Cleaning and analyzing crime data

Project group 30

Francisco Chavez Gonzalez

Xueshi Bai

Santosh Kumar Kushal Yelamandala

Nixon Lobo

Pruthviraj Chintakindi

—

Course: **Foundations for Data Analytics Engineering**

—

Professor: *Sivarit (Tony) Sultornsanee*

In this project, we worked with a real-world dataset containing crime data from 2020 to the present. We have cleaned and prepared the dataset for analysis, performed exploratory data analysis, and answered specific questions related to crime trends, patterns, and factors influencing crime rates.

Data Acquisition: We Downloaded the dataset from the provided link and loaded it into Jupyter Notebook.

Data Acquisition: Download the dataset from the provided link and load it into your preferred data analysis tool

In [1]:
```
#Loading Data and Displaying first few Rows
import pandas as pd
df = pd.read_csv('Crime_Data_from_2020_to_Present.csv')
df.head()
```

Out[1]:

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	...	Status	Status Desc	Crm Cd 1	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION	Cross Street	LAT	LON
0	10304468	01/08/2020 12:00:00 AM	01/08/2020 12:00:00 AM	2230	3	Southwest	377	2	624	BATTERY - SIMPLE ASSAULT	...	AO	Adult Other	624.0	NaN	NaN	NaN	1100 W 39TH PL	NaN	34.0141	-118.2978
1	190101086	01/02/2020 12:00:00 AM	01/01/2020 12:00:00 AM	330	1	Central	163	2	624	BATTERY - SIMPLE ASSAULT	...	IC	Invest Cont	624.0	NaN	NaN	NaN	700 S HILL ST	NaN	34.0459	-118.2545
2	200110444	04/14/2020 12:00:00 AM	02/13/2020 12:00:00 AM	1200	1	Central	155	2	845	SEX OFFENDER REGISTRANT OUT OF COMPLIANCE	...	AA	Adult Arrest	845.0	NaN	NaN	NaN	200 E 6TH ST	NaN	34.0448	-118.2474
3	191501505	01/01/2020 12:00:00 AM	01/01/2020 12:00:00 AM	1730	15	N Hollywood	1543	2	745	VANDALISM - MISDEAMEANOR ($399 OR UNDER)	...	IC	Invest Cont	745.0	998.0	NaN	NaN	5400 CORTEEN PL	NaN	34.1685	-118.4019
4	191921269	01/01/2020 12:00:00 AM	01/01/2020 12:00:00 AM	415	19	Mission	1998	2	740	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA...	...	IC	Invest Cont	740.0	NaN	NaN	NaN	14400 TITUS ST	NaN	34.2198	-118.4468

5 rows × 28 columns

Data Inspection: We thoroughly went through the given data set.

Data Inspection: Display the first few rows of the dataset, Check the data types of each column, Review column names and descriptions, if available.

In [2]:
```
df.head()
```

Out[2]:

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	...	Status	Status Desc	Crm Cd 1	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION	Cross Street	LAT	LON
0	10304468	01/08/2020 12:00:00 AM	01/08/2020 12:00:00 AM	2230	3	Southwest	377	2	624	BATTERY - SIMPLE ASSAULT	...	AO	Adult Other	624.0	NaN	NaN	NaN	1100 W 39TH PL	NaN	34.0141	-118.2978
1	190101086	01/02/2020 12:00:00 AM	01/01/2020 12:00:00 AM	330	1	Central	163	2	624	BATTERY - SIMPLE ASSAULT	...	IC	Invest Cont	624.0	NaN	NaN	NaN	700 S HILL ST	NaN	34.0459	-118.2545
2	200110444	04/14/2020 12:00:00 AM	02/13/2020 12:00:00 AM	1200	1	Central	155	2	845	SEX OFFENDER REGISTRANT OUT OF COMPLIANCE	...	AA	Adult Arrest	845.0	NaN	NaN	NaN	200 E 6TH ST	NaN	34.0448	-118.2474
3	191501505	01/01/2020 12:00:00 AM	01/01/2020 12:00:00 AM	1730	15	N Hollywood	1543	2	745	VANDALISM - MISDEAMEANOR ($399 OR UNDER)	...	IC	Invest Cont	745.0	998.0	NaN	NaN	5400 CORTEEN PL	NaN	34.1685	-118.4019
4	191921269	01/01/2020 12:00:00 AM	01/01/2020 12:00:00 AM	415	19	Mission	1998	2	740	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA...	...	IC	Invest Cont	740.0	NaN	NaN	NaN	14400 TITUS ST	NaN	34.2198	-118.4468

5 rows × 28 columns

In [3]:
```
df.dtypes
```
Out[3]:
```
DR_NO              int64
Date Rptd         object
DATE OCC          object
TIME OCC           int64
AREA               int64
AREA NAME         object
Rpt Dist No        int64
Part 1-2           int64
Crm Cd             int64
Crm Cd Desc       object
Mocodes           object
Vict Age           int64
Vict Sex          object
Vict Descent      object
Premis Cd        float64
Premis Desc       object
Weapon Used Cd   float64
Weapon Desc       object
Status            object
Status Desc       object
Crm Cd 1         float64
Crm Cd 2         float64
Crm Cd 3         float64
Crm Cd 4         float64
LOCATION          object
Cross Street      object
LAT              float64
LON              float64
dtype: object
```
In [4]:
```
df.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 807377 entries, 0 to 807376
Data columns (total 28 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   DR_NO           807377 non-null  int64
 1   Date Rptd       807377 non-null  object
 2   DATE OCC        807377 non-null  object
 3   TIME OCC        807377 non-null  int64
 4   AREA            807377 non-null  int64
 5   AREA NAME       807377 non-null  object
 6   Rpt Dist No     807377 non-null  int64
 7   Part 1-2        807377 non-null  int64
 8   Crm Cd          807377 non-null  int64
 9   Crm Cd Desc     807377 non-null  object
 10  Mocodes         696010 non-null  object
 11  Vict Age        807377 non-null  int64
 12  Vict Sex        701468 non-null  object
 13  Vict Descent    701460 non-null  object
 14  Premis Cd       807368 non-null  float64
 15  Premis Desc     806901 non-null  object
 16  Weapon Used Cd  281174 non-null  float64
 17  Weapon Desc     281174 non-null  object
 18  Status          807377 non-null  object
 19  Status Desc     807377 non-null  object
 20  Crm Cd 1        807367 non-null  float64
 21  Crm Cd 2        59483 non-null   float64
 22  Crm Cd 3        1987 non-null    float64
 23  Crm Cd 4        58 non-null      float64
 24  LOCATION        807377 non-null  object
 25  Cross Street    129232 non-null  object
 26  LAT             807377 non-null  float64
 27  LON             807377 non-null  float64
dtypes: float64(8), int64(7), object(13)
memory usage: 172.5+ MB
```
In [5]:
```
df.shape[0]
```
Out[5]:
```
811663
```
In [6]:
```
#Indexing the Columns
df.columns
```
Out[6]:
```
Index(['DR_NO', 'Date Rptd', 'DATE OCC', 'TIME OCC', 'AREA', 'AREA NAME',
       'Rpt Dist No', 'Part 1-2', 'Crm Cd', 'Crm Cd Desc', 'Mocodes',
       'Vict Age', 'Vict Sex', 'Vict Descent', 'Premis Cd', 'Premis Desc',
       'Weapon Used Cd', 'Weapon Desc', 'Status', 'Status Desc', 'Crm Cd 1',
       'Crm Cd 2', 'Crm Cd 3', 'Crm Cd 4', 'LOCATION', 'Cross Street', 'LAT',
       'LON'],
      dtype='object')
```

Data Cleaning: After going through the data, we cleaned it up to then start performing analysis on it.

Data Cleaning: Identify and handle missing data appropriately.

```
In [7]: df.isna().sum()

Out[7]: DR_NO                 0
        Date Rptd             0
        DATE OCC              0
        TIME OCC              0
        AREA                  0
        AREA NAME             0
        Rpt Dist No           0
        Part 1-2              0
        Crm Cd                0
        Crm Cd Desc           0
        Mocodes          111367
        Vict Age              0
        Vict Sex         105909
        Vict Descent     105917
        Premis Cd             9
        Premis Desc         476
        Weapon Used Cd   526203
        Weapon Desc      526203
        Status                0
        Status Desc           0
        Crm Cd 1             10
        Crm Cd 2         747894
        Crm Cd 3         805390
        Crm Cd 4         807319
        LOCATION              0
        Cross Street     678145
        LAT                   0
        LON                   0
        dtype: int64
```

We analyze the importance of all missing values in each column in order to understand the significance.

After cleaning :

```
In [25]: df = df.drop_duplicates()
         df.shape

Out[25]: (808821, 33)

In [26]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 808821 entries, 0 to 811662
Data columns (total 33 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   DR_NO              808821 non-null  int64
 1   Date Rptd          808821 non-null  datetime64[ns]
 2   DATE OCC           808821 non-null  datetime64[ns]
 3   TIME OCC           808821 non-null  int64
 4   AREA               808821 non-null  int64
 5   AREA NAME          808821 non-null  object
 6   Rpt Dist No        808821 non-null  int64
 7   Part 1-2           808821 non-null  int64
 8   Crm Cd             808821 non-null  int64
 9   Crm Cd Desc        808821 non-null  object
 10  Mocodes            808821 non-null  object
 11  Vict Age           808821 non-null  int64
 12  Vict Sex           808821 non-null  object
 13  Vict Descent       808821 non-null  object
 14  Premis Cd          808821 non-null  float64
 15  Premis Desc        808821 non-null  object
 16  Weapon Used Cd     808821 non-null  float64
 17  Weapon Desc        808821 non-null  object
 18  Status             808821 non-null  object
 19  Status Desc        808821 non-null  object
 20  Crm Cd 1           808821 non-null  float64
 21  Crm Cd 2           808821 non-null  float64
 22  Crm Cd 3           808821 non-null  float64
 23  Crm Cd 4           808821 non-null  float64
 24  LOCATION           808821 non-null  object
 25  Cross Street       808821 non-null  object
 26  LAT                808821 non-null  float64
 27  LON                808821 non-null  float64
 28  DAY OF WEEK OCC    808821 non-null  object
 29  DAY OF WEEK RPTD   808821 non-null  object
 30  Year               808821 non-null  int64
 31  Month              808821 non-null  int64
 32  Hour               808821 non-null  int32
dtypes: datetime64[ns](2), float64(8), int32(1), int64(9), object(13)
memory usage: 206.7+ MB
```

Exploratory data analysis: After cleaning the data, we performed EDA on it to find insights, visualize, and see the statistics in the data. Also, answered questions that were asked of us.

Importing some libraries

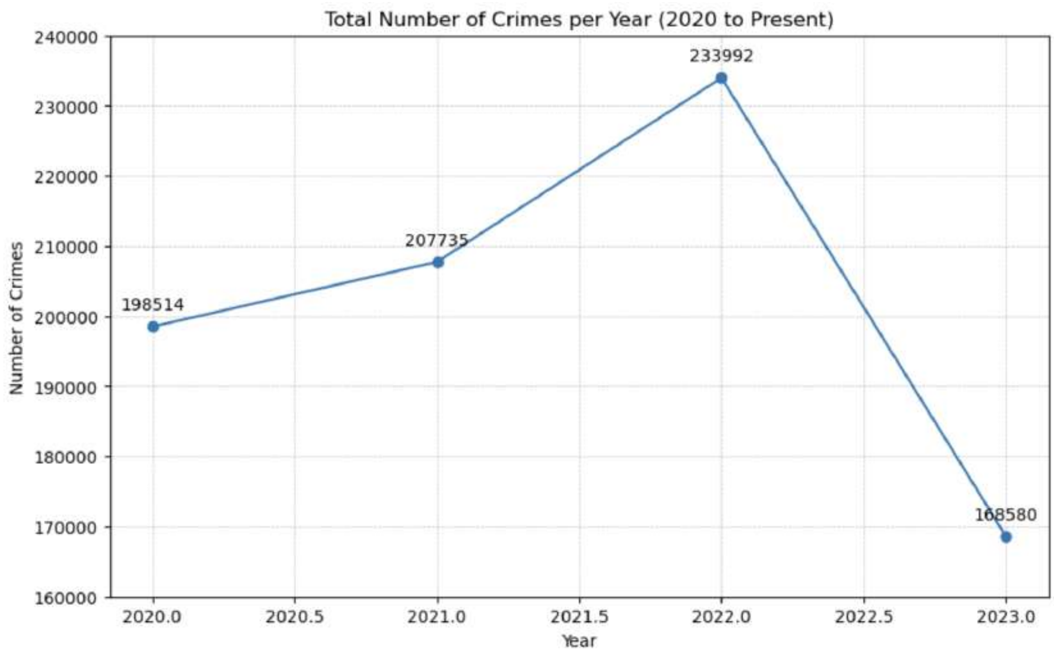
In [29]:
```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import calendar
import folium
from folium.plugins import HeatMap
import pandas as pd
```

Calculation and plotting of the total number of crimes per year to visualize the Overall crime trends:

In [28]:
```
duration = (df["DATE OCC"].max() - df["DATE OCC"].min()).days
print("There are {} crimes commited over {} days. On average, there are {} crimes each day.".format(len(df), duratio
```

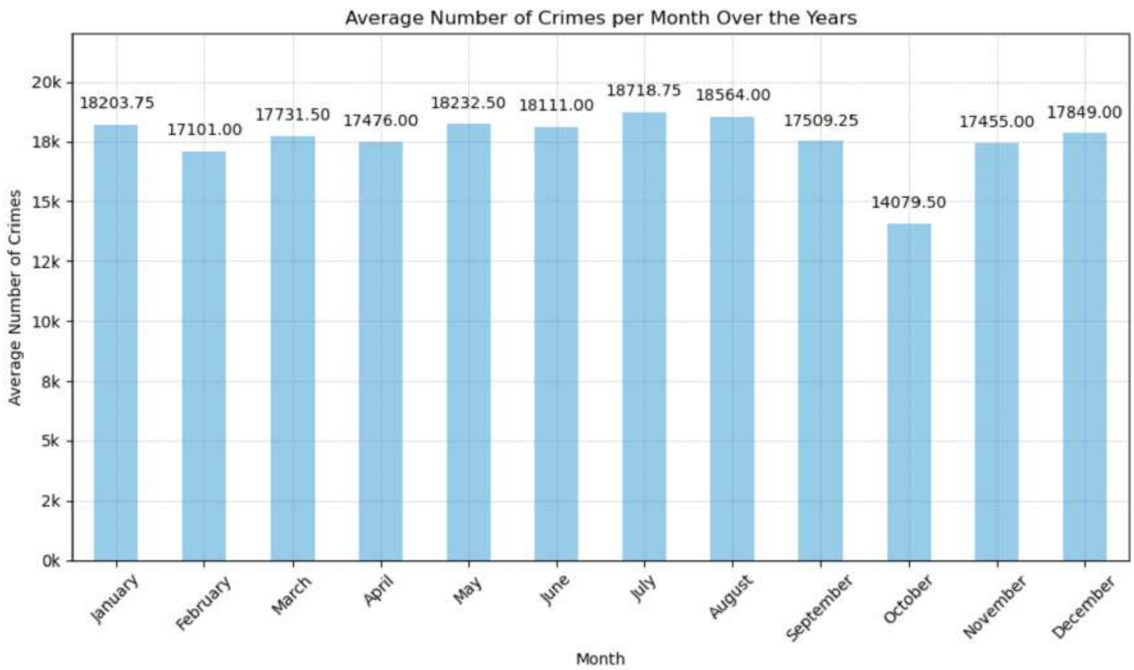
```
There are 808821 crimes commited over 1370 days. On average, there are 590 crimes each day.
```

In [29]:
```
# Group by year and count the number of crimes in each year
crime_counts_per_year = df['Year'].value_counts().sort_index()

# Plot the total number of crimes per year
plt.figure(figsize=(10, 6))
plt.plot(crime_counts_per_year.index, crime_counts_per_year.values, marker='o', linestyle='-')
plt.title('Total Number of Crimes per Year (2020 to Present)')
plt.xlabel('Year')
plt.ylabel('Number of Crimes')
plt.grid(color='gray', linestyle='--', linewidth=0.5, alpha=0.5)
plt.ylim(160000, 240000)

# Annotate each data point with the total number of crimes
for year, count in zip(crime_counts_per_year.index, crime_counts_per_year.values):
    plt.annotate(str(count), (year, count), textcoords="offset points", xytext=(0, 10), ha='center')

plt.show()
```



Finding the average number of crimes per month over the years to see the seasonal patterns:

In [30]:
```
from matplotlib.ticker import FuncFormatter
import calendar

# Group the data by year and month and calculate the average number of crimes for each month
average_crimes_per_month = df.groupby(['Year', 'Month']).size().groupby('Month').mean()

# Rename month numbers to month names
average_crimes_per_month.index = [calendar.month_name[i] for i in range(1, 13)]

# Plot the average number of crimes per month
plt.figure(figsize=(10, 6))
ax = average_crimes_per_month.plot(kind='bar', color='skyblue')
plt.title('Average Number of Crimes per Month Over the Years')
plt.grid(color='gray', linestyle='--', linewidth=0.5, alpha=0.5)
plt.xlabel('Month')
plt.ylabel('Average Number of Crimes')
plt.xticks(rotation=45)
plt.ylim(0, 22000)
def format_thousands(x, pos):
    return f'{x/1000:.0f}k'

formatter = FuncFormatter(format_thousands)
ax.yaxis.set_major_formatter(formatter)

for i, count in enumerate(average_crimes_per_month):
    plt.annotate(f'{count:.2f}', (i, count), textcoords="offset points", xytext=(0, 10), ha='center')


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



Counting the occurrences of each crime type and identify the one with the highest frequency:

```
In [34]: unique_crime_type = df['Crm Cd Desc'].unique()
         unique_crime_type
```

```
Out[34]: array(['BATTERY - SIMPLE ASSAULT',
       'SEX OFFENDER REGISTRANT OUT OF COMPLIANCE',
       'VANDALISM - MISDEAMEANOR ($399 OR UNDER)',
       'VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)',
       'RAPE, FORCIBLE', 'SHOPLIFTING - PETTY THEFT ($950 & UNDER)',
       'OTHER MISCELLANEOUS CRIME',
       'THEFT-GRAND ($950.01 & OVER)EXCPT,GUNS,FOWL,LIVESTK,PROD',
       'BURGLARY FROM VEHICLE', 'CRIMINAL THREATS - NO WEAPON DISPLAYED',
       'ARSON', 'INTIMATE PARTNER - SIMPLE ASSAULT',
       'THEFT PLAIN - PETTY ($950 & UNDER)', 'THEFT OF IDENTITY',
       'ROBBERY', 'ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT',
       'BURGLARY', 'VEHICLE - STOLEN',
       'THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)',
       'BRANDISH WEAPON', 'INTIMATE PARTNER - AGGRAVATED ASSAULT',
       'BUNCO, GRAND THEFT', 'THEFT, PERSON',
       'BATTERY WITH SEXUAL CONTACT', 'BIKE - STOLEN',
       'BATTERY POLICE (SIMPLE)',
       'LETTERS, LEWD  -  TELEPHONE CALLS, LEWD',
       'VIOLATION OF COURT ORDER', 'TRESPASSING',
       'THEFT FROM MOTOR VEHICLE - GRAND ($950.01 AND OVER)',
       'VIOLATION OF RESTRAINING ORDER', 'DISTURBING THE PEACE',
       'THEFT FROM MOTOR VEHICLE - ATTEMPT',
       'THROWING OBJECT AT MOVING VEHICLE', 'EXTORTION',
       'SEX,UNLAWFUL(INC MUTUAL CONSENT, PENETRATION W/ FRGN OBJ',
       'CHILD STEALING',
       'CRM AGNST CHLD (13 OR UNDER) (14-15 & SUSP 10 YRS OLDER)',
       'ATTEMPTED ROBBERY', 'OTHER ASSAULT', 'BOMB SCARE',
       'DOCUMENT FORGERY / STOLEN FELONY',
       'SEXUAL PENETRATION W/FOREIGN OBJECT',
       'SHOTS FIRED AT INHABITED DWELLING', 'BURGLARY, ATTEMPTED',
       'FAILURE TO YIELD', 'PURSE SNATCHING', 'INDECENT EXPOSURE',
       'ORAL COPULATION', 'EMBEZZLEMENT, GRAND THEFT ($950.01 & OVER)',
       'VIOLATION OF TEMPORARY RESTRAINING ORDER', 'BUNCO, PETTY THEFT',
       'KIDNAPPING - GRAND ATTEMPT',
       'SHOPLIFTING-GRAND THEFT ($950.01 & OVER)', 'RESISTING ARREST',
       'DISCHARGE FIREARMS/SHOTS FIRED',
       'THREATENING PHONE CALLS/LETTERS', 'KIDNAPPING',
       'LEWD/LASCIVIOUS ACTS WITH CHILD', 'LEWD CONDUCT',
       'UNAUTHORIZED COMPUTER ACCESS',
       'SODOMY/SEXUAL CONTACT B/W PENIS OF ONE PERS TO ANUS OTH',
       'CHILD NEGLECT (SEE 300 W.I.C.)', 'CONTEMPT OF COURT',
       'CHILD ANNOYING (17YRS & UNDER)', 'BUNCO, ATTEMPT',
       'CHILD ABUSE (PHYSICAL) - SIMPLE ASSAULT', 'PIMPING', 'STALKING',
       'THEFT PLAIN - ATTEMPT', 'RAPE, ATTEMPTED',
       'SHOPLIFTING - ATTEMPT', 'THEFT FROM PERSON - ATTEMPT',
       'VEHICLE - ATTEMPT STOLEN', 'FALSE IMPRISONMENT',
       'BURGLARY FROM VEHICLE, ATTEMPTED', 'PICKPOCKET',
       'EMBEZZLEMENT, PETTY THEFT ($950 & UNDER)',
       'DEFRAUDING INNKEEPER/THEFT OF SERVICES, $950 & UNDER',
       'COUNTERFEIT', 'CREDIT CARDS, FRAUD USE ($950 & UNDER',
       'SHOTS FIRED AT MOVING VEHICLE, TRAIN OR AIRCRAFT',
       'CRIMINAL HOMICIDE', 'DOCUMENT WORTHLESS ($200 & UNDER)',
       'PROWLER', 'DEFRAUDING INNKEEPER/THEFT OF SERVICES, OVER $950.01',
       'ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER',
       'DISHONEST EMPLOYEE - GRAND THEFT',
       'HUMAN TRAFFICKING - COMMERCIAL SEX ACTS', 'CHILD PORNOGRAPHY',
       'PEEPING TOM', 'BATTERY ON A FIREFIGHTER',
       'TILL TAP - PETTY ($950 & UNDER)',
       'CHILD ABUSE (PHYSICAL) - AGGRAVATED ASSAULT',
       'TILL TAP - GRAND THEFT ($950.01 & OVER)',
       'HUMAN TRAFFICKING - INVOLUNTARY SERVITUDE',
       'FIREARMS RESTRAINING ORDER (FIREARMS RO)',
       'DRIVING WITHOUT OWNER CONSENT (DWOC)',
       'DOCUMENT WORTHLESS ($200.01 & OVER)', 'PANDERING',
       'CRUELTY TO ANIMALS', 'CREDIT CARDS, FRAUD USE ($950.01 & OVER)',
```

```
In [32]: crimes_per_type_total = df['Crm Cd Desc'].value_counts().reset_index()
         crimes_per_type_total=pd.DataFrame(crimes_per_type_total)
         crimes_per_type_total.columns=['CRIME', 'NUMBER OF CRIMES']
         crimes_per_type_total.head(5)
```

Out[32]:

	CRIME	NUMBER OF CRIMES
0	VEHICLE - STOLEN	86748
1	BATTERY - SIMPLE ASSAULT	64204
2	THEFT OF IDENTITY	51494
3	BURGLARY FROM VEHICLE	49735
4	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA...	49443

```
In [33]: total_count_tp = crimes_per_type_total['NUMBER OF CRIMES'].sum()
         total_count_tp
```

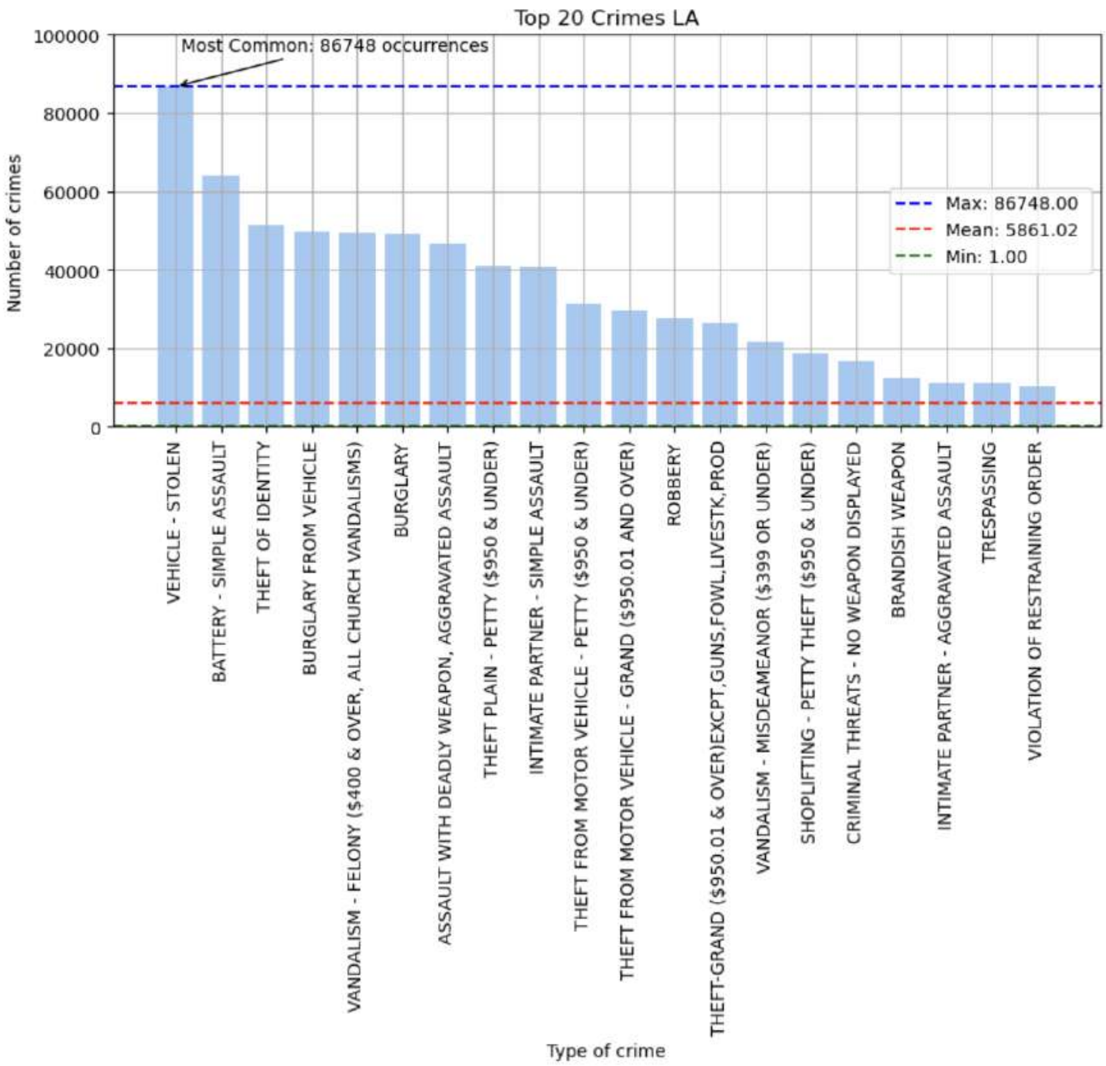
Out[33]: 808821

```
In [34]: crimes_per_type_total['NUMBER OF CRIMES'] = crimes_per_type_total['NUMBER OF CRIMES'].astype(int)
         crimes_per_type_total['%']=round((crimes_per_type_total['NUMBER OF CRIMES']/total_count_tp)*100,2)
         crimes_per_type_total.head(5)
```

Out[34]:

	CRIME	NUMBER OF CRIMES	%
0	VEHICLE - STOLEN	86748	10.73
1	BATTERY - SIMPLE ASSAULT	64204	7.94
2	THEFT OF IDENTITY	51494	6.37
3	BURGLARY FROM VEHICLE	49735	6.15
4	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA...	49443	6.11

We picked to plot the 20 most common crimes because those represent more than the 80% of the crimes comitted



Grouping the data by region or city and compare crime rates between them using descriptive statistics or visualizations:

In [37]:
```
unique_area_names = df['AREA NAME'].unique()
```

In [38]:
```
unique_area_names
```

Out[38]:
```
array(['Southwest', 'Central', 'N Hollywood', 'Mission', 'Devonshire',
       'Northeast', 'Harbor', 'Van Nuys', 'West Valley', 'West LA',
       'Wilshire', 'Pacific', 'Rampart', '77th Street', 'Hollenbeck',
       'Southeast', 'Hollywood', 'Newton', 'Topanga', 'Foothill',
       'Olympic'], dtype=object)
```

In [39]:
```
crimes_per_area_total = df['AREA NAME'].value_counts().reset_index()
crimes_per_area_total=pd.DataFrame(crimes_per_area_total)
crimes_per_area_total.columns=['AREA NAME', 'NUMBER OF CRIMES']
crimes_per_area_total.head(5)
```

Out[39]:

	AREA NAME	NUMBER OF CRIMES
0	Central	54335
1	77th Street	51130
2	Pacific	47282
3	Southwest	45291
4	Hollywood	42702

In [40]:
```
total_count = crimes_per_area_total['NUMBER OF CRIMES'].sum()
total_count
```

Out[40]:
```
808821
```

```
In [40]: total_count = crimes_per_area_total['NUMBER OF CRIMES'].sum()
         total_count
```

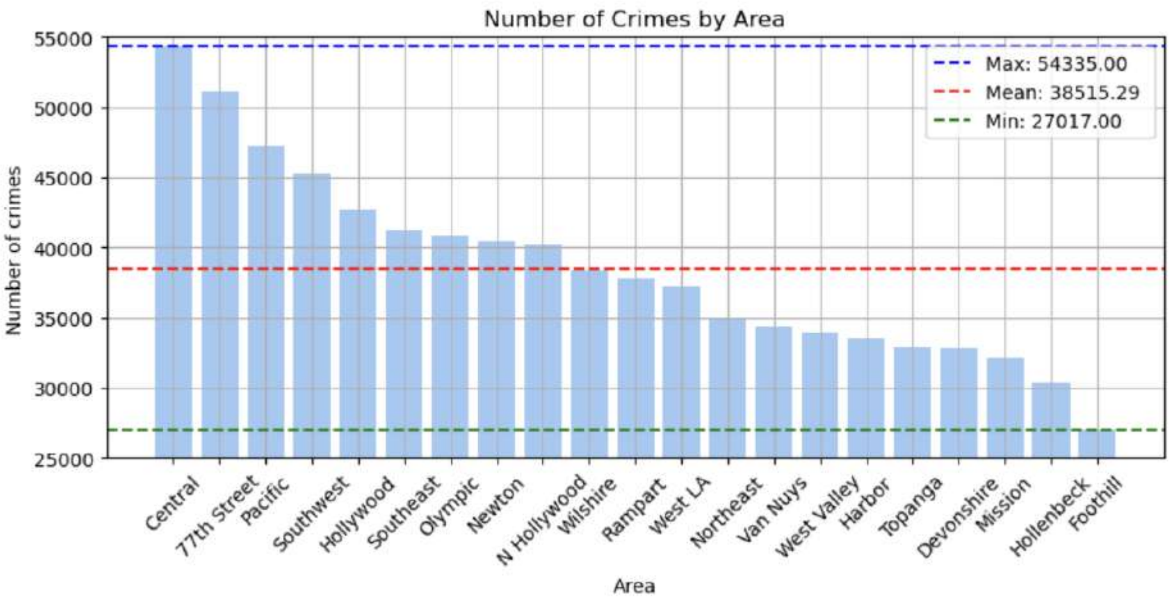
```
Out[40]: 808821
```

```
In [41]: crimes_per_area_total['NUMBER OF CRIMES'] = crimes_per_area_total['NUMBER OF CRIMES'].astype(int)
         crimes_per_area_total['%']=round((crimes_per_area_total['NUMBER OF CRIMES']/total_count)*100,2)
```

```
In [42]: crimes_per_area_total
```

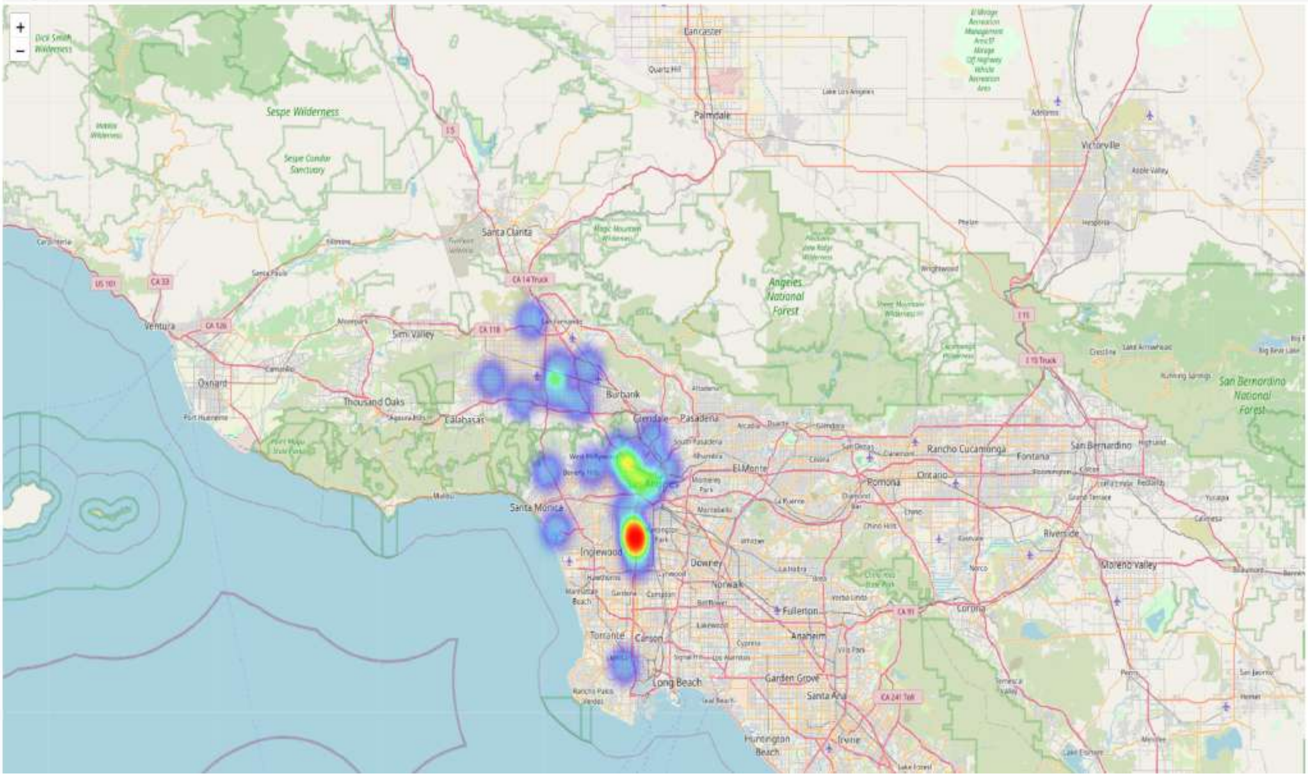
Out[42]:

	AREA NAME	NUMBER OF CRIMES	%
0	Central	54335	6.72
1	77th Street	51130	6.32
2	Pacific	47282	5.85
3	Southwest	45291	5.60
4	Hollywood	42702	5.28
5	Southeast	41217	5.10
6	Olympic	40784	5.04
7	Newton	40396	4.99
8	N Hollywood	40215	4.97
9	Wilshire	38339	4.74
10	Rampart	37797	4.67
11	West LA	37199	4.60
12	Northeast	35004	4.33
13	Van Nuys	34360	4.25
14	West Valley	33947	4.20
15	Harbor	33535	4.15
16	Topanga	32900	4.07
17	Devonshire	32815	4.06
18	Mission	32171	3.98
19	Hollenbeck	30385	3.76
20	Foothill	27017	3.34



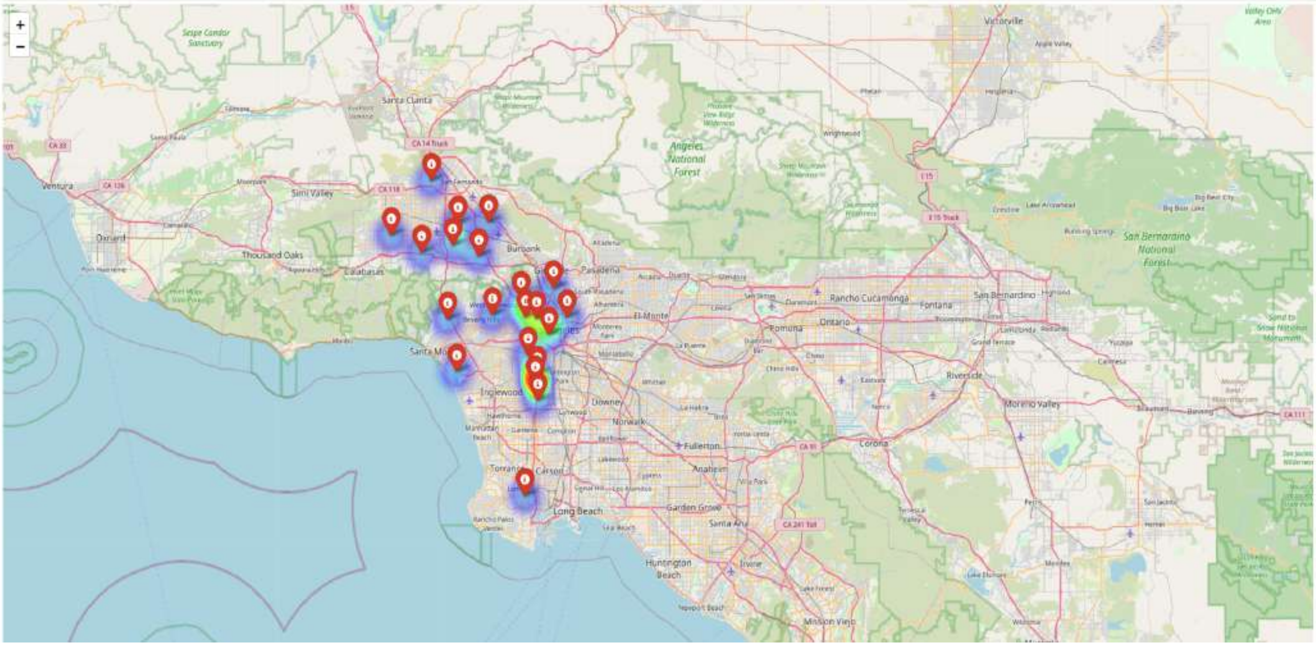
In [50]:
```python
crime_map = folium.Map(location=[34.0522, -118.2437], zoom_start=10)

heatmap_data = []
for region_name, count in crime_counts_by_region.items():
    region_df = df[df['AREA NAME'] == region_name]
    region_lat = region_df['LAT'].iloc[0]
    region_lon = region_df['LON'].iloc[0]
    heatmap_data.append([region_lat, region_lon, count])

folium.plugins.HeatMap(heatmap_data).add_to(crime_map)

crime_map
```

Out[50]:



In [51]:
```python
crime_counts_by_region = df.groupby('AREA NAME')['DR_NO'].count()
crime_counts_by_region
for region_name, count in crime_counts_by_region.items():
    region_df = df[df['AREA NAME'] == region_name]
    region_lat = region_df['LAT'].iloc[0]
    region_lon = region_df['LON'].iloc[0]

    tooltip = f"{region_name}: {count} crimes"

    folium.Marker(
        location=[region_lat, region_lon],
        popup=region_name,
        tooltip=tooltip,
        icon=folium.Icon(color='red', icon_color='white'),
    ).add_to(crime_map)

crime_map
```

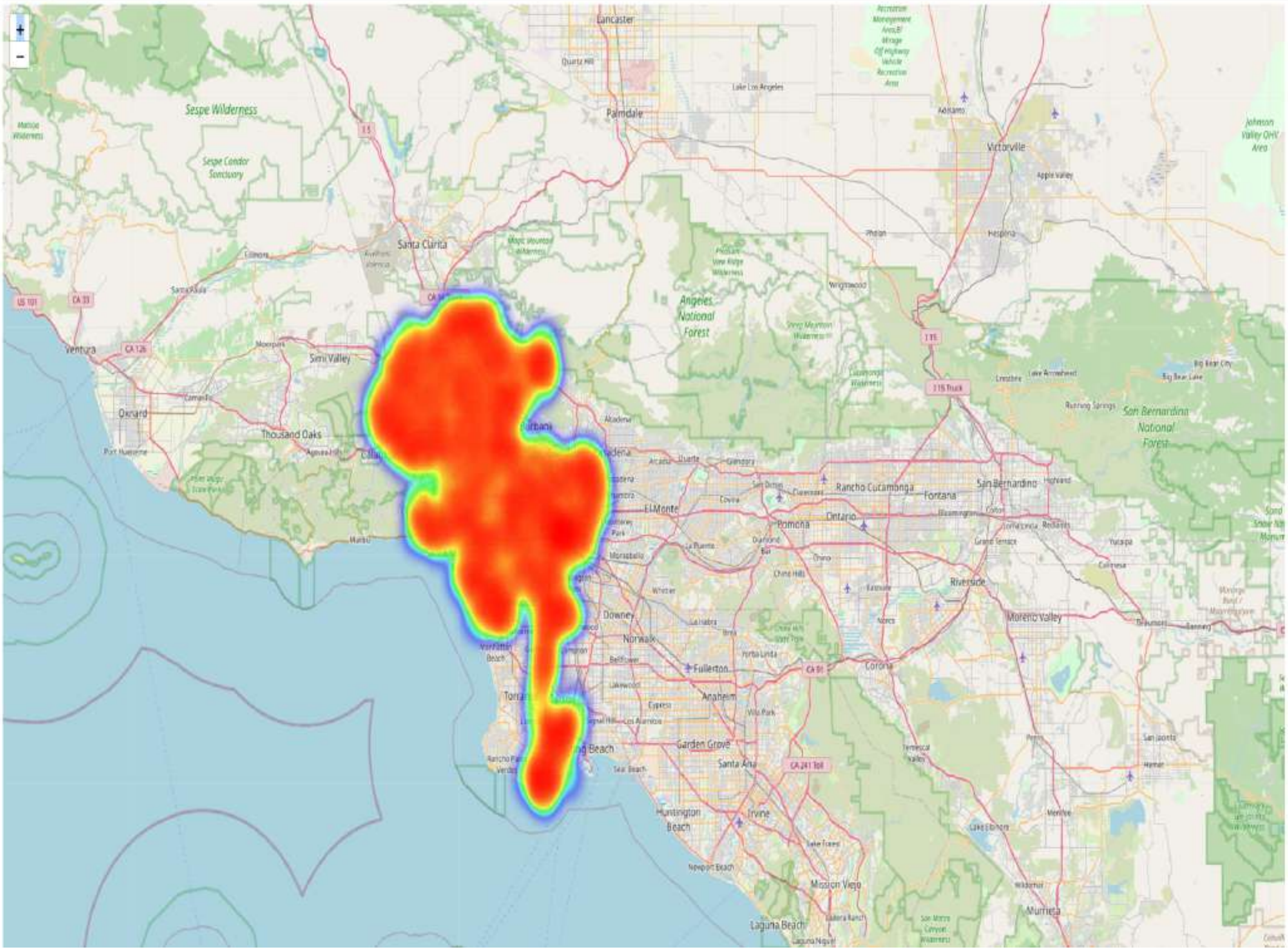
Out[51]:



In [52]:

```python
crime_map = folium.Map(location=[34.0522, -118.2437], zoom_start=10)

heatmap_data = df[['LAT', 'LON']].values.tolist()
HeatMap(heatmap_data).add_to(crime_map)

crime_map
```

Out[52]:



Collecting economic data for the same time frame and using statistical methods like correlation analysis to assess the relationship between economic factors and crime rates:

We gather all the information from the next websites https://data.census.gov/ https://censusreporter.org/ .This inoformation represent the economical analysis done by the US Census Bureau during the years 2020 and 2021. We compiled this information by police station because even though each crime had different locations those where the areas where those were reported.

In [53]:
```
df.head()
```

Out[53]:

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	...	Crm Cd 4	LOCATION	Cross Street	LAT	LON	DAY OF WEEK OCC	DAY OF WEEK RPTD	Year	Month	Hour
0	10304468	2020-01-08	2020-01-08	2230	3	Southwest	377	2	624	BATTERY - SIMPLE ASSAULT	...	0.0	1100 W 39TH PL		34.0141	-118.2978	Wednesday	Wednesday	2020	1	22
1	190101086	2020-01-02	2020-01-01	330	1	Central	163	2	624	BATTERY - SIMPLE ASSAULT	...	0.0	700 S HILL ST		34.0459	-118.2545	Wednesday	Thursday	2020	1	3
2	200110444	2020-04-14	2020-02-13	1200	1	Central	155	2	845	SEX OFFENDER REGISTRANT OUT OF COMPLIANCE	...	0.0	200 E 6TH ST		34.0448	-118.2474	Thursday	Tuesday	2020	2	12
3	191501505	2020-01-01	2020-01-01	1730	15	N Hollywood	1543	2	745	VANDALISM - MISDEAMEANOR ($399 OR UNDER)	...	0.0	5400 CORTEEN PL		34.1685	-118.4019	Wednesday	Wednesday	2020	1	17
4	191921269	2020-01-01	2020-01-01	415	19	Mission	1998	2	740	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA...	...	0.0	14400 TITUS ST		34.2198	-118.4468	Wednesday	Wednesday	2020	1	4

5 rows × 33 columns

In [54]:
```
area_data = {
    '77th Street': {
        'DESCRIPTION': '7600 S Broadway, Los Angeles, CA 90003',
        'ZIP CODE': 90003,
        'Population': 72764,
        'Employment %': 55.10,
        'Median Household Income': 47733.00,
        'Housing': 18244,
        'Education %': 7.30,
        'Median value of owner-occupied housing units': 430800
    },
    'Olympic': {
        'DESCRIPTION': '1130 Vermont Ave, Los Angeles, CA 90006',
        'ZIP CODE': 90006,
        'Population': 58229,
        'Employment %': 63.00,
        'Median Household Income': 41068.00,
        'Housing': 21425,
        'Education %': 20.40,
        'Median value of owner-occupied housing units': 716900
    },
    'Newton': {
        'DESCRIPTION': '3400 South Central Ave. Los Angeles, CA 90011',
        'ZIP CODE': 90011,
        'Population': 102308,
        'Employment %': 58.70,
        'Median Household Income': 47126.00,
        'Housing': 24348,
        'Education %': 6.90,
        'Median value of owner-occupied housing units': 452100
    },
    'Central': {
        'DESCRIPTION': '251 E 6th St, Los Angeles, CA 90014',
        'ZIP CODE': 90014,
        'Population': 9254,
        'Employment %': 55.80,
        'Median Household Income': 31332.00,
        'Housing': 6788,
        'Education %': 42.40,
        'Median value of owner-occupied housing units': 625000
    },
    'Rampart': {
        'DESCRIPTION': '1401 W 6th St, Los Angeles, CA 90017',
        'ZIP CODE': 90017,
        'Population': 27295,
        'Employment %': 65.40,
        'Median Household Income': 44607.00,
        'Housing': 15191,
        'Education %': 34.00,
        'Median value of owner-occupied housing units': 697100
    },
    'Wilshire': {
        'DESCRIPTION': '4861 Venice Blvd., Los Angeles, CA 90019',
        'ZIP CODE': 90019,
        'Population': 62002,
        'Employment %': 61.90,
        'Median Household Income': 61616.00,
        'Housing': 25266,
        'Education %': 35.10,
        'Median value of owner-occupied housing units': 1033900
    },
    'West LA': {
        'DESCRIPTION': '1663 Butler Ave, Los Angeles, CA 90025',
        'ZIP CODE': 90025,
        'Population': 45466,
        'Employment %': 70.90,
        'Median Household Income': 100453.00,
        'Housing': 24402,
        'Education %': 71.00,
        'Median value of owner-occupied housing units': 905900
    },
    'Hollywood': {
        'DESCRIPTION': '1358 Wilcox Ave, Los Angeles, CA 90028',
        'ZIP CODE': 90028,
        'Population': 32330,
        'Employment %': 66.30,
        'Median Household Income': 52814.00,
        'Housing': 20700,
        'Education %': 51.30,
        'Median value of owner-occupied housing units': 723000
    },
    'Hollenbeck': {
        'DESCRIPTION': '2111 1st St, Los Angeles, CA 90033',
        'ZIP CODE': 90033,
        'Population': 46081,
        'Employment %': 55.60,
        'Median Household Income': 49734.00,
        'Housing': 13843,
        'Education %': 13.50,
        'Median value of owner-occupied housing units': 538900
    },
```

```
    'Southeast': {
        'DESCRIPTION': '145 W 108th St, Los Angeles, CA 90061',
        'ZIP CODE': 90061,
        'Population': 29570,
        'Employment %': 55.80,
        'Median Household Income': 50427.00,
        'Housing': 8116,
        'Education %': 11.70,
        'Median value of owner-occupied housing units': 438800
    },
    'Southwest': {
        'DESCRIPTION': '1546 W Martin Luther King Jr Blvd, Los Angeles, CA 90062',
        'ZIP CODE': 90062,
        'Population': 32524,
        'Employment %': 59.30,
        'Median Household Income': 56500.00,
        'Housing': 9839,
        'Education %': 15.50,
        'Median value of owner-occupied housing units': 550600
    },
    'Northeast': {
        'DESCRIPTION': '3353 N San Fernando Rd, Los Angeles, CA 90065',
        'ZIP CODE': 90065,
        'Population': 44328,
        'Employment %': 64.90,
        'Median Household Income': 80386.00,
        'Housing': 16373,
        'Education %': 38.60,
        'Median value of owner-occupied housing units': 833400
    },
    'Pacific': {
        'DESCRIPTION': '12312 Culver Blvd, Los Angeles, CA 90066',
        'ZIP CODE': 90066,
        'Population': 55304,
        'Employment %': 67.10,
        'Median Household Income': 90983.00,
        'Housing': 25186,
        'Education %': 57.40,
        'Median value of owner-occupied housing units': 1282100
    },
    'Harbor': {
        'DESCRIPTION': '2175 John S Gibson Blvd, Los Angeles, CA 90731',
        'ZIP CODE': 90731,
        'Population': 61270,
        'Employment %': 59.90,
        'Median Household Income': 61144.00,
        'Housing': 24189,
        'Education %': 23.30,
        'Median value of owner-occupied housing units': 657700
    },
    'Topanga': {
        'DESCRIPTION': '21501 Schoenborn St, Canoga Park, CA 91304',
        'ZIP CODE': 91304,
        'Population': 52386,
        'Employment %': 62.80,
        'Median Household Income': 74987.00,
        'Housing': 17963,
        'Education %': 34.40,
        'Median value of owner-occupied housing units': 677700
    },
    'Devonshire': {
        'DESCRIPTION': '10250 Etiwanda Ave, Northridge, CA 91324',
        'ZIP CODE': 91324,
        'Population': 29500,
        'Employment %': 61.70,
        'Median Household Income': 88003.00,
        'Housing': 10998,
        'Education %': 40.90,
        'Median value of owner-occupied housing units': 680200
    },
    'Foothill': {
        'DESCRIPTION': '12760 Osborne St, Pacoima, CA 91331',
        'ZIP CODE': 91331,
        'Population': 100720,
        'Employment %': 58.60,
        'Median Household Income': 72089.00,
        'Housing': 23996,
        'Education %': 10.50,
        'Median value of owner-occupied housing units': 487600
    },
    'West Valley': {
        'DESCRIPTION': '19020 Vanowen St, Reseda, CA 91335',
        'ZIP CODE': 91335,
        'Population': 76650,
        'Employment %': 62.30,
        'Median Household Income': 60163.00,
        'Housing': 24819,
        'Education %': 29.20,
        'Median value of owner-occupied housing units': 580500
    },
```

In [56]:
```
area_data=pd.DataFrame(area_data).T
area_data
```

Out[56]:

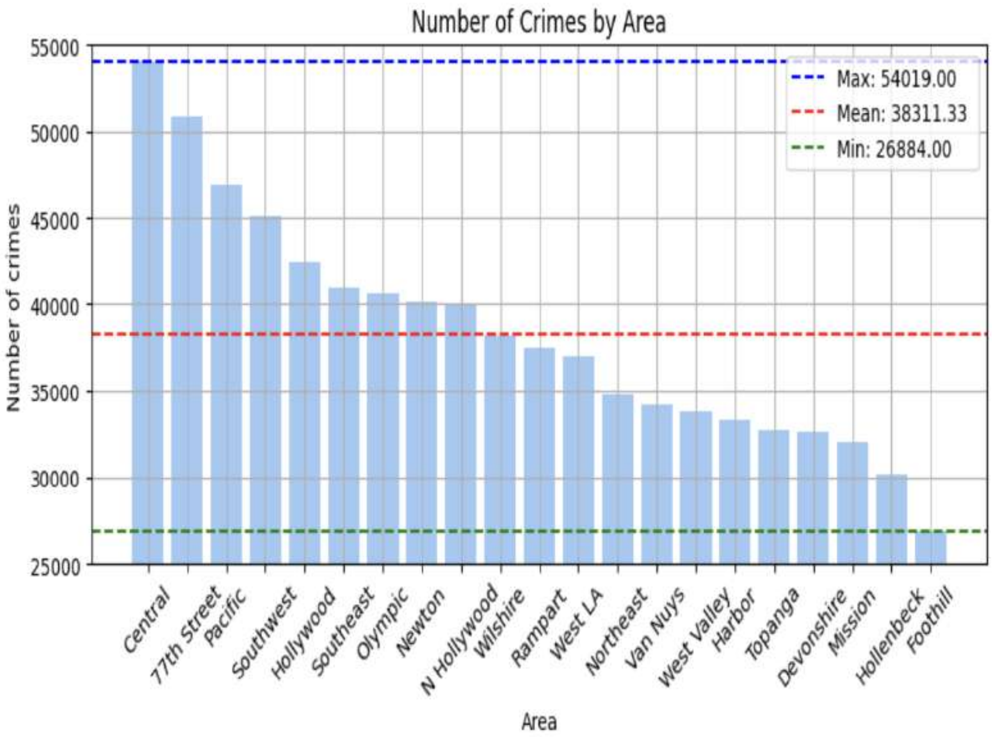
	DESCRIPTION	ZIP CODE	Population	Employment %	Median Household Income	Housing	Education %	Median value of owner-occupied housing units
77th Street	7600 S Broadway, Los Angeles, CA 90003	90003	72764	55.1	47733.0	18244	7.3	430800
Olympic	1130 Vermont Ave, Los Angeles, CA 90006	90006	58229	63.0	41068.0	21425	20.4	716900
Newton	3400 South Central Ave. Los Angeles, CA 90011	90011	102308	58.7	47126.0	24348	6.9	452100
Central	251 E 6th St, Los Angeles, CA 90014	90014	9254	55.8	31332.0	6788	42.4	625000
Rampart	1401 W 6th St, Los Angeles, CA 90017	90017	27295	65.4	44607.0	15191	34.0	697100
Wilshire	4861 Venice Blvd., Los Angeles, CA 90019	90019	62002	61.9	61616.0	25266	35.1	1033900
West LA	1663 Butler Ave, Los Angeles, CA 90025	90025	45466	70.9	100453.0	24402	71.0	905900
Hollywood	1358 Wilcox Ave, Los Angeles, CA 90028	90028	32330	66.3	52814.0	20700	51.3	723000
Hollenbeck	2111 1st St, Los Angeles, CA 90033	90033	46081	55.6	49734.0	13843	13.5	538900
Southeast	145 W 108th St, Los Angeles, CA 90061	90061	29570	55.8	50427.0	8116	11.7	438800
Southwest	1546 W Martin Luther King Jr Blvd, Los Angeles...	90062	32524	59.3	56500.0	9839	15.5	550600
Northeast	3353 N San Fernando Rd, Los Angeles, CA 90065	90065	44328	64.9	80386.0	16373	38.6	833400
Pacific	12312 Culver Blvd, Los Angeles, CA 90066	90066	55304	67.1	90983.0	25186	57.4	1282100
Harbor	2175 John S Gibson Blvd, Los Angeles, CA 90731	90731	61270	59.9	61144.0	24189	23.3	657700
Topanga	21501 Schoenborn St, Canoga Park, CA 91304	91304	52386	62.8	74987.0	17963	34.4	677700
Devonshire	10250 Etiwanda Ave, Northridge, CA 91324	91324	29500	61.7	88003.0	10998	40.9	680200
Foothill	12760 Osborne St, Pacoima, CA 91331	91331	100720	58.6	72089.0	23996	10.5	487600
West Valley	19020 Vanowen St, Reseda, CA 91335	91335	76650	62.3	68163.0	24819	29.2	580500
Mission	11121 Sepulveda Blvd, Mission Hills, CA 91345	91345	18895	60.8	85659.0	5541	21.9	558000
Van Nuys	6240 Sylmar Ave, Van Nuys, CA 91401	91401	39621	64.5	65458.0	15437	38.9	798900
N Hollywood	11640 Burbank Blvd, North Hollywood, CA 91601	91601	39429	65.6	61761.0	19476	46.0	744500

In [57]:
```
merged_df = crimes_per_area_total.merge(area_data, left_on='AREA NAME', right_index=True)
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 21 entries, 0 to 20
Data columns (total 11 columns):
 #   Column                                        Non-Null Count  Dtype  
---  ------                                        --------------  -----  
 0   AREA NAME                                     21 non-null     object 
 1   NUMBER OF CRIMES                              21 non-null     int32  
 2   %                                             21 non-null     float64
 3   DESCRIPTION                                   21 non-null     object 
 4   ZIP CODE                                      21 non-null     object 
 5   Population                                    21 non-null     object 
 6   Employment %                                  21 non-null     object 
 7   Median Household Income                       21 non-null     object 
 8   Housing                                       21 non-null     object 
 9   Education %                                   21 non-null     object 
 10  Median value of owner-occupied housing units  21 non-null     object 
dtypes: float64(1), int32(1), object(9)
memory usage: 1.9+ KB
```

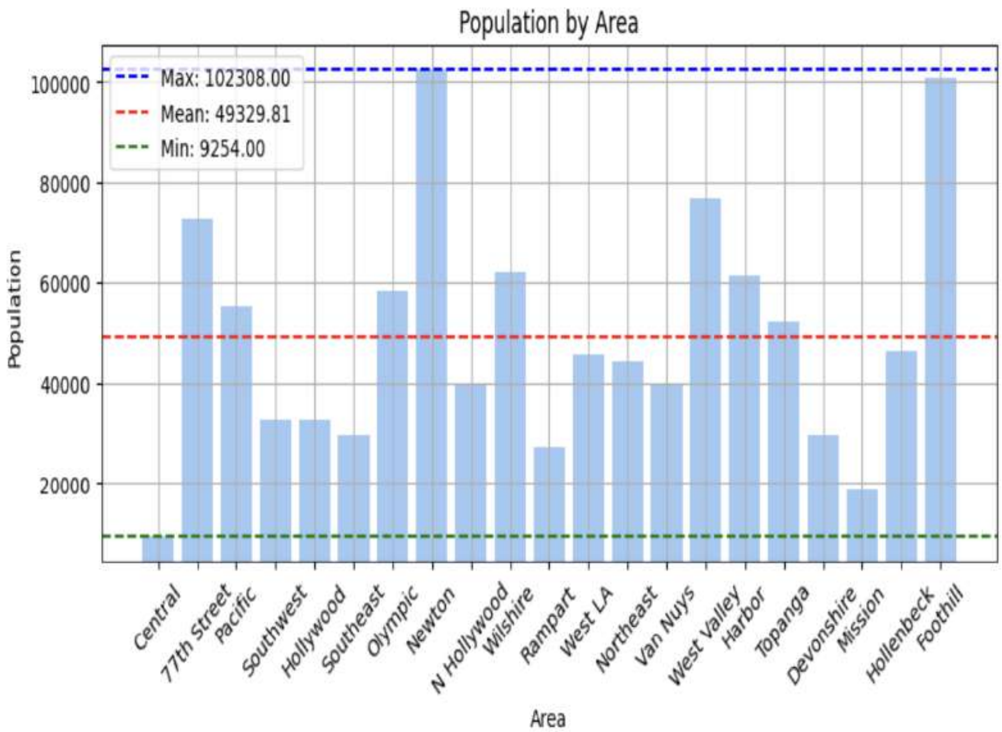
Crimes by Area

In [60]:
```
plt.figure(figsize=(10, 4))
sns.set_palette("pastel")
plt.bar(merged_df['AREA NAME'], merged_df['NUMBER OF CRIMES'])

max_crm = merged_df['NUMBER OF CRIMES'].max()
plt.axhline(max_crm, color='blue', linestyle='--', label=f'Max: {max_crm:.2f} ')

mean_crm = merged_df['NUMBER OF CRIMES'].mean()
plt.axhline(mean_crm, color='red', linestyle='--', label=f'Mean: {mean_crm:.2f} ')

min_crm = merged_df['NUMBER OF CRIMES'].min()
plt.axhline(min_crm, color='green', linestyle='--', label=f'Min: {min_crm:.2f} ')

plt.title('Number of Crimes by Area Name')
plt.grid()
plt.xlabel('Area')
plt.ylabel('Number of crimes')
plt.ylim(25000, 55000)
plt.title('Number of Crimes by Area')

plt.legend()
plt.xticks(rotation=45)
plt.show()
```



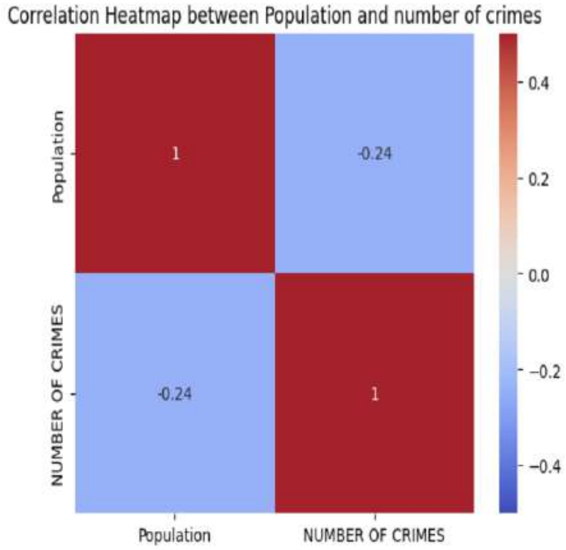
Population by Area

In [61]:
```
plt.figure(figsize=(10, 4))
sns.set_palette("pastel")
plt.bar(merged_df['AREA NAME'], merged_df['Population'])

max_pop = merged_df['Population'].max()
plt.axhline(max_pop, color='blue', linestyle='--', label=f'Max: {max_pop:.2f} ')

mean_pop = merged_df['Population'].mean()
plt.axhline(mean_pop, color='red', linestyle='--', label=f'Mean: {mean_pop:.2f} ')

min_pop = merged_df['Population'].min()
plt.axhline(min_pop, color='green', linestyle='--', label=f'Min: {min_pop:.2f} ')

plt.title('Number of Crimes by Area Name')
plt.grid()
plt.xlabel('Area')
plt.ylabel('Population')
plt.ylim(merged_df['Population'].min()-5000, merged_df['Population'].max()+5000)
plt.title('Population by Area')

plt.legend()
plt.xticks(rotation=45)
plt.show()
```

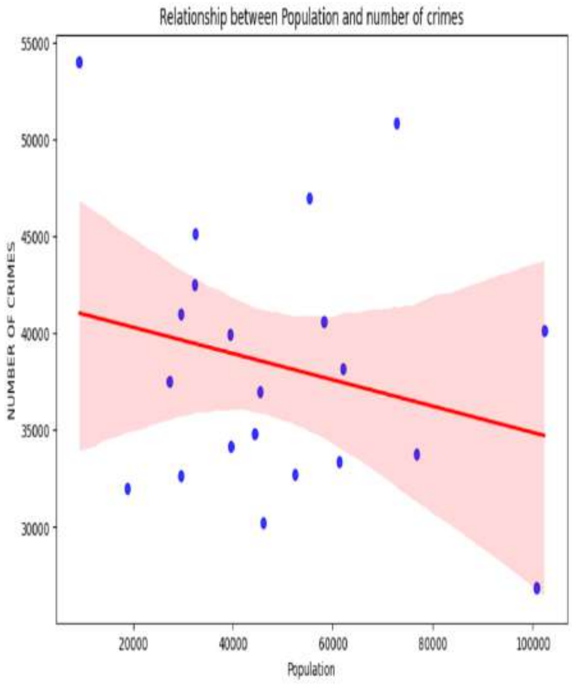


Relationship Population and number of crimes

In [62]:
```
correlation = merged_df[['Population', 'NUMBER OF CRIMES']].corr()
sns.heatmap(correlation, annot=True, cmap='coolwarm', vmin=-0.5, vmax=0.5)
plt.title('Correlation Heatmap between Population and number of crimes')

plt.show()
```



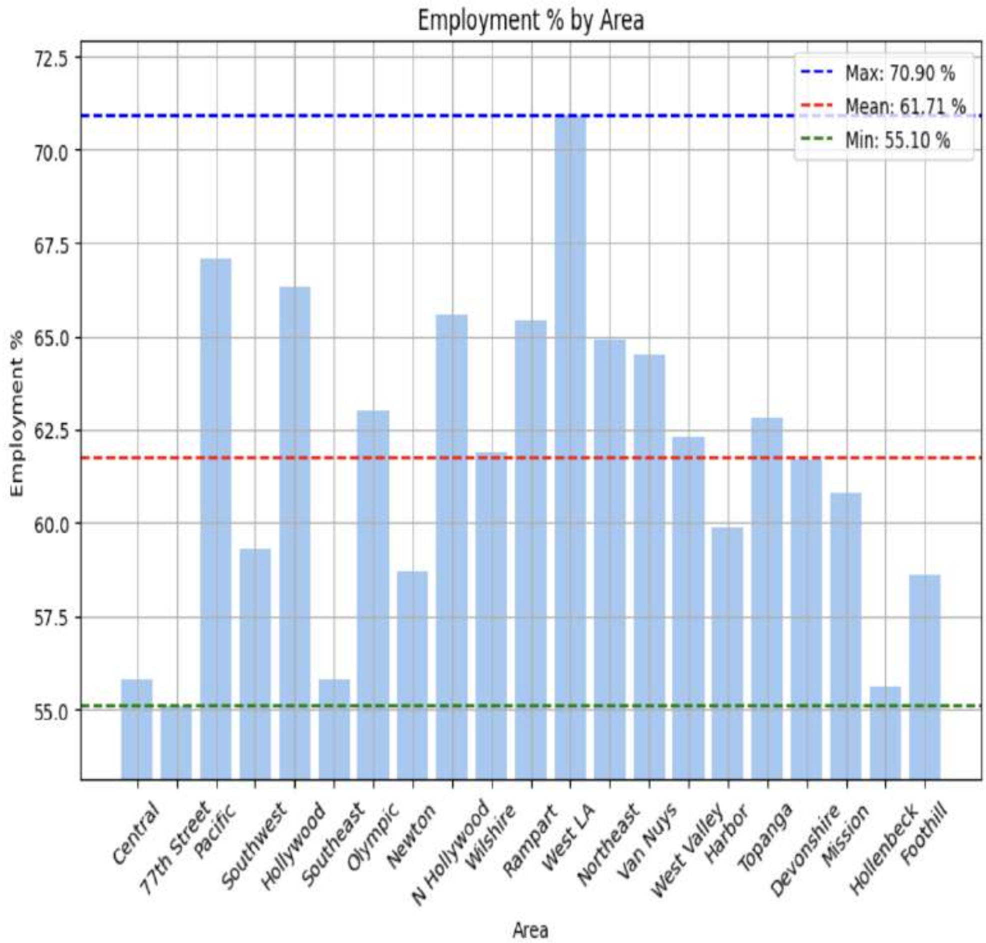
In [63]:
```
plt.figure(figsize=(10, 6))
sns.regplot(x='Population', y='NUMBER OF CRIMES', data=merged_df, scatter_kws={'color': 'blue'}, line_kws={'color': 'red'})
plt.title('Relationship between Population and number of crimes')
plt.xlabel('Population')
plt.ylabel('NUMBER OF CRIMES')
plt.show()
```

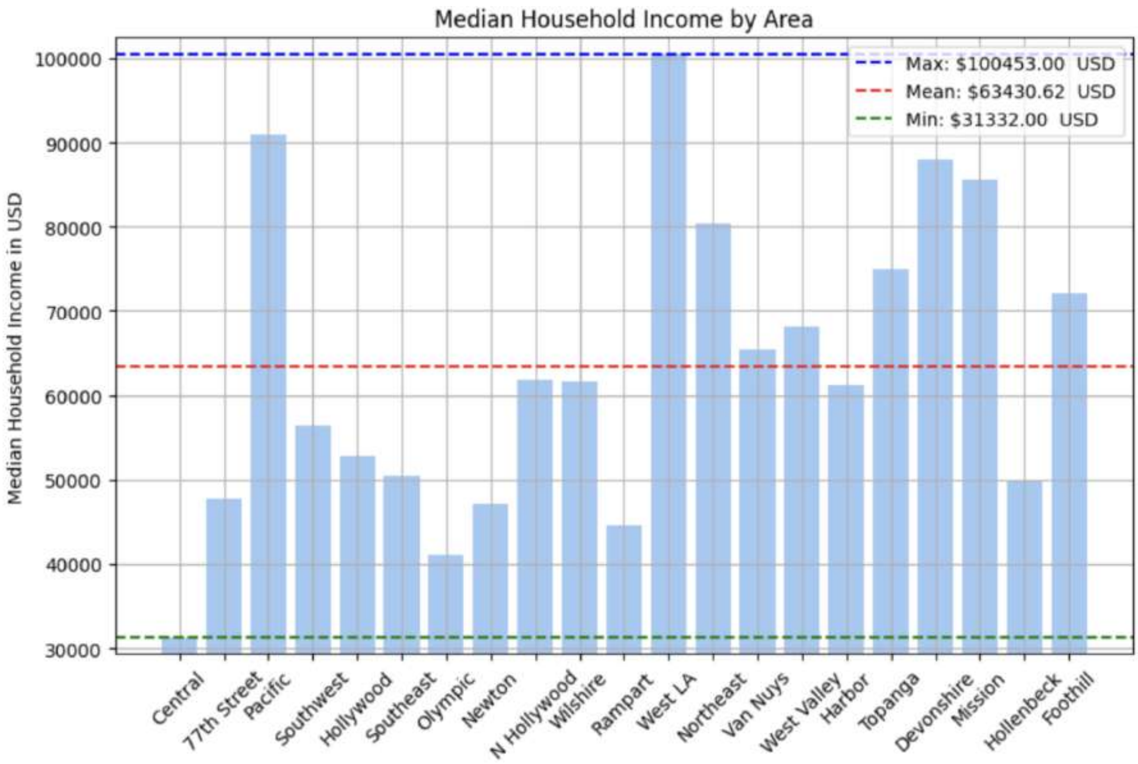


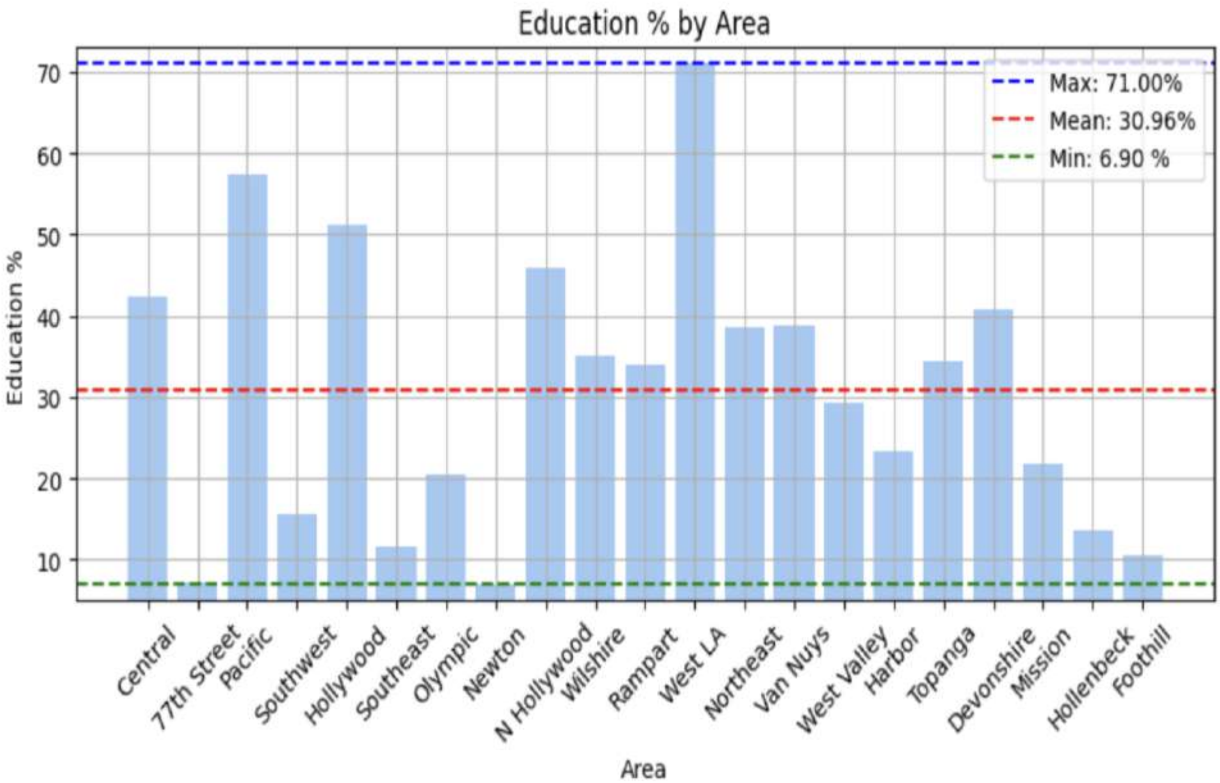
We have a negative relationship between the population and the number of crimes which means that in 24% of the cases if the population increase the numbers of crimes will decrease. This is just a tendency that in some areas is really clear such as 'Central' where the population is low and the crimes are a lot, and in areas like ''West Valley' and 'Foothill' the crime is lower because they have more population. This is a sligh relationship.

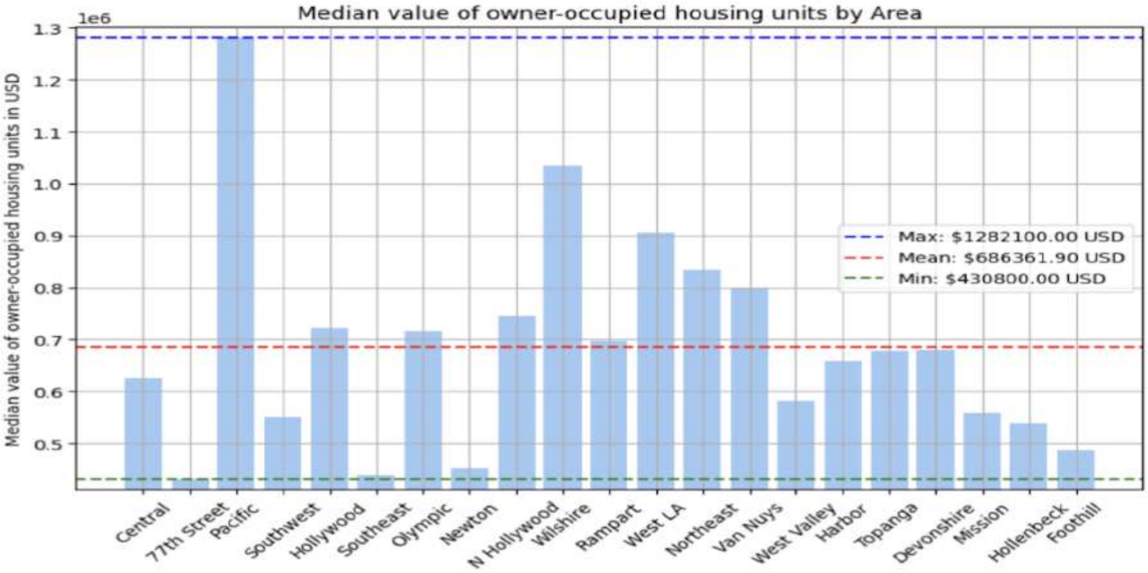
Employment % by AREA

In [64]:
```
plt.figure(figsize=(10, 6))
sns.set_palette("pastel")
plt.bar(merged_df['AREA NAME'], merged_df['Employment %'])

max_emp = merged_df['Employment %'].max()
plt.axhline(max_emp, color='blue', linestyle='--', label=f'Max: {max_emp:.2f} %')

mean_emp = merged_df['Employment %'].mean()
plt.axhline(mean_emp, color='red', linestyle='--', label=f'Mean: {mean_emp:.2f} %')

min_emp = merged_df['Employment %'].min()
plt.axhline(min_emp, color='green', linestyle='--', label=f'Min: {min_emp:.2f} %')

plt.grid()
plt.xlabel('Area')
plt.ylabel('Employment %')
plt.ylim(merged_df['Employment %'].min()-2, merged_df['Employment %'].max()+2)
plt.title('Employment % by Area')

plt.legend()
plt.xticks(rotation=45)
plt.show()
```

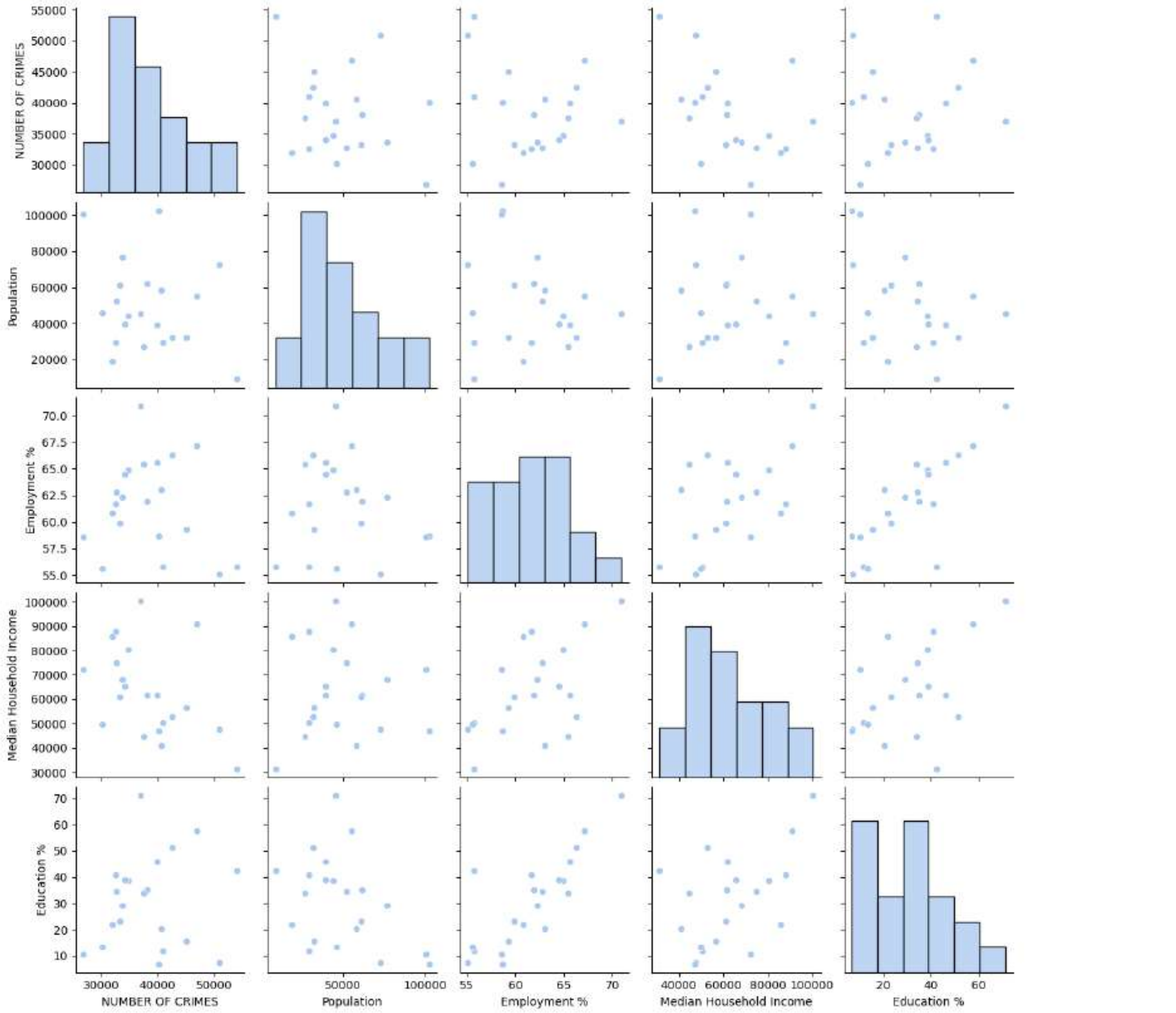



Median Household Income by Area
Max: $100453.00 USD
Mean: $63430.62 USD
Min: $31332.00 USD
Median Household Income in USD
100000
90000
80000
70000
60000
50000
40000
30000
Central
77th Street
Pacific
Southwest
Hollywood
Southeast
Olympic
Newton
N Hollywood
Wilshire
Rampart
West LA
Northeast
Van Nuys
West Valley
Harbor
Topanga
Devonshire
Mission
Hollenbeck
Foothill


Education % by Area
Max: 71.00%
Mean: 30.96%
Min: 6.90 %
Education %
70
60
50
40
30
20
10
Central
77th Street
Pacific
Southwest
Hollywood
Southeast
Olympic
Newton
N Hollywood
Wilshire
Rampart
West LA
Northeast
Van Nuys
West Valley
Harbor
Topanga
Devonshire
Mission
Hollenbeck
Foothill
Area



In [77]:
```
variables_of_interest = ['NUMBER OF CRIMES', 'Population', 'Employment %', 'Median Household Income','Education %']
sns.pairplot(merged_df[variables_of_interest])
plt.show()
```



Grouping the data by day of the week and analyzing crime frequencies:

In [75]:
```
days = df['DAY OF WEEK OCC'].unique()
days
```

Out[75]:
```
array(['Wednesday', 'Thursday', 'Saturday', 'Tuesday', 'Sunday', 'Monday',
       'Friday'], dtype=object)
```

In [76]:
```
crimes_per_day_total = df['DAY OF WEEK OCC'].value_counts().reset_index()
crimes_per_day_total=pd.DataFrame(crimes_per_day_total)
crimes_per_day_total.columns=['DAY OF WEEK', 'NUMBER OF CRIMES']
crimes_per_day_total
```

Out[76]:

	DAY OF WEEK	NUMBER OF CRIMES
0	Friday	123391
1	Saturday	118164
2	Wednesday	114896
3	Monday	114597
4	Thursday	114147
5	Sunday	112753
6	Tuesday	110873

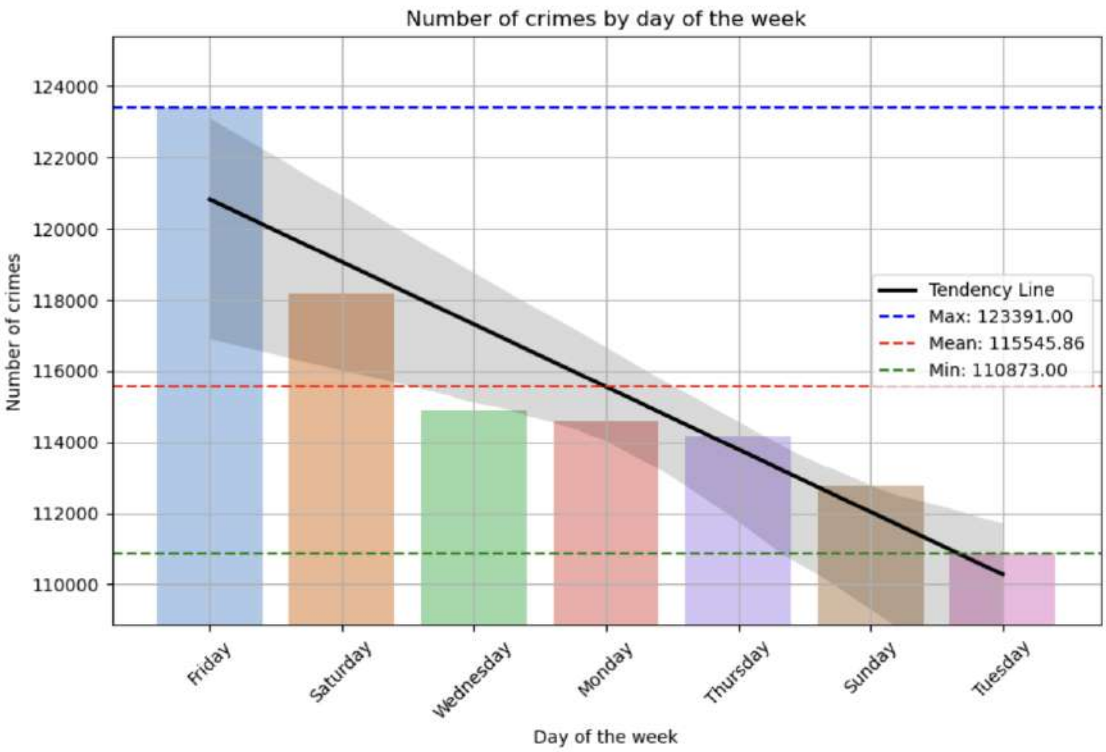
In [77]:
```
total_count_day = crimes_per_day_total['NUMBER OF CRIMES'].sum()
total_count_day
```

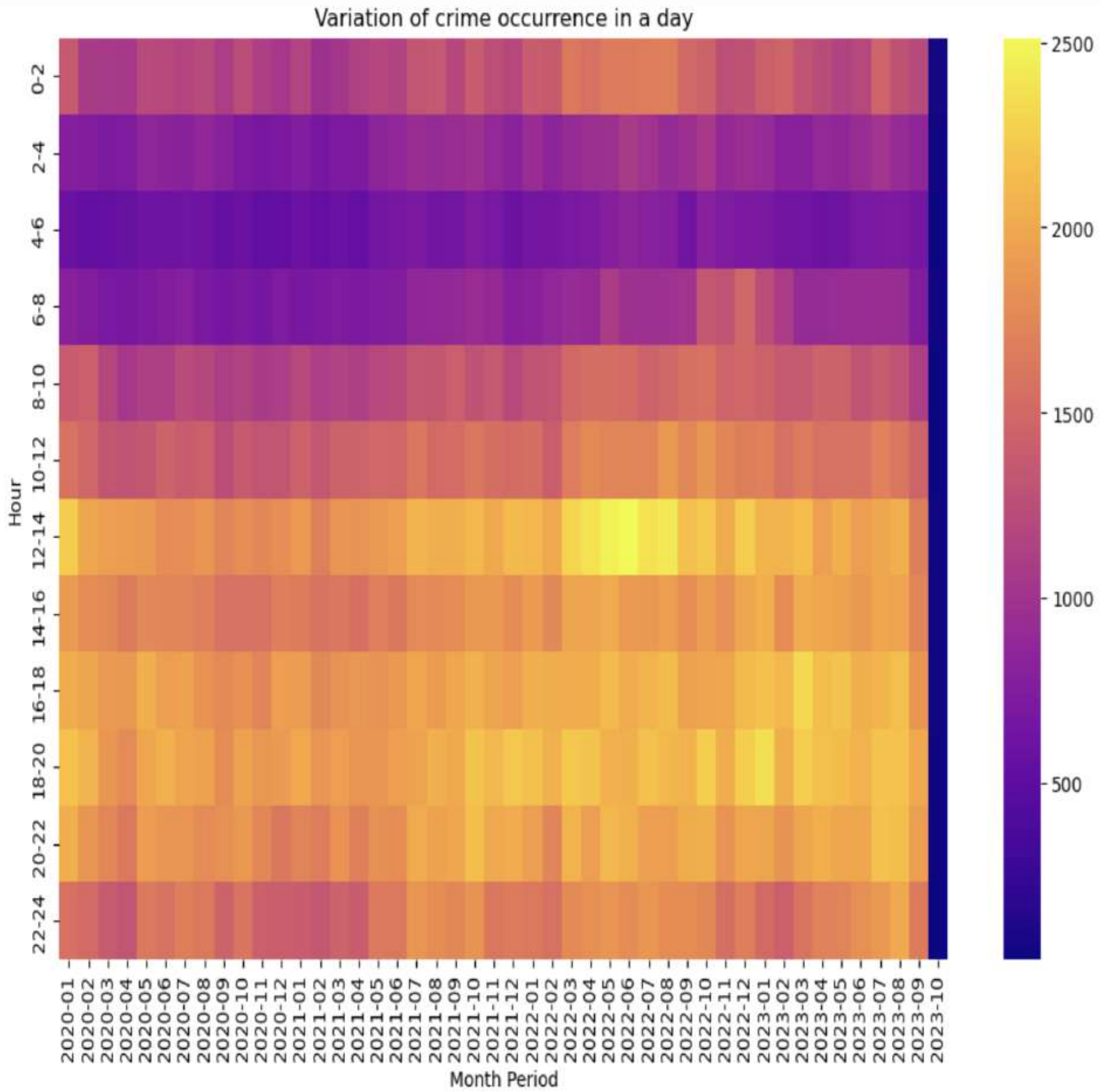
Out[77]: 808821

In [78]:
```
crimes_per_day_total['NUMBER OF CRIMES'] = crimes_per_day_total['NUMBER OF CRIMES'].astype(int)
crimes_per_day_total['%']=round((crimes_per_day_total['NUMBER OF CRIMES']/total_count_day)*100,2)
crimes_per_day_total
```

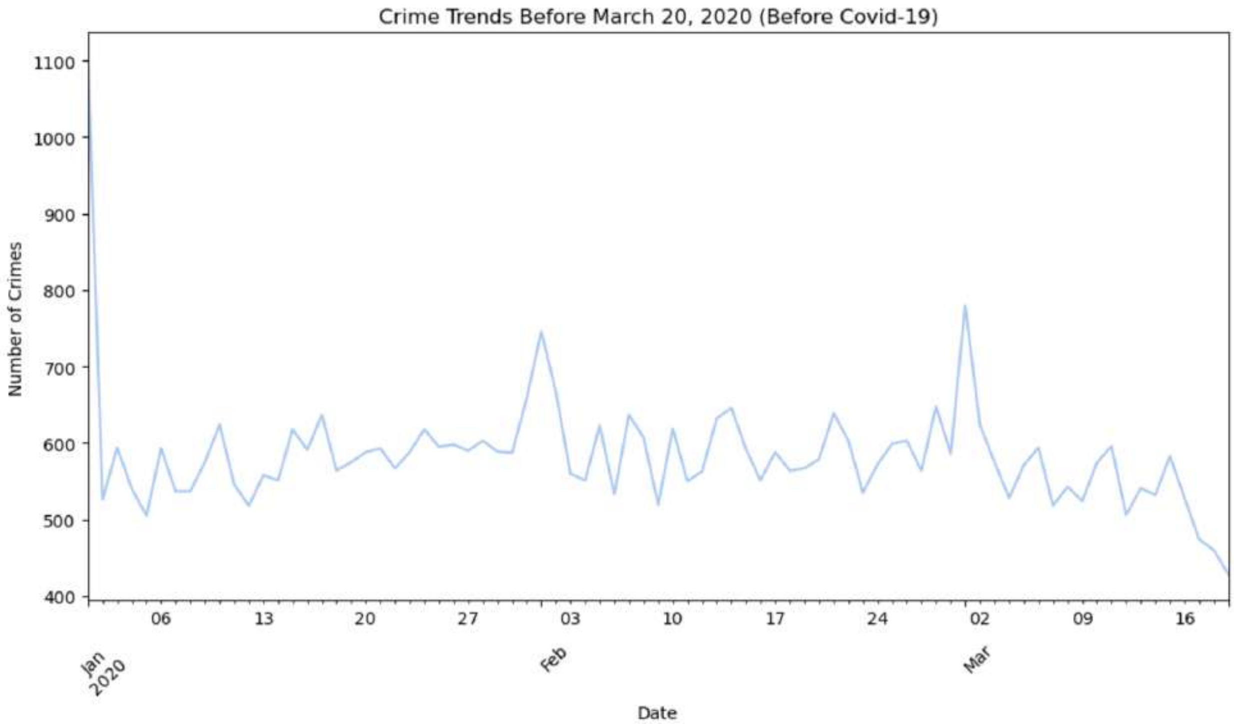
Out[78]:

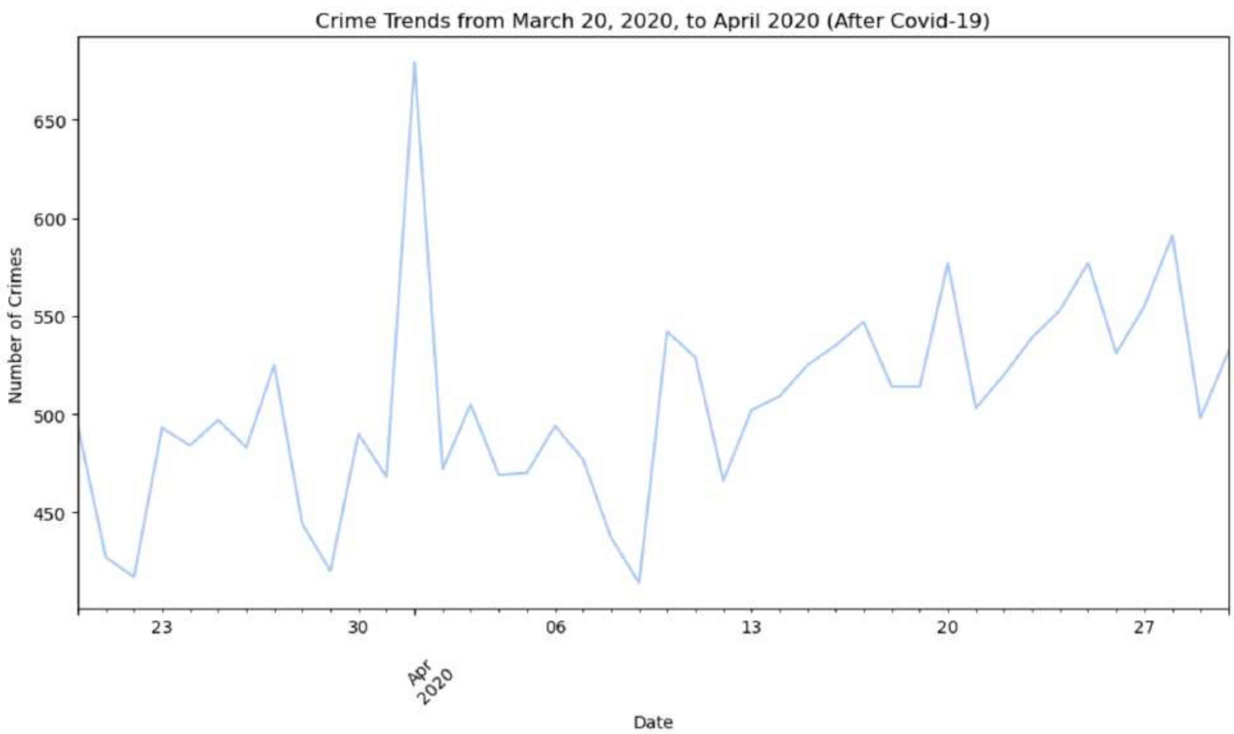
	DAY OF WEEK	NUMBER OF CRIMES	%
0	Friday	123391	15.26
1	Saturday	118164	14.61
2	Wednesday	114896	14.21
3	Monday	114597	14.17
4	Thursday	114147	14.11
5	Sunday	112753	13.94
6	Tuesday	110873	13.71

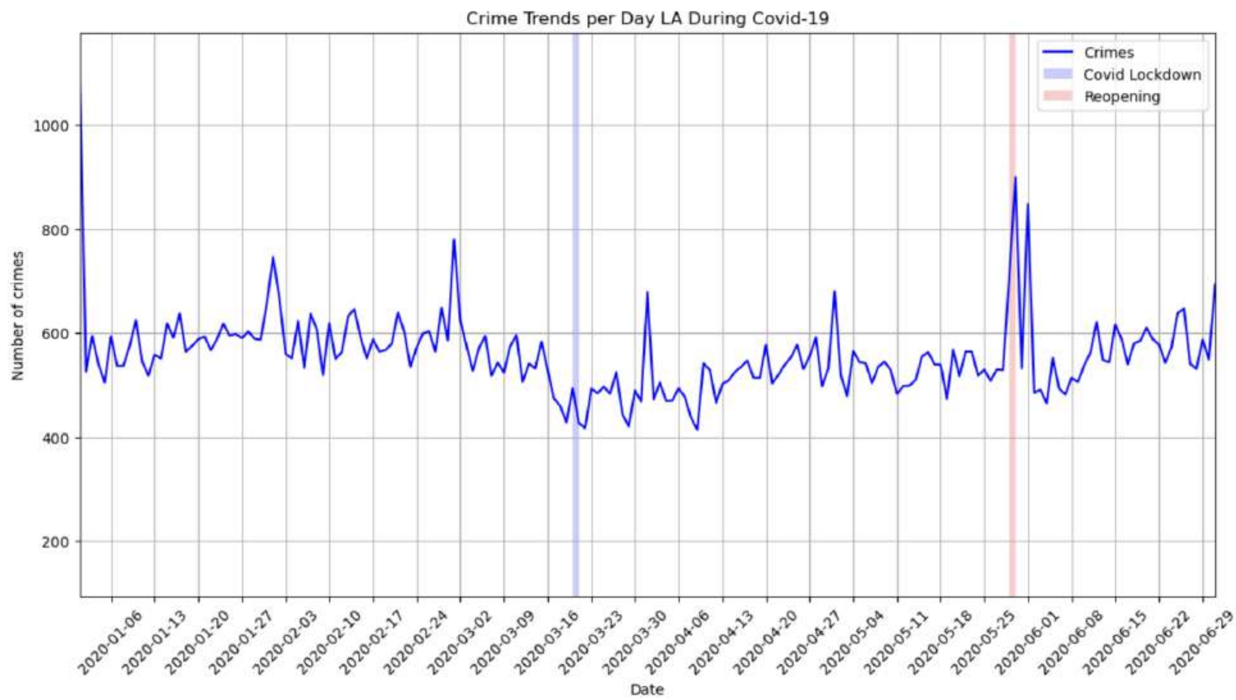


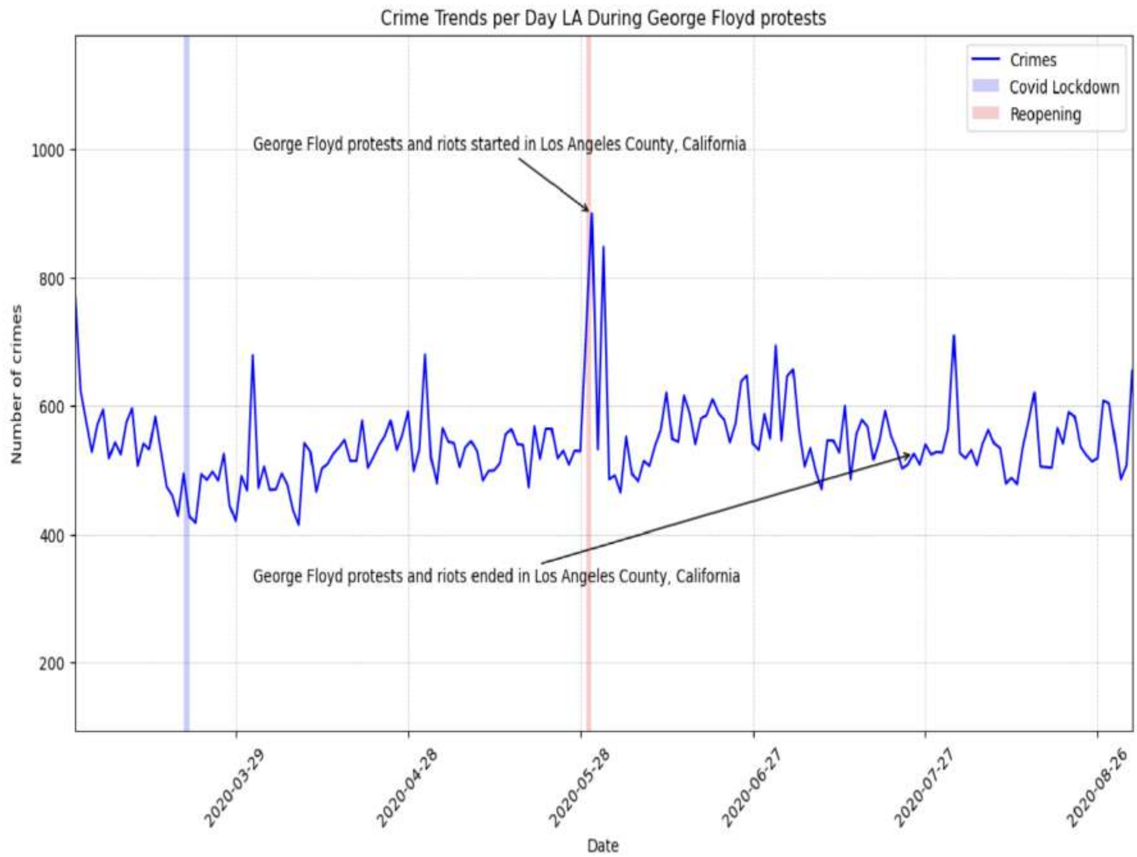

Variation of crime occurrence in a day
Hour
0-2
2-4
4-6
6-8
8-10
10-12
12-14
14-16
16-18
18-20
20-22
22-24
Month Period
2020-01
2020-02
2020-03
2020-04
2020-05
2020-06
2020-07
2020-08
2020-09
2020-10
2020-11
2020-12
2021-01
2021-02
2021-03
2021-04
2021-05
2021-06
2021-07
2021-08
2021-09
2021-10
2021-11
2021-12
2022-01
2022-02
2022-03
2022-04
2022-05
2022-06
2022-07
2022-08
2022-09
2022-10
2022-11
2022-12
2023-01
2023-02
2023-03
2023-04
2023-05
2023-06
2023-07
2023-08
2023-09
2023-10
2500
2000
1500
1000
500

Events or policy changes during the dataset period and analyze crime rate changes:



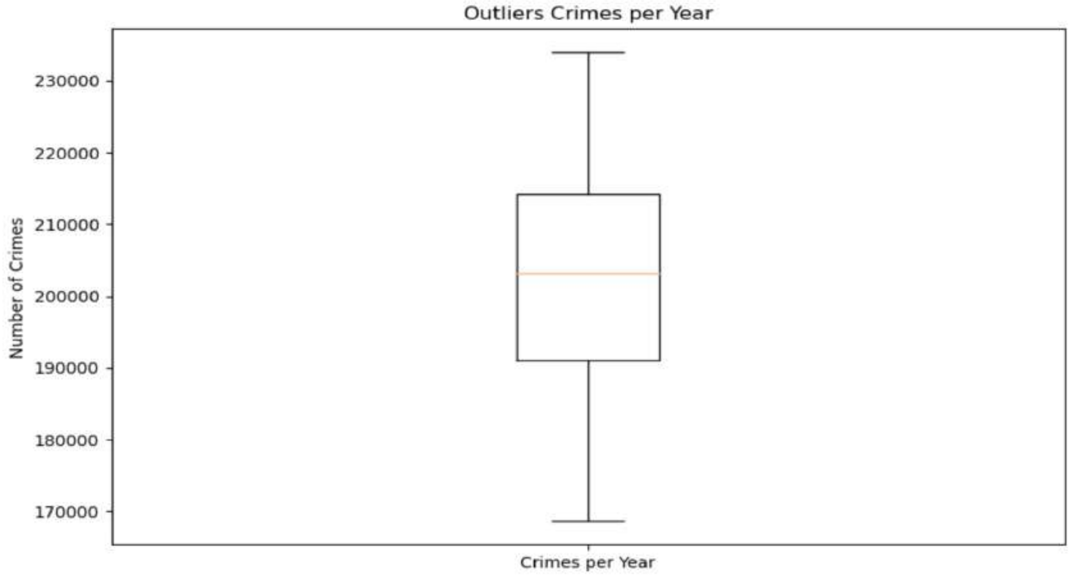



Crime Trends per Day LA During Covid-19
Crimes
Covid Lockdown
Reopening
Number of crimes
Date

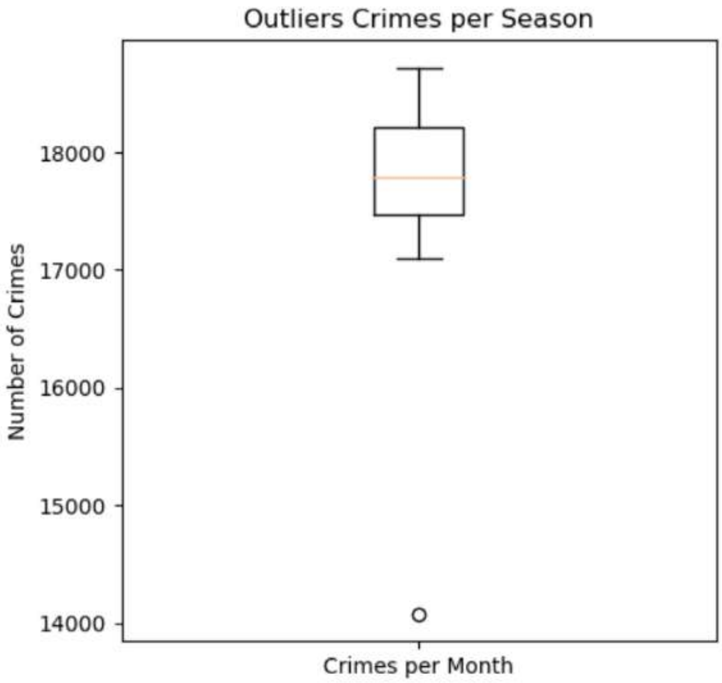

Crime Trends per Day LA During George Floyd protests
Crimes
Covid Lockdown
Reopening
George Floyd protests and riots started in Los Angeles County, California
George Floyd protests and riots ended in Los Angeles County, California
Number of crimes
Date

Data visualization techniques to identify dataset outliers:

In [126]:
```
plt.figure(figsize=(10, 6))
plt.title('Outliers Crimes per Year' )
plt.ylabel('Number of Crimes')
plt.boxplot(crime_counts_per_year)
plt.xticks([1], ['Crimes per Year'])
plt.show()
```

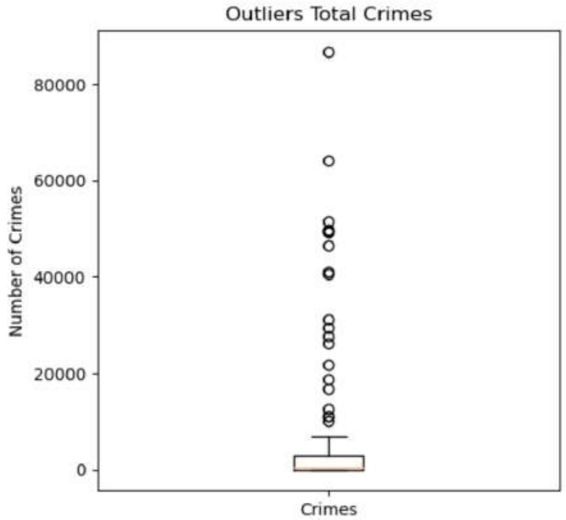


In [134]:
```
plt.figure(figsize=(5, 5))
plt.title('Outliers Crimes per Season' )
plt.ylabel('Number of Crimes')
plt.boxplot(average_crimes_per_month)
plt.xticks([1], ['Crimes per Month'])
plt.show()
```



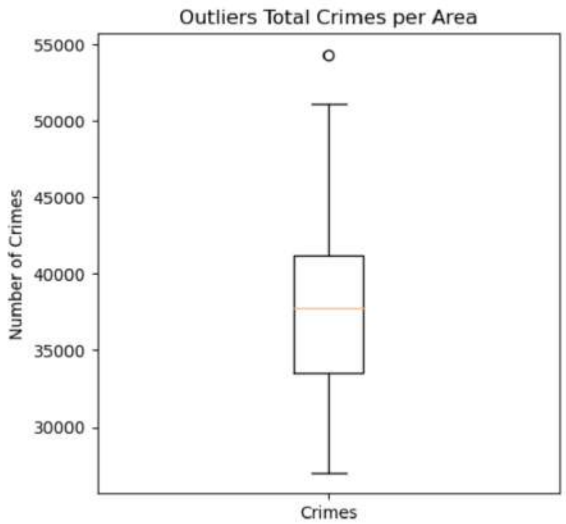
The outlier that we can see corresponds to October, however this can not be removed because is an important data

In [135]:
```
plt.figure(figsize=(5, 5))
plt.title('Outliers Total Crimes' )
plt.ylabel('Number of Crimes')
plt.boxplot(crimes_per_type_total['NUMBER OF CRIMES'])
plt.xticks([1], ['Crimes'])
plt.show()
```



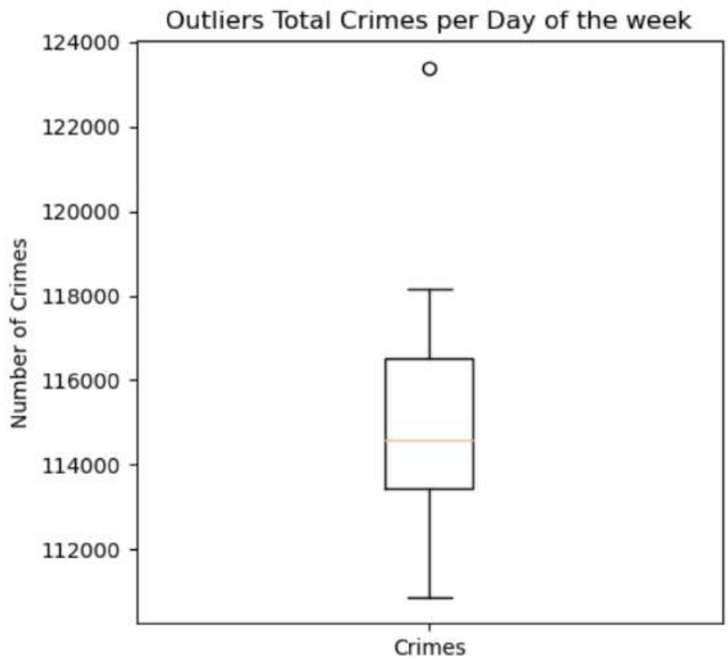
These outliers happened because there are multiple crimes that have been committed just once and because of that the graph shows that the most common ones are outliers. However, this is not a sign of emergency because this is the distribution of crimes and we are focus on the most important ones.

In [137]:
```
plt.figure(figsize=(5, 5))
plt.title('Outliers Total Crimes per Area' )
plt.ylabel('Number of Crimes')
plt.boxplot(crimes_per_area_total['NUMBER OF CRIMES'])
plt.xticks([1], ['Crimes'])
plt.show()
```



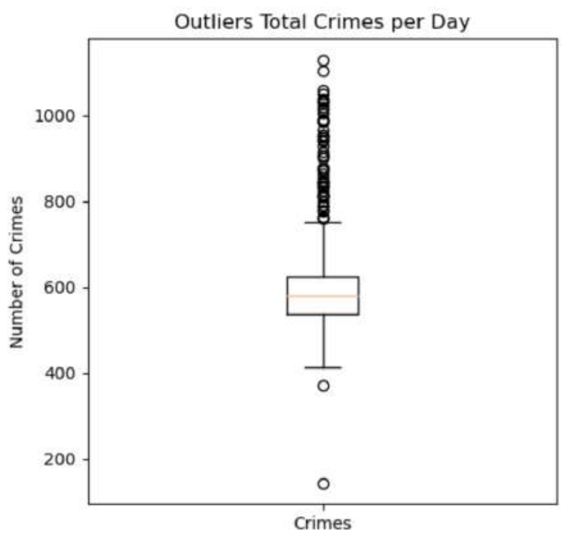
There is not an important number of outliers in Crimes per Area. The point that is out is just the most committed crime that is over 10% of the overall.

In [138]:
```
plt.figure(figsize=(5, 5))
plt.title('Outliers Total Crimes per Day of the week' )
plt.ylabel('Number of Crimes')
plt.boxplot(crimes_per_day_total['NUMBER OF CRIMES'])
plt.xticks([1], ['Crimes'])
plt.show()
```



We do not have outliers, again what we have here is that one value is bigger than the other ones for more than 15%.

In [141]:
```
plt.figure(figsize=(5, 5))
plt.title('Outliers Total Crimes per Day' )
plt.ylabel('Number of Crimes')
plt.boxplot(crimes_per_day['Number of Crimes'])
plt.xticks([1], ['Crimes'])
plt.show()
```



The unique point that can be considered as a real outlier is the one that is below 200 because it is associated with the last day of the study which can mean that they did not get all the crimes for that date. Other than that, the outliers above are associated with the different peaks that we had over time for different events.

Analyzing the dataset to identify any patterns or correlations between demographic factors (e.g., age, gender) and specific types of crimes:

In [99]:
```
print(df[['Vict Age', 'Vict Sex']].describe())
crime_counts = df['Crm Cd Desc'].value_counts()
print(crime_counts)
```

```
            Vict Age
count  808821.000000
mean       29.835287
std        21.767512
min        -3.000000
25%         8.000000
50%        31.000000
75%        45.000000
max       120.000000
VEHICLE - STOLEN                                           86748
BATTERY - SIMPLE ASSAULT                                   64204
THEFT OF IDENTITY                                          51494
BURGLARY FROM VEHICLE                                      49735
VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)    49443
                                                           ...
GRAND THEFT / AUTO REPAIR                                      5
FIREARMS RESTRAINING ORDER (FIREARMS RO)                       4
FAILURE TO DISPERSE                                            3
DISHONEST EMPLOYEE ATTEMPTED THEFT                             2
INCITING A RIOT                                                1
Name: Crm Cd Desc, Length: 138, dtype: int64
```

In [100]:
```
gender = df['Vict Sex'].unique()
gender
```

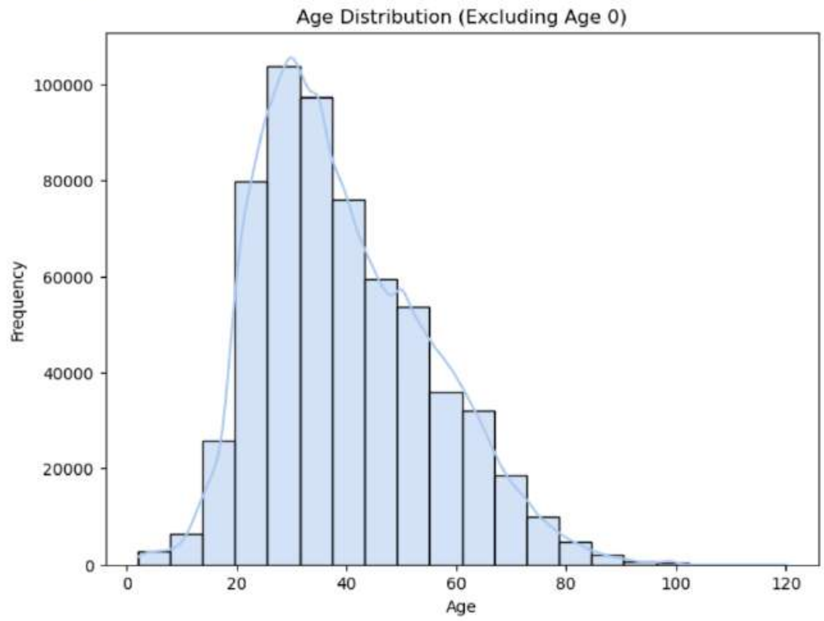
Out[100]: array(['F', 'M', 'X'], dtype=object)

In [101]:
```
crimes_gender = df['Vict Sex'].value_counts().reset_index()
crimes_gender=pd.DataFrame(crimes_gender)
crimes_gender.columns=['Vict Sex', 'NUMBER OF CRIMES']
crimes_gender
```

Out[101]:

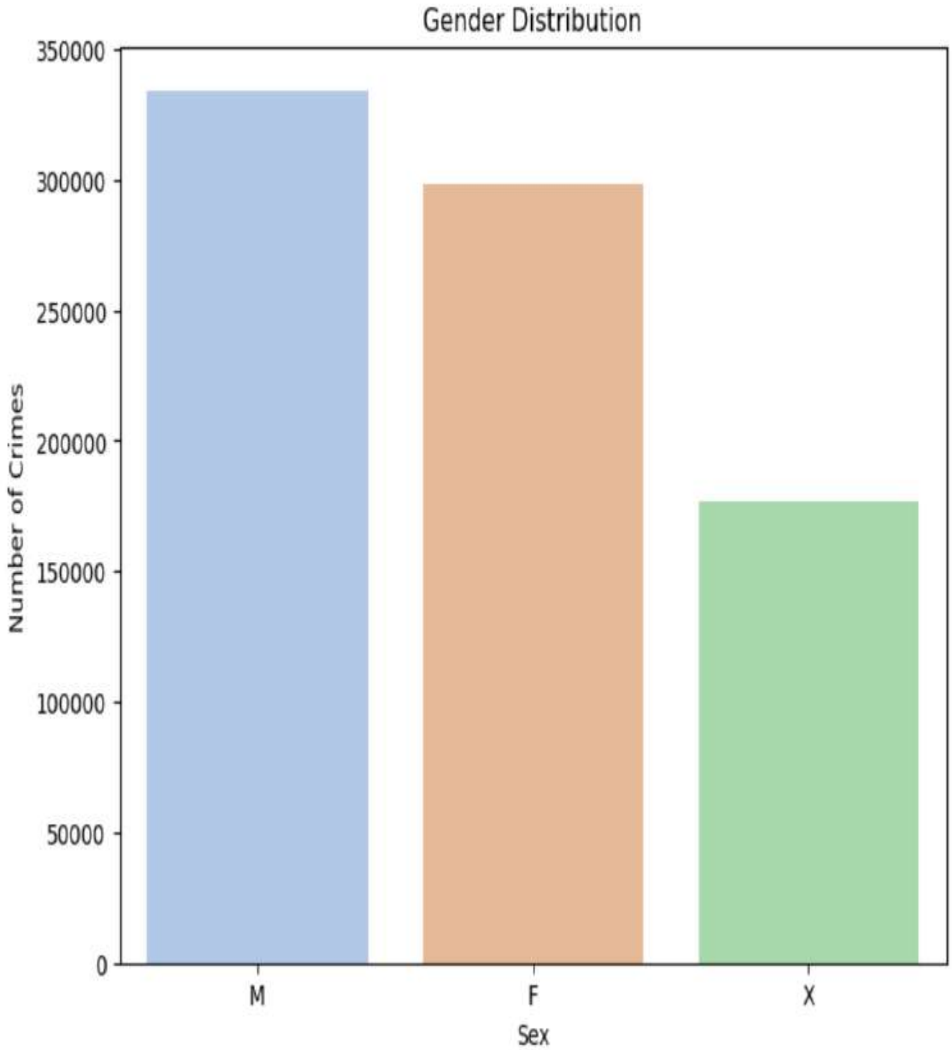
	Vict Sex	NUMBER OF CRIMES
0	M	334253
1	F	298138
2	X	176430

In [652…
```
df_filtered = df[df['Vict Age'] > 0]
plt.figure(figsize=(8, 6))
sns.histplot(df_filtered['Vict Age'], bins=20, kde=True)
plt.title('Age Distribution (Excluding Age 0)')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

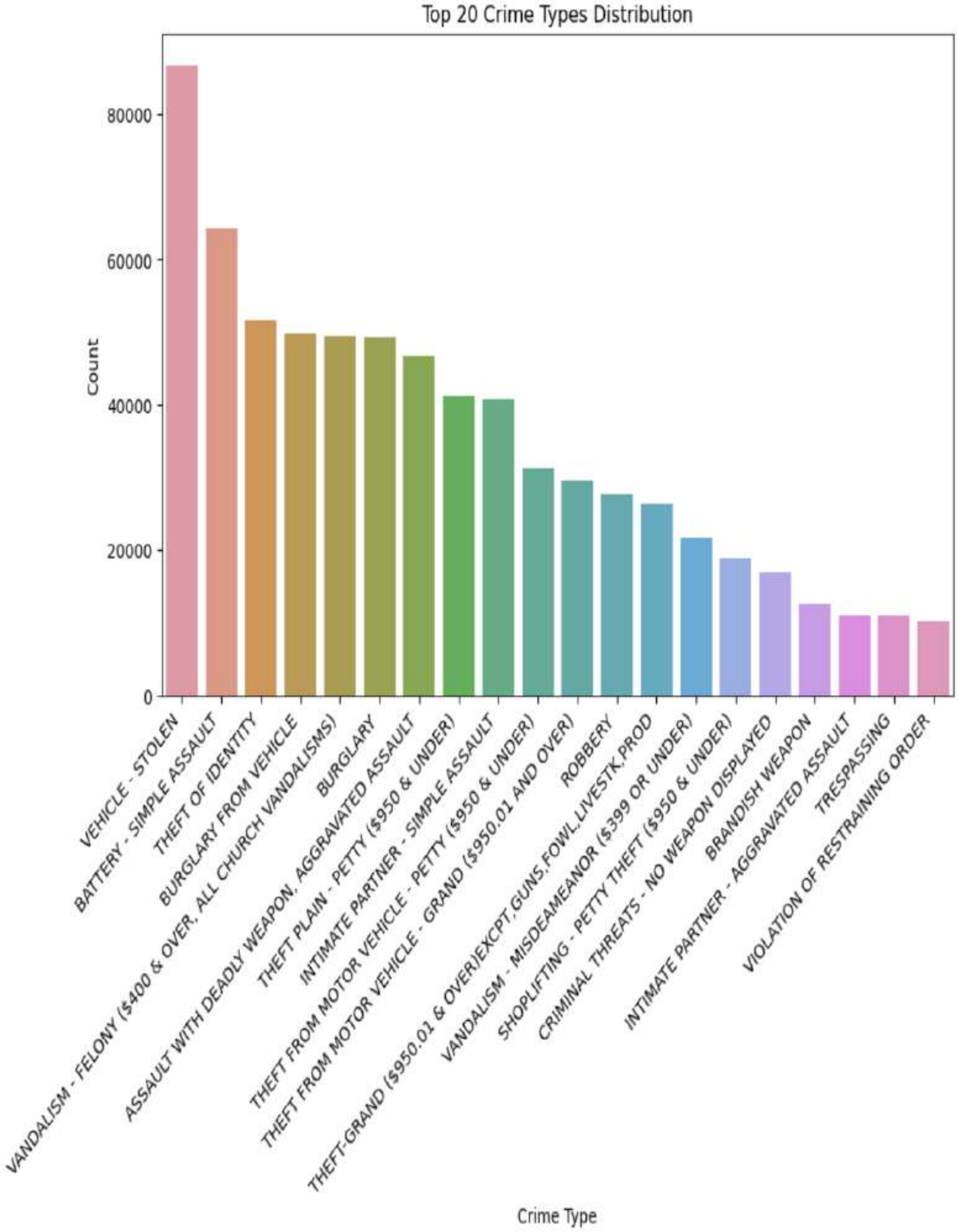


In [664…
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Vict Sex', order=df['Vict Sex'].value_counts().index)
plt.title('Gender Distribution')
plt.xlabel('Sex')
plt.ylabel('Number of Crimes')
plt.show()
```

In [664…
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Vict Sex', order=df['Vict Sex'].value_counts().index)
plt.title('Gender Distribution')
plt.xlabel('Sex')
plt.ylabel('Number of Crimes')
plt.show()
```



```python
In [665… top_20_crimes = crime_counts[:20]
plt.figure(figsize=(10, 6))
sns.barplot(x=top_20_crimes.index, y=top_20_crimes.values)
plt.title('Top 20 Crime Types Distribution')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Crime Type')
plt.ylabel('Count')
plt.show()
age_crime_corr = df['Vict Age'].corr(df['Crm Cd'])
print(f"Correlation between age and crime frequency: {age_crime_corr}")
```



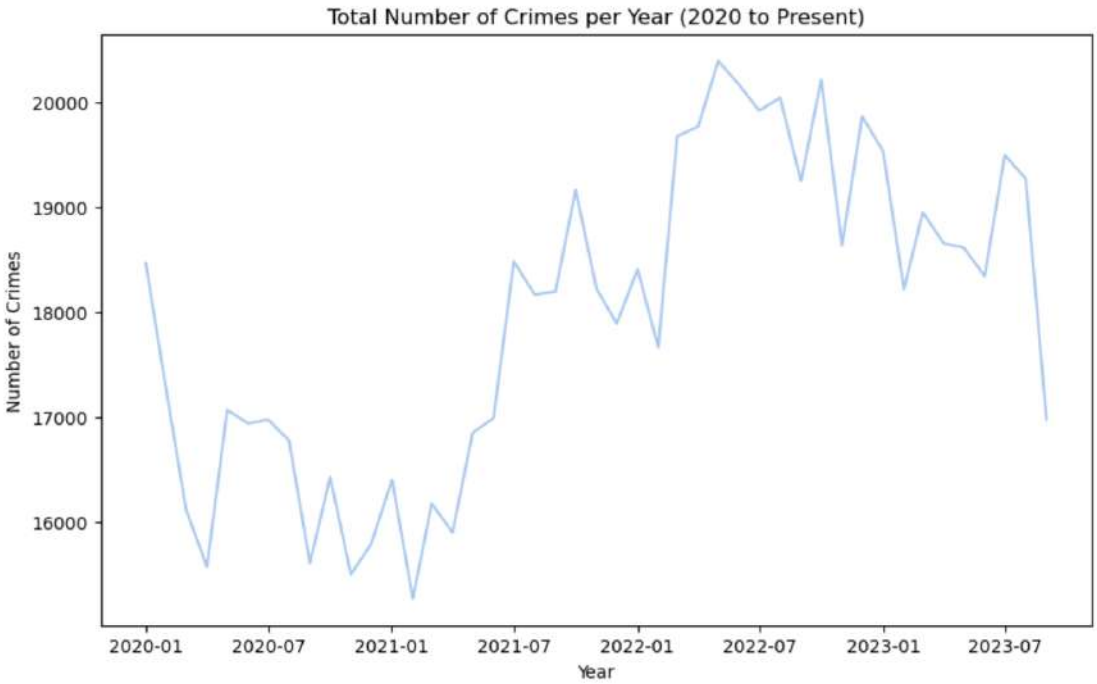

```
Correlation between age and crime frequency: -0.010950231136746072
```

Employing time series forecasting methods, such as ARIMA or Prophet, to predict future crime trends based on historical data:

In [105]:
```
crimes_per_month = df.groupby(['Year', 'Month'], as_index = False).size()
crimes_per_month['yearMonth'] = pd.to_datetime(crimes_per_month['Year'].astype(str) + '-' +crimes_per_month['Month']
crimes_per_month.set_index('yearMonth', inplace=True)
crimes_per_month.rename(columns={'size': 'Num'}, inplace=True)
crimes_per_month = crimes_per_month.drop(crimes_per_month.index[-1])
crimes_per_month
```

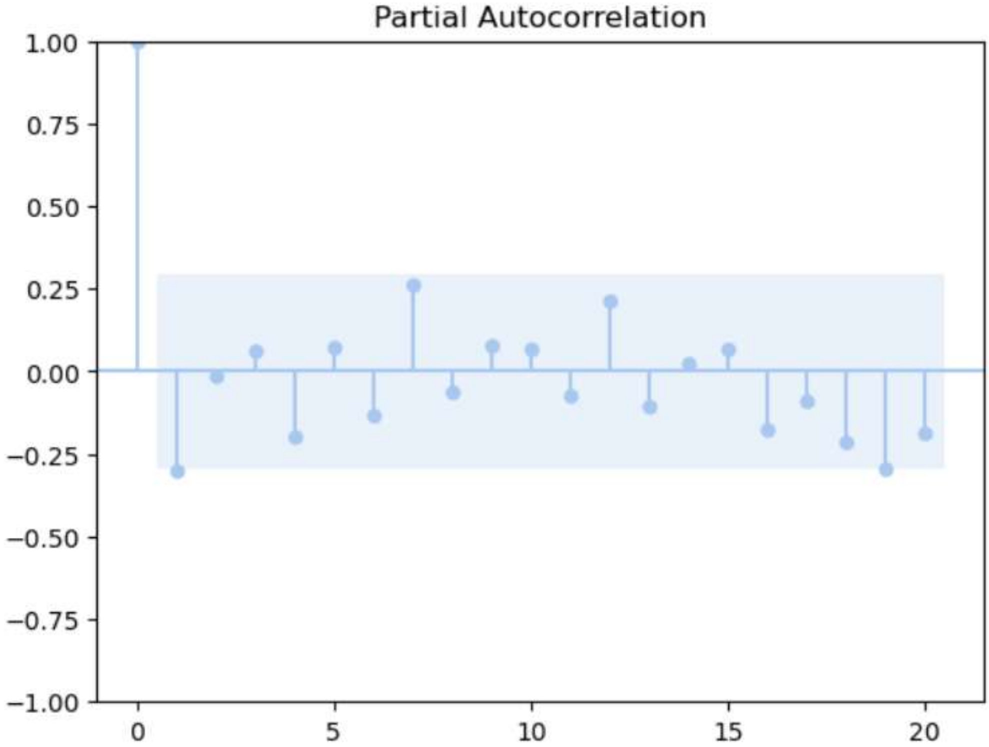
Out[105]:

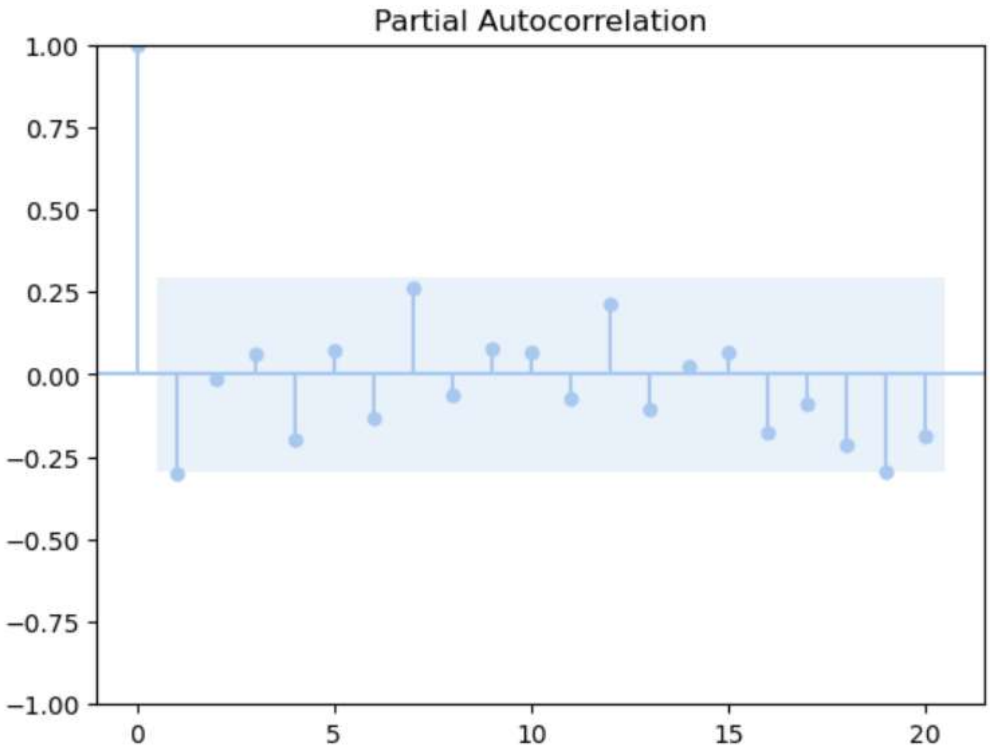
yearMonth	Year	Month	Num
2020-01-01	2020	1	18466
2020-02-01	2020	2	17243
2020-03-01	2020	3	16121
2020-04-01	2020	4	15576
2020-05-01	2020	5	17070
2020-06-01	2020	6	16942
2020-07-01	2020	7	16979
2020-08-01	2020	8	16777
2020-09-01	2020	9	15613
2020-10-01	2020	10	16430
2020-11-01	2020	11	15505
2020-12-01	2020	12	15792
2021-01-01	2021	1	16406
2021-02-01	2021	2	15275
2021-03-01	2021	3	16179
2021-04-01	2021	4	15903
2021-05-01	2021	5	16854
2021-06-01	2021	6	16993
2021-07-01	2021	7	18483
2021-08-01	2021	8	18168
2021-09-01	2021	9	18196
2021-10-01	2021	10	19163
2021-11-01	2021	11	18224
2021-12-01	2021	12	17891
2022-01-01	2022	1	18409
2022-02-01	2022	2	17666
2022-03-01	2022	3	19675
2022-04-01	2022	4	19768
2022-05-01	2022	5	20391
2022-06-01	2022	6	20165
2022-07-01	2022	7	19917
2022-08-01	2022	8	20042
2022-09-01	2022	9	19248
2022-10-01	2022	10	20211


Total Number of Crimes per Year (2020 to Present)
Number of Crimes
20000
19000
18000
17000
16000
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
Year

In [111]:
```
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

plot_pacf(crimes_per_month['shiftDiff'].dropna(), lags=20)
```

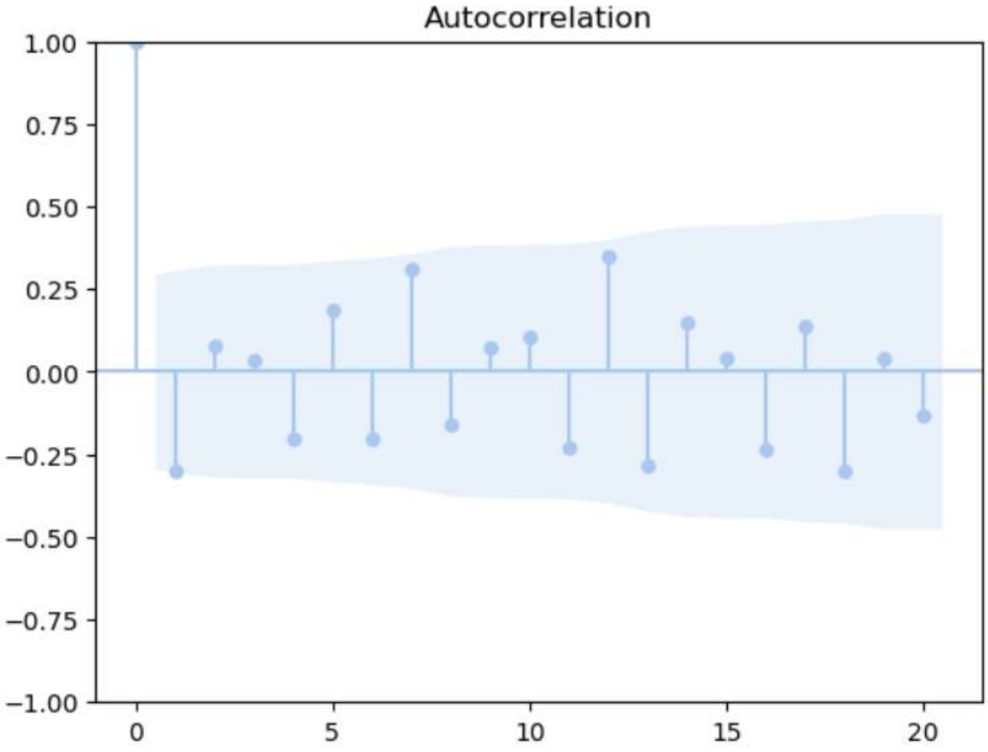
Out[111]:

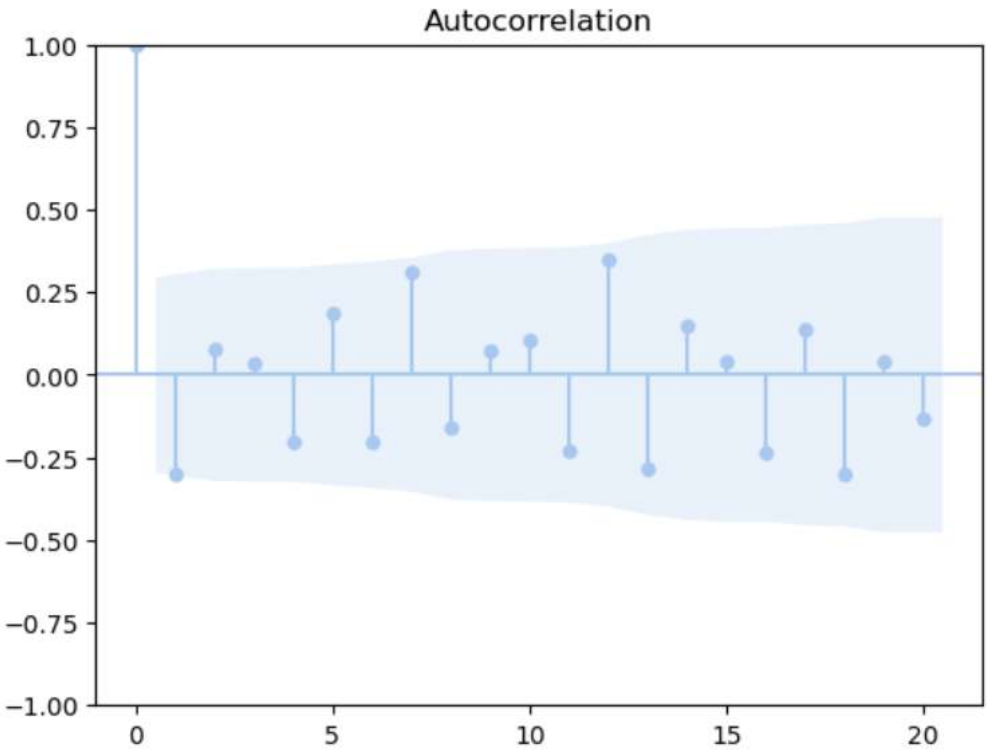




In [112]:
```python
plot_acf(crimes_per_month['shiftDiff'].dropna(), lags=20)
```

Out[112]:





```
In [113]: train = crimes_per_month[:round(len(crimes_per_month)*0.8)]
          test = crimes_per_month[round(len(crimes_per_month)*0.8):]
```

```
In [114]: from statsmodels.tsa.arima.model import ARIMA
          model = ARIMA(train['Num'], order=(1,1,12))
          result = model.fit()
          print(result.summary())
```

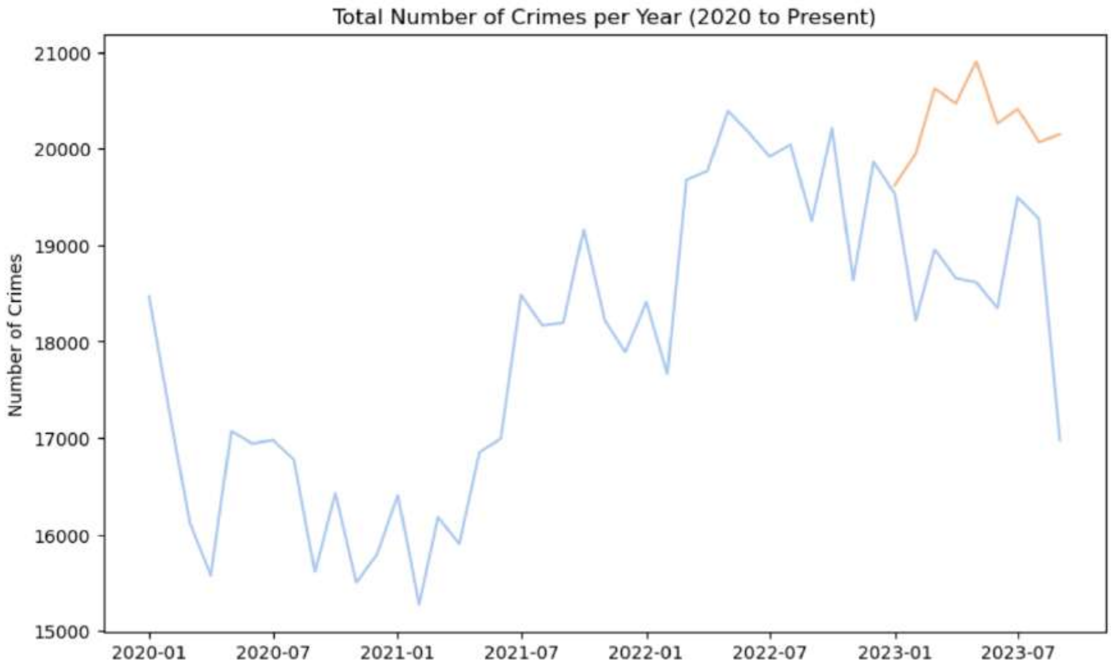
```
                               SARIMAX Results
==============================================================================
Dep. Variable:                    Num   No. Observations:                   36
Model:                 ARIMA(1, 1, 12)  Log Likelihood                -277.839
Date:                Fri, 03 Nov 2023   AIC                            583.678
Time:                        15:34:32   BIC                            605.453
Sample:                    01-01-2020   HQIC                           591.195
                         - 12-01-2022
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.8130      0.231     -3.514      0.000      -1.267      -0.360
ma.L1          0.5832      0.563      1.035      0.300      -0.521       1.687
ma.L2         -0.3263      0.432     -0.755      0.450      -1.174       0.521
ma.L3         -0.0209      0.692     -0.030      0.976      -1.376       1.335
ma.L4          0.2411      0.871      0.277      0.782      -1.465       1.947
ma.L5          0.1385      0.465      0.298      0.766      -0.773       1.050
ma.L6         -0.1639      0.359     -0.457      0.648      -0.867       0.539
ma.L7         -0.2445      0.631     -0.387      0.698      -1.481       0.992
ma.L8         -0.2780      0.619     -0.449      0.654      -1.492       0.936
ma.L9         -0.0231      0.371     -0.062      0.950      -0.749       0.703
ma.L10         0.1304      0.337      0.387      0.699      -0.531       0.792
ma.L11         0.1355      0.420      0.322      0.747      -0.688       0.959
ma.L12         0.2833      0.330      0.858      0.391      -0.364       0.931
sigma2      5.722e+05   4.75e+05      1.205      0.228   -3.59e+05     1.5e+06
===================================================================================
Ljung-Box (L1) (Q):                   0.89   Jarque-Bera (JB):                 1.70
Prob(Q):                              0.35   Prob(JB):                         0.43
Heteroskedasticity (H):               3.43   Skew:                             0.53
Prob(H) (two-sided):                  0.04   Kurtosis:                         3.19
===================================================================================
```

In [115]:
```
prediction = result.predict(start=test.index[0], end=test.index[-1])
crimes_per_month['prediction'] = prediction
crimes_per_month.tail()
```

Out[115]:

	Year	Month	Num	shift	shiftDiff	prediction
yearMonth						
2023-05-01	2023	5	18615	18657.0	-42.0	20904.748053
2023-06-01	2023	6	18344	18615.0	-271.0	20259.976362
2023-07-01	2023	7	19496	18344.0	1152.0	20410.908683
2023-08-01	2023	8	19269	19496.0	-227.0	20068.492666
2023-09-01	2023	9	16980	19269.0	-2289.0	20147.669986

In [116]:
```
crimes_per_month.dropna()
plt.figure(figsize=(10, 6))
plt.plot(crimes_per_month.index, crimes_per_month['Num'])
plt.title('Total Number of Crimes per Year (2020 to Present)')
plt.xlabel('Year')
plt.ylabel('Number of Crimes')
sns.lineplot(data=crimes_per_month, x=crimes_per_month.index, y='prediction')
plt.show()
```



In [341…

```
futureDate = pd.DataFrame(pd.date_range(start='2023-09-01', end='2024-12-01', freq='MS'), columns=['Dates'])
futureDate.set_index('Dates', inplace=True)
futureDate
```

Out[341]:

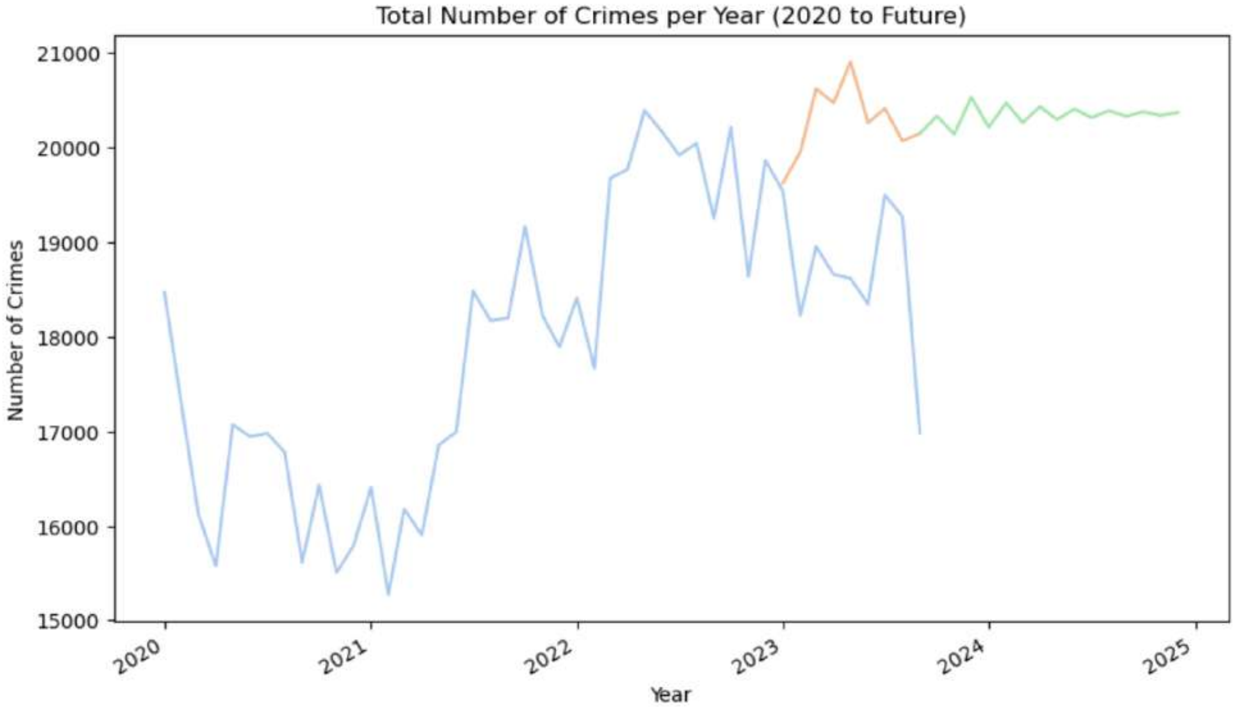
Dates
2023-09-01
2023-10-01
2023-11-01
2023-12-01
2024-01-01
2024-02-01
2024-03-01
2024-04-01
2024-05-01
2024-06-01
2024-07-01
2024-08-01
2024-09-01
2024-10-01
2024-11-01
2024-12-01

In [345…

```
result.predict(start=futureDate.index[0], end=futureDate.index[-1])
```

Out[345]:

```
2023-09-01    19541.501727
2023-10-01    19687.157629
2023-11-01    19428.617857
2023-12-01    19639.706732
2024-01-01    19467.359891
2024-02-01    19608.075188
2024-03-01    19493.185964
2024-04-01    19586.989083
2024-05-01    19510.402047
2024-06-01    19572.932741
2024-07-01    19521.878569
2024-08-01    19563.562553
2024-09-01    19529.529007
2024-10-01    19557.316232
2024-11-01    19534.628914
2024-12-01    19553.152333
Freq: MS, Name: predicted_mean, dtype: float64
```


Total Number of Crimes per Year (2020 to Future)
21000
20000
19000
18000
17000
16000
15000
Number of Crimes
2020
2021
2022
2023
2024
2025
Year