

1. Import a 311 NYC service request.

In [1]:

```
# importing required libraries

import numpy as np
import pandas as pd
from pandas import Series
import datetime as dt #to convert Created Date and Closed Date to datetime format
from datetime import datetime
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:

```
#importing the data set

df_Service_Requests = pd.read_csv("C:\\Venkat\\Data Science with Python - SimpliLearn\\Data Science with Python Two\\311_Service_Requests_from_2010_to_Present.csv")
```

C:\Users\vkakarla\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3058: DtypeWarning: Columns (48,49) have mixed types. Specify dtype option on import or set low_memory=False.
interactivity=interactivity, compiler=compiler, result=result)

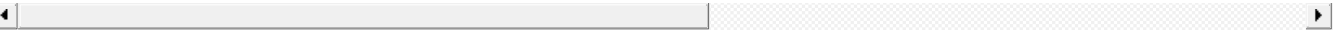
In [3]:

```
#viewing the first 5 records
df_Service_Requests.head()
```

Out[3]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name
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
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
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
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
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

5 rows × 53 columns



In [4]:

```
# Setting options to display all columns
pd.set_option('display.max_columns', None)
df_Service_Requests.head()
```

Out[4]:

Unique	Created	Closed	Agency	Complaint	Incident	Incident	Street
--------	---------	--------	--------	-----------	----------	----------	--------

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Street Name
	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Street Name
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	VERMILYEA AVENUE
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	23 AVENUE
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	VALENTINE AVENUE
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	BAISLEY AVENUE
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	57 ROAD

In [5]:

```
# Some of the columns contains "Unspecified" as value.
# Replacing all those vallues with NaN, so that we can get actual null counts.

df_Service_Requests.replace('Unspecified', np.NaN, inplace=True)
df_Service_Requests.head()
```

Out[5]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Street Name
	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Street Name
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	VERMILYEA AVENUE
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	23 AVENUE
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	VALENTINE AVENUE
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	BAISLEY AVENUE
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	57 ROAD

In [6]:

```
df_Service_Requests.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 53 columns):
Unique Key                300698 non-null int64
Created Date              300698 non-null object
Closed Date              298534 non-null object
Agency                  300698 non-null object
Agency Name             300698 non-null object
Complaint Type           300698 non-null object
Descriptor               294784 non-null object
Location Type            300567 non-null object
Incident Zip             298083 non-null float64
Incident Address         256288 non-null object
Street Name              256288 non-null object
Cross Street 1           251419 non-null object
Cross Street 2           250919 non-null object
Intersection Street 1    43858 non-null object
Intersection Street 2    43362 non-null object
Address Type             297883 non-null object
City                    298084 non-null object
Landmark                 349 non-null object
Facility Type            298527 non-null object
Status                   300698 non-null object
Due Date                 300695 non-null object
Resolution Description    300698 non-null object
Resolution Action Updated Date 298511 non-null object
Community Board          300698 non-null object
Borough                  298124 non-null object
X Coordinate (State Plane) 297158 non-null float64
Y Coordinate (State Plane) 297158 non-null float64
Park Facility Name       1 non-null object
Park Borough             298124 non-null object
School Name              1 non-null object
School Number            1 non-null object
School Region            0 non-null float64
School Code              0 non-null float64
School Phone Number      1 non-null object
School Address           1 non-null object
School City              1 non-null object
School State             1 non-null object
School Zip               0 non-null float64
School Not Found         300698 non-null object
School or Citywide Complaint 0 non-null float64
Vehicle Type             0 non-null float64
Taxi Company Borough     0 non-null float64
Taxi Pick Up Location    0 non-null float64
Bridge Highway Name      243 non-null object
Bridge Highway Direction 243 non-null object
Road Ramp                213 non-null object
Bridge Highway Segment   213 non-null object
Garage Lot Name          0 non-null float64
Ferry Direction          1 non-null object
Ferry Terminal Name      2 non-null object
Latitude                 297158 non-null float64
Longitude                297158 non-null float64
Location                 297158 non-null object
dtypes: float64(13), int64(1), object(39)
memory usage: 121.6+ MB

```

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. (Hint: Explore the package/module datetime)

In [7]:

```

#Checking the type of Created Date and Closed Date

print('Created Date data type is', df_Service_Requests['Created Date'].dtype)
print('Closed Date data type is', df_Service_Requests['Closed Date'].dtype)

```

Created Date data type is object
Closed Date data type is object

In [8]:

```
#Converting the columns 'Created Date' and Closed Date' to datetime datatype

df_Service_Requests['Created Date'] = pd.to_datetime(df_Service_Requests['Created Date'])
df_Service_Requests['Closed Date'] = pd.to_datetime(df_Service_Requests['Closed Date'])
```

In [9]:

```
#Checking the type of Created Date and Closed Date after conversion

print('Created Date data type is', df_Service_Requests['Created Date'].dtype)
print('Closed Date data type is', df_Service_Requests['Closed Date'].dtype)
```

Created Date data type is datetime64[ns]
Closed Date data type is datetime64[ns]

In [10]:

```
# creating a new column 'Request_Closing_Time' as the time elapsed between request creation and request closing

# Adding new column next to 'Closed Date' column
df_Service_Requests.insert(3, 'Request Closing Time', '')

# passing values to the new column

df_Service_Requests['Request Closing Time'] = (df_Service_Requests['Closed Date'] - df_Service_Requests['Created Date']) / pd.Timedelta(hours=1)

#Creating another new column 'Request_Closing_Time_dt' as the time elapsed between request creation and request closing as datetime column

df_Service_Requests.insert(4, 'Request_Closing_Time_dt', '')
df_Service_Requests['Request_Closing_Time_dt'] = df_Service_Requests['Closed Date'] - df_Service_Requests['Created Date']

#checking the new dataset
df_Service_Requests.head()
```

Out[10]:

	Unique Key	Created Date	Closed Date	Request Closing Time	Request_Closing_Time_dt	Agency	Agency Name	Complaint Type	Descriptor	Location Type
0	32310363	2015-12-31 23:59:45	2016-01-01 00:55:00	0.920833	00:55:15	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk
1	32309934	2015-12-31 23:59:44	2016-01-01 01:26:00	1.437778	01:26:16	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk
2	32309159	2015-12-31 23:59:29	2016-01-01 04:51:00	4.858611	04:51:31	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk
3	32305098	2015-12-31 23:57:46	2016-01-01 07:43:00	7.753889	07:45:14	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk

	Unique Key	Created Date	Closed Date	Request Closing Time	Request_Closing_Time_dt	Agency	Agency Name	Complaint Type	Descriptor	Location Type
4	3230652	2019-12-29 23:56:58	2019-01-01 03:24:00	3.45	03:27:02	NYPD	New York City Department of Transportation	Illegal Parking	Blocked Sidewalk	Street/Sidewalk

In [11]:

```
#Checking the dataset shape and size
```

```
print(df_Service_Requests.shape)
print(df_Service_Requests.size)
df_Service_Requests.describe()
```

(300698, 55)
16538390

Out[11]:

	Unique Key	Request Closing Time	Request_Closing_Time_dt	Incident Zip	X Coordinate (State Plane)	Y Coordinate (State Plane)	School Region	School Code	School Zip
count	3.006980e+05	298534.000000	298534	298083.000000	2.971580e+05	297158.000000	0.0	0.0	0.0
mean	3.130054e+07	4.314398	0 days 04:18:51.832782	10848.888645	1.004854e+06	203754.534416	NaN	NaN	NaN
std	5.738547e+05	6.089484	0 days 06:05:22.141833	583.182081	2.175338e+04	29880.183529	NaN	NaN	NaN
min	3.027948e+07	0.016667	0 days 00:01:00	83.000000	9.133570e+05	121219.000000	NaN	NaN	NaN
25%	3.080118e+07	1.275833	0 days 01:16:33	10310.000000	9.919752e+05	183343.000000	NaN	NaN	NaN
50%	3.130436e+07	2.715417	0 days 02:42:55.500000	11208.000000	1.003158e+06	201110.500000	NaN	NaN	NaN
75%	3.178446e+07	5.350000	0 days 05:21:00	11238.000000	1.018372e+06	224125.250000	NaN	NaN	NaN
max	3.231065e+07	592.872778	24 days 16:52:22	11697.000000	1.067173e+06	271876.000000	NaN	NaN	NaN

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 [12]:

```
# Checking for null values if any
df[ServiceRequests.isnull().any()]
```

Out[12]:

Unique Key	False
Created Date	False
Closed Date	True
Request Closing Time	True
Request_Closing_Time_dt	True
Agency	False
Agency Name	False
Complaint Type	False
Descriptor	True
Location Type	True
Incident Zip	True
Incident Address	True
Street Name	True
Cross Street 1	True
Cross Street 2	True
Intersection Street 1	True
Intersection Street 2	True
Address Type	True
City	True
Landmark	True
Facility Type	True
Status	False
Due Date	True

```

Closed Date      True
Resolution Description      False
Resolution Action Updated Date      True
Community Board      False
Borough      True
X Coordinate (State Plane)      True
Y Coordinate (State Plane)      True
Park Facility Name      True
Park Borough      True
School Name      True
School Number      True
School Region      True
School Code      True
School Phone Number      True
School Address      True
School City      True
School State      True
School Zip      True
School Not Found      False
School or Citywide Complaint      True
Vehicle Type      True
Taxi Company Borough      True
Taxi Pick Up Location      True
Bridge Highway Name      True
Bridge Highway Direction      True
Road Ramp      True
Bridge Highway Segment      True
Garage Lot Name      True
Ferry Direction      True
Ferry Terminal Name      True
Latitude      True
Longitude      True
Location      True
dtype: bool

```

In [13]:

```

#Checking the counts of null for ech column
df_Service_Requests_Counts = pd.DataFrame(df_Service_Requests.isnull().sum(), columns=['Null_Count'])
df_Service_Requests_Counts.insert(0, 'Field_Names', '')
df_Service_Requests_Counts['Field_Names'] = df_Service_Requests.columns
df_Service_Requests_Counts["Total_Rows"]=300698
df_Service_Requests_Counts["Null_Percentage"]=df_Service_Requests_Counts["Null_Count"]/df_Service_Requests_Counts["Total_Rows"]*100
df_Service_Requests_Counts

```

Out[13]:

	Field_Names	Null_Count	Total_Rows	Null_Percentage
Unique Key	Unique Key	0	300698	0.000000
Created Date	Created Date	0	300698	0.000000
Closed Date	Closed Date	2164	300698	0.719659
Request Closing Time	Request Closing Time	2164	300698	0.719659
Request_Closing_Time_dt	Request_Closing_Time_dt	2164	300698	0.719659
Agency	Agency	0	300698	0.000000
Agency Name	Agency Name	0	300698	0.000000
Complaint Type	Complaint Type	0	300698	0.000000
Descriptor	Descriptor	5914	300698	1.966757
Location Type	Location Type	131	300698	0.043565
Incident Zip	Incident Zip	2615	300698	0.869643
Incident Address	Incident Address	44410	300698	14.768971
Street Name	Street Name	44410	300698	14.768971
Cross Street 1	Cross Street 1	49279	300698	16.388203
Cross Street 2	Cross Street 2	49779	300698	16.554483
Intersection Street 1	Intersection Street 1	256840	300698	85.414602
Intersection Street 2	Intersection Street 2	257822	300698	85.775755

Intersection Street 2	Intersection Street 2	257336	300698	85.579552
Field Names	Field Names	Null_Count	Total_Rows	Null_Percentage
Address Type	Address Type	2815	300698	0.936155
City	City	2614	300698	0.869311
Landmark	Landmark	300349	300698	99.883937
Facility Type	Facility Type	2171	300698	0.721987
Status	Status	0	300698	0.000000
Due Date	Due Date	3	300698	0.000998
Resolution Description	Resolution Description	0	300698	0.000000
Resolution Action Updated Date	Resolution Action Updated Date	2187	300698	0.727308
Community Board	Community Board	0	300698	0.000000
Borough	Borough	2574	300698	0.856008
X Coordinate (State Plane)	X Coordinate (State Plane)	3540	300698	1.177261
Y Coordinate (State Plane)	Y Coordinate (State Plane)	3540	300698	1.177261
Park Facility Name	Park Facility Name	300697	300698	99.999667
Park Borough	Park Borough	2574	300698	0.856008
School Name	School Name	300697	300698	99.999667
School Number	School Number	300697	300698	99.999667
School Region	School Region	300698	300698	100.000000
School Code	School Code	300698	300698	100.000000
School Phone Number	School Phone Number	300697	300698	99.999667
School Address	School Address	300697	300698	99.999667
School City	School City	300697	300698	99.999667
School State	School State	300697	300698	99.999667
School Zip	School Zip	300698	300698	100.000000
School Not Found	School Not Found	0	300698	0.000000
School or Citywide Complaint	School or Citywide Complaint	300698	300698	100.000000
Vehicle Type	Vehicle Type	300698	300698	100.000000
Taxi Company Borough	Taxi Company Borough	300698	300698	100.000000
Taxi Pick Up Location	Taxi Pick Up Location	300698	300698	100.000000
Bridge Highway Name	Bridge Highway Name	300455	300698	99.919188
Bridge Highway Direction	Bridge Highway Direction	300455	300698	99.919188
Road Ramp	Road Ramp	300485	300698	99.929165
Bridge Highway Segment	Bridge Highway Segment	300485	300698	99.929165
Garage Lot Name	Garage Lot Name	300698	300698	100.000000
Ferry Direction	Ferry Direction	300697	300698	99.999667
Ferry Terminal Name	Ferry Terminal Name	300696	300698	99.999335
Latitude	Latitude	3540	300698	1.177261
Longitude	Longitude	3540	300698	1.177261
Location	Location	3540	300698	1.177261

In [14]:

```
#Checking the data info
df_Service_Requests.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 55 columns):
Unique Key                300698 non-null int64
Created Date              300698 non-null datetime64[ns]
Closed Date              298534 non-null datetime64[ns]
Request Closing Time      298534 non-null float64
Request Closing Time dt   298534 non-null timedelta64[ns]
```

```

Agency 300698 non-null object
Agency Name 300698 non-null object
Complaint Type 300698 non-null object
Descriptor 294784 non-null object
Location Type 300567 non-null object
Incident Zip 298083 non-null float64
Incident Address 256288 non-null object
Street Name 256288 non-null object
Cross Street 1 251419 non-null object
Cross Street 2 250919 non-null object
Intersection Street 1 43858 non-null object
Intersection Street 2 43362 non-null object
Address Type 297883 non-null object
City 298084 non-null object
Landmark 349 non-null object
Facility Type 298527 non-null object
Status 300698 non-null object
Due Date 300695 non-null object
Resolution Description 300698 non-null object
Resolution Action Updated Date 298511 non-null object
Community Board 300698 non-null object
Borough 298124 non-null object
X Coordinate (State Plane) 297158 non-null float64
Y Coordinate (State Plane) 297158 non-null float64
Park Facility Name 1 non-null object
Park Borough 298124 non-null object
School Name 1 non-null object
School Number 1 non-null object
School Region 0 non-null float64
School Code 0 non-null float64
School Phone Number 1 non-null object
School Address 1 non-null object
School City 1 non-null object
School State 1 non-null object
School Zip 0 non-null float64
School Not Found 300698 non-null object
School or Citywide Complaint 0 non-null float64
Vehicle Type 0 non-null float64
Taxi Company Borough 0 non-null float64
Taxi Pick Up Location 0 non-null float64
Bridge Highway Name 243 non-null object
Bridge Highway Direction 243 non-null object
Road Ramp 213 non-null object
Bridge Highway Segment 213 non-null object
Garage Lot Name 0 non-null float64
Ferry Direction 1 non-null object
Ferry Terminal Name 2 non-null object
Latitude 297158 non-null float64
Longitude 297158 non-null float64
Location 297158 non-null object
dtypes: datetime64[ns](2), float64(14), int64(1), object(37), timedelta64[ns](1)
memory usage: 126.2+ MB

```

In [15]:

```
#checking the column names/headers
```

```
df_Service_Requests.columns
```

Out[15]:

```

Index(['Unique Key', 'Created Date', 'Closed Date', 'Request Closing Time',
      'Request_Closing_Time_dt', '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',

```



```

    'Bridge Highway Segment', 'Garage Lot Name', 'Ferry Direction',
    'Ferry Terminal Name', 'Latitude', 'Longitude', 'Location'],
    dtype='object')

```

In [16]:

```

#Checking the index range of the dataset
df_Service_Requests.index

```

Out[16]:

```

RangeIndex(start=0, stop=300698, step=1)

```

In [17]:

```

#Describing the dataset
df_Service_Requests.describe(include = 'all')

```

Out[17]:

	Unique Key	Created Date	Closed Date	Request Closing Time	Request_Closing_Time_dt	Agency	Agency Name	Complaint Type	Descriptor	Location
count	3.006980e+05	300698	298534	298534.000000	298534	300698	300698	300698	294784	
unique	NaN	259493	237165	NaN	NaN	1	3	24	45	
top	NaN	2015-06-06 22:23:00	2015-11-08 07:34:00	NaN	NaN	NYPD	New York City Police Department	Blocked Driveway	Loud Music/Party	Street
freq	NaN	9	24	NaN	NaN	300698	300690	77044	61430	
first	NaN	2015-03-29 00:33:01	2015-03-29 00:57:23	NaN	NaN	NaN	NaN	NaN	NaN	
last	NaN	2015-12-31 23:59:45	2016-01-03 16:22:00	NaN	NaN	NaN	NaN	NaN	NaN	
mean	3.130054e+07	NaN	NaN	4.314398	0 days 04:18:51.832782	NaN	NaN	NaN	NaN	
std	5.738547e+05	NaN	NaN	6.089484	0 days 06:05:22.141833	NaN	NaN	NaN	NaN	
min	3.027948e+07	NaN	NaN	0.016667	0 days 00:01:00	NaN	NaN	NaN	NaN	
25%	3.080118e+07	NaN	NaN	1.275833	0 days 01:16:33	NaN	NaN	NaN	NaN	
50%	3.130436e+07	NaN	NaN	2.715417	0 days 02:42:55.500000	NaN	NaN	NaN	NaN	
75%	3.178446e+07	NaN	NaN	5.350000	0 days 05:21:00	NaN	NaN	NaN	NaN	
max	3.231065e+07	NaN	NaN	592.872778	24 days 16:52:22	NaN	NaN	NaN	NaN	

In [18]:

```

#checking the unique values of each columns
df_Service_Requests.nunique()

```

Out[18]:

```

Unique Key          300698
Created Date        259493
Closed Date         237165
Request Closing Time 47608
Request_Closing_Time_dt 47608
Agency              1
Agency Name         3
Complaint Type       24
Descriptor           45
Location Type        18
Incident Zip         201
Incident Address     107652

```

```

Street Name          7320
Cross Street 1       5982
Cross Street 2       5823
Intersection Street 1 4413
Intersection Street 2 4172
Address Type         5
City                 53
Landmark             116
Facility Type        1
Status               4
Due Date             259851
Resolution Description 18
Resolution Action Updated Date 237895
Community Board      75
Borough              5
X Coordinate (State Plane) 63226
Y Coordinate (State Plane) 73694
Park Facility Name   1
Park Borough         5
School Name          1
School Number        1
School Region        0
School Code          0
School Phone Number  1
School Address       1
School City          1
School State         1
School Zip           0
School Not Found     1
School or Citywide Complaint 0
Vehicle Type         0
Taxi Company Borough 0
Taxi Pick Up Location 0
Bridge Highway Name  29
Bridge Highway Direction 34
Road Ramp            2
Bridge Highway Segment 160
Garage Lot Name      0
Ferry Direction      1
Ferry Terminal Name  2
Latitude             125122
Longitude            125216
Location             126048
dtype: int64

```

In [19]:

```

# Dropping some columns that have more missing values and creating a new data frame
# Dropping columns with more than 70% missing values

df_Service_Requests_drop_columns = df_Service_Requests_Counts.loc[df_Service_Requests_Counts['Null_Percentage']>70,['Field_Names']]

df_Service_Requests_drop_columns['Field_Names'].values
df_Service_Requests2 = df_Service_Requests.copy()
print(df_Service_Requests.shape)
print(df_Service_Requests2.shape)
df_Service_Requests2.drop(df_Service_Requests_drop_columns['Field_Names'].values, axis=1,inplace =True)
#df_Service_Requests_Counts

# Let us remove drop the records if the columns "Created Date", "Closed Date" and "City" have any missing values

df_Service_Requests2.dropna(subset=['Created Date','Closed Date','City','Location Type'], inplace=True)

print(df_Service_Requests.shape)
print(df_Service_Requests2.shape)

```

```

(300698, 55)
(300698, 55)
(300698, 55)
(297906, 31)

```

In [20]:

In [20]:

```
non_category_columns = ['Created Date', 'Closed Date', 'Request Closing Time', 'Request_Closing_Time_dt']

for columns in df_Service_Requests2.columns:
    if df_Service_Requests2[columns].nunique() < 210 and columns not in non_category_columns:
        df_Service_Requests2[columns] = df_Service_Requests2[columns].astype('category')
```

In [21]:

```
#Checking the distinct values of few columns

print("Unique values in 'Agency Name' column are : \n",df_Service_Requests2['Agency Name'].unique(), '\n')
print("Unique values in 'Complaint Type' column are : \n",df_Service_Requests['Complaint Type'].unique(), '\n')
print("Unique values in 'Location Type' column are : \n",df_Service_Requests['Location Type'].unique(), '\n')
print("Unique values in 'City' column are : \n",df_Service_Requests['City'].unique(), '\n')
```

Unique values in 'Agency Name' column are :
[New York City Police Department]
Categories (1, object): [New York City Police Department]

Unique values in 'Complaint Type' column are :
['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']

Unique values in 'Location Type' column are :
['Street/Sidewalk' 'Club/Bar/Restaurant' 'Store/Commercial'
'House of Worship' 'Residential Building/House' 'Residential Building'
'Park/Playground' 'Vacant Lot' 'House and Store' 'Highway' 'Commercial'
'Roadway Tunnel' 'Subway Station' 'Parking Lot' 'Bridge' 'Terminal' nan
'Ferry' 'Park']

Unique values in 'City' column are :
['NEW YORK' 'ASTORIA' 'BRONX' 'ELMHURST' 'BROOKLYN' 'KEW GARDENS'
'JACKSON HEIGHTS' 'MIDDLE VILLAGE' 'REGO PARK' 'SAINT ALBANS' 'JAMAICA'
'SOUTH RICHMOND HILL' nan 'RIDGEWOOD' 'HOWARD BEACH' 'FOREST HILLS'
'STATEN ISLAND' 'OZONE PARK' 'RICHMOND HILL' 'WOODHAVEN' 'FLUSHING'
'CORONA' 'QUEENS VILLAGE' 'OAKLAND GARDENS' 'HOLLIS' 'MASPETH'
'EAST ELMHURST' 'SOUTH OZONE PARK' 'WOODSIDE' 'FRESH MEADOWS'
'LONG ISLAND CITY' 'ROCKAWAY PARK' 'SPRINGFIELD GARDENS' 'COLLEGE POINT'
'BAYSIDE' 'GLEN OAKS' 'FAR ROCKAWAY' 'BELLEROSE' 'LITTLE NECK'
'CAMBRIA HEIGHTS' 'ROSEDALE' 'SUNNYSIDE' 'WHITESTONE' 'ARVERNE'
'FLORAL PARK' 'NEW HYDE PARK' 'CENTRAL PARK' 'BREEZY POINT' 'QUEENS'
'Astoria' 'Long Island City' 'Woodside' 'East Elmhurst' 'Howard Beach']

Doing basic exploratory data analysis

In [22]:

```
# Checking complaint counts for different cities
complaint_dist = df_Service_Requests2.groupby('Complaint Type').City.count()
```

In [23]:

```
print(complaint_dist)
```

Complaint Type	
Animal Abuse	7764
Animal in a Park	1
Bike/Roller/Skate Chronic	422

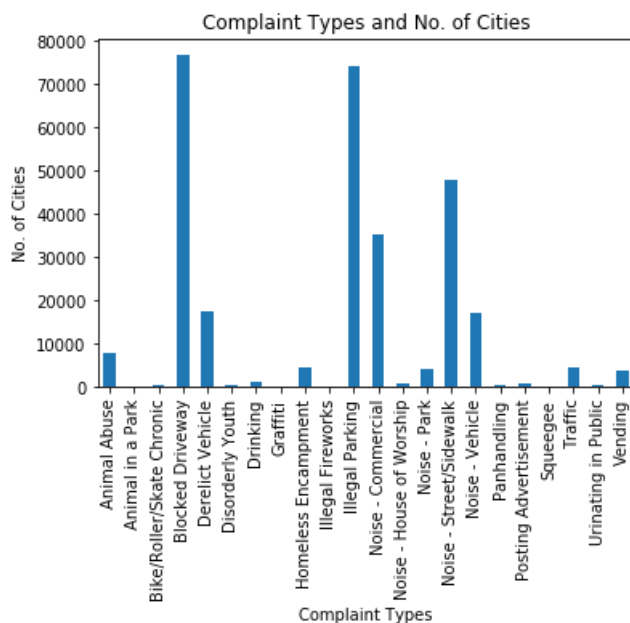
Blocked Driveway	76715
Derelict Vehicle	17534
Disorderly Youth	286
Drinking	1271
Graffiti	113
Homeless Encampment	4406
Illegal Fireworks	168
Illegal Parking	74276
Noise - Commercial	35180
Noise - House of Worship	927
Noise - Park	4014
Noise - Street/Sidewalk	47977
Noise - Vehicle	17024
Panhandling	305
Posting Advertisement	647
Squeegee	4
Traffic	4489
Urinating in Public	591
Vending	3792

Name: City, dtype: int64

In [24]:

```
# Graph to show diff Complaint Types and the count of cities under each Complaint Type

x=df_Service_Requests2.groupby('Complaint Type').City.count()
x.plot(kind="bar")
plt.xlabel("Complaint Types")
plt.ylabel("No. of Cities")
plt.title("Complaint Types and No. of Cities")
plt.show()
```



Conclusion 1:

More complaints are under the complaint type "Blocked Driveway" followed by "Illegal Parking" and "Noise - Street/Sidewalk"

In [25]:

```
# # Count plot

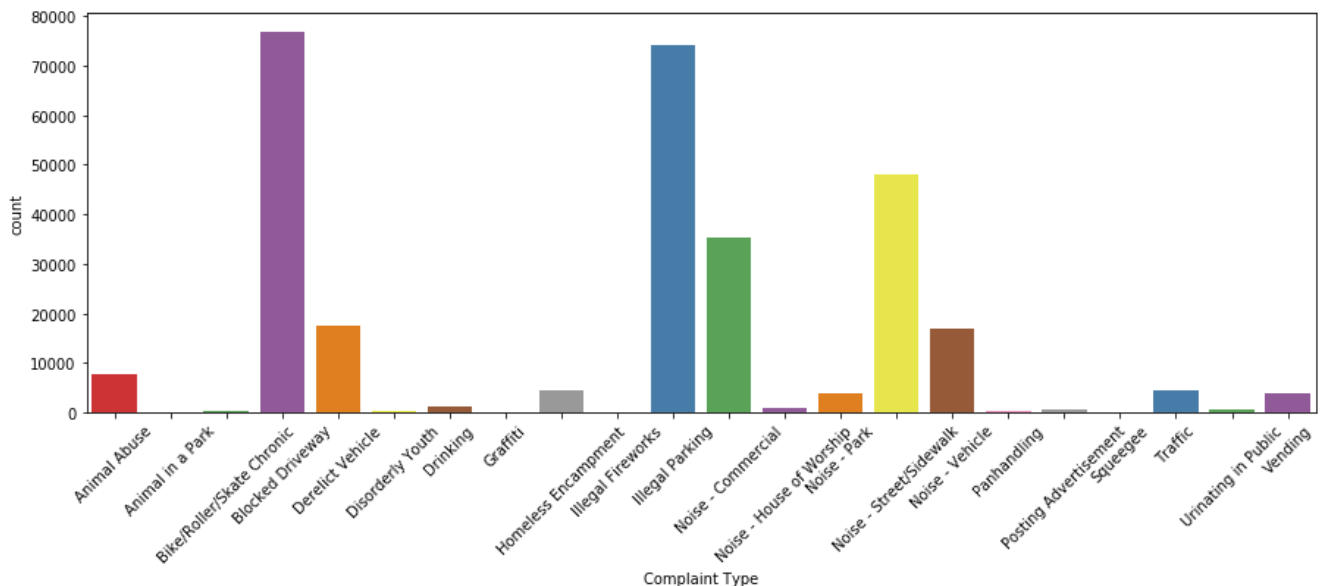
# only apply with categorical values
# give the count details
# univariate

sns.countplot(x='Complaint Type',data=df_Service_Requests2).set_xticklabels(get_xticklabels(), rotation=45)
```

```
plt.figure(figsize=(15,5))
chart = sns.countplot(x='Complaint Type',data=df_Service_Requests2,palette='Set1')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
```

Out[25]:

```
[Text(0, 0, 'Animal Abuse'),
Text(0, 0, 'Animal in a Park'),
Text(0, 0, 'Bike/Roller/Skate Chronic'),
Text(0, 0, 'Blocked Driveway'),
Text(0, 0, 'Derelict Vehicle'),
Text(0, 0, 'Disorderly Youth'),
Text(0, 0, 'Drinking'),
Text(0, 0, 'Graffiti'),
Text(0, 0, 'Homeless Encampment'),
Text(0, 0, 'Illegal Fireworks'),
Text(0, 0, 'Illegal Parking'),
Text(0, 0, 'Noise - Commercial'),
Text(0, 0, 'Noise - House of Worship'),
Text(0, 0, 'Noise - Park'),
Text(0, 0, 'Noise - Street/Sidewalk'),
Text(0, 0, 'Noise - Vehicle'),
Text(0, 0, 'Panhandling'),
Text(0, 0, 'Posting Advertisement'),
Text(0, 0, 'Squeegee'),
Text(0, 0, 'Traffic'),
Text(0, 0, 'Urinating in Public'),
Text(0, 0, 'Vending')]
```



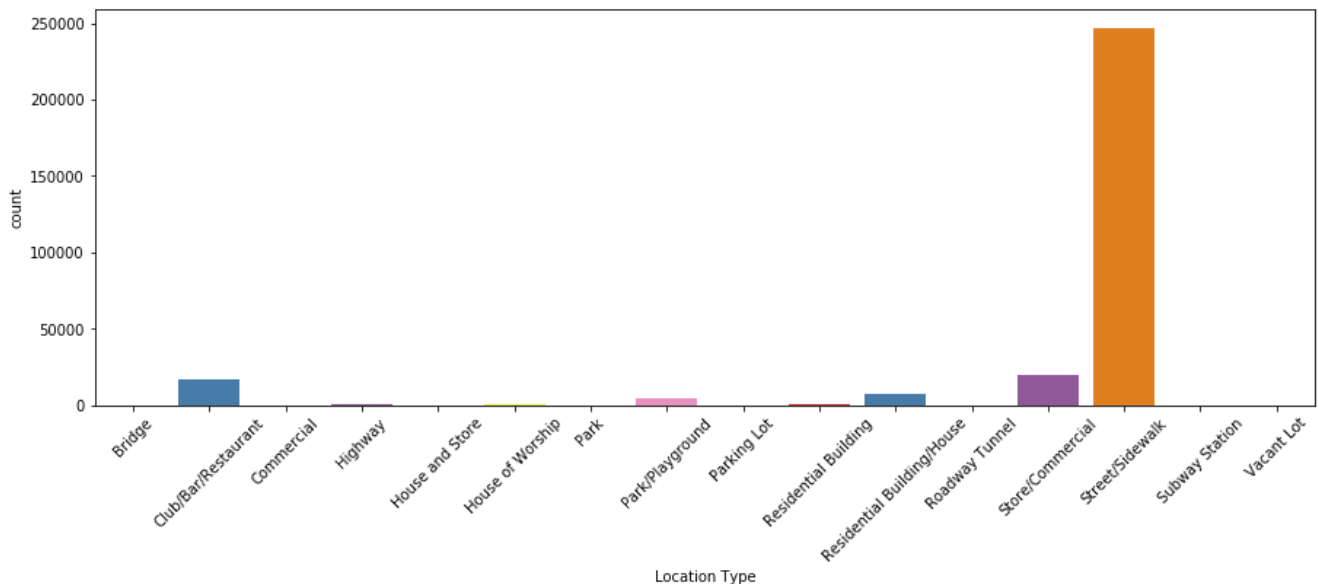
In [26]:

```
plt.figure(figsize=(15,5))
chart = sns.countplot(x='Location Type',data=df_Service_Requests2,palette='Set1')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
```

Out[26]:

```
[Text(0, 0, 'Bridge'),
Text(0, 0, 'Club/Bar/Restaurant'),
Text(0, 0, 'Commercial'),
Text(0, 0, 'Highway'),
Text(0, 0, 'House and Store'),
Text(0, 0, 'House of Worship'),
Text(0, 0, 'Park'),
Text(0, 0, 'Park/Playground'),
Text(0, 0, 'Parking Lot'),
Text(0, 0, 'Residential Building'),
Text(0, 0, 'Residential Building/House'),
Text(0, 0, 'Roadway Tunnel'),
Text(0, 0, 'Store/Commercial'),
Text(0, 0, 'Street/Sidewalk'),
Text(0, 0, 'Subway Station'),
```

```
Text(0, 0, 'Vacant Lot')]
```



In [27]:

```
# No of Complaints by Location Type
```

```
location_dist = df_Service_Requests2.groupby('Location Type')['City'].count()
```

```
location_count = df_Service_Requests2['Location Type'].count()
```

```
location_st_sw_count = location_dist['Street/Sidewalk']
```

```
#print(location_dist)
```

```
print("Total Complaints are: ", location_count)
```

```
print("No. of Complaints from the Location Type 'Street/Sidewalk' are: ", location_st_sw_count)
```

```
print("Percentage of Complaints from the Location Type 'Street/Sidewalk' is {:.2%}".format(location_st_sw_count/location_count))
```

Total Complaints are: 297906

No. of Complaints from the Location Type 'Street/Sidewalk' are: 247076

Percentage of Complaints from the Location Type 'Street/Sidewalk' is 82.94%

Conclusion 2:

More than 80% complaints are under the Location Type = 'Street/Sidewalk'

In [28]:

```
# City wise complaints distribution
```

```
city_dist = pd.DataFrame(df_Service_Requests2.groupby('City')['Complaint Type'].count())
```

```
#df_Service_Requests2.columns
```

```
city_dist.rename(columns={'Complaint Type': 'Complaints_Count'}, inplace=True)
```

```
city_dist['Complaints %'] = city_dist['Complaints_Count'] / city_dist['Complaints_Count'].sum() * 100
```

```
city_dist
```

Out[28]:

	Complaints_Count	Complaints %
City		
ARVERNE	220	0.073849
ASTORIA	6330	2.124831
Astoria	716	0.240344
BAYSIDE	1218	0.408854

BELLEROSE	Complaints_Count	Complaints%
BREEZY POINT	30	0.010070
BRONX	40683	13.656321
BROOKLYN	98251	32.980537
CAMBRIA HEIGHTS	477	0.160118
CENTRAL PARK	97	0.032561
COLLEGE POINT	1219	0.409189
CORONA	4293	1.441059
EAST ELMHURST	2731	0.916732
ELMHURST	2672	0.896927
East Elmhurst	14	0.004699
FAR ROCKAWAY	1179	0.395762
FLORAL PARK	152	0.051023
FLUSHING	5965	2.002309
FOREST HILLS	1688	0.566622
FRESH MEADOWS	1897	0.636778
GLEN OAKS	306	0.102717
HOLLIS	1009	0.338697
HOWARD BEACH	930	0.312179
Howard Beach	1	0.000336
JACKSON HEIGHTS	1688	0.566622
JAMAICA	7286	2.445738
KEW GARDENS	771	0.258806
LITTLE NECK	559	0.187643
LONG ISLAND CITY	2434	0.817036
Long Island City	134	0.044981
MASPETH	2460	0.825764
MIDDLE VILLAGE	1765	0.592469
NEW HYDE PARK	96	0.032225
NEW YORK	65957	22.140205
OAKLAND GARDENS	551	0.184958
OZONE PARK	2752	0.923781
QUEENS	32	0.010742
QUEENS VILLAGE	1814	0.608917
REGO PARK	1486	0.498815
RICHMOND HILL	1902	0.638456
RIDGEWOOD	5162	1.732761
ROCKAWAY PARK	745	0.250079
ROSEDALE	922	0.309494
SAINT ALBANS	834	0.279954
SOUTH OZONE PARK	2173	0.729425
SOUTH RICHMOND HILL	2774	0.931166
SPRINGFIELD GARDENS	882	0.296067
STATEN ISLAND	12332	4.139561
SUNNYSIDE	723	0.242694
WHITESTONE	1098	0.368573
WOODHAVEN	2461	0.826100
WOODSIDE	3541	1.188630
WYCKHOF	100	0.003300

In [29]:

```
print("No of. Cities that have more than 1% of the overall complaints are"
      , city_dist.loc[city_dist["Complaints %"]>1,["Complaints %"]].count().values, end="")
print(" with a share of complaint percent"
      ,city_dist.loc[city_dist["Complaints %"]>1,["Complaints %"]].sum().values)

print("No of. Cities that have less than 1% of the overall complaints are"
      , city_dist.loc[city_dist["Complaints %"]<=1,["Complaints %"]].count().values,end="")
print(" with a share of complaint percent"
      ,city_dist.loc[city_dist["Complaints %"]<=1,["Complaints %"]].sum().values)
```

No of. Cities that have more than 1% of the overall complaints are [10] with a share of complaint percent [83.8519533]
No of. Cities that have less than 1% of the overall complaints are [43] with a share of complaint percent [16.1480467]

Conclusion 3:

Approximately 20%the Cities account for more than 80%of the complaints

Rest of the 80%cities account for less than 20%of the complaints

Major complaints are from BROOKLYN (32.9%) followed by NEW YORK (22.1%) and BRONX (13.6%)

In [30]:

```
df_Service_Requests2.groupby('Agency Name')['Complaint Type'].count()
```

Out[30]:

```
Agency Name
New York City Police Department    297906
Name: Complaint Type, dtype: int64
```

Conclusion 4:

All most all the complaints are under the agency 'New York City Police Department'

In [31]:

```
df_Service_Requests2.groupby('Address Type')['Complaint Type'].count()
```

Out[31]:

```
Address Type
ADDRESS      238507
BLOCKFACE    12011
INTERSECTION 43339
LATLONG      3460
PLACENAME    350
Name: Complaint Type, dtype: int64
```

In [32]:

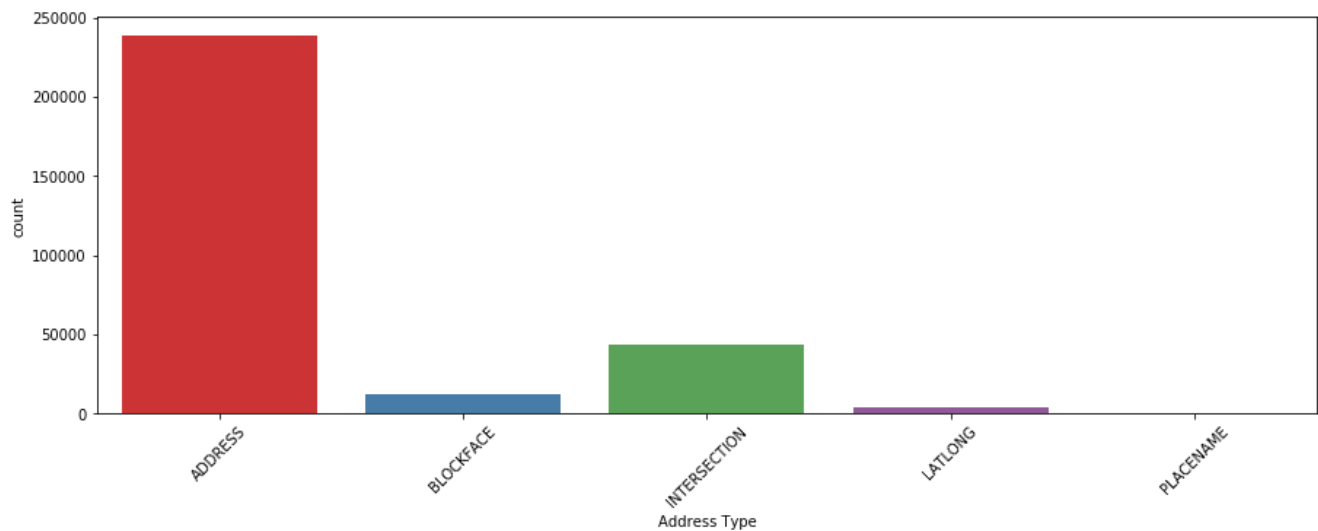
```
plt.figure(figsize=(15,5))
chart = sns.countplot(x='Address Type',data=df_Service_Requests2,palette='Set1')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
```

Out[32]:

```
[Text(0, 0, 'ADDRESS'),
 Text(0, 0, 'BLOCKFACE'),
 Text(0, 0, 'INTERSECTION'),
```



```
Text(0, 0, 'LATLONG'),  
Text(0, 0, 'PLACENAME')]
```



In [33]:

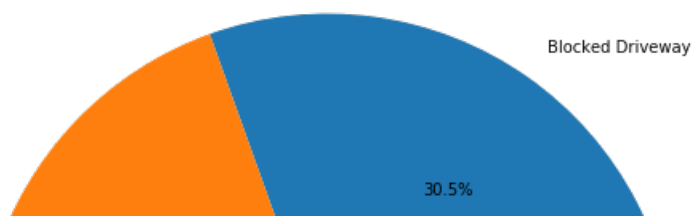
```
Top_Complaints=df_Service_Requests2.dropna(subset=["Complaint Type"])  
Top_Complaints=df_Service_Requests2.groupby("Complaint Type")  
Top_Complaints2 = Top_Complaints.size().sort_values(ascending = False)  
Top_Complaints2 = Top_Complaints2.to_frame('count').reset_index()  
  
Top_Complaints2  
Top_Complaints2.head(10)
```

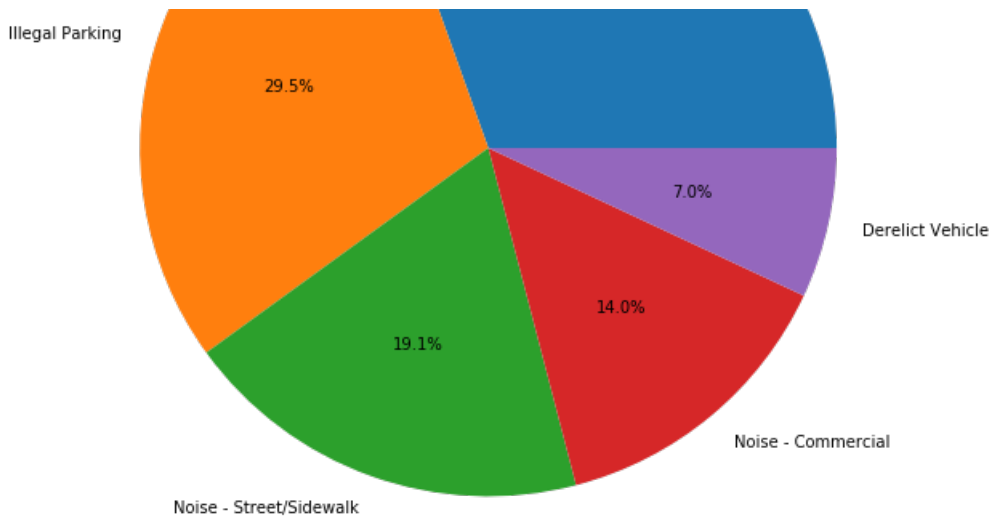
Out[33]:

	Complaint Type	count
0	Blocked Driveway	76715
1	Illegal Parking	74276
2	Noise - Street/Sidewalk	47977
3	Noise - Commercial	35180
4	Derelict Vehicle	17534
5	Noise - Vehicle	17024
6	Animal Abuse	7764
7	Traffic	4489
8	Homeless Encampment	4406
9	Noise - Park	4014

In [34]:

```
Top_Complaints2 = Top_Complaints2.head()  
plt.figure(figsize=(10,10))  
plt.pie(Top_Complaints2['count'],labels=Top_Complaints2["Complaint Type"], autopct="%1.1f%%")  
plt.show()
```





In [35]:

```
# Looking at No. of Complaints per each Borough
```

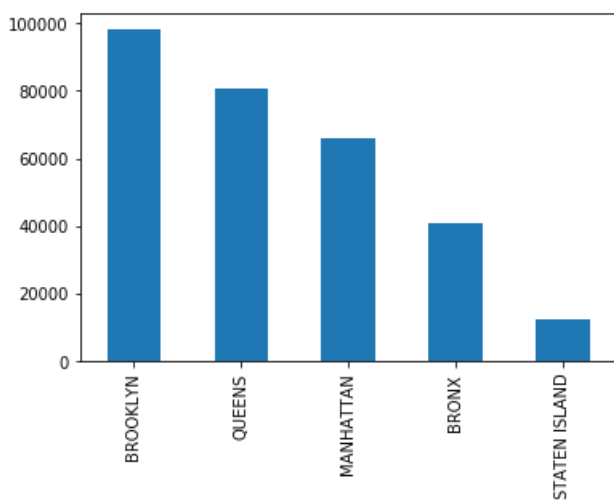
```
df_Service_Requests2['Borough'].value_counts().plot('bar')
```

C:\Users\vkakarla\anaconda3\lib\site-packages\ipykernel_launcher.py:3: FutureWarning: `Series.plot()` should not be called with positional arguments, only keyword arguments. The order of positional arguments will change in the future. Use `Series.plot(kind='bar')` instead of `Series.plot('bar',)`.

This is separate from the ipykernel package so we can avoid doing imports until

Out[35]:

<matplotlib.axes._subplots.AxesSubplot at 0x1b5020d3b88>

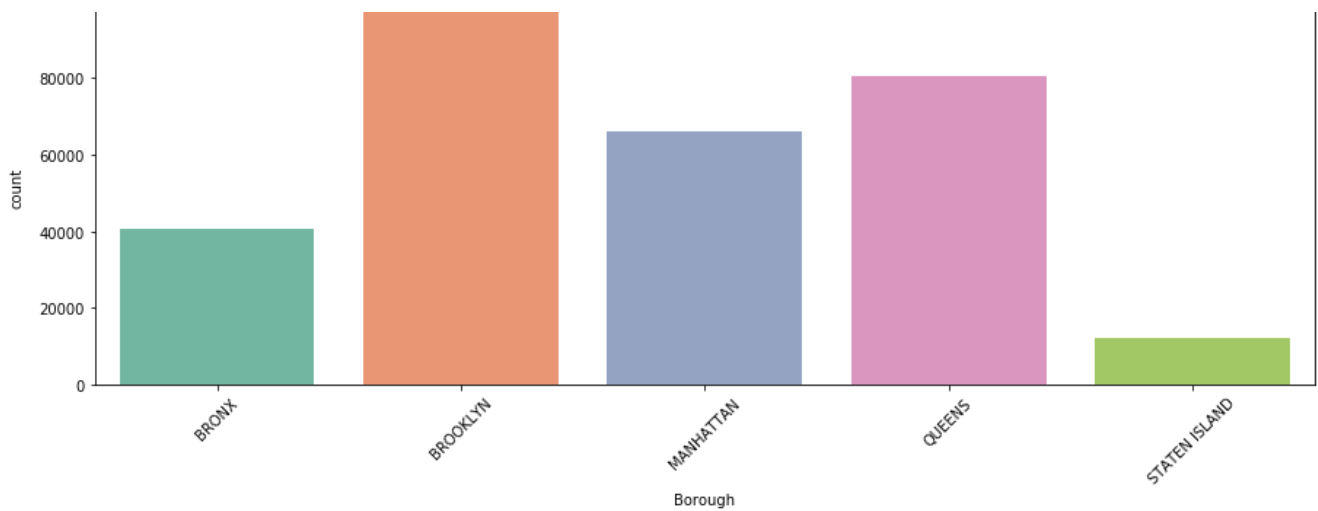


In [36]:

```
plt.figure(figsize=(15,5))
chart = sns.countplot(x='Borough',data=df_Service_Requests2,palette='Set2')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
```

Out[36]:

```
[Text(0, 0, 'BRONX'),
Text(0, 0, 'BROOKLYN'),
Text(0, 0, 'MANHATTAN'),
Text(0, 0, 'QUEENS'),
Text(0, 0, 'STATEN ISLAND')]
```



Conclusion 5:

Brooklyn has the highest No. of Complaints followed by Queens and Manhattan

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

In [37]:

```
df_Service_Requests2['Request_Closing_Time'].describe()
```

Out[37]:

```
count    297906.000000
mean         4.306074
std         6.078652
min         0.016667
25%         1.273056
50%         2.709444
75%         5.339722
max        592.872778
Name: Request_Closing_Time, dtype: float64
```

In [38]:

```
df_Service_Requests2['Request_Closing_Time_dt'].describe()
```

Out[38]:

```
count          297906
mean    0 days 04:18:21.867038
std      0 days 06:04:43.146383
min           0 days 00:01:00
25%           0 days 01:16:23
50%           0 days 02:42:34
75%           0 days 05:20:23
max          24 days 16:52:22
Name: Request_Closing_Time_dt, dtype: object
```

In [39]:

```
# Let us convert Request_Closing_Time_dt from days to hours

df_Service_Requests2['Request_Closing_Hours'] = df_Service_Requests2['Request_Closing_Time_dt'].astype(
    'timedelta64[h]')+1
df_Service_Requests2[['Request_Closing_Time_dt', 'Request_Closing_Hours']].head()
```

Out[39]:

	Request_Closing_Time_dt	Request_Closing_Hours
0	00:55:15	1.0
1	01:26:16	2.0
2	04:51:31	5.0
3	07:45:14	8.0
4	03:27:02	4.0

In [40]:

```
# Borough wise

Closing_by_Borough = df_Service_Requests2.groupby(['Borough','Complaint Type'],as_index=False)['Request_Closing_Hours'].mean().sort_values(['Borough','Request_Closing_Hours'],ascending=[True, False])

Closing_by_Borough.rename(columns={'Request_Closing_Hours': 'Average_Request_Closing_Hours'}, inplace=True)

Closing_by_Borough
```

Out[40]:

	Borough	Complaint Type	Average_Request_Closing_Hours
16	BRONX	Panhandling	14.842105
4	BRONX	Derelict Vehicle	9.730907
7	BRONX	Graffiti	9.333333
8	BRONX	Homeless Encampment	7.967347
0	BRONX	Animal Abuse	7.840170
...
100	STATEN ISLAND	Noise - House of Worship	3.058824
108	STATEN ISLAND	Urinating in Public	2.857143
105	STATEN ISLAND	Posting Advertisement	2.071845
89	STATEN ISLAND	Animal in a Park	NaN
106	STATEN ISLAND	Squeegee	NaN

110 rows × 3 columns

In [41]:

```
# Location Type wise

Closing_by_Location_Type = df_Service_Requests2.groupby(['Location Type','Complaint Type'],as_index=False)['Request_Closing_Hours'].mean().sort_values(['Location Type','Request_Closing_Hours'],ascending=[True, False])

Closing_by_Location_Type.rename(columns={'Request_Closing_Hours': 'Average_Request_Closing_Hours'}, inplace=True)

pd.set_option('display.max_rows', None)
Closing_by_Location_Type
```

Out[41]:

	Location Type	Complaint Type	Average_Request_Closing_Hours
8	Bridge	Homeless Encampment	4.000000
0	Bridge	Animal Abuse	NaN
1	Bridge	Animal in a Park	NaN
2	Bridge	Bike/Roller/Skate Chronic	NaN

3	Location Type	Complaint Type	Average Request Closing Hours
	Bridge	Blocked Driveway	NaN
4	Bridge	Derelict Vehicle	NaN
5	Bridge	Disorderly Youth	NaN
6	Bridge	Drinking	NaN
7	Bridge	Graffiti	NaN
9	Bridge	Illegal Fireworks	NaN
10	Bridge	Illegal Parking	NaN
11	Bridge	Noise - Commercial	NaN
12	Bridge	Noise - House of Worship	NaN
13	Bridge	Noise - Park	NaN
14	Bridge	Noise - Street/Sidewalk	NaN
15	Bridge	Noise - Vehicle	NaN
16	Bridge	Panhandling	NaN
17	Bridge	Posting Advertisement	NaN
18	Bridge	Squeegee	NaN
19	Bridge	Traffic	NaN
20	Bridge	Urinating in Public	NaN
21	Bridge	Vending	NaN
42	Club/Bar/Restaurant	Urinating in Public	8.476190
28	Club/Bar/Restaurant	Drinking	5.054795
33	Club/Bar/Restaurant	Noise - Commercial	3.577277
22	Club/Bar/Restaurant	Animal Abuse	NaN
23	Club/Bar/Restaurant	Animal in a Park	NaN
24	Club/Bar/Restaurant	Bike/Roller/Skate Chronic	NaN
25	Club/Bar/Restaurant	Blocked Driveway	NaN
26	Club/Bar/Restaurant	Derelict Vehicle	NaN
27	Club/Bar/Restaurant	Disorderly Youth	NaN
29	Club/Bar/Restaurant	Graffiti	NaN
30	Club/Bar/Restaurant	Homeless Encampment	NaN
31	Club/Bar/Restaurant	Illegal Fireworks	NaN
32	Club/Bar/Restaurant	Illegal Parking	NaN
34	Club/Bar/Restaurant	Noise - House of Worship	NaN
35	Club/Bar/Restaurant	Noise - Park	NaN
36	Club/Bar/Restaurant	Noise - Street/Sidewalk	NaN
37	Club/Bar/Restaurant	Noise - Vehicle	NaN
38	Club/Bar/Restaurant	Panhandling	NaN
39	Club/Bar/Restaurant	Posting Advertisement	NaN
40	Club/Bar/Restaurant	Squeegee	NaN
41	Club/Bar/Restaurant	Traffic	NaN
43	Club/Bar/Restaurant	Vending	NaN
44	Commercial	Animal Abuse	5.790323
45	Commercial	Animal in a Park	NaN
46	Commercial	Bike/Roller/Skate Chronic	NaN
47	Commercial	Blocked Driveway	NaN
48	Commercial	Derelict Vehicle	NaN
49	Commercial	Disorderly Youth	NaN
50	Commercial	Drinking	NaN
51	Commercial	Graffiti	NaN

52	Commercial	Homeless Encampment	Average_Request_Closing_Hours
	Location Type	Complaint Type	
53	Commercial	Illegal Fireworks	NaN
54	Commercial	Illegal Parking	NaN
55	Commercial	Noise - Commercial	NaN
56	Commercial	Noise - House of Worship	NaN
57	Commercial	Noise - Park	NaN
58	Commercial	Noise - Street/Sidewalk	NaN
59	Commercial	Noise - Vehicle	NaN
60	Commercial	Panhandling	NaN
61	Commercial	Posting Advertisement	NaN
62	Commercial	Squeegee	NaN
63	Commercial	Traffic	NaN
64	Commercial	Urinating in Public	NaN
65	Commercial	Vending	NaN
70	Highway	Derelict Vehicle	8.692308
85	Highway	Traffic	3.896739
74	Highway	Homeless Encampment	3.785714
66	Highway	Animal Abuse	NaN
67	Highway	Animal in a Park	NaN
68	Highway	Bike/Roller/Skate Chronic	NaN
69	Highway	Blocked Driveway	NaN
71	Highway	Disorderly Youth	NaN
72	Highway	Drinking	NaN
73	Highway	Graffiti	NaN
75	Highway	Illegal Fireworks	NaN
76	Highway	Illegal Parking	NaN
77	Highway	Noise - Commercial	NaN
78	Highway	Noise - House of Worship	NaN
79	Highway	Noise - Park	NaN
80	Highway	Noise - Street/Sidewalk	NaN
81	Highway	Noise - Vehicle	NaN
82	Highway	Panhandling	NaN
83	Highway	Posting Advertisement	NaN
84	Highway	Squeegee	NaN
86	Highway	Urinating in Public	NaN
87	Highway	Vending	NaN
88	House and Store	Animal Abuse	5.505376
89	House and Store	Animal in a Park	NaN
90	House and Store	Bike/Roller/Skate Chronic	NaN
91	House and Store	Blocked Driveway	NaN
92	House and Store	Derelict Vehicle	NaN
93	House and Store	Disorderly Youth	NaN
94	House and Store	Drinking	NaN
95	House and Store	Graffiti	NaN
96	House and Store	Homeless Encampment	NaN
97	House and Store	Illegal Fireworks	NaN
98	House and Store	Illegal Parking	NaN
99	House and Store	Noise - Commercial	NaN
100	House and Store	Noise - House of Worship	NaN

	Location Type	Complaint Type	Average_Request_Closing_Hours
101	House and Store	Noise - Park	NaN
102	House and Store	Noise - Street/Sidewalk	NaN
103	House and Store	Noise - Vehicle	NaN
104	House and Store	Panhandling	NaN
105	House and Store	Posting Advertisement	NaN
106	House and Store	Squeegee	NaN
107	House and Store	Traffic	NaN
108	House and Store	Urinating in Public	NaN
109	House and Store	Vending	NaN
122	House of Worship	Noise - House of Worship	3.722762
110	House of Worship	Animal Abuse	NaN
111	House of Worship	Animal in a Park	NaN
112	House of Worship	Bike/Roller/Skate Chronic	NaN
113	House of Worship	Blocked Driveway	NaN
114	House of Worship	Derelict Vehicle	NaN
115	House of Worship	Disorderly Youth	NaN
116	House of Worship	Drinking	NaN
117	House of Worship	Graffiti	NaN
118	House of Worship	Homeless Encampment	NaN
119	House of Worship	Illegal Fireworks	NaN
120	House of Worship	Illegal Parking	NaN
121	House of Worship	Noise - Commercial	NaN
123	House of Worship	Noise - Park	NaN
124	House of Worship	Noise - Street/Sidewalk	NaN
125	House of Worship	Noise - Vehicle	NaN
126	House of Worship	Panhandling	NaN
127	House of Worship	Posting Advertisement	NaN
128	House of Worship	Squeegee	NaN
129	House of Worship	Traffic	NaN
130	House of Worship	Urinating in Public	NaN
131	House of Worship	Vending	NaN
133	Park	Animal in a Park	337.000000
132	Park	Animal Abuse	NaN
134	Park	Bike/Roller/Skate Chronic	NaN
135	Park	Blocked Driveway	NaN
136	Park	Derelict Vehicle	NaN
137	Park	Disorderly Youth	NaN
138	Park	Drinking	NaN
139	Park	Graffiti	NaN
140	Park	Homeless Encampment	NaN
141	Park	Illegal Fireworks	NaN
142	Park	Illegal Parking	NaN
143	Park	Noise - Commercial	NaN
144	Park	Noise - House of Worship	NaN
145	Park	Noise - Park	NaN
146	Park	Noise - Street/Sidewalk	NaN
147	Park	Noise - Vehicle	NaN
148	Park	Panhandling	NaN

149	Location	Park Type	Posting Advertisment Complaint Type	Average_Request_Closing_Hours
150		Park	Squeegee	NaN
151		Park	Traffic	NaN
152		Park	Urinating in Public	NaN
153		Park	Vending	NaN
163		Park/Playground	Illegal Fireworks	5.500000
162		Park/Playground	Homeless Encampment	4.439093
154		Park/Playground	Animal Abuse	4.213115
175		Park/Playground	Vending	3.990476
167		Park/Playground	Noise - Park	3.919532
160		Park/Playground	Drinking	3.835052
174		Park/Playground	Urinating in Public	3.368421
170		Park/Playground	Panhandling	1.666667
155		Park/Playground	Animal in a Park	NaN
156		Park/Playground	Bike/Roller/Skate Chronic	NaN
157		Park/Playground	Blocked Driveway	NaN
158		Park/Playground	Derelict Vehicle	NaN
159		Park/Playground	Disorderly Youth	NaN
161		Park/Playground	Graffiti	NaN
164		Park/Playground	Illegal Parking	NaN
165		Park/Playground	Noise - Commercial	NaN
166		Park/Playground	Noise - House of Worship	NaN
168		Park/Playground	Noise - Street/Sidewalk	NaN
169		Park/Playground	Noise - Vehicle	NaN
171		Park/Playground	Posting Advertisment	NaN
172		Park/Playground	Squeegee	NaN
173		Park/Playground	Traffic	NaN
176		Parking Lot	Animal Abuse	6.072727
193		Parking Lot	Posting Advertisment	2.571429
177		Parking Lot	Animal in a Park	NaN
178		Parking Lot	Bike/Roller/Skate Chronic	NaN
179		Parking Lot	Blocked Driveway	NaN
180		Parking Lot	Derelict Vehicle	NaN
181		Parking Lot	Disorderly Youth	NaN
182		Parking Lot	Drinking	NaN
183		Parking Lot	Graffiti	NaN
184		Parking Lot	Homeless Encampment	NaN
185		Parking Lot	Illegal Fireworks	NaN
186		Parking Lot	Illegal Parking	NaN
187		Parking Lot	Noise - Commercial	NaN
188		Parking Lot	Noise - House of Worship	NaN
189		Parking Lot	Noise - Park	NaN
190		Parking Lot	Noise - Street/Sidewalk	NaN
191		Parking Lot	Noise - Vehicle	NaN
192		Parking Lot	Panhandling	NaN
194		Parking Lot	Squeegee	NaN
195		Parking Lot	Traffic	NaN
196		Parking Lot	Urinating in Public	NaN
197		Parking Lot	Vending	NaN

Case Number	Location Type	Complaint Type	Average Request Closing Hours
198	Residential Building	Animal Abuse	5.356828
199	Residential Building	Animal in a Park	NaN
200	Residential Building	Bike/Roller/Skate Chronic	NaN
201	Residential Building	Blocked Driveway	NaN
202	Residential Building	Derelict Vehicle	NaN
203	Residential Building	Disorderly Youth	NaN
204	Residential Building	Drinking	NaN
205	Residential Building	Graffiti	NaN
206	Residential Building	Homeless Encampment	NaN
207	Residential Building	Illegal Fireworks	NaN
208	Residential Building	Illegal Parking	NaN
209	Residential Building	Noise - Commercial	NaN
210	Residential Building	Noise - House of Worship	NaN
211	Residential Building	Noise - Park	NaN
212	Residential Building	Noise - Street/Sidewalk	NaN
213	Residential Building	Noise - Vehicle	NaN
214	Residential Building	Panhandling	NaN
215	Residential Building	Posting Advertisement	NaN
216	Residential Building	Squeegee	NaN
217	Residential Building	Traffic	NaN
218	Residential Building	Urinating in Public	NaN
219	Residential Building	Vending	NaN
227	Residential Building/House	Graffiti	6.375000
220	Residential Building/House	Animal Abuse	5.903365
236	Residential Building/House	Panhandling	5.625000
228	Residential Building/House	Homeless Encampment	5.431333
240	Residential Building/House	Urinating in Public	4.775362
241	Residential Building/House	Vending	4.716418
225	Residential Building/House	Disorderly Youth	4.311688
226	Residential Building/House	Drinking	4.152249
237	Residential Building/House	Posting Advertisement	4.111111
222	Residential Building/House	Bike/Roller/Skate Chronic	4.080000
229	Residential Building/House	Illegal Fireworks	3.606061
221	Residential Building/House	Animal in a Park	NaN
223	Residential Building/House	Blocked Driveway	NaN
224	Residential Building/House	Derelict Vehicle	NaN
230	Residential Building/House	Illegal Parking	NaN
231	Residential Building/House	Noise - Commercial	NaN
232	Residential Building/House	Noise - House of Worship	NaN
233	Residential Building/House	Noise - Park	NaN
234	Residential Building/House	Noise - Street/Sidewalk	NaN
235	Residential Building/House	Noise - Vehicle	NaN
238	Residential Building/House	Squeegee	NaN
239	Residential Building/House	Traffic	NaN
246	Roadway Tunnel	Derelict Vehicle	18.600000
250	Roadway Tunnel	Homeless Encampment	3.000000
261	Roadway Tunnel	Traffic	2.724138
242	Roadway Tunnel	Animal Abuse	NaN

243	Location Type	Complaint Type	Average_Request_Closing_Hours
244	Roadway Tunnel	Bike/Roller/Skate Chronic	NaN
245	Roadway Tunnel	Blocked Driveway	NaN
247	Roadway Tunnel	Disorderly Youth	NaN
248	Roadway Tunnel	Drinking	NaN
249	Roadway Tunnel	Graffiti	NaN
251	Roadway Tunnel	Illegal Fireworks	NaN
252	Roadway Tunnel	Illegal Parking	NaN
253	Roadway Tunnel	Noise - Commercial	NaN
254	Roadway Tunnel	Noise - House of Worship	NaN
255	Roadway Tunnel	Noise - Park	NaN
256	Roadway Tunnel	Noise - Street/Sidewalk	NaN
257	Roadway Tunnel	Noise - Vehicle	NaN
258	Roadway Tunnel	Panhandling	NaN
259	Roadway Tunnel	Posting Advertisement	NaN
260	Roadway Tunnel	Squeegee	NaN
262	Roadway Tunnel	Urinating in Public	NaN
263	Roadway Tunnel	Vending	NaN
271	Store/Commercial	Graffiti	6.031250
280	Store/Commercial	Panhandling	4.850000
264	Store/Commercial	Animal Abuse	4.773512
272	Store/Commercial	Homeless Encampment	4.732422
285	Store/Commercial	Vending	4.493056
266	Store/Commercial	Bike/Roller/Skate Chronic	4.264151
284	Store/Commercial	Urinating in Public	3.969697
270	Store/Commercial	Drinking	3.733333
275	Store/Commercial	Noise - Commercial	3.720833
269	Store/Commercial	Disorderly Youth	3.375000
281	Store/Commercial	Posting Advertisement	3.000000
273	Store/Commercial	Illegal Fireworks	2.500000
265	Store/Commercial	Animal in a Park	NaN
267	Store/Commercial	Blocked Driveway	NaN
268	Store/Commercial	Derelict Vehicle	NaN
274	Store/Commercial	Illegal Parking	NaN
276	Store/Commercial	Noise - House of Worship	NaN
277	Store/Commercial	Noise - Park	NaN
278	Store/Commercial	Noise - Street/Sidewalk	NaN
279	Store/Commercial	Noise - Vehicle	NaN
282	Store/Commercial	Squeegee	NaN
283	Store/Commercial	Traffic	NaN
293	Street/Sidewalk	Graffiti	12.600000
290	Street/Sidewalk	Derelict Vehicle	7.855955
286	Street/Sidewalk	Animal Abuse	5.608781
289	Street/Sidewalk	Blocked Driveway	5.240527
296	Street/Sidewalk	Illegal Parking	4.994736
302	Street/Sidewalk	Panhandling	4.887892
304	Street/Sidewalk	Squeegee	4.750000
294	Street/Sidewalk	Homeless Encampment	4.732782
287	Street/Sidewalk	Vending	4.520424

307	Street/Sidewalk	vending	4.530124
	Location Type	Complaint Type	Average_Request_Closing_Hours
288	Street/Sidewalk	Bike/Roller/Skate Chronic	4.279070
292	Street/Sidewalk	Drinking	4.167442
301	Street/Sidewalk	Noise - Vehicle	4.101680
305	Street/Sidewalk	Traffic	3.976380
291	Street/Sidewalk	Disorderly Youth	3.975124
300	Street/Sidewalk	Noise - Street/Sidewalk	3.954457
306	Street/Sidewalk	Urinating in Public	3.787975
295	Street/Sidewalk	Illegal Fireworks	3.104000
303	Street/Sidewalk	Posting Advertisement	2.350000
287	Street/Sidewalk	Animal in a Park	NaN
297	Street/Sidewalk	Noise - Commercial	NaN
298	Street/Sidewalk	Noise - House of Worship	NaN
299	Street/Sidewalk	Noise - Park	NaN
308	Subway Station	Animal Abuse	3.636364
328	Subway Station	Urinating in Public	1.750000
309	Subway Station	Animal in a Park	NaN
310	Subway Station	Bike/Roller/Skate Chronic	NaN
311	Subway Station	Blocked Driveway	NaN
312	Subway Station	Derelict Vehicle	NaN
313	Subway Station	Disorderly Youth	NaN
314	Subway Station	Drinking	NaN
315	Subway Station	Graffiti	NaN
316	Subway Station	Homeless Encampment	NaN
317	Subway Station	Illegal Fireworks	NaN
318	Subway Station	Illegal Parking	NaN
319	Subway Station	Noise - Commercial	NaN
320	Subway Station	Noise - House of Worship	NaN
321	Subway Station	Noise - Park	NaN
322	Subway Station	Noise - Street/Sidewalk	NaN
323	Subway Station	Noise - Vehicle	NaN
324	Subway Station	Panhandling	NaN
325	Subway Station	Posting Advertisement	NaN
326	Subway Station	Squeegee	NaN
327	Subway Station	Traffic	NaN
329	Subway Station	Vending	NaN
334	Vacant Lot	Derelict Vehicle	7.974026
330	Vacant Lot	Animal Abuse	NaN
331	Vacant Lot	Animal in a Park	NaN
332	Vacant Lot	Bike/Roller/Skate Chronic	NaN
333	Vacant Lot	Blocked Driveway	NaN
335	Vacant Lot	Disorderly Youth	NaN
336	Vacant Lot	Drinking	NaN
337	Vacant Lot	Graffiti	NaN
338	Vacant Lot	Homeless Encampment	NaN
339	Vacant Lot	Illegal Fireworks	NaN
340	Vacant Lot	Illegal Parking	NaN
341	Vacant Lot	Noise - Commercial	NaN
342	Vacant Lot	Noise - House of Worship	NaN

	Location Type	Complaint Type	Average Request Closing Hours
343	Vacant Lot	Noise - Street/Sidewalk	NaN
344	Vacant Lot	Noise - Vehicle	NaN
345	Vacant Lot	Panhandling	NaN
346	Vacant Lot	Posting Advertisement	NaN
347	Vacant Lot	Squeegee	NaN
348	Vacant Lot	Traffic	NaN
349	Vacant Lot	Urinating in Public	NaN
350	Vacant Lot	Vending	NaN
351	Vacant Lot		

4. 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)

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

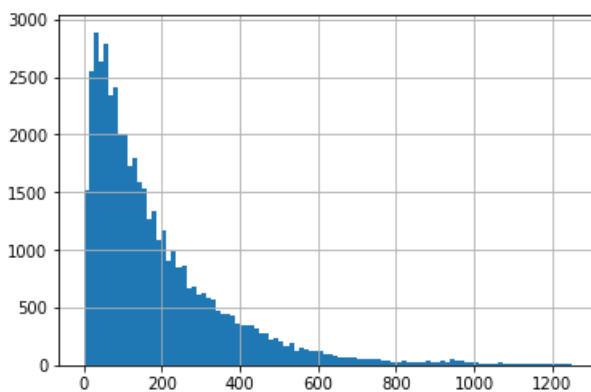
In [42]:

```
df_Service_Requests2['Request_Closing_Minutes'] = df_Service_Requests2['Request_Closing_Time_dt'].astype('timedelta64[m]')+1

#First, let us look at one Complaint Type and its distribution
# Let us look take a loo at "Noise - Street/Sidewalk"
complaint_noise_st_sw = df_Service_Requests2[df_Service_Requests2['Complaint Type']=='Noise - Street/Sidewalk']['Request_Closing_Minutes']
complaint_noise_st_sw.hist(bins=100,range=(0,1250))
```

Out[42]:

<matplotlib.axes._subplots.AxesSubplot at 0x1b569a53d48>



In [43]:

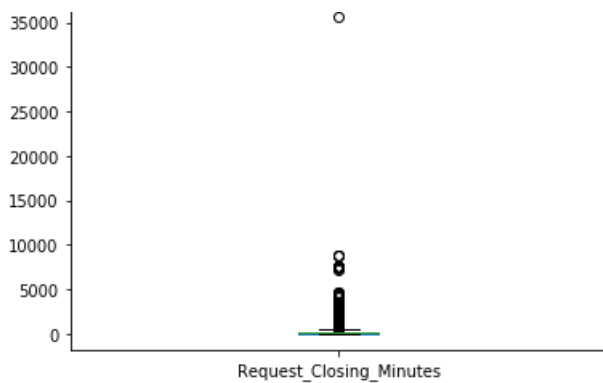
```
complaint_noise_st_sw.plot('box')
```

C:\Users\vkakarla\anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: `Series.plot()` should not be called with positional arguments, only keyword arguments. The order of positional argument s will change in the future. Use `Series.plot(kind='box')` instead of `Series.plot('box',)`.

"""Entry point for launching an IPython kernel.

Out[43]:

<matplotlib.axes._subplots.AxesSubplot at 0x1b500be5cc8>



In [44]:

```
complaint_noise_st_sw.describe()
```

Out[44]:

```
count    47977.000000
mean      207.028430
std       326.639027
min         3.000000
25%        61.000000
50%       133.000000
75%       259.000000
max      35573.000000
Name: Request_Closing_Minutes, dtype: float64
```

Looking at the above summary and the histogram and box plot, we can understand that there are outliers in the data.

It is right skewed data.

Let us remove the skewness using log

In [45]:

```
df_Service_Requests_log = {}
for complaint in df_Service_Requests2['Complaint Type'].unique():
    df_Service_Requests_log[complaint] = np.log(df_Service_Requests2[df_Service_Requests2['Complaint Type']==complaint]['Request_Closing_Minutes'])
```

In [46]:

```
df_Service_Requests_log.keys()
```

Out[46]:

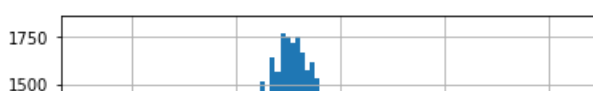
```
dict_keys(['Noise - Street/Sidewalk', 'Blocked Driveway', 'Illegal Parking', 'Derelict Vehicle', 'Noise - Commercial', 'Noise - House of Worship', 'Posting Advertisement', 'Noise - Vehicle', 'Animal Abuse', 'Vending', 'Traffic', 'Drinking', 'Bike/Roller/Skate Chronic', 'Panhandling', 'Noise - Park',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

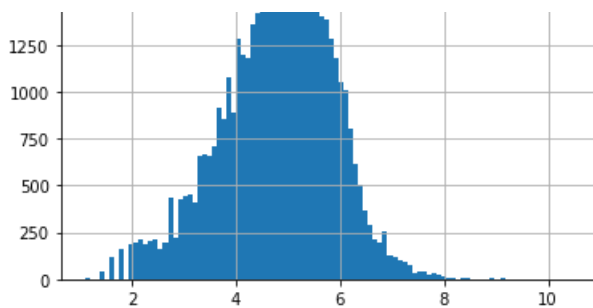
In [47]:

```
df_Service_Requests_log['Noise - Street/Sidewalk'].hist(bins=100)
```

Out[47]:

<matplotlib.axes._subplots.AxesSubplot at 0x1b500c4a348>





In [48]:

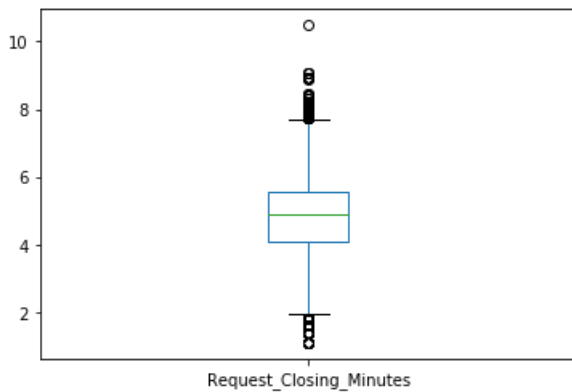
```
df_Service_Requests_log['Noise - Street/Sidewalk'].plot('box')
```

C:\Users\vkakarla\anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: `Series.plot()` should not be called with positional arguments, only keyword arguments. The order of positional arguments will change in the future. Use `Series.plot(kind='box')` instead of `Series.plot('box',)`.

"""Entry point for launching an IPython kernel.

Out[48]:

<matplotlib.axes._subplots.AxesSubplot at 0x1b500de9088>



The above histogram looks like a bell curve which means that there is no skewness, boxplot also shows the same.

Now the skewness is removed from the data after using log.

In [49]:

```
# We need conduct an One-Way ANOVA Test since we need to compare the average of more than two groups.
# Looking at mean and standard deviation of each complaint type

print("                Mean  Std Dev")
for complaint in df_Service_Requests_log.keys():
    print(complaint.ljust(30, " "), end=" ")
    print("{:.2} ".format(df_Service_Requests_log[complaint].mean()), end=" "),
    print("{:.2} ".format(df_Service_Requests_log[complaint].std()))
```

	Mean	Std Dev
Noise - Street/Sidewalk	4.8	1.1
Blocked Driveway	5.2	0.97
Illegal Parking	5.1	1.1
Derelict Vehicle	5.4	1.2
Noise - Commercial	4.7	1.1
Noise - House of Worship	4.6	1.2
Posting Advertisement	4.2	1.2
Noise - Vehicle	4.9	1.1
Animal Abuse	5.2	1.0
Vending	5.0	1.1
Traffic	4.7	1.2
Drinking	5.0	1.0

Bike/Roller/Skate Chronic	4.9	1.2
Panhandling	5.0	1.1
Noise - Park	4.8	1.1
Homeless Encampment	5.1	1.0
Urinating in Public	4.8	1.1
Graffiti	5.5	1.1
Disorderly Youth	4.9	1.0
Illegal Fireworks	4.5	1.2
Squeegee	5.3	0.85
Animal in a Park	9.9	nan

One-Way ANOVA

Null Hypothesis (H0): The average response time across complaint types is same (overall).

Alternate Hypothesis (H1): The average response time across complaint types is not same (overall).

let us conduct the test for top "5" complaints types

Let us choose "level of significane (alpha)" at 5% which give 95% confidence of the test

In [50]:

```
df_Service_Requests_log.keys()
```

Out[50]:

```
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', 'Squeegee', 'Animal in a Park'])
```

In [51]:

```
# let us import the f_oneway from scipy.stats library for the test

from scipy.stats import f_oneway

F_Statistic, p_value = f_oneway(df_Service_Requests_log['Noise - Street/Sidewalk'],
                                df_Service_Requests_log['Blocked Driveway'],
                                df_Service_Requests_log['Illegal Parking'],
                                df_Service_Requests_log['Derelict Vehicle'],
                                df_Service_Requests_log['Noise - Commercial'])

print('Statistics=%.3f, p=%.3f' % (F_Statistic, p_value))

# Test Result as per level of significance:

alpha = 0.05 #5% level of significance

if p_value > alpha:
    print('Failed to Reject Null Hypothesis (H0)')
else:
    print('Rejecting Null Hypothesis (H0)')
```

```
Statistics=2467.485, p=0.000
Rejecting Null Hypothesis (H0)
```

Null hypothesis is rejected, so we have enough evidence to conclude as below:

"The average response time across complaint types is not same (overall)."

In []:

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

Let us consider the below columns to find the correlation between location and complaint types

Complaint Type, Borough, City, Longitude, and Latitude

In [61]:

```
df_Service_Requests_corr_data = df_Service_Requests2[['Complaint Type', 'Borough', 'Longitude', 'Latitude', 'City', 'Location Type']]
```

In [62]:

```
# Since Complaint Type, Borough and City are not a numerical type we can not calculate correlation directly.
```

```
# Let us convert them to numerical data by mapping the category to category codes
```

```
df_Service_Requests_corr_data['Complaint Type'] = df_Service_Requests_corr_data['Complaint Type'].cat.codes
df_Service_Requests_corr_data['Borough'] = df_Service_Requests_corr_data['Borough'].cat.codes
df_Service_Requests_corr_data['City'] = df_Service_Requests_corr_data['City'].cat.codes
df_Service_Requests_corr_data['Location Type'] = df_Service_Requests_corr_data['Location Type'].cat.codes
```

```
C:\Users\vkakarla\anaconda3\lib\site-packages\ipykernel_launcher.py:5: 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: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
C:\Users\vkakarla\anaconda3\lib\site-packages\ipykernel_launcher.py:6: 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: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
C:\Users\vkakarla\anaconda3\lib\site-packages\ipykernel_launcher.py:7: 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: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
import sys
C:\Users\vkakarla\anaconda3\lib\site-packages\ipykernel_launcher.py:8: 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: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

In [63]:

```
df_Service_Requests_corr_data.corr(method='pearson')
```

Out[63]:

	Complaint Type	Borough	Longitude	Latitude	City	Location Type
Complaint Type	1.000000	-0.066257	-0.180978	0.152894	0.094990	-0.101088
Borough	-0.066257	1.000000	0.021367	-0.249487	0.717788	-0.024583
Longitude	-0.180978	0.021367	1.000000	0.364936	-0.123996	0.066888
Latitude	0.152894	-0.249487	0.364936	1.000000	-0.000533	-0.026850
City	0.094990	0.717788	-0.123996	-0.000533	1.000000	-0.045244
Location Type	-0.101088	-0.024583	0.066888	-0.026850	-0.045244	1.000000

Location Type	-0.101088	-0.024583	0.066888	-0.026850	-0.045244	1.000000
Complaint Type		Borough	Longitude	Latitude	City	Location Type

Conclusion:

We can observe from the above table that there is no significant correlation ($>.70$) between complaint type and location.

Hence, type of complaint or service requested and location are not related

In []:

In []: