	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction	I
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/Sidewalk	10034.0	71 VERMILYEA AVENUE	...	NaN	NaN	
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/Sidewalk	11105.0	27-07 23 AVENUE	...	NaN	NaN	
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/Sidewalk	10458.0	2897 VALENTINE AVENUE	...	NaN	NaN	
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/Sidewalk	10461.0	2940 BAISLEY AVENUE	...	NaN	NaN	
4	32306529	12/31/2015 11:56:58 PM	01/01/2016 03:24:42 AM	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	...	NaN	NaN	

5 rows × 53 columns

```
pd.set_option('display.max_columns',100)
pd.set_option('display.width',1000)
pd.set_option('display.float_format','{:2f}'.format)
import matplotlib as plt
%matplotlib inline
%config InlineBackend.figure_format='retina'
```

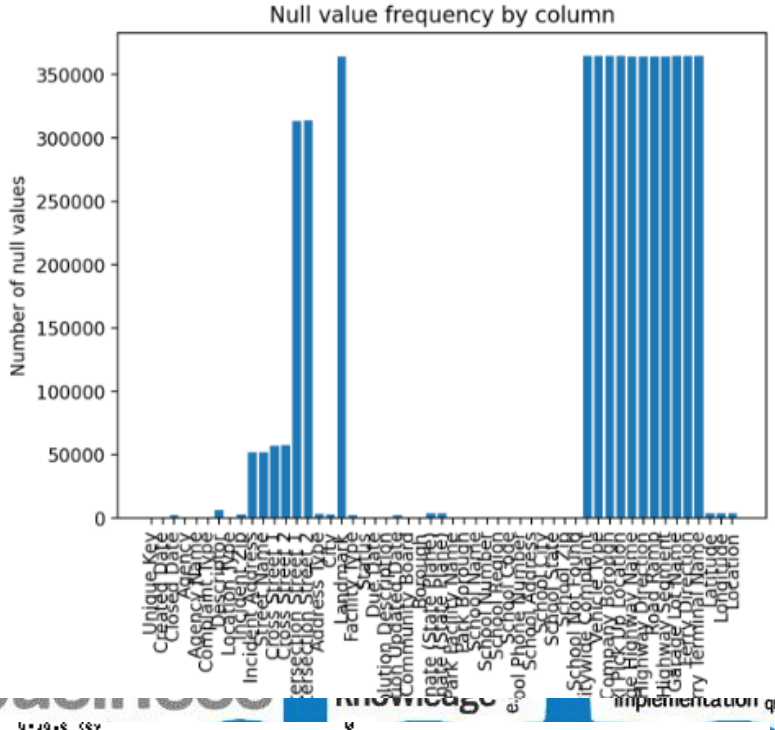
```
print(df.columns.tolist())
```

```
['Unique Key', 'Created Date', 'Closed Date', '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', 'Ferry Terminal Name', 'Latitude', 'Longitude', 'Location']
```

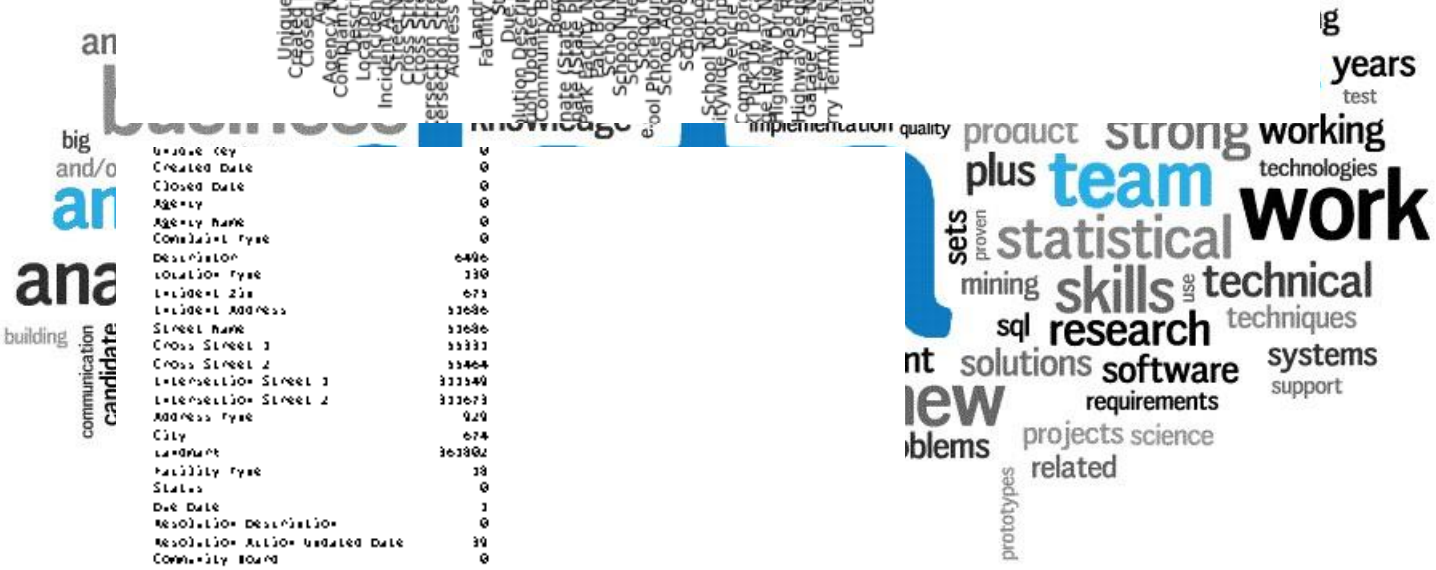
```
print(df.shape)
```

```
(364558, 53)
```

```
plt.bar(null_counts.index, null_counts.values)
plt.xticks(rotation=90)
plt.xlabel('Columns')
plt.ylabel('Number of null values')
plt.title('Null value frequency by column')
plt.show()
```




```
Unique Key                                0
Created Date                              0
Closed Date                               0
Agency                                    0
Agency Name                               0
Complaint Type                            0
Descriptor                             6496
Location Type                           130
Incident Zip                            675
Incident Address                      33686
Street Name                           33686
Cross Street 1                        33333
Cross Street 2                        33464
Intersection Street 1                 333349
Intersection Street 2                 333673
Address Type                             929
City                                     674
Landmark                              363802
Facility Type                             38
Status                                     0
Due Date                                   1
Resolution Description                     0
Resolution Action Updated Date            39
Community Board                            0
Borough                                    0
X Coordinate (State Plane)              3707
Y Coordinate (State Plane)              3707
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          362377
Vehicle Type                          362377
Taxi Company Borough                  362377
Taxi Pick Up Location                 362377
Bridge Highway Name                   363880
Bridge Highway Direction              363880
Road Ramp                             363933
Bridge Highway Segment                363933
Garage Lot Name                       362377
Ferry Direction                       362377
Ferry Terminal Name                   362377
Latitude                                3707
Longitude                               3707
Location                                3707
dtype: int64
```

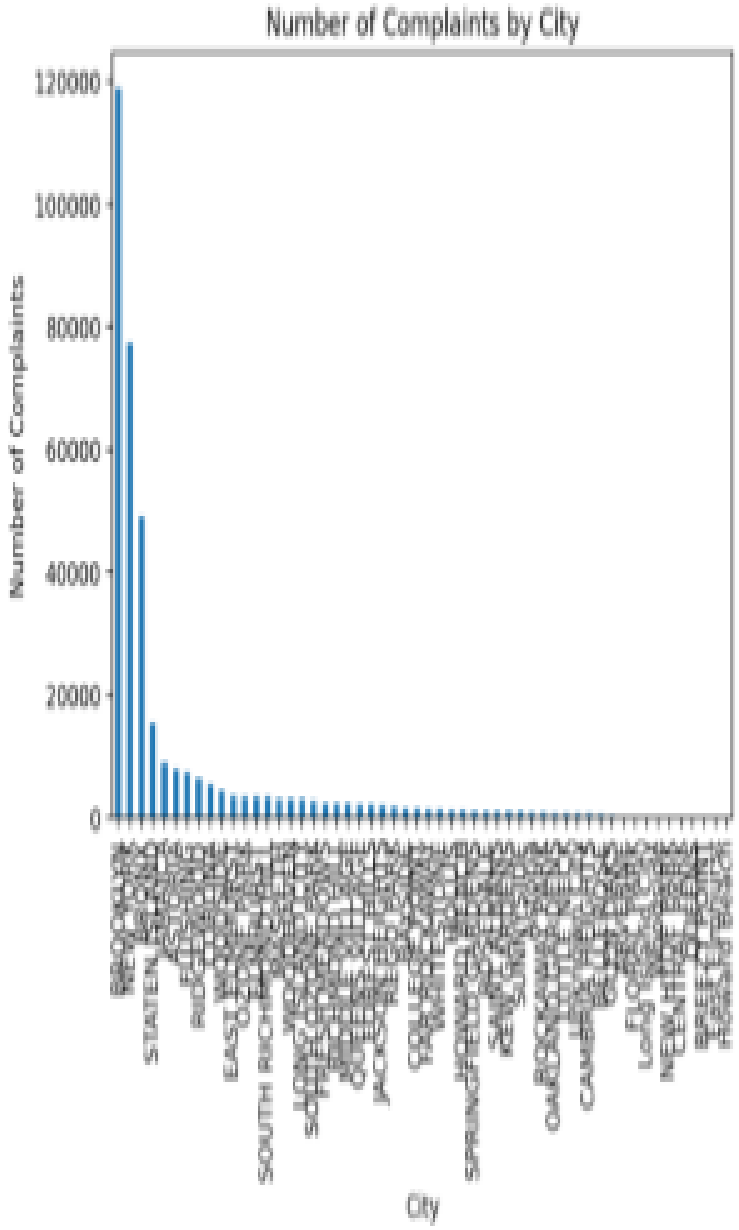


```
from datetime import date,time,datetime
df['Created Date']=df['Created Date'].astype('datetime64[ns]')
df['Closed Date']=df['Closed Date'].astype('datetime64[ns]')
df['time_diff']=df['Closed Date']-df['Created Date']
df['Request_Closing_Time']=(df["time_diff"].astype('timedelta64[s]'))/(60*60)
```

```
df.head()
```

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Street Name	Cross Street 1	Cross Street 2	Intersection St
0	33310353	2015-12-31 23:59:45	2016-01-01 00:55:16	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034	71 VERMILYEA AVENUE	VERMILYEA AVENUE	ACADEMY STREET	WEST 204 STREET	
1	33309534	2015-12-31 23:59:44	2016-01-01 01:26:57	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105	27-07 23 AVENUE	23 AVENUE	27 STREET	28 STREET	
2	33309159	2015-12-31 23:59:29	2016-01-01 04:51:03	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458	2897 VALENTINE AVENUE	VALENTINE AVENUE	EAST 198 STREET	EAST 199 STREET	
3	33305638	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	10461	2940 BAISLEY AVENUE	BAISLEY AVENUE	EDISON AVENUE	B STREET	
4	33305629	2015-12-31 23:56:58	2016-01-01 03:24:42	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373	87-14 57 ROAD	57 ROAD	SEABURY STREET	HOFFMAN DRIVE	

[28]:
```
# Count the number of complaints for each city
complaints_by_city = df['City'].value_counts()

# Create a bar plot of the number of complaints for each city
complaints_by_city.plot(kind='bar')

# Set the title and axis labels
plt.title('Number of Complaints by City')
plt.xlabel('City')
plt.ylabel('Number of Complaints')

# Display the plot
plt.show()
```



In [3]:
```
df.info()
```

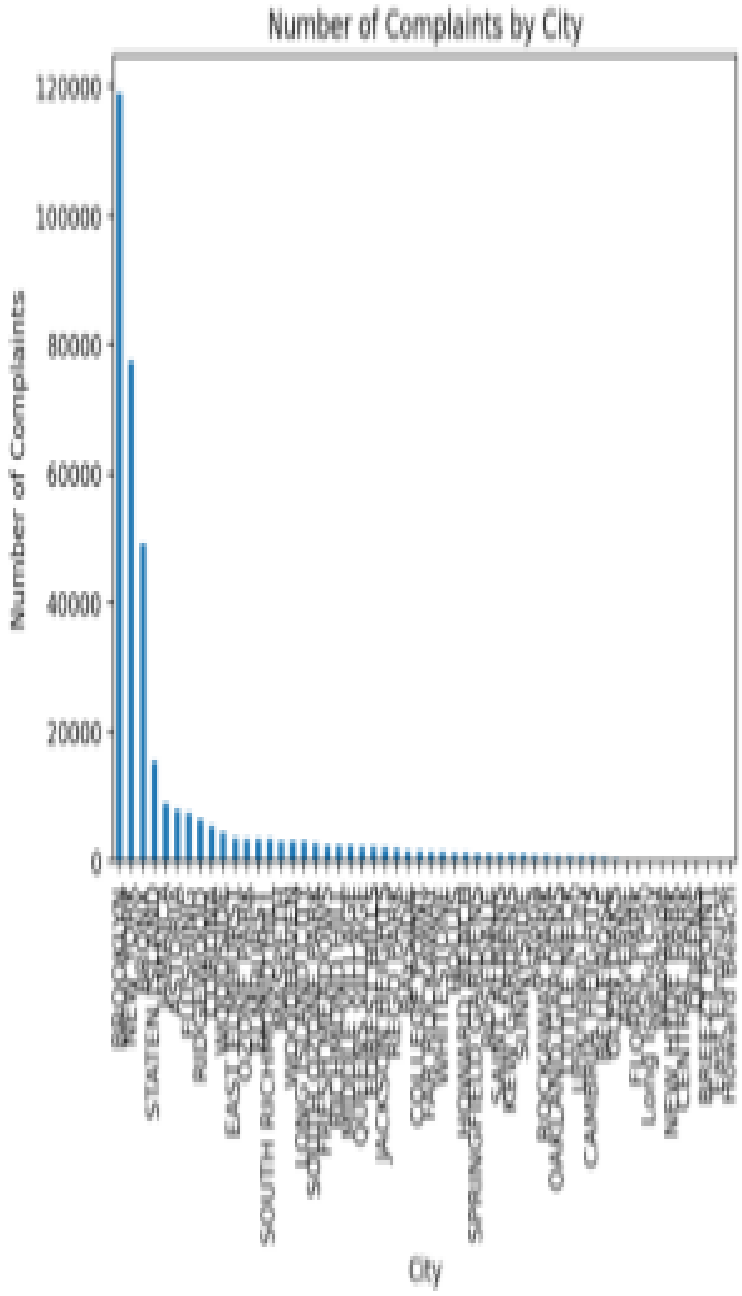
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 364558 entries, 0 to 364557
Data columns (total 53 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   Unique Key                      364558 non-null  int64
 1   Created Date                    364558 non-null  object
 2   Closed Date                     362377 non-null  object
 3   Agency                          364558 non-null  object
 4   Agency Name                     364558 non-null  object
 5   Complaint Type                  364558 non-null  object
 6   Descriptor                      358037 non-null  object
 7   Location Type                   364421 non-null  object
 8   Incident Zip                    361960 non-null  float64
 9   Incident Address                312859 non-null  object
 10  Street Name                     312859 non-null  object
 11  Cross Street 1                  307370 non-null  object
 12  Cross Street 2                  306733 non-null  object
 13  Intersection Street 1           51120 non-null   object
 14  Intersection Street 2           50112 non-null   object
 15  Address Type                    361306 non-null  object
 16  City                            361961 non-null  object
 17  Landmark                        371 non-null     object
 18  Facility Type                   362360 non-null  object
 19  Status                          364558 non-null  object
 20  Due Date                        364555 non-null  object
 21  Resolution Description          364558 non-null  object
 22  Resolution Action Updated Date  362356 non-null  object
 23  Community Board                 364558 non-null  object
 24  Borough                         364558 non-null  object
 25  X Coordinate (State Plane)      360529 non-null  float64
 26  Y Coordinate (State Plane)      360529 non-null  float64
 27  Park Facility Name              364558 non-null  object
 28  Park Borough                    364558 non-null  object
 29  School Name                     364558 non-null  object
 30  School Number                   364558 non-null  object
 31  School Region                   364557 non-null  object
 32  School Code                     364557 non-null  object
 33  School Phone Number             364558 non-null  object
 34  School Address                  364558 non-null  object
 35  School City                     364558 non-null  object
 36  School State                    364558 non-null  object
 37  School Zip                      364557 non-null  object
 38  School Not Found                364558 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             297 non-null     object
 44  Bridge Highway Direction        297 non-null     object
 45  Road Ramp                       262 non-null     object
 46  Bridge Highway Segment          262 non-null     object
 47  Garage Lot Name                 0 non-null       float64
 48  Ferry Direction                 3 non-null       object
 49  Ferry Terminal Name             2 non-null       object
 50  Latitude                        360529 non-null  float64
 51  Longitude                       360529 non-null  float64
 52  Location                        360529 non-null  object
dtypes: float64(10), int64(1), object(42)
memory usage: 147.4+ MB
```

```
print('Number of null values in City column.', df['City'].isnull().sum
print(df.isnull().sum())
```

```
Number of null values in City column. 674
Unique Key                                0
Created Date                              0
Closed Date                               0
Agency                                    0
Agency Name                               0
Complaint Type                            0
Descriptor                             6406
Location Type                           330
Incident Zip                            675
Incident Address                      53686
Street Name                           53686
Cross Street 1                        55333
Cross Street 2                        55464
Intersection Street 1                333540
Intersection Street 2                333673
Address Type                            929
City                                    674
Landmark                             363802
Facility Type                            38
Status                                    0
Due Date                                  1
Resolution Description                    0
Resolution Action Updated Date           30
Community Board                           0
Borough                                   0
X Coordinate (State Plane)             3707
Y Coordinate (State Plane)             3707
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         362377
Vehicle Type                         362377
Taxi Company Borough                 362377
Taxi Pick Up Location                362377
Bridge Highway Name                  363880
Bridge Highway Direction             363880
Road Ramp                            363915
Bridge Highway Segment               363915
Garage Lot Name                      362377
Ferry Direction                      362377
Ferry Terminal Name                  362377
Latitude                               3707
Longitude                              3707
Location                               3707
time_diff                                 0
Request_Closing_Time                      0
New_Column                                0
dtype: int64
```


data
experience
business
analytics
analysis
work
team
statistical
skills
technical
research
tools
strong
working
modeling
statistics
knowledge
insights
algorithms
develop
management
solutions
software
systems
including
new
models
problems
learning
large
hadoop
python
sql
mining
sets
plus
product
implementation
quality
products
platforms
preferred
analytical
development
degree
distributed
familiarity
customer
customers
field
java
machine
must
language
programming
processing
recommendations
production
understanding
using
well
years
test
technologies
techniques
support
requirements
projects
science
related
prototypes
marketing
languages
help
improve
environment
design
greenplum
datasets
engineering
computer
deliver
company
complex
candidate
communication
building
big
and/or
communicate
database
least
proven
use
e.g

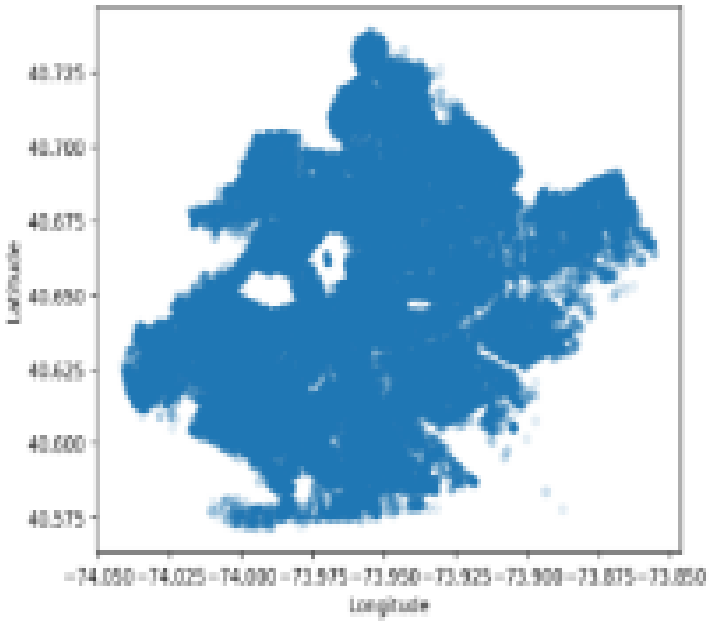
[28]:
```
# Count the number of complaints for each city
complaints_by_city = df['City'].value_counts()

# Create a bar plot of the number of complaints for each city
complaints_by_city.plot(kind='bar')

# Set the title and axis labels
plt.title('Number of Complaints by City')
plt.xlabel('City')
plt.ylabel('Number of Complaints')

# Display the plot
plt.show()
```

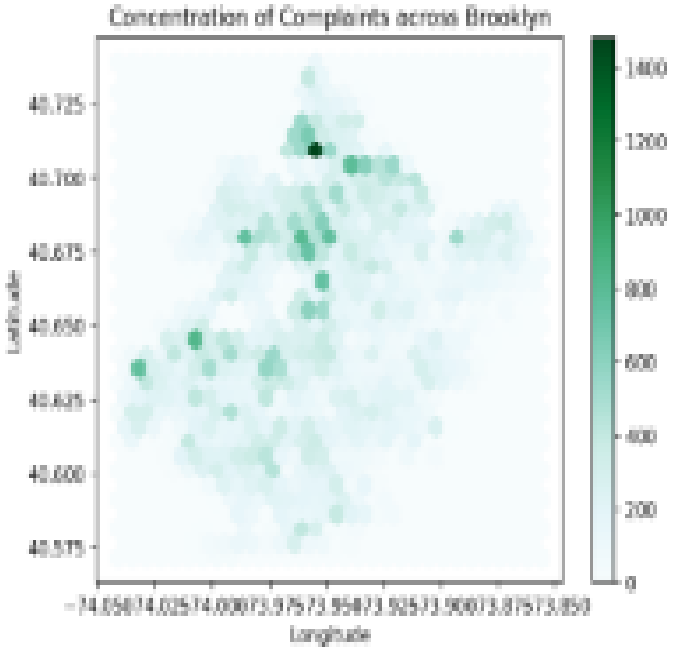



data
experience
business
analytics
analysis
work
team
statistical
skills
technical
research
tools
strong
working
modeling
statistics
knowledge
insights
algorithms
develop
management
solutions
software
systems
including
new
models
problems
learning
large
hadoop
python
sql
mining
sets
plus
product
implementation
quality
products
platforms
preferred
analytical
development
degree
distributed
familiarity
customer
customers
field
java
machine
must
language
programming
processing
recommendations
production
understanding
using
well
years
test
technologies
techniques
support
requirements
projects
science
related
prototypes
marketing
languages
help
improve
environment
design
greenplum
datasets
engineering
computer
deliver
company
complex
candidate
communication
building
big
and/or
communicate
database
least
proven
use
e.g

In [29]:

```
# Filter the DataFrame to include only complaints in Brooklyn
brooklyn_df = df[df['Borough'] == 'BROOKLYN']

# Create a scatter plot of the concentration of complaints across Brooklyn
brooklyn_df.plot(kind='scatter', x='Longitude', y='Latitude', alpha=0.1)

# Create a density plot of the concentration of complaints across Brooklyn
brooklyn_df.plot(kind='hexbin', x='Longitude', y='Latitude', gridsize=100)

# Add the title and axis labels
plt.title('Concentration of Complaints across Brooklyn')
plt.xlabel('Longitude')
plt.ylabel('Latitude')

# Display the plot
plt.show()
```





```
# Filter the dataframe to include only complaints in New York City
nyc_df = df[df['City'] == 'NEW YORK']

# Count the number of complaints for each type
complaints_by_type = nyc_df['Complaint Type'].value_counts()

# Print the counts for each type of complaint
print(complaints_by_type)
```

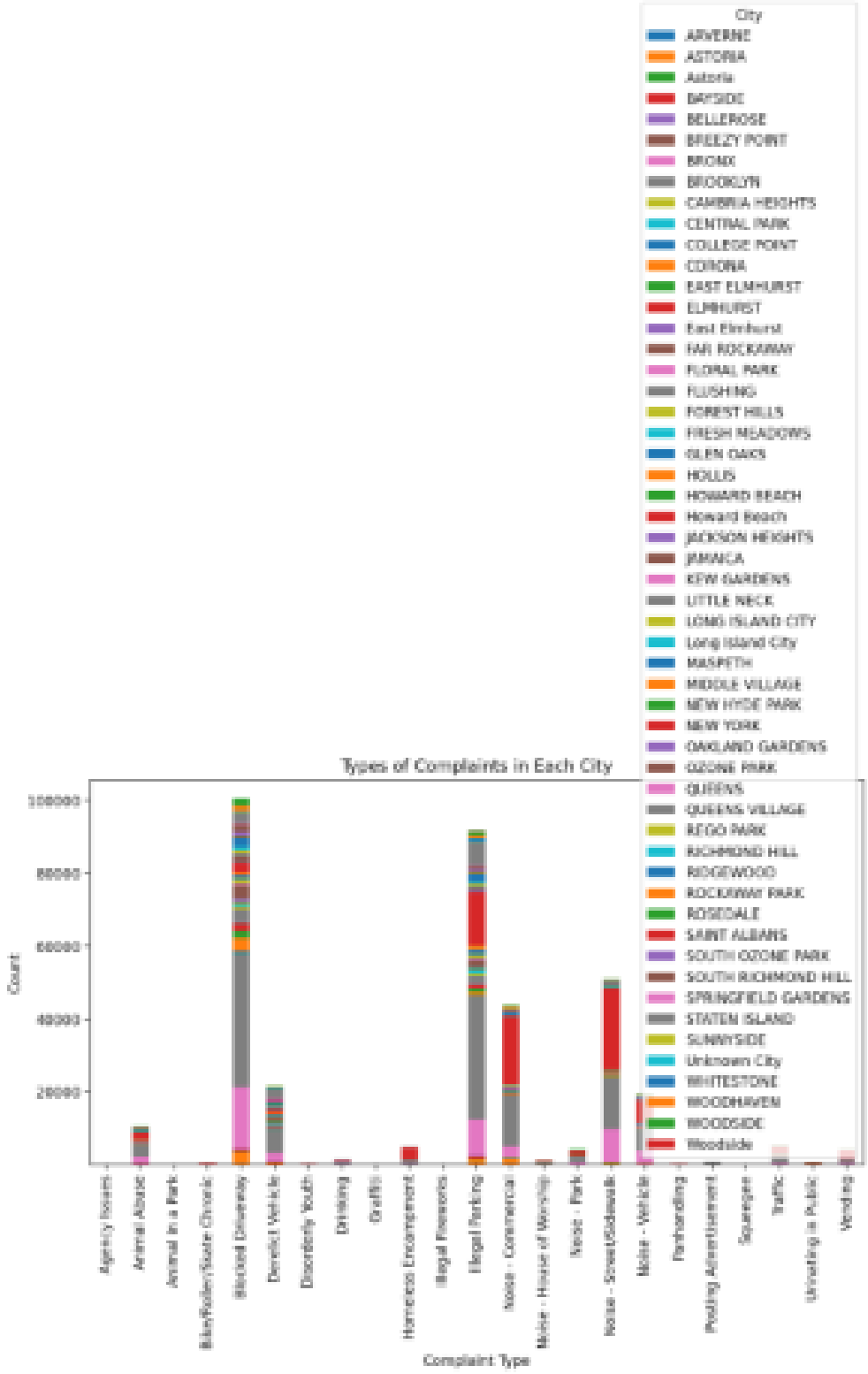
```
Noise - Street/Sidewalk       22245
Noise - Commercial            18686
Illegal Parking               14349
Noise - Vehicle                6294
Homeless Encampment            3060
Blocked Driveway               2705
Vending                        2639
Animal Abuse                   1943
Graffiti                       1769
Noise - Park                   1243
Derelict Vehicle                693
Drinking                        323
Urinating in Public             264
Bike/Roller/Skate Chronic       254
Noise - House of Worship        222
Panhandling                     206
Disorderly Youth                 93
Posting Advertisement            49
Illegal Fireworks                38
Graffiti                         25
Squeegee                          4
Name: Complaint Type, dtype: int64
```

```
# Count the number of complaints for each type
complaints_by_type = df['Complaint Type'].value_counts()

# Display the top 10 types of complaints
top_10_complaints = complaints_by_type.head(10)
print(top_10_complaints)
```

```
Blocked Driveway            100624
Illegal Parking              91736
Noise - Street/Sidewalk      51330
Noise - Commercial           43753
Derelict Vehicle             21338
Noise - Vehicle              19301
Animal Abuse                 10530
Graffiti                      5106
Homeless Encampment           4879
Vending                       4385
Name: Complaint Type, dtype: int64
```


data
experience
business
analytics
analysis
work
team
statistical
skills
technical
research
tools
strong
working
modeling
statistics
knowledge
insights
algorithms
develop
management
solutions
software
systems
including
new
models
problems
learning
large
hadoop
python
sql
mining
sets
plus
product
implementation
quality
products
platforms
preferred
analytical
development
degree
distributed
familiarity
customer
customers
field
java
machine
must
language
programming
processing
recommendations
production
understanding
using
well
years
test
technologies
techniques
support
requirements
projects
science
related
prototypes
marketing
languages
help
improve
environment
design
greenplum
datasets
engineering
computer
deliver
company
complex
candidate
communication
building
big
and/or
communicate
database
least
proven
use
e.g

```
[ ]: # Create a stacked bar plot of complaint types by city
     pivot_table = pd.pivot_table(df, values='Complaint Type', index='City', aggfunc='count')

     # Plot a stacked bar plot for each city with the types of complaints
     ax = pivot_table.plot(kind='bar', stacked=True, figsize=(10,6))
     ax.set_xlabel('Complaint Type')
     ax.set_ylabel('Count')
     ax.set_title('Types of Complaints in Each City')

     plt.show()
```



```
# Convert the 'Created Date' and 'Closed Date' columns to datetime format
df['Created Date'] = pd.to_datetime(df['Created Date'], format='%m/%d/%Y %I:%M:%S %p')
df['Closed Date'] = pd.to_datetime(df['Closed Date'], format='%m/%d/%Y %I:%M:%S %p')

# Calculate the Request_Closing_Time for each complaint
df['Request_Closing_Time'] = df['Closed Date'] - df['Created Date']

# Group the data by city and complaint type, and calculate the mean of the Request_Closing_Time column
df_grouped = df.groupby(['City', 'Complaint Type'])['Request_Closing_Time'].mean().reset_index()

# Sort the data by the mean of the Request_Closing_Time column
df_sorted = df_grouped.sort_values(by='Request_Closing_Time')

# Print the sorted data
print(df_sorted)
```

	City	Complaint Type	Request_Closing_Time
4	ARVERNE	Drinking	0 days 00:14:19
585	RIDGEWOOD	Posting Advertisement	0 days 00:14:55
170	EAST ELMHURST	Bike/Roller/Skate Chronic	0 days 00:15:08
497	OZONE PARK	Illegal Fireworks	0 days 00:19:51
520	QUEENS	Urinating in Public	0 days 00:20:26
..	...	...	...
482	OAKLAND GARDENS	Homeless Encampment	1 days 04:39:16
653	SOUTH RICHMOND HILL	Bike/Roller/Skate Chronic	1 days 06:54:37
63	BELLEROSE	Homeless Encampment	1 days 15:08:50
525	QUEENS VILLAGE	Graffiti	2 days 05:19:55
510	QUEENS	Animal in a Park	14 days 00:50:34

[294 rows x 3 columns]

In [41]:

```
# Convert the 'Created Date' and 'Closed Date' columns to datetime format
df['Created Date'] = pd.to_datetime(df['Created Date'], format='%m/%d/%Y %I:%M:%S %p')
df['Closed Date'] = pd.to_datetime(df['Closed Date'], format='%m/%d/%Y %I:%M:%S %p')

# Calculate the Request_Closing_Time for each complaint
df['Request_Closing_Time'] = df['Closed Date'] - df['Created Date']

# Calculate the average Request_Closing_Time for each complaint type
avg_time = df.groupby('Complaint Type')['Request_Closing_Time'].mean()

# Create a bar chart to visualize the average Request_Closing_Time for each complaint type
fig, ax = plt.subplots(figsize=(30,5))
ax.bar(avg_time.index, avg_time.values)
ax.set_xticklabels(avg_time.index, rotation=90)
ax.set_xlabel('Complaint Type')
ax.set_ylabel('Average Request Closing Time')
ax.set_title('Average Request Closing Time by Complaint Type')
plt.show()
```

```
C:\Users\91805\AppData\Local\Temp\ipykernel_7012\2148917335.py:14: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_xticklabels(avg_time.index, rotation=90)
```