In [1]:

```
#Importing required libraries
import numpy as np
import pandas as pd
import matplotlib as mpl
from matplotlib import pyplot as plt
from datetime import datetime
import seaborn as sns
```

1. Import 311 NYC service request.

In [2]:

```
#Import a 311 NYC service request.

df = pd.read_csv('311_Service_Requests_from_2010_to_Present.csv')

df.head()
```

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

Out[2]:

	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 [3]:

```
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.describe()
```

Out[4]:

	Unique Key	Incident Zip	X Coordinate (State Plane)	Y Coordinate (State Plane)	School or Citywide Complaint	Vehicle Type	Taxi Company Borough	T Pick Locat
count	3.006980e+05	298083.000000	2.971580e+05	297158.000000	0.0	0.0	0.0	
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 [5]:

```
df.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
```

In [6]:

```
df.shape
```

Out[6]:

```
(300698, 53)
```

In [7]:

```
df1=df.drop(['Unique Key','Agency','Agency Name','Incident Address','Address Type','Des
'Bridge Highway Direction',"Incident Zip",'Community Board','Facility Type','Park Facil
df1.head()
```

Out[7]:

	Created Date	Closed Date	Complaint Type	Location Type	City	Status	Due Date	Resolution Action Updated Date	Boroug
0	12/31/2015 11:59:45 PM	01-01-16 0:55	Noise - Street/Sidewalk	Street/Sidewalk	NEW YORK	Closed	01-01-16 7:59	01-01-16 0:55	MANHATTA
1	12/31/2015 11:59:44 PM	01-01-16 1:26	Blocked Driveway	Street/Sidewalk	ASTORIA	Closed	01-01-16 7:59	01-01-16 1:26	QUEEN
2	12/31/2015 11:59:29 PM	01-01-16 4:51	Blocked Driveway	Street/Sidewalk	BRONX	Closed	01-01-16 7:59	01-01-16 4:51	BRON
3	12/31/2015 11:57:46 PM	01-01-16 7:43	Illegal Parking	Street/Sidewalk	BRONX	Closed	01-01-16 7:57	01-01-16 7:43	BRON
4	12/31/2015 11:56:58 PM	01-01-16 3:24	Illegal Parking	Street/Sidewalk	ELMHURST	Closed	01-01-16 7:56	01-01-16 3:24	QUEEN

In [8]:

```
df1.shape
```

Out[8]:

```
(300698, 9)
```

In [9]:

```
df1.isnull().sum()
```

Out[9]:

```
Created Date                         0
Closed Date                       2164
Complaint Type                       0
Location Type                      131
City                              2614
Status                               0
Due Date                             3
Resolution Action Updated Date    2187
Borough                              0
dtype: int64
```

In [10]:

```
df1["Complaint Type"].unique()
```

Out[10]:

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

In [11]:

```
df1=df1.dropna(axis=0)
df1.head()
```

Out[11]:

	Created Date	Closed Date	Complaint Type	Location Type	City	Status	Due Date	Resolution Action Updated Date	Boroug
0	12/31/2015 11:59:45 PM	01-01-16 0:55	Noise - Street/Sidewalk	Street/Sidewalk	NEW YORK	Closed	01-01-16 7:59	01-01-16 0:55	MANHATTA
1	12/31/2015 11:59:44 PM	01-01-16 1:26	Blocked Driveway	Street/Sidewalk	ASTORIA	Closed	01-01-16 7:59	01-01-16 1:26	QUEEN
2	12/31/2015 11:59:29 PM	01-01-16 4:51	Blocked Driveway	Street/Sidewalk	BRONX	Closed	01-01-16 7:59	01-01-16 4:51	BRON
3	12/31/2015 11:57:46 PM	01-01-16 7:43	Illegal Parking	Street/Sidewalk	BRONX	Closed	01-01-16 7:57	01-01-16 7:43	BRON

	Created Date	Closed Date	Complaint Type	Location Type	City	Status	Due Date	Resolution Action Updated Date	Boroug
4	12/31/2015 11:56:58 PM	01-01-16 3:24	Illegal Parking	Street/Sidewalk	ELMHURST	Closed	01-01-16 7:56	01-01-16 3:24	QUEEN

In [12]:

```
df1.shape
```

Out[12]:

```
(297904, 9)
```

In [13]:

```
df1['City'].unique()
```

Out[13]:

```
array(['NEW YORK', 'ASTORIA', 'BRONX', 'ELMHURST', 'BROOKLYN',
       'KEW GARDENS', 'JACKSON HEIGHTS', 'MIDDLE VILLAGE', 'REGO PARK',
       'SAINT ALBANS', 'JAMAICA', 'SOUTH RICHMOND HILL', 'RIDGEWOOD',
       'HOWARD BEACH', 'FOREST HILLS', 'STATEN ISLAND', 'OZONE PARK',
       'RICHMOND HILL', 'WOODHAVEN', 'FLUSHING', 'CORONA',
       'QUEENS VILLAGE', 'OAKLAND GARDENS', 'HOLLIS', 'MASPETH',
       'EAST ELMHURST', 'SOUTH OZONE PARK', 'WOODSIDE', 'FRESH MEADOWS',
       'LONG ISLAND CITY', 'ROCKAWAY PARK', 'SPRINGFIELD GARDENS',
       'COLLEGE POINT', 'BAYSIDE', 'GLEN OAKS', 'FAR ROCKAWAY',
       'BELLEROSE', 'LITTLE NECK', 'CAMBRIA HEIGHTS', 'ROSEDALE',
       'SUNNYSIDE', 'WHITESTONE', 'ARVERNE', 'FLORAL PARK',
       'NEW HYDE PARK', 'CENTRAL PARK', 'BREEZY POINT', 'QUEENS',
       'Astoria', 'Long Island City', 'Woodside', 'East Elmhurst',
       'Howard Beach'], dtype=object)
```

In [14]:

```
df1['City']=df1['City'].str.upper()


df1['City'].unique()
```

Out[14]:

```
array(['NEW YORK', 'ASTORIA', 'BRONX', 'ELMHURST', 'BROOKLYN',
       'KEW GARDENS', 'JACKSON HEIGHTS', 'MIDDLE VILLAGE', 'REGO PARK',
       'SAINT ALBANS', 'JAMAICA', 'SOUTH RICHMOND HILL', 'RIDGEWOOD',
       'HOWARD BEACH', 'FOREST HILLS', 'STATEN ISLAND', 'OZONE PARK',
       'RICHMOND HILL', 'WOODHAVEN', 'FLUSHING', 'CORONA',
       'QUEENS VILLAGE', 'OAKLAND GARDENS', 'HOLLIS', 'MASPETH',
       'EAST ELMHURST', 'SOUTH OZONE PARK', 'WOODSIDE', 'FRESH MEADOWS',
       'LONG ISLAND CITY', 'ROCKAWAY PARK', 'SPRINGFIELD GARDENS',
       'COLLEGE POINT', 'BAYSIDE', 'GLEN OAKS', 'FAR ROCKAWAY',
       'BELLEROSE', 'LITTLE NECK', 'CAMBRIA HEIGHTS', 'ROSEDALE',
       'SUNNYSIDE', 'WHITESTONE', 'ARVERNE', 'FLORAL PARK',
       'NEW HYDE PARK', 'CENTRAL PARK', 'BREEZY POINT', 'QUEENS'],
      dtype=object)
```

In [15]:

```
df1.isnull().sum()
```

Out[15]:

```
Created Date                      0
Closed Date                       0
```

```
Complaint Type                    0
Location Type                     0
City                              0
Status                            0
Due Date                          0
Resolution Action Updated Date    0
Borough                           0
dtype: int64
```

2.Read or convert the columns ‘Created Date’ and Closed Date’ to datetime datatype.

In [16]:

```
df1['Created Date'].dtype
```

Out[16]:

```
dtype('O')
```

In [17]:

```
df1['Created Date'] = pd.to_datetime(df1['Created Date'])
df1['Closed Date'] = pd.to_datetime(df1['Closed Date'])
```

In [18]:

```
df1['Created Date'].dtype
```

Out[18]:

```
dtype('<M8[ns]')
```

In [19]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 297904 entries, 0 to 300697
Data columns (total 9 columns):
 #   Column                          Non-Null Count   Dtype         
---  ------                          --------------   -----         
 0   Created Date                    297904 non-null  datetime64[ns]
 1   Closed Date                     297904 non-null  datetime64[ns]
 2   Complaint Type                  297904 non-null  object        
 3   Location Type                   297904 non-null  object        
 4   City                            297904 non-null  object        
 5   Status                          297904 non-null  object        
 6   Due Date                        297904 non-null  object        
 7   Resolution Action Updated Date  297904 non-null  object        
 8   Borough                         297904 non-null  object        
dtypes: datetime64[ns](2), object(7)
memory usage: 22.7+ MB
```

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

In [20]:

```
df1['Request_Closing_Time']= df1['Closed Date']-df1['Created Date']
df1['Request_Closing_Time(in seconds)'] = df1['Request_Closing_Time'].astype('timedelta
df1['Request_Closing_Time(in hour)'] =df1['Request_Closing_Time(in seconds)'].apply(lam
```

In [21]:

```
df1.head()
```

Out[21]:

	Created Date	Closed Date	Complaint Type	Location Type	City	Status	Due Date	Resolution Action Updated Date	Borough
0	2015-12-31 23:59:45	2016-01-01 00:55:00	Noise - Street/Sidewalk	Street/Sidewalk	NEW YORK	Closed	01-01-16 7:59	01-01-16 0:55	MANHATTAN
1	2015-12-31 23:59:44	2016-01-01 01:26:00	Blocked Driveway	Street/Sidewalk	ASTORIA	Closed	01-01-16 7:59	01-01-16 1:26	QUEENS
2	2015-12-31 23:59:29	2016-01-01 04:51:00	Blocked Driveway	Street/Sidewalk	BRONX	Closed	01-01-16 7:59	01-01-16 4:51	BRONX
3	2015-12-31 23:57:46	2016-01-01 07:43:00	Illegal Parking	Street/Sidewalk	BRONX	Closed	01-01-16 7:57	01-01-16 7:43	BRONX
4	2015-12-31 23:56:58	2016-01-01 03:24:00	Illegal Parking	Street/Sidewalk	ELMHURST	Closed	01-01-16 7:56	01-01-16 3:24	QUEENS

4.Insight

Insight 1: Greater Number of complaints were received regarding ***Blocked Driveway*** and ***Illegal Parking***

In [22]:

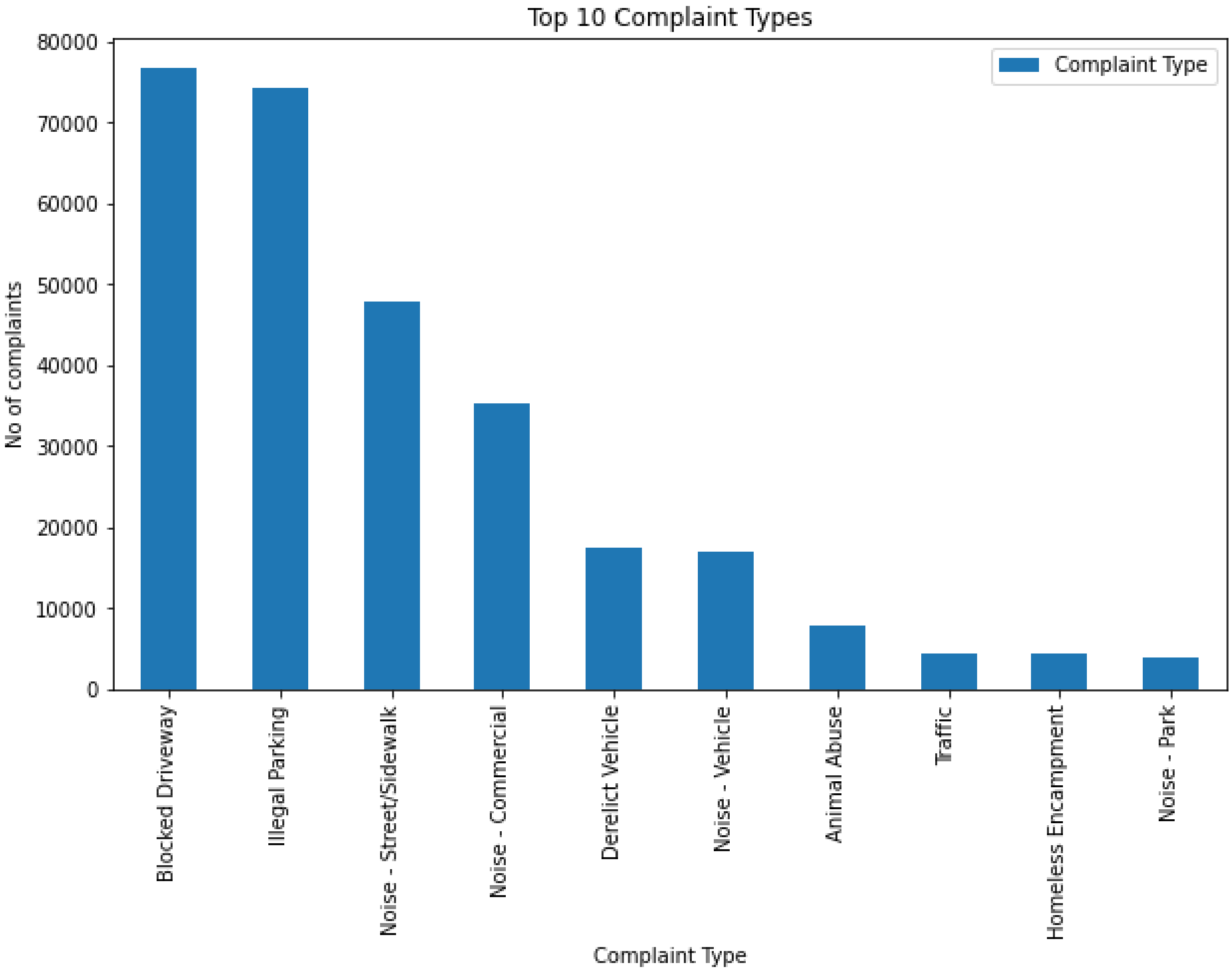
```
(df1['Complaint Type'].value_counts().nlargest(10)).plot(kind='bar', figsize=(10,6))

plt.xlabel('Complaint Type')
plt.ylabel('No of complaints')

plt.title("Top 10 Complaint Types")
plt.legend()
```

Out[22]:

```
<matplotlib.legend.Legend at 0x1fb23c56460>
```



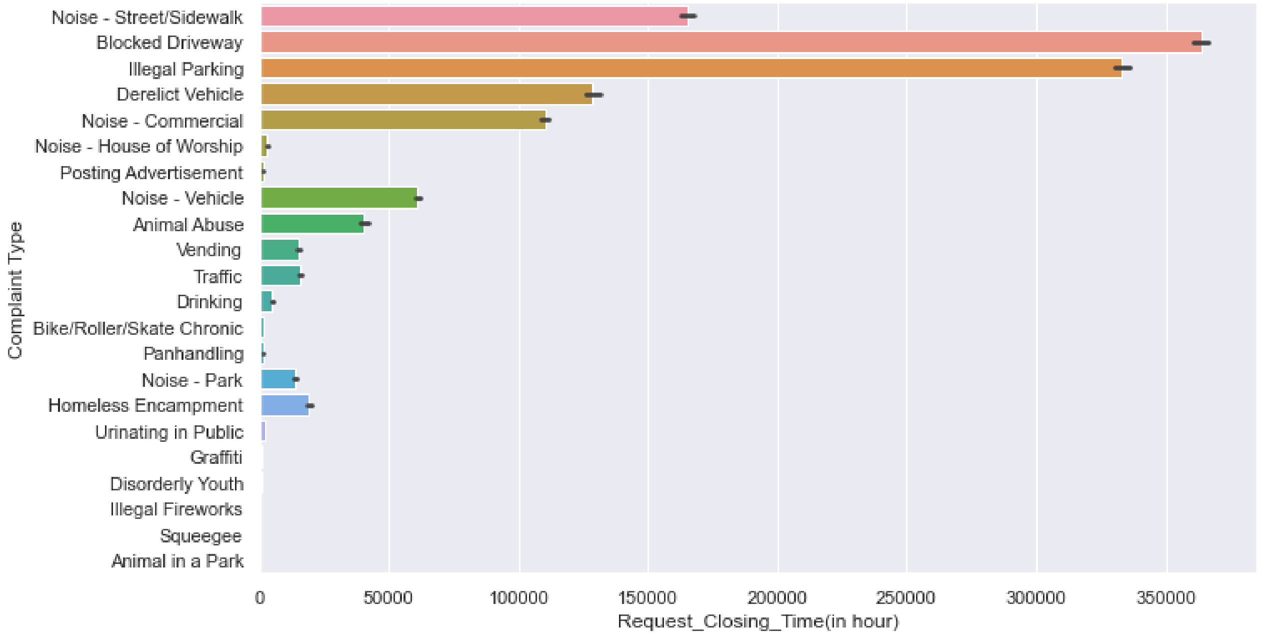
Insight 2: Blocked driveway complaint type took majority share of total resolving time of all complaint type.

In [23]:

```
sns.set(rc={'figure.figsize':(10.7,6.27)})
sns.barplot(x=df1['Request_Closing_Time(in hour)'],y=df1['Complaint Type'],estimator=np
```

Out[23]:

```
<AxesSubplot:xlabel='Request_Closing_Time(in hour)', ylabel='Complaint Type'>
```



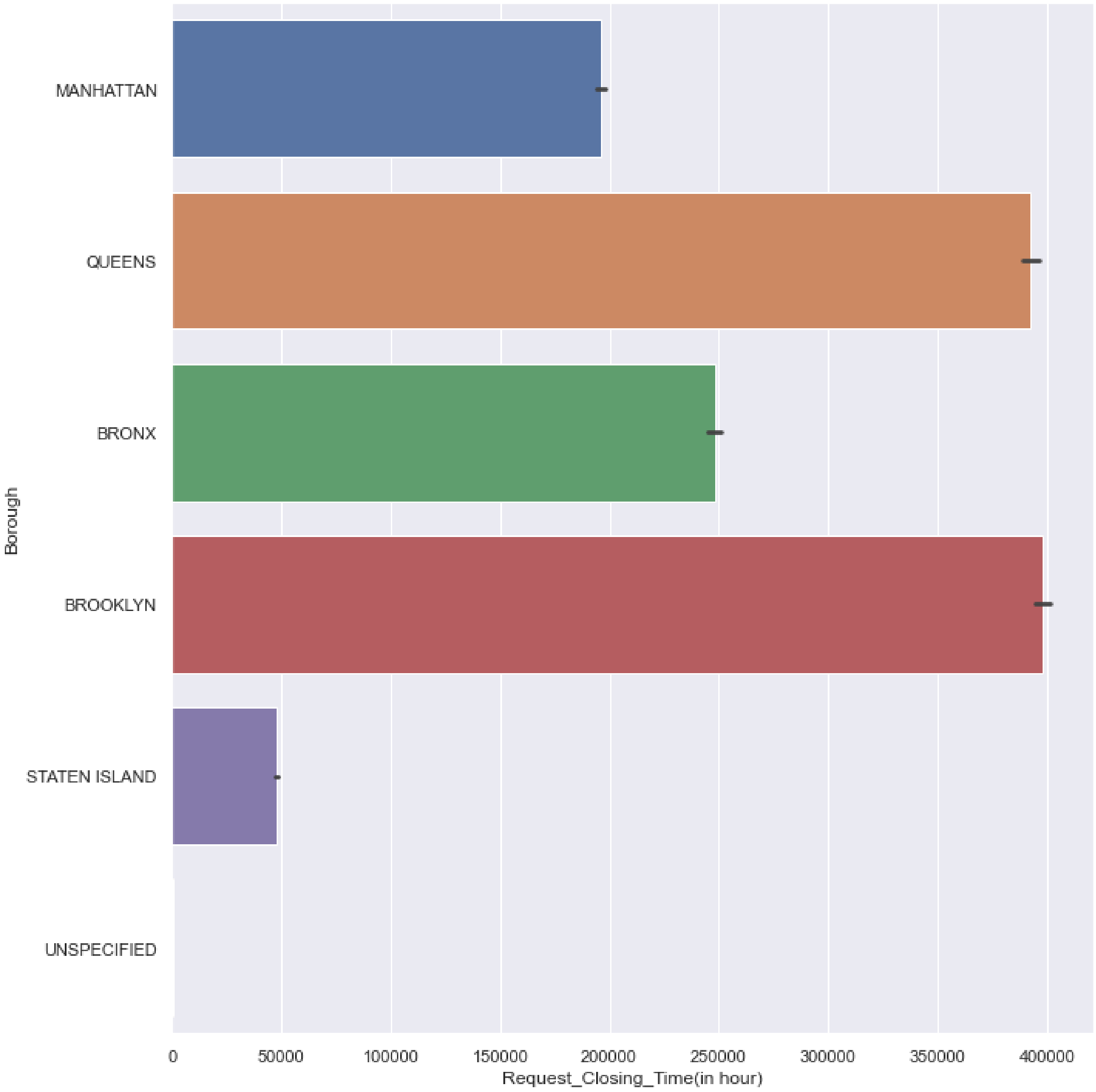
Insight 3: Total closing time in hour for each Borough. Longest Closing time was seen on Brooklyn, followed by Queens and Bronx.

In [24]:

```
sns.set(rc={'figure.figsize':(10.7,12.27)})
sns.barplot(x=df1['Request_Closing_Time(in hour)'],y= df1['Borough'].str.upper(),estima
```

Out[24]:

```
<AxesSubplot:xlabel='Request_Closing_Time(in hour)', ylabel='Borough'>
```



Insight 4: Complaint type against each Borough. In most of the Borough,

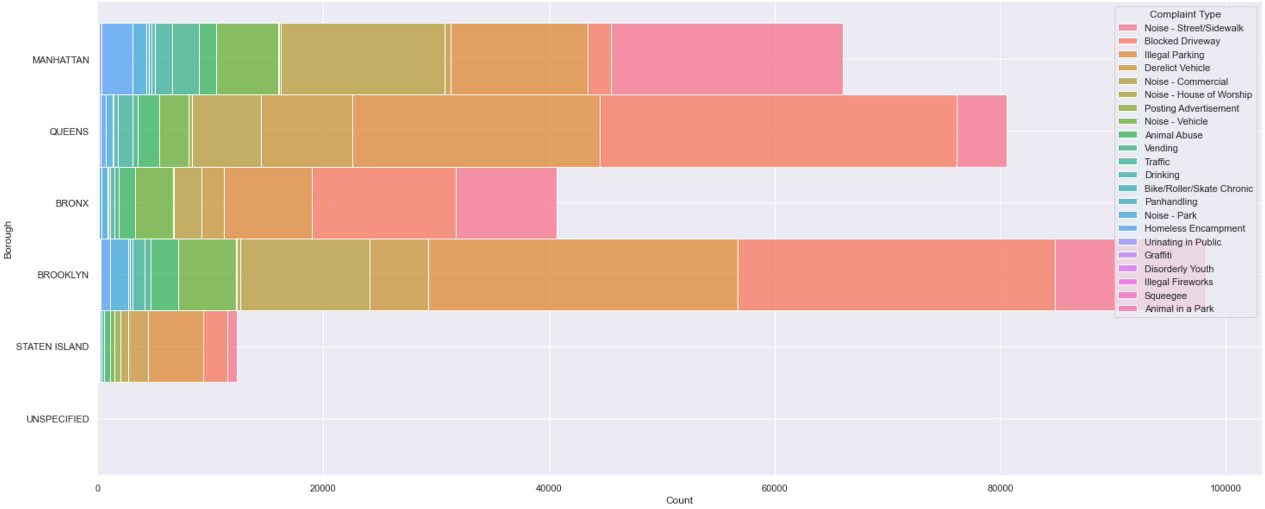
Blocked Driveway,***Illegal Parking*** and ***Noise*** were the most received complaints.

In [25]:

```
sns.set(rc={'figure.figsize':(24.2,10.10)})
sns.histplot(y=df1['Borough'].str.upper(),hue=df1['Complaint Type'],multiple='stack')
```

Out[25]:

```
<AxesSubplot:xlabel='Count', ylabel='Borough'>
```



5.Order the complaint types based on the average ‘Request_Closing_Time’, grouping them for different locations.

In [26]:

```
location_grp=pd.DataFrame(df1.groupby(["Location Type","Complaint Type"])["Request_Clos
```

In [27]:

```
location_grp
```

Out[27]:

		Request_Closing_Time(in hour)
Location Type	**Complaint Type**	
Subway Station	**Urinating in Public**	1.153333
Park/Playground	**Panhandling**	1.216667
Street/Sidewalk	**Posting Advertisement**	1.819103
Store/Commercial	**Illegal Fireworks**	1.925000
Parking Lot	**Posting Advertisement**	2.115714
...	**...**	...
Club/Bar/Restaurant	**Urinating in Public**	7.920000
Highway	**Derelict Vehicle**	8.196923
Street/Sidewalk	**Graffiti**	12.036800
Roadway Tunnel	**Derelict Vehicle**	17.964000
Park	**Animal in a Park**	336.830000

69 rows × 1 columns

In [28]:

```
city_complaint_grp = df1.groupby(['City','Complaint Type'])
city_complaint_grp['Request_Closing_Time(in hour)'].agg(['mean']).round(2)
```

Out[28]:

		mean
City	**Complaint Type**	

		mean
City	Complaint Type	
ARVERNE	Animal Abuse	2.15
	Blocked Driveway	2.53
	Derelict Vehicle	2.97
	Disorderly Youth	3.60
	Drinking	0.24
...	...	...
WOODSIDE	Noise - Street/Sidewalk	6.56
	Noise - Vehicle	5.48
	Traffic	4.84
	Urinating in Public	6.42
	Vending	7.30

746 rows × 1 columns

7. Statistical Testing

7 a. Null Hypothesis(h0): Average response time across complaint type is similar

Alternative Hypothesis(ha): Average response time across complaint type is not similar

In [29]:

```
complaint_grp = df1.groupby(['Complaint Type'])
mean_ser=complaint_grp['Request_Closing_Time(in hour)'].agg(['mean']).round(2)
```

In [30]:

```
mean_ser
```

Out[30]:

	mean
Complaint Type	
Animal Abuse	5.21
Animal in a Park	336.83
Bike/Roller/Skate Chronic	3.76
Blocked Driveway	4.74
Derelict Vehicle	7.35
Disorderly Youth	3.56
Drinking	3.86
Graffiti	7.15

	mean
Complaint Type	
Homeless Encampment	4.36
Illegal Fireworks	2.76
Illegal Parking	4.49
Noise - Commercial	3.14
Noise - House of Worship	3.20
Noise - Park	3.40
Noise - Street/Sidewalk	3.44
Noise - Vehicle	3.59
Panhandling	4.37
Posting Advertisement	1.98
Squeegee	4.05
Traffic	3.45
Urinating in Public	3.63
Vending	4.01

In [31]:

```
import scipy.stats as stats
z_score = stats.zscore(mean_ser,nan_policy='omit').round(3)
z_score=z_score.values
z_score
```

Out[31]:

```
array([[-0.202],
       [ 4.582],
       [-0.223],
       [-0.209],
       [-0.171],
       [-0.226],
       [-0.221],
       [-0.174],
       [-0.214],
       [-0.237],
       [-0.212],
       [-0.232],
       [-0.231],
       [-0.228],
       [-0.227],
       [-0.225],
       [-0.214],
       [-0.248],
       [-0.218],
       [-0.227],
       [-0.225],
       [-0.219]])
```

Let's take three random 'z_score' values and find the 'p_values' for them. Let's take the "significance level(alpha)=0.05"

In [32]:

```
P_value_1 = stats.norm.sf(abs(4.69)).round(5)
P_value_2 = stats.norm.sf(abs(-0.233))
P_value_3 = stats.norm.sf(abs(-0.221))

print('p-value for z-score 4.69 is : {}'.format(P_value_1))
print('p-value for z-score -0.233 is : {}'.format(P_value_2))
print('p-value for z-score -0.221 is : {}'.format(P_value_3))
```

```
p-value for z-score 4.69 is : 0.0
p-value for z-score -0.233 is : 0.4078807028841309
p-value for z-score -0.221 is : 0.41254621648911394
```

Conclusion : As we can see that "p-value" for any random "z-score" is greater than the "significance level(alpha)" except the one at 3rd index i.e "z_score=4.69" in z_score list which is less than "significance level(alpha)" , we "Reject" the null hypothesis

7 b. Relation between Complaint Type & Location

Null Hypthesis(h0): Complaint type and Location are independent

Alternative Hypothesis(ha): Complaint type and location are not independent

In [33]:

```
Contingency_tab = pd.crosstab(df1['Complaint Type'],df1['City'])
Contingency_tab
```

Out[33]:

City	ARVERNE	ASTORIA	BAYSIDE	BELLEROSE	BREEZY POINT	BRONX	BROOKLYN	CAMBRIA HEIGHTS
Complaint Type								
Animal Abuse	38	125	37	7	2	1414	2393	11
Animal in a Park	0	0	0	0	0	0	0	0
Bike/Roller/Skate Chronic	0	15	0	1	0	20	111	0
Blocked Driveway	35	2734	376	95	3	12750	28130	147
Derelict Vehicle	27	363	198	89	3	1951	5176	115
Disorderly Youth	2	3	1	2	0	63	72	0
Drinking	1	35	1	1	1	187	257	0
Graffiti	1	4	3	0	0	9	43	0
Homeless Encampment	4	32	2	1	0	246	854	5
Illegal Fireworks	0	4	0	1	0	24	61	1
Illegal Parking	58	1281	512	105	15	7855	27446	76
Noise - Commercial	2	1554	40	37	4	2433	11457	12

City	ARVERNE	ASTORIA	BAYSIDE	BELLEROSE	BREEZY POINT	BRONX	BROOKLYN	CAMBRIA HEIGHTS
Complaint Type								
Noise - House of Worship	11	19	2	1	0	79	340	2
Noise - Park	2	61	4	1	0	547	1554	0
Noise - Street/Sidewalk	29	500	15	13	1	8890	13353	25
Noise - Vehicle	7	204	16	10	1	3395	5175	77
Panhandling	1	1	0	1	0	19	49	0
Posting Advertisement	0	1	0	1	0	16	45	0
Squeegee	0	0	0	0	0	0	0	0
Traffic	0	47	9	7	0	355	1085	6
Urinating in Public	1	9	0	1	0	51	136	0
Vending	1	54	2	0	0	379	514	0

22 rows × 48 columns

In [34]:

```
Chi_val,p_val,dof,exp_val=stats.chi2_contingency(Contingency_tab)
print(f'Chi-square Value: {Chi_val}\nP-value: {p_val}\nDegree of Freedom: {dof}\nExpect
```

```
Chi-square Value: 118788.3273808606
P-value: 0.0
Degree of Freedom: 987
Expected Value: [[5.73365917e+00 1.83633466e+02 3.17436221e+01 ... 2.86161717e+01
  6.41387964e+01 9.54133009e+01]
 [7.38492937e-04 2.36519147e-02 4.08856544e-03 ... 3.68575111e-03
  8.26105054e-03 1.22891938e-02]
 [3.11644020e-01 9.98110801e+00 1.72537462e+00 ... 1.55538697e+00
  3.48616333e+00 5.18603980e+00]
 ...
 [3.31509480e+00 1.06173445e+02 1.83535703e+01 ... 1.65453368e+01
  3.70838559e+01 5.51661911e+01]
 [4.36449326e-01 1.39782816e+01 2.41634218e+00 ... 2.17827891e+00
  4.88228087e+00 7.26291356e+00]
 [2.80036522e+00 8.96880606e+01 1.55038402e+01 ... 1.39763682e+01
  3.13259036e+01 4.66006230e+01]]
```

Conclusion : Reject Null Hypothesis