Project_Customer Service Requests Analysis

November 7, 2022

[1]:
```python
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```python
import pandas as pd
import numpy as np
```

[3]:
```python
data = pd.read_csv('311_Service_Requests_from_2010.csv')
```

[4]:
```python
data.head()
```

[4]:
```
   Unique Key            Created Date            Closed Date Agency  \
0    32310363  12/31/2015 11:59:45 PM  01/01/2016 12:55:15 AM   NYPD
1    32309934  12/31/2015 11:59:44 PM  01/01/2016 01:26:57 AM   NYPD
2    32309159  12/31/2015 11:59:29 PM  01/01/2016 04:51:03 AM   NYPD
3    32305098  12/31/2015 11:57:46 PM  01/01/2016 07:43:13 AM   NYPD
4    32306529  12/31/2015 11:56:58 PM  01/01/2016 03:24:42 AM   NYPD

                       Agency Name           Complaint Type  \
0  New York City Police Department  Noise - Street/Sidewalk
1  New York City Police Department         Blocked Driveway
2  New York City Police Department         Blocked Driveway
3  New York City Police Department           Illegal Parking
4  New York City Police Department           Illegal Parking

                    Descriptor    Location Type  Incident Zip  \
0             Loud Music/Party  Street/Sidewalk       10034.0
1                    No Access  Street/Sidewalk       11105.0
2                    No Access  Street/Sidewalk       10458.0
3  Commercial Overnight Parking  Street/Sidewalk      10461.0
4             Blocked Sidewalk  Street/Sidewalk       11373.0

        Incident Address  … Bridge Highway Name Bridge Highway Direction  \
0    71 VERMILYEA AVENUE  …                 NaN                      NaN
1        27-07 23 AVENUE  …                 NaN                      NaN
2  2897 VALENTINE AVENUE  …                 NaN                      NaN
3    2940 BAISLEY AVENUE  …                 NaN                      NaN
4          87-14 57 ROAD  …                 NaN                      NaN
```

```
  Road Ramp Bridge Highway Segment Garage Lot Name Ferry Direction  \
0       NaN                    NaN             NaN             NaN
1       NaN                    NaN             NaN             NaN
2       NaN                    NaN             NaN             NaN
3       NaN                    NaN             NaN             NaN
4       NaN                    NaN             NaN             NaN

  Ferry Terminal Name   Latitude  Longitude  \
0                 NaN  40.865682 -73.923501
1                 NaN  40.775945 -73.915094
2                 NaN  40.870325 -73.888525
3                 NaN  40.835994 -73.828379
4                 NaN  40.733060 -73.874170

                                  Location
0   (40.86568153633767, -73.92350095571744)
1  (40.775945312321085, -73.91509393898605)
2  (40.870324522111424, -73.88852464418646)
3   (40.83599404683083, -73.82837939584206)
4  (40.733059618956815, -73.87416975810375)

[5 rows x 53 columns]
```

[5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 364558 entries, 0 to 364557
Data columns (total 53 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   Unique Key                      364558 non-null  int64
 1   Created Date                    364558 non-null  object
 2   Closed Date                     362177 non-null  object
 3   Agency                          364558 non-null  object
 4   Agency Name                     364558 non-null  object
 5   Complaint Type                  364558 non-null  object
 6   Descriptor                      358057 non-null  object
 7   Location Type                   364425 non-null  object
 8   Incident Zip                    361560 non-null  float64
 9   Incident Address                312859 non-null  object
 10  Street Name                     312859 non-null  object
 11  Cross Street 1                  307370 non-null  object
 12  Cross Street 2                  306753 non-null  object
 13  Intersection Street 1           51120 non-null   object
 14  Intersection Street 2           50512 non-null   object
 15  Address Type                    361306 non-null  object
```

```
 16  City                            361561 non-null  object
 17  Landmark                        375 non-null     object
 18  Facility Type                   362169 non-null  object
 19  Status                          364558 non-null  object
 20  Due Date                        364555 non-null  object
 21  Resolution Description          364558 non-null  object
 22  Resolution Action Updated Date  362156 non-null  object
 23  Community Board                 364558 non-null  object
 24  Borough                         364558 non-null  object
 25  X Coordinate (State Plane)      360528 non-null  float64
 26  Y Coordinate (State Plane)      360528 non-null  float64
 27  Park Facility Name              364558 non-null  object
 28  Park Borough                    364558 non-null  object
 29  School Name                     364558 non-null  object
 30  School Number                   364558 non-null  object
 31  School Region                   364557 non-null  object
 32  School Code                     364557 non-null  object
 33  School Phone Number             364558 non-null  object
 34  School Address                  364558 non-null  object
 35  School City                     364558 non-null  object
 36  School State                    364558 non-null  object
 37  School Zip                      364557 non-null  object
 38  School Not Found                364558 non-null  object
 39  School or Citywide Complaint    0 non-null       float64
 40  Vehicle Type                    0 non-null       float64
 41  Taxi Company Borough            0 non-null       float64
 42  Taxi Pick Up Location           0 non-null       float64
 43  Bridge Highway Name             297 non-null     object
 44  Bridge Highway Direction        297 non-null     object
 45  Road Ramp                       262 non-null     object
 46  Bridge Highway Segment          262 non-null     object
 47  Garage Lot Name                 0 non-null       float64
 48  Ferry Direction                 1 non-null       object
 49  Ferry Terminal Name             2 non-null       object
 50  Latitude                        360528 non-null  float64
 51  Longitude                       360528 non-null  float64
 52  Location                        360528 non-null  object
dtypes: float64(10), int64(1), object(42)
memory usage: 147.4+ MB
```

1 1. Understand the dataset:

```
1. Identify the shape of the dataset

2. Identify variables with null values
```

[6]:
```
# Identify the shape of the dataset
data.shape
```

[6]:
```
(364558, 53)
```

[7]:
```
data.shape[0]
```

[7]:
```
364558
```

[8]:
```
# Identify variables with null values
data.isna().sum()
```

[8]:
```
Unique Key                              0
Created Date                            0
Closed Date                          2381
Agency                                  0
Agency Name                             0
Complaint Type                          0
Descriptor                           6501
Location Type                         133
Incident Zip                         2998
Incident Address                    51699
Street Name                         51699
Cross Street 1                      57188
Cross Street 2                      57805
Intersection Street 1              313438
Intersection Street 2              314046
Address Type                         3252
City                                 2997
Landmark                           364183
Facility Type                        2389
Status                                  0
Due Date                                3
Resolution Description                  0
Resolution Action Updated Date       2402
Community Board                         0
Borough                                 0
X Coordinate (State Plane)           4030
Y Coordinate (State Plane)           4030
Park Facility Name                      0
Park Borough                            0
School Name                             0
School Number                           0
School Region                           1
School Code                             1
School Phone Number                     0
School Address                          0
```

```
School City                           0
School State                          0
School Zip                            1
School Not Found                      0
School or Citywide Complaint     364558
Vehicle Type                     364558
Taxi Company Borough             364558
Taxi Pick Up Location            364558
Bridge Highway Name              364261
Bridge Highway Direction         364261
Road Ramp                        364296
Bridge Highway Segment           364296
Garage Lot Name                  364558
Ferry Direction                  364557
Ferry Terminal Name              364556
Latitude                           4030
Longitude                          4030
Location                           4030
dtype: int64
```

[9]:
```
data.isna().sum()/data.shape[0]
```

[9]:
```
Unique Key                          0.000000
Created Date                        0.000000
Closed Date                         0.006531
Agency                              0.000000
Agency Name                         0.000000
Complaint Type                      0.000000
Descriptor                          0.017833
Location Type                       0.000365
Incident Zip                        0.008224
Incident Address                    0.141813
Street Name                         0.141813
Cross Street 1                      0.156869
Cross Street 2                      0.158562
Intersection Street 1               0.859775
Intersection Street 2               0.861443
Address Type                        0.008920
City                                0.008221
Landmark                            0.998971
Facility Type                       0.006553
Status                              0.000000
Due Date                            0.000008
Resolution Description              0.000000
Resolution Action Updated Date      0.006589
Community Board                     0.000000
Borough                             0.000000
```

```
X Coordinate (State Plane)        0.011054
Y Coordinate (State Plane)        0.011054
Park Facility Name                0.000000
Park Borough                      0.000000
School Name                       0.000000
School Number                     0.000000
School Region                     0.000003
School Code                       0.000003
School Phone Number               0.000000
School Address                    0.000000
School City                       0.000000
School State                      0.000000
School Zip                        0.000003
School Not Found                  0.000000
School or Citywide Complaint      1.000000
Vehicle Type                      1.000000
Taxi Company Borough              1.000000
Taxi Pick Up Location             1.000000
Bridge Highway Name               0.999185
Bridge Highway Direction          0.999185
Road Ramp                         0.999281
Bridge Highway Segment            0.999281
Garage Lot Name                   1.000000
Ferry Direction                   0.999997
Ferry Terminal Name               0.999995
Latitude                          0.011054
Longitude                         0.011054
Location                          0.011054
dtype: float64
```

2 2. Perform basic data exploratory analysis:

1. Utilize missing value treatment

2. Analyze the date column and remove the entries if it has an incorrect timeline

3. Draw a frequency plot for city-wise complaints

4. Draw scatter and hexbin plots for complaint concentration across Brooklyn

[10]:
```
# data=data.drop(columns=['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','Unique␣
↪Key','Agency','Vehicle Type','Taxi Company Borough',
```

```
#                         'Taxi Pick Up Location','Garage Lot Name','Ferry␣
↪Direction','Ferry Terminal Name'],axis=1)
# data.head()
```

[11]:
```
# 1. Utilize missing value treatment

data_imputed = data.fillna(0)
round((data_imputed.isna().sum() / data_imputed.shape[0])*100)
```

[11]:
```
Unique Key                        0.0
Created Date                      0.0
Closed Date                       0.0
Agency                            0.0
Agency Name                       0.0
Complaint Type                    0.0
Descriptor                        0.0
Location Type                     0.0
Incident Zip                      0.0
Incident Address                  0.0
Street Name                       0.0
Cross Street 1                    0.0
Cross Street 2                    0.0
Intersection Street 1             0.0
Intersection Street 2             0.0
Address Type                      0.0
City                              0.0
Landmark                          0.0
Facility Type                     0.0
Status                            0.0
Due Date                          0.0
Resolution Description            0.0
Resolution Action Updated Date    0.0
Community Board                   0.0
Borough                           0.0
X Coordinate (State Plane)        0.0
Y Coordinate (State Plane)        0.0
Park Facility Name                0.0
Park Borough                      0.0
School Name                       0.0
School Number                     0.0
School Region                     0.0
School Code                       0.0
School Phone Number               0.0
School Address                    0.0
School City                       0.0
School State                      0.0
School Zip                        0.0
```

```
School Not Found                  0.0
School or Citywide Complaint      0.0
Vehicle Type                      0.0
Taxi Company Borough              0.0
Taxi Pick Up Location             0.0
Bridge Highway Name               0.0
Bridge Highway Direction          0.0
Road Ramp                         0.0
Bridge Highway Segment            0.0
Garage Lot Name                   0.0
Ferry Direction                   0.0
Ferry Terminal Name               0.0
Latitude                          0.0
Longitude                         0.0
Location                          0.0
dtype: float64
```

[12]:
```
# 2. Analyze the date column and remove the entries if it has an incorrect␣
↪timeline

data['Created Date'] = pd.to_datetime(data['Created Date'])
data['Closed Date'] = pd.to_datetime(data['Closed Date'])
data['Due Date'] = pd.to_datetime(data['Due Date'])
data_date = data[['Created Date','Closed Date']]
```

[13]:
```
data_date.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 364558 entries, 0 to 364557
Data columns (total 2 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   Created Date  364558 non-null  datetime64[ns]
 1   Closed Date   362177 non-null  datetime64[ns]
dtypes: datetime64[ns](2)
memory usage: 5.6 MB
```

[14]:
```
data_date.head()
```

[14]:
```
         Created Date         Closed Date
0 2015-12-31 23:59:45 2016-01-01 00:55:15
1 2015-12-31 23:59:44 2016-01-01 01:26:57
2 2015-12-31 23:59:29 2016-01-01 04:51:03
3 2015-12-31 23:57:46 2016-01-01 07:43:13
4 2015-12-31 23:56:58 2016-01-01 03:24:42
```

[15]:

```python
# 3. Draw a frequency plot for city-wise complaints
data_city = data['City'].value_counts()
data_city = data_city.reset_index()
data_city = data_city.rename(columns={'index':'City','City':'counts'})
data_city["Percentage"]=np.around((data_city.counts/data_city.counts.
 ↪sum())*100,decimals=2)
data_city
```

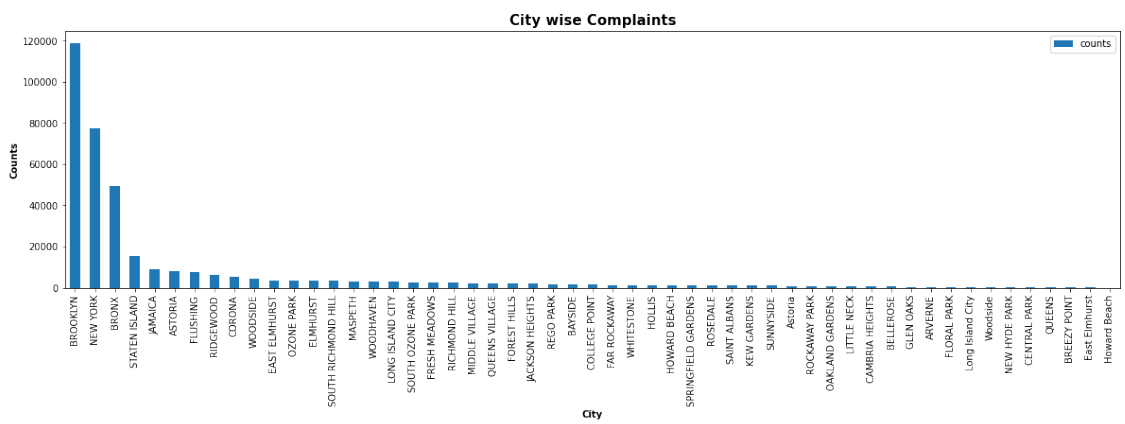
[15]:

	City	counts	Percentage
0	BROOKLYN	118862	32.87
1	NEW YORK	77312	21.38
2	BRONX	49171	13.60
3	STATEN ISLAND	15340	4.24
4	JAMAICA	8932	2.47
5	ASTORIA	7991	2.21
6	FLUSHING	7487	2.07
7	RIDGEWOOD	6392	1.77
8	CORONA	5383	1.49
9	WOODSIDE	4357	1.21
10	EAST ELMHURST	3558	0.98
11	OZONE PARK	3446	0.95
12	ELMHURST	3438	0.95
13	SOUTH RICHMOND HILL	3431	0.95
14	MASPETH	3118	0.86
15	WOODHAVEN	3103	0.86
16	LONG ISLAND CITY	3028	0.84
17	SOUTH OZONE PARK	2668	0.74
18	FRESH MEADOWS	2453	0.68
19	RICHMOND HILL	2335	0.65
20	MIDDLE VILLAGE	2291	0.63
21	QUEENS VILLAGE	2251	0.62
22	FOREST HILLS	2122	0.59
23	JACKSON HEIGHTS	2106	0.58
24	REGO PARK	1807	0.50
25	BAYSIDE	1550	0.43
26	COLLEGE POINT	1544	0.43
27	FAR ROCKAWAY	1397	0.39
28	WHITESTONE	1369	0.38
29	HOLLIS	1231	0.34
30	HOWARD BEACH	1144	0.32
31	SPRINGFIELD GARDENS	1094	0.30
32	ROSEDALE	1091	0.30
33	SAINT ALBANS	1047	0.29
34	KEW GARDENS	1008	0.28
35	SUNNYSIDE	944	0.26
36	Astoria	906	0.25
37	ROCKAWAY PARK	831	0.23

38	OAKLAND GARDENS	717	0.20
39	LITTLE NECK	712	0.20
40	CAMBRIA HEIGHTS	617	0.17
41	BELLEROSE	487	0.13
42	GLEN OAKS	361	0.10
43	ARVERNE	259	0.07
44	FLORAL PARK	196	0.05
45	Long Island City	170	0.05
46	Woodside	166	0.05
47	NEW HYDE PARK	129	0.04
48	CENTRAL PARK	110	0.03
49	QUEENS	37	0.01
50	BREEZY POINT	31	0.01
51	East Elmhurst	30	0.01
52	Howard Beach	1	0.00

[16]:
```
import matplotlib.pyplot as plt
```

[17]:
```
data_city.plot(x='City',y='counts',kind='bar',figsize=(20,5))
plt.title('City wise Complaints',fontsize=15,weight='bold')
plt.xlabel('City',weight='bold')
plt.ylabel('Counts',weight='bold')
plt.show()
```



[18]:
```
# 4. Draw scatter and hexbin plots for complaint concentration across Brooklyn
data['Borough'].value_counts()
```

[18]:

BROOKLYN	118864
QUEENS	100766
MANHATTAN	77462
BRONX	49169
STATEN ISLAND	15339

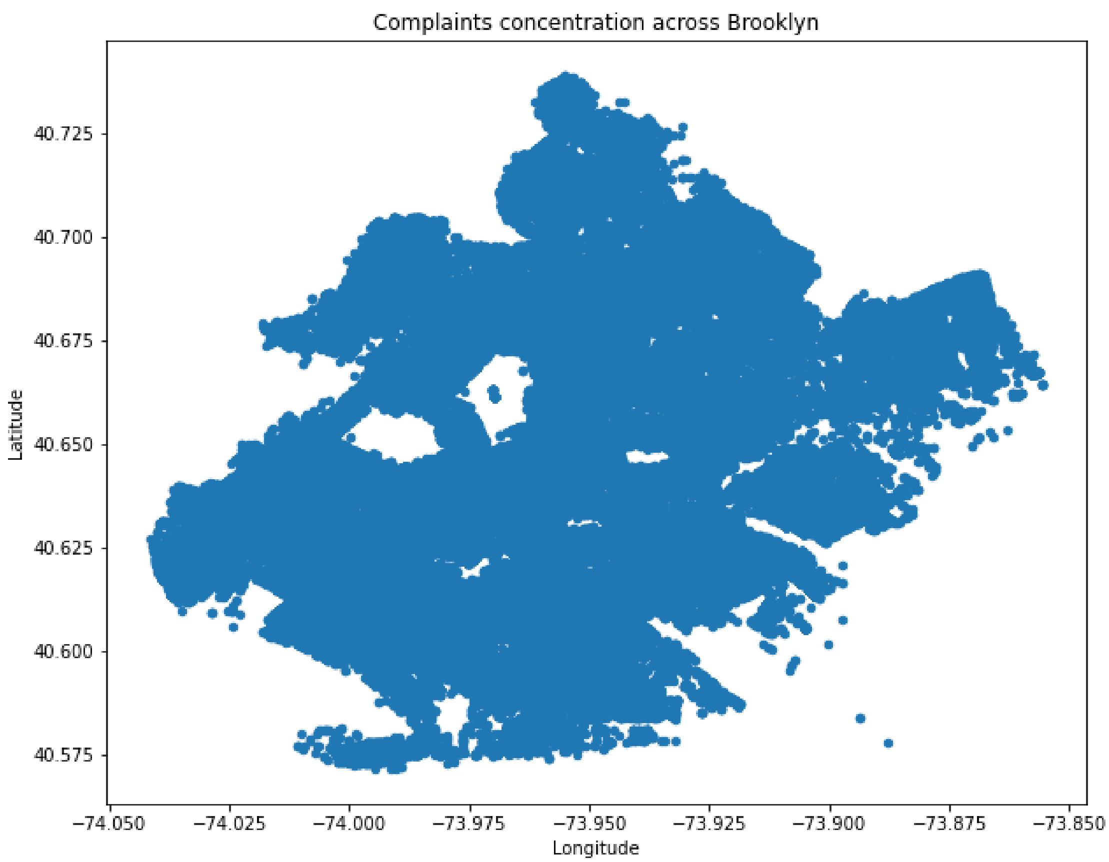
```
Unspecified        2958
Name: Borough, dtype: int64
```

[19]:
```python
data_brooklyn = data[data['Borough'] == 'BROOKLYN']
data_brooklyn.shape
```

[19]:
```
(118864, 53)
```

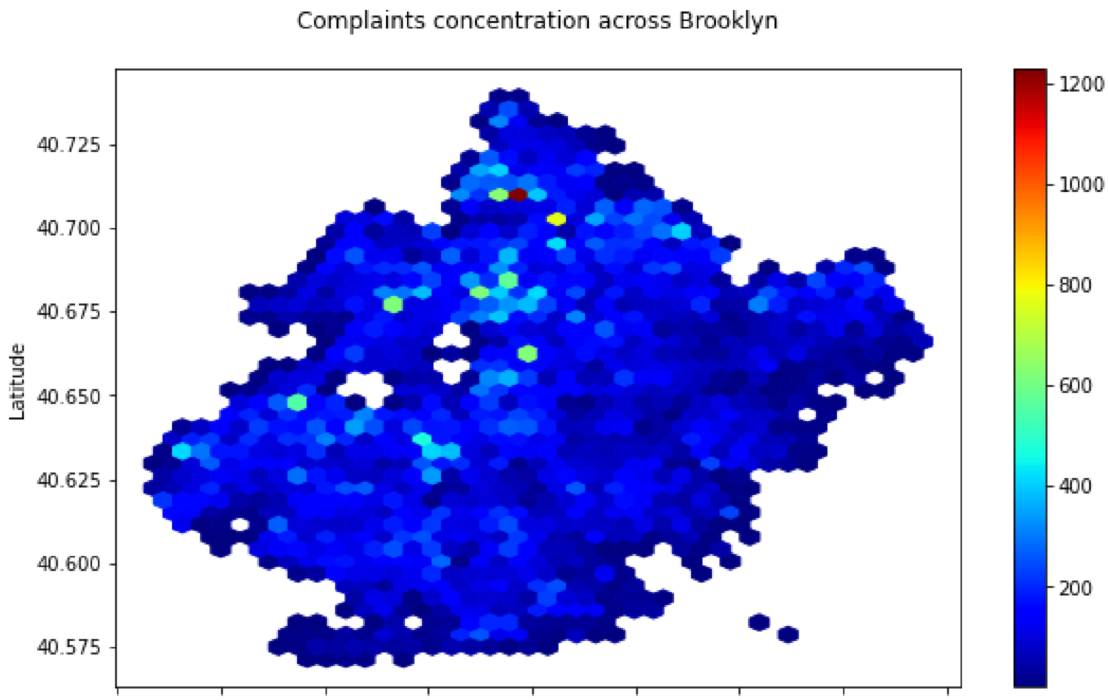
[20]:
```python
data_brooklyn[['Longitude', 'Latitude']].plot(kind='scatter',x='Longitude',␣
↪y='Latitude', figsize=(10,8),
                                              title = 'Complaints concentration␣
↪across Brooklyn')
```

[20]:
```
<AxesSubplot:title={'center':'Complaints concentration across Brooklyn'},
xlabel='Longitude', ylabel='Latitude'>
```



[21]:
```python
data_brooklyn.plot(kind='hexbin', x='Longitude', y='Latitude', gridsize=40,␣
↪colormap = 'jet',mincnt=1,
                   title = 'Complaints concentration across Brooklyn\n',␣
↪figsize=(10,6))
```

[21]: <AxesSubplot:title={'center':'Complaints concentration across Brooklyn\n'}, xlabel='Longitude', ylabel='Latitude'>



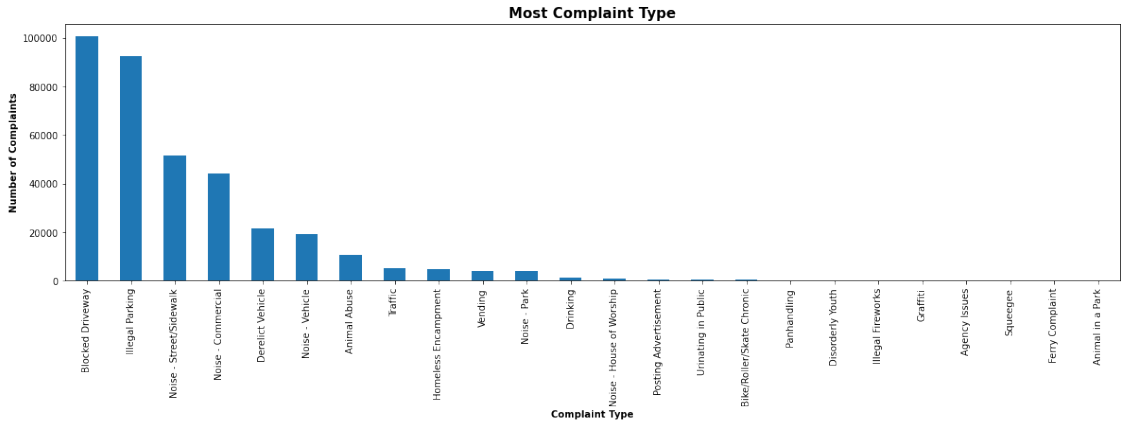
3 3. Find major types of complaints:

1. Plot a bar graph of count vs. complaint types

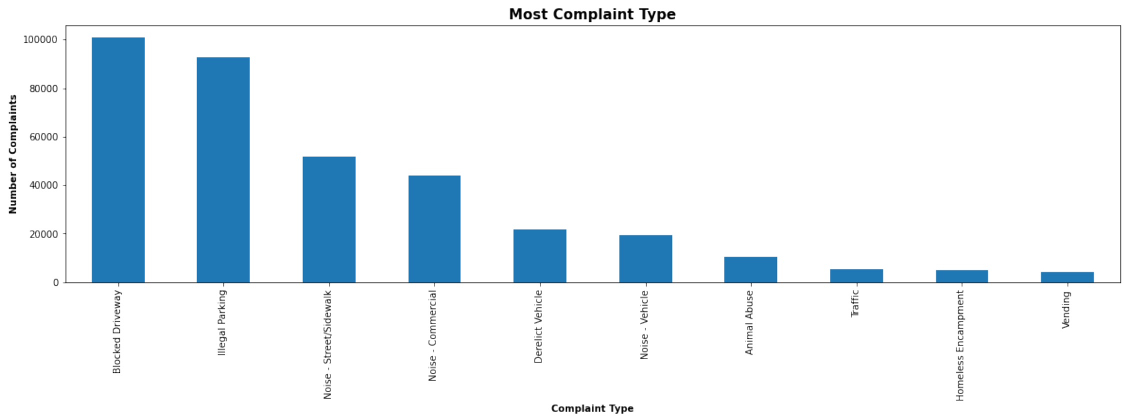
2. Find the top 10 types of complaints

3. Display the types of complaints in each city in a separate dataset

[22]:
```
# 1. Plot a bar graph of count vs. complaint types
data_complaint = data['Complaint Type'].value_counts().
↪plot(kind='bar',figsize=(20,5))
plt.xlabel('Complaint Type',weight='bold',fontsize=10)
plt.ylabel('Number of Complaints',weight='bold',fontsize=10)
plt.title('Most Complaint Type',weight='bold',fontsize=15)
plt.show()
```



[23]:
```python
# 2. Find the top 10 types of complaints
data_complaint = data['Complaint Type'].value_counts().head(10).
↪plot(kind='bar',figsize=(20,5))
plt.xlabel('Complaint Type',weight='bold',fontsize=10)
plt.ylabel('Number of Complaints',weight='bold',fontsize=10)
plt.title('Most Complaint Type',weight='bold',fontsize=15)
plt.show()
```



[24]:
```python
# 3. Display the types of complaints in each city in a separate dataset
city_complaint = data.groupby(['Complaint Type','City'],as_index=False)['City'].
↪size()
city_complaint.rename(columns={'size':'Counts'})
```

[24]:

	Complaint Type	City	Counts
0	Animal Abuse	ARVERNE	46
1	Animal Abuse	ASTORIA	170
2	Animal Abuse	BAYSIDE	53

```
3      Animal Abuse      BELLEROSE      15
4      Animal Abuse   BREEZY POINT       2
..              …              …      …
772         Vending  STATEN ISLAND      25
773         Vending      SUNNYSIDE      15
774         Vending     WHITESTONE       1
775         Vending      WOODHAVEN       6
776         Vending       WOODSIDE      15

[777 rows x 3 columns]
```

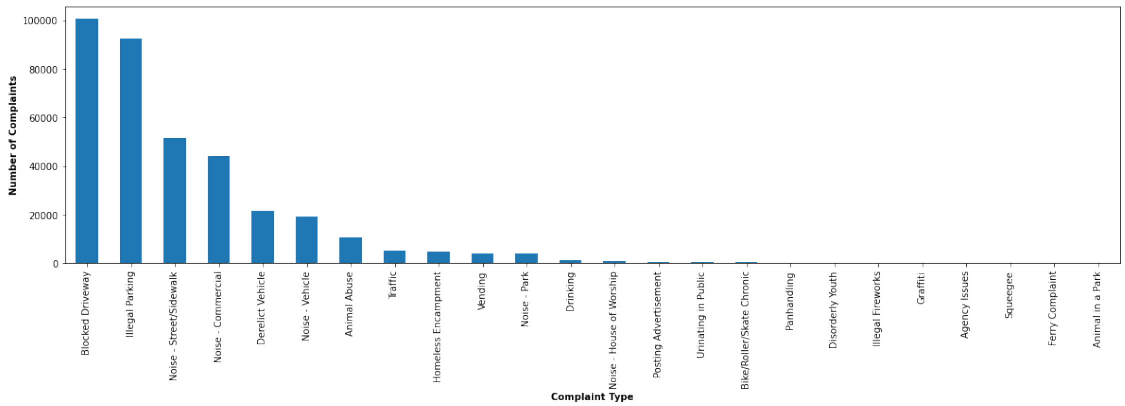
4 4. Visualize the major types of complaints in each city

[25]:
```
complaint_city = data.groupby(['City','Complaint␣
↪Type'],as_index=False)['Complaint Type'].size()
complaint_city.rename(columns={'size':'Counts'})
```

[25]:
```
          City             Complaint Type  Counts
0      ARVERNE               Animal Abuse      46
1      ARVERNE           Blocked Driveway      50
2      ARVERNE           Derelict Vehicle      32
3      ARVERNE           Disorderly Youth       2
4      ARVERNE                   Drinking       1
..         …                          …      …
772   Woodside           Blocked Driveway      27
773   Woodside           Derelict Vehicle       8
774   Woodside            Illegal Parking     124
775   Woodside         Noise - Commercial       2
776   Woodside  Noise - Street/Sidewalk       5

[777 rows x 3 columns]
```

[26]:
```
data['Complaint Type'].value_counts().plot(kind='bar',figsize=(20,5))
plt.xlabel('Complaint Type',weight='bold',fontsize=10)
plt.ylabel('Number of Complaints',weight='bold',fontsize=10)
plt.show()
```



5 5. Check if the average response time across various types of complaints

[27]:
```
data_date['Request_Response_Time'] = data['Closed Date'] - data['Created Date']
data_date['Request_Response_Time'].head()
```

[27]:
```
0    0 days 00:55:30
1    0 days 01:27:13
2    0 days 04:51:34
3    0 days 07:45:27
4    0 days 03:27:44
Name: Request_Response_Time, dtype: timedelta64[ns]
```

[28]:
```
data_date['Request_Response_Time'].describe()
```

[28]:
```
count                       362177
mean     0 days 04:11:53.299632500
std      0 days 05:51:42.547519569
min                0 days 00:01:01
25%                0 days 01:15:33
50%                0 days 02:40:16
75%                0 days 05:14:38
max               24 days 16:52:22
Name: Request_Response_Time, dtype: object
```

- 04 hrs 11 min 53 sec average response time across various types of complaints

6 6. Identify significant variables by performing a statistical analysis using p-values and chi-square values

[29]:
```
from scipy import stats
```

[30]:
```
# help(stats.chi2_contingency)
```

[31]:
```
data_cross = pd.crosstab(data['Complaint Type'],data['City'])
```

[32]:
```
coff,pval,dof,expec=stats.chi2_contingency(data_cross)
print("chisquare",coff)
print("Pvalue",pval)
print("DOF",dof)
print("Expected",expec)
```

```
chisquare 141373.60935271924
Pvalue 0.0
DOF 1092
Expected [[7.54232619e+00 2.32705516e+02 2.63835812e+01 … 9.03623095e+01
  1.26879982e+02 4.83407779e+00]
 [7.16338322e-04 2.21013881e-02 2.50580123e-03 … 8.58223094e-03
  1.20505254e-02 4.59120314e-04]
 [3.38828026e-01 1.04539566e+01 1.18524398e+00 … 4.05939523e+00
  5.69989850e+00 2.17163909e-01]
 …
 [3.72137758e+00 1.14816711e+02 1.30176374e+01 … 4.45846897e+01
  6.26024792e+01 2.38513003e+00]
 [4.59172864e-01 1.41669898e+01 1.60621859e+00 … 5.50121003e+00
  7.72438676e+00 2.94296122e-01]
 [2.99787588e+00 9.24943094e+01 1.04867782e+01 … 3.59166365e+01
  5.04314486e+01 1.92141852e+00]]
```

[33]:
```
if pval<0.05:
    print("Alter Hypo----->relation exist")
else:
    print("Null Hypo---->No relation")
```

```
Alter Hypo----->relation exist
```

[ ]:
```
```