In [155]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

%matplotlib inline
sns.set_style('darkgrid')

import warnings
warnings.filterwarnings('ignore')
```

In [156]:
```
df = pd.read_csv(r"C:\Users\Yogale01\OneDrive - TMF Group\Downloads\Customer_S
```

In [157]:
```
df.head(4)
```

Out[157]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location T
0	32310363	12/31/2015 11:59:45 PM	01/01/2016 12:55:15 AM	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Side
1	32309934	12/31/2015 11:59:44 PM	01/01/2016 01:26:57 AM	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Side
2	32309159	12/31/2015 11:59:29 PM	01/01/2016 04:51:03 AM	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Side
3	32305098	12/31/2015 11:57:46 PM	01/01/2016 07:43:13 AM	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Side

4 rows × 53 columns

In [158]:
```
df.tail(4)
```

Out[158]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Loca
364554	29608392	01/01/2015 12:04:28 AM	01/01/2015 02:25:02 AM	NYPD	New York City Police Department	Noise - Vehicle	Car/Truck Horn	Stree
364555	29607589	01/01/2015 12:01:30 AM	01/01/2015 12:20:33 AM	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Stree
364556	29610889	01/01/2015 12:01:29 AM	01/01/2015 02:42:22 AM	NYPD	New York City Police Department	Blocked Driveway	No Access	Stree
364557	29611816	01/01/2015 12:00:50 AM	01/01/2015 02:47:50 AM	NYPD	New York City Police Department	Blocked Driveway	No Access	Stree

4 rows × 53 columns

In [159]:
```
df.describe()
```

Out[159]:

	Unique Key	Incident Zip	X Coordinate (State Plane)	Y Coordinate (State Plane)	School or Citywide Complaint	Vehicle Type	Tax Company Borough
count	3.645580e+05	361560.000000	3.605280e+05	360528.000000	0.0	0.0	0.(
mean	3.106595e+07	10858.496659	1.005043e+06	203425.305782	NaN	NaN	NaN
std	7.331531e+05	578.263114	2.196362e+04	29842.192857	NaN	NaN	NaN
min	2.960737e+07	83.000000	9.133570e+05	121185.000000	NaN	NaN	NaN
25%	3.049938e+07	10314.000000	9.919460e+05	182945.000000	NaN	NaN	NaN
50%	3.108795e+07	11209.000000	1.003470e+06	201023.000000	NaN	NaN	NaN
75%	3.167433e+07	11238.000000	1.019134e+06	222790.000000	NaN	NaN	NaN
max	3.231065e+07	11697.000000	1.067186e+06	271876.000000	NaN	NaN	NaN

In [160]:
```
df.shape
```

Out[160]: (364558, 53)

In [161]:
```
df.isnull().sum()
```

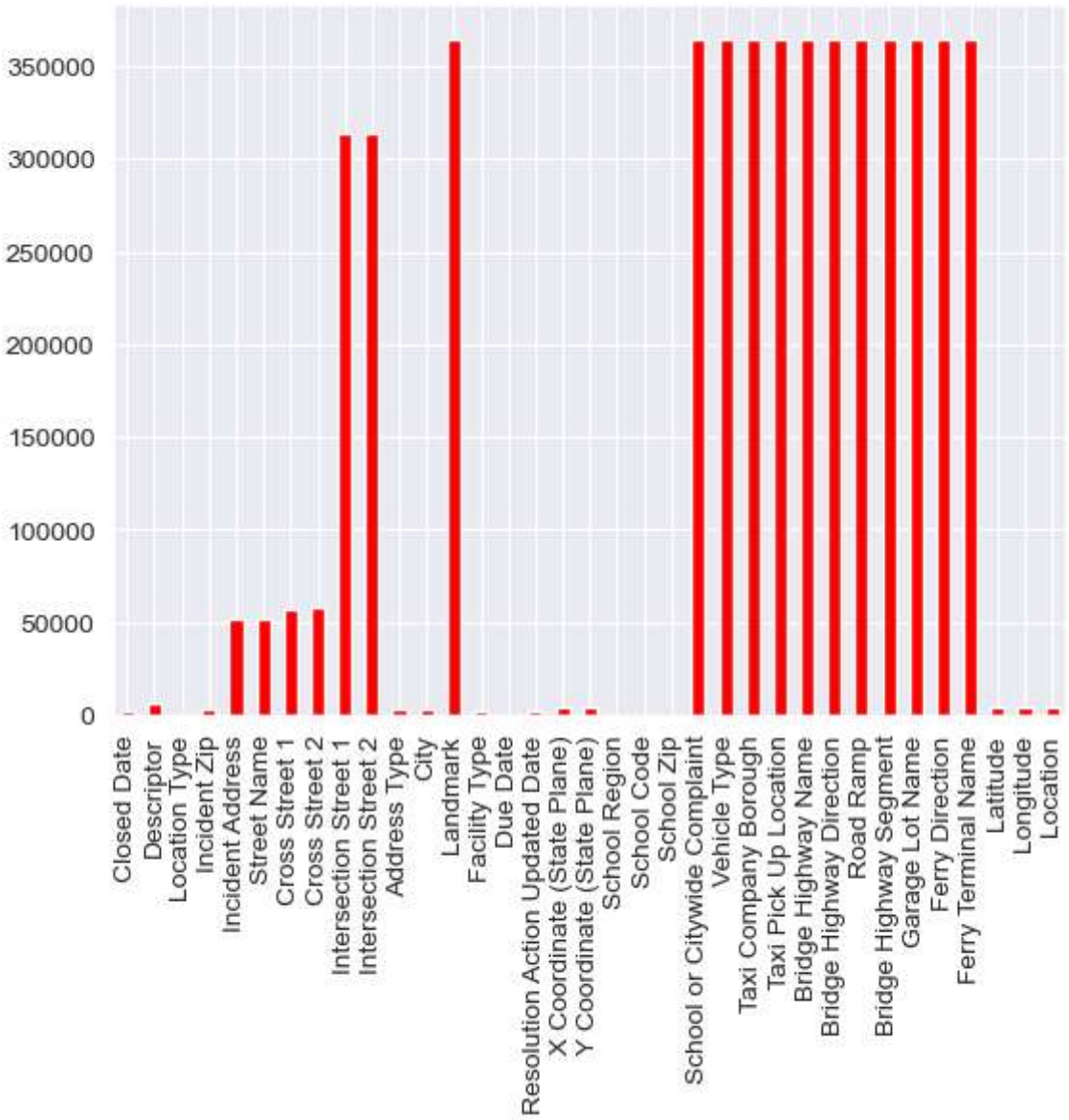
Out[161]:

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 [162]:
```
df.isna().sum() [df.isnull().sum()!=0].plot(kind='bar', color = '#ff0000')
```

Out[162]: <Axes: >



In [163]:
```
df.dropna(subset='Closed Date',inplace =True)
```

In [164]:
```
df['Closed Date']=pd.to_datetime(df['Closed Date'])
```

In [165]:
```
df['Created Date']=pd.to_datetime(df['Created Date'])
```

In [166]:
```
df.head(4)
```

Out[166]:

	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 00:55:15	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 01:26:57	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk
2	32309159	2015-12-31 23:59:29	2016-01-01 04:51:03	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk
3	32305098	2015-12-31 23:57:46	2016-01-01 07:43:13	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk

4 rows × 53 columns

In [167]:
```
from datetime import datetime
```

In [168]:
```
df['Closed Date']=pd.to_datetime(df['Closed Date'],errors='corece')
```

In [169]:
```
df['Created Date']=pd.to_datetime(df['Created Date'],errors='corece')
```

In [170]:
```
df['Elapsed Time'] = df['Closed Date'] - df['Created Date']
```

In [171]:
```
df['Elapsed Time in seconds']= df['Elapsed Time'].map(lambda x: x.total_secon
```

In [172]:
```
df.describe()
```

Out[172]:

	Unique Key	Created Date	Closed Date	Incident Zip	X Coordinate (State Plane)	Y C (St
count	3.621770e+05	362177	362177	361502.000000	3.604700e+05	3604
mean	3.106545e+07	2015-07-13 19:49:41.535144192	2015-07-14 00:01:34.834776576	10858.533377	1.005044e+06	2034
min	2.960737e+07	2015-01-01 00:00:50	2015-01-01 00:20:33	83.000000	9.133570e+05	1211
25%	3.049763e+07	2015-04-28 08:29:08	2015-04-28 12:20:38	10314.000000	9.919460e+05	1829
50%	3.108661e+07	2015-07-15 22:07:35	2015-07-16 01:22:54	11209.000000	1.003470e+06	2010
75%	3.167497e+07	2015-10-04 00:12:31	2015-10-04 03:22:45	11238.000000	1.019135e+06	2227
max	3.231065e+07	2015-12-31 23:59:45	2016-01-03 16:22:52	11697.000000	1.067186e+06	2718
std	7.337572e+05	NaN	NaN	578.254027	2.196323e+04	298

In [173]:
```
df['Complaint Type'].isnull().sum()
```

Out[173]:
```
0
```

In [174]:
```
df['City'].isnull().sum()
```

Out[174]:
```
674
```

In [175]:
```
df['City'].fillna(value='Unkonwn City', inplace=True)
```

In [176]:
```
df['City'].isnull().sum()
```

Out[176]:
```
0
```

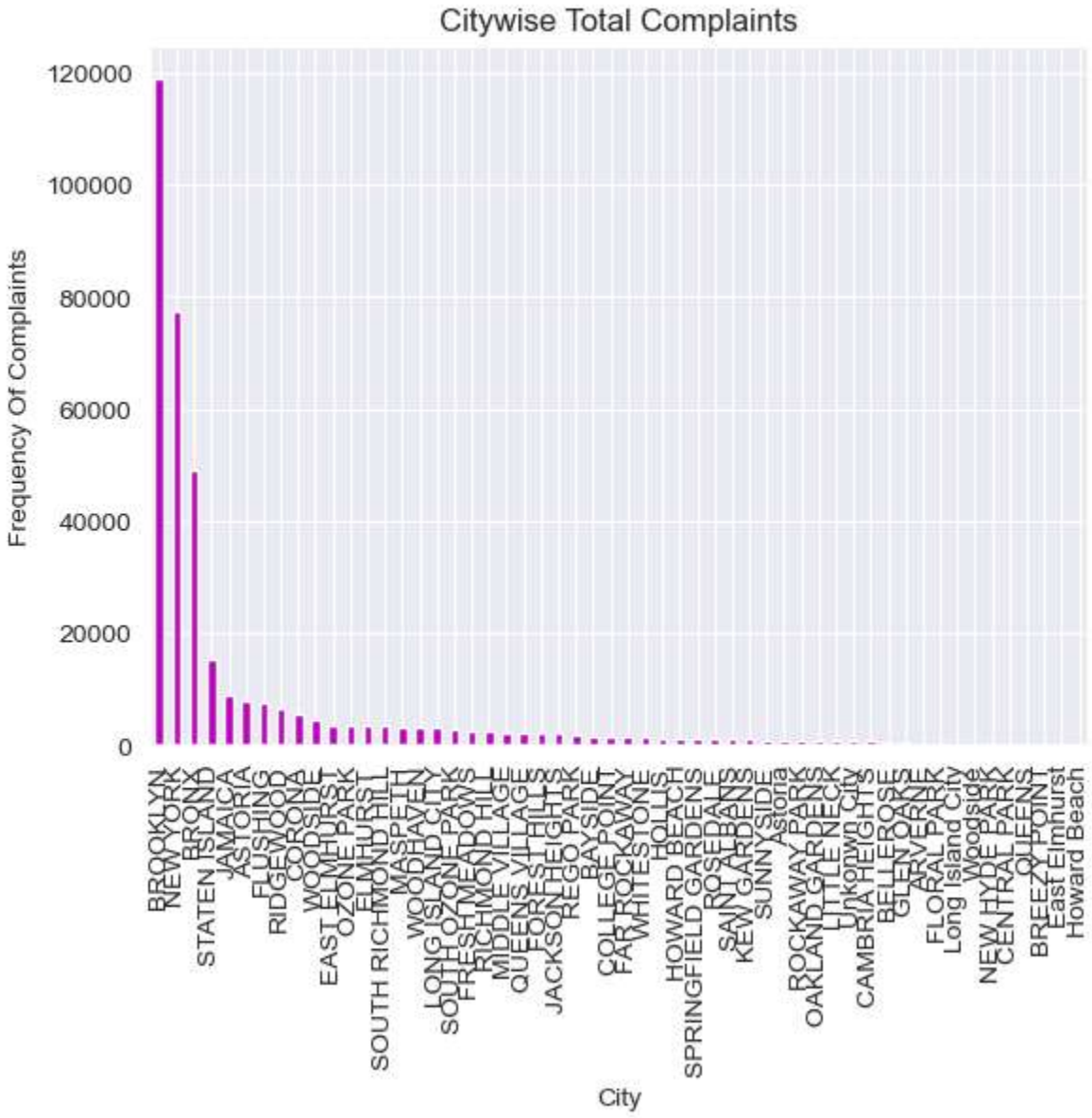
In [177]:
```
# CityWise Complaints Frequency Plot
```

In [178]:
```
plt.figure(figsize=(30,15))
```

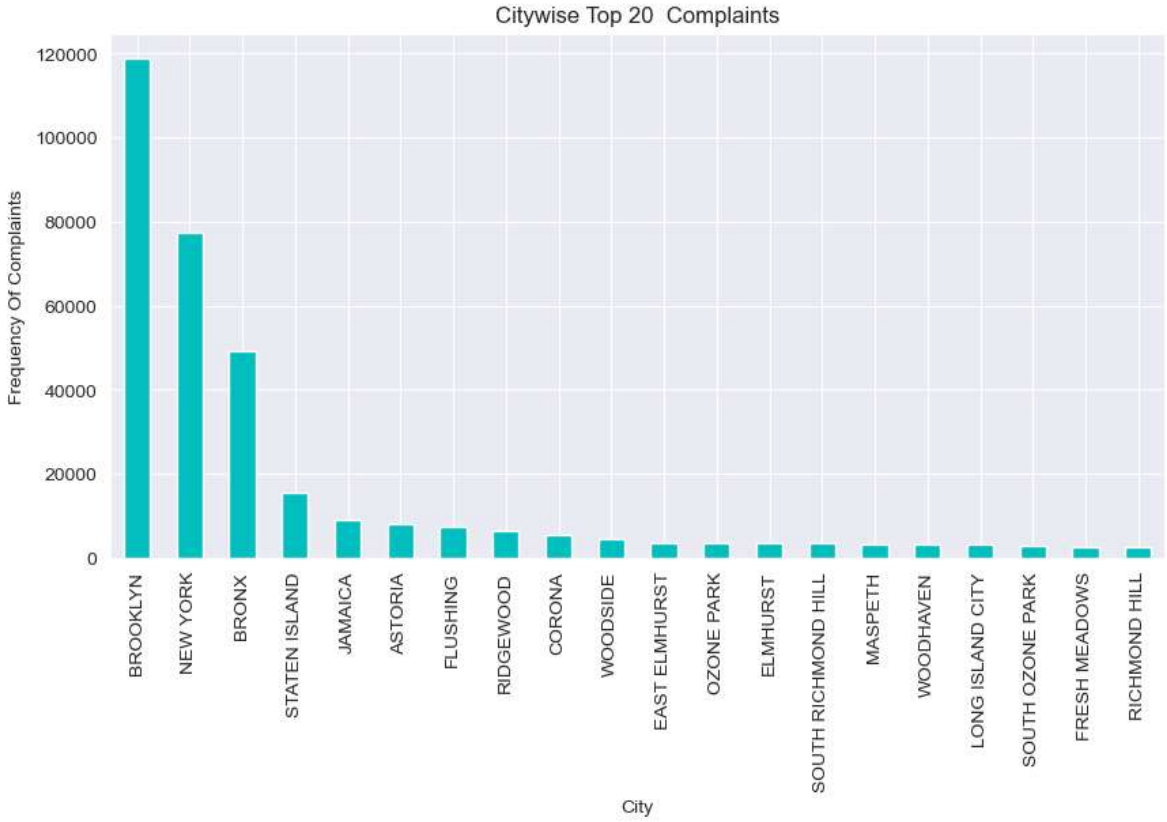
Out[178]:
```
<Figure size 3000x1500 with 0 Axes>
```

```
<Figure size 3000x1500 with 0 Axes>
```

In [179]:
```
df['City'].value_counts().plot(kind='bar', title='Citywise Total Complaints'
plt.show()
```



In [180]:
```
plt.figure(figsize=(10,5))
df['City'].value_counts().head(20).plot(kind='bar', title='Citywise Top 20  C
plt.show()
```



In [181]:
```
df1 = df[df.City=='BROOKLYN']
df1.head(2)
```

Out[181]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Inci
5	32306554	2015-12-31 23:56:30	2016-01-01 01:50:11	NYPD	New York City Police Department	Illegal Parking	Posted Parking Sign Violation	Street/Sidewalk	112
9	32308391	2015-12-31 23:53:58	2016-01-01 01:17:40	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	112

2 rows × 55 columns

In [182]:
```
df1.shape
```

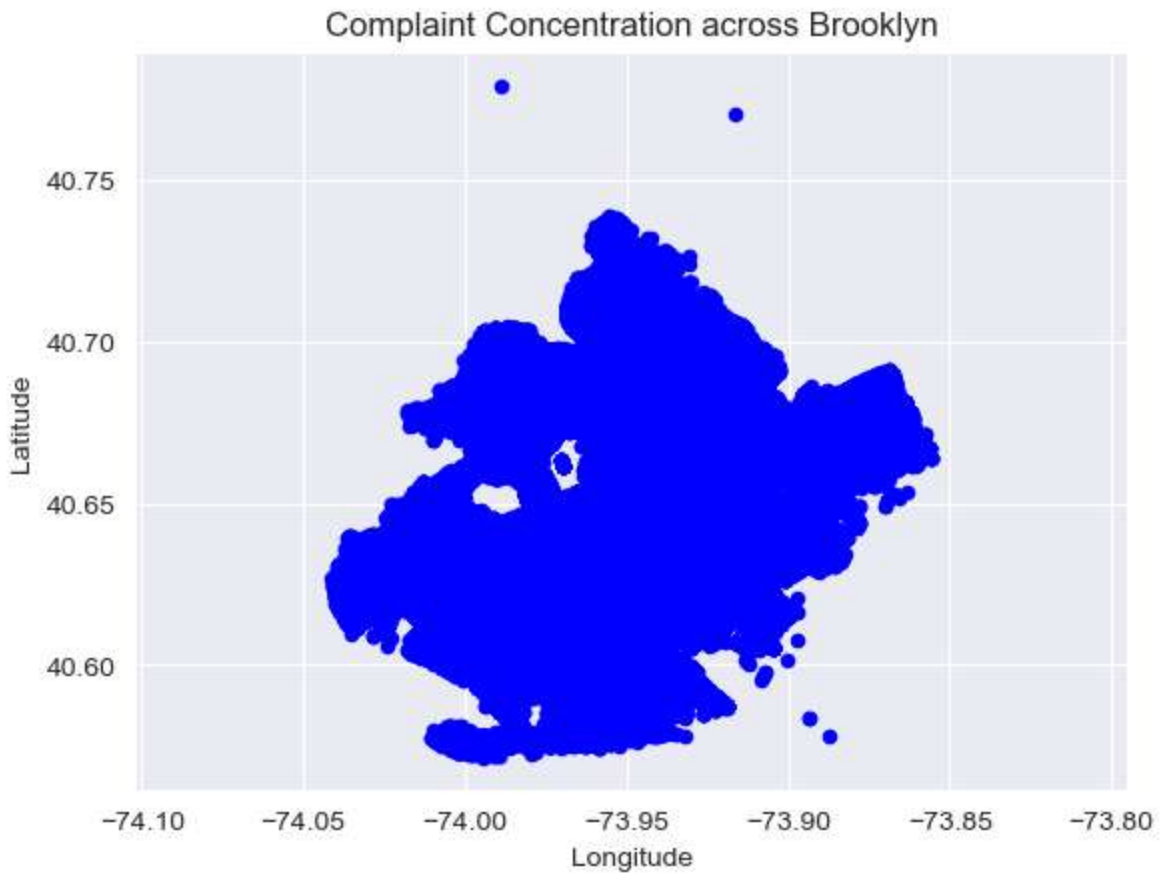
Out[182]: (118849, 55)

In [183]:
```
df1['Complaint Type'].value_counts()
```

Out[183]:
```
Complaint Type
Blocked Driveway            36445
Illegal Parking             33532
Noise - Street/Sidewalk     13982
Noise - Commercial          13855
Derelict Vehicle             6257
Noise - Vehicle              5965
Animal Abuse                 3191
Noise - Park                 1575
Traffic                      1258
Homeless Encampment           948
Vending                       575
Noise - House of Worship      389
Drinking                      291
Urinating in Public           155
Bike/Roller/Skate Chronic     124
Disorderly Youth               79
Illegal Fireworks              61
Graffiti                       60
Posting Advertisement          58
Panhandling                    49
Name: count, dtype: int64
```

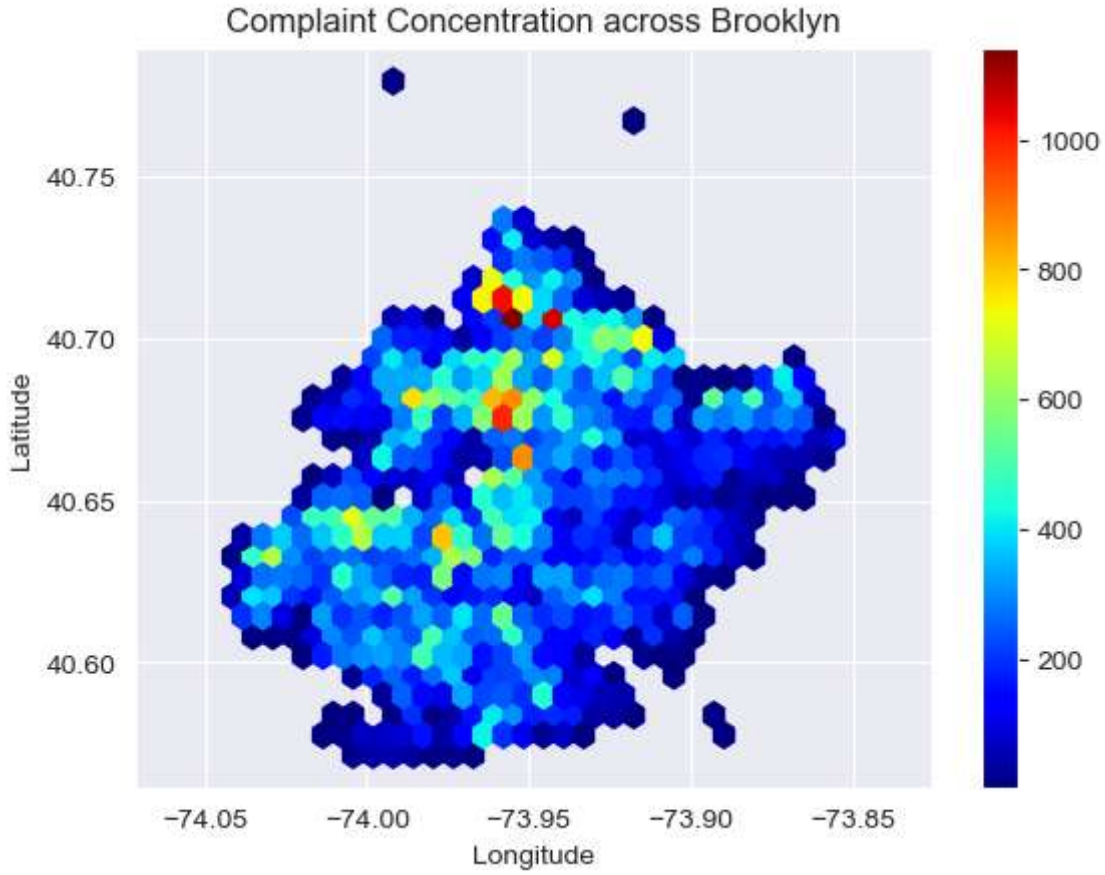
In [184]:
```
df1.plot(kind='scatter',color='b',x='Longitude',y='Latitude',title='Complaint
```

Out[184]:
```
(-74.05061403028367, -73.84647934348564, 40.561126853754885, 40.789798386232
55)
```



In [185]:
```
df1.plot(kind='hexbin',x='Longitude',y='Latitude',colormap='jet', mincnt=1, g
```

Out[185]:
```
(-74.05061403048781, -73.8464793432815, 40.561126853754885, 40.7897983862325
5)
```

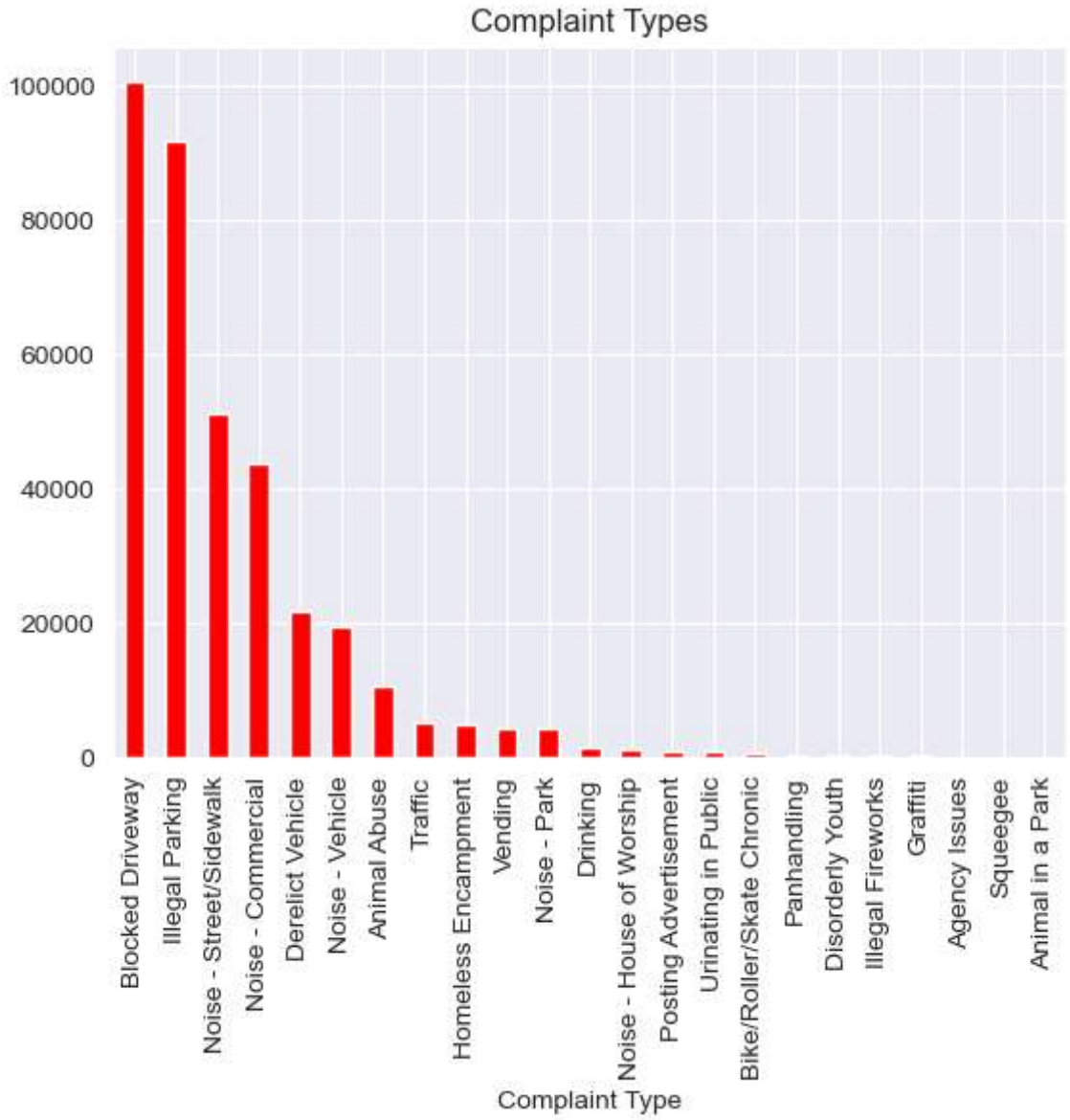


In [186]:
```
df['Complaint Type'].unique()
```

Out[186]:
```
array(['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',
       'Agency Issues', 'Squeegee', 'Animal in a Park'], dtype=object)
```

In [187]:
```
df['Complaint Type'].value_counts().plot(kind='bar' , color='r', title='Compla
```

Out[187]:
```
<Axes: title={'center': 'Complaint Types'}, xlabel='Complaint Type'>
```

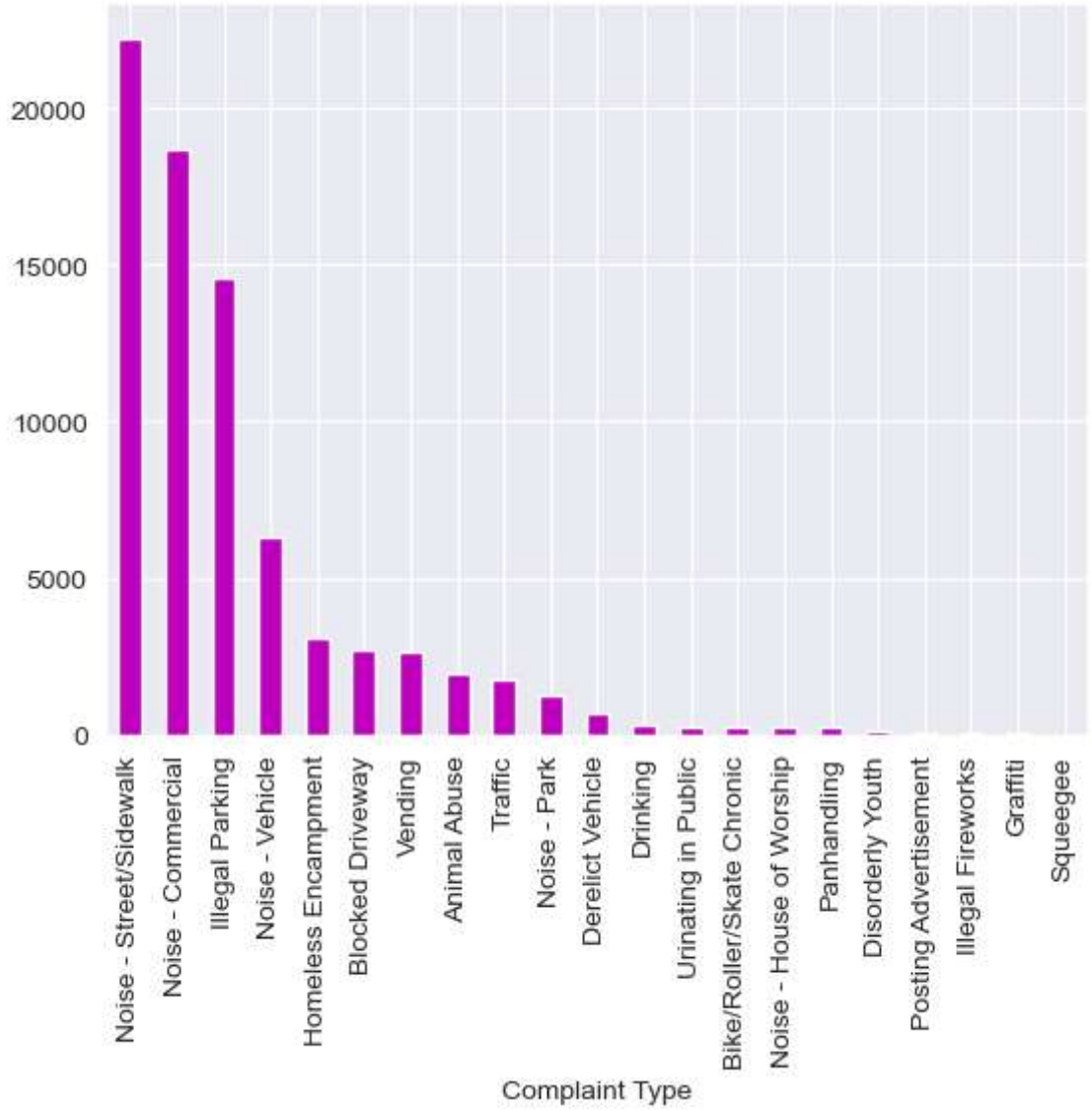


In [188]:
```
df2= df[df.City=='NEW YORK']
df2['Complaint Type'].value_counts()
```

Out[188]:
```
Complaint Type
Noise - Street/Sidewalk      22245
Noise - Commercial           18686
Illegal Parking              14549
Noise - Vehicle               6294
Homeless Encampment           3060
Blocked Driveway              2705
Vending                       2638
Animal Abuse                  1941
Traffic                       1769
Noise - Park                  1243
Derelict Vehicle               695
Drinking                       321
Urinating in Public            264
Bike/Roller/Skate Chronic      254
Noise - House of Worship       222
Panhandling                    206
Disorderly Youth                81
Posting Advertisement           49
Illegal Fireworks               38
Graffiti                        25
Squeegee                         4
Name: count, dtype: int64
```

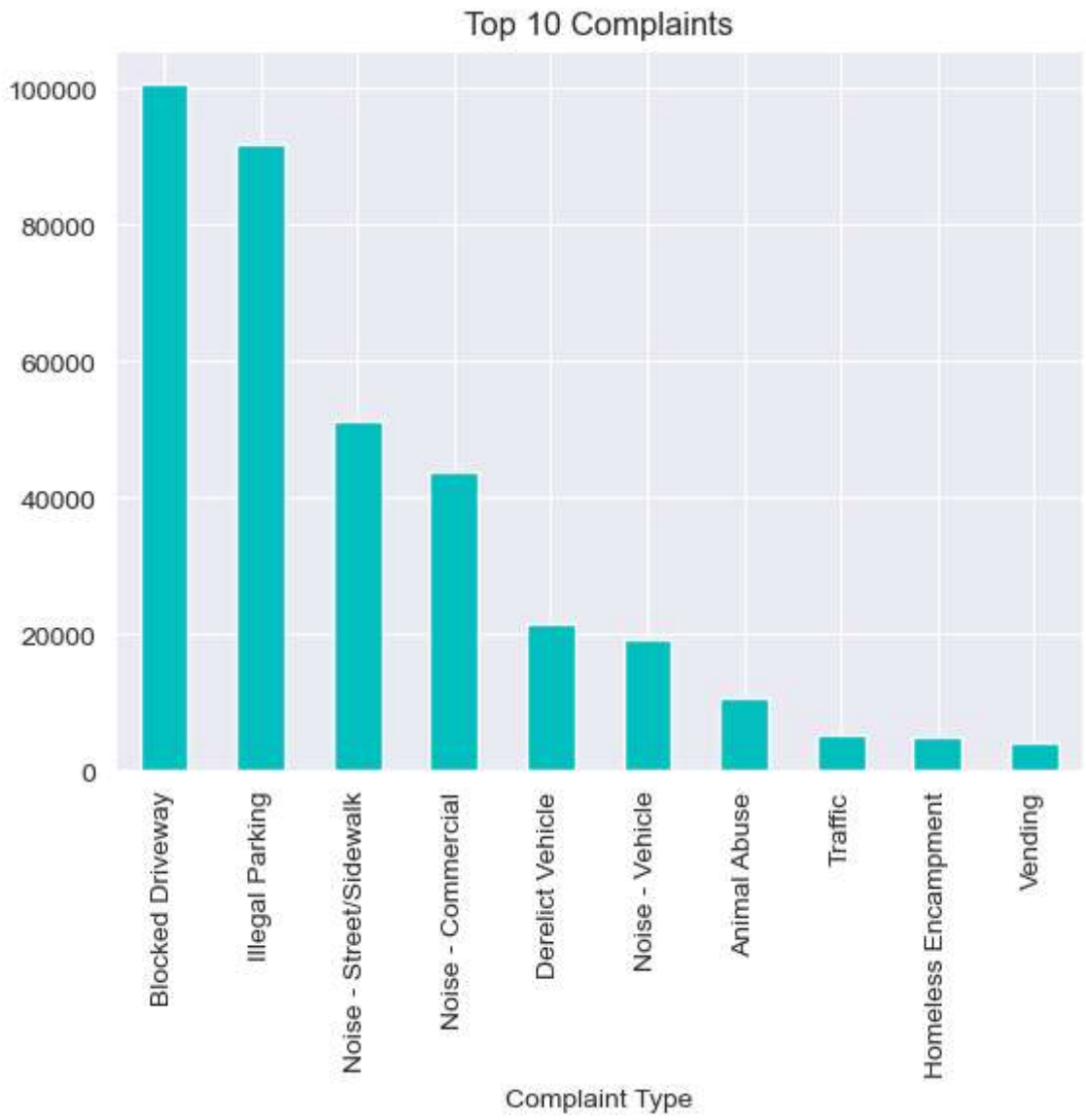
In [189]:
```
df2['Complaint Type'].value_counts().plot(kind='bar', color='#bf00bf')
```

Out[189]: <Axes: xlabel='Complaint Type'>



In [190]:
```
df['Complaint Type'].value_counts().head(10).plot(kind='bar',color='#00bfbf',
```

Out[190]: <Axes: title={'center': 'Top 10 Complaints'}, xlabel='Complaint Type'>



In [191]:
```
df[['City','Complaint Type']].value_counts()
```

Out[191]:
```
City                 Complaint Type
BROOKLYN             Blocked Driveway          36445
                     Illegal Parking           33532
NEW YORK             Noise - Street/Sidewalk   22245
                     Noise - Commercial        18686
BRONX                Blocked Driveway          17062
                                               ...
SOUTH RICHMOND HILL  Urinating in Public           1
FRESH MEADOWS        Urinating in Public           1
FOREST HILLS         Drinking                      1
BELLEROSE            Urinating in Public           1
BREEZY POINT         Noise - Vehicle               1
Name: count, Length: 792, dtype: int64
```

In [192]:
```
new_df=df.groupby(['Complaint Type','City']).size().unstack()
```

In [193]:
```
new_df.head()
```

Out[193]:

City	ARVERNE	ASTORIA	Astoria	BAYSIDE	BELLEROSE	BREEZY POINT	BRONX	BRC
Complaint Type								
Agency Issues	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Animal Abuse	46.0	170.0	NaN	53.0	15.0	2.0	1971.0	
Animal in a Park	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Bike/Roller/Skate Chronic	NaN	16.0	NaN	NaN	1.0	NaN	22.0	
Blocked Driveway	50.0	3436.0	159.0	514.0	138.0	3.0	17062.0	

5 rows × 54 columns

In [194]:
```
temp = df['City'].value_counts()
temp[ :10]
```

Out[194]:
```
City
BROOKLYN         118849
NEW YORK          77289
BRONX             49166
STATEN ISLAND     15335
JAMAICA            8930
ASTORIA            7991
FLUSHING           7486
RIDGEWOOD          6391
CORONA             5383
WOODSIDE           4357
Name: count, dtype: int64
```

In [195]:
```
temp[ :10].keys()
```

Out[195]:
```
Index(['BROOKLYN', 'NEW YORK', 'BRONX', 'STATEN ISLAND', 'JAMAICA', 'ASTORI
A',
       'FLUSHING', 'RIDGEWOOD', 'CORONA', 'WOODSIDE'],
      dtype='object', name='City')
```

In [196]:
```
top_complaints=new_df[['BROOKLYN', 'NEW YORK', 'BRONX', 'STATEN ISLAND', 'JAM
       'FLUSHING', 'RIDGEWOOD', 'CORONA', 'WOODSIDE']]
```

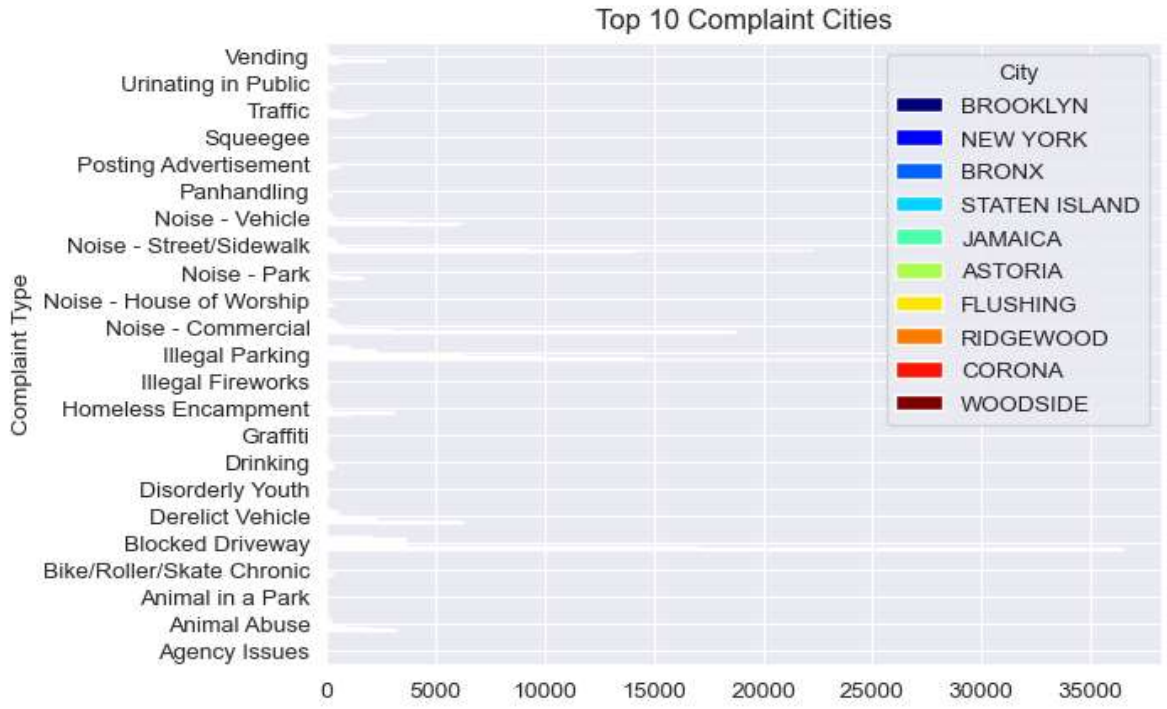
In [197]:
```
top_complaints
```

Out[197]:

City	BROOKLYN	NEW YORK	BRONX	STATEN ISLAND	JAMAICA	ASTORIA	FLUSHING	RIDC
Complaint Type								
Agency Issues	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Animal Abuse	3191.0	1941.0	1971.0	786.0	317.0	170.0	191.0	
Animal in a Park	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Bike/Roller/Skate Chronic	124.0	254.0	22.0	10.0	3.0	16.0	3.0	
Blocked Driveway	36445.0	2705.0	17062.0	2845.0	3620.0	3436.0	3640.0	
Derelict Vehicle	6257.0	695.0	2402.0	2184.0	1132.0	426.0	532.0	
Disorderly Youth	79.0	81.0	66.0	25.0	9.0	5.0	2.0	
Drinking	291.0	321.0	206.0	188.0	40.0	43.0	47.0	
Graffiti	60.0	25.0	15.0	6.0	3.0	4.0	6.0	
Homeless Encampment	948.0	3060.0	275.0	77.0	93.0	32.0	26.0	
Illegal Fireworks	61.0	38.0	24.0	11.0	4.0	4.0	2.0	
Illegal Parking	33532.0	14549.0	9889.0	6224.0	1698.0	1340.0	2250.0	
Noise - Commercial	13855.0	18686.0	2944.0	783.0	552.0	1653.0	222.0	
Noise - House of Worship	389.0	222.0	90.0	18.0	15.0	21.0	5.0	
Noise - Park	1575.0	1243.0	548.0	67.0	38.0	64.0	61.0	
Noise - Street/Sidewalk	13982.0	22245.0	9144.0	885.0	365.0	409.0	241.0	
Noise - Vehicle	5965.0	6294.0	3556.0	424.0	337.0	236.0	147.0	
Panhandling	49.0	206.0	20.0	13.0	3.0	2.0	2.0	
Posting Advertisement	58.0	49.0	18.0	516.0	8.0	3.0	1.0	
Squeegee	NaN	4.0	NaN	NaN	NaN	NaN	NaN	
Traffic	1258.0	1769.0	427.0	229.0	632.0	60.0	59.0	
Urinating in Public	155.0	264.0	54.0	19.0	37.0	10.0	12.0	
Vending	575.0	2638.0	433.0	25.0	24.0	57.0	37.0	

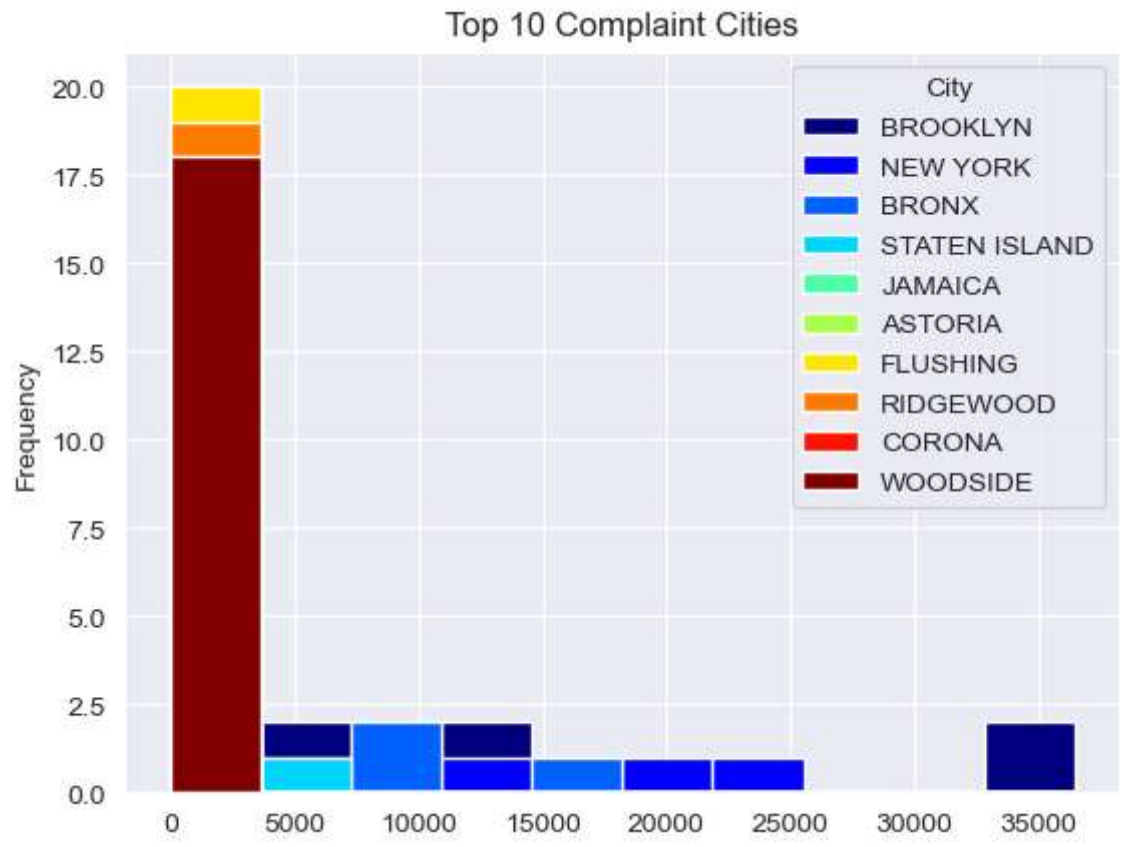
In [198]:
```
top_complaints.plot(kind='barh', colormap='jet', title='Top 10 Complaint Citie
```

Out[198]: <Axes: title={'center': 'Top 10 Complaint Cities'}, ylabel='Complaint Type'>



In [199]:
```
top_complaints.plot(kind='hist', colormap='jet', title='Top 10 Complaint Citie
```

Out[199]: <Axes: title={'center': 'Top 10 Complaint Cities'}, ylabel='Frequency'>



In [200]:
```
cities=df['City'].value_counts().index.to_list()
```

In [201]:
```
ds=df[df.City.isin(cities)]
```

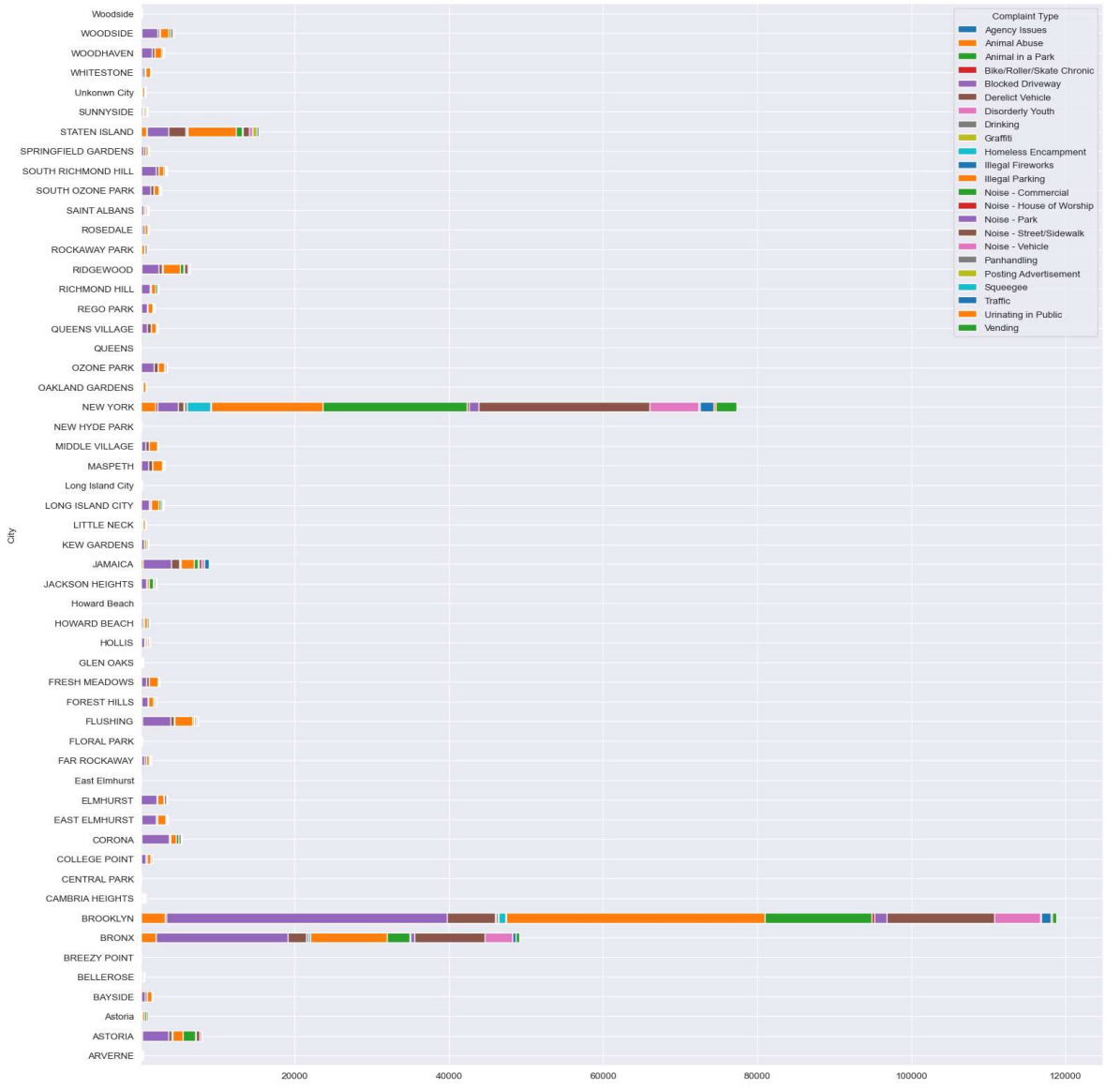
In [202]:
```
df1= pd.crosstab(ds['City'],ds['Complaint Type'])
```

In [203]:
```
df1
```

Out[203]:

City	Agency Issues	Animal Abuse	Animal in a Park	Bike/Roller/Skate Chronic	Blocked Driveway	Derelict Vehicle	Disorderly Youth	Di
ARVERNE	0	46	0	0	50	32	2	
ASTORIA	0	170	0	16	3436	426	5	
Astoria	0	0	0	0	159	14	0	
BAYSIDE	0	53	0	0	514	231	2	
BELLEROSE	0	15	0	1	138	120	2	
BREEZY POINT	0	2	0	0	3	3	0	
BRONX	0	1971	0	22	17062	2402	66	

In [204]:
```
df1.plot(kind='barh',stacked=True,figsize=(18,20))
plt.title=('Category Wise Complaints Per City')
plt.show()
```



In [205]:
```
df['Request_Closing_Time']=df['Closed Date']-df['Created Date']
df['Request_Closing_Hours']=df['Request_Closing_Time'].astype('timedelta64[ms]

df[['Request_Closing_Time','Request_Closing_Hours']].head(10)
```

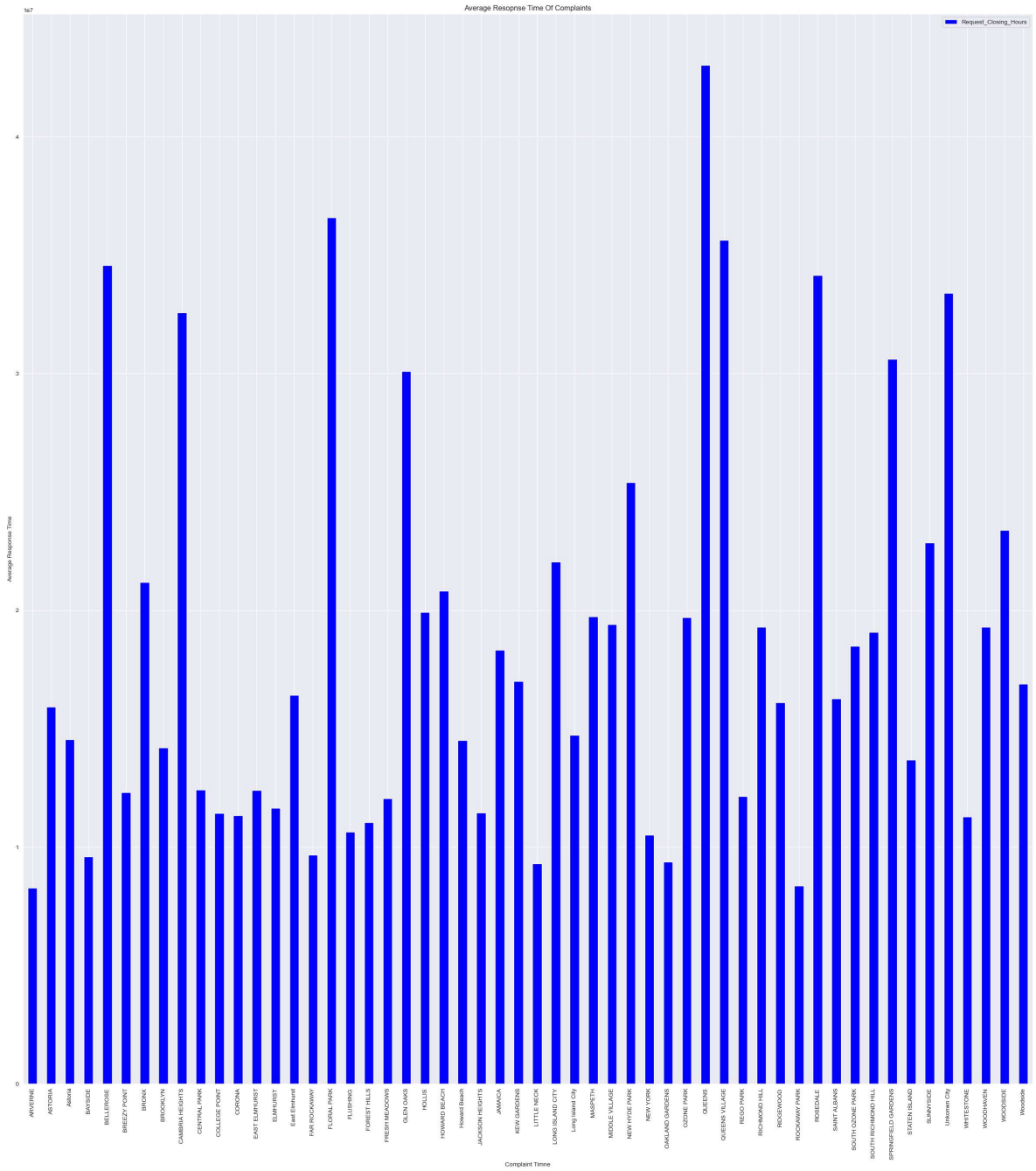
Out[205]:

	Request_Closing_Time	Request_Closing_Hours
0	0 days 00:55:30	0 days 00:55:30
1	0 days 01:27:13	0 days 01:27:13
2	0 days 04:51:34	0 days 04:51:34
3	0 days 07:45:27	0 days 07:45:27
4	0 days 03:27:44	0 days 03:27:44
5	0 days 01:53:41	0 days 01:53:41
6	0 days 01:58:22	0 days 01:58:22
7	0 days 01:48:49	0 days 01:48:49
8	0 days 08:33:34	0 days 08:33:34
9	0 days 01:23:42	0 days 01:23:42

In [206]:
```
Resolution = df.groupby(['City'])[['Request_Closing_Hours']].mean()

Resolution.plot(kind='bar',title='Average Resopnse Time Of Complaints', xlabel
```

Out[206]:
```
<Axes: title={'center': 'Average Resopnse Time Of Complaints'}, xlabel='Comp
laint Timne', ylabel='Average Response Time'>
```



In [207]:
```
nex_df= df.groupby(['City', 'Complaint Type']) ['Elapsed Time in seconds'].mea
nex_df.sort_values(by=['City', 'Elapsed Time in seconds'])
```

Out[207]:

	City	Complaint Type	Elapsed Time in seconds
4	ARVERNE	Drinking	859.000000
16	ARVERNE	Vending	1735.000000
15	ARVERNE	Urinating in Public	2491.000000
13	ARVERNE	Panhandling	3673.000000
14	ARVERNE	Traffic	4014.000000
...	...	...	...
790	Woodside	Noise - Commercial	8619.000000
791	Woodside	Noise - Street/Sidewalk	12285.600000
787	Woodside	Blocked Driveway	15566.185185
789	Woodside	Illegal Parking	17293.459677
788	Woodside	Derelict Vehicle	19994.500000

792 rows × 3 columns

In [208]:
```
complaintstypes= df['Complaint Type'].unique()

for i in range(len(complaintstypes)):
    exec("c{} =df.loc[(df['Complaint Type']=='{}'),'Elapsed Time in seconds']"
```

In [209]:
```
fscore,pvalue = stats.f_oneway(c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,
print(fscore,pvalue)
```

```
603.320672207129 0.0
```

In [210]:
```
from scipy import stats
```

In [211]:
```
df7= df['Latitude'].mean()
df8= df['Longitude'].mean()
df9= df['Elapsed Time in seconds'].mean()

result = stats.kruskal(df7,df8,df9)
print(result)
```

```
KruskalResult(statistic=2.0, pvalue=0.36787944117144245)
```

In [ ]:
```

```