

Customer Service Requests Analysis.

Analysis tasks to performed.

- 1.Import a 311 NYC service request.
- 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)
- 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.
- 4.Order the complaint types based on the average 'Request_Closing_Time', grouping them for different locations.
- 5.Perform a statistical test for the following: Please note: For the below statements you need to state the Null and Alternate and then provide a statistical test to accept or reject the Null Hypothesis along with the corresponding 'p-value'.

Whether the average response time across complaint types is similar or not (overall) Are the type of complaint or service requested and location related?

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
In [2]: #Import a 311 NYC service request.
NYC=pd.read_csv('311_Service_Requests_from_2010_to_Present.csv')
```

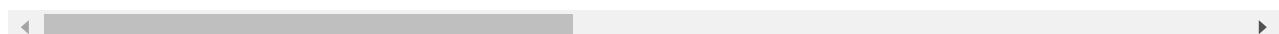
C:\Users\Admin\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3444: DtypeWarning: Columns (48,49) have mixed types. Specify dtype option on import or set low_memory=False.
exec(code_obj, self.user_global_ns, self.user_ns)

```
In [3]: NYC.head()
```

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Number
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	100-000000000000000000
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	111-000000000000000000

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incid
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 [4]:

`NYC.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: 121.6+ MB

In [5]:

```
NYC.isnull().sum()
```

Out[5]: 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
dtype: int64

```

Descriptive Analysis

In [6]:

```
##Read or convert the columns 'Created Date' and Closed Date' to datetime datatype.
NYC.describe()
```

Out[6]:

	Unique Key	Incident Zip	X Coordinate (State Plane)	Y Coordinate (State Plane)	School or Citywide Complaint	Vehicle Type	Taxi Company Borough	Pick Locat
count	3.006980e+05	298083.000000	2.971580e+05	297158.000000	0.0	0.0	0.0	N
mean	3.130054e+07	10848.888645	1.004854e+06	203754.534416	NaN	NaN	NaN	N
std	5.738547e+05	583.182081	2.175338e+04	29880.183529	NaN	NaN	NaN	N
min	3.027948e+07	83.000000	9.133570e+05	121219.000000	NaN	NaN	NaN	N
25%	3.080118e+07	10310.000000	9.919752e+05	183343.000000	NaN	NaN	NaN	N
50%	3.130436e+07	11208.000000	1.003158e+06	201110.500000	NaN	NaN	NaN	N
75%	3.178446e+07	11238.000000	1.018372e+06	224125.250000	NaN	NaN	NaN	N
max	3.231065e+07	11697.000000	1.067173e+06	271876.000000	NaN	NaN	NaN	N

In [7]:

```
NYC.shape
```

Out[7]:

```
(300698, 53)
```

Now we perform exploratory data analysis because all the values given in the above does not provides us very clear insights.

In [8]:

```
NYC['Created Date']=pd.to_datetime(NYC['Created Date'])
NYC['Closed Date']=pd.to_datetime(NYC['Closed Date'])
```

```
In [9]: #create a new column 'Request_Closing_Time' as the time elapsed between request creation and closure
NYC['Request_Closing_Time']=(NYC['Closed Date']-NYC['Created Date']).dt.total_seconds()
nyc1=NYC[NYC['Request_Closing_Time'].notnull()]
nyc_clean=nyc1[nyc1['Closed Date']>=nyc1['Created Date']]
nyc_clean['day of week']=nyc_clean['Created Date'].dt.dayofweek
nyc_clean['month']=nyc_clean['Created Date'].dt.month
nyc_clean['year']=nyc_clean['Created Date'].dt.year
nyc_clean=nyc_clean[nyc_clean.Borough!='Unspecified']
```

```
In [10]: nyc_clean.shape
```

```
Out[10]: (298068, 57)
```

```
In [11]: NYC['Agency'].unique()
```

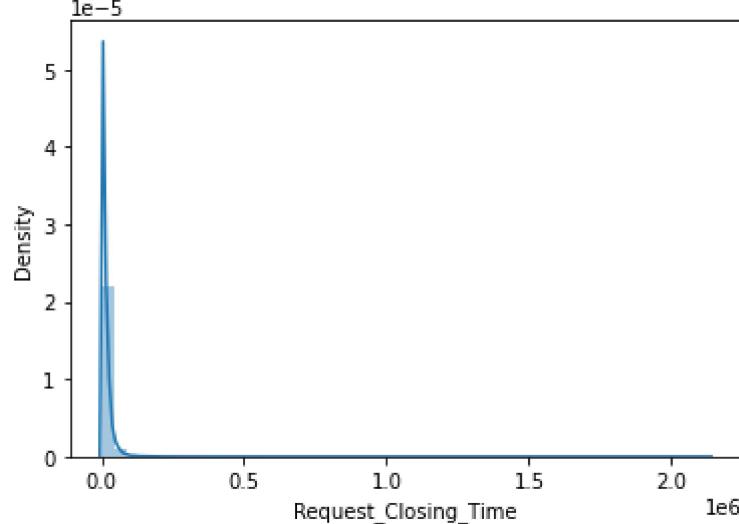
```
Out[11]: array(['NYPD'], dtype=object)
```

All of our data belongs to a single agency NYPD i.e New York City Police Department.

```
In [12]: #Univariate distribution plot for Request closing time.
sns.distplot(NYC['Request_Closing_Time'])
plt.show()
```

C:\Users\Admin\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```



```
In [13]: NYC.head()
```

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip
------------	--------------	-------------	--------	-------------	----------------	------------	---------------	--------------

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident ID
0	32310363	2015-12-31 23:59:45	2016-01-01 00:55:00	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034
1	32309934	2015-12-31 23:59:44	2016-01-01 01:26:00	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105
2	32309159	2015-12-31 23:59:29	2016-01-01 04:51:00	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458
3	32305098	2015-12-31 23:57:46	2016-01-01 07:43:00	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461
4	32306529	2015-12-31 23:56:58	2016-01-01 03:24:00	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373

5 rows × 54 columns



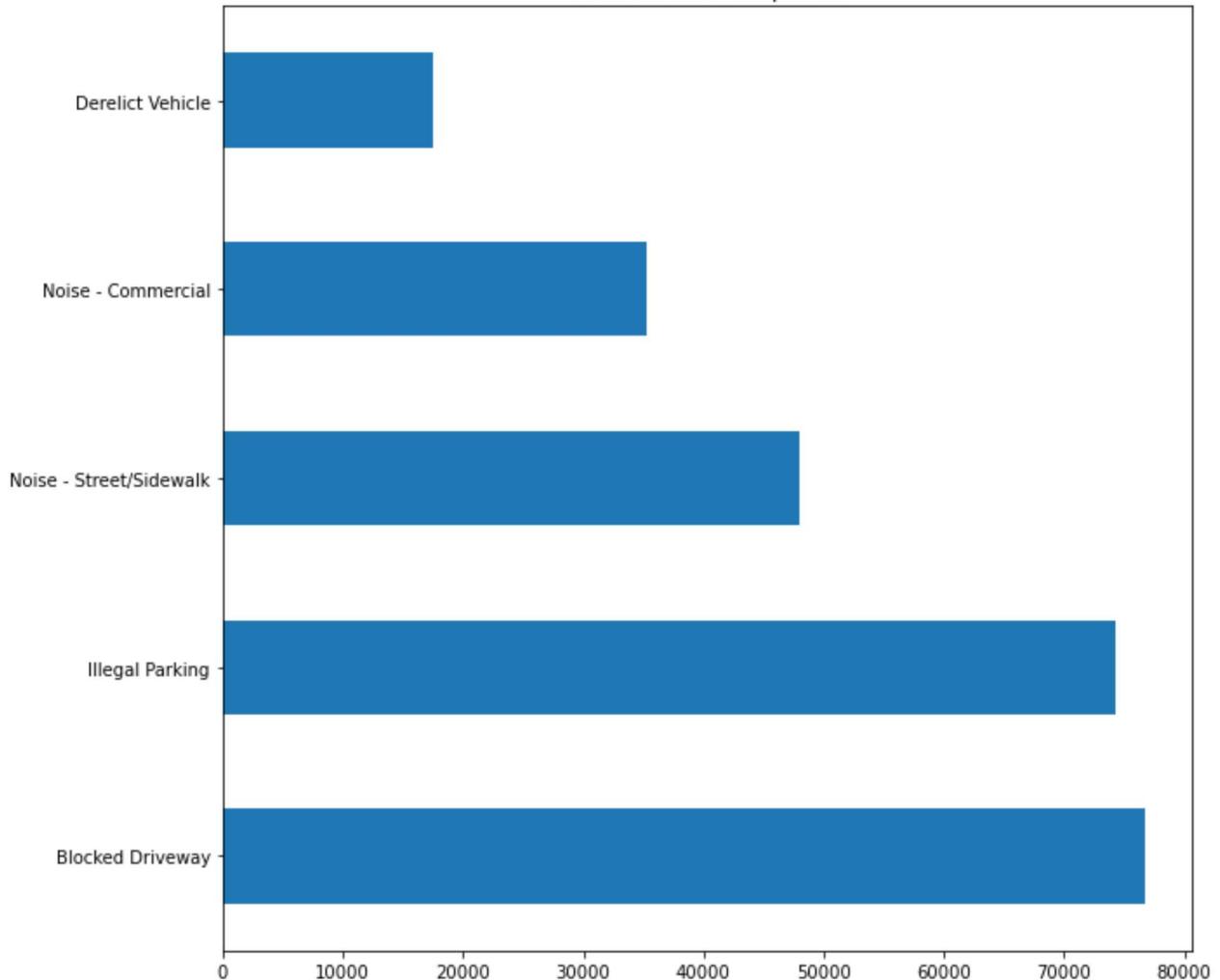
Major Insights

In [14]:

```
#Most frequent complaints
(nyc_clean['Complaint Type'].value_counts()).head().plot(kind='barh',figsize=(10,10),title='Most common complaints')
```

Out[14]:

```
<AxesSubplot:title={'center':'Most common complaints'}>
```

Most common complaints

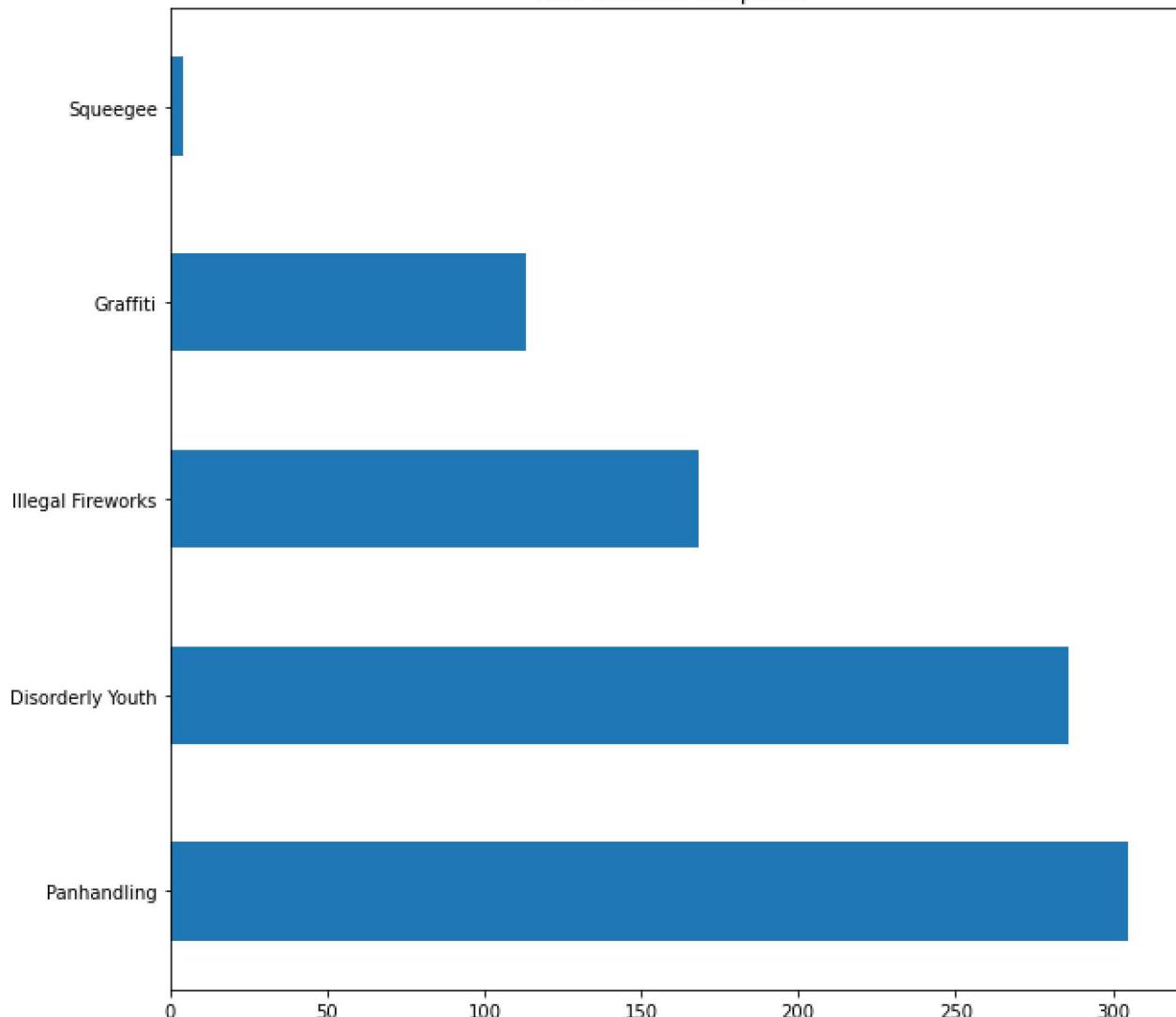
In [15]:

```
#Least common complaints
(nyc_clean['Complaint Type'].value_counts()).tail().plot(kind='barh',figsize=(10,10),ti
```

Out[15]:

```
<AxesSubplot:title={'center':'Most common complaints'}>
```

Most common complaints

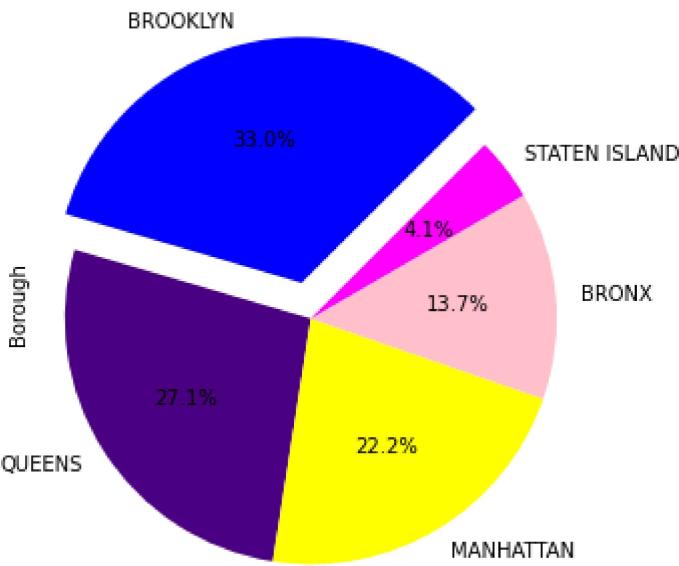


```
In [16]: nyc_clean.shape
```

```
Out[16]: (298068, 57)
```

```
In [17]: # complaints distribution across Borough
colors = ['blue','indigo','yellow','pink','magenta','cyan','violet']
nyc_clean['Borough'].value_counts().plot(kind='pie', autopct='%1.1f%%', explode=(0.15,0,0)
plt.axis('equal')
plt.title("complaints distribution across Borough")
plt.tight_layout()
plt.show()
```

complaints distribution across Borough



Maximum complaint requests were registered from Brooklyn.

```
In [18]: brook=nyc_clean[nyc_clean['Borough']=='BROOKLYN']
```

```
In [19]: nyc_clean.shape
```

```
Out[19]: (298068, 57)
```

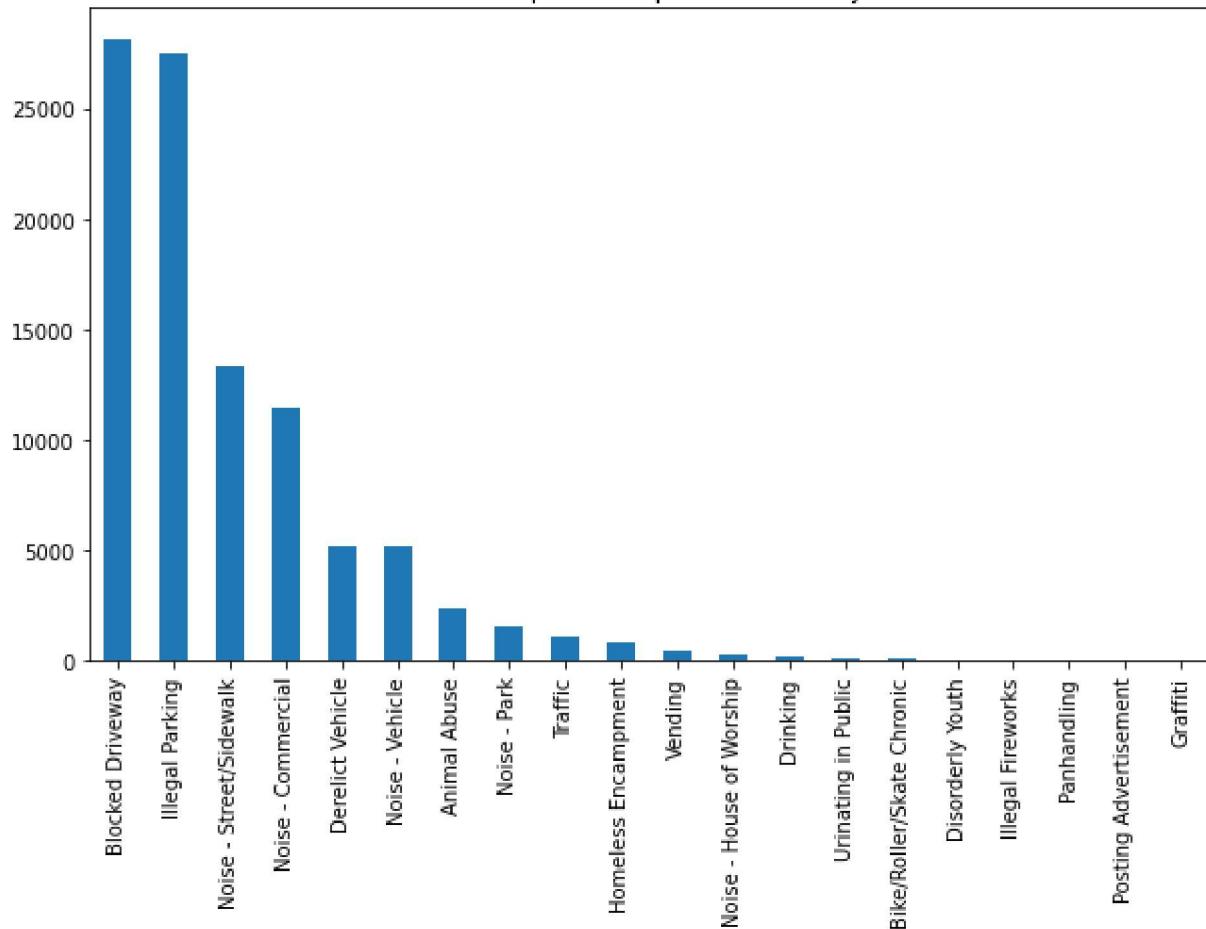
```
In [20]: brook.shape
```

```
Out[20]: (98295, 57)
```

```
In [21]: (brook['Complaint Type'].value_counts()).head(25).plot(kind='bar', figsize=(10,6), title
```

```
Out[21]: <AxesSubplot:title={'center':'Most Frequent Complaints in Brooklyn'}>
```

Most Frequent Complaints in Brooklyn

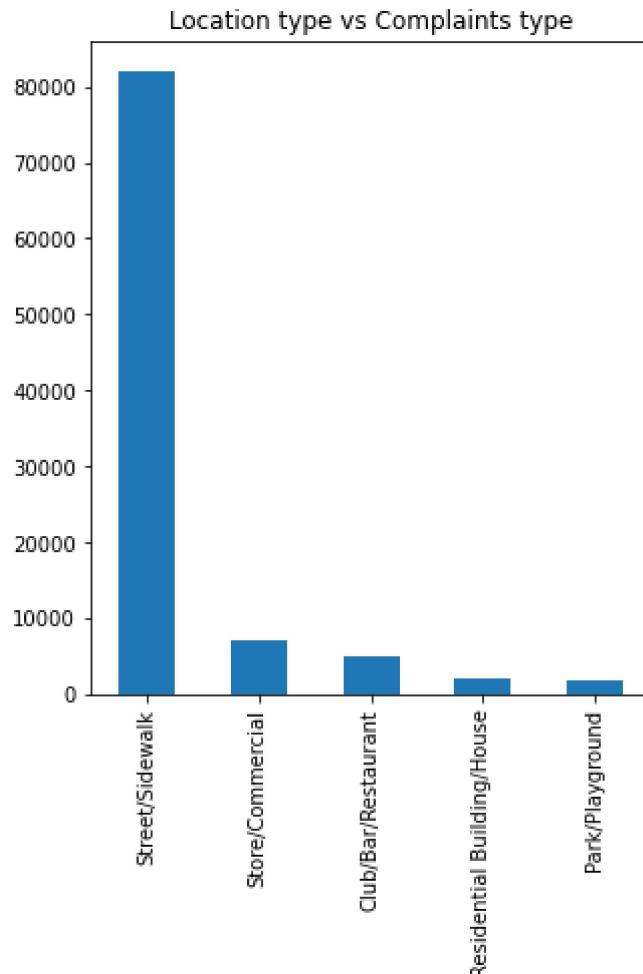


In [22]:

```
brook['Location Type'].value_counts().head().plot(kind='bar', figsize=(5,6), title='Locat
```

Out[22]:

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



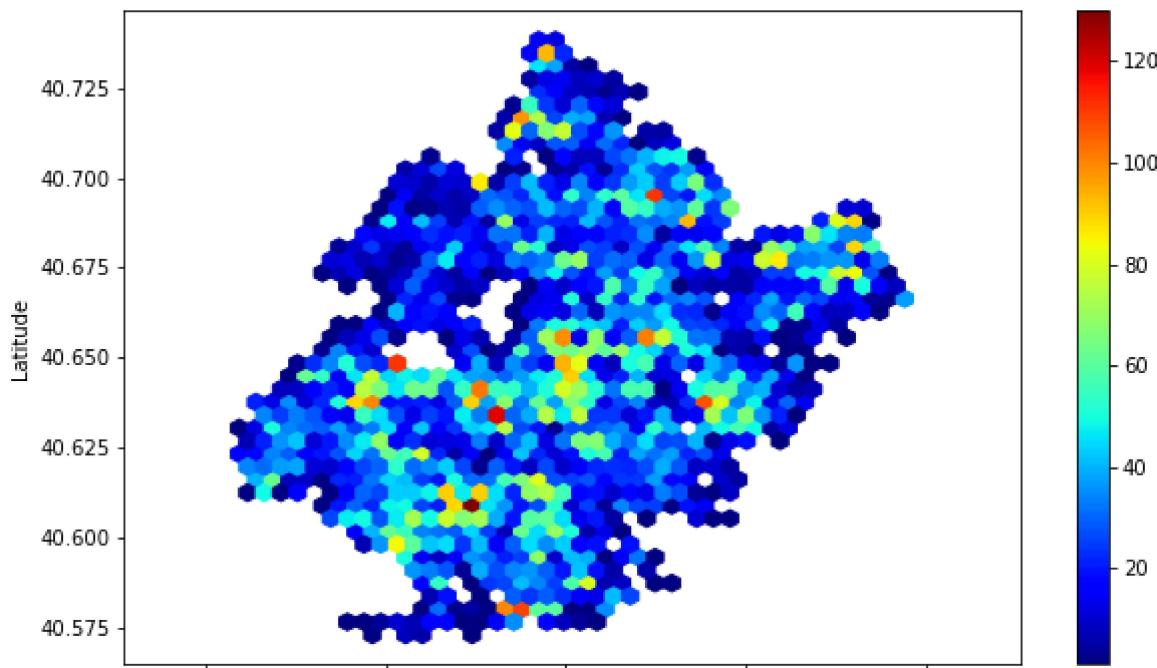
```
In [23]: nyc_clean[nyc_clean['Complaint Type']=='Blocked Driveway']['Descriptor'].value_counts()
```

```
Out[23]: No Access      56786
Partial Access    19967
Name: Descriptor, dtype: int64
```

```
In [24]: brook_blocked=brook[brook['Complaint Type'] == 'Blocked Driveway']
```

```
In [25]: brook_blocked.plot(kind='hexbin',x='Longitude',y='Latitude',gridsize=40,title='Blocked
Out[25]: (-74.04994270070343, -73.84653002929657, 40.5645918345, 40.7464960955)
```

Blocked driveway issues concentration across Brooklyn



Request closing time in seconds, grouping them for different location.

```
In [26]: nyc_avg_res_time=nyc_clean.groupby(['City','Complaint Type']).Request_Closing_Time.mean
```

```
In [27]: nyc_avg_res_time.head(18)
```

```
Out[27]:
```

City	Complaint Type	Request_Closing_Time
ARVERNE	Animal Abuse	7753.052632
	Blocked Driveway	9093.485714
	Derelict Vehicle	10685.592593
	Disorderly Youth	12928.500000
	Drinking	859.000000
	Graffiti	5520.000000
	Homeless Encampment	6533.250000
	Illegal Parking	8338.913793
	Noise - Commercial	8234.000000
	Noise - House of Worship	5623.909091
	Noise - Park	4620.000000
	Noise - Street/Sidewalk	7172.620690
	Noise - Vehicle	6695.571429
	Panhandling	3720.000000
	Urinating in Public	2491.000000
	Vending	1740.000000
ASTORIA	Animal Abuse	18000.608000
	Bike/Roller/Skate Chronic	6261.533333

Name: Request_Closing_Time, dtype: float64

```
In [28]: nyc_avg_res_time=nyc_clean.groupby(['Complaint Type']).Request_Closing_Time.mean().sort
```

```
In [29]: nyc_avg_res_time.head(15)
```

```
Out[29]: Complaint Type
          Posting Advertisement      7112.891975
          Illegal Fireworks        9940.101190
          Noise - Commercial     11291.632884
          Noise - House of Worship 11495.874058
          Noise - Park            12246.158157
          Noise - Street/Sidewalk  12377.738882
          Traffic                  12415.252002
          Disorderly Youth       12810.902098
          Noise - Vehicle          12918.914430
          Urinating in Public     13055.991554
          Bike/Roller/Skate Chronic 13523.545024
          Drinking                 13879.309748
          Vending                  14449.060358
          Squeegie                 14564.250000
          Homeless Encampment      15716.052536
          Name: Request_Closing_Time, dtype: float64
```

From the above, Null hypothesis: avg response time across complaint type are not equal

Alternate hypothesis: avg response time across complaint type are equal

Following complaints have resolution times which are very close. Disorderly Youth 12810.902098 ,Noise - Vehicle 12918.914430 One group can be formed for these complaints and one way Anova for this can be performed

```
In [30]: nyc_des_youth=nyc_clean[nyc_clean['Complaint Type']=='Disorderly Youth']
```

```
In [31]: nyc_des_youth=nyc_des_youth.loc[:,['Request_Closing_Time']]
```

```
In [32]: nyc_des_youth.head()
```

```
Out[32]: Request_Closing_Time
```

4670	713.0
9034	4605.0
12027	2345.0
12176	19415.0
17181	6849.0

```
In [33]: nyc_nos_veh=nyc_clean[nyc_clean['Complaint Type']=='Noise - Vehicle']
nyc_nos_veh=nyc_nos_veh.loc[:,['Request_Closing_Time']]
nyc_nos_veh.head()
```

```
Out[33]: Request_Closing_Time
```

Request_Closing_Time	
87	22949.0
156	7254.0
172	11319.0
221	10937.0
319	2615.0

```
In [34]: nyc_type_res=nyc_clean.loc[:,['Complaint Type','Request_Closing_Time']]
nyc_type_res.head()
nyc_type_res.columns
```

```
Out[34]: Index(['Complaint Type', 'Request_Closing_Time'], dtype='object')
```

```
In [37]: import scipy.stats as stats
#stats f_oneway functions takes the groups as input and returns F and P-value
fvalue, pvalue = stats.f_oneway(nyc_des_youth, nyc_nos_veh)
pvalue
```

```
Out[37]: array([0.91269878])
```

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

```
In [39]: nyc_post_adv=nyc_clean[nyc_clean['Complaint Type']=='Posting Advertisement']
nyc_post_adv=nyc_post_adv.loc[:,['Request_Closing_Time']]
nyc_post_adv.head()
```

Request_Closing_Time	
39	7596.0
42	7745.0
46	7834.0
49	8042.0
51	8137.0

```
In [40]: nyc_der=nyc_clean[nyc_clean['Complaint Type']=='Derelict Vehicle']
nyc_der=nyc_der.loc[:,['Request_Closing_Time']]
nyc_der.head()
```

Request_Closing_Time	
14	37763.0
151	14221.0
255	4913.0
256	14879.0

Request_Closing_Time

295	2712.0
-----	--------

```
In [41]: fvalue, pvalue = stats.f_oneway(nyc_post_adv, nyc_der)
pvalue
```

```
Out[41]: array([7.28776953e-35])
```

Null hypothesis for Posting Advertisement and Derelict Vehicle to be rejected p-value < 0.05

```
In [ ]:
```