```python
import pandas as pd
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from prophet import Prophet
import warnings
warnings.filterwarnings('ignore')
```

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

```python
#Task 1 - Download the dataset from the provided link and load it into your preferred data analysis tool

df_crime = pd.read_csv("Crime_Data_from_2020_to_Present.csv")
```

2. Data Inspection:

```python
#Task 2.1 - Display the first few rows of the dataset

df_crime.head()
```

	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
0	190326475	03/01/2020 12:00:00 AM	03/01/2020 12:00:00 AM	2130	7	Wilshire	784	1	510	VEHICLE - STOLEN	...	AA	Adult Arrest	510.0	998.0	NaN	NaN	1900 S LONGWOOD AV	NaN
1	200106753	02/09/2020 12:00:00 AM	02/08/2020 12:00:00 AM	1800	1	Central	182	1	330	BURGLARY FROM VEHICLE	...	IC	Invest Cont	330.0	998.0	NaN	NaN	1000 S FLOWER ST	NaN
2	200320258	11/11/2020 12:00:00 AM	11/04/2020 12:00:00 AM	1700	3	Southwest	356	1	480	BIKE - STOLEN	...	IC	Invest Cont	480.0	NaN	NaN	NaN	1400 W 37TH ST	NaN
3	200907217	05/10/2023 12:00:00 AM	03/10/2020 12:00:00 AM	2037	9	Van Nuys	964	1	343	SHOPLIFTING-GRAND THEFT ($950.01 & OVER)	...	IC	Invest Cont	343.0	NaN	NaN	NaN	14000 RIVERSIDE DR	NaN
4	220614831	08/18/2022 12:00:00 AM	08/17/2020 12:00:00 AM	1200	6	Hollywood	666	2	354	THEFT OF IDENTITY	...	IC	Invest Cont	354.0	NaN	NaN	NaN	1900 TRANSIENT	NaN

5 rows × 28 columns

```python
#Task 2.2 - Check the data types of each column

df_crime.dtypes
```

```
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
```

```python
#Task 2.3 - Review column names and descriptions, if available

df_crime.describe()
```

	DR_NO	TIME OCC	AREA	Rpt Dist No	Part 1-2	Crm Cd	Vict Age	Premis Cd	Weapon Used Cd	Crm Cd 1
count	9.826380e+05	982638.000000	982638.000000	982638.000000	982638.000000	982638.000000	982638.000000	982624.000000	326167.000000	982627.000000
mean	2.197437e+08	1338.945426	10.700277	1116.459887	1.404253	500.823555	29.079817	306.133008	363.840882	500.57866
std	1.294954e+07	651.537830	6.107808	610.893787	0.490747	206.211940	21.970094	219.053795	123.684663	206.01036
min	8.170000e+02	1.000000	1.000000	101.000000	1.000000	110.000000	-4.000000	101.000000	101.000000	110.000000
25%	2.106089e+08	900.000000	5.000000	587.000000	1.000000	331.000000	0.000000	101.000000	311.000000	331.000000
50%	2.208146e+08	1420.000000	11.000000	1141.000000	1.000000	442.000000	30.000000	203.000000	400.000000	442.000000
75%	2.309153e+08	1900.000000	16.000000	1617.000000	2.000000	626.000000	44.000000	501.000000	400.000000	626.000000
max	2.499253e+08	2359.000000	21.000000	2199.000000	2.000000	956.000000	120.000000	976.000000	516.000000	956.000000

3.Data Cleaning:

```python
#Task 3.1 - Identify and handle missing data appropriately
df_crime.isnull().sum()


df_crime['Vict Sex'] = df_crime['Vict Sex'].replace({' ': 'X', '-': 'X', 'H': 'X'})

missing_values = ["n.a.", "NA", "n/a", "na", 0]
df_crime = pd.read_csv('Crime_Data_from_2020_to_Present.csv', na_values = missing_values)
df_crime.dropna(axis= 0, inplace = True, how = 'all')
```

```python
#Task 3.2 - Check for and remove duplicate rows

df_crime = df_crime.drop_duplicates()
print(df_crime.duplicated().sum())
```

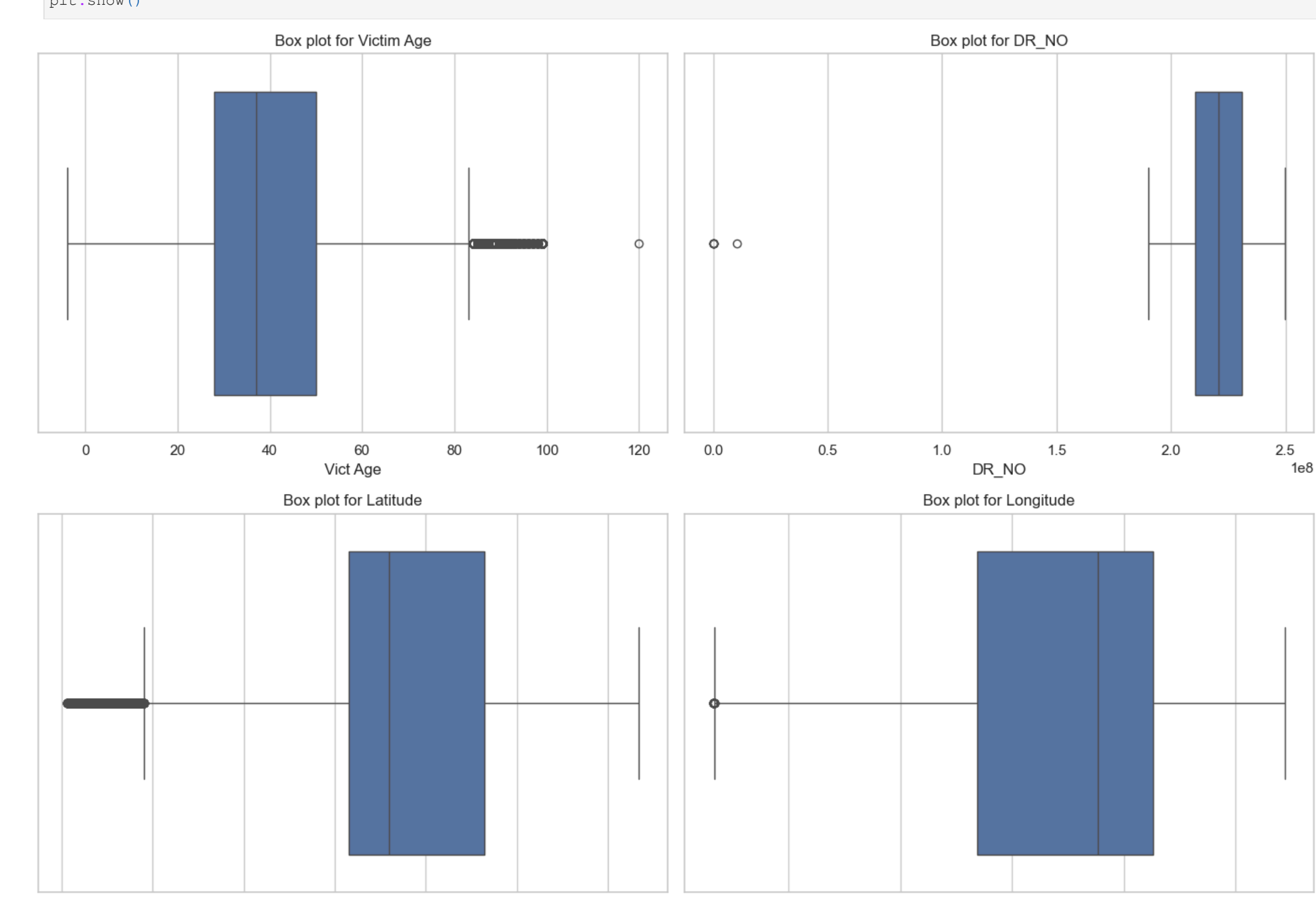
0

```python
#Task 3.3 - Convert data types if needed (e.g., dates to date format, numerical values to appropriate numeric types)

df_crime['Date Rptd'] = pd.to_datetime(df_crime['Date Rptd']).dt.date
df_crime['DATE OCC'] = pd.to_datetime(df_crime['DATE OCC']).dt.date
df_crime['AREA'] = df_crime['AREA'].astype(str)
df_crime['Rpt Dist No'] = df_crime['Rpt Dist No'].astype(int)

def convert_time(time):
    time = str(time).zfill(4)
    return f"{time[:2]}:{time[2:]}"
df_crime['TIME OCC'] = df_crime['TIME OCC'].apply(convert_time)

df_crime
```

	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	LOCAT
0	190326475	2020-03-01	2020-03-01	21:30	7	Wilshire	784	1	510	VEHICLE - STOLEN	...	AA	Adult Arrest	510.0	998.0	NaN	NaN	19 LONGW
1	200106753	2020-02-09	2020-02-08	18:00	1	Central	182	1	330	BURGLARY FROM VEHICLE	...	IC	Invest Cont	330.0	998.0	NaN	NaN	10 FLOWEI
2	200320258	2020-11-11	2020-11-04	17:00	3	Southwest	356	1	480	BIKE - STOLEN	...	IC	Invest Cont	480.0	NaN	NaN	NaN	140 37TI
3	200907217	2023-05-10	2020-03-10	20:37	9	Van Nuys	964	1	343	SHOPLIFTING-GRAND THEFT ($950.01 & OVER)	...	IC	Invest Cont	343.0	NaN	NaN	NaN	1 RIVERS
4	220614831	2022-08-18	2020-08-17	12:00	6	Hollywood	666	2	354	THEFT OF IDENTITY	...	IC	Invest Cont	354.0	NaN	NaN	NaN	TRANSI
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
982633	242011172	2024-08-20	2024-08-17	23:00	20	Olympic	2033	1	341	THEFT-GRAND ($950.01 & OVER)EXCPT,GUNS,FOWL,LI...	...	IC	Invest Cont	341.0	NaN	NaN	NaN	WILSI
982634	240710284	2024-07-24	2024-07-23	14:00	7	Wilshire	788	1	510	VEHICLE - STOLEN	...	IC	Invest Cont	510.0	NaN	NaN	NaN	400 23RI
982635	240104953	2024-01-15	2024-01-15	01:00	1	Central	101	2	745	VANDALISM - MISDEAMEANOR ($399 OR UNDER)	...	IC	Invest Cont	745.0	NaN	NaN	NaN	130 SUNSE
982636	240309674	2024-04-24	2024-04-24	15:00	3	Southwest	358	1	230	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	...	IC	Invest Cont	230.0	NaN	NaN	NaN	FLOWEI
982637	240910892	2024-08-13	2024-08-12	23:00	9	Van Nuys	914	1	510	VEHICLE - STOLEN	...	IC	Invest Cont	510.0	NaN	NaN	NaN	VESPEI

982638 rows × 28 columns

```python
#Task 3.4 Deal with outliers if relevant to your analysis.

sns.set(style="whitegrid")
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(14, 10))
sns.boxplot(x=df_crime['Vict Age'], ax=axes[0, 0])
axes[0, 0].set_title('Box plot for Victim Age')
sns.boxplot(x=df_crime['DR_NO'], ax=axes[0, 1])
axes[0, 1].set_title('Box plot for DR_NO')
sns.boxplot(x=df_crime['LAT'], ax=axes[1, 0])
axes[1, 0].set_title('Box plot for Latitude')
sns.boxplot(x=df_crime['LON'], ax=axes[1, 1])
axes[1, 1].set_title('Box plot for Longitude')
plt.tight_layout()
plt.show()
```




```python
#Task 3.4 - Deal with outliers if relevant to your analysis
# Replace 'Vict Age' less than or equal to 0 with NaN values to remove outliers

df_crime['Vict Age'] = np.where(df_crime['Vict Age'] <= 0, np.nan, df_crime['Vict Age'])

df_crime
```

	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	LOCAT
0	190326475	2020-03-01	2020-03-01	21:30	7	Wilshire	784	1	510	VEHICLE - STOLEN	...	AA	Adult Arrest	510.0	998.0	NaN	NaN	19 LONGW
1	200106753	2020-02-09	2020-02-08	18:00	1	Central	182	1	330	BURGLARY FROM VEHICLE	...	IC	Invest Cont	330.0	998.0	NaN	NaN	10 FLOWEI
2	200320258	2020-11-11	2020-11-04	17:00	3	Southwest	356	1	480	BIKE - STOLEN	...	IC	Invest Cont	480.0	NaN	NaN	NaN	140 37TI
3	200907217	2023-05-10	2020-03-10	20:37	9	Van Nuys	964	1	343	SHOPLIFTING-GRAND THEFT ($950.01 & OVER)	...	IC	Invest Cont	343.0	NaN	NaN	NaN	1 RIVERS
4	220614831	2022-08-18	2020-08-17	12:00	6	Hollywood	666	2	354	THEFT OF IDENTITY	...	IC	Invest Cont	354.0	NaN	NaN	NaN	TRANSI
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
982633	242011172	2024-08-20	2024-08-17	23:00	20	Olympic	2033	1	341	THEFT-GRAND ($950.01 & OVER)EXCPT,GUNS,FOWL,LI...	...	IC	Invest Cont	341.0	NaN	NaN	NaN	WILSI
982634	240710284	2024-07-24	2024-07-23	14:00	7	Wilshire	788	1	510	VEHICLE - STOLEN	...	IC	Invest Cont	510.0	NaN	NaN	NaN	400 23RI
982635	240104953	2024-01-15	2024-01-15	01:00	1	Central	101	2	745	VANDALISM - MISDEAMEANOR ($399 OR UNDER)	...	IC	Invest Cont	745.0	NaN	NaN	NaN	130 SUNSE
982636	240309674	2024-04-24	2024-04-24	15:00	3	Southwest	358	1	230	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	...	IC	Invest Cont	230.0	NaN	NaN	NaN	FLOWEI
982637	240910892	2024-08-13	2024-08-12	23:00	9	Van Nuys	914	1	510	VEHICLE - STOLEN	...	IC	Invest Cont	510.0	NaN	NaN	NaN	VESPEI

982638 rows × 28 columns

```python
#Task 3.5 - Standardize or normalize numerical data as necessary.

num_clm = ['Vict Age', 'Rpt Dist No', 'Part 1-2', 'Premis Cd', 'LAT', 'LON']
# Standardization
scaler = StandardScaler()
df_crime[num_clm] = scaler.fit_transform(df_crime[num_clm])

# Normalization
scaler = MinMaxScaler()
df_crime[num_clm] = scaler.fit_transform(df_crime[num_clm])

df_crime
```

	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	LC
0	190326475	2020-03-01	2020-03-01	21:30	7	Wilshire	0.325548	0.0	510	VEHICLE - STOLEN	...	AA	Adult Arrest	510.0	998.0	NaN	NaN	LON
1	200106753	2020-02-09	2020-02-08	18:00	1	Central	0.038608	0.0	330	BURGLARY FROM VEHICLE	...	IC	Invest Cont	330.0	998.0	NaN	NaN	FLC
2	200320258	2020-11-11	2020-11-04	17:00	3	Southwest	0.121544	0.0	480	BIKE - STOLEN	...	IC	Invest Cont	480.0	NaN	NaN	NaN	
3	200907217	2023-05-10	2020-03-10	20:37	9	Van Nuys	0.411344	0.0	343	SHOPLIFTING-GRAND THEFT ($950.01 & OVER)	...	IC	Invest Cont	343.0	NaN	NaN	NaN	RI
4	220614831	2022-08-18	2020-08-17	12:00	6	Hollywood	0.269304	1.0	354	THEFT OF IDENTITY	...	IC	Invest Cont	354.0	NaN	NaN	NaN	TR
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
982633	242011172	2024-08-20	2024-08-17	23:00	20	Olympic	0.920877	0.0	341	THEFT-GRAND ($950.01 & OVER)EXCPT,GUNS,FOWL,LI...	...	IC	Invest Cont	341.0	NaN	NaN	NaN	W
982634	240710284	2024-07-24	2024-07-23	14:00	7	Wilshire	0.327455	0.0	510	VEHICLE - STOLEN	...	IC	Invest Cont	510.0	NaN	NaN	NaN	
982635	240104953	2024-01-15	2024-01-15	01:00	1	Central	0.000000	1.0	745	VANDALISM - MISDEAMEANOR ($399 OR UNDER)	...	IC	Invest Cont	745.0	NaN	NaN	NaN	SU
982636	240309674	2024-04-24	2024-04-24	15:00	3	Southwest	0.122498	0.0	230	ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	...	IC	Invest Cont	230.0	NaN	NaN	NaN	FLC
982637	240910892	2024-08-13	2024-08-12	23:00	9	Van Nuys	0.387512	0.0	510	VEHICLE - STOLEN	...	IC	Invest Cont	510.0	NaN	NaN	NaN	VE

982638 rows × 28 columns

```python
#Task 3.6 - Encode categorical data if present.
#There is no categorical data that required encoding.
```

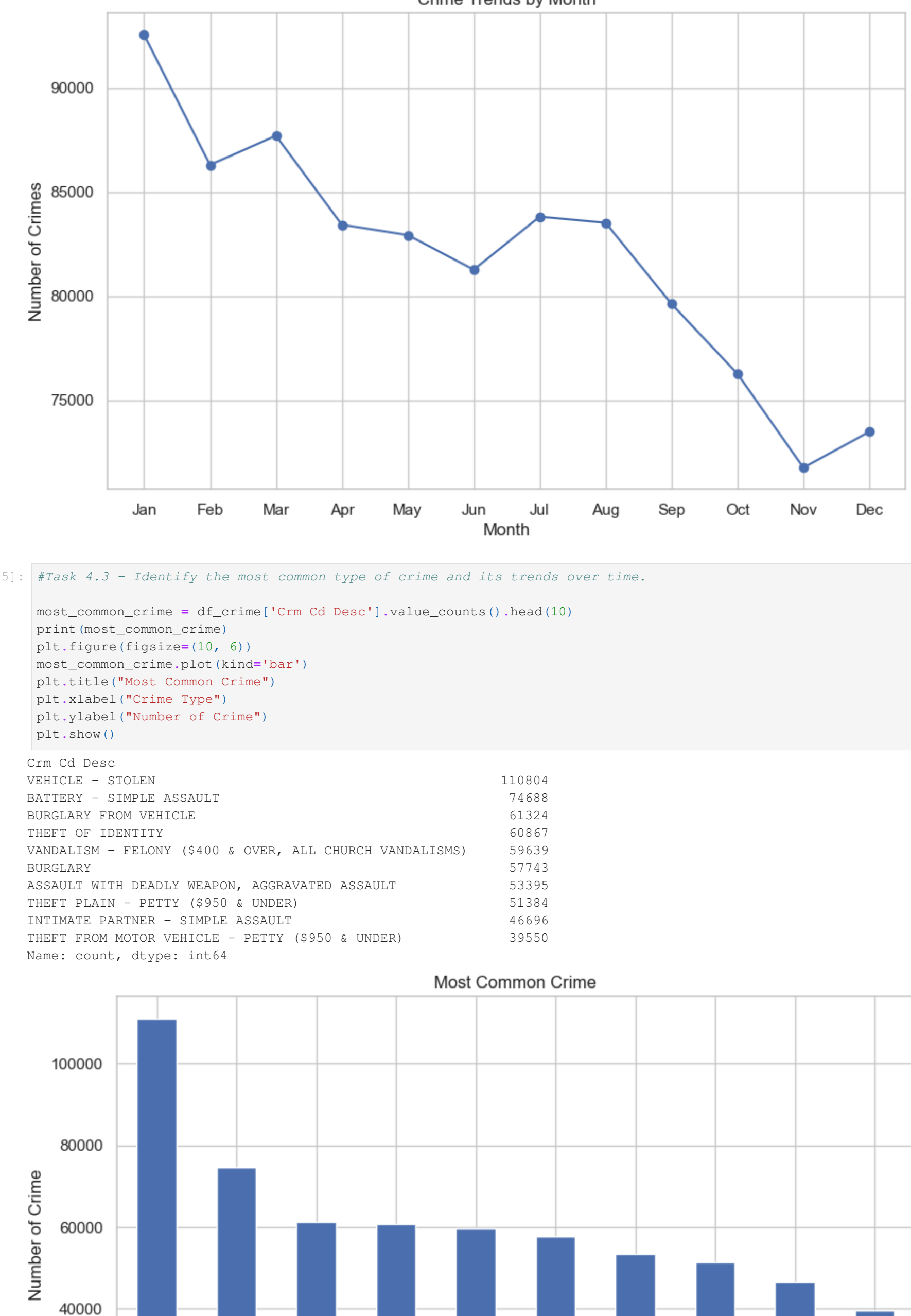
4.Exploratory Data Analysis (EDA)

```python
#Task 4.1 - Visualize overall crime trends from 2020 to the present year

df_crime['year'] = pd.to_datetime(df_crime['DATE OCC']).dt.year
crime_per_year = df_crime.groupby('year').size()

plt.figure(figsize=(8,6))
plt.plot(crime_per_year.index, crime_per_year.values, marker='o')
plt.title('Total Number of Crimes Per Year (2020 - Present)')
plt.xlabel('Year')
plt.ylabel('Number of Crimes')
plt.xticks(rotation=50)
plt.grid(True)
plt.show()
```


```python
#Task 4.2 - Analyze and visualize seasonal patterns in crime data.

df_crime['month'] = pd.to_datetime(df_crime['DATE OCC']).dt.month
crime_by_month = df_crime.groupby('month').size()

plt.figure(figsize=(10,6))
crime_by_month.plot(kind='line', marker='o')
plt.title('Crime Trends by Month')
plt.xlabel('Month')
plt.ylabel('Number of Crimes')
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.show()
```

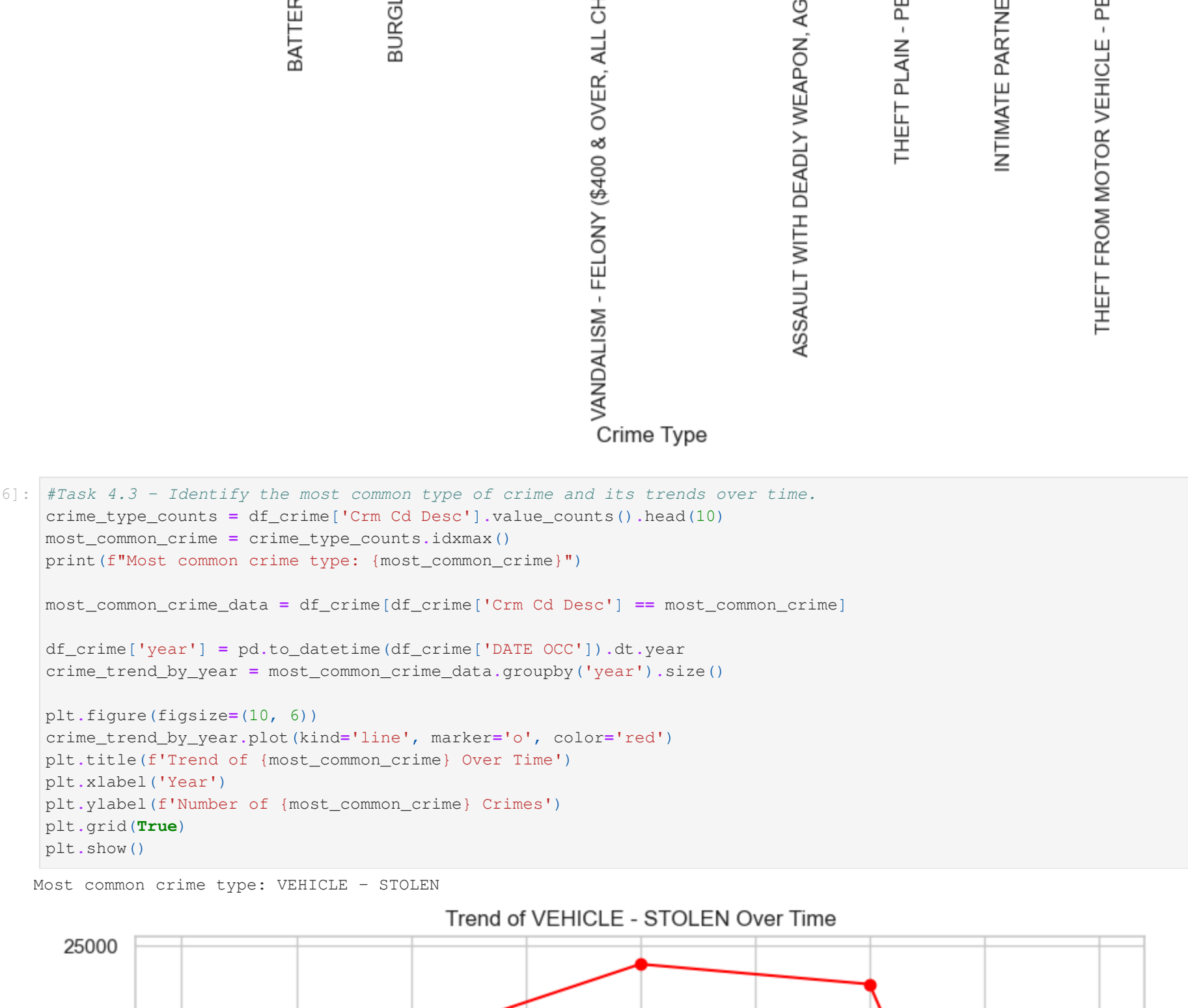

```python
#Task 4.3 - Identify the most common type of crime and its trends over time.

most_common_crime = df_crime['Crm Cd Desc'].value_counts().head(10)
print(most_common_crime)
plt.figure(figsize=(10, 6))
most_common_crime.plot(kind='bar')
plt.title("Most Common Crime")
plt.xlabel("Crime Type")
plt.ylabel("Number of Crime")
plt.show()
```

```
Crm Cd Desc
VEHICLE - STOLEN                                           110804
BATTERY - SIMPLE ASSAULT                                    74688
BURGLARY FROM VEHICLE                                       61324
THEFT OF IDENTITY                                           60867
VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)     59639
BURGLARY                                                    57743
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT              53395
THEFT PLAIN - PETTY ($950 & UNDER)                          51384
INTIMATE PARTNER - SIMPLE ASSAULT                           46696
THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)             39550
Name: count, dtype: int64
```



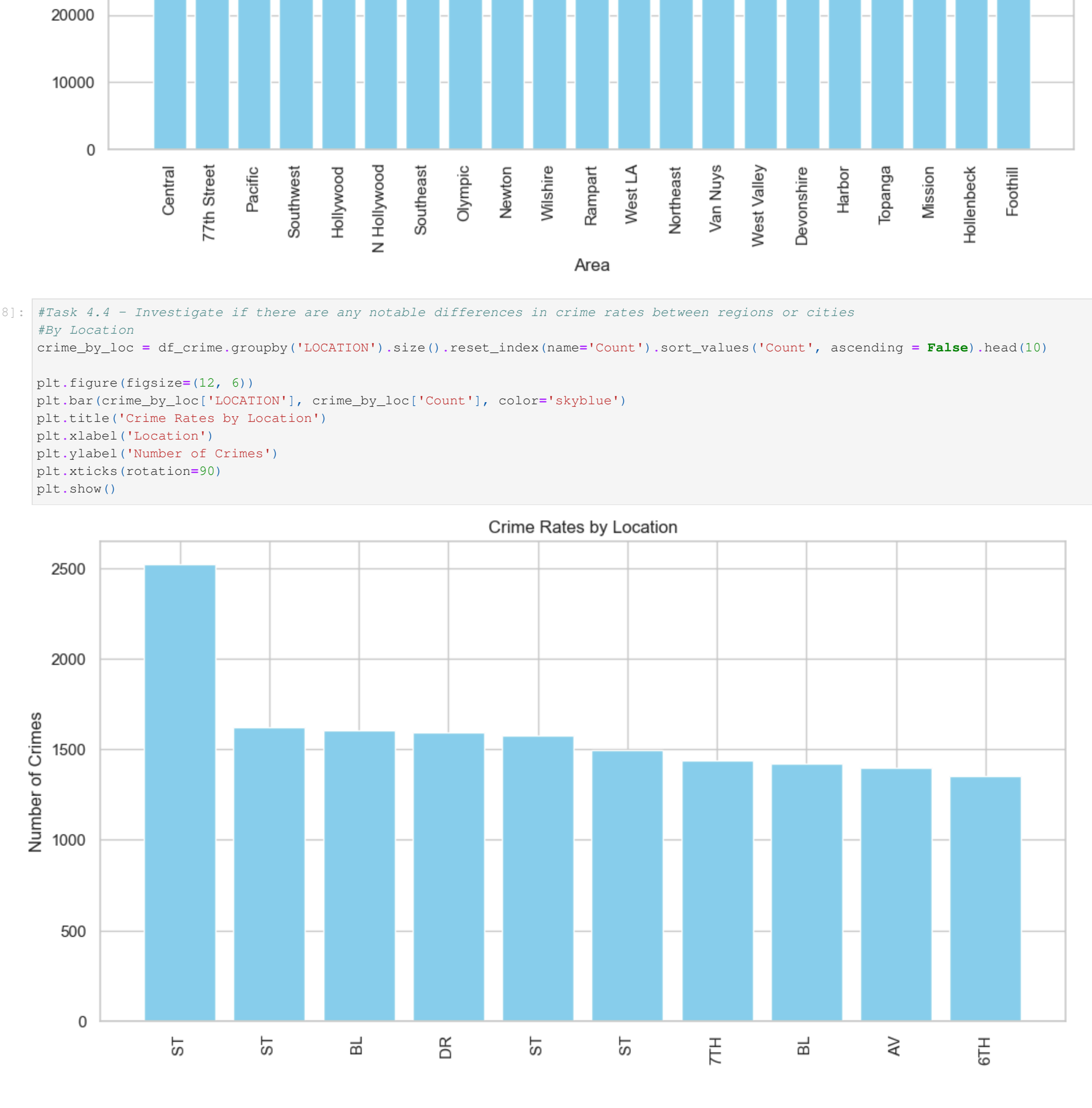

```python
#Task 4.3 - Identify the most common type of crime and its trends over time.
crime_type_counts = df_crime['Crm Cd Desc'].value_counts().head(10)
most_common_crime = crime_type_counts.idxmax()
print(f"Most common crime type: {most_common_crime}")

most_common_crime_data = df_crime[df_crime['Crm Cd Desc'] == most_common_crime]

df_crime['year'] = pd.to_datetime(df_crime['DATE OCC']).dt.year
crime_trend_by_year = most_common_crime_data.groupby('year').size()

plt.figure(figsize=(10, 6))
crime_trend_by_year.plot(kind='line', marker='o', color='red')
plt.title(f'Trend of {most_common_crime} Over Time')
plt.xlabel('Year')
plt.ylabel(f'Number of {most_common_crime} Crimes')
plt.grid(True)
plt.show()
```

```
Most common crime type: VEHICLE - STOLEN
```




```python
#Task 4.4 - Investigate if there are any notable differences in crime rates between regions or cities
#By Area
crime_by_area = df_crime.groupby('AREA NAME').size().reset_index(name='Count').sort_values('Count', ascending = False)

plt.figure(figsize=(12, 6))
plt.bar(crime_by_area['AREA NAME'], crime_by_area['Count'], color='skyblue')
plt.title('Crime Rates by Area')
plt.xlabel('Area')
plt.ylabel('Number of Crimes')
plt.xticks(rotation=90)
plt.show()
```


```python
#Task 4.4 - Investigate if there are any notable differences in crime rates between regions or cities
#By Location
crime_by_loc = df_crime.groupby('LOCATION').size().reset_index(name='Count').sort_values('Count', ascending = False).head(10)

plt.figure(figsize=(12, 6))
plt.bar(crime_by_loc['LOCATION'], crime_by_loc['Count'], color='skyblue')
plt.title('Crime Rates by Location')
plt.xlabel('Location')
plt.ylabel('Number of Crimes')
plt.xticks(rotation=90)
plt.show()
```




```python
#Task 4.5 - Explore correlations between economic factors (if available) and crime rates
#Cannot be visualized since economic data is not avaliable.
```

```python
#Task 4.6 - Analyze the relationship between the day of the week and the frequency of certain types of crimes.

df_crime['Day of Week'] = pd.to_datetime(df_crime['DATE OCC']).dt.day_name()
top_5_crimes = df_crime['Crm Cd Desc'].value_counts().head(5).index.tolist()
top_5_crime_data = df_crime[df_crime['Crm Cd Desc'].isin(top_5_crimes)]
top_5_crime_by_day = top_5_crime_data.groupby(['Day of Week', 'Crm Cd Desc']).size().reset_index(name='Count')
top_5_crime_by_day_pivot = top_5_crime_by_day.pivot(index='Day of Week', columns='Crm Cd Desc', values='Count')
ordered_days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
top_5_crime_by_day_pivot = top_5_crime_by_day_pivot.loc[ordered_days]

top_5_crime_by_day_pivot.plot(kind='line', marker='o', figsize=(14, 7))
plt.title('Frequency of Top 5 Crimes by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Crimes')
plt.grid(True, axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.legend(title='Crime Type')
plt.show()
```


```python
#Task 5 - a) Advanced Analysis
#Use predictive modeling techniques (e.g., time series forecasting) to predict future crime trends
df_crime['DATE OCC'] = pd.to_datetime(df_crime['DATE OCC'])
daily_crimes = df_crime.resample('D', on='DATE OCC').size().reset_index(name='Count')
daily_crimes.columns = ['ds', 'y']
# Fitting the Prophet model
model = Prophet()
model.fit(daily_crimes)
future = model.make_future_dataframe(periods=365)
forecast = model.predict(future)
fig = model.plot(forecast)
plt.title('Forecasting Daily Crime Trends')
plt.xlabel('Date')
plt.ylabel('Number of Crimes')
plt.legend(['Observed', 'Forecasted', 'Confidence Interval'], loc='upper left')
plt.grid(True, axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

```
01:05:37 - cmdstanpy - INFO - Chain [1] start processing
01:05:37 - cmdstanpy - INFO - Chain [1] done processing
```