NYC crimes using Pyspark

In [79]:

```
# importing libraries:
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
%matplotlib inline
from pyspark.sql import SparkSession
import matplotlib.pyplot as plt
import seaborn as sns
from random import sample
from folium.plugins import HeatMap


# create a spark session app with the name of NYC_CRIME_ANALYSIS:
spark_session = SparkSession.builder.appName("NYC_crime_analysis").getOrCreate()
```

In [80]:

```
# read the crimes dataframe:
file = "/Users/karimaidrissi/Desktop/DSSA 5101/complaint.csv"
df = spark_session.read.csv(file, header = "True", inferSchema = "True")
```

In [81]:

```
# checking the columns and the data types:
df.columns
df.dtypes
```

Out[81]:

```
[('CMPLNT_NUM', 'int'),
 ('ADDR_PCT_CD', 'int'),
 ('BORO_NM', 'string'),
 ('CMPLNT_FR_DT', 'string'),
 ('CMPLNT_FR_TM', 'string'),
 ('JURIS_DESC', 'string'),
 ('KY_CD', 'int'),
 ('LAW_CAT_CD', 'string'),
 ('LOC_OF_OCCUR_DESC', 'string'),
 ('OFNS_DESC', 'string'),
 ('PATROL_BORO', 'string'),
 ('PD_CD', 'int'),
 ('PD_DESC', 'string'),
 ('PREM_TYP_DESC', 'string'),
 ('RPT_DT', 'string'),
 ('SUSP_AGE_GROUP', 'string'),
 ('SUSP_RACE', 'string'),
 ('SUSP_SEX', 'string'),
 ('VIC_AGE_GROUP', 'string'),
 ('VIC_RACE', 'string'),
 ('VIC_SEX', 'string'),
 ('X_COORD_CD', 'int'),
 ('Y_COORD_CD', 'int'),
 ('Latitude', 'double'),
 ('Longitude', 'double'),
 ('Lat_Lon', 'string')]
```

In [82]:

```
# counting how many records in the data:
df.count()
```

Out[82]:

```
222398
```

In [83]:

```
# printing the schema:
df.printSchema()
```

```
root
 |-- CMPLNT_NUM: integer (nullable = true)
 |-- ADDR_PCT_CD: integer (nullable = true)
 |-- BORO_NM: string (nullable = true)
 |-- CMPLNT_FR_DT: string (nullable = true)
 |-- CMPLNT_FR_TM: string (nullable = true)
 |-- JURIS_DESC: string (nullable = true)
 |-- KY_CD: integer (nullable = true)
 |-- LAW_CAT_CD: string (nullable = true)
 |-- LOC_OF_OCCUR_DESC: string (nullable = true)
 |-- OFNS_DESC: string (nullable = true)
 |-- PATROL_BORO: string (nullable = true)
 |-- PD_CD: integer (nullable = true)
 |-- PD_DESC: string (nullable = true)
 |-- PREM_TYP_DESC: string (nullable = true)
 |-- RPT_DT: string (nullable = true)
 |-- SUSP_AGE_GROUP: string (nullable = true)
 |-- SUSP_RACE: string (nullable = true)
 |-- SUSP_SEX: string (nullable = true)
 |-- VIC_AGE_GROUP: string (nullable = true)
 |-- VIC_RACE: string (nullable = true)
 |-- VIC_SEX: string (nullable = true)
 |-- X_COORD_CD: integer (nullable = true)
 |-- Y_COORD_CD: integer (nullable = true)
 |-- Latitude: double (nullable = true)
 |-- Longitude: double (nullable = true)
 |-- Lat_Lon: string (nullable = true)
```

In [84]:

```
# summary of the data:
df.describe().show()
```

```
+-------+-------------------+-----------------+-------------+----------
--+------------+----------------+------------------+----------+--------
---------+--------------------+--------------------+------------------+
--------------------+------------------+------+------------------+-----
---------------+--------+-----------------+--------------------+-------
+------------------+------------------+-------------------+------------
-------+--------------------+
|summary|         CMPLNT_NUM|      ADDR_PCT_CD|      BORO_NM|CMPLNT_FR_
DT|CMPLNT_FR_TM|      JURIS_DESC|             KY_CD|LAW_CAT_CD|LOC_OF_O
CCUR_DESC|           OFNS_DESC|         PATROL_BORO|             PD_CD|
PD_DESC|     PREM_TYP_DESC|RPT_DT|    SUSP_AGE_GROUP|           SUSP_RA
CE|SUSP_SEX|    VIC_AGE_GROUP|            VIC_RACE|VIC_SEX|        X_CO
ORD_CD|        Y_COORD_CD|           Latitude|          Longitude|
Lat_Lon|
+-------+-------------------+-----------------+-------------+----------
--+------------+----------------+------------------+----------+--------
---------+--------------------+--------------------+------------------+
--------------------+------------------+------+------------------+-----
---------------+--------+-----------------+--------------------+-------
+------------------+------------------+-------------------+------------
-------+--------------------+
|  count|             222398|           222393|       222260|      2223
98|      222398|          222398|            222398|    222398|
182366|              222389|              222261|            222261|
222261|            221461|222398|            170016|             17001
6|  170016|           222398|              222398| 222398|           2
22376|            222376|             222376|             222376|
222376|
|   mean|5.507290972476372E8|62.58275215496891|         null|        nu
ll|        null|            null|308.10353510373295|      null|
null|                null|                null|  404.278798349688|
null|              null|  null|             718.4|                null|
null|40.94117647058823|                null|   null|1005117.9980078785|
207800.42674569198| 40.736998631524784| -73.92467272973211|
null|
| stddev|2.602162294006241E8|34.63591142332019|         null|        nu
ll|        null|            null|155.90499971083656|      null|
null|                null|                null|220.49448351145233|
null|              null|  null|1094.9854402163137|                null|
null|856.0620940232836|                null|   null| 21161.08783726843|
30137.47531887374|0.08272394890111402|0.07631667308134345|
null|
|    min|          100001492|                1|        BRONX|  01/13/10
19|     0:00:00|         AMTRACK|               101|    FELONY|
FRONT OF| ADMINISTRATIVE CODE|PATROL BORO BKLYN...|               100|
A.B.C.,FALSE PROO...|ABANDONED BUILDING|1/1/19|                -1|AMERI
CAN INDIAN/A...|       F|               -2|AMERICAN INDIAN/A...|      D
|            913512|            121282|         40.4993236|         -7
4.254377|(40.4993236, -74....|
|    max|          999994546|              123|STATEN ISLAND|      9/9/
18|     9:59:00|U.S. PARK POLICE|               881| VIOLATION|
REAR OF|VEHICLE AND TRAFF...|PATROL BORO STATE...|               969| W
OUNDS,REPORTING OF|       VIDEO STORE|6/9/19|           UNKNOWN|      W
HITE HISPANIC|       U|          UNKNOWN|      WHITE HISPANIC|      M|
1067185|            271820|         40.9127234|       -73.70072029|(40.
912723396, -7...|
+-------+-------------------+-----------------+-------------+----------
```

```
--+------------+----------------+------------------+----------+--------
---------+--------------------+--------------------+------------------+
--------------------+------------------+------+------------------+-----
---------------+--------+-----------------+--------------------+-------
+------------------+------------------+-------------------+------------
-------+--------------------+
```

In [85]:

```
# calculate the statistics of some columns:

df.select(["X_COORD_CD", "Y_COORD_CD", "latitude", "longitude"]).describ
e().show()
```

```
+-------+------------------+------------------+-------------------+----
---------------+
|summary|        X_COORD_CD|        Y_COORD_CD|           latitude|
longitude|
+-------+------------------+------------------+-------------------+----
---------------+
|  count|            222376|            222376|             222376|
222376|
|   mean|1005117.9980078785|207800.42674569198| 40.736998631524784| -7
3.92467272973211|
| stddev| 21161.08783726843| 30137.47531887374|0.08272394890111402|0.07
631667308134345|
|    min|            913512|            121282|         40.4993236|
-74.254377|
|    max|           1067185|            271820|         40.9127234|
-73.70072029|
+-------+------------------+------------------+-------------------+----
---------------+
```

In [86]:

```
#  drop some columns:
df1 = df.drop("ADDR_PCT_CD", "LOC_OF_OCCUR_DESC", "PD_DESC","RPT_DT","PA
TROL_BORO","PD_CD","RDT_DT","X_COORD_CD","Y_COORD_CD", "lat_lon")
```

In [87]:

```
# convert Spark data into Pandas DataFrame.
df1.toPandas()
```

Out[87]:

	CMPLNT_NUM	BORO_NM	CMPLNT_FR_DT	CMPLNT_FR_TM	JURIS_DESC	KY_CD	
0	857927015	MANHATTAN	1/29/19	16:37:00	N.Y. POLICE DEPT	106	
1	479254687	QUEENS	3/29/19	17:00:00	N.Y. POLICE DEPT	107	
2	320007604	BRONX	2/6/19	2:00:00	N.Y. POLICE DEPT	105	
3	746022144	BROOKLYN	1/8/19	22:49:00	N.Y. POLICE DEPT	117	
4	593941718	BRONX	3/17/19	5:00:00	N.Y. POLICE DEPT	344	M
5	613547550	MANHATTAN	2/22/19	13:35:00	N.Y. POLICE DEPT	578	
6	585652917	QUEENS	2/1/19	10:00:00	N.Y. POLICE DEPT	578	
7	407860526	QUEENS	1/27/19	4:00:00	N.Y. POLICE DEPT	105	
8	145366108	MANHATTAN	2/11/19	12:07:00	N.Y. STATE POLICE	236	M
9	746680655	STATEN ISLAND	3/23/19	20:06:00	N.Y. POLICE DEPT	341	M
10	513320708	BROOKLYN	1/14/19	17:35:00	N.Y. HOUSING POLICE	106	
11	821304454	QUEENS	11/26/18	15:01:00	N.Y. POLICE DEPT	125	
12	827038864	MANHATTAN	3/19/19	20:00:00	N.Y. POLICE DEPT	344	M
13	889702556	MANHATTAN	3/11/19	21:40:00	N.Y. POLICE DEPT	105	
14	291569019	BRONX	2/16/19	17:15:00	N.Y. POLICE DEPT	344	M
15	336865313	BRONX	2/19/19	19:20:00	N.Y. POLICE DEPT	361	M
16	671142050	BRONX	2/8/19	6:00:00	N.Y. POLICE DEPT	121	
17	883793011	QUEENS	1/12/09	0:01:00	N.Y. POLICE DEPT	116	
18	944638748	QUEENS	2/23/19	22:30:00	N.Y. POLICE DEPT	344	M
19	758966473	BRONX	2/10/19	16:00:00	N.Y. HOUSING POLICE	344	M

	CMPLNT_NUM	BORO_NM	CMPLNT_FR_DT	CMPLNT_FR_TM	JURIS_DESC	KY_CD	
20	272821005	BROOKLYN	3/16/19	23:30:00	N.Y. POLICE DEPT	341	M
21	836139486	BRONX	1/31/19	16:00:00	N.Y. POLICE DEPT	341	M
22	877424333	MANHATTAN	11/11/18	12:00:00	N.Y. POLICE DEPT	361	M
23	804396233	MANHATTAN	12/19/18	8:00:00	N.Y. POLICE DEPT	341	M
24	811464670	BROOKLYN	2/9/19	20:36:00	N.Y. POLICE DEPT	111	
25	605345964	BRONX	3/14/19	18:20:00	N.Y. POLICE DEPT	118	
26	816741975	MANHATTAN	3/30/19	14:58:00	N.Y. POLICE DEPT	109	
27	403194332	QUEENS	3/17/19	15:53:00	N.Y. POLICE DEPT	113	
28	195765193	QUEENS	3/5/19	16:30:00	N.Y. POLICE DEPT	344	M
29	375209270	BRONX	2/27/19	5:38:00	N.Y. HOUSING POLICE	114	
...	...	...	...	...	...	...	
222368	871799835	QUEENS	5/13/19	11:44:00	N.Y. POLICE DEPT	351	M
222369	386832109	MANHATTAN	5/22/19	18:22:00	N.Y. POLICE DEPT	341	M
222370	526795281	BROOKLYN	5/4/19	17:45:00	N.Y. HOUSING POLICE	578	
222371	829042094	BROOKLYN	6/5/19	21:45:00	N.Y. POLICE DEPT	578	
222372	698629525	BROOKLYN	5/3/19	15:00:00	N.Y. POLICE DEPT	107	
222373	935708403	QUEENS	6/11/19	13:50:00	N.Y. POLICE DEPT	341	M
222374	283202420	BRONX	6/4/19	22:00:00	N.Y. POLICE DEPT	361	M
222375	998128516	BROOKLYN	4/30/19	14:30:00	N.Y. POLICE DEPT	578	
222376	383359978	MANHATTAN	6/17/19	9:00:00	N.Y. POLICE DEPT	341	M
222377	301752049	BROOKLYN	4/8/19	21:30:00	N.Y. HOUSING POLICE	351	M

	CMPLNT_NUM	BORO_NM	CMPLNT_FR_DT	CMPLNT_FR_TM	JURIS_DESC	KY_CD	
222378	664586224	QUEENS	5/12/19	19:50:00	N.Y. POLICE DEPT	344	M
222379	142105031	MANHATTAN	6/15/19	23:45:00	N.Y. POLICE DEPT	341	M
222380	121833520	MANHATTAN	2/22/19	9:00:00	N.Y. POLICE DEPT	109	
222381	713294929	QUEENS	5/2/19	16:00:00	N.Y. POLICE DEPT	341	M
222382	762039568	MANHATTAN	5/25/19	12:20:00	N.Y. POLICE DEPT	109	
222383	915876963	BROOKLYN	5/21/19	23:30:00	N.Y. POLICE DEPT	578	
222384	115756814	BROOKLYN	4/21/19	22:00:00	N.Y. POLICE DEPT	110	
222385	158258939	BRONX	6/2/19	1:00:00	N.Y. POLICE DEPT	236	M
222386	322958106	BROOKLYN	5/1/19	21:35:00	N.Y. POLICE DEPT	341	M
222387	339650129	MANHATTAN	6/5/19	8:00:00	N.Y. TRANSIT POLICE	578	
222388	729155081	MANHATTAN	6/27/19	21:30:00	N.Y. POLICE DEPT	105	
222389	138938099	MANHATTAN	6/19/19	20:35:00	N.Y. POLICE DEPT	126	
222390	443989732	BRONX	4/1/19	15:00:00	N.Y. POLICE DEPT	578	
222391	426822007	BROOKLYN	5/16/19	6:00:00	N.Y. POLICE DEPT	351	M
222392	824778502	QUEENS	4/12/19	17:30:00	N.Y. HOUSING POLICE	344	M
222393	587294745	BROOKLYN	4/15/19	12:00:00	N.Y. POLICE DEPT	109	
222394	326362764	BROOKLYN	6/22/19	13:25:00	N.Y. POLICE DEPT	126	
222395	992657534	MANHATTAN	6/17/19	20:10:00	N.Y. TRANSIT POLICE	106	
222396	577523166	QUEENS	6/7/19	9:28:00	N.Y. POLICE DEPT	340	M
222397	956145385	MANHATTAN	5/2/19	18:30:00	N.Y. TRANSIT POLICE	230	M

222398 rows × 17 columns

In [88]:

```
# Renaming the columns:
new_names = ("ID","Borough","Date","Time","Jurisdiction","Code","Level o
f offense", "Offense", "Premise" , "Suspicious age", "Suspicious race",
"Suspicious sex", "Victim age", "Victim race", "Victim sex","Latitude",
"Longitude")
df2 = df1.toDF(*new_names)
```

In [89]:

```
# showing only df2 columns:
df2.columns
```

Out[89]:

```
['ID',
 'Borough',
 'Date',
 'Time',
 'Jurisdiction',
 'Code',
 'Level of offense',
 'Offense',
 'Premise',
 'Suspicious age',
 'Suspicious race',
 'Suspicious sex',
 'Victim age',
 'Victim race',
 'Victim sex',
 'Latitude',
 'Longitude']
```

In [90]:

```
# counting how many Boroughs in the data :
df2.select("Borough").distinct().count()
```

Out[90]:

```
6
```

In [91]:

```
# counting how many offenses:
df2.select("Offense").distinct().count()
```

Out[91]:

```
62
```

In [92]:

```python
# showing only 20 offenses in our data
df2.select("Offense").distinct().show(n =20)
```

```
+--------------------+
|             Offense|
+--------------------+
|OTHER TRAFFIC INF...|
|ANTICIPATORY OFFE...|
|   FELONY SEX CRIMES|
|OTHER OFFENSES RE...|
|VEHICLE AND TRAFF...|
|KIDNAPPING & RELA...|
|HOMICIDE-NEGLIGEN...|
|OFF. AGNST PUB OR...|
|PETIT LARCENY OF ...|
|      FELONY ASSAULT|
|ALCOHOLIC BEVERAG...|
|OFFENSES RELATED ...|
|CRIMINAL MISCHIEF...|
|         THEFT-FRAUD|
|                null|
|   THEFT OF SERVICES|
|           JOSTLING|
|MISCELLANEOUS PEN...|
|LOITERING/GAMBLIN...|
|               ARSON|
+--------------------+
only showing top 20 rows
```

In [93]:

```python
# counting how many Harrassment 2 in the dataset:
df2.where(df2["Offense"] == "HARRASSMENT 2").count()
```

Out[93]:

```
35048
```

In [94]:

```python
df2.columns
```

Out[94]:

```
['ID',
 'Borough',
 'Date',
 'Time',
 'Jurisdiction',
 'Code',
 'Level of offense',
 'Offense',
 'Premise',
 'Suspicious age',
 'Suspicious race',
 'Suspicious sex',
 'Victim age',
 'Victim race',
 'Victim sex',
 'Latitude',
 'Longitude']
```

In [95]:

```
# filter Date bw 1/1/19 and 6/31/19:
#d = df2[(df2['Date'] >= '01/01/19') & (df2['Date'] <= '06/31/19')]
#d.show()
```

In [96]:

```
# split the date column into year, month and day :
split_date = F.split(df2['Date'], "/")

df2 = df2.withColumn('Year', split_date.getItem(2))
df2 = df2.withColumn('Month', split_date.getItem(0))
df2 = df2.withColumn('Day', split_date.getItem(1))
```

In [97]:

```
df2.columns
```

Out[97]:

```
['ID',
 'Borough',
 'Date',
 'Time',
 'Jurisdiction',
 'Code',
 'Level of offense',
 'Offense',
 'Premise',
 'Suspicious age',
 'Suspicious race',
 'Suspicious sex',
 'Victim age',
 'Victim race',
 'Victim sex',
 'Latitude',
 'Longitude',
 'Year',
 'Month',
 'Day']
```

```
In [98]: df2.select('Year', 'Month', 'Day').show()
```

```
+----+-----+---+
|Year|Month|Day|
+----+-----+---+
|  19|    1| 29|
|  19|    3| 29|
|  19|    2|  6|
|  19|    1|  8|
|  19|    3| 17|
|  19|    2| 22|
|  19|    2|  1|
|  19|    1| 27|
|  19|    2| 11|
|  19|    3| 23|
|  19|    1| 14|
|  18|   11| 26|
|  19|    3| 19|
|  19|    3| 11|
|  19|    2| 16|
|  19|    2| 19|
|  19|    2|  8|
|  09|    1| 12|
|  19|    2| 23|
|  19|    2| 10|
+----+-----+---+
only showing top 20 rows
```

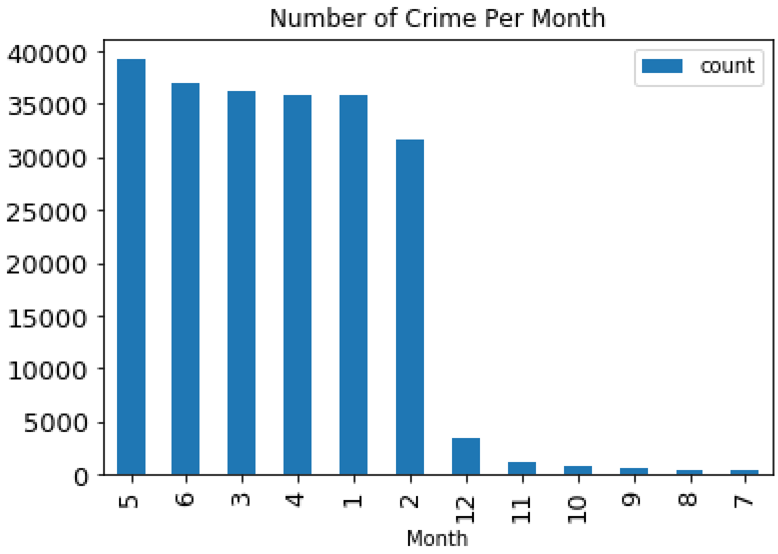
```
In [99]: # counting total number of crimes per month:
count_month =df2.groupBy(['Month']).count().filter("`count`>3").sort('count', ascending = False).toPandas()
count_month
```

Out[99]:

	Month	count
0	5	39271
1	6	36916
2	3	36154
3	4	35914
4	1	35874
5	2	31723
6	12	3432
7	11	1096
8	10	744
9	9	528
10	8	373
11	7	364

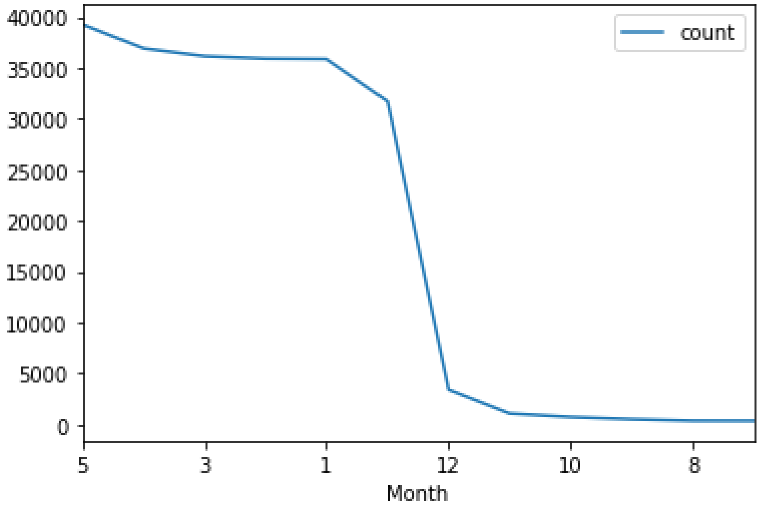
```
In [100]: # total number of crimes for each month using bar graph:
          count_month.plot(kind = "bar" , x = "Month", y ="count", fontsize =13, t
          itle = "Number of Crime Per Month")


          # May is the highest crime month during the first 6 month of 2019
```

```
Out[100]: <matplotlib.axes._subplots.AxesSubplot at 0x1a237196d8>
```



```
In [101]: # line graph of number of crimes for each month :
          count_month.plot(kind = "line" , x = "Month", y ="count")
```

```
Out[101]: <matplotlib.axes._subplots.AxesSubplot at 0x1a226d78d0>
```



In [102]:

```
# counting which days have the highest crimes number:
count_day = df2.groupBy('Day').count().sort("Day", ascending = True).toPandas()
count_day
```

Out[102]:

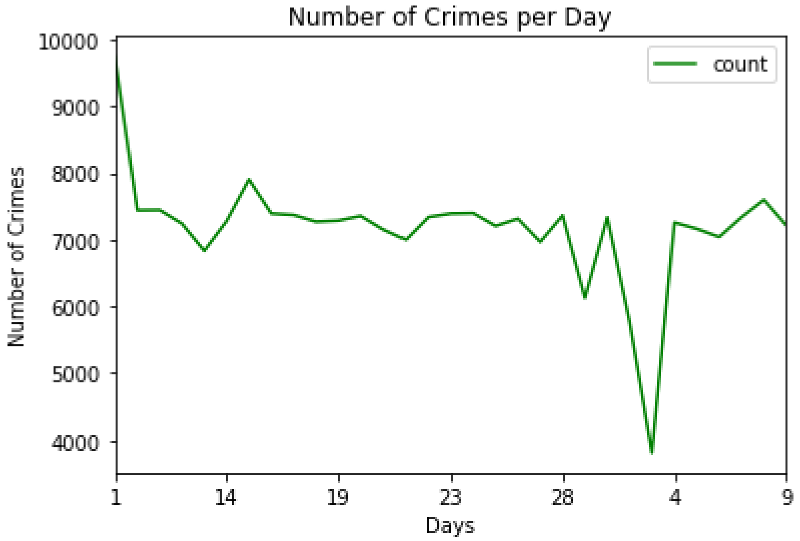
	Day	count
0	1	9754
1	10	7444
2	11	7447
3	12	7242
4	13	6834
5	14	7281
6	15	7905
7	16	7393
8	17	7372
9	18	7271
10	19	7287
11	2	7357
12	20	7154
13	21	7001
14	22	7341
15	23	7393
16	24	7399
17	25	7207
18	26	7316
19	27	6968
20	28	7365
21	29	6127
22	3	7337
23	30	5770
24	31	3814
25	4	7259
26	5	7164
27	6	7042
28	7	7340
29	8	7600
30	9	7214

In [103]:

```
count_day.plot(kind = "line", x = "Day",y = "count", color = "green")
plt.xlabel("Days")
plt.ylabel("Number of Crimes")
plt.title("Number of Crimes per Day")
```

Out[103]:

```
Text(0.5, 1.0, 'Number of Crimes per Day')
```



In [104]:

```
# We can use SQL query to interact with spark DataFrame:
# first we will make a temporary table called Crimes

df2.createOrReplaceTempView("Crimes")
avg_month = spark_session.sql("""
    SELECT Month, COUNT(*) AS Total
    FROM Crimes
    WHERE Month BETWEEN '1' AND '6'
    GROUP BY Month
    ORDER BY Total DESC
""")
avg_month = avg_month.toPandas()
avg_month
```

Out[104]:

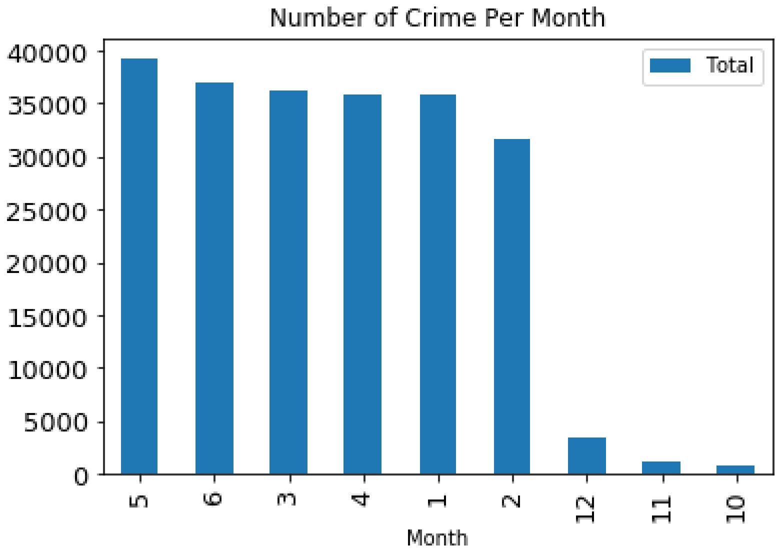
	Month	Total
0	5	39271
1	6	36916
2	3	36154
3	4	35914
4	1	35874
5	2	31723
6	12	3432
7	11	1096
8	10	744

In [105]:

```
# Crimes Count for each month by using sparkSQL:
avg_month.plot(kind = "bar" , x = "Month", y ="Total", fontsize =13, title = "Number of Crime Per Month")
```

Out[105]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1a567128>
```



In [106]:

```
# Counting the crimes for each Borough:

boro_count=df2.groupBy(["Borough"]).count().sort("count", ascending = False).toPandas()
boro_count
```

Out[106]:

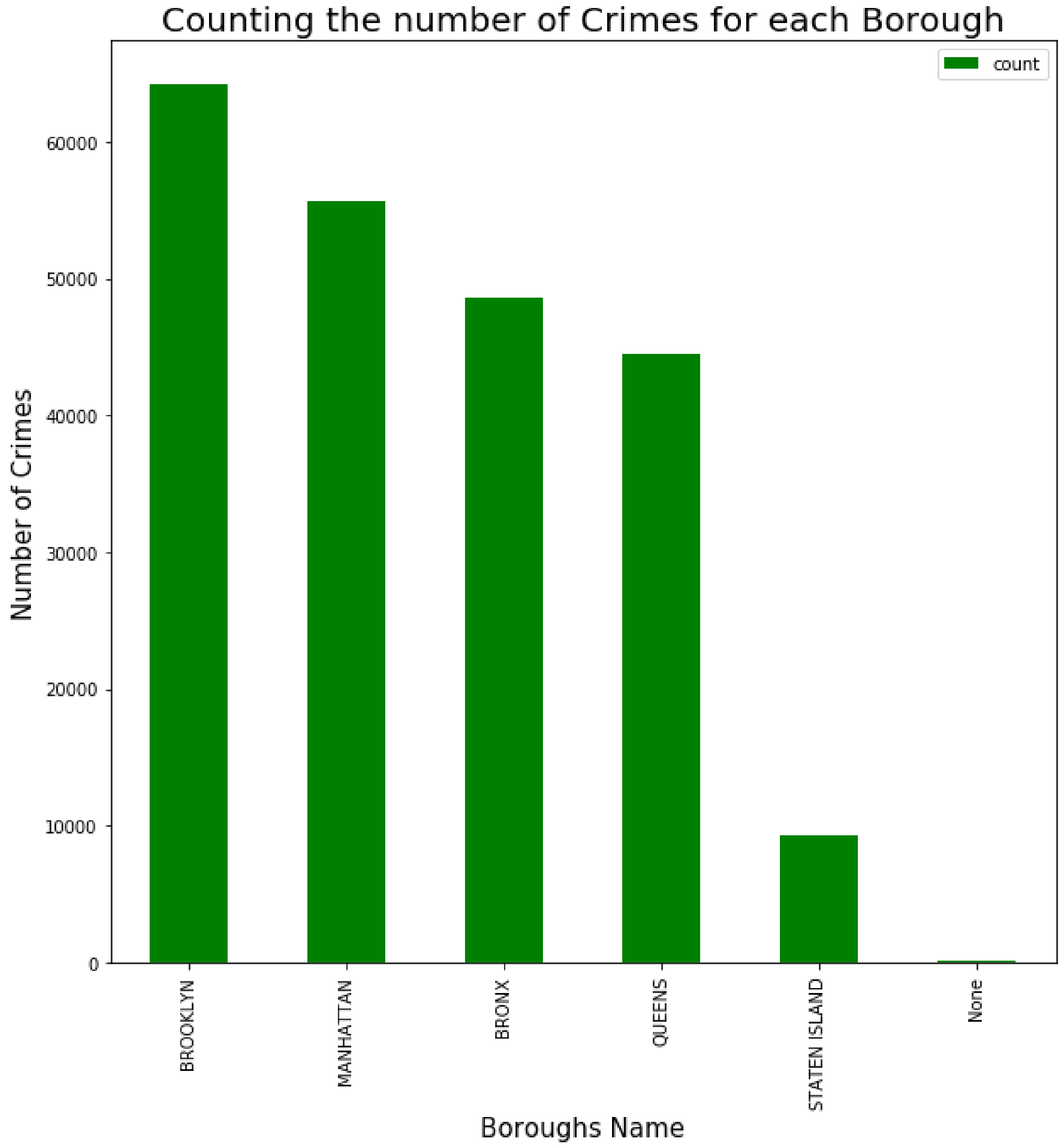
	Borough	count
0	BROOKLYN	64160
1	MANHATTAN	55666
2	BRONX	48605
3	QUEENS	44506
4	STATEN ISLAND	9323
5	None	138

In [107]:

```
# plottinh bar graph counting the number of crimes per Borough:
boro_count.plot(kind="bar", x ="Borough",color = "green", y = "count", figsize =(10,10))

plt.xlabel("Boroughs Name", fontsize = 15)
plt.ylabel("Number of Crimes" , fontsize = 15)
plt.title("Counting the number of Crimes for each Borough", fontsize = 20)


#BROKLYN has the most crime complaints during 2019, followed by Manhattan
```

Out[107]:

```
Text(0.5, 1.0, 'Counting the number of Crimes for each Borough')
```



In [108]:

```
# counting the level of offense per Borough:

off_boro = df2.groupBy(["Borough", "Level of offense"]).count().toPandas
()
off_boro
```

Out[108]:

	Borough	Level of offense	count
0	MANHATTAN	FELONY	16876
1	MANHATTAN	MISDEMEANOR	31359
2	STATEN ISLAND	FELONY	2257
3	QUEENS	MISDEMEANOR	23332
4	None	FELONY	138
5	MANHATTAN	VIOLATION	7431
6	BRONX	MISDEMEANOR	27094
7	BRONX	FELONY	13226
8	BROOKLYN	FELONY	20907
9	BROOKLYN	VIOLATION	10074
10	QUEENS	FELONY	13693
11	STATEN ISLAND	MISDEMEANOR	4949
12	QUEENS	VIOLATION	7481
13	BRONX	VIOLATION	8285
14	BROOKLYN	MISDEMEANOR	33179
15	STATEN ISLAND	VIOLATION	2117

In [109]:

```python
# counting the offenses and filter the count:
offense= df2.groupBy(["offense"]).count().filter("`count`>10").sort('count', ascending = True).toPandas()
offense
```

Out[109]:

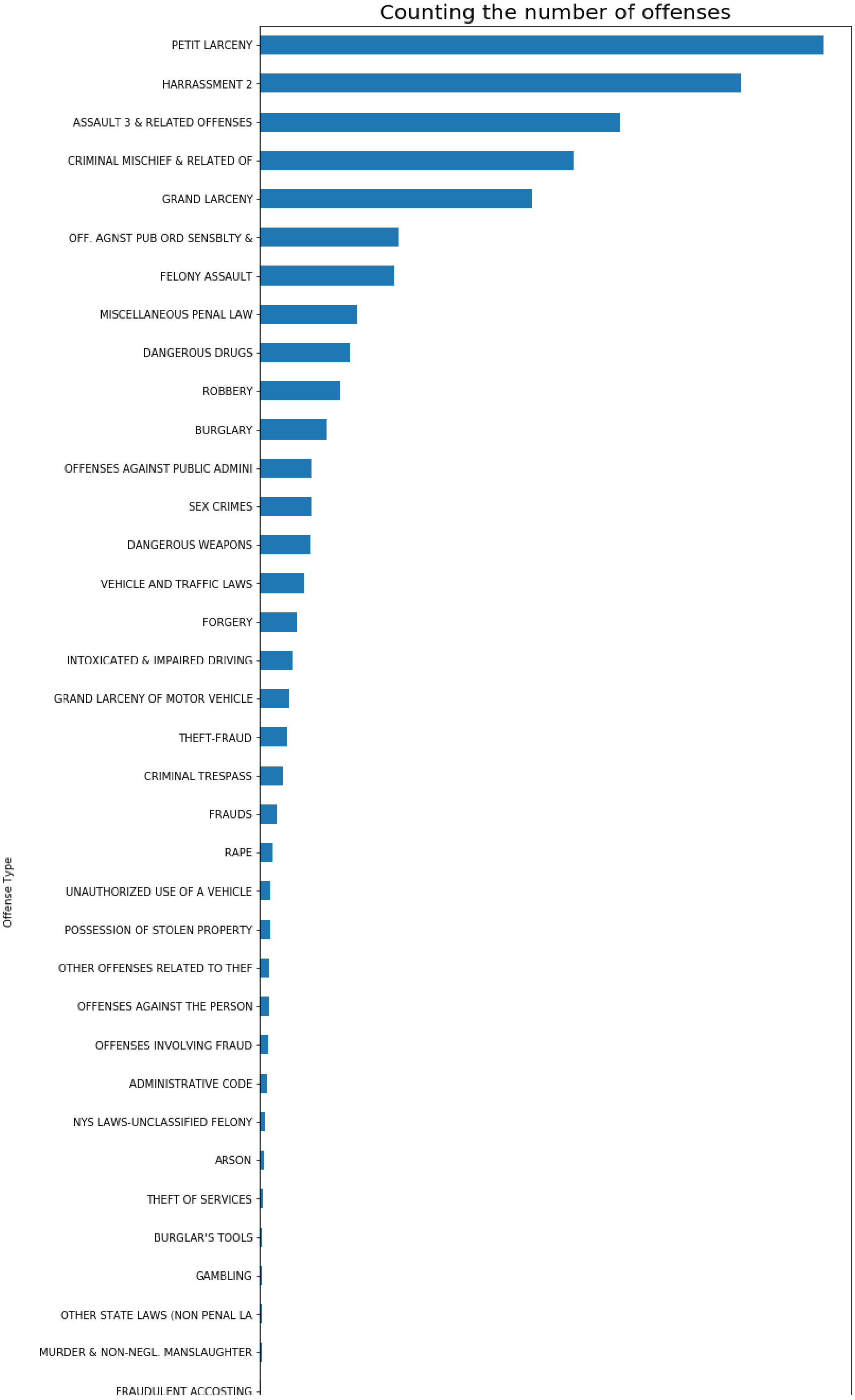
	offense	count
0	JOSTLING	13
1	DISORDERLY CONDUCT	19
2	OFFENSES AGAINST PUBLIC SAFETY	21
3	PETIT LARCENY OF MOTOR VEHICLE	34
4	PROSTITUTION & RELATED OFFENSES	34
5	OFFENSES RELATED TO CHILDREN	36
6	KIDNAPPING & RELATED OFFENSES	58
7	ALCOHOLIC BEVERAGE CONTROL LAW	60
8	AGRICULTURE & MRKTS LAW-UNCLASSIFIED	61
9	FRAUDULENT ACCOSTING	78
10	MURDER & NON-NEGL. MANSLAUGHTER	137
11	OTHER STATE LAWS (NON PENAL LA	139
12	GAMBLING	143
13	BURGLAR'S TOOLS	158
14	THEFT OF SERVICES	192
15	ARSON	318
16	NYS LAWS-UNCLASSIFIED FELONY	378
17	ADMINISTRATIVE CODE	501
18	OFFENSES INVOLVING FRAUD	598
19	OFFENSES AGAINST THE PERSON	654
20	OTHER OFFENSES RELATED TO THEF	679
21	POSSESSION OF STOLEN PROPERTY	730
22	UNAUTHORIZED USE OF A VEHICLE	738
23	RAPE	874
24	FRAUDS	1229
25	CRIMINAL TRESPASS	1646
26	THEFT-FRAUD	1992
27	GRAND LARCENY OF MOTOR VEHICLE	2157
28	INTOXICATED & IMPAIRED DRIVING	2350
29	FORGERY	2703
30	VEHICLE AND TRAFFIC LAWS	3212
31	DANGEROUS WEAPONS	3669
32	SEX CRIMES	3769
33	OFFENSES AGAINST PUBLIC ADMINI	3796

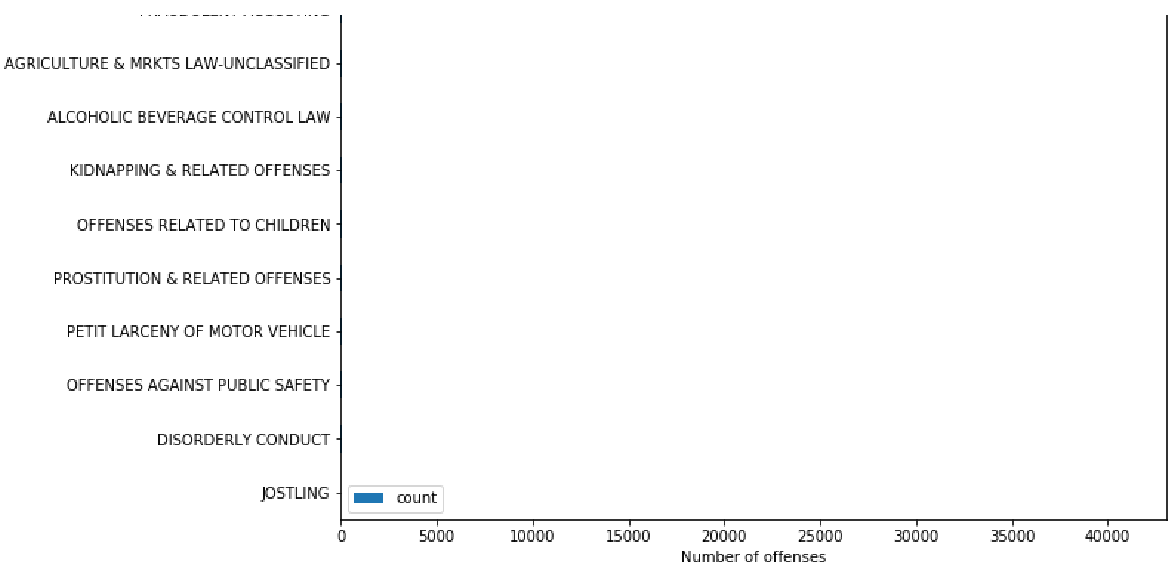
	offense	count
34	BURGLARY	4824
35	ROBBERY	5824
36	DANGEROUS DRUGS	6568
37	MISCELLANEOUS PENAL LAW	7085
38	FELONY ASSAULT	9800
39	OFF. AGNST PUB ORD SENSBLTY &	10099
40	GRAND LARCENY	19823
41	CRIMINAL MISCHIEF & RELATED OF	22801
42	ASSAULT 3 & RELATED OFFENSES	26246
43	HARRASSMENT 2	35048
44	PETIT LARCENY	41039

In [110]:

```
# plotting the number of crimes per offenses type:
offense.plot(kind = "barh", x = "offense", y = "count", figsize = (10,30
))
plt.xlabel("Number of offenses", fontsize = 10)
plt.ylabel("Offense Type", fontsize = 10)
plt.title("Counting the number of offenses", fontsize = 20)


# PETIT LARCENY is the most frequent offense during 2019.
```

```
Out[110]: Text(0.5, 1.0, 'Counting the number of offenses')
```


Counting the number of offenses
Offense Type
PETIT LARCENY
HARRASSMENT 2
ASSAULT 3 & RELATED OFFENSES
CRIMINAL MISCHIEF & RELATED OF
GRAND LARCENY
OFF. AGNST PUB ORD SENSBLTY &
FELONY ASSAULT
MISCELLANEOUS PENAL LAW
DANGEROUS DRUGS
ROBBERY
BURGLARY
OFFENSES AGAINST PUBLIC ADMINI
SEX CRIMES
DANGEROUS WEAPONS
VEHICLE AND TRAFFIC LAWS
FORGERY
INTOXICATED & IMPAIRED DRIVING
GRAND LARCENY OF MOTOR VEHICLE
THEFT-FRAUD
CRIMINAL TRESPASS
FRAUDS
RAPE
UNAUTHORIZED USE OF A VEHICLE
POSSESSION OF STOLEN PROPERTY
OTHER OFFENSES RELATED TO THEF
OFFENSES AGAINST THE PERSON
OFFENSES INVOLVING FRAUD
ADMINISTRATIVE CODE
NYS LAWS-UNCLASSIFIED FELONY
ARSON
THEFT OF SERVICES
BURGLAR'S TOOLS
GAMBLING
OTHER STATE LAWS (NON PENAL LA
MURDER & NON-NEGL. MANSLAUGHTER
FRAUDULENT ACCOSTING


AGRICULTURE & MRKTS LAW-UNCLASSIFIED
ALCOHOLIC BEVERAGE CONTROL LAW
KIDNAPPING & RELATED OFFENSES
OFFENSES RELATED TO CHILDREN
PROSTITUTION & RELATED OFFENSES
PETIT LARCENY OF MOTOR VEHICLE
OFFENSES AGAINST PUBLIC SAFETY
DISORDERLY CONDUCT
JOSTLING
count
0
5000
10000
15000
20000
25000
30000
35000
40000
Number of offenses

In [111]:

```
# counting the number of crimes by premises :
premises = df2.groupBy("premise"). count().sort( 'count', ascending = True).toPandas()
premises
```

Out[111]:

	premise	count
0	DAYCARE FACILITY	2
1	TRAMWAY	4
2	LOAN COMPANY	11
3	PHOTO/COPY	20
4	CEMETERY	22
5	VIDEO STORE	27
6	TAXI/LIVERY (UNLICENSED)	35
7	OTHER HOUSE OF WORSHIP	35
8	ABANDONED BUILDING	57
9	MOSQUE	65
10	MARINA/PIER	75
11	FERRY/FERRY TERMINAL	78
12	SYNAGOGUE	91
13	TUNNEL	103
14	BOOK/CARD	108
15	MAILBOX INSIDE	110
16	BUS (OTHER)	112
17	ATM	128
18	JEWELRY	139
19	SOCIAL CLUB/POLICY	149
20	SHOE	175
21	BUS STOP	178
22	TRANSIT FACILITY (OTHER)	181
23	FACTORY/WAREHOUSE	182
24	BUS TERMINAL	183
25	HOMELESS SHELTER	187
26	TAXI (YELLOW LICENSED)	196
27	STORAGE FACILITY	223
28	CHECK CASHING BUSINESS	226
29	BRIDGE	228
...	...	...
45	SMALL MERCHANT	668
46	STORE UNCLASSIFIED	767
47	GYM/FITNESS FACILITY	795

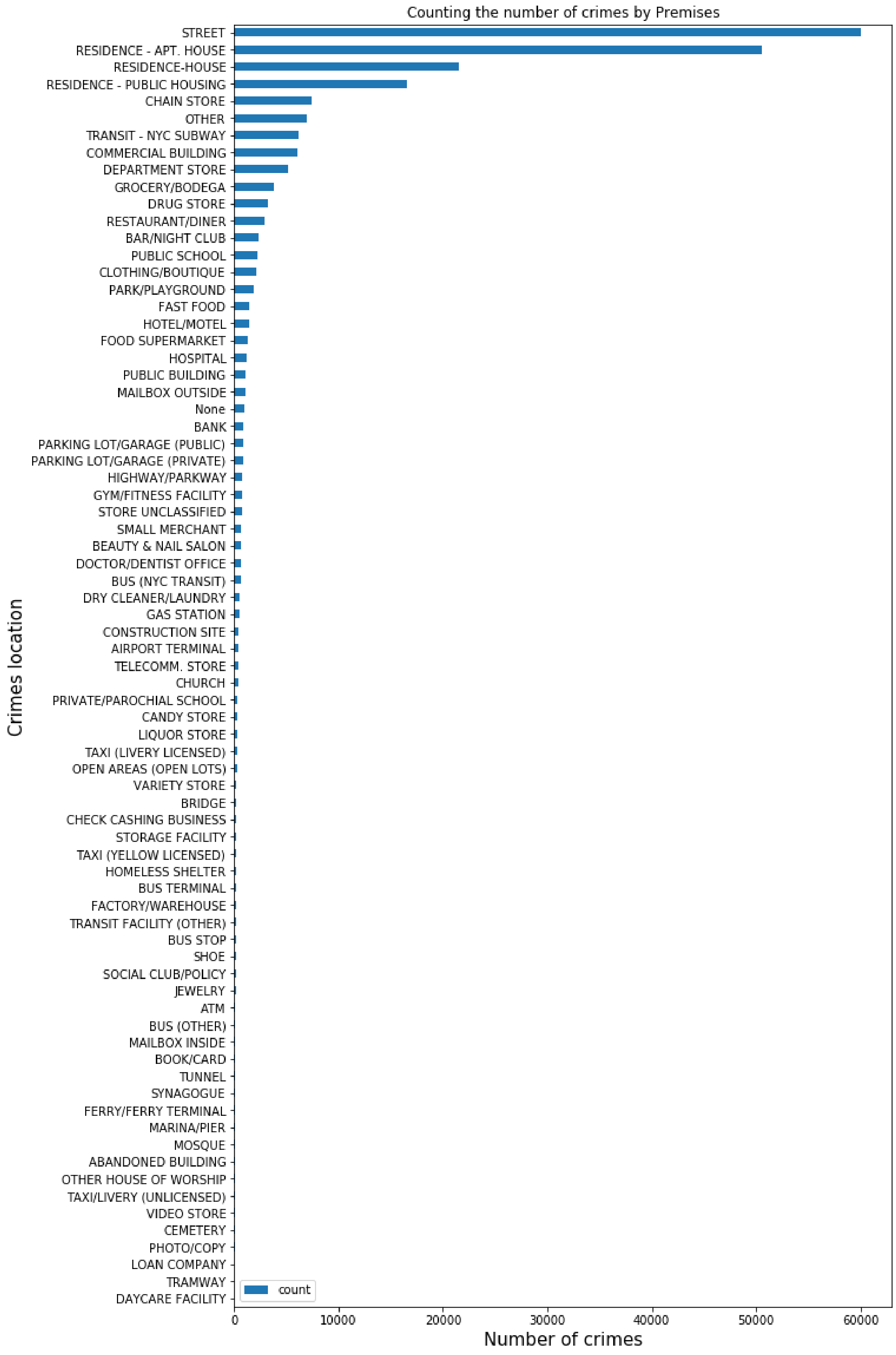
	premise	count
48	HIGHWAY/PARKWAY	798
49	PARKING LOT/GARAGE (PRIVATE)	826
50	PARKING LOT/GARAGE (PUBLIC)	873
51	BANK	879
52	None	937
53	MAILBOX OUTSIDE	1092
54	PUBLIC BUILDING	1148
55	HOSPITAL	1201
56	FOOD SUPERMARKET	1341
57	HOTEL/MOTEL	1402
58	FAST FOOD	1479
59	PARK/PLAYGROUND	1837
60	CLOTHING/BOUTIQUE	2097
61	PUBLIC SCHOOL	2269
62	BAR/NIGHT CLUB	2375
63	RESTAURANT/DINER	2903
64	DRUG STORE	3187
65	GROCERY/BODEGA	3830
66	DEPARTMENT STORE	5169
67	COMMERCIAL BUILDING	6114
68	TRANSIT - NYC SUBWAY	6163
69	OTHER	6910
70	CHAIN STORE	7452
71	RESIDENCE - PUBLIC HOUSING	16502
72	RESIDENCE-HOUSE	21474
73	RESIDENCE - APT. HOUSE	50560
74	STREET	59992

75 rows × 2 columns

In [115]:

```
premises.plot(kind = "barh", x = "premise", y = "count", figsize = (10,2
0), title = "Counting the number of crimes by Premises")
plt.xlabel("Number of crimes", fontsize = 15)
plt.ylabel("Crimes location", fontsize = 15)

# The most crimes occurred in street followed by Residence in apartement
or house.
```

```
Out[115]: Text(0, 0.5, 'Crimes location')
```



In [119]:

```
df2 = df2.toPandas()
df2
```

```
Out[119]:
```

	ID	Borough	Date	Time	Jurisdiction	Code	Level of offense	
0	857927015	MANHATTAN	1/29/19	16:37:00	N.Y. POLICE DEPT	106	FELONY	
1	479254687	QUEENS	3/29/19	17:00:00	N.Y. POLICE DEPT	107	FELONY	BU
2	320007604	BRONX	2/6/19	2:00:00	N.Y. POLICE DEPT	105	FELONY	R
3	746022144	BROOKLYN	1/8/19	22:49:00	N.Y. POLICE DEPT	117	FELONY	DAN
4	593941718	BRONX	3/17/19	5:00:00	N.Y. POLICE DEPT	344	MISDEMEANOR	ASS O
5	613547550	MANHATTAN	2/22/19	13:35:00	N.Y. POLICE DEPT	578	VIOLATION	HARRAS
6	585652917	QUEENS	2/1/19	10:00:00	N.Y. POLICE DEPT	578	VIOLATION	HARRAS
7	407860526	QUEENS	1/27/19	4:00:00	N.Y. POLICE DEPT	105	FELONY	R
8	145366108	MANHATTAN	2/11/19	12:07:00	N.Y. STATE POLICE	236	MISDEMEANOR	DAN W
9	746680655	STATEN ISLAND	3/23/19	20:06:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT L
10	513320708	BROOKLYN	1/14/19	17:35:00	N.Y. HOUSING POLICE	106	FELONY	
11	821304454	QUEENS	11/26/18	15:01:00	N.Y. POLICE DEPT	125	FELONY	NY UNCL
12	827038864	MANHATTAN	3/19/19	20:00:00	N.Y. POLICE DEPT	344	MISDEMEANOR	ASS O
13	889702556	MANHATTAN	3/11/19	21:40:00	N.Y. POLICE DEPT	105	FELONY	R
14	291569019	BRONX	2/16/19	17:15:00	N.Y. POLICE DEPT	344	MISDEMEANOR	ASS O
15	336865313	BRONX	2/19/19	19:20:00	N.Y. POLICE DEPT	361	MISDEMEANOR	OFF. AG ORD SE
16	671142050	BRONX	2/8/19	6:00:00	N.Y. POLICE DEPT	121	FELONY	C MI RE
17	883793011	QUEENS	1/12/09	0:01:00	N.Y. POLICE DEPT	116	FELONY	SE
18	944638748	QUEENS	2/23/19	22:30:00	N.Y. POLICE DEPT	344	MISDEMEANOR	ASS O
19	758966473	BRONX	2/10/19	16:00:00	N.Y. HOUSING POLICE	344	MISDEMEANOR	ASS O

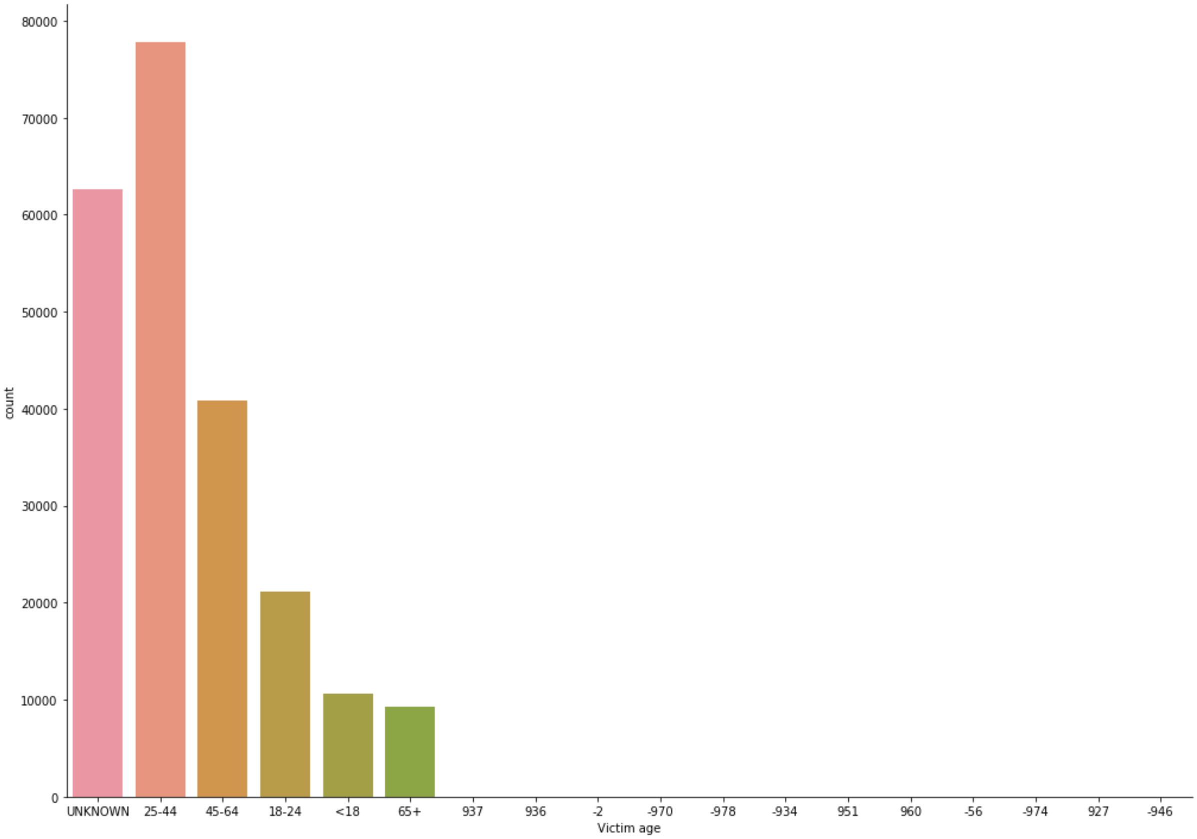
	ID	Borough	Date	Time	Jurisdiction	Code	Level of offense	
20	272821005	BROOKLYN	3/16/19	23:30:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT
21	836139486	BRONX	1/31/19	16:00:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT
22	877424333	MANHATTAN	11/11/18	12:00:00	N.Y. POLICE DEPT	361	MISDEMEANOR	OFF. AG ORD SE
23	804396233	MANHATTAN	12/19/18	8:00:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT
24	811464670	BROOKLYN	2/9/19	20:36:00	N.Y. POLICE DEPT	111	FELONY	POSSES P
25	605345964	BRONX	3/14/19	18:20:00	N.Y. POLICE DEPT	118	FELONY	DAN W
26	816741975	MANHATTAN	3/30/19	14:58:00	N.Y. POLICE DEPT	109	FELONY	GRAND
27	403194332	QUEENS	3/17/19	15:53:00	N.Y. POLICE DEPT	113	FELONY	
28	195765193	QUEENS	3/5/19	16:30:00	N.Y. POLICE DEPT	344	MISDEMEANOR	ASS O
29	375209270	BRONX	2/27/19	5:38:00	N.Y. HOUSING POLICE	114	FELONY	
...	...	...	...	...	...	...	...	
222368	871799835	QUEENS	5/13/19	11:44:00	N.Y. POLICE DEPT	351	MISDEMEANOR	MI RE
222369	386832109	MANHATTAN	5/22/19	18:22:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT
222370	526795281	BROOKLYN	5/4/19	17:45:00	N.Y. HOUSING POLICE	578	VIOLATION	HARRAS
222371	829042094	BROOKLYN	6/5/19	21:45:00	N.Y. POLICE DEPT	578	VIOLATION	HARRAS
222372	698629525	BROOKLYN	5/3/19	15:00:00	N.Y. POLICE DEPT	107	FELONY	B
222373	935708403	QUEENS	6/11/19	13:50:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT
222374	283202420	BRONX	6/4/19	22:00:00	N.Y. POLICE DEPT	361	MISDEMEANOR	OFF. AG ORD SE
222375	998128516	BROOKLYN	4/30/19	14:30:00	N.Y. POLICE DEPT	578	VIOLATION	HARRAS
222376	383359978	MANHATTAN	6/17/19	9:00:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT
222377	301752049	BROOKLYN	4/8/19	21:30:00	N.Y. HOUSING POLICE	351	MISDEMEANOR	MI RE

	ID	Borough	Date	Time	Jurisdiction	Code	Level of offense	
222378	664586224	QUEENS	5/12/19	19:50:00	N.Y. POLICE DEPT	344	MISDEMEANOR	ASS O
222379	142105031	MANHATTAN	6/15/19	23:45:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT
222380	121833520	MANHATTAN	2/22/19	9:00:00	N.Y. POLICE DEPT	109	FELONY	GRAND
222381	713294929	QUEENS	5/2/19	16:00:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT
222382	762039568	MANHATTAN	5/25/19	12:20:00	N.Y. POLICE DEPT	109	FELONY	GRAND
222383	915876963	BROOKLYN	5/21/19	23:30:00	N.Y. POLICE DEPT	578	VIOLATION	HARRAS
222384	115756814	BROOKLYN	4/21/19	22:00:00	N.Y. POLICE DEPT	110	FELONY	GRAND O
222385	158258939	BRONX	6/2/19	1:00:00	N.Y. POLICE DEPT	236	MISDEMEANOR	DAN W
222386	322958106	BROOKLYN	5/1/19	21:35:00	N.Y. POLICE DEPT	341	MISDEMEANOR	PETIT
222387	339650129	MANHATTAN	6/5/19	8:00:00	N.Y. TRANSIT POLICE	578	VIOLATION	HARRAS
222388	729155081	MANHATTAN	6/27/19	21:30:00	N.Y. POLICE DEPT	105	FELONY	F
222389	138938099	MANHATTAN	6/19/19	20:35:00	N.Y. POLICE DEPT	126	FELONY	MISCELL PE
222390	443989732	BRONX	4/1/19	15:00:00	N.Y. POLICE DEPT	578	VIOLATION	HARRAS
222391	426822007	BROOKLYN	5/16/19	6:00:00	N.Y. POLICE DEPT	351	MISDEMEANOR	C MI REI
222392	824778502	QUEENS	4/12/19	17:30:00	N.Y. HOUSING POLICE	344	MISDEMEANOR	ASS O
222393	587294745	BROOKLYN	4/15/19	12:00:00	N.Y. POLICE DEPT	109	FELONY	GRAND
222394	326362764	BROOKLYN	6/22/19	13:25:00	N.Y. POLICE DEPT	126	FELONY	MISCELL PE
222395	992657534	MANHATTAN	6/17/19	20:10:00	N.Y. TRANSIT POLICE	106	FELONY	
222396	577523166	QUEENS	6/7/19	9:28:00	N.Y. POLICE DEPT	340	MISDEMEANOR	
222397	956145385	MANHATTAN	5/2/19	18:30:00	N.Y. TRANSIT POLICE	230	MISDEMEANOR	J

222398 rows × 20 columns

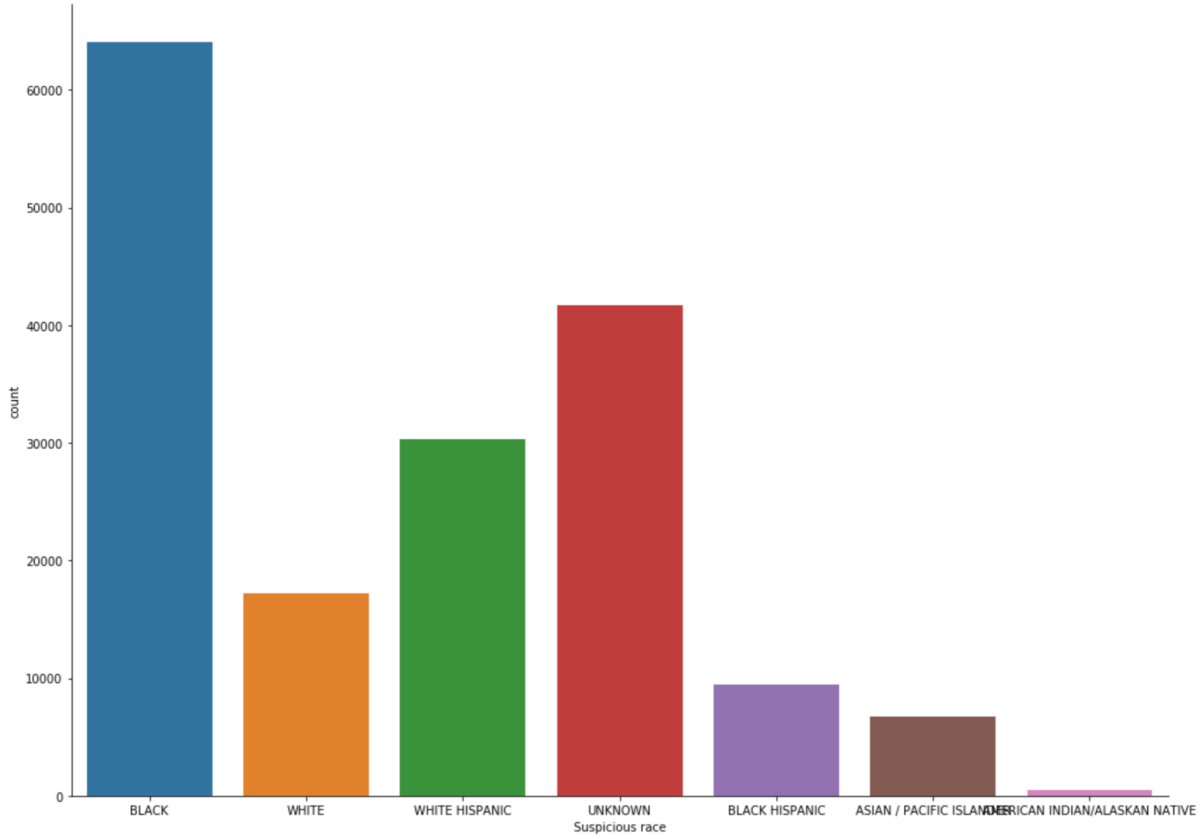
In [120]:
```
# plotting the age group the most attacked during 2019:


sns.catplot(x = "Victim age", kind = "count", data = df2, height =10, as
pect =1.4)


# seems like the victim age is between 25 and 44
```

Out[120]:
```
<seaborn.axisgrid.FacetGrid at 0x1a42188320>
```



```
In [121]: # plotting the suspicious age:

          sns.catplot(x = "Suspicious race", kind = "count", data = df2, height =1
          0, aspect =1.4)

          # the black are more suspected than others
```

```
Out[121]: <seaborn.axisgrid.FacetGrid at 0x1a5b832cc0>
```



In [122]:

```
import folium
import pandas as pd
map2 = folium.Map(location=[40.712776, -74.005974], tiles ="cartodbdark_matter", zoom_start =10)
map2
```

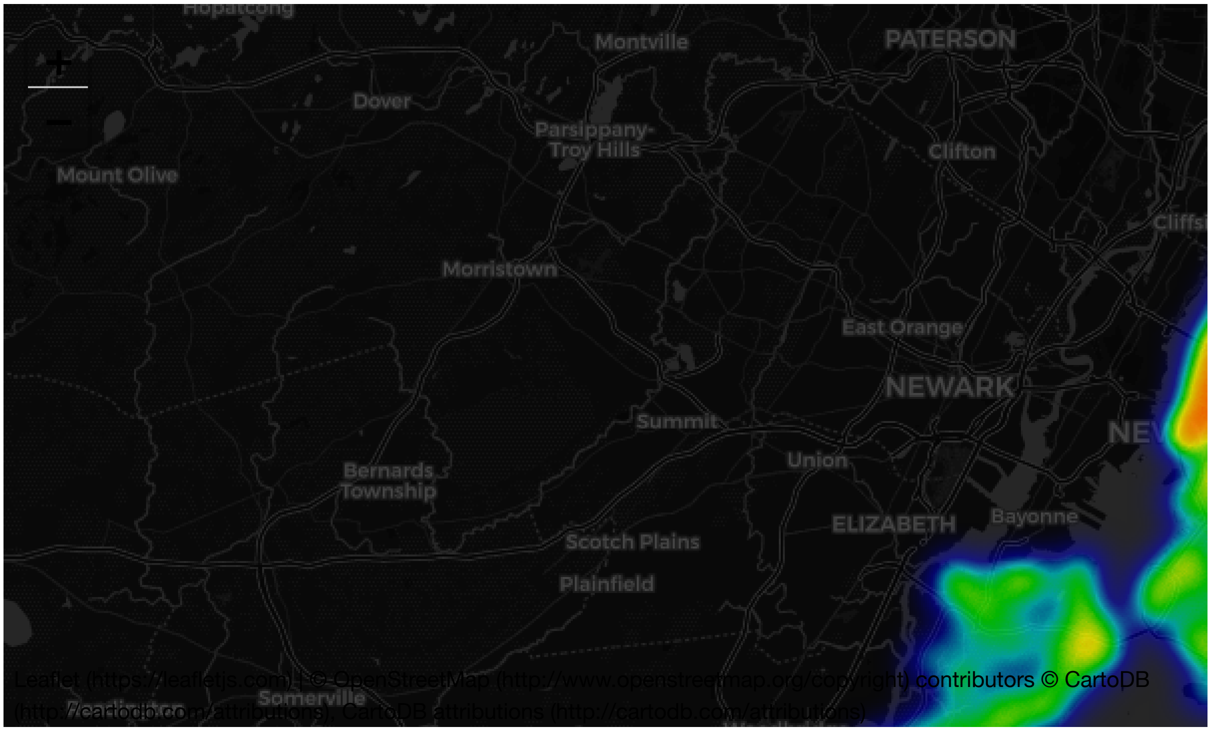
Out[122]:

+
−
BINGHAMTON
Newburgh
C
Waterbu
SCRANTON
NEW
WILKES-BARRE
Leaflet (https://leafletjs.com) | © OpenStreetMap (http://www.openstreetmap.org/copyright) contributors © CartoDB (http://cartodb.com/attributions), CartoDB attributions (http://cartodb.com/attributions)
BRIDGEPORT

In [123]:

```
# plotting a map of the most crimes in NYC by using Latitude and longtitude.

# first I will drop NA from the data
d = df2.dropna()

positions = list(zip(d['Latitude'], d['Longitude']))
map1 = folium.Map(location=[40.712776, -74.005974], zoom_start=10, tiles = "cartodbdark_matter")
HeatMap(positions[:30000], radius = 10).add_to(map1)
map1


# by looking at the map , we can confirm that Manhattan and Bronx have more crimes during the first 6 month of 2019.
```

Out[123]:


+
−
Hopatcong
Montville
PATERSON
Dover
Parsippany-Troy Hills
Clifton
Mount Olive
Cliffs
Morristown
East Orange
NEWARK
NE
Summit
Union
Bernards Township
ELIZABETH
Bayonne
Scotch Plains
Plainfield
Somerville
Leaflet (https://leafletjs.com) | © OpenStreetMap (http://www.openstreetmap.org/copyright) contributors © CartoDB (http://cartodb.com/attributions), CartoDB attributions (http://cartodb.com/attributions)

In [ ]: