Task 1

Importing the Data Set

In [1]:
```
#Importing packages And libraries for python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
import seaborn as sns

import itertools
```

In [2]:
```
#Imorting Dataset file
filename="311_Service_Requests_from_2010_to_Present.csv"
df=pd.read_csv(filename, low_memory=False)
df.head()
```

Out[2]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	In
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	1
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	1
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	1
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	1
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	1

5 rows × 53 columns

In [3]:
```
df["Unique Key"].unique()
Header=df.columns
df=pd.read_csv(filename, names=Header,low_memory=False, index_col=0)
df.head(2)# will show top 2 rows only
```

Out[3]:

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incid
Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incic
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	10

2 rows × 52 columns

In [4]:
```
df = df.drop(["Intersection Street 1","Cross Street 1","Cross Street 2","Landmark","Ir
#"Taxi Company Borough","Vehicle Type", "Taxi Company Borough", "Taxi Pick Up Locatior
df.head(2)
```

Out[4]:

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incid
Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incic
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	10

2 rows × 33 columns

In [5]:
```
#fill NAN value with 0s
df.fillna(0)
df.head(2)
```

Out[5]:

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incid
Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incic
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	10

2 rows × 33 columns

In [6]:
```
#Checking Null Values
df.isna().sum()
```

Out[6]:
```
Created Date                      0
Closed Date                    2164
Agency                            0
Agency Name                       0
Complaint Type                    0
Descriptor                     5914
Location Type                   131
Incident Zip                   2615
Address Type                   2815
City                           2614
Facility Type                  2171
Status                            0
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
Latitude                       3540
Longitude                      3540
Location                       3540
dtype: int64
```

In [7]:
```
#Dropping all indexes having null value for "City" and "Latitude" Columns
df.dropna(subset=["City","Latitude"], inplace=True)
```

In [8]:
```
df["Descriptor"].fillna("Unknown", inplace=True)# fill "Unknown" inplace of null value
df["Location Type"].fillna("Unknown", inplace=True)# fill "Unknown" inplace of null va
df.dropna(inplace=True)
```

In [9]:
```
df.isna().sum()# Final Check whether null value exists anymore or not.
```

Out[9]:
```
Created Date                      0
Closed Date                       0
Agency                            0
Agency Name                       0
Complaint Type                    0
Descriptor                        0
Location Type                     0
Incident Zip                      0
Address Type                      0
City                              0
Facility Type                     0
Status                            0
Resolution Description            0
Resolution Action Updated Date    0
Community Board                   0
Borough                           0
X Coordinate (State Plane)        0
Y Coordinate (State Plane)        0
Park Facility Name                0
Park Borough                      0
School Name                       0
School Number                     0
School Region                     0
School Code                       0
School Phone Number               0
School Address                    0
School City                       0
School State                      0
School Zip                        0
School Not Found                  0
Latitude                          0
Longitude                         0
Location                          0
dtype: int64
```

In [10]:
```
#No null values exists.
# Lets Check Datatypes for all the columns
df.info(verbose=None, buf=None, max_cols=None, memory_usage=None, show_counts=None, nu
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 296777 entries, Unique Key to 30281825
Data columns (total 33 columns):
 #   Column                          Non-Null Count   Dtype 
---  ------                          --------------   ----- 
 0   Created Date                    296777 non-null  object
 1   Closed Date                     296777 non-null  object
 2   Agency                          296777 non-null  object
 3   Agency Name                     296777 non-null  object
 4   Complaint Type                  296777 non-null  object
 5   Descriptor                      296777 non-null  object
 6   Location Type                   296777 non-null  object
 7   Incident Zip                    296777 non-null  object
 8   Address Type                    296777 non-null  object
 9   City                            296777 non-null  object
 10  Facility Type                   296777 non-null  object
 11  Status                          296777 non-null  object
 12  Resolution Description          296777 non-null  object
 13  Resolution Action Updated Date  296777 non-null  object
 14  Community Board                 296777 non-null  object
 15  Borough                         296777 non-null  object
 16  X Coordinate (State Plane)      296777 non-null  object
 17  Y Coordinate (State Plane)      296777 non-null  object
 18  Park Facility Name              296777 non-null  object
 19  Park Borough                    296777 non-null  object
 20  School Name                     296777 non-null  object
 21  School Number                   296777 non-null  object
 22  School Region                   296777 non-null  object
 23  School Code                     296777 non-null  object
 24  School Phone Number             296777 non-null  object
 25  School Address                  296777 non-null  object
 26  School City                     296777 non-null  object
 27  School State                    296777 non-null  object
 28  School Zip                      296777 non-null  object
 29  School Not Found                296777 non-null  object
 30  Latitude                        296777 non-null  object
 31  Longitude                       296777 non-null  object
 32  Location                        296777 non-null  object
dtypes: object(33)
memory usage: 77.0+ MB
```

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

Out[11]:
```
(296777, 33)
```

In [12]:
```
df.describe()# to Check all statistical values
```

Out[12]:

	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Ad
count	296777	296777	296777	296777	296777	296777	296777	296777	29
unique	256343	235916	2	2	22	43	16	200	
top	07-11-15 23:04	11-08-15 7:34	NYPD	New York City Police Department	Blocked Driveway	Loud Music/Party	Street/Sidewalk	11385	ADD
freq	9	24	296776	296776	76707	60443	246257	5162	23

4 rows × 33 columns

In [13]:
```
#Printing last row
df.tail(1)
```

Out[13]:

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	I
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	

1 rows × 33 columns

In [14]:
```
#we are checking all the null value
missing_values=df.isnull()
#True shows that the value at the place was null
missing_values.head()
# Optional
```

Out[14]:

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Address Type	C
Unique Key										
Unique Key	False	False	False	False	False	False	False	False	False	Fa
32310363	False	False	False	False	False	False	False	False	False	Fa
32309934	False	False	False	False	False	False	False	False	False	Fa
32309159	False	False	False	False	False	False	False	False	False	Fa
32305098	False	False	False	False	False	False	False	False	False	Fa

5 rows × 33 columns

In [15]:
```
for col in df.columns.values.tolist():
    print(col)
    print(df[col].value_counts ())#it tells for each column value counts
    print("")
```

```
Created Date
07-11-15 23:04            9
11-06-15 23:34            9
06-06-15 22:23            9
08-08-15 22:05            8
05-05-15 21:20            8
                         ..
09/22/2015 09:20:13 PM    1
09/22/2015 09:16:24 PM    1
09/22/2015 09:15:57 PM    1
09/22/2015 09:15:48 PM    1
03/29/2015 12:33:01 AM    1
Name: Created Date, Length: 256343, dtype: int64

Closed Date
11-08-15 7:34             24
10-11-15 7:03             22
05-10-15 7:01             18
12-08-15 7:44             18
12-07-15 23:17            17
                          ..
09/21/2015 10:28:35 AM     1
09/23/2015 05:45:59 AM     1
09/21/2015 10:29:37 AM     1
09/22/2015 03:52:44 AM     1
03/29/2015 04:41:50 AM     1
Name: Closed Date, Length: 235916, dtype: int64

Agency
NYPD      296776
Agency         1
Name: Agency, dtype: int64

Agency Name
New York City Police Department    296776
Agency Name                             1
Name: Agency Name, dtype: int64

Complaint Type
Blocked Driveway             76707
Illegal Parking              74045
Noise - Street/Sidewalk      47746
Noise - Commercial           35147
Derelict Vehicle             17502
Noise - Vehicle              16869
Animal Abuse                  7746
Homeless Encampment           4345
Traffic                       4258
Noise - Park                  3927
Vending                       3773
Drinking                      1271
Noise - House of Worship       922
Posting Advertisement          647
Urinating in Public            592
Bike/Roller/Skate Chronic      414
Panhandling                    300
Disorderly Youth               285
Illegal Fireworks              163
Graffiti                       113
Squeegee                         4
```

```
Complaint Type                   1
Name: Complaint Type, dtype: int64

Descriptor
Loud Music/Party                60443
No Access                       56748
Posted Parking Sign Violation   22113
Loud Talking                    21255
Partial Access                  19959
With License Plate              17502
Blocked Hydrant                 15842
Commercial Overnight Parking    11911
Car/Truck Music                 11114
Blocked Sidewalk                10931
Unknown                          5818
Double Parked Blocking Traffic   5561
Double Parked Blocking Vehicle   4148
Engine Idling                    4135
Banging/Pounding                 4093
Neglected                        3773
Car/Truck Horn                   3478
Congestion/Gridlock              2556
In Prohibited Area               2017
Other (complaint details)        1961
Unlicensed                       1756
Overnight Commercial Storage     1747
Unauthorized Bus Layover         1333
Truck Route Violation             982
In Public                         924
Tortured                          849
Vehicle                           587
Chained                           534
Detached Trailer                  459
No Shelter                        381
Chronic Stoplight Violation       280
Underage - Licensed Est           270
Chronic Speeding                  266
In Car                            248
Playing in Unsuitable Place       245
Drag Racing                       174
Loud Television                    93
Police Report Requested            90
After Hours - Licensed Est         77
Building                           60
Nuisance/Truant                    40
Police Report Not Requested        23
Descriptor                          1
Name: Descriptor, dtype: int64

Location Type
Street/Sidewalk              246257
Store/Commercial              20116
Club/Bar/Restaurant           17193
Residential Building/House     6942
Park/Playground                4645
House of Worship                920
Residential Building            226
Parking Lot                     116
House and Store                  93
Unknown                          85
```

```
Vacant Lot                      77
Commercial                      62
Subway Station                  34
Highway                          5
Roadway Tunnel                   5
Location Type                    1
Name: Location Type, dtype: int64

Incident Zip
11385           5162
11368           4292
11211           4219
11234           4141
11238           3769
                ...
Incident Zip       1
11242              1
10153              1
11451              1
10123              1
Name: Incident Zip, Length: 200, dtype: int64

Address Type
ADDRESS         238570
INTERSECTION     43349
BLOCKFACE        11047
LATLONG           3461
PLACENAME          349
Address Type         1
Name: Address Type, dtype: int64

City
BROOKLYN               98047
NEW YORK               65293
BRONX                  40548
STATEN ISLAND          12302
JAMAICA                 7264
ASTORIA                 6313
FLUSHING                5961
RIDGEWOOD               5159
CORONA                  4292
WOODSIDE                3538
SOUTH RICHMOND HILL     2773
OZONE PARK              2755
EAST ELMHURST           2730
ELMHURST                2671
WOODHAVEN               2462
MASPETH                 2457
LONG ISLAND CITY        2427
SOUTH OZONE PARK        2167
RICHMOND HILL           1902
FRESH MEADOWS           1891
QUEENS VILLAGE          1812
MIDDLE VILLAGE          1765
JACKSON HEIGHTS         1688
FOREST HILLS            1683
REGO PARK               1483
COLLEGE POINT           1220
BAYSIDE                 1217
FAR ROCKAWAY            1178
```

```
WHITESTONE              1095
HOLLIS                  1012
HOWARD BEACH             928
ROSEDALE                 915
SPRINGFIELD GARDENS      880
SAINT ALBANS             834
KEW GARDENS              771
ROCKAWAY PARK            743
SUNNYSIDE                722
Astoria                  716
LITTLE NECK              558
OAKLAND GARDENS          549
CAMBRIA HEIGHTS          477
BELLEROSE                375
GLEN OAKS                306
ARVERNE                  220
FLORAL PARK              152
Long Island City         134
Woodside                 120
NEW HYDE PARK             98
CENTRAL PARK              97
QUEENS                    31
BREEZY POINT              30
East Elmhurst             14
City                       1
Howard Beach               1
Name: City, dtype: int64

Facility Type
Precinct         296776
Facility Type         1
Name: Facility Type, dtype: int64

Status
Closed      296750
Assigned        26
Status           1
Name: Status, dtype: int64

Resolution Description
The Police Department responded to the complaint and with the information available o
bserved no evidence of the violation at that time.
89875
The Police Department responded to the complaint and took action to fix the conditio
n.
61197
The Police Department responded and upon arrival those responsible for the condition
were gone.
57781
The Police Department responded to the complaint and determined that police action wa
s not necessary.
37953
The Police Department issued a summons in response to the complaint.
28203
The Police Department reviewed your complaint and provided additional information bel
ow.
13736
Your request can not be processed at this time because of insufficient contact inform
ation. Please create a new Service Request on NYC.gov and provide more detailed conta
ct information.     4257
```

```
This complaint does not fall under the Police Department's jurisdiction.
1769
The Police Department responded to the complaint but officers were unable to gain ent
ry into the premises.
1208
The Police Department responded to the complaint and a report was prepared.
673
The Police Department made an arrest in response to the complaint.
124
Resolution Description
1
Name: Resolution Description, dtype: int64

Resolution Action Updated Date
11-08-15 7:34             24
10-11-15 7:03             22
05-10-15 7:01             18
12-08-15 7:44             18
12-07-15 23:17            17
                          ..
09/21/2015 09:25:25 PM     1
09/21/2015 10:26:54 PM     1
09/22/2015 02:04:21 AM     1
09/22/2015 12:07:14 AM     1
03/29/2015 04:41:50 AM     1
Name: Resolution Action Updated Date, Length: 236677, dtype: int64

Community Board
12 MANHATTAN       12337
01 BROOKLYN        10891
05 QUEENS           9415
01 QUEENS           9170
09 QUEENS           8006
                   ...
84 QUEENS             11
26 BRONX              10
80 QUEENS              7
56 BROOKLYN            4
Community Board        1
Name: Community Board, Length: 72, dtype: int64

Borough
BROOKLYN         98046
QUEENS           80490
MANHATTAN        65391
BRONX            40547
STATEN ISLAND    12302
Borough              1
Name: Borough, dtype: int64

X Coordinate (State Plane)
1021327    910
1000311    563
1037000    507
982967     344
1042290    342
          ...
984815       1
1000891      1
1024005      1
```

```
1026090      1
1016436      1
Name: X Coordinate (State Plane), Length: 63163, dtype: int64

Y Coordinate (State Plane)
241829    901
202363    506
195702    500
175044    364
197554    345
         ...
164474      1
209955      1
239002      1
159386      1
222234      1
Name: Y Coordinate (State Plane), Length: 73626, dtype: int64

Park Facility Name
Unspecified           296776
Park Facility Name         1
Name: Park Facility Name, dtype: int64

Park Borough
BROOKLYN         98046
QUEENS           80490
MANHATTAN        65391
BRONX            40547
STATEN ISLAND    12302
Park Borough         1
Name: Park Borough, dtype: int64

School Name
Unspecified    296776
School Name         1
Name: School Name, dtype: int64

School Number
Unspecified      296776
School Number         1
Name: School Number, dtype: int64

School Region
Unspecified      296776
School Region         1
Name: School Region, dtype: int64

School Code
Unspecified    296776
School Code         1
Name: School Code, dtype: int64

School Phone Number
Unspecified            296776
School Phone Number         1
Name: School Phone Number, dtype: int64

School Address
Unspecified       296776
School Address         1
```

```
Name: School Address, dtype: int64

School City
Unspecified    296776
School City         1
Name: School City, dtype: int64

School State
Unspecified     296776
School State         1
Name: School State, dtype: int64

School Zip
Unspecified    296776
School Zip          1
Name: School Zip, dtype: int64

School Not Found
N                   296776
School Not Found         1
Name: School Not Found, dtype: int64

Latitude
40.83036236    901
40.72195913    505
40.70381897    480
40.6471319     362
40.70872649    341
               ...
40.82991286      1
40.62301064      1
40.73944459      1
40.69419227      1
40.71605291      1
Name: Latitude, Length: 124925, dtype: int64

Longitude
-73.86602154    901
-73.80969682    505
-73.94207345    480
-73.79065392    341
-74.00462341    340
                ...
-73.89950141      1
-73.9490832       1
-73.85696098      1
-73.89308544      1
-73.9913785       1
Name: Longitude, Length: 125019, dtype: int64

Location
(40.83036235589997, -73.86602154214397)     901
(40.72195913199264, -73.80969682426189)     505
(40.703818970933284, -73.94207345177706)    476
(40.708726489323325, -73.7906539235748)     341
(40.64713190020787, -74.00462341153786)     340
                                           ...
(40.83482408901492, -73.91738581364538)       1
(40.76069205946853, -73.82100378683009)       1
(40.84747086926786, -73.8310353247154)        1
```

```
(40.8731850797772, -73.88631056524476)        1
(40.71605290789855, -73.99137850370803)       1
Name: Location, Length: 125850, dtype: int64
```

In [16]:
```
check_missing_values=df.isnull()# boolean counter for checking "True" and "False" in p
for col1 in check_missing_values.columns.values.tolist():
    print(col1)
    print(check_missing_values[col1].value_counts ())#it tells for each column value c
```

```
Created Date
False    296777
Name: Created Date, dtype: int64
Closed Date
False    296777
Name: Closed Date, dtype: int64
Agency
False    296777
Name: Agency, dtype: int64
Agency Name
False    296777
Name: Agency Name, dtype: int64
Complaint Type
False    296777
Name: Complaint Type, dtype: int64
Descriptor
False    296777
Name: Descriptor, dtype: int64
Location Type
False    296777
Name: Location Type, dtype: int64
Incident Zip
False    296777
Name: Incident Zip, dtype: int64
Address Type
False    296777
Name: Address Type, dtype: int64
City
False    296777
Name: City, dtype: int64
Facility Type
False    296777
Name: Facility Type, dtype: int64
Status
False    296777
Name: Status, dtype: int64
Resolution Description
False    296777
Name: Resolution Description, dtype: int64
Resolution Action Updated Date
False    296777
Name: Resolution Action Updated Date, dtype: int64
Community Board
False    296777
Name: Community Board, dtype: int64
Borough
False    296777
Name: Borough, dtype: int64
X Coordinate (State Plane)
False    296777
Name: X Coordinate (State Plane), dtype: int64
Y Coordinate (State Plane)
False    296777
Name: Y Coordinate (State Plane), dtype: int64
Park Facility Name
False    296777
Name: Park Facility Name, dtype: int64
Park Borough
False    296777
Name: Park Borough, dtype: int64
```

```
School Name
False    296777
Name: School Name, dtype: int64
School Number
False    296777
Name: School Number, dtype: int64
School Region
False    296777
Name: School Region, dtype: int64
School Code
False    296777
Name: School Code, dtype: int64
School Phone Number
False    296777
Name: School Phone Number, dtype: int64
School Address
False    296777
Name: School Address, dtype: int64
School City
False    296777
Name: School City, dtype: int64
School State
False    296777
Name: School State, dtype: int64
School Zip
False    296777
Name: School Zip, dtype: int64
School Not Found
False    296777
Name: School Not Found, dtype: int64
Latitude
False    296777
Name: Latitude, dtype: int64
Longitude
False    296777
Name: Longitude, dtype: int64
Location
False    296777
Name: Location, dtype: int64
```

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 [17]:
```
#column converted to datetime datatype

import datetime

df["Created Date"]= pd.to_datetime(df["Created Date"][1:])
df['Closed Date']= pd.to_datetime(df['Closed Date'][1:])

print(type(df))
print(df["Created Date"].dtype)# Checking the Conversion of Datatype from string to Da
```

```
<class 'pandas.core.frame.DataFrame'>
datetime64[ns]
```

In [18]:
```
df["Request_Closing_Time"] =((df["Closed Date"][:])-(df["Created Date"][:]))
df.head(2)
```

Out[18]:

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incide Z
Unique Key	NaT	NaT	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incide Z
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	1003

2 rows × 34 columns

In [19]:
```
df['Request_Closing_Time'] = df['Closed Date'].values - df['Created Date'].values
df['Request_Closing_Time_Mins'] = df['Request_Closing_Time']/np.timedelta64(1,'m')
```

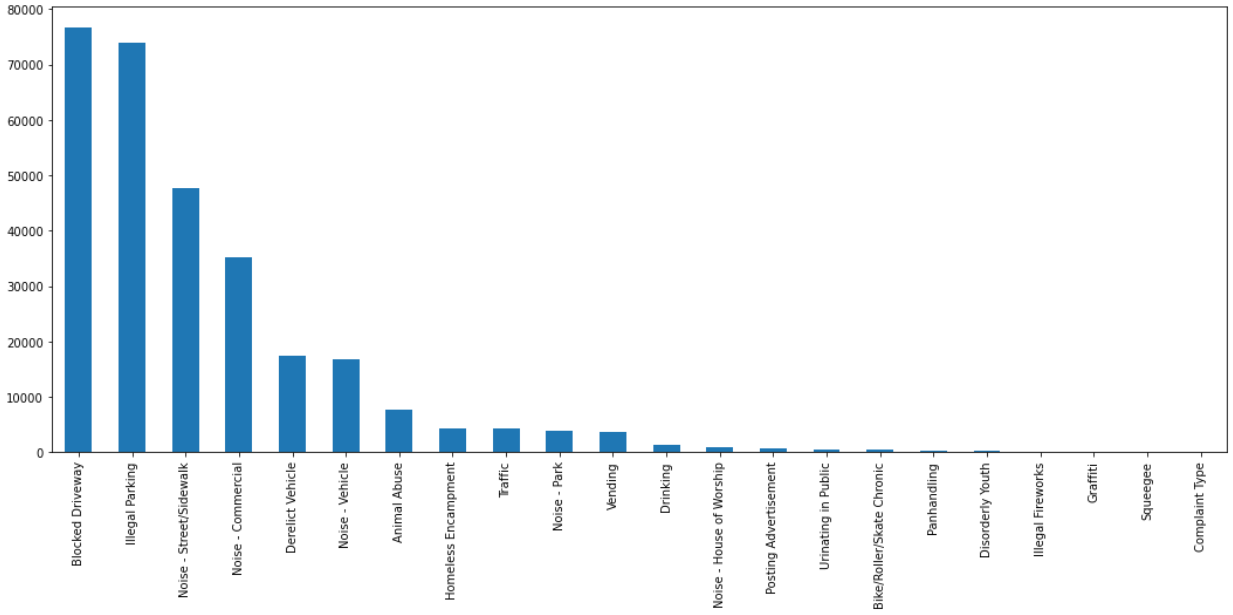
Task 3

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.

Insight 1st

would be that maximum complaint type among all the the complaints are of "Blocked Driveway" followed by "Illeagal Parking".

In [20]:
```
#complaint type breakdown with bars plot to figure out the top 10 complaints
df['Complaint Type'].value_counts().plot(kind='bar',alpha=1, figsize=(18,7))
plt.show()
```



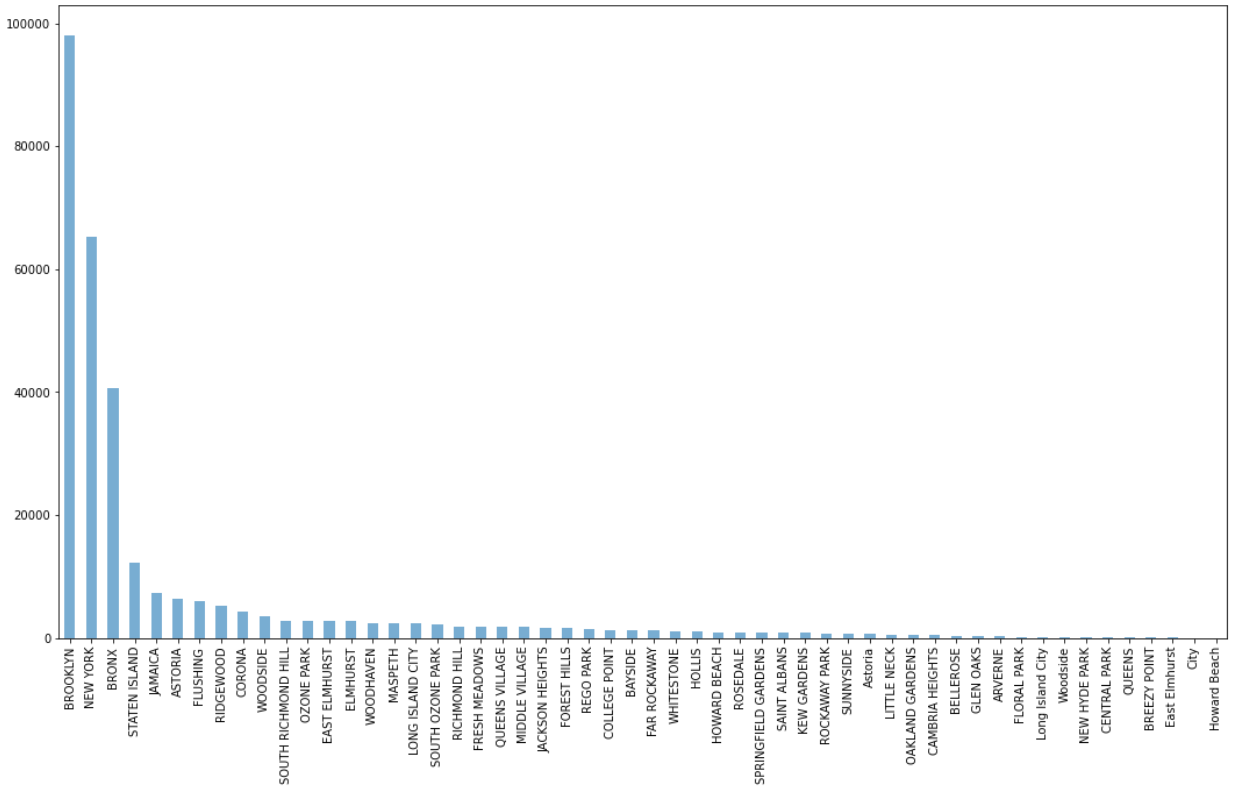
Insight 2nd

Maximum Complaints Registered are from Brooklyn City

In [21]:
```
#Now we count the City with the highest complaint
df['City'].value_counts()
```

Out[21]:
```
BROOKLYN               98047
NEW YORK               65293
BRONX                  40548
STATEN ISLAND          12302
JAMAICA                 7264
ASTORIA                 6313
FLUSHING                5961
RIDGEWOOD               5159
CORONA                  4292
WOODSIDE                3538
SOUTH RICHMOND HILL     2773
OZONE PARK              2755
EAST ELMHURST           2730
ELMHURST                2671
WOODHAVEN               2462
MASPETH                 2457
LONG ISLAND CITY        2427
SOUTH OZONE PARK        2167
RICHMOND HILL           1902
FRESH MEADOWS           1891
QUEENS VILLAGE          1812
MIDDLE VILLAGE          1765
JACKSON HEIGHTS         1688
FOREST HILLS            1683
REGO PARK               1483
COLLEGE POINT           1220
BAYSIDE                 1217
FAR ROCKAWAY            1178
WHITESTONE              1095
HOLLIS                  1012
HOWARD BEACH             928
ROSEDALE                 915
SPRINGFIELD GARDENS      880
SAINT ALBANS             834
KEW GARDENS              771
ROCKAWAY PARK            743
SUNNYSIDE                722
Astoria                  716
LITTLE NECK              558
OAKLAND GARDENS          549
CAMBRIA HEIGHTS          477
BELLEROSE                375
GLEN OAKS                306
ARVERNE                  220
FLORAL PARK              152
Long Island City         134
Woodside                 120
NEW HYDE PARK             98
CENTRAL PARK              97
QUEENS                    31
BREEZY POINT              30
East Elmhurst             14
City                       1
Howard Beach               1
Name: City, dtype: int64
```

In [22]:
```
#complaint by city
df['City'].value_counts().plot(kind='bar', alpha=0.6, figsize=(18, 10))
plt.show()
```



In [23]:
```
#display complaint type and city together
df.groupby('Complaint Type')['City'].head(10)
```

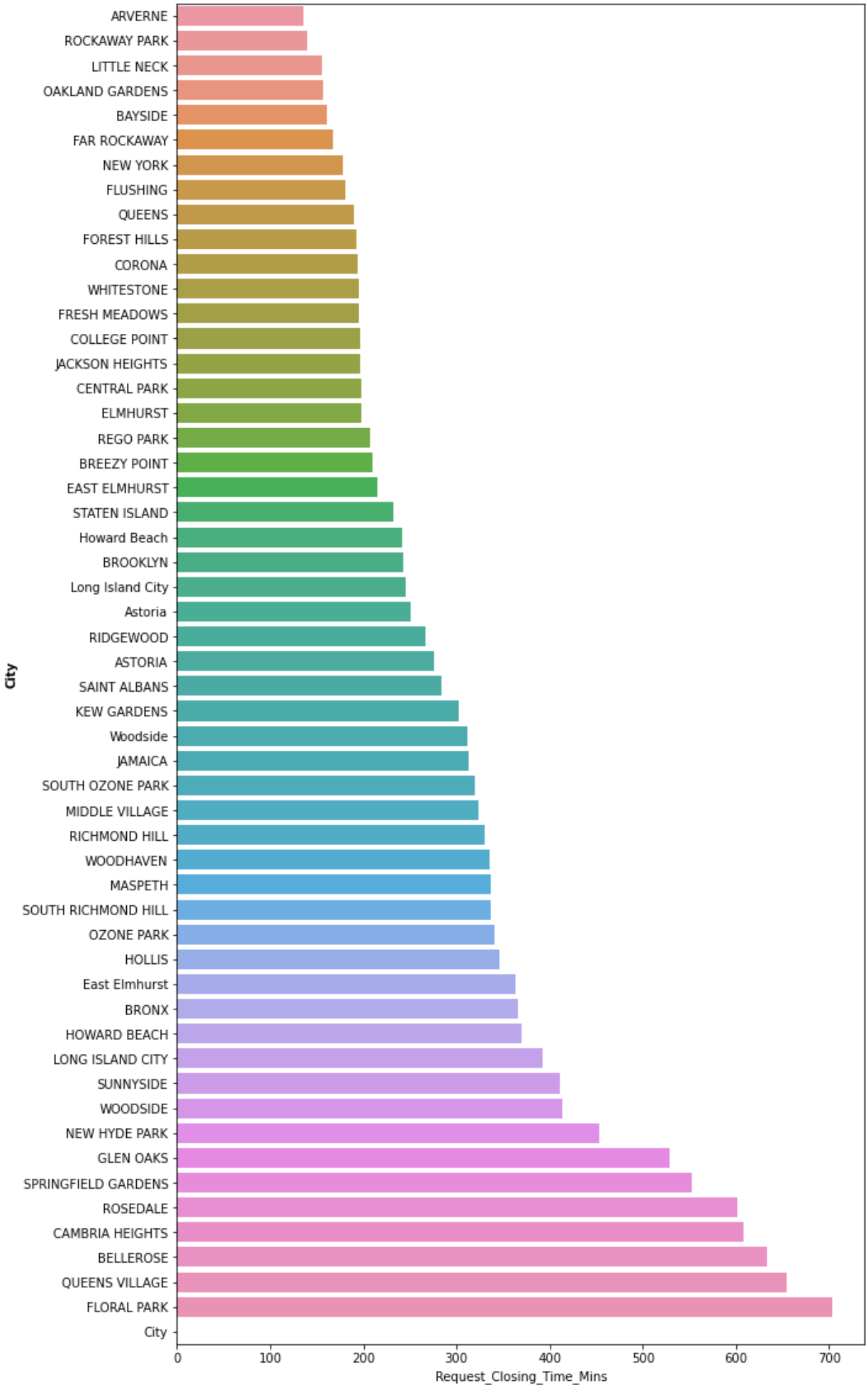
Out[23]:
```
Unique Key
Unique Key        City
32310363      NEW YORK
32309934       ASTORIA
32309159         BRONX
32305098         BRONX
                 ...
31701324      WOODSIDE
31044728      NEW YORK
30871215      NEW YORK
30427320      NEW YORK
30314122      NEW YORK
Name: City, Length: 205, dtype: object
```

Insight 3rd

Cities with average Response Time.

In [45]:
```
# visualizing Cities with average response time
viz1 = df[['City','Request_Closing_Time_Mins']]
c1 = viz1.groupby('City')['Request_Closing_Time_Mins'].mean().to_frame()
c1 = c1.sort_values('Request_Closing_Time_Mins')
c1['City'] = c1.index
txt={'weight':'bold'}
plt.figure(figsize=(10,20))
sns.barplot(y='City',x='Request_Closing_Time_Mins',data=c1)
plt.ylabel("City",fontdict=txt)
```

Out[45]: Text(0, 0.5, 'City')

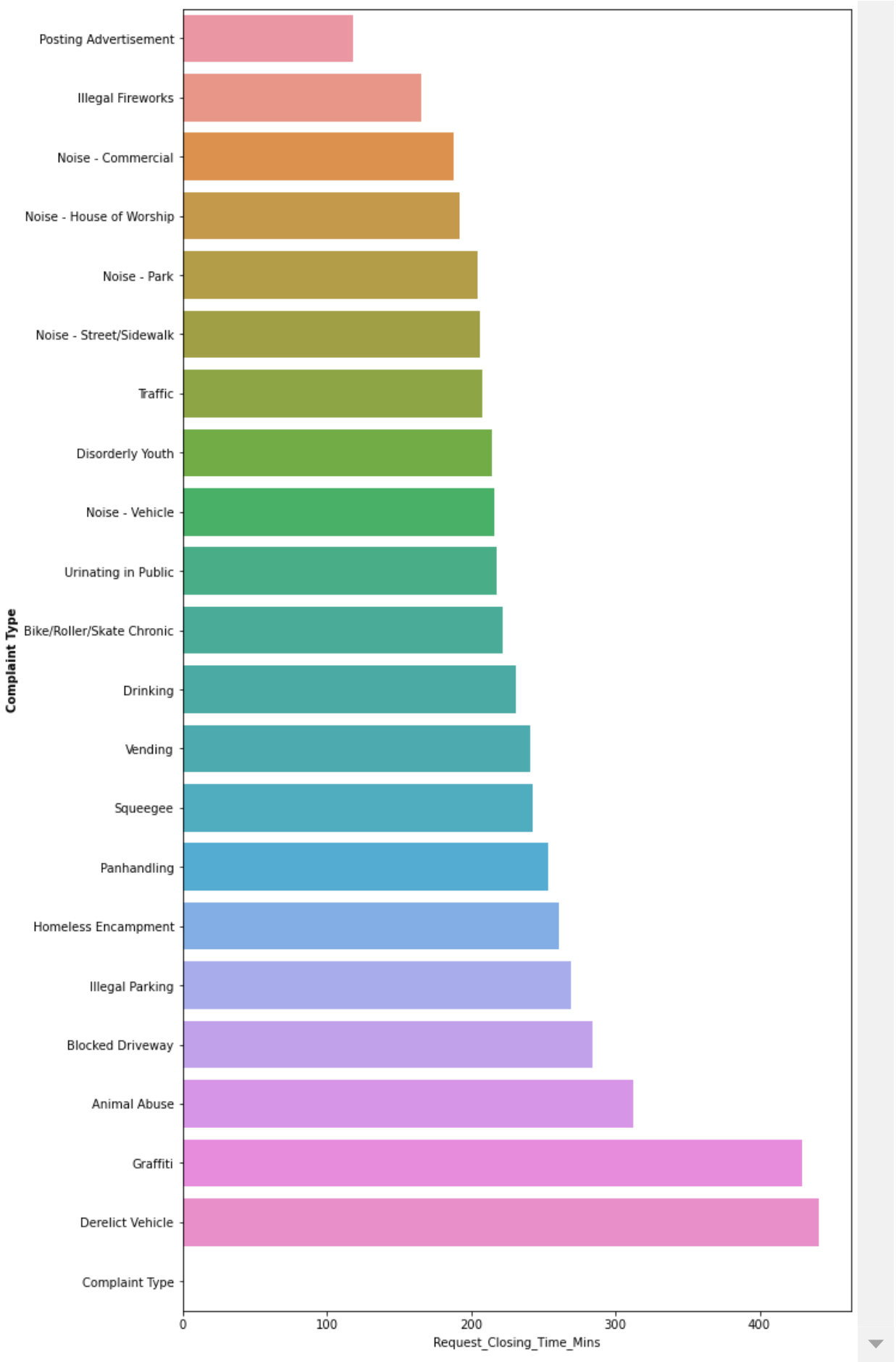

ARVERNE
ROCKAWAY PARK
LITTLE NECK
OAKLAND GARDENS
BAYSIDE
FAR ROCKAWAY
NEW YORK
FLUSHING
QUEENS
FOREST HILLS
CORONA
WHITESTONE
FRESH MEADOWS
COLLEGE POINT
JACKSON HEIGHTS
CENTRAL PARK
ELMHURST
REGO PARK
BREEZY POINT
EAST ELMHURST
STATEN ISLAND
Howard Beach
BROOKLYN
Long Island City
Astoria
RIDGEWOOD
ASTORIA
SAINT ALBANS
KEW GARDENS
Woodside
JAMAICA
SOUTH OZONE PARK
MIDDLE VILLAGE
RICHMOND HILL
WOODHAVEN
MASPETH
SOUTH RICHMOND HILL
OZONE PARK
HOLLIS
East Elmhurst
BRONX
HOWARD BEACH
LONG ISLAND CITY
SUNNYSIDE
WOODSIDE
NEW HYDE PARK
GLEN OAKS
SPRINGFIELD GARDENS
ROSEDALE
CAMBRIA HEIGHTS
BELLEROSE
QUEENS VILLAGE
FLORAL PARK
City
City
0
100
200
300
400
500
600
700
Request_Closing_Time_Mins

Insight 4th

"Complaint Types" with average Response Time.

In [49]:
```
# visualizing Cities with average response time
viz1 = df[['Complaint Type','Request_Closing_Time_Mins']]
c1 = viz1.groupby('Complaint Type')['Request_Closing_Time_Mins'].mean().to_frame()
c1 = c1.sort_values('Request_Closing_Time_Mins')
c1['Complaint Type'] = c1.index
txt={'weight':'bold'}
plt.figure(figsize=(10,20))
sns.barplot(y='Complaint Type',x='Request_Closing_Time_Mins',data=c1)
plt.ylabel("Complaint Type",fontdict=txt)
```

Out[49]:
```
Text(0, 0.5, 'Complaint Type')
```


Posting Advertisement
Illegal Fireworks
Noise - Commercial
Noise - House of Worship
Noise - Park
Noise - Street/Sidewalk
Traffic
Disorderly Youth
Noise - Vehicle
Urinating in Public
Bike/Roller/Skate Chronic
Drinking
Vending
Squeegee
Panhandling
Homeless Encampment
Illegal Parking
Blocked Driveway
Animal Abuse
Graffiti
Derelict Vehicle
Complaint Type
Complaint Type
0
100
200
300
400
Request_Closing_Time_Mins

Task 4

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

In [24]:
```
# Grouping complaints by cities and finiding mean response time for each complaint typ
# Sorting the mean response time of different complaint types for each city
city_complaintype_group = df.groupby(['City','Complaint Type'])['Request_Closing_Time_
city_complaintype_group = city_complaintype_group.T
col = city_complaintype_group.columns
for i in col:
 exec("{} = city_complaintype_group['{}'].sort_values()".format(i,i))
```

```
Traceback (most recent call last):

  File C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:33
69 in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)

  Input In [24] in <cell line: 6>
    exec("{} = city_complaintype_group['{}'].sort_values()".format(i,i))

  File <string>:1
    BREEZY POINT = city_complaintype_group['BREEZY POINT'].sort_values()
           ^
SyntaxError: invalid syntax
```

In [ ]:
```
```

Task 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?

Create Hypothesis for statement 1: Whether the average response time across complaint types is similar or not (overall)

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 types
- Alternate Hypothesis : H1 : There is a significant difference in average response time across different complaint types

In [ ]:
```
complaints = df['Complaint Type'].value_counts().index
for i in range(len(complaints)):
 exec("sample{} = df.loc[(df['Complaint Type'] == '{}') , 'Request Closing Time Mins']
```

In [ ]:
```
from scipy import stats
fscore,pvalue = stats.f_oneway(sample1,sample2,sample3,sample4,sample5,sample6,sample7
sample10,sample11,sample12,sample13,sample14,sample15,sample16,sample17,sample18,sampl
print("score : {:.2f} , pvalue : {:.2f}".format(fscore,pvalue))
```

- Here , pvalue (0.00) < alpha value(0.05)
- We reject our Null Hypothesis
- There is a significant difference in average response time across different complaint types
- (i.e) the average response time across different complaint types is not similar (overall)

Create Hypothesis for Statement 2: Are the type of complaint or service requested and location related?

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

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

- Here , pvalue (0.00) < alpha value(0.05)
- We reject our Null Hypothesis
- There is some significant relation between type of complaint and location (i.e) The type of complaint or service requested and the location are related