# Customer Service Requests Analysis

August 7, 2023

[2]:
```
# importing libraries

import numpy as n
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

import warnings
warnings.filterwarnings("ignore")

from scipy import stats
from scipy.stats import chi2_contingency

import statsmodels.api as sm
from statsmodels.formula.api import ols
```

[ ]:
```
2nd PROJECT
DESCRIPTION

Reduce the time a Mercedes-Benz spends on the test bench.

Problem Statement Scenario:
Since the first automobile, the Benz Patent Motor Car in 1886, Mercedes-Benz␣
 ↪has stood for important automotive innovations. These include the passenger␣
 ↪safety cell with a crumple zone, the airbag, and intelligent assistance␣
 ↪systems. Mercedes-Benz applies for nearly 2000 patents per year, making the␣
 ↪brand the European leader among premium carmakers. Mercedes-Benz is the␣
 ↪leader in the premium car industry. With a huge selection of features and␣
 ↪options, customers can choose the customized Mercedes-Benz of their dreams.
```

```
To ensure the safety and reliability of every unique car configuration before␣
 ↪they hit the road, the company's engineers have developed a robust testing␣
 ↪system. As one of the world's biggest manufacturers of premium cars, safety␣
 ↪and efficiency are paramount on Mercedes-Benz's production lines. However,␣
 ↪optimizing the speed of their testing system for many possible feature␣
 ↪combinations is complex and time-consuming without a powerful algorithmic␣
 ↪approach.

You are required to reduce the time that cars spend on the test bench. Others␣
 ↪will work with a dataset representing different permutations of features in␣
 ↪a Mercedes-Benz car to predict the time it takes to pass testing. Optimal␣
 ↪algorithms will contribute to faster testing, resulting in lower carbon␣
 ↪dioxide emissions without reducing Mercedes-Benz's standards.

Following actions should be performed:

If for any column(s), the variance is equal to zero, then you need to remove␣
 ↪those variable(s).
Check for null and unique values for test and train sets.
Apply label encoder.
Perform dimensionality reduction.
Predict your test_df values using XGBoost.
3rd PROJECT
Cardiovascular diseases are the leading cause of death globally. It is␣
 ↪therefore necessary to identify the causes and develop a system to predict␣
 ↪heart attacks in an effective manner. The data below has the information␣
 ↪about the factors that might have an impact on cardiovascular health.
4th project
DESCRIPTION

NIDDK (National Institute of Diabetes and Digestive and Kidney Diseases)␣
 ↪research creates knowledge about and treatments for the most chronic,␣
 ↪costly, and consequential diseases.

The dataset used in this project is originally from NIDDK. The objective is to␣
 ↪predict whether or not a patient has diabetes, based on certain diagnostic␣
 ↪measurements included in the dataset.
Build a model to accurately predict whether the patients in the dataset have␣
 ↪diabetes or not.


Dataset Description

The datasets consists of several medical predictor variables and one target␣
 ↪variable (Outcome). Predictor variables includes the number of pregnancies␣
 ↪the patient has had, their BMI, insulin level, age, and more.
```

```
Variables        Description
Pregnancies        Number of times pregnant
Glucose        Plasma glucose concentration in an oral glucose tolerance test
BloodPressure        Diastolic blood pressure (mm Hg)
SkinThickness        Triceps skinfold thickness (mm)
Insulin        Two hour serum insulin
BMI        Body Mass Index
DiabetesPedigreeFunction        Diabetes pedigree function
Age        Age in years
Outcome        Class variable (either 0 or 1). 268 of 768 values are 1, and the␣
 ↪others are 0
Project Task: Week 1

Data Exploration:

Perform descriptive analysis. Understand the variables and their corresponding␣
 ↪values. On the columns below, a value of zero does not make sense and thus␣
 ↪indicates missing value:

Glucose

BloodPressure

SkinThickness

Insulin

BMI

Visually explore these variables using histograms. Treat the missing values␣
 ↪accordingly.

There are integer and float data type variables in this dataset. Create a count␣
 ↪(frequency) plot describing the data types and the count of variables.



Data Exploration:

Check the balance of the data by plotting the count of outcomes by their value.␣
 ↪Describe your findings and plan future course of action.

Create scatter charts between the pair of variables to understand the␣
 ↪relationships. Describe your findings.
```

```
Perform correlation analysis. Visually explore it using a heat map.



Project Task: Week 2

Data Modeling:

Devise strategies for model building. It is important to decide the right␣
 ↪validation framework. Express your thought process.

Apply an appropriate classification algorithm to build a model.

Compare various models with the results from KNN algorithm.

Create a classification report by analyzing sensitivity, specificity, AUC (ROC␣
 ↪curve), etc.

Please be descriptive to explain what values of these parameter you have used.



Data Reporting:

Create a dashboard in tableau by choosing appropriate chart types and metrics␣
 ↪useful for the business. The dashboard must entail the following:

Pie chart to describe the diabetic or non-diabetic population

Scatter charts between relevant variables to analyze the relationships

Histogram or frequency charts to analyze the distribution of the data

Heatmap of correlation analysis among the relevant variables

Create bins of these age values: 20-25, 25-30, 30-35, etc. Analyze different␣
 ↪variables for these age brackets using a bubble chart.

 5th project
DESCRIPTION

For safe and secure lending experience, it's important to analyze the past data.
 ↪ In this project, you have to build a deep learning model to predict the␣
 ↪chance of default for future loans using the historical data. As you will␣
 ↪see, this dataset is highly imbalanced and includes a lot of features that␣
 ↪make this problem more challenging.
```

```
Objective: Create a model that predicts whether or not an applicant will be able to repay a loan using historical data.

Domain: Finance

Analysis to be done: Perform data preprocessing and build a deep learning prediction model.

Steps to be done:

    Load the dataset that is given to you
    Check for null values in the dataset
    Print percentage of default to payer of the dataset for the TARGET column
    Balance the dataset if the data is imbalanced
    Plot the balanced data or imbalanced data
    Encode the columns that is required for the model
    Calculate Sensitivity as a metrice
    Calculate area under receiver operating characteristics curve . i want data sciences keyword oriented cv.
```

# 1 uploading the raw data

[3]:
```
data= pd.read_csv("a_Present.csv")
```

[4]:
```
data.head(8)
```

[4]:

| | Unique Key | Created Date | Closed Date | Agency | Agency Name | Complaint Type |
|---|---|---|---|---|---|---|
| 0 | 32310363 | 12/31/2015 11:59:45 PM | 01-01-16 0:55 | NYPD | New York City Police Department | Noise - Street/Sidewalk |
| 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 |
| 5 | 32306554 | 12/31/2015 11:56:30 PM | 01-01-16 1:50 | NYPD | New York City Police Department | Illegal Parking |
| 6 | 32306559 | 12/31/2015 11:55:32 PM | 01-01-16 1:53 | NYPD | New York City Police Department | Illegal Parking |
| 7 | 32307009 | 12/31/2015 11:54:05 PM | 01-01-16 1:42 | NYPD | New York City Police Department | Illegal Parking |

```
6  New York City Police Department          Illegal Parking
7  New York City Police Department         Blocked Driveway

                     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
5  Posted Parking Sign Violation  Street/Sidewalk       11215.0
6                 Blocked Hydrant  Street/Sidewalk       10032.0
7                     No Access  Street/Sidewalk       10457.0

         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
5          260 21 STREET  …                 NaN                       NaN
6    524 WEST 169 STREET  …                 NaN                       NaN
7    501 EAST 171 STREET  …                 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
5       NaN                    NaN             NaN             NaN
6       NaN                    NaN             NaN             NaN
7       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
5                 NaN  40.660823 -73.992568
6                 NaN  40.840848 -73.937375
7                 NaN  40.837503 -73.902905

                                  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   (40.66082272389114, -73.99256786342693)
6   (40.840847591440415, -73.9373750864581)
7   (40.83750262540012, -73.90290517326568)

[8 rows x 53 columns]
```

[5]:
```
data.tail()
```

[5]:
```
        Unique Key            Created Date            Closed Date Agency  \
300693    30281872  03/29/2015 12:33:41 AM                     NaN   NYPD
300694    30281230  03/29/2015 12:33:28 AM  03/29/2015 02:33:59 AM   NYPD
300695    30283424  03/29/2015 12:33:03 AM  03/29/2015 03:40:20 AM   NYPD
300696    30280004  03/29/2015 12:33:02 AM  03/29/2015 04:38:35 AM   NYPD
300697    30281825  03/29/2015 12:33:01 AM  03/29/2015 04:41:50 AM   NYPD

                           Agency Name      Complaint Type        Descriptor  \
300693  New York City Police Department  Noise - Commercial  Loud Music/Party
300694  New York City Police Department    Blocked Driveway    Partial Access
300695  New York City Police Department  Noise - Commercial  Loud Music/Party
300696  New York City Police Department  Noise - Commercial  Loud Music/Party
300697  New York City Police Department  Noise - Commercial  Loud Music/Party

              Location Type  Incident Zip          Incident Address  …  \
300693  Club/Bar/Restaurant           NaN           CRESCENT AVENUE  …
300694     Street/Sidewalk       11418.0          100-17 87 AVENUE  …
300695  Club/Bar/Restaurant       11206.0         162 THROOP AVENUE  …
300696  Club/Bar/Restaurant       10461.0  3151 EAST TREMONT AVENUE  …
300697    Store/Commercial       10036.0        251 WEST 48 STREET  …

       Bridge Highway Name Bridge Highway Direction Road Ramp  \
300693                 NaN                      NaN       NaN
300694                 NaN                      NaN       NaN
300695                 NaN                      NaN       NaN
300696                 NaN                      NaN       NaN
300697                 NaN                      NaN       NaN

       Bridge Highway Segment Garage Lot Name Ferry Direction  \
300693                    NaN             NaN              NaN
300694                    NaN             NaN              NaN
300695                    NaN             NaN              NaN
300696                    NaN             NaN              NaN
300697                    NaN             NaN              NaN

       Ferry Terminal Name   Latitude  Longitude  \
300693                 NaN        NaN        NaN
```

```
300694                NaN  40.694077 -73.846087
300695                NaN  40.699590 -73.944234
300696                NaN  40.837708 -73.834587
300697                NaN  40.760583 -73.985922

                                     Location
300693                                    NaN
300694   (40.69407728322387, -73.8460866160573)
300695  (40.69959035300927, -73.94423377144169)
300696   (40.8377075854206, -73.83458731019586)
300697  (40.76058322950115, -73.98592204392392)

[5 rows x 53 columns]
```

[6]:
```
data.shape
```

[6]:
```
(300698, 53)
```

[7]:
```
data.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
```

## 2 Desctriptive analysis

[8]:
```
data.describe()
```

[8]:

| | 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
```

[9]:
```
data.isna().sum()
```

[9]:
```
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
```

3 the raw data shows abdundance of missing values.Therefore, to understanding the data with this innaccurate information its kind of difficult. So, move on with next process.

[10]:
```
complaintTypecity = pd.DataFrame({'count':
                                  data.groupby(['Complaint Type','City']).
 ↪size()}).reset_index()
complaintTypecity
```

[10]:

| | Complaint Type | City | count |
|---|---|---|---|
| 0 | Animal Abuse | ARVERNE | 38 |
| 1 | Animal Abuse | ASTORIA | 125 |
| 2 | Animal Abuse | BAYSIDE | 37 |
| 3 | Animal Abuse | BELLEROSE | 7 |
| 4 | Animal Abuse | BREEZY POINT | 2 |
| .. | … | … | … |
| 759 | Vending | STATEN ISLAND | 25 |
| 760 | Vending | SUNNYSIDE | 15 |
| 761 | Vending | WHITESTONE | 1 |
| 762 | Vending | WOODHAVEN | 6 |
| 763 | Vending | WOODSIDE | 15 |

[764 rows x 3 columns]

## 4 feature creation

[11]:
```
# Converting the data into datetime
data["Created Date"]=pd.to_datetime(data["Created Date"])
data["Closed Date"]=pd.to_datetime(data["Closed Date"])
```

[12]:
```
#Creating the new column that consist the amount of time taken to resolve the␣
 ↪complaint
data["Request_Closing_Time"]=(data["Closed Date"]-data["Created Date"])

Request_Closing_Time=[]
for x in (data["Closed Date"]-data["Created Date"]):
    close=x.total_seconds()/60
    Request_Closing_Time.append(close)

data["Request_Closing_Time"]=Request_Closing_Time
```

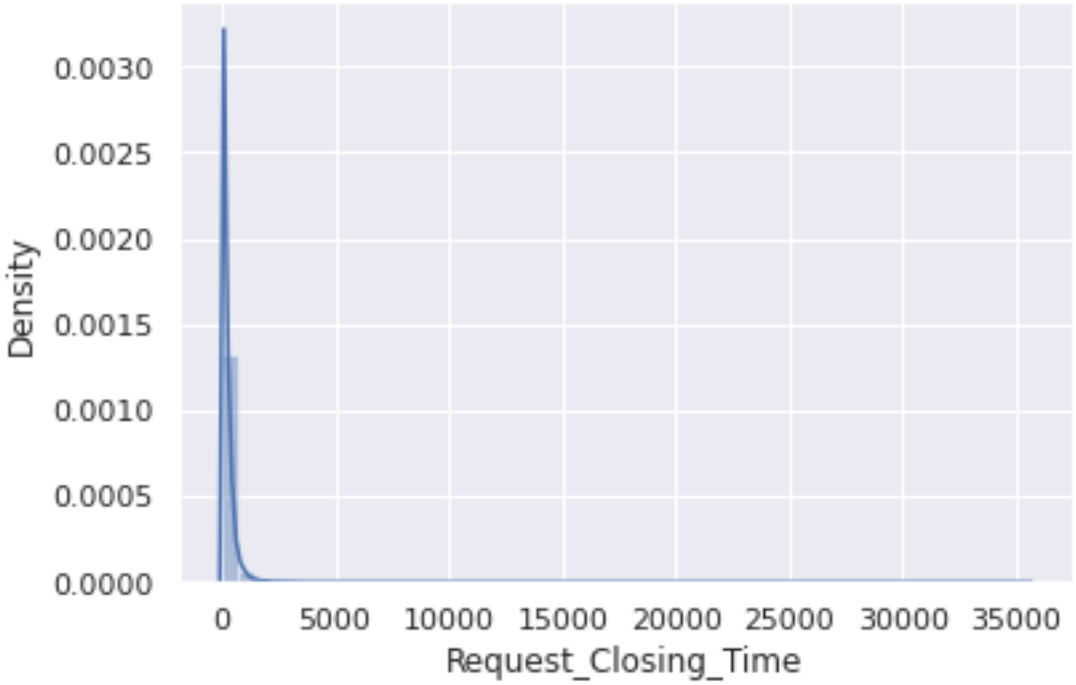
[13]:
```
data["Agency"].unique()
```

[13]:
```
array(['NYPD'], dtype=object)
```

## 5 all data belongs to New York City Police Department.

[14]:
```
#Univariate Distribution Plot for Request Closing Time
sns.distplot(data["Request_Closing_Time"])
plt.show
```

[14]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

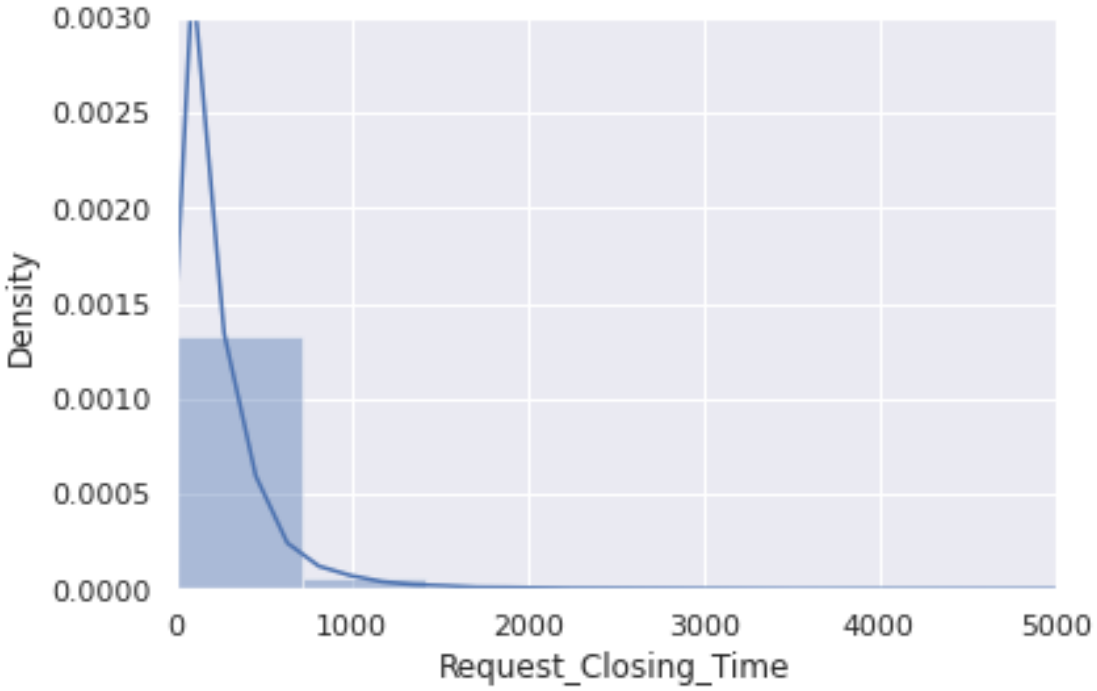



[15]:
```
print("Total Number of Concerns : ",len(data),"\n")
print("Percentage of Requests took less than 100 hour to get solved   :␣
 ↪",round((len(data)-(data["Request_Closing_Time"]>100).sum())/
 ↪len(data)*100,2),"%")
print("Percentage of Requests took less than 1000 hour to get solved :␣
 ↪",round((len(data)-(data["Request_Closing_Time"]>1000).sum())/
 ↪len(data)*100,2),"%")
```

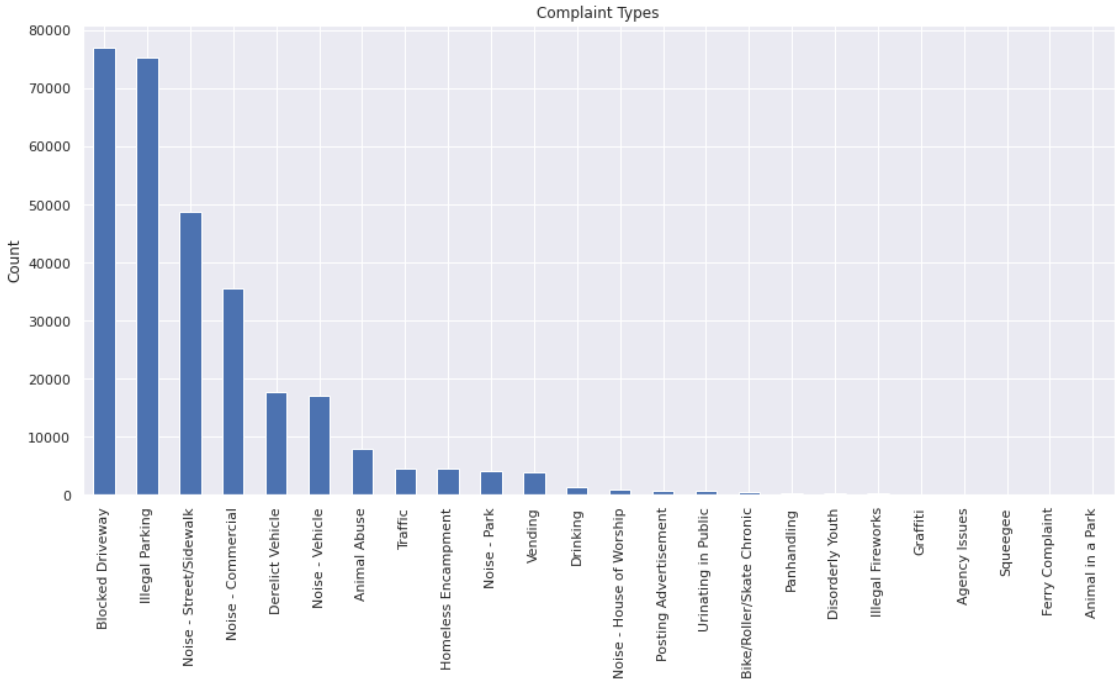
```
Total Number of Concerns :  300698

Percentage of Requests took less than 100 hour to get solved   :  33.32 %
Percentage of Requests took less than 1000 hour to get solved :  97.19 %
```

**6 as we can see in the above statements 33% of requests slovevd in 4 days(100hr) and 97% of requests sloved 41 days(1000hr)**

[16]:
```
#Univariate Distribution Plot for Request Closing Time
sns.distplot(data["Request_Closing_Time"])
plt.xlim((0,5000))
plt.ylim((0,0.003))
plt.show()
```




[17]:
```
# Count plot to understand the type of the complaint raised
data['Complaint Type'].value_counts().plot(kind = 'bar', figsize=(15, 7),␣
↪title='Complaint Types', ylabel='Count', grid=True)
plt.show()
```



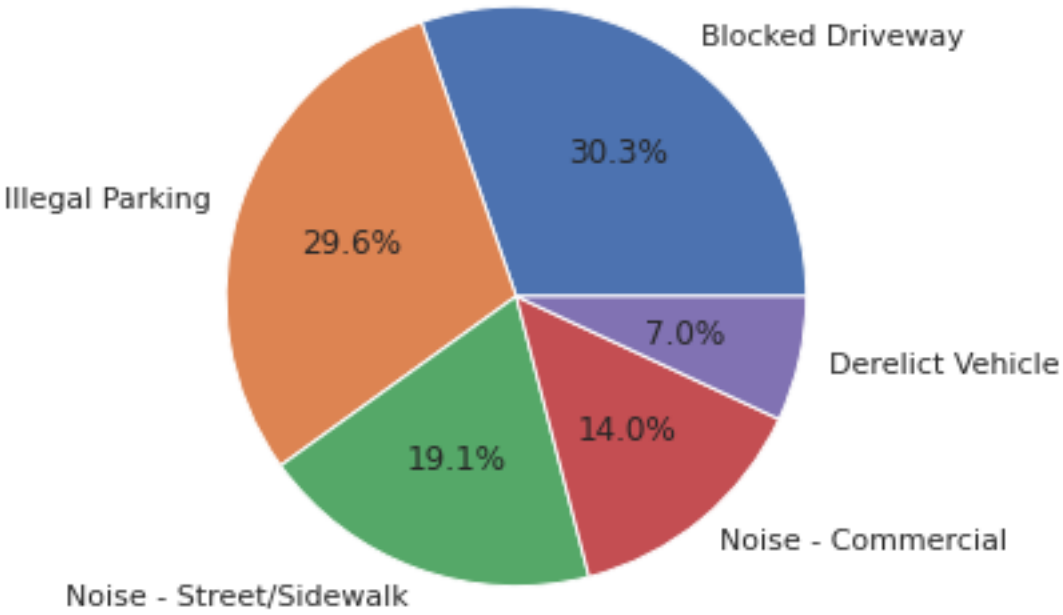

[18]:
```
majorcomplints=data.dropna(subset=["Complaint Type"])
majorcomplints=data.groupby("Complaint Type")

sortedComplaintType = majorcomplints.size().sort_values(ascending = False)
sortedComplaintType = sortedComplaintType.to_frame('count').reset_index()

sortedComplaintType
sortedComplaintType.head(10)
```

[18]:

| | Complaint Type | count |
|---|---|---|
| 0 | Blocked Driveway | 77044 |
| 1 | Illegal Parking | 75361 |
| 2 | Noise - Street/Sidewalk | 48612 |
| 3 | Noise - Commercial | 35577 |
| 4 | Derelict Vehicle | 17718 |
| 5 | Noise - Vehicle | 17083 |
| 6 | Animal Abuse | 7778 |
| 7 | Traffic | 4498 |
| 8 | Homeless Encampment | 4416 |
| 9 | Noise - Park | 4042 |

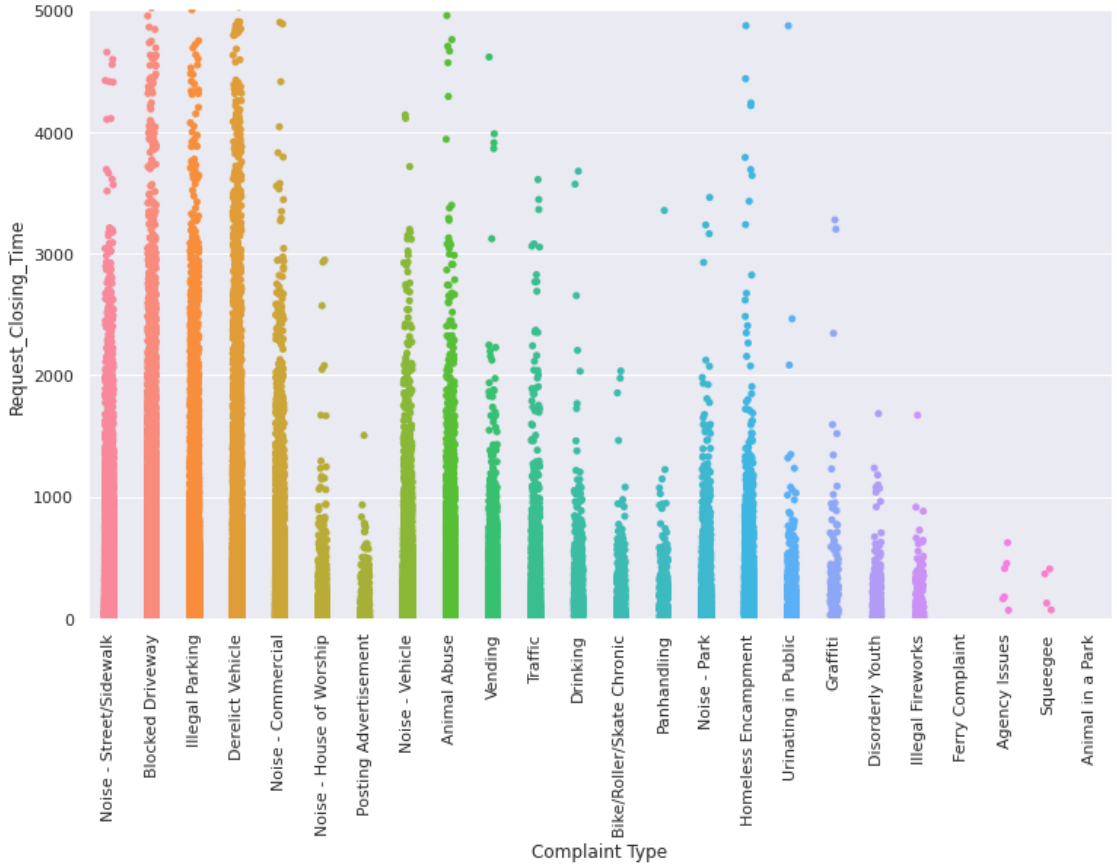
[49]:
```
sortedComplaintType = sortedComplaintType.head()
plt.figure(figsize=(5,5))
plt.pie(sortedComplaintType['count'],labels=sortedComplaintType["Complaint␣
 ↪Type"], autopct="%1.1f%%")
```

```
plt.show()
```



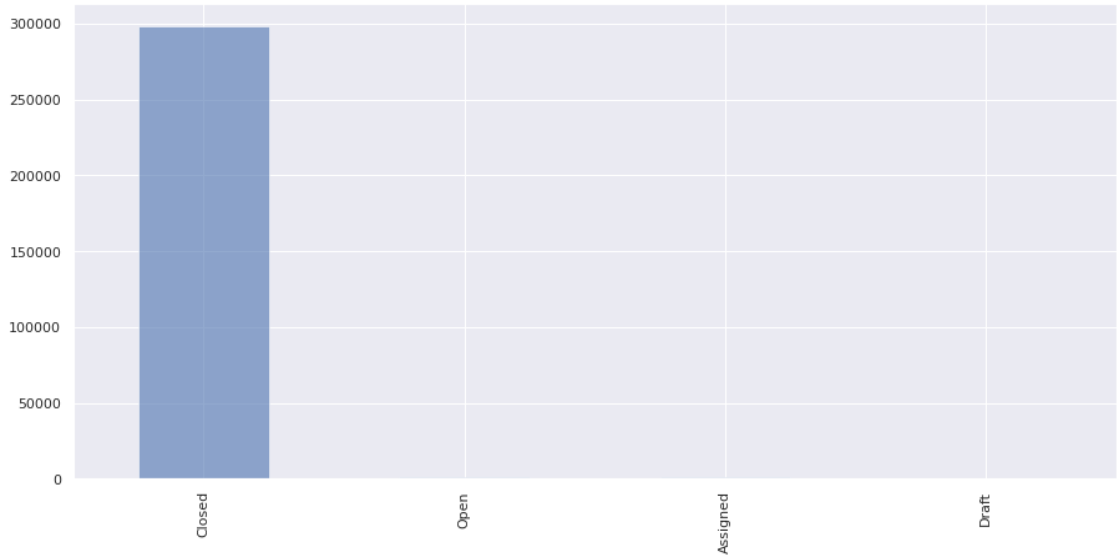

- 85% of the the requests belongs to transport (blocked driveway,illegal parking,noise-street/sidewalk..etc)

[20]:
```
#type of complaints are taking more time to get resolved, use by Categorical␣
↪Scatter Plot
g=sns.catplot(x='Complaint Type', y="Request_Closing_Time",data=data)
g.fig.set_figwidth(15)
g.fig.set_figheight(7)
plt.xticks(rotation=90)
plt.ylim((0,5000))
plt.show()
```




- almost 90% of resquest belongs to transport (street/sidewalk, Blocked driveway,Illegal Parking, derelict Vehicle , Road Traffic etc ).

[21]:
```
# Count plot to know the status of the requests
data['Status'].value_counts().plot(kind='bar',alpha=0.6,figsize=(15,7))
plt.show()
```




- almost 95% of cases are closed

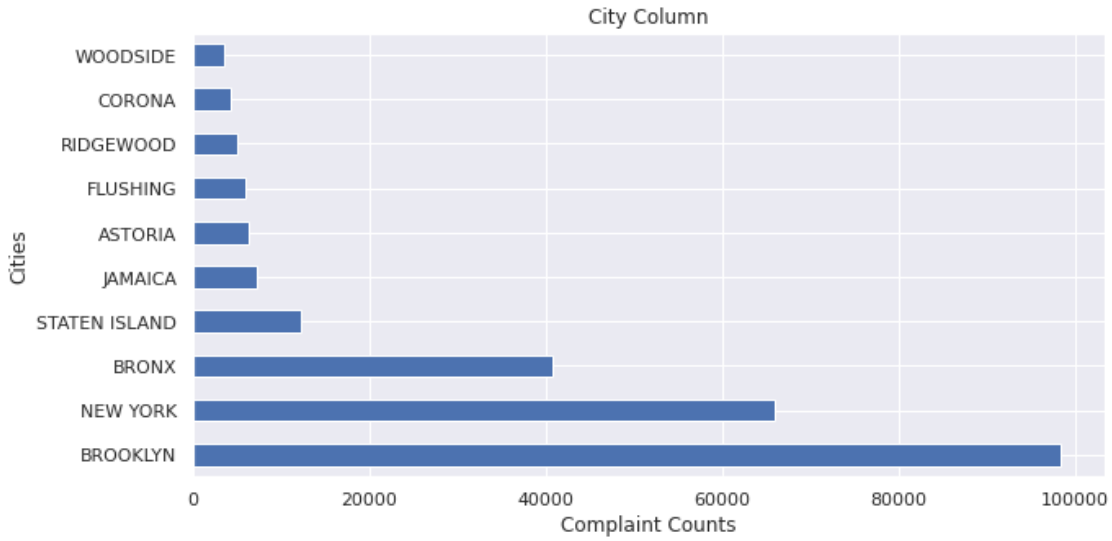
## 7 Cities

[22]:
```
data['City'].value_counts().head(10)
```

[22]:
```
BROOKLYN         98307
NEW YORK         65994
BRONX            40702
STATEN ISLAND    12343
JAMAICA           7296
ASTORIA           6330
FLUSHING          5971
RIDGEWOOD         5163
CORONA            4295
WOODSIDE          3544
Name: City, dtype: int64
```

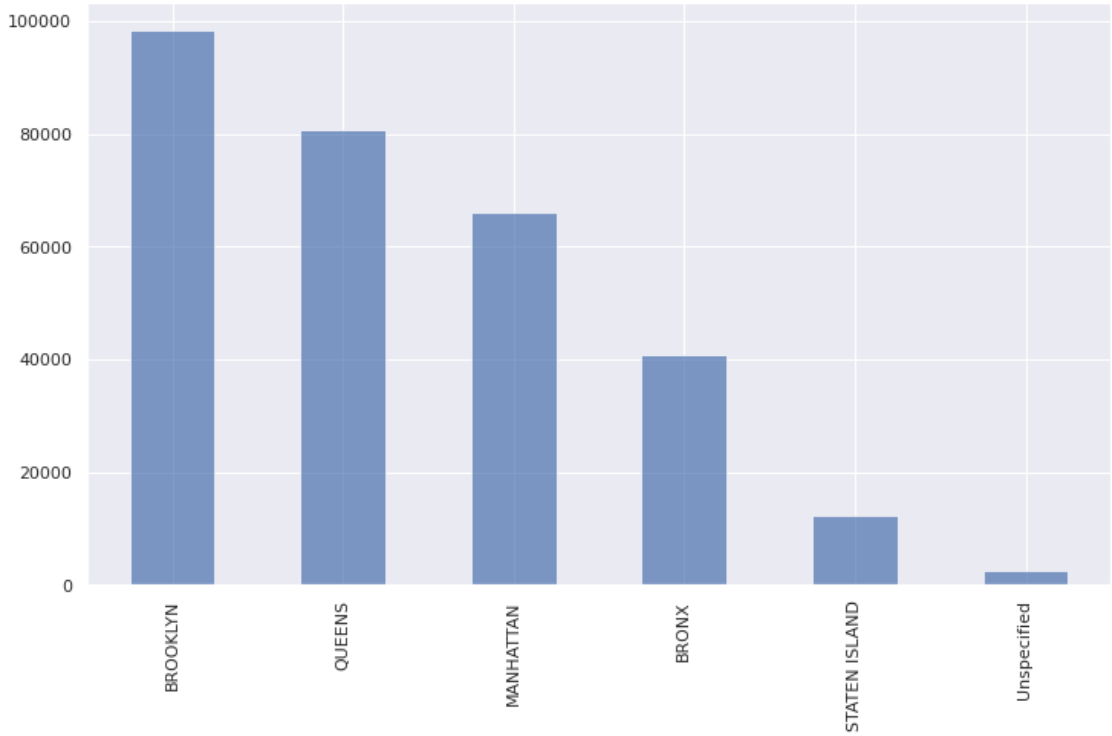
[23]:
```
# plotting the cities
data['City'].value_counts().head(10).plot( kind ='barh', grid=True,␣
 ↪figsize=(10, 5), title='City Column', ylabel='Cities')
plt.xlabel('Complaint Counts')
```

[23]:
```
Text(0.5, 0, 'Complaint Counts')
```




- majority of the complaints are from BROOKLYN.

[24]:
```
#Count Plot for Coloumn Borough
plt.figure(figsize=(12,7))
data['Borough'].value_counts().plot(kind='bar',alpha=0.7)
plt.show()
```



[25]:
```
for x in data["Borough"].unique():
    print("Percentage of Request from ",x," Division :␣
 ↪",round((data["Borough"]==x).sum()/len(data)*100,2))
```

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

we only analyse one column. Lets analyse Borough and Complaint Types

[26]:
```
top_6_complaints = data['Complaint Type'].value_counts()[:6].keys()
top_6_complaints
```

[26]:
```
Index(['Blocked Driveway', 'Illegal Parking', 'Noise - Street/Sidewalk',
       'Noise - Commercial', 'Derelict Vehicle', 'Noise - Vehicle'],
      dtype='object')
```

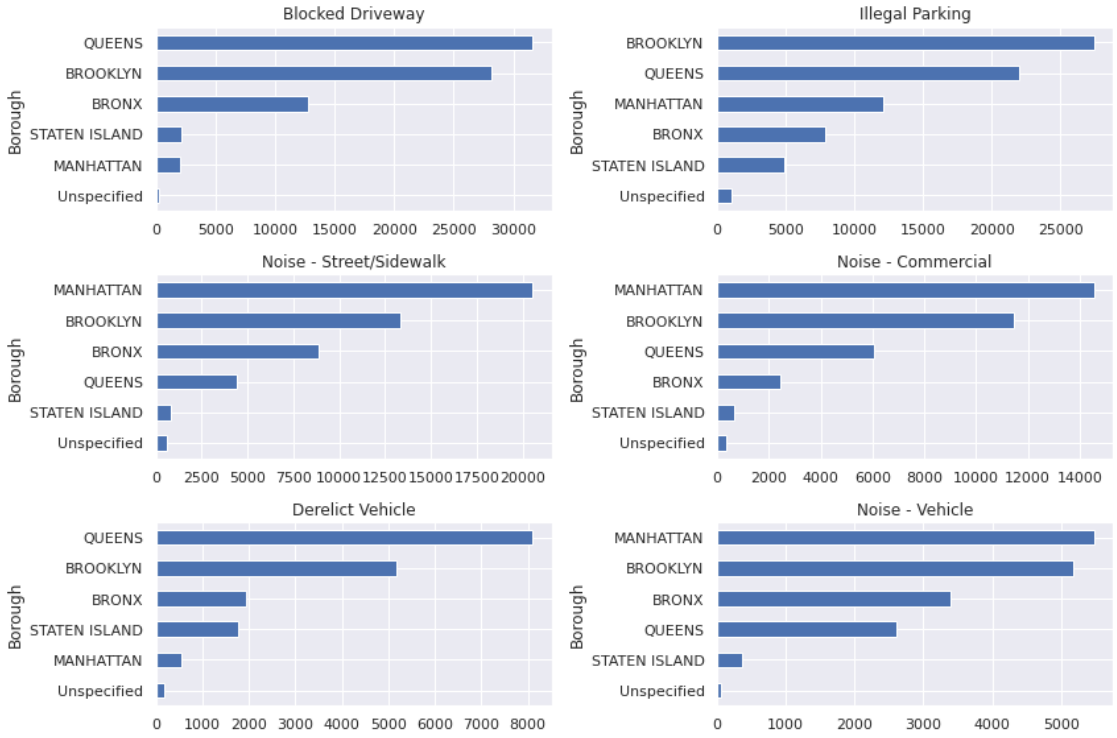
[27]:
```
borough_complaints = data.groupby(['Borough', 'Complaint Type']).size().
 ↪unstack()
borough_complaints = borough_complaints[top_6_complaints]
borough_complaints
```

[27]:

| Complaint Type | Blocked Driveway | Illegal Parking | Noise - Street/Sidewalk | Noise - Commercial | Derelict Vehicle | Noise - Vehicle |
|---|---|---|---|---|---|---|
| Borough | | | | | | |
| BRONX | 12755.0 | 7859.0 | 8891.0 | 2434.0 | 1953.0 | 3396.0 |
| BROOKLYN | 28148.0 | 27462.0 | 13355.0 | 11463.0 | 5181.0 | 5177.0 |
| MANHATTAN | 2073.0 | 12132.0 | 20550.0 | 14560.0 | 537.0 | 5485.0 |
| QUEENS | 31644.0 | 21982.0 | 4407.0 | 6075.0 | 8110.0 | 2615.0 |
| STATEN ISLAND | 2142.0 | 4886.0 | 820.0 | 679.0 | 1766.0 | 356.0 |
| Unspecified | 282.0 | 1040.0 | 589.0 | 366.0 | 171.0 | 54.0 |

[28]:
```
# Plotting Borough per Complaint Type
col_number = 2
row_number = 3
```

```
fig, axes = plt.subplots(row_number,col_number, figsize=(12,8))

for i, (label,col) in enumerate(borough_complaints.iteritems()):
    ax = axes[int(i/col_number), i%col_number]
    col = col.sort_values(ascending=True)[:15]
    col.plot(kind='barh', ax=ax, grid=True)
    ax.set_title(label)

plt.tight_layout()
```




ANALYSIS:

- BROOKLYN, QUEENS and BRONX has most complaints of Blocked Driveway.
- MANHATTAN has most complaints of Noise - Street/Sidewalk.
- STATEN ISLAND has most complaints of Illegal Parking

[29]:
```
#Request Closing Time for all location Type sorted in ascending Order
pd.DataFrame(data.groupby("Location Type")["Request_Closing_Time"].mean()).
 ↪sort_values("Request_Closing_Time")
```

[29]:

| 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
```

We see that maximum(mean) time to resolve the complaint is taken in Park,Vacant Lot and Commercial areas whereas the cases in the Subway Station and Restaurent are resolved in very less time

[30]:
```
#Request Closing Time for all City sorted in ascending Order
pd.DataFrame(data.groupby("City")["Request_Closing_Time"].mean()).
 ↪sort_values("Request_Closing_Time")
```

[30]:

| 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
```

[31]:
```
#Percentage Of Missing Value
pd.DataFrame((data.isnull().sum()/data.shape[0]*100)).
 ↪sort_values(0,ascending=False)[:20]
```

[31]:
```
                                    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
```

[32]:
```
#Remove the column with very high percentage of missing value
new_df=data.loc[:,(data.isnull().sum()/data.shape[0]*100)<=50]
new_df.head()
```

[32]:
```
   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  … School Phone Number School Address  School City  \
0    71 VERMILYEA AVENUE  …         Unspecified    Unspecified  Unspecified
1        27-07 23 AVENUE  …         Unspecified    Unspecified  Unspecified
2  2897 VALENTINE AVENUE  …         Unspecified    Unspecified  Unspecified
3    2940 BAISLEY AVENUE  …         Unspecified    Unspecified  Unspecified
4          87-14 57 ROAD  …         Unspecified    Unspecified  Unspecified

  School State   School Zip School Not Found   Latitude  Longitude  \
0  Unspecified  Unspecified                N  40.865682 -73.923501
```

```
1  Unspecified  Unspecified                N  40.775945 -73.915094
2  Unspecified  Unspecified                N  40.870325 -73.888525
3  Unspecified  Unspecified                N  40.835994 -73.828379
4  Unspecified  Unspecified                N  40.733060 -73.874170

                                  Location Request_Closing_Time
0   (40.86568153633767, -73.92350095571744)            55.250000
1  (40.775945312321085, -73.91509393898605)            86.266667
2  (40.870324522111424, -73.88852464418646)           291.516667
3   (40.83599404683083, -73.82837939584206)           465.233333
4  (40.733059618956815, -73.87416975810375)           207.033333

[5 rows x 40 columns]
```

[33]:
```
print("Old DataFrame Shape :",data.shape)
print("New DataFrame Shape : ",new_df.shape)
```

```
Old DataFrame Shape : (300698, 54)
New DataFrame Shape :  (300698, 40)
```

[34]:
```
rem=[]
for x in new_df.columns.tolist():
    if new_df[x].nunique()<=3:
        print(x+ " "*10+" : ",new_df[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']
```

[35]:
```
new_df.drop(rem,axis=1,inplace=True)
```

[36]:
```
new_df.shape
```

[36]: (300698, 26)

[37]:
```
#Remove columns that are not needed for our analysis
rem1=["Unique Key","Incident Address","Descriptor","Street Name","Cross Street␣
 ↪1","Cross Street 2","Due Date","Resolution Description","Resolution Action␣
 ↪Updated Date","Community Board","X Coordinate (State Plane)","Y Coordinate␣
 ↪(State Plane)","Park Borough","Latitude","Longitude","Location"]

new_df.drop(rem1,axis=1,inplace=True)
```

[38]:
```
new_df.head()
```

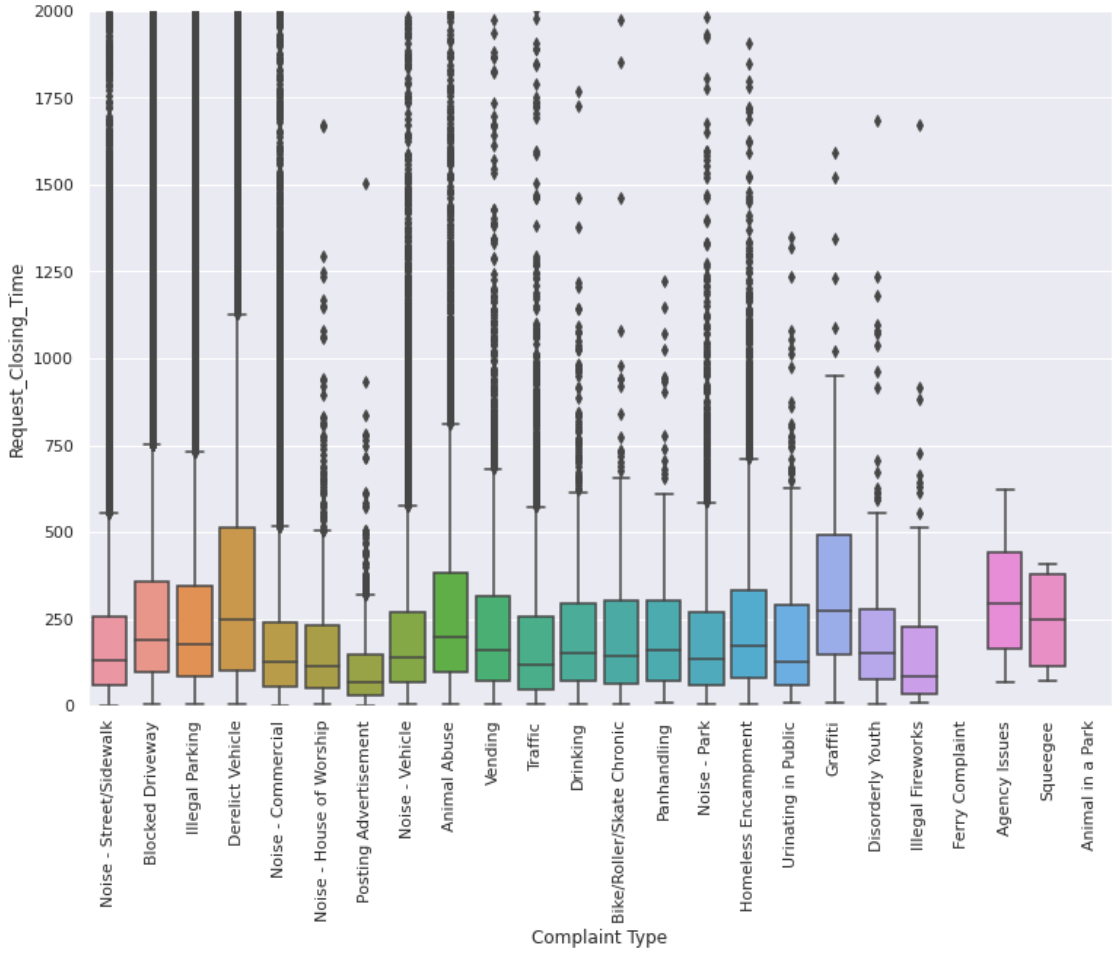
[38]:

| | Created Date | Closed Date | Complaint Type | Location Type | Incident Zip | Address Type | City | Status | Borough | Request_Closing_Time |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2015-12-31 23:59:45 | 2016-01-01 00:55:00 | Noise - Street/Sidewalk | Street/Sidewalk | 10034.0 | ADDRESS | NEW YORK | Closed | MANHATTAN | 55.250000 |
| 1 | 2015-12-31 23:59:44 | 2016-01-01 01:26:00 | Blocked Driveway | Street/Sidewalk | 11105.0 | ADDRESS | ASTORIA | Closed | QUEENS | 86.266667 |
| 2 | 2015-12-31 23:59:29 | 2016-01-01 04:51:00 | Blocked Driveway | Street/Sidewalk | 10458.0 | ADDRESS | BRONX | Closed | BRONX | 291.516667 |
| 3 | 2015-12-31 23:57:46 | 2016-01-01 07:43:00 | Illegal Parking | Street/Sidewalk | 10461.0 | ADDRESS | BRONX | Closed | BRONX | 465.233333 |
| 4 | 2015-12-31 23:56:58 | 2016-01-01 03:24:00 | Illegal Parking | Street/Sidewalk | 11373.0 | ADDRESS | ELMHURST | Closed | QUEENS | 207.033333 |

[ ]:

[39]:
```
g=sns.catplot(x="Complaint␣
 ↪Type",y="Request_Closing_Time",kind="box",data=new_df)
g.fig.set_figheight(8)
g.fig.set_figwidth(15)
plt.xticks(rotation=90)
plt.ylim((0,2000))
```

[39]: (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

[44]:
```
anova_df=pd.DataFrame()
anova_df["Request_Closing_Time"]=new_df["Request_Closing_Time"]
anova_df["Complaint"]=new_df["Complaint Type"]

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

[44]:

| | 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 |

[45]:
```
lm=ols("Request_Closing_Time~Complaint",data=anova_df).fit()
table=sm.stats.anova_lm(lm)
table
```

[45]:

| | 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 |

H0:Complaint Type and Location Type are independent

H1:Complaint Type and Location Type are related

[46]:
```
chi_sq=pd.DataFrame()
chi_sq["Location Type"]=new_df["Location Type"]
chi_sq["Complaint Type"]=new_df["Complaint Type"]

chi_sq.dropna(inplace=True)
```

[47]:
```
data_crosstab = pd.crosstab( chi_sq["Location Type"],chi_sq["Complaint Type"])
```

[48]:
```
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)
```

# 8 Conclusion

- almost 90% of resquest belongs to transport (street/sidewalk, Blocked driveway,Illegal Parking, derelict Vehicle , Road Traffic etc ).
- On an average complains are closed in an span of 150 to 300 hours
- BROOKLYN, QUEENS and BRONX has most complaints of Blocked Driveway.
- Complaint Type are Depentent on Location Type.
- Time taken for solving different complaint type are different

[ ]:
```

```