Assignment02. Bigger data and communications

Load the NYC 311 Data as a JSON API

In [1]:
```
# To ignore unimporant system warnings
import warnings
warnings.filterwarnings("ignore")

# We will use Pandas, Numpy, and Matplotlib which is a package for visualization with P
import pandas as pd
import numpy as np

# Load a required package
# This is a library for accessing and parsing data through URLs
from urllib.parse import urlencode
import urllib.request, json
from bs4 import BeautifulSoup # for web scraping
import fiona
import plotly.express as px
import geopandas as gpd
import seaborn as sns # visualization styling package


# A magic functin that renders the figure in a notebook
%matplotlib inline
```

In [2]:
```
API = 'https://data.cityofnewyork.us/resource/erm2-nwe9.csv?'
# https://data.cityofnewyork.us/resource/erm2-nwe9.json
```

Query the Noise Data

In [3]:
```
query = {'$select': 'created_date, complaint_type, resolution_description, incident_zip
         '$where': 'complaint_type="Noise" OR complaint_type="Noise - Residential" OR c
         '$order': 'created_date',
         '$limit': 300000000}

df = pd.read_csv(API + urlencode(query),
                 parse_dates = ['created_date'],
                 dtype = {'incident_zip':'str'})
df.head()
```

Out[3]:

	created_date	complaint_type	resolution_description	incident_zip	borough	latitude	longitude
0	2010-01-01 00:03:46	Noise - Residential	The Police Department responded to the complai...	11230	BROOKLYN	40.612155	-73.955036
1	2010-01-01 00:08:02	Noise - Residential	The Police Department responded to the complai...	11220	BROOKLYN	40.635708	-74.006853

	created_date	complaint_type	resolution_description	incident_zip	borough	latitude	longitude
2	2010-01-01 00:08:29	Noise - Residential	The Police Department responded to the complai...	10036	MANHATTAN	40.759486	-73.98913
3	2010-01-01 00:08:30	Noise - Residential	The Police Department responded to the complai...	10303	STATEN ISLAND	40.634762	-74.16972
4	2010-01-01 00:15:08	Noise - Residential	The Police Department responded to the complai...	11355	QUEENS	40.755060	-73.83225

Convert the JSON data to Pandas DataFrame

In [4]:
```
df.shape[0]
```

Out[4]:
```
4802218
```

In [5]:
```
df.dtypes
```

Out[5]:
```
created_date              datetime64[ns]
complaint_type                    object
resolution_description            object
incident_zip                      object
borough                           object
latitude                         float64
longitude                        float64
dtype: object
```

In [6]:
```
df = df.rename(columns={"created_date": "Date", "complaint_type": "Complaint", "resolut
df["Complaint"] = df["Complaint"].str.replace("Noise -", "")
df.head()
```

Out[6]:

	Date	Complaint	Resolution	Zipcode	Borough	latitude	longitude
0	2010-01-01 00:03:46	Residential	The Police Department responded to the complai...	11230	BROOKLYN	40.612155	-73.955036
1	2010-01-01 00:08:02	Residential	The Police Department responded to the complai...	11220	BROOKLYN	40.635708	-74.006853
2	2010-01-01 00:08:29	Residential	The Police Department responded to the complai...	10036	MANHATTAN	40.759486	-73.989135
3	2010-01-01 00:08:30	Residential	The Police Department responded to the complai...	10303	STATEN ISLAND	40.634762	-74.169727
4	2010-01-01 00:15:08	Residential	The Police Department responded to the complai...	11355	QUEENS	40.755060	-73.832251

Drop Empty Values

In [7]:
```
df.Borough.unique()
```

Out[7]:
```
array(['BROOKLYN', 'MANHATTAN', 'STATEN ISLAND', 'QUEENS', 'BRONX',
       'Unspecified', nan], dtype=object)
```

In [8]:
```
df.Zipcode.unique()
```

Out[8]:
```
array(['11230', '11220', '10036', '10303', '11355', '10014', '10453',
       '10011', '10473', '11218', '11215', '10463', '10310', '11233',
       '10301', '11219', '11434', '11234', '10025', '11209', '10466',
       '11203', '10451', '10023', '11373', '11377', '10304', '10029',
       '11206', '11354', '11372', '11375', '11369', '10459', '10013',
       '10312', '11208', '11211', '11231', '11221', '10028', '10452',
       '10454', '11374', '10458', '11420', '10306', '10003', '10027',
       '10031', '10456', '11224', '11201', '11415', '10032', '11238',
       '11358', '11104', '10033', '11217', '11421', '10035', '10019',
       '10460', '11210', '10075', '11385', '11226', '11213', '11207',
       '11426', '11225', '11212', '10034', '10001', '11106', '10030',
       '10472', '11435', '11368', '10462', '11102', '11691', '10002',
       '11222', '10467', '10040', '10455', '11223', '10302', '11236',
       '10469', '11423', '10009', '10457', '11204', '10012', '11694',
       '11237', '11216', '11214', '10024', '11103', '11229', '11205',
       '11379', '10039', '11433', '11418', '11432', '11361', '10468',
       '11101', '10065', '11419', '10465', '10026', '11364', '10037',
       '11367', '10307', '11422', '11365', '11105', '10016', '10471',
       '11412', '11370', '10314', '10038', '10021', '10128', '11357',
       '11228', '10010', '10475', '11004', '11235', '10470', nan, '11411',
       '10069', '10309', '11413', '10461', '11378', '11429', '11416',
       '11427', '11436', '10022', '11428', '11232', '10305', '11693',
       '10005', '10017', '10018', '10006', '11360', '10280', '11109',
       '10474', '11249', '10464', '11414', '11239', '10007', '11417',
       '10308', '11356', '11362', '11692', '11366', '10004', '10281',
       '10020', '11001', '11363', '10000', '10044', '10282', '11005',
       '10119', '10162', '11697', '10048', '11040', '11430', '10153',
       '10118', '11371', '10041', '11243', '10278', '10106', '10103',
       '10112', '10174', '10129', '10107', '10111', '10803', '11242',
       '10271', '00083', '10169', '11241', '10275', '10123', '10279',
       '10121', '10177', '10045', '10120', '11359', '10170', '10178',
       '10583', '10165', '10158', '10176', '10173', '10110', '10122',
       '11096', '10105', '10155', '10171', '10172', '10168', '10167',
       '11695', '12345', '10154', '10152', '10115', '11251', '10166'],
      dtype=object)
```

In [9]:
```
df['Borough'].replace('', np.nan, inplace=True)
df['Borough'].replace('Unspecified', np.nan, inplace=True)
df.dropna(subset=['Borough'], inplace=True)
```

In [10]:
```
df['Zipcode'].replace('', np.nan, inplace=True)
df.dropna(subset=['Zipcode'], inplace=True)
```

In [11]:
```
df.shape[0]
```

Out[11]:
```
4789704
```

View the Number of Complaints by Zipcode

In [13]:
```
complaints_by_zip = df['Zipcode'].value_counts().to_frame('nc_freq')
complaints_by_zip = pd.DataFrame(complaints_by_zip).reset_index()
complaints_by_zip.columns = ['ZIPCODE', 'Complaint_Count']
complaints_by_zip.head()
```

Out[13]:

	ZIPCODE	Complaint_Count
0	10466	141795
1	10031	90140
2	11226	88550
3	10032	82163
4	10034	78926

Read Zipcode Geometry Shapefile as GeoDataFrame

In [14]:
```
gdf = gpd.read_file("G:/My Drive/GSAPP/1 Fall 21/Urban Informatics/Assignments/2/ZIP_CO
gdf.head()
```

Out[14]:

	ZIPCODE	BLDGZIP	PO_NAME	POPULATION	AREA	STATE	COUNTY	ST_FIPS	CTY_FIPS	
0	11436	0	Jamaica	18681.0	2.269930e+07	NY	Queens	36	081	ht
1	11213	0	Brooklyn	62426.0	2.963100e+07	NY	Kings	36	047	ht
2	11212	0	Brooklyn	83866.0	4.197210e+07	NY	Kings	36	047	ht
3	11225	0	Brooklyn	56527.0	2.369863e+07	NY	Kings	36	047	ht
4	11218	0	Brooklyn	72280.0	3.686880e+07	NY	Kings	36	047	ht

In [15]:
```
gdf.dtypes
```

Out[15]:

```
ZIPCODE          object
BLDGZIP          object
PO_NAME          object
POPULATION      float64
AREA            float64
STATE            object
COUNTY           object
ST_FIPS          object
CTY_FIPS         object
URL              object
SHAPE_AREA      float64
SHAPE_LEN       float64
geometry       geometry
dtype: object
```

In [16]:

```
gdf = gdf.merge(complaints_by_zip, on ='ZIPCODE')
gdf.head()
```

Out[16]:

	ZIPCODE	BLDGZIP	PO_NAME	POPULATION	AREA	STATE	COUNTY	ST_FIPS	CTY_FIPS	
0	11436	0	Jamaica	18681.0	2.269930e+07	NY	Queens	36	081	ht
1	11213	0	Brooklyn	62426.0	2.963100e+07	NY	Kings	36	047	ht
2	11212	0	Brooklyn	83866.0	4.197210e+07	NY	Kings	36	047	ht
3	11225	0	Brooklyn	56527.0	2.369863e+07	NY	Kings	36	047	ht
4	11218	0	Brooklyn	72280.0	3.686880e+07	NY	Kings	36	047	ht

In [19]:

```
print(gdf['Complaint_Count'].max())
print(gdf['Complaint_Count'].min())
```

```
141795
1
```

In [17]:

```
gdf.plot("Complaint_Count", legend=True)
```

Out[17]: <AxesSubplot:>