In [1]:
```
#   Author: Rahul ArvindKumar Thakur
#   Email: rahulkumarr2080@gmail.com
#   Date: 31st July 2021
```

In [2]:
```
#importing required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from datetime import datetime
plt.rcParams["figure.figsize"] = (20,7)
```

Task 1 - Import a 311 NYC service request.

In [3]:
```
# 1)Import a 311 NYC service request.
customer_service_request=pd.read_csv('DataSets/311_Service_Requests_from_2010_to_Presen

#Giving Low Memory Warning so put low_memory=False above
```

In [4]:
```
#Check CSV Imported or not, This code will help to view top 5 Rows
customer_service_request.head()
```

Out[4]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incid
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	100
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	111
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	104
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	104
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	113

5 rows × 53 columns

In [5]:
```
# Check Number of Rows(observations) & Number of Columns(Variables)
customer_service_request.shape
```

Out[5]: (300698, 53)

In [6]:
```
#This means this Dataset has 300698 Rows(observations) and 53 Columns(Variables)
```

In [7]:
```
##############################################################################################
#            Data Preprocessing  - Cleaning up the data and make it ready for building
##############################################################################################
```

In [8]:
```
# Check the duplicate columns or Variables with duplicate name
customer_service_request.columns
```

Out[8]:
```
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')
```

In [9]:
```
# list of empty columns name given in data description they are of no use so we have to
empty_data_cols = ['Landmark','School Not Found','School or Citywide Complaint','Vehicl
'Taxi Pick Up Location','Bridge Highway Name','Bridge Highway Direction','Road Ramp',
'Bridge Highway Segment','Garage Lot Name','Ferry Direction','Ferry Terminal Name']

unspecified_data_cols =['Park Facility Name','School Name','School Number','School Regi
'School Phone Number','School Address','School City','School State','School Zip']

# Deleteing both Empty Columns and Unspecified columns from data frame
customer_service_request= customer_service_request.drop(empty_data_cols, axis=1)
customer_service_request= customer_service_request.drop(unspecified_data_cols, axis=1)
customer_service_request.shape
```

Out[9]: (300698, 30)

In [10]:
```
# Deleting duplicate representation in multiple columns
# Location with lat long info already avaiable in Lattitude and Longtitude. Deleting lo
customer_service_request= customer_service_request.drop(['Latitude','Longitude'], axis=
customer_service_request.shape
```

Out[10]: (300698, 28)

In [11]:
```
#Removing ID Column 'Unique Key' which doesnt have any significance in analytics model
customer_service_request = customer_service_request.drop(['Unique Key'],axis=1)
```

In [12]:
```
# Now we have to identify columns which has null or empty or single value and these col
# any significance while processing model
single_value_columns = []
for i in customer_service_request.columns:
    if len(customer_service_request[i].value_counts()) ==1:
        single_value_columns.append(i)
        print('Following columns has single value ', single_value_columns)
#Deleting single value columns from dataframe
customer_service_request= customer_service_request.drop(single_value_columns, axis=1)
```

```
Following columns has single value  ['Agency']
Following columns has single value  ['Agency', 'Facility Type']
```

In [13]:
```
# Removing columns which has more than 70% data is null or empty and doesnt have any im
customer_service_request.isnull().sum().sort_values(ascending=False)
```

Out[13]:
```
Intersection Street 2             257336
Intersection Street 1             256840
Cross Street 2                     49779
Cross Street 1                     49279
Incident Address                   44410
Street Name                        44410
Descriptor                          5914
Y Coordinate (State Plane)          3540
X Coordinate (State Plane)          3540
Location                            3540
Address Type                        2815
Incident Zip                        2615
City                                2614
Resolution Action Updated Date      2187
Closed Date                         2164
Location Type                        131
Due Date                               3
Status                                 0
Resolution Description                 0
Community Board                        0
Borough                                0
Complaint Type                         0
Agency Name                            0
Park Borough                           0
Created Date                           0
dtype: int64
```

In [14]:
```
# Above cell listed Intersection Street 2 and Intersection Street 1 has more than 70% d
# doesnt have much impact on analytics model building.hence dropping these columns
customer_service_request =customer_service_request.drop(['Intersection Street 1','Inter
customer_service_request.shape
```

Out[14]:
```
(300698, 23)
```

In [15]:
```
# Seperating object columns and non object columns from dataframe not required.
# non object columns are Incident Zip, X Coordinate (State Plane), Y Coordinate (State
# these columns represented in number format but these are more categorized data than n
# hence converting these columns as object columns
customer_service_request=customer_service_request.astype(
    {'Incident Zip':'str','X Coordinate (State Plane)':'str','Y Coordinate (State Plane
```

In [16]:
```
#Converting string data to upper case to make model case insentive
```

```
for col in customer_service_request.columns:
    customer_service_request[col] = customer_service_request[col].str.upper()

customer_service_request.head()
```

Out[16]:

	Created Date	Closed Date	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Inc Ad
0	12/31/2015 11:59:45 PM	01-01-16 0:55	NEW YORK CITY POLICE DEPARTMENT	NOISE - STREET/SIDEWALK	LOUD MUSIC/PARTY	STREET/SIDEWALK	10034.0	VERM AV
1	12/31/2015 11:59:44 PM	01-01-16 1:26	NEW YORK CITY POLICE DEPARTMENT	BLOCKED DRIVEWAY	NO ACCESS	STREET/SIDEWALK	11105.0	27- AV
2	12/31/2015 11:59:29 PM	01-01-16 4:51	NEW YORK CITY POLICE DEPARTMENT	BLOCKED DRIVEWAY	NO ACCESS	STREET/SIDEWALK	10458.0	VALE AV
3	12/31/2015 11:57:46 PM	01-01-16 7:43	NEW YORK CITY POLICE DEPARTMENT	ILLEGAL PARKING	COMMERCIAL OVERNIGHT PARKING	STREET/SIDEWALK	10461.0	BA AV
4	12/31/2015 11:56:58 PM	01-01-16 3:24	NEW YORK CITY POLICE DEPARTMENT	ILLEGAL PARKING	BLOCKED SIDEWALK	STREET/SIDEWALK	11373.0	87-

5 rows × 23 columns

In [17]:

```
#Due Date and Resolution Action Updated Date cannot be less than Created Date. Filling
customer_service_request['Due Date']=customer_service_request['Due Date'].fillna(custom
customer_service_request['Resolution Action Updated Date']=customer_service_request[
    'Resolution Action Updated Date'].fillna(customer_service_request['Created Date'])
```

In [18]:

```
#Filling empty/null columns of object columns with max occurance item value
for col in customer_service_request:
    customer_service_request[col]=customer_service_request[col].fillna(
        customer_service_request[col].value_counts().idxmax())
#Checking is there any null cells present in Dataframe
customer_service_request.isnull().sum().sort_values()
```

Out[18]:

```
Created Date                    0
Y Coordinate (State Plane)      0
X Coordinate (State Plane)      0
```

```
Borough                         0
Community Board                 0
Resolution Action Updated Date  0
Resolution Description          0
Due Date                        0
Status                          0
City                            0
Park Borough                    0
Address Type                    0
Cross Street 1                  0
Street Name                     0
Incident Address                0
Incident Zip                    0
Location Type                   0
Descriptor                      0
Complaint Type                  0
Agency Name                     0
Closed Date                     0
Cross Street 2                  0
Location                        0
dtype: int64
```

In [19]:
```
customer_service_request.Status.value_counts()
```

Out[19]:
```
CLOSED      298471
OPEN          1439
ASSIGNED       786
DRAFT            2
Name: Status, dtype: int64
```

In [20]:
```
# Converting all the date columns datatype from str object column to datetime object co
date_cols =['Created Date', 'Closed Date','Due Date','Resolution Action Updated Date']
for col in date_cols:
    customer_service_request[col] = pd.to_datetime(customer_service_request[col])
```

In [21]:
```
customer_service_request.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 23 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   Created Date                    300698 non-null  datetime64[ns]
 1   Closed Date                     300698 non-null  datetime64[ns]
 2   Agency Name                     300698 non-null  object
 3   Complaint Type                  300698 non-null  object
 4   Descriptor                      300698 non-null  object
 5   Location Type                   300698 non-null  object
 6   Incident Zip                    300698 non-null  object
 7   Incident Address                300698 non-null  object
 8   Street Name                     300698 non-null  object
 9   Cross Street 1                  300698 non-null  object
 10  Cross Street 2                  300698 non-null  object
 11  Address Type                    300698 non-null  object
 12  City                            300698 non-null  object
 13  Status                          300698 non-null  object
 14  Due Date                        300698 non-null  datetime64[ns]
 15  Resolution Description          300698 non-null  object
 16  Resolution Action Updated Date  300698 non-null  datetime64[ns]
 17  Community Board                 300698 non-null  object
 18  Borough                         300698 non-null  object
```

```
 19  X Coordinate (State Plane)      300698 non-null  object
 20  Y Coordinate (State Plane)      300698 non-null  object
 21  Park Borough                    300698 non-null  object
 22  Location                        300698 non-null  object
dtypes: datetime64[ns](4), object(19)
memory usage: 52.8+ MB
```

In [22]:

```
closed_status_requests = customer_service_request[customer_service_request['Status']=='
non_closed_status_requests = customer_service_request[customer_service_request['Status'
open_status_requests = customer_service_request[customer_service_request['Status']=='OP
assigned_status_requests = customer_service_request[customer_service_request['Status']=
draft_status_requests = customer_service_request[customer_service_request['Status']=='D
print('Closed requests',closed_status_requests.shape)
print('Not closed requests (OPEN+ASSIGNED+DRAFT)', non_closed_status_requests.shape)
print('Open requests',open_status_requests.shape)
print('Assigned requests',assigned_status_requests.shape)
print('Drafted requests',draft_status_requests.shape)
```

```
Closed requests (298471, 23)
Not closed requests (OPEN+ASSIGNED+DRAFT) (2227, 23)
Open requests (1439, 23)
Assigned requests (786, 23)
Drafted requests (2, 23)
```

In [23]:

```
#Function to range classify observations based on Request_Closing_Time for CLOSED statu
def segregate_duration_range(duration):
    if duration <=60:
        return "00-1 Hrs"
    elif duration >60 and duration<= 120:
        return "1-2 Hrs"
    elif duration >120 and duration<= 240:
        return "2-4 Hrs"
    elif duration >240 and duration<= 480:
        return "4-8 Hrs"
    elif duration >480 and duration< 960:
        return "8-16 Hrs"
    elif duration >960 and duration< 1440:
        return "16- 24 Hrs"
    elif duration >1440 and duration< 2880:
        return "1-2 Days"
    elif duration >2880 and duration< 5760:
        return "2-4 Days"
    elif duration >5760 and duration< 11520:
        return "4-8 Days"
    elif duration >11520 :
        return "More than 8 Days"
```

Task 2 - Read or convert the columns 'Created Date' and Closed Date' to datetime datatype and create a new column 'Request_Closing_Time' as the time elapsed between request creation and request closing. (Hint: Explore the package/module datetime)

In [24]:

```
closed_status_requests.shape
```

Out[24]: (298471, 23)

In [25]:

```
print('Both Created Date and Closed Date is available for closed CSRs')
closed_status_requests['Request_Closing_Time']=0
closed_status_requests['Resolved_Duration_Grp']=''
```

```
closed_status_requests['Closed-MM-YYYY']=''
 # request_closing_time_index = closed_status_requests.columns.get_loc('Request_Closing_
for index, csr in closed_status_requests.iterrows():
    Request_Closing_Time = int(((csr['Closed Date']-csr['Created Date']).total_seconds(
    closed_status_requests.at[index,'Request_Closing_Time'] = Request_Closing_Time
    closed_status_requests.at[index,'Resolved_Duration_Grp'] = segregate_duration_range
closed_status_requests.head()
```

```
Both Created Date and Closed Date is available for closed CSRs
<ipython-input-25-308fd748276a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  closed_status_requests['Request_Closing_Time']=0
<ipython-input-25-308fd748276a>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  closed_status_requests['Resolved_Duration_Grp']=''
<ipython-input-25-308fd748276a>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  closed_status_requests['Closed-MM-YYYY']=''
```

Out[25]:

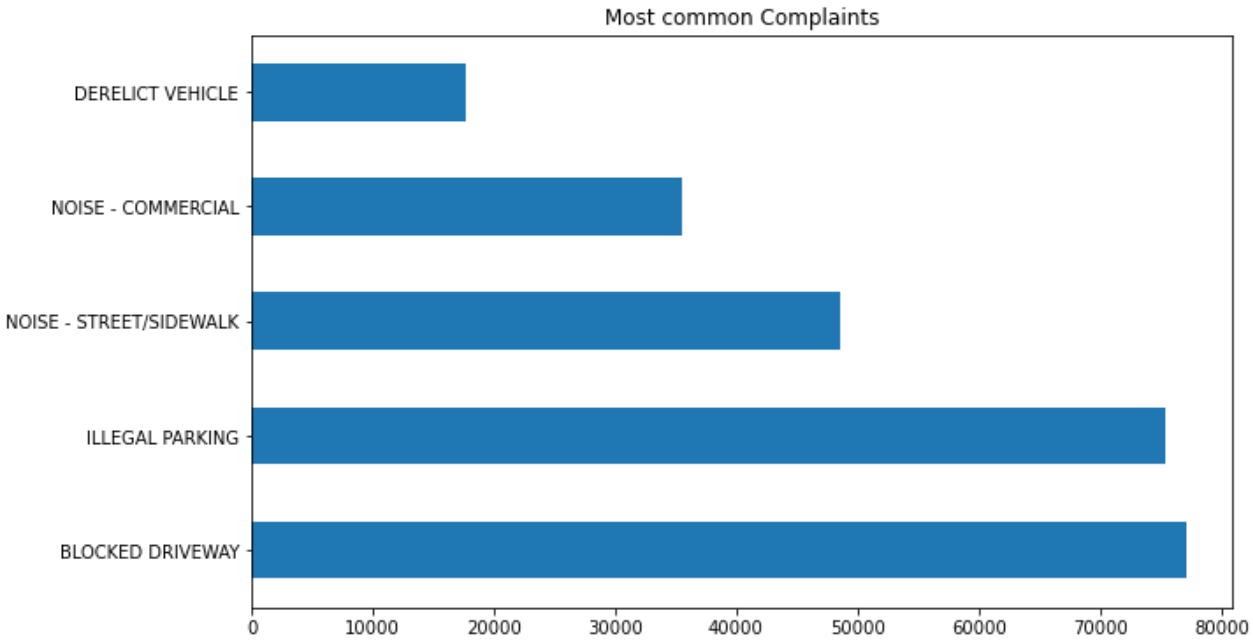
	Created Date	Closed Date	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incid Addı
0	2015-12-31 23:59:45	2016-01-01 00:55:00	NEW YORK CITY POLICE DEPARTMENT	NOISE - STREET/SIDEWALK	LOUD MUSIC/PARTY	STREET/SIDEWALK	10034.0	VERMIL AVEI
1	2015-12-31 23:59:44	2016-01-01 01:26:00	NEW YORK CITY POLICE DEPARTMENT	BLOCKED DRIVEWAY	NO ACCESS	STREET/SIDEWALK	11105.0	27-07 AVEI
2	2015-12-31 23:59:29	2016-01-01 04:51:00	NEW YORK CITY POLICE DEPARTMENT	BLOCKED DRIVEWAY	NO ACCESS	STREET/SIDEWALK	10458.0	2 VALENT AVEI
3	2015-12-31 23:57:46	2016-01-01 07:43:00	NEW YORK CITY POLICE DEPARTMENT	ILLEGAL PARKING	COMMERCIAL OVERNIGHT PARKING	STREET/SIDEWALK	10461.0	2 BAIS AVEI
4	2015-12-31 23:56:58	2016-01-01 03:24:00	NEW YORK CITY POLICE DEPARTMENT	ILLEGAL PARKING	BLOCKED SIDEWALK	STREET/SIDEWALK	11373.0	87-14 RC

5 rows × 26 columns

Task 3 - Provide major insights/patterns that you can offer in a visual format (graphs or tables); at least 4 major conclusions that you can come up with after generic data mining.

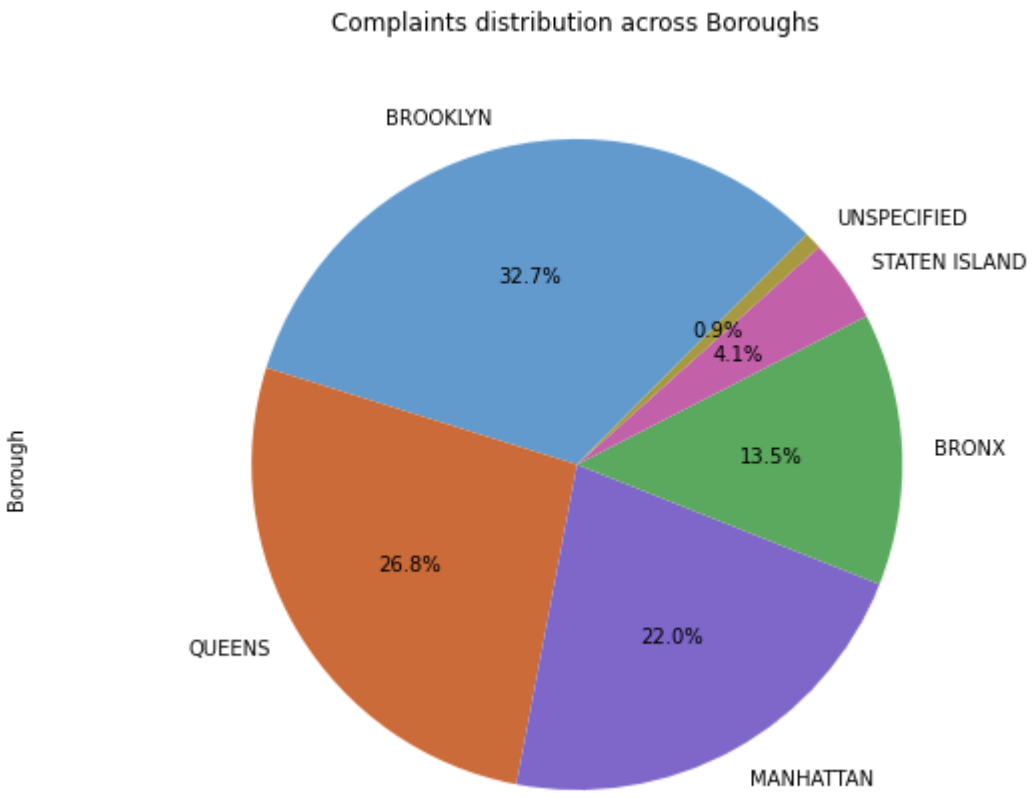
In [26]:
```
# Conclusion -1 Most frequent Complaints - Top 5
# Used .head() to show top 5 major Complaint Type
(customer_service_request['Complaint Type'].value_counts()).head().plot(kind='barh',
                    figsize=(10,6), title = 'Most common Complaints')
```

Out[26]: <AxesSubplot:title={'center':'Most common Complaints'}>



1)This graph shows that Blocked Driveway and Illegal Parking are the common complaints

In [27]:
```
# Conclusion -2 complaints distribution across Boroughs
colors = ['#639ace','#ca6b39','#7f67ca','#5ba85f','#c360aa','#a7993f','#cc566a']
customer_service_request['Borough'].value_counts().plot(kind='pie',autopct='%1.1f%%',
                        startangle=45, colors = colors,
                        figsize = (8,6))
#plt.legend(title='BOROUGH', loc='upper right', bbox_to_anchor=(1.5,1))
plt.axis('equal')
plt.title('Complaints distribution across Boroughs\n\n')
plt.tight_layout()
plt.show()
```



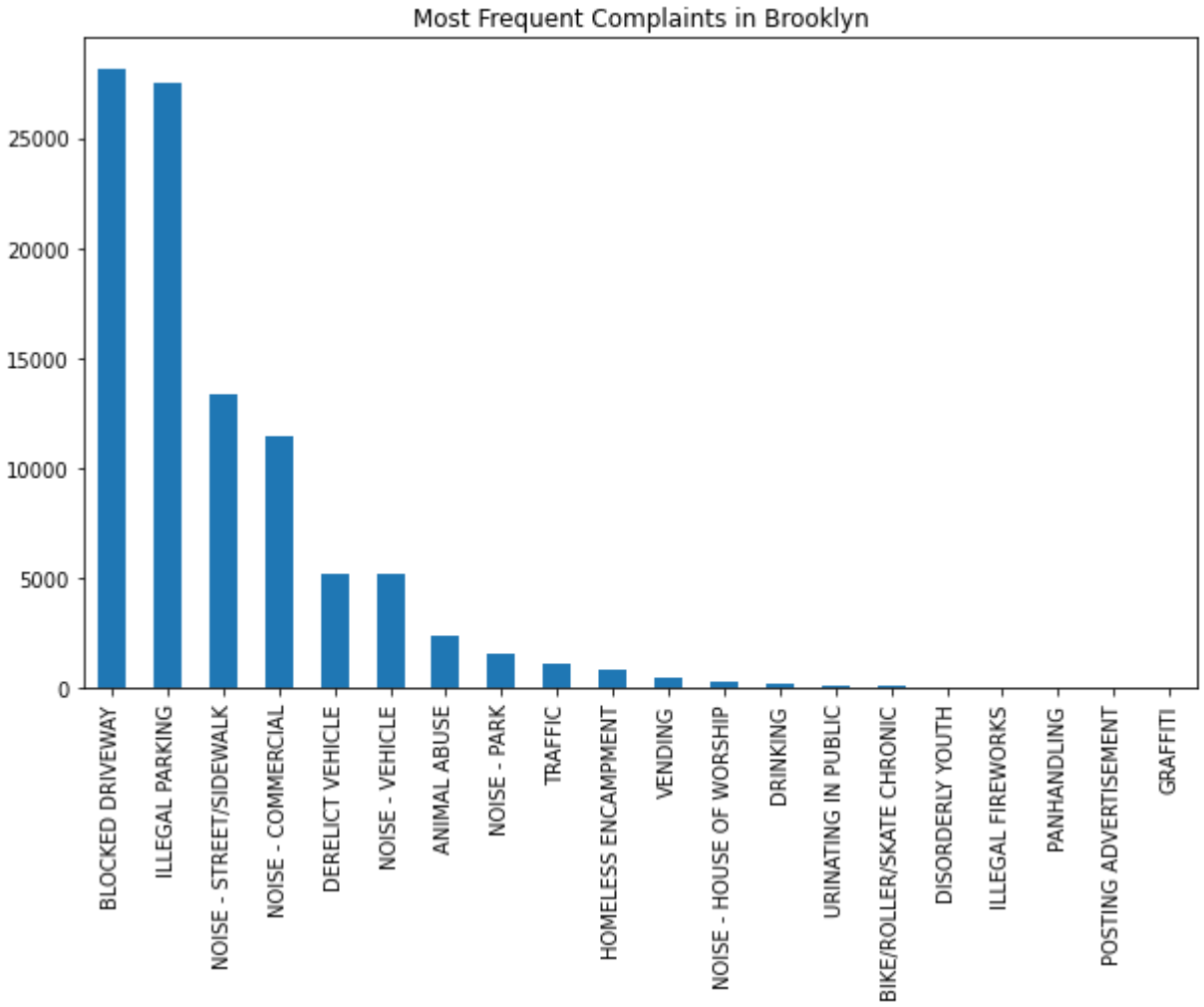
2)This Conclusion indicates that Brooklyn has more complaints with (32.7%) and after that Queens (26.8%)

In [28]:
```
#Conclusion - 3 Analysis for Brooklyn borough which has highest number of complains

df_Brooklyn = customer_service_request[customer_service_request['Borough']=='BROOKLYN']
```

In [29]:
```
(df_Brooklyn['Complaint Type'].value_counts()).head(25).plot(kind='bar',figsize=(10,6),
```

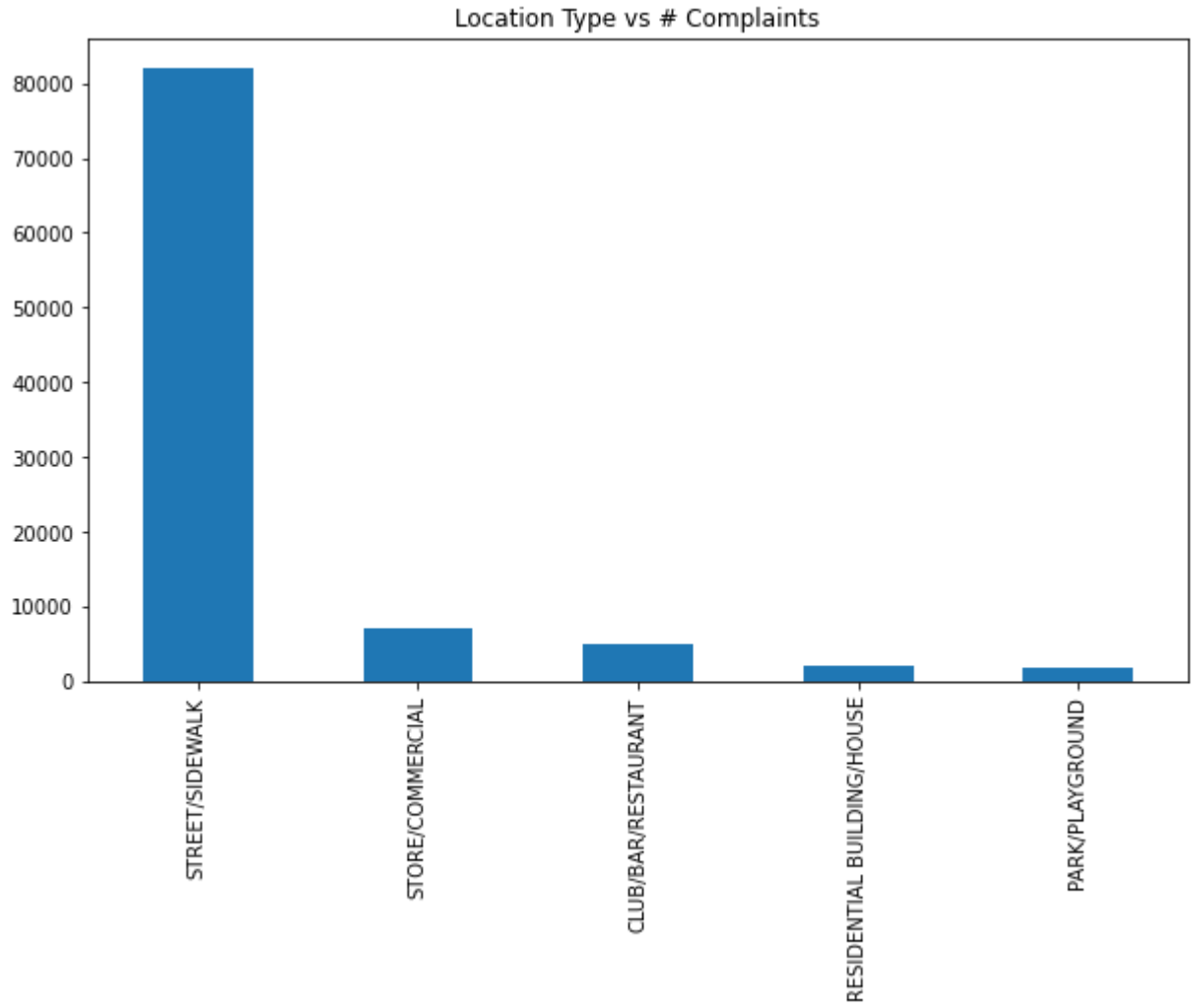
Out[29]: <AxesSubplot:title={'center':'Most Frequent Complaints in Brooklyn'}>



3) Blocked Driveway and Illegal Parking is the Most Frequent Complaints in Brooklyn

In [30]:
```
#Conclusion - 4 location type vs complaints
(df_Brooklyn['Location Type'].value_counts()).head().plot(kind='bar',
                                                          figsize=(10,6),title = 'Location Ty
```

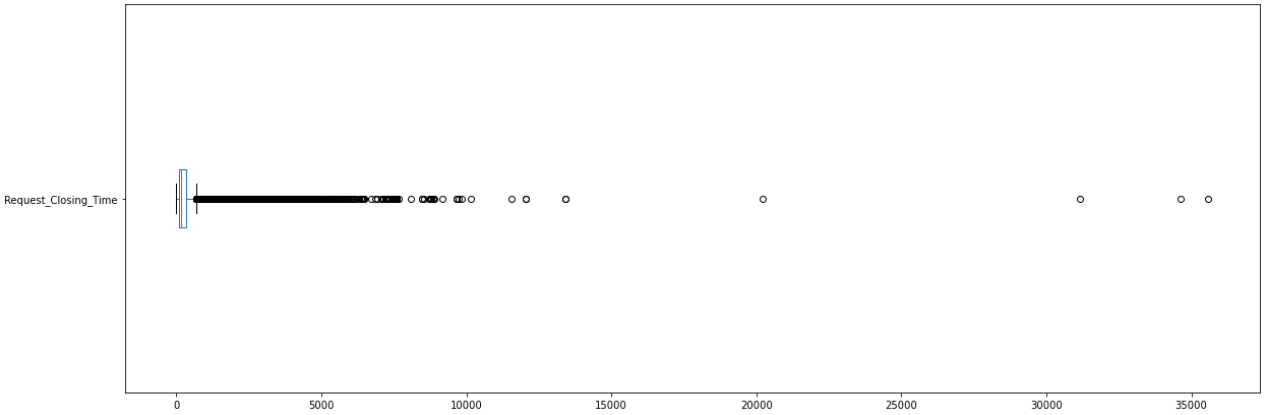
Out[30]:
```
<AxesSubplot:title={'center':'Location Type vs # Complaints'}>
```



4) Street/Sidewalk has more complaints

In [31]:
```
#There are some outliers that means some of the requests taken longer time than the avg
closed_status_requests.Request_Closing_Time.plot(kind='box', vert=False)
```

Out[31]: <AxesSubplot:>



*Task 4 - Order the complaint types based on the average 'Request_Closing_Time', grouping them for different locations.

In [32]:
```
order_data= closed_status_requests.groupby(['City','Complaint Type']).Request_Closing_T
df = order_data.to_frame()
```

```
df = df.sort_values(by=['City','Request_Closing_Time'],ascending=False)
df.shape
```

Out[32]: (747, 1)

In [33]:
```
#Storing the sorted results
#df.to_csv('sorted_data.csv')
with pd.option_context('display.max_rows', None, 'display.max_columns', None):
    print(df)
```

City	Complaint Type	Request_Closing_Time
WOODSIDE	BIKE/ROLLER/SKATE CHRONIC	728.500000
	DERELICT VEHICLE	560.562249
	GRAFFITI	539.333333
	ANIMAL ABUSE	506.101449
	VENDING	437.800000
	ILLEGAL PARKING	422.168517
	NOISE - PARK	404.710526
	HOMELESS ENCAMPMENT	402.696970
	NOISE - COMMERCIAL	398.445498
	NOISE - STREET/SIDEWALK	393.329412
	BLOCKED DRIVEWAY	388.239063
	URINATING IN PUBLIC	385.125000
	NOISE - VEHICLE	328.590476
	DRINKING	328.466667
	TRAFFIC	289.923077
	NOISE - HOUSE OF WORSHIP	284.000000
	ILLEGAL FIREWORKS	148.000000
	DISORDERLY YOUTH	73.000000
WOODHAVEN	DERELICT VEHICLE	446.672078
	HOMELESS ENCAMPMENT	426.000000
	ILLEGAL PARKING	343.778267
	BLOCKED DRIVEWAY	331.081209
	NOISE - STREET/SIDEWALK	314.000000
	ANIMAL ABUSE	297.844444
	NOISE - COMMERCIAL	233.205714
	URINATING IN PUBLIC	204.500000
	NOISE - VEHICLE	203.932432
	NOISE - HOUSE OF WORSHIP	198.333333
	DRINKING	171.666667
	VENDING	170.166667
	TRAFFIC	109.666667
	NOISE - PARK	82.666667
	BIKE/ROLLER/SKATE CHRONIC	74.500000
WHITESTONE	GRAFFITI	528.000000
	NOISE - COMMERCIAL	266.562500
	ILLEGAL FIREWORKS	262.000000
	DRINKING	217.500000
	NOISE - PARK	217.500000
	NOISE - STREET/SIDEWALK	207.030303
	DERELICT VEHICLE	204.555066
	NOISE - VEHICLE	201.642857
	DISORDERLY YOUTH	195.000000
	ILLEGAL PARKING	190.266667
	BLOCKED DRIVEWAY	188.913462
	BIKE/ROLLER/SKATE CHRONIC	188.500000
	ANIMAL ABUSE	168.107143
	TRAFFIC	162.823529
	VENDING	140.000000
SUNNYSIDE	ANIMAL ABUSE	692.800000
	DERELICT VEHICLE	589.100000

	VENDING	583.000000
	POSTING ADVERTISEMENT	435.500000
	HOMELESS ENCAMPMENT	431.090909
	BLOCKED DRIVEWAY	417.752427
	ILLEGAL PARKING	399.663934
	NOISE - COMMERCIAL	397.099379
	NOISE - PARK	387.533333
	NOISE - STREET/SIDEWALK	373.646154
	TRAFFIC	356.875000
	NOISE - VEHICLE	295.041667
	DRINKING	276.000000
	DISORDERLY YOUTH	224.000000
	URINATING IN PUBLIC	204.000000
	BIKE/ROLLER/SKATE CHRONIC	72.000000
	GRAFFITI	35.000000
STATEN ISLAND	GRAFFITI	573.500000
	DERELICT VEHICLE	302.134278
	HOMELESS ENCAMPMENT	298.464789
	ANIMAL ABUSE	297.874327
	PANHANDLING	280.000000
	VENDING	256.000000
	BIKE/ROLLER/SKATE CHRONIC	244.428571
	BLOCKED DRIVEWAY	243.966853
	DISORDERLY YOUTH	233.304348
	ILLEGAL PARKING	230.471853
	ILLEGAL FIREWORKS	223.900000
	TRAFFIC	214.190955
	DRINKING	209.251429
	NOISE - VEHICLE	193.946629
	NOISE - COMMERCIAL	179.722304
	NOISE - STREET/SIDEWALK	177.844363
	NOISE - PARK	175.805970
	NOISE - HOUSE OF WORSHIP	149.882353
	URINATING IN PUBLIC	148.428571
	POSTING ADVERTISEMENT	92.225243
SPRINGFIELD GARDENS	POSTING ADVERTISEMENT	1170.000000
	ANIMAL ABUSE	891.375000
	DERELICT VEHICLE	702.411483
	TRAFFIC	620.363636
	ILLEGAL PARKING	550.428571
	BLOCKED DRIVEWAY	541.862595
	VENDING	411.000000
	PANHANDLING	381.000000
	ILLEGAL FIREWORKS	333.000000
	NOISE - VEHICLE	302.476190
	DRINKING	290.333333
	URINATING IN PUBLIC	287.000000
	NOISE - STREET/SIDEWALK	252.789474
	HOMELESS ENCAMPMENT	226.600000
	NOISE - COMMERCIAL	223.138889
	NOISE - HOUSE OF WORSHIP	222.000000
	NOISE - PARK	100.000000
SOUTH RICHMOND HILL	BIKE/ROLLER/SKATE CHRONIC	1854.000000
	DERELICT VEHICLE	712.311419
	TRAFFIC	346.272727
	DRINKING	345.739130
	ILLEGAL PARKING	342.190476
	HOMELESS ENCAMPMENT	319.909091
	VENDING	315.750000
	BLOCKED DRIVEWAY	291.661499
	ANIMAL ABUSE	261.807692
	NOISE - STREET/SIDEWALK	247.032967
	NOISE - COMMERCIAL	232.065657
	NOISE - VEHICLE	216.160494
	NOISE - PARK	215.500000

	DISORDERLY YOUTH	183.500000
	NOISE - HOUSE OF WORSHIP	119.000000
	ILLEGAL FIREWORKS	28.500000
SOUTH OZONE PARK	DERELICT VEHICLE	623.606145
	NOISE - PARK	342.250000
	DRINKING	292.538462
	ILLEGAL PARKING	281.447368
	BLOCKED DRIVEWAY	265.338641
	BIKE/ROLLER/SKATE CHRONIC	243.000000
	NOISE - VEHICLE	227.282353
	NOISE - STREET/SIDEWALK	219.371429
	TRAFFIC	213.071429
	VENDING	210.800000
	ANIMAL ABUSE	209.218182
	NOISE - COMMERCIAL	200.142857
	DISORDERLY YOUTH	160.500000
	NOISE - HOUSE OF WORSHIP	129.333333
	HOMELESS ENCAMPMENT	109.750000
	URINATING IN PUBLIC	101.500000
	POSTING ADVERTISEMENT	77.000000
	ILLEGAL FIREWORKS	30.000000
SAINT ALBANS	HOMELESS ENCAMPMENT	453.875000
	ANIMAL ABUSE	441.933333
	URINATING IN PUBLIC	390.000000
	DERELICT VEHICLE	353.722772
	BLOCKED DRIVEWAY	280.872951
	NOISE - HOUSE OF WORSHIP	278.000000
	ILLEGAL PARKING	250.712707
	TRAFFIC	237.727273
	NOISE - VEHICLE	207.195122
	NOISE - STREET/SIDEWALK	201.658228
	NOISE - COMMERCIAL	185.758621
	VENDING	143.000000
	DISORDERLY YOUTH	105.000000
	DRINKING	57.333333
	NOISE - PARK	49.000000
ROSEDALE	HOMELESS ENCAMPMENT	1131.250000
	ANIMAL ABUSE	875.333333
	DERELICT VEHICLE	870.149038
	TRAFFIC	631.913043
	ILLEGAL PARKING	613.191336
	BIKE/ROLLER/SKATE CHRONIC	511.500000
	NOISE - COMMERCIAL	465.280000
	BLOCKED DRIVEWAY	447.170616
	VENDING	407.625000
	NOISE - STREET/SIDEWALK	352.083333
	NOISE - VEHICLE	350.600000
	NOISE - PARK	346.376812
	DRINKING	330.000000
	NOISE - HOUSE OF WORSHIP	270.000000
	GRAFFITI	9.000000
ROCKAWAY PARK	TRAFFIC	213.142857
	DISORDERLY YOUTH	205.000000
	VENDING	176.500000
	DERELICT VEHICLE	163.555556
	ILLEGAL PARKING	156.958991
	BLOCKED DRIVEWAY	144.728571
	DRINKING	137.950000
	ANIMAL ABUSE	136.700000
	NOISE - VEHICLE	135.185185
	NOISE - COMMERCIAL	120.555556
	NOISE - STREET/SIDEWALK	109.698413
	HOMELESS ENCAMPMENT	102.750000
	NOISE - PARK	61.500000
	URINATING IN PUBLIC	43.000000

RIDGEWOOD	DERELICT VEHICLE	454.278788
	ANIMAL ABUSE	385.418803
	HOMELESS ENCAMPMENT	346.347826
	ILLEGAL PARKING	265.799674
	NOISE - VEHICLE	260.714286
	DRINKING	260.600000
	URINATING IN PUBLIC	248.625000
	NOISE - STREET/SIDEWALK	247.411085
	BLOCKED DRIVEWAY	240.255759
	DISORDERLY YOUTH	239.666667
	NOISE - COMMERCIAL	220.690955
	TRAFFIC	217.761905
	NOISE - PARK	215.357143
	BIKE/ROLLER/SKATE CHRONIC	208.333333
	NOISE - HOUSE OF WORSHIP	184.000000
	VENDING	182.625000
	ILLEGAL FIREWORKS	64.500000
	GRAFFITI	59.000000
	POSTING ADVERTISEMENT	14.000000
RICHMOND HILL	DERELICT VEHICLE	575.240964
	HOMELESS ENCAMPMENT	484.785714
	URINATING IN PUBLIC	376.800000
	VENDING	348.076923
	ILLEGAL PARKING	346.421320
	ANIMAL ABUSE	332.093750
	NOISE - STREET/SIDEWALK	308.632184
	BLOCKED DRIVEWAY	302.847302
	NOISE - COMMERCIAL	245.064815
	NOISE - VEHICLE	244.312500
	ILLEGAL FIREWORKS	208.750000
	TRAFFIC	195.571429
	POSTING ADVERTISEMENT	180.000000
	DRINKING	176.888889
	GRAFFITI	145.000000
	NOISE - PARK	134.250000
REGO PARK	NOISE - HOUSE OF WORSHIP	558.000000
	URINATING IN PUBLIC	335.000000
	VENDING	306.333333
	DERELICT VEHICLE	272.716049
	ANIMAL ABUSE	265.153846
	TRAFFIC	225.571429
	NOISE - PARK	224.772727
	BLOCKED DRIVEWAY	215.875614
	ILLEGAL PARKING	201.718808
	NOISE - VEHICLE	176.534884
	NOISE - STREET/SIDEWALK	155.982456
	DRINKING	153.250000
	NOISE - COMMERCIAL	143.253165
	HOMELESS ENCAMPMENT	91.333333
	GRAFFITI	56.000000
QUEENS VILLAGE	GRAFFITI	3200.000000
	DERELICT VEHICLE	964.313514
	VENDING	875.500000
	ANIMAL ABUSE	768.954545
	NOISE - COMMERCIAL	602.000000
	ILLEGAL PARKING	595.823529
	BLOCKED DRIVEWAY	571.994872
	HOMELESS ENCAMPMENT	562.666667
	PANHANDLING	543.000000
	NOISE - VEHICLE	532.414634
	TRAFFIC	425.615385
	URINATING IN PUBLIC	380.000000
	NOISE - STREET/SIDEWALK	334.590909
	DRINKING	280.600000
	NOISE - PARK	211.000000

	ILLEGAL FIREWORKS	197.000000
	POSTING ADVERTISEMENT	183.000000
	NOISE - HOUSE OF WORSHIP	150.500000
QUEENS	ANIMAL IN A PARK	20210.000000
	DERELICT VEHICLE	521.000000
	HOMELESS ENCAMPMENT	432.000000
	BLOCKED DRIVEWAY	239.500000
	NOISE - STREET/SIDEWALK	219.000000
	ILLEGAL PARKING	205.000000
	NOISE - HOUSE OF WORSHIP	156.000000
	TRAFFIC	126.500000
	NOISE - VEHICLE	79.500000
	NOISE - COMMERCIAL	79.166667
	URINATING IN PUBLIC	21.000000
OZONE PARK	DERELICT VEHICLE	640.380952
	BIKE/ROLLER/SKATE CHRONIC	431.000000
	ILLEGAL PARKING	306.898223
	POSTING ADVERTISEMENT	298.000000
	ANIMAL ABUSE	297.500000
	BLOCKED DRIVEWAY	297.184420
	PANHANDLING	274.571429
	URINATING IN PUBLIC	263.500000
	TRAFFIC	260.263158
	VENDING	256.000000
	DRINKING	236.421053
	NOISE - COMMERCIAL	234.217391
	NOISE - STREET/SIDEWALK	229.080292
	NOISE - VEHICLE	200.126761
	NOISE - PARK	174.055556
	DISORDERLY YOUTH	131.750000
	HOMELESS ENCAMPMENT	117.500000
	NOISE - HOUSE OF WORSHIP	50.000000
	ILLEGAL FIREWORKS	19.000000
OAKLAND GARDENS	HOMELESS ENCAMPMENT	1719.000000
	VENDING	227.000000
	DERELICT VEHICLE	222.802326
	ANIMAL ABUSE	165.526316
	NOISE - PARK	164.714286
	BLOCKED DRIVEWAY	152.477273
	DRINKING	146.000000
	TRAFFIC	144.333333
	NOISE - VEHICLE	138.600000
	ILLEGAL PARKING	137.273764
	NOISE - STREET/SIDEWALK	98.421053
	DISORDERLY YOUTH	85.000000
	BIKE/ROLLER/SKATE CHRONIC	67.000000
NEW YORK	GRAFFITI	303.545455
	DERELICT VEHICLE	255.636872
	SQUEEGEE	242.500000
	ANIMAL ABUSE	220.807869
	HOMELESS ENCAMPMENT	220.795381
	BLOCKED DRIVEWAY	213.216425
	PANHANDLING	208.155440
	ILLEGAL PARKING	203.007341
	VENDING	198.110555
	DRINKING	183.159322
	POSTING ADVERTISEMENT	177.048780
	NOISE - PARK	176.380165
	URINATING IN PUBLIC	173.338645
	BIKE/ROLLER/SKATE CHRONIC	173.226667
	NOISE - COMMERCIAL	163.794211
	NOISE - STREET/SIDEWALK	163.690022
	TRAFFIC	158.785252
	NOISE - VEHICLE	157.015682
	DISORDERLY YOUTH	144.130435

	NOISE - HOUSE OF WORSHIP	138.020619
	ILLEGAL FIREWORKS	102.944444
NEW HYDE PARK	DERELICT VEHICLE	467.714286
	BLOCKED DRIVEWAY	464.018868
	ILLEGAL PARKING	455.178571
	NOISE - VEHICLE	200.000000
	ANIMAL ABUSE	115.000000
MIDDLE VILLAGE	BIKE/ROLLER/SKATE CHRONIC	940.000000
	DERELICT VEHICLE	494.395270
	ANIMAL ABUSE	412.545455
	NOISE - COMMERCIAL	306.200000
	ILLEGAL PARKING	299.583808
	NOISE - STREET/SIDEWALK	293.378378
	NOISE - VEHICLE	284.214286
	NOISE - PARK	272.500000
	BLOCKED DRIVEWAY	261.838074
	HOMELESS ENCAMPMENT	260.800000
	TRAFFIC	211.416667
	DRINKING	74.000000
MASPETH	NOISE - PARK	661.666667
	BIKE/ROLLER/SKATE CHRONIC	530.000000
	NOISE - HOUSE OF WORSHIP	474.000000
	DERELICT VEHICLE	463.519630
	ANIMAL ABUSE	436.444444
	HOMELESS ENCAMPMENT	430.600000
	ILLEGAL PARKING	329.421107
	NOISE - COMMERCIAL	295.269231
	BLOCKED DRIVEWAY	288.632514
	VENDING	277.833333
	NOISE - VEHICLE	275.789474
	NOISE - STREET/SIDEWALK	252.280992
	DISORDERLY YOUTH	240.500000
	TRAFFIC	224.890909
	URINATING IN PUBLIC	221.000000
	DRINKING	139.333333
	ILLEGAL FIREWORKS	97.000000
LONG ISLAND CITY	DERELICT VEHICLE	626.854271
	VENDING	556.500000
	ILLEGAL PARKING	438.230769
	HOMELESS ENCAMPMENT	420.700000
	BIKE/ROLLER/SKATE CHRONIC	405.333333
	ANIMAL ABUSE	363.266667
	BLOCKED DRIVEWAY	357.853598
	DISORDERLY YOUTH	329.000000
	NOISE - PARK	319.685185
	NOISE - VEHICLE	303.457944
	NOISE - COMMERCIAL	267.024194
	URINATING IN PUBLIC	251.333333
	NOISE - STREET/SIDEWALK	248.389262
	PANHANDLING	233.500000
	GRAFFITI	210.000000
	TRAFFIC	202.763889
	DRINKING	181.571429
	POSTING ADVERTISEMENT	49.000000
LITTLE NECK	DISORDERLY YOUTH	260.500000
	DERELICT VEHICLE	215.508197
	URINATING IN PUBLIC	184.000000
	ILLEGAL PARKING	170.710843
	BLOCKED DRIVEWAY	144.743802
	ANIMAL ABUSE	137.866667
	POSTING ADVERTISEMENT	134.000000
	TRAFFIC	131.764706
	NOISE - VEHICLE	128.600000
	NOISE - STREET/SIDEWALK	118.625000
	DRINKING	94.000000

	NOISE - COMMERCIAL	80.921053
	NOISE - PARK	64.000000
KEW GARDENS	URINATING IN PUBLIC	431.333333
	DERELICT VEHICLE	431.142857
	VENDING	373.000000
	BLOCKED DRIVEWAY	367.479233
	ILLEGAL PARKING	284.896226
	NOISE - COMMERCIAL	231.115854
	TRAFFIC	227.400000
	NOISE - VEHICLE	217.333333
	ANIMAL ABUSE	189.684211
	NOISE - STREET/SIDEWALK	154.700000
	HOMELESS ENCAMPMENT	113.800000
	NOISE - HOUSE OF WORSHIP	69.000000
	DRINKING	61.000000
JAMAICA	HOMELESS ENCAMPMENT	484.379747
	DERELICT VEHICLE	476.499475
	VENDING	450.350000
	POSTING ADVERTISEMENT	329.000000
	GRAFFITI	324.666667
	NOISE - HOUSE OF WORSHIP	320.769231
	URINATING IN PUBLIC	306.242424
	ILLEGAL PARKING	304.430683
	DISORDERLY YOUTH	304.375000
	ANIMAL ABUSE	303.253275
	BLOCKED DRIVEWAY	298.505682
	TRAFFIC	282.105357
	DRINKING	264.441176
	NOISE - COMMERCIAL	235.344262
	NOISE - STREET/SIDEWALK	220.649701
	NOISE - VEHICLE	207.509934
	NOISE - PARK	190.578947
	BIKE/ROLLER/SKATE CHRONIC	174.500000
	ILLEGAL FIREWORKS	156.250000
	PANHANDLING	104.000000
JACKSON HEIGHTS	DRINKING	343.444444
	BIKE/ROLLER/SKATE CHRONIC	247.000000
	ANIMAL ABUSE	243.285714
	URINATING IN PUBLIC	229.000000
	DERELICT VEHICLE	225.724138
	BLOCKED DRIVEWAY	224.052817
	VENDING	222.564103
	ILLEGAL PARKING	218.644809
	TRAFFIC	194.538462
	NOISE - COMMERCIAL	177.129590
	HOMELESS ENCAMPMENT	156.454545
	NOISE - PARK	144.875000
	NOISE - STREET/SIDEWALK	139.018433
	NOISE - VEHICLE	132.431034
	PANHANDLING	101.000000
	POSTING ADVERTISEMENT	83.000000
	NOISE - HOUSE OF WORSHIP	66.000000
	ILLEGAL FIREWORKS	42.000000
HOWARD BEACH	DERELICT VEHICLE	688.318841
	NOISE - HOUSE OF WORSHIP	408.000000
	NOISE - COMMERCIAL	349.782946
	ILLEGAL PARKING	345.145907
	ANIMAL ABUSE	264.354839
	BLOCKED DRIVEWAY	254.857143
	DISORDERLY YOUTH	251.000000
	NOISE - STREET/SIDEWALK	236.428571
	NOISE - PARK	232.500000
	PANHANDLING	198.000000
	TRAFFIC	185.333333
	NOISE - VEHICLE	184.600000

	HOMELESS ENCAMPMENT	158.666667
	BIKE/ROLLER/SKATE CHRONIC	158.000000
	DRINKING	120.000000
	ILLEGAL FIREWORKS	110.333333
	VENDING	108.000000
HOLLIS	DERELICT VEHICLE	693.629371
	DRINKING	437.333333
	ILLEGAL PARKING	407.278146
	NOISE - COMMERCIAL	402.520000
	BLOCKED DRIVEWAY	287.406433
	ANIMAL ABUSE	261.969697
	HOMELESS ENCAMPMENT	256.777778
	NOISE - HOUSE OF WORSHIP	242.839572
	URINATING IN PUBLIC	237.000000
	DISORDERLY YOUTH	232.000000
	TRAFFIC	228.727273
	NOISE - PARK	210.588235
	NOISE - VEHICLE	191.148936
	NOISE - STREET/SIDEWALK	169.658537
GLEN OAKS	DERELICT VEHICLE	914.428571
	ANIMAL ABUSE	779.400000
	NOISE - STREET/SIDEWALK	714.500000
	NOISE - VEHICLE	685.250000
	BLOCKED DRIVEWAY	677.566667
	URINATING IN PUBLIC	660.500000
	ILLEGAL PARKING	537.675676
	NOISE - COMMERCIAL	366.679487
	TRAFFIC	332.666667
	VENDING	290.388889
	NOISE - PARK	263.891892
FRESH MEADOWS	HOMELESS ENCAMPMENT	353.600000
	DERELICT VEHICLE	270.415808
	BLOCKED DRIVEWAY	237.675299
	DRINKING	223.500000
	NOISE - PARK	204.750000
	ANIMAL ABUSE	196.511111
	VENDING	157.000000
	ILLEGAL PARKING	156.692308
	NOISE - STREET/SIDEWALK	147.857143
	NOISE - COMMERCIAL	144.142857
	NOISE - VEHICLE	128.488636
	TRAFFIC	97.538462
	URINATING IN PUBLIC	96.000000
	PANHANDLING	94.000000
FOREST HILLS	POSTING ADVERTISEMENT	351.666667
	PANHANDLING	348.600000
	BIKE/ROLLER/SKATE CHRONIC	289.200000
	DISORDERLY YOUTH	249.000000
	DRINKING	246.000000
	NOISE - HOUSE OF WORSHIP	234.000000
	DERELICT VEHICLE	223.250000
	BLOCKED DRIVEWAY	222.594268
	HOMELESS ENCAMPMENT	210.277778
	ILLEGAL PARKING	199.158416
	GRAFFITI	196.333333
	ANIMAL ABUSE	195.488889
	VENDING	164.400000
	NOISE - VEHICLE	142.719298
	NOISE - STREET/SIDEWALK	138.936842
	TRAFFIC	120.116667
	NOISE - COMMERCIAL	112.531915
	NOISE - PARK	92.050000
	URINATING IN PUBLIC	82.000000
	ILLEGAL FIREWORKS	40.000000
FLUSHING	POSTING ADVERTISEMENT	368.000000

	BIKE/ROLLER/SKATE CHRONIC	343.666667
	DERELICT VEHICLE	221.268182
	NOISE - HOUSE OF WORSHIP	218.000000
	ANIMAL ABUSE	213.711268
	NOISE - VEHICLE	201.310078
	DRINKING	182.775000
	HOMELESS ENCAMPMENT	180.000000
	ILLEGAL PARKING	178.616184
	BLOCKED DRIVEWAY	176.793770
	NOISE - PARK	173.120690
	NOISE - COMMERCIAL	170.417143
	NOISE - STREET/SIDEWALK	169.520000
	VENDING	146.666667
	DISORDERLY YOUTH	128.000000
	TRAFFIC	123.382979
	GRAFFITI	103.250000
	URINATING IN PUBLIC	73.166667
	PANHANDLING	68.500000
	ILLEGAL FIREWORKS	41.500000
FLORAL PARK	ANIMAL ABUSE	1594.000000
	DERELICT VEHICLE	999.017857
	ILLEGAL PARKING	554.984375
	DRINKING	494.000000
	BLOCKED DRIVEWAY	461.750000
	NOISE - STREET/SIDEWALK	413.666667
	NOISE - COMMERCIAL	261.666667
	DISORDERLY YOUTH	195.000000
	NOISE - VEHICLE	117.000000
FAR ROCKAWAY	DERELICT VEHICLE	219.614973
	DISORDERLY YOUTH	201.000000
	NOISE - STREET/SIDEWALK	181.992647
	HOMELESS ENCAMPMENT	180.642857
	VENDING	168.000000
	ANIMAL ABUSE	162.741573
	ILLEGAL PARKING	162.040678
	BLOCKED DRIVEWAY	157.795775
	DRINKING	144.750000
	TRAFFIC	142.900000
	NOISE - VEHICLE	130.506494
	NOISE - COMMERCIAL	115.270833
	NOISE - PARK	90.086957
	URINATING IN PUBLIC	90.000000
	NOISE - HOUSE OF WORSHIP	68.000000
ELMHURST	DERELICT VEHICLE	288.833333
	BIKE/ROLLER/SKATE CHRONIC	277.500000
	VENDING	239.285714
	ANIMAL ABUSE	231.921053
	HOMELESS ENCAMPMENT	220.218750
	BLOCKED DRIVEWAY	205.694329
	PANHANDLING	197.666667
	ILLEGAL PARKING	196.440323
	DRINKING	183.307692
	NOISE - COMMERCIAL	167.086420
	TRAFFIC	157.428571
	NOISE - VEHICLE	157.297872
	URINATING IN PUBLIC	152.000000
	NOISE - STREET/SIDEWALK	150.263393
	NOISE - PARK	138.617647
	NOISE - HOUSE OF WORSHIP	112.800000
	ILLEGAL FIREWORKS	59.000000
	DISORDERLY YOUTH	51.000000
	POSTING ADVERTISEMENT	44.000000
EAST ELMHURST	GRAFFITI	458.333333
	DISORDERLY YOUTH	414.000000
	HOMELESS ENCAMPMENT	378.000000

	DERELICT VEHICLE	342.921053
	ANIMAL ABUSE	243.016949
	VENDING	241.000000
	BLOCKED DRIVEWAY	223.690341
	ILLEGAL PARKING	204.479190
	TRAFFIC	161.050000
	NOISE - PARK	159.000000
	NOISE - STREET/SIDEWALK	148.158879
	NOISE - COMMERCIAL	140.857143
	NOISE - VEHICLE	129.786885
	URINATING IN PUBLIC	125.200000
	DRINKING	111.111111
	POSTING ADVERTISEMENT	111.000000
	NOISE - HOUSE OF WORSHIP	86.000000
	BIKE/ROLLER/SKATE CHRONIC	15.000000
CORONA	DERELICT VEHICLE	274.842105
	NOISE - HOUSE OF WORSHIP	225.000000
	HOMELESS ENCAMPMENT	219.894737
	ANIMAL ABUSE	219.868852
	DRINKING	218.545455
	ILLEGAL PARKING	201.371212
	BLOCKED DRIVEWAY	198.939515
	VENDING	185.774194
	DISORDERLY YOUTH	172.333333
	NOISE - VEHICLE	159.350000
	NOISE - COMMERCIAL	157.004032
	NOISE - STREET/SIDEWALK	142.126582
	NOISE - PARK	139.000000
	TRAFFIC	136.916667
	URINATING IN PUBLIC	117.000000
	POSTING ADVERTISEMENT	92.000000
	PANHANDLING	70.000000
	GRAFFITI	44.000000
COLLEGE POINT	GRAFFITI	711.000000
	VENDING	279.000000
	ANIMAL ABUSE	278.428571
	NOISE - COMMERCIAL	238.114286
	DERELICT VEHICLE	210.809783
	BLOCKED DRIVEWAY	202.501149
	NOISE - PARK	190.500000
	ILLEGAL PARKING	187.198864
	NOISE - STREET/SIDEWALK	180.666667
	NOISE - VEHICLE	158.732824
	TRAFFIC	137.357143
	HOMELESS ENCAMPMENT	86.333333
	DISORDERLY YOUTH	36.000000
CENTRAL PARK	ILLEGAL PARKING	275.000000
	NOISE - STREET/SIDEWALK	195.747368
CAMBRIA HEIGHTS	HOMELESS ENCAMPMENT	1367.000000
	DERELICT VEHICLE	966.347826
	ANIMAL ABUSE	680.909091
	ILLEGAL PARKING	674.328947
	TRAFFIC	512.333333
	BLOCKED DRIVEWAY	461.517007
	NOISE - VEHICLE	414.792208
	NOISE - STREET/SIDEWALK	276.160000
	NOISE - COMMERCIAL	228.250000
	NOISE - HOUSE OF WORSHIP	158.000000
	ILLEGAL FIREWORKS	91.000000
BROOKLYN	GRAFFITI	494.232558
	DERELICT VEHICLE	360.920191
	AGENCY ISSUES	315.333333
	BIKE/ROLLER/SKATE CHRONIC	300.884956
	ANIMAL ABUSE	289.598162
	HOMELESS ENCAMPMENT	281.410047

	VENDING	271.295146
	BLOCKED DRIVEWAY	264.795297
	PANHANDLING	258.673469
	ILLEGAL PARKING	258.533008
	DISORDERLY YOUTH	248.680556
	URINATING IN PUBLIC	233.698529
	DRINKING	214.246154
	POSTING ADVERTISEMENT	201.422222
	NOISE - STREET/SIDEWALK	198.943916
	NOISE - VEHICLE	196.810691
	NOISE - PARK	189.916720
	TRAFFIC	187.090156
	NOISE - HOUSE OF WORSHIP	183.873529
	NOISE - COMMERCIAL	180.791753
	ILLEGAL FIREWORKS	140.180328
BRONX	PANHANDLING	852.684211
	DERELICT VEHICLE	553.342725
	GRAFFITI	533.666667
	HOMELESS ENCAMPMENT	446.222672
	ANIMAL ABUSE	439.826855
	VENDING	409.245383
	ILLEGAL PARKING	394.493445
	BLOCKED DRIVEWAY	375.418948
	DRINKING	347.297872
	ILLEGAL FIREWORKS	336.333333
	NOISE - VEHICLE	333.379676
	URINATING IN PUBLIC	323.196078
	NOISE - STREET/SIDEWALK	313.278965
	TRAFFIC	295.121127
	NOISE - PARK	281.606947
	NOISE - COMMERCIAL	281.542951
	NOISE - HOUSE OF WORSHIP	273.265823
	DISORDERLY YOUTH	254.015873
	POSTING ADVERTISEMENT	213.250000
	BIKE/ROLLER/SKATE CHRONIC	207.300000
BREEZY POINT	DERELICT VEHICLE	428.333333
	ILLEGAL PARKING	236.266667
	DRINKING	158.000000
	ANIMAL ABUSE	156.500000
	NOISE - COMMERCIAL	152.000000
	BLOCKED DRIVEWAY	79.333333
	NOISE - VEHICLE	79.000000
	NOISE - STREET/SIDEWALK	60.000000
BELLEROSE	HOMELESS ENCAMPMENT	2348.000000
	DERELICT VEHICLE	1029.764045
	ANIMAL ABUSE	763.142857
	BLOCKED DRIVEWAY	605.726316
	NOISE - STREET/SIDEWALK	543.846154
	ILLEGAL PARKING	491.849057
	URINATING IN PUBLIC	452.000000
	PANHANDLING	449.000000
	NOISE - COMMERCIAL	404.243243
	ILLEGAL FIREWORKS	400.000000
	TRAFFIC	345.142857
	BIKE/ROLLER/SKATE CHRONIC	294.000000
	DRINKING	235.000000
	NOISE - VEHICLE	154.700000
	POSTING ADVERTISEMENT	135.000000
	NOISE - HOUSE OF WORSHIP	131.000000
	DISORDERLY YOUTH	111.000000
	NOISE - PARK	84.000000
BAYSIDE	GRAFFITI	272.666667
	NOISE - HOUSE OF WORSHIP	212.000000
	DERELICT VEHICLE	201.242424
	NOISE - PARK	196.250000

	ANIMAL ABUSE	196.135135
	DISORDERLY YOUTH	178.000000
	HOMELESS ENCAMPMENT	172.500000
	BLOCKED DRIVEWAY	153.718085
	ILLEGAL PARKING	153.606238
	NOISE - COMMERCIAL	133.725000
	DRINKING	114.000000
	VENDING	112.500000
	NOISE - VEHICLE	102.312500
	NOISE - STREET/SIDEWALK	91.666667
	TRAFFIC	91.222222
ASTORIA	GRAFFITI	845.000000
	DERELICT VEHICLE	574.206612
	POSTING ADVERTISEMENT	352.000000
	TRAFFIC	324.297872
	ANIMAL ABUSE	299.664000
	VENDING	295.851852
	HOMELESS ENCAMPMENT	294.781250
	BLOCKED DRIVEWAY	288.893197
	ILLEGAL PARKING	288.491803
	DRINKING	283.000000
	URINATING IN PUBLIC	277.111111
	NOISE - STREET/SIDEWALK	210.402000
	NOISE - VEHICLE	210.220588
	NOISE - COMMERCIAL	191.805663
	NOISE - PARK	179.409836
	DISORDERLY YOUTH	173.666667
	ILLEGAL FIREWORKS	166.250000
	NOISE - HOUSE OF WORSHIP	120.947368
	BIKE/ROLLER/SKATE CHRONIC	104.133333
	PANHANDLING	69.000000
ARVERNE	DISORDERLY YOUTH	215.000000
	DERELICT VEHICLE	177.740741
	BLOCKED DRIVEWAY	151.200000
	ILLEGAL PARKING	138.724138
	NOISE - COMMERCIAL	136.500000
	ANIMAL ABUSE	128.894737
	NOISE - STREET/SIDEWALK	119.275862
	NOISE - VEHICLE	111.142857
	HOMELESS ENCAMPMENT	108.250000
	NOISE - HOUSE OF WORSHIP	93.272727
	GRAFFITI	92.000000
	NOISE - PARK	77.000000
	PANHANDLING	62.000000
	URINATING IN PUBLIC	41.000000
	VENDING	29.000000
	DRINKING	14.000000

Task 5 - Perform a statistical test for the following:

In [34]:
```
#Please note: For the below statements you need to state the Null and Alternate and the
#a statistical test to accept or reject the Null Hypothesis along with the correspondin
# a. Whether the average response time across complaint types is similar or
# b. Are the type of complaint or service requested and location related?
```

In [35]:
```
#a. Whether the average response time across complaint types is similar or not (overall
#Here complaint type is categorical data and response time is numerical data. Hence, nu
closed_status_requests.Request_Closing_Time.groupby(closed_status_requests['Complaint T
#Mean of response time based on Complaint type
```

Out[35]:
```
Complaint Type
```

```
AGENCY ISSUES                  315.333333
ANIMAL ABUSE                   312.512490
ANIMAL IN A PARK             20210.000000
BIKE/ROLLER/SKATE CHRONIC      225.693396
BLOCKED DRIVEWAY               284.142643
DERELICT VEHICLE               441.499119
DISORDERLY YOUTH               213.167832
DRINKING                       231.391373
GRAFFITI                       428.752212
HOMELESS ENCAMPMENT            261.000907
ILLEGAL FIREWORKS              165.470238
ILLEGAL PARKING                269.724982
NOISE - COMMERCIAL             188.524159
NOISE - HOUSE OF WORSHIP       191.287406
NOISE - PARK                   204.239741
NOISE - STREET/SIDEWALK        206.425917
NOISE - VEHICLE                215.039220
PANHANDLING                    262.072131
POSTING ADVERTISEMENT          118.262751
SQUEEGEE                       242.500000
TRAFFIC                        206.695526
URINATING IN PUBLIC            217.302365
VENDING                        240.542578
Name: Request_Closing_Time, dtype: float64
```

In [36]:
```
# import anova
from scipy.stats import f_oneway
```

In [37]:
```
#Programatically adding compaint types Request closing time params for f_oneway functio
result = closed_status_requests.groupby('Complaint Type')['Request_Closing_Time'].apply
f_oneway(*result)
```

Out[37]:
```
F_onewayResult(statistic=513.9606866025152, pvalue=0.0)
```

In [38]:
```
#Response Summary: ● Whether the average response time across complaint types is simila
# f_oneway(for each complaint type request closing time avg)
# Response F_onewayResult(statistic=513.9606866025152, pvalue=0.0)
#Since pvalue = 0.0 is less than 0.05, Reject Null
# Null - No Significant difference in average request closing time by complaint type
# Al - Significant difference in average request closing time by complaint type
#Finally:Reject_null There is relationship between avg response time for each complaint
```

In [39]:
```
#5.b. Are the type of complaint or service requested and location related?
#Both complaint type and locations are categorical data. Hence p score should be calucl
#Import chi2_contigency module
from scipy.stats import chi2_contingency
```

In [40]:
```
chi2_contingency(pd.crosstab(closed_status_requests['Complaint Type'], closed_status_re
```

Out[40]:
```
(119069.84639856996,
 0.0,
 1034,
 array([[4.42254021e-03, 1.41641902e-01, 2.45048933e-02, ...,
         2.20724962e-02, 4.94922455e-02, 7.36352946e-02],
        [5.72424122e+00, 1.83331835e+02, 3.17175002e+01, ...,
         2.85691675e+01, 6.40594631e+01, 9.53086162e+01],
```

```
       [7.37090036e-04, 2.36069836e-02, 4.08414888e-03, ...,
        3.67874936e-03, 8.24870758e-03, 1.22725491e-02],
       ...,
       [3.31174553e+00, 1.06066177e+02, 1.83500809e+01, ...,
        1.65286209e+01, 3.70614432e+01, 5.51405631e+01],
       [4.36357301e-01, 1.39753343e+01, 2.41781614e+00, ...,
        2.17781962e+00, 4.88323489e+00, 7.26534906e+00],
       [2.79578250e+00, 8.95412888e+01, 1.54911767e+01, ...,
        1.39534963e+01, 3.12873478e+01, 4.65497787e+01]]))
```

In [41]:

```
#Response summary: Are the type of complaint or service requested and location related?
#Response Summary: chi2_contigency(cross tabe between complaint type and city)
#Response : (119069.84639856996, (ch2 stat),
# 0.0 (pvalue),
# 1034,
#Since pvalue = 0.0 is less than 0.05, Reject Null
# Null - No Significant difference in city and complaint type
# Al - Significant difference in city and complaint type
#Finally: Reject null, means There is relationship between complaint type location of i
```

Thank You