Basic Packages

In [1]:
```python
#dependencies and setup
import pandas as pd
pd.options.display.float_format = '{:,.2f}'.format
import os
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.express as px
from openpyxl import Workbook
import numpy as np
# SQLite dependencies
import sqlite3
from sqlalchemy import create_engine, text
from sqlalchemy import Column, Integer, String, Float
from pandas_profiling import ProfileReport
```

DB Connection

In [2]:
```python
#  SQLite DB creation and establishing connection
database_path = "NJ_County_DB.sqlite"
engine = create_engine(f"sqlite:///{database_path}", echo=True)
sqlite_connection = engine.connect()
```

Data Pull

In [3]:
```python
sql_query = """SELECT name FROM sqlite_master
  WHERE type='table';"""
tbls= pd.read_sql(sql_query,sqlite_connection)
tbls
```

```
2023-03-23 01:08:45,119 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT name FROM sqlite_master
  WHERE type='table';")
2023-03-23 01:08:45,123 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,126 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT name FROM sqlite_master
  WHERE type='table';")
2023-03-23 01:08:45,127 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,128 INFO sqlalchemy.engine.Engine SELECT name FROM sqlite_master
  WHERE type='table';
2023-03-23 01:08:45,129 INFO sqlalchemy.engine.Engine [raw sql] ()
```

Out[3]:

	name
0	nj_property_tax
1	nj_mortgage_rates
2	nj_population
3	nj_zillow_house_value_index
4	nj_food_desert
5	nj_poverty_median_income
6	nj_crime_detail
7	nj_school_performance
8	nj_zillow_observed_rent_index
9	nj_adi
10	nj_counties_dist_to_major_cities

In [4]:
```python
for i in tbls['name'].tolist():
    sql_query = f"""SELECT * FROM {i};"""
    globals()[f'{i}_df'] = pd.read_sql(sql_query,sqlite_connection)
    print(f'Data from {i} retreived!')
```

```
2023-03-23 01:08:45,164 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_property_tax;")
2023-03-23 01:08:45,165 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,167 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_property_tax;")
2023-03-23 01:08:45,168 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,170 INFO sqlalchemy.engine.Engine SELECT * FROM nj_property_tax;
2023-03-23 01:08:45,174 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_property_tax retreived!
2023-03-23 01:08:45,197 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_mortgage_rate
s;")
2023-03-23 01:08:45,198 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,201 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_mortgage_rate
s;")
2023-03-23 01:08:45,203 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,205 INFO sqlalchemy.engine.Engine SELECT * FROM nj_mortgage_rates;
2023-03-23 01:08:45,206 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_mortgage_rates retreived!
2023-03-23 01:08:45,211 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_population;")
2023-03-23 01:08:45,212 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,215 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_population;")
2023-03-23 01:08:45,217 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,219 INFO sqlalchemy.engine.Engine SELECT * FROM nj_population;
2023-03-23 01:08:45,220 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_population retreived!
2023-03-23 01:08:45,224 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_zillow_house_val
ue_index;")
2023-03-23 01:08:45,225 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,227 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_zillow_house_val
ue_index;")
2023-03-23 01:08:45,228 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,230 INFO sqlalchemy.engine.Engine SELECT * FROM nj_zillow_house_value_index;
2023-03-23 01:08:45,232 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_zillow_house_value_index retreived!
2023-03-23 01:08:45,246 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_food_desert;")
2023-03-23 01:08:45,247 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,249 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_food_desert;")
2023-03-23 01:08:45,250 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,251 INFO sqlalchemy.engine.Engine SELECT * FROM nj_food_desert;
2023-03-23 01:08:45,252 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_food_desert retreived!
2023-03-23 01:08:45,401 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_poverty_median_i
ncome;")
2023-03-23 01:08:45,402 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,403 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_poverty_median_i
ncome;")
2023-03-23 01:08:45,404 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,404 INFO sqlalchemy.engine.Engine SELECT * FROM nj_poverty_median_income;
2023-03-23 01:08:45,405 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_poverty_median_income retreived!
2023-03-23 01:08:45,410 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_crime_detail;")
2023-03-23 01:08:45,410 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,412 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_crime_detail;")
2023-03-23 01:08:45,412 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,413 INFO sqlalchemy.engine.Engine SELECT * FROM nj_crime_detail;
2023-03-23 01:08:45,414 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_crime_detail retreived!
2023-03-23 01:08:45,475 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_school_performan
ce;")
2023-03-23 01:08:45,475 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,477 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_school_performan
ce;")
2023-03-23 01:08:45,477 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,478 INFO sqlalchemy.engine.Engine SELECT * FROM nj_school_performance;
2023-03-23 01:08:45,478 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_school_performance retreived!
2023-03-23 01:08:45,559 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_zillow_observed_
rent_index;")
2023-03-23 01:08:45,560 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,561 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_zillow_observed_
rent_index;")
2023-03-23 01:08:45,561 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,562 INFO sqlalchemy.engine.Engine SELECT * FROM nj_zillow_observed_rent_index;
2023-03-23 01:08:45,563 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_zillow_observed_rent_index retreived!
2023-03-23 01:08:45,567 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_adi;")
2023-03-23 01:08:45,567 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,569 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_adi;")
2023-03-23 01:08:45,570 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,572 INFO sqlalchemy.engine.Engine SELECT * FROM nj_adi;
2023-03-23 01:08:45,574 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_adi retreived!
2023-03-23 01:08:45,599 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("SELECT * FROM nj_counties_dist_to
_major_cities;")
2023-03-23 01:08:45,600 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,602 INFO sqlalchemy.engine.Engine PRAGMA temp.table_info("SELECT * FROM nj_counties_dist_to
_major_cities;")
2023-03-23 01:08:45,602 INFO sqlalchemy.engine.Engine [raw sql] ()
2023-03-23 01:08:45,603 INFO sqlalchemy.engine.Engine SELECT * FROM nj_counties_dist_to_major_cities;
2023-03-23 01:08:45,604 INFO sqlalchemy.engine.Engine [raw sql] ()
Data from nj_counties_dist_to_major_cities retreived!
```

In [6]:
```python
for name in vars().keys():
    if '_df' in name:
        print(name)
```

```
nj_property_tax_df
nj_mortgage_rates_df
nj_population_df
nj_zillow_house_value_index_df
nj_food_desert_df
nj_poverty_median_income_df
nj_crime_detail_df
nj_school_performance_df
nj_zillow_observed_rent_index_df
nj_adi_df
nj_counties_dist_to_major_cities_df
```

Aggregations

Property Tax

In [7]:
```python
nj_property_tax_df.head()
```

Out[7]:

	county_code	county_name	district_code	district_name	year	tax_rate
0	001	ATLANTIC	01	ABSECON CITY	2017	3.27
1	001	ATLANTIC	01	ATLANTIC CITY CITY	2017	3.42
2	001	ATLANTIC	01	BRIGANTINE CITY	2017	1.76
3	001	ATLANTIC	01	BUENA BORO	2017	3.03
4	001	ATLANTIC	01	BUENA VISTA TWP	2017	2.46

In [8]:
```python
agg_nj_property_tax_df=nj_property_tax_df.groupby(['county_name','year'],as_index=False).\
                        agg(min_tax_rate=('tax_rate','min'),\
                          avg_tax_rate=('tax_rate','mean'),\
                          max_tax_rate=('tax_rate','max'))
agg_nj_property_tax_df.head()
```

Out[8]:

	county_name	year	min_tax_rate	avg_tax_rate	max_tax_rate
0	ATLANTIC	2017	0.96	2.91	4.69
1	ATLANTIC	2018	0.98	2.93	5.26
2	ATLANTIC	2019	0.98	2.90	5.25
3	ATLANTIC	2020	0.99	2.94	5.28
4	ATLANTIC	2021	0.97	2.97	5.23

Crime

In [9]:
```python
nj_crime_detail_df.head()
```

Out[9]:

	county_name	year	agency	report_type	population	murder	rape	robbery	assault	burglary	larceny	auto_theft	total
0	ATLANTIC	2017	Absecon	Number of Offenses	8,261.00	1.00	2.00	5.00	3.00	27.00	194.00	8.00	240.00
1	ATLANTIC	2017	Absecon	Rate Per 100,000	8,261.00	12.11	24.21	60.53	36.32	326.84	2,348.38	96.84	2,905.22
2	ATLANTIC	2017	Atlantic City	Number of Offenses	38,601.00	13.00	24.00	227.00	161.00	319.00	1,298.00	115.00	2,157.00
3	ATLANTIC	2017	Atlantic City	Rate Per 100,000	38,601.00	33.68	62.17	588.07	417.09	826.40	3,362.61	297.92	5,587.94
4	ATLANTIC	2017	Brigantine	Number of Offenses	8,976.00	0.00	0.00	0.00	2.00	24.00	84.00	2.00	112.00

In [10]:
```python
agg_nj_crime_detail_df=nj_crime_detail_df[nj_crime_detail_df['report_type']=='Number of Offenses'\
                                          ].groupby(['county_name','year'],as_index=False).sum()
agg_nj_crime_detail_df.drop(['population','total'],axis=1,inplace=True)
agg_nj_crime_detail_df.head()
```

Out[10]:

	county_name	year	murder	rape	robbery	assault	burglary	larceny	auto_theft
0	ATLANTIC	2017	21.00	45.00	337.00	399.00	1,159.00	4,756.00	271.00
1	ATLANTIC	2018	16.00	46.00	231.00	391.00	865.00	4,657.00	196.00
2	ATLANTIC	2019	13.00	49.00	272.00	338.00	857.00	5,174.00	283.00
3	ATLANTIC	2020	17.00	55.00	183.00	410.00	793.00	3,978.00	283.00
4	BERGEN	2017	4.00	64.00	237.00	388.00	1,166.00	7,463.00	543.00

Shool Rankings

In [11]:
```python
nj_school_performance_df.head()
```

Out[11]:

	rank	school	grades	district	students	free_lunch_rec	total_exp	score	year	school_type	type	address	city	
0	1	Deane-Porter Elementary School	PK, KG-3	Rumson Borough School District	384	0.00	11,020.00	98.30	2017	Elementary	Public	50 Blackpoint Road	Rumson	077
1	2	School 28	PK, KG-8	Paterson Public School District	488	82.60	9,579.00	97.80	2017	Elementary	Public	200 Presidential Boulevard	Paterson	075
2	3	Thomas Edison EnergySmart Charter School	K-12	Thomas Edison EnergySmart Charter School	421	9.70	7,387.00	96.50	2017	Elementary	Public, Charter, Alternative	150 Pierce Street	Somerset	088
3	4	Terence C. Reilly School No 7	2-8	Elizabeth Public Schools	1055	73.50	7,777.00	96.20	2017	Elementary	Public	436 First Avenue	Elizabeth	072
4	5	Deerfield School	K-4	Millburn Township School District	563	0.50	13,641.00	96.10	2017	Elementary	Public	26 Troy Lane	Short Hills	070

In [12]:
```python
nj_school_performance_df['type'].unique()
```

Out[12]:
```
array(['Public', 'Public, Charter, Alternative', 'Public, Charter',
       'Public, Alternative'], dtype=object)
```

In [13]:
```python
nj_school_performance_df['school_type'].unique()
```

Out[13]:
```
array(['Elementary', 'Middle', 'High'], dtype=object)
```

In [14]:
```python
for i in nj_school_performance_df['school_type'].unique().tolist():
    globals()[f'{i.lower()}_school_df']=nj_school_performance_df.groupby(['county_name','year'],\
                                                   as_index=False).agg(avg_std_cnt=('students','mean'),\
                                                                  avg_exp=('total_exp','mean'),\
                                                                  avg_score=('score','mean'),\
                                                                  min_std_cnt=('students','min'),\
                                                                  min_exp=('total_exp','min'),\
                                                                  min_score=('score','min'),\
                                                                  max_std_cnt=('students','max'),\
                                                                  max_exp=('total_exp','max'),\
                                                                  max_score=('score','max'))
```

In [15]:
```python
school_df=pd.merge(elementary_school_df, middle_school_df, on=['county_name','year'],\
                  how='inner', suffixes=('_ele', '_mid'))
high_school_df.columns = ['county_name', 'year','avg_std_cnt_high', 'avg_exp_high', 'avg_score_high',\
                          'min_std_cnt_high', 'min_exp_high', 'min_score_high', 'max_std_cnt_high',\
                          'max_exp_high', 'max_score_high']
school_df=pd.merge(school_df, high_school_df, on=['county_name','year'],\
                  how='inner')
school_df.head()
```

Out[15]:

	county_name	year	avg_std_cnt_ele	avg_exp_ele	avg_score_ele	min_std_cnt_ele	min_exp_ele	min_score_ele	max_std_cnt_ele	max_exp_ele
0	ATLANTIC	2017	625.61	12,472.54	36.37	118	6,322.00	2.90	2365	19,102.00
1	ATLANTIC	2018	618.25	12,597.75	34.69	114	6,322.00	2.00	2367	19,102.00
2	ATLANTIC	2019	634.47	12,778.11	31.46	101	6,322.00	2.90	2357	19,102.00
3	ATLANTIC	2020	558.77	12,570.73	34.88	105	6,322.00	5.30	2332	19,102.00
4	BERGEN	2017	528.94	11,803.85	69.79	13	3,269.00	13.90	1832	27,835.00

5 rows × 29 columns

Merging all

In [16]:
```python
nj_zillow_observed_rent_index_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 3 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   county_name          180 non-null    object
 1   year                 180 non-null    int64
 2   observed_rent_index  157 non-null    float64
dtypes: float64(1), int64(1), object(1)
memory usage: 4.3+ KB
```

In [17]:
```python
nj_zillow_observed_rent_index_df['county_name'].unique()
```

Out[17]:
```
array(['ATLANTIC', 'BERGEN', 'BURLINGTON', 'CAMDEN', 'CUMBERLAND',
       'ESSEX', 'GLOUCESTER', 'HUDSON', 'HUNTERDON', 'MERCER',
       'MIDDLESEX', 'MONMOUTH', 'MORRIS', 'OCEAN', 'PASSAIC', 'SALEM',
       'SOMERSET', 'SUSSEX', 'UNION', 'WARREN'], dtype=object)
```

In [18]:
```python
nj_zillow_observed_rent_index_df[nj_zillow_observed_rent_index_df['observed_rent_index'].isnull()]
```

Out[18]:

	county_name	year	observed_rent_index
36	CUMBERLAND	2015	NaN
37	CUMBERLAND	2016	NaN
38	CUMBERLAND	2017	NaN
39	CUMBERLAND	2018	NaN
40	CUMBERLAND	2019	NaN
41	CUMBERLAND	2020	NaN
42	CUMBERLAND	2021	NaN
43	CUMBERLAND	2022	NaN
135	SALEM	2015	NaN
136	SALEM	2016	NaN
137	SALEM	2017	NaN
138	SALEM	2018	NaN
139	SALEM	2019	NaN
140	SALEM	2020	NaN
141	SALEM	2021	NaN
142	SALEM	2022	NaN
153	SUSSEX	2015	NaN
154	SUSSEX	2016	NaN
155	SUSSEX	2017	NaN
156	SUSSEX	2018	NaN
157	SUSSEX	2019	NaN
158	SUSSEX	2020	NaN
159	SUSSEX	2021	NaN

Zillow rent data is incomplete and will be disregarded.

In [19]:
```python
final_df=pd.merge(agg_nj_property_tax_df, agg_nj_crime_detail_df, on = ['county_name','year'],how='inner')
final_df=pd.merge(final_df, school_df, on = ['county_name','year'],how='inner')
final_df=pd.merge(final_df, nj_population_df, on = ['county_name','year'],how='inner')
final_df=pd.merge(final_df, nj_mortgage_rates_df, on = 'year',how='inner')
final_df=pd.merge(final_df, nj_zillow_house_value_index_df, on = ['county_name','year'],how='left')
# final_df=pd.merge(final_df, nj_zillow_observed_rent_index_df, on = ['county_name','year'],how='left')
final_df=pd.merge(final_df, nj_poverty_median_income_df[['county_name','year',\
                                                       'median_hh_income','poverty_count',\
                                                       'poverty_rate']], on =['county_name','year'],how='inner
len(final_df)
```

Out[19]:
```
420
```

In [20]:
```python
final_df.head()
```

Out[20]:

	county_name	year	min_tax_rate	avg_tax_rate	max_tax_rate	murder	rape	robbery	assault	burglary	...	est_pop	apr_30	points_30	ap
0	ATLANTIC	2017	0.96	2.91	4.69	21.00	45.00	337.00	399.00	1,159.00	...	265446	3.99	0.50	
1	ATLANTIC	2017	0.96	2.91	4.69	21.00	45.00	337.00	399.00	1,159.00	...	265446	3.99	0.50	
2	ATLANTIC	2017	0.96	2.91	4.69	21.00	45.00	337.00	399.00	1,159.00	...	265446	3.99	0.50	
3	ATLANTIC	2017	0.96	2.91	4.69	21.00	45.00	337.00	399.00	1,159.00	...	265446	3.99	0.50	
4	ATLANTIC	2017	0.96	2.91	4.69	21.00	45.00	337.00	399.00	1,159.00	...	265446	3.99	0.50	

5 rows × 49 columns

In [21]:
```python
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 420 entries, 0 to 419
Data columns (total 49 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   county_name        420 non-null    object
 1   year               420 non-null    int64
 2   min_tax_rate       420 non-null    float64
 3   avg_tax_rate       420 non-null    float64
 4   max_tax_rate       420 non-null    float64
 5   murder             420 non-null    float64
 6   rape               420 non-null    float64
 7   robbery            420 non-null    float64
 8   assault            420 non-null    float64
 9   burglary           420 non-null    float64
 10  larceny            420 non-null    float64
 11  auto_theft         420 non-null    float64
 12  avg_std_cnt_ele    420 non-null    float64
 13  avg_exp_ele        420 non-null    float64
 14  avg_score_ele      420 non-null    float64
 15  min_std_cnt_ele    420 non-null    int64
 16  min_exp_ele        420 non-null    float64
 17  min_score_ele      420 non-null    float64
 18  max_std_cnt_ele    420 non-null    int64
 19  max_exp_ele        420 non-null    float64
 20  max_score_ele      420 non-null    float64
 21  avg_std_cnt_mid    420 non-null    float64
 22  avg_exp_mid        420 non-null    float64
 23  avg_score_mid      420 non-null    float64
 24  min_std_cnt_mid    420 non-null    int64
 25  min_exp_mid        420 non-null    float64
 26  min_score_mid      420 non-null    float64
 27  max_std_cnt_mid    420 non-null    int64
 28  max_exp_mid        420 non-null    float64
 29  max_score_mid      420 non-null    float64
 30  avg_std_cnt_high   420 non-null    float64
 31  avg_exp_high       420 non-null    float64
 32  avg_score_high     420 non-null    float64
 33  min_std_cnt_high   420 non-null    int64
 34  min_exp_high       420 non-null    float64
 35  min_score_high     420 non-null    float64
 36  max_std_cnt_high   420 non-null    int64
 37  max_exp_high       420 non-null    float64
 38  max_score_high     420 non-null    float64
 39  est_pop            420 non-null    int64
 40  apr_30             420 non-null    float64
 41  points_30          420 non-null    float64
 42  apr_15             420 non-null    float64
 43  points_15          420 non-null    float64
 44  num_of_bedrooms    420 non-null    int64
 45  house_value_index  420 non-null    float64
 46  median_hh_income   420 non-null    int64
 47  poverty_count      420 non-null    int64
 48  poverty_rate       420 non-null    float64
dtypes: float64(37), int64(11), object(1)
memory usage: 164.1+ KB
```

In [22]:
```python
final_df.to_csv('../Resources/final_data.csv',index=False)
```

In [23]:
```python
# close connection
sqlite_connection.close()
```