Customer Service Requests Analysis

By Kumar Anurag

Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Load the CSV data

```
# Define data types for specific columns
dtypes = {
    'Unique Key': 'int64',
    'Incident Zip': 'float64',
    'X Coordinate (State Plane)': 'float64',
    'Y Coordinate (State Plane)': 'float64',
    'School or Citywide Complaint': 'float64',
    'Vehicle Type': 'float64',
    'Taxi Company Borough': 'float64',
    'Taxi Pick Up Location': 'float64',
    'Garage Lot Name': 'float64',
    'Latitude': 'float64',
    'Longitude': 'float64'
}

# Load the CSV data into a DataFrame with specified data types and
low_memory set to False
df = pd.read_csv("311_Service_Requests_from_2010_to_Present.csv",
dtype=dtypes, low_memory=False)
df.head()
```

```
   Unique Key            Created Date    Closed Date Agency  \
0    32310363  12/31/2015 11:59:45 PM  01-01-16 0:55   NYPD
1    32309934  12/31/2015 11:59:44 PM  01-01-16 1:26   NYPD
2    32309159  12/31/2015 11:59:29 PM  01-01-16 4:51   NYPD
3    32305098  12/31/2015 11:57:46 PM  01-01-16 7:43   NYPD
4    32306529  12/31/2015 11:56:58 PM  01-01-16 3:24   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  ... Bridge Highway Name Bridge Highway
Direction  \
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

  Road Ramp Bridge Highway Segment Garage Lot Name Ferry Direction  \
0       NaN                     NaN              NaN              NaN
1       NaN                     NaN              NaN              NaN
2       NaN                     NaN              NaN              NaN
3       NaN                     NaN              NaN              NaN
4       NaN                     NaN              NaN              NaN

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

                                  Location
0   (40.86568153633767, -73.92350095571744)
1  (40.775945312321085, -73.91509393898605)
2  (40.870324522111424, -73.88852464418646)
3   (40.83599404683083, -73.82837939584206)
4  (40.733059618956815, -73.87416975810375)

[5 rows x 53 columns]
```

Data Exploration and Preprocessing

```
# Display basic information about the dataset
print(df.info())

# Display the first few rows of the dataset
print(df.head())
```

```python
# Check for missing values
print(df.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 53 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   Unique Key                      300698 non-null  int64  
 1   Created Date                    300698 non-null  object 
 2   Closed Date                     298534 non-null  object 
 3   Agency                          300698 non-null  object 
 4   Agency Name                     300698 non-null  object 
 5   Complaint Type                  300698 non-null  object 
 6   Descriptor                      294784 non-null  object 
 7   Location Type                   300567 non-null  object 
 8   Incident Zip                    298083 non-null  float64
 9   Incident Address                256288 non-null  object 
 10  Street Name                     256288 non-null  object 
 11  Cross Street 1                  251419 non-null  object 
 12  Cross Street 2                  250919 non-null  object 
 13  Intersection Street 1           43858 non-null   object 
 14  Intersection Street 2           43362 non-null   object 
 15  Address Type                    297883 non-null  object 
 16  City                            298084 non-null  object 
 17  Landmark                        349 non-null     object 
 18  Facility Type                   298527 non-null  object 
 19  Status                          300698 non-null  object 
 20  Due Date                        300695 non-null  object 
 21  Resolution Description          300698 non-null  object 
 22  Resolution Action Updated Date  298511 non-null  object 
 23  Community Board                 300698 non-null  object 
 24  Borough                         300698 non-null  object 
 25  X Coordinate (State Plane)      297158 non-null  float64
 26  Y Coordinate (State Plane)      297158 non-null  float64
 27  Park Facility Name              300698 non-null  object 
 28  Park Borough                    300698 non-null  object 
 29  School Name                     300698 non-null  object 
 30  School Number                   300698 non-null  object 
 31  School Region                   300697 non-null  object 
 32  School Code                     300697 non-null  object 
 33  School Phone Number             300698 non-null  object 
 34  School Address                  300698 non-null  object 
 35  School City                     300698 non-null  object 
 36  School State                    300698 non-null  object 
 37  School Zip                      300697 non-null  object 
 38  School Not Found                300698 non-null  object 
 39  School or Citywide Complaint    0 non-null       float64
 40  Vehicle Type                    0 non-null       float64
```

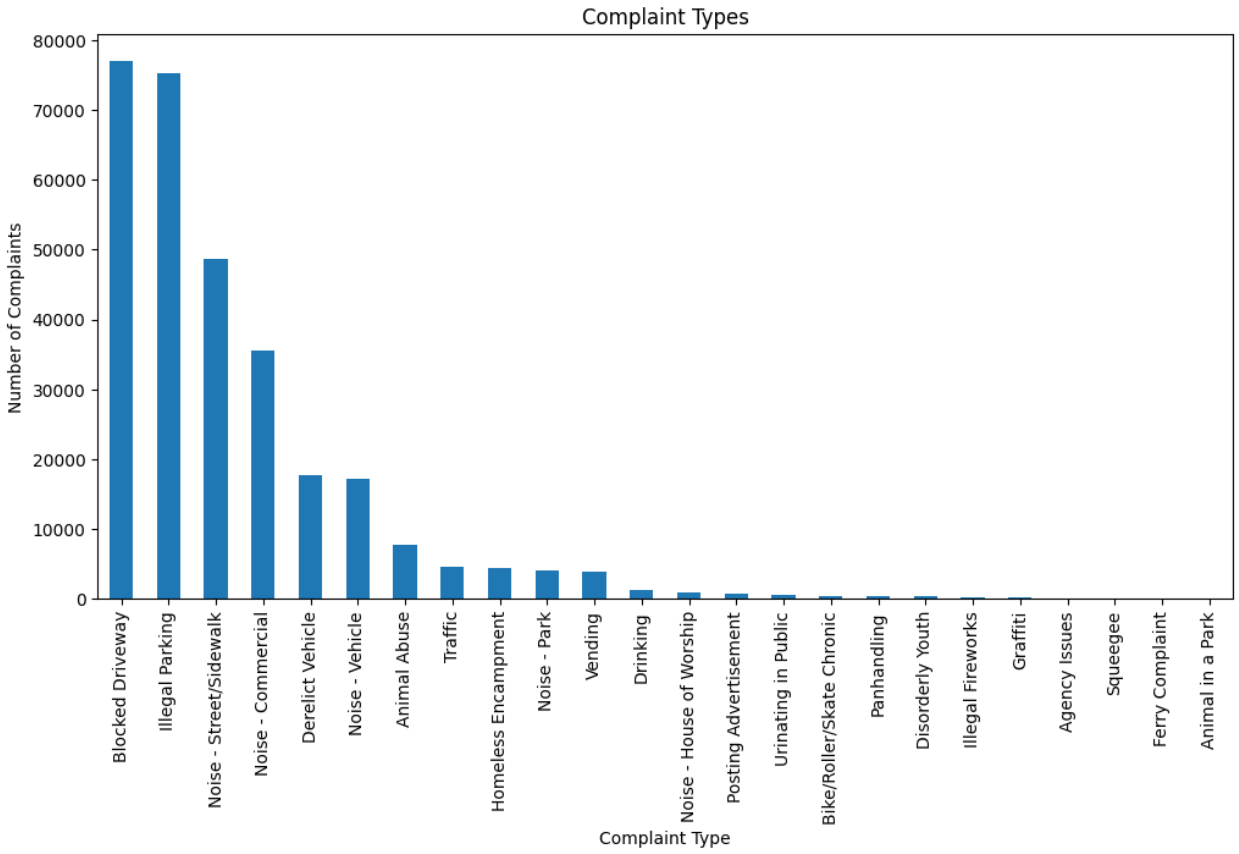
```
 41  Taxi Company Borough            0 non-null       float64
 42  Taxi Pick Up Location           0 non-null       float64
 43  Bridge Highway Name             243 non-null     object
 44  Bridge Highway Direction        243 non-null     object
 45  Road Ramp                       213 non-null     object
 46  Bridge Highway Segment          213 non-null     object
 47  Garage Lot Name                 0 non-null       float64
 48  Ferry Direction                 1 non-null       object
 49  Ferry Terminal Name             2 non-null       object
 50  Latitude                        297158 non-null  float64
 51  Longitude                       297158 non-null  float64
 52  Location                        297158 non-null  object
dtypes: float64(10), int64(1), object(42)
memory usage: 121.6+ MB
None
   Unique Key            Created Date    Closed Date Agency  \
0    32310363  12/31/2015 11:59:45 PM  01-01-16 0:55   NYPD
1    32309934  12/31/2015 11:59:44 PM  01-01-16 1:26   NYPD
2    32309159  12/31/2015 11:59:29 PM  01-01-16 4:51   NYPD
3    32305098  12/31/2015 11:57:46 PM  01-01-16 7:43   NYPD
4    32306529  12/31/2015 11:56:58 PM  01-01-16 3:24   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  ... Bridge Highway Name Bridge Highway
Direction  \
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

  Road Ramp Bridge Highway Segment Garage Lot Name Ferry Direction  \
```

```
0       NaN                       NaN             NaN             NaN
1       NaN                       NaN             NaN             NaN
2       NaN                       NaN             NaN             NaN
3       NaN                       NaN             NaN             NaN
4       NaN                       NaN             NaN             NaN

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

                                  Location
0   (40.86568153633767, -73.92350095571744)
1  (40.775945312321085, -73.91509393898605)
2  (40.870324522111424, -73.88852464418646)
3   (40.83599404683083, -73.82837939584206)
4  (40.733059618956815, -73.87416975810375)

[5 rows x 53 columns]
Unique Key                              0
Created Date                            0
Closed Date                          2164
Agency                                  0
Agency Name                             0
Complaint Type                          0
Descriptor                           5914
Location Type                         131
Incident Zip                         2615
Incident Address                    44410
Street Name                         44410
Cross Street 1                      49279
Cross Street 2                      49779
Intersection Street 1              256840
Intersection Street 2              257336
Address Type                         2815
City                                 2614
Landmark                           300349
Facility Type                        2171
Status                                  0
Due Date                                3
Resolution Description                  0
Resolution Action Updated Date       2187
Community Board                         0
Borough                                 0
X Coordinate (State Plane)           3540
Y Coordinate (State Plane)           3540
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    300698
Vehicle Type                    300698
Taxi Company Borough            300698
Taxi Pick Up Location           300698
Bridge Highway Name             300455
Bridge Highway Direction        300455
Road Ramp                       300485
Bridge Highway Segment          300485
Garage Lot Name                 300698
Ferry Direction                 300697
Ferry Terminal Name             300696
Latitude                          3540
Longitude                         3540
Location                          3540
dtype: int64
```

Data Analysis and Visualization

```
# Example: Visualize complaint types
complaint_counts = df['Complaint Type'].value_counts()
complaint_counts.plot(kind='bar', figsize=(12, 6))
plt.title("Complaint Types")
plt.xlabel("Complaint Type")
plt.ylabel("Number of Complaints")
plt.show()
```



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 and request closing. (Hint: Explore the package/module datetime)

```
# Define data types for specific columns
dtypes = {
    'Unique Key': 'int64',
    'Incident Zip': 'float64',
    'X Coordinate (State Plane)': 'float64',
    'Y Coordinate (State Plane)': 'float64',
    'School or Citywide Complaint': 'float64',
    'Vehicle Type': 'float64',
    'Taxi Company Borough': 'float64',
    'Taxi Pick Up Location': 'float64',
    'Garage Lot Name': 'float64',
    'Latitude': 'float64',
    'Longitude': 'float64'
}
```

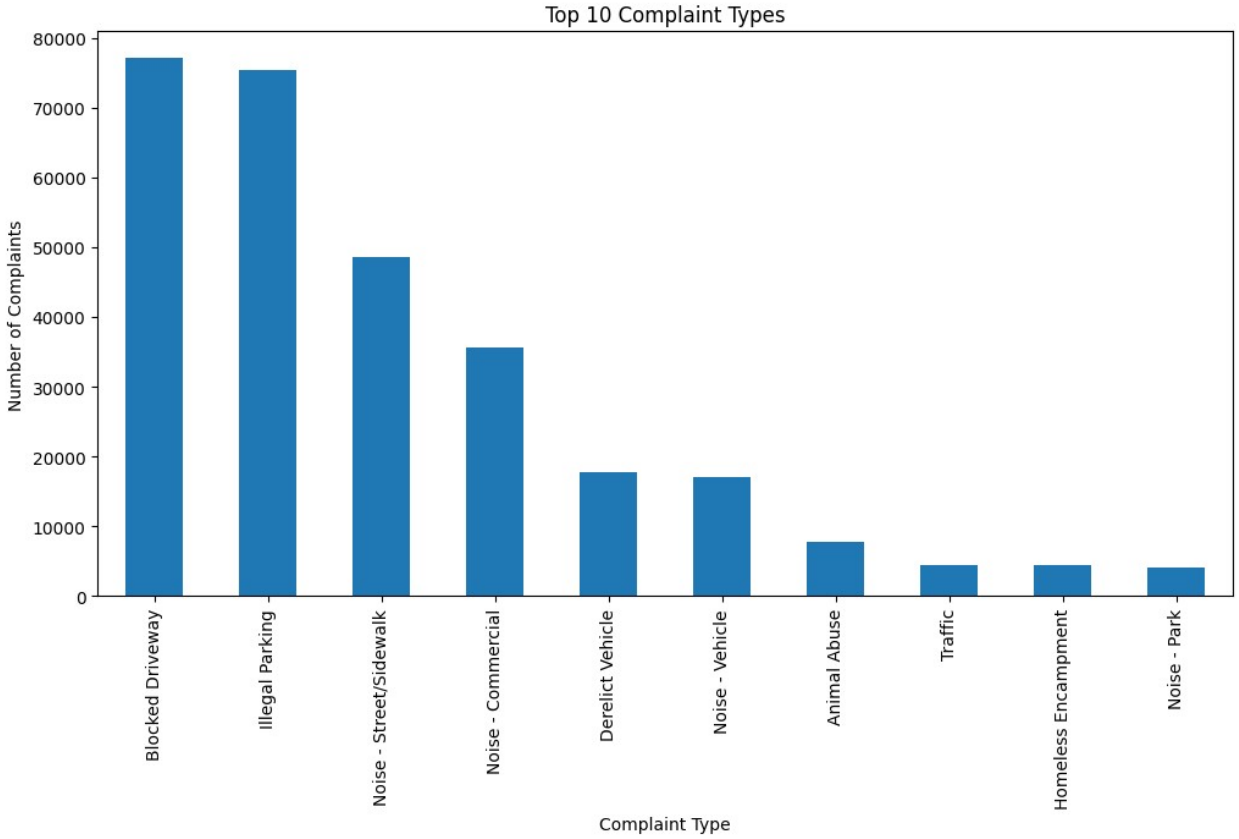
```
# Load the CSV data into a DataFrame with specified data types and
low_memory set to False
df = pd.read_csv("311_Service_Requests_from_2010_to_Present.csv",
dtype=dtypes, low_memory=False)

# Convert 'Created Date' to datetime data type with multiple formats
df['Created Date'] = pd.to_datetime(df['Created Date'],
format='%m/%d/%Y %I:%M:%S %p', errors='coerce')
df['Created Date'].combine_first(pd.to_datetime(df['Created Date'],
format='%m-%d-%y %H:%M', errors='coerce'))

# Do the same for 'Closed Date' if needed
df['Closed Date'] = pd.to_datetime(df['Closed Date'], format='%m/%d/%Y
%I:%M:%S %p', errors='coerce')
df['Closed Date'].combine_first(pd.to_datetime(df['Closed Date'],
format='%m-%d-%y %H:%M', errors='coerce'))



# Calculate the time elapsed between request creation and request
closing
df['Request_Closing_Time'] = df['Closed Date'] - df['Created Date']

# Display the first few rows of the DataFrame with the new
'Request_Closing_Time' column
print(df[['Created Date', 'Closed Date',
'Request_Closing_Time']].head())
```

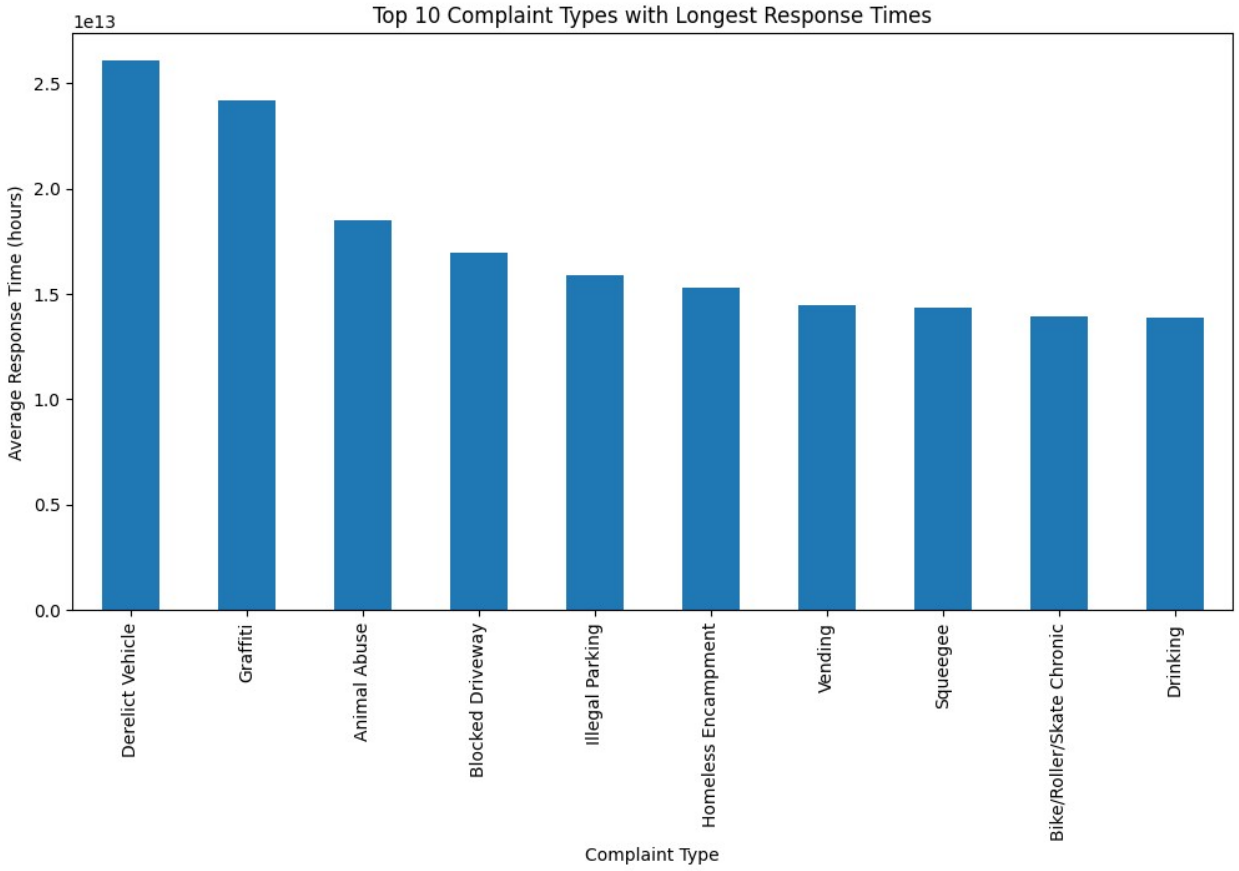
```
         Created Date Closed Date Request_Closing_Time
0 2015-12-31 23:59:45         NaT                  NaT
1 2015-12-31 23:59:44         NaT                  NaT
2 2015-12-31 23:59:29         NaT                  NaT
3 2015-12-31 23:57:46         NaT                  NaT
4 2015-12-31 23:56:58         NaT                  NaT
```

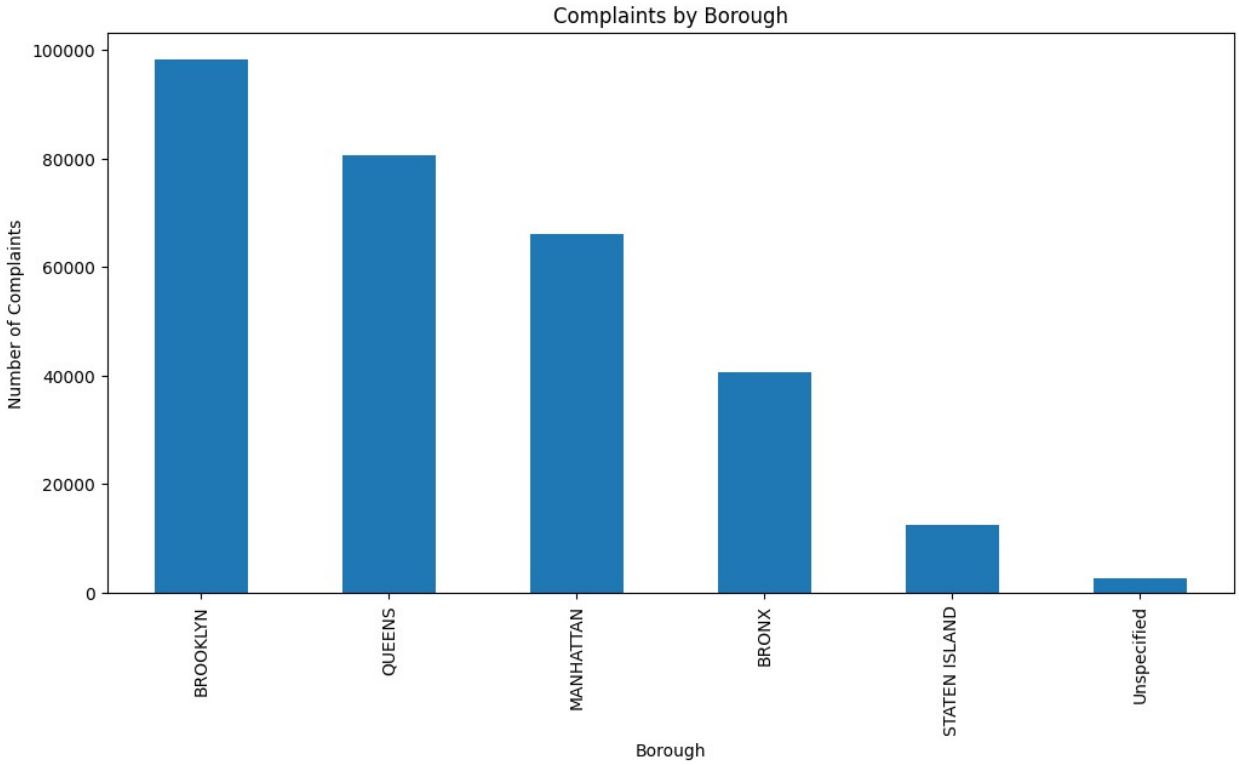
Provide major insights/patterns that you can offer in a visual format (graphs or tables); at least 4 major conclusions that you can come up with after generic data mining.

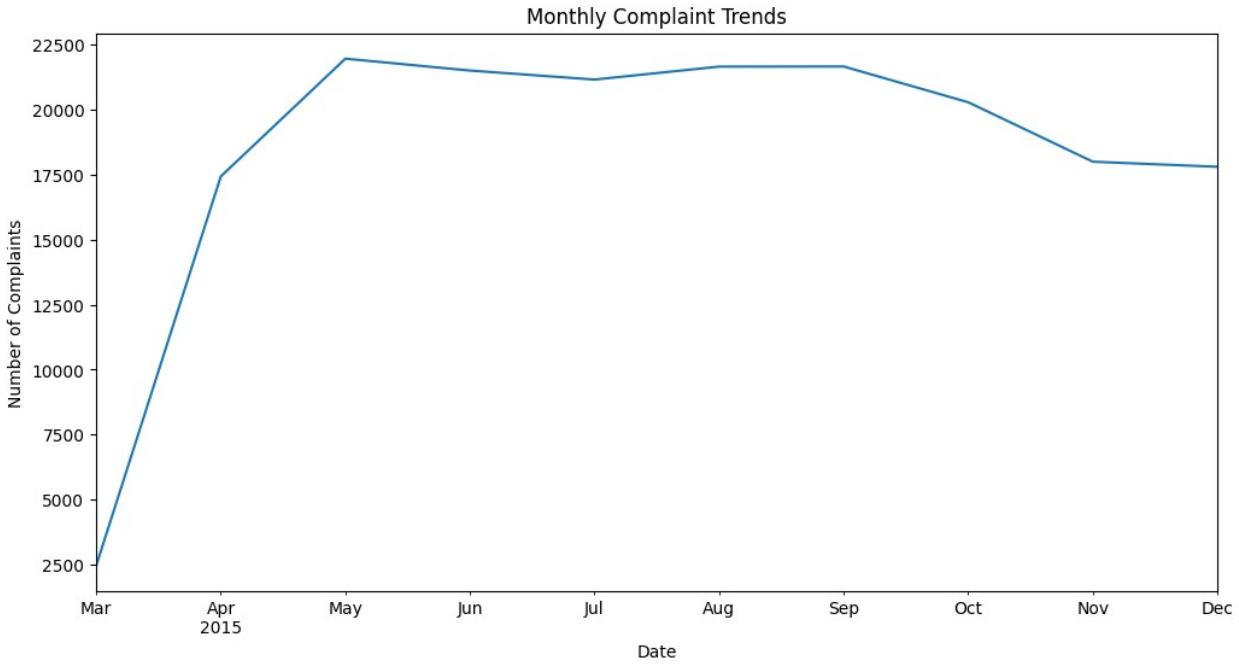
```
import matplotlib.pyplot as plt

# 1. Distribution of Complaint Types
complaint_counts = df['Complaint Type'].value_counts()
plt.figure(figsize=(12, 6))
complaint_counts.head(10).plot(kind='bar')
plt.title("Top 10 Complaint Types")
plt.xlabel("Complaint Type")
```

```
plt.ylabel("Number of Complaints")

# 2. Response Time Analysis
avg_response_time = df.groupby('Complaint Type')
['Request_Closing_Time'].mean()
avg_response_time.sort_values(ascending=False, inplace=True)
plt.figure(figsize=(12, 6))
avg_response_time.head(10).plot(kind='bar')
plt.title("Top 10 Complaint Types with Longest Response Times")
plt.xlabel("Complaint Type")
plt.ylabel("Average Response Time (hours)")

# 3. Borough-wise Complaint Analysis
borough_counts = df['Borough'].value_counts()
plt.figure(figsize=(12, 6))
borough_counts.plot(kind='bar')
plt.title("Complaints by Borough")
plt.xlabel("Borough")
plt.ylabel("Number of Complaints")

# 4. Temporal Trends
df['Created Date'] = pd.to_datetime(df['Created Date'])
complaints_by_month = df.resample('M', on='Created Date').size()
plt.figure(figsize=(12, 6))
complaints_by_month.plot()
plt.title("Monthly Complaint Trends")
plt.xlabel("Date")
plt.ylabel("Number of Complaints")

plt.show()
```


Top 10 Complaint Types
Number of Complaints
80000
70000
60000
50000
40000
30000
20000
10000
0
Blocked Driveway
Illegal Parking
Noise - Street/Sidewalk
Noise - Commercial
Derelict Vehicle
Noise - Vehicle
Animal Abuse
Traffic
Homeless Encampment
Noise - Park
Complaint Type


Top 10 Complaint Types with Longest Response Times
1e13
2.5
2.0
1.5
1.0
0.5
0.0
Average Response Time (hours)
Derelict Vehicle
Graffiti
Animal Abuse
Blocked Driveway
Illegal Parking
Homeless Encampment
Vending
Squeegee
Bike/Roller/Skate Chronic
Drinking
Complaint Type


Complaints by Borough
100000
80000
60000
40000
20000
0
Number of Complaints
BROOKLYN
QUEENS
MANHATTAN
BRONX
STATEN ISLAND
Unspecified
Borough



Order the complaint types based on the average 'Request_Closing_Time', grouping them for different locations

```
# Group the data by 'Borough' and 'Complaint Type' and calculate the
average 'Request_Closing_Time' for each group
complaints_grouped = df.groupby(['Borough', 'Complaint Type'])
['Request_Closing_Time'].mean()

# Reset the index to make it easier to work with
complaints_grouped = complaints_grouped.reset_index()

# Sort the data by the average 'Request_Closing_Time' in ascending
order
complaints_grouped =
complaints_grouped.sort_values(by='Request_Closing_Time')

# Display the result
print(complaints_grouped)
```

```
           Borough              Complaint Type
Request_Closing_Time
97   STATEN ISLAND      Posting Advertisement 0 days
01:12:57.407665505
104    Unspecified  Bike/Roller/Skate Chronic           0 days
01:39:12
69           QUEENS          Illegal Fireworks 0 days
01:49:54.363636363
```

```
48       MANHATTAN          Illegal Fireworks 0 days
01:57:03.866666666
89   STATEN ISLAND          Illegal Fireworks    0 days
02:14:16.500000
..             ...                        ...                        ..
.
108    Unspecified            Ferry Complaint
NaT
111    Unspecified   Noise - House of Worship
NaT
115    Unspecified                Panhandling
NaT
116    Unspecified      Posting Advertisement
NaT
117    Unspecified                    Traffic
NaT

[119 rows x 3 columns]
```

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

```
import scipy.stats as stats

# Filter out complaint types with no data for 'Request_Closing_Time'
data = [df[df['Complaint Type'] == complaint]
['Request_Closing_Time'].dropna() for complaint in df['Complaint
Type'].unique() if not df[df['Complaint Type'] == complaint]
['Request_Closing_Time'].dropna().empty]

# Perform the ANOVA test if there are complaint types with data
if len(data) > 1:
    f_statistic, p_value = stats.f_oneway(*data)

    # Set the significance level (alpha)
    alpha = 0.05

    # Check the p-value against the significance level to determine
whether to accept or reject the null hypothesis
    if p_value < alpha:
        print("Reject the null hypothesis. There is a significant
difference in average response time across complaint types.")
    else:
        print("Accept the null hypothesis. The average response time
is similar across complaint types.")

    # Display the p-value
    print("p-value:", p_value)
```

```
else:
    print("Insufficient data for ANOVA. Unable to perform the test.")
```

```
Reject the null hypothesis. There is a significant difference in
average response time across complaint types.
p-value: 0.0
```

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

```
import pandas as pd
from scipy.stats import chi2_contingency

# Create a cross-tabulation (contingency table) of 'Complaint Type'
and 'Borough'
contingency_table = pd.crosstab(df['Complaint Type'], df['Borough'])

# Perform the Chi-Square test of independence
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Set the significance level (alpha)
alpha = 0.05

# Check the p-value against the significance level to determine
whether to accept or reject the null hypothesis
if p < alpha:
    print("Reject the null hypothesis. There is a significant
relationship between complaint type and location.")
else:
    print("Accept the null hypothesis. Complaint type and location are
independent.")

# Display the p-value
print("p-value:", p)
```

```
Reject the null hypothesis. There is a significant relationship
between complaint type and location.
p-value: 0.0
```