# Simplilearn- Data Science with Python

# Name- SARATH S # Date- 30/07/2021

In [37]:
```
# Analysis task to be performed

# Perform a service request data analysis of New York City 311 calls)

# 1. Import a 311 NYC service request.

# 2. Read or convert the columns ‘Created Date’ and Closed Date’ to datetime datatype
#   and create a new column ‘Request_Closing_Time’ as the time elapsed between request creation and request closi
#   (Hint: Explore the package/module datetime)

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

# 4. Order the complaint types based on the average ‘Request_Closing_Time’, grouping them for different locations

# 5. Perform a statistical test for the following:

# Please note: For the below statements you need to state the Null and Alternate and then provide a statistical t
# 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 relate.
```

# 1. Import a 311 NYC service request.

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [4]:
```
import io
%cd "C:\Data Science\Data Science with Python\Project1\Data Science with Python Two"
```

```
C:\Data Science\Data Science with Python\Project1\Data Science with Python Two
```

In [38]:
```
# Import a 311 NYC service request.
service_311=pd.read_csv('311_Service_Requests_from_2010_to_Present.csv')
```

```
C:\Users\Sarath S Kumar\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3165: DtypeWarning: Columns
(48,49) have mixed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [6]:
```
service_311.head()
```

Out[6]:

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

5 rows × 53 columns

In [7]:
```
service_311.describe()
```

Out[7]:

| | Unique Key | Incident Zip | X Coordinate (State Plane) | Y Coordinate (State Plane) | School or Citywide Complaint | Vehicle Type | Taxi Company Borough | Taxi Pick Up Location | Garage Lot Name | Latitude | Longitu |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 3.006980e+05 | 298083.000000 | 2.971580e+05 | 297158.000000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 297158.000000 | 297158.0000 |
| **mean** | 3.130054e+07 | 10848.888645 | 1.004854e+06 | 203754.534416 | NaN | NaN | NaN | NaN | NaN | 40.725885 | -73.9256 |
| **std** | 5.738547e+05 | 583.182081 | 2.175338e+04 | 29880.183529 | NaN | NaN | NaN | NaN | NaN | 0.082012 | 0.0784 |
| **min** | 3.027948e+07 | 83.000000 | 9.133570e+05 | 121219.000000 | NaN | NaN | NaN | NaN | NaN | 40.499135 | -74.2549 |
| **25%** | 3.080118e+07 | 10310.000000 | 9.919752e+05 | 183343.000000 | NaN | NaN | NaN | NaN | NaN | 40.669796 | -73.9721 |
| **50%** | 3.130436e+07 | 11208.000000 | 1.003158e+06 | 201110.500000 | NaN | NaN | NaN | NaN | NaN | 40.718661 | -73.9317 |
| **75%** | 3.178446e+07 | 11238.000000 | 1.018372e+06 | 224125.250000 | NaN | NaN | NaN | NaN | NaN | 40.781840 | -73.8768 |
| **max** | 3.231065e+07 | 11697.000000 | 1.067173e+06 | 271876.000000 | NaN | NaN | NaN | NaN | NaN | 40.912869 | -73.7007 |

In [8]:
```
# Printing the datatypes information
service_311.dtypes
```

Out[8]:
```
Unique Key                        int64
Created Date                     object
Closed Date                      object
Agency                           object
Agency Name                      object
Complaint Type                   object
Descriptor                       object
Location Type                    object
Incident Zip                    float64
Incident Address                 object
Street Name                      object
Cross Street 1                   object
Cross Street 2                   object
Intersection Street 1            object
Intersection Street 2            object
Address Type                     object
City                             object
Landmark                         object
Facility Type                    object
Status                           object
Due Date                         object
Resolution Description           object
Resolution Action Updated Date   object
Community Board                  object
Borough                          object
X Coordinate (State Plane)      float64
Y Coordinate (State Plane)      float64
Park Facility Name               object
Park Borough                     object
School Name                      object
School Number                    object
School Region                    object
School Code                      object
School Phone Number              object
School Address                   object
School City                      object
School State                     object
School Zip                       object
School Not Found                 object
School or Citywide Complaint    float64
Vehicle Type                    float64
Taxi Company Borough            float64
Taxi Pick Up Location           float64
Bridge Highway Name              object
Bridge Highway Direction         object
Road Ramp                        object
Bridge Highway Segment           object
Garage Lot Name                 float64
Ferry Direction                  object
Ferry Terminal Name              object
Latitude                        float64
Longitude                       float64
Location                         object
dtype: object
```

In [20]:

In [39]:
```
# Printing the rows and columns information
service_311.shape
```

Out[39]: (300698, 53)

In [10]:
```
# Deleting colums that are having more than 70% data missing
service_311.isnull().sum().sort_values(ascending=False)
```

Out[10]:
```
School or Citywide Complaint      300698
Vehicle Type                      300698
Taxi Company Borough              300698
Taxi Pick Up Location             300698
Garage Lot Name                   300698
Ferry Direction                   300697
Ferry Terminal Name               300696
Road Ramp                         300485
Bridge Highway Segment            300485
Bridge Highway Name               300455
Bridge Highway Direction          300455
Landmark                          300349
Intersection Street 2             257336
Intersection Street 1             256840
Cross Street 2                     49779
Cross Street 1                     49279
Street Name                        44410
Incident Address                   44410
Descriptor                          5914
X Coordinate (State Plane)          3540
Latitude                            3540
Longitude                           3540
Y Coordinate (State Plane)          3540
Location                            3540
Address Type                        2815
Incident Zip                        2615
City                                2614
Resolution Action Updated Date      2187
Facility Type                       2171
Closed Date                         2164
Location Type                        131
Due Date                               3
School Region                          1
School Code                            1
School Zip                             1
Borough                                0
Agency                                 0
Agency Name                            0
Complaint Type                         0
Status                                 0
School Not Found                       0
Resolution Description                 0
Community Board                        0
School State                           0
School City                            0
School Address                         0
School Phone Number                    0
School Number                          0
School Name                            0
Park Borough                           0
Park Facility Name                     0
Created Date                           0
Unique Key                             0
dtype: int64
```

In [11]:
```
df=service_311.drop(['School or Citywide Complaint', 'Vehicle Type', 'Taxi Company Borough', 'Taxi Pick Up Locati
                     'Garage Lot Name', 'Ferry Direction', 'Ferry Terminal Name', 'Road Ramp', 'Bridge Highway Seg
                     'Bridge Highway Name','Bridge Highway Direction', 'Landmark', 'Intersection Street 2',
                     'Intersection Street 1'], axis=1)
```

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

In [ ]:
```
import datetime as dt
```

In [45]:
```
df['Created Date'].dtype
```

Out[45]:
```
dtype('O')
```

In [46]:
```
df['Closed Date'].dtype
```

Out[46]:
```
dtype('O')
```

In [49]:
```
df['Created Date_DT'] = pd.to_datetime(df['Created Date'])
df['Closed Date_DT'] = pd.to_datetime(df['Closed Date'])
df['Request_Closing_Time_DT'] = (df['Closed Date_DT'] - df['Created Date_DT'])
df['Request_Closing_Time'] = df['Request_Closing_Time_DT'] / np.timedelta64(1, 'h')
```

In [50]:
```
df.head()
```

Out[50]:

| | Unique Key | Created Date | Closed Date | Agency | Agency Name | Complaint Type | Descriptor | Location Type | Incident Zip | Incident Address | ... | School Not Found | Latitude |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 32310363 | 12/31/2015 11:59:45 PM | 01-01-16 0:55 | NYPD | New York City Police Department | Noise - Street/Sidewalk | Loud Music/Party | Street/Sidewalk | 10034.0 | 71 VERMILYEA AVENUE | ... | N | 40.865682 |
| 1 | 32309934 | 12/31/2015 11:59:44 PM | 01-01-16 1:26 | NYPD | New York City Police Department | Blocked Driveway | No Access | Street/Sidewalk | 11105.0 | 27-07 23 AVENUE | ... | N | 40.775945 |
| 2 | 32309159 | 12/31/2015 11:59:29 PM | 01-01-16 4:51 | NYPD | New York City Police Department | Blocked Driveway | No Access | Street/Sidewalk | 10458.0 | 2897 VALENTINE AVENUE | ... | N | 40.870325 |
| 3 | 32305098 | 12/31/2015 11:57:46 PM | 01-01-16 7:43 | NYPD | New York City Police Department | Illegal Parking | Commercial Overnight Parking | Street/Sidewalk | 10461.0 | 2940 BAISLEY AVENUE | ... | N | 40.835994 |
| 4 | 32306529 | 12/31/2015 11:56:58 PM | 01-01-16 3:24 | NYPD | New York City Police Department | Illegal Parking | Blocked Sidewalk | Street/Sidewalk | 11373.0 | 87-14 57 ROAD | ... | N | 40.733060 |

5 rows × 45 columns

# 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.

In [96]:
```
#Identifying the top 10 Complaint Types.
groupbyComplaintType['Complaint Type'].value_counts().nlargest(10)
```

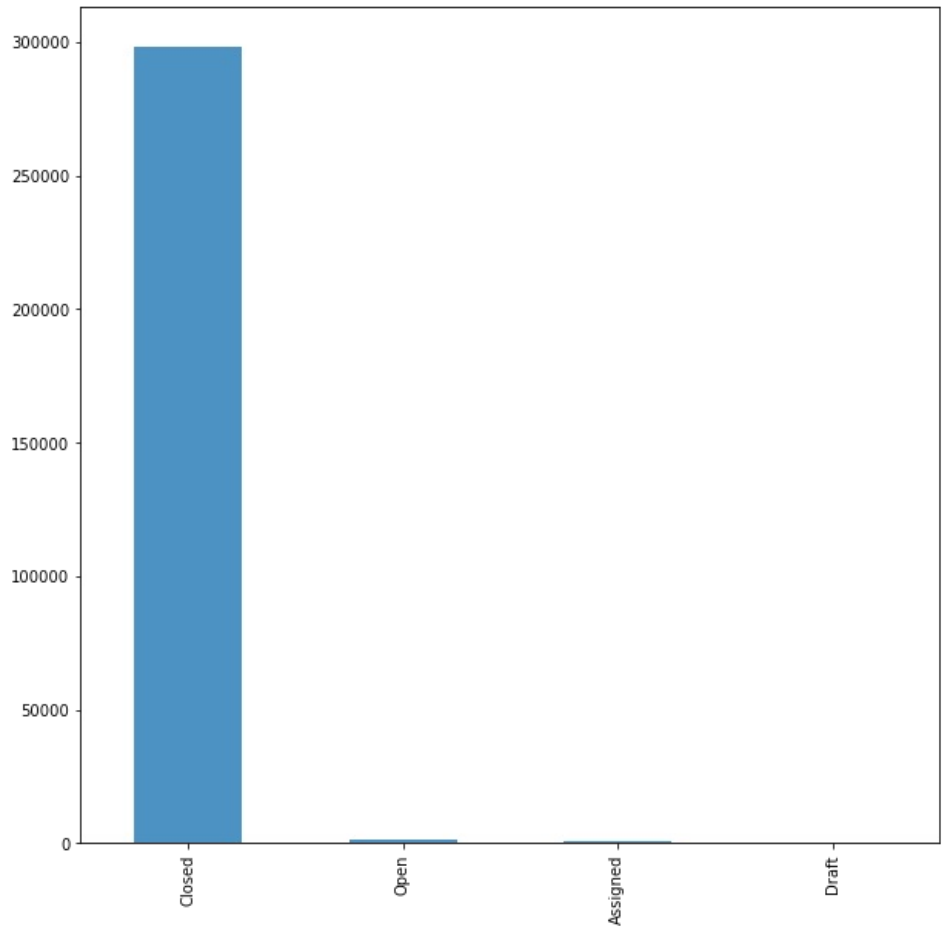
Out[96]:
```
Complaint Type           Complaint Type
Blocked Driveway         Blocked Driveway           77044
Illegal Parking          Illegal Parking            75361
Noise - Street/Sidewalk  Noise - Street/Sidewalk    48612
Noise - Commercial       Noise - Commercial         35577
Derelict Vehicle         Derelict Vehicle           17718
Noise - Vehicle          Noise - Vehicle            17083
Animal Abuse             Animal Abuse                7778
Traffic                  Traffic                     4498
Homeless Encampment      Homeless Encampment         4416
Noise - Park             Noise - Park                4042
Name: Complaint Type, dtype: int64
```

In [98]:
```
groupbyComplaintType=df.groupby('Complaint Type')
```

In [99]:
```
# Plotting the status of the complaints in a bar graph.
```

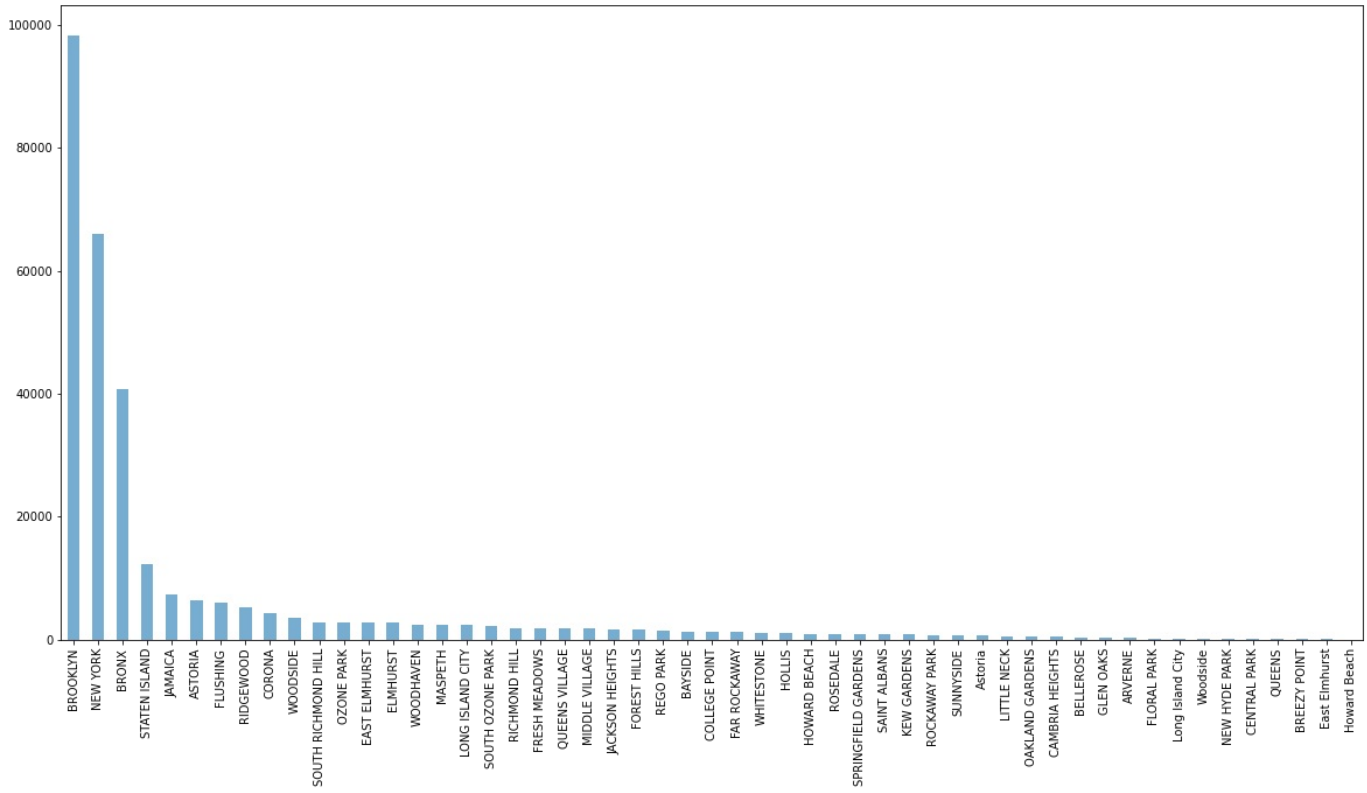
```
df['Status'].value_counts().plot(kind='bar', alpha=0.8, figsize=(10,10))
# The status of closed complaints is nearly 300000.
```

Out[99]: <AxesSubplot:>



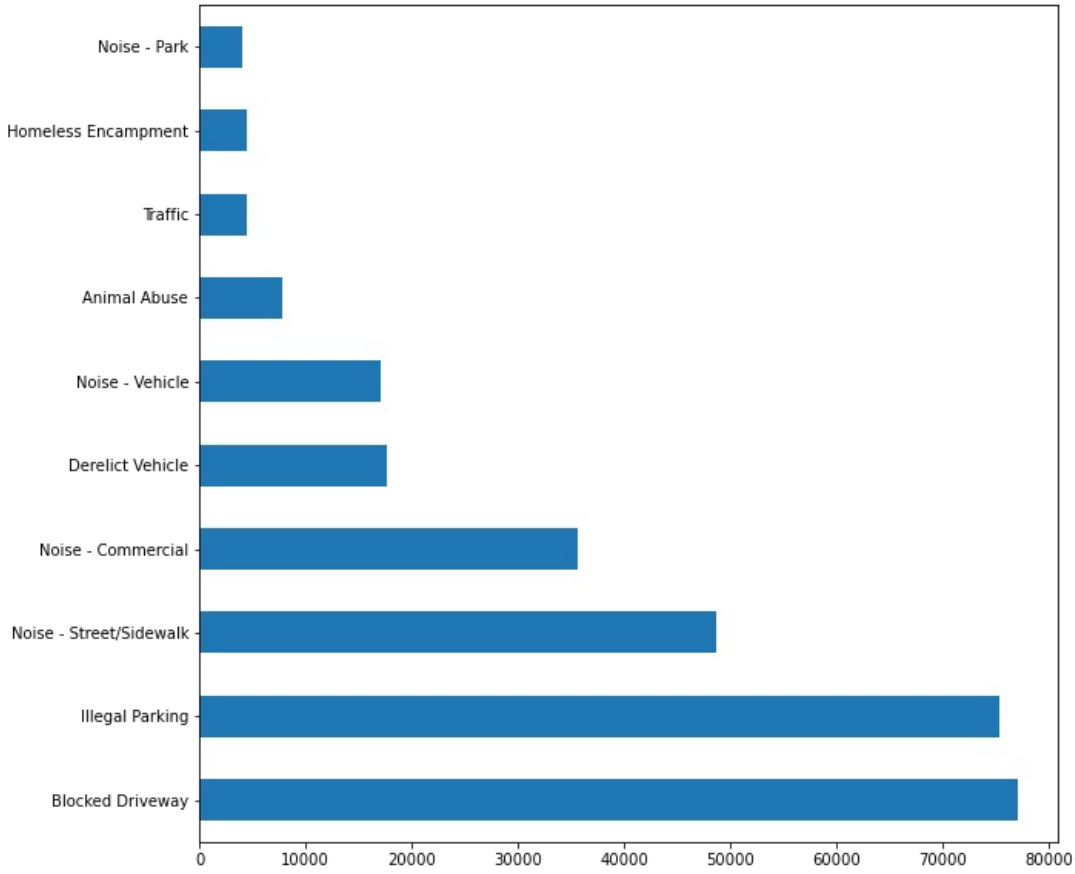

In [30]:
```
#Plottiting the count of complaint types of different cities using a bar graph.
df['City'].value_counts().plot(kind='bar', alpha=0.6, figsize=(20,10))
```

Out[30]: <AxesSubplot:>



In [31]:
```
#Plotting the count of different complain types in a bar graph
df['Complaint Type'].value_counts().head(10).plot(kind='barh',figsize=(10,10))
```

Out[31]: <AxesSubplot:>




In [32]:
```
#Analysing the count of complaint types with respect to each city.
df1 = pd.DataFrame({'count': df.groupby(['Complaint Type','City']).size()}).reset_index()
```

In [33]:
```
df1.head(50)
```

Out[33]:

| | Complaint Type | City | count |
|---|---|---|---|
| 0 | Animal Abuse | ARVERNE | 38 |
| 1 | Animal Abuse | ASTORIA | 125 |
| 2 | Animal Abuse | BAYSIDE | 37 |
| 3 | Animal Abuse | BELLEROSE | 7 |
| 4 | Animal Abuse | BREEZY POINT | 2 |
| 5 | Animal Abuse | BRONX | 1415 |
| 6 | Animal Abuse | BROOKLYN | 2394 |
| 7 | Animal Abuse | CAMBRIA HEIGHTS | 11 |
| 8 | Animal Abuse | COLLEGE POINT | 28 |
| 9 | Animal Abuse | CORONA | 61 |
| 10 | Animal Abuse | EAST ELMHURST | 59 |
| 11 | Animal Abuse | ELMHURST | 38 |
| 12 | Animal Abuse | FAR ROCKAWAY | 89 |
| 13 | Animal Abuse | FLORAL PARK | 2 |
| 14 | Animal Abuse | FLUSHING | 143 |
| 15 | Animal Abuse | FOREST HILLS | 45 |
| 16 | Animal Abuse | FRESH MEADOWS | 45 |
| 17 | Animal Abuse | GLEN OAKS | 5 |
| 18 | Animal Abuse | HOLLIS | 33 |
| 19 | Animal Abuse | HOWARD BEACH | 31 |
| 20 | Animal Abuse | JACKSON HEIGHTS | 42 |

| | | | |
|---|---|---|---|
| **21** | Animal Abuse | JAMAICA | 229 |
| **22** | Animal Abuse | KEW GARDENS | 19 |
| **23** | Animal Abuse | LITTLE NECK | 15 |
| **24** | Animal Abuse | LONG ISLAND CITY | 30 |
| **25** | Animal Abuse | MASPETH | 36 |
| **26** | Animal Abuse | MIDDLE VILLAGE | 22 |
| **27** | Animal Abuse | NEW HYDE PARK | 1 |
| **28** | Animal Abuse | NEW YORK | 1525 |
| **29** | Animal Abuse | OAKLAND GARDENS | 19 |
| **30** | Animal Abuse | OZONE PARK | 48 |
| **31** | Animal Abuse | QUEENS VILLAGE | 66 |
| **32** | Animal Abuse | REGO PARK | 26 |
| **33** | Animal Abuse | RICHMOND HILL | 32 |
| **34** | Animal Abuse | RIDGEWOOD | 117 |
| **35** | Animal Abuse | ROCKAWAY PARK | 30 |
| **36** | Animal Abuse | ROSEDALE | 33 |
| **37** | Animal Abuse | SAINT ALBANS | 30 |
| **38** | Animal Abuse | SOUTH OZONE PARK | 55 |
| **39** | Animal Abuse | SOUTH RICHMOND HILL | 26 |
| **40** | Animal Abuse | SPRINGFIELD GARDENS | 24 |
| **41** | Animal Abuse | STATEN ISLAND | 557 |
| **42** | Animal Abuse | SUNNYSIDE | 35 |
| **43** | Animal Abuse | WHITESTONE | 28 |
| **44** | Animal Abuse | WOODHAVEN | 45 |
| **45** | Animal Abuse | WOODSIDE | 69 |
| **46** | Animal in a Park | QUEENS | 1 |
| **47** | Bike/Roller/Skate Chronic | ASTORIA | 15 |
| **48** | Bike/Roller/Skate Chronic | BELLEROSE | 1 |
| **49** | Bike/Roller/Skate Chronic | BRONX | 20 |

# 4. Order the complaint types based on the average ‘Request_Closing_Time’, grouping them for different locations.

In [42]:
```
print(f'Unique number of "Location"s in the given dataset --> {len(df.Location.unique())}')
print(f'Unique number of "City"s in the given dataset --> {len(df.City.unique())}')
print(f'Unique number of "Borough"s in the given dataset --> {len(df.Borough.unique())}')
```

```
Unique number of "Location"s in the given dataset --> 126049
Unique number of "City"s in the given dataset --> 54
Unique number of "Borough"s in the given dataset --> 6
```

In [77]:
```
df.groupby(['Location','Complaint Type'])[['Request_Closing_Time']].mean().sort_values(by='Request_Closing_Time')
```

Out[77]:

| | | Request_Closing_Time |
|---|---|---|
| **Location** | **Complaint Type** | |
| **(40.678429539269835, -73.98361397723242)** | **Noise - Commercial** | 0.043611 |
| **(40.76848580086362, -73.91235250532725)** | **Noise - Vehicle** | 0.045278 |
| **(40.69371028050496, -73.95499211670034)** | **Illegal Parking** | 0.046389 |
| **(40.71598512070559, -73.9509008064274)** | **Illegal Parking** | 0.047500 |
| **(40.72895633655987, -74.00074325193769)** | **Noise - Commercial** | 0.050000 |
| **...** | **...** | ... |
| **(40.64465625507198, -73.95663234950166)** | **Derelict Vehicle** | 200.606250 |
| **(40.64466438582295, -73.95635848114169)** | **Derelict Vehicle** | 223.351667 |
| **(40.64496727639598, -73.95897801142875)** | **Derelict Vehicle** | 223.370000 |

| | | |
|---|---|---|
| **(40.68594971751218, -73.95942153955151)** | **Noise - Street/Sidewalk** | 297.859306 |
| **(40.59814521498835, -73.98935198928409)** | **Illegal Parking** | 577.360000 |

151508 rows × 1 columns

# 5. Perform a statistical test for the following:

# Please note: For the below statements you need to state the Null and Alternate and then provide a statistical test to accept # or reject the Null Hypothesis along with the corresponding ‘p-value’. # a. Whether the average response time across complaint types is similar or not (overall) Are the type of complaint or service # requested and location related?

In [ ]:
```
# Null Hypothesis (H0) - Avg Response Time across complaint types is similar.

# Alternate Hypothesis (Ha) - Avg Response Time across complaint types is NOT similar.

# Since we need to compare the avg response time across different categories, we will use the Anova One-way F-tes
# the given Null Hypothesis.
```

In [82]:
```
alpha = 0.05
```

In [83]:
```
len(df['Complaint Type'].unique())
```

Out[83]: 24

In [84]:
```
complaint_types_df = df.groupby('Complaint Type').mean()
complaint_types = complaint_types_df.index

print(complaint_types)
print(len(complaint_types))
```

```
Index(['Agency Issues', 'Animal Abuse', 'Animal in a Park',
       'Bike/Roller/Skate Chronic', 'Blocked Driveway', 'Derelict Vehicle',
       'Disorderly Youth', 'Drinking', 'Ferry Complaint', 'Graffiti',
       'Homeless Encampment', 'Illegal Fireworks', 'Illegal Parking',
       'Noise - Commercial', 'Noise - House of Worship', 'Noise - Park',
       'Noise - Street/Sidewalk', 'Noise - Vehicle', 'Panhandling',
       'Posting Advertisement', 'Squeegee', 'Traffic', 'Urinating in Public',
       'Vending'],
      dtype='object', name='Complaint Type')
24
```

In [85]:
```
service_requests_stats = df[['Complaint Type', 'Request_Closing_Time']]
service_requests_stats
```

Out[85]:

| | Complaint Type | Request_Closing_Time |
|---|---|---|
| **0** | Noise - Street/Sidewalk | 0.920833 |
| **1** | Blocked Driveway | 1.437778 |
| **2** | Blocked Driveway | 4.858611 |
| **3** | Illegal Parking | 7.753889 |
| **4** | Illegal Parking | 3.450556 |
| **...** | ... | ... |
| **300693** | Noise - Commercial | NaN |
| **300694** | Blocked Driveway | 2.008611 |
| **300695** | Noise - Commercial | 3.121389 |
| **300696** | Noise - Commercial | 4.092500 |
| **300697** | Noise - Commercial | 4.146944 |

300698 rows × 2 columns

In [86]:
```
complaint_samples = []
```

```
sample_size = 300

for i in range(len(complaint_types)):
    complaint_pop = service_requests_stats[service_requests_stats['Complaint Type'] == complaint_types[i]]
    if(len(complaint_pop)) < 2 * sample_size:
        continue
    complaint_times_sample =  np.random.choice(complaint_pop['Request_Closing_Time'].dropna(),size=sample_size,re
    complaint_samples.append(complaint_times_sample)
```

In [67]:
```
from scipy.stats import f_oneway
```

In [87]:
```
f_statistic,p_value = f_oneway(*complaint_samples)
```

In [88]:
```
print(f'The f-statistic from the One-way Anova test is {f_statistic} and the corresponding p-value is {p_value}')
```

```
The f-statistic from the One-way Anova test is 18.09283609930872 and the corresponding p-value is 1.3509123867431
636e-41
```

In [90]:
```
# Since the p-value(1.3509123867431636e-41) is less than the level of significance (0.05),
# the Null Hypothesis (H0) can be rejected.

# The Null Hypothesis that "the average response times across Complaint types are similar" can be rejected.
# And the alternative hypothesis that "the average response time across Complaint types are different" should be
```

# b. Are the type of complaint or service requested and location related?

In [ ]:
```
# Null Hypothesis (H0) - There is no association between Complaint Type and Location
# Alternative Hypothesis (Ha) - There is association between Complaint Type and Location
```

In [92]:
```
pd.crosstab(df['Complaint Type'], df['Location'])
```

Out[92]:

| Location | (40.49913462101514, -74.24348482977875) | (40.49967332981336, -74.2379063249761) | (40.49994886080869, -74.23740031497493) | (40.49999700116009, -74.23801175120917) | (40.50002168207532, -74.23802262609722) | (40.50004910779944, -74.238033510764) |
|---|---|---|---|---|---|---|
| **Complaint Type** | | | | | | |
| **Animal Abuse** | 0 | 0 | 0 | 0 | 0 | 2 |
| **Bike/Roller/Skate Chronic** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Blocked Driveway** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Derelict Vehicle** | 0 | 1 | 0 | 0 | 0 | 0 |
| **Disorderly Youth** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Drinking** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Graffiti** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Homeless Encampment** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Illegal Fireworks** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Illegal Parking** | 1 | 0 | 1 | 1 | 1 | 0 |
| **Noise - Commercial** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Noise - House of Worship** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Noise - Park** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Noise - Street/Sidewalk** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Noise - Vehicle** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Panhandling** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Posting Advertisement** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Squeegee** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Traffic** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Urinating in Public** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Vending** | 0 | 0 | 0 | 0 | 0 | 0 |

21 rows × 126048 columns

In [93]:
```
from scipy.stats import chi2_contingency
```

In [94]:
```
chi2_contingency(pd.crosstab(df['Complaint Type'], df['Location']))
```

Out[94]:
```
(4161473.224993255,
 0.0,
 2520940,
 array([[0.02607031, 0.02607031, 0.02607031, ..., 0.02607031, 0.02607031,
         0.10428122],
        [0.0013932 , 0.0013932 , 0.0013932 , ..., 0.0013932 , 0.0013932 ,
         0.00557279],
        [0.25818925, 0.25818925, 0.25818925, ..., 0.25818925, 0.25818925,
         1.03275698],
        ...,
        [0.01505933, 0.01505933, 0.01505933, ..., 0.01505933, 0.01505933,
         0.06023731],
        [0.00199221, 0.00199221, 0.00199221, ..., 0.00199221, 0.00199221,
         0.00796882],
        [0.01270704, 0.01270704, 0.01270704, ..., 0.01270704, 0.01270704,
         0.05082818]]))
```

In [95]:
```
# Since the p-value(0.0) is less than the level of significance (0.05), the Null Hypothesis (H0) can be rejected.
# The null hypothesis - There is no association between Complaint Type and Location - can be rejected.
# There is a relationship between Complaint Type and Location.
```