In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from scipy.stats import chi2_contingency
%matplotlib inline
```

In [2]:

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

```
C:\Users\Amrutha\anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py:3071: DtypeWarning: Columns (48,49) have mixed types.Specify dtype op
tion on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [3]:

```
pd.set_option('display.max_columns', None)
df.sample(10)
```

Out[3]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	
109791	31564611	09/18/2015 04:24:27 AM	09/18/2015 06:16:14 AM	NYPD	New York City Police Department	Blocked Driveway	No Access	
354996	29724557	01/15/2015 08:37:15 PM	01/15/2015 09:07:15 PM	NYPD	New York City Police Department	Blocked Driveway	No Access	
4731	32274000	12/26/2015 02:52:05 PM	12/26/2015 07:20:04 PM	NYPD	New York City Police Department	Blocked Driveway	No Access	
32293	32085798	11/29/2015 02:03:29 AM	11/29/2015 02:51:25 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Talking	Clu
34072	32079927	11/27/2015 05:56:13 AM	11/27/2015 07:32:57 AM	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	
23972	32148323	12/07/2015 11:46:16 AM	12/07/2015 10:14:04 PM	NYPD	New York City Police Department	Illegal Parking	Blocked Hydrant	
82598	31730956	10/11/2015 10:55:53 PM	10/12/2015 03:05:54 AM	NYPD	New York City Police Department	Noise - Vehicle	Car/Truck Music	
182605	31080124	07/15/2015 08:07:19 PM	07/15/2015 08:37:52 PM	NYPD	New York City Police Department	Noise - House of Worship	Loud Music/Party	
99719	31616798	09/26/2015 12:14:20 AM	09/26/2015 12:51:39 AM	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	
19916	32179791	12/11/2015 12:54:25 PM	12/11/2015 02:15:59 PM	NYPD	New York City Police Department	Blocked Driveway	No Access	

In [4]:

```
df.columns
```

Out[4]:

```
Index(['Unique Key', 'Created Date', 'Closed Date', 'Agency', 'Agency Nam
e',
       '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 Sta
te',
       'School Zip', 'School Not Found', 'School or Citywide Complaint',
       'Vehicle Type', 'Taxi Company Borough', 'Taxi Pick Up Location',
       'Bridge Highway Name', 'Bridge Highway Direction', 'Road Ramp',
       'Bridge Highway Segment', 'Garage Lot Name', 'Ferry Direction',
       'Ferry Terminal Name', 'Latitude', 'Longitude', 'Location'],
      dtype='object')
```

In [5]:

```
df.shape
```

Out[5]:

```
(364558, 53)
```

In [6]:

```
df.isna().sum()
```

Out[6]:

```
Unique Key                          0
Created Date                        0
Closed Date                      2381
Agency                              0
Agency Name                         0
Complaint Type                      0
Descriptor                       6501
Location Type                     133
Incident Zip                     2998
Incident Address                51699
Street Name                     51699
Cross Street 1                  57188
Cross Street 2                  57805
Intersection Street 1          313438
Intersection Street 2          314046
Address Type                     3252
City                             2997
Landmark                       364183
Facility Type                    2389
Status                              0
Due Date                            3
Resolution Description              0
Resolution Action Updated Date   2402
Community Board                     0
Borough                             0
X Coordinate (State Plane)       4030
Y Coordinate (State Plane)       4030
Park Facility Name                  0
Park Borough                        0
School Name                         0
School Number                       0
School Region                       1
School Code                         1
School Phone Number                 0
School Address                      0
School City                         0
School State                        0
School Zip                          1
School Not Found                    0
School or Citywide Complaint   364558
Vehicle Type                   364558
Taxi Company Borough           364558
Taxi Pick Up Location          364558
Bridge Highway Name            364261
Bridge Highway Direction       364261
Road Ramp                      364296
Bridge Highway Segment         364296
Garage Lot Name                364558
Ferry Direction                364557
Ferry Terminal Name            364556
Latitude                         4030
Longitude                        4030
Location                         4030
dtype: int64
```

In [7]:

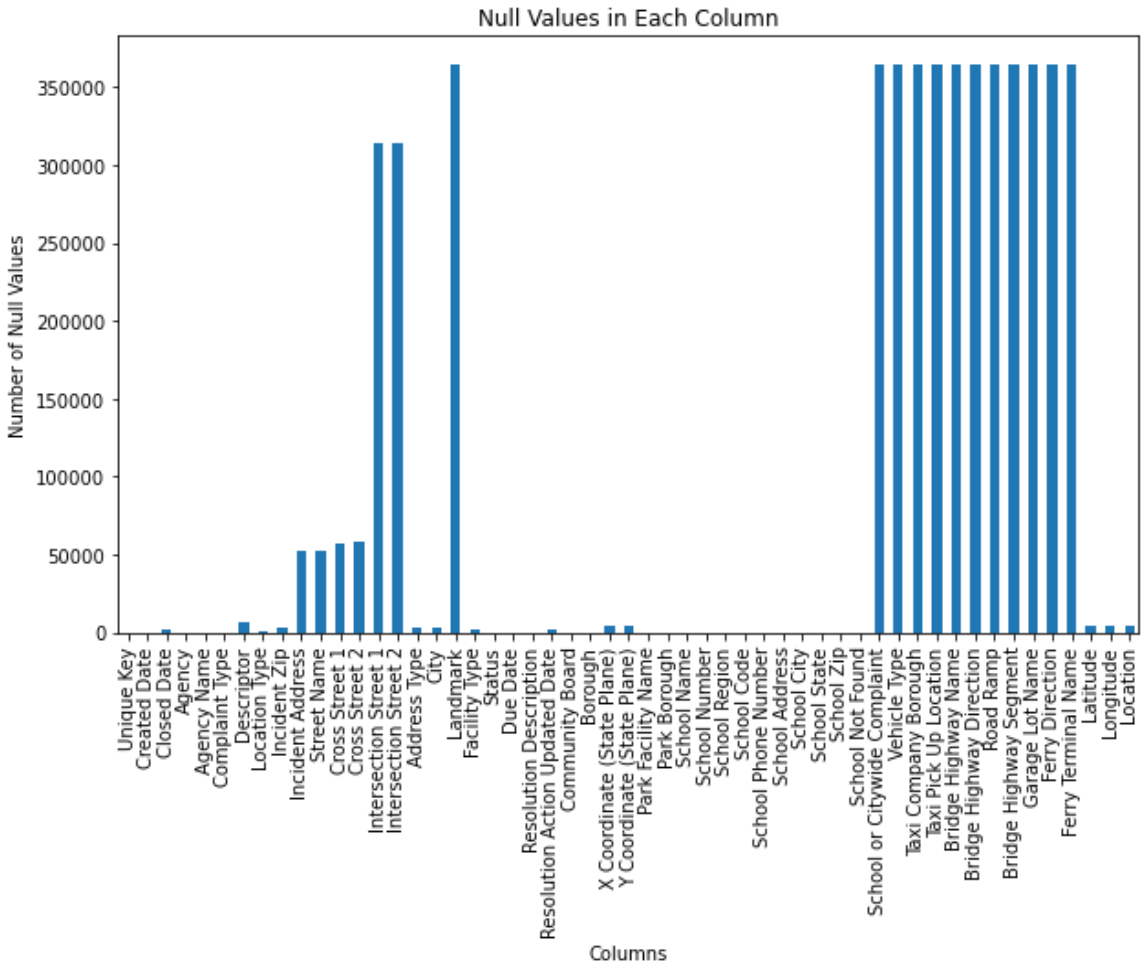
```
df.describe()
```

Out[7]:

	Unique Key	Incident Zip	X Coordinate (State Plane)	Y Coordinate (State Plane)	School or Citywide Complaint	Vehicle Type	T Compa Borou
count	3.645580e+05	361560.000000	3.605280e+05	360528.000000	0.0	0.0	
mean	3.106595e+07	10858.496659	1.005043e+06	203425.305782	NaN	NaN	N
std	7.331531e+05	578.263114	2.196362e+04	29842.192857	NaN	NaN	N
min	2.960737e+07	83.000000	9.133570e+05	121185.000000	NaN	NaN	N
25%	3.049938e+07	10314.000000	9.919460e+05	182945.000000	NaN	NaN	N
50%	3.108795e+07	11209.000000	1.003470e+06	201023.000000	NaN	NaN	N
75%	3.167433e+07	11238.000000	1.019134e+06	222790.000000	NaN	NaN	N
max	3.231065e+07	11697.000000	1.067186e+06	271876.000000	NaN	NaN	N

In [8]:

```
null_counts = df.isnull().sum()  # Count null values in each column
plt.figure(figsize=(10, 6))  # Set the figure size
null_counts.plot(kind='bar')  # Create a bar plot
plt.xlabel('Columns')
plt.ylabel('Number of Null Values')
plt.title('Null Values in Each Column')
plt.show()
```



In [9]:

```
df_1 = df.drop(columns=['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'],axis=1)
```

In [10]:

```
df_1 = df_1.drop(columns=['Vehicle Type','Taxi Company Borough','Taxi Pick Up Locatio
n'],axis=1)
```

In [11]:

```
df_1 = df_1.drop(columns=['Garage Lot Name','Ferry Direction','Ferry Terminal Name'],ax
is=1)
```

In [12]:

```
#Remove the records whose Closed Date values are null
df_1.dropna(subset=['Closed Date'], inplace=True)
```

In [13]:

```
#now to add a new col 'calculated date'
df_1['Closed Date'] = pd.to_datetime(df_1['Closed Date'])
df_1['Created Date'] = pd.to_datetime(df_1['Created Date'])

df_1['Request_Closing_Time'] = df_1['Closed Date'] - df_1['Created Date']
```

In [45]:

```
df_1['Request_Closing_Time_Mins']= df_1['Request_Closing_Time']/np.timedelta64(1,'m')
```

In [14]:

```
df_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 362177 entries, 0 to 364557
Data columns (total 37 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   Unique Key                      362177 non-null  int64
 1   Created Date                    362177 non-null  datetime64[ns]
 2   Closed Date                     362177 non-null  datetime64[ns]
 3   Agency                          362177 non-null  object
 4   Agency Name                     362177 non-null  object
 5   Complaint Type                  362177 non-null  object
 6   Descriptor                      355681 non-null  object
 7   Location Type                   362047 non-null  object
 8   Incident Zip                    361502 non-null  float64
 9   Incident Address                310491 non-null  object
 10  Street Name                     310491 non-null  object
 11  Cross Street 1                  306846 non-null  object
 12  Cross Street 2                  306713 non-null  object
 13  Intersection Street 1           50628 non-null   object
 14  Intersection Street 2           50504 non-null   object
 15  Address Type                    361248 non-null  object
 16  City                            361503 non-null  object
 17  Landmark                        375 non-null     object
 18  Facility Type                   362159 non-null  object
 19  Status                          362177 non-null  object
 20  Due Date                        362176 non-null  object
 21  Resolution Description          362177 non-null  object
 22  Resolution Action Updated Date  362138 non-null  object
 23  Community Board                 362177 non-null  object
 24  Borough                         362177 non-null  object
 25  X Coordinate (State Plane)      360470 non-null  float64
 26  Y Coordinate (State Plane)      360470 non-null  float64
 27  Park Facility Name              362177 non-null  object
 28  Park Borough                    362177 non-null  object
 29  Bridge Highway Name             297 non-null     object
 30  Bridge Highway Direction        297 non-null     object
 31  Road Ramp                       262 non-null     object
 32  Bridge Highway Segment          262 non-null     object
 33  Latitude                        360470 non-null  float64
 34  Longitude                       360470 non-null  float64
 35  Location                        360470 non-null  object
 36  Request_Closing_Time            362177 non-null  timedelta64[ns]
dtypes: datetime64[ns](2), float64(5), int64(1), object(28), timedelta64[n
s](1)
memory usage: 105.0+ MB
```

In [15]:

```
df_1['Request_Closing_Time'].describe()
```

Out[15]:

```
count                     362177
mean     0 days 04:11:53.299632
std      0 days 05:51:42.547519
min             0 days 00:01:01
25%             0 days 01:15:33
50%             0 days 02:40:16
75%             0 days 05:14:38
max            24 days 16:52:22
Name: Request_Closing_Time, dtype: object
```

In [16]:

```
df_1['Complaint Type'].isna().sum()
```

Out[16]:

```
0
```

In [17]:

```
df_1['City'].isna().sum()
```

Out[17]:

```
674
```

In [18]:

```
df_1['City'].fillna('Unknown City',inplace=True)
```

In [19]:

```
df_1['City'].isna().sum()
```

Out[19]:

```
0
```

In [20]:

```
df_1.sample(4)
```

Out[20]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location T
67797	31822508	2015-10-25 16:42:50	2015-10-26 01:49:39	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidev
88275	31697311	2015-10-06 22:10:17	2015-10-06 23:08:22	NYPD	New York City Police Department	Noise - House of Worship	Loud Music/Party	Hous Wors
262095	30573809	2015-05-08 18:46:43	2015-05-08 18:57:03	NYPD	New York City Police Department	Vending	Unlicensed	Street/Sidev
177880	31113475	2015-07-19 17:10:34	2015-07-19 17:42:23	NYPD	New York City Police Department	Noise - Vehicle	Car/Truck Music	Street/Sidev

In [21]:

```
df_1['City'].unique()
```

Out[21]:

```
array(['NEW YORK', 'ASTORIA', 'BRONX', 'ELMHURST', 'BROOKLYN',
       'KEW GARDENS', 'JACKSON HEIGHTS', 'MIDDLE VILLAGE', 'REGO PARK',
       'SAINT ALBANS', 'JAMAICA', 'SOUTH RICHMOND HILL', 'Unknown City',
       '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'], dtype=object)
```

In [22]:

```
unknown_cities = df_1['City'].value_counts()['Unknown City']
unknown_cities
```

Out[22]:

```
674
```

In [23]:

```
df_1.columns
```

Out[23]:

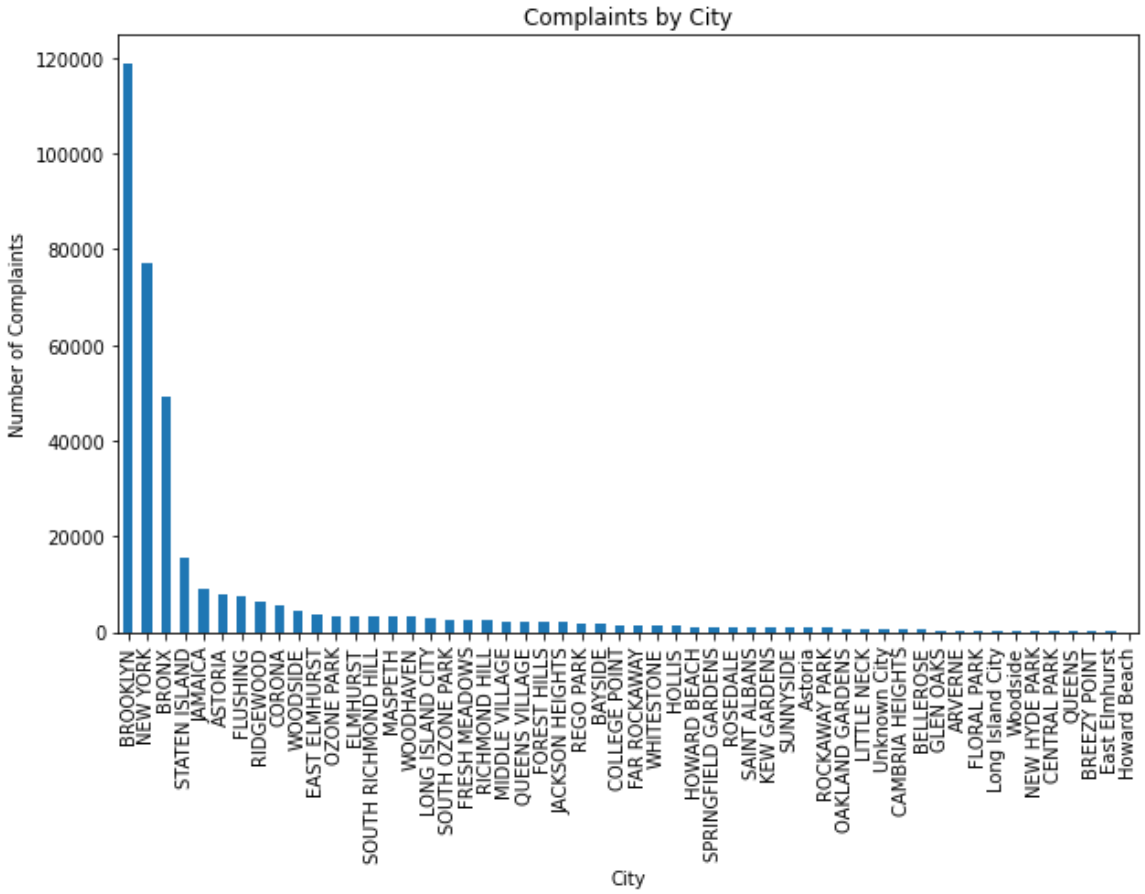
```
Index(['Unique Key', 'Created Date', 'Closed Date', 'Agency', 'Agency Nam
e',
       'Complaint Type', 'Descriptor', 'Location Type', 'Incident Zip',
       'Incident Address', 'Street Name', 'Cross Street 1', 'Cross Street
2',
       'Intersection Street 1', 'Intersection Street 2', 'Address Type',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

In [24]:

```
complaints_by_city = df_1.groupby('City')['Complaint Type'].count().sort_values(ascendi
ng=False)
```

In [25]:

```
plt.figure(figsize=(10, 6))  # Set the figure size
complaints_by_city.plot(kind='bar')
plt.xlabel('City')
plt.ylabel('Number of Complaints')
plt.title('Complaints by City')
plt.show()
```



In [26]:

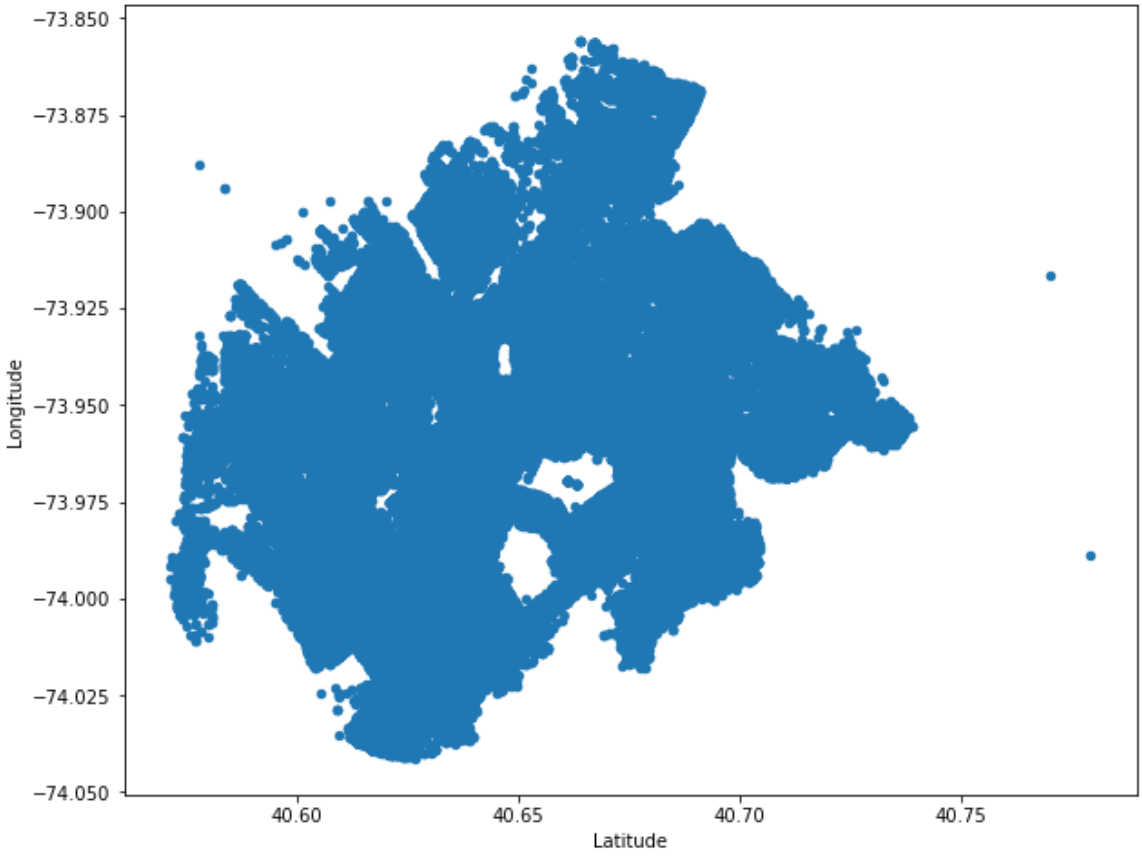
```
comp_brook = df_1.loc[df['City']=='BROOKLYN']
```

In [27]:

```
comp_brook[['Latitude','Longitude']].plot(kind='scatter',x='Latitude',y='Longitude',fig
size = (10, 8))
```

Out[27]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a38e9a4610>
```

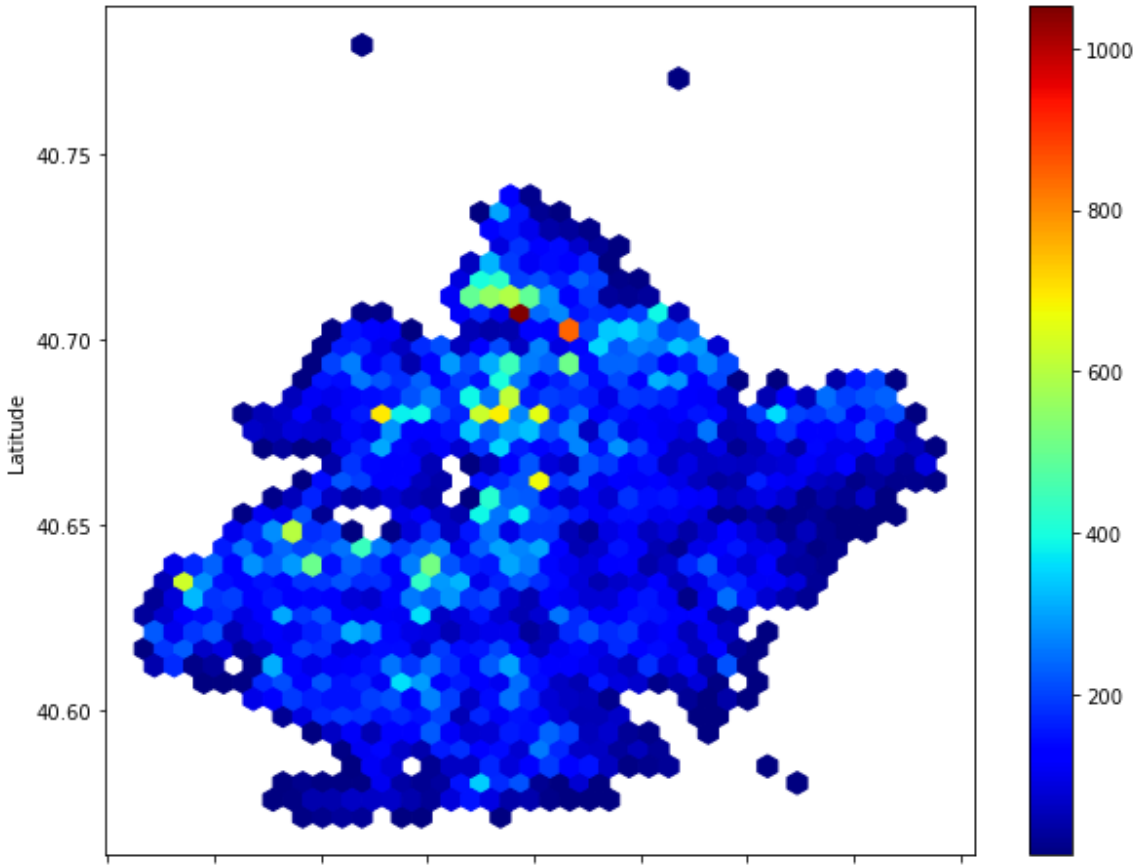


In [28]:

```
comp_brook[['Longitude','Latitude']].plot(kind='hexbin',x='Longitude',y='Latitude',grid
size=40,colormap='jet',mincnt=1,figsize = (10, 8))
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a38f304f70>
```

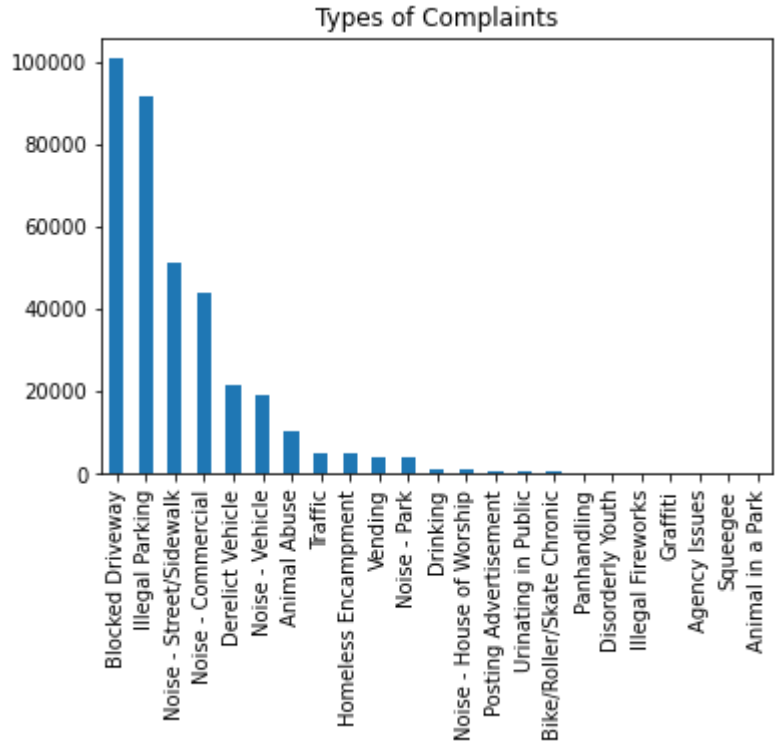


In [29]:

```
#types of complaints
df_1['Complaint Type'].value_counts().plot(kind='bar',title='Types of Complaints')
```

Out[29]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a38f3af7c0>
```

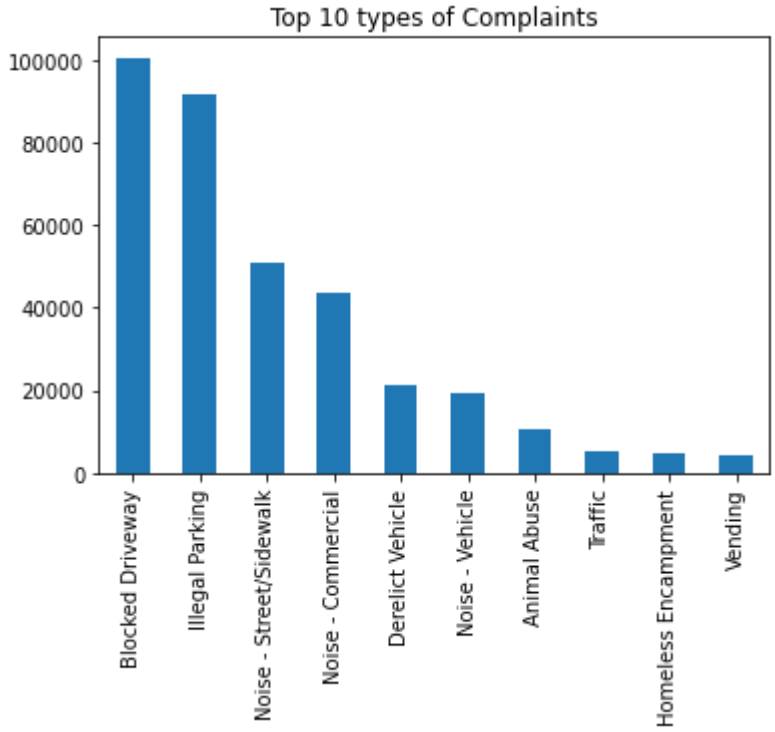


In [30]:

```
#Top 10 Complaint types
df_1['Complaint Type'].value_counts().head(10).plot(kind='bar',title='Top 10 types of C
omplaints')
```

Out[30]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a38fc55cd0>
```

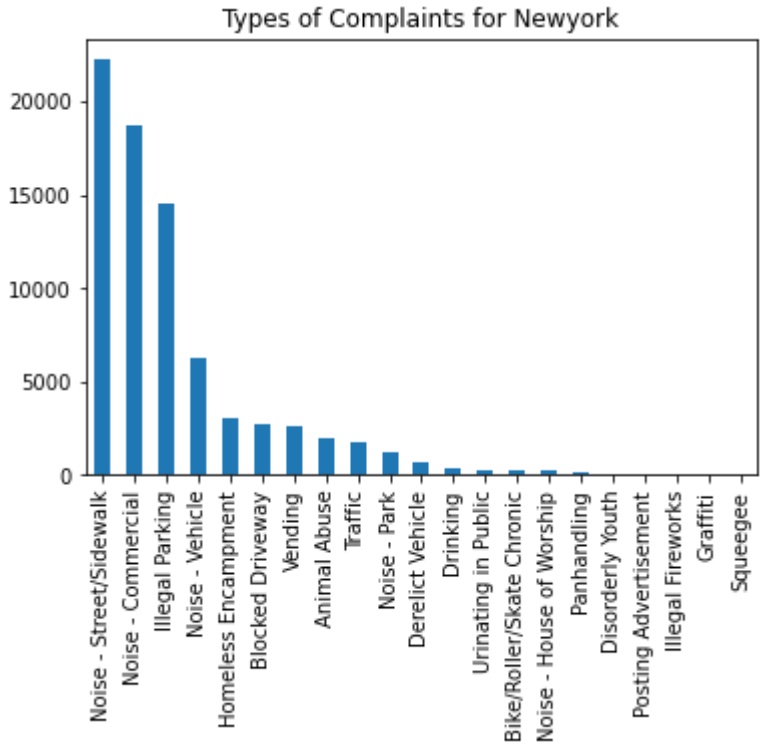


In [31]:

```
#various types of complaints for NEw York city
comp_ny = df_1.loc[df['City']=='NEW YORK']
comp_ny['Complaint Type'].value_counts().plot(kind='bar',title='Types of Complaints for Newyork')
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a38fcc7070>
```

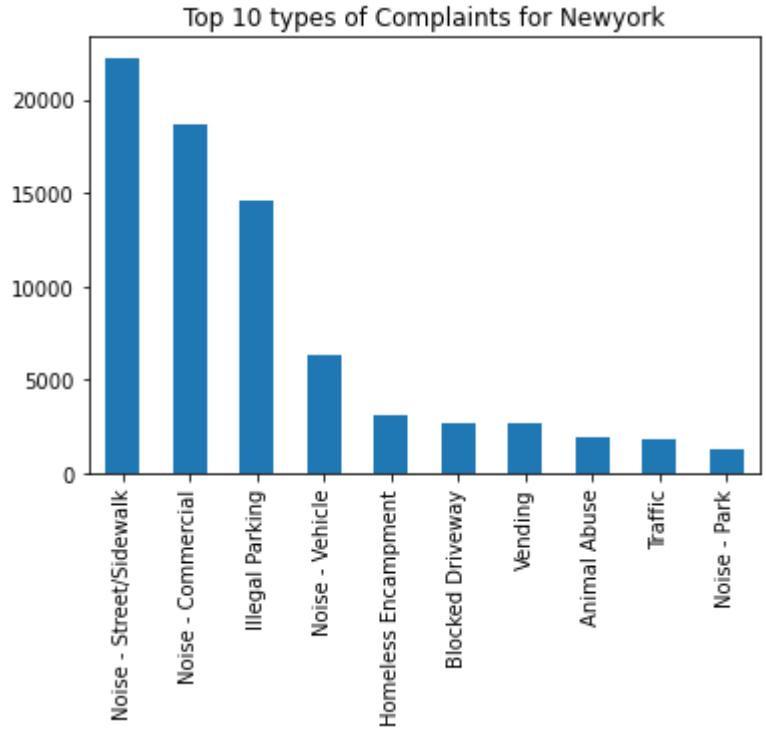


In [32]:

```
#Top 10 various types of complaints for NEw York city
comp_ny['Complaint Type'].value_counts().head(10).plot(kind='bar',title='Top 10 types o
f Complaints for Newyork')
```

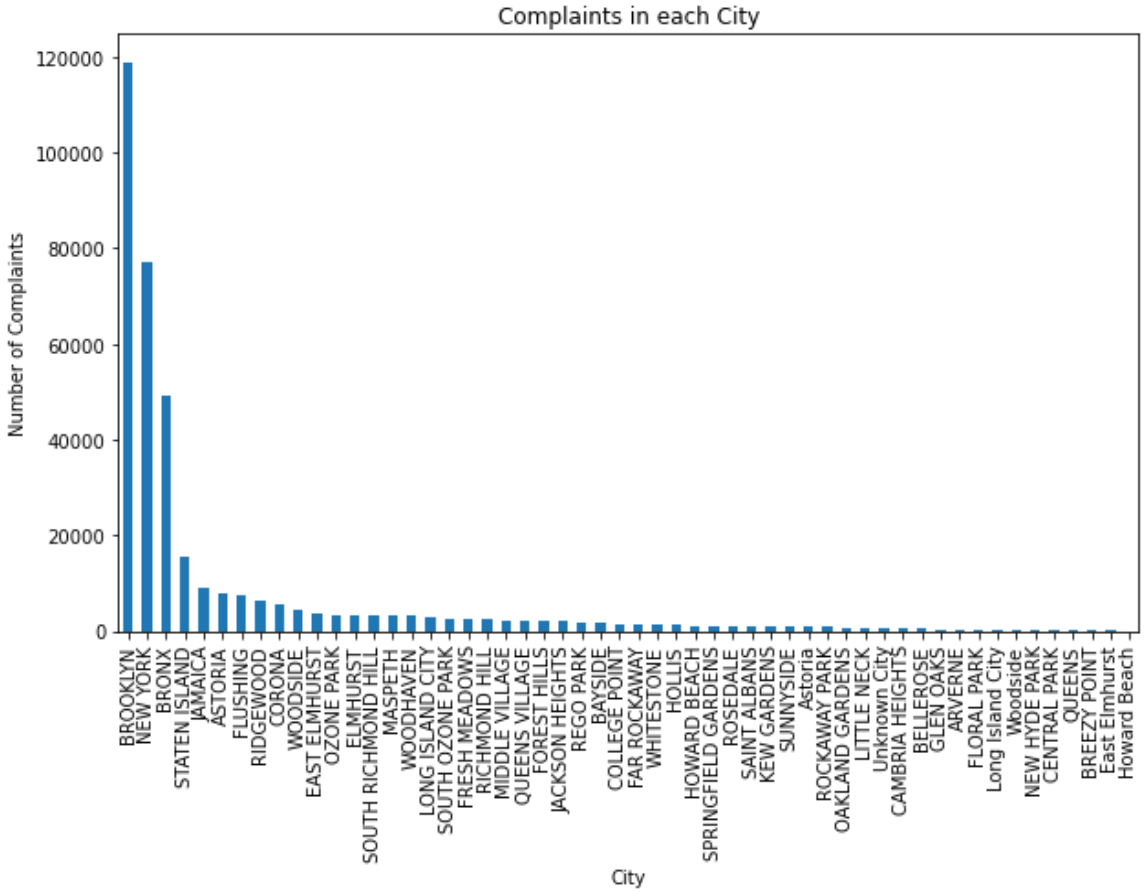
Out[32]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a38f1db670>
```



In [33]:

```
#various types of complaints in each city
complaints_by_city = df_1.groupby('City')['Complaint Type'].count().sort_values(ascendi
ng=False)
plt.figure(figsize=(10, 6))  # Set the figure size
complaints_by_city.plot(kind='bar')
plt.xlabel('City')
plt.ylabel('Number of Complaints')
plt.title('Complaints in each City')
plt.show()
```



In [ ]:

```

```

In [34]:

```
df_new=df.pivot_table(index='Complaint Type',columns='City')
df_new
```

Out[34]:

	Incident Zip					
City	ARVERNE	ASTORIA	Astoria	BAYSIDE	BELLEROSE	BREEZY POINT
Complaint Type						
Animal Abuse	11692.0	11104.264706	NaN	11360.811321	11426.0	11697.0
Animal in a Park	NaN	NaN	NaN	NaN	NaN	NaN
Bike/Roller/Skate Chronic	NaN	11103.250000	NaN	NaN	11426.0	NaN
Blocked Driveway	11692.0	11103.845169	11104.113208	11360.826848	11426.0	11697.0
Derelict Vehicle	11692.0	11104.016432	11103.428571	11360.658009	11426.0	11697.0
Disorderly Youth	11692.0	11104.400000	NaN	11361.000000	11426.0	NaN
Drinking	11692.0	11103.837209	NaN	11361.000000	11426.0	11697.0
Graffiti	11692.0	11104.000000	NaN	11361.000000	NaN	NaN
Homeless Encampment	11692.0	11104.625000	NaN	11360.500000	11426.0	NaN
Illegal Fireworks	NaN	11105.500000	NaN	NaN	11426.0	NaN
Illegal Parking	11692.0	11104.214925	11104.007220	11360.619122	11426.0	11697.0
Noise - Commercial	11692.0	11104.035088	11103.517685	11360.872340	11426.0	11697.0
Noise - House of Worship	11692.0	11102.904762	NaN	11361.000000	11426.0	NaN
Noise - Park	11692.0	11103.750000	NaN	11360.250000	11426.0	NaN
Noise - Street/Sidewalk	11692.0	11104.051345	11104.317241	11360.823529	11426.0	11697.0
Noise - Vehicle	11692.0	11104.135593	NaN	11360.666667	11426.0	11697.0
Panhandling	11692.0	11104.500000	NaN	NaN	11426.0	NaN
Posting Advertisement	NaN	11102.666667	NaN	NaN	11426.0	NaN
Squeegee	NaN	NaN	NaN	NaN	NaN	NaN
Traffic	11692.0	11108.233333	NaN	11360.888889	11426.0	NaN
Urinating in Public	11692.0	11103.600000	NaN	NaN	11426.0	NaN
Vending	11692.0	11104.210526	NaN	11360.000000	NaN	NaN

In [35]:

```
df_new = pd.crosstab(df_1['Complaint Type'], df_1['City'])
df_new
```

Out[35]:

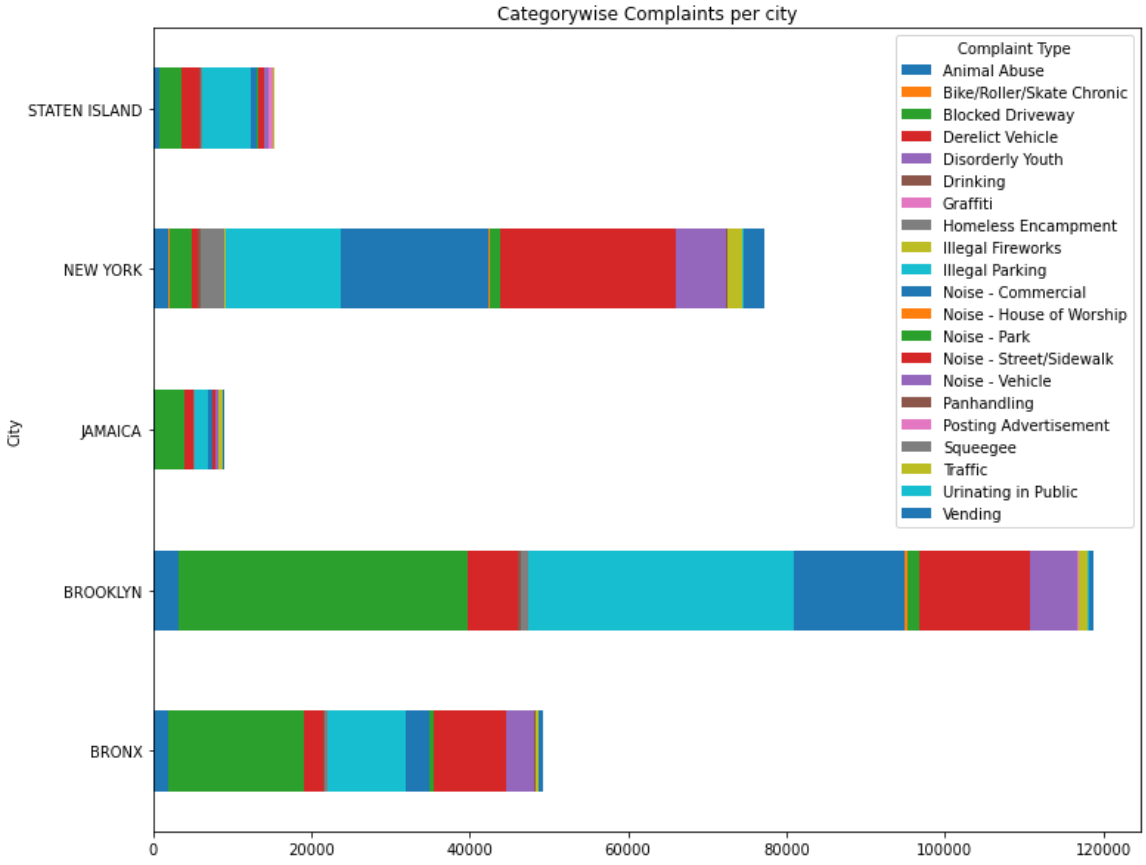
City	ARVERNE	ASTORIA	Astoria	BAYSIDE	BELLEROSE	BREEZY POINT	BRONX	BI
Complaint Type								
Agency Issues	0	0	0	0	0	0	0	
Animal Abuse	46	170	0	53	15	2	1971	
Animal in a Park	0	0	0	0	0	0	0	
Bike/Roller/Skate Chronic	0	16	0	0	1	0	22	
Blocked Driveway	50	3436	159	514	138	3	17062	
Derelict Vehicle	32	426	14	231	120	3	2402	
Disorderly Youth	2	5	0	2	2	0	66	
Drinking	1	43	0	1	1	1	206	
Graffiti	1	4	0	3	0	0	15	
Homeless Encampment	4	32	0	2	1	0	275	
Illegal Fireworks	0	4	0	0	1	0	24	
Illegal Parking	62	1340	277	638	132	16	9889	
Noise - Commercial	2	1653	310	47	38	4	2944	
Noise - House of Worship	14	21	0	3	1	0	90	
Noise - Park	2	64	0	4	1	0	548	
Noise - Street/Sidewalk	29	409	145	17	13	1	9144	
Noise - Vehicle	10	236	0	24	11	1	3556	
Panhandling	1	2	0	0	1	0	20	
Posting Advertisement	0	3	0	0	1	0	18	
Squeegee	0	0	0	0	0	0	0	
Traffic	1	60	0	9	9	0	427	
Urinating in Public	1	10	0	0	1	0	54	
Vending	1	57	0	2	0	0	433	

In [37]:

```
top5cities = df_1['City'].value_counts().head(5).index.to_list()
dstop5 = df_1[df_1.City.isin(top5cities)]
#citywise complaint counts(typewise)
df_new_1 = pd.crosstab(dstop5['City'],dstop5['Complaint Type'])
```

In [38]:

```
##citywise complaint counts(typewise)
df_new_1.plot(kind='barh',stacked=True,figsize=(12,10))
plt.title('Categorywise Complaints per city')
plt.show()
```



In [52]:

```
# Storing mean response time for various complaint types
complaintTypes = df_1['Complaint Type'].unique()

for i in range(len(complaintTypes)):
    exec("c{} = df_1.loc[(df_1['Complaint Type'] == '{}'),'Request_Closing_Time_Min
s']".format(i+1,complaintTypes[i]))
```

In [55]:

```
#Performing F-statics
fscore,pvalue = stats.f_oneway(c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16)
print(fscore)
print(pvalue)
```

```
603.3206722071288
0.0
```

In [56]:

```
#As pvalue is less than alpha(0.05) we reject null hypothesis hence
#Reject H0: One or more sample distributions are not equal
```

In [ ]:

```

```