Customer Service Requests Analysis 311

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

Q-1. Import a 311 NYC service request.

In [2]:
```
df = pd.read_csv('311_Service_Requests_from_2010_to_Present.csv')
```

```
C:\Users\comac 12\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3071: DtypeWarning: Columns (48,49) have mixed types.Specif
y dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

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

Out[3]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction	Road Ramp	Bridge Highway Segment	Garage Lot Name	D
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	71 VERMILYEA AVENUE	...	NaN	NaN	NaN	NaN	NaN	
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	27-07 23 AVENUE	...	NaN	NaN	NaN	NaN	NaN	
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	2897 VALENTINE AVENUE	...	NaN	NaN	NaN	NaN	NaN	
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	2940 BAISLEY AVENUE	...	NaN	NaN	NaN	NaN	NaN	
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	87-14 57 ROAD	...	NaN	NaN	NaN	NaN	NaN	

5 rows × 53 columns

In [5]:
```
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: 121.6+ MB
```

In [4]:
```
df.isnull().sum()/len(df)*100
```

Out[4]:
```
Unique Key                        0.000000
Created Date                      0.000000
Closed Date                       0.719659
Agency                            0.000000
Agency Name                       0.000000
Complaint Type                    0.000000
Descriptor                        1.966757
Location Type                     0.043565
Incident Zip                      0.869643
Incident Address                 14.768971
Street Name                      14.768971
Cross Street 1                   16.388203
Cross Street 2                   16.554483
Intersection Street 1            85.414602
Intersection Street 2            85.579552
Address Type                      0.936155
City                              0.869311
Landmark                         99.883937
Facility Type                     0.721987
Status                            0.000000
Due Date                          0.000998
Resolution Description            0.000000
Resolution Action Updated Date    0.727308
Community Board                   0.000000
Borough                           0.000000
X Coordinate (State Plane)        1.177261
Y Coordinate (State Plane)        1.177261
Park Facility Name                0.000000
Park Borough                      0.000000
School Name                       0.000000
School Number                     0.000000
School Region                     0.000333
School Code                       0.000333
School Phone Number               0.000000
School Address                    0.000000
School City                       0.000000
School State                      0.000000
School Zip                        0.000333
School Not Found                  0.000000
School or Citywide Complaint    100.000000
Vehicle Type                    100.000000
Taxi Company Borough            100.000000
Taxi Pick Up Location           100.000000
Bridge Highway Name              99.919188
Bridge Highway Direction         99.919188
Road Ramp                        99.929165
Bridge Highway Segment           99.929165
Garage Lot Name                 100.000000
Ferry Direction                  99.999667
Ferry Terminal Name              99.999335
Latitude                          1.177261
Longitude                         1.177261
Location                          1.177261
dtype: float64
```

Data Cleaning with Null Values

In [6]:
```
#Dropping columns with more than 30% null values:
df.drop(labels=['Intersection Street 1',
                'Intersection Street 2',
                'Landmark',
                'Vehicle Type',
                'School or Citywide Complaint',
                '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',
                ],axis=1, inplace=True)
```

In [7]:
```
# percentage of missing values greater than 5% and less than 30%: Replacing values with mode
df['Incident Address'].fillna(value=df['Incident Address'].mode()[0],inplace=True)
df['Street Name'].fillna(value=df['Street Name'].mode()[0],inplace=True)
df['Cross Street 1'].fillna(value=df['Cross Street 1'].mode()[0],inplace=True)
df['Cross Street 2'].fillna(value=df['Cross Street 2'].mode()[0],inplace=True)
```

In [9]:
```
#percentage of missing value in column is less than 5%, so dropping rows:
['Closed Date','Descriptor','Location Type','Incident Zip','Adddress Type','City','Facility Type','Due Date',
 'Resolution Action updated Date','X Coordinate (State Plane)','Y Coordinate (State Plane)','School Region',
 'School Code','School Zip','Latitude','Longitude','Location']
df.dropna(inplace=True)
```

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 290881 entries, 0 to 300697
Data columns (total 39 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   Unique Key                      290881 non-null  int64  
 1   Created Date                    290881 non-null  object 
 2   Closed Date                     290881 non-null  object 
 3   Agency                          290881 non-null  object 
 4   Agency Name                     290881 non-null  object 
 5   Complaint Type                  290881 non-null  object 
 6   Descriptor                      290881 non-null  object 
 7   Location Type                   290881 non-null  object 
 8   Incident Zip                    290881 non-null  float64
 9   Incident Address                290881 non-null  object 
 10  Street Name                     290881 non-null  object 
 11  Cross Street 1                  290881 non-null  object 
 12  Cross Street 2                  290881 non-null  object 
 13  Address Type                    290881 non-null  object 
 14  City                            290881 non-null  object 
 15  Facility Type                   290881 non-null  object 
 16  Status                          290881 non-null  object 
 17  Due Date                        290881 non-null  object 
 18  Resolution Description          290881 non-null  object 
 19  Resolution Action Updated Date  290881 non-null  object 
 20  Community Board                 290881 non-null  object 
 21  Borough                         290881 non-null  object 
 22  X Coordinate (State Plane)      290881 non-null  float64
 23  Y Coordinate (State Plane)      290881 non-null  float64
 24  Park Facility Name              290881 non-null  object 
 25  Park Borough                    290881 non-null  object 
 26  School Name                     290881 non-null  object 
 27  School Number                   290881 non-null  object 
 28  School Region                   290881 non-null  object 
 29  School Code                     290881 non-null  object 
 30  School Phone Number             290881 non-null  object 
 31  School Address                  290881 non-null  object 
 32  School City                     290881 non-null  object 
 33  School State                    290881 non-null  object 
 34  School Zip                      290881 non-null  object 
 35  School Not Found                290881 non-null  object 
 36  Latitude                        290881 non-null  float64
 37  Longitude                       290881 non-null  float64
 38  Location                        290881 non-null  object 
dtypes: float64(5), int64(1), object(33)
memory usage: 88.8+ MB
```

In [11]:
```
df.shape
```

Out[11]: (290881, 39)

In [12]:
```
df.isnull().sum()
```

...

Q-2. Convert the columns ‘Created Date’ and Closed Date’ to datetime datatype

In [13]:
```
# convert the columns ‘Created Date’ and Closed Date’ to datetime datatype.
df['Created Date'] = pd.to_datetime(df['Created Date'])
df['Closed Date'] = pd.to_datetime(df['Closed Date'])
```

create a new column ‘Request_Closing_Time’ as the time elapsed between request creation and request closing.

In [14]:
```
d = df['Closed Date'] - df['Created Date']
```

In [15]:
```
df['Request_Closing_Time'] = d.dt.seconds/3600
```

In [16]:
```
df.keys()
```

Out[16]:
```
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',
       'Address Type', 'City', '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', 'Latitude', 'Longitude', 'Location',
       'Request_Closing_Time'],
      dtype='object')
```

In [17]:
```
df['Request_Closing_Time']
```

...

In [18]:
```
df['Agency Name'].unique()
```

Out[18]: array(['New York City Police Department'], dtype=object)

In [19]:
```
df['Agency Name'].value_counts()
```

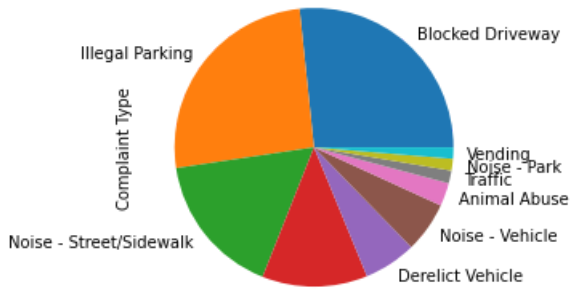
Out[19]:
```
New York City Police Department    290881
Name: Agency Name, dtype: int64
```

In [20]:
```
df['Complaint Type'].value_counts()
```

Out[20]:
```
Blocked Driveway            76675
Illegal Parking             74020
Noise - Street/Sidewalk     47745
Noise - Commercial          35144
Derelict Vehicle            17496
Noise - Vehicle             16867
Animal Abuse                 7743
Traffic                      4256
Noise - Park                 3927
Vending                      3773
Drinking                     1270
Noise - House of Worship      920
Posting Advertisement         647
Disorderly Youth              285
Graffiti                      113
Name: Complaint Type, dtype: int64
```

In [21]:
```
df['Complaint Type'].value_counts().head(10).plot(kind='pie')
```

Out[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x68a06c4f0>
```

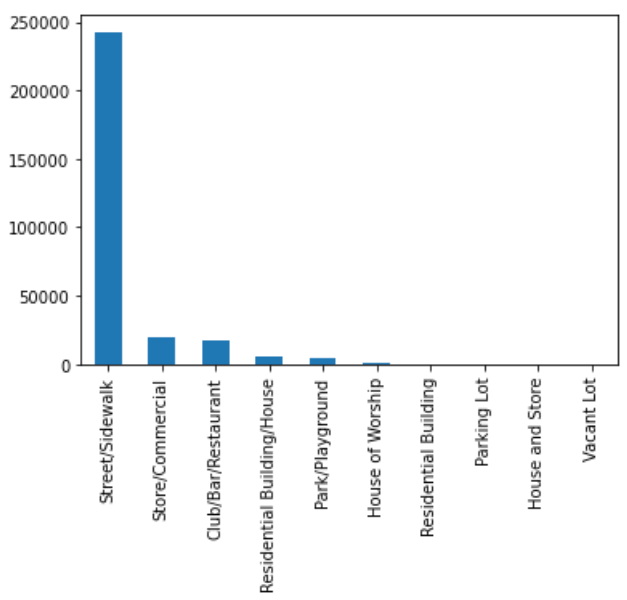


In [22]:
```
df['Location Type'].value_counts()
```

Out[22]:
```
Street/Sidewalk              242760
Store/Commercial              19425
Club/Bar/Restaurant           17172
Residential Building/House     5752
Park/Playground                4246
House of Worship                920
Residential Building            226
Parking Lot                     116
House and Store                  93
Vacant Lot                       77
Commercial                       62
Subway Station                   22
Roadway Tunnel                    5
Highway                           5
Name: Location Type, dtype: int64
```

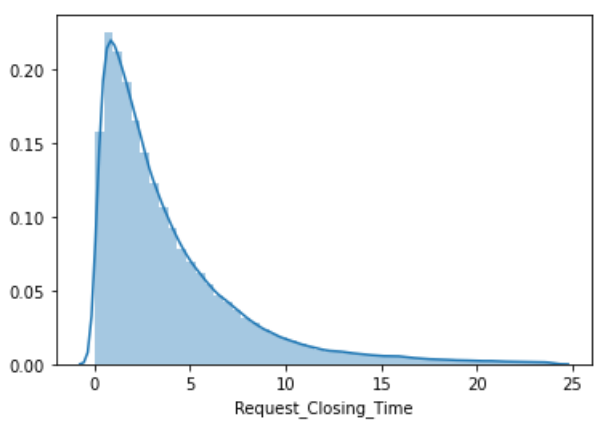
In [23]:
```
df['Location Type'].value_counts().head(10).plot(kind='bar')
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x68c483880>
```



In [24]:
```
sns.distplot(df['Request_Closing_Time'])
```

Out[24]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x68c50e9d0>
```



In [25]:
```
df['Request_Closing_Time'].describe()
```

Out[25]:
```
count    290881.000000
mean          3.925422
std           3.857343
min           0.000000
25%           1.266667
50%           2.684722
75%           5.257500
max          23.998611
Name: Request_Closing_Time, dtype: float64
```

Q-3. Major insights/patterns with 4 major conclusions:

1) Top 5 "location type" having maximum complaints

- Street/Sidewalk
- Store/Commercial
- Club/Bar/Restaurant
- Residential Building/House
- Park/Playground

2) Top 5 "complaint type" received

- Street/Sidewalk
- Store/Commercial
- Club/Bar/Restaurant
- Residential Building/House
- Park/Playground

3) Out of the 3 Agencies, maximum complaints falls under the New York City Police Department

4) The mean/average time to resolve the complaints is 3.92

Q.4. 'Request_Closing_Time', grouping them for different locations.

In [32]:
```
df_Loc_hours = df[['Location Type','Request_Closing_Time']]
```

In [33]:
```
df_Request_Closing_Time_Loc = df_Loc_hours.groupby(by='Location Type').mean()
```

In [36]:
```
df_Request_Closing_Time_Loc.head()
```

Out[36]:

Location Type	Request_Closing_Time
Club/Bar/Restaurant	2.913868
Commercial	4.568575
Highway	6.781389
House and Store	4.497133
House of Worship	2.964972

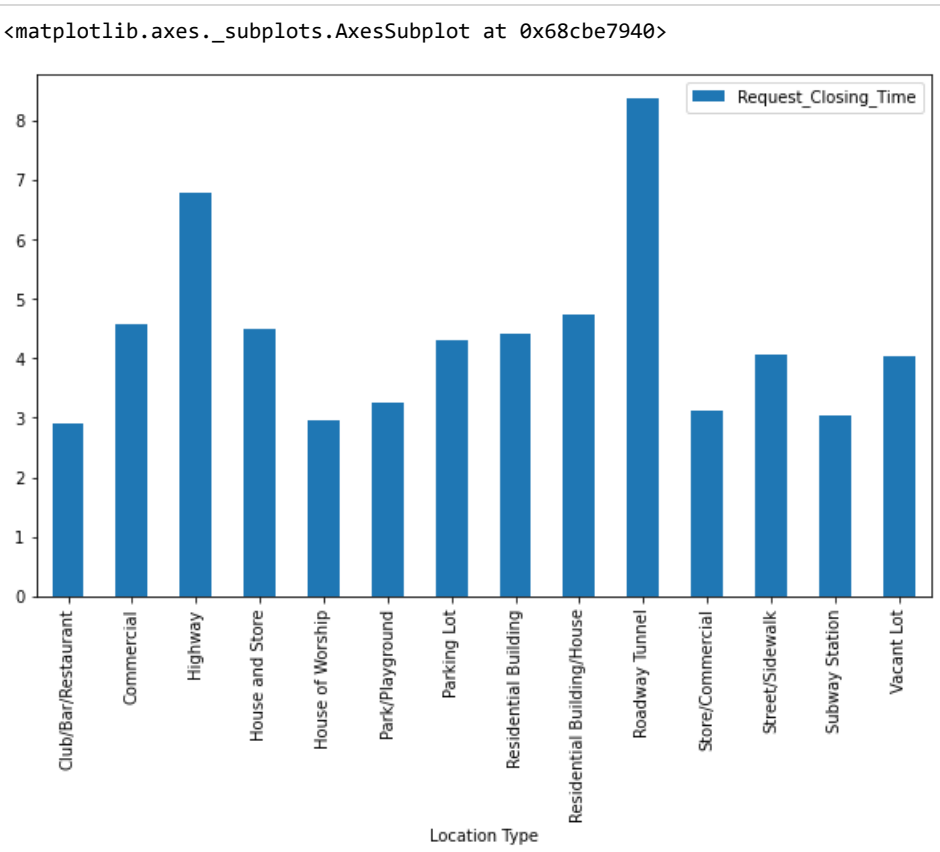
In [39]:
```
df1 = df_Request_Closing_Time_Loc.reset_index()
```

In [40]:
```
df1.keys()
```

Out[40]:
```
Index(['Location Type', 'Request_Closing_Time'], dtype='object')
```

In [41]:
```
fig,ax = plt.subplots(figsize=(10,6))
df_Request_Closing_Time_Loc.plot(kind='bar',ax=ax)
```

Out[41]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x68cbe7940>
```



In [42]:
```
df_Loc_complaint_hours = df[['Location Type','Complaint Type','Request_Closing_Time']]
```

In [43]:
```
df_Loc_Request_Closing_Time = df_Loc_complaint_hours.groupby(by=['Location Type','Complaint Type']).mean()
```

In [44]:
```
df_Loc_Request_Closing_Time.sort_values(by='Request_Closing_Time')
```

Out[44]:

Location Type	Complaint Type	Request_Closing_Time
Street/Sidewalk	Posting Advertisement	1.777614
Parking Lot	Posting Advertisement	2.115754
Store/Commercial	Posting Advertisement	2.369167
	Disorderly Youth	2.839479
Club/Bar/Restaurant	Noise - Commercial	2.889850
House of Worship	Noise - House of Worship	2.964972
Subway Station	Animal Abuse	3.035606
Store/Commercial	Noise - Commercial	3.066891
Street/Sidewalk	Traffic	3.202763
	Noise - Street/Sidewalk	3.217912
Park/Playground	Noise - Park	3.243770
Store/Commercial	Drinking	3.252596
Park/Playground	Animal Abuse	3.266972
	Drinking	3.356238
	Disorderly Youth	3.368008
Street/Sidewalk	Noise - Vehicle	3.383090
	Drinking	3.408078
Park/Playground	Vending	3.474647
Residential Building/House	Drinking	3.595236
	Posting Advertisement	3.600509
Highway	Traffic	3.687778
Street/Sidewalk	Vending	3.786017
Residential Building/House	Disorderly Youth	3.854477
Store/Commercial	Vending	3.862727
Club/Bar/Restaurant	Drinking	4.019785
Vacant Lot	Derelict Vehicle	4.045354
Residential Building/House	Vending	4.077681
Street/Sidewalk	Illegal Parking	4.132335
Store/Commercial	Animal Abuse	4.141185
Street/Sidewalk	Animal Abuse	4.241630
	Blocked Driveway	4.382400
Residential Building	Animal Abuse	4.401292
Parking Lot	Animal Abuse	4.445872
House and Store	Animal Abuse	4.497133
Commercial	Animal Abuse	4.568575
Residential Building/House	Animal Abuse	4.848593
	Graffiti	5.021657
Store/Commercial	Graffiti	5.560642
Street/Sidewalk	Derelict Vehicle	5.588368
	Graffiti	7.237522
Roadway Tunnel	Derelict Vehicle	8.364167
Highway	Derelict Vehicle	8.843796

In [45]:
```
df2 = df_Loc_Request_Closing_Time.reset_index()
df2.sort_values(by='Request_Closing_Time').reset_index()
```

Out[45]:

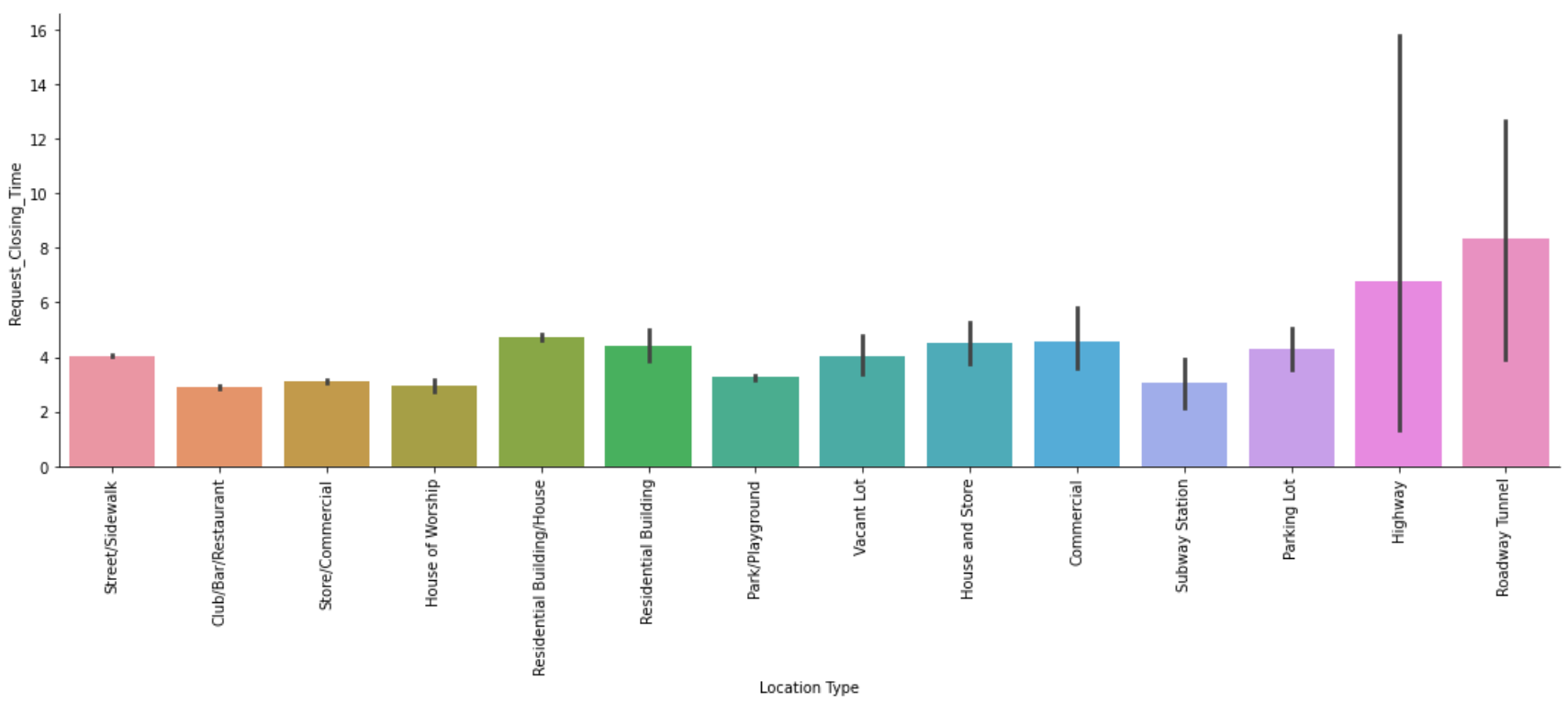
	index	Location Type	Complaint Type	Request_Closing_Time
0	37	Street/Sidewalk	Posting Advertisement	1.777614
1	12	Parking Lot	Posting Advertisement	2.115754
2	26	Store/Commercial	Posting Advertisement	2.369167
3	22	Store/Commercial	Disorderly Youth	2.839479
4	1	Club/Bar/Restaurant	Noise - Commercial	2.889850
5	6	House of Worship	Noise - House of Worship	2.964972
6	40	Subway Station	Animal Abuse	3.035606
7	25	Store/Commercial	Noise - Commercial	3.066891
8	38	Street/Sidewalk	Traffic	3.202763
9	35	Street/Sidewalk	Noise - Street/Sidewalk	3.217912
10	9	Park/Playground	Noise - Park	3.243770
11	23	Store/Commercial	Drinking	3.252596
12	7	Park/Playground	Animal Abuse	3.266972
13	8	Park/Playground	Drinking	3.356238
14	31	Street/Sidewalk	Disorderly Youth	3.368008
15	36	Street/Sidewalk	Noise - Vehicle	3.383090
16	32	Street/Sidewalk	Drinking	3.408078
17	10	Park/Playground	Vending	3.474647
18	16	Residential Building/House	Drinking	3.595236
19	18	Residential Building/House	Posting Advertisement	3.600509
20	4	Highway	Traffic	3.687778
21	39	Street/Sidewalk	Vending	3.786017
22	15	Residential Building/House	Disorderly Youth	3.854477
23	27	Store/Commercial	Vending	3.862727
24	0	Club/Bar/Restaurant	Drinking	4.019785
25	41	Vacant Lot	Derelict Vehicle	4.045354
26	19	Residential Building/House	Vending	4.077681
27	34	Street/Sidewalk	Illegal Parking	4.132335
28	21	Store/Commercial	Animal Abuse	4.141185
29	28	Street/Sidewalk	Animal Abuse	4.241630
30	29	Street/Sidewalk	Blocked Driveway	4.382400
31	13	Residential Building	Animal Abuse	4.401292
32	11	Parking Lot	Animal Abuse	4.445872
33	5	House and Store	Animal Abuse	4.497133
34	2	Commercial	Animal Abuse	4.568575
35	14	Residential Building/House	Animal Abuse	4.848593
36	17	Residential Building/House	Graffiti	5.021657
37	24	Store/Commercial	Graffiti	5.560642
38	30	Street/Sidewalk	Derelict Vehicle	5.588368
39	33	Street/Sidewalk	Graffiti	7.237522
40	20	Roadway Tunnel	Derelict Vehicle	8.364167
41	3	Highway	Derelict Vehicle	8.843796

In [46]:
```
df2.keys()
```

Out[46]: Index(['Location Type', 'Complaint Type', 'Request_Closing_Time'], dtype='object')

In [48]:
```
fig = sns.catplot(x='Location Type',y='Request_Closing_Time',kind='bar',data=df,aspect=3)
fig.set_xticklabels(rotation=90)
```

Out[48]: <seaborn.axisgrid.FacetGrid at 0x68cb0c670>



Q-5. Statistical Test

Q-5-I. Average response time across complaint types is similar or not

Hypothesis Testing:

- H0: Average response time across complaint types is not similar

  if p-val >0.05

- HA: Average response time across complaint types is similar

  if p-val <0.05

In [70]:
```
from scipy import stats
```

In [71]:
```
df['Complaint Type'].value_counts().head()
Top5_Complaints_type = df['Complaint Type'].value_counts().head()
```

In [72]:
```
Top5_Complaints_type_names = Top5_Complaints_type.index
```

In [73]:
```
Top5_Complaints_type_names
```

Out[73]:
```
Index(['Blocked Driveway', 'Illegal Parking', 'Noise - Street/Sidewalk',
       'Noise - Commercial', 'Derelict Vehicle'],
      dtype='object')
```

In [74]:
```
data = df.loc[df['Complaint Type'].isin(Top5_Complaints_type_names),['Complaint Type', 'Request_Closing_Time']]
data.head()
```

Out[74]:

	Complaint Type	Request_Closing_Time
0	Noise - Street/Sidewalk	0.920833
1	Blocked Driveway	1.437778
2	Blocked Driveway	4.858611
3	Illegal Parking	7.753889
4	Illegal Parking	3.450556

In [75]:
```
d1 = df[df['Complaint Type']== Top5_Complaints_type_names[0]].Request_Closing_Time
d1.head()
```

Out[75]:
```
1     1.437778
2     4.858611
7     1.798611
9     1.383889
10    7.800556
Name: Request_Closing_Time, dtype: float64
```

In [76]:
```
d2 = df[df['Complaint Type']== Top5_Complaints_type_names[1]].Request_Closing_Time
d2.head()
```

Out[76]:
```
3    7.753889
4    3.450556
5    1.891667
6    1.957778
8    8.550556
Name: Request_Closing_Time, dtype: float64
```

In [77]:
```
d3 = df[df['Complaint Type']== Top5_Complaints_type_names[2]].Request_Closing_Time
d3.head()
```

Out[77]:
```
0     0.920833
12    2.482500
19    0.784722
38    0.491111
54    1.495556
Name: Request_Closing_Time, dtype: float64
```

In [78]:
```
d4 = df[df['Complaint Type']== Top5_Complaints_type_names[3]].Request_Closing_Time
d4.head()
```

Out[78]:
```
17    0.852222
18    2.933611
22    1.261667
29    2.499722
30    1.985833
Name: Request_Closing_Time, dtype: float64
```

In [79]:
```
d5 = df[df['Complaint Type']== Top5_Complaints_type_names[4]].Request_Closing_Time
d5.head()
```

Out[79]:
```
14     10.489722
151     3.950278
255     1.364722
256     4.133056
295     0.753333
Name: Request_Closing_Time, dtype: float64
```

In [80]:
```
stats.f_oneway(d1,d2,d3,d4,d5)
```

Out[80]:
```
F_onewayResult(statistic=2080.3676275129005, pvalue=0.0)
```

Here we can observe that p-value is less than 0.05, so we reject the null hypothesis and accept the alternate hypothesis. Hence we can say that response time across complaint type is similar.

Q-5-II. Relation between complaint or service request and location

Hypothesis Testing:

- H0: complaints are not depends on location

  if p-val >0.05

- HA: complaints are depends on location

  if p-val <0.05

In [81]:
```
df['Location Type'].value_counts().head(3)
```

Out[81]:
```
Street/Sidewalk        242760
Store/Commercial        19425
Club/Bar/Restaurant     17172
Name: Location Type, dtype: int64
```

In [82]:
```
df['Complaint Type'].value_counts().head(2)
```

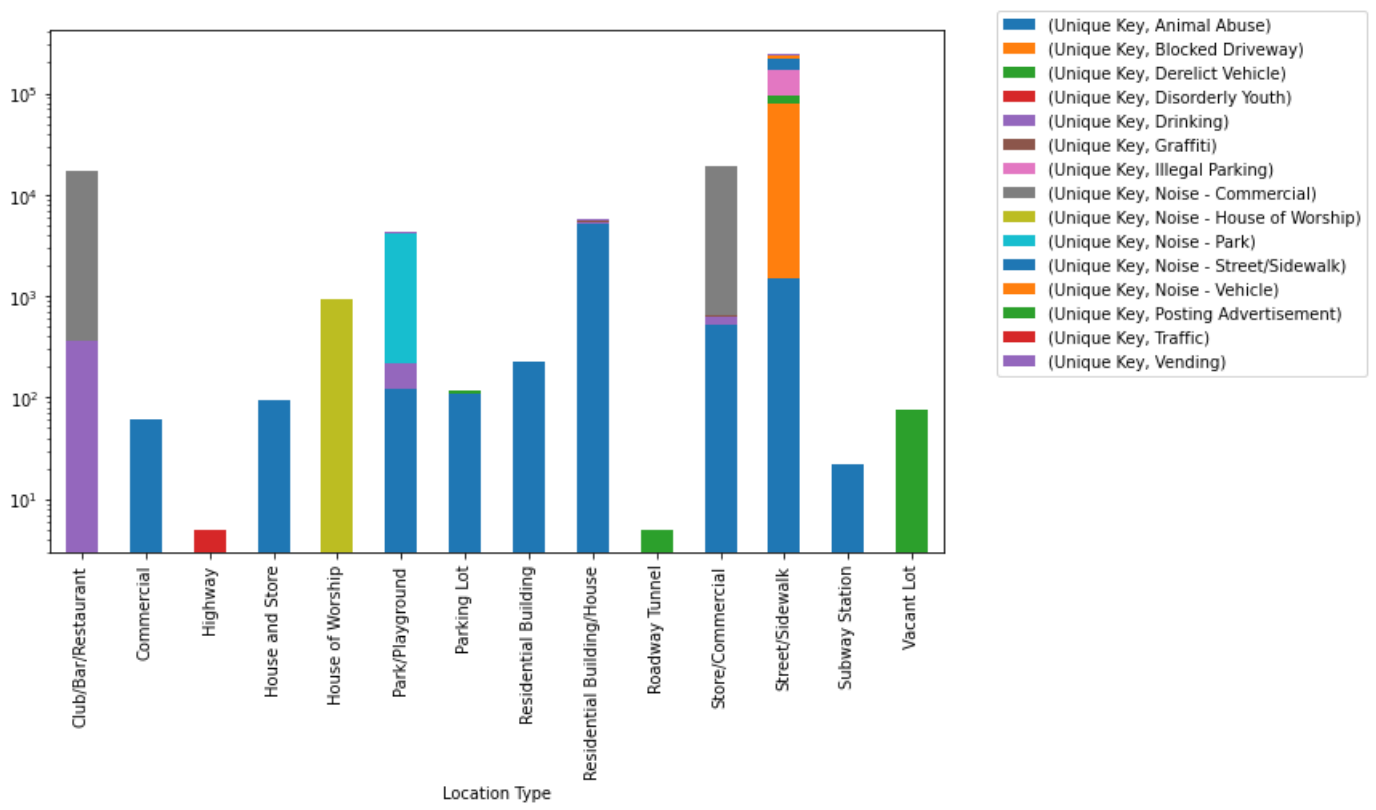
Out[82]:
```
Blocked Driveway    76675
Illegal Parking     74020
Name: Complaint Type, dtype: int64
```

In [83]:
```
group = df[['Unique Key','Location Type','Complaint Type']].groupby(by=['Location Type','Complaint Type'])
```

In [84]:
```
group_count = group.count()
```

In [85]:
```
group_un = group_count.unstack() #.reset_index()
```

In [86]:
```
fig,ax = plt.subplots(figsize=(10,6))
fig= group_un.plot(kind='bar',stacked=True,ax=ax)
fig.legend(bbox_to_anchor=[1.05,1.05])
plt.yscale('log')
```



In [87]:
```
from scipy.stats import chi2_contingency
```

In [88]:
```
pd.crosstab(df['Location Type'],df['Complaint Type'], margins=True)
```

Out[88]:

Complaint Type	Animal Abuse	Blocked Driveway	Derelict Vehicle	Disorderly Youth	Drinking	Graffiti	Illegal Parking	Noise - Commercial	Noise - House of Worship	Noise - Park	Noise - Street/Sidewalk	Noise - Vehicle	Posting Advertisement	Traffic	Vendin
Location Type															
Club/Bar/Restaurant	0	0	0	0	365	0	0	16807	0	0	0	0	0	0	
Commercial	62	0	0	0	0	0	0	0	0	0	0	0	0	0	
Highway	0	0	3	0	0	0	0	0	0	0	0	0	0	2	
House and Store	93	0	0	0	0	0	0	0	0	0	0	0	0	0	
House of Worship	0	0	0	0	0	0	0	0	920	0	0	0	0	0	
Park/Playground	120	0	0	0	96	0	0	0	0	3927	0	0	0	0	10
Parking Lot	109	0	0	0	0	0	0	0	0	0	0	0	7	0	
Residential Building	226	0	0	0	0	0	0	0	0	0	0	0	0	0	
Residential Building/House	5075	0	0	77	289	56	0	0	0	0	0	0	54	0	20
Roadway Tunnel	0	0	5	0	0	0	0	0	0	0	0	0	0	0	
Store/Commercial	520	0	0	8	90	32	0	18337	0	0	0	0	6	0	43
Street/Sidewalk	1516	76675	17411	200	430	25	74020	0	0	0	47745	16867	580	4254	303
Subway Station	22	0	0	0	0	0	0	0	0	0	0	0	0	0	
Vacant Lot	0	0	77	0	0	0	0	0	0	0	0	0	0	0	
All	7743	76675	17496	285	1270	113	74020	35144	920	3927	47745	16867	647	4256	377

In [89]:
```
ch2,p_value,dof,exp_freq = stats.chi2_contingency(pd.crosstab(df['Location Type'],df['Complaint Type'], margins=True))
print(ch2)
print(p_value)
print(dof)
```

```
1036031.2859946301
0.0
210
```

Here we can observe that p-value is less than 0.05, so we reject the null hypothesis and accept the alternate hypothesis. Hence we can say that complaints and location are related.

Regards:

Anu Verma

In [ ]:
```
```