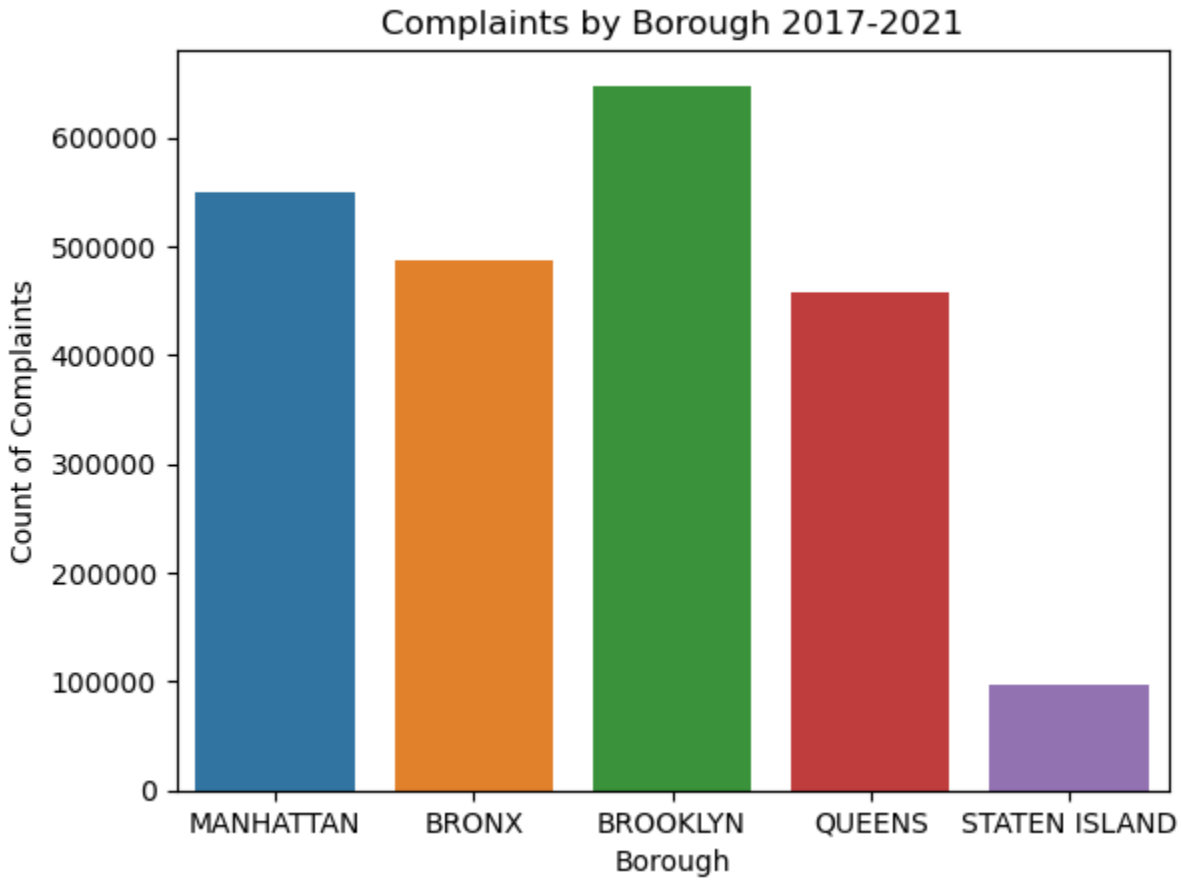
EDA Report

Fritz Grunert, Mason Lonoff, Sara Douglas, Susan Lu

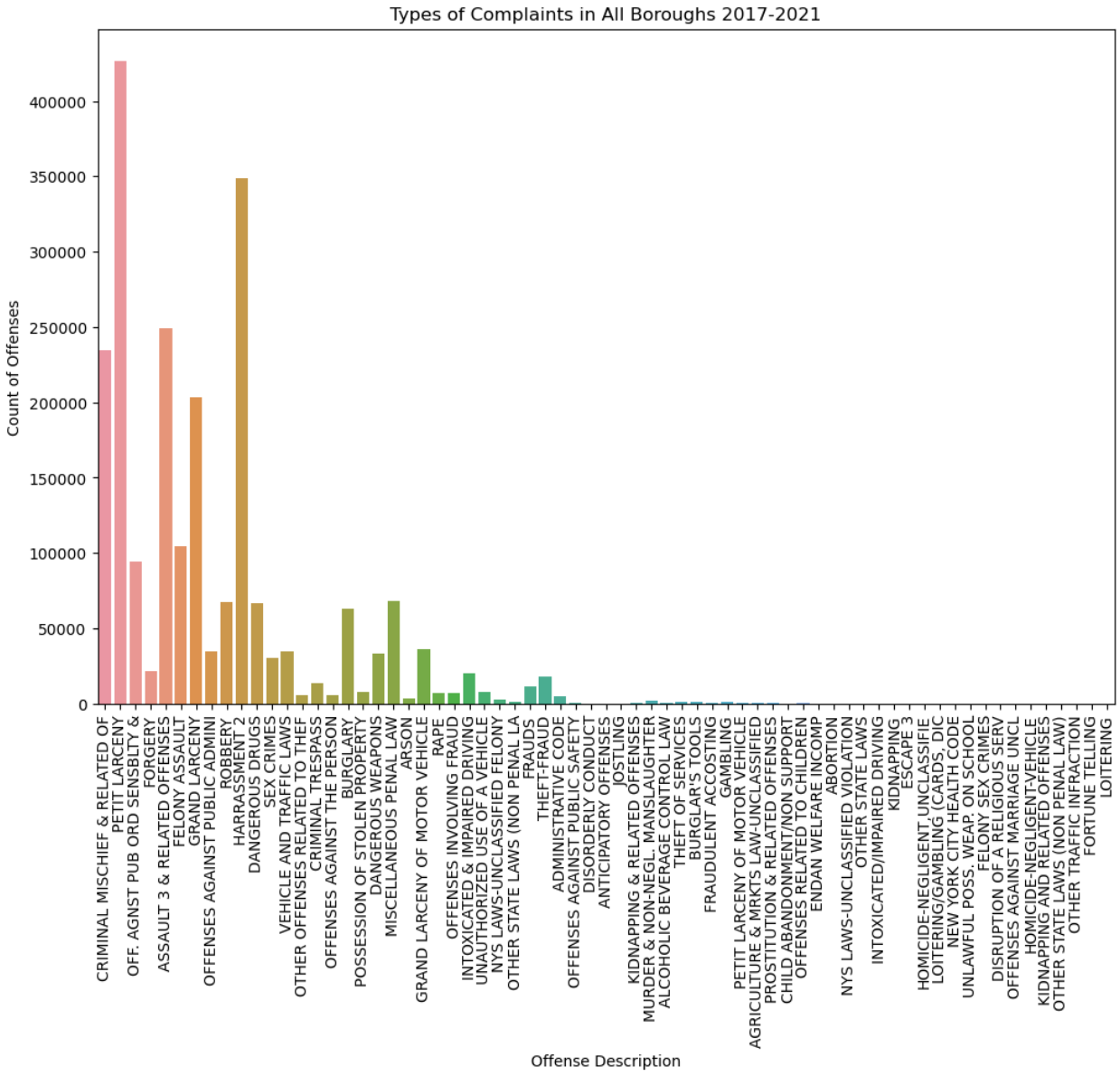
Exploring Complaints by Borough

```
sns.countplot(
    x = crime2['BORO_NM'],
    data = crime2
)
plt.title('Complaints by Borough 2017-2021')
plt.xlabel('Borough')
plt.ylabel('Count of Complaints')
plt.show()
```



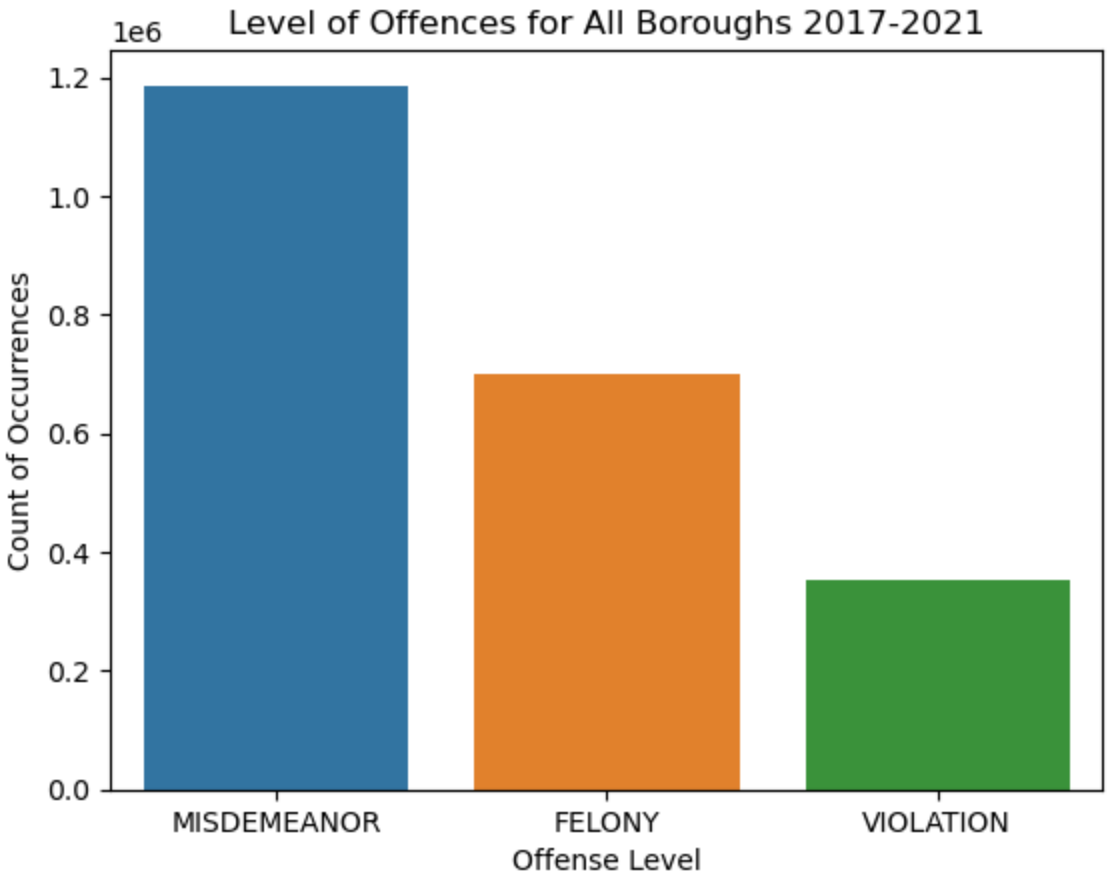
Exploring Types of Offences

```
plt.figure(figsize=(12,8))
sns.countplot(
    x = crime2['OFNS_DESC'],
    data = crime2
)
plt.xticks(rotation = 90)
plt.title('Types of Complaints in All Boroughs 2017-2021')
plt.xlabel('Offense Description')
plt.ylabel('Count of Offenses')
plt.show()
```



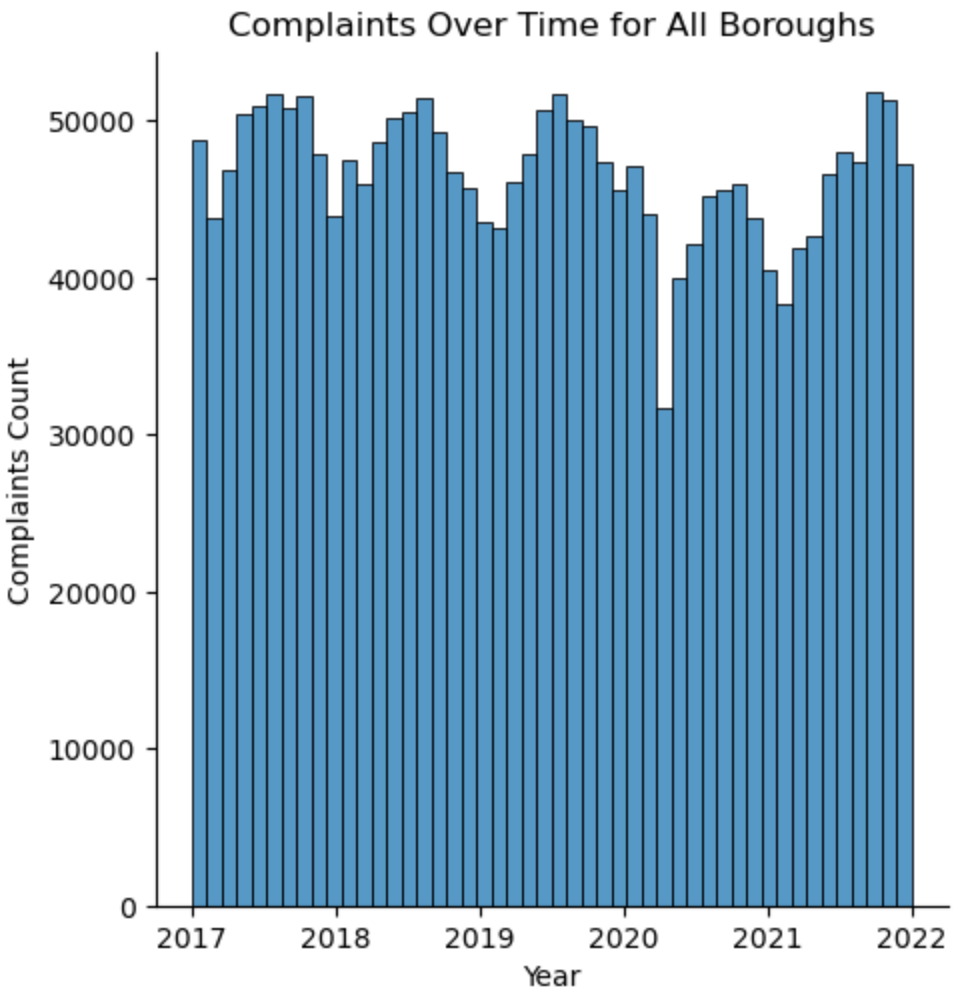
Exploring Levels of Offences

```
sns.countplot(
    x = crime2['LAW_CAT_CD'],
    data = crime2
)
plt.title('Level of Offences for All Boroughs 2017-2021')
plt.xlabel('Offense Level')
plt.ylabel('Count of Occurrences')
plt.show()
```



Exploring Complaints Across Years

```
sns.displot(
    data = crime2,
    x = crime2['DATE'],
    bins = 48
)
plt.title('Complaints Over Time for All Boroughs')
plt.xlabel('Year')
plt.ylabel('Complaints Count')
plt.show()
```



Null Counts for Crime Dataset

After removing nulls in date, longitude and latitude

CMPLNT_NUM	0
CMPLNT_FR_TM	0
CMPLNT_TO_DT	272607
CMPLNT_TO_TM	271641
ADDR_PCT_CD	0
RPT_DT	0
KY_CD	0
OFNS_DESC	57
PD_CD	1783
PD_DESC	1783
CRM_ATPT_CPTD_CD	161
LAW_CAT_CD	0

BORO_NM	2487
LOC_OF_OCCUR_DESC	397233
PREM_TYP_DESC	7941
JURIS_DESC	0
JURISDICTION_CODE	1783
PARKS_NM	2222620
HADEVELOPT	2170008
HOUSING_PSA	2073836
X_COORD_CD	0
Y_COORD_CD	0
SUSP_AGE_GROUP	529003
SUSP_RACE	529003
SUSP_SEX	529003
TRANSIT_DISTRICT	2188541
Latitude	0
Longitude	0
Lat_Lon	0
PATROL_BORO	1783
STATION_NAME	2188541
VIC_AGE_GROUP	5
VIC_RACE	86
VIC_SEX	5
DATE	0
DATE_AND_TIME	0

dtype: int64

 (easter egg!!!)

Alerts EDA

High level view of the data frame:

```
full_alerts.info()
```

```
#   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Alert_Code        85837 non-null  object
 1   DateTime          85837 non-null  datetime64[ns]
 2   Agency            85837 non-null  object
 3   Title             85837 non-null  object
 4   Message           85837 non-null  object
 5   Train_Line        85832 non-null  object
 6   Borough           85832 non-null  object
 7   Message_Category  85837 non-null  object
```

A count of the nulls in each column:

```
full_alerts.isnull().sum()
```

```
Alert_Code          0
DateTime            0
Agency              0
Title               0
Message             0
Train_Line          5
Borough             5
Message_Category    0
```

A view of the first and last 3 rows:

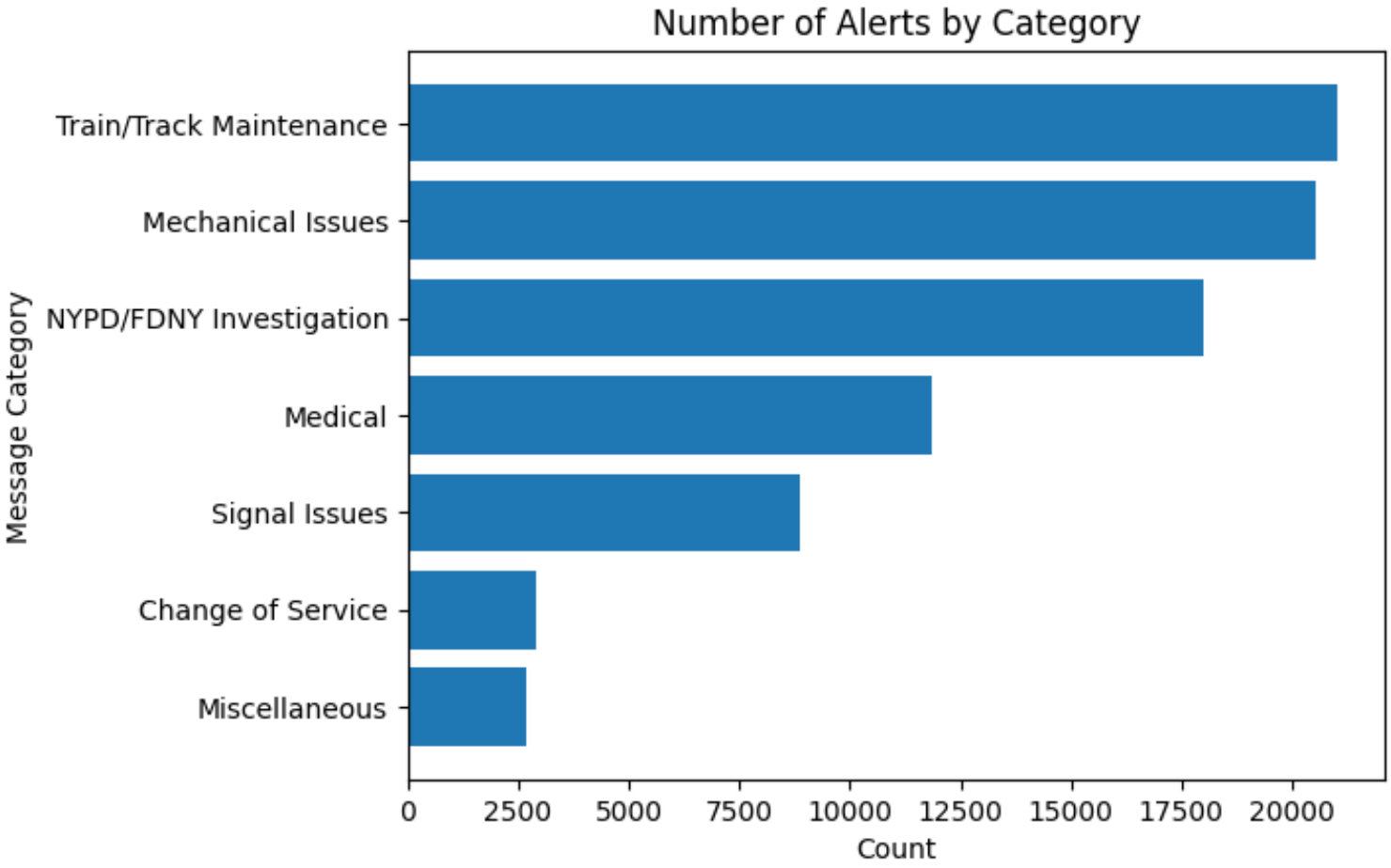
```
full_alerts.head(3)
```

	Alert_Code	DateTime	Agency	Title	Message	Train_Line	Borough	Message_Category
0	1172692	1/17/22 4:23 PM	NYC	MANH, 1 Train, Some Delays	1 trains are delayed entering/leaving South Fe...	1	MANH	Mechanical Issues
1	1172650	1/17/22 1:57 PM	NYC	MANH, 1 Train, Some Delays	1 trains may experience delays while entering ...	1	MANH	NYPD/FDNY Investigation
2	1172337	1/16/22 9:52 PM	NYC	MANH, 1 Train, Local to Express	Uptown 1 trains are running express from Times...	1	MANH	Medical

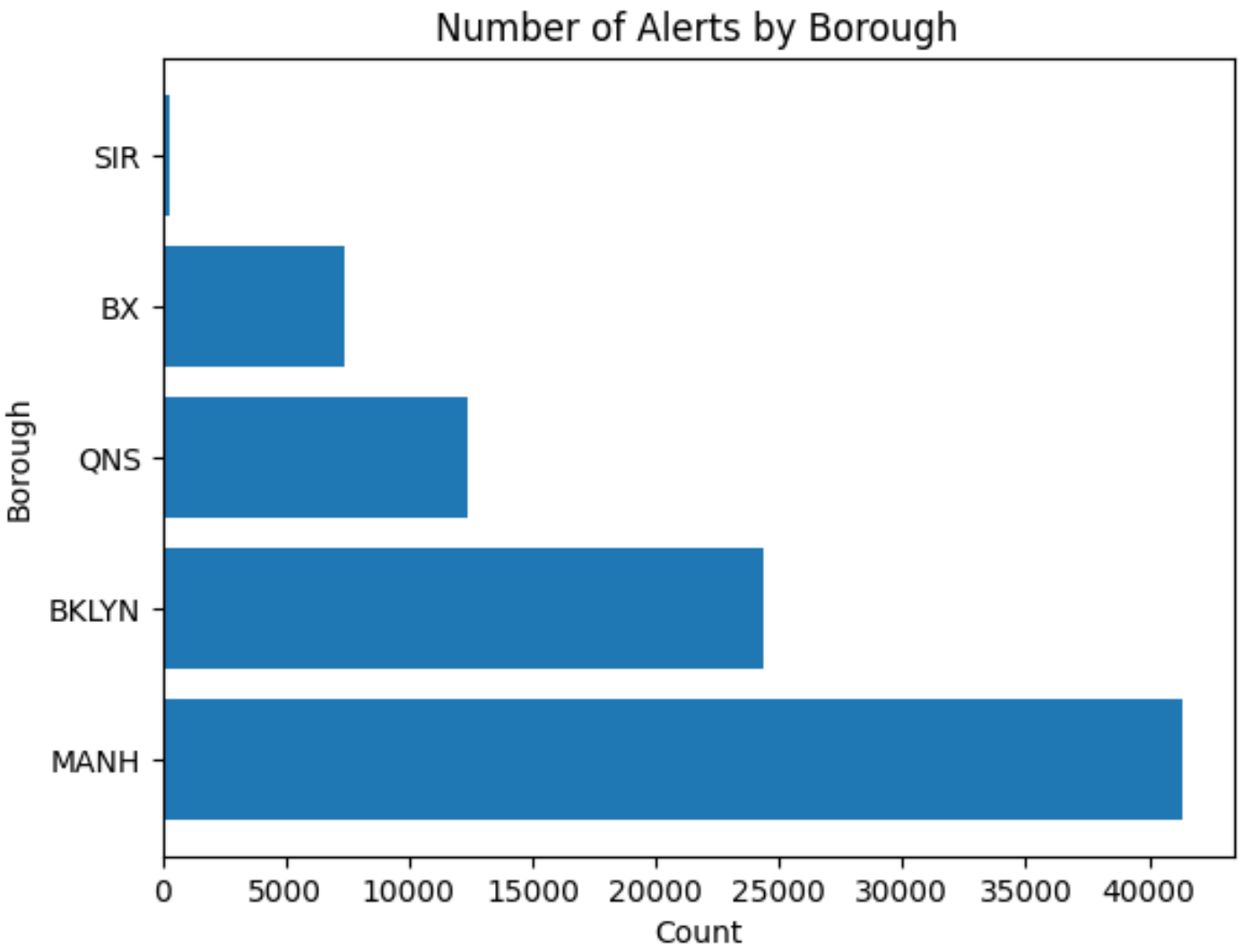
```
full_alerts.tail(3)
```

	Alert_Code	DateTime	Agency	Title	Message	Train_Line	Borough	Message_Category
86149	417474	4/20/17 7:30 AM	NYC	SI, SIR Trains, Sick passenger	Due to an earlier incident at Prince's Bay, SI...	SIR	SIR	Miscellaneous
86150	411342	4/1/17 12:34 PM	NYC	SI, SIR Trains, Injured Passenger	Due to an injured passenger at New Dorp, There...	SIR	SIR	Medical
86151	409578	3/27/17 7:53 AM	NYC	SI, SIR Trains, NYPD Activity	Due to NYPD activity at Great Kills, northboun...	SIR	SIR	NYPD/FDNY Investigation

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

category_counts = full_alerts['Message_Category'].value_counts()
category_counts = category_counts.sort_index()

# sort the data by counts in descending order
category_counts =
full_alerts['Message_Category'].value_counts().sort_values(ascending=True)

plt.barh(category_counts.index, category_counts.values)
plt.ylabel('Message Category')
plt.xlabel('Count')
plt.title('Number of Alerts by Category')
plt.show()
```



```
borough_counts = full_alerts['Borough'].value_counts()

plt.barh(borough_counts.index, borough_counts.values)
plt.ylabel('Borough')
plt.xlabel('Count')
plt.title('Number of Alerts by Borough')
plt.show()
```




```
import matplotlib.pyplot as plt


# Extract the year from the datetime object
full_alerts['Year'] = full_alerts['DateTime'].dt.year
```

```
# Filter the data to exclude the year 2022
filtered_alerts = full_alerts[full_alerts['Year'] != 2022]

# Group the data by year and count the number of complaints for each year
complaints_by_year =
filtered_alerts.groupby('Year').size().reset_index(name='Counts')


# Create a line chart
complaints_by_year.plot(x='Year', y='Counts')
```

```
# Set the x-axis labels to whole years
ax = plt.gca()
ax.xaxis.set_major_locator(plt.MaxNLocator(integer=True))

plt.title("Number of MTA Alerts by Year")

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