

In [1]:

```
#import all the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import optimize
import scipy.stats as stats
from math import sqrt
%matplotlib inline
```

In [2]:

```
#reading the file from the source
nyc=pd.read_csv('/Users/apple/Desktop/data folder/311_Service_Requests_from_2010_to_Present.csv',low_memory=False)
nyc
```

Out[2]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction	Road Ramp
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	...	NaN	NaN	NaN
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	...	NaN	NaN	NaN
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	...	NaN	NaN	NaN
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	...	NaN	NaN	NaN
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	...	NaN	NaN	NaN
...
300693	30281872	03/29/2015 12:33:41 AM	NaN	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	NaN	CRESCENT AVENUE	...	NaN	NaN	NaN
300694	30281230	03/29/2015 12:33:28 AM	03/29/2015 02:33:59 AM	NYPD	New York City Police Department	Blocked Driveway	Partial Access	Street/Sidewalk	11418.0	100-17 87 AVENUE	...	NaN	NaN	NaN
300695	30283424	03/29/2015 12:33:03 AM	03/29/2015 03:40:20 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	11206.0	162 THROOP AVENUE	...	NaN	NaN	NaN
300696	30280004	03/29/2015 12:33:02 AM	03/29/2015 04:38:35 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	10461.0	3151 EAST TREMONT AVENUE	...	NaN	NaN	NaN
300697	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	10036.0	251 WEST 48 STREET	...	NaN	NaN	NaN

300698 rows Ã 53 columns

In [3]:

```
#further exploration and pattern finding
nyc.head()
```

Out[3]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction	Road Ramp	Bridge Highway Segment	...
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	...	NaN	NaN	NaN	NaN	NaN
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	...	NaN	NaN	NaN	NaN	NaN
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	...	NaN	NaN	NaN	NaN	NaN
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	...	NaN	NaN	NaN	NaN	NaN
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	...	NaN	NaN	NaN	NaN	NaN

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction	Road Ramp	Bridge Highway Segment	G
		PM	3:24		Department											

5 rows Ã 53 columns
In [4]:

```
nyc.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 53 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Unique Key                            300698 non-null int64
1   Created Date                           300698 non-null object
2   Closed Date                            298534 non-null object
3   Agency                                 300698 non-null object
4   Agency Name                            300698 non-null object
5   Complaint Type                         300698 non-null object
6   Descriptor                             294784 non-null object
7   Location Type                          300567 non-null object
8   Incident Zip                           298083 non-null float64
9   Incident Address                       256288 non-null object
10  Street Name                            256288 non-null object
11  Cross Street 1                          251419 non-null object
12  Cross Street 2                          250919 non-null object
13  Intersection Street 1                   43858 non-null object
14  Intersection Street 2                   43362 non-null object
15  Address Type                            297883 non-null object
16  City                                    298084 non-null object
17  Landmark                                349 non-null object
18  Facility Type                           298527 non-null object
19  Status                                  300698 non-null object
20  Due Date                                300695 non-null object
21  Resolution Description                  300698 non-null object
22  Resolution Action Updated Date          298511 non-null object
23  Community Board                        300698 non-null object
24  Borough                                300698 non-null object
25  X Coordinate (State Plane)              297158 non-null float64
26  Y Coordinate (State Plane)              297158 non-null float64
27  Park Facility Name                      300698 non-null object
28  Park Borough                           300698 non-null object
29  School Name                             300698 non-null object
30  School Number                           300698 non-null object
31  School Region                           300697 non-null object
32  School Code                             300697 non-null object
33  School Phone Number                     300698 non-null object
34  School Address                          300698 non-null object
35  School City                             300698 non-null object
36  School State                            300698 non-null object
37  School Zip                              300697 non-null object
38  School Not Found                        300698 non-null object
39  School or Citywide Complaint            0 non-null float64
40  Vehicle Type                            0 non-null float64
41  Taxi Company Borough                    0 non-null float64
42  Taxi Pick Up Location                    0 non-null float64
43  Bridge Highway Name                     243 non-null object
44  Bridge Highway Direction                243 non-null object
45  Road Ramp                               213 non-null object
46  Bridge Highway Segment                  213 non-null object
47  Garage Lot Name                          0 non-null float64
48  Ferry Direction                          1 non-null object
49  Ferry Terminal Name                      2 non-null object
50  Latitude                                297158 non-null float64
51  Longitude                                297158 non-null float64
52  Location                                297158 non-null object
dtypes: float64(10), int64(1), object(42)
memory usage: 121.6+ MB
```

```
In [6]:

#dropping all the columns with no values
nyc311=nyc.drop(['Bridge Highway Name','Bridge Highway Direction','Road Ramp','Bridge Highway Segment',
'Garage Lot Name','Ferry Direction','Ferry Terminal Name','School State','School Zip',
'School Not Found','School or Citywide Complaint','Vehicle Type','Taxi Company Borough',
'Taxi Pick Up Location','School Name','School Number','School Region','School Code',
'School Phone Number','School Address','School City'],axis=1)

nyc311
```

Out[6]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Resolution Action Updated Date	Communit Boar
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	...	01-01-16 0:55	12 MANHATTAN
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	...	01-01-16 1:26	01 QUEENS

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Resolution Action Updated Date	Communit Boar
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	01-01-16 4:51	07 BRONX
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	01-01-16 7:43	10 BRONX
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	01-01-16 3:24	04 QUEENS
...
300693	30281872	03/29/2015 12:33:41 AM	NaN	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	NaN	CRESCENT AVENUE	NaN	0 Unspecified
300694	30281230	03/29/2015 12:33:28 AM	03/29/2015 02:33:59 AM	NYPD	New York City Police Department	Blocked Driveway	Partial Access	Street/Sidewalk	11418.0	100-17 87 AVENUE	03/29/2015 02:33:59 AM	09 QUEENS
300695	30283424	03/29/2015 12:33:03 AM	03/29/2015 03:40:20 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	11206.0	162 THROOP AVENUE	03/29/2015 03:40:20 AM	03 BROOKLYN
300696	30280004	03/29/2015 12:33:02 AM	03/29/2015 04:38:35 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	10461.0	3151 EAST TREMONT AVENUE	03/29/2015 04:38:35 AM	10 BRONX
300697	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	10036.0	251 WEST 48 STREET	03/29/2015 04:41:50 AM	05 MANHATTAN

300698 rows Æ 32 columns

In [7]:

```
#using fillna to fill all the NaN values with zero
nyc311.fillna(0)
nyc311
```

Out[7]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Resolution Action Updated Date	Communit Boar
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	01-01-16 0:55	12 MANHATTAN
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	01-01-16 1:26	01 QUEENS
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	01-01-16 4:51	07 BRONX
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	01-01-16 7:43	10 BRONX
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	01-01-16 3:24	04 QUEENS
...
300693	30281872	03/29/2015 12:33:41 AM	NaN	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	NaN	CRESCENT AVENUE	NaN	0 Unspecified
300694	30281230	03/29/2015 12:33:28 AM	03/29/2015 02:33:59 AM	NYPD	New York City Police Department	Blocked Driveway	Partial Access	Street/Sidewalk	11418.0	100-17 87 AVENUE	03/29/2015 02:33:59 AM	09 QUEENS
300695	30283424	03/29/2015 12:33:03 AM	03/29/2015 03:40:20 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	11206.0	162 THROOP AVENUE	03/29/2015 03:40:20 AM	03 BROOKLYN
300696	30280004	03/29/2015 12:33:02 AM	03/29/2015 04:38:35 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	10461.0	3151 EAST TREMONT AVENUE	03/29/2015 04:38:35 AM	10 BRONX
300697	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	10036.0	251 WEST 48 STREET	03/29/2015 04:41:50 AM	05 MANHATTAN

300698 rows Æ 32 columns

In [8]:

```
# Read or convert the columns 'Created Date' and 'Closed Date' to datetime datatype and create a new column 'Request_Closing_Time' as the time el
#Question -2
```

```
import datetime as dt
import time, datetime
```

In [9]:

```
# Convert "Closed Date" to datetime dtype
nyc311['Closed Date'] = pd.to_datetime(nyc311['Closed Date'])
nyc311['Closed Date'].dtype
```

Out[9]:

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

In [10]:

```
nyc311['Created Date'] = pd.to_datetime(nyc311['Created Date'])
nyc311['Created Date'].dtype
```

Out[10]:

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

In [11]:

```
# Create new column Request_Closing_Time
```

```
nyc311['Request_Closing_Time'] = nyc311['Closed Date'] - nyc311['Created Date']
```

```
df=nyc311['Request_Closing_Time']
df.head()
```

Out[11]:

```
0    00:55:15
1    01:26:16
2    04:51:31
3    07:45:14
4    03:27:02
Name: Request_Closing_Time, dtype: timedelta64[ns]
```

In [12]:

```
#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.
#Question -3
```

In [13]:

```
import math
# Function to convert TimeDelta in Hour
def ConvertTohour(timeDel):
    days = timeDel.days
    hours = round(timeDel.seconds/3600, 2)
    result = (days * 24) + hours
    #print(days)
    #print(hours)
    return result
#return round(pd.Timedelta(timeDel).seconds / 3600, 2)
```

In [14]:

```
# Apply this function to every row of column Request_Closing_Time
nyc311['Request_Closing_Hr'] = nyc311['Request_Closing_Time'].apply(ConvertTohour)
nyc311['Request_Closing_Hr'].head()
```

Out[14]:

```
0    0.92
1    1.44
2    4.86
3    7.75
4    3.45
Name: Request_Closing_Hr, dtype: float64
```

In [58]:

```
# Function to categorize hours - Less than 2 hours - Excellent, Between 2 to 4 hours - Moderate,
#Between 4 to 6 - Average, More than 6 hours - Bad
def Ratings(hr):
    if (math.isnan(hr)):
        return 'Unspecified'
    elif (hr < 2.0):
        return 'Excellent'
    elif (4.0 > hr >= 2.0):
        return 'Moderate'
    elif (6.0 > hr >= 4.0):
        return 'Average'
    else:
        return 'Bad'
```

In [59]:

```
# Apply function on column Request_Closing_In_Hr
```

```
nyc311['Request_Closing_time_Category'] = nyc311['Request_Closing_Hr'].apply(Ratings)
```

```
nyc311['Request_Closing_time_Category'].head()
```

```
Out[59]:
```

```
0    Excellent
1    Excellent
2      Average
3         Bad
4    Moderate
```

```
Name: Request_Closing_time_Category, dtype: object
```

```
In [17]:
```

```
nyc311.columns
```

```
Out[17]:
```

```
Index(['Unique Key', 'Created Date', 'Closed Date', 'Agency', 'Agency Name',
      'Complaint Type', 'Descriptor', 'Location Type', 'Incident Zip',
      'Incident Address', 'Street Name', 'Cross Street 1', 'Cross Street 2',
      'Intersection Street 1', 'Intersection Street 2', 'Address Type',
      'City', 'Landmark', 'Facility Type', 'Status', 'Due Date',
      'Resolution Description', 'Resolution Action Updated Date',
      'Community Board', 'Borough', 'X Coordinate (State Plane)',
      'Y Coordinate (State Plane)', 'Park Facility Name', 'Park Borough',
      'Latitude', 'Longitude', 'Location', 'Request_Closing_Time',
      'Request_Closing_Hr', 'Request_Closing_time_Category'],
      dtype='object')
```

```
In [18]:
```

```
#Pattern 1
```

```
nyc311['Request_Closing_time_Category'].value_counts()
```

```
# Create Bar plot for Request_Closing_Time_Category which shows on average close time is Fast
```

```
nyc311['Request_Closing_time_Category'].value_counts().plot(kind="bar", color=list('rgbkymc'), alpha=0.7, figsize=(15,7))
plt.show()
```

```
In [19]:
```

```
#Pattern 2
```

```
#mostly complaints have been taken care of and closed
```

```
nyc311['Status'].value_counts().plot(kind='barh',color=list('rgbkymc'), alpha=0.7, figsize=(15,7))
plt.show()
```

```
In [20]:
```

```
# Pattern 3
```

```
# Check count in each complain type - sorted decreasing order
```

```
nyc311['Complaint Type'].value_counts().head()
```

```
Out[20]:
```

```
Blocked Driveway      77044
Illegal Parking       75361
Noise - Street/Sidewalk 48612
Noise - Commercial    35577
Derelict Vehicle      17718
Name: Complaint Type, dtype: int64
```

```
In [21]:
```

```
# Create Bar plot for complain type to check frequency .Blocked driveway is the popular complaint
```

```
nyc311['Complaint Type'].value_counts().plot(kind="barh", color=list('rgbkymc'), alpha=0.7, figsize=(15,10))
plt.show()
```

```
In [22]:
```

```
#Pattern 4
```

```
#Manhattan has the highest number of complaints
```

```
pd.value_counts(nyc311['Park Borough'].values)
```

```
Out[22]:
```

```
BROOKLYN      98307
QUEENS        80641
MANHATTAN     66131
BRONX         40702
STATEN ISLAND 12343
Unspecified    2574
dtype: int64
```

```
In [23]:
```

```
nyc311['Park Borough'].value_counts().plot(kind="barh", color=list('rgbkymc'), alpha=0.7, figsize=(15,7))
plt.show()
```

```
In [24]:
```

```
nyc311.columns
```

```
Out[24]:
```

```
Index(['Unique Key', 'Created Date', 'Closed Date', 'Agency', 'Agency Name',
      'Complaint Type', 'Descriptor', 'Location Type', 'Incident Zip',
      'Incident Address', 'Street Name', 'Cross Street 1', 'Cross Street 2',
```

```
'Intersection Street 1', 'Intersection Street 2', 'Address Type',
'City', 'Landmark', 'Facility Type', 'Status', 'Due Date',
'Resolution Description', 'Resolution Action Updated Date',
'Community Board', 'Borough', 'X Coordinate (State Plane)',
'Y Coordinate (State Plane)', 'Park Facility Name', 'Park Borough',
'Latitude', 'Longitude', 'Location', 'Request_Closing_Time',
'Request_Closing_Hr', 'Request_Closing_time_Category'],
dtype='object')
In [25]:
nyc311['City'].head()
Out[25]:
0    NEW YORK
1    ASTORIA
2    BRONX
3    BRONX
4    ELMHURST
Name: City, dtype: object
In [26]:
nyc311_grouped = nyc311.groupby(['City', 'Complaint Type'])
In [27]:
# get average of this grouped dataframe, and get Request_Closing_Time column from there
nyc311_mean = nyc311_grouped.mean()['Request_Closing_Hr']
nyc311_mean.sum()
Out[27]:
3894.907028001004
In [28]:
# Group by City(location) first and then Complain Type and showing average of Request Closing in Hour
nyc311_grouped = nyc311.groupby(['City','Complaint Type']).agg({'Request_Closing_Hr': 'mean'})
nyc311_grouped.head()
Out[28]:


|         |                  | Request_Closing_Hr |
|---------|------------------|--------------------|
| City    | Complaint Type   |                    |
| ARVERNE | Animal Abuse     | 2.153158           |
|         | Blocked Driveway | 2.526000           |
|         | Derelict Vehicle | 2.968889           |
|         | Disorderly Youth | 3.595000           |
|         | Drinking         | 0.240000           |


In [29]:
nyc311_grouped.info()
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 764 entries, ('ARVERNE', 'Animal Abuse') to ('Woodside', 'Noise - Street/Sidewalk')
Data columns (total 1 columns):
#   Column              Non-Null Count  Dtype
---  ---
0   Request_Closing_Hr  764 non-null   float64
dtypes: float64(1)
memory usage: 8.2+ KB
In [30]:
# Question 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
# Whether the average response time across complaint types is similar or not (overall)
# Are the type of complaint or service requested and location related?
In [31]:
# Null hypotheses: Groups means are equal (no variation in means of groups)
# Average response time across complaint types is similar
# Alternative hypotheses: At least, one group mean is different from other groups
# Average response time across complaint types is NOT the same
In [32]:
#FIRST PART
In [33]:
#Preparing Sample data to test for Anova
In [34]:
nyc311['Complaint Type'].value_counts()
Out[34]:
Blocked Driveway      77044
Illegal Parking       75361
Noise - Street/Sidewalk 48612
```

```
Noise - Commercial      35577
Derelect Vehicle        17718
Noise - Vehicle          17083
Animal Abuse            7778
Traffic                 4498
Homeless Encampment     4416
Noise - Park            4042
Vending                 3802
Drinking                1280
Noise - House of Worship 931
Posting Advertisement    650
Urinating in Public      592
Bike/Roller/Skate Chronic 427
Panhandling             307
Disorderly Youth        286
Illegal Fireworks        168
Graffiti               113
Agency Issues           6
Squeegee                4
Ferry Complaint          2
Animal in a Park         1
Name: Complaint Type, dtype: int64
In [35]:
```

```
sample5_complaints=nyc311['Complaint Type'].value_counts()[:5]
sample5_complaints
```

```
Out[35]:
Blocked Driveway      77044
Illegal Parking        75361
Noise - Street/Sidewalk 48612
Noise - Commercial     35577
Derelect Vehicle       17718
Name: Complaint Type, dtype: int64
```

```
In [36]:
sample5_complaint_names = sample5_complaints.index
sample5_complaint_names
```

```
Out[36]:
Index(['Blocked Driveway', 'Illegal Parking', 'Noise - Street/Sidewalk',
      'Noise - Commercial', 'Derelect Vehicle'],
      dtype='object')
```

```
In [37]:
test = nyc311.loc[nyc311['Complaint Type'].isin(sample5_complaint_names), ['Complaint Type', 'Request_Closing_Hr']]
test.head()
```

Out[37]:

	Complaint Type	Request_Closing_Hr
0	Noise - Street/Sidewalk	0.92
1	Blocked Driveway	1.44
2	Blocked Driveway	4.86
3	Illegal Parking	7.75
4	Illegal Parking	3.45

```
In [38]:
test.shape
```

```
Out[38]:
(254312, 2)
```

```
In [39]:
test.dropna(how='any', inplace=True)
test.isnull().sum()
```

```
Out[39]:
Complaint Type      0
Request_Closing_Hr  0
dtype: int64
```

```
In [40]:
Sample_1 = test[test['Complaint Type'] == sample5_complaint_names[0]].Request_Closing_Hr
Sample_1.head()
```

```
Out[40]:
1      1.44
2      4.86
7      1.80
9      1.38
10     7.80
Name: Request_Closing_Hr, dtype: float64
```

```
In [41]:
```

```
Sample_2 = test[test['Complaint Type'] == sample5_complaint_names[1]].Request_Closing_Hr
Sample_2.head()
Out[41]:
```

```
3    7.75
4    3.45
5    1.89
6    1.96
8    8.55
```

```
Name: Request_Closing_Hr, dtype: float64
```

```
In [42]:
```

```
Sample_3 = test[test['Complaint Type'] == sample5_complaint_names[2]].Request_Closing_Hr
Sample_3.head()
```

```
Out[42]:
```

```
0    0.92
12   2.48
19   0.78
38   0.49
54   1.50
```

```
Name: Request_Closing_Hr, dtype: float64
```

```
In [43]:
```

```
Sample_4 = test[test['Complaint Type'] == sample5_complaint_names[3]].Request_Closing_Hr
Sample_4.head()
```

```
Out[43]:
```

```
17   0.85
18   2.93
22   1.26
29   2.50
30   1.99
```

```
Name: Request_Closing_Hr, dtype: float64
```

```
In [44]:
```

```
Sample_5 = test[test['Complaint Type'] == sample5_complaint_names[4]].Request_Closing_Hr
Sample_5.head()
```

```
Out[44]:
```

```
14   10.49
151   3.95
255   1.36
256   4.13
295   0.75
```

```
Name: Request_Closing_Hr, dtype: float64
```

```
In [45]:
```

```
print(Sample_1.isnull().sum())
print(Sample_2.isnull().sum())
print(Sample_3.isnull().sum())
print(Sample_4.isnull().sum())
print(Sample_5.isnull().sum())
```

```
0
0
0
0
0
```

```
In [46]:
```

```
stats.f_oneway(Sample_1,Sample_2,Sample_3,Sample_4,Sample_5)
```

```
Out[46]:
```

```
F_onewayResult(statistic=1799.598683238952, pvalue=0.0)
```

```
In [47]:
```

```
#CONCLUSION : SINCE P-VALUE IS (P<0.05). WE CAN REJECT THE NULL HYPOTHESIS AND SAY THAT THE RESPONSE TIME
#FOR COMPLAINTS IS NOT THE SAME OVERALL
```

```
In [48]:
```

```
#SECOND PART
# Are the type of complaint or service requested and location related?
```

```
In [49]:
```

```
#Null Hypothesis:Complaint type or service request and location are related
#Alternative Hypothesis: Complaint type or service request and location are NOT related
```

```
In [50]:
```

```
#Preparing data to perform Chi-square test
```

```
In [51]:
```

```
location = nyc311['City'].value_counts()[:5]
location
```



```
Out[51]:

BROOKLYN      98307
NEW YORK      65994
BRONX         40702
STATEN ISLAND 12343
JAMAICA       7296
Name: City, dtype: int64
In [52]:

location_names = location.index
location_names

Out[52]:

Index(['BROOKLYN', 'NEW YORK', 'BRONX', 'STATEN ISLAND', 'JAMAICA'], dtype='object')

In [53]:

location_test = nyc311.loc[(nyc311['Complaint Type'].isin(sample5_complaint_names)) & (nyc311['City'].isin(location_names)), ['Complaint Type', 'City']]
location_test.head()

Out[53]:

```

	Complaint Type	City
0	Noise - Street/Sidewalk	NEW YORK
2	Blocked Driveway	BRONX
3	Illegal Parking	BRONX
5	Illegal Parking	BROOKLYN
6	Illegal Parking	NEW YORK

```

In [54]:

pd.crosstab(location_test['Complaint Type'], location_test['City'], margins=True)

Out[54]:

```

City	BRONX	BROOKLYN	JAMAICA	NEW YORK	STATEN ISLAND	All
Complaint Type						
Blocked Driveway	12755	28148	2818	2072	2142	47935
Derelict Vehicle	1953	5181	954	537	1766	10391
Illegal Parking	7859	27462	1421	12128	4886	53756
Noise - Commercial	2434	11463	429	14550	678	29554
Noise - Street/Sidewalk	8892	13356	339	20433	819	43839
All	33893	85610	5961	49720	10291	185475

```

In [55]:

ch2, p_value, df, exp_frq = stats.chi2_contingency(pd.crosstab(location_test['Complaint Type'],location_test['City']))

In [56]:

print('ch2:',ch2,'p_value:',p_value)

ch2: 40522.79928349593 p_value: 0.0

In [ ]:

#CONCLUSION : SINCE P-VALUE IS (P<0.05). WE CAN REJECT THE NULL HYPOTHESIS AND SAY THAT
#Complaint type or service request and location are NOT related
```