PREDICT THE TYPE OF CRIME

Kartikay Gupta(1801ME28)
Jay Kabra(1801EE08)
Ayush Shah(1801EE08)
Aparsh Gupta(1801EE08)

WORKFLOW

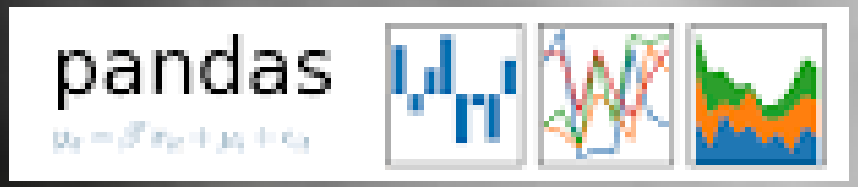

Chicago crime dataset
Data pre-processing
Crime rates over years
Time window for each type of crime
Zone-wise intensity of crime

Data pre-processing

- Converted the data into .csv format.
- Eliminated the data which had any null field.

Time window for each type of crime

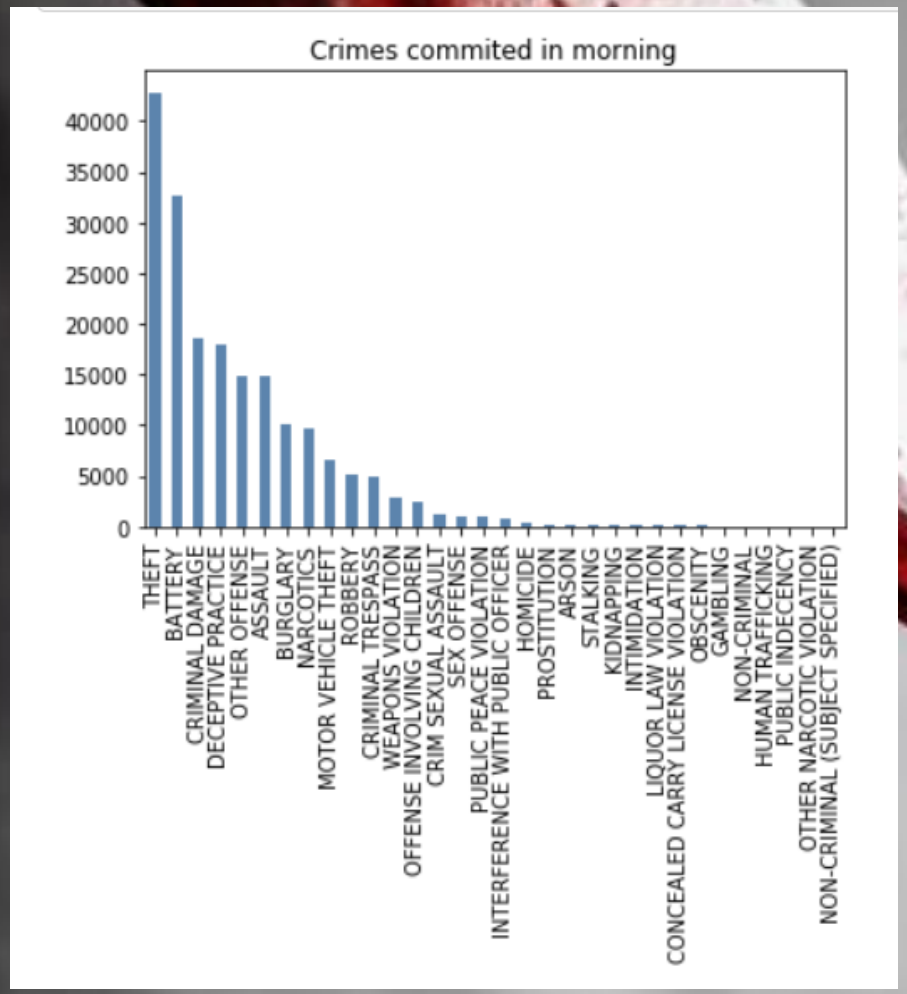
- **Extracted the time of crime from the date time string.**
- **Divided the crimes into four groups morning, evening, afternoon and night based upon time slabs.**
- **Plotted the variation in different crimes in different time slabs.**

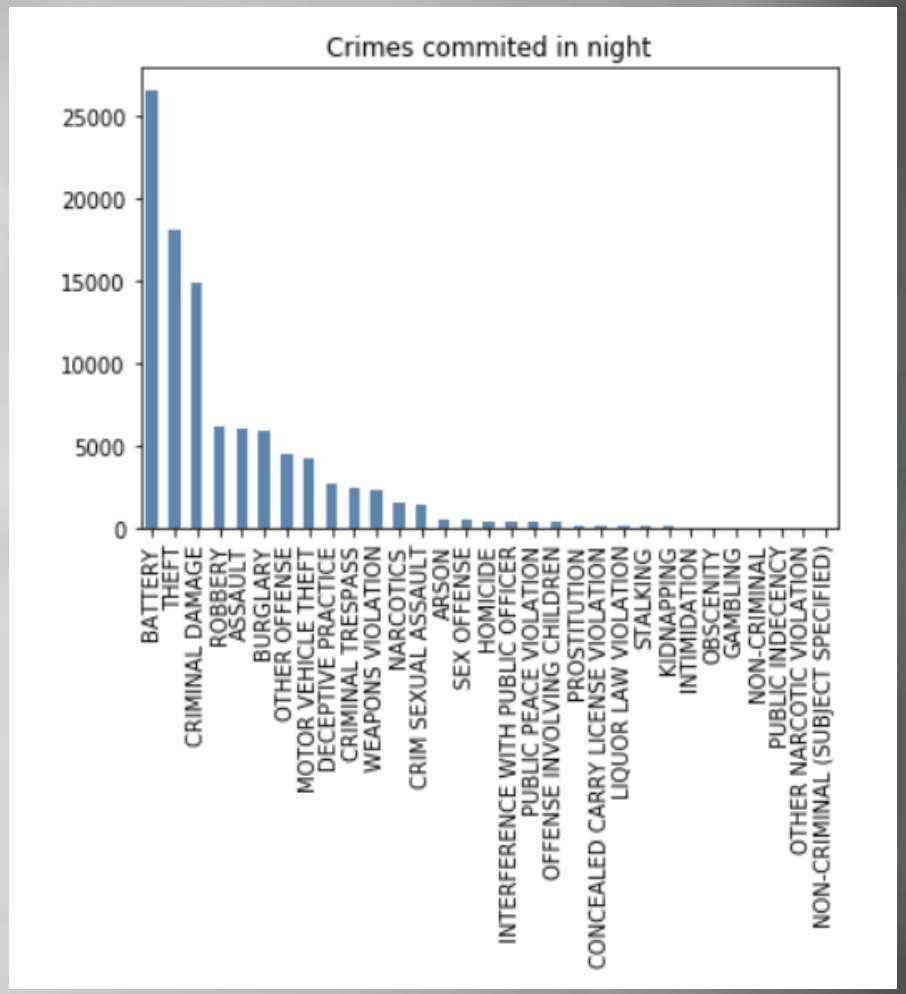


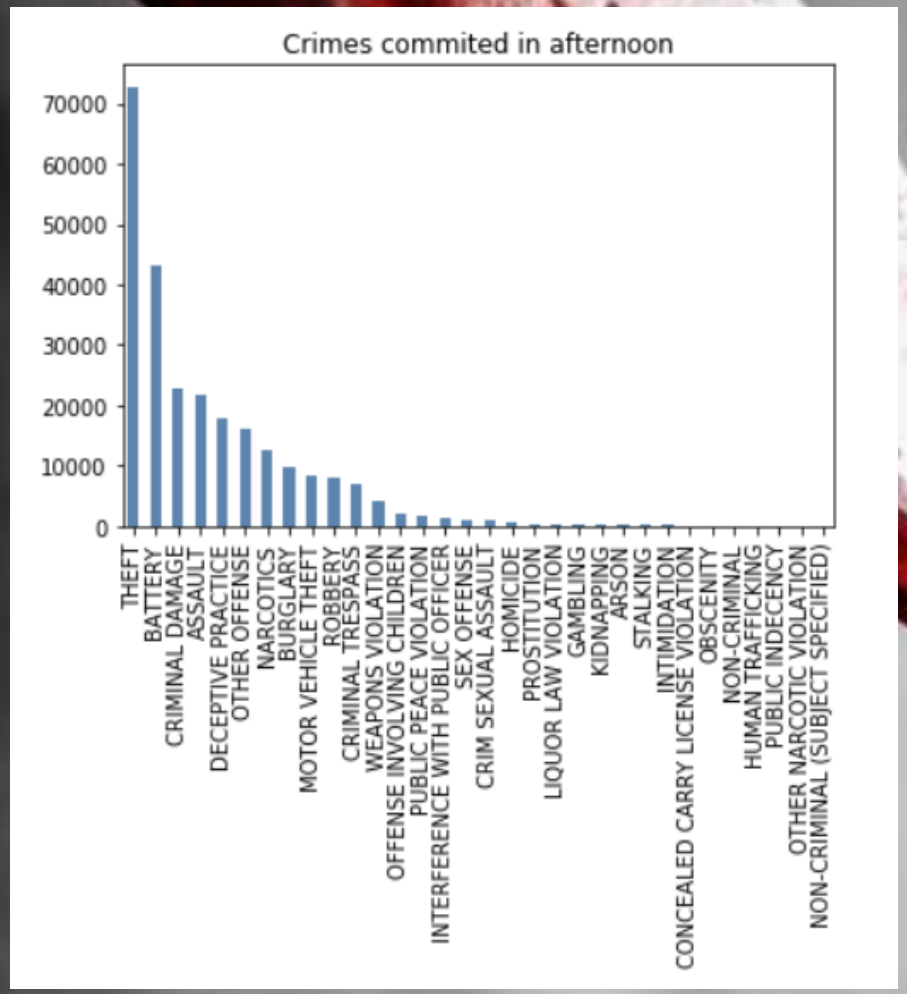
Time window for each type of crime

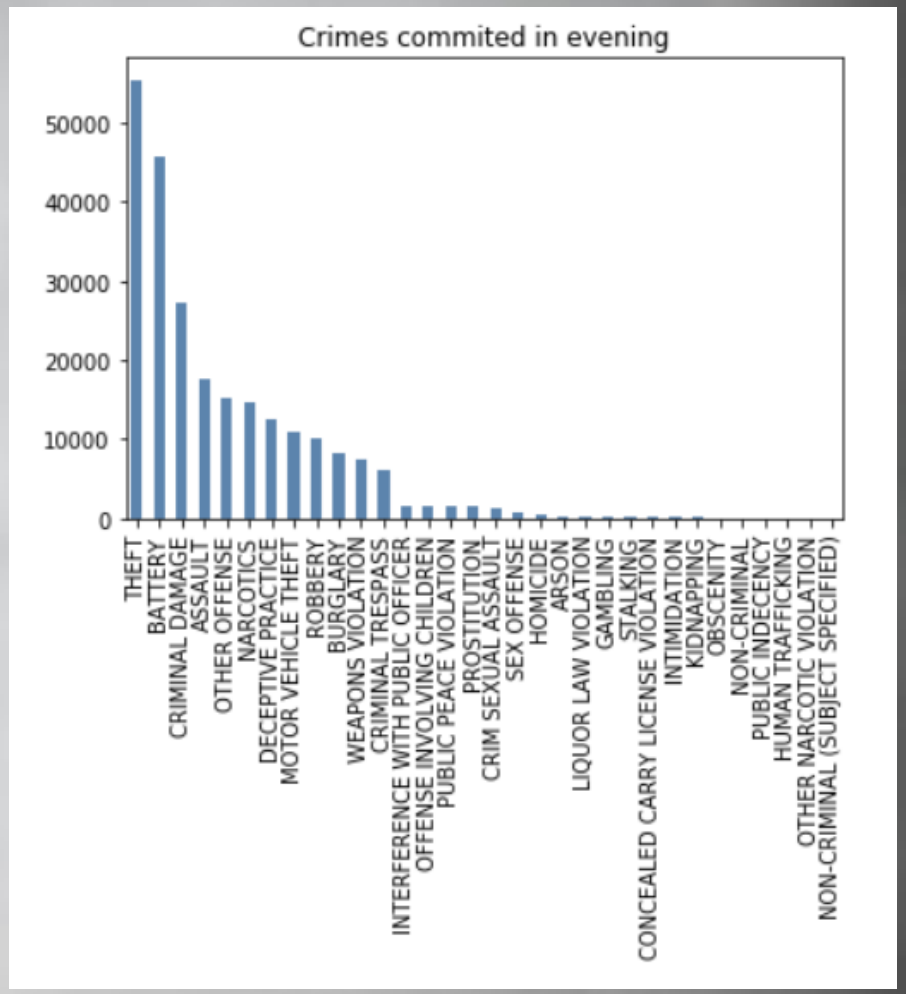
In [5]:
```
# Extracting time_window from Date column
def make_window(row):
    date, time, half = row.split()
    h = int(time[0:2])
    if(half == 'AM'):
        if(h <= 6):
            return 'night'
        else:
            return 'morning'
    if(half == 'PM'):
        if(h <= 6):
            return 'afternoon'
        else:
            return 'evening'
```

In [6]:
```
df['time_window'] = df.apply(lambda row: make_window(row['Date']), axis=1)
```


Crimes commited in morning
40000
35000
30000
25000
20000
15000
10000
5000
0
THEFT
BATTERY
CRIMINAL DAMAGE
DECEPTIVE PRACTICE
OTHER OFFENSE
ASSAULT
BURGLARY
NARCOTICS
MOTOR VEHICLE THEFT
ROBBERY
CRIMINAL TRESPASS
WEAPONS VIOLATION
OFFENSE INVOLVING CHILDREN
CRIM SEXUAL ASSAULT
SEX OFFENSE
PUBLIC PEACE VIOLATION
INTERFERENCE WITH PUBLIC OFFICER
HOMICIDE
PROSTITUTION
ARSON
STALKING
KIDNAPPING
INTIMIDATION
LIQUOR LAW VIOLATION
CONCEALED CARRY LICENSE VIOLATION
OBSCENITY
GAMBLING
NON-CRIMINAL
HUMAN TRAFFICKING
PUBLIC INDECENCY
OTHER NARCOTIC VIOLATION
NON-CRIMINAL (SUBJECT SPECIFIED)


Crimes commited in night
25000
20000
15000
10000
5000
0
BATTERY
THEFT
CRIMINAL DAMAGE
ROBBERY
ASSAULT
BURGLARY
OTHER OFFENSE
MOTOR VEHICLE THEFT
DECEPTIVE PRACTICE
CRIMINAL TRESPASS
WEAPONS VIOLATION
NARCOTICS
CRIM SEXUAL ASSAULT
ARSON
SEX OFFENSE
HOMICIDE
INTERFERENCE WITH PUBLIC OFFICER
PUBLIC PEACE VIOLATION
OFFENSE INVOLVING CHILDREN
PROSTITUTION
CONCEALED CARRY LICENSE VIOLATION
LIQUOR LAW VIOLATION
STALKING
KIDNAPPING
INTIMIDATION
OBSCENITY
GAMBLING
NON-CRIMINAL
PUBLIC INDECENCY
OTHER NARCOTIC VIOLATION
NON-CRIMINAL (SUBJECT SPECIFIED)


Crimes commited in afternoon
70000
60000
50000
40000
30000
20000
10000
0
THEFT
BATTERY
CRIMINAL DAMAGE
ASSAULT
DECEPTIVE PRACTICE
OTHER OFFENSE
NARCOTICS
BURGLARY
MOTOR VEHICLE THEFT
ROBBERY
CRIMINAL TRESPASS
WEAPONS VIOLATION
OFFENSE INVOLVING CHILDREN
PUBLIC PEACE VIOLATION
INTERFERENCE WITH PUBLIC OFFICER
SEX OFFENSE
CRIM SEXUAL ASSAULT
HOMICIDE
PROSTITUTION
LIQUOR LAW VIOLATION
GAMBLING
KIDNAPPING
ARSON
STALKING
INTIMIDATION
CONCEALED CARRY LICENSE VIOLATION
OBSCENITY
NON-CRIMINAL
HUMAN TRAFFICKING
PUBLIC INDECENCY
OTHER NARCOTIC VIOLATION
NON-CRIMINAL (SUBJECT SPECIFIED)


Crimes commited in evening
50000
40000
30000
20000
10000
0
THEFT
BATTERY
CRIMINAL DAMAGE
ASSAULT
OTHER OFFENSE
NARCOTICS
DECEPTIVE PRACTICE
MOTOR VEHICLE THEFT
ROBBERY
BURGLARY
WEAPONS VIOLATION
CRIMINAL TRESPASS
INTERFERENCE WITH PUBLIC OFFICER
OFFENSE INVOLVING CHILDREN
PUBLIC PEACE VIOLATION
PROSTITUTION
CRIM SEXUAL ASSAULT
SEX OFFENSE
HOMICIDE
ARSON
LIQUOR LAW VIOLATION
GAMBLING
STALKING
CONCEALED CARRY LICENSE VIOLATION
INTIMIDATION
KIDNAPPING
OBSCENITY
NON-CRIMINAL
PUBLIC INDECENCY
HUMAN TRAFFICKING
OTHER NARCOTIC VIOLATION
NON-CRIMINAL (SUBJECT SPECIFIED)

Crime rates over years

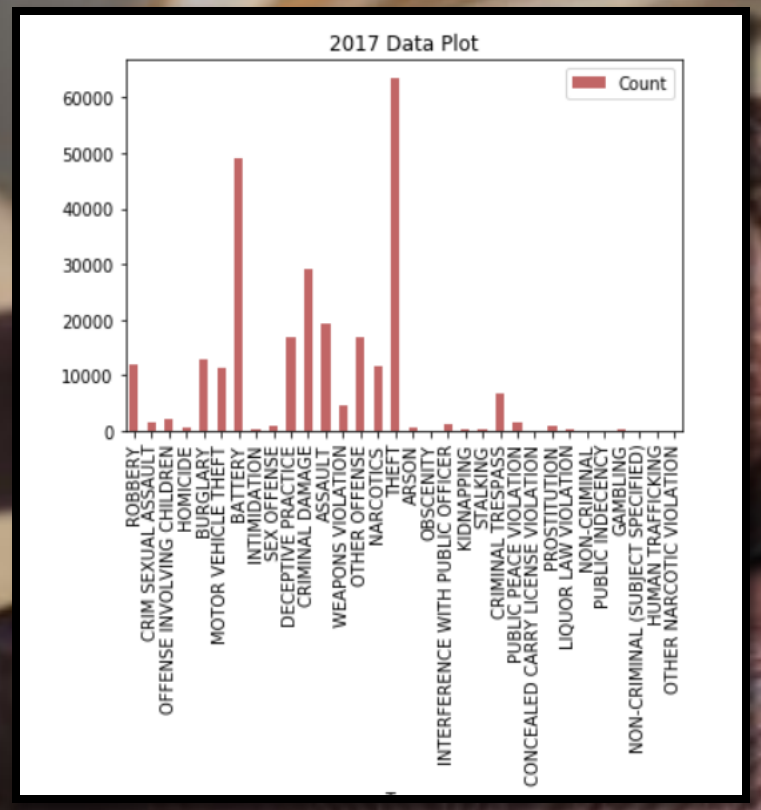
- Started by using the pre-processed data.
- Year wise divided the crimes.
- Plotted the individual year-crime graphs.
- Plotted the variation over years in different crimes which clearly displays the increase and decrease in crimes.

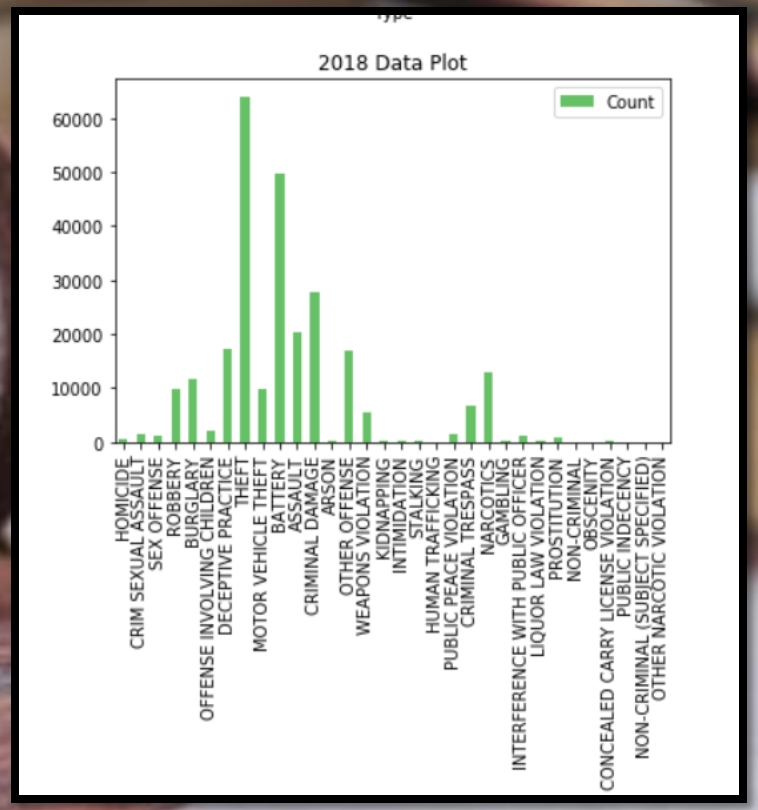
Crime rates over years

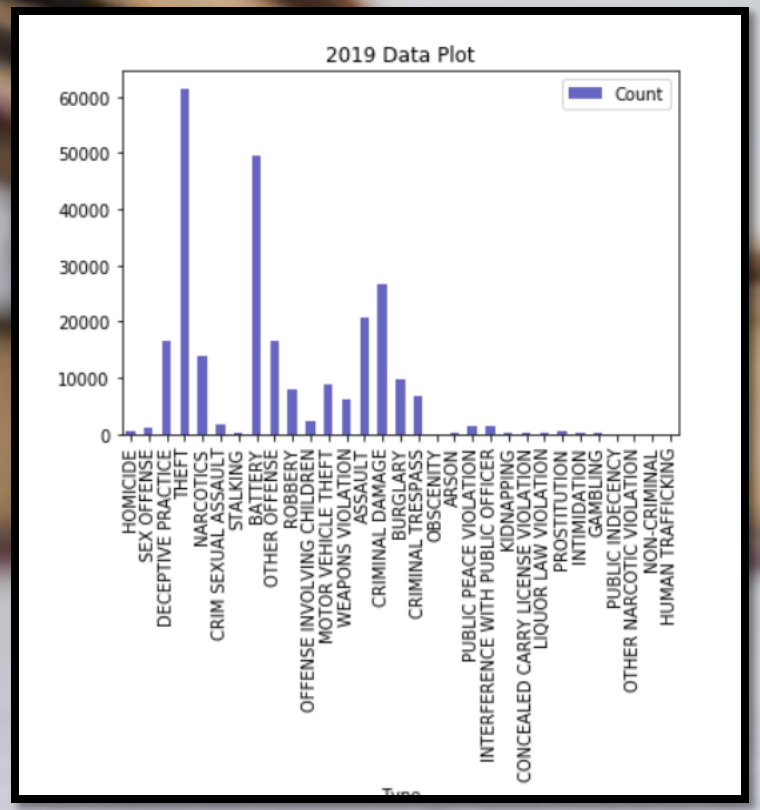
In [5]:
```
#calculating the size of data points of each year
print(df.groupby('Year').size())
```

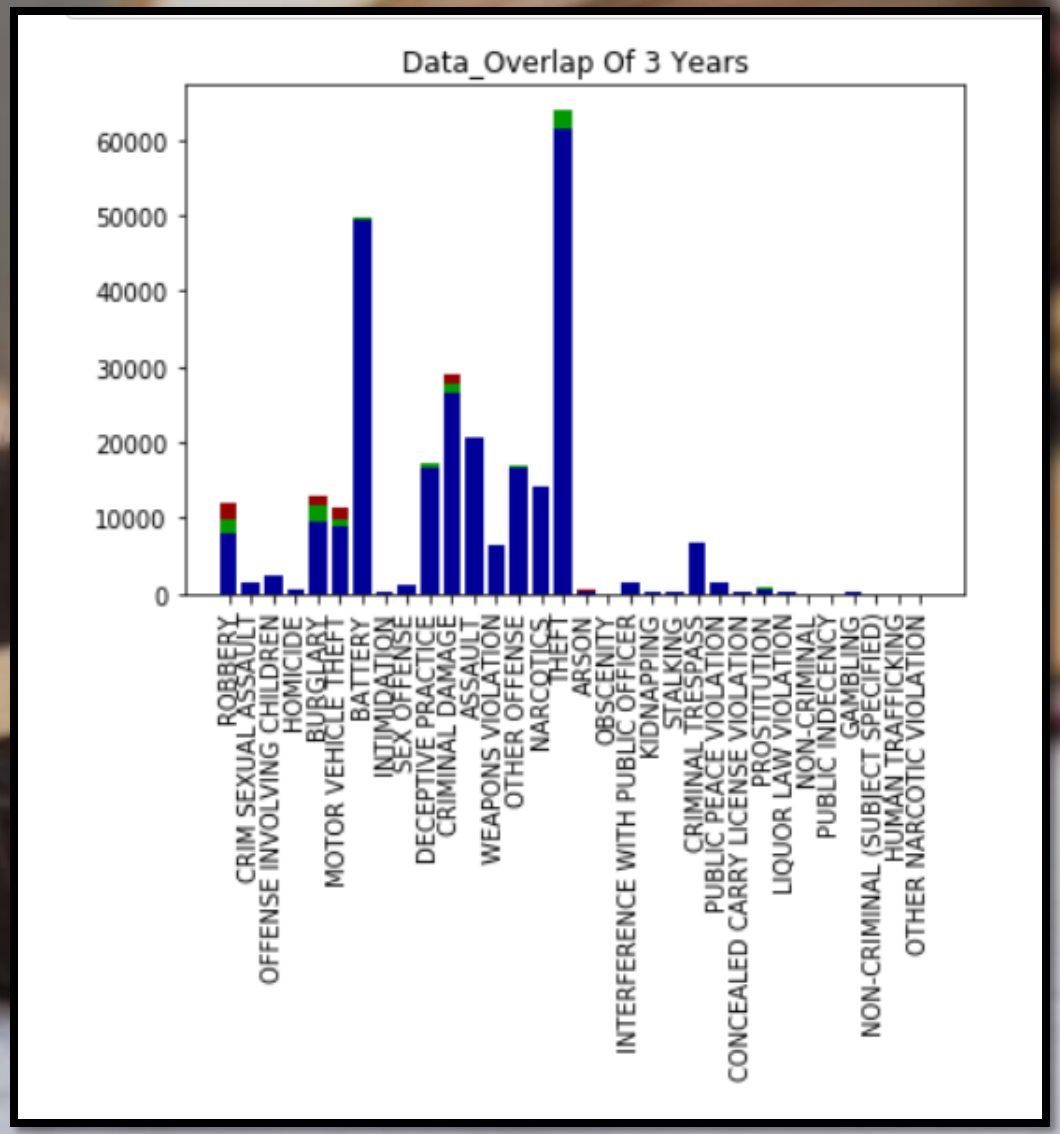
```
Year
2017    264116
2018    262787
2019    256023
dtype: int64
```

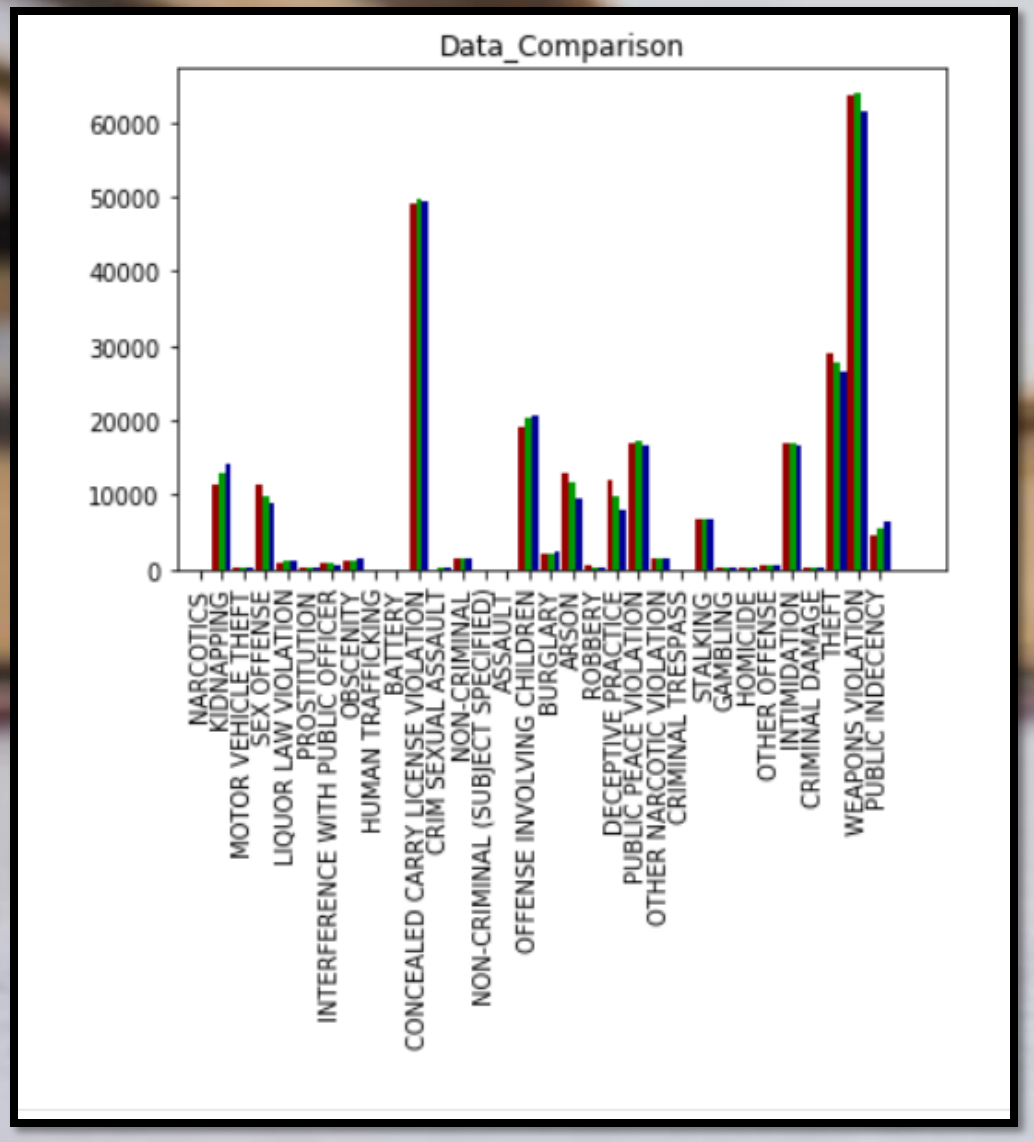
In [6]:
```
#calculation for the year 2017
get_type1 = []

#applying nested for loop to calculate the occurance of each type of crime
for i in range(0,264116):
    primary_type = crime_data['Primary Type'].iloc[i]
    get_index = -1

    for j in range(0, len(get_type1)):
        if (get_type1[j][0] == primary_type):
            get_index = j
            get_type1[j][1]+=1

    if get_index == -1:
        get_type1.append([primary_type, 1])


#plot for 2017 data
get_type1
added_data1 = pd.DataFrame(columns=['Type', 'Count'], data=get_type1)
added_data1.head()
added_data1.plot(x='Type', y='Count', kind='bar',color=(0.6, 0.0, 0.0, 0.6))
plt.title("2017 Data PLot")
plt.show()
```


2017 Data Plot
Count
60000
50000
40000
30000
20000
10000
0
ROBBERY
CRIM SEXUAL ASSAULT
OFFENSE INVOLVING CHILDREN
HOMICIDE
BURGLARY
MOTOR VEHICLE THEFT
BATTERY
INTIMIDATION
SEX OFFENSE
DECEPTIVE PRACTICE
CRIMINAL DAMAGE
ASSAULT
WEAPONS VIOLATION
OTHER OFFENSE
NARCOTICS
THEFT
ARSON
OBSCENITY
INTERFERENCE WITH PUBLIC OFFICER
KIDNAPPING
STALKING
CRIMINAL TRESPASS
PUBLIC PEACE VIOLATION
CONCEALED CARRY LICENSE VIOLATION
PROSTITUTION
LIQUOR LAW VIOLATION
NON-CRIMINAL
PUBLIC INDECENCY
GAMBLING
NON-CRIMINAL (SUBJECT SPECIFIED)
HUMAN TRAFFICKING
OTHER NARCOTIC VIOLATION


2018 Data Plot
Count
60000
50000
40000
30000
20000
10000
0
HOMICIDE
CRIM SEXUAL ASSAULT
SEX OFFENSE
ROBBERY
BURGLARY
OFFENSE INVOLVING CHILDREN
DECEPTIVE PRACTICE
THEFT
MOTOR VEHICLE THEFT
BATTERY
ASSAULT
CRIMINAL DAMAGE
ARSON
OTHER OFFENSE
WEAPONS VIOLATION
KIDNAPPING
INTIMIDATION
STALKING
HUMAN TRAFFICKING
PUBLIC PEACE VIOLATION
CRIMINAL TRESPASS
NARCOTICS
GAMBLING
INTERFERENCE WITH PUBLIC OFFICER
LIQUOR LAW VIOLATION
PROSTITUTION
NON-CRIMINAL
OBSCENITY
CONCEALED CARRY LICENSE VIOLATION
PUBLIC INDECENCY
NON-CRIMINAL (SUBJECT SPECIFIED)
OTHER NARCOTIC VIOLATION

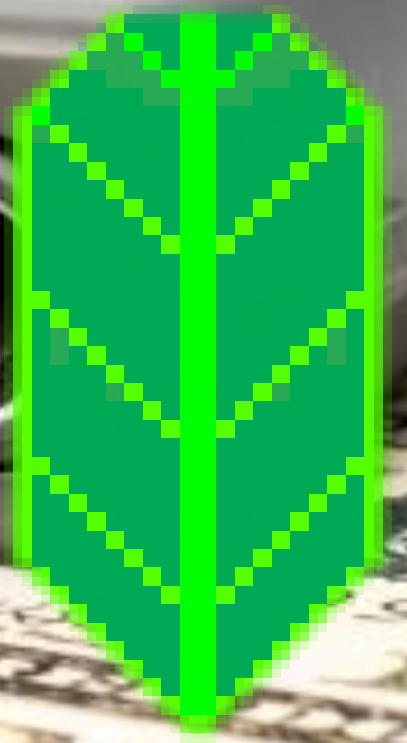

2019 Data Plot
Count
60000
50000
40000
30000
20000
10000
0
HOMICIDE
SEX OFFENSE
DECEPTIVE PRACTICE
THEFT
NARCOTICS
CRIM SEXUAL ASSAULT
STALKING
BATTERY
OTHER OFFENSE
ROBBERY
OFFENSE INVOLVING CHILDREN
MOTOR VEHICLE THEFT
WEAPONS VIOLATION
ASSAULT
CRIMINAL DAMAGE
BURGLARY
CRIMINAL TRESPASS
OBSCENITY
ARSON
PUBLIC PEACE VIOLATION
INTERFERENCE WITH PUBLIC OFFICER
KIDNAPPING
CONCEALED CARRY LICENSE VIOLATION
LIQUOR LAW VIOLATION
PROSTITUTION
INTIMIDATION
GAMBLING
PUBLIC INDECENCY
OTHER NARCOTIC VIOLATION
NON-CRIMINAL
HUMAN TRAFFICKING


Data_Overlap Of 3 Years
60000
50000
40000
30000
20000
10000
0
ROBBERY
CRIM SEXUAL ASSAULT
OFFENSE INVOLVING CHILDREN
HOMICIDE
BURGLARY
MOTOR VEHICLE THEFT
BATTERY
INTIMIDATION
SEX OFFENSE
DECEPTIVE PRACTICE
CRIMINAL DAMAGE
ASSAULT
WEAPONS VIOLATION
OTHER OFFENSE
NARCOTICS
THEFT
ARSON
OBSCENITY
INTERFERENCE WITH PUBLIC OFFICER
KIDNAPPING
STALKING
CRIMINAL TRESPASS
PUBLIC PEACE VIOLATION
CONCEALED CARRY LICENSE VIOLATION
PROSTITUTION
LIQUOR LAW VIOLATION
NON-CRIMINAL
PUBLIC INDECENCY
GAMBLING
NON-CRIMINAL (SUBJECT SPECIFIED)
HUMAN TRAFFICKING
OTHER NARCOTIC VIOLATION


Data_Comparison
60000
50000
40000
30000
20000
10000
0
NARCOTICS
KIDNAPPING
MOTOR VEHICLE THEFT
SEX OFFENSE
LIQUOR LAW VIOLATION
PROSTITUTION
INTERFERENCE WITH PUBLIC OFFICER
OBSCENITY
HUMAN TRAFFICKING
BATTERY
CONCEALED CARRY LICENSE VIOLATION
CRIM SEXUAL ASSAULT
NON-CRIMINAL
NON-CRIMINAL (SUBJECT SPECIFIED)
ASSAULT
OFFENSE INVOLVING CHILDREN
BURGLARY
ARSON
ROBBERY
DECEPTIVE PRACTICE
PUBLIC PEACE VIOLATION
OTHER NARCOTIC VIOLATION
CRIMINAL TRESPASS
STALKING
GAMBLING
HOMICIDE
OTHER OFFENSE
INTIMIDATION
CRIMINAL DAMAGE
THEFT
WEAPONS VIOLATION
PUBLIC INDECENCY

District-wise crime frequency

- Divided the data, area wise.
- Calculated the number of crimes in each area.
- Plotted the **frequency map** of crimes using **folium and seaborn** library.
- Classified the frequencies of crimes in **6 slabs**. (**Thresholding**)



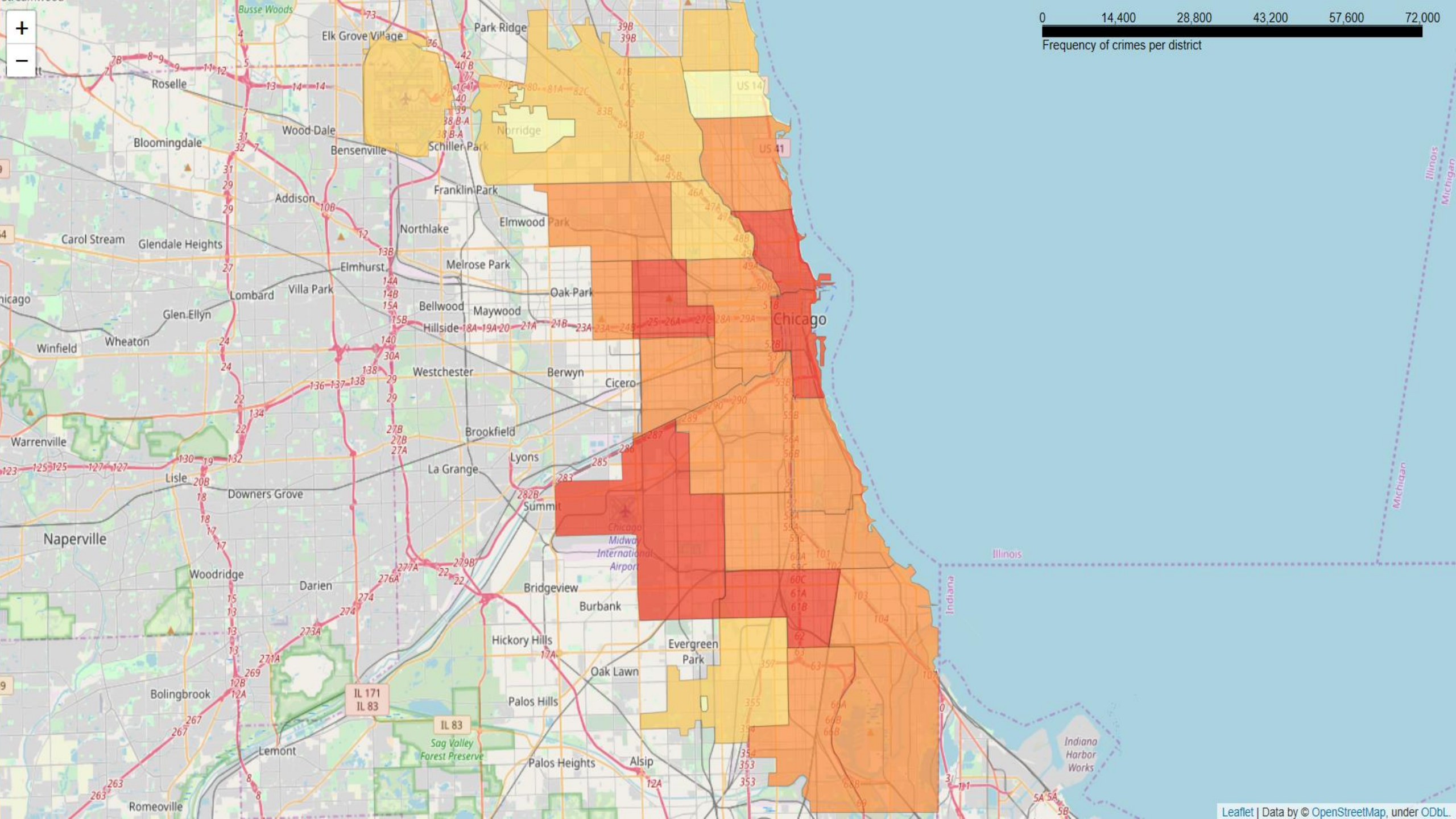
FOLIUM

District-wise crime frequency

In [10]:
```
## Reset index and name the district and crime count columns
district_crime = district_crime.reset_index()
district_crime.columns = ['District', 'Count']
```

In [11]:
```
cleaned_df = df[df['Latitude'].notnull() & df['Longitude'].notnull()].copy()
```

In [12]:
```
# creation of the choropleth
geo_path = 'C:/Users/aayus/Desktop/Boundaries - Police Districts (current).geojson'
district_map = folium.Map(location=[cleaned_df['Latitude'].mean(), cleaned_df['Longitude'].mean()], zoom_s
tart=10)
threshold_scale = list(np.linspace(0,72000,6))
district_map.choropleth(geo_data=geo_path,
                name='choropleth',
                data = district_crime,
                threshold_scale=threshold_scale,
                columns = ['District', 'Count'],
                key_on = 'feature.properties.dist_num',
                fill_color = 'YlOrRd',
                fill_opacity = 0.7,
                line_opacity = 0.2,
                legend_name = 'Frequency of crimes per district',
                highlight=True)


district_map.save("Crime-per-district-choropleth.html")
district_map
```


0
14,400
28,800
43,200
57,600
72,000
Frequency of crimes per district
Chicago
Park Ridge
Elk Grove Village
Roselle
Wood Dale
Bloomingdale
Bensenville
Schiller Park
Norridge
Franklin Park
Addison
Northlake
Elmwood Park
Melrose Park
Carol Stream
Glendale Heights
Elmhurst
Lombard
Villa Park
Oak Park
Bellwood
Maywood
Hillside
Glen Ellyn
Wheaton
Winfield
Westchester
Berwyn
Cicero
Brookfield
Warrenville
Lyons
La Grange
Lisle
Downers Grove
Summit
Naperville
Woodridge
Darien
Bridgeview
Burbank
Hickory Hills
Evergreen Park
Oak Lawn
Palos Hills
Bolingbrook
IL 171
IL 83
Sag Valley Forest Preserve
Lemont
Palos Heights
Alsip
Romeoville
Indiana Harbor Works
Illinois
Indiana
Michigan
Leaflet | Data by © OpenStreetMap, under ODbL.