_Name
Avocado from Mexico   17258.600000
Budlight               2996.409777
Budweiser              4885.215297
Coca-Cola              7035.234763
Doritos                3370.938760
Jeep                   2486.914725
Mars                   4262.438222
Pepsi                  4258.319173
T-Mobile               4999.523524
Tide                   4194.474553
Toyota                 3716.802556
amazon alexa           6212.674797
pringles               2701.273408
sprint                 1826.213333
squarespace            5161.949074
turbotax               7730.812665
```

```
uber eats              4349.078219
weather tech           3918.312500
wix                    6029.181250
```

```
[ ]: before_df.groupby('Company_Name').sum().filter(regex='Count$',axis=1)
```

```
[ ]:                    Retweet Count  Reply Count  Like Count  Followers Count  \
Company_Name
Avocado from Mexico              1449          524        1330         35197839
Budlight                        22331         6101      107807        598347645
Budweiser                       16070         3763       42637        559299517
Coca-Cola                       34533         4521       84755        765374131
Doritos                         18471         2644      127120        284104363
Jeep                            14909         3226       59067        112495029
Mars                            64555        10177      388576        962450990
Pepsi                           55033        18440      212757       1085216758
T-Mobile                        41419         8713      202900        940499915
Tide                            26053         4522      105995        264863056
Toyota                          47003         9610      233907        884626521
amazon alexa                     2136          330        4839        146515624
pringles                         2282          849       11454        176251897
sprint                          11165         1492       34211        213116547
squarespace                      1186          174        8103         31343941
turbotax                         3288         1965        5995         62864099
uber eats                       13506         3905       72572        307635752
weather tech                      340          172        2933          9821566
wix                               784          149        2415         26878932

                    Friends Count
Company_Name
Avocado from Mexico        3451720
Budlight                   4965051
Budweiser                  8622405
Coca-Cola                 18699654
Doritos                    4348511
Jeep                       5978543
Mars                      19180972
Pepsi                     15236266
T-Mobile                  11688886
Tide                      12445006
Toyota                    16283312
amazon alexa               3056636
pringles                    721240
sprint                     4108980
squarespace                1114981
turbotax                   2929978
uber eats                  3614084
```

```
weather tech                376158
wix                         964669
```

[ ]:
```
before_df.groupby(['Company_Name', 'Year']).sum().filter(regex='Count$',axis=1).
↪sort_values(by = "Retweet Count",ascending = False).reset_index()
#T-mobile, coca-cola, pepsi tend to have high social engagement.
#Mars shows high number of retweet count which is mostly due to it pulling␣
↪bruno mars related tweets
```

[ ]:
```
           Company_Name  Year  Retweet Count  Reply Count  Like Count  \
0                  Mars  2018          26240         2969      109296
1              T-Mobile  2021          21228         4447      129149
2             Coca-Cola  2020          17337         1166       39144
3                 Pepsi  2015          15057         2139       20632
4                Toyota  2022          14383         4450       69698
..                  ...   ...            ...          ...         ...
87         weather tech  2016             69           18         106
88         weather tech  2022             58           23         461
89         weather tech  2018             48           15         111
90  Avocado from Mexico  2018             24            3          57
91  Avocado from Mexico  2022              7            3          20

    Followers Count  Friends Count
0         226958527        2483630
1          77505391        1576775
2          82906963        1966534
3         124483867        3166859
4         177956523        1921719
..              ...            ...
87          1609593         132362
88           645375          46910
89          1082882         130858
90            13418           7807
91             3825           5227

[92 rows x 7 columns]
```

[ ]:
```
before_df.groupby('Company_Name').median().filter(regex='Count$',axis=1)
# data is highly skeweded thats why mean and median values are so different
```

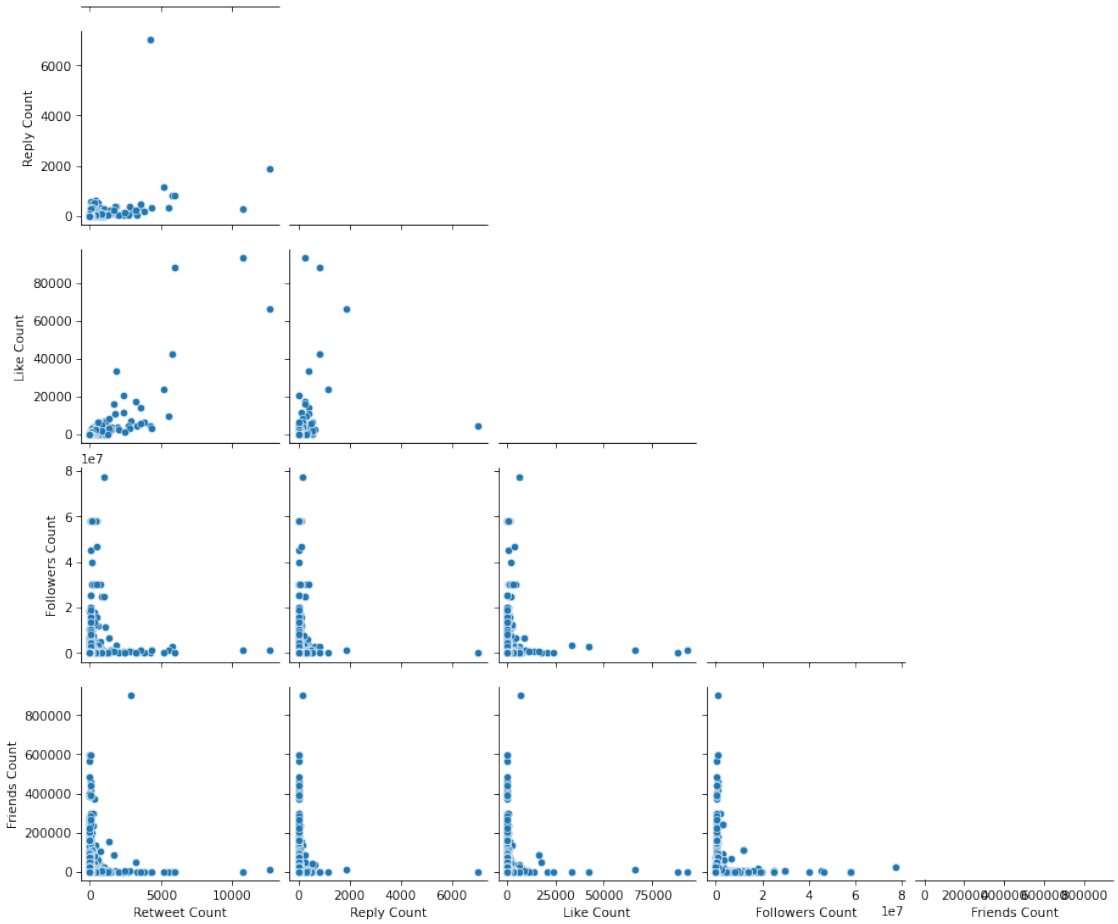
[ ]:
```
                     Retweet Count  Reply Count  Like Count  Followers Count  \
Company_Name
Avocado from Mexico            1.0          0.0         1.0          27948.0
Budlight                       1.0          0.0         3.0           4024.0
Budweiser                      1.0          0.0         2.0           2038.0
Coca-Cola                      2.0          0.0         3.0           5179.5
Doritos                        1.0          0.0         3.0           1537.5
```

Jeep	1.0	0.0	3.0	1594.5
Mars	1.0	0.0	3.0	2161.0
Pepsi	2.0	0.0	4.0	2531.0
T-Mobile	1.0	0.0	3.0	2665.5
Tide	2.0	0.0	5.0	2774.0
Toyota	2.0	0.0	3.0	4201.0
amazon alexa	1.0	0.0	3.0	7309.0
pringles	1.0	0.0	4.0	8609.0
sprint	1.0	0.0	3.0	1765.0
squarespace	2.0	0.0	3.0	4700.0
turbotax	1.0	1.0	3.0	13180.0
uber eats	1.0	1.0	3.0	1151.0
weather tech	1.0	0.0	2.0	2641.0
wix	1.0	0.0	2.0	5956.0

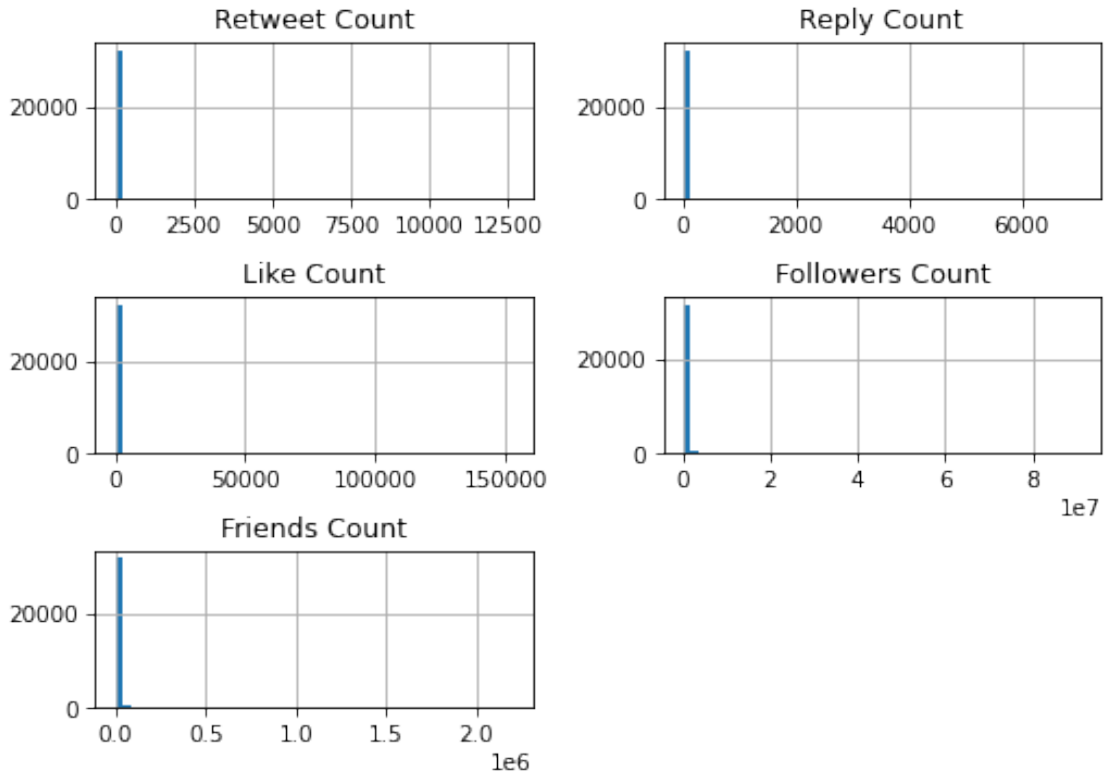
Company_Name	Friends Count
Avocado from Mexico	4987.0
Budlight	589.0
Budweiser	820.0
Coca-Cola	979.5
Doritos	699.5
Jeep	609.5
Mars	768.0
Pepsi	782.5
T-Mobile	774.0
Tide	904.0
Toyota	849.0
amazon alexa	1120.5
pringles	816.0
sprint	575.0
squarespace	942.0
turbotax	2372.0
uber eats	576.0
weather tech	874.5
wix	1255.5

[ ]:
```
sns.pairplot(before_df.filter(regex='Count$', axis = 1).reset_index(drop=True),␣
↪corner=True)
#There might be some correlation between like, reply and retweet metric
```

[ ]:
```
<seaborn.axisgrid.PairGrid at 0x7f9f592b2e80>
```



[ ]:
```
before_df.filter(regex='Count$', axis = 1).reset_index(drop=True).hist(bins =␣
↪50)
plt.show()
```



1.0.4 After Files

[ ]:
```
after_df.head()
```

[ ]:
```
                                                 url  \
0  https://twitter.com/billyg67/status/5623991633…
1  https://twitter.com/thefox1019/status/56239886…
2  https://twitter.com/ChadHaase/status/562397634…
3  https://twitter.com/CiscaPr/status/56239722094…
4  https://twitter.com/FlexinJC/status/5623969134…

                   Datetime            Tweet Id  \
0  2015-02-02 23:56:24+00:00  562399163350925312
1  2015-02-02 23:55:12+00:00  562398865127530498
2  2015-02-02 23:50:19+00:00  562397634976301056
3  2015-02-02 23:48:40+00:00  562397220943589378
4  2015-02-02 23:47:27+00:00  562396913417617408

                                                Text  Retweet Count  \
0  @SolarTJChick I would love for all jeepgirls t…              4
1  We're paying a visit to Bundoora RIGHT NOW! Ge…              1
2  Jeep stuck in mud in France 1944. http://t.co/…             11
```

```
3  Good morning sexy people! Have a sexy Tuesday!…              2
4  @YellowJeepJewel Wow. You don't need people li…              1

   Reply Count  Like Count    Username          Display Name  Followers Count  \
0            3           9    billyg67    Porsche & schmoopy             4435
1            0           2  thefox1019         The Fox 101.9            42613
2            1          14   ChadHaase            Chad Haase             1086
3            1           7     CiscaPr  Francisca Prandayani             6077
4            1           2    FlexinJC          James Miller             1156

   Friends Count                                   Rendered Content  Day  \
0           4885  @SolarTJChick I would love for all jeepgirls t…    2
1            296  We're paying a visit to Bundoora RIGHT NOW! Ge…    2
2            673  Jeep stuck in mud in France 1944. http://t.co/…    2
3           1992  Good morning sexy people! Have a sexy Tuesday!…    2
4           2090  @YellowJeepJewel Wow. You don't need people li…    2

   Month  Year Company_Name Directory
0      2  2015         Jeep     after
1      2  2015         Jeep     after
2      2  2015         Jeep     after
3      2  2015         Jeep     after
4      2  2015         Jeep     after
```

[ ]:
```
after_df.shape
```

[ ]:
```
(52017, 17)
```

[ ]:
```
after_df.describe()
```

[ ]:
```
           Tweet Id  Retweet Count   Reply Count     Like Count  \
count  5.201700e+04   52017.000000  52017.000000   52017.000000
mean   9.721033e+17      14.674876      2.624950      59.207240
std    2.970745e+17     436.852727     32.534084    1716.294218
min    5.616792e+17       1.000000      0.000000       0.000000
25%    6.967073e+17       1.000000      0.000000       1.000000
50%    9.603698e+17       1.000000      0.000000       3.000000
75%    1.224160e+18       3.000000      1.000000      11.000000
max    1.493374e+18   56260.000000   4830.000000  251520.000000

       Followers Count  Friends Count           Day    Month          Year
count     5.201700e+04   5.201700e+04  52017.000000  52017.0  52017.000000
mean      2.704329e+05   5.500573e+03      5.722821      2.0   2018.086279
std       1.446864e+06   1.932250e+04      3.294681      0.0      2.238787
min       0.000000e+00   0.000000e+00      1.000000      2.0   2015.000000
25%       6.180000e+02   3.740000e+02      3.000000      2.0   2016.000000
50%       3.076000e+03   9.580000e+02      5.000000      2.0   2018.000000
```

```
75%       2.873800e+04   2.853000e+03      8.000000      2.0   2020.000000
max       9.108921e+07   1.428329e+06     14.000000      2.0   2022.000000
```

[ ]:
```
after_df.columns
```

[ ]:
```
Index(['url', 'Datetime', 'Tweet Id', 'Text', 'Retweet Count', 'Reply Count',
       'Like Count', 'Username', 'Display Name', 'Followers Count',
       'Friends Count', 'Rendered Content', 'Day', 'Month', 'Year',
       'Company_Name', 'Directory'],
      dtype='object')
```

[ ]:
```
after_df.dtypes
```

[ ]:
```
url                object
Datetime           object
Tweet Id            int64
Text               object
Retweet Count       int64
Reply Count         int64
Like Count          int64
Username           object
Display Name       object
Followers Count     int64
Friends Count       int64
Rendered Content   object
Day                 int64
Month               int64
Year                int64
Company_Name       object
Directory          object
dtype: object
```

[ ]:
```
after_df.describe()
```

[ ]:
```
           Tweet Id  Retweet Count   Reply Count     Like Count  \
count  5.201700e+04   52017.000000  52017.000000   52017.000000
mean   9.721033e+17      14.674876      2.624950      59.207240
std    2.970745e+17     436.852727     32.534084    1716.294218
min    5.616792e+17       1.000000      0.000000       0.000000
25%    6.967073e+17       1.000000      0.000000       1.000000
50%    9.603698e+17       1.000000      0.000000       3.000000
75%    1.224160e+18       3.000000      1.000000      11.000000
max    1.493374e+18   56260.000000   4830.000000  251520.000000

       Followers Count  Friends Count           Day    Month          Year
count     5.201700e+04   5.201700e+04  52017.000000  52017.0  52017.000000
mean      2.704329e+05   5.500573e+03      5.722821      2.0   2018.086279
```

```
std       1.446864e+06   1.932250e+04      3.294681      0.0      2.238787
min       0.000000e+00   0.000000e+00      1.000000      2.0   2015.000000
25%       6.180000e+02   3.740000e+02      3.000000      2.0   2016.000000
50%       3.076000e+03   9.580000e+02      5.000000      2.0   2018.000000
75%       2.873800e+04   2.853000e+03      8.000000      2.0   2020.000000
max       9.108921e+07   1.428329e+06     14.000000      2.0   2022.000000
```

[ ]:
```
after_df.groupby('Company_Name').mean().filter(regex='Count$',axis=1)
```

[ ]:
```
                    Retweet Count  Reply Count  Like Count  Followers Count  \
Company_Name
Avocado from Mexico       7.191904     2.259370   17.219640     2.478452e+05
Budlight                 13.925171     3.357402   63.965914     1.964389e+05
Budweiser                26.420755     2.447484  100.568868     9.852033e+04
Coca-Cola                11.083294     1.477327   32.898091     1.149393e+06
Doritos                  28.838797     4.508478  133.415555     1.394784e+05
Jeep                      8.602133     2.314400   36.438933     1.136707e+05
Mars                     11.784000     1.924889   51.903778     1.676772e+05
Pepsi                    14.153778     2.732889   49.945111     1.914844e+05
T-Mobile                 16.315431     3.939682   44.693452     5.238889e+05
Tide                      9.708333     1.821333   67.296000     1.333936e+05
Toyota                    9.678152     2.289350   36.759984     1.512429e+05
amazon alexa              5.615578     1.359296   29.694724     2.103804e+05
pringles                  8.160291     2.229508   45.500911     2.878679e+05
sprint                    9.244000     1.202667   16.490222     1.183010e+05
squarespace               9.825516     1.463415   28.346154     1.379584e+05
turbotax                 10.129477     2.185491   21.375574     1.168477e+05
uber eats                38.535961     4.487552  233.738589     1.567512e+05
weather tech             10.601869     1.912150   14.528972     1.433798e+05
wix                      14.992032     1.264940   11.874502     1.905277e+05

                    Friends Count
Company_Name
Avocado from Mexico   5948.222639
Budlight              2884.718233
Budweiser             3929.434277
Coca-Cola            20977.045823
Doritos               3675.637124
Jeep                  4301.316533
Mars                  3206.970222
Pepsi                 4538.964000
T-Mobile              5519.112993
Tide                  3947.286667
Toyota                4189.750196
amazon alexa          7893.133166
pringles              2713.440801
sprint                4452.500889
```

```
squarespace            5326.975610
turbotax               3442.832874
uber eats              3008.665975
weather tech           5346.927103
wix                    3708.673307
```

[ ]:
```
after_df.groupby('Company_Name').sum().filter(regex='Count$',axis=1)
```

[ ]:
```
                     Retweet Count  Reply Count  Like Count  Followers Count  \
Company_Name
Avocado from Mexico           9594         3014       22971        330625471
Budlight                     69041        16646      317143        973943997
Budweiser                    84018         7783      319809        313294658
Coca-Cola                    46439         6190      137843       4815956816
Doritos                     127554        19941      590097        616912817
Jeep                         32258         8679      136646        426265229
Mars                         53028         8662      233567        754547250
Pepsi                        63692        12298      224753        861679950
T-Mobile                     78983        19072      216361       2536146224
Tide                         29125         5464      201888        400180732
Toyota                       49436        11694      187770        772548681
amazon alexa                  4470         1082       23637        167462786
pringles                      4480         1224       24980        158039480
sprint                       20799         2706       37103        266177245
squarespace                  10474         1560       30217        147063655
turbotax                     11031         2380       23278        127247177
uber eats                    55723         6489      337986        226662247
weather tech                  5672         1023        7773         76708183
wix                           7526          635        5961         95644914

                     Friends Count
Company_Name
Avocado from Mexico        7934929
Budlight                  14302433
Budweiser                 12495601
Coca-Cola                 87893822
Doritos                   16257343
Jeep                      16129937
Mars                      14431366
Pepsi                     20425338
T-Mobile                  26718026
Tide                      11841860
Toyota                    21401244
amazon alexa               6282934
pringles                   1489679
sprint                    10018127
squarespace                5678556
```

```
turbotax                    3749245
uber eats                   4350531
weather tech                2860606
wix                         1861754
```

[ ]:
```
after_df.groupby(['Company_Name', 'Year']).sum().filter(regex='Count$',axis=1).
↪sort_values(by = "Retweet Count",ascending = False).reset_index()
#Doritos, uber eats, Budweiser, T-mobile, pepsi tend to have high social␣
↪engagement.
```

[ ]:
```
           Company_Name  Year  Retweet Count  Reply Count  Like Count  \
0               Doritos  2018          67440         2607      261161
1             uber eats  2021          51912         3941      304557
2             Budweiser  2019          51144         2520      174237
3              T-Mobile  2021          44015         7339       71772
4                 Pepsi  2016          27852         1578       21624
..                  …   …               …            …           …
87                  wix  2018            534          291        2381
88          squarespace  2021            414          219        3133
89  Avocado from Mexico  2018            237           58         799
90         weather tech  2021            224          119        2399
91         weather tech  2022            136           75         863

    Followers Count  Friends Count
0          81322050        2725017
1         103351617        2296850
2          60397339        2107096
3         204411226        3062553
4         207102497        3965550
..              …               …
87         25227975         425222
88         24606683         757358
89          3759682         565774
90         44259564         518532
91          3502302         411503

[92 rows x 7 columns]
```

[ ]:
```
after_df.groupby('Company_Name').median().filter(regex='Count$',axis=1)
# data is highly skeweded thats why mean and median values are so different
```

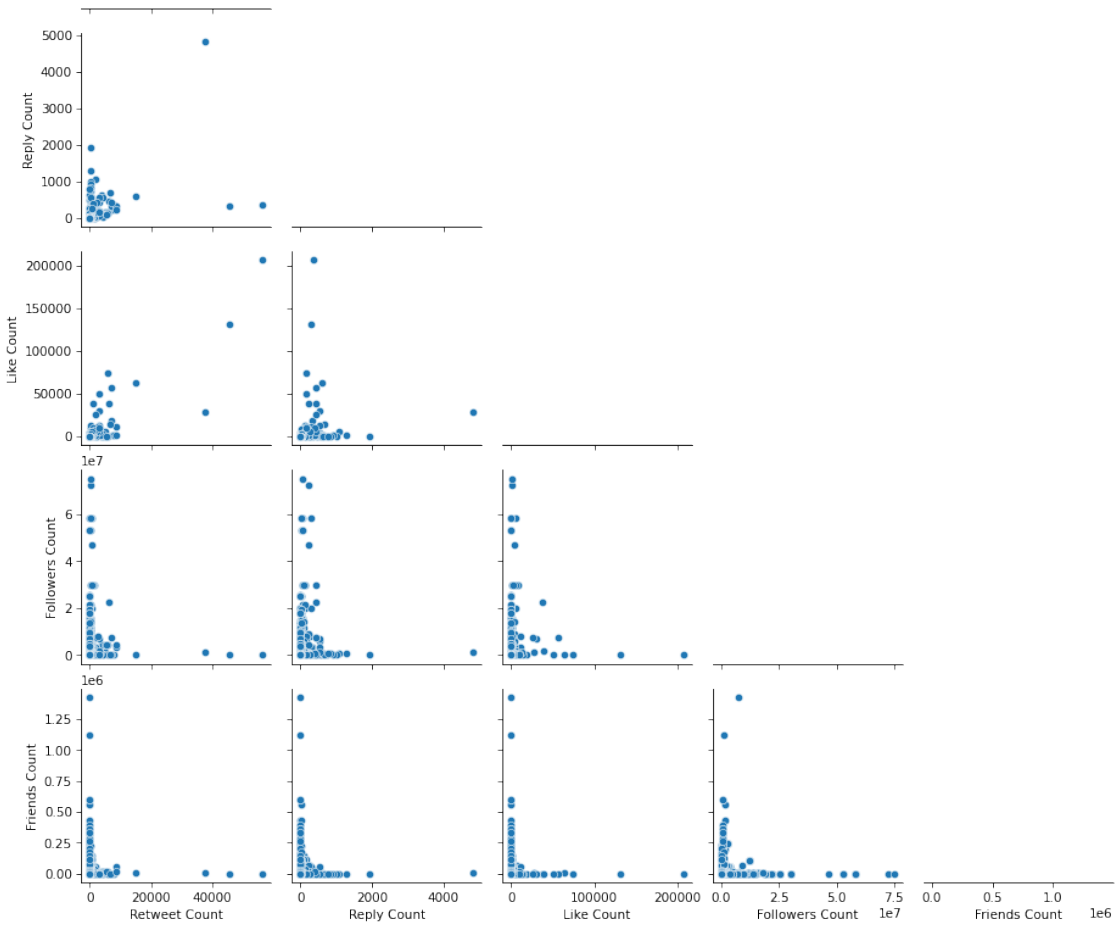
[ ]:
```
                     Retweet Count  Reply Count  Like Count  Followers Count  \
Company_Name
Avocado from Mexico            1.0          0.0         2.0           3692.0
Budlight                       1.0          0.0         4.0           3076.5
Budweiser                      1.0          0.0         3.0           2531.5
Coca-Cola                      1.0          0.0         2.0          15701.0
```

Doritos	1.0	0.0	4.0	2712.0
Jeep	1.0	0.0	4.0	2485.0
Mars	1.0	0.0	3.0	2341.5
Pepsi	1.0	0.0	3.0	1654.0
T-Mobile	1.0	0.0	3.0	4097.0
Tide	1.0	0.0	5.0	2621.0
Toyota	1.0	0.0	4.0	3876.5
amazon alexa	1.0	0.0	3.0	5351.5
pringles	1.0	0.0	6.0	2707.0
sprint	1.0	0.0	3.0	2172.5
squarespace	1.0	0.0	3.0	4690.0
turbotax	1.0	0.0	3.0	5460.0
uber eats	1.0	1.0	6.0	2281.0
weather tech	1.0	0.0	3.0	2942.0
wix	1.0	0.0	3.0	10463.0

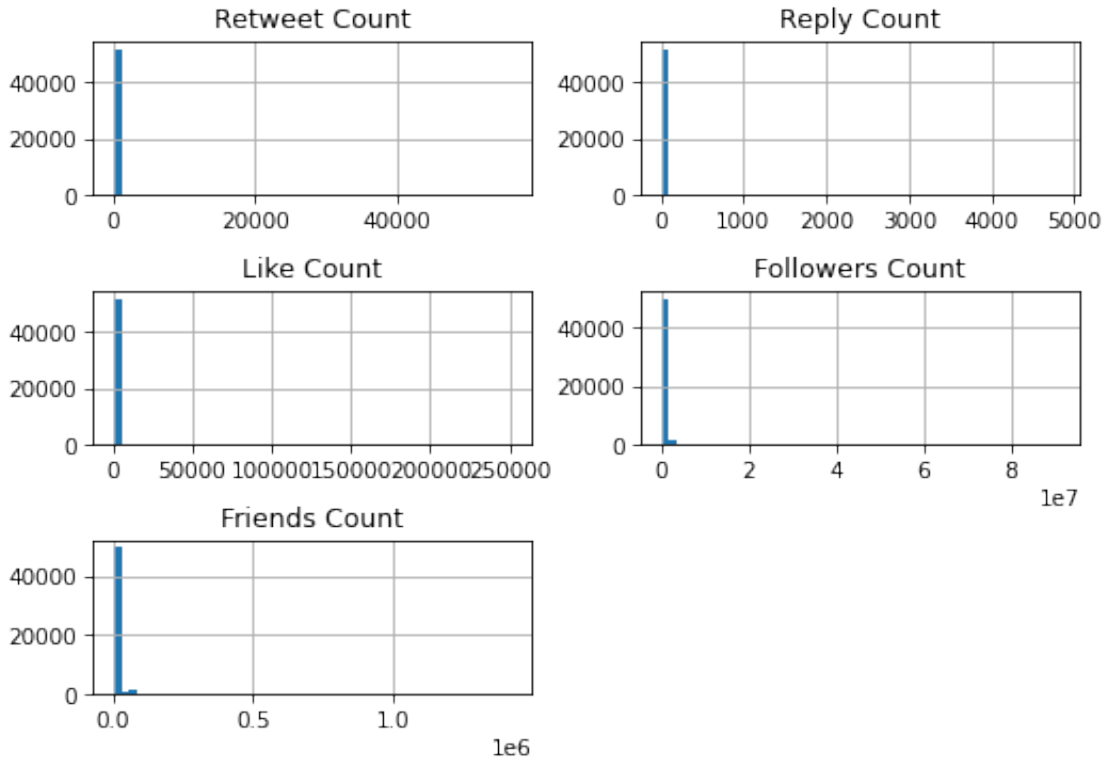
Company_Name	Friends Count
Avocado from Mexico	1281.0
Budlight	769.0
Budweiser	987.0
Coca-Cola	2444.0
Doritos	879.0
Jeep	955.0
Mars	687.0
Pepsi	983.0
T-Mobile	1129.0
Tide	941.0
Toyota	1019.0
amazon alexa	1136.5
pringles	782.0
sprint	727.5
squarespace	1064.5
turbotax	927.0
uber eats	839.0
weather tech	1163.0
wix	1910.5

[ ]:
```
sns.pairplot(after_df.filter(regex='Count$', axis = 1).reset_index(drop=True),␣
↪corner=True)
#There might be some correlation between like, reply and retweet metric
```

[ ]: <seaborn.axisgrid.PairGrid at 0x7f9ebc0f0e20>




```
[ ]: after_df.filter(regex='Count$', axis = 1).reset_index(drop=True).hist(bins = 50)
     plt.show()
     #skewed data set
```



1.0.5 After Superbowl Data

[ ]:
```
after_superbowl_df.head()
```

[ ]:
```
                                                 url  \
0  https://twitter.com/NihadAwad/status/562388095…
1  https://twitter.com/i4unews/status/56238505431…
2  https://twitter.com/LoveInshAllah/status/56237…
3  https://twitter.com/MirDAliZ/status/5623749644…
4  https://twitter.com/MaryWbn/status/56236490636…

                   Datetime            Tweet Id  \
0  2015-02-02 23:12:25+00:00  562388095522193410
1  2015-02-02 23:00:20+00:00  562385054312464384
2  2015-02-02 22:26:35+00:00  562376561643372544
3  2015-02-02 22:20:14+00:00  562374964473372672
4  2015-02-02 21:40:16+00:00  562364906365005826

                                                Text Retweet Count  \
0  We should all commend @Jeep for being inclusiv…              2
1  Jeep #SuperBowl 2015 Ad - I4U News http://t.co…              2
2  Make sure to thank @Jeep! "Racists whine abt i…              1
```

```
3  Peace is the message that was being conveyed b…             2
4  Embarrassing "@Adweek: Jeep's #SuperBowl ad is…             1

  Reply Count Like Count       Username      Display Name Followers Count  \
0           0          5      NihadAwad        Nihad Awad           35774
1           0          0        i4unews          I4U News           12101
2           0          2  LoveInshAllah   Love, InshAllah            6774
3           0          1       MirDAliZ    MIR DAWOOD ALI              15
4           0          0        MaryWbn  Mary Weatherburn            1378

  Friends Count                                   Rendered Content Day Month  \
0          1849  We should all commend @Jeep for being inclusiv…   2     2
1          4650  Jeep #SuperBowl 2015 Ad - I4U News buff.ly/1HI…   2     2
2          1842  Make sure to thank @Jeep! "Racists whine abt i…   2     2
3            89  Peace is the message that was being conveyed b…   2     2
4          1779  Embarrassing "@Adweek: Jeep's #SuperBowl ad is…   2     2

   Year Company_Name Directory
0  2015         Jeep     after
1  2015         Jeep     after
2  2015         Jeep     after
3  2015         Jeep     after
4  2015         Jeep     after
```

[ ]:
```
after_superbowl_df.shape
```

[ ]:
```
(13107, 17)
```

[ ]:
```
after_superbowl_df.dtypes
```

[ ]:
```
url               object
Datetime          object
Tweet Id          object
Text              object
Retweet Count     object
Reply Count       object
Like Count        object
Username          object
Display Name      object
Followers Count   object
Friends Count     object
Rendered Content  object
Day               object
Month             object
Year              object
Company_Name      object
Directory         object
```

```
dtype: object
```

[ ]:
```
after_superbowl_df[['Like Count','Followers Count', 'Friends Count','Reply␣
↪Count', 'Day','Month', 'Year', 'Tweet Id', 'Retweet Count']] =␣
↪after_superbowl_df[['Like Count','Followers Count', 'Friends Count','Reply␣
↪Count', 'Day','Month', 'Year','Tweet Id', 'Retweet Count']].astype(int)
```

[ ]:
```
after_superbowl_df.dtypes
```

[ ]:
```
url                 object
Datetime            object
Tweet Id             int64
Text                object
Retweet Count        int64
Reply Count          int64
Like Count           int64
Username            object
Display Name        object
Followers Count      int64
Friends Count        int64
Rendered Content    object
Day                  int64
Month                int64
Year                 int64
Company_Name        object
Directory           object
dtype: object
```

[ ]:
```
after_superbowl_df.describe()
```

[ ]:
```
           Tweet Id  Retweet Count   Reply Count     Like Count  \
count  1.310700e+04   13107.000000  13107.000000   13107.000000
mean   9.277131e+17      11.595789      3.191882      44.808270
std    2.773782e+17     239.282196    172.259675    1118.097405
min    5.616825e+17       1.000000      0.000000       0.000000
25%    6.965084e+17       1.000000      0.000000       1.000000
50%    9.603002e+17       1.000000      0.000000       4.000000
75%    1.092232e+18       3.000000      1.000000      12.000000
max    1.493367e+18   25666.000000  19694.000000  121226.000000

       Followers Count  Friends Count           Day    Month          Year
count     1.310700e+04   1.310700e+04  13107.000000  13107.0  13107.000000
mean      1.319085e+05   6.067258e+03      5.378729      2.0   2017.752270
std       1.060089e+06   3.755447e+04      2.959465      0.0      2.090809
min       0.000000e+00   0.000000e+00      1.000000      2.0   2015.000000
25%       5.870000e+02   4.440000e+02      3.000000      2.0   2016.000000
50%       2.823000e+03   1.051000e+03      5.000000      2.0   2018.000000
```

```
75%       1.858600e+04   2.791500e+03      7.000000      2.0   2019.000000
max       5.652033e+07   1.431616e+06     14.000000      2.0   2022.000000
```

[ ]:
```
after_superbowl_df.columns
```

[ ]:
```
Index(['url', 'Datetime', 'Tweet Id', 'Text', 'Retweet Count', 'Reply Count',
       'Like Count', 'Username', 'Display Name', 'Followers Count',
       'Friends Count', 'Rendered Content', 'Day', 'Month', 'Year',
       'Company_Name', 'Directory'],
      dtype='object')
```

[ ]:
```
after_superbowl_df.describe()
```

[ ]:
```
           Tweet Id  Retweet Count   Reply Count     Like Count  \
count  1.310700e+04   13107.000000  13107.000000   13107.000000
mean   9.277131e+17      11.595789      3.191882      44.808270
std    2.773782e+17     239.282196    172.259675    1118.097405
min    5.616825e+17       1.000000      0.000000       0.000000
25%    6.965084e+17       1.000000      0.000000       1.000000
50%    9.603002e+17       1.000000      0.000000       4.000000
75%    1.092232e+18       3.000000      1.000000      12.000000
max    1.493367e+18   25666.000000  19694.000000  121226.000000

       Followers Count  Friends Count           Day    Month          Year
count     1.310700e+04   1.310700e+04  13107.000000  13107.0  13107.000000
mean      1.319085e+05   6.067258e+03      5.378729      2.0   2017.752270
std       1.060089e+06   3.755447e+04      2.959465      0.0      2.090809
min       0.000000e+00   0.000000e+00      1.000000      2.0   2015.000000
25%       5.870000e+02   4.440000e+02      3.000000      2.0   2016.000000
50%       2.823000e+03   1.051000e+03      5.000000      2.0   2018.000000
75%       1.858600e+04   2.791500e+03      7.000000      2.0   2019.000000
max       5.652033e+07   1.431616e+06     14.000000      2.0   2022.000000
```

[ ]:
```
after_superbowl_df.groupby('Company_Name').mean().filter(regex='Count$',axis=1)
```

[ ]:
```
                    Retweet Count  Reply Count  Like Count  Followers Count  \
Company_Name
Avocado from Mexico      11.680412     2.340206   33.546392    183068.932990
Budlight                  8.896965     2.003195   31.234026    155610.766773
Budweiser                 9.150483     1.553663   25.322544    108450.647927
Coca-Cola                10.299903     1.159688   29.458617    101250.444985
Doritos                   7.120177     1.314991   26.405440    101498.378874
Jeep                      7.729323     2.742213   39.822771    178738.532760
Mars                     27.071063     1.209987  109.729193    127234.216389
Pepsi                    14.805297    15.576951   47.765927    114611.262706
T-Mobile                 12.120690     1.727011   31.830460    209313.149425
Tide                     10.729242     2.128159   71.792419    105836.990975
```

```
Toyota                    5.205357     1.317602   21.477041    105388.982143
amazon alexa              6.924528     1.295597   27.427673    185127.226415
pringles                  6.197802     1.538462   33.494505    164583.439560
sprint                   12.142180     0.753555   11.872038    157998.962085
squarespace               9.819512     1.526829   52.702439    233409.073171
turbotax                  9.131868     5.450549   37.895604    128935.675824
uber eats                 8.367925     3.650943   62.452830    373575.216981
weather tech              5.500000     0.500000   11.500000     21032.750000
wix                       5.948980     0.877551   15.632653    171546.040816

                     Friends Count
Company_Name
Avocado from Mexico    7923.639175
Budlight               5566.335463
Budweiser              6266.590006
Coca-Cola              5221.542356
Doritos                5506.662872
Jeep                   5920.239527
Mars                   2370.967990
Pepsi                  9856.545455
T-Mobile               6892.537356
Tide                   5737.084838
Toyota                 7023.528061
amazon alexa           7277.471698
pringles               7355.857143
sprint                 9706.815166
squarespace           10410.848780
turbotax               4736.170330
uber eats              5628.820755
weather tech           2339.000000
wix                    6053.571429
```

[ ]:
```
after_superbowl_df.groupby('Company_Name').sum().filter(regex='Count$',axis=1)
```

[ ]:
```
                     Retweet Count  Reply Count  Like Count  Followers Count  \
Company_Name
Avocado from Mexico           2266          454        6508         35515373
Budlight                     11139         2508       39105        194824680
Budweiser                    16114         2736       44593        190981591
Coca-Cola                    10578         1191       30254        103984207
Doritos                      11257         2079       41747        160468937
Jeep                          7196         2553       37075        166405574
Mars                         42285         1890      171397        198739846
Pepsi                        20683        21761       66729        160111934
T-Mobile                      4218          601       11077         72840976
Tide                         11888         2358       79546        117267386
Toyota                        4081         1033       16838         82624962
```

```
amazon alexa                1101          206        4361          29435229
pringles                     564          140        3048          14977093
sprint                      2562          159        2505          33337781
squarespace                 2013          313       10804          47848860
turbotax                    1662          992        6897          23466293
uber eats                   1774          774       13240          79197946
weather tech                  22            2          46             84131
wix                          583           86        1532          16811512

                      Friends Count
Company_Name
Avocado from Mexico         1537186
Budlight                    6969052
Budweiser                  11035465
Coca-Cola                   5362524
Doritos                     8706034
Jeep                        5511743
Mars                        3703452
Pepsi                      13769594
T-Mobile                    2398603
Tide                        6356690
Toyota                      5506446
amazon alexa                1157118
pringles                     669383
sprint                      2048138
squarespace                 2134224
turbotax                     861983
uber eats                   1193310
weather tech                   9356
wix                          593250
```

[ ]:
```
after_superbowl_df.groupby(['Company_Name', 'Year']).sum().
↪filter(regex='Count$',axis=1).sort_values(by = "Retweet Count",ascending =␣
↪False).reset_index()
#Tide, coca-cola, budweiser tend to have high social engagement.
#mars is showing more retweets which might be from Bruno mars
```

[ ]:
```
           Company_Name  Year  Retweet Count  Reply Count  Like Count  \
0                  Mars  2018          27966          415      128886
1                  Mars  2016           8733          451       14946
2                  Tide  2018           7660         1227       48945
3             Coca-Cola  2017           7400          744       21968
4             Budweiser  2017           7100          986       20196
..                  ...   ...            ...          ...         ...
85  Avocado from Mexico  2016             22            1          24
86         weather tech  2015             20            2          41
87  Avocado from Mexico  2019             15            3          44
```

```
88          weather tech  2016              1            0           1
89          weather tech  2018              1            0           4

    Followers Count  Friends Count
0           2412506         308350
1         106003566        2355075
2          62231819        2733554
3          61625682        2626066
4          94319408        3755739
..              …               …
85            54271           18403
86            81099            6711
87          1068739          103345
88              693             617
89             2339            2028

[90 rows x 7 columns]
```

[ ]:
```
after_superbowl_df.groupby('Company_Name').median().
 ↪filter(regex='Count$',axis=1)
# data is highly skeweded thats why mean and median values are so different
```

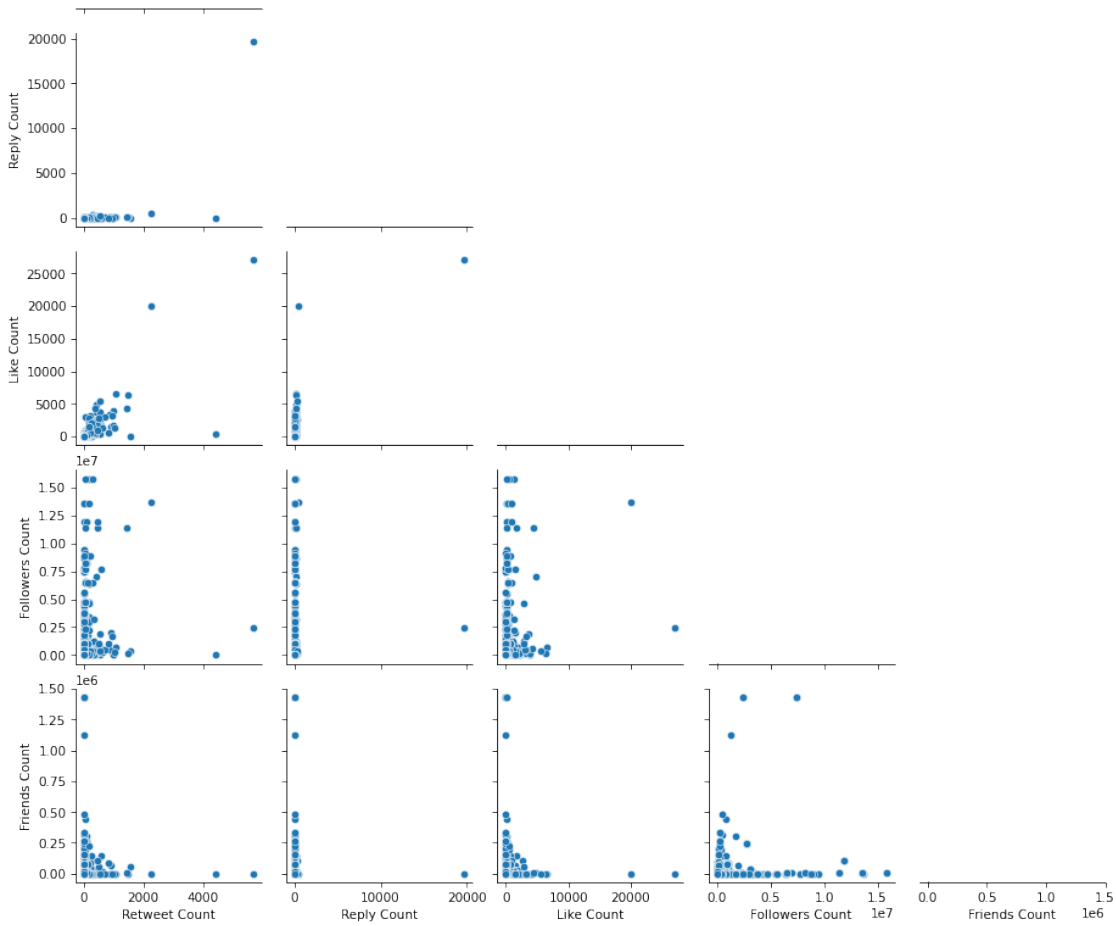
[ ]:
```
                    Retweet Count  Reply Count  Like Count  Followers Count  \
Company_Name
Avocado from Mexico            2.0          0.0         5.0           5066.0
Budlight                       1.0          0.0         4.0           3585.5
Budweiser                      2.0          0.0         4.0           3411.0
Coca-Cola                      2.0          0.0         4.0           2764.0
Doritos                        1.0          0.0         3.0           2367.0
Jeep                           1.0          0.0         5.0           4630.0
Mars                           2.0          0.0         3.0           1333.5
Pepsi                          1.0          0.0         4.0           2035.0
T-Mobile                       1.0          0.0         3.0           4113.0
Tide                           1.0          0.0         4.0           1681.5
Toyota                         1.0          0.0         5.0           4856.0
amazon alexa                   2.0          0.0         4.0           5138.0
pringles                       2.0          0.0         6.0           9670.0
sprint                         1.0          0.0         3.0           3076.0
squarespace                    2.0          0.0         5.0           9670.0
turbotax                       2.0          1.0         5.0          19223.0
uber eats                      1.0          0.0         5.0           3727.0
weather tech                   2.0          0.0         4.0          18026.0
wix                            1.5          0.0         3.0           6406.5

                    Friends Count
Company_Name
Avocado from Mexico         1190.5
```

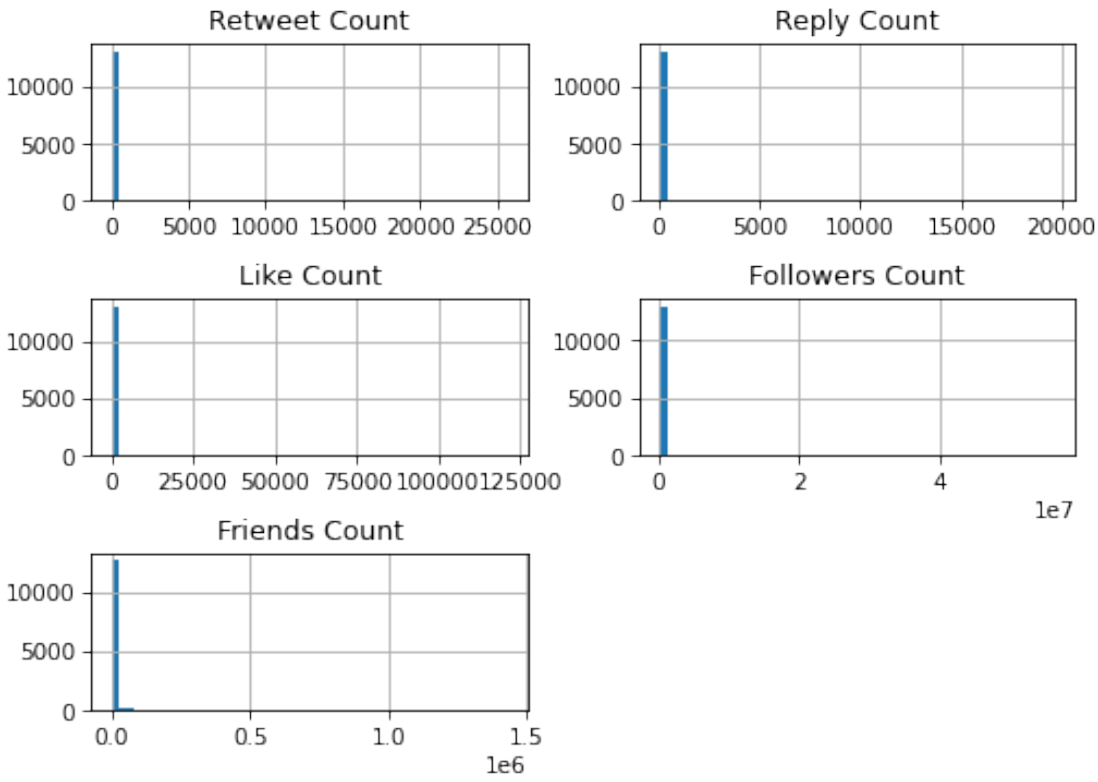
```
Budlight                  1006.0
Budweiser                 1184.0
Coca-Cola                 1155.0
Doritos                   1069.0
Jeep                      1395.0
Mars                       686.0
Pepsi                     1010.0
T-Mobile                  1179.0
Tide                       955.5
Toyota                    1462.0
amazon alexa              1392.0
pringles                  1392.0
sprint                    1066.0
squarespace               1708.0
turbotax                   927.0
uber eats                 1391.5
weather tech              1322.5
wix                       1416.5
```

[ ]:
```
sns.pairplot(after_superbowl_df.filter(regex='Count$', axis = 1).
 ↪reset_index(drop=True), corner=True)
#There might be some correlation between like, reply and retweet metric
```

[ ]:
```
<seaborn.axisgrid.PairGrid at 0x7f9b3b102a00>
```



[ ]:
```
after_superbowl_df.filter(regex='Count$', axis = 1).reset_index(drop=True).
↪hist(bins = 50)
plt.show()
```


Retweet Count
10000
5000
0
0
5000
10000
15000
20000
25000
Reply Count
10000
5000
0
0
5000
10000
15000
20000
Like Count
10000
5000
0
0
25000
50000
75000
100000
125000
Followers Count
10000
5000
0
0
2
4
1e7
Friends Count
10000
5000
0
0.0
0.5
1.0
1.5
1e6

[ ]: