Customer_Service_Request

November 21, 2023

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

#1. Understand the dataset: 1.1 Import the dataset 1.2 Visualize the dataset 1.3 Print the columns of the DataFrame 1.4 Identify the shape of the dataset 1.5 Identify the variables with null values

[3]:
```
# 1.1 Import the datase
df = pd.read_csv("311_Service_Requests_from_2010_to_Present.csv",␣
 ↪low_memory=False)
```

[3]:
```
# 1.2 Visualize the dataset
df
```

[3]:
```
        Unique Key             Created Date             Closed Date Agency  \
0         32310363  12/31/2015 11:59:45 PM  01/01/2016 12:55:15 AM   NYPD
1         32309934  12/31/2015 11:59:44 PM  01/01/2016 01:26:57 AM   NYPD
2         32309159  12/31/2015 11:59:29 PM  01/01/2016 04:51:03 AM   NYPD
3         32305098  12/31/2015 11:57:46 PM  01/01/2016 07:43:13 AM   NYPD
4         32306529  12/31/2015 11:56:58 PM  01/01/2016 03:24:42 AM   NYPD
…            …                      …                      …    …
257866    30598450  05/11/2015 07:31:50 PM  05/11/2015 08:02:37 PM   NYPD
257867    30591102  05/11/2015 07:28:52 PM  05/11/2015 07:58:26 PM   NYPD
257868    30595924  05/11/2015 07:28:11 PM  05/11/2015 10:38:36 PM   NYPD
257869    30595246  05/11/2015 07:24:54 PM  05/11/2015 11:32:36 PM   NYPD
257870    30590985  05/11/2015 07:24:53 PM  05/12/2015 01:25:35 AM   NYPD

                            Agency Name           Complaint Type  \
0       New York City Police Department  Noise - Street/Sidewalk
1       New York City Police Department         Blocked Driveway
2       New York City Police Department         Blocked Driveway
3       New York City Police Department          Illegal Parking
4       New York City Police Department          Illegal Parking
…                                     …                       …
257866  New York City Police Department          Illegal Parking
257867  New York City Police Department         Derelict Vehicle
257868  New York City Police Department         Blocked Driveway
```

```
257869  New York City Police Department          Illegal Parking
257870  New York City Police Department                  Traffic

                          Descriptor    Location Type  Incident Zip  \
0                   Loud Music/Party  Street/Sidewalk       10034.0
1                          No Access  Street/Sidewalk       11105.0
2                          No Access  Street/Sidewalk       10458.0
3       Commercial Overnight Parking  Street/Sidewalk       10461.0
4                   Blocked Sidewalk  Street/Sidewalk       11373.0
…                              …              …           …
257866               Blocked Hydrant  Street/Sidewalk       11219.0
257867            With License Plate  Street/Sidewalk       10460.0
257868                     No Access  Street/Sidewalk       11228.0
257869      Unauthorized Bus Layover  Street/Sidewalk       10013.0
257870         Truck Route Violation  Street/Sidewalk       11432.0

             Incident Address  … Bridge Highway Name  \
0         71 VERMILYEA AVENUE  …                 NaN
1             27-07 23 AVENUE  …                 NaN
2       2897 VALENTINE AVENUE  …                 NaN
3         2940 BAISLEY AVENUE  …                 NaN
4               87-14 57 ROAD  …                 NaN
…                       …  …                 …
257866         1057 66 STREET  …                 NaN
257867       629 BAKER AVENUE  …                 NaN
257868         1302 76 STREET  …                 NaN
257869       38 GREENE STREET  …                 NaN
257870                    NaN  …                 NaN

       Bridge Highway Direction Road Ramp Bridge Highway Segment  \
0                           NaN       NaN                    NaN
1                           NaN       NaN                    NaN
2                           NaN       NaN                    NaN
3                           NaN       NaN                    NaN
4                           NaN       NaN                    NaN
…                         …       …                    …
257866                      NaN       NaN                    NaN
257867                      NaN       NaN                    NaN
257868                      NaN       NaN                    NaN
257869                      NaN       NaN                    NaN
257870                      NaN       NaN                    NaN

       Garage Lot Name Ferry Direction Ferry Terminal Name   Latitude  \
0                  NaN             NaN                 NaN  40.865682
1                  NaN             NaN                 NaN  40.775945
2                  NaN             NaN                 NaN  40.870325
3                  NaN             NaN                 NaN  40.835994
```

```
4                  NaN             NaN                 NaN  40.733060
…                …             …                 …        …
257866             NaN             NaN                 NaN  40.628646
257867             NaN             NaN                 NaN  40.841797
257868             NaN             NaN                 NaN  40.619475
257869             NaN             NaN                 NaN  40.721757
257870             NaN             NaN                 NaN        NaN

        Longitude                                     Location
0      -73.923501   (40.86568153633767, -73.92350095571744)
1      -73.915094  (40.775945312321085, -73.91509393898605)
2      -73.888525  (40.870324522111424, -73.88852464418646)
3      -73.828379   (40.83599404683083, -73.82837939584206)
4      -73.874170  (40.733059618956815, -73.87416975810375)
…           …                                       …
257866 -74.007537   (40.62864560850486, -74.00753663120338)
257867 -73.866270   (40.84179712181627, -73.86626957360414)
257868 -74.007698   (40.61947525219153, -74.00769769033148)
257869 -74.002078   (40.72175682225491, -74.00207799450742)
257870        NaN                                       NaN

[257871 rows x 53 columns]
```

[4]:

```python
# 1.3 Print the columns of the DataFrame
df.columns
```

[4]:

```
Index(['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'],
      dtype='object')
```

[5]:

```python
# 1.4 Identify the shape of the dataset
df.shape
```

[5]:

```
(257871, 53)
```

[6]:

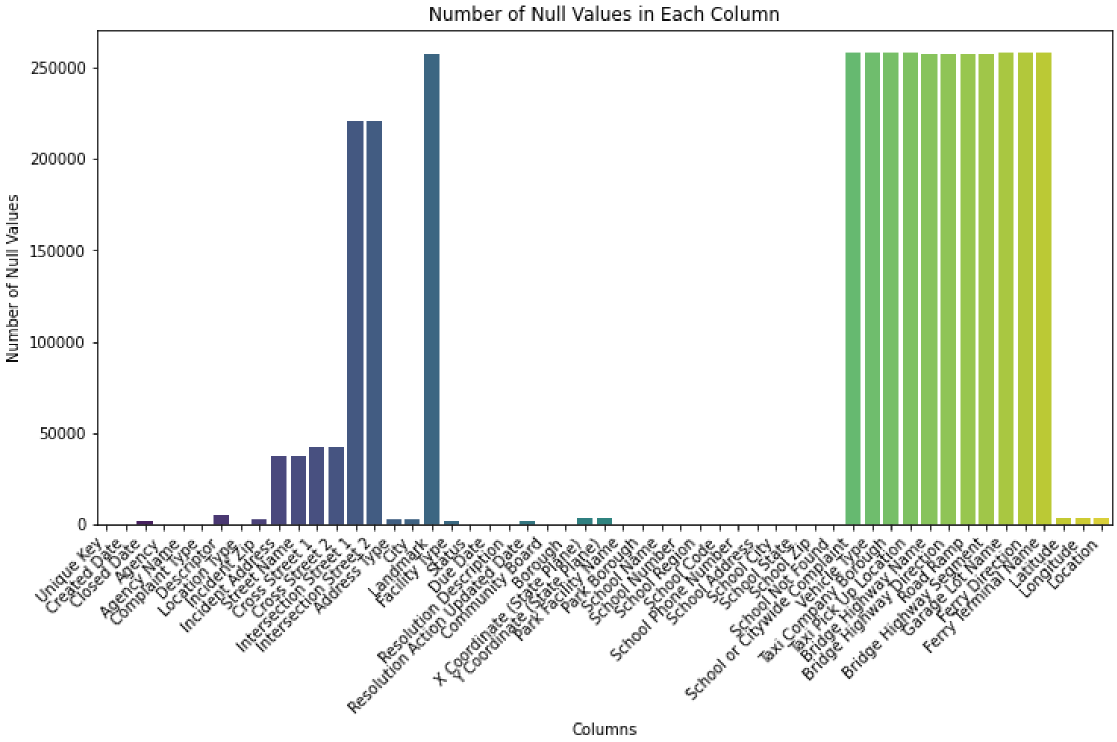
```python
# 1.5 Identify the variables with null values
missing_values = df.isnull().sum()
print("Missing Values:")
print(missing_values)
```

```
Missing Values:
Unique Key                             0
Created Date                           0
Closed Date                         1936
Agency                                 0
Agency Name                            0
Complaint Type                         0
Descriptor                          5269
Location Type                        130
Incident Zip                        2304
Incident Address                   37779
Street Name                        37779
Cross Street 1                     42073
Cross Street 2                     42486
Intersection Street 1             220568
Intersection Street 2             220990
Address Type                        2480
City                                2304
Landmark                          257573
Facility Type                       1940
Status                                 0
Due Date                               3
Resolution Description                 0
Resolution Action Updated Date      1958
Community Board                        0
Borough                                0
X Coordinate (State Plane)          3100
Y Coordinate (State Plane)          3100
Park Facility Name                     0
Park Borough                           0
School Name                            0
School Number                          0
School Region                          0
School Code                            0
School Phone Number                    0
School Address                         1
School City                            1
School State                           1
School Zip                             1
School Not Found                       1
School or Citywide Complaint      257871
Vehicle Type                      257871
```

```
Taxi Company Borough            257871
Taxi Pick Up Location           257871
Bridge Highway Name             257654
Bridge Highway Direction        257654
Road Ramp                       257682
Bridge Highway Segment          257682
Garage Lot Name                 257871
Ferry Direction                 257870
Ferry Terminal Name             257869
Latitude                          3101
Longitude                         3101
Location                          3101
dtype: int64
```

[8]:

```
# 2.1 Draw a frequency plot to show the number of null values in each column of␣
 ↪the DataFrame.

# Calculate the number of null values in each column
null_counts = df.isnull().sum()

# Plot the frequency of null values
plt.figure(figsize=(12, 6))
sns.barplot(x=null_counts.index, y=null_counts.values, palette="viridis")
plt.title("Number of Null Values in Each Column")
plt.xticks(rotation=45, ha="right")
plt.xlabel("Columns")
plt.ylabel("Number of Null Values")
plt.show()
```



[9]:
```
#2.2 Missing value treatment
#2.2.1 Remove the records whose Closed Date values are null

# Remove records with null values in the 'Closed Date' column
df = df.dropna(subset=['Closed Date'])

# Optionally, reset the index after removing rows
df = df.reset_index(drop=True)
```

[10]:
```
missing_values = df.isnull().sum()
print("Missing Values:")
print(missing_values)
```

```
Missing Values:
Unique Key                            0
Created Date                          0
Closed Date                           0
Agency                                0
Agency Name                           0
Complaint Type                        0
Descriptor                         5264
Location Type                       127
Incident Zip                        420
Incident Address                   37766
Street Name                        37766
Cross Street 1                     40523
Cross Street 2                     40584
Intersection Street 1             218992
Intersection Street 2             219062
Address Type                         596
City                                 420
Landmark                          255637
Facility Type                         12
Status                                 0
Due Date                               1
Resolution Description                 0
Resolution Action Updated Date        38
Community Board                        0
Borough                                0
X Coordinate (State Plane)          1216
Y Coordinate (State Plane)          1216
Park Facility Name                     0
Park Borough                           0
School Name                            0
School Number                          0
School Region                          0
School Code                            0
School Phone Number                    0
School Address                         1
School City                            1
School State                           1
School Zip                             1
School Not Found                       1
School or Citywide Complaint      255935
Vehicle Type                      255935
Taxi Company Borough              255935
Taxi Pick Up Location             255935
Bridge Highway Name               255718
Bridge Highway Direction          255718
Road Ramp                         255746
Bridge Highway Segment            255746
Garage Lot Name                   255935
Ferry Direction                   255935
Ferry Terminal Name               255935
Latitude                            1217
Longitude                           1217
Location                            1217
dtype: int64
```

[11]:

```
# 2.3.1 Calculate the time elapsed in "Closed Date" and "Created Date":

# Convert 'Closed Date' and 'Created Date' to datetime format
df['Closed Date'] = pd.to_datetime(df['Closed Date'])
df['Created Date'] = pd.to_datetime(df['Created Date'])

# Calculate time elapsed in hours
df['TimeElapsed'] = (df['Closed Date'] - df['Created Date']).dt.total_seconds()␣
 ↪/ 3600

# 2.3.2 Convert the calculated date to seconds:
df['TimeElapsedSeconds'] = df['TimeElapsed'] * 3600

#2.3.3 View the descriptive statistics for the newly created column:
# View descriptive statistics for the new column
print(df['TimeElapsedSeconds'].describe())
```

```
count    2.559350e+05
mean     1.584580e+04
std      2.216819e+04
min      6.100000e+01
25%      4.675000e+03
50%      9.961000e+03
75%      1.964300e+04
max      2.134342e+06
Name: TimeElapsedSeconds, dtype: float64
```

[12]:

```
df.head()
```

[12]:

```
   Unique Key        Created Date         Closed Date Agency  \
0    32310363 2015-12-31 23:59:45 2016-01-01 00:55:15   NYPD
1    32309934 2015-12-31 23:59:44 2016-01-01 01:26:57   NYPD
2    32309159 2015-12-31 23:59:29 2016-01-01 04:51:03   NYPD
3    32305098 2015-12-31 23:57:46 2016-01-01 07:43:13   NYPD
4    32306529 2015-12-31 23:56:58 2016-01-01 03:24:42   NYPD

                       Agency Name           Complaint Type  \
0  New York City Police Department  Noise - Street/Sidewalk
1  New York City Police Department         Blocked Driveway
2  New York City Police Department         Blocked Driveway
3  New York City Police Department          Illegal Parking
4  New York City Police Department          Illegal Parking

                    Descriptor    Location Type  Incident Zip  \
0             Loud Music/Party  Street/Sidewalk       10034.0
1                    No Access  Street/Sidewalk       11105.0
2                    No Access  Street/Sidewalk       10458.0
```

```
3  Commercial Overnight Parking  Street/Sidewalk       10461.0
4              Blocked Sidewalk  Street/Sidewalk       11373.0

        Incident Address  … Road Ramp Bridge Highway Segment  \
0    71 VERMILYEA AVENUE  …       NaN                    NaN
1         27-07 23 AVENUE  …       NaN                    NaN
2  2897 VALENTINE AVENUE  …       NaN                    NaN
3    2940 BAISLEY AVENUE  …       NaN                    NaN
4           87-14 57 ROAD  …       NaN                    NaN

  Garage Lot Name Ferry Direction Ferry Terminal Name   Latitude  Longitude  \
0             NaN             NaN                 NaN  40.865682 -73.923501
1             NaN             NaN                 NaN  40.775945 -73.915094
2             NaN             NaN                 NaN  40.870325 -73.888525
3             NaN             NaN                 NaN  40.835994 -73.828379
4             NaN             NaN                 NaN  40.733060 -73.874170

                                   Location TimeElapsed TimeElapsedSeconds
0   (40.86568153633767, -73.92350095571744)    0.925000             3330.0
1  (40.775945312321085, -73.91509393898605)    1.453611             5233.0
2  (40.870324522111424, -73.88852464418646)    4.859444            17494.0
3   (40.83599404683083, -73.82837939584206)    7.757500            27927.0
4  (40.733059618956815, -73.87416975810375)    3.462222            12464.0

[5 rows x 55 columns]
```

[13]:

```python
# 2.3.4 Check the number of null values in the Complaint_Type and City columns

# Check the number of null values in Complaint_Type and City columns
null_counts = df[['Complaint Type', 'City']].isna().sum()

# Display the null counts
print(null_counts)
```

```
Complaint Type      0
City              420
dtype: int64
```

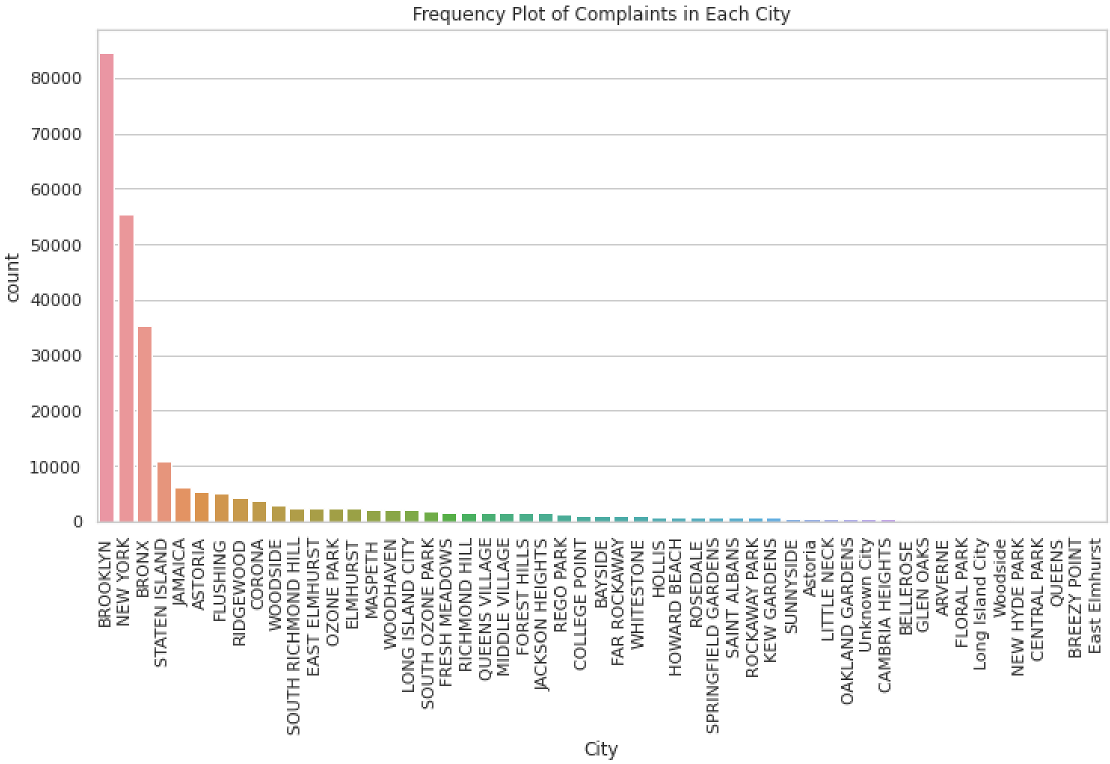
[14]:

```python
# 2.3.5 Impute the NA value with Unknown City

# Impute missing values in the 'City' column with 'Unknown City'
df['City'].fillna('Unknown City', inplace=True)
```

[15]:

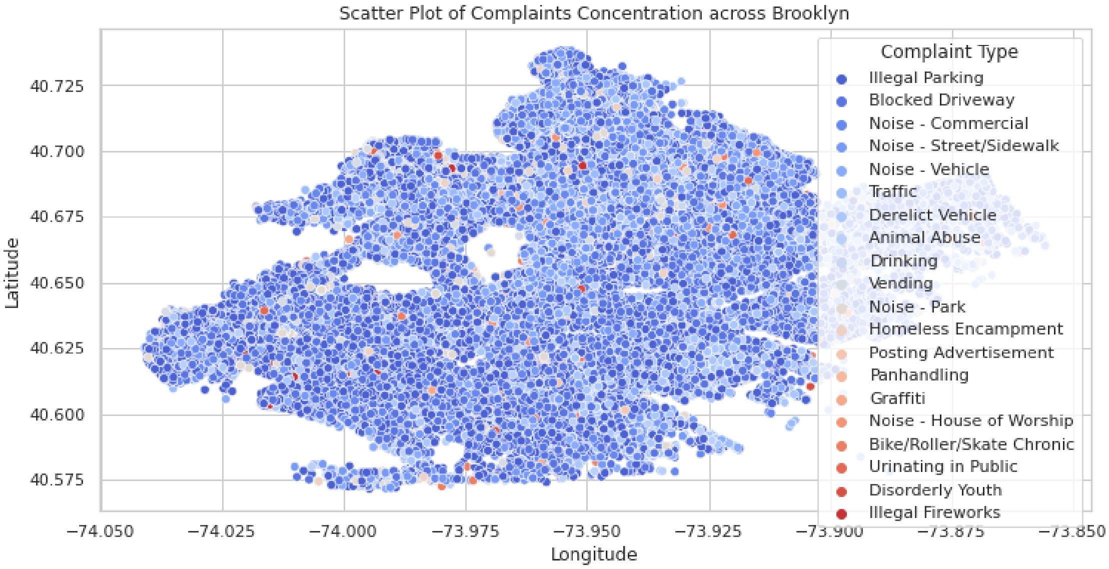
```python
# 2.3.6 Draw a frequency plot for the complaints in each city

# Set the style of seaborn
sns.set(style="whitegrid")
```

```
# Draw a frequency plot for the complaints in each city
plt.figure(figsize=(12, 6))
sns.countplot(x='City', data=df, order=df['City'].value_counts().index)
plt.xticks(rotation=90)
plt.title('Frequency Plot of Complaints in Each City')
plt.show()
```



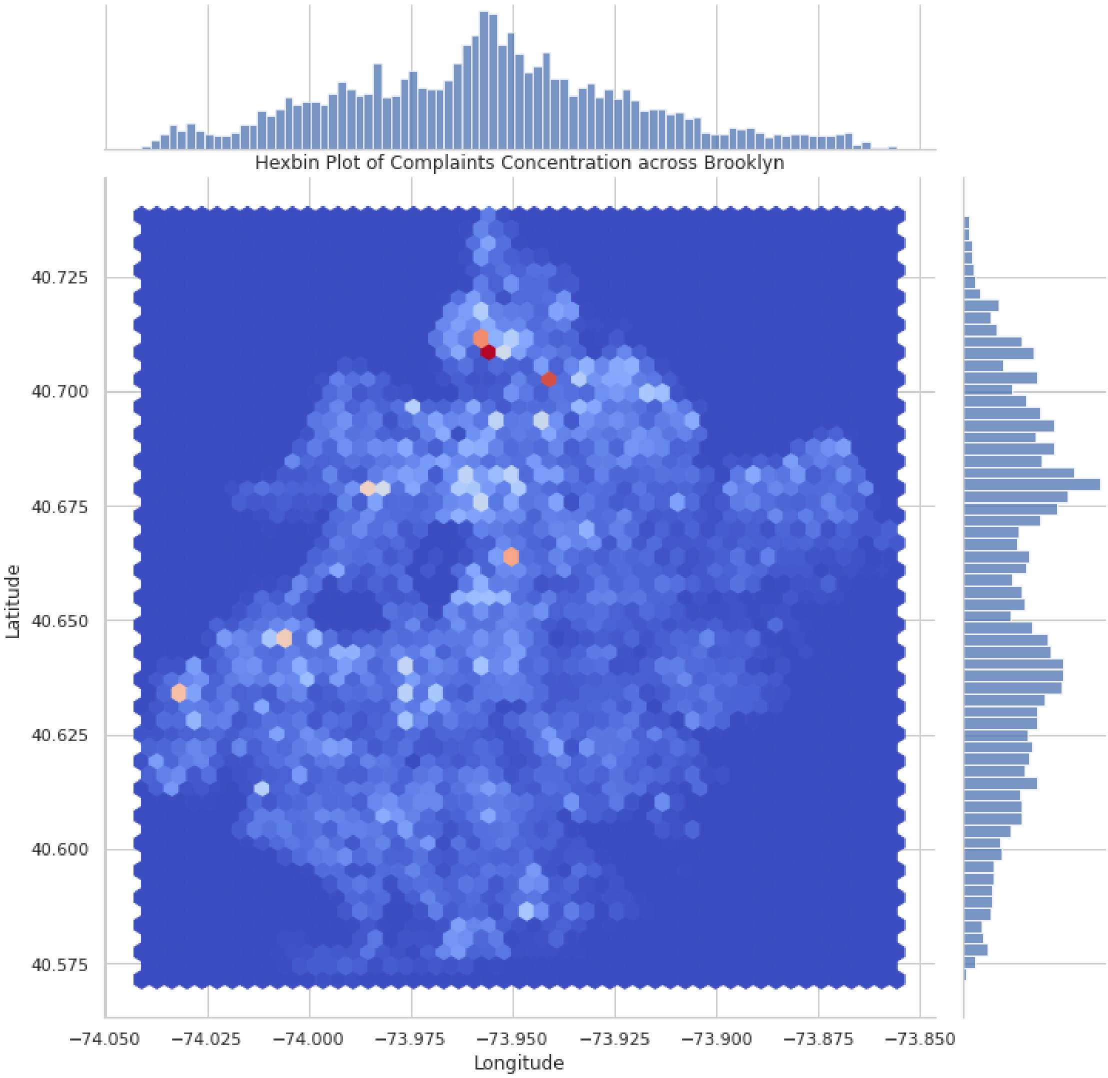
[16]:

```
# 2.3.7 Create a scatter and hexbin plot of the concentration of complaints␣
↪across Brooklyn.

# Filter the dataframe to include only Brooklyn
brooklyn_df = df[df['Borough'] == 'BROOKLYN']

# Set the style of seaborn
sns.set(style="whitegrid")

# Create a scatter plot
plt.figure(figsize=(12, 6))
sns.scatterplot(x='Longitude', y='Latitude', data=brooklyn_df, hue='Complaint␣
↪Type', palette='coolwarm')
plt.title('Scatter Plot of Complaints Concentration across Brooklyn')
plt.show()
```



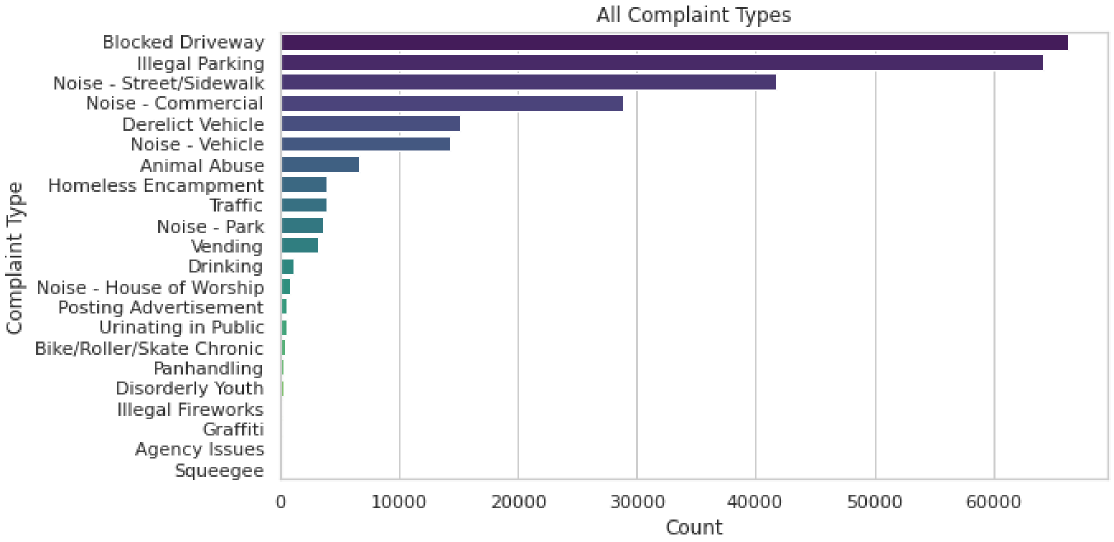
[17]:
```
# Filter the dataframe to include only Brooklyn
brooklyn_df = df[df['Borough'] == 'BROOKLYN']

# Set the style of seaborn
sns.set(style="whitegrid")

# Create a hexbin plot
plt.figure(figsize=(12, 8))
sns.jointplot(x='Longitude', y='Latitude', data=brooklyn_df, kind='hex',␣
  ↪cmap='coolwarm', height=10)
plt.title('Hexbin Plot of Complaints Concentration across Brooklyn')
plt.show()
```

```
<Figure size 864x576 with 0 Axes>
```



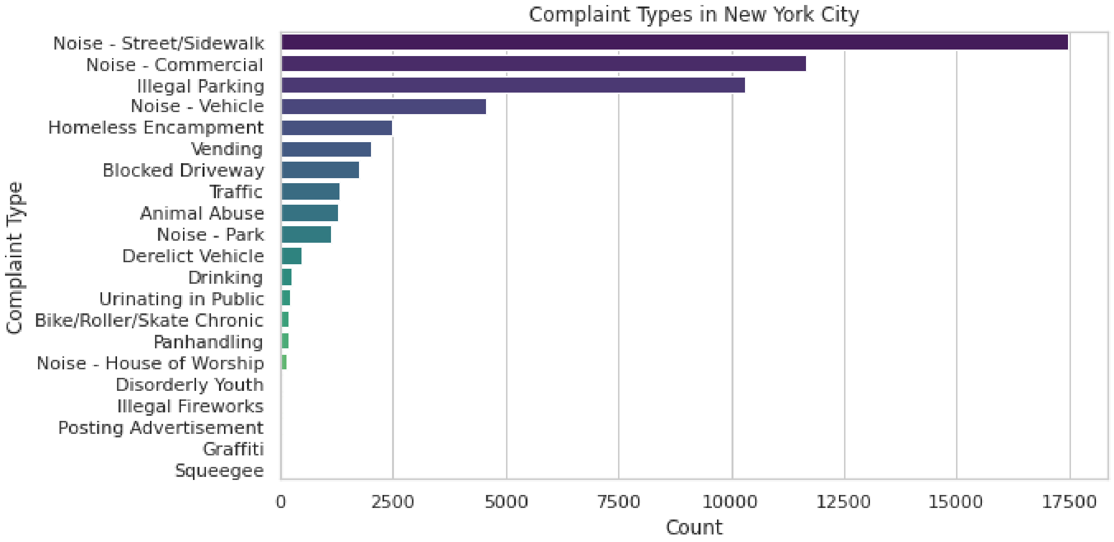
```
[ ]: # 3. Find major types of complaints:
```

```
[18]: # 3.1 Plot a bar graph to show the types of complaints
plt.figure(figsize=(9, 5))
all_complaints = df['Complaint Type'].value_counts()
sns.barplot(x=all_complaints.values, y=all_complaints.index, palette='viridis')
plt.title('All Complaint Types')
plt.xlabel('Count')
plt.ylabel('Complaint Type')
plt.show()
```



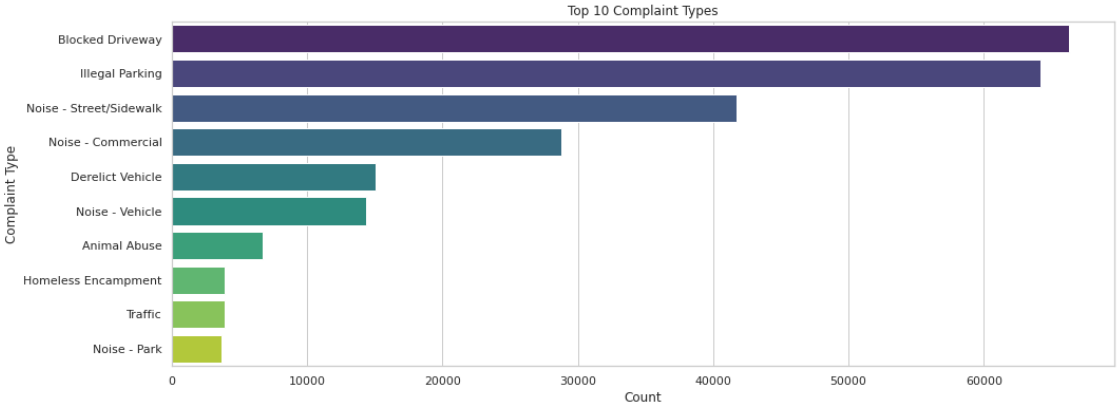
[19]:

```
# 3.2 Check the frequency of various types of complaints for New York City.
# Filter data for New York City
nyc_complaints = df[df['City'] == 'NEW YORK']['Complaint Type'].value_counts()

# Plot a bar graph for NYC complaint types
plt.figure(figsize=(9, 5))
sns.barplot(x=nyc_complaints.values, y=nyc_complaints.index, palette='viridis')
plt.title('Complaint Types in New York City')
plt.xlabel('Count')
plt.ylabel('Complaint Type')
plt.show()
```



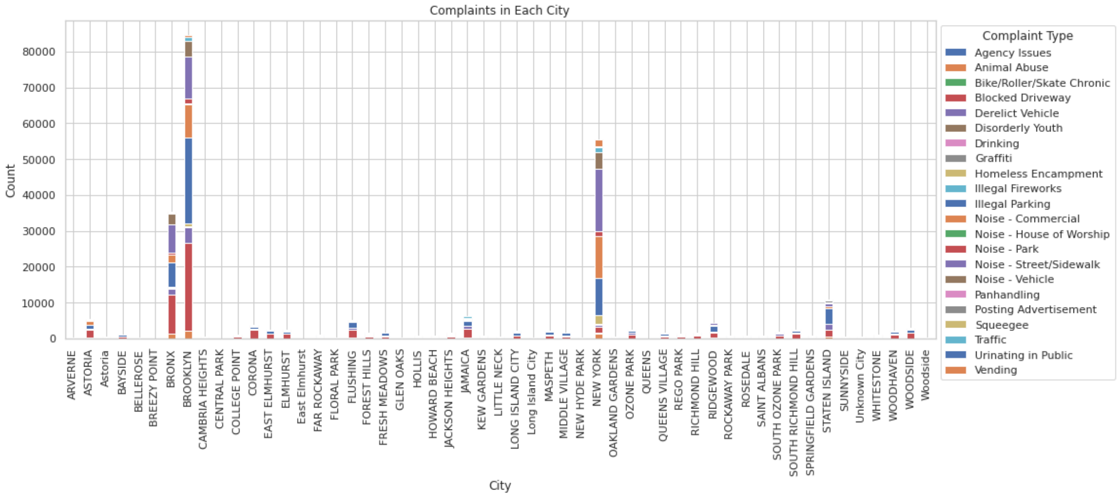
[20]:

```
# 3.3 Find the top 10 complaint types
top_10_complaints = df['Complaint Type'].value_counts().head(10)
# Plot a bar graph for the top 10 complaint types
plt.figure(figsize=(16, 6))
sns.barplot(x=top_10_complaints.values, y=top_10_complaints.index,␣
  ↪palette='viridis')
plt.title('Top 10 Complaint Types')
plt.xlabel('Count')
plt.ylabel('Complaint Type')
plt.show()
```



[21]:

```
# 3.4 Display the various types of complaints in each city

# Group by 'City' and 'Complaint Type' and count the occurrences
complaints_by_city = df.groupby(['City', 'Complaint Type']).size().unstack()

# Plot a stacked bar chart
complaints_by_city.plot(kind='bar', stacked=True, figsize=(16, 6))
plt.title('Complaints in Each City')
plt.xlabel('City')
plt.ylabel('Count')
plt.legend(title='Complaint Type', bbox_to_anchor=(1, 1))
plt.show()
```



[4]:

```
# 3.5 Create a DataFrame, df_new, which contains cities as columns and␣
  ↪complaint types in rows.

# Group by 'City' and 'Complaint Type' and count the occurrences
complaints_by_city = df.groupby(['City', 'Complaint Type']).size().unstack()

# Create a new DataFrame with cities as columns and complaint types as rows
df_new = complaints_by_city.transpose()

# Display the new DataFrame
print(df_new)
```

```
City                       ARVERNE  ASTORIA  Astoria  BAYSIDE  BELLEROSE  \
Complaint Type
Animal Abuse                  36.0    111.0      NaN     30.0        6.0
Blocked Driveway              33.0   2257.0     93.0    325.0       81.0
Derelict Vehicle              25.0    293.0      9.0    173.0       70.0
Disorderly Youth               2.0      3.0      NaN      1.0        2.0
Homeless Encampment            4.0     31.0      NaN      2.0        1.0
Illegal Parking               54.0    941.0    164.0    438.0       95.0
Noise - Commercial             2.0   1043.0    211.0     39.0       22.0
Noise - House of Worship      11.0     14.0      NaN      2.0        NaN
Noise - Park                   2.0     57.0      NaN      4.0        1.0
Noise - Street/Sidewalk       28.0    334.0     86.0      8.0       13.0
Noise - Vehicle                6.0    159.0      NaN     11.0        8.0
Panhandling                    1.0      1.0      NaN      NaN        1.0
Urinating in Public            1.0      9.0      NaN      NaN        1.0
Vending                        1.0     51.0      NaN      1.0        NaN
Bike/Roller/Skate Chronic      NaN     15.0      NaN      NaN        1.0
Drinking                       NaN     31.0      NaN      1.0        NaN
```

```
Graffiti                       NaN      2.0      NaN      3.0        NaN
Illegal Fireworks              NaN      4.0      NaN      NaN        1.0
Traffic                        NaN     40.0      NaN      8.0        6.0
Posting Advertisement          NaN      NaN      NaN      NaN        NaN
Squeegee                       NaN      NaN      NaN      NaN        NaN

City                       BREEZY POINT    BRONX  BROOKLYN  CAMBRIA HEIGHTS  \
Complaint Type
Animal Abuse                        1.0   1205.0    2055.0             11.0
Blocked Driveway                    3.0  10967.0   24412.0            127.0
Derelict Vehicle                    NaN   1680.0    4466.0             94.0
Disorderly Youth                    NaN     50.0      66.0              NaN
Homeless Encampment                 NaN    212.0     769.0              4.0
Illegal Parking                    14.0   6773.0   23758.0             63.0
Noise - Commercial                  4.0   2113.0    9266.0             11.0
Noise - House of Worship            NaN     72.0     297.0              2.0
Noise - Park                        NaN    504.0    1403.0              NaN
Noise - Street/Sidewalk             1.0   7943.0   11645.0             23.0
Noise - Vehicle                     1.0   2953.0    4388.0             61.0
Panhandling                         NaN     17.0      42.0              NaN
Urinating in Public                 NaN     46.0     127.0              NaN
Vending                             NaN    295.0     441.0              NaN
Bike/Roller/Skate Chronic           NaN     16.0      97.0              NaN
Drinking                            NaN    172.0     218.0              NaN
Graffiti                            NaN      9.0      33.0              NaN
Illegal Fireworks                   NaN     24.0      56.0              1.0
Traffic                             NaN    291.0     959.0              6.0
Posting Advertisement               NaN     15.0      43.0              NaN
Squeegee                            NaN      NaN       NaN              NaN

City                       CENTRAL PARK  …  SAINT ALBANS  SOUTH OZONE PARK  \
Complaint Type                           …
Animal Abuse                        NaN  …          26.0              51.0
Blocked Driveway                    NaN  …         213.0             827.0
Derelict Vehicle                    NaN  …         168.0             308.0
Disorderly Youth                    NaN  …           1.0               1.0
Homeless Encampment                 NaN  …           6.0               4.0
Illegal Parking                     2.0  …         166.0             420.0
Noise - Commercial                  NaN  …          27.0              59.0
Noise - House of Worship            NaN  …           1.0               3.0
Noise - Park                        NaN  …           1.0               4.0
Noise - Street/Sidewalk            79.0  …          75.0              90.0
Noise - Vehicle                     NaN  …          28.0              69.0
Panhandling                         NaN  …           NaN               NaN
Urinating in Public                 NaN  …           1.0               2.0
Vending                             NaN  …           2.0               5.0
Bike/Roller/Skate Chronic           NaN  …           NaN               1.0
Drinking                            NaN  …           3.0              10.0
```

```
Graffiti                            NaN  …           NaN               NaN
Illegal Fireworks                   NaN  …           NaN               1.0
Traffic                             NaN  …          11.0              24.0
Posting Advertisement               NaN  …           NaN               NaN
Squeegee                            NaN  …           NaN               NaN

City                       SOUTH RICHMOND HILL  SPRINGFIELD GARDENS  \
Complaint Type
Animal Abuse                              22.0                 22.0
Blocked Driveway                        1346.0                229.0
Derelict Vehicle                         246.0                181.0
Disorderly Youth                           2.0                  NaN
Homeless Encampment                       10.0                  4.0
Illegal Parking                          409.0                198.0
Noise - Commercial                       181.0                 35.0
Noise - House of Worship                   3.0                  1.0
Noise - Park                               2.0                  1.0
Noise - Street/Sidewalk                   74.0                 34.0
Noise - Vehicle                           62.0                 38.0
Panhandling                                NaN                  2.0
Urinating in Public                        NaN                  3.0
Vending                                   21.0                  1.0
Bike/Roller/Skate Chronic                  1.0                  NaN
Drinking                                  18.0                  6.0
Graffiti                                   NaN                  NaN
Illegal Fireworks                          2.0                  1.0
Traffic                                   10.0                 11.0
Posting Advertisement                      NaN                  1.0
Squeegee                                   NaN                  NaN

City                       STATEN ISLAND  SUNNYSIDE  WHITESTONE  WOODHAVEN  \
Complaint Type
Animal Abuse                       478.0       33.0        25.0       39.0
Blocked Driveway                  1877.0      172.0       181.0      939.0
Derelict Vehicle                  1533.0       10.0       190.0      262.0
Disorderly Youth                    12.0        2.0         1.0        NaN
Homeless Encampment                 60.0        9.0         NaN        8.0
Illegal Parking                   4218.0       99.0       443.0      575.0
Noise - Commercial                 590.0      126.0        15.0      120.0
Noise - House of Worship            14.0        NaN         NaN        3.0
Noise - Park                        52.0       12.0         6.0        2.0
Noise - Street/Sidewalk            697.0       56.0        29.0       79.0
Noise - Vehicle                    276.0       37.0        21.0       60.0
Panhandling                         12.0        NaN         NaN        NaN
Urinating in Public                 13.0        2.0         NaN        2.0
Vending                             21.0       13.0         NaN        5.0
Bike/Roller/Skate Chronic            4.0        2.0         4.0        2.0
Drinking                           163.0       10.0         2.0        2.0
```

```
Graffiti                             2.0        1.0         NaN        NaN
Illegal Fireworks                   10.0        NaN         1.0        NaN
Traffic                            178.0       15.0        15.0        4.0
Posting Advertisement              516.0        2.0         NaN        NaN
Squeegee                             NaN        NaN         NaN        NaN

City                       WOODSIDE  Woodside
Complaint Type
Animal Abuse                   58.0       NaN
Blocked Driveway             1389.0       9.0
Derelict Vehicle              210.0       1.0
Disorderly Youth                NaN       NaN
Homeless Encampment            27.0       NaN
Illegal Parking               746.0      75.0
Noise - Commercial            180.0       NaN
Noise - House of Worship        3.0       NaN
Noise - Park                   35.0       NaN
Noise - Street/Sidewalk       212.0       3.0
Noise - Vehicle                95.0       NaN
Panhandling                     NaN       NaN
Urinating in Public             8.0       NaN
Vending                        14.0       NaN
Bike/Roller/Skate Chronic       4.0       NaN
Drinking                       15.0       NaN
Graffiti                        2.0       NaN
Illegal Fireworks               1.0       NaN
Traffic                        34.0       NaN
Posting Advertisement           NaN       NaN
Squeegee                        NaN       NaN

[21 rows x 52 columns]
```

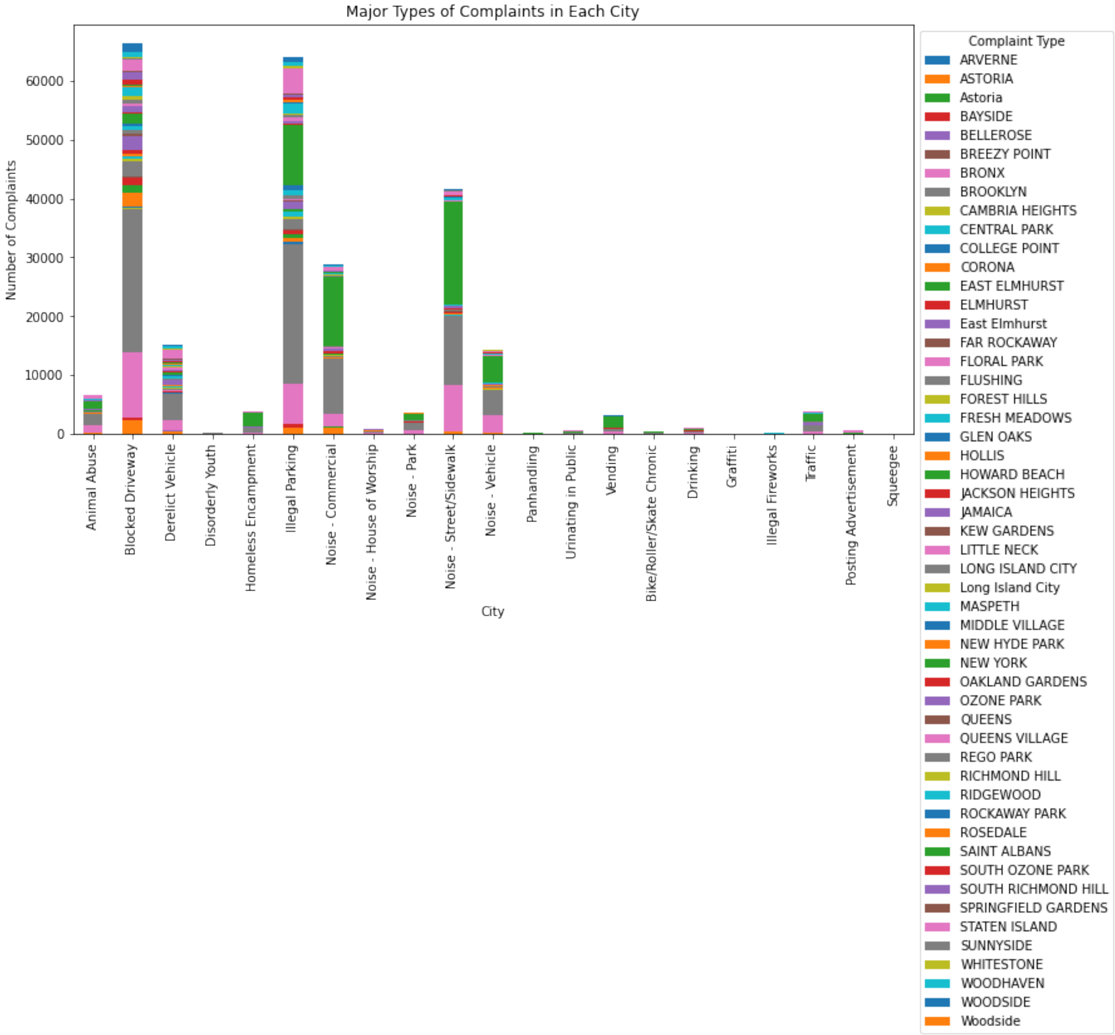
[9]:
```
print(df_new.isna().sum())
```

```
City
ARVERNE                 7
ASTORIA                 2
Astoria                16
BAYSIDE                 6
BELLEROSE               6
BREEZY POINT           15
BRONX                   1
BROOKLYN                1
CAMBRIA HEIGHTS        10
CENTRAL PARK           19
COLLEGE POINT           8
CORONA                  4
EAST ELMHURST           3
```

```
ELMHURST                 4
East Elmhurst           19
FAR ROCKAWAY             6
FLORAL PARK             12
FLUSHING                 1
FOREST HILLS             1
FRESH MEADOWS            8
GLEN OAKS               10
HOLLIS                   8
HOWARD BEACH             6
JACKSON HEIGHTS          4
JAMAICA                  2
KEW GARDENS              8
LITTLE NECK             11
LONG ISLAND CITY         3
Long Island City        16
MASPETH                  4
MIDDLE VILLAGE           9
NEW HYDE PARK           16
NEW YORK                 0
OAKLAND GARDENS          8
OZONE PARK               3
QUEENS                  11
QUEENS VILLAGE           4
REGO PARK                7
RICHMOND HILL            5
RIDGEWOOD                4
ROCKAWAY PARK            7
ROSEDALE                 6
SAINT ALBANS             6
SOUTH OZONE PARK         4
SOUTH RICHMOND HILL      5
SPRINGFIELD GARDENS      4
STATEN ISLAND            1
SUNNYSIDE                4
WHITESTONE               8
WOODHAVEN                6
WOODSIDE                 4
Woodside                17
dtype: int64
```

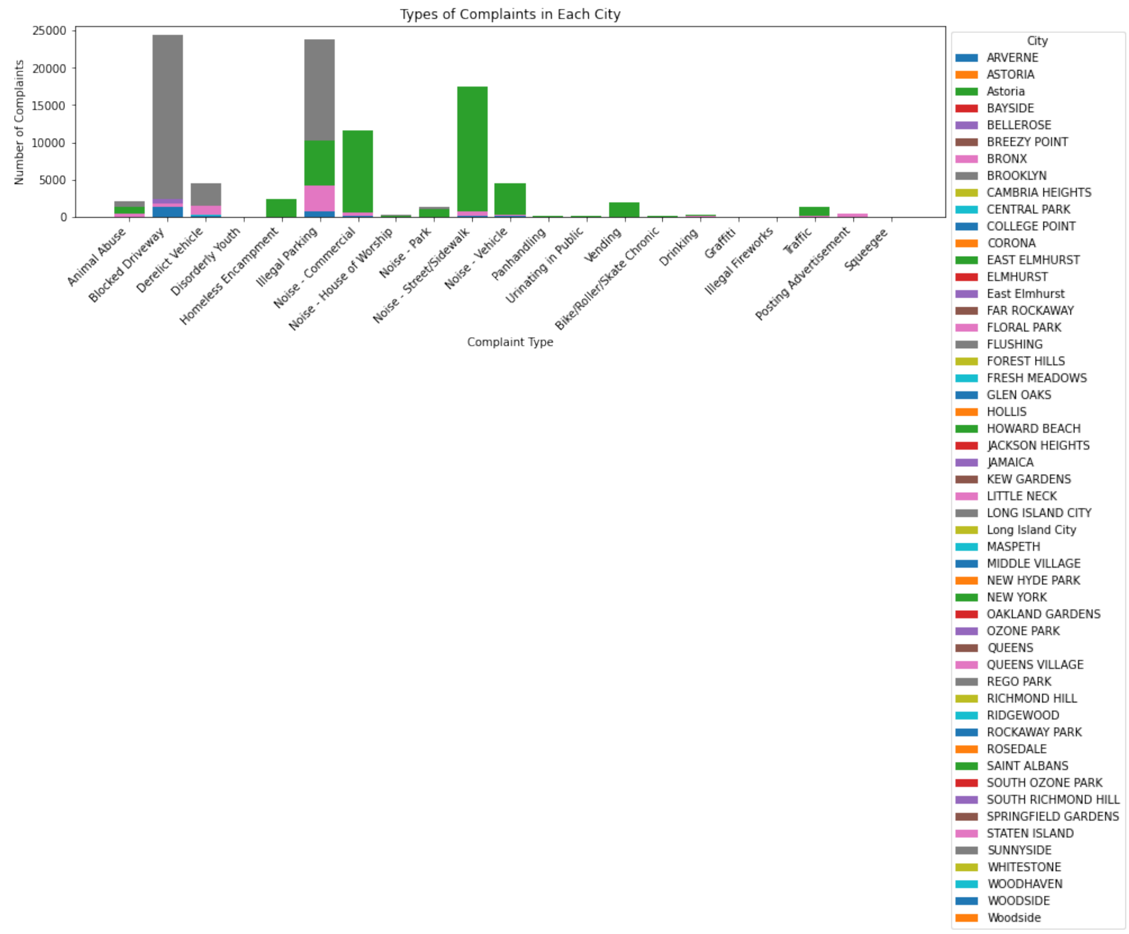
[5]:

```python
# 4. Visualize the major types of complaints in each city

# Plot a stacked bar chart for major types of complaints in each city
df_new.plot(kind='bar', stacked=True, figsize=(12, 6))
plt.title('Major Types of Complaints in Each City')
plt.xlabel('City')
plt.ylabel('Number of Complaints')
plt.legend(title='Complaint Type', bbox_to_anchor=(1, 1))
plt.show()
```



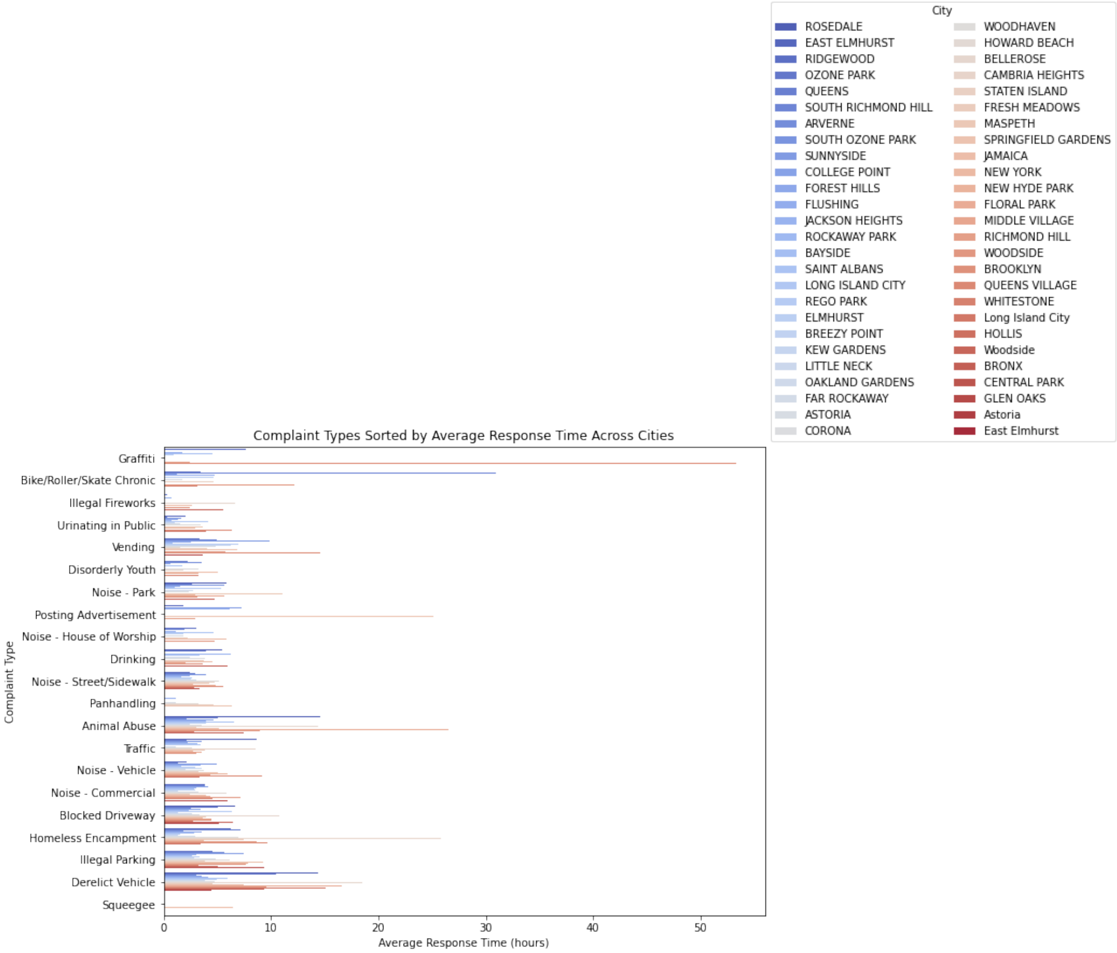
[5]:
```
# 4.1 Draw another chart that shows the types of complaints in each city in a␣
 ↪single chart, where different colors show the
# different types of complaints.

# Set the figure size
plt.figure(figsize=(14, 8))

# Loop through each city and plot the bar chart
for city in df_new.columns:
    plt.bar(df_new.index, df_new[city], label=city)
```

```
# Customize the plot
plt.title('Types of Complaints in Each City')
plt.xlabel('Complaint Type')
plt.ylabel('Number of Complaints')
plt.legend(title='City', bbox_to_anchor=(1, 1))
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better␣
 ↪readability

plt.tight_layout()
plt.show()
```



[42]:
```
# 4.2 Sort the complaint types based on the average Request_Closing_Time␣
 ↪grouping them for different locations

# Convert 'Closed Date' and 'Created Date' to datetime format
df['Closed Date'] = pd.to_datetime(df['Closed Date'])
df['Created Date'] = pd.to_datetime(df['Created Date'])
```

```
# Calculate the time elapsed in hours
df['Response Time'] = (df['Closed Date'] - df['Created Date']).dt.
 ↪total_seconds() / 3600

# Group by 'City' and 'Complaint Type' and calculate the mean Response Time
avg_response_time_df = df.groupby(['City', 'Complaint Type'])['Response Time'].
 ↪mean().reset_index()

# Sort the DataFrame based on the rank within each city
sorted_df = avg_response_time_df.sort_values(by='Response Time')

# Plot the bar chart
plt.figure(figsize=(10, 8))
sns.barplot(x='Response Time', y='Complaint Type', hue='City', data=sorted_df,␣
 ↪palette='coolwarm')
plt.title('Complaint Types Sorted by Average Response Time Across Cities')
plt.xlabel('Average Response Time (hours)')
plt.ylabel('Complaint Type')
plt.legend(title='City', bbox_to_anchor=(1, 1), ncol=2)
plt.show()
```



```
[45]: # 5. See whether the average response time across different complaint types is␣
        ↪similar (overall).

      df['Response Time'] = (df['Closed Date'] - df['Created Date']).dt.
        ↪total_seconds() / 3600  # in hours

      # Group the data by complaint type and calculate the average response time
      average_response_time_by_type = df.groupby('Complaint Type')['Response Time'].
        ↪mean().reset_index()

      # Display the average response time for each type
      print(average_response_time_by_type)
```

```
               Complaint Type  Response Time
0               Agency Issues       5.720333
1                Animal Abuse       5.282471
2   Bike/Roller/Skate Chronic       3.806924
```

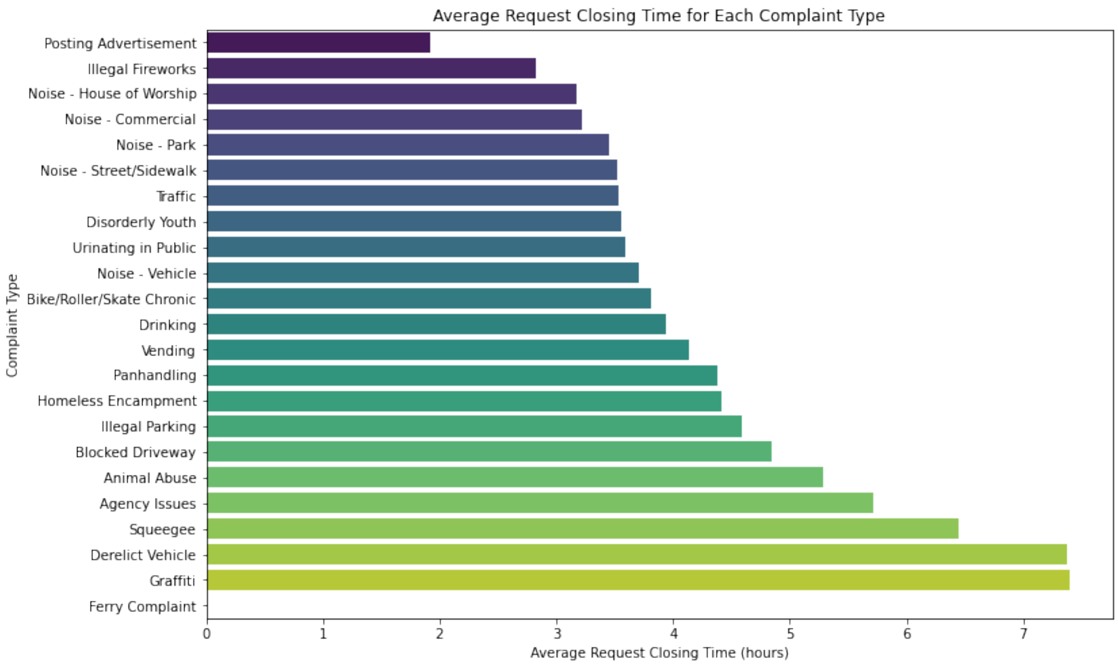
```
3            Blocked Driveway       4.847326
4            Derelict Vehicle       7.378833
5            Disorderly Youth       3.561831
6                    Drinking       3.944382
7             Ferry Complaint            NaN
8                    Graffiti       7.401216
9         Homeless Encampment       4.412038
10          Illegal Fireworks       2.826893
11            Illegal Parking       4.586629
12         Noise - Commercial       3.223074
13   Noise - House of Worship       3.169555
14               Noise - Park       3.456051
15    Noise - Street/Sidewalk       3.518111
16            Noise - Vehicle       3.712814
17                Panhandling       4.378483
18      Posting Advertisement       1.921980
19                   Squeegee       6.445694
20                    Traffic       3.538889
21         Urinating in Public       3.586154
22                    Vending       4.131121
```

[ ]:

```
# By observing average response time above, we can say that average response␣
 ↪time across different complaint types are not similiar.
```

[52]:

```
# 5.1 Visualize the average of Request_Closing_Time
# Create a new column Request_Closing_Time
df['Request_Closing_Time'] = (df['Closed Date'] - df['Created Date']).dt.
 ↪total_seconds() / 3600  # in hours

# 5.1 Visualize the average of Request_Closing_Time

# Group by 'Complaint Type' and calculate the mean Request_Closing_Time
avg_closing_time_by_type = df.groupby('Complaint Type')['Request_Closing_Time'].
 ↪mean().reset_index()

# Sort the DataFrame by average Request_Closing_Time
sorted_df = avg_closing_time_by_type.sort_values(by='Request_Closing_Time')

# Plot the sorted DataFrame
plt.figure(figsize=(12, 8))
sns.barplot(x='Request_Closing_Time', y='Complaint Type', data=sorted_df,␣
 ↪palette='viridis')
plt.title('Average Request Closing Time for Each Complaint Type')
plt.xlabel('Average Request Closing Time (hours)')
plt.ylabel('Complaint Type')
plt.show()
```



[7]:

```
# 6. Identify significant variables by performing a statistical analysis using␣
  ↪p-values and chi-square values

from scipy.stats import chi2_contingency

# Identify categorical variables
categorical_vars = df.select_dtypes(include='object').columns

# Set a maximum number of unique values for each variable to reduce the size of␣
  ↪the contingency table
max_unique_values = 10

# Perform chi-square test for each pair of categorical variables
significant_variables = []

for col1 in categorical_vars:
    for col2 in categorical_vars:
        if col1 != col2:
            if df[col1].nunique() <= max_unique_values and df[col2].nunique()␣
  ↪<= max_unique_values:
                contingency_table = pd.crosstab(df[col1], df[col2])

                # Check if the contingency table has non-zero size
                if contingency_table.size > 0:
                    chi2, p_value, _, _ = chi2_contingency(contingency_table)
                    if p_value < 0.05:  # You can adjust the significance level
                        significant_variables.append((col1, col2))

print("Significant Variables:", significant_variables)
```

```
Significant Variables: [('Agency Name', 'Status'), ('Agency Name', 'Borough'),
('Agency Name', 'Park Borough'), ('Address Type', 'Status'), ('Address Type',
'Borough'), ('Address Type', 'Park Borough'), ('Status', 'Agency Name'),
('Status', 'Address Type'), ('Status', 'Borough'), ('Status', 'Park Borough'),
('Borough', 'Agency Name'), ('Borough', 'Address Type'), ('Borough', 'Status'),
('Borough', 'Park Borough'), ('Park Borough', 'Agency Name'), ('Park Borough',
'Address Type'), ('Park Borough', 'Status'), ('Park Borough', 'Borough')]
```

[17]:

```
# 7. Perform a Kruskal-Wallis H test
from scipy.stats import kruskal

# Convert 'Closed Date' and 'Created Date' to datetime objects
df['Closed Date'] = pd.to_datetime(df['Closed Date'])
df['Created Date'] = pd.to_datetime(df['Created Date'])

# Create a new column 'Request_Closing_Time'
df['Request_Closing_Time'] = (df['Closed Date'] - df['Created Date']).dt.
 ↪total_seconds() / 3600  # in hours

# Identify continuous variable and categorical variable
continuous_var = 'Request_Closing_Time'
categorical_var = 'Complaint Type'  # Adjust this based on your actual column␣
 ↪name

# Perform Kruskal-Wallis H test
groups = [df[df[categorical_var] == category][continuous_var] for category in␣
 ↪df[categorical_var].unique()]
h_statistic, p_value = kruskal(*groups)

# Print overall statistics
print("Overall Statistics:")
print(df.groupby(categorical_var)[continuous_var].describe()[['min', 'max',␣
 ↪'std']])

# Interpret the results
print("\nKruskal-Wallis H Statistic:", h_statistic)
print("P-value:", p_value)

# Check for significance
if p_value < 0.05:
    print("There are significant differences between groups.")
```

```
else:
    print("There are no significant differences between groups.")
```

```
Overall Statistics:
                                min         max        std
Complaint Type
Agency Issues              1.131389   10.383611   3.756644
Animal Abuse               0.064722  519.254444   9.080823
Bike/Roller/Skate Chronic  0.071111   33.914444   4.396013
Blocked Driveway           0.047500  148.286667   5.750926
Derelict Vehicle           0.061667  223.370000  11.258130
Disorderly Youth           0.100833   28.057500   3.846192
Drinking                   0.082222   94.770000   5.372015
Graffiti                   0.156389   54.611944   9.781655
Homeless Encampment        0.091944   91.312222   5.417903
Illegal Fireworks          0.135000   27.852778   3.578705
Illegal Parking            0.043611  577.351667   6.052256
Noise - Commercial         0.016944   81.657778   4.019584
Noise - House of Worship   0.072222   49.091111   4.415029
Noise - Park               0.071389   57.680556   4.100291
Noise - Street/Sidewalk    0.038056  592.872778   5.359315
Noise - Vehicle            0.052222  147.447778   4.843991
Panhandling                0.149444  145.082222   9.289124
Posting Advertisement      0.040556   25.086944   2.345099
Squeegee                   6.104722    6.786667   0.482208
Traffic                    0.077778   60.132778   4.846149
Urinating in Public        0.143333   81.188333   5.194064
Vending                    0.052500   76.924444   4.907832

Kruskal-Wallis H Statistic: 8551.021378882862
P-value: 0.0
There are significant differences between groups.
```

[ ]:

```
The Kruskal-Wallis H test results in a significant p-value (p < 0.05),
suggesting that there are significant differences in the distribution of 'Request_Closing_Time' across different 'Complaint Types.'
In other words, the 'Request_Closing_Time' is not the same across all complaint types.

#7.1 Fail to reject H0: All sample distributions are equal
#7.2 Reject H0: One or more sample distributions are not equal

Reject H0: The p-value is very close to zero, so you reject the null hypothesis. Therefore, you do not "fail to reject" the null hypothesis. Instead, you reject it.
```