service-request-analysis

September 8, 2023

[12]:
```
#Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

[13]:
```
from scipy import stats
from scipy.stats import chi2_contingency
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

[14]:
```
#Importing the dataset
df=pd.read_csv("311_Service_Requests_from_2010_to_Present.csv")
```

```
C:\Users\lrnem\AppData\Local\Temp\ipykernel_8728\2134542247.py:2: DtypeWarning:
Columns (48,49) have mixed types. Specify dtype option on import or set
low_memory=False.
  df=pd.read_csv("311_Service_Requests_from_2010_to_Present.csv")
```

[15]:
```
df.head()
```

[15]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	New York City Police Department	Blocked Driveway			
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	New York City Police Department	Blocked Driveway			
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	New York City Police Department	Illegal Parking			
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	NYPD	New York City Police Department	Illegal Parking			

```
1                    No Access  Street/Sidewalk       11105.0
2                    No Access  Street/Sidewalk       10458.0
3  Commercial Overnight Parking  Street/Sidewalk       10461.0
4              Blocked Sidewalk  Street/Sidewalk       11373.0

        Incident Address  … Bridge Highway Name Bridge Highway Direction  \
0    71 VERMILYEA AVENUE  …                  NaN                       NaN
1        27-07 23 AVENUE  …                  NaN                       NaN
2  2897 VALENTINE AVENUE  …                  NaN                       NaN
3    2940 BAISLEY AVENUE  …                  NaN                       NaN
4          87-14 57 ROAD  …                  NaN                       NaN

  Road Ramp Bridge Highway Segment Garage Lot Name Ferry Direction  \
0       NaN                     NaN             NaN             NaN
1       NaN                     NaN             NaN             NaN
2       NaN                     NaN             NaN             NaN
3       NaN                     NaN             NaN             NaN
4       NaN                     NaN             NaN             NaN

  Ferry Terminal Name   Latitude  Longitude  \
0                 NaN  40.865682 -73.923501
1                 NaN  40.775945 -73.915094
2                 NaN  40.870325 -73.888525
3                 NaN  40.835994 -73.828379
4                 NaN  40.733060 -73.874170

                                Location
0   (40.86568153633767, -73.92350095571744)
1  (40.775945312321085, -73.91509393898605)
2  (40.870324522111424, -73.88852464418646)
3   (40.83599404683083, -73.82837939584206)
4  (40.733059618956815, -73.87416975810375)

[5 rows x 53 columns]
```

[16]:
```
#Understanding the data
df.describe()
```

[16]:
```
          Unique Key   Incident Zip  X Coordinate (State Plane)  \
count  3.006980e+05  298083.000000                2.971580e+05
mean   3.130054e+07   10848.888645                1.004854e+06
std    5.738547e+05     583.182081                2.175338e+04
min    3.027948e+07      83.000000                9.133570e+05
25%    3.080118e+07   10310.000000                9.919752e+05
50%    3.130436e+07   11208.000000                1.003158e+06
75%    3.178446e+07   11238.000000                1.018372e+06
max    3.231065e+07   11697.000000                1.067173e+06
```

```
       Y Coordinate (State Plane)  School or Citywide Complaint  Vehicle Type  \
count                297158.000000                           0.0           0.0
mean                 203754.534416                           NaN           NaN
std                   29880.183529                           NaN           NaN
min                  121219.000000                           NaN           NaN
25%                  183343.000000                           NaN           NaN
50%                  201110.500000                           NaN           NaN
75%                  224125.250000                           NaN           NaN
max                  271876.000000                           NaN           NaN

       Taxi Company Borough  Taxi Pick Up Location  Garage Lot Name  \
count                   0.0                    0.0              0.0
mean                    NaN                    NaN              NaN
std                     NaN                    NaN              NaN
min                     NaN                    NaN              NaN
25%                     NaN                    NaN              NaN
50%                     NaN                    NaN              NaN
75%                     NaN                    NaN              NaN
max                     NaN                    NaN              NaN

            Latitude      Longitude
count  297158.000000  297158.000000
mean       40.725885     -73.925630
std         0.082012       0.078454
min        40.499135     -74.254937
25%        40.669796     -73.972142
50%        40.718661     -73.931781
75%        40.781840     -73.876805
max        40.912869     -73.700760
```

[17]:
```
df.shape
```

[17]:
```
(300698, 53)
```

[18]:
```
#Conversion to datetime format
df["Created Date"]=pd.to_datetime(df["Created Date"])
df["Closed Date"]=pd.to_datetime(df["Closed Date"])
```

[19]:
```
#Creating a new column 'Request_Closing_Time' as the time elapsed between␣
 ↪request creation and request closing
df["Request_Closing_Time"]=(df["Closed Date"]-df["Created Date"])
Request_Closing_Time=[]
for x in (df["Closed Date"]-df["Created Date"]):
    close=x.total_seconds()/60
    Request_Closing_Time.append(close)
df["Request_Closing_Time"]=Request_Closing_Time
```

[20]:
```
df.head()
```

[20]:

```
   Unique Key        Created Date         Closed Date Agency  \
0    32310363 2015-12-31 23:59:45 2016-01-01 00:55:00   NYPD
1    32309934 2015-12-31 23:59:44 2016-01-01 01:26:00   NYPD
2    32309159 2015-12-31 23:59:29 2016-01-01 04:51:00   NYPD
3    32305098 2015-12-31 23:57:46 2016-01-01 07:43:00   NYPD
4    32306529 2015-12-31 23:56:58 2016-01-01 03:24:00   NYPD

                      Agency Name           Complaint Type  \
0  New York City Police Department  Noise - Street/Sidewalk
1  New York City Police Department         Blocked Driveway
2  New York City Police Department         Blocked Driveway
3  New York City Police Department          Illegal Parking
4  New York City Police Department          Illegal Parking

                    Descriptor    Location Type  Incident Zip  \
0              Loud Music/Party  Street/Sidewalk       10034.0
1                     No Access  Street/Sidewalk       11105.0
2                     No Access  Street/Sidewalk       10458.0
3  Commercial Overnight Parking  Street/Sidewalk       10461.0
4              Blocked Sidewalk  Street/Sidewalk       11373.0

        Incident Address  … Bridge Highway Direction Road Ramp  \
0    71 VERMILYEA AVENUE  …                       NaN       NaN
1        27-07 23 AVENUE  …                       NaN       NaN
2  2897 VALENTINE AVENUE  …                       NaN       NaN
3    2940 BAISLEY AVENUE  …                       NaN       NaN
4          87-14 57 ROAD  …                       NaN       NaN

  Bridge Highway Segment Garage Lot Name Ferry Direction Ferry Terminal Name  \
0                    NaN             NaN              NaN                 NaN
1                    NaN             NaN              NaN                 NaN
2                    NaN             NaN              NaN                 NaN
3                    NaN             NaN              NaN                 NaN
4                    NaN             NaN              NaN                 NaN

    Latitude  Longitude                                    Location  \
0  40.865682 -73.923501   (40.86568153633767, -73.92350095571744)
1  40.775945 -73.915094  (40.775945312321085, -73.91509393898605)
2  40.870325 -73.888525  (40.870324522111424, -73.88852464418646)
3  40.835994 -73.828379   (40.83599404683083, -73.82837939584206)
4  40.733060 -73.874170  (40.733059618956815, -73.87416975810375)

  Request_Closing_Time
0            55.250000
1            86.266667
```

```
2           291.516667
3           465.233333
4           207.033333

[5 rows x 54 columns]
```

[21]:
```
#EDA
df["Agency"].unique()
```

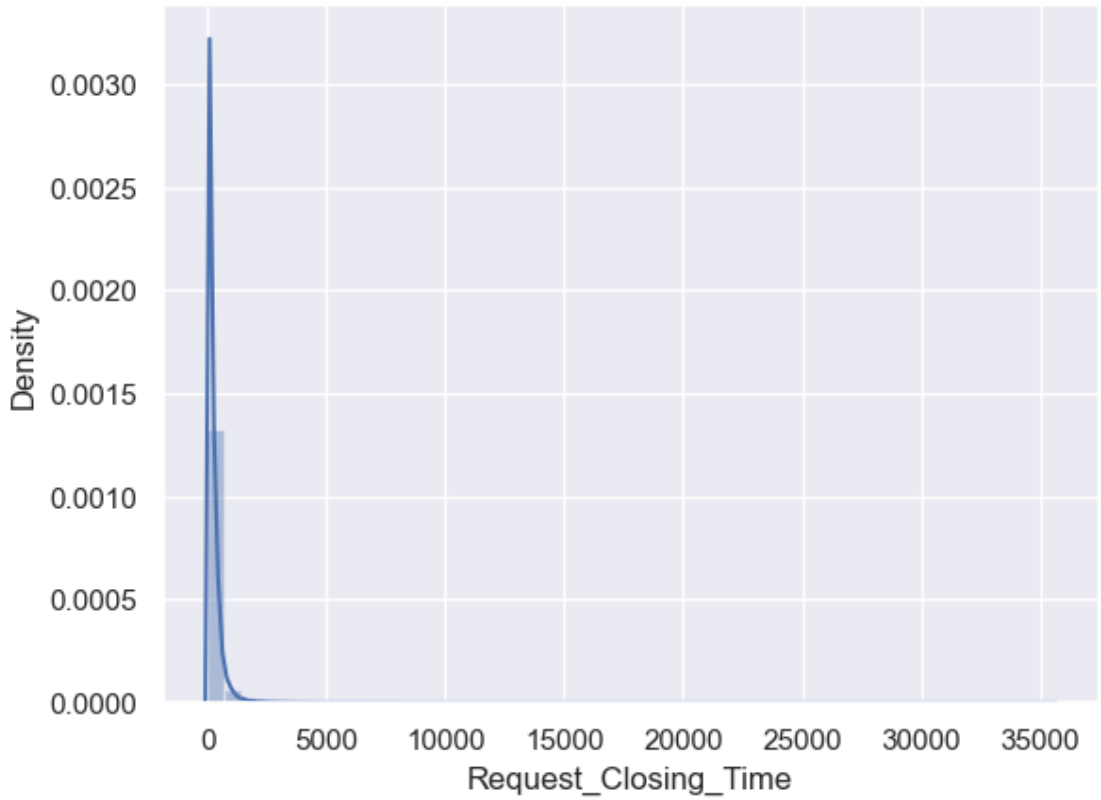
[21]:
```
array(['NYPD'], dtype=object)
```

We can see the above data belongs to the NYPD.

[22]:
```
sns.distplot(df["Request_Closing_Time"])
plt.show
```

```
C:\Users\lrnem\AppData\Local\Temp\ipykernel_8728\5426915.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df["Request_Closing_Time"])
```

[22]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



[23]:
```
print("Total Number of Complaints : ",len(df),"\n")
print("Percentage of complaints that took 99 hours or less :␣
  ↪",round((len(df)-(df["Request_Closing_Time"]>=99).sum())/len(df)*100,2),"%")
print("Percentage of complaints that took 999 hours or less :␣
  ↪",round((len(df)-(df["Request_Closing_Time"]>=999).sum())/len(df)*100,2),"%")
```

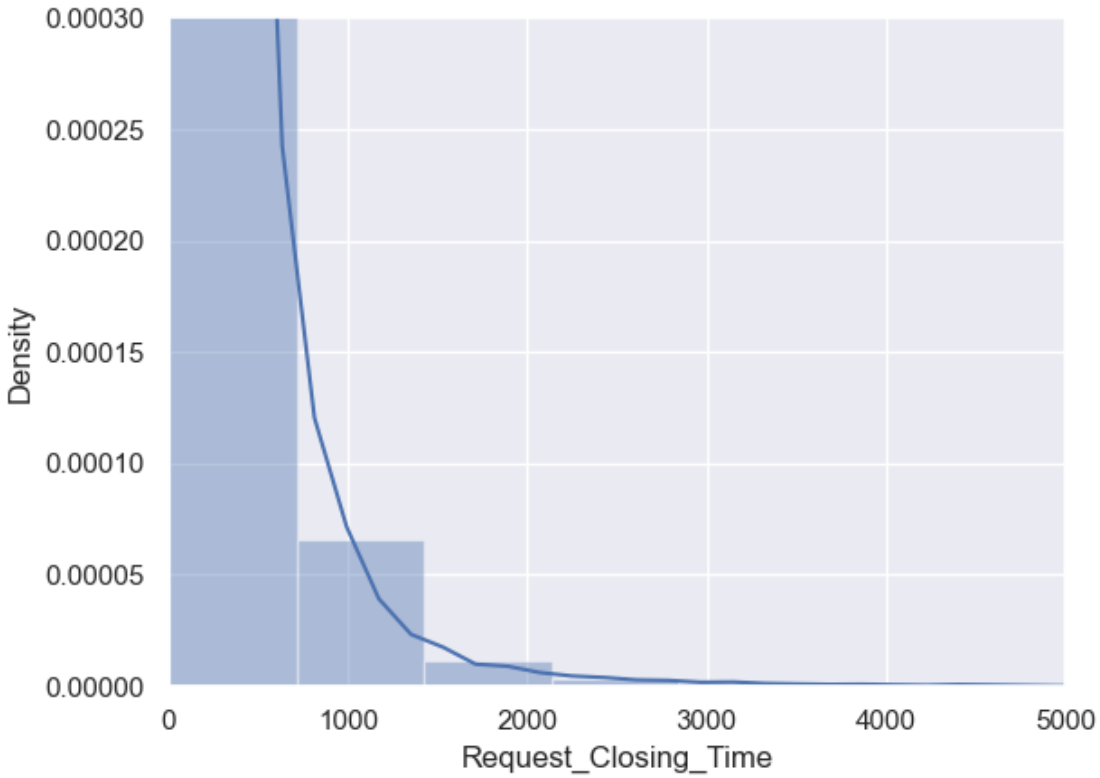
```
Total Number of Complaints :  300698

Percentage of complaints that took 99 hours or less :  32.88 %
Percentage of complaints that took 999 hours or less :  97.18 %
```

From the above data we can see that majority of the complaints needed more than 99 hours to be dealt with.

[24]:
```
sns.distplot(df["Request_Closing_Time"])
plt.xlim((0,5000))
plt.ylim((0,0.0003))
plt.show()
```

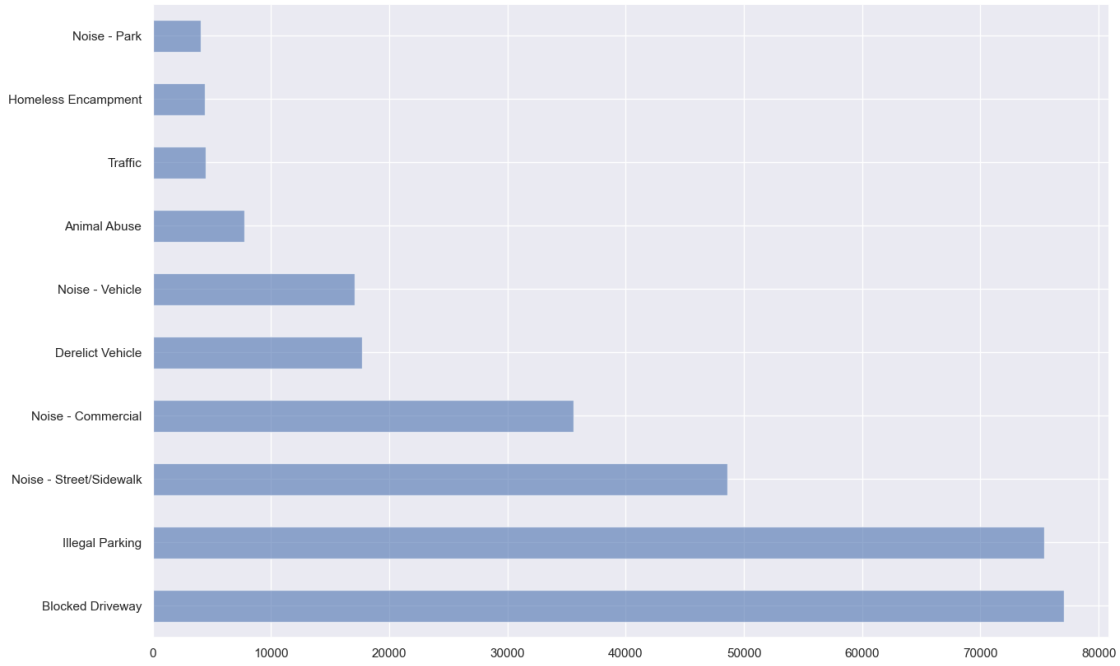
```
C:\Users\lrnem\AppData\Local\Temp\ipykernel_8728\2810770643.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df["Request_Closing_Time"])
```



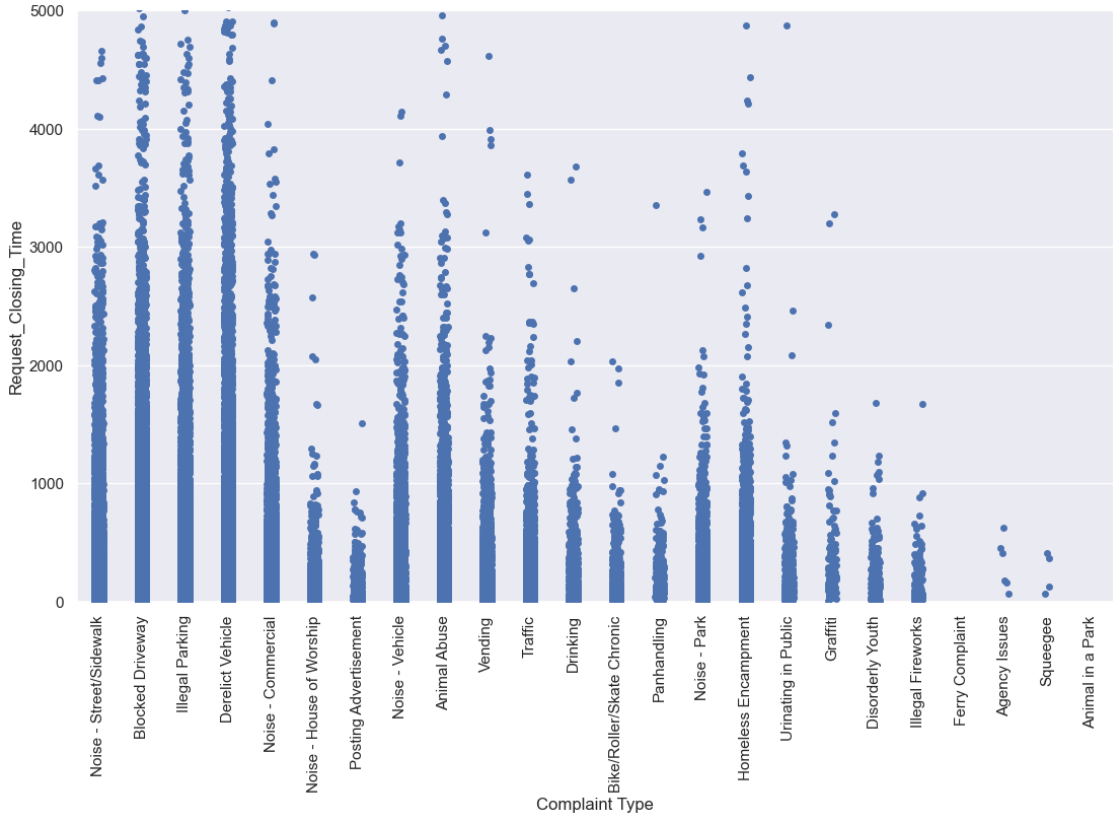
Now let us try to understand the major complaint types

[25]:
```
df['Complaint Type'].value_counts()[:10].plot(kind='barh',alpha=0.
  ↪6,figsize=(15,10))
plt.show()
```



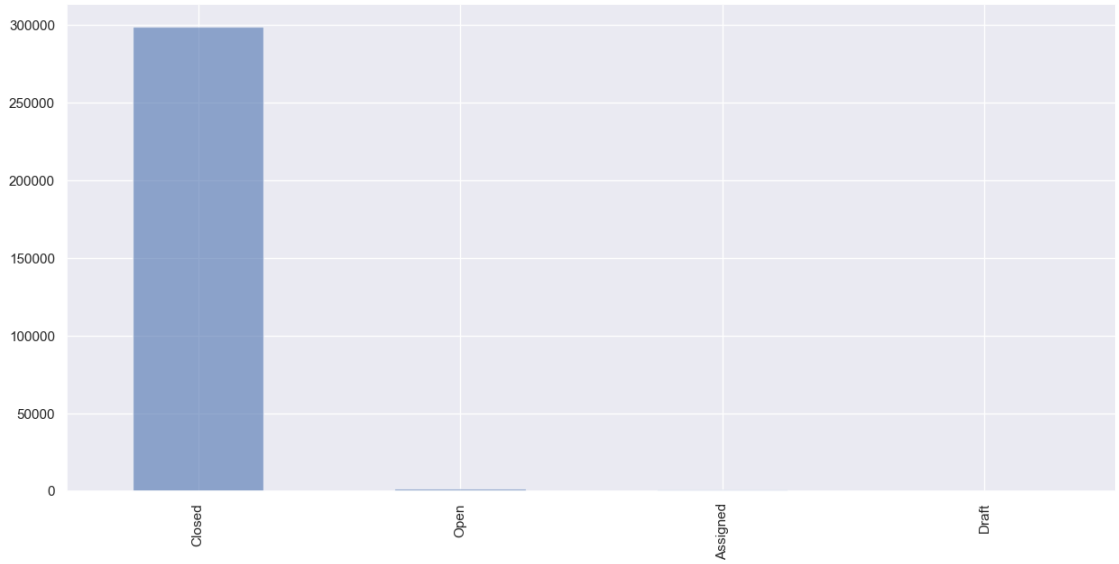
From the above graph we can see that majority of complaints are related to that of transporttion and parking and as such it needed more time to be resolved.

[26]:
```
crt = sns.catplot(x='Complaint Type', y="Request_Closing_Time",data=df)
crt.fig.set_figwidth(15)
crt.fig.set_figheight(7)
plt.xticks(rotation=90)
plt.ylim((0,5000))
plt.show()
```



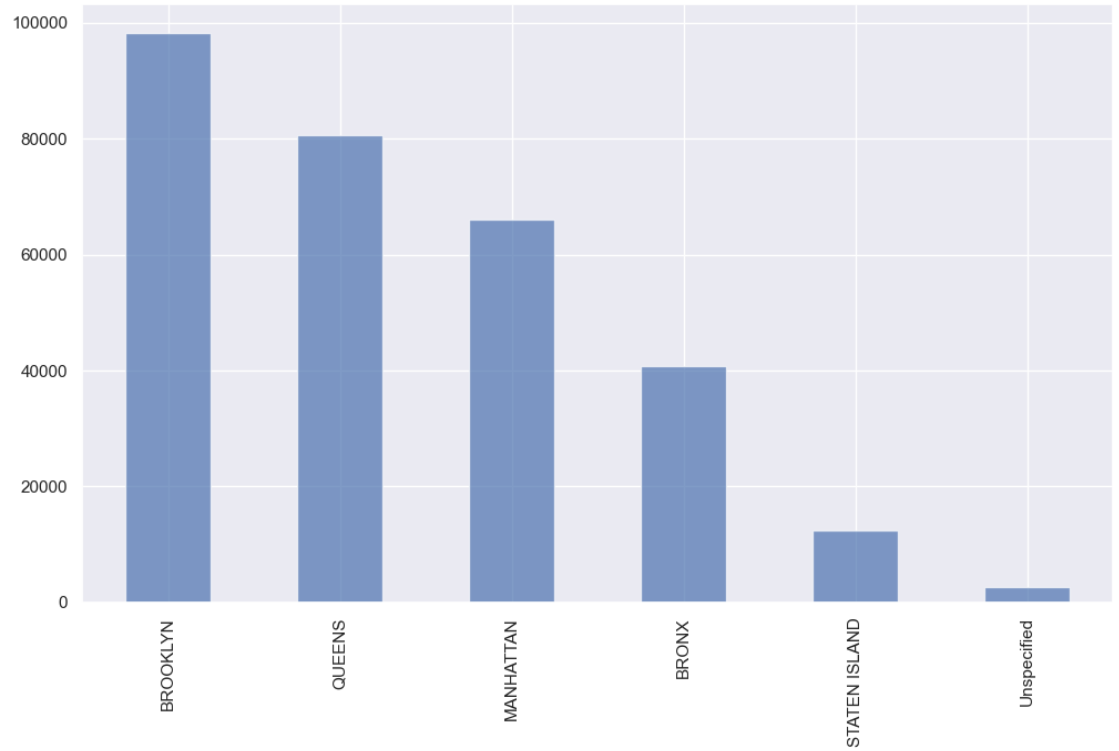
We can see from the above lot that major complaints arises from transport and take up huge time to be resolved and as such the government should take steps to improve the transport system and implement stricter vehicle laws.

[27]:
```
df['Status'].value_counts().plot(kind='bar',alpha=0.6,figsize=(15,7))
plt.show()
```



From the above plot we can see all complaints are now closed that is they are resolved.

[28]:
```
plt.figure(figsize=(12,7))
df['Borough'].value_counts().plot(kind='bar',alpha=0.7)
plt.show()
```



[29]:
```
for x in df["Borough"].unique():
    print("Percentage of complaints from ",x," : ",round((df["Borough"]==x).
 ↪sum()/len(df)*100,2))
```

```
Percentage of complaints from  MANHATTAN  :  21.99
Percentage of complaints from  QUEENS  :  26.82
Percentage of complaints from  BRONX  :  13.54
Percentage of complaints from  BROOKLYN  :  32.69
Percentage of complaints from  Unspecified  :  0.86
Percentage of complaints from  STATEN ISLAND  :  4.1
```

[30]:
```
#All unique locations
df["Location Type"].unique()
```

[30]:
```
array(['Street/Sidewalk', 'Club/Bar/Restaurant', 'Store/Commercial',
       'House of Worship', 'Residential Building/House',
       'Residential Building', 'Park/Playground', 'Vacant Lot',
       'House and Store', 'Highway', 'Commercial', 'Roadway Tunnel',
       'Subway Station', 'Parking Lot', 'Bridge', 'Terminal', nan,
       'Ferry', 'Park'], dtype=object)
```

[31]:
```
pd.DataFrame(df.groupby("Location Type")["Request_Closing_Time"].mean()).
 ↪sort_values("Request_Closing_Time")
```

[31]:

Location Type	Request_Closing_Time
Subway Station	142.250980
Club/Bar/Restaurant	186.074330
House of Worship	191.833279
Store/Commercial	198.089073
Park/Playground	207.137129
Highway	223.424221
Bridge	229.158333
Roadway Tunnel	266.525714
Street/Sidewalk	268.515306
Residential Building	289.089941
House and Store	300.795699
Residential Building/House	309.505679
Parking Lot	320.130342
Commercial	320.566129
Vacant Lot	448.435498
Park	20210.083333
Ferry	NaN
Terminal	NaN

Conclusion : Maximum time taken to resolved a complaint is in park and vacant lots whereas

complaints from subway or club/bar/restaurant take the lowest.

[32]:
```
#losing time of complaints with respect to city
pd.DataFrame(df.groupby("City")["Request_Closing_Time"].mean()).
 ↪sort_values("Request_Closing_Time")
```

[32]:

City	Request_Closing_Time
ARVERNE	135.895606
ROCKAWAY PARK	139.133736
LITTLE NECK	154.660316
OAKLAND GARDENS	157.853146
BAYSIDE	160.759992
FAR ROCKAWAY	167.399774
NEW YORK	178.357371
FLUSHING	181.081826
FOREST HILLS	193.449032
CORONA	193.670512
WHITESTONE	194.688843
FRESH MEADOWS	195.843207
COLLEGE POINT	196.417842
JACKSON HEIGHTS	196.419964
CENTRAL PARK	197.658591
ELMHURST	198.631095
REGO PARK	207.665668
BREEZY POINT	209.789444
EAST ELMHURST	214.659709
STATEN ISLAND	232.796699
Howard Beach	241.750000
BROOKLYN	242.878848
Long Island City	246.045522
Astoria	251.076304
RIDGEWOOD	266.507613
ASTORIA	275.934779
SAINT ALBANS	283.252098
KEW GARDENS	302.578556
Woodside	312.083333
JAMAICA	312.606051
SOUTH OZONE PARK	319.678662
MIDDLE VILLAGE	323.097583
RICHMOND HILL	329.658614
WOODHAVEN	335.728705
MASPETH	335.985805
SOUTH RICHMOND HILL	337.049201
OZONE PARK	340.863702
HOLLIS	345.610161
East Elmhurst	362.867857

```
BRONX                          365.769723
HOWARD BEACH                   369.652291
LONG ISLAND CITY               392.351457
SUNNYSIDE                      411.120332
WOODSIDE                       413.606029
NEW HYDE PARK                  453.365646
GLEN OAKS                      528.943900
SPRINGFIELD GARDENS            551.145130
ROSEDALE                       601.867552
CAMBRIA HEIGHTS                607.426555
BELLEROSE                      633.386578
QUEENS VILLAGE                 654.411273
FLORAL PARK                    703.171272
QUEENS                         815.586458
```

[33]:
```
#Percentage of missing values
pd.DataFrame((df.isnull().sum()/df.shape[0]*100)).
  ↪sort_values(0,ascending=False)[:20]
```

[33]:
```
                                    0
School or Citywide Complaint  100.000000
Garage Lot Name               100.000000
Vehicle Type                  100.000000
Taxi Pick Up Location         100.000000
Taxi Company Borough          100.000000
Ferry Direction                99.999667
Ferry Terminal Name            99.999335
Road Ramp                      99.929165
Bridge Highway Segment         99.929165
Bridge Highway Direction       99.919188
Bridge Highway Name            99.919188
Landmark                       99.883937
Intersection Street 2          85.579552
Intersection Street 1          85.414602
Cross Street 2                 16.554483
Cross Street 1                 16.388203
Street Name                    14.768971
Incident Address               14.768971
Descriptor                      1.966757
Latitude                        1.177261
```

We can see that school or city wide complaint, garage lot name, vehicle type, taxi pickup location, taxi company borough have 100% missing values, that could mean there are no complaints from those sectors.

[34]:
```
#We will now drop the missing values
dfn=df.loc[:,(df.isnull().sum()/df.shape[0]*100)<=50]
```

[35]:
```
print("Old df shape :",df.shape)
print("New df shape: ",dfn.shape)
```

```
Old df shape : (300698, 54)
New df shape:  (300698, 40)
```

[38]:
```
rem=[]
for x in dfn.columns.tolist():
    if dfn[x].nunique()<=3:
        print(x+ " "*10+" : ",dfn[x].unique())
        rem.append(x)
```

```
Agency           :  ['NYPD']
Agency Name           :  ['New York City Police Department' 'NYPD' 'Internal
Affairs Bureau']
Facility Type           :  ['Precinct' nan]
Park Facility Name           :  ['Unspecified' 'Alley Pond Park - Nature
Center']
School Name           :  ['Unspecified' 'Alley Pond Park - Nature Center']
School Number           :  ['Unspecified' 'Q001']
School Region           :  ['Unspecified' nan]
School Code           :  ['Unspecified' nan]
School Phone Number           :  ['Unspecified' '7182176034']
School Address           :  ['Unspecified' 'Grand Central Parkway, near the
soccer field']
School City           :  ['Unspecified' 'QUEENS']
School State           :  ['Unspecified' 'NY']
School Zip           :  ['Unspecified' nan]
School Not Found           :  ['N']
```

We can remove the unspecified data

[39]:
```
dfn.drop(rem,axis=1,inplace=True)
```

```
C:\Users\lrnem\AppData\Local\Temp\ipykernel_8728\3503437274.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  dfn.drop(rem,axis=1,inplace=True)
```

[40]:
```
dfn.shape
```

[40]: (300698, 26)

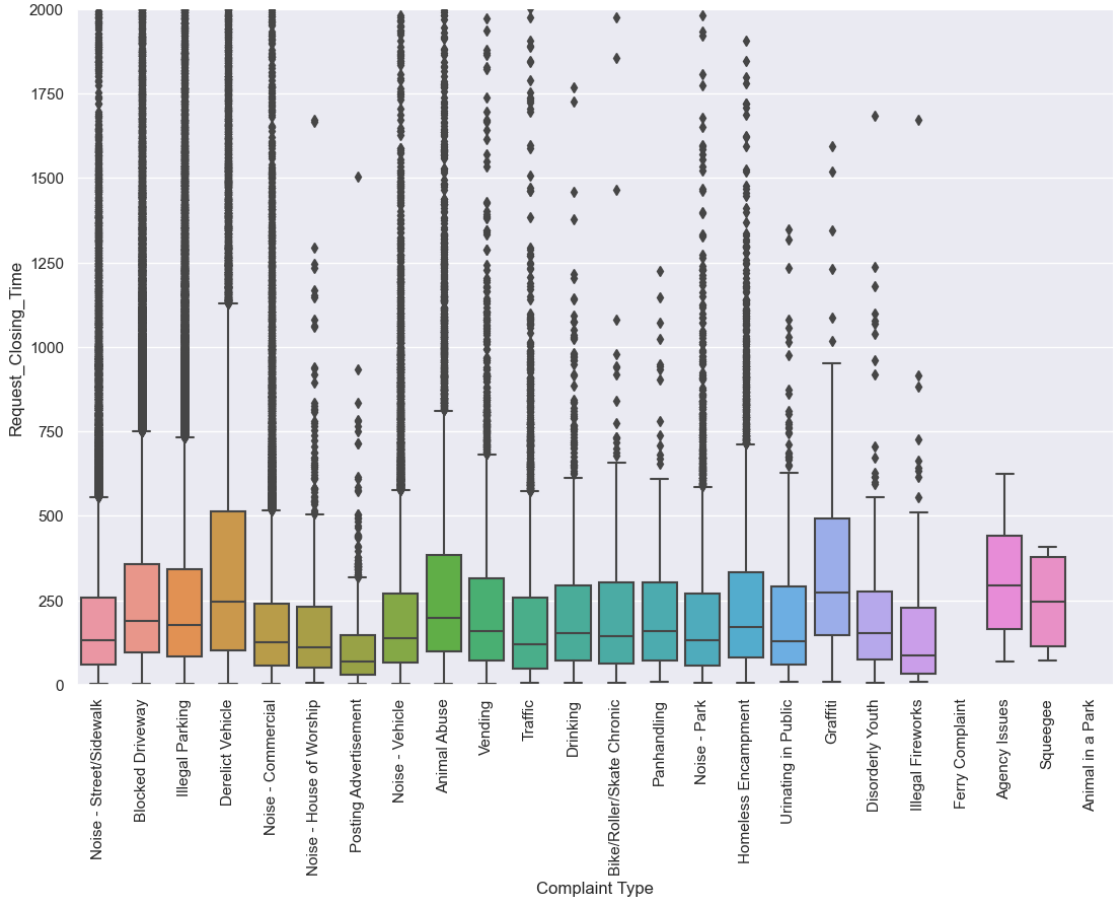
[41]:
```
dfn.head()
```

[41]:
```
   Unique Key        Created Date         Closed Date  \
0    32310363 2015-12-31 23:59:45 2016-01-01 00:55:00
1    32309934 2015-12-31 23:59:44 2016-01-01 01:26:00
2    32309159 2015-12-31 23:59:29 2016-01-01 04:51:00
3    32305098 2015-12-31 23:57:46 2016-01-01 07:43:00
4    32306529 2015-12-31 23:56:58 2016-01-01 03:24:00

             Complaint Type                   Descriptor    Location Type  \
0  Noise - Street/Sidewalk              Loud Music/Party  Street/Sidewalk
1         Blocked Driveway                     No Access  Street/Sidewalk
2         Blocked Driveway                     No Access  Street/Sidewalk
3          Illegal Parking  Commercial Overnight Parking  Street/Sidewalk
4          Illegal Parking              Blocked Sidewalk  Street/Sidewalk

   Incident Zip       Incident Address       Street Name   Cross Street 1  \
0       10034.0    71 VERMILYEA AVENUE  VERMILYEA AVENUE   ACADEMY STREET
1       11105.0        27-07 23 AVENUE         23 AVENUE        27 STREET
2       10458.0  2897 VALENTINE AVENUE  VALENTINE AVENUE  EAST 198 STREET
3       10461.0    2940 BAISLEY AVENUE    BAISLEY AVENUE    EDISON AVENUE
4       11373.0          87-14 57 ROAD           57 ROAD   SEABURY STREET

   … Resolution Action Updated Date Community Board    Borough  \
0  …                  01-01-16 0:55    12 MANHATTAN  MANHATTAN
1  …                  01-01-16 1:26       01 QUEENS     QUEENS
2  …                  01-01-16 4:51        07 BRONX      BRONX
3  …                  01-01-16 7:43        10 BRONX      BRONX
4  …                  01-01-16 3:24       04 QUEENS     QUEENS

  X Coordinate (State Plane) Y Coordinate (State Plane) Park Borough  \
0                  1005409.0                   254678.0    MANHATTAN
1                  1007766.0                   221986.0       QUEENS
2                  1015081.0                   256380.0        BRONX
3                  1031740.0                   243899.0        BRONX
4                  1019123.0                   206375.0       QUEENS

    Latitude  Longitude                                  Location  \
0  40.865682 -73.923501   (40.86568153633767, -73.92350095571744)
1  40.775945 -73.915094  (40.775945312321085, -73.91509393898605)
2  40.870325 -73.888525  (40.870324522111424, -73.88852464418646)
3  40.835994 -73.828379   (40.83599404683083, -73.82837939584206)
4  40.733060 -73.874170  (40.733059618956815, -73.87416975810375)

   Request_Closing_Time
0             55.250000
1             86.266667
2            291.516667
3            465.233333
```

```
4              207.033333

[5 rows x 26 columns]
```

[43]:
```
#Hypothesis testing
crt=sns.catplot(x="Complaint Type",y="Request_Closing_Time",kind="box",data=dfn)
crt.fig.set_figheight(8)
crt.fig.set_figwidth(15)
plt.xticks(rotation=90)
plt.ylim((0,2000))
```

[43]: (0.0, 2000.0)



H0: There is no significant different in mean of Request_Closing_Time for different Complaint

H1: There is signficant different in mean of Request_Closing_Time for different Complaint

[46]:
```
anova_df=pd.DataFrame()
anova_df["Request_Closing_Time"]=dfn["Request_Closing_Time"]
anova_df["Complaint"]=dfn["Complaint Type"]
```

```
anova_df.dropna(inplace=True)
anova_df.head()
```

[46]:

	Request_Closing_Time	Complaint
0	55.250000	Noise - Street/Sidewalk
1	86.266667	Blocked Driveway
2	291.516667	Blocked Driveway
3	465.233333	Illegal Parking
4	207.033333	Illegal Parking

[47]:
```
lm=ols("Request_Closing_Time~Complaint",data=anova_df).fit()
table=sm.stats.anova_lm(lm)
table
```

[47]:

	df	sum_sq	mean_sq	F	PR(>F)
Complaint	22.0	1.455049e+09	6.613860e+07	514.177089	0.0
Residual	298511.0	3.839747e+10	1.286300e+05	NaN	NaN

[48]:
```
chi_sq=pd.DataFrame()
chi_sq["Location Type"]=dfn["Location Type"]
chi_sq["Complaint Type"]=dfn["Complaint Type"]
chi_sq.dropna(inplace=True)
```

[49]:
```
data_crosstab = pd.crosstab( chi_sq["Location Type"],chi_sq["Complaint Type"])
```

[50]:
```
stat, p, dof, expected = chi2_contingency(data_crosstab)

alpha = 0.05
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
```

```
Dependent (reject H0)
```

Conclusions : 1. Different complaints last for different duration. 2. Complaints are different in different locations. 3. Majority complaints are from transport sector. 4. School sector has the lowest number of complaints(next to none).

[ ]:
```

```