In [1]:

```
import pandas as pd
import datetime as dt

##1)Import a 311 NYC service request.
df=pd.read_csv('311_Service_Requests_from_2010_to_Present.csv',low_memory=False)

##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 a
nd request closing
df['Closed Date']=pd.to_datetime(df['Closed Date'])
df['Created Date']=pd.to_datetime(df['Created Date'])
```

In [2]:

```
df['Request_Closing_Time']=(df['Closed Date'] - df['Created Date']).dt.total_seconds()
```

In [3]:

```
df.shape
```

Out[3]:

```
(300698, 54)
```

In [7]:

```
###Checking the Null values
df.Request_Closing_Time.isnull().sum()
```

Out[7]:

```
2164
```

In [8]:

```
#####Data Imputation
df=df[df['Request_Closing_Time'].notnull()]
```

In [9]:

```
df.shape
```

Out[9]:

```
(298534, 54)
```

In [20]:

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

In [21]:

```
##A.Complaints based on Location Type
df1=df['Location Type'].value_counts()
df1
```

Out[21]:

```
Street/Sidewalk               247503
Store/Commercial               20183
Club/Bar/Restaurant            17227
Residential Building/House      6953
Park/Playground                 4751
House of Worship                 927
Residential Building             227
Highway                          214
```

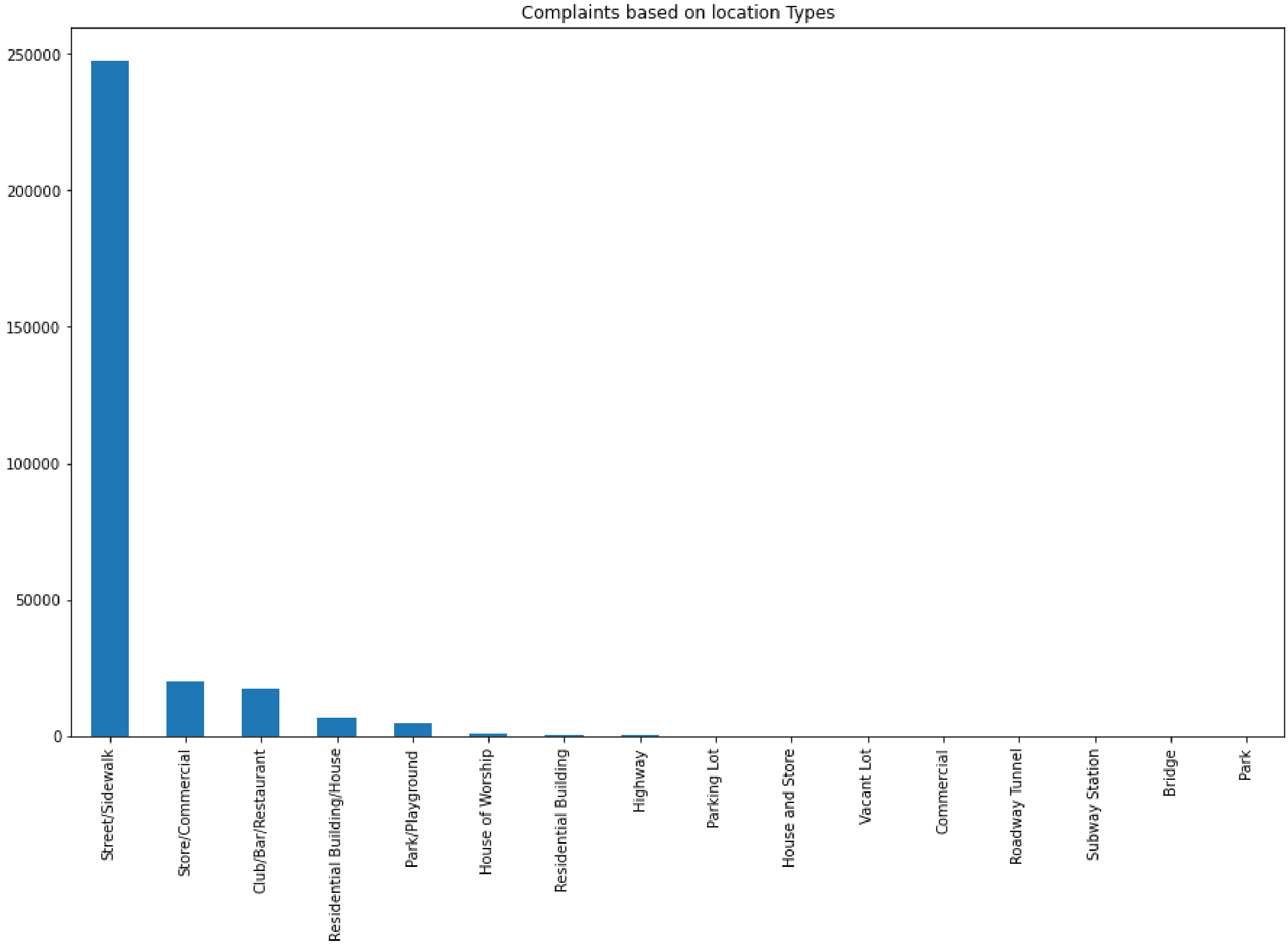
```
Parking Lot                    117
House and Store                 93
Vacant Lot                      77
Commercial                      62
Roadway Tunnel                  35
Subway Station                  34
Bridge                           2
Park                             1
Name: Location Type, dtype: int64
```

In [29]:

```python
df1.plot(kind='bar',figsize=(15,9),title='Complaints based on location Types')
```

Out[29]:

```
<AxesSubplot:title={'center':'Complaints based on location Types'}>
```



In [23]:

```python
###Insights on complaints
df2=df['Complaint Type'].value_counts()
df2
```

Out[23]:

```
Blocked Driveway             76810
Illegal Parking              74532
Noise - Street/Sidewalk      48076
Noise - Commercial           35247
Derelict Vehicle             17588
Noise - Vehicle              17033
Animal Abuse                  7768
Traffic                       4496
Homeless Encampment           4416
Noise - Park                  4022
Vending                       3795
Drinking                      1275
```

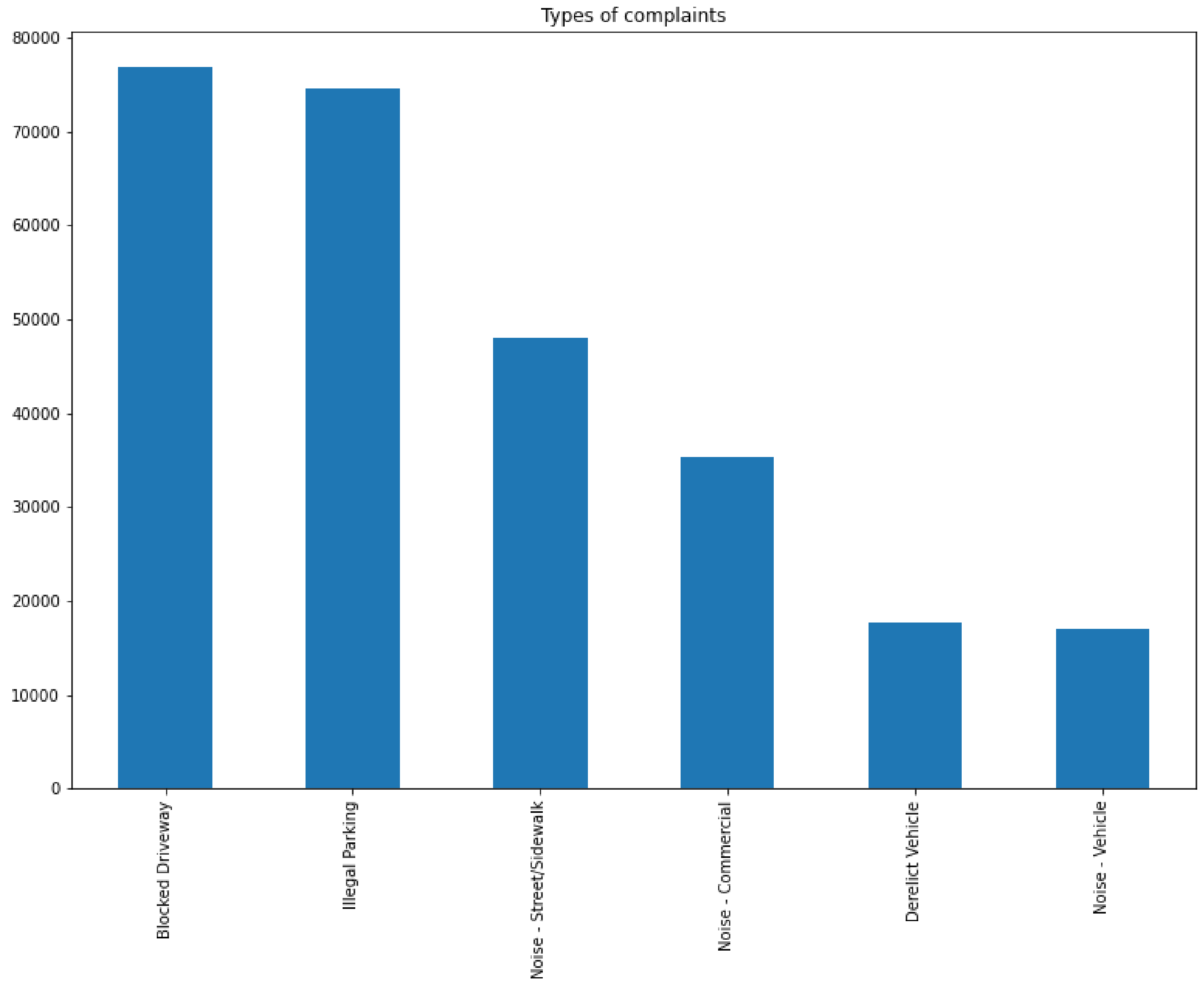
```
Drinking                      1275
Noise - House of Worship       929
Posting Advertisement          648
Urinating in Public            592
Bike/Roller/Skate Chronic      424
Panhandling                    305
Disorderly Youth               286
Illegal Fireworks              168
Graffiti                       113
Agency Issues                    6
Squeegee                         4
Animal in a Park                 1
Name: Complaint Type, dtype: int64
```

In [28]:

```
###Insights on Top 6 complaints
(df['Complaint Type'].value_counts()).head(6).plot(kind='bar',figsize=(13,9),title='Type
s of complaints')
```

Out[28]:

```
<AxesSubplot:title={'center':'Types of complaints'}>
```



In [35]:

```
###Analysis based on Complaints from each City
df3=df['City'].value_counts()
df3
```

Out[35]:

```
BROOKLYN             98295
NEW YORK             65972
```

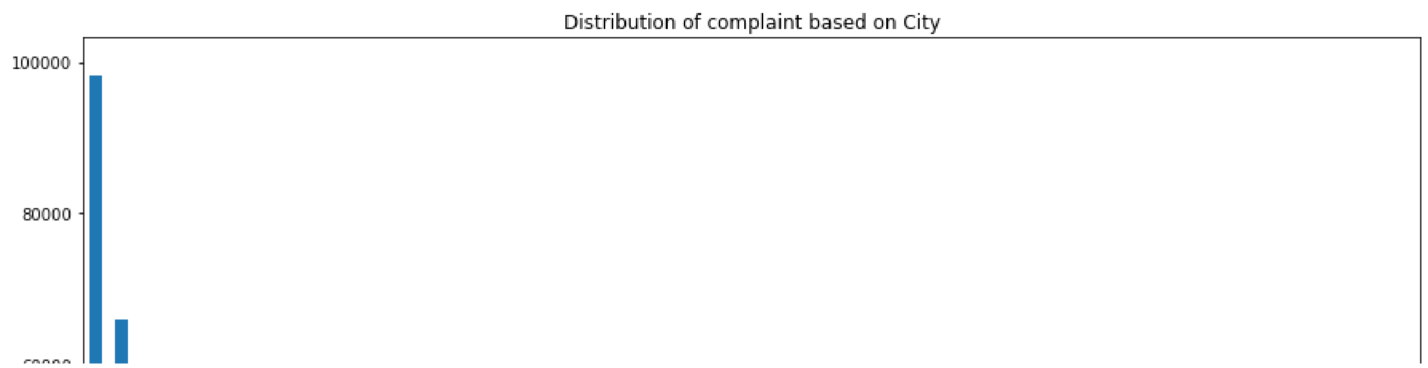
```
BRONX                      40697
STATEN ISLAND              12338
JAMAICA                     7294
ASTORIA                     6330
FLUSHING                    5970
RIDGEWOOD                   5162
CORONA                      4295
WOODSIDE                    3544
SOUTH RICHMOND HILL         2774
OZONE PARK                  2755
EAST ELMHURST               2733
ELMHURST                    2673
WOODHAVEN                   2463
MASPETH                     2461
LONG ISLAND CITY            2436
SOUTH OZONE PARK            2173
RICHMOND HILL               1902
FRESH MEADOWS               1899
QUEENS VILLAGE              1814
MIDDLE VILLAGE              1765
JACKSON HEIGHTS             1688
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                      716
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
```

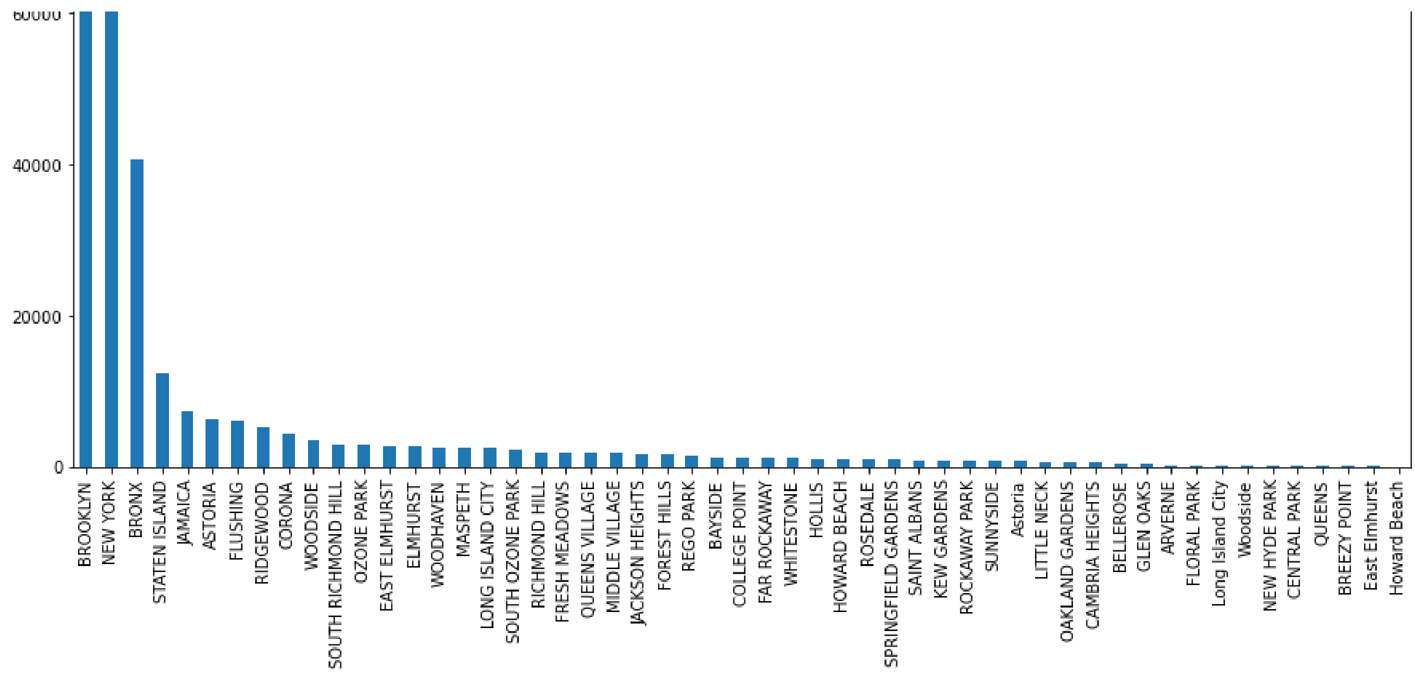
In [36]:

```
df3.plot(kind='bar',figsize=(15,9),title='Distribution of complaint based on City')
```

Out[36]:

```
<AxesSubplot:title={'center':'Distribution of complaint based on City'}>
```



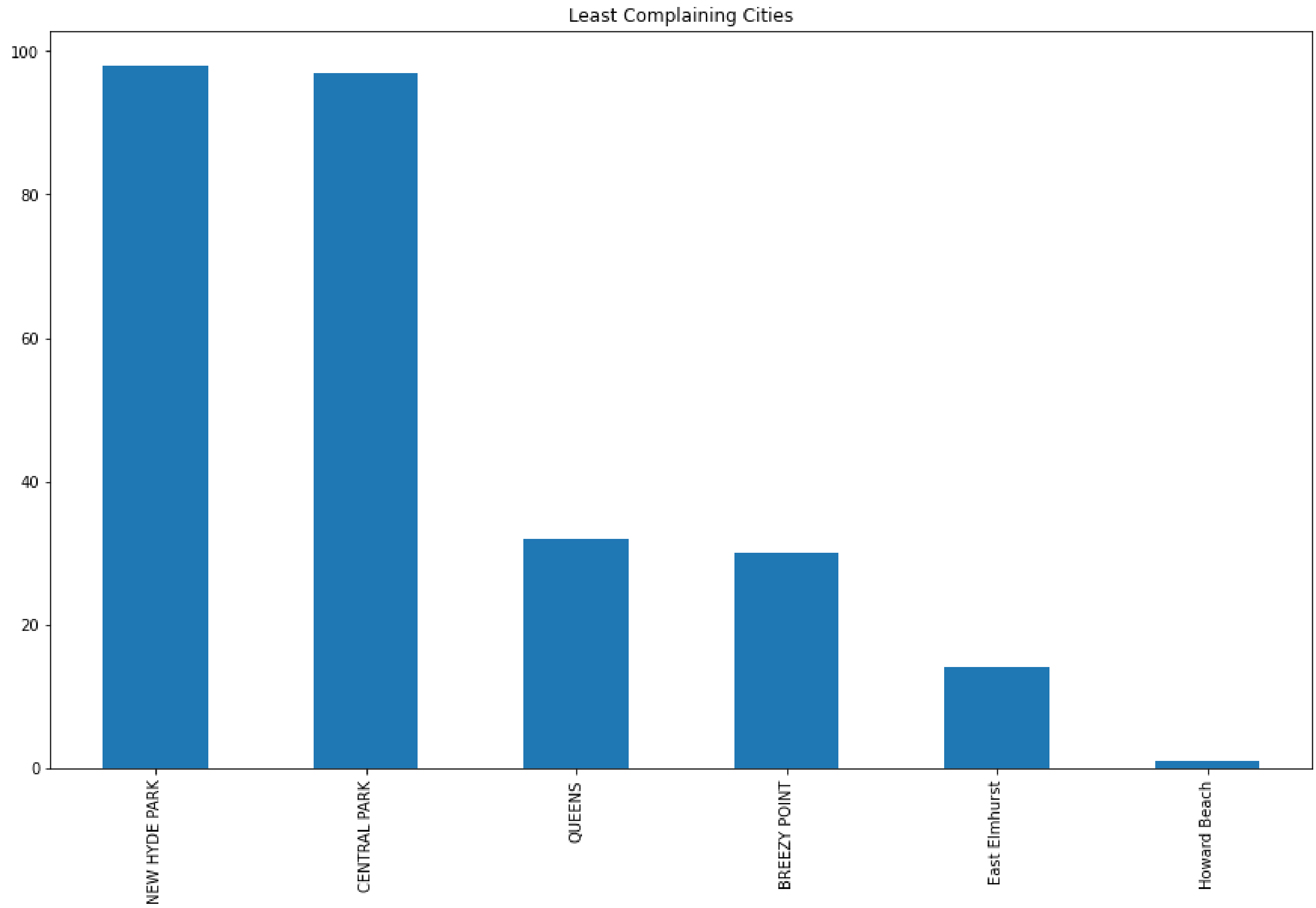


In [38]:

```
###Insights on Top 6 complaints
(df['City'].value_counts()).tail(6).plot(kind='bar',figsize=(15,9),title='Least Complain
ing Cities')
```

Out[38]:

```
<AxesSubplot:title={'center':'Least Complaining Cities'}>
```

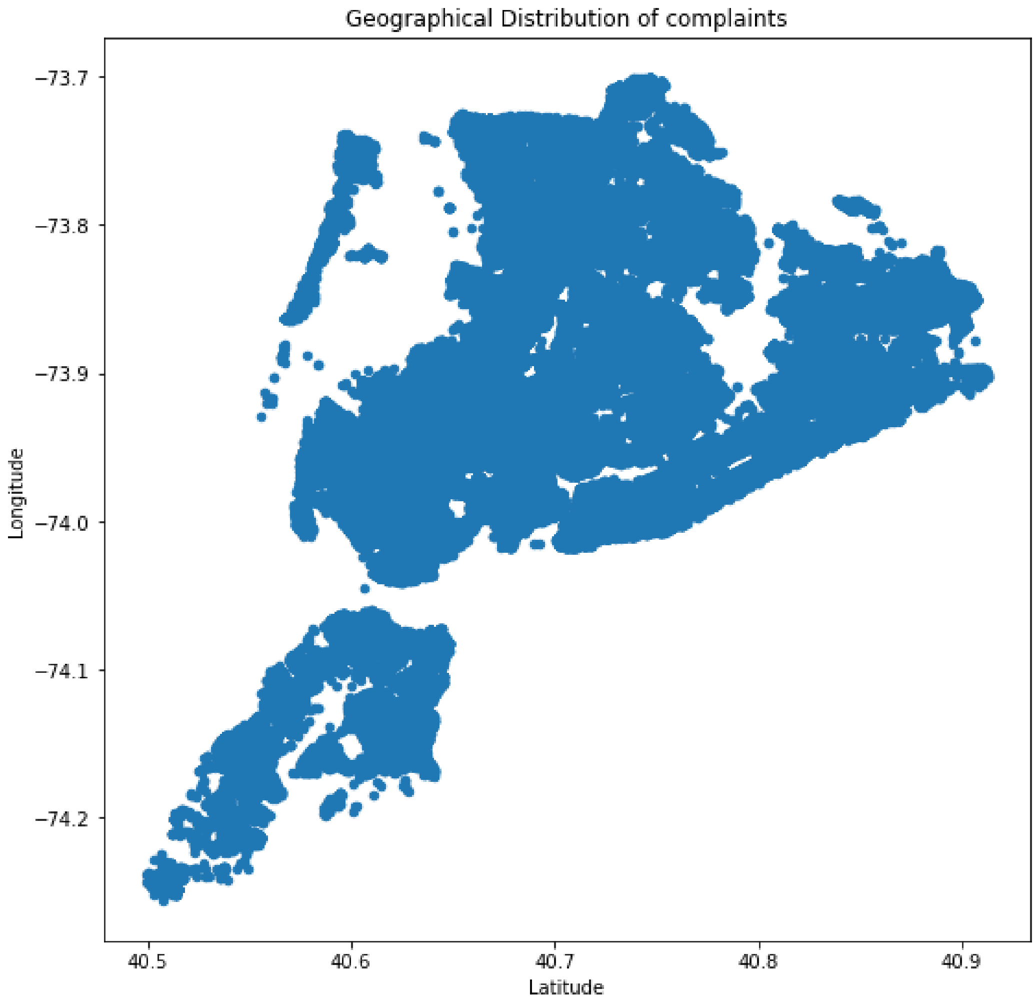


In [43]:

```
####Geographical Distribution of complaints
df.plot(kind='scatter',x='Latitude',y='Longitude',figsize=(9,9),title='Geographical Dist
ribution of complaints')
```

Out[43]:

```
<AxesSubplot:title={'center':'Geographical Distribution of complaints'}, xlabel='Latitude
', ylabel='Longitude'>
```



In [44]:

```
### (4) Order the complaint types based on the average 'Request_Closing_Time', grouping t
hem for different locations.
```

In [96]:

```
df_new=df.groupby(['City','Complaint Type']).Request_Closing_Time.mean()
df_new.head(20)
```

Out[96]:

```
City      Complaint Type
ARVERNE   Animal Abuse                    7753.052632
          Blocked Driveway                9093.485714
          Derelict Vehicle               10685.592593
          Disorderly Youth               12928.500000
          Drinking                         859.000000
          Graffiti                        5520.000000
          Homeless Encampment             6533.250000
          Illegal Parking                 8338.913793
          Noise - Commercial              8234.000000
          Noise - House of Worship        5623.909091
          Noise - Park                    4620.000000
          Noise - Street/Sidewalk         7172.620690
          Noise - Vehicle                 6695.571429
          Panhandling                     3720.000000
          Urinating in Public             2491.000000
          Vending                         1740.000000
ASTORIA   Animal Abuse                   18000.608000
          Bike/Roller/Skate Chronic       6261.533333
          Blocked Driveway               17338.024064
          Derelict Vehicle               34881.299145
Name: Request_Closing_Time, dtype: float64
```

In [101]:

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

In [104]:

```
##ANOVA
#h0: mu(Average response time) = mu(Complaint Type)
#hA: mu(Average response time) =! mu(Complaint Type)
from statsmodels.formula.api import ols
import statsmodels.api as sm
df['Complaint_Type']=df['Complaint Type']
```

In [105]:

```
model=ols('Request_Closing_Time ~ Complaint_Type',data=df).fit()
anova_mod=sm.stats.anova_lm(model)
anova_mod
```

Out[105]:

	df	sum_sq	mean_sq	F	PR(>F)
Complaint_Type	22.0	5.238177e+12	2.380990e+11	514.177089	0.0
Residual	298511.0	1.382309e+14	4.630680e+08	NaN	NaN

In [106]:

```
##Null hypothesis is rejected as pvalue < 0.05 .
##So,Average response time across complaints are not similar
```

In [107]:

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

In [108]:

```
#h0: mu(Location) = mu(Type of complaint requested)
#hA: mu(Location) != mu(Type of complaint requested)
```

In [109]:

```
contingency=pd.crosstab(df['Complaint Type'],df['Location Type'])
contingency
```

Out[109]:

Location Type	Bridge	Club/Bar/Restaurant	Commercial	Highway	House and Store	House of Worship	Park	Park/Playground	Parking Lot	Resi Bu
Complaint Type										
Animal Abuse	0	0	62	0	93	0	0	122	110	
Animal in a Park	0	0	0	0	0	0	1	0	0	
Bike/Roller/Skate Chronic	0	0	0	0	0	0	0	0	0	
Blocked Driveway	0	0	0	0	0	0	0	0	0	
Derelict Vehicle	0	0	0	13	0	0	0	0	0	
Disorderly Youth	0	0	0	0	0	0	0	0	0	
Drinking	0	365	0	0	0	0	0	98	0	
Graffiti	0	0	0	0	0	0	0	0	0	
Homeless Encampment	2	0	0	15	0	0	0	353	0	

Complaint Type	Bridge	Club/Bar/Restaurant	Commercial	Highway	House and Store	House of Worship	Park	Park/Playground	Parking Lot
Illegal Fireworks	0	0	0	0	0	0	0	8	0
Illegal Parking	0	0	0	0	0	0	0	0	0
Noise - Commercial	0	16841	0	0	0	0	0	0	0
Noise - House of Worship	0	0	0	0	0	927	0	0	0
Noise - Park	0	0	0	0	0	0	0	4021	0
Noise - Street/Sidewalk	0	0	0	0	0	0	0	0	0
Noise - Vehicle	0	0	0	0	0	0	0	0	0
Panhandling	0	0	0	0	0	0	0	6	0
Posting Advertisement	0	0	0	0	0	0	0	0	7
Squeegee	0	0	0	0	0	0	0	0	0
Traffic	0	0	0	186	0	0	0	0	0
Urinating in Public	0	21	0	0	0	0	0	38	0
Vending	0	0	0	0	0	0	0	105	0

In [110]:

```
contingency_pcr=pd.crosstab(df['Complaint Type'],df['Location Type'],normalize='index')
contingency_pcr
```

Out[110]:

Location Type Complaint Type	Bridge	Club/Bar/Restaurant	Commercial	Highway	House and Store	House of Worship	Park	Park/Playground	Parking Lot
Animal Abuse	0.000000	0.000000	0.007985	0.000000	0.011977	0.0	0.0	0.015712	0.014166
Animal in a Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	1.0	0.000000	0.000000
Bike/Roller/Skate Chronic	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
Blocked Driveway	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
Derelict Vehicle	0.000000	0.000000	0.000000	0.000739	0.000000	0.0	0.0	0.000000	0.000000
Disorderly Youth	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
Drinking	0.000000	0.286499	0.000000	0.000000	0.000000	0.0	0.0	0.076923	0.000000
Graffiti	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
Homeless Encampment	0.000454	0.000000	0.000000	0.003404	0.000000	0.0	0.0	0.080100	0.000000
Illegal Fireworks	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.047619	0.000000
Illegal Parking	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
Noise - Commercial	0.000000	0.477867	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
Noise - House of Worship	0.000000	0.000000	0.000000	0.000000	0.000000	1.0	0.0	0.000000	0.000000
Noise - Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	1.000000	0.000000
Noise - Street/Sidewalk	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
Noise - Vehicle	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
Panhandling	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.019672	0.000000

Location Type / Complaint Type	Bridge	Club/Bar/Restaurant	Commercial	Highway	House and Store	House of Worship	Park	Park/Playground	Parking Lot
Panhandling	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.019372	0.000000
Posting Advertisement	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.010819
Squeegee	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
Traffic	0.000000	0.000000	0.000000	0.041416	0.000000	0.0	0.0	0.000000	0.000000
Urinating in Public	0.000000	0.035533	0.000000	0.000000	0.000000	0.0	0.0	0.064298	0.000000
Vending	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.027683	0.000000

In [112]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,8))
```

Out[112]:

```
<Figure size 576x576 with 0 Axes>

<Figure size 576x576 with 0 Axes>
```

In [118]:

```
from scipy.stats import chi2_contingency
c,p,dof,expected=chi2_contingency(contingency)
print('pvalue is:' ,p, 'and dof is', dof)
```

```
pvalue is: 0.0 and dof is 315
```

In [ ]:

```
#since pvalue is less than 0.05,H0 is rejected.So,the type of complaint or service reques
ted and location not related.
```