

In [1]:

```
import pandas as pd
```

In [2]:

```
nyc=pd.read_csv('311_Service_Requests_from_2010_to_Present.csv')
```

C:\Users\Sumit\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)

In [3]:

```
#Import a 311 NYC service request.
```

In [4]:

```
nyc.head()
```

Out[4]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Typ
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/Sidewal
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/Sidewal
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/Sidewal
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/Sidewal
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/Sidewal

5 rows × 53 columns

In [5]:

```
#Read or convert the columns 'Created Date' and Closed Date' to datetime datatype and creat
```

In [6]:

```
pd.to_datetime(nyc['Created Date'])
```

Out[6]:

```
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
...
300693 2015-03-29 00:33:41
300694 2015-03-29 00:33:28
300695 2015-03-29 00:33:03
300696 2015-03-29 00:33:02
300697 2015-03-29 00:33:01
Name: Created Date, Length: 300698, dtype: datetime64[ns]
```

In [7]:

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

In [8]:

```
#pandas.to_datetime()
nyc['Closed Date']=pd.to_datetime(nyc['Closed Date'])
```

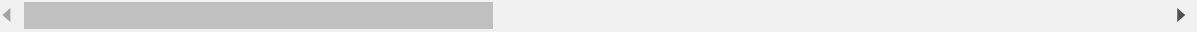
In [9]:

```
nyc.head()
```

Out[9]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type
0	32310363	2015-12-31 23:59:45	2016-01-01 00:55:00	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	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	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	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk
4	32306529	2015-12-31 23:56:58	2016-01-01 03:24:00	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk

5 rows × 53 columns



In [10]:

nyc.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 300698 entries, 0 to 300697

Data columns (total 53 columns):

#	Column	Non-Null Count	Dtype
0	Unique Key	300698 non-null	int64
1	Created Date	300698 non-null	datetime64[ns]
2	Closed Date	298534 non-null	datetime64[ns]
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: datetime64[ns](2), float64(10), int64(1), object(40)
memory usage: 121.6+ MB

In [11]:

```
# pd.to_datetime()  
nyc['Closed Date'] = pd.to_datetime(nyc['Closed Date'])
```

In [15]:

```
# 'Request_Closing_Time' as the time elapsed between request creation and request closing.  
nyc['Request_Closing_Time'] = nyc['Created Date'] - nyc['Closed Date']
```

In [17]:

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

In [18]:

```
#yyyy:mm:dd hh:mm:ss  
nyc['Request_Closing_Time']
```

Out[18]:

```
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  
...  
300693      NaT  
300694      0 days 02:00:31  
300695      0 days 03:07:17  
300696      0 days 04:05:33  
300697      0 days 04:08:49  
Name: Request_Closing_Time, Length: 300698, dtype: timedelta64[ns]
```

In [20]:

```
# request closing time in days  
import numpy as np  
nyc['response_days'] = nyc['Request_Closing_Time'] / np.timedelta64(1, 'D')
```

In [21]:

```
nyc['response_days']
```

Out[21]:

```
0      0.038368
1      0.059907
2      0.202442
3      0.323079
4      0.143773
...
300693      NaN
300694      0.083692
300695      0.130058
300696      0.170521
300697      0.172789
Name: response_days, Length: 300698, dtype: float64
```

In [23]:

```
# request closing time to hours
nyc['response_hrs']=nyc['Request_Closing_Time']/np.timedelta64(1,'h')
```

In [24]:

```
nyc['response_hrs']
```

Out[24]:

```
0      0.920833
1      1.437778
2      4.858611
3      7.753889
4      3.450556
...
300693      NaN
300694      2.008611
300695      3.121389
300696      4.092500
300697      4.146944
Name: response_hrs, Length: 300698, dtype: float64
```

In [25]:

```
an offer in a visual format (graphs or tables); at least 4 major conclusions that you can co
```

In [26]:

```
# frequency of complaint types
```

```
nyc['Complaint Type'].value_counts()
```

Out[26]:

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

In [29]:

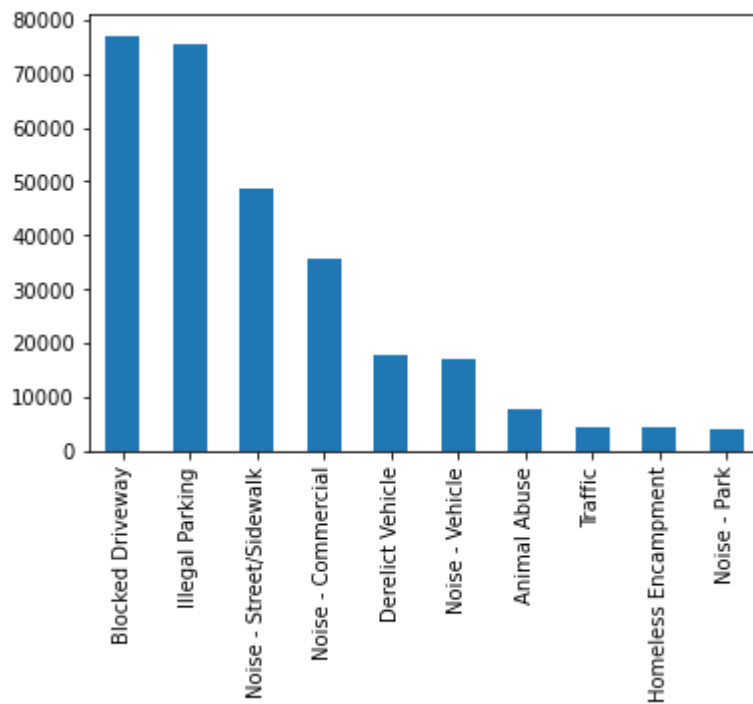
```
top_10_types=nyc['Complaint Type'].value_counts()[:10]
```

In [31]:

```
# barplot for to 10  
top_10_types.plot(kind='bar')
```

Out[31]:

<AxesSubplot:>



In [32]:

```
nyc['Location Type'].value_counts()
```

Out[32]:

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

In [34]:

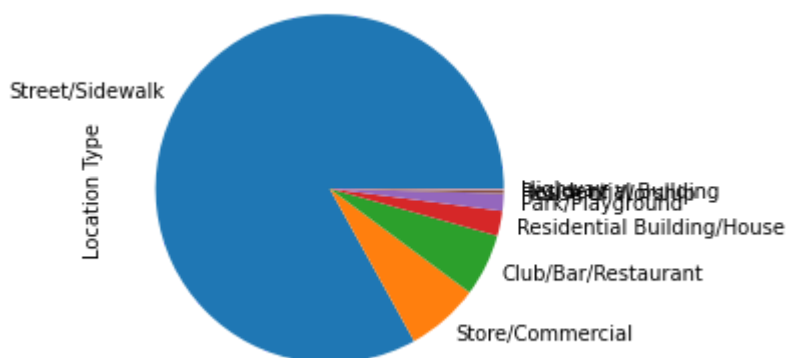
```
top_loc=nyc['Location Type'].value_counts()[:8]
```

In [36]:

```
top_loc.plot(kind='pie')
```

Out[36]:

<AxesSubplot:ylabel='Location Type'>



In [37]:

```
nyc['City'].value_counts()
```

Out[37]:

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

In [39]:

```
# average response in days by city
```

```
nyc.groupby('City')['response_days'].mean()
```

Out[39]:

City	
ARVERNE	0.094372
ASTORIA	0.191621
Astoria	0.174359
BAYSIDE	0.111639
BELLEROSE	0.439852
BREEZY POINT	0.145687
BRONX	0.254007
BROOKLYN	0.168666
CAMBRIA HEIGHTS	0.421824
CENTRAL PARK	0.137263
COLLEGE POINT	0.136401
CORONA	0.134493
EAST ELMHURST	0.149069
ELMHURST	0.137938
East Elmhurst	0.251992
FAR ROCKAWAY	0.116250
FLORAL PARK	0.488313
FLUSHING	0.125751
FOREST HILLS	0.134340
FRESH MEADOWS	0.136002
GLEN OAKS	0.367322
HOLLIS	0.240007
HOWARD BEACH	0.256703
Howard Beach	0.167882
JACKSON HEIGHTS	0.136403
JAMAICA	0.217088
KEW GARDENS	0.210124
LITTLE NECK	0.107403
LONG ISLAND CITY	0.272466
Long Island City	0.170865
MASPETH	0.233323
MIDDLE VILLAGE	0.224373
NEW HYDE PARK	0.314837
NEW YORK	0.123859
OAKLAND GARDENS	0.109620
OZONE PARK	0.236711
QUEENS	0.566379
QUEENS VILLAGE	0.454452
REGO PARK	0.144212
RICHMOND HILL	0.228930
RIDGEWOOD	0.185075
ROCKAWAY PARK	0.096621
ROSEDALE	0.417964
SAINT ALBANS	0.196703
SOUTH OZONE PARK	0.221999
SOUTH RICHMOND HILL	0.234062
SPRINGFIELD GARDENS	0.382740
STATEN ISLAND	0.161664
SUNNYSIDE	0.285500
WHITESTONE	0.135201
WOODHAVEN	0.233145
WOODSIDE	0.287226

Woodside 0.216725
Name: response_days, dtype: float64

In [40]:

```
#4-Order the complaint types based on the average 'Request_Closing_Time', grouping them for
```

In [41]:

```
# order the complaint types based on the averages 'Request_Closing_Time'
```

In [45]:

```
nyc.groupby('Complaint Type')['response_hrs'].mean()
```

Out[45]:

Complaint Type	
Agency Issues	5.260324
Animal Abuse	5.213244
Animal in a Park	336.834722
Bike/Roller/Skate Chronic	3.766462
Blocked Driveway	4.740905
Derelict Vehicle	7.364140
Disorderly Youth	3.558584
Drinking	3.861833
Ferry Complaint	NaN
Graffiti	7.151251
Homeless Encampment	4.365570
Illegal Fireworks	2.761139
Illegal Parking	4.501152
Noise - Commercial	3.147154
Noise - House of Worship	3.193298
Noise - Park	3.410732
Noise - Street/Sidewalk	3.445224
Noise - Vehicle	3.588993
Panhandling	4.372768
Posting Advertisement	1.975803
Squeegee	4.045625
Traffic	3.448681
Urinating in Public	3.626664
Vending	4.013919

Name: response_hrs, dtype: float64

In [49]:

```
df=nyc.groupby(['City', 'Complaint Type'])['response_days'].mean()
```

In [48]:

```
df
```

Out[48]:

```
City      Complaint Type      response_hrs
ARVERNE   Animal Abuse         2.153626
          Blocked Driveway      2.525968
          Derelict Vehicle       2.968220
          Disorderly Youth       3.591250
          Drinking              0.238611
          ...
Woodside  Blocked Driveway      6.406212
          Derelict Vehicle       4.966667
          Illegal Parking        5.219203
          Noise - Commercial     2.394167
          Noise - Street/Sidewalk 3.411278
Name: response_hrs, Length: 764, dtype: float64
```

In [51]:

```
pd.pivot_table(nyc,index='City',columns='Complaint Type',values='response_hrs',aggfunc='mean')
```

ROCKAWAY PARK	2.282750	NaN	NaN	2.417353	2.727346	3.426389	2.303750	NaN
ROSEDALE	14.593712	NaN	8.525000	7.457450	14.507449	NaN	5.505556	0.156389
SAINT ALBANS	7.371750	NaN	NaN	4.686787	5.900492	1.759167	0.961389	NaN
SOUTH OZONE PARK	3.492338	NaN	4.055000	4.427600	10.398522	2.682222	4.881197	NaN
SOUTH RICHMOND	4.367703	NaN	30.910278	4.866324	11.876894	3.065000	5.766679	NaN

In [52]:

```
# 5- Perform a statistical test for the following:

#Please note: For the below statements you need to state the Null and Alternate and then pr

#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 [54]:

```
# y=response_time_days

# by complaint types
```

In [55]:

```
# H0: the average response time in days is same for different complaint types  
# Ha: the average response time in days is different for different complaint types
```

In [56]:

```
nyc['Complaint Type'].value_counts()[:5]
```

Out[56]:

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

In [58]:

```
df1=nyc[nyc['Complaint Type']=='Blocked Driveway']
```

In [68]:

```
df2=nyc[nyc['Complaint Type']=='Illegal Parking ']
```

In [69]:

```
df3=nyc[nyc['Complaint Type']=='Noise - Street/Sidewalk']
```

In [70]:

```
df4=nyc[nyc['Complaint Type']=='Noise - Commercial']
```

In [71]:

```
df5=nyc[nyc['Complaint Type']=='Derelict Vehicle ']
```

In [72]:

```
ar1=df1['response_days'].dropna()
```

In [73]:

```
ar1
```

Out[73]:

```
1      0.059907
2      0.202442
7      0.074942
9      0.057662
10     0.325023
...
300665 0.050544
300672 0.654468
300679 0.050347
300687 0.093669
300694 0.083692
Name: response_days, Length: 76810, dtype: float64
```

In [87]:

```
ar2=df2['response_days'].dropna()
```

```
ar2
```

In [88]:

```
ar2
```

Out[88]:

```
Series([], Name: response_days, dtype: float64)
```

In [75]:

```
ar3=df3['response_days'].dropna()
```

In [85]:

```
ar3
```

Out[85]:

```
0      0.038368
12     0.103438
19     0.032697
38     0.020463
54     0.062315
...
300674 0.097685
300675 0.042708
300681 0.120949
300683 0.122037
300689 0.152072
Name: response_days, Length: 48076, dtype: float64
```

In [76]:

```
ar4=df4['response_days'].dropna()
```

In [89]:

```
ar4
```

Out[89]:

```
17      0.035509
18      0.122234
22      0.052569
29      0.104155
30      0.082743
...
300691  0.095417
300692  0.026725
300695  0.130058
300696  0.170521
300697  0.172789
Name: response_days, Length: 35247, dtype: float64
```

In [77]:

```
ar5=df5['response_days'].dropna()
```

In [90]:

```
ar5
```

Out[90]:

```
Series([], Name: response_days, dtype: float64)
```

In [91]:

```
# run the anova model

import scipy.stats as stats
```

In [92]:

```
stats.f_oneway(ar1,ar2,ar3,ar4,ar5)
```

Out[92]:

```
F_onewayResult(statistic=nan, pvalue=nan)
```

In [93]:

```
# H0: the average response time in days is same for different complaint types
# Ha: the average response time in days is different for different complaint types
```

In [95]:

```
# as p value ~ 0, so we will reject

#the average time in days is not same for different complaint tpyes
```

In [96]:

```
# are the type of complaint or service requested and location related
```

```
nyc['City'].value_counts()
```

Out[96]:

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

In [97]:

```
top_5_complaints=nyc['Complaint Type'].value_counts()[:5]
```

In [100]:

```
top_5_complaints.index
```

Out[100]:

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

In [103]:

```
# filter the dataframe for the top_5_complaints
```

```
city_top5_comp=nyc[nyc['Complaint Type'].isin(top_5_complaints.index)]
```

In [104]:

```
city_top5_comp
```

Out[104]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location
0	32310363	2015-12-31 23:59:45	2016-01-01 00:55:00	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street
1	32309934	2015-12-31 23:59:44	2016-01-01 01:26:00	NYPD	New York City Police Department	Blocked Driveway	No Access	Street
2	32309159	2015-12-31 23:59:29	2016-01-01 04:51:00	NYPD	New York City Police Department	Blocked Driveway	No Access	Street
3	32305098	2015-12-31 23:57:46	2016-01-01 07:43:00	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street
4	32306529	2015-12-31 23:56:58	2016-01-01 03:24:00	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street
...
300693	30281872	2015-03-29 00:33:41	NaT	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/
300694	30281230	2015-03-29 00:33:28	2015-03-29 02:33:59	NYPD	New York City Police Department	Blocked Driveway	Partial Access	Street
300695	30283424	2015-03-29 00:33:03	2015-03-29 03:40:20	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/
300696	30280004	2015-03-29 00:33:02	2015-03-29 04:38:35	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/
300697	30281825	2015-03-29 00:33:01	2015-03-29 04:41:50	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Store/C

254312 rows × 56 columns

In [106]:

```
# get a crosstab of the City and the Complaint Types

city_comp_table=pd.crosstab(city_top5_comp['City'],city_top5_comp['Complaint Type'])
```

In [107]:

city_comp_table

NEW YORK	2072	537	12128	14550	20433
OAKLAND GARDENS	132	86	263	0	19
OZONE PARK	1259	420	619	115	137
QUEENS	2	1	8	6	6
QUEENS VILLAGE	585	370	578	43	66
REGO PARK	611	81	537	79	57
RICHMOND HILL	872	167	394	216	87
RIDGEWOOD	1694	330	1842	398	433
ROCKAWAY PARK	70	9	317	63	189
ROSEDALE	211	208	277	25	24
SAINT ALBANS	244	202	181	29	79
SOUTH OZONE PARK	942	358	494	70	105

In [108]:

City and complaint

chi square - categorical values

ho: the city and the complaint type is independent of each other

ha: the city and the complaint type is not independent of each other

In [109]:

chi2_stats,pvalue,df,mean_val=stats.chi2_contingency(city_comp_table)

In [110]:

chi2_stats

Out[110]:

72968.04166267907

In [112]:

#chi2 stats=sum of all (actual-expected)/expected

In [113]:

pvalue

Out[113]:

0.0

In [114]:

```
# as p value = 0 < alpha , so we will reject the H0  
# the city and the complaint type is not independent of each other
```

In []: