This project is part of the continuous assessment for the Master Course in Data Analytics with Python. The project was completed using jupyter notebook running on anaconda and python 3,

- I started by importing the dataset into jupyter notebook and view the head of the dataset

```python
#import libraries
import pandas as pd
from pandas import Series, DataFrame

#get the train data
df = pd.read_csv("file:/Users/online1/Documents/311_Service_Requests_from_2010_to_Present.csv")
df.head()
```

View the data information
```
df.info()
```

<table>
<thead>
<tr>
<th>Unique Key</th>
<th>Created Date</th>
<th>Closed Date</th>
<th>Agency</th>
<th>Agency Name</th>
<th>Complaint Type</th>
<th>Descriptor</th>
<th>Location Type</th>
<th>Incident Zip</th>
<th>Incident Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>12:01:2015</td>
<td>12:59:15 AM</td>
<td>NYPD</td>
<td>New York City Police Department</td>
<td>Noise - Street/Sidewalk</td>
<td>Loud Music/Party</td>
<td>Street/Sidewalk</td>
<td>10034.0</td>
<td>71 VERMONT AVENUE</td>
</tr>
<tr>
<td>1</td>
<td>12:01:2015</td>
<td>12:59:44 PM</td>
<td>NYPD</td>
<td>New York City Police Department</td>
<td>Blocked Driveway</td>
<td>No Access</td>
<td>Street/Sidewalk</td>
<td>11025.0</td>
<td>27-29 AVENUE</td>
</tr>
<tr>
<td>2</td>
<td>12:01:2015</td>
<td>12:39:29 AM</td>
<td>NYPD</td>
<td>New York City Police Department</td>
<td>Blocked Driveway</td>
<td>No Access</td>
<td>Street/Sidewalk</td>
<td>10458.0</td>
<td>889 VALENTINE AVENUE</td>
</tr>
<tr>
<td>3</td>
<td>12:01:2015</td>
<td>07:43:13 AM</td>
<td>NYPD</td>
<td>New York City Police Department</td>
<td>Illegal Parking</td>
<td>Commercial Overnight Parking</td>
<td>Street/Sidewalk</td>
<td>10401.0</td>
<td>2040 BADGELEY AVENUE</td>
</tr>
<tr>
<td>4</td>
<td>12:01:2015</td>
<td>01:56:58 PM</td>
<td>NYPD</td>
<td>New York City Police Department</td>
<td>Illegal Parking</td>
<td>Blocked Sidewalk</td>
<td>Street/Sidewalk</td>
<td>11373.0</td>
<td>67-14 57 ROAD</td>
</tr>
</tbody>
</table>
```

5 rows × 53 columns
import Matplotlib, seaborn and view the shape of the dataset

```
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import seaborn as sns
```

# - import Matplotlib, seaborn and view the shape of the dataset

```
#check the number of observations in the dataset
df.shape
```

- We see that our data set contains 364558 records with 53 rows, we are going to analysis and visualized the data using matplotlip

```
In [4]:
   import matplotlib.pyplot as plt
   import numpy as np
   import seaborn as sns

In [10]:
   #check the number of observations in the dataset
   df.shape
```

```
Out[10]:
(364558, 53)
```

Lets look at the description of the data and get some inside from the data.

df.describe()
We now look at the columns so we can know how they are called and be able to view the data.

```
In [7]: df.columns
Out[7]: Index(['Unique Key', 'Created Date', 'Closed Date', 'Agency', 'Agency Name', 'Complaint Type', 'Descriptor', 'Location Type', 'Incident Zip', 'Incident Address', 'Street Name', 'Cross Street 1', 'Cross Street 2', 'Intersection Street 1', 'Intersection Street 2', 'Address Type', 'City', 'Landmark', 'Facility Type', 'Status', 'Due Date', 'Resolution Description', 'Resolution Action Updated Date', 'Community Board', 'Borough', 'X Coordinate (State Plane)', 'Y Coordinate (State Plane)', 'Park Facility Name', 'Park Borough', 'School Name', 'School Number', 'School Region', 'School Code', 'School Phone Number', 'School Address', 'School City', 'School State', 'School Zip', 'School Not Found', 'School or Citywide Complaint', 'Vehicle Type', 'Taxi Company Borough', 'Taxi Pick Up Location', 'Bridge Highway Name', 'Bridge Highway Direction', 'Road Ramp', 'Bridge Highway Segment', 'Garage Lot Name', 'Ferry Direction', 'Ferry Terminal Name', 'Latitude', 'Longitude', 'Location'],
       dtype='object')
```

#complaint type breakdown with bars plot to figure out the top 10 complaints
```
df['Complaint Type'].value_counts().plot(kind='barh', alpha=0.6, figsize=(12,15))
plt.show()
```

#top complaint by City
Now we count the City with the highest complaint

df['City'].value_counts()

#complaint by city
df['City'].value_counts().plot(kind='bar', alpha=0.6, figsize=(20,15))
plt.show()
# Display complaint type and city together

df[['Complaint Type', 'City']].head(50)
In [44]:
# display complaint type and city together
df[['Complaint Type', 'City']].head(50)

Out[44]:

<table>
<thead>
<tr>
<th>Complaint Type</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Noise - Street/Sidewalk</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>1 Blocked Driveway</td>
<td>ASTORIA</td>
</tr>
<tr>
<td>2 Blocked Driveway</td>
<td>BRONX</td>
</tr>
<tr>
<td>3 Illegal Parking</td>
<td>BRONX</td>
</tr>
<tr>
<td>4 Illegal Parking</td>
<td>ELMHURST</td>
</tr>
<tr>
<td>5 Illegal Parking</td>
<td>BROOKLYN</td>
</tr>
<tr>
<td>6 Illegal Parking</td>
<td>NEW YORK</td>
</tr>
<tr>
<td>7 Blocked Driveway</td>
<td>BRONX</td>
</tr>
<tr>
<td>8 Illegal Parking</td>
<td>KEW GARDENS</td>
</tr>
<tr>
<td>9 Blocked Driveway</td>
<td>BROOKLYN</td>
</tr>
<tr>
<td>10 Blocked Driveway</td>
<td>JACKSON HEIGHTS</td>
</tr>
<tr>
<td>11 Blocked Driveway</td>
<td>BRONX</td>
</tr>
<tr>
<td>12 Noise - Street/Sidewalk</td>
<td>BRONX</td>
</tr>
<tr>
<td>13 Illegal Parking</td>
<td>BROOKLYN</td>
</tr>
<tr>
<td>14 Derelict Vehicle</td>
<td>MIDDLE VILLAGE</td>
</tr>
<tr>
<td>15 Blocked Driveway</td>
<td>REGO PARK</td>
</tr>
<tr>
<td>16 Blocked Driveway</td>
<td>SAINT ALBANS</td>
</tr>
<tr>
<td>17 Noise - Commercial</td>
<td>BROOKLYN</td>
</tr>
</tbody>
</table>

In [21]:
# group data by complaint to display plot against city
groupByComplainType = df.groupby('Complaint Type')

In [23]:
grp_data = groupByComplainType.get_group('Blocked Driveway')
grp_data.shape

Out[23]:
(100881, 51)

In [24]:
# to get nan values in the entire dataset
df.isnull().sum()

Out[24]:
Unique key               0
Created Date             0
Closed Date              2381
Agency                   0
Agency Name              0
Complaint Type           0
Descriptor               6501
Location Type            133
Incident Zip             2898
Incident Address         51699
Street Name              51699
Cross Street 1           57188
Cross Street 2           57805
In [25]: # fill blank values in the City column
df['City'].fillna('unknown city', inplace=True)

In [26]: # shape after dropping nan values
df['City'].shape

Out[26]: (361561,)

In [27]: # count of null values in grouped city column data
grp_data['City'].isnull().sum()

Out[27]: 3569

In [28]: # fix those NaN with 'unknown city' value instead
grp_data['City'].fillna('unknown city', inplace=True)

/users/online/anaconda3/lib/python3.6/site-packages/pandas/core/generics.py:4355: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy

self._update_inplace(new_data)

In [29]: grpd_city.head()
```python
In [32]: # find the top 10 major complaint type and their counts
groupByComplainType('Complaint Type')['value_counts'].nlargest(10)

Out[32]:
Complaint Type     Complaint Type
Blocked Driveway    Blocked Driveway  108881
Illegal Parking     Illegal Parking    92679
Noise - Street/Sidewalk Noise - Street/Sidewalk  51692
Noise - Commercial  Noise - Commercial     44109
Derelict Vehicle    Derelict Vehicle    21661
Noise - Vehicle     Noise - Vehicle     19352
Animal Abuse        Animal Abuse        10541
Traffic             Traffic             5198
Homeless Encampment Homeless Encampment     4879
Vending             Vending             4192
Name: Complaint Type, dtype: int64

In [49]: # fix Location type those NaN with 'unknown location' value instead
df['Location Type'].fillna('Unknown Loc', inplace = True)

In [50]: df['Location Type'].values

Out[50]: array(['Street/Sidewalk', 'Street/Sidewalk', 'Street/Sidewalk', ..., 
              'Street/Sidewalk', 'Street/Sidewalk', 'Street/Sidewalk'],
             dtype='object')

In [51]: # count of null values in grouped location type column data
grp_data['Location Type'].isnull().sum()

Out[51]: 37

In [52]: # plot Major complaint type against location type to check for any patterns
plt.figure(figsize=(3,3))
plt.scatter(grp_data['Complaint Type'], grp_data['Location Type'])
plt.title='Plot complaint type against location type'
plt.xlabel='Complaint Type'
plt.ylabel='Location Type'
plt.show()
```