

In [1]:

```
#importing the required packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import seaborn as sns
```

Task# 1

In [2]:

```
# loading the file
df=pd.read_csv("311_Service_Requests_from_2010_to_Present.csv")
df

c:\users\703196200\appdata\local\programs\python\python38-32\lib\site-packages\IPython\core\interactiveshell.py:3169: DtypeWarning: Columns (48,49) have mixed types.Specify dtype option on import or set low_memory=False.
    has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

Out[2]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip
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	10034.0
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	11105.0
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	10458.0
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	10461.0
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	11373.0
...
300693	30281872	03/29/2015 12:33:41 AM	NaN	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	NaN
300694	30281230	03/29/2015 12:33:28 AM	03/29/2015 02:33:59 AM	NYPD	New York City Police Department	Blocked Driveway	Partial Access	Street/Sidewalk	11418.0
300695	30283424	03/29/2015 12:33:03 AM	03/29/2015 03:40:20 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	11206.0
300696	30280004	03/29/2015 12:33:02 AM	03/29/2015 04:38:35 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	10461.0
300697	30281825	03/29/2015 12:33:01 AM	03/29/2015 04:41:50 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Store/Commercial	10036.0

300698 rows x 53 columns

In [3]:

```
df.shape
```

Out[3]:

```
(300698, 53)
```

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 53 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   Unique Key                          300698 non-null int64
 1   Created Date                        300698 non-null object
 2   Closed Date                         298534 non-null object
 3   Agency                             300698 non-null object
 4   Agency Name                        300698 non-null object
 5   Complaint Type                     300698 non-null object
 6   Descriptor                          294784 non-null object
 7   Location Type                     300567 non-null object
 8   Incident Zip                      298083 non-null float64
 9   Incident Address                   256288 non-null object
10   Street Name                       256288 non-null object
11   Cross Street 1                     251419 non-null object
12   Cross Street 2                     250919 non-null object
13   Intersection Street 1              43858 non-null object
14   Intersection Street 2              43362 non-null object
15   Address Type                       297883 non-null object
16   City                              298084 non-null object
17   Landmark                          349 non-null object
18   Facility Type                     298527 non-null object
19   Status                            300698 non-null object
20   Due Date                          300695 non-null object
21   Resolution Description              300698 non-null object
22   Resolution Action Updated Date     298511 non-null object
23   Community Board                    300698 non-null object
24   Borough                           300698 non-null object
25   X Coordinate (State Plane)         297158 non-null float64
26   Y Coordinate (State Plane)         297158 non-null float64
27   Park Facility Name                 300698 non-null object
28   Park Borough                      300698 non-null object
29   School Name                       300698 non-null object
30   School Number                     300698 non-null object
31   School Region                     300697 non-null object
32   School Code                       300697 non-null object
33   School Phone Number               300698 non-null object
34   School Address                    300698 non-null object
35   School City                       300698 non-null object
36   School State                      300698 non-null object
37   School Zip                        300697 non-null object
38   School Not Found                  300698 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                243 non-null object
44   Bridge Highway Direction           243 non-null object
45   Road Ramp                         213 non-null object
46   Bridge Highway Segment             213 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                          297158 non-null float64
51   Longitude                         297158 non-null float64
52   Location                          297158 non-null object
dtypes: float64(10), int64(1), object(42)
memory usage: 73.4+ MB
```

In [5]:

```
#Check for duplicates and NA values
dup=df.duplicated().sum()
nulval=df.isna().sum()
```

In [6]:

```
dup
```

Out[6]:

```
0
```

In [7]:

```
nulval
```

Out[7]:

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

```
Longitude      3540
Location      3540
dtype: int64
```

Task# 2

In [8]:

```
#Converting "Created Date" and "Closing Date" in Datetime datatype
df['Created Date'] = pd.to_datetime(df['Created Date'])
df['Closed Date'] = pd.to_datetime(df['Closed Date'])
```

In [9]:

```
#Creating new column "Request_Closing_Time" as difference of "Created Date" and "Closing Date"
Request_Closing_Time=df['Closed Date']-df['Created Date']
```

In [10]:

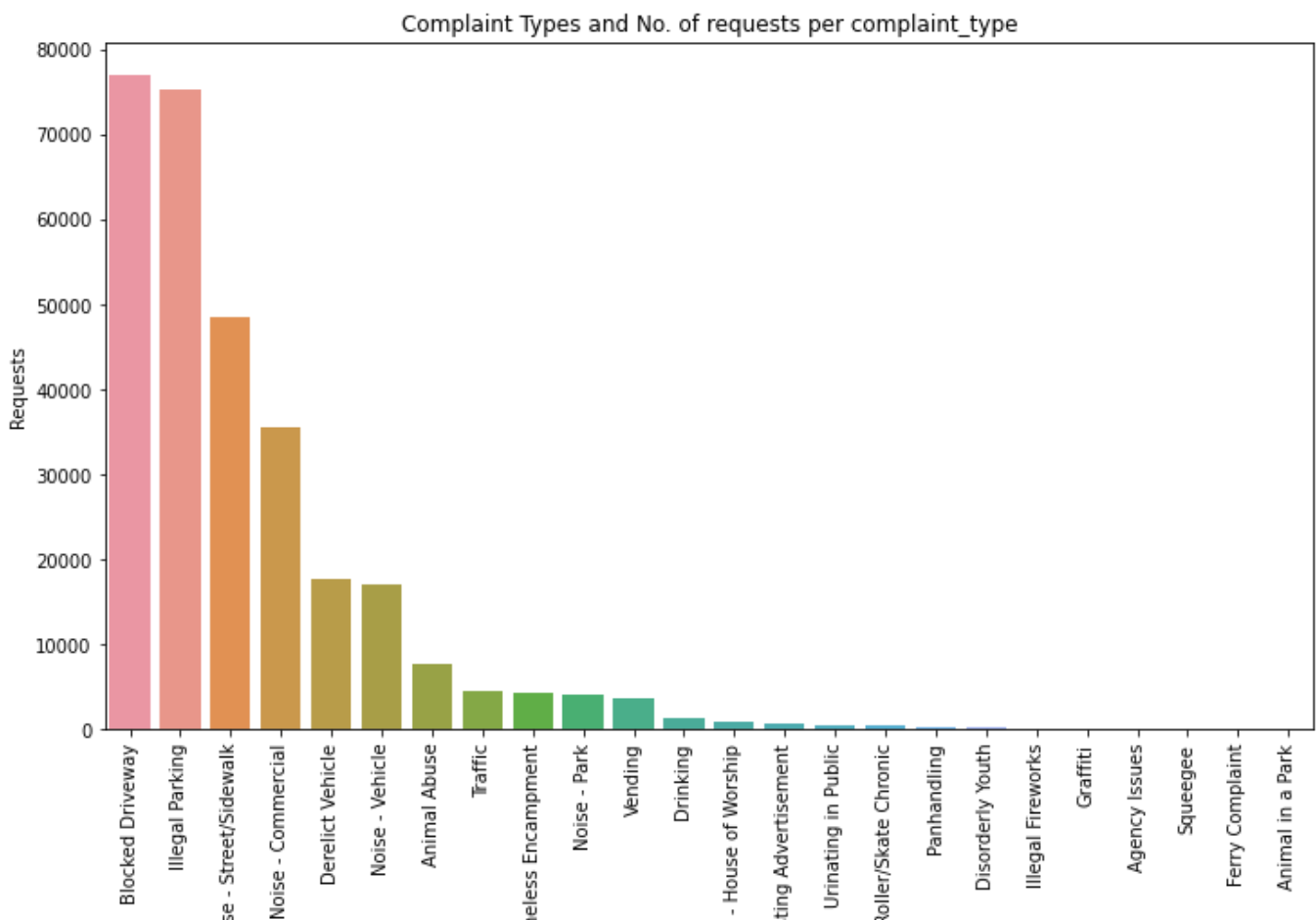
```
#Adding "Request_Closing_Time" in dataframe
df["Request_Closing_Time"]=Request_Closing_Time
```

Task# 3

In [11]:

```
# Number of Compliant Type based on the Requets/Complaint

plt.figure(figsize=(12,7))
sns.countplot(x='Complaint Type',data=df,order=df['Complaint Type'].value_counts().index)
plt.xticks(rotation=90)
plt.title("Complaint Types and No. of requests per complaint_type")
plt.xlabel('Complaint Type')
plt.ylabel('Requests')
plt.show()
```



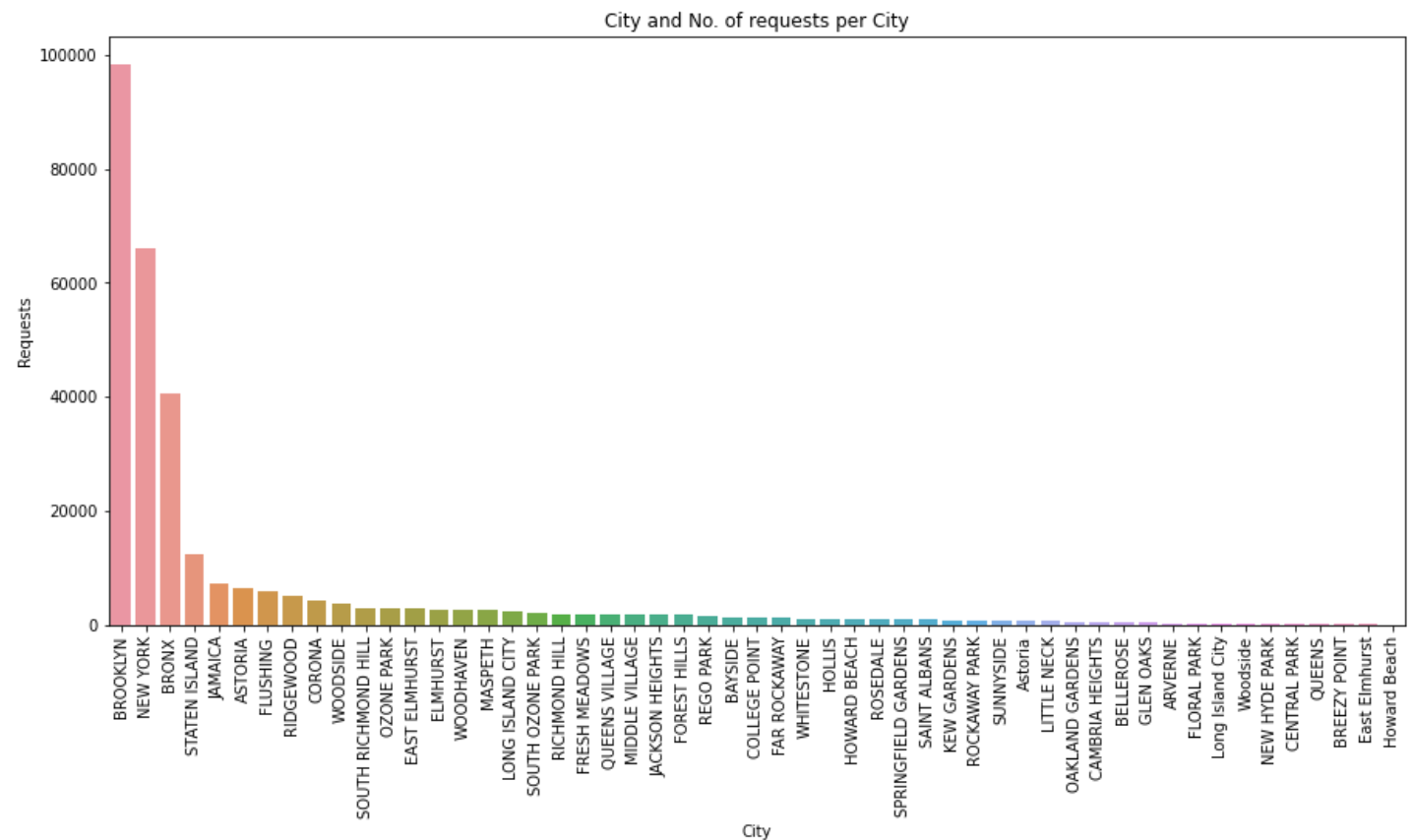
Observation 1 :: Top 5 Complaint Type are :

1. Blocked Driveway
2. Illegal Parking
3. Noise - Street/Sidewalk
4. Noise - Commercial
5. Dereleict Vehicle

In [12]:

```
# Number of Complaint Request based on the City

plt.figure(figsize=(15,7))
sns.countplot(x='City',data=df,order=df['City'].value_counts().index)
plt.xticks(rotation=90)
plt.title("City and No. of requests per City")
plt.xlabel('City')
plt.ylabel('Requests')
plt.show()
```



Observation 2 :: Top 5 City where requests are lodged at highest :

1. Brooklyn
2. Ney York
3. Bronx
4. Staten Island
5. Jamaica

In [13]:

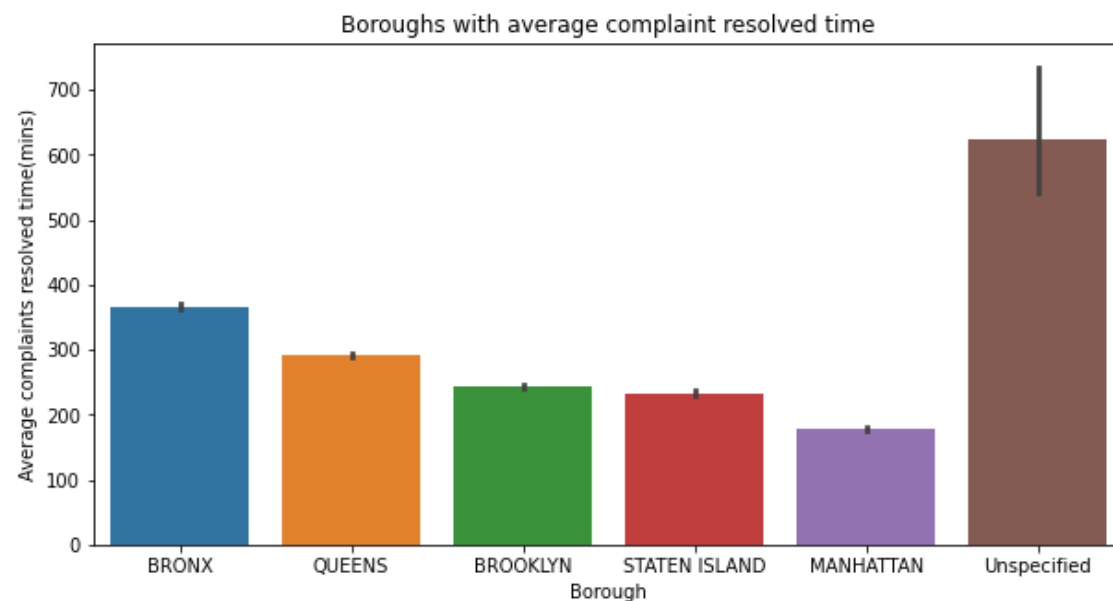
```
#Converting the Request Closing Time in hours and minutes
```

```
df['Request_Closing_Time_H']= df['Closed Date'].values- df['Created Date'].values  
df['Request_Closing_Time_M']= df['Request_Closing_Time_H']/np.timedelta64(1, 'm')
```

In [14]:

```
# Average Closing time of request w.r.t. Borough's
```

```
plt.figure(figsize=(10,5))  
sns.barplot(x='Borough',y='Request_Closing_Time_M',data=df,order=['BRONX','QUEENS','BROO  
KLYN','STATEN ISLAND','MANHATTAN','Unspecified'])  
plt.title("Boroughs with average complaint resolved time")  
plt.xlabel("Borough")  
plt.ylabel("Average complaints resolved time(mins)")  
plt.show()
```



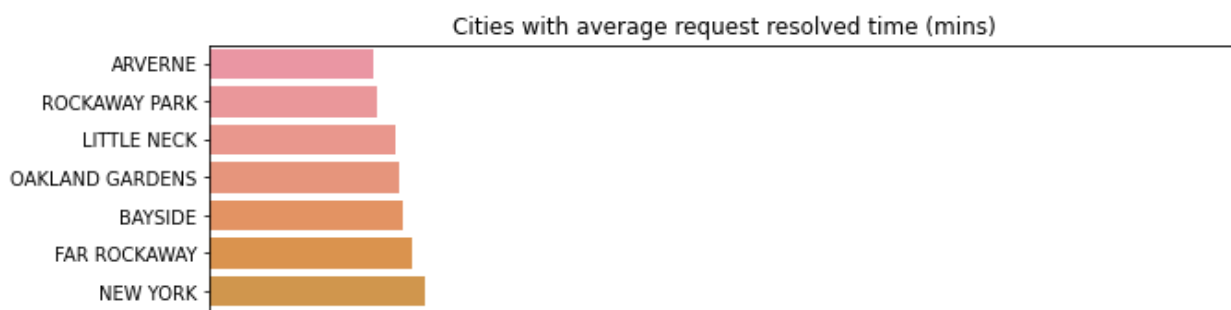
Observation 3 :: Average Complaint resolved in Boroughs :

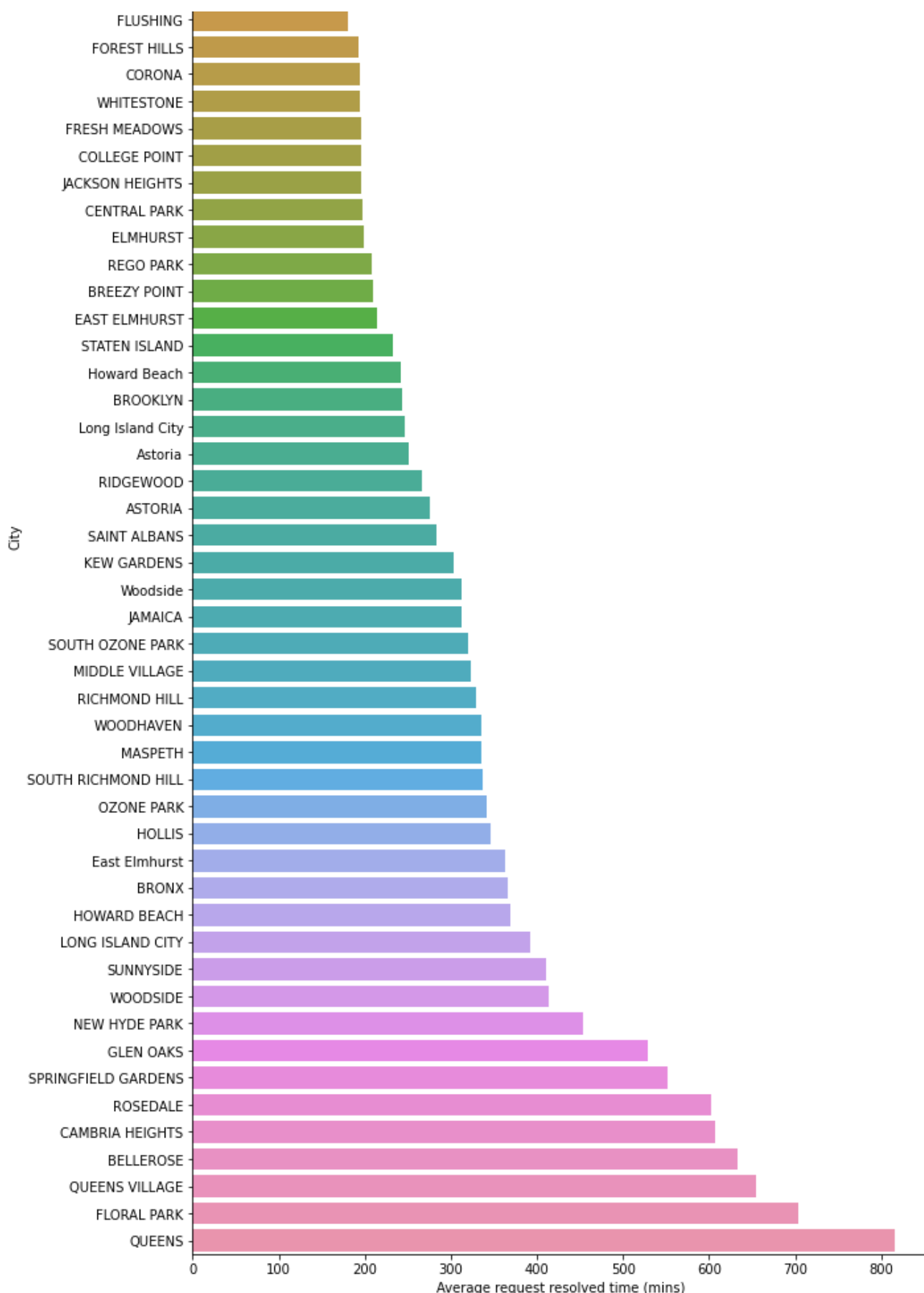
"Manhattan" Borough has the minimum average complaint response time and "Bronx" Borough has the maximum average complaint response time

In [15]:

```
# Average Closing time of request w.r.t. City
```

```
df1= df[['City','Request_Closing_Time_M']]  
c1= df1.groupby('City')['Request_Closing_Time_M'].mean().to_frame()  
c1= c1.sort_values('Request_Closing_Time_M')  
c1['City']= c1.index  
plt.figure(figsize=(10,20))  
sns.barplot(y='City',x='Request_Closing_Time_M',data=c1)  
plt.title("Cities with average request resolved time (mins)")  
plt.ylabel("City")  
plt.xlabel("Average request resolved time (mins)")  
plt.show()
```





Observation 4 :: Visualizing Cities with average response time :

"Arverne" has the minimum complaint request request respond time and "Queens" has the maximum complaint request respond time

In [16]:

```
# Average Closing time of request w.r.t. Location
df2= df[['Location Type','Request_Closing_Time_M']]
```

Location types with average request resolved time (mins)

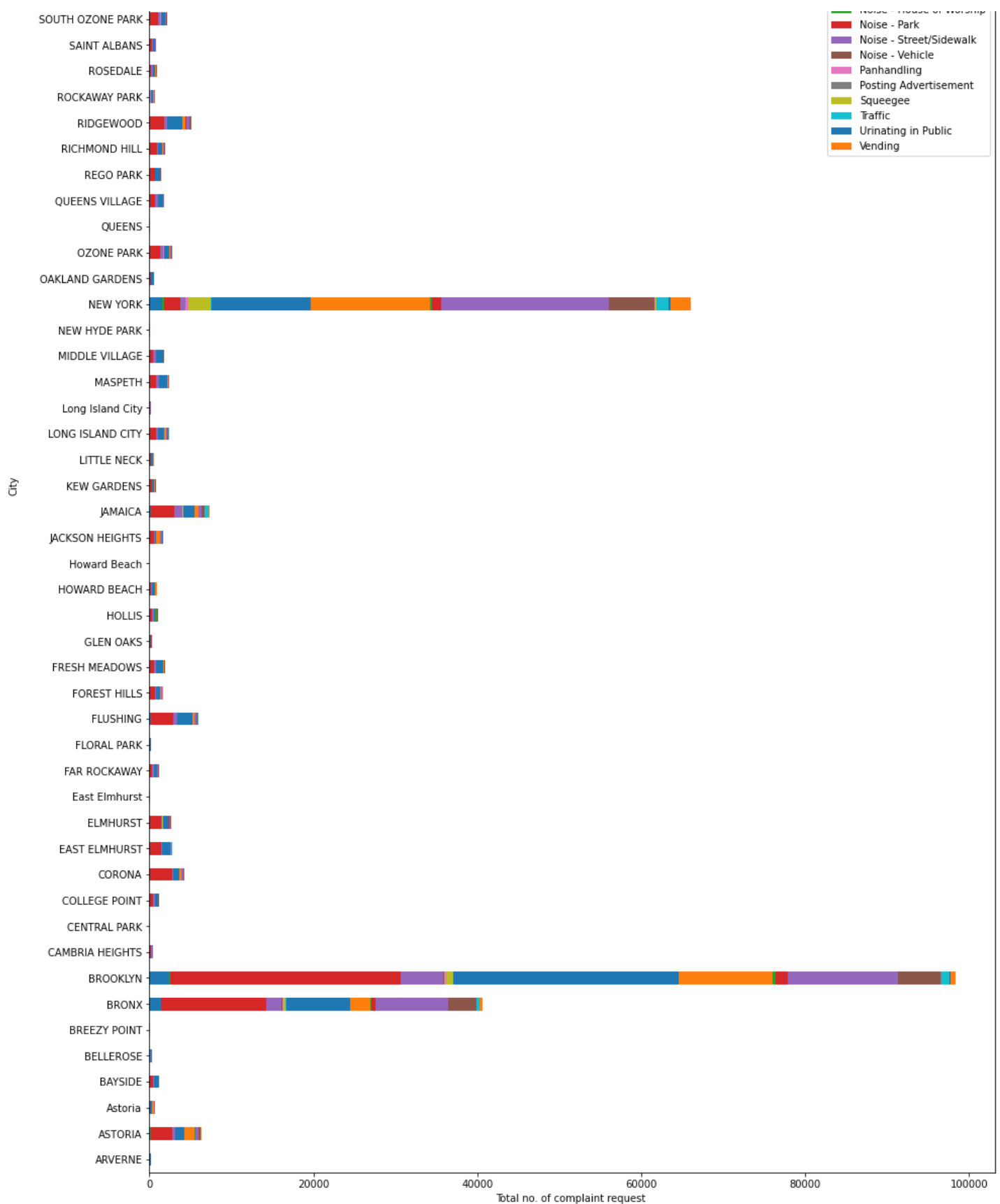
Location Type	Average request resolved time (mins)
Bridge	~200
Club/Bar/Restaurant	~200
Commercial	~300
Ferry	~0
Highway	~200
House and Store	~300
House of Worship	~200
Park	~20500
Park/Playground	~200
Parking Lot	~300
Residential Building	~300
Residential Building/House	~300
Roadway Tunnel	~300
Store/Commercial	~200
Street/Sidewalk	~300
Subway Station	~200
Terminal	~0
Vacant Lot	~500

Complaints at "Subway Station" has the least respond time and Complaints at "Park" has the maximum respond time

Total number of request lodged w.r.t. City

<Figure size 1440x720 with 0 Axes>





Observation 6 :: Visualizing Complaint Request against each City :

Brooklyn has the maximum complaint types received and it has the maximum number of complaints requested as well than any other city.

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

In [18]:

```
df_avg_res_time_city = df.groupby(['Location Type', 'Complaint Type']).Request_Closing_Ti
```

```
me_M.mean()
#df_avg_res_time_city = df.groupby(['City', 'Complaint Type']).Request_Closing_Time_M.mean()
df_avg_res_time_city.head(50)
```

Out[18]:

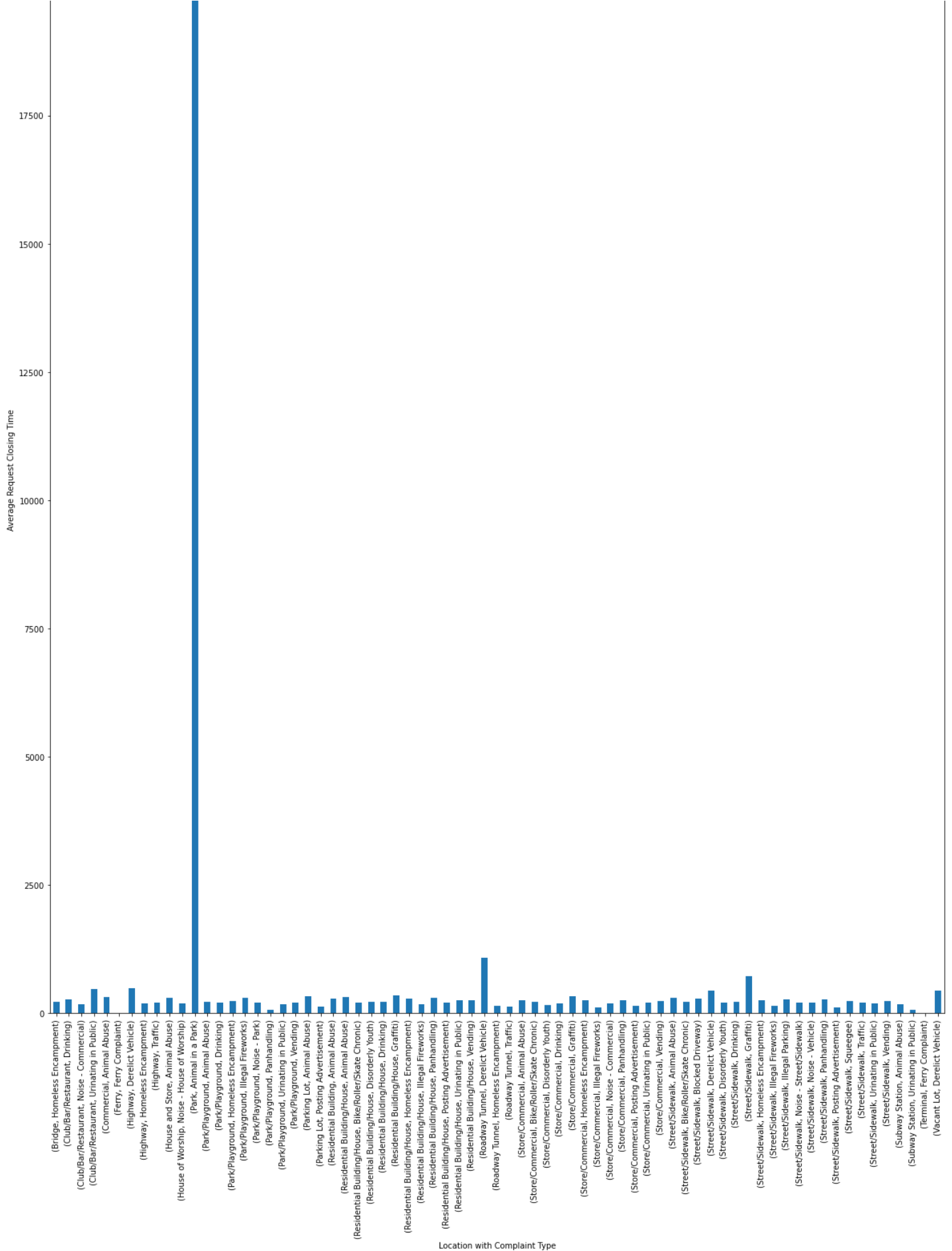
Location Type	Complaint Type	
Bridge	Homeless Encampment	229.158333
Club/Bar/Restaurant	Drinking	272.748721
	Noise - Commercial	183.835283
	Urinating in Public	475.200000
Commercial	Animal Abuse	320.566129
Ferry	Ferry Complaint	NaN
Highway	Derelict Vehicle	491.742308
	Homeless Encampment	196.270000
	Traffic	206.860663
House and Store	Animal Abuse	300.795699
House of Worship	Noise - House of Worship	191.833279
Park	Animal in a Park	20210.083333
Park/Playground	Animal Abuse	222.149590
	Drinking	206.479762
	Homeless Encampment	235.429792
	Illegal Fireworks	300.183333
	Noise - Park	204.539907
	Panhandling	73.119444
	Urinating in Public	171.749561
	Vending	208.027460
Parking Lot	Animal Abuse	332.423939
	Posting Advertisement	126.945238
Residential Building	Animal Abuse	289.089941
Residential Building/House	Animal Abuse	323.971065
	Bike/Roller/Skate Chronic	216.678000
	Disorderly Youth	231.268615
	Drinking	220.696886
	Graffiti	352.727976
	Homeless Encampment	295.446270
	Illegal Fireworks	185.982828
	Panhandling	309.933333
	Posting Advertisement	216.030556
	Urinating in Public	256.048309
	Vending	251.825041
Roadway Tunnel	Derelict Vehicle	1077.850000
	Homeless Encampment	151.083333
	Traffic	130.622989
Store/Commercial	Animal Abuse	256.368490
	Bike/Roller/Skate Chronic	230.400000
	Disorderly Youth	170.368750
	Drinking	195.155741
	Graffiti	333.638542
	Homeless Encampment	254.241276
	Illegal Fireworks	115.450000
	Noise - Commercial	193.420127
	Panhandling	260.780556
	Posting Advertisement	142.150000
	Urinating in Public	206.384596
	Vending	238.430285
Street/Sidewalk	Animal Abuse	306.649629

Name: Request_Closing_Time_M, dtype: float64

In [19]:

```
df_avg_res_time_city.plot(kind='bar', figsize=(20,25), stacked=True)
plt.title("Complaint at different location with their average closing time")
plt.xlabel("Location with Complaint Type")
plt.ylabel("Average Request Closing Time")
plt.show()
```

Complaint at different location with their average closing time



Task # 5 :

F-Test

Testing at Confidence level(95%) => alpha value = 0.05

- Null Hypothesis : H0 • There is no significant difference in average response time across different complaint

- Null Hypothesis : H0 : There is no significant difference in average response time across different complaint types
- Alternate Hypothesis : H1 : There is a significant difference in average response time across different complaint types

In [20]:

```
# Dropping NA values for statistical testing
df.dropna(subset=['Request_Closing_Time_H', 'Request_Closing_Time_M'], inplace=True)
```

In [21]:

```
# Storing mean response time for various complaint types
complaints = df['Complaint Type'].value_counts().index
for i in range(len(complaints)):
    exec("sample{} = df.loc[(df['Complaint Type'] == '{}') , 'Request_Closing_Time_M']").format(i+1, complaints[i])
```

In [22]:

```
# Performing F-statics
fscore, pvalue = stats.f_oneway(sample1, sample2, sample3, sample4, sample5, sample6, sample7, sample8, sample9, sample10, sample11, sample12, sample13, sample14, sample15, sample16, sample17, sample18, sample19, sample20, sample21)
print("score : {:.2f} , pvalue : {:.2f}".format(fscore, pvalue))
```

score : 410.86 , pvalue : 0.00

Here , pvalue (0.00) < alpha value(0.05) ,hence, We reject our Null Hypothesis

Observation : There is a significant difference in average response time across different complaint types ,the average response time across different complaint types is not similar (overall)

Chi-Square Test of Independence

Testing at Confidence level(95%) => alpha value = 0.05

- Null Hypothesis : H0 : There is no significant relation between type of complaint and location
- Alternate Hypothesis : H1 : There is some significant relation between type of complaint and location

In [23]:

```
# Performing Chi-square test of independence
location_complaint_type = pd.crosstab(df['Complaint Type'], df['Location'])
```

In [24]:

```
cscore, pval, df, et = stats.chi2_contingency(location_complaint_type)
print("score : {:.2f} , pvalue : {:.2f}".format(cscore, pval))
```

score : 4160841.63 , pvalue : 0.00

Here , pvalue (0.00) < alpha value(0.05)

Observation : We reject our Null Hypothesis as there is some significant relation between type of complaint and location which means that the type of complaint or service requested and the location are related

In []: