

```
In [2]: #Importing Required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

```
In [3]: #Importing the data file and reading it
df_nyc = pd.read_csv('311_Service_Requests.csv')
```

C:\Users\HP\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3444: DtypeWarning: Columns (48,49) have mixed types.Specify dtype option on import or set low\_memory=False.  
exec(code\_obj, self.user\_global\_ns, self.user\_ns)

```
In [14]: df_nyc.shape() #no of rows and columns
```

```
Out[14]: (300698, 53)
```

```
In [17]: df_nyc.head()
```

```
Out[17]:
```

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction
0	32310363	12/31/2015 23:59	1/1/2016 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	...	NaN	NaN
1	32309934	12/31/2015 23:59	1/1/2016 1:26	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	...	NaN	NaN
2	32309159	12/31/2015 23:59	1/1/2016 4:51	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	...	NaN	NaN
3	32305098	12/31/2015 23:57	1/1/2016 7:43	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	...	NaN	NaN

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction
4	32306529	12/31/2015 23:56	1/1/2016 3:24	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	...	NaN	NaN

5 rows × 53 columns



## DATA CLEANING AND PROCESSING

```
In [15]: #Changing Created Date and Closed Date to datetime
import datetime
df_nyc = pd.read_csv('311_Service_Requests.csv', parse_dates=['Created Date', 'Closed Date'], low_memory=False)
```

```
In [16]: #Creating column Request_closing_time
df_nyc['Request Closing Time'] = df_nyc['Closed Date'] - df_nyc['Created Date']
```

```
In [17]: df_nyc['Request Closing Time Sec'] = df_nyc['Request Closing Time'].apply(lambda x : x.seconds)
df_nyc['Request Closing Time Sec'] = df_nyc['Request Closing Time Sec']
df_nyc = df_nyc[df_nyc['Request Closing Time Sec'].notnull()]
```

```
In [186... df_nyc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 298534 entries, 0 to 300697
Data columns (total 41 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Unique Key                           298534 non-null  int64
1   Created Date                          298534 non-null  datetime64[ns]
2   Closed Date                           298534 non-null  datetime64[ns]
3   Agency                                298534 non-null  object
4   Agency Name                           298534 non-null  object
5   Complaint Type                         298534 non-null  object
```

```

6  Descriptor                292625 non-null object
7  Location Type             298406 non-null object
8  Incident Zip              298027 non-null float64
9  Incident Address          254137 non-null object
10 Street Name               254137 non-null object
11 Cross Street 1            250971 non-null object
12 Cross Street 2            250881 non-null object
13 Address Type              297827 non-null object
14 City                      298028 non-null object
15 Facility Type             298519 non-null object
16 Status                    298534 non-null object
17 Due Date                  298533 non-null object
18 Resolution Description     298534 non-null object
19 Resolution Action Updated Date 298495 non-null object
20 Community Board           298534 non-null object
21 Borough                   298534 non-null object
22 X Coordinate (State Plane) 297102 non-null float64
23 Y Coordinate (State Plane) 297102 non-null float64
24 Park Facility Name        298534 non-null object
25 Park Borough              298534 non-null object
26 School Name               298534 non-null object
27 School Number             298534 non-null object
28 School Region             298533 non-null object
29 School Code               298533 non-null object
30 School Phone Number       298534 non-null object
31 School Address            298534 non-null object
32 School City               298534 non-null object
33 School State              298534 non-null object
34 School Zip                298533 non-null object
35 School Not Found          298534 non-null object
36 Latitude                  297102 non-null float64
37 Longitude                 297102 non-null float64
38 Location                  297102 non-null object
39 Request Closing Time       298534 non-null timedelta64[ns]
40 Request Closing Time Sec   298534 non-null float64
dtypes: datetime64[ns](2), float64(6), int64(1), object(31), timedelta64[ns](1)
memory usage: 95.7+ MB

```

```
In [32]: df_nyc.columns
```

```
Out[32]: 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',
'Request Closing Time'],
dtype='object')
```

```
In [245... df_nyc['Descriptor'].value_counts().head()
```

```
Out[245... Loud Music/Party          60829
No Access                  56822
Posted Parking Sign Violation 22274
Loud Talking               21377
Partial Access             19988
Name: Descriptor, dtype: int64
```

```
In [38]: #Finding null values in the date set
df_nyc.isnull().sum()
```

```
Out[38]: Unique Key          0
Created Date                0
Closed Date                2164
Agency                    0
Agency Name                0
Complaint Type              0
Descriptor                 5914
Location Type              131
Incident Zip               2615
Incident Address           44410
Street Name                44410
Cross Street 1             49279
Cross Street 2             49779
Intersection Street 1      256840
Intersection Street 2      257336
Address Type               2815
City                       2614
Landmark                   300349
```

Facility Type	2171
Status	0
Due Date	3
Resolution Description	0
Resolution Action Updated Date	2187
Community Board	0
Borough	0
X Coordinate (State Plane)	3540
Y Coordinate (State Plane)	3540
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	300698
Vehicle Type	300698
Taxi Company Borough	300698
Taxi Pick Up Location	300698
Bridge Highway Name	300455
Bridge Highway Direction	300455
Road Ramp	300485
Bridge Highway Segment	300485
Garage Lot Name	300698
Ferry Direction	300697
Ferry Terminal Name	300696
Latitude	3540
Longitude	3540
Location	3540
Request Closing Time	2164

dtype: int64

```
In [45]: #Removing columns with majority null values
null_col = pd.DataFrame(df_nyc.columns.to_list()).set_index(0)
null_col = null_col[df_nyc.isnull().sum() < 200000].reset_index()
null_col
```

Out[45]: 0

0	
0	Unique Key
1	Created Date
2	Closed Date
3	Agency
4	Agency Name
5	Complaint Type
6	Descriptor
7	Location Type
8	Incident Zip
9	Incident Address
10	Street Name
11	Cross Street 1
12	Cross Street 2
13	Address Type
14	City
15	Facility Type
16	Status
17	Due Date
18	Resolution Description
19	Resolution Action Updated Date
20	Community Board
21	Borough
22	X Coordinate (State Plane)
23	Y Coordinate (State Plane)
24	Park Facility Name

0

25	Park Borough
26	School Name
27	School Number
28	School Region
29	School Code
30	School Phone Number
31	School Address
32	School City
33	School State
34	School Zip
35	School Not Found
36	Latitude
37	Longitude
38	Location
39	Request Closing Time

In [47]:

```
df_nyc = df_nyc[null_col[0].to_list()]
df_nyc.head()
```

Out[47]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	School Phone Number	Scho Addre
0	32310363	2015-12-31 23:59:00	2016-01-01 00:55:00	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	...	Unspecified	Unspecifi
1	32309934	2015-12-31 23:59:00	2016-01-01 01:26:00	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	...	Unspecified	Unspecifi

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	School Phone Number	School Address
2	32309159	2015-12-31 23:59:00	2016-01-01 04:51:00	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	...	Unspecified	Unspecified
3	32305098	2015-12-31 23:57:00	2016-01-01 07:43:00	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	...	Unspecified	Unspecified
4	32306529	2015-12-31 23:56:00	2016-01-01 03:24:00	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	...	Unspecified	Unspecified

5 rows × 40 columns



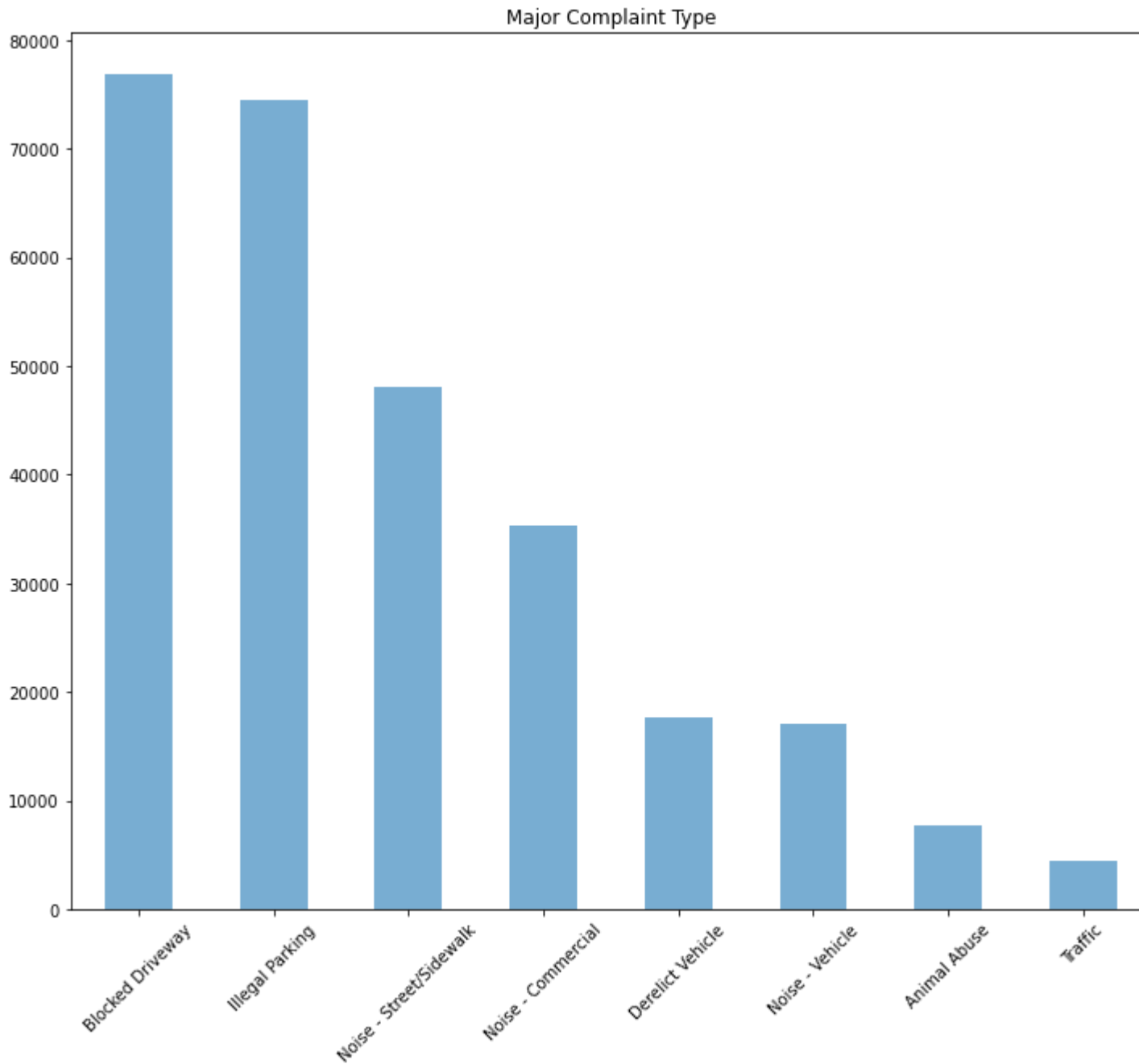
In [48]: `df_nyc.shape`

Out[48]: `(300698, 40)`

In [277... `#Bar plot to figure out major complaints received`  
`df_nyc['Complaint Type'].value_counts().head(8).plot(kind='bar', alpha=0.6, figsize=(12,10),rot=45,`  
`title='Major Complaint Type')`

Out[277... `<AxesSubplot:title={'center':'Major Complaint Type'}>`





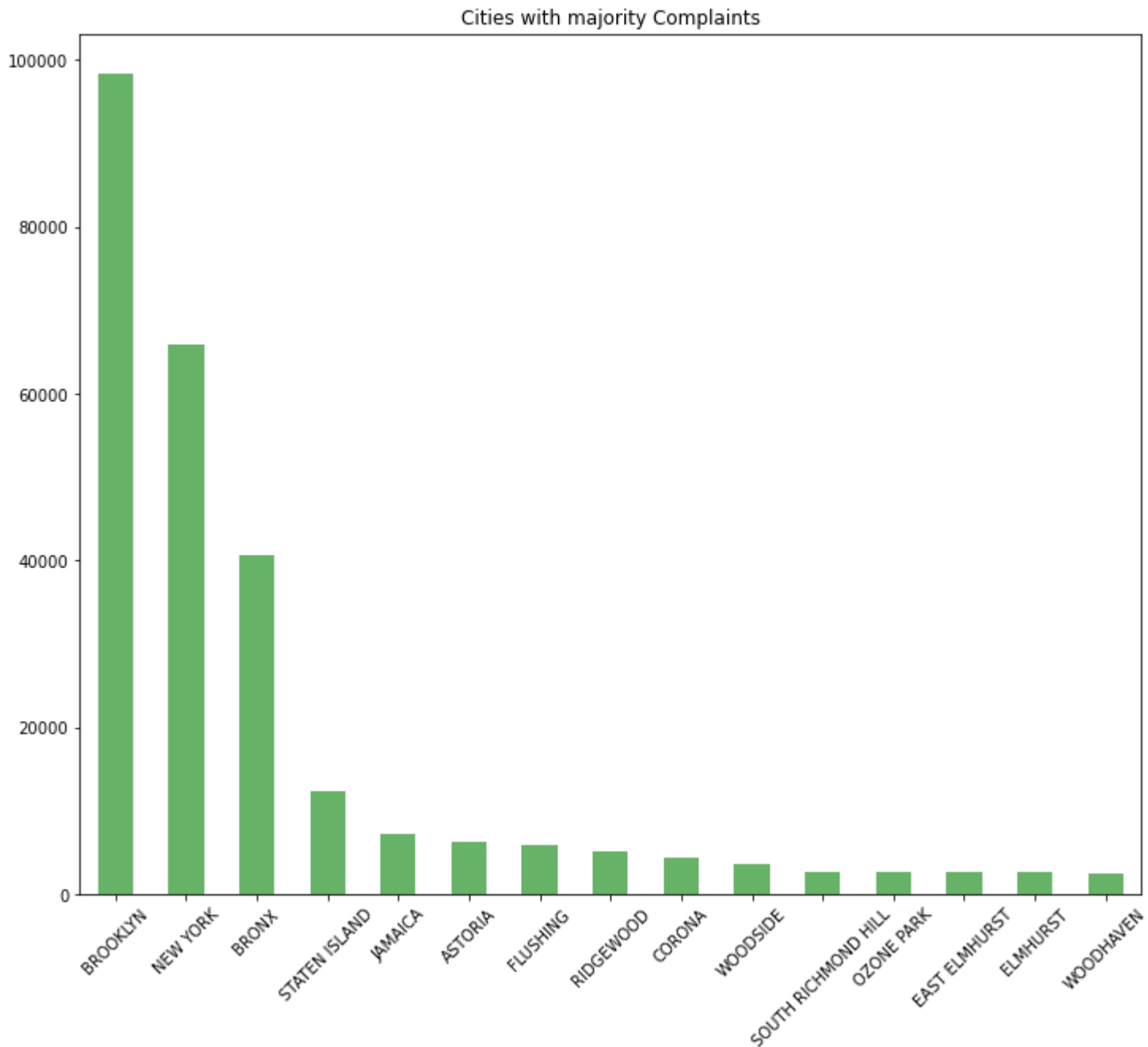
In [273...

```
#Top 15 cities having the most number of complaints  
df_city = df_nyc['City'].value_counts().nlargest(15)
```

```
df_city.plot(kind='bar', alpha=0.6, figsize=(12,10),rot=45, color='g', title='Cities with majority Complaints')
```

Out[273...

```
<AxesSubplot:title={'center':'Cities with majority Complaints'}>
```

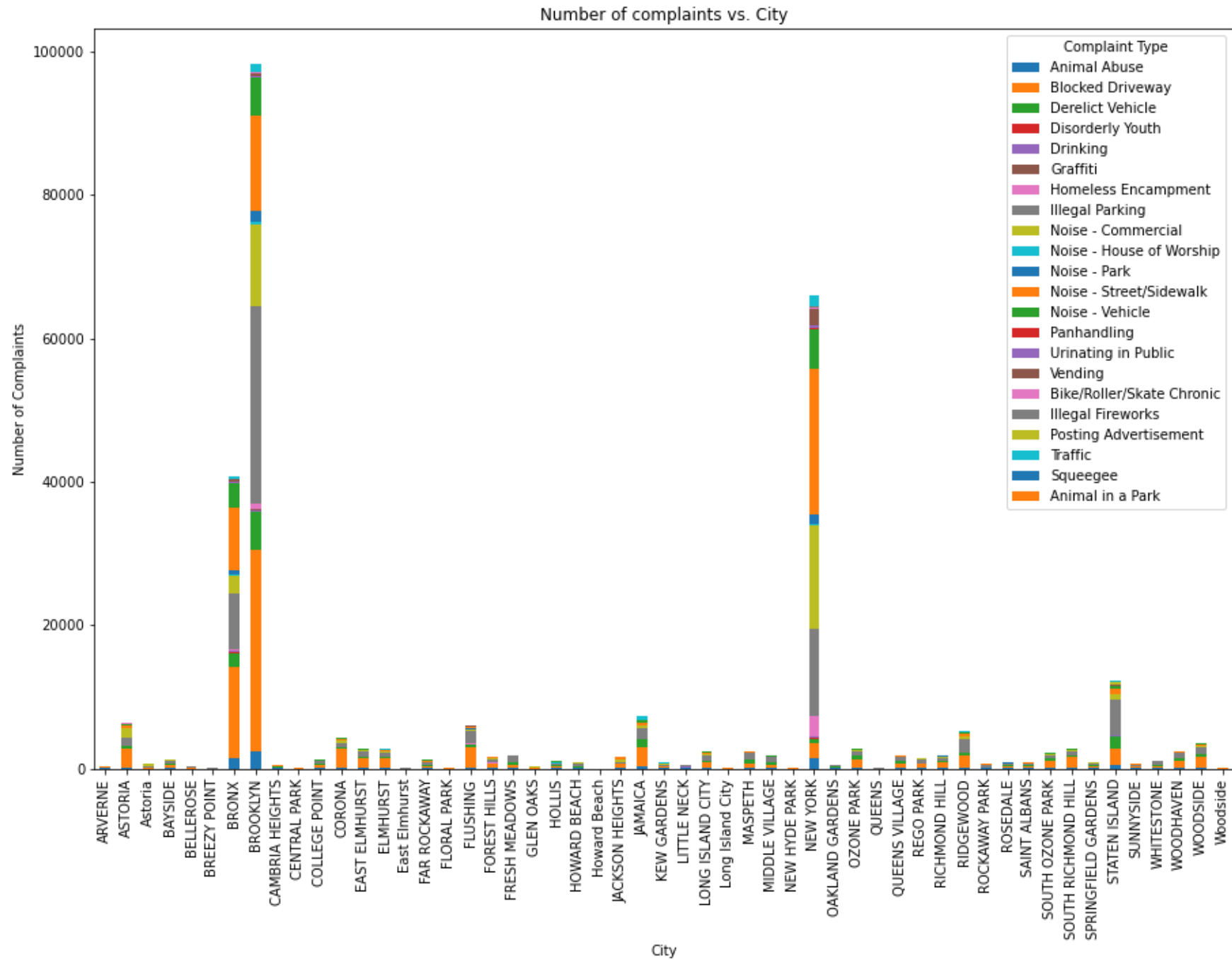


In [294...

```
#Number of Complaints vs City  
df_nyc_cVc=df_nyc.groupby(['City','Complaint Type']).size().unstack().fillna(0)  
df_nyc_cVc.plot(kind='bar', figsize=(15,10), stacked=True, title='Number of complaints vs. City',  
                ylabel='Number of Complaints')
```

Out[294...

```
<AxesSubplot:title={'center':'Number of complaints vs. City'}, xlabel='City', ylabel='Number of Complaints'>
```



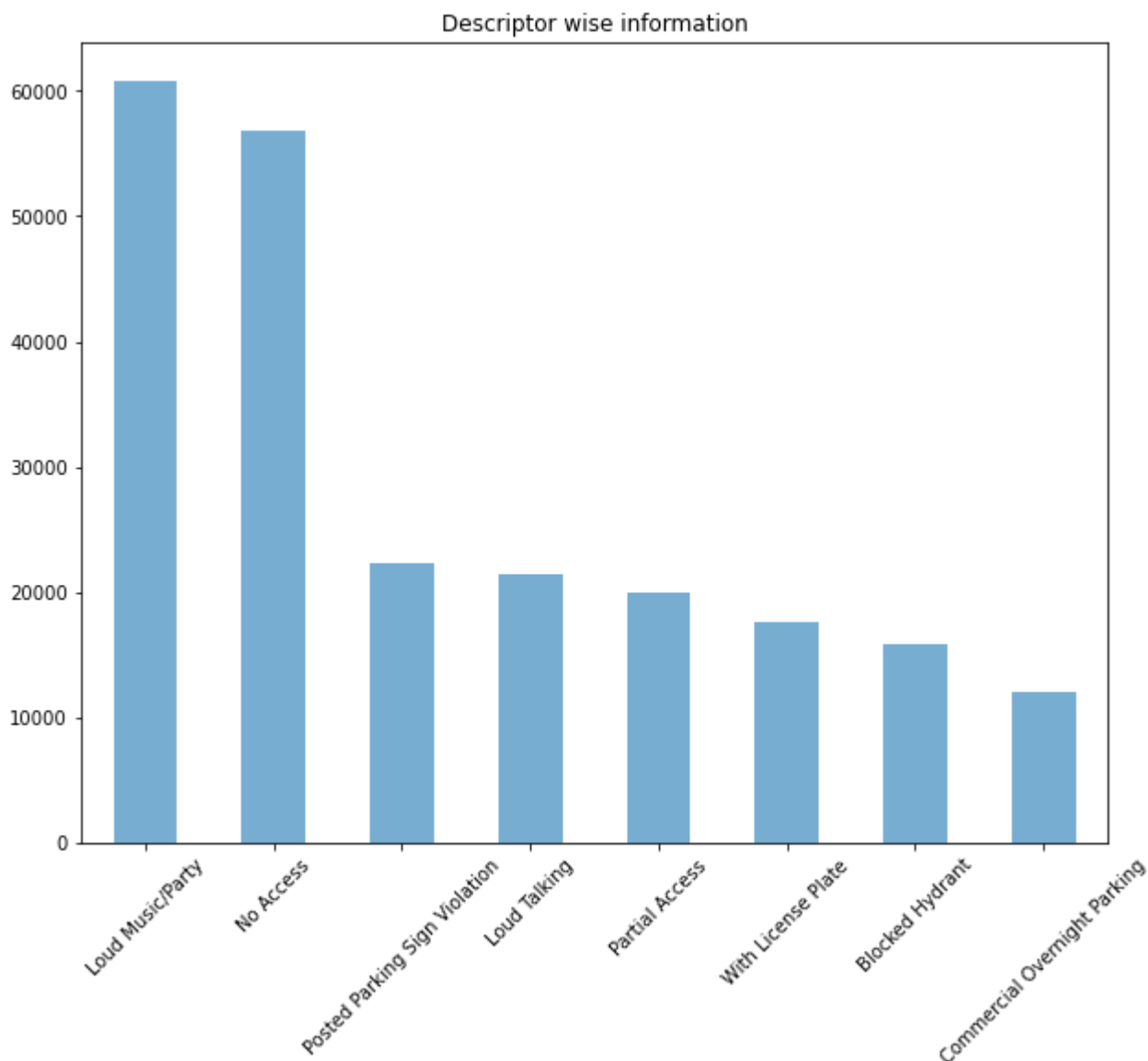
In [276...

#Descriptor wise information

```
df_nyc['Descriptor'].value_counts().head(8).plot(kind='bar',  
alpha=0.6, figsize=(10,8),rot=45, title='Descriptor wise information')
```

Out[276...

```
<AxesSubplot:title={'center':'Descriptor wise information'}>
```



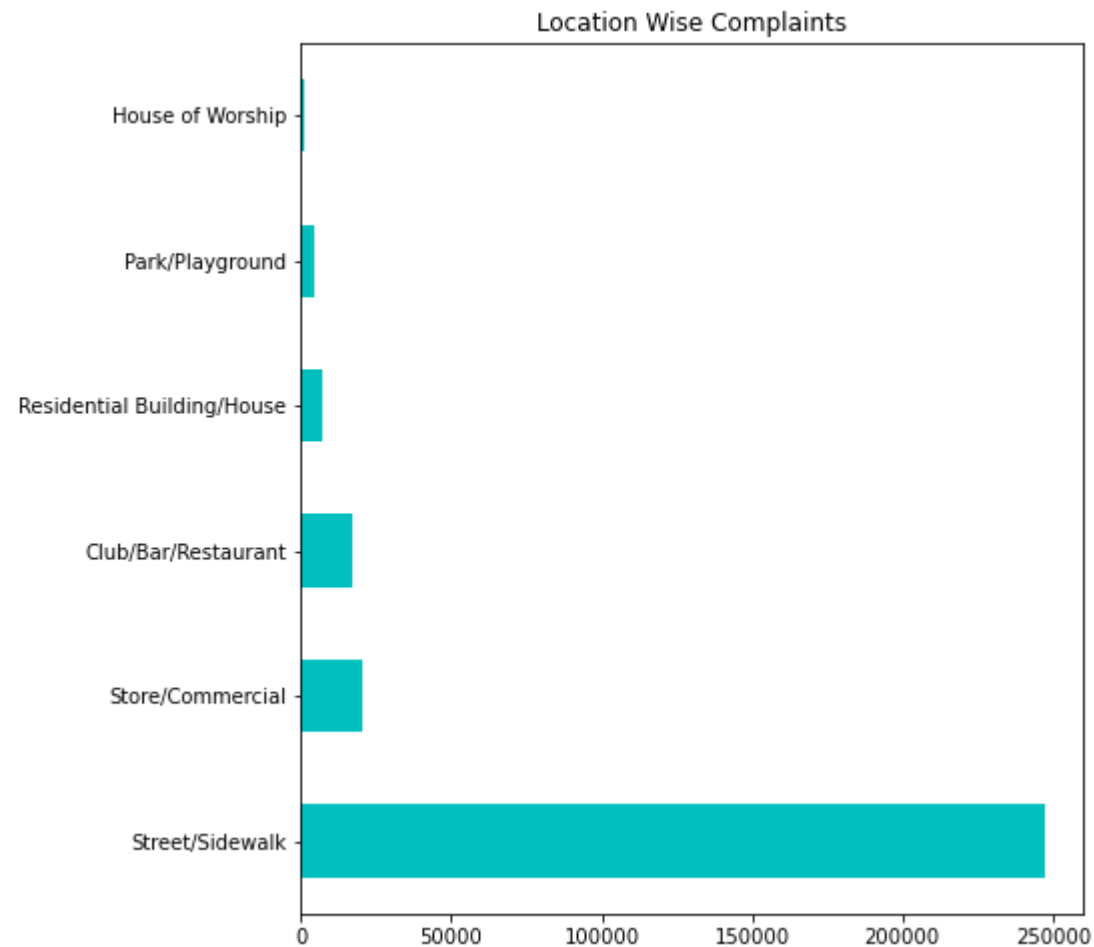
In [242...

```
#Location Type Complaints  
df_location = df_nyc['Location Type'].value_counts().nlargest(6)
```

```
df_location.plot(kind='barh', figsize=(7,8), color='c', title='Location Wise Complaints')
```

Out[242...

```
<AxesSubplot:title={'center':'Location Wise Complaints'}>
```



In [175...

```
#Analysing Borough information  
df_nyc['Borough'].value_counts()
```

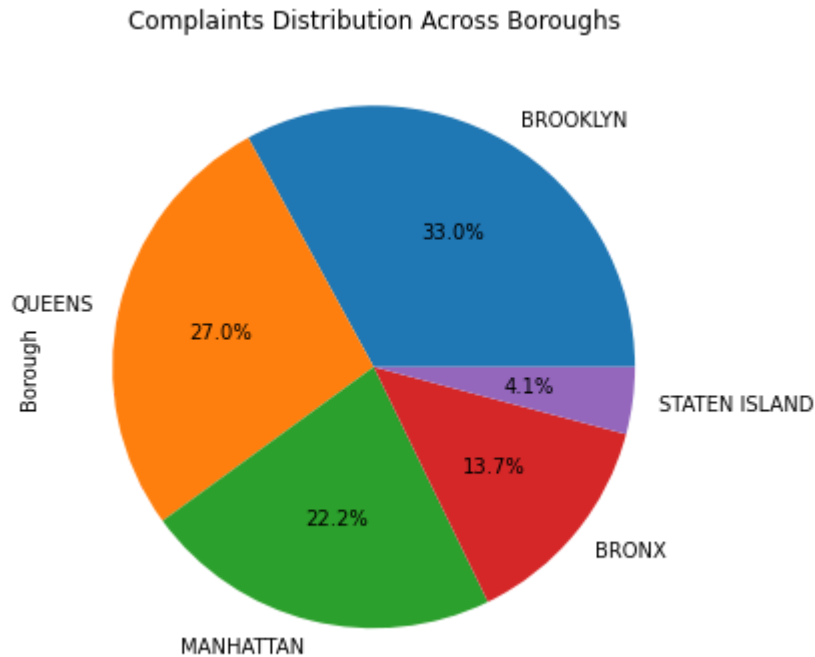
Out[175...

```
BROOKLYN      98307  
QUEENS        80641  
MANHATTAN     66131  
BRONX         40702  
STATEN ISLAND 12343  
Unspecified   2574  
Name: Borough, dtype: int64
```

```
In [165... #Dropping Unspecified Borough
df_nyc_borough1=df_nyc[df_nyc.Borough!='Unspecified']
```

```
In [293... df_nyc_borough1['Borough'].value_counts().plot(kind='pie',autopct='%1.1f%', figsize = (8,6),
                                                title='Complaints Distribution Across Boroughs')
```

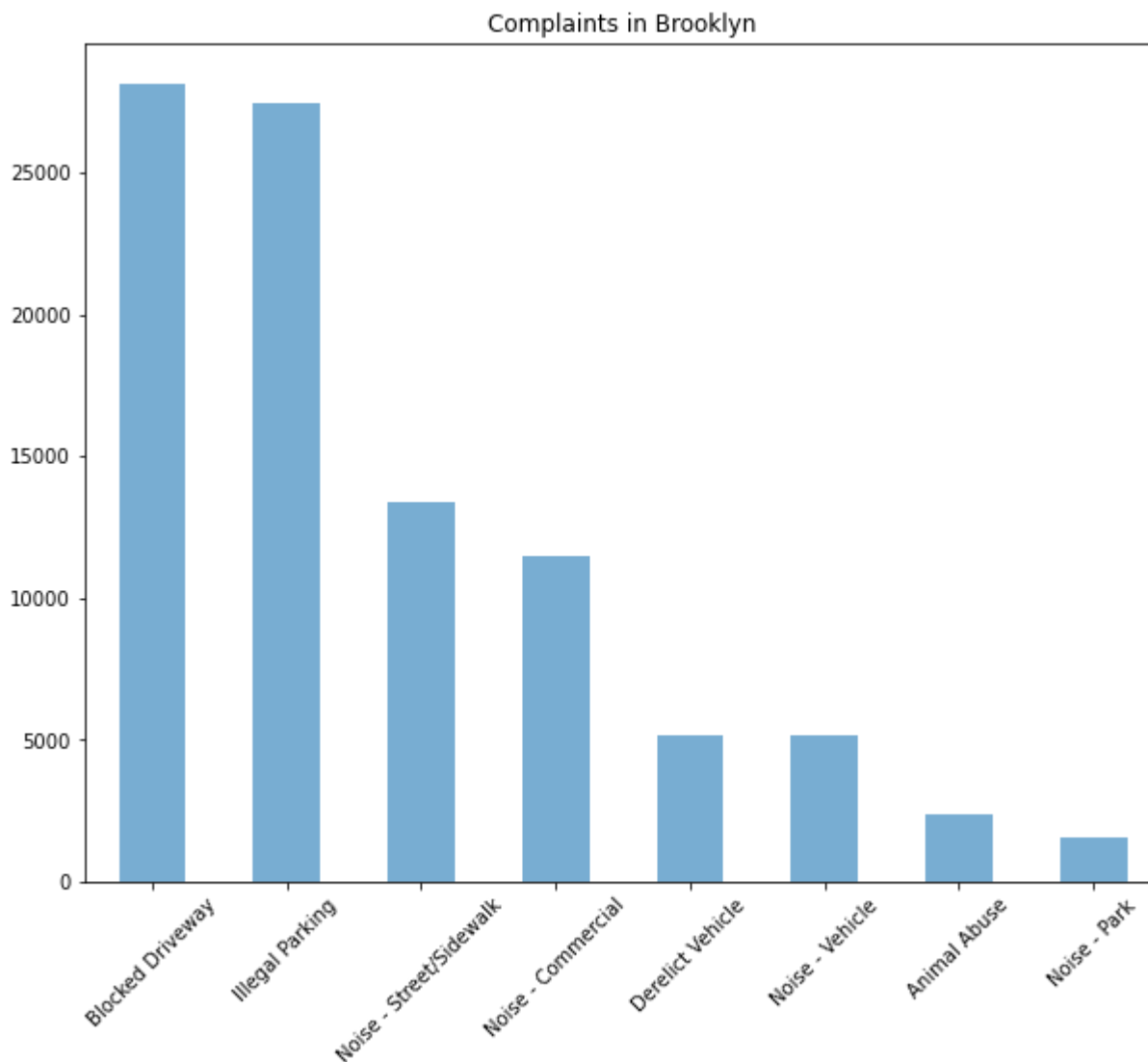
```
Out[293... <AxesSubplot:title={'center':'Complaints Distribution Across Boroughs'}, ylabel='Borough'>
```



```
In [4]: #Analysing Brooklyn Borough as it has recorded majority Complaints
df_nyc_brooklyn=df_nyc[df_nyc.Borough=='BROOKLYN']
```

```
In [6]: #Graph showing top 8 Complaints in Brooklyn
df_nyc_brooklyn['Complaint Type'].value_counts().head(8).plot(kind='bar',
                                                             alpha=0.6, figsize=(10,8), rot=45,
                                                             title='Complaints in Brooklyn')
```

```
Out[6]: <AxesSubplot:title={'center':'Complaints in Brooklyn'}>
```



In [74]:

```
#Graph showing avg resolution time for each complaint type in Brooklyn
df_nyc_brooklyn_res = df_nyc_brooklyn.groupby(['Complaint Type'])['Request Closing Time Sec'].mean().unstack()

y_labels = ['Illegal Parking', 'Blocked Driveway', 'Noise - Commercial',
            'Noise - Street/Sidewalk', 'Noise - Vehicle', 'Traffic',
            'Derelict Vehicle', 'Animal Abuse', 'Drinking', 'Vending',
            'Noise - Park', 'Homeless Encampment', 'Posting Advertisement',
            'Panhandling', 'Graffiti', 'Noise - House of Worship',
            'Bike/Roller/Skate Chronic', 'Urinating in Public',
```

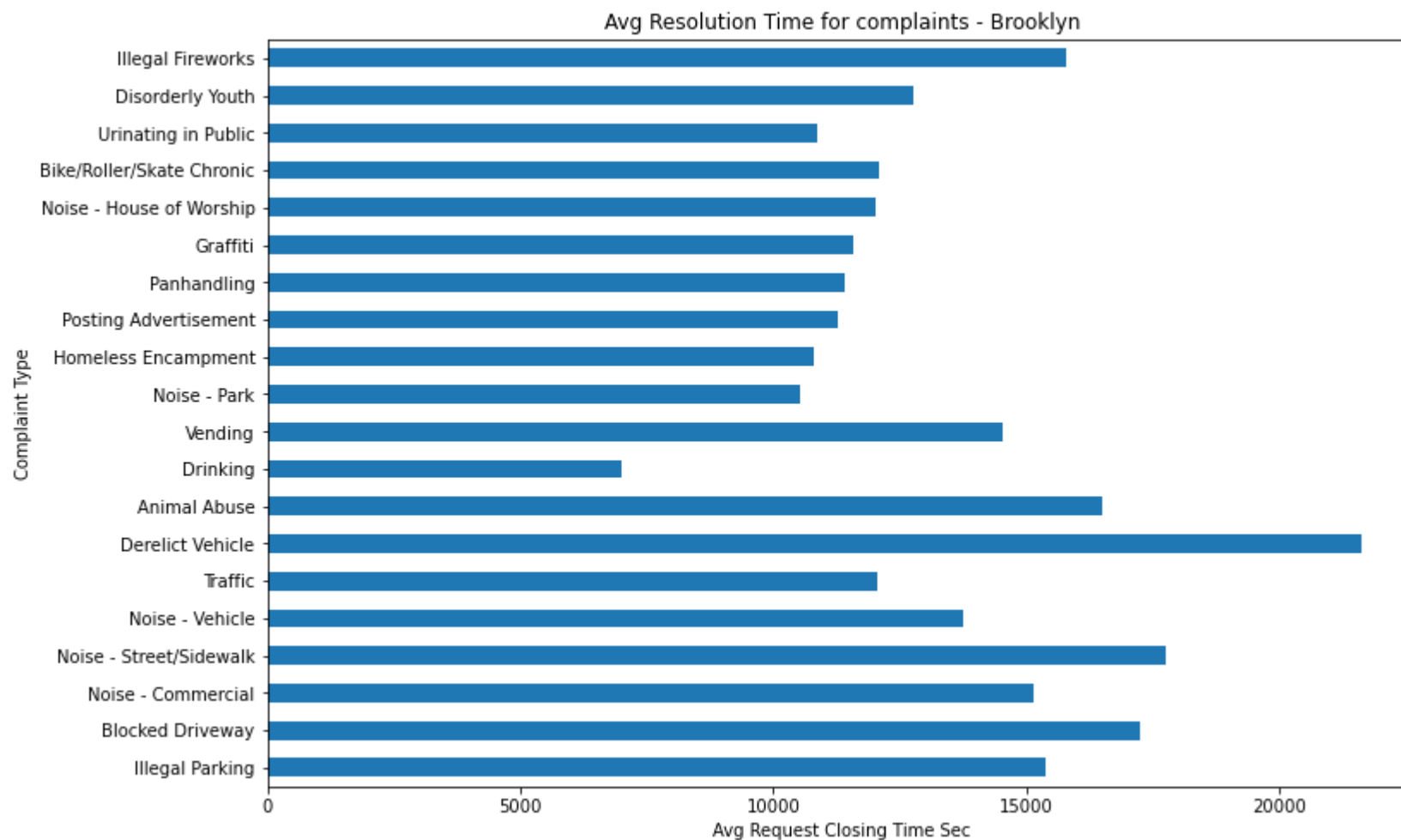


```
'Disorderly Youth', 'Illegal Fireworks']
```

```
plt.figure(figsize=(12, 8))
fig = df_nyc_brooklyn_res.plot(kind='barh')
fig.set_title('Avg Resolution Time for complaints - Brooklyn')
fig.set_xlabel('Avg Request Closing Time Sec')
fig.set_ylabel('Complaint Type')
fig.set_yticklabels(y_labels)
```

Out[74]:

```
[Text(0, 0, 'Illegal Parking'),
Text(0, 1, 'Blocked Driveway'),
Text(0, 2, 'Noise - Commercial'),
Text(0, 3, 'Noise - Street/Sidewalk'),
Text(0, 4, 'Noise - Vehicle'),
Text(0, 5, 'Traffic'),
Text(0, 6, 'Derelict Vehicle'),
Text(0, 7, 'Animal Abuse'),
Text(0, 8, 'Drinking'),
Text(0, 9, 'Vending'),
Text(0, 10, 'Noise - Park'),
Text(0, 11, 'Homeless Encampment'),
Text(0, 12, 'Posting Advertisement'),
Text(0, 13, 'Panhandling'),
Text(0, 14, 'Graffiti'),
Text(0, 15, 'Noise - House of Worship'),
Text(0, 16, 'Bike/Roller/Skate Chronic'),
Text(0, 17, 'Urinating in Public'),
Text(0, 18, 'Disorderly Youth'),
Text(0, 19, 'Illegal Fireworks')]
```



In [236...

```
#Ordering Complaints by Average Request Closing Time and Grouping by Location
df_nyc_group = df_nyc.sort_values('Request Closing Time Sec').groupby(['Complaint Type', 'Location Type'])
                                                         ['Request Closing Time Sec'].mean()

df_location = pd.DataFrame(df_nyc_group).head(50)
df_location.style.set_table_styles([{'selector' : 'tr', 'props':[( 'font size', '9pt'),('borderline' , 'solid')]}])
```

Out[236...

Request Closing Time Sec		
Complaint Type	Location Type	
Animal Abuse	Commercial	16445.806452
	House and Store	16190.967742

		Request Closing Time Sec
Complaint Type	Location Type	
	Park/Playground	11914.426230
	Parking Lot	16020.000000
	Residential Building	15821.233480
	Residential Building/House	17465.561897
	Store/Commercial	14884.606526
	Street/Sidewalk	15230.412574
	Subway Station	10930.909091
Animal in a Park	Park	3060.000000
	Residential Building/House	12998.400000
Bike/Roller/Skate Chronic	Store/Commercial	12193.584906
	Street/Sidewalk	12809.132948
Blocked Driveway	Street/Sidewalk	15783.620023
	Highway	16213.846154
	Roadway Tunnel	30120.000000
Derelict Vehicle	Street/Sidewalk	20143.576780
	Vacant Lot	14564.415584
	Residential Building/House	13874.805195
Disorderly Youth	Store/Commercial	10222.500000
	Street/Sidewalk	12075.223881
	Club/Bar/Restaurant	14472.657534
	Park/Playground	12388.775510
Drinking	Residential Building/House	12941.314879
	Store/Commercial	11705.333333
	Street/Sidewalk	12271.250000

		Request Closing Time Sec
Complaint Type	Location Type	
Graffiti	Residential Building/House	18078.214286
	Store/Commercial	20015.625000
	Street/Sidewalk	26059.200000
	Bridge	13740.000000
	Highway	11776.000000
Homeless Encampment	Park/Playground	13635.807365
	Residential Building/House	16495.544252
	Roadway Tunnel	9060.000000
	Store/Commercial	14411.015625
	Street/Sidewalk	14275.537190
Illegal Fireworks	Park/Playground	18015.000000
	Residential Building/House	11158.181818
	Store/Commercial	6930.000000
Illegal Parking	Street/Sidewalk	8461.920000
	Street/Sidewalk	14913.141469
Noise - Commercial	Club/Bar/Restaurant	10409.830770
	Store/Commercial	11097.945764
Noise - House of Worship	House of Worship	10671.585761
Noise - Park	Park/Playground	11713.698085
Noise - Street/Sidewalk	Street/Sidewalk	11596.277280
Noise - Vehicle	Street/Sidewalk	12154.400470
	Park/Playground	4370.000000
Panhandling	Residential Building/House	18600.000000
	Store/Commercial	15643.000000

In [76]:

```
#Ordering Complaints by Average Request Closing Time and Grouping by Location
df_nyc_group1 = df_nyc.sort_values('Request Closing Time Sec').groupby(['Complaint Type', 'City'])
                                                         ['Request Closing Time Sec'].mean()

df_city = pd.DataFrame(df_nyc_group1).head(50)
df_city.style.set_table_styles([{'selector' : 'tr', 'props':[('font size', '9pt'),('borderline' , 'solid')]}])
```

Out[76]:

		Request Closing Time Sec
Complaint Type	City	
Animal Abuse	ARVERNE	7754.210526
	ASTORIA	18000.480000
	BAYSIDE	11789.189189
	BELLEROSE	33462.857143
	BREEZY POINT	9390.000000
	BRONX	22316.777385
	BROOKLYN	15375.964912
	CAMBRIA HEIGHTS	9463.636364
	COLLEGE POINT	16722.857143
	CORONA	13203.934426
	EAST ELMHURST	14596.271186
	ELMHURST	13932.631579
	FAR ROCKAWAY	9791.460674
	FLORAL PARK	9300.000000
	FLUSHING	12809.790210
	FOREST HILLS	11752.000000
	FRESH MEADOWS	11808.000000
	GLEN OAKS	46776.000000
	HOLLIS	15736.363636
	HOWARD BEACH	15882.580645

Complaint Type	Request Closing Time Sec	
	City	
	JACKSON HEIGHTS	14615.714286
	JAMAICA	17082.183406
	KEW GARDENS	11387.368421
	LITTLE NECK	8292.000000
	LONG ISLAND CITY	21812.000000
	MASPETH	26203.333333
	MIDDLE VILLAGE	20841.818182
	NEW HYDE PARK	6900.000000
	NEW YORK	12869.822951
	OAKLAND GARDENS	9944.210526
	OZONE PARK	16066.250000
	QUEENS VILLAGE	29132.727273
	REGO PARK	15936.923077
	RICHMOND HILL	14538.750000
	RIDGEWOOD	20189.743590
	ROCKAWAY PARK	8222.000000
	ROSEDALE	26352.727273
	SAINT ALBANS	23658.000000
	SOUTH OZONE PARK	12571.636364
	SOUTH RICHMOND HILL	12399.230769
	SPRINGFIELD GARDENS	31907.500000
	STATEN ISLAND	17423.590664
	SUNNYSIDE	34186.285714
	WHITESTONE	10107.857143

		Request Closing Time Sec
Complaint Type	City	
Animal in a Park	WOODHAVEN	15965.333333
	WOODSIDE	25374.782609
	QUEENS	3060.000000
	ASTORIA	6256.000000
Bike/Roller/Skate Chronic	BELLEROSE	17640.000000
	BRONX	12453.000000

## STATISTICAL TESTS

### a. Whether the average response time across complaint types is similar or not (overall)

Null Hypothesis: There is no significant difference in average response time across different complaint types

Alternate Hypothesis: There is a significant difference in average response time across different complaint types

```
In [330... df_nyc['Complaint Type'].unique()
```

```
Out[330... array(['Noise - Street/Sidewalk', 'Blocked Driveway', 'Illegal Parking',
      'Derelict Vehicle', 'Noise - Commercial',
      'Noise - House of Worship', 'Posting Advertisement',
      'Noise - Vehicle', 'Animal Abuse', 'Vending', 'Traffic',
      'Drinking', 'Bike/Roller/Skate Chronic', 'Panhandling',
      'Noise - Park', 'Homeless Encampment', 'Urinating in Public',
      'Graffiti', 'Disorderly Youth', 'Illegal Fireworks',
      'Agency Issues', 'Squeegee', 'Animal in a Park'], dtype=object)
```

```
In [343... df_nyc_group2 = df_nyc.sort_values('Request Closing Time Sec').groupby(['Complaint Type'])['Request Closing Time Sec'].me
df_city = pd.DataFrame(df_nyc_group2)
df_city.style.set_table_styles([{'selector' : 'tr', 'props':[( 'font size', '9pt'), ('borderline' , 'solid')]}])
```

```
Out[343...
```

<b>Request Closing Time Sec</b>	
<b>Complaint Type</b>	
<b>Agency Issues</b>	18930.000000
<b>Animal Abuse</b>	16654.302266
<b>Animal in a Park</b>	3060.000000
<b>Bike/Roller/Skate Chronic</b>	12743.349057
<b>Blocked Driveway</b>	15783.826064
<b>Derelict Vehicle</b>	20114.938594
<b>Disorderly Youth</b>	12507.902098
<b>Drinking</b>	13021.505882
<b>Graffiti</b>	20392.566372
<b>Homeless Encampment</b>	14757.146739
<b>Illegal Fireworks</b>	9428.214286
<b>Illegal Parking</b>	14915.040251
<b>Noise - Commercial</b>	10768.560729
<b>Noise - House of Worship</b>	10659.268030
<b>Noise - Park</b>	11720.079562
<b>Noise - Street/Sidewalk</b>	11595.727598
<b>Noise - Vehicle</b>	12154.383843
<b>Panhandling</b>	13473.639344
<b>Posting Advertisement</b>	6979.166667
<b>Squeegee</b>	14565.000000
<b>Traffic</b>	11511.939502
<b>Urinating in Public</b>	12326.554054
<b>Vending</b>	13699.272727

From the above data it is observed that for majority of the population average response time is not equal.



However it is also observed that the Average Response Time for few Complaint types are significantly close. To analyse this further we'll perform ANOVA test for those complaint types.

The two complaint types compared here are: Disorderly Youth:12507.902098 and Noise - Vehicle: 12154.383843

To perform the hypothesis test we will conduct One Way ANOVA.

```
In [345... df_nyc_youth = df_nyc[df_nyc['Complaint Type']=='Disorderly Youth']  
df_nyc_youth = df_nyc_youth.loc[:,['Request Closing Time Sec']]  
df_nyc_youth.head()
```

```
Out[345... Request Closing Time Sec
```

<b>4670</b>	720.0
<b>9034</b>	4620.0
<b>12027</b>	2340.0
<b>12176</b>	19440.0
<b>17181</b>	6840.0

```
In [346... df_nyc_noise = df_nyc[df_nyc['Complaint Type']=='Noise - Vehicle']  
df_nyc_noise = df_nyc_noise.loc[:,['Request Closing Time Sec']]  
df_nyc_noise.head()
```

```
Out[346... Request Closing Time Sec
```

<b>87</b>	22980.0
<b>156</b>	7260.0
<b>172</b>	11280.0
<b>221</b>	10920.0
<b>319</b>	2640.0

```
In [347... #One way Anova for Disorderly Youth and Noise - Vehicle  
fvalue, pvalue = stats.f_oneway(df_nyc_youth, df_nyc_noise)  
pvalue
```

Out[347... array([0.62622334])

Null hypothesis to be accepted for Disorderly Youth and Noise - Vehicle p-value close to 1

## Performing ANOVA for Complaint Type and Request Closing Time

```
In [354... from statsmodels.formula.api import ols
import scipy.stats as stats
import statsmodels.api as sm
```

```
In [356... import warnings
warnings.filterwarnings("ignore")
```

```
In [357... df_nyc['Complaint_type'] = df_nyc['Complaint Type']
df_nyc['Request_Closing_Time'] = df_nyc['Request Closing Time Sec']
df_type_res = df_nyc.loc[:, ['Complaint_Type', 'Request_Closing_Time']] #Complaint Type
# Ordinary Least Squares (OLS) model
model = ols('Request_Closing_Time ~ Complaint_Type', data=df_type_res).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
anova_table
```

```
Out[357...
```

	sum_sq	df	F	PR(>F)
<b>Complaint_Type</b>	1.997856e-13	22.0	4202.592407	0.0
<b>Residual</b>	6.450369e-13	298511.0	NaN	NaN

Null Hypothesis to be rejected as P Value is < 0.05

**CONCLUSION:** Reject Null Hypothesis, there is a significant difference in average response time across different complaint types.

## b. Are the type of complaint or service requested and Location related?

To check the above statement we use Chi-Square test of Independence

Null Hypothesis: There is no relationship between the complaint type and location

Alternate Hypothesis: There is a relationship between the complaint type and location

```
In [360... df_nyc_chi_sq = pd.crosstab(df_nyc['Complaint Type'], df_nyc['Location Type'])
```

```
In [361... chisq,pval,df,et = stats.chi2_contingency(df_nyc_chi_sq)
print('chisq Score : {:.2f}, pvalue : {:.3f}'.format(chisq,pval))
```

```
chisq Score : 1328207.97, pvalue : 0.000
```

**Null Hypothesis to be rejected as P Value is  $< 0.05$**

**CONCLUSION: We reject Null Hypothesis - There is some significant relation between type of complaint and location**