SRD_NYC_311

June 14, 2022

1 Customer Service Requests Analysis

1.1 Step 1:

1.1.1 Identify Problem:

NYC 311's mission is to provide the public with quick and easy access to all New York City government services and information while offering the best customer service. Each day, NYC311 receives thousands of requests related to several hundred types of non-emergency services, including noise complaints, plumbing issues, and illegally parked cars. These requests are received by NYC311 and forwarded to the relevant agencies such as the police, buildings, or transportation. The agency responds to the request, addresses it, and then closes it.

Problem Objective :

Perform a service request data analysis of New York City 311 calls. You will focus on the data wrangling techniques to understand the pattern in the data and also visualize the major complaint types.

Domain: Customer Service Analysis Tasks to be performed: ##### (Perform a service request data analysis of New York City 311 calls)

Import a 311 NYC service request.

Read or convert the columns 'Created Date' and Closed Date' to datetime datatype and create a new column 'Request_Closing_Time' as the time elapsed between request creation and request closing.

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.

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

Perform a statistical test for the following:

Please note: For the below statements you need to state the Null and Alternate and then provide a statistical test to accept or reject the Null Hypothesis along with the corresponding 'p-value'.

Whether the average response time across complaint types is similar or not (overall) Are the type of complaint or service requested and location related?

1.2 Step 2: Data Acquisition

[1]:
```
#Importing Required Libraries for Data Analysis
import numpy as np
import pandas as pd
```

[2]:
```
#Reading the Service Request Data csv
SRD_raw = pd.read_csv("C:
 ↪\\Users\\grkum\\Downloads\\Data-Science-with-Python-Project-2--master\\Data␣
 ↪Science with Python Two\\311_Service_Requests_from_2010_to_Present.csv")
```

```
C:\ProgramData\Anaconda3\lib\site-
packages\IPython\core\interactiveshell.py:3444: DtypeWarning: Columns (48,49)
have mixed types.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

[3]:
```
#Checking the first 5 entries of Data
SRD_raw.head()
```

[3]:

	Unique Key	Created Date	Closed Date	Agency	\
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	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 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]
```

[4]:
```
#Identify features of the dataset(Columns)
SRD_raw.columns
```

[4]:
```
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',
       'School Name', 'School Number', 'School Region', 'School Code',
       'School Phone Number', 'School Address', 'School City', 'School State',
       'School Zip', 'School Not Found', 'School or Citywide Complaint',
       'Vehicle Type', 'Taxi Company Borough', 'Taxi Pick Up Location',
       'Bridge Highway Name', 'Bridge Highway Direction', 'Road Ramp',
       'Bridge Highway Segment', 'Garage Lot Name', 'Ferry Direction',
       'Ferry Terminal Name', 'Latitude', 'Longitude', 'Location'],
      dtype='object')
```

[5]:
```
#View the data(observations),shape,info,describe to get more insights on the␣
↪data.
SRD_raw.shape
```

[5]: (300698, 53)

[6]:
```
SRD_raw.describe()
```

[6]:
```
         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
```

[7]:
```
SRD_raw.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
```

[8]:
```
# Using the option to view all columns of the dataset and view the data for any␣
↪6 random entries.

pd.set_option('display.max_columns',None)
SRD_raw.sample(6)
```

[8]:

	Unique Key	Created Date	Closed Date	Agency	\
31093	32094558	11/30/2015 12:25:43 PM	11/30/2015 06:05:33 PM	NYPD	
282574	30429768	04/18/2015 08:14:48 PM	04/19/2015 12:12:18 AM	NYPD	
79663	31754823	10/14/2015 08:52:32 PM	10/15/2015 02:02:23 AM	NYPD	
182011	31094038	07/16/2015 09:33:53 AM	07/16/2015 05:12:21 PM	NYPD	
129800	31432827	08/31/2015 02:55:00 PM	09-01-15 6:54	NYPD	
16399	32201934	12/14/2015 08:27:56 AM	12/14/2015 08:35:48 AM	NYPD	

	Agency Name	Complaint Type	\
31093	New York City Police Department	Derelict Vehicle	
282574	New York City Police Department	Noise - Street/Sidewalk	
79663	New York City Police Department	Blocked Driveway	
182011	New York City Police Department	Derelict Vehicle	
129800	New York City Police Department	Illegal Parking	
16399	New York City Police Department	Blocked Driveway	

	Descriptor	Location Type	Incident Zip	\
31093	With License Plate	Street/Sidewalk	10453.0	
282574	Loud Music/Party	Street/Sidewalk	10452.0	
79663	No Access	Street/Sidewalk	11226.0	
182011	With License Plate	Street/Sidewalk	11412.0	
129800	Posted Parking Sign Violation	Street/Sidewalk	10466.0	
16399	No Access	Street/Sidewalk	11218.0	

	Incident Address	Street Name	Cross Street 1	\
31093	1783 UNDERCLIFF AVENUE	UNDERCLIFF AVENUE	WEST 176 STREET	
282574	1476 TOWNSEND AVENUE	TOWNSEND AVENUE	EAST 171 STREET	
79663	250 MARTENSE STREET	MARTENSE STREET	ROGERS AVENUE	
182011	205-13 114 DRIVE	114 DRIVE	205 STREET	

```
129800    3900 ROMBOUTS AVENUE    ROMBOUTS AVENUE         DEAD END
16399       206 ALBEMARLE ROAD     ALBEMARLE ROAD    EAST 2 STREET

                Cross Street 2 Intersection Street 1 Intersection Street 2  \
31093           SEDGWICK AVENUE                  NaN                   NaN
282574          EAST 172 STREET                  NaN                   NaN
79663           NOSTRAND AVENUE                  NaN                   NaN
182011  FRANCIS LEWIS BOULEVARD                  NaN                   NaN
129800              DARK STREET                  NaN                   NaN
16399             EAST 3 STREET                  NaN                   NaN

       Address Type          City Landmark Facility Type  Status  \
31093       ADDRESS         BRONX      NaN      Precinct  Closed
282574      ADDRESS         BRONX      NaN      Precinct  Closed
79663       ADDRESS      BROOKLYN      NaN      Precinct  Closed
182011      ADDRESS  SAINT ALBANS      NaN      Precinct  Closed
129800      ADDRESS         BRONX      NaN      Precinct  Closed
16399       ADDRESS      BROOKLYN      NaN      Precinct  Closed

                     Due Date  \
31093   11/30/2015 08:25:43 PM
282574  04/19/2015 04:14:48 AM
79663   10/15/2015 04:52:32 AM
182011  07/16/2015 05:33:53 PM
129800  08/31/2015 10:55:00 PM
16399   12/14/2015 04:27:56 PM

                                  Resolution Description  \
31093   The Police Department responded to the complai…
282574  The Police Department responded to the complai…
79663   The Police Department responded and upon arriv…
182011  The Police Department reviewed your complaint …
129800  The Police Department responded to the complai…
16399   Your request can not be processed at this time…

       Resolution Action Updated Date Community Board   Borough  \
31093          11/30/2015 06:05:33 PM        05 BRONX     BRONX
282574         04/19/2015 12:12:18 AM        04 BRONX     BRONX
79663          10/15/2015 02:02:23 AM     17 BROOKLYN  BROOKLYN
182011         07/16/2015 05:12:21 PM       12 QUEENS    QUEENS
129800                   09-01-15 6:54        12 BRONX     BRONX
16399          12/14/2015 08:35:48 AM     12 BROOKLYN  BROOKLYN

        X Coordinate (State Plane)  Y Coordinate (State Plane)  \
31093                    1006471.0                    249686.0
282574                   1007597.0                    245868.0
79663                     997900.0                    176686.0
```

```
182011                  1053774.0                   194380.0
129800                  1031152.0                   263476.0
16399                    990244.0                   174364.0

       Park Facility Name Park Borough  School Name School Number  \
31093         Unspecified        BRONX  Unspecified   Unspecified
282574        Unspecified        BRONX  Unspecified   Unspecified
79663         Unspecified     BROOKLYN  Unspecified   Unspecified
182011        Unspecified       QUEENS  Unspecified   Unspecified
129800        Unspecified        BRONX  Unspecified   Unspecified
16399         Unspecified     BROOKLYN  Unspecified   Unspecified

       School Region  School Code School Phone Number School Address  \
31093    Unspecified  Unspecified         Unspecified    Unspecified
282574   Unspecified  Unspecified         Unspecified    Unspecified
79663    Unspecified  Unspecified         Unspecified    Unspecified
182011   Unspecified  Unspecified         Unspecified    Unspecified
129800   Unspecified  Unspecified         Unspecified    Unspecified
16399    Unspecified  Unspecified         Unspecified    Unspecified

        School City School State   School Zip School Not Found  \
31093   Unspecified  Unspecified  Unspecified                N
282574  Unspecified  Unspecified  Unspecified                N
79663   Unspecified  Unspecified  Unspecified                N
182011  Unspecified  Unspecified  Unspecified                N
129800  Unspecified  Unspecified  Unspecified                N
16399   Unspecified  Unspecified  Unspecified                N

        School or Citywide Complaint  Vehicle Type  Taxi Company Borough  \
31093                            NaN           NaN                   NaN
282574                           NaN           NaN                   NaN
79663                            NaN           NaN                   NaN
182011                           NaN           NaN                   NaN
129800                           NaN           NaN                   NaN
16399                            NaN           NaN                   NaN

        Taxi Pick Up Location Bridge Highway Name Bridge Highway Direction  \
31093                     NaN                 NaN                       NaN
282574                    NaN                 NaN                       NaN
79663                     NaN                 NaN                       NaN
182011                    NaN                 NaN                       NaN
129800                    NaN                 NaN                       NaN
16399                     NaN                 NaN                       NaN

       Road Ramp Bridge Highway Segment  Garage Lot Name Ferry Direction  \
31093        NaN                   NaN             NaN             NaN
282574       NaN                   NaN             NaN             NaN
```

```
79663        NaN                    NaN              NaN              NaN
182011       NaN                    NaN              NaN              NaN
129800       NaN                    NaN              NaN              NaN
16399        NaN                    NaN              NaN              NaN

       Ferry Terminal Name   Latitude  Longitude  \
31093                  NaN  40.851977 -73.919678
282574                 NaN  40.841495 -73.915621
79663                  NaN  40.651628 -73.950808
182011                 NaN  40.699932 -73.749265
129800                 NaN  40.889730 -73.830367
16399                  NaN  40.645264 -73.978401

                                        Location
31093    (40.85197739088619, -73.91967791239406)
282574   (40.841495203820855, -73.9156210646667)
79663   (40.651628406909204, -73.95080761949374)
182011   (40.69993185824646, -73.74926494425105)
129800   (40.88973003766137, -73.83036731666976)
16399   (40.645263511616335, -73.97840066016514)
```

1.2.1 Step 3: Data Wrangling

From above step we understood each feature(column) one by one. If the feature contains all Nun or all same entries or maybe, all different entries, then we will omit such columns, since these columns do not contain information that is statistically meaningful or can give us any trend.

For instance, the 'Unique Key' column has a different number for each entry whereas columns like 'Descriptor', 'Complaint Type' have a different group of lists from which we can predict which kind of complaint occurs more often (e.g frequency distribution). So we can improve our future prediction and even can take some precautions (if permissible).

[9]:
```
#Converting the columns as an array and then convert to a list using the method␣
 ↪tolist() method
column_names=SRD_raw.columns.values.tolist()

#Observing Frequency of each feature
for SRD in range(len(column_names)):
   name = column_names[SRD]
   print(SRD_raw[name].value_counts(),"\n--------------------------\n")
```

```
32310363    1
30964902    1
30963768    1
30961544    1
30964777    1
           ..
31611925    1
```

```
31615926    1
31612695    1
31617117    1
30281825    1
Name: Unique Key, Length: 300698, dtype: int64
--------------------------

07-11-15 23:04            9
11-06-15 23:34            9
06-06-15 22:23            9
10-09-15 23:56            8
11-01-15 22:12            8
                         ..
09/22/2015 05:52:17 PM    1
09/22/2015 05:50:43 PM    1
09/22/2015 05:49:55 PM    1
09/22/2015 05:49:47 PM    1
03/29/2015 12:33:01 AM    1
Name: Created Date, Length: 259493, dtype: int64
--------------------------

11-08-15 7:34             24
10-11-15 7:03             22
12-08-15 7:44             18
05-10-15 7:01             18
12-07-15 23:17            17
                          ..
09/21/2015 11:03:55 AM     1
09/21/2015 08:52:27 AM     1
09/21/2015 09:13:15 AM     1
09/21/2015 08:26:57 AM     1
03/29/2015 04:41:50 AM     1
Name: Closed Date, Length: 237165, dtype: int64
--------------------------

NYPD    300698
Name: Agency, dtype: int64
--------------------------

New York City Police Department    300690
Internal Affairs Bureau                 6
NYPD                                    2
Name: Agency Name, dtype: int64
--------------------------

Blocked Driveway             77044
Illegal Parking              75361
Noise - Street/Sidewalk      48612
```

```
Noise - Commercial           35577
Derelict 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
--------------------------

Loud Music/Party                 61430
No Access                        56976
Posted Parking Sign Violation    22440
Loud Talking                     21584
Partial Access                   20068
With License Plate               17718
Blocked Hydrant                  16081
Commercial Overnight Parking     12189
Car/Truck Music                  11273
Blocked Sidewalk                 11121
Double Parked Blocking Traffic    5731
Double Parked Blocking Vehicle    4211
Engine Idling                     4189
Banging/Pounding                  4165
Neglected                         3787
Car/Truck Horn                    3511
Congestion/Gridlock               2761
In Prohibited Area                2025
Other (complaint details)         1969
Unlicensed                        1777
Overnight Commercial Storage      1757
Unauthorized Bus Layover          1367
Truck Route Violation             1014
In Public                          932
```

```
Tortured                            854
Vehicle                             590
Chained                             535
Detached Trailer                    464
No Shelter                          382
Chronic Stoplight Violation         280
Underage - Licensed Est             271
Chronic Speeding                    268
In Car                              251
Playing in Unsuitable Place         245
Drag Racing                         175
Loud Television                      93
Police Report Requested              90
After Hours - Licensed Est           77
Building                             60
Nuisance/Truant                      41
Police Report Not Requested          23
Language Access Complaint             6
Homeless Issue                        1
Disruptive Passenger                  1
Animal Waste                          1
Name: Descriptor, dtype: int64
--------------------------

Street/Sidewalk               249299
Store/Commercial               20381
Club/Bar/Restaurant            17360
Residential Building/House      6960
Park/Playground                 4773
House of Worship                 929
Residential Building             227
Highway                          215
Parking Lot                      117
House and Store                   93
Vacant Lot                        77
Commercial                        62
Roadway Tunnel                    35
Subway Station                    34
Bridge                             2
Terminal                           1
Ferry                              1
Park                               1
Name: Location Type, dtype: int64
--------------------------

11385.0    5167
11368.0    4298
11211.0    4225
```

```
11234.0    4150
11206.0    3781
           …
10153.0       1
11242.0       1
11371.0       1
11451.0       1
11241.0       1
Name: Incident Zip, Length: 201, dtype: int64
--------------------------

1207 BEACH AVENUE          904
78-15 PARSONS BOULEVARD    505
89 MOORE STREET            480
177 LAREDO AVENUE          311
2117 3 AVENUE              295
                          …
131 NORTH    6 STREET        1
233W WEST 115 STREET         1
235 MONTROSE AVENUE          1
250 EAST 25 STREET           1
100-17 87 AVENUE             1
Name: Incident Address, Length: 107652, dtype: int64
--------------------------

BROADWAY            3237
3 AVENUE            1241
SHERMAN AVENUE      1156
BEACH AVENUE        1109
BEDFORD AVENUE       979
                    …
84TH DRIVE             1
BONNER PLACE           1
CEDARCROFT PLACE       1
GRAMERCY PARK          1
COOPER AVE             1
Name: Street Name, Length: 7320, dtype: int64
--------------------------

BROADWAY            4338
BEND                4129
3 AVENUE            3112
5 AVENUE            3035
AMSTERDAM AVENUE    2651
                    …
BAY 11 STREET          1
HANK PLACE             1
PYRAMID COURT          1
```

```
BOWDOIN STREET         1
EAST 186               1
Name: Cross Street 1, Length: 5982, dtype: int64
--------------------------

BEND                  4391
BROADWAY              3784
8 AVENUE              2766
DEAD END              2144
7 AVENUE              2140
                      …
CONNOR STREET            1
SANDY LANE               1
MC LAUGHLIN AVENUE       1
TIEMAN AVENUE            1
GRAMERCY PARK            1
Name: Cross Street 2, Length: 5823, dtype: int64
--------------------------

BROADWAY            672
170 STREET          441
44 STREET           355
6 AVENUE            348
85 STREET           237
                   …
ELGAR PLACE           1
MANHATTAN STREET      1
TENBROCK AVENUE       1
PETER STREET          1
east 186              1
Name: Intersection Street 1, Length: 4413, dtype: int64
--------------------------

BROADWAY            1358
6 AVENUE             715
2 AVENUE             617
5 AVENUE             551
3 AVENUE             487
                    …
ST PAULS PLACE         1
HAUGHWOUT AVENUE       1
MADOC AVENUE           1
WEST  103 STREET       1
LAWTON STREET          1
Name: Intersection Street 2, Length: 4172, dtype: int64
--------------------------

ADDRESS         238644
```

```
INTERSECTION     43366
BLOCKFACE        12014
LATLONG           3509
PLACENAME          350
Name: Address Type, dtype: int64
--------------------------

BROOKLYN               98307
NEW YORK               65994
BRONX                  40702
STATEN ISLAND          12343
JAMAICA                 7296
ASTORIA                 6330
FLUSHING                5971
RIDGEWOOD               5163
CORONA                  4295
WOODSIDE                3544
SOUTH RICHMOND HILL     2774
OZONE PARK              2755
EAST ELMHURST           2734
ELMHURST                2673
WOODHAVEN               2464
MASPETH                 2462
LONG ISLAND CITY        2437
SOUTH OZONE PARK        2173
RICHMOND HILL           1904
FRESH MEADOWS           1899
QUEENS VILLAGE          1814
MIDDLE VILLAGE          1765
JACKSON HEIGHTS         1689
FOREST HILLS            1688
REGO PARK               1486
BAYSIDE                 1221
COLLEGE POINT           1220
FAR ROCKAWAY            1179
WHITESTONE              1098
HOLLIS                  1012
HOWARD BEACH             931
ROSEDALE                 922
SPRINGFIELD GARDENS      883
SAINT ALBANS             834
KEW GARDENS              771
ROCKAWAY PARK            745
SUNNYSIDE                723
Astoria                  717
LITTLE NECK              559
OAKLAND GARDENS          551
CAMBRIA HEIGHTS          477
```

```
BELLEROSE                375
GLEN OAKS                306
ARVERNE                  220
FLORAL PARK              152
Long Island City         134
Woodside                 120
NEW HYDE PARK             98
CENTRAL PARK              97
QUEENS                    32
BREEZY POINT              30
East Elmhurst             14
Howard Beach               1
Name: City, dtype: int64
--------------------------

CENTRAL PARK               67
PROSPECT PARK              22
WASHINGTON SQUARE PARK     16
SUNSET PARK                13
UNION SQUARE PARK          13
                           ..
KOLBERT PARK                1
SEWARD PARK                 1
WOODHULL MEDICAL CENTER     1
ST JOHN THE DIVINE          1
INTREPID MUSEUM             1
Name: Landmark, Length: 116, dtype: int64
--------------------------

Precinct    298527
Name: Facility Type, dtype: int64
--------------------------

Closed      298471
Open          1439
Assigned       786
Draft            2
Name: Status, dtype: int64
--------------------------

11-07-15 7:34             9
06-07-15 6:23             9
07-12-15 7:04             9
11-02-15 6:12             8
05-03-15 9:32             8
                         ..
09/22/2015 09:39:41 PM    1
09/22/2015 09:39:38 PM    1
```

```
09/22/2015 09:38:34 PM    1
09/22/2015 09:37:02 PM    1
03/29/2015 08:33:01 AM    1
Name: Due Date, Length: 259851, dtype: int64
--------------------------

The Police Department responded to the complaint and with the information
available observed no evidence of the violation at that time.
90490
The Police Department responded to the complaint and took action to fix the
condition.
61624
The Police Department responded and upon arrival those responsible for the
condition were gone.
58031
The Police Department responded to the complaint and determined that police
action was not necessary.
38211
The Police Department issued a summons in response to the complaint.
28246
The Police Department reviewed your complaint and provided additional
information below.
13821
Your request can not be processed at this time because of insufficient contact
information. Please create a new Service Request on NYC.gov and provide more
detailed contact information.
4310
Your complaint has been forwarded to the New York Police Department for a non-
emergency response. 311 will have additional information in 8 hours. Please note
your service request number for future reference.
1916
This complaint does not fall under the Police Department's jurisdiction.
1797
The Police Department responded to the complaint but officers were unable to
gain entry into the premises.
1211
The Police Department responded to the complaint and a report was prepared.
675
Your complaint has been forwarded to the New York Police Department for a non-
emergency response. If the police determine the vehicle is illegally parked,
they will ticket the vehicle and then you may either contact a private towing
company to remove the vehicle or ask your local precinct to contact 'rotation
tow'. Any fees charged for towing will have to be paid by the vehicle owner. 311
will have additional information in 8 hours. Please note your service request
number for future reference.      232
The Police Department made an arrest in response to the complaint.
124
The New York City Police Department received your comments and forwarded them to
```

```
the appropriate unit for resolution. You may follow up by calling (646) 610-6952
after 60 days from submitting your agency issue.
6
Your complaint has been received by the Police Department and it has been
determined that a long-term investigation may be necessary.  Additional
information will be available at the conclusion of the investigation.
1
The Department of Transportation contacted the customer and resolved the Service
Request or provided the information requested.
1
The Department of Transportation requires 30 days to respond to this type of
complaint.  Please note your Service Request number for future reference.
1
The condition was determined to be an issue appropriate for handling by an
alternate entity.  The Department of Parks and Recreation has notified the
appropriate resource.
1
Name: Resolution Description, dtype: int64
---------------------------

11-08-15 7:34             24
10-11-15 7:03             22
05-10-15 7:01             18
12-08-15 7:44             18
12-07-15 23:17            17
                          ..
09/22/2015 12:43:13 AM     1
09/21/2015 08:30:02 PM     1
09/22/2015 07:32:20 AM     1
09/22/2015 01:47:57 AM     1
03/29/2015 04:41:50 AM     1
Name: Resolution Action Updated Date, Length: 237895, dtype: int64
---------------------------

12 MANHATTAN                12390
01 BROOKLYN                 10920
05 QUEENS                    9422
01 QUEENS                    9197
09 QUEENS                    8013
                            …
84 QUEENS                      11
56 BROOKLYN                     9
80 QUEENS                       7
Unspecified STATEN ISLAND       2
Unspecified QUEENS              2
Name: Community Board, Length: 75, dtype: int64
---------------------------
```

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

1021327.0    911
1000311.0    563
1037000.0    507
982967.0     344
1042290.0    342
             …
1000891.0      1
1024005.0      1
1026090.0      1
945160.0       1
1016436.0      1
Name: X Coordinate (State Plane), Length: 63226, dtype: int64
--------------------------

241829.0    902
202363.0    506
195702.0    500
175044.0    364
197554.0    345
            …
215171.0      1
241607.0      1
209946.0      1
201837.0      1
222234.0      1
Name: Y Coordinate (State Plane), Length: 73694, dtype: int64
--------------------------

Unspecified                         300697
Alley Pond Park - Nature Center          1
Name: Park Facility Name, dtype: int64
--------------------------

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

```
Name: Park Borough, dtype: int64
--------------------------

Unspecified                          300697
Alley Pond Park - Nature Center           1
Name: School Name, dtype: int64
--------------------------

Unspecified    300697
Q001                1
Name: School Number, dtype: int64
--------------------------

Unspecified    300697
Name: School Region, dtype: int64
--------------------------

Unspecified    300697
Name: School Code, dtype: int64
--------------------------

Unspecified    300697
7182176034          1
Name: School Phone Number, dtype: int64
--------------------------

Unspecified                                          300697
Grand Central Parkway, near the soccer field              1
Name: School Address, dtype: int64
--------------------------

Unspecified    300697
QUEENS              1
Name: School City, dtype: int64
--------------------------

Unspecified    300697
NY                  1
Name: School State, dtype: int64
--------------------------

Unspecified    300697
Name: School Zip, dtype: int64
--------------------------

N    300698
Name: School Not Found, dtype: int64
--------------------------
```

```
Series([], Name: School or Citywide Complaint, dtype: int64)
--------------------------

Series([], Name: Vehicle Type, dtype: int64)
--------------------------

Series([], Name: Taxi Company Borough, dtype: int64)
--------------------------

Series([], Name: Taxi Pick Up Location, dtype: int64)
--------------------------

FDR Dr                                        33
Belt Pkwy                                     30
BQE/Gowanus Expwy                             27
Staten Island Expwy                           21
Cross Bronx Expwy                             19
Battery Park Underpass                        15
Long Island Expwy                             12
Henry Hudson Pkwy/Rt 9A                       10
Bronx River Pkwy                               7
Major Deegan Expwy                             7
West Street                                    7
Grand Central Pkwy                             7
Jackie Robinson/Interboro Pkwy                 6
Van Wyck Expwy                                 6
Harlem River Dr                                6
Prospect Expwy                                 4
Whitestone Expwy                               4
First Ave Tunnel - UN Plaza                    3
Park Ave Tunnel - E 34th St./Grand Central     3
Cross Island Pkwy                              2
Bruckner Expwy                                 2
Sheridan Expwy                                 2
FDR Southbound                                 2
Richmond Pkwy/Korean War Vets                  2
Hutchinson River Pkwy                          2
Third Ave Br - Fifth St Basin                  1
Nassau Expwy                                   1
Third Ave Br - E 129th St                      1
Clearview Expwy                                1
Name: Bridge Highway Name, dtype: int64
--------------------------

East/Queens Bound                21
Northbound/Uptown                20
North/Bronx Bound                20
```

```
West/Staten Island Bound          18
North/Westbound (To GW Br)        17
East/Long Island Bound            17
East/Brooklyn Bound               14
Southbound/Downtown               13
North Bound                        9
To FDR/East Side                   9
West/Brooklyn Bound                7
Southbound                         7
South Bound                        7
West/Manhattan Bound               7
Westbound/To Goethals Br           7
North/Westchester County Bound     6
To West St/West Side               6
South/Downtown                     5
South/Toward Triborough Br         4
Eastbound                          4
Westbound                          3
Westbound/To BQE                   3
West/Toward Triborough Br          3
East/Bronx Bound                   3
Northbound                         2
South/JFK Airport Bound            2
South/East (To Throgs Neck Br)     2
Eastbound/To Ocean Pkwy            1
Southbound/To Triborough Br        1
Manhattan Bound                    1
North/Eastbound                    1
South/New Jersey Bound             1
South/Queens Bound                 1
South/Long Island Bound            1
Name: Bridge Highway Direction, dtype: int64
--------------------------

Roadway    162
Ramp        51
Name: Road Ramp, dtype: int64
--------------------------

East 96th St (Exit 14) - Triborough Br (Exit 17)
6
Bronx River Pkwy (Exit 4B) - Westchester Ave / White Plains Road (Exit 5A)
5
Westchester Ave / White Plains Road (Exit 5A) - Castle Hill Ave (Exit 5B)
3
Richmond Ave (Exit 7) - Victory Blvd (Exit 8)
3
BEGIN Staten Island Expwy (Exit 15N) - Lily Pond Ave/Bay St (Exit 15S)
```

```
3
     ..
East 96th St (Exit 14)
1
Brooklyn-Queens Expwy (I-278) (Exit 4) - La Guardia Airport/Astoria Blvd. (Exit
5)    1
Grand Central Pkwy (Split) (Exit 39)
1
Cross Bronx / GWB (Exit 14) - Riverside Dr (Exit 15)
1
20th Ave (Exit 15) - Linden Place (Exit 14)
1
Name: Bridge Highway Segment, Length: 160, dtype: int64
--------------------------

Series([], Name: Garage Lot Name, dtype: int64)
--------------------------

Manhattan Bound    1
Name: Ferry Direction, dtype: int64
--------------------------

St. George Terminal (Staten Island)    1
Barberi                                1
Name: Ferry Terminal Name, dtype: int64
--------------------------

40.830362    902
40.721959    505
40.703819    480
40.647132    362
40.708726    341
            …
40.847772      1
40.754813      1
40.726701      1
40.774942      1
40.716053      1
Name: Latitude, Length: 125122, dtype: int64
--------------------------

-73.866022    902
-73.809697    505
-73.942073    480
-73.790654    341
-74.004623    340
             …
-73.993975      1
```

```
-73.940633      1
-73.940517      1
-73.957122      1
-73.991378      1
Name: Longitude, Length: 125216, dtype: int64
--------------------------

(40.83036235589997, -73.86602154214397)     902
(40.72195913199264, -73.80969682426189)     505
(40.703818970933284, -73.94207345177706)    476
(40.708726489323325, -73.7906539235748)     341
(40.64713190020787, -74.00462341153786)     340
                                           ...
(40.62696139198204, -73.96729285396994)       1
(40.7101085852056, -73.79451990362605)        1
(40.67528836018061, -73.87727461085845)       1
(40.64073327311908, -73.9879939750202)        1
(40.71605290789855, -73.99137850370803)       1
Name: Location, Length: 126048, dtype: int64
--------------------------
```

[10]:
```
#Dropping unwanted columns and making a copy of it

SRD_mod = SRD_raw.drop(columns=['Unique Key','School Name', 'School Number',␣
↪'School Region', 'School Code','School Phone Number',
                                'School Address', 'School City', 'School␣
↪State','School Zip', 'School Not Found',
                                'School or Citywide Complaint','Vehicle␣
↪Type','Taxi Company Borough','Taxi Pick Up Location',
                                'Garage Lot Name','Ferry Direction','Ferry␣
↪Terminal Name'],axis=1)
```

[11]:
```
#View the Shape of data after dropping columns
SRD_mod.shape
```

[11]:
```
(300698, 35)
```

[12]:
```
#View the data after cleansing
SRD_mod.columns
```

[12]:
```
Index(['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',
       'Bridge Highway Name', 'Bridge Highway Direction', 'Road Ramp',
       'Bridge Highway Segment', 'Latitude', 'Longitude', 'Location'],
      dtype='object')
```

[13]:
```
SRD_mod.sample(3)
```

[13]:

	Created Date	Closed Date	Agency	Agency Name
293277	04-06-15 22:07	04-07-15 5:10	NYPD	New York City Police Department
229161	06-06-15 1:13	06-06-15 8:04	NYPD	New York City Police Department
25281	12-06-15 5:37	12-06-15 7:51	NYPD	New York City Police Department

	Complaint Type	Descriptor	Location Type	Incident Zip
293277	Noise - Commercial	Loud Music/Party	Store/Commercial	10003.0
229161	Blocked Driveway	No Access	Street/Sidewalk	11234.0
25281	Noise - Vehicle	Car/Truck Music	Street/Sidewalk	10465.0

	Incident Address	Street Name	Cross Street 1	Cross Street 2
293277	64 3 AVENUE	3 AVENUE	EAST 10 STREET	EAST 11 STREET
229161	1774 EAST 37 STREET	EAST 37 STREET	QUENTIN ROAD	AVENUE R
25281	NaN	NaN	NaN	NaN

	Intersection Street 1	Intersection Street 2	Address Type	City
293277	NaN	NaN	ADDRESS	NEW YORK
229161	NaN	NaN	ADDRESS	BROOKLYN
25281	VINCENT AVENUE	LAFAYETTE AVENUE	INTERSECTION	BRONX

	Landmark	Facility Type	Status	Due Date
293277	NaN	Precinct	Closed	04-07-15 6:07
229161	NaN	Precinct	Closed	06-06-15 9:13
25281	NaN	Precinct	Closed	12-06-15 13:37

	Resolution Description
293277	The Police Department responded to the complai…
229161	The Police Department responded to the complai…
25281	The Police Department responded to the complai…

	Resolution Action Updated Date	Community Board	Borough
293277	04-07-15 5:10	03 MANHATTAN	MANHATTAN
229161	06-06-15 8:04	18 BROOKLYN	BROOKLYN
25281	12-06-15 7:51	10 BRONX	BRONX

	X Coordinate (State Plane)	Y Coordinate (State Plane)
293277	987385.0	205686.0
229161	1002358.0	162968.0
25281	1034185.0	242557.0

```
       Park Facility Name Park Borough Bridge Highway Name  \
293277        Unspecified    MANHATTAN                 NaN
229161        Unspecified     BROOKLYN                 NaN
25281         Unspecified        BRONX                 NaN

       Bridge Highway Direction Road Ramp Bridge Highway Segment   Latitude  \
293277                      NaN       NaN                    NaN  40.731237
229161                      NaN       NaN                    NaN  40.613967
25281                       NaN       NaN                    NaN  40.832297

        Longitude                                  Location
293277 -73.988688  (40.73123667508269, -73.98868847296802)
229161 -73.934779  (40.61396742104142, -73.93477852675497)
25281  -73.819554  (40.83229716540471, -73.81955360554016)
```

[14]:
```
SRD_mod.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 35 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   Created Date                    300698 non-null  object
 1   Closed Date                     298534 non-null  object
 2   Agency                          300698 non-null  object
 3   Agency Name                     300698 non-null  object
 4   Complaint Type                  300698 non-null  object
 5   Descriptor                      294784 non-null  object
 6   Location Type                   300567 non-null  object
 7   Incident Zip                    298083 non-null  float64
 8   Incident Address                256288 non-null  object
 9   Street Name                     256288 non-null  object
 10  Cross Street 1                  251419 non-null  object
 11  Cross Street 2                  250919 non-null  object
 12  Intersection Street 1           43858 non-null   object
 13  Intersection Street 2           43362 non-null   object
 14  Address Type                    297883 non-null  object
 15  City                            298084 non-null  object
 16  Landmark                        349 non-null     object
 17  Facility Type                   298527 non-null  object
 18  Status                          300698 non-null  object
 19  Due Date                        300695 non-null  object
 20  Resolution Description          300698 non-null  object
 21  Resolution Action Updated Date  298511 non-null  object
 22  Community Board                 300698 non-null  object
 23  Borough                         300698 non-null  object
```

```
 24  X Coordinate (State Plane)      297158 non-null  float64
 25  Y Coordinate (State Plane)      297158 non-null  float64
 26  Park Facility Name              300698 non-null  object
 27  Park Borough                    300698 non-null  object
 28  Bridge Highway Name             243 non-null     object
 29  Bridge Highway Direction        243 non-null     object
 30  Road Ramp                       213 non-null     object
 31  Bridge Highway Segment          213 non-null     object
 32  Latitude                        297158 non-null  float64
 33  Longitude                       297158 non-null  float64
 34  Location                        297158 non-null  object
dtypes: float64(5), object(30)
memory usage: 80.3+ MB
```

Query 1: Read or convert the columns 'Created Date' and Closed Date' to datetime datatype and create a new column 'Request_Closing_Time' as the time elapsed between request creation and request closing

[15]:
```
#importing datetime library to perform above operation
import datetime
```

[16]:
```
SRD_mod["Created Date"] = pd.to_datetime(SRD_mod["Created Date"])
SRD_mod["Closed Date"] = pd.to_datetime(SRD_mod["Closed Date"])
SRD_mod["Request_Closing_Time"] = SRD_mod["Closed Date"] - SRD_mod["Created␣
 ↪Date"]
SRD_mod['Request_Closing_Time_mins'] = SRD_mod['Request_Closing_Time']/np.
 ↪timedelta64(1,'m')
print(SRD_mod["Request_Closing_Time"].head())
print("----------------------------")
print(SRD_mod['Request_Closing_Time_mins'].head(3))
```

```
0   0 days 00:55:15
1   0 days 01:26:16
2   0 days 04:51:31
3   0 days 07:45:14
4   0 days 03:27:02
Name: Request_Closing_Time, dtype: timedelta64[ns]
----------------------------
0     55.250000
1     86.266667
2    291.516667
Name: Request_Closing_Time_mins, dtype: float64
```

[17]:
```
SRD_mod.shape
```

[17]: (300698, 37)

[18]:
```
SRD_mod.sample(2)
```

[18]:

```
             Created Date         Closed Date Agency  \
86078 2015-10-08 22:58:00 2015-10-08 23:13:00   NYPD
39031 2015-11-21 22:42:02 2015-11-22 03:55:13   NYPD

                          Agency Name      Complaint Type        Descriptor  \
86078  New York City Police Department  Noise - Commercial  Loud Music/Party
39031  New York City Police Department    Blocked Driveway         No Access

             Location Type  Incident Zip     Incident Address  \
86078  Club/Bar/Restaurant       11225.0  545 FLATBUSH AVENUE
39031      Street/Sidewalk       10462.0  866 KINSELLA STREET

           Street Name   Cross Street 1    Cross Street 2  \
86078  FLATBUSH AVENUE     LINCOLN ROAD              BEND
39031  KINSELLA STREET  MATTHEWS AVENUE  BRONXDALE AVENUE

      Intersection Street 1 Intersection Street 2 Address Type      City  \
86078                   NaN                   NaN      ADDRESS  BROOKLYN
39031                   NaN                   NaN      ADDRESS     BRONX

      Landmark Facility Type  Status               Due Date  \
86078      NaN      Precinct  Closed           10-09-15 6:58
39031      NaN      Precinct  Closed  11/22/2015 06:42:02 AM

                                Resolution Description  \
86078  The Police Department responded to the complai…
39031  The Police Department responded and upon arriv…

      Resolution Action Updated Date Community Board   Borough  \
86078                 10-08-15 23:13     09 BROOKLYN  BROOKLYN
39031        11/22/2015 03:55:13 AM        11 BRONX     BRONX

       X Coordinate (State Plane)  Y Coordinate (State Plane)  \
86078                    995173.0                    179969.0
39031                   1022953.0                    247412.0

      Park Facility Name Park Borough Bridge Highway Name  \
86078       Unspecified     BROOKLYN                 NaN
39031       Unspecified        BRONX                 NaN

      Bridge Highway Direction Road Ramp Bridge Highway Segment   Latitude  \
86078                      NaN       NaN                    NaN  40.660643
39031                      NaN       NaN                    NaN  40.845679

       Longitude                                    Location  \
86078 -73.960630   (40.66064330544147, -73.96062996504574)
39031 -73.860114  (40.845679075440366, -73.86011375262585)
```

```
       Request_Closing_Time  Request_Closing_Time_mins
86078      0 days 00:15:00                  15.000000
39031      0 days 05:13:11                 313.183333
```

3. Provide major insights/patterns that you can offer in a visual format (graphs or tables); at least 4 major conclusions that you can come up with after generic data mining

1.2.2 Step 4: Exploratory Data Analysis and Data Exploration

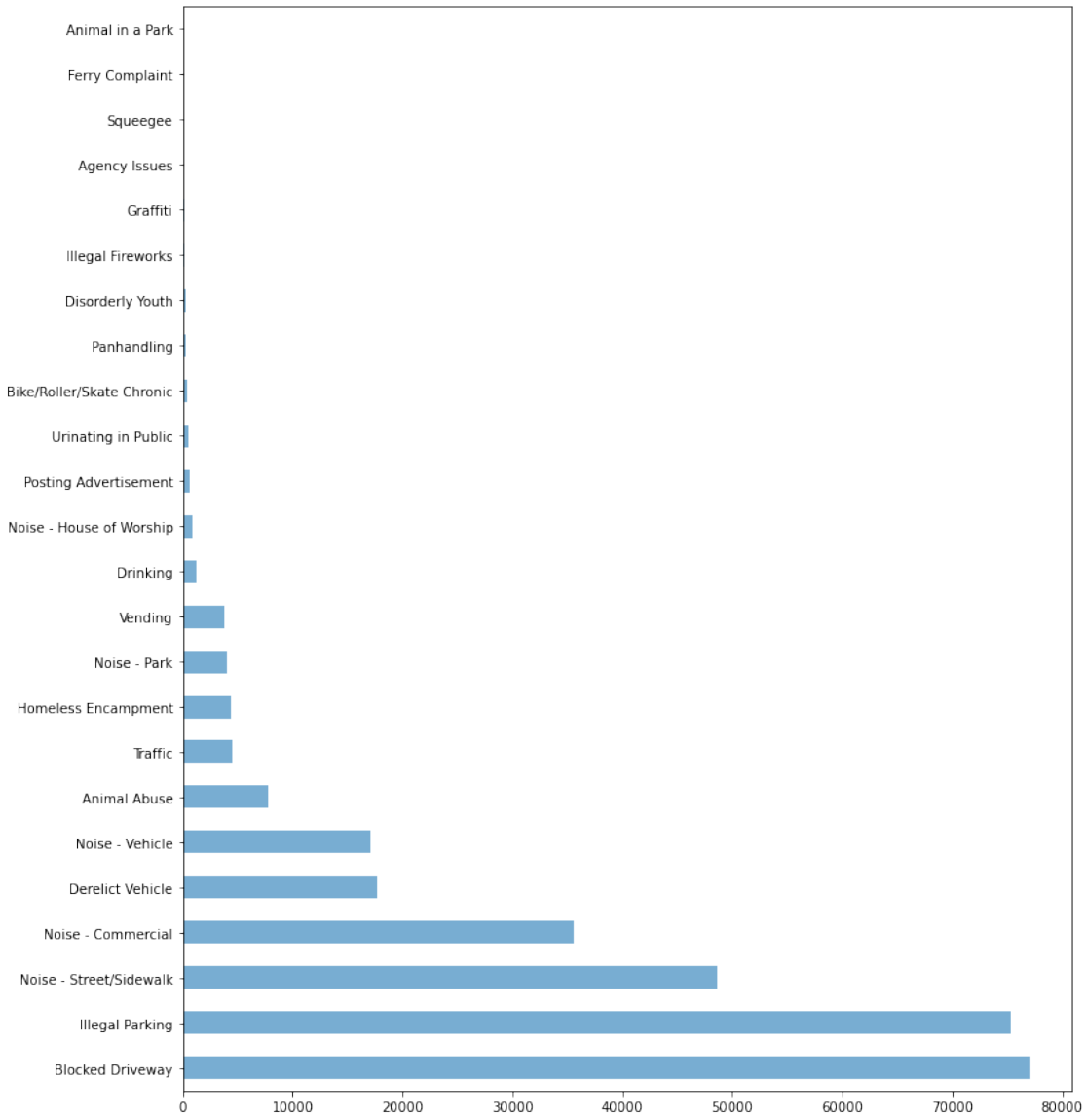
[19]:
```
#importing the libraries necessary for above step
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns
%matplotlib inline
```

[20]:
```
# Measuring the frequency (occurence) of the different complaint

SRD_complaint = SRD_mod['Complaint Type'].value_counts()
SRD_complaint = SRD_complaint.to_frame()
SRD_complaint = SRD_complaint.rename(columns={'Complaint Type':'Counts'})
SRD_complaint
```

[20]:
```
                            Counts
Blocked Driveway             77044
Illegal Parking              75361
Noise - Street/Sidewalk      48612
Noise - Commercial           35577
Derelict 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
```

[21]:
```
SRD_mod['Complaint Type'].value_counts().plot(kind='barh',alpha=0.6,␣
↪figsize=(12,15))
plt.show()
```



[22]:
```
#Lets evaluate above in percentage for clear picture

SRD_complaint['Percentage'] = np.around((SRD_complaint.Counts/SRD_complaint.
↪Counts.sum())*100,decimals=2)
#Sorting the Complaint feature based on percentage values
```

```
SRD_complaint.sort_values("Percentage")
print("First five observations \n",SRD_complaint.head(5))
print("----------------------------")
print("last five observations \n",SRD_complaint.tail(5))
```

```
First five observations
                          Counts  Percentage
Blocked Driveway          77044       25.62
Illegal Parking           75361       25.06
Noise - Street/Sidewalk   48612       16.17
Noise - Commercial        35577       11.83
Derelict Vehicle          17718        5.89
----------------------------
last five observations
                   Counts  Percentage
Graffiti              113        0.04
Agency Issues           6        0.00
Squeegee                4        0.00
Ferry Complaint         2        0.00
Animal in a Park        1        0.00
```

[23]:
```
# Keeping the complaint types >1.0 percentage ---------------

print("Before sorting \n",SRD_complaint.shape)
SRD_complaint = SRD_complaint[SRD_complaint.Percentage>1.0]
print("After  sorting \n",SRD_complaint.shape)
```

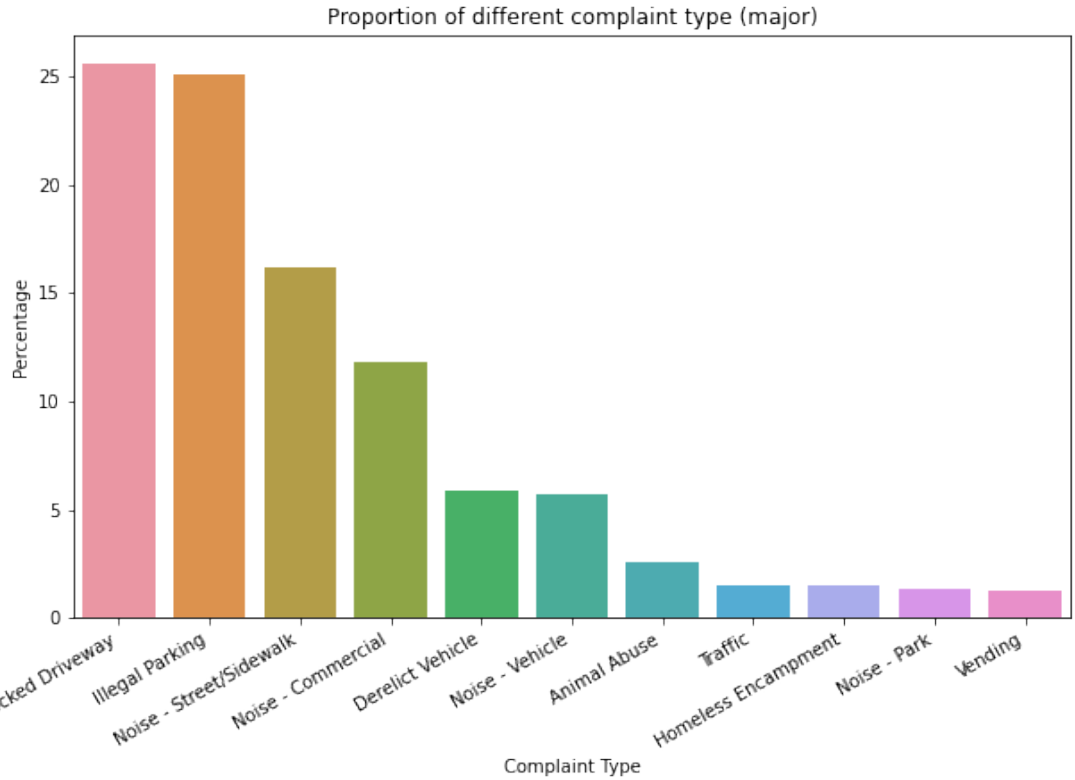
```
Before sorting
 (24, 2)
After  sorting
 (11, 2)
```

[24]:
```
print("Before re-indexing \n",SRD_complaint)
SRD_complaint = SRD_complaint.reset_index()
SRD_complaint = SRD_complaint.rename(columns={'index':'Complaint Type'})
print("After re-indexing \n",SRD_complaint)
```

```
Before re-indexing
                          Counts  Percentage
Blocked Driveway          77044       25.62
Illegal Parking           75361       25.06
Noise - Street/Sidewalk   48612       16.17
Noise - Commercial        35577       11.83
Derelict Vehicle          17718        5.89
Noise - Vehicle           17083        5.68
Animal Abuse               7778        2.59
Traffic                    4498        1.50
Homeless Encampment        4416        1.47
```

```
Noise - Park                4042        1.34
Vending                     3802        1.26
After re-indexing
              Complaint Type  Counts  Percentage
0          Blocked Driveway   77044       25.62
1           Illegal Parking   75361       25.06
2   Noise - Street/Sidewalk   48612       16.17
3        Noise - Commercial   35577       11.83
4          Derelict Vehicle   17718        5.89
5           Noise - Vehicle   17083        5.68
6              Animal Abuse    7778        2.59
7                   Traffic    4498        1.50
8       Homeless Encampment    4416        1.47
9              Noise - Park    4042        1.34
10                  Vending    3802        1.26
```

[25]:
```
# Visualization of the above evaluated dataset
plt.figure(figsize=(10,6))
SRD_complaint_barplot = sns.barplot(x=SRD_complaint['Complaint␣
 ↪Type'],y=SRD_complaint.Percentage,data=SRD_complaint)
SRD_complaint_barplot.set_xticklabels(SRD_complaint_barplot.get_xticklabels(),␣
 ↪rotation=30, ha="right")
plt.title('Proportion of different complaint type (major)')
plt.show()
plt.tight_layout()
```




```
<Figure size 432x288 with 0 Axes>
```

From the above data (Counts and Percentage), it is clear that main complaint comes from 'Blocked Driveway', 'Illegal Parking' and noise from Street/Sidewalk or Commercial. However, it is alluring to represent such results via visualization. And it is easy to realise the facts also. Now, we will do the same for several features.

[26]:
```
# Applying the above procedure for Descriptor

SRD_descriptor = np.around(((SRD_mod['Descriptor'].value_counts()*100) /␣
 ↪SRD_mod['Descriptor'].value_counts().sum()),decimals=2)
SRD_descriptor = SRD_descriptor.to_frame()
SRD_descriptor = SRD_descriptor.rename(columns={'Descriptor':'Percentage'})
SRD_descriptor['Descriptor'] = SRD_descriptor.index
cols = SRD_descriptor.columns.tolist()
cols = cols[-1:]+cols[:-1]
SRD_descriptor = SRD_descriptor[cols]
SRD_descriptor = SRD_descriptor[(SRD_descriptor.Percentage) >= 2.0]
SRD_descriptor = SRD_descriptor.reset_index()
SRD_descriptor = SRD_descriptor.drop(columns=['index'],axis=1)
SRD_descriptor
```

[26]:

	Descriptor	Percentage
0	Loud Music/Party	20.84
1	No Access	19.33
2	Posted Parking Sign Violation	7.61
3	Loud Talking	7.32
4	Partial Access	6.81
5	With License Plate	6.01
6	Blocked Hydrant	5.46
7	Commercial Overnight Parking	4.13
8	Car/Truck Music	3.82
9	Blocked Sidewalk	3.77

[27]:
```
# Applying the above procedure for Location Type
SRD_location = np.around((SRD_mod["Location Type"].value_counts()*100)/
 ↪SRD_mod["Location Type"].value_counts().sum(),decimals=2)
SRD_location = SRD_location.to_frame()
SRD_location = SRD_location.rename(columns={"Location Type":"Percentage"})
SRD_location = SRD_location.reset_index()
SRD_location = SRD_location.rename(columns={'index':'Location Type'})
SRD_location = SRD_location[(SRD_location.Percentage) >= 0.1]
SRD_location
```

[27]:

	Location Type	Percentage
0	Street/Sidewalk	82.94
1	Store/Commercial	6.78
2	Club/Bar/Restaurant	5.78
3	Residential Building/House	2.32
4	Park/Playground	1.59
5	House of Worship	0.31

[28]:
```
# Applying the above procedure for City
SRD_City = np.around((SRD_mod["City"].value_counts()*100 / SRD_mod["City"].
 ↪value_counts().sum()),decimals=2)
SRD_City = SRD_City.to_frame()
SRD_City = SRD_City.rename(columns={"City":"Percentage"})
SRD_City = SRD_City.reset_index()
SRD_City = SRD_City.rename(columns={"index":"City"})
SRD_City = SRD_City[(SRD_City.Percentage) >= 1.0]
SRD_City
```

[28]:

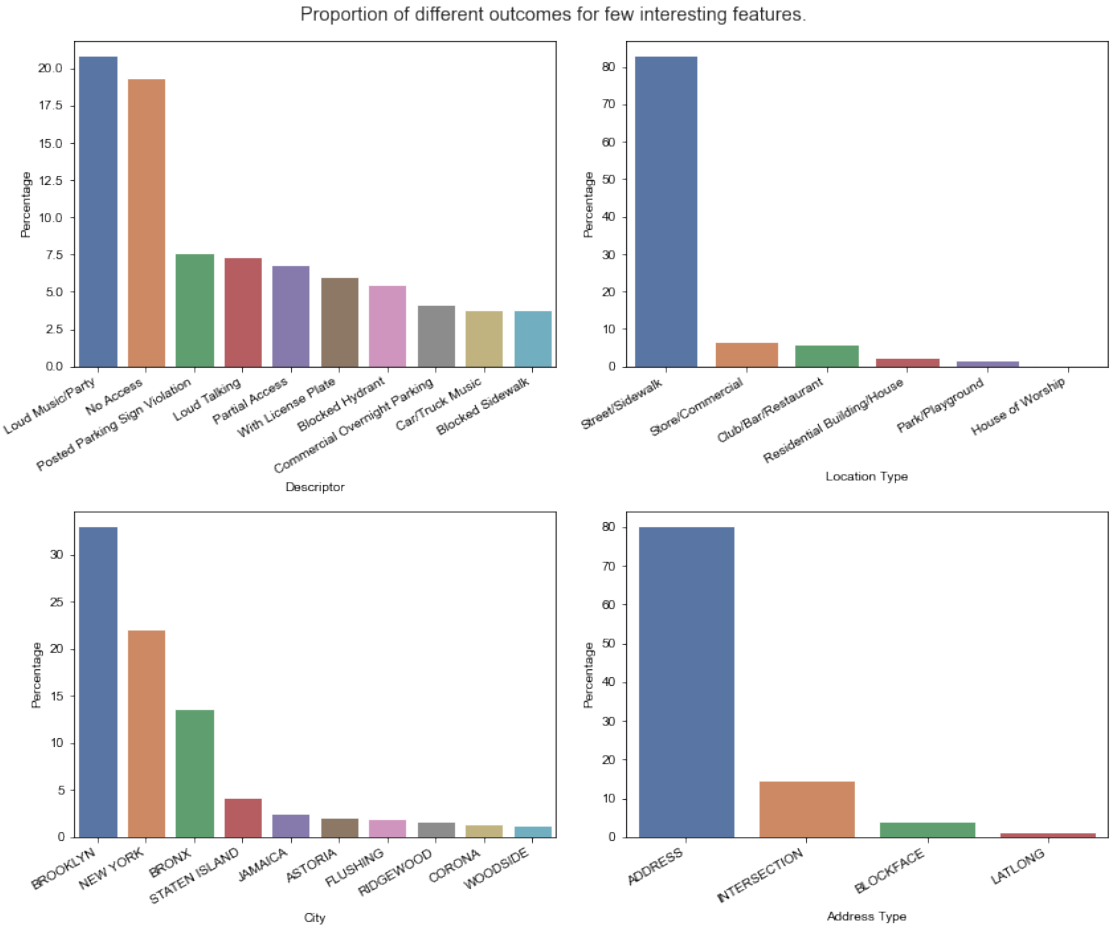
	City	Percentage
0	BROOKLYN	32.98
1	NEW YORK	22.14
2	BRONX	13.65
3	STATEN ISLAND	4.14
4	JAMAICA	2.45
5	ASTORIA	2.12

```
6       FLUSHING        2.00
7      RIDGEWOOD        1.73
8         CORONA        1.44
9       WOODSIDE        1.19
```

[29]:
```
# Applying the above procedure for Address Type
SRD_Address = np.around((SRD_mod["Address Type"].value_counts()*100 /␣
 ↪SRD_mod["Address Type"].value_counts().sum()),decimals=2)
SRD_Address = SRD_Address.to_frame()
SRD_Address = SRD_Address.rename(columns={"Address Type":"Percentage"})
SRD_Address = SRD_Address.reset_index()
SRD_Address = SRD_Address.rename(columns={"index":"Address Type"})
SRD_Address = SRD_Address[(SRD_Address.Percentage) >= 1.0 ]
SRD_Address.head()
```

[29]:
```
    Address Type  Percentage
0        ADDRESS       80.11
1   INTERSECTION       14.56
2      BLOCKFACE        4.03
3        LATLONG        1.18
```

[30]:
```
fig,ax = plt.subplots(2, 2, figsize=(12, 10))

sns.set_theme(style="whitegrid")
plt.suptitle("Proportion of different outcomes for few interesting features.")

descriptor = sns.
 ↪barplot(ax=ax[0,0],x=SRD_descriptor["Descriptor"],y=SRD_descriptor.
 ↪Percentage,)
descriptor.set_xticklabels(descriptor.get_xticklabels(), rotation=30,␣
 ↪ha="right")

location_type = sns.barplot(ax=ax[0,1],x=SRD_location['Location␣
 ↪Type'],y=SRD_location.Percentage,)
location_type.set_xticklabels(location_type.get_xticklabels(), rotation=30,␣
 ↪ha="right")

city = sns.barplot(ax=ax[1,0],x=SRD_City['City'],y=SRD_City.Percentage,)
city.set_xticklabels(city.get_xticklabels(), rotation=30, ha="right")

address = sns.barplot(ax=ax[1,1],x=SRD_Address['Address Type'],y=SRD_Address.
 ↪Percentage,)
address.set_xticklabels(address.get_xticklabels(), rotation=30, ha="right")

plt.tight_layout()
```



So it is obvious that the Loud Music/party causes the biggest problem for the citizens. And it seems most complaints occur at Street/Sidewalk. And 'Brooklyn' faces the largest problems among all other cities. However, we have mostly solid information. The place where the problem occurs is pinpointed (Proper Address)

These observations are very preliminary. One can expect or guess the outcomes from these visualizations, regarding the corresponding features. However, it needs to be realized that we can not infer/predict from here without any proper statistical explanation.

Now, let's convert the time data ('timedelta64') into integer and store them (converting into hours) in a new column. Besides that let us cut the ambiguous data.

[31]:
```
#Creating a new Dataframe with needed features
data_place_CType_RCTime = SRD_mod[['City','Complaint␣
 ↪Type','Request_Closing_Time']]
print("Null values in the new data set were : \n",data_place_CType_RCTime.
 ↪isnull().sum())
data_place_CType_RCTime.dropna(subset = ['City','Complaint␣
 ↪Type','Request_Closing_Time'], inplace = True)
```

```
print("Verifying Null values in the new data set were :␣
 ↪\n",data_place_CType_RCTime.isnull().sum())
```

```
Null values in the new data set were :
 City                    2614
Complaint Type             0
Request_Closing_Time    2164
dtype: int64
Verifying Null values in the new data set were :
 City                    0
Complaint Type          0
Request_Closing_Time    0
dtype: int64
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\util\_decorators.py:311:
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
  return func(*args, **kwargs)
```

[32]:
```
""" Converting time data to integer and storing them as hour - Method 1 using␣
 ↪lambda
data_place_CType_RCTime['DeltaT(in_hr.)'] = np.around( (data_place_CType_RCTime.
 ↪Request_Closing_Time.apply(lambda x: pd.Timedelta(x).total_seconds() \
                             / 3600.0 ) ), decimals=2)
data_place_CType_RCTime['DeltaT(in_hr.)'] """
```

[32]:
```
" Converting time data to integer and storing them as hour - Method 1 using
lambda\ndata_place_CType_RCTime['DeltaT(in_hr.)'] = np.around(
(data_place_CType_RCTime.Request_Closing_Time.apply(lambda x:
pd.Timedelta(x).total_seconds() \\ \n                              / 3600.0 ) ),
decimals=2)\ndata_place_CType_RCTime['DeltaT(in_hr.)'] "
```

[33]:
```
#Method 2- Casting DeltaT type to Interger using astype and then dividing by 10␣
 ↪pow 9 to get the sec. Later dividing by 3600 to get hours.
data_place_CType_RCTime['DeltaT(in_hr.)'] = np.around(␣
 ↪(data_place_CType_RCTime['Request_Closing_Time'].astype(np.int64)/
                                                    (pow(10,9)*3600) ),␣
 ↪decimals=2)
```

```
C:\Users\grkum\AppData\Local\Temp/ipykernel_36704/186797137.py:2: FutureWarning:
casting timedelta64[ns] values to int64 with .astype(…) is deprecated and will
raise in a future version. Use .view(…) instead.
  data_place_CType_RCTime['DeltaT(in_hr.)'] = np.around(
(data_place_CType_RCTime['Request_Closing_Time'].astype(np.int64)/
C:\Users\grkum\AppData\Local\Temp/ipykernel_36704/186797137.py:2:
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data_place_CType_RCTime['DeltaT(in_hr.)'] = np.around(
(data_place_CType_RCTime['Request_Closing_Time'].astype(np.int64)/
```

[34]:
```
#Check if any negative times exist or not
neg_time = data_place_CType_RCTime[data_place_CType_RCTime['DeltaT(in_hr.)'] <␣
 ↪0]

neg_time.head()
```

[34]:
```
Empty DataFrame
Columns: [City, Complaint Type, Request_Closing_Time, DeltaT(in_hr.)]
Index: []
```

[35]:
```
print('The no negative time difference (Created Time > Clossing Time, which is␣
 ↪not possible) = \n',neg_time)
#data_place_CType_RCTime['DeltaT(in sec)/Avg.'] = np.
 ↪around((data_place_CType_RCTime['DeltaT(in sec)']/Average_time),decimals=1)
data_place_CType_RCTime.head(6)
```

```
The no negative time difference (Created Time > Clossing Time, which is not
possible) =
 Empty DataFrame
Columns: [City, Complaint Type, Request_Closing_Time, DeltaT(in_hr.)]
Index: []
```

[35]:

	City	Complaint Type	Request_Closing_Time	DeltaT(in_hr.)
0	NEW YORK	Noise - Street/Sidewalk	0 days 00:55:15	0.92
1	ASTORIA	Blocked Driveway	0 days 01:26:16	1.44
2	BRONX	Blocked Driveway	0 days 04:51:31	4.86
3	BRONX	Illegal Parking	0 days 07:45:14	7.75
4	ELMHURST	Illegal Parking	0 days 03:27:02	3.45
5	BROOKLYN	Illegal Parking	0 days 01:53:30	1.89

Let us calculate some statistical parameters, in order to draw a conclusion on the solution time taken so that we can group them into different categories depending on the time interval.

[36]:
```
Average_time = np.around((data_place_CType_RCTime['DeltaT(in_hr.)'].
 ↪mean()),decimals=2)
print('Average time gap between logging the complaint and problem solved =␣
 ↪',Average_time, 'hour')
Central_val = np.around((data_place_CType_RCTime['DeltaT(in_hr.)'].
 ↪median()),decimals=2)
```

```
print('Central value of the distribution = ',Central_val, 'hour')
Most_occoor = np.around((data_place_CType_RCTime['DeltaT(in_hr.)'].
 ↪mode()),decimals=2)
print('Most occered value = ',Most_occoor, 'hour')
stand_dev = np.around((data_place_CType_RCTime['DeltaT(in_hr.)'].
 ↪std()),decimals=2)
print('Deviation is = ',stand_dev)
```

```
Average time gap between logging the complaint and problem solved =  4.31 hour
Central value of the distribution =  2.71 hour
Most occered value =  0    0.88
dtype: float64 hour
Deviation is =  6.08
```

So, one can take the central value as the normal time taken to solve the problem/issue. However, as it is clear from the deviation that it spreads around 6 hr.(more than the central value) from the distribution, so it is more practical to choose average time as the normal time to solve the problem. And categorize time interval as per the codes written below.

[37]:
```
conditions = [data_place_CType_RCTime['DeltaT(in_hr.)'] <= 0.5,
              (0.50 < data_place_CType_RCTime['DeltaT(in_hr.)']) &␣
 ↪(data_place_CType_RCTime['DeltaT(in_hr.)'] <= 1.00),
              (1.00 < data_place_CType_RCTime['DeltaT(in_hr.)']) &␣
 ↪(data_place_CType_RCTime['DeltaT(in_hr.)'] <= 2.00),
              (2.00 < data_place_CType_RCTime['DeltaT(in_hr.)']) &␣
 ↪(data_place_CType_RCTime['DeltaT(in_hr.)'] <= 6.00),
              (6.00 < data_place_CType_RCTime['DeltaT(in_hr.)']) &␣
 ↪(data_place_CType_RCTime['DeltaT(in_hr.)'] <= 10.00),
              (10.00 < data_place_CType_RCTime['DeltaT(in_hr.)'])]

choices = ['Super fast','Very fast','Fast','Normal','Slow','Super Slow']

data_place_CType_RCTime['Solution Status'] = np.select(conditions,choices)
```

```
C:\Users\grkum\AppData\Local\Temp/ipykernel_36704/3789049267.py:10:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data_place_CType_RCTime['Solution Status'] = np.select(conditions,choices)
```

[38]:
```
data_place_CType_RCTime.head(6)
```

[38]:

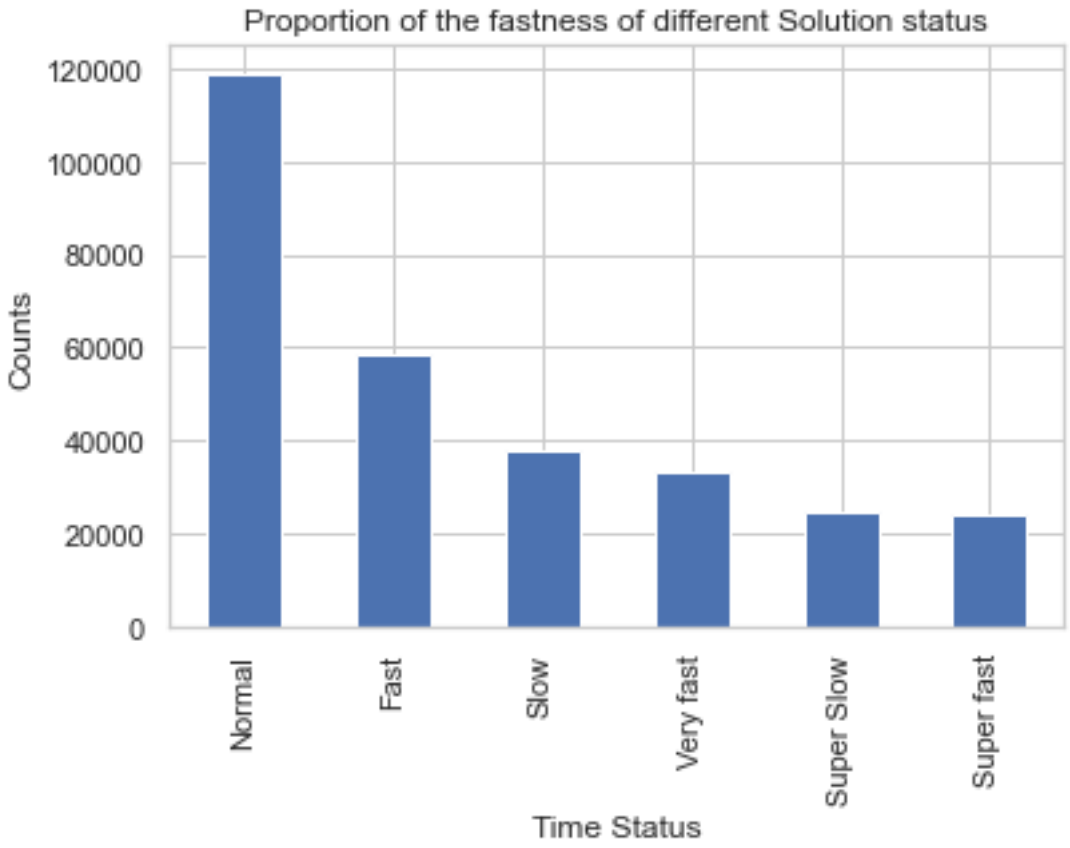
	City	Complaint Type	Request_Closing_Time	DeltaT(in_hr.)	\
0	NEW YORK	Noise - Street/Sidewalk	0 days 00:55:15	0.92	

```
1   ASTORIA         Blocked Driveway      0 days 01:26:16            1.44
2     BRONX         Blocked Driveway      0 days 04:51:31            4.86
3     BRONX          Illegal Parking      0 days 07:45:14            7.75
4  ELMHURST          Illegal Parking      0 days 03:27:02            3.45
5  BROOKLYN          Illegal Parking      0 days 01:53:30            1.89

  Solution Status
0       Very fast
1            Fast
2          Normal
3            Slow
4          Normal
5            Fast
```

[39]:
```
data_place_CType_RCTime["Solution Status"].value_counts()
```

[39]:
```
Normal        118955
Fast           58549
Slow           38068
Very fast      33459
Super Slow     24871
Super fast     24126
Name: Solution Status, dtype: int64
```

[40]:
```
data_place_CType_RCTime['Solution Status'].value_counts().plot(kind='bar')
plt.xlabel('Time Status')
plt.ylabel('Counts')
plt.title('Proportion of the fastness of different Solution status')
plt.show()
plt.tight_layout()
```




```
<Figure size 432x288 with 0 Axes>
```

Based on the above-discussed approximation, the proportion of the time interval (expressed in different groups/status) to solve the problem, is depicted here. And it is obvious that the 'Normal' status will dominant since the range is chosen around the average value.

Now, let's see, is there any pattern for lodging a complaint?

Does it depend on a particular day or is there any month where too much or fewer problems are recorded?

[41]:
```
SRD_mod['Created Date'].head(5)
```

[41]:
```
0   2015-12-31 23:59:45
1   2015-12-31 23:59:44
2   2015-12-31 23:59:29
3   2015-12-31 23:57:46
4   2015-12-31 23:56:58
Name: Created Date, dtype: datetime64[ns]
```

[42]:
```
# Creating a data frame Contain Days and Months of Complaint date
# Importing Calender to do the operation
```

```
import calendar
Year_Month_Day = pd.to_datetime(SRD_mod['Created Date'].dt.date)
Month_Day = pd.DataFrame()
Month_Day['Date'] = pd.to_datetime(Year_Month_Day.dt.date)
Month_Day['Month'] = Year_Month_Day.dt.month
Month_Day['Day'] = Year_Month_Day.dt.day
Month_Day['Month Name'] = Month_Day['Month'].apply(lambda x: calendar.
 ↪month_abbr[x])
Month_Day['Day No'] = Month_Day['Date'].dt.weekday
Month_Day['Day Name'] = Month_Day['Day No'].map({0:'Monday',1:'Tuesday',2:
 ↪'Wednesday',3:'Thursday',4:'Friday',
                                                  5:'Saturday',6:'Sunday'})
Month_Day.sample(20)
```

[42]:

	Date	Month	Day	Month Name	Day No	Day Name
300474	2015-03-29	3	29	Mar	6	Sunday
288597	2015-04-12	4	12	Apr	6	Sunday
299568	2015-03-30	3	30	Mar	0	Monday
240002	2015-05-27	5	27	May	2	Wednesday
20987	2015-12-10	12	10	Dec	3	Thursday
232228	2015-06-03	6	3	Jun	2	Wednesday
230417	2015-06-05	6	5	Jun	4	Friday
200927	2015-06-29	6	29	Jun	0	Monday
23630	2015-12-07	12	7	Dec	0	Monday
267984	2015-05-03	5	3	May	6	Sunday
163125	2015-08-01	8	1	Aug	5	Saturday
5405	2015-12-25	12	25	Dec	4	Friday
296017	2015-04-03	4	3	Apr	4	Friday
250409	2015-05-18	5	18	May	0	Monday
7086	2015-12-23	12	23	Dec	2	Wednesday
139342	2015-08-23	8	23	Aug	6	Sunday
182862	2015-07-15	7	15	Jul	2	Wednesday
179032	2015-07-18	7	18	Jul	5	Saturday
107690	2015-09-19	9	19	Sep	5	Saturday
136100	2015-08-26	8	26	Aug	2	Wednesday

[43]:
```
Month_plot = Month_Day['Month Name'].value_counts()
Month_plot = Month_plot.to_frame()
Month_plot = Month_plot.rename(columns={'Month Name':'Counts'})
Month_plot
```

[43]:

	Counts
May	36437
Sep	35427
Jun	35315
Aug	34956
Jul	34888

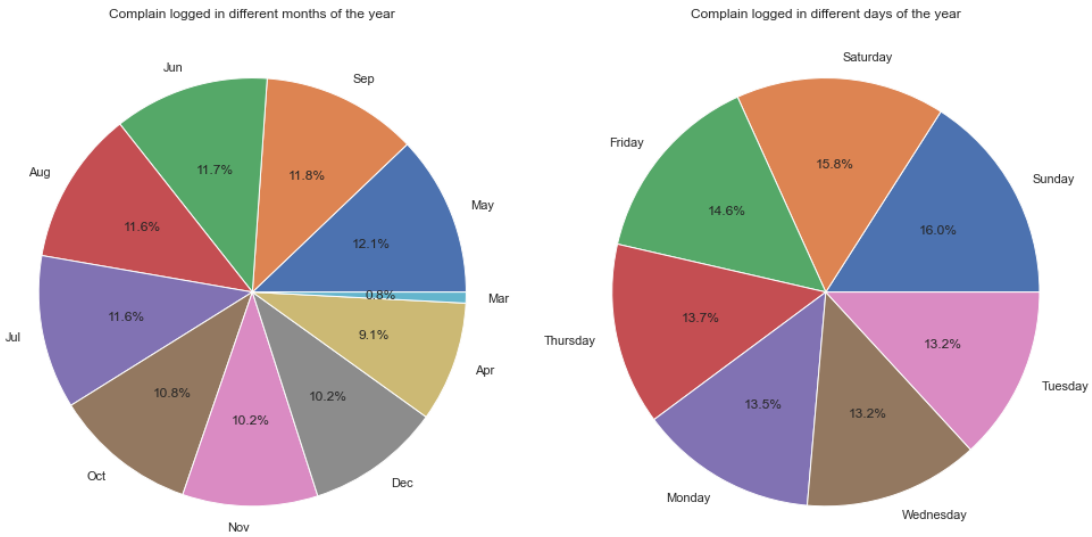
```
Oct    32605
Nov    30773
Dec    30521
Apr    27305
Mar     2471
```

[44]:
```
Day_plot = Month_Day['Day Name'].value_counts()
Day_plot = Day_plot.to_frame()
Day_plot = Day_plot.rename(columns={'Day Name':'Counts'})
Day_plot
```

[44]:

	Counts
Sunday	47969
Saturday	47564
Friday	43995
Thursday	41342
Monday	40489
Wednesday	39788
Tuesday	39551

[45]:
```
fig, axes = plt.subplots(1,2, figsize=(14,8))

axes[0].pie(Month_plot['Counts'], labels = Month_plot.index,autopct='%1.1f%%')
axes[0].set_title('Complain logged in different months of the year')

axes[1].pie(Day_plot['Counts'], labels = Day_plot.index,autopct='%1.1f%%')
axes[1].set_title('Complain logged in different days of the year')

plt.tight_layout()
```



So there is nothing abrupt for the months of lodging complaint. However, a very small amount of complaints recorded in the month of March.

The same observation can be made for the days. But if we look carefully, there is a small increment on the weekends compared to the weekly days.

However, looking at the days of a year might hide some extra information. It is better to check the days of each month of the year.

[46]:
```
Month_Day_grouped = Month_Day.groupby(['Month Name','Day␣
↪Name'],as_index=False)['Day No'].count()
Month_Day_grouped_final = Month_Day_grouped.rename(columns={'Day No':'Counts'})
Month_Day_grouped_final.head(15)
```

[46]:

	Month Name	Day Name	Counts
0	Apr	Friday	3565
1	Apr	Monday	3222
2	Apr	Saturday	4227
3	Apr	Sunday	4069
4	Apr	Thursday	4323
5	Apr	Tuesday	3586
6	Apr	Wednesday	4313
7	Aug	Friday	4684
8	Aug	Monday	5042
9	Aug	Saturday	6913
10	Aug	Sunday	6293
11	Aug	Thursday	4198
12	Aug	Tuesday	3893
13	Aug	Wednesday	3933
14	Dec	Friday	4000

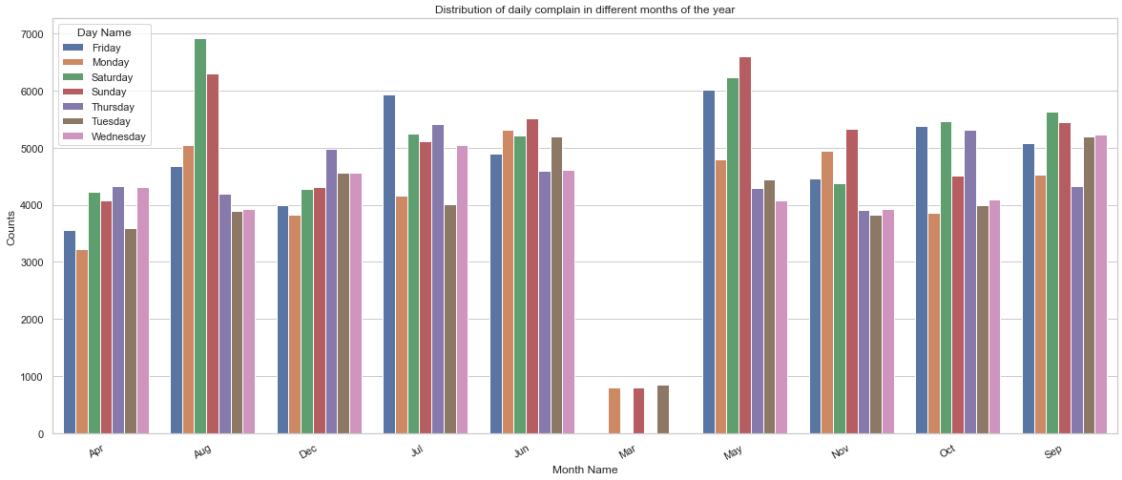
[47]:
```
Month_Day[( (Month_Day['Month Name'] == 'Apr') & (Month_Day['Day Name'] ==␣
↪'Monday') )].count()
```

[47]:
```
Date          3222
Month         3222
Day           3222
Month Name    3222
Day No        3222
Day Name      3222
dtype: int64
```

This is just to check whether the grouping operation is done correctly or not.

As you can see below, complaints created in each month for all seven days of the week are plotted. As we already counter that in March there is an abrupt decrement of complaint lodging compared to the other months. And Only three days of a week contributed here. It may contain seven days of the week, but with a very lesser amount. So let's check that to as well from the numbers.

[48]:
```
plt.figure(figsize=(20,8))

month_day_plot = sns.barplot(x=Month_Day_grouped_final['Month Name'],␣
↪y=Month_Day_grouped_final['Counts'],
                             hue=Month_Day_grouped_final['Day Name'],␣
↪data=Month_Day_grouped_final)
month_day_plot.set_xticklabels(month_day_plot.get_xticklabels(), rotation=30,␣
↪ha="right")
plt.title('Distribution of daily complain in different months of the year')
plt.show()
plt.tight_layout()
```




```
<Figure size 432x288 with 0 Axes>
```

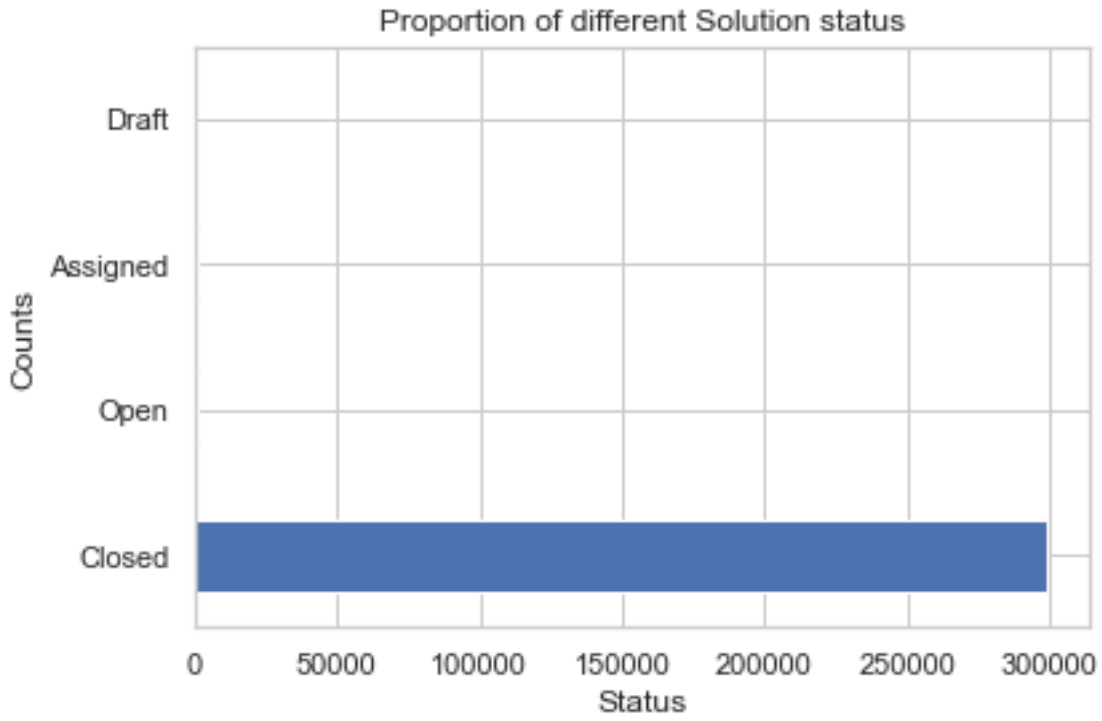
[49]:
```
Month_Day_grouped[Month_Day_grouped['Month Name'] == 'Mar']
```

[49]:

	Month Name	Day Name	Day No
35	Mar	Monday	807
36	Mar	Sunday	802
37	Mar	Tuesday	862

So complaints are recorded only in these three days of March. And let's have a look quickly at the status of the complaints.

[50]:
```
SRD_mod['Status'].value_counts().plot(kind='barh')
plt.xlabel('Status')
plt.ylabel('Counts')
plt.title('Proportion of different Solution status')
plt.show()
plt.tight_layout()
```




```
<Figure size 432x288 with 0 Axes>
```

4. Order the complaint types based on the average 'Request_Closing_Time', grouping them for different locations Ordering the complaint types based on the average 'Request_Closing_Time' (converted into integer and kept in column 'DeltaT(in_hr.)') and grouping them for different locations (such as 'City').

[51]:
```
Complaint_City_AvgTime_grouped = data_place_CType_RCTime.
 ↪groupby(['City','Complaint Type']).agg({'DeltaT(in_hr.)':'mean'})
Complaint_City_AvgTime_grouped = Complaint_City_AvgTime_grouped.rename( \
    columns={'DeltaT(in_hr.)':'Avg. Time(Given City, Complaint Type)'})
Complaint_City_AvgTime_grouped = Complaint_City_AvgTime_grouped.sort_values(
    ['City','Avg. Time(Given City, Complaint Type)'])
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
Complaint_City_AvgTime_grouped.head(10)
```

[51]:

City	Complaint Type	Avg. Time(Given City, Complaint Type)
ARVERNE	Drinking	0.240000
	Vending	0.480000
	Urinating in Public	0.690000
	Panhandling	1.030000

```
Noise - Park                                          1.285000
Graffiti                                              1.530000
Noise - House of Worship                              1.562727
Homeless Encampment                                   1.812500
Noise - Vehicle                                       1.860000
Noise - Street/Sidewalk                               1.992759
```

[52]:
```
#Displaying data in the form of table
Complaint_City_AvgTime_grouped.unstack().fillna(0).head()
```

[52]:
```
               Avg. Time(Given City, Complaint Type)                  \
Complaint Type                           Animal Abuse Animal in a Park
City
ARVERNE                                      2.153158             0.0
ASTORIA                                      5.000640             0.0
Astoria                                      0.000000             0.0
BAYSIDE                                      3.274865             0.0
BELLEROSE                                   12.725714             0.0

                                                                      \
Complaint Type Bike/Roller/Skate Chronic Blocked Driveway Derelict Vehicle
City
ARVERNE                          0.000000         2.526286         2.968519
ASTORIA                          1.740667         4.816108         9.689145
Astoria                          0.000000         4.915172         6.234167
BAYSIDE                          0.000000         2.562997         3.360000
BELLEROSE                        4.900000        10.099474        17.167978

                                                                     \
Complaint Type Disorderly Youth  Drinking   Graffiti Homeless Encampment
City
ARVERNE                 3.595000  0.240000   1.530000             1.81250
ASTORIA                 2.903333  4.722571  14.097500             4.91875
Astoria                 0.000000  0.000000   0.000000             0.00000
BAYSIDE                 2.970000  1.900000   4.553333             2.87500
BELLEROSE               1.850000  3.920000   0.000000            39.13000

                                                                   \
Complaint Type Illegal Fireworks Illegal Parking Noise - Commercial
City
ARVERNE                    0.0000        2.316207          2.285000
ASTORIA                    2.7725        4.833371          3.133039
Astoria                    0.0000        4.711362          3.542069
BAYSIDE                    0.0000        2.562763          2.234500
BELLEROSE                  6.6700        8.203019          6.740811

                                                                    \
```

```
Complaint Type Noise - House of Worship Noise - Park Noise - Street/Sidewalk
City
ARVERNE                          1.562727     1.285000                1.992759
ASTORIA                          2.022632     2.994754                3.450881
Astoria                          0.000000     0.000000                3.713333
BAYSIDE                          3.535000     3.275000                1.530667
BELLEROSE                        2.200000     1.410000                9.069231

                                                                              \
Complaint Type Noise - Vehicle Panhandling Posting Advertisement Squeegee
City
ARVERNE                1.860000        1.03                  0.00      0.0
ASTORIA                3.509020        1.15                  5.87      0.0
Astoria                0.000000        0.00                  0.00      0.0
BAYSIDE                1.709375        0.00                  0.00      0.0
BELLEROSE              2.584000        7.48                  2.26      0.0


Complaint Type   Traffic Urinating in Public   Vending
City
ARVERNE         0.000000            0.690000  0.480000
ASTORIA         5.410851            4.626667  4.935556
Astoria         0.000000            0.000000  0.000000
BAYSIDE         1.526667            0.000000  1.880000
BELLEROSE       5.760000            7.540000  0.000000
```

5. Perform a statistical test for the following: (For the below statements you need to state the Null and Alternate and then provide a statistical test to accept or reject the Null Hypothesis along with the corresponding 'p-value'.)

- Whether the average response time across complaint types is similar or not (overall)
- Are the type of complaint or service requested and location related?

[53]:
```
import scipy.stats as stat
```

- Whether the average response time across complaint types is similar or not (overall)

[54]:
```
# Average response time across complaint types --------------------

Complaint_AvgTime = data_place_CType_RCTime.groupby(['Complaint Type']).
 ↪agg({'DeltaT(in_hr.)':'mean'})
Complaint_AvgTime = pd.DataFrame(Complaint_AvgTime)
Complaint_AvgTime = Complaint_AvgTime.sort_values(['DeltaT(in_hr.)']).
 ↪reset_index()
Complaint_AvgTime
```

[54]:

	Complaint Type	DeltaT(in_hr.)
0	Posting Advertisement	1.975926
1	Illegal Fireworks	2.761190
2	Noise - Commercial	3.136907
3	Noise - House of Worship	3.193240
4	Noise - Park	3.401706
5	Noise - Street/Sidewalk	3.438573
6	Traffic	3.446291
7	Disorderly Youth	3.558916
8	Noise - Vehicle	3.588570
9	Urinating in Public	3.626486
10	Bike/Roller/Skate Chronic	3.756611
11	Drinking	3.855354
12	Vending	4.013619
13	Squeegee	4.047500
14	Homeless Encampment	4.366029
15	Panhandling	4.372852
16	Illegal Parking	4.486005
17	Blocked Driveway	4.738187
18	Animal Abuse	5.213471
19	Graffiti	7.151062
20	Derelict Vehicle	7.346105
21	Animal in a Park	336.830000

1. T-test

(a) 1-sample T-test

- t-test is statistical hypothesis test used to compare the means of two population groups.
- https://www.voxco.com/blog/anova-vs-t-test-with-a-comparison-chart/#:~:text=Comparison%20variable-,T%2DTEST,more%20than%20two%20population%20groups.

It is noteworthy that the value of the Avg. time due to complaint type 'Animal in a Park' quite out of the range. Let's find out the average with or without this particular complaint type.

[55]:
```
Tmean_without = Complaint_AvgTime[Complaint_AvgTime['Complaint Type']!='Animal␣
↪in a Park']
Tmean_without = float(Tmean_without['DeltaT(in_hr.)'].mean())
print("Without complaint type 'Animal in a Park' ----- ",Tmean_without)
Tmean_with = float(Complaint_AvgTime['DeltaT(in_hr.)'].mean())
print("With complaint type 'Animal in a Park' ----- ",Tmean_with)
```

```
Without complaint type 'Animal in a Park' -----  4.070219157949681
With complaint type 'Animal in a Park' -----  19.19566374167924
```

With complaint type 'Animal in a Park'

[56]:
```
ttest_with, pval_with = stat.ttest_1samp(Complaint_AvgTime['DeltaT(in_hr.)'],␣
 ↪Tmean_with)
print('T-statistic is =',ttest_with)
print('p value is =',np.around(pval_with))
if (pval_with<0.05):
    print('Null hypothesis is rejected since p value ({}) is less than 0.05'.
 ↪format(np.around(pval_with,decimals=2)))
else:
    print('Null hypothesis is accepted since p value ({}) is greater than 0.05'.
 ↪format(np.around(pval_with,decimals=2)))
```

```
T-statistic is = 0.0
p value is = 1.0
Null hypothesis is accepted since p value (1.0) is greater than 0.05
```

Without complaint type 'Animal in a Park'

[57]:
```
Complaint_AvgTime_without = Complaint_AvgTime.
 ↪drop([len(Complaint_AvgTime)-1],axis=0)
Complaint_AvgTime_without
```

[57]:

	Complaint Type	DeltaT(in_hr.)
0	Posting Advertisement	1.975926
1	Illegal Fireworks	2.761190
2	Noise - Commercial	3.136907
3	Noise - House of Worship	3.193240
4	Noise - Park	3.401706
5	Noise - Street/Sidewalk	3.438573
6	Traffic	3.446291
7	Disorderly Youth	3.558916
8	Noise - Vehicle	3.588570
9	Urinating in Public	3.626486
10	Bike/Roller/Skate Chronic	3.756611
11	Drinking	3.855354
12	Vending	4.013619
13	Squeegee	4.047500
14	Homeless Encampment	4.366029
15	Panhandling	4.372852
16	Illegal Parking	4.486005
17	Blocked Driveway	4.738187
18	Animal Abuse	5.213471
19	Graffiti	7.151062
20	Derelict Vehicle	7.346105

[58]:
```
ttest_without, pval_without = stat.
 ↪ttest_1samp(Complaint_AvgTime_without['DeltaT(in_hr.)'], Tmean_without)
print('T-statistic is =',ttest_without)
```

```
print('p value is =',np.around(pval_without,decimals=2))
if (pval_with<0.05):
    print('Null hypothesis is rejected since p value ({}) is less than 0.05'.
 ↪format(np.around(pval_with,decimals=2)))
else:
    print('Null hypothesis is accepted since p value ({}) is greater than 0.05'.
 ↪format(np.around(pval_with,decimals=2)))
```

```
T-statistic is = 0.0
p value is = 1.0
Null hypothesis is accepted since p value (1.0) is greater than 0.05
```

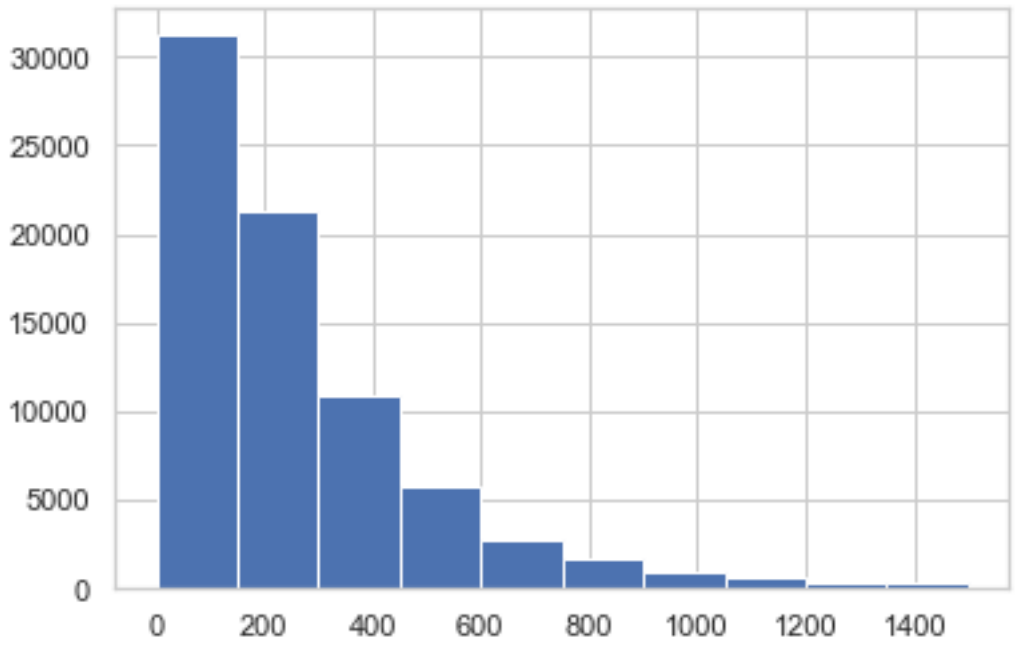
With or without the Hypothesis remain the same.

[59]:
```
SRD_mod['Complaint Type'].unique()
```

[59]:
```
array(['Noise - Street/Sidewalk', 'Blocked Driveway', 'Illegal Parking',
       'Derelict Vehicle', 'Noise - Commercial',
       'Noise - House of Worship', 'Posting Advertisement',
       'Noise - Vehicle', 'Animal Abuse', 'Vending', 'Traffic',
       'Drinking', 'Bike/Roller/Skate Chronic', 'Panhandling',
       'Noise - Park', 'Homeless Encampment', 'Urinating in Public',
       'Graffiti', 'Disorderly Youth', 'Illegal Fireworks',
       'Ferry Complaint', 'Agency Issues', 'Squeegee', 'Animal in a Park'],
      dtype=object)
```

[60]:
```
SRD_mod.head()
SRD_hypo_blocked_driveeway= SRD_mod[SRD_mod['Complaint Type']== 'Blocked␣
 ↪Driveway']['Request_Closing_Time_mins']
SRD_hypo_blocked_driveeway.hist(range=(0,1500))
```

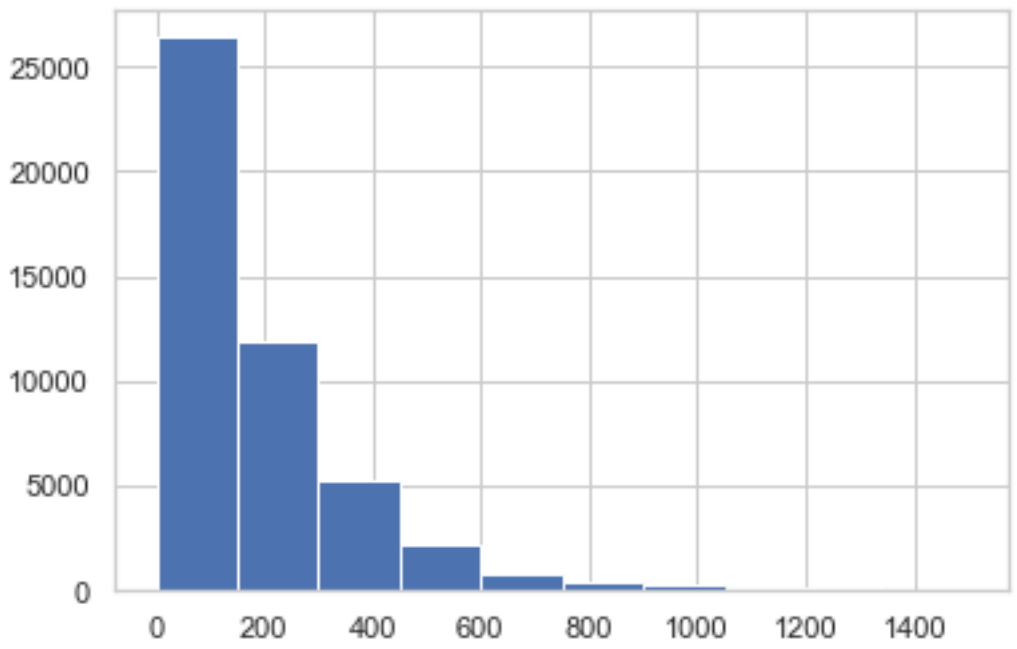
[60]:
```
<AxesSubplot:>
```



The data is left skewed, needs to be converted to gaussian

[61]:
```
SRD_hypo_noice_street= SRD_mod[SRD_mod['Complaint Type']== 'Noise - Street/
↪Sidewalk']['Request_Closing_Time_mins']
SRD_hypo_noice_street.hist(range=(0,1500))
```

[61]: <AxesSubplot:>



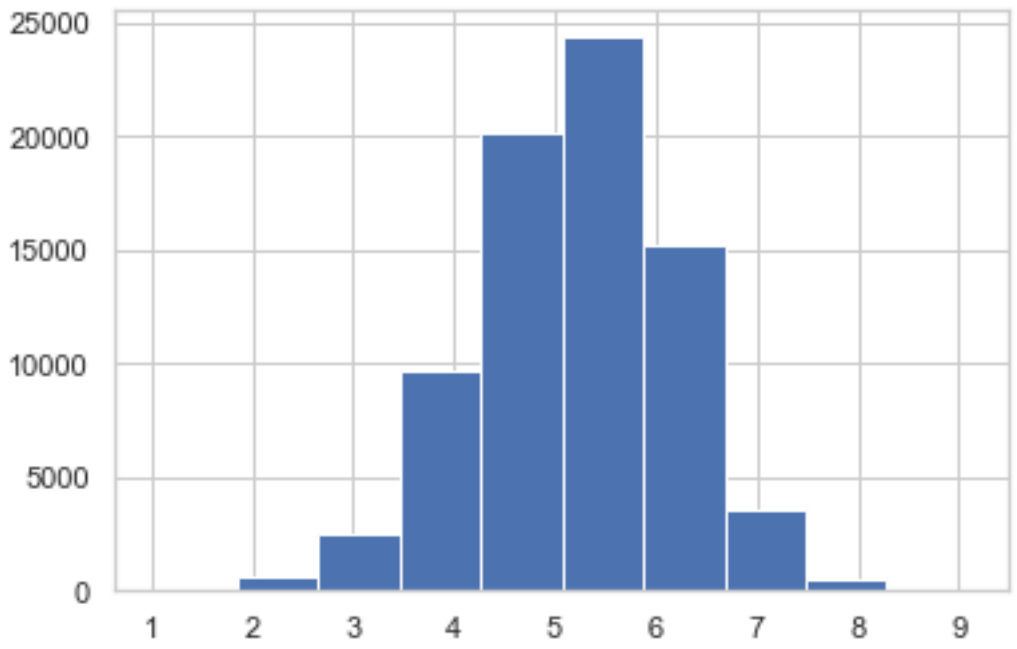
[62]:
```
 # Simiar result
 # Applying log transformation
dataset_log_transformed={}
for i in SRD_mod['Complaint Type'].unique():
    dataset_log_transformed[i]= np.log(SRD_mod[SRD_mod['Complaint␣
 ↪Type']==i]['Request_Closing_Time_mins'])
```

[63]:
```
dataset_log_transformed.keys()
```

[63]:
```
dict_keys(['Noise - Street/Sidewalk', 'Blocked Driveway', 'Illegal Parking',
'Derelict Vehicle', 'Noise - Commercial', 'Noise - House of Worship', 'Posting
Advertisement', 'Noise - Vehicle', 'Animal Abuse', 'Vending', 'Traffic',
'Drinking', 'Bike/Roller/Skate Chronic', 'Panhandling', 'Noise - Park',
'Homeless Encampment', 'Urinating in Public', 'Graffiti', 'Disorderly Youth',
'Illegal Fireworks', 'Ferry Complaint', 'Agency Issues', 'Squeegee', 'Animal in
a Park'])
```

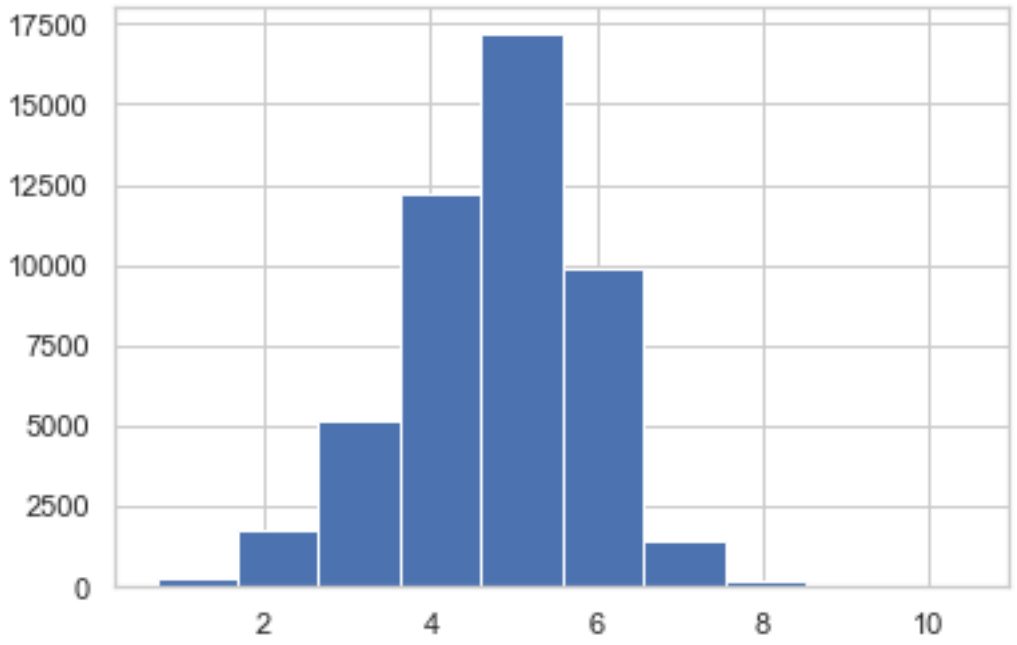
[64]:
```
dataset_log_transformed['Blocked Driveway'].hist()
```

[64]:
```
<AxesSubplot:>
```



[65]:
```
dataset_log_transformed['Noise - Street/Sidewalk'].hist()
```

[65]: <AxesSubplot:>



ANOVA Analysis (Checking for top 5 complaints)

- 1. Null Hypothesis: The average response time across complaint types is not different
- 2. Alternate Hypothesis: The average response time across complaint types is different

[66]:
```
#Perform one-way ANOVA.
#The one-way ANOVA tests the null hypothesis that two or more groups have the␣
 ↪same population mean. The test is applied to samples from two or more␣
 ↪groups,
# possibly with differing sizes.

from scipy.stats import f_oneway
stat,p = f_oneway(dataset_log_transformed['Noise - Street/Sidewalk'],␣
 ↪dataset_log_transformed['Blocked Driveway'],␣
 ↪dataset_log_transformed['Illegal Parking'],
                   dataset_log_transformed['Derelict Vehicle'],␣
 ↪dataset_log_transformed['Noise - Commercial'])

alpha=0.05
if p>0.05:
    print('Null Hypothesis is accepted')
else:
    print('Null hypothesis is rejected')
```

```
Null hypothesis is rejected
```

- Are the type of complaint or service requested and location related?

[67]:
```
print('Null data in Complaint Type =',SRD_mod['Complaint Type'].isnull().sum())
print('Null data in City =',SRD_mod['City'].isnull().sum())
```

```
Null data in Complaint Type = 0
Null data in City = 2614
```

[68]:
```
df_cc = SRD_mod[['Complaint Type','City']]
df_cc = df_cc.dropna()
#df_cc.isnull().sum()
#df_cc
```

[69]:
```
City_Complaint = pd.crosstab(SRD_mod['Complaint␣
 ↪Type'],SRD_mod['City'],margins=True, margins_name='Total')
#City_Complaint = pd.crosstab(df_cc['Complaint Type'],df_cc['City'])
City_Complaint.head(6)
```

[69]:

City	ARVERNE	ASTORIA	Astoria	BAYSIDE	BELLEROSE	\
Complaint Type						
Animal Abuse	38	125	0	37	7	
Animal in a Park	0	0	0	0	0	

Bike/Roller/Skate Chronic	0	15	0	0	1
Blocked Driveway	35	2618	116	377	95
Derelict Vehicle	27	351	12	198	89
Disorderly Youth	2	3	0	1	2

City	BREEZY POINT	BRONX	BROOKLYN	CAMBRIA HEIGHTS	\
Complaint Type					
Animal Abuse	2	1415	2394	11	
Animal in a Park	0	0	0	0	
Bike/Roller/Skate Chronic	0	20	111	0	
Blocked Driveway	3	12755	28148	147	
Derelict Vehicle	3	1953	5181	115	
Disorderly Youth	0	63	72	0	

City	CENTRAL PARK	COLLEGE POINT	CORONA	EAST ELMHURST	\
Complaint Type					
Animal Abuse	0	28	61	59	
Animal in a Park	0	0	0	0	
Bike/Roller/Skate Chronic	0	0	0	1	
Blocked Driveway	0	435	2761	1408	
Derelict Vehicle	0	184	57	113	
Disorderly Youth	0	1	6	1	

City	ELMHURST	East Elmhurst	FAR ROCKAWAY	FLORAL PARK	\
Complaint Type					
Animal Abuse	38	0	89	2	
Animal in a Park	0	0	0	0	
Bike/Roller/Skate Chronic	2	0	0	0	
Blocked Driveway	1446	0	284	20	
Derelict Vehicle	78	1	187	56	
Disorderly Youth	2	0	1	1	

City	FLUSHING	FOREST HILLS	FRESH MEADOWS	GLEN OAKS	\
Complaint Type					
Animal Abuse	143	45	45	5	
Animal in a Park	0	0	0	0	
Bike/Roller/Skate Chronic	3	5	0	0	
Blocked Driveway	2795	663	503	30	
Derelict Vehicle	440	52	291	49	
Disorderly Youth	2	1	0	0	

City	HOLLIS	HOWARD BEACH	Howard Beach	\
Complaint Type				
Animal Abuse	33	31	0	
Animal in a Park	0	0	0	
Bike/Roller/Skate Chronic	0	1	0	
Blocked Driveway	342	167	1	

Derelict Vehicle	143	138	0
Disorderly Youth	1	1	0

City	JACKSON HEIGHTS	JAMAICA	KEW GARDENS	LITTLE NECK	\
Complaint Type					
Animal Abuse	42	229	19	15	
Animal in a Park	0	0	0	0	
Bike/Roller/Skate Chronic	2	2	0	0	
Blocked Driveway	568	2818	313	121	
Derelict Vehicle	29	954	14	61	
Disorderly Youth	0	8	0	2	

City	LONG ISLAND CITY	Long Island City	MASPETH	\
Complaint Type				
Animal Abuse	30	0	36	
Animal in a Park	0	0	0	
Bike/Roller/Skate Chronic	3	0	1	
Blocked Driveway	772	34	732	
Derelict Vehicle	195	4	434	
Disorderly Youth	1	0	2	

City	MIDDLE VILLAGE	NEW HYDE PARK	NEW YORK	\
Complaint Type				
Animal Abuse	22	1	1525	
Animal in a Park	0	0	0	
Bike/Roller/Skate Chronic	1	0	225	
Blocked Driveway	457	53	2072	
Derelict Vehicle	296	14	537	
Disorderly Youth	0	0	69	

City	OAKLAND GARDENS	OZONE PARK	QUEENS	\
Complaint Type				
Animal Abuse	19	48	0	
Animal in a Park	0	0	1	
Bike/Roller/Skate Chronic	2	1	0	
Blocked Driveway	132	1259	2	
Derelict Vehicle	86	420	1	
Disorderly Youth	1	4	0	

City	QUEENS VILLAGE	REGO PARK	RICHMOND HILL	\
Complaint Type				
Animal Abuse	66	26	32	
Animal in a Park	0	0	0	
Bike/Roller/Skate Chronic	0	0	0	
Blocked Driveway	585	611	872	
Derelict Vehicle	370	81	167	
Disorderly Youth	0	0	0	

```
City                        RIDGEWOOD  ROCKAWAY PARK  ROSEDALE  SAINT ALBANS  \
Complaint Type
Animal Abuse                      117             30        33            30
Animal in a Park                    0              0         0             0
Bike/Roller/Skate Chronic           3              0         2             0
Blocked Driveway                 1694             70       211           244
Derelict Vehicle                  330              9       208           202
Disorderly Youth                    3              4         0             1

City                        SOUTH OZONE PARK  SOUTH RICHMOND HILL  \
Complaint Type
Animal Abuse                              55                   26
Animal in a Park                           0                    0
Bike/Roller/Skate Chronic                  1                    1
Blocked Driveway                         942                 1548
Derelict Vehicle                         358                  289
Disorderly Youth                           2                    2

City                        SPRINGFIELD GARDENS  STATEN ISLAND  SUNNYSIDE  \
Complaint Type
Animal Abuse                                 24            557         35
Animal in a Park                              0              0          0
Bike/Roller/Skate Chronic                     0              7          2
Blocked Driveway                            262           2142        206
Derelict Vehicle                            210           1766         10
Disorderly Youth                              0             23          2

City                        WHITESTONE  WOODHAVEN  WOODSIDE  Woodside  Total
Complaint Type
Animal Abuse                        28         45        69         0   7767
Animal in a Park                     0          0         0         0      1
Bike/Roller/Skate Chronic            4          2         4         0    422
Blocked Driveway                   208       1060      1613        11  76761
Derelict Vehicle                   227        308       247         2  17547
Disorderly Youth                     1          0         1         0    286
```

Applying the ANOVA for a few combinations and let's see how does it go?

[70]:
```
import scipy.stats as stat
print("For 'ARVERNE' and 'ASTORIA' pair -------")
f_val,p_val = stat.f_oneway(City_Complaint['ARVERNE'],City_Complaint['ASTORIA'])
print('F-statistic is =',f_val)
print('p value is =',np.around(p_val,decimals=2))
```

```
For 'ARVERNE' and 'ASTORIA' pair -------
F-statistic is = 3.3097701947747975
p value is = 0.08
```

[71]:
```
print("For 'ARVERNE' and 'BROOKLYN' pair -------")
f_val,p_val = stat.
 ↪f_oneway(City_Complaint['ARVERNE'],City_Complaint['BROOKLYN'])
print('F-statistic is =',f_val)
print('p value is =',np.around(p_val,decimals=2))
```

```
For 'ARVERNE' and 'BROOKLYN' pair -------
F-statistic is = 3.716772993046823
p value is = 0.06
```

[72]:
```
print("For 'HOLLIS' and 'JAMAICA' pair -------")
f_val,p_val = stat.f_oneway(City_Complaint['HOLLIS'],City_Complaint['JAMAICA'])
print('F-statistic is =',f_val)
print('p value is =',np.around(p_val,decimals=2))
```

```
For 'HOLLIS' and 'JAMAICA' pair -------
F-statistic is = 2.666621070410633
p value is = 0.11
```

[73]:
```
print("For 'MASPETH' and 'QUEENS' pair -------")
f_val,p_val = stat.f_oneway(City_Complaint['MASPETH'],City_Complaint['QUEENS'])
print('F-statistic is =',f_val)
print('p value is =',np.around(p_val,decimals=2))
```

```
For 'MASPETH' and 'QUEENS' pair -------
F-statistic is = 3.368313812374042
p value is = 0.07
```

We have seen a few of the pairs. And it seems p-value is around 0.05. This is a very insufficient number of pair checking. So, though it looks like 'neglecting Null Hypothesis', but we can not certain unless checking all pairs (${}^{53}C_2$ combinations for 53 cities). Even for 21 complaint types, it is still ${}^{21}C_2$ combinations.

It is more proper to use the chi square contingency test for such data structure. It gives us the correlation between different features (here different cities for a given complaint type).

- Null Hypothesis states - there is no dependence or relation among the features
- Alternate Hypothesis states - there is a relation among the features

Chi square Contigency test

[74]:
```
chai2, p_val, df, exp_frq = stat.chi2_contingency(City_Complaint)
```

[75]:
```
print('Chai square value =',chai2)
print('p-value is =',p_val)
```

```
Chai square value = 119769.34666374495
p-value is = 0.0
```

[76]:
```
if (p_val<0.05):
    print('Null hypothesis is rejected since p value ({}) is less than 0.05'.
 ↪format(np.around(p_val,decimals=2)))
else:
    print('Null hypothesis is accepted since p value ({}) is greater than 0.05'.
 ↪format(np.around(p_val,decimals=2)))
```

```
Null hypothesis is rejected since p value (0.0) is less than 0.05
```

Thus we may conclude that there is a relationship between the type of complaint or service requested and location.

[77]:
```
### Method 2: Pearson Correlation Method
```

[78]:
```
df_corr_pearson= SRD_mod[['Complaint␣
 ↪Type','Location','Latitude','Longitude','City','Borough']]
df_corr_pearson.head(10)
```

[78]:

	Complaint Type	Location	Latitude	Longitude	City	Borough
0	Noise - Street/Sidewalk	(40.86568153633767, -73.92350095571744)	40.865682	-73.923501	NEW YORK	MANHATTAN
1	Blocked Driveway	(40.775945312321085, -73.91509393898605)	40.775945	-73.915094	ASTORIA	QUEENS
2	Blocked Driveway	(40.870324522111424, -73.88852464418646)	40.870325	-73.888525	BRONX	BRONX
3	Illegal Parking	(40.83599404683083, -73.82837939584206)	40.835994	-73.828379	BRONX	BRONX
4	Illegal Parking	(40.733059618956815, -73.87416975810375)	40.733060	-73.874170	ELMHURST	QUEENS
5	Illegal Parking	(40.66082272389114, -73.99256786342693)	40.660823	-73.992568	BROOKLYN	BROOKLYN
6	Illegal Parking	(40.840847591440415, -73.9373750864581)	40.840848	-73.937375	NEW YORK	MANHATTAN
7	Blocked Driveway	(40.83750262540012, -73.90290517326568)	40.837503	-73.902905	BRONX	BRONX
8	Illegal Parking	(40.704977164399935, -73.8326047502584)	40.704977	-73.832605	KEW GARDENS	QUEENS
9	Blocked Driveway	(40.623793065806524, -73.99953890121567)	40.623793	-73.999539	BROOKLYN	BROOKLYN

[79]:
```
#https://stackoverflow.com/questions/51102205/
 ↪how-to-know-the-labels-assigned-by-astypecategory-cat-codes
df_corr_pearson['Complaint Type']=df_corr_pearson['Complaint Type'].
 ↪astype('category').cat.codes
df_corr_pearson['City']= df_corr_pearson['City'].astype('category').cat.codes
```

```
df_corr_pearson['Borough']= df_corr_pearson['Borough'].astype('category').cat.
 ↪codes
df_corr_pearson.head()
```

```
C:\Users\grkum\AppData\Local\Temp/ipykernel_36704/1886232764.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_corr_pearson['Complaint Type']=df_corr_pearson['Complaint
Type'].astype('category').cat.codes
C:\Users\grkum\AppData\Local\Temp/ipykernel_36704/1886232764.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_corr_pearson['City']= df_corr_pearson['City'].astype('category').cat.codes
C:\Users\grkum\AppData\Local\Temp/ipykernel_36704/1886232764.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_corr_pearson['Borough']=
df_corr_pearson['Borough'].astype('category').cat.codes
```

[79]:

	Complaint Type	Location	Latitude	Longitude	City	Borough
0	16	(40.86568153633767, -73.92350095571744)	40.865682	-73.923501	33	2
1	4	(40.775945312321085, -73.91509393898605)	40.775945	-73.915094	1	3
2	4	(40.870324522111424, -73.88852464418646)	40.870325	-73.888525	6	0
3	12	(40.83599404683083, -73.82837939584206)	40.835994	-73.828379	6	0
4	12	(40.733059618956815, -73.87416975810375)	40.733060	-73.874170	13	3

[80]:
```
df_corr_pearson.corr(method='pearson')
```

[80]:

	Complaint Type	Latitude	Longitude	City	Borough
Complaint Type	1.000000	0.150197	-0.184391	0.091711	-0.057730
Latitude	0.150197	1.000000	0.364966	-0.000571	-0.249501
Longitude	-0.184391	0.364966	1.000000	-0.123933	0.021277
City	0.091711	-0.000571	-0.123933	1.000000	0.654637
Borough	-0.057730	-0.249501	0.021277	0.654637	1.000000

From the first line it can be seen that the complaint types has little correlation with the location