IE6400_Project1_code

October 15, 2025

[48]:
```python
import pandas as pd
import io
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
```

[49]:
```python
file_name ="/content/Crime_Data_from_2020_to_Present-2.csv"
df = pd.read_csv(file_name)
display(df.head())
```

```
       DR_NO               Date Rptd                DATE OCC  TIME OCC  AREA  \
0  211507896  04/11/2021 12:00:00 AM  11/07/2020 12:00:00 AM       845    15
1  201516622  10/21/2020 12:00:00 AM  10/18/2020 12:00:00 AM      1845    15
2  240913563  12/10/2024 12:00:00 AM  10/30/2020 12:00:00 AM      1240     9
3  210704711  12/24/2020 12:00:00 AM  12/24/2020 12:00:00 AM      1310     7
4  201418201  10/03/2020 12:00:00 AM  09/29/2020 12:00:00 AM      1830    14

     AREA NAME  Rpt Dist No  Part 1-2  Crm Cd  \
0  N Hollywood         1502         2     354
1  N Hollywood         1521         1     230
2     Van Nuys          933         2     354
3     Wilshire          782         1     331
4      Pacific         1454         1     420

                                        Crm Cd Desc  … Status  Status Desc  \
0                                 THEFT OF IDENTITY  …     IC  Invest Cont
1    ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT  …     IC  Invest Cont
2                                 THEFT OF IDENTITY  …     IC  Invest Cont
3  THEFT FROM MOTOR VEHICLE - GRAND ($950.01 AND …  …     IC  Invest Cont
4   THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)  …     IC  Invest Cont

  Crm Cd 1 Crm Cd 2  Crm Cd 3 Crm Cd 4  \
0    354.0      NaN       NaN      NaN
1    230.0      NaN       NaN      NaN
2    354.0      NaN       NaN      NaN
3    331.0      NaN       NaN      NaN
4    420.0      NaN       NaN      NaN
```

```
                                    LOCATION Cross Street      LAT       LON
0   7800     BEEMAN                       AV          NaN  34.2124 -118.4092
1            ATOLL                        AV      N  GAULT  34.1993 -118.4203
2  14600     SYLVAN                       ST          NaN  34.1847 -118.4509
3   6000     COMEY                        AV          NaN  34.0339 -118.3747
4                  4700    LA VILLA MARINA          NaN  33.9813 -118.4350

[5 rows x 28 columns]
```

[50]:
```python
df.shape
```

[50]:
```
(1004991, 28)
```

[51]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1004991 entries, 0 to 1004990
Data columns (total 28 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   DR_NO           1004991 non-null  int64
 1   Date Rptd       1004991 non-null  object
 2   DATE OCC        1004991 non-null  object
 3   TIME OCC        1004991 non-null  int64
 4   AREA            1004991 non-null  int64
 5   AREA NAME       1004991 non-null  object
 6   Rpt Dist No     1004991 non-null  int64
 7   Part 1-2        1004991 non-null  int64
 8   Crm Cd          1004991 non-null  int64
 9   Crm Cd Desc     1004991 non-null  object
 10  Mocodes         853372 non-null   object
 11  Vict Age        1004991 non-null  int64
 12  Vict Sex        860347 non-null   object
 13  Vict Descent    860335 non-null   object
 14  Premis Cd       1004975 non-null  float64
 15  Premis Desc     1004403 non-null  object
 16  Weapon Used Cd  327247 non-null   float64
 17  Weapon Desc     327247 non-null   object
 18  Status          1004990 non-null  object
 19  Status Desc     1004991 non-null  object
 20  Crm Cd 1        1004980 non-null  float64
 21  Crm Cd 2        69160 non-null    float64
 22  Crm Cd 3        2314 non-null     float64
 23  Crm Cd 4        64 non-null       float64
 24  LOCATION        1004991 non-null  object
 25  Cross Street    154236 non-null   object
 26  LAT             1004991 non-null  float64
```

```
 27  LON              1004991 non-null  float64
dtypes: float64(8), int64(7), object(13)
memory usage: 214.7+ MB
```

[52]:
```
df.describe(include='all')
```

[52]:
```
               DR_NO              Date Rptd               DATE OCC  \
count   1.004991e+06                1004991                1004991
unique           NaN                   1896                   1879
top              NaN  02/02/2023 12:00:00 AM  01/01/2020 12:00:00 AM
freq             NaN                    929                   1164
mean    2.202215e+08                    NaN                    NaN
std     1.319718e+07                    NaN                    NaN
min     8.170000e+02                    NaN                    NaN
25%     2.106169e+08                    NaN                    NaN
50%     2.209159e+08                    NaN                    NaN
75%     2.311103e+08                    NaN                    NaN
max     2.521041e+08                    NaN                    NaN

            TIME OCC          AREA AREA NAME   Rpt Dist No      Part 1-2  \
count   1.004991e+06  1.004991e+06   1004991  1.004991e+06  1.004991e+06
unique           NaN           NaN        21           NaN           NaN
top              NaN           NaN   Central           NaN           NaN
freq             NaN           NaN     69670           NaN           NaN
mean    1.339900e+03  1.069174e+01       NaN  1.115633e+03  1.400348e+00
std     6.510613e+02  6.110255e+00       NaN  6.111605e+02  4.899691e-01
min     1.000000e+00  1.000000e+00       NaN  1.010000e+02  1.000000e+00
25%     9.000000e+02  5.000000e+00       NaN  5.870000e+02  1.000000e+00
50%     1.420000e+03  1.100000e+01       NaN  1.139000e+03  1.000000e+00
75%     1.900000e+03  1.600000e+01       NaN  1.613000e+03  2.000000e+00
max     2.359000e+03  2.100000e+01       NaN  2.199000e+03  2.000000e+00

              Crm Cd       Crm Cd Desc  …   Status  Status Desc  \
count   1.004991e+06           1004991  …  1004990      1004991
unique           NaN               140  …        6            6
top              NaN  VEHICLE - STOLEN  …       IC  Invest Cont
freq             NaN            115190  …   802862       802862
mean    5.001568e+02               NaN  …      NaN          NaN
std     2.052731e+02               NaN  …      NaN          NaN
min     1.100000e+02               NaN  …      NaN          NaN
25%     3.310000e+02               NaN  …      NaN          NaN
50%     4.420000e+02               NaN  …      NaN          NaN
75%     6.260000e+02               NaN  …      NaN          NaN
max     9.560000e+02               NaN  …      NaN          NaN

            Crm Cd 1      Crm Cd 2     Crm Cd 3  Crm Cd 4  \
count   1.004980e+06  69160.000000  2314.000000  64.00000
```

```
unique           NaN          NaN          NaN         NaN
top              NaN          NaN          NaN         NaN
freq             NaN          NaN          NaN         NaN
mean    4.999174e+02   958.101258   984.015990   991.21875
std     2.050736e+02   110.354348    52.350982    27.06985
min     1.100000e+02   210.000000   310.000000   821.00000
25%     3.310000e+02   998.000000   998.000000   998.00000
50%     4.420000e+02   998.000000   998.000000   998.00000
75%     6.260000e+02   998.000000   998.000000   998.00000
max     9.560000e+02   999.000000   999.000000   999.00000

                                         LOCATION Cross Street           LAT  \
count                                     1004991       154236  1.004991e+06
unique                                      66566        10413           NaN
top     800 N  ALAMEDA                         ST     BROADWAY           NaN
freq                                         2598         2486           NaN
mean                                          NaN          NaN  3.399821e+01
std                                           NaN          NaN  1.610713e+00
min                                           NaN          NaN  0.000000e+00
25%                                           NaN          NaN  3.401470e+01
50%                                           NaN          NaN  3.405890e+01
75%                                           NaN          NaN  3.416490e+01
max                                           NaN          NaN  3.433430e+01

                 LON
count   1.004991e+06
unique           NaN
top              NaN
freq             NaN
mean   -1.180909e+02
std     5.582386e+00
min    -1.186676e+02
25%    -1.184305e+02
50%    -1.183225e+02
75%    -1.182739e+02
max     0.000000e+00

[11 rows x 28 columns]
```

Already, we can see that most of the data is present. Some columns are more complete than others (as seen under count); however, we will now print the column names in a list and check for missing values with more certainty.

[53]:
```
print("\nColumn names:")
print(df.columns.tolist())
print("Missing values per column:")
print(df.isnull().sum())
```

```
Column names:
['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']
Missing values per column:
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            151619
Vict Age                0
Vict Sex           144644
Vict Descent       144656
Premis Cd              16
Premis Desc           588
Weapon Used Cd     677744
Weapon Desc        677744
Status                  1
Status Desc             0
Crm Cd 1               11
Crm Cd 2           935831
Crm Cd 3          1002677
Crm Cd 4          1004927
LOCATION                0
Cross Street       850755
LAT                     0
LON                     0
dtype: int64
```

An in-depth description of the columns was not found, so we will infer column titles and entries (most seem straightforward). Now we can start cleaning the data. We will define new dataframes as we go to ensure no duplication or confusing of dataframes.

[54]:
```python
# Cleaning: check for duplicate rows.
print("\nNumber of duplicate rows:", df.duplicated().sum())
```

```
Number of duplicate rows: 0
```

[55]:
```python
# Cleaning: convert dates and times to datetime format.
df['Date Rptd'] = pd.to_datetime(df['Date Rptd'], errors='coerce')
df['DATE OCC'] = pd.to_datetime(df['DATE OCC'], errors='coerce')
df['TIME OCC'] = df['TIME OCC'].astype(str).str.zfill(4)
df['TIME OCC'] = pd.to_datetime(df['TIME OCC'], format='%H%M', errors='coerce').
 ↪dt.time
```

```
/tmp/ipython-input-3215737464.py:2: UserWarning: Could not infer format, so each
element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df['Date Rptd'] = pd.to_datetime(df['Date Rptd'], errors='coerce')
/tmp/ipython-input-3215737464.py:3: UserWarning: Could not infer format, so each
element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df['DATE OCC'] = pd.to_datetime(df['DATE OCC'], errors='coerce')
```

[56]:
```python
# Cleaning: drop unnecessary columns:
# We will drop Area because it is redundant with Area Desc, and we will plot
# Area Desc
df = df.drop(columns=['AREA'])

# We will drop Crm Cd because they are vague and we do not have descriptions of
# them. Note that we will create our own crime descriptions from Crm Desc to
# replace and evaluate different crimes. There are also many missing values.
df = df.drop(columns=['Crm Cd', 'Crm Cd 1', 'Crm Cd 2', 'Crm Cd 3', 'Crm Cd 4'])

# Cross Street is vague and not needed. There are also many missing values.
df = df.drop(columns=['Cross Street'])

# Here, we are getting rid of more vague columns.
df = df.drop(columns=['Premis Cd', 'Weapon Used Cd', 'Status', 'Mocodes'])
```

[57]:
```python
# Fill missing values
df['Weapon Desc'] = df['Weapon Desc'].fillna("No Weapon Used")

descent_map = {
    'A': 'Other Asian',
    'B': 'Black',
    'C': 'Chinese',
    'D': 'Cambodian',
    'F': 'Filipino',
    'G': 'Guamanian',
    'H': 'Hispanic/Latin/Mexican',
    'I': 'American Indian/Alaskan Native',
    'J': 'Japanese',
    'K': 'Korean',
    'L': 'Laotian',
    'O': 'Other',
```

```
    'P': 'Pacific Islander',
    'S': 'Samoan',
    'U': 'Hawaiian',
    'V': 'Vietnamese',
    'W': 'White',
    'X': 'Unknown',
    'Z': 'Asian Indian',
    '-': 'Unknown'
}
df['Vict Descent'] = df['Vict Descent'].map(descent_map)
df['Vict Descent'] = df['Vict Descent'].fillna("Descent Unknown")

sex_map = {
    'M': 'Male',
    'F': 'Female',
    'X': 'Unknown Sex'
}

df['Vict Sex'] = df['Vict Sex'].map(sex_map).fillna("Unknown Sex")

df['Premis Desc'] = df['Premis Desc'].fillna("Unknown Premise")
```

[58]:
```
# Define column names
df["Crime_type"] = df["Part 1-2"].map({1: "Serious Crime", 2: "Less Serious␣
↪Crime"})
df = df.drop(columns=['Part 1-2'])

df["Crm Cd Desc"] = df["Crm Cd Desc"].astype(str).str.strip().str.title()
```

[59]:
```
# Break-up of ages in new category
def categorize_age(age):
    if age < 18:
        return "Juvenile"
    elif age < 60:
        return "Adult"
    else:
        return "Senior Citizen"

# Apply function to create new column
df["Age_category"] = df["Vict Age"].apply(categorize_age)
```

[60]:
```
# view the df
df
```

[60]:
```
             DR_NO  Date Rptd   DATE OCC  TIME OCC   AREA NAME  Rpt Dist No  \
0        211507896 2021-04-11 2020-11-07  08:45:00  N Hollywood         1502
1        201516622 2020-10-21 2020-10-18  18:45:00  N Hollywood         1521
```

```
2        240913563 2024-12-10 2020-10-30  12:40:00     Van Nuys          933
3        210704711 2020-12-24 2020-12-24  13:10:00     Wilshire          782
4        201418201 2020-10-03 2020-09-29  18:30:00      Pacific         1454
…              …          …          …        …            …           …
1004986  252104112 2025-02-02 2025-02-02  01:30:00      Topanga         2103
1004987  250404100 2025-02-18 2025-02-18  10:00:00   Hollenbeck          479
1004988  251304095 2025-01-31 2025-01-30  15:54:00       Newton         1372
1004989  251704066 2025-01-17 2025-01-17  16:00:00   Devonshire         1774
1004990  251904210 2025-03-25 2025-03-25  12:35:00      Mission         1944

                                               Crm Cd Desc  Vict Age Vict Sex  \
0                                        Theft Of Identity        31     Male
1           Assault With Deadly Weapon, Aggravated Assault        32     Male
2                                        Theft Of Identity        30     Male
3        Theft From Motor Vehicle - Grand ($950.01 And …        47   Female
4          Theft From Motor Vehicle - Petty ($950 & Under)        63     Male
…                                                      …       …       …
1004986                          Other Miscellaneous Crime        35     Male
1004987                        Child Neglect (See 300 W.I.C.)     11     Male
1004988                                  Indecent Exposure        16   Female
1004989                           Battery - Simple Assault        17     Male
1004990                                  Indecent Exposure        35   Female

                   Vict Descent             Premis Desc  \
0        Hispanic/Latin/Mexican  SINGLE FAMILY DWELLING
1        Hispanic/Latin/Mexican                SIDEWALK
2                         White  SINGLE FAMILY DWELLING
3                   Other Asian                  STREET
4        Hispanic/Latin/Mexican                   ALLEY
…                           …                     …
1004986                 Unknown                  STREET
1004987                   Black  SINGLE FAMILY DWELLING
1004988  Hispanic/Latin/Mexican                  STREET
1004989  Hispanic/Latin/Mexican             HIGH SCHOOL
1004990  Hispanic/Latin/Mexican             HIGH SCHOOL

                                            Weapon Desc  Status Desc  \
0                                        No Weapon Used  Invest Cont
1                        KNIFE WITH BLADE 6INCHES OR LESS  Invest Cont
2                                        No Weapon Used  Invest Cont
3                                        No Weapon Used  Invest Cont
4                                        No Weapon Used  Invest Cont
…                                                    …          …
1004986                                  No Weapon Used  Invest Cont
1004987                                  No Weapon Used  Invest Cont
1004988                                  No Weapon Used  Invest Cont
1004989  STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)  Invest Cont
```

```
1004990                                  No Weapon Used  Invest Cont

                                         LOCATION      LAT       LON  \
0         7800    BEEMAN                       AV  34.2124 -118.4092
1                 ATOLL                        AV  34.1993 -118.4203
2        14600    SYLVAN                       ST  34.1847 -118.4509
3         6000    COMEY                        AV  34.0339 -118.3747
4                         4700    LA VILLA MARINA  33.9813 -118.4350
…                                         …        …         …
1004986  22100    ROSCOE                       BL  34.2259 -118.6126
1004987   3500    PERCY                        ST  34.0277 -118.1979
1004988    300 E  53RD                         ST  33.9942 -118.2701
1004989   9600    ZELZAH                       AV  34.2450 -118.5233
1004990  11100    OMELVENY                     AV  34.2722 -118.4417

                  Crime_type    Age_category
0        Less Serious Crime           Adult
1             Serious Crime           Adult
2        Less Serious Crime           Adult
3             Serious Crime           Adult
4             Serious Crime  Senior Citizen
…                       …              …
1004986  Less Serious Crime           Adult
1004987  Less Serious Crime        Juvenile
1004988  Less Serious Crime        Juvenile
1004989  Less Serious Crime        Juvenile
1004990  Less Serious Crime           Adult

[1004991 rows x 18 columns]
```

[61]:
```
cleaned_df = df
cleaned_df
```

[61]:
```
             DR_NO  Date Rptd   DATE OCC   TIME OCC    AREA NAME  Rpt Dist No  \
0        211507896 2021-04-11 2020-11-07  08:45:00  N Hollywood         1502
1        201516622 2020-10-21 2020-10-18  18:45:00  N Hollywood         1521
2        240913563 2024-12-10 2020-10-30  12:40:00     Van Nuys          933
3        210704711 2020-12-24 2020-12-24  13:10:00     Wilshire          782
4        201418201 2020-10-03 2020-09-29  18:30:00      Pacific         1454
…              …         …         …         …           …           …
1004986  252104112 2025-02-02 2025-02-02  01:30:00      Topanga         2103
1004987  250404100 2025-02-18 2025-02-18  10:00:00   Hollenbeck          479
1004988  251304095 2025-01-31 2025-01-30  15:54:00       Newton         1372
1004989  251704066 2025-01-17 2025-01-17  16:00:00   Devonshire         1774
1004990  251904210 2025-03-25 2025-03-25  12:35:00      Mission         1944

                                              Crm Cd Desc  Vict Age Vict Sex  \
```

```
0                                       Theft Of Identity        31     Male
1            Assault With Deadly Weapon, Aggravated Assault        32     Male
2                                       Theft Of Identity        30     Male
3         Theft From Motor Vehicle - Grand ($950.01 And …        47   Female
4           Theft From Motor Vehicle - Petty ($950 & Under)        63     Male
…                                                     …       …        …
1004986                         Other Miscellaneous Crime        35     Male
1004987                      Child Neglect (See 300 W.I.C.)        11     Male
1004988                                  Indecent Exposure        16   Female
1004989                             Battery - Simple Assault        17     Male
1004990                                  Indecent Exposure        35   Female

                   Vict Descent            Premis Desc  \
0        Hispanic/Latin/Mexican  SINGLE FAMILY DWELLING
1        Hispanic/Latin/Mexican                SIDEWALK
2                         White  SINGLE FAMILY DWELLING
3                   Other Asian                  STREET
4        Hispanic/Latin/Mexican                   ALLEY
…                             …                       …
1004986                 Unknown                  STREET
1004987                   Black  SINGLE FAMILY DWELLING
1004988  Hispanic/Latin/Mexican                  STREET
1004989  Hispanic/Latin/Mexican             HIGH SCHOOL
1004990  Hispanic/Latin/Mexican             HIGH SCHOOL

                                            Weapon Desc  Status Desc  \
0                                        No Weapon Used  Invest Cont
1                    KNIFE WITH BLADE 6INCHES OR LESS  Invest Cont
2                                        No Weapon Used  Invest Cont
3                                        No Weapon Used  Invest Cont
4                                        No Weapon Used  Invest Cont
…                                                     …           …
1004986                                  No Weapon Used  Invest Cont
1004987                                  No Weapon Used  Invest Cont
1004988                                  No Weapon Used  Invest Cont
1004989  STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)  Invest Cont
1004990                                  No Weapon Used  Invest Cont

                                        LOCATION      LAT       LON  \
0          7800    BEEMAN                       AV  34.2124 -118.4092
1                  ATOLL                        AV  34.1993 -118.4203
2         14600    SYLVAN                       ST  34.1847 -118.4509
3          6000    COMEY                        AV  34.0339 -118.3747
4                          4700    LA VILLA MARINA  33.9813 -118.4350
…                                              …        …         …
1004986   22100    ROSCOE                       BL  34.2259 -118.6126
1004987    3500    PERCY                        ST  34.0277 -118.1979
```

```
1004988    300 E   53RD                         ST  33.9942 -118.2701
1004989   9600     ZELZAH                       AV  34.2450 -118.5233
1004990  11100     OMELVENY                     AV  34.2722 -118.4417

                 Crime_type     Age_category
0        Less Serious Crime            Adult
1             Serious Crime            Adult
2        Less Serious Crime            Adult
3             Serious Crime            Adult
4             Serious Crime   Senior Citizen
…                       …              …
1004986  Less Serious Crime            Adult
1004987  Less Serious Crime         Juvenile
1004988  Less Serious Crime         Juvenile
1004989  Less Serious Crime         Juvenile
1004990  Less Serious Crime            Adult

[1004991 rows x 18 columns]
```

[62]:

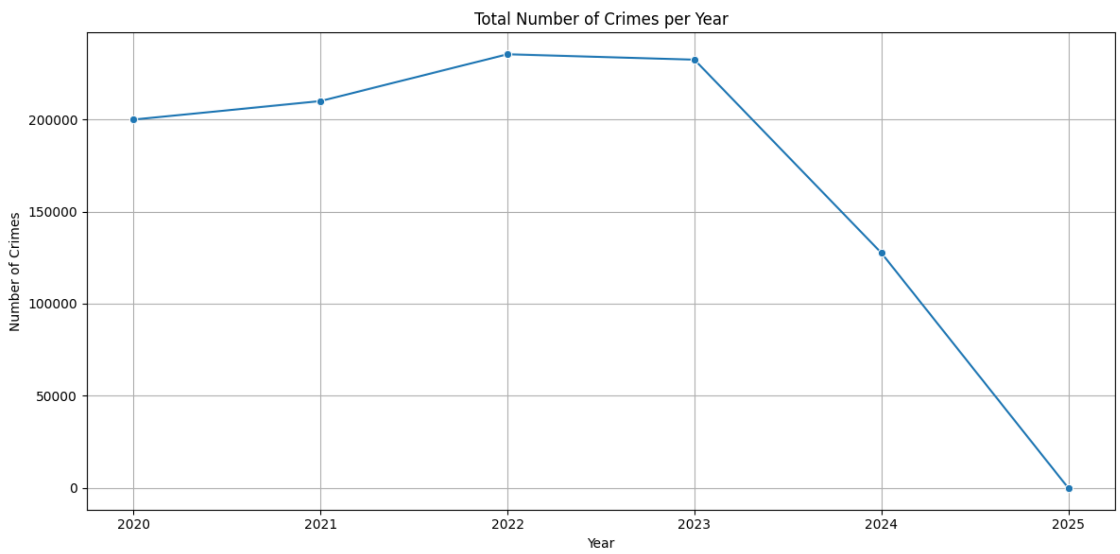
```python
print("Missing values per column:")
print(cleaned_df.isnull().sum())
```

```
Missing values per column:
DR_NO           0
Date Rptd       0
DATE OCC        0
TIME OCC        0
AREA NAME       0
Rpt Dist No     0
Crm Cd Desc     0
Vict Age        0
Vict Sex        0
Vict Descent    0
Premis Desc     0
Weapon Desc     0
Status Desc     0
LOCATION        0
LAT             0
LON             0
Crime_type      0
Age_category    0
dtype: int64
```

Data cleaning has been completed. Now start on Data Analysis.

1. Overall Crime Trends: • Calculate and plot the total number of crimes per year to visualize the trends

[63]:
```
# Extract the year from 'DATE OCC'
cleaned_df['Year'] = cleaned_df['DATE OCC'].dt.year

crimes_per_year = cleaned_df['Year'].value_counts().sort_index()

plt.figure(figsize=(12, 6))
sns.lineplot(x = crimes_per_year.index, y = crimes_per_year.values, marker =␣
↪'o')

plt.title('Total Number of Crimes per Year')
plt.xlabel('Year')
plt.ylabel('Number of Crimes')
plt.grid(True)
plt.tight_layout()
plt.show()
```



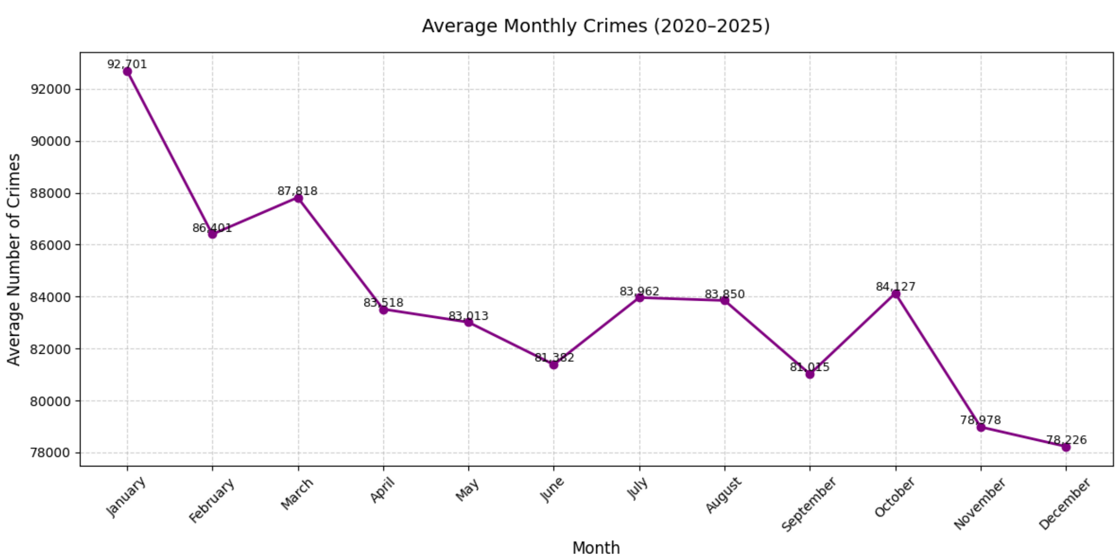
Crime peaks in 2022 and 2023 before decreasing in 2024. The steep descrease in 2025 may be the result of missing or incomplete data due to LAPD issues as mentioned in the data description.

2. Seasonal Patterns: • Group the data by month and analyze the average number of monthly crimes over the years.

[64]:
```
cleaned_df['Month'] = cleaned_df['DATE OCC'].dt.month_name()
monthly_crimes = cleaned_df.groupby(["Month"]).size().
↪reset_index(name="Total_Crimes")

month_order = [
    "January", "February", "March", "April", "May", "June",
    "July", "August", "September", "October", "November", "December"
```

```
]
monthly_crimes["Month"] = pd.Categorical(monthly_crimes["Month"],␣
↪categories=month_order, ordered=True)
avg_monthly = monthly_crimes.groupby("Month")["Total_Crimes"].mean().
↪reset_index()


plt.figure(figsize=(12,6))
plt.plot(avg_monthly["Month"], avg_monthly["Total_Crimes"], marker='o',␣
↪linewidth=2, color='purple')

# Add labels
for i, row in avg_monthly.iterrows():
    plt.text(row["Month"], row["Total_Crimes"] + 100,
             f"{int(row['Total_Crimes']):,}", ha='center', fontsize=9)

plt.title("Average Monthly Crimes (2020-2025)", fontsize=14, pad=15)
plt.xlabel("Month", fontsize=12)
plt.ylabel("Average Number of Crimes", fontsize=12)
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-2306259654.py:9: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass
observed=False to retain current behavior or observed=True to adopt the future
default and silence this warning.
  avg_monthly =
monthly_crimes.groupby("Month")["Total_Crimes"].mean().reset_index()
```



Monthly crimes in greatest in January and least in December, showing decreasing crime throughout the year. This may be the result of efforts to crack down on crime throughout the year; otherwise, January may simply be an outlier as most of the data is clumped together.

[65]:
```python
cleaned_df['Monthly_count'] = cleaned_df['DATE OCC'].dt.to_period('M').dt.
↪to_timestamp()
cleaned_df
```

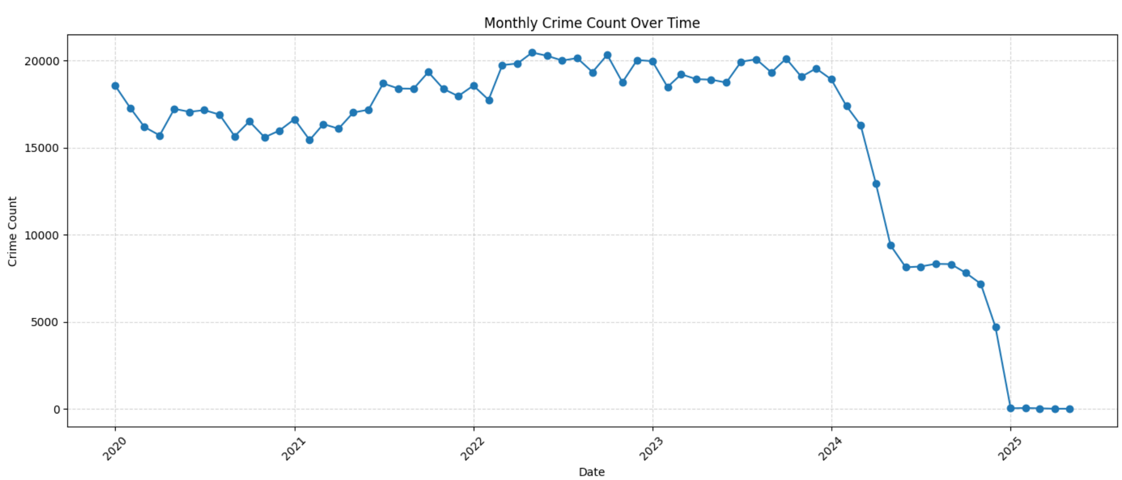
[65]:
```
             DR_NO  Date Rptd   DATE OCC  TIME OCC     AREA NAME  Rpt Dist No  \
0        211507896 2021-04-11 2020-11-07  08:45:00  N Hollywood         1502
1        201516622 2020-10-21 2020-10-18  18:45:00  N Hollywood         1521
2        240913563 2024-12-10 2020-10-30  12:40:00     Van Nuys          933
3        210704711 2020-12-24 2020-12-24  13:10:00     Wilshire          782
4        201418201 2020-10-03 2020-09-29  18:30:00      Pacific         1454
…              …          …          …        …            …           …
1004986  252104112 2025-02-02 2025-02-02  01:30:00      Topanga         2103
1004987  250404100 2025-02-18 2025-02-18  10:00:00   Hollenbeck          479
1004988  251304095 2025-01-31 2025-01-30  15:54:00       Newton         1372
1004989  251704066 2025-01-17 2025-01-17  16:00:00   Devonshire         1774
1004990  251904210 2025-03-25 2025-03-25  12:35:00      Mission         1944

                                              Crm Cd Desc  Vict Age Vict Sex  \
0                                       Theft Of Identity        31     Male
1            Assault With Deadly Weapon, Aggravated Assault        32     Male
2                                       Theft Of Identity        30     Male
3        Theft From Motor Vehicle - Grand ($950.01 And …       47   Female
4          Theft From Motor Vehicle - Petty ($950 & Under)        63     Male
…                                                     …       …      …
1004986                          Other Miscellaneous Crime        35     Male
1004987                      Child Neglect (See 300 W.I.C.)        11     Male
1004988                                  Indecent Exposure        16   Female
1004989                            Battery - Simple Assault        17     Male
1004990                                  Indecent Exposure        35   Female

                   Vict Descent  …  \
0        Hispanic/Latin/Mexican  …
1        Hispanic/Latin/Mexican  …
2                         White  …
3                   Other Asian  …
4        Hispanic/Latin/Mexican  …
…                            …  …
1004986                 Unknown  …
1004987                   Black  …
1004988  Hispanic/Latin/Mexican  …
1004989  Hispanic/Latin/Mexican  …
1004990  Hispanic/Latin/Mexican  …
```

```
                                          Weapon Desc  Status Desc  \
0                                      No Weapon Used  Invest Cont
1                      KNIFE WITH BLADE 6INCHES OR LESS  Invest Cont
2                                      No Weapon Used  Invest Cont
3                                      No Weapon Used  Invest Cont
4                                      No Weapon Used  Invest Cont
…                                                   …           …
1004986                                No Weapon Used  Invest Cont
1004987                                No Weapon Used  Invest Cont
1004988                                No Weapon Used  Invest Cont
1004989  STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)  Invest Cont
1004990                                No Weapon Used  Invest Cont

                                    LOCATION      LAT       LON  \
0         7800    BEEMAN                    AV  34.2124 -118.4092
1                 ATOLL                     AV  34.1993 -118.4203
2        14600    SYLVAN                    ST  34.1847 -118.4509
3         6000    COMEY                     AV  34.0339 -118.3747
4                        4700    LA VILLA MARINA  33.9813 -118.4350
…                                         …        …        …
1004986  22100    ROSCOE                    BL  34.2259 -118.6126
1004987   3500    PERCY                     ST  34.0277 -118.1979
1004988    300 E  53RD                      ST  33.9942 -118.2701
1004989   9600    ZELZAH                    AV  34.2450 -118.5233
1004990  11100    OMELVENY                  AV  34.2722 -118.4417

                 Crime_type    Age_category  Year      Month Monthly_count
0        Less Serious Crime           Adult  2020   November    2020-11-01
1             Serious Crime           Adult  2020    October    2020-10-01
2        Less Serious Crime           Adult  2020    October    2020-10-01
3             Serious Crime           Adult  2020   December    2020-12-01
4             Serious Crime  Senior Citizen  2020  September    2020-09-01
…                       …             …     …        …           …
1004986  Less Serious Crime           Adult  2025   February    2025-02-01
1004987  Less Serious Crime        Juvenile  2025   February    2025-02-01
1004988  Less Serious Crime        Juvenile  2025    January    2025-01-01
1004989  Less Serious Crime        Juvenile  2025    January    2025-01-01
1004990  Less Serious Crime           Adult  2025      March    2025-03-01

[1004991 rows x 21 columns]
```

[66]:
```
monthly_crime_counts = cleaned_df.groupby('Monthly_count').size().
  ↪reset_index(name='crime_count')
monthly_crime_counts
```

[66]:
```
    Monthly_count  crime_count
0      2020-01-01        18576
1      2020-02-01        17284
2      2020-03-01        16188
3      2020-04-01        15706
4      2020-05-01        17230
..            …            …
60     2025-01-01           26
61     2025-02-01           44
62     2025-03-01           24
63     2025-04-01            1
64     2025-05-01            2

[65 rows x 2 columns]
```

[67]:
```python
plt.figure(figsize=(14, 6))
plt.plot(monthly_crime_counts['Monthly_count'],␣
 ↪monthly_crime_counts['crime_count'], marker='o', linestyle='-')

# Labels and formatting
plt.title('Monthly Crime Count Over Time')
plt.xlabel('Date')
plt.ylabel('Crime Count')
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()

plt.show()
```



Crime seemed to be slightly increasing monthly until 2024, when it began to decline. The decline in 2025 is likely due to LAPD reporting and publication issues, as mentioned earlier.

3. Most Common Crime Type: • Count the occurrences of each crime type and identify the one with the highest frequency

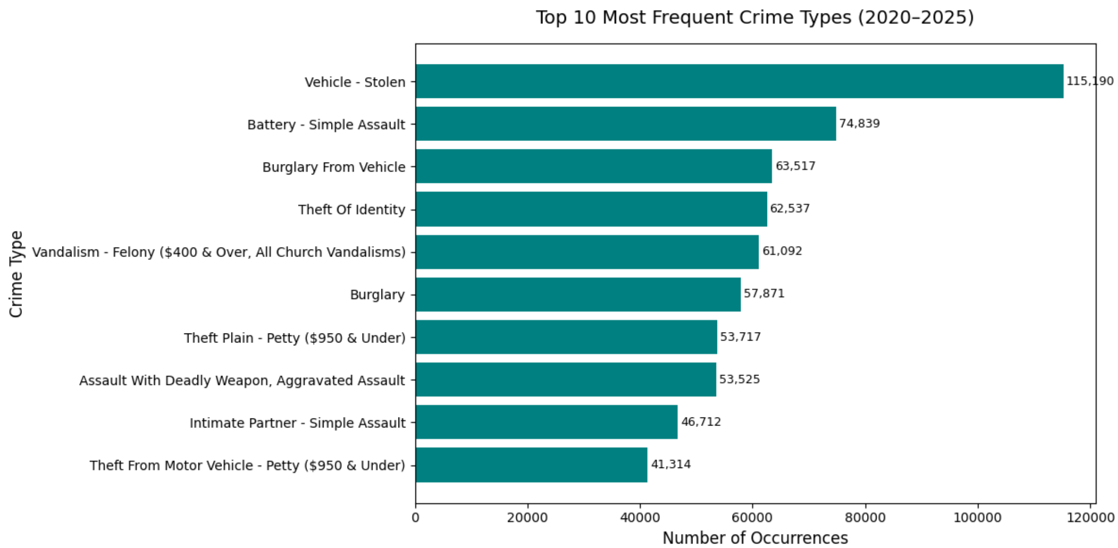
[68]:

```
crime_counts = cleaned_df["Crm Cd Desc"].value_counts().reset_index()
crime_counts.columns = ["Crime_Type", "Count"]

top10_crimes = crime_counts.head(10)
print(top10_crimes)

most_frequent_crime = top10_crimes.iloc[0]
print(f"\n Most frequent crime: {most_frequent_crime['Crime_Type']} "
      f"({most_frequent_crime['Count']:,} occurrences)")
```

```
                                         Crime_Type   Count
0                                  Vehicle - Stolen  115190
1                          Battery - Simple Assault   74839
2                             Burglary From Vehicle   63517
3                                 Theft Of Identity   62537
4  Vandalism - Felony ($400 & Over, All Church Va…   61092
5                                          Burglary   57871
6                 Theft Plain - Petty ($950 & Under)   53717
7    Assault With Deadly Weapon, Aggravated Assault   53525
8                 Intimate Partner - Simple Assault   46712
9   Theft From Motor Vehicle - Petty ($950 & Under)   41314

 Most frequent crime: Vehicle - Stolen (115,190 occurrences)
```

[69]:

```
plt.figure(figsize=(12,6))
plt.barh(top10_crimes["Crime_Type"][::-1], top10_crimes["Count"][::-1],␣
 ↪color='teal')
plt.title("Top 10 Most Frequent Crime Types (2020-2025)", fontsize=14, pad=15)
plt.xlabel("Number of Occurrences", fontsize=12)
plt.ylabel("Crime Type", fontsize=12)

for index, value in enumerate(top10_crimes["Count"][::-1]):
    plt.text(value + 500, index, f"{value:,}", va='center', fontsize=9)

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



A stolen vehicle is the most frequent crime in the last six years, with simple assault and burglary from vehicle the next two common, respectively.

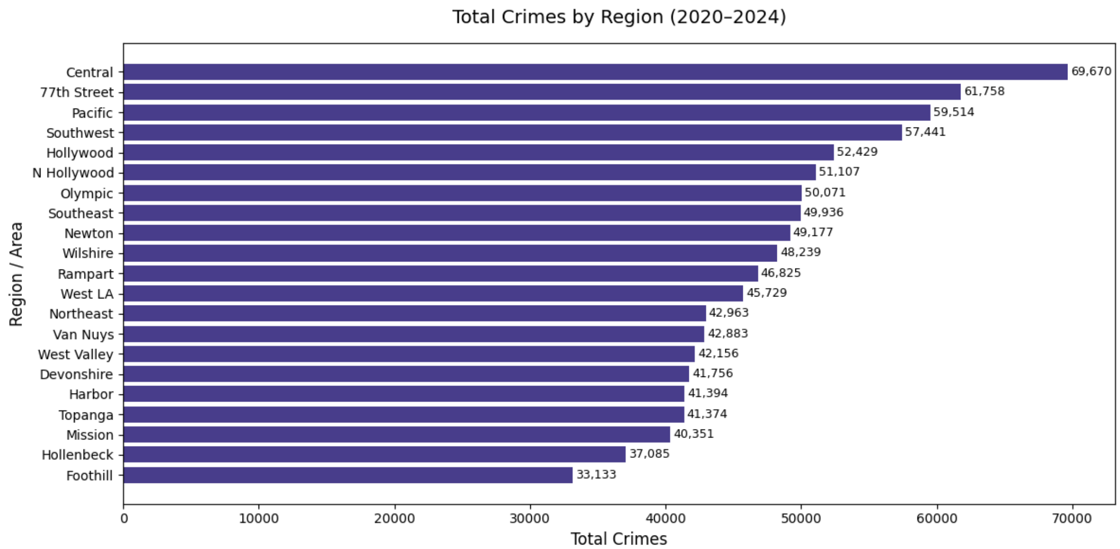
4. Regional Differences: • Group the data by region or city and compare crime rates using descriptive statistics or visualizations.

[70]:
```
region_crimes = cleaned_df.groupby("AREA NAME").size().
    ↪reset_index(name="Total_Crimes")

region_crimes = region_crimes.sort_values(by="Total_Crimes", ascending=False)

print("  Descriptive Statistics for Regional Crime Counts:")
print(region_crimes["Total_Crimes"].describe())
```

```
 Descriptive Statistics for Regional Crime Counts:
count       21.000000
mean     47856.714286
std       8764.537336
min      33133.000000
25%      41756.000000
50%      46825.000000
75%      51107.000000
max      69670.000000
Name: Total_Crimes, dtype: float64
```

[71]:
```
plt.figure(figsize=(12,6))
plt.barh(region_crimes["AREA NAME"][::-1], region_crimes["Total_Crimes"][::-1],␣
    ↪color='darkslateblue')
plt.title("Total Crimes by Region (2020-2024)", fontsize=14, pad=15)
plt.xlabel("Total Crimes", fontsize=12)
```

```python
plt.ylabel("Region / Area", fontsize=12)

for index, value in enumerate(region_crimes["Total_Crimes"][::-1]):
    plt.text(value + 200, index, f"{value:,}", va='center', fontsize=9)

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



Crime is greatest in Central, while Foothill appears to be the safest (least amount of crime) region.

5. Correlation with Economic Factors: • Collect economic data for the same time frame and use statistical methods, such as correlation analysis, to assess the relationship between economic factors and crime rates.

Here, we will recall our code from question #2 (monthly crime counts).

[72]:
```python
monthly_crime_counts
```

[72]:

	Monthly_count	crime_count
0	2020-01-01	18576
1	2020-02-01	17284
2	2020-03-01	16188
3	2020-04-01	15706
4	2020-05-01	17230
..	…	…
60	2025-01-01	26
61	2025-02-01	44
62	2025-03-01	24
63	2025-04-01	1
64	2025-05-01	2

```
[65 rows x 2 columns]
```

Here, below, we are importing verified data about LA unemployment from the Federal Reserve Bank of St. Louis (FRED).

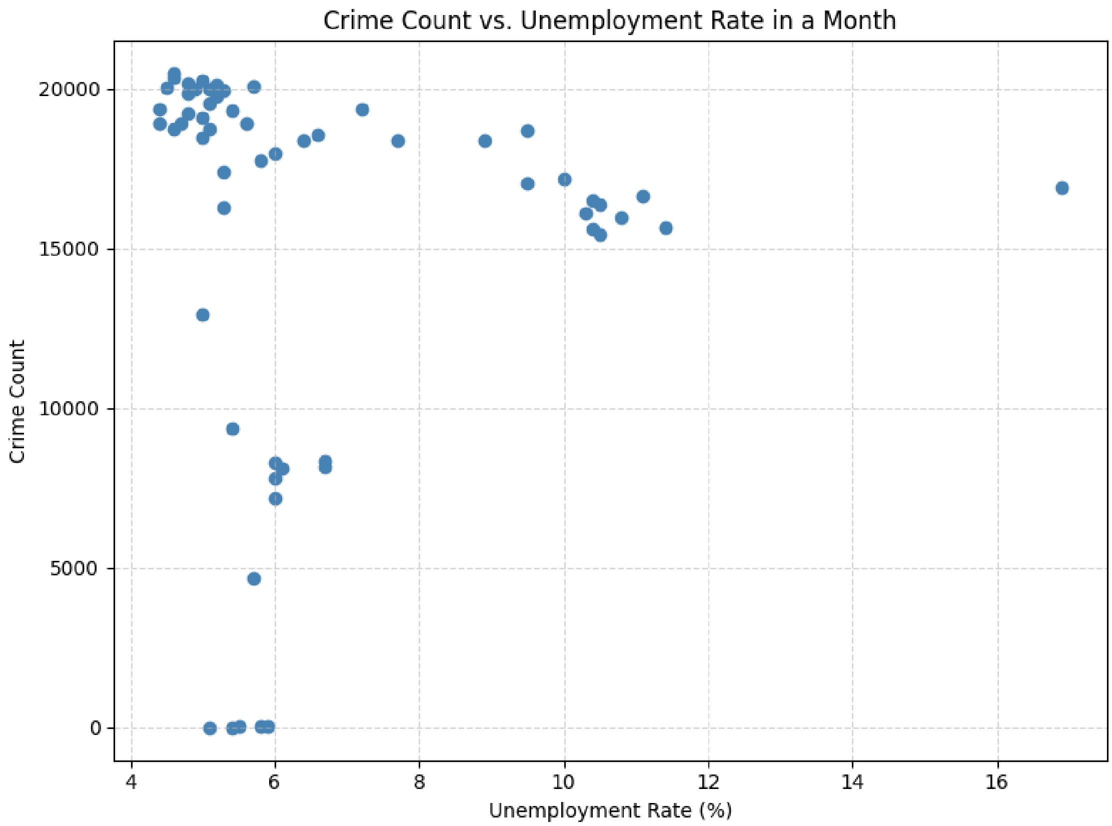
[73]:
```
la_unemployment = pd.read_csv('CALOSA7URN.csv')
la_unemployment.rename(columns={'observation_date': 'Monthly_count'},␣
 ↪inplace=True)
la_unemployment['Monthly_count'] = pd.
 ↪to_datetime(la_unemployment['Monthly_count'])

merged = pd.merge(la_unemployment, monthly_crime_counts, on='Monthly_count',␣
 ↪how='inner')
merged
```

[73]:

	Monthly_count	CALOSA7URN	crime_count
0	2020-08-01	16.9	16902
1	2020-09-01	11.4	15658
2	2020-10-01	10.4	16510
3	2020-11-01	10.4	15596
4	2020-12-01	10.8	15979
5	2021-01-01	11.1	16636
6	2021-02-01	10.5	15440
7	2021-03-01	10.5	16354
8	2021-04-01	10.3	16091
9	2021-05-01	9.5	17020
10	2021-06-01	10.0	17182
11	2021-07-01	9.5	18690
12	2021-08-01	8.9	18398
13	2021-09-01	7.7	18386
14	2021-10-01	7.2	19343
15	2021-11-01	6.4	18374
16	2021-12-01	6.0	17962
17	2022-01-01	6.6	18567
18	2022-02-01	5.8	17750
19	2022-03-01	5.2	19745
20	2022-04-01	4.8	19837
21	2022-05-01	4.6	20467
22	2022-06-01	5.0	20273
23	2022-07-01	4.9	20009
24	2022-08-01	4.8	20144
25	2022-09-01	4.4	19341
26	2022-10-01	4.6	20335
27	2022-11-01	4.6	18755
28	2022-12-01	4.5	20036
29	2023-01-01	5.1	19970

30	2023-02-01	5.0	18489
31	2023-03-01	4.8	19214
32	2023-04-01	4.4	18937
33	2023-05-01	4.7	18908
34	2023-06-01	5.1	18741
35	2023-07-01	5.3	19935
36	2023-08-01	5.7	20082
37	2023-09-01	5.4	19321
38	2023-10-01	5.2	20122
39	2023-11-01	5.0	19074
40	2023-12-01	5.1	19552
41	2024-01-01	5.6	18926
42	2024-02-01	5.3	17394
43	2024-03-01	5.3	16293
44	2024-04-01	5.0	12946
45	2024-05-01	5.4	9386
46	2024-06-01	6.1	8126
47	2024-07-01	6.7	8170
48	2024-08-01	6.7	8324
49	2024-09-01	6.0	8309
50	2024-10-01	6.0	7817
51	2024-11-01	6.0	7179
52	2024-12-01	5.7	4697
53	2025-01-01	5.8	26
54	2025-02-01	5.9	44
55	2025-03-01	5.5	24
56	2025-04-01	5.1	1
57	2025-05-01	5.4	2

[74]:
```
plt.figure(figsize=(8, 6))
plt.scatter(merged['CALOSA7URN'], merged['crime_count'], color='steelblue')

# Labels and title
plt.xlabel('Unemployment Rate (%)')
plt.ylabel('Crime Count')
plt.title('Crime Count vs. Unemployment Rate in a Month')
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```



It appears as though unemployment rate and number of crimes in a month are negatively correlated. This might have something to do with Covid-19 and less crime during that year when unemployment rates were higher. Many of the values in the lower left are probably from 2025 which has seen a steep drop in crime. Whether this is due to misreporting or actually drops in crime is hard to tell since 2025 is the current year.

6. Day of the Week Analysis: • Group the data by day of the week and analyze crime frequencies for each day.

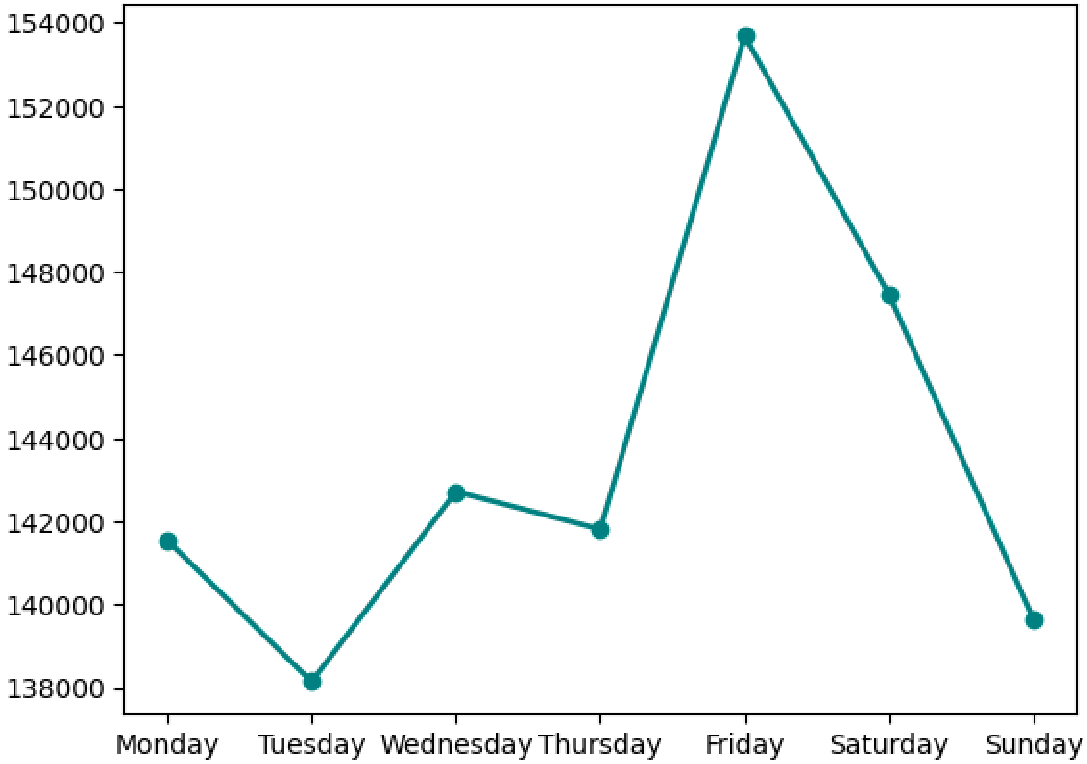
[75]:
```
cleaned_df["DayOfWeek"] = cleaned_df["DATE OCC"].dt.day_name()

dow_order =␣
 ↪["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]
df["DayOfWeek"] = pd.Categorical(df["DayOfWeek"], categories=dow_order,␣
 ↪ordered=True)

by_dow = (
    df.groupby("DayOfWeek")
      .size()
      .reset_index(name="Crime_Count")
      .sort_values("DayOfWeek")
```

```
)

total = by_dow["Crime_Count"].sum()
by_dow["Share_%"] = (by_dow["Crime_Count"] / total * 100).round(2)

print("Crime frequency by day of week (2020-2024):")
print(by_dow)


plt.plot(by_dow["DayOfWeek"], by_dow["Crime_Count"], marker='o', linewidth=2,␣
↪color='teal')
```

```
Crime frequency by day of week (2020-2024):
   DayOfWeek  Crime_Count  Share_%
0     Monday       141543    14.08
1    Tuesday       138141    13.75
2  Wednesday       142714    14.20
3   Thursday       141810    14.11
4     Friday       153676    15.29
5   Saturday       147459    14.67
6     Sunday       139648    13.90
```

```
/tmp/ipython-input-1907830983.py:7: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass
observed=False to retain current behavior or observed=True to adopt the future
default and silence this warning.
  df.groupby("DayOfWeek")
```

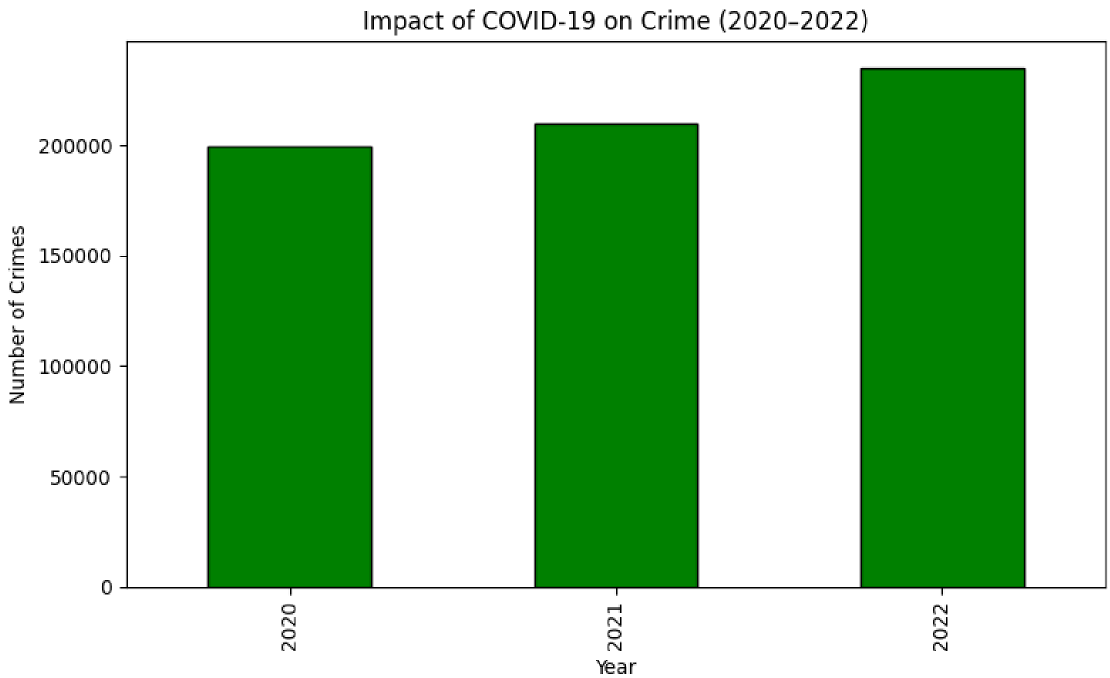
```
[75]: [<matplotlib.lines.Line2D at 0x7b0001e5e360>]
```



Most crimes occur over the weekend (Friday and Saturday), while they occur least on Tuesdays. This is likely due to more frequent travel or outings on Fridays and Saturdays.

7. Impact of Major Events: • Identify significant events or policy changes during the dataset period and analyze crime rate changes before and after these events.

[76]:
```
df_event = df[df['Year'].between(2020, 2022)]
plt.figure(figsize=(8,5))
df_event['Year'].value_counts().sort_index().plot(kind='bar', color='green',␣
↪edgecolor='black')
plt.title("Impact of COVID-19 on Crime (2020-2022)")
plt.xlabel("Year")
plt.ylabel("Number of Crimes")
plt.tight_layout()
plt.show()
```



As Covid-19 subsided and stay-at-home orders were lifted, crime rose in LA.

8. Outliers and Anomalies: • Use statistical methods or data visualization techniques to identify dataset outliers and investigate unusual patterns.

[77]:
```
# Recall Monthly_count and monthly_crime_counts.

Q1 = monthly_crime_counts["crime_count"].quantile(0.25)
Q3 = monthly_crime_counts["crime_count"].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers_iqr = monthly_crime_counts[
    (monthly_crime_counts["crime_count"] < lower_bound) |␣
 ↪(monthly_crime_counts["crime_count"] > upper_bound)
]
print("  Outlier Months (Unusual Crime Activity):")
print(outliers_iqr)
```

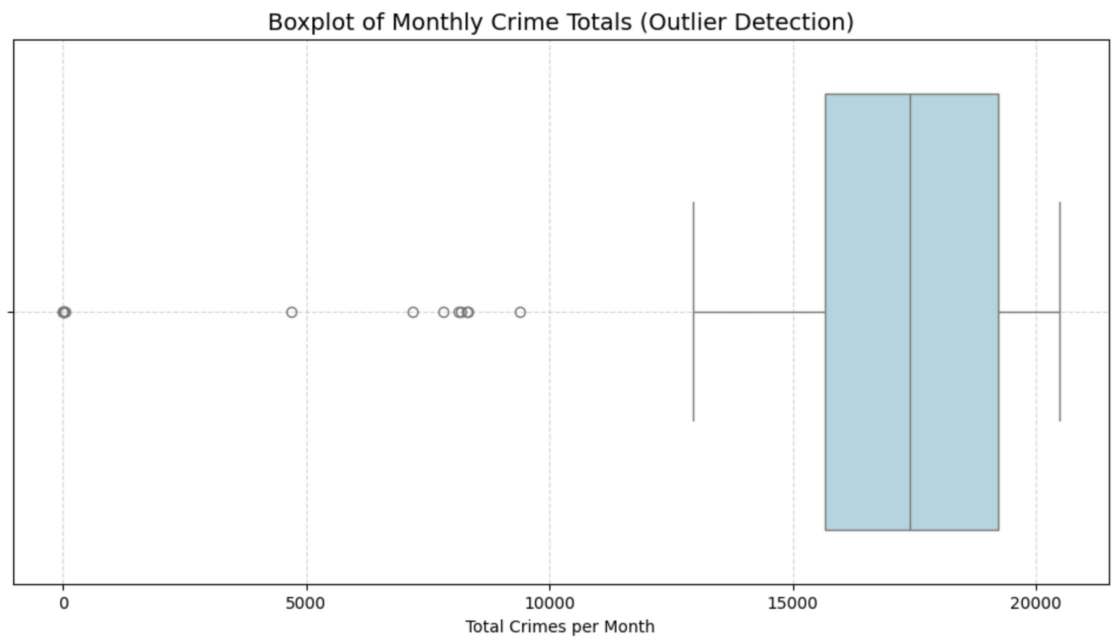
```
 Outlier Months (Unusual Crime Activity):
   Monthly_count  crime_count
52    2024-05-01         9386
53    2024-06-01         8126
54    2024-07-01         8170
55    2024-08-01         8324
```

```
56     2024-09-01          8309
57     2024-10-01          7817
58     2024-11-01          7179
59     2024-12-01          4697
60     2025-01-01            26
61     2025-02-01            44
62     2025-03-01            24
63     2025-04-01             1
64     2025-05-01             2
```

[78]:
```
z_scores = (monthly_crime_counts["crime_count"] -␣
 ↪monthly_crime_counts["crime_count"].mean()) /␣
 ↪monthly_crime_counts["crime_count"].std()
outliers_z = monthly_crimes[np.abs(z_scores) > 3]

outlier_months = pd.concat([outliers_iqr, outliers_z]).drop_duplicates().
 ↪sort_values("crime_count", ascending=False)
print("  Outlier Months (Unusual Crime Activity):")
print(outlier_months)


plt.figure(figsize=(12,6))
sns.boxplot(x = monthly_crime_counts["crime_count"], color="lightblue")
plt.title("Boxplot of Monthly Crime Totals (Outlier Detection)", fontsize=14)
plt.xlabel("Total Crimes per Month")
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

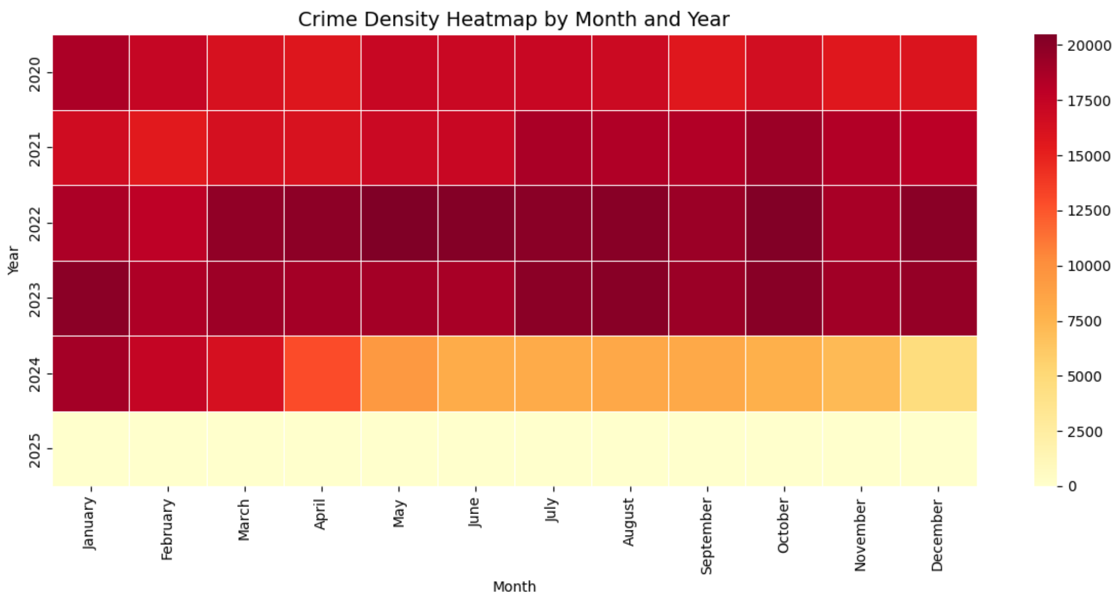
```
  Outlier Months (Unusual Crime Activity):
   Monthly_count  crime_count Month  Total_Crimes
52     2024-05-01       9386.0   NaN           NaN
55     2024-08-01       8324.0   NaN           NaN
56     2024-09-01       8309.0   NaN           NaN
54     2024-07-01       8170.0   NaN           NaN
53     2024-06-01       8126.0   NaN           NaN
57     2024-10-01       7817.0   NaN           NaN
58     2024-11-01       7179.0   NaN           NaN
59     2024-12-01       4697.0   NaN           NaN
61     2025-02-01         44.0   NaN           NaN
60     2025-01-01         26.0   NaN           NaN
62     2025-03-01         24.0   NaN           NaN
64     2025-05-01          2.0   NaN           NaN
63     2025-04-01          1.0   NaN           NaN
```

```
/tmp/ipython-input-341720165.py:2: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  outliers_z = monthly_crimes[np.abs(z_scores) > 3]
```



As predicted, all of the 2025 entries and some of the 2024 entries are outliers. This lines up with police reports of changes in reporting in 2024 and issues in 2025.

[79]:
```
heatmap_data = cleaned_df.groupby(["Year", "Month"]).size().
  ↪unstack(fill_value=0)
# Order months properly
month_order =␣
  ↪["January","February","March","April","May","June","July","August","September","October","N
heatmap_data = heatmap_data[month_order]

plt.figure(figsize=(12,6))
sns.heatmap(heatmap_data, cmap="YlOrRd", linewidths=0.5, annot=False)
plt.title("Crime Density Heatmap by Month and Year", fontsize=14)
plt.xlabel("Month")
plt.ylabel("Year")
plt.tight_layout()
plt.show()
```

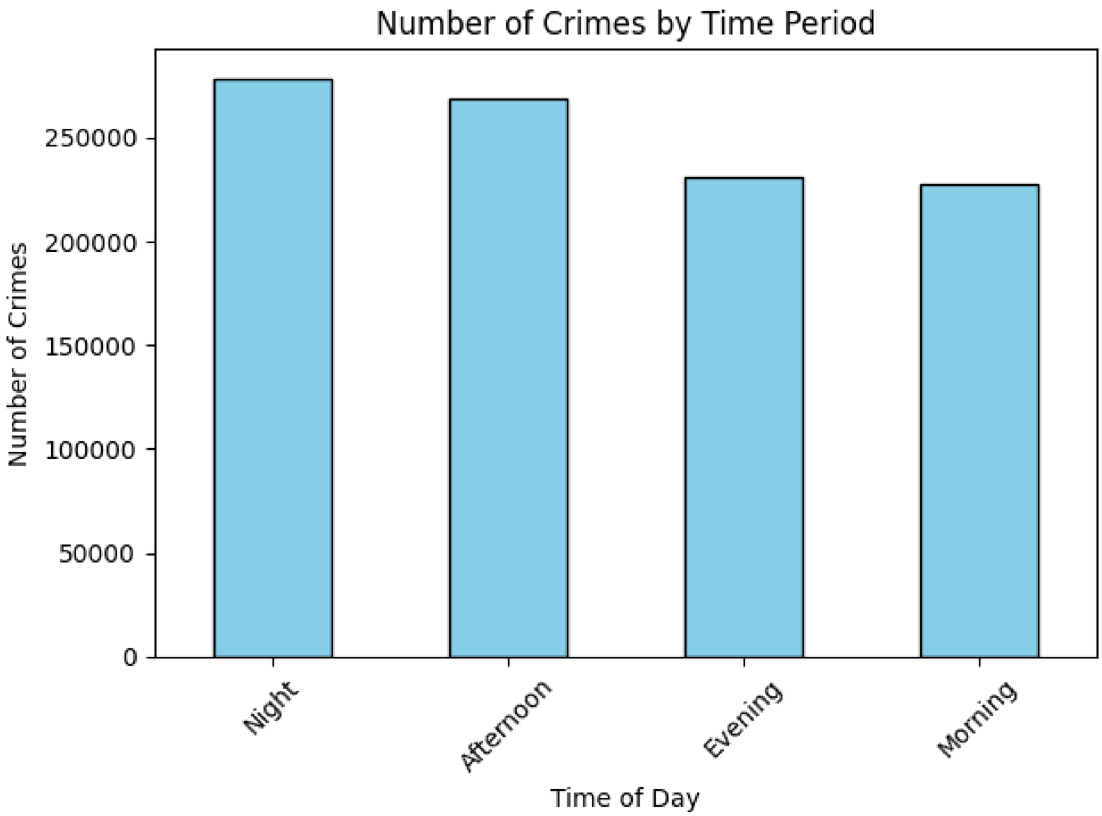


The heatmap above shows the decline of crime in 2024 and 2025 as data was affected by outliers and possible missing data.

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

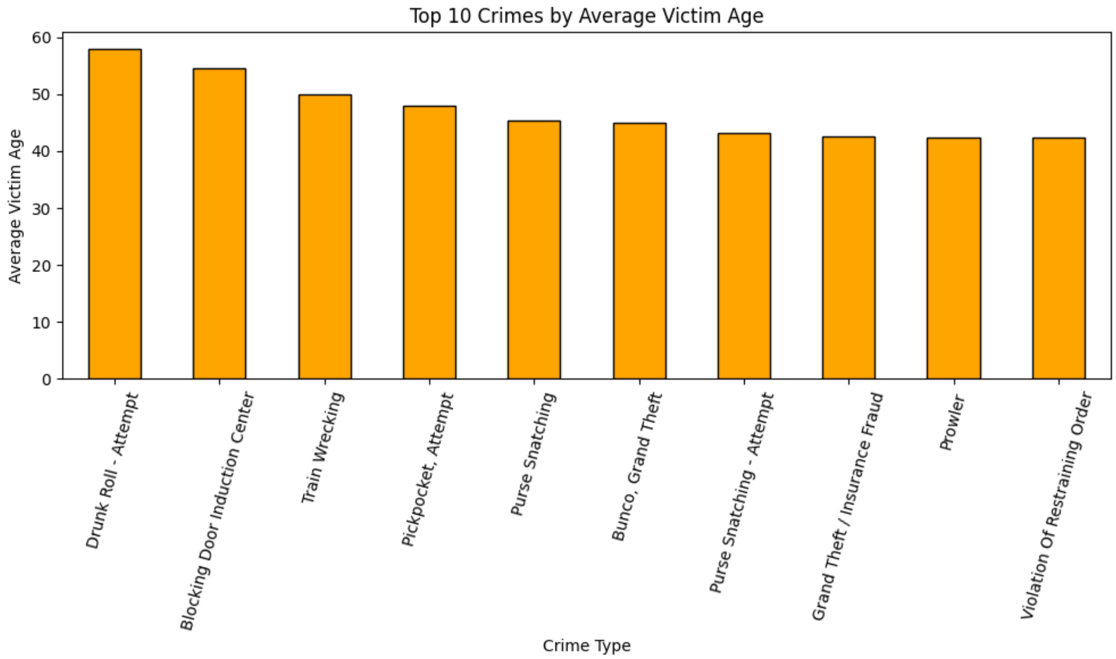
[46]:
```
def get_time_period(t):
    if pd.isnull(t):
        return "Unknown"
    dt_object = pd.to_datetime(str(t), format='%H:%M:%S')
    hour = dt_object.hour
    if 5 <= hour < 12:
        return "Morning"
    elif 12 <= hour < 17:
        return "Afternoon"
    elif 17 <= hour < 21:
        return "Evening"
    else:  # 0 <= hour < 5 or 21 <= hour < 24
        return "Night"

cleaned_df['Time Period'] = cleaned_df['TIME OCC'].apply(get_time_period)

cleaned_df['Time Period'].value_counts().plot(kind='bar', color='skyblue',␣
 ↪edgecolor='black')
plt.title("Number of Crimes by Time Period")
plt.xlabel("Time of Day")
plt.ylabel("Number of Crimes")
plt.xticks(rotation=45)
```

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



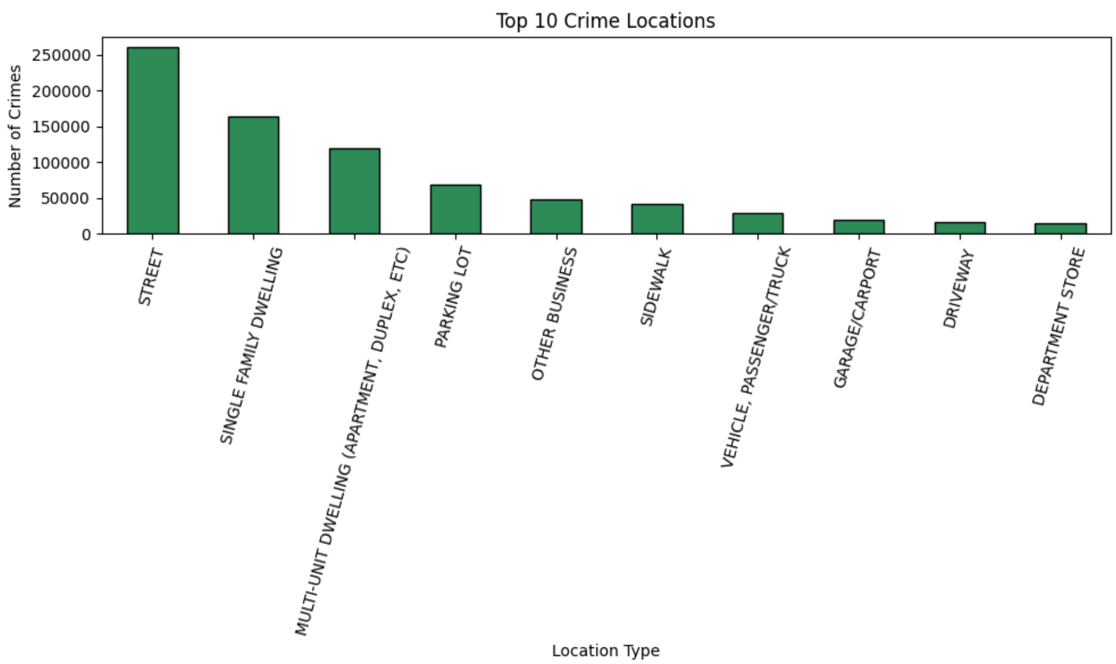
Most crimes occur at night, with the least occuring between 5 AM and 12 PM.

[47]:
```
plt.figure(figsize=(10,6))
cleaned_df.groupby('Crm Cd Desc')['Vict Age'].mean().
 ↪sort_values(ascending=False).head(10).plot(
    kind='bar', color='orange', edgecolor='black')
plt.title("Top 10 Crimes by Average Victim Age")
plt.xlabel("Crime Type")
plt.ylabel("Average Victim Age")
plt.xticks(rotation=75)
plt.tight_layout()
plt.show()
```



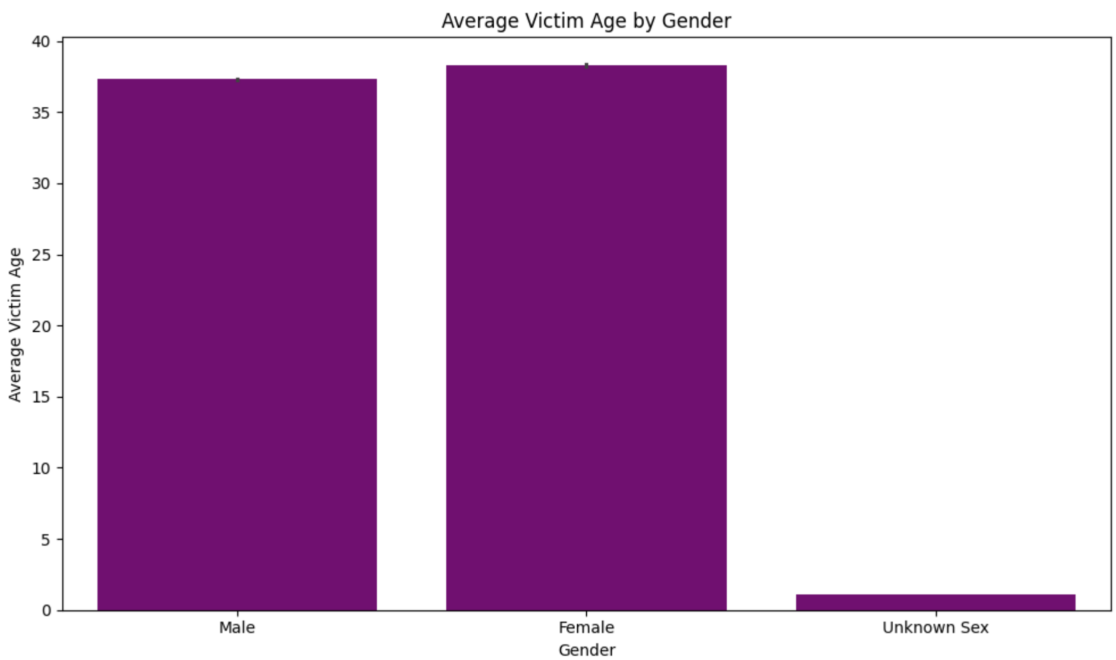
Stealing from inebriated people is the crime associated with the oldest average victim age. Purse snatching (both actual snatching and attempts) also have some of the highest victim ages.

[80]:
```
plt.figure(figsize=(10,6))
cleaned_df['Premis Desc'].value_counts().head(10).plot(kind='bar',␣
 ↪color='seagreen', edgecolor='black')
plt.title("Top 10 Crime Locations")
plt.xlabel("Location Type")
plt.ylabel("Number of Crimes")
plt.xticks(rotation=75)
plt.tight_layout()
plt.show()
```



The large majority of crimes occur in the street, with the next most occurring in family dwellings (either single family or multi-unit).

[81]:
```
plt.figure(figsize=(10,6))
sns.barplot(x='Vict Sex', y='Vict Age', data = cleaned_df, estimator='mean',␣
  ↪color = 'purple')
plt.title("Average Victim Age by Gender")
plt.xlabel("Gender")
plt.ylabel("Average Victim Age")
plt.tight_layout()
plt.show()
```

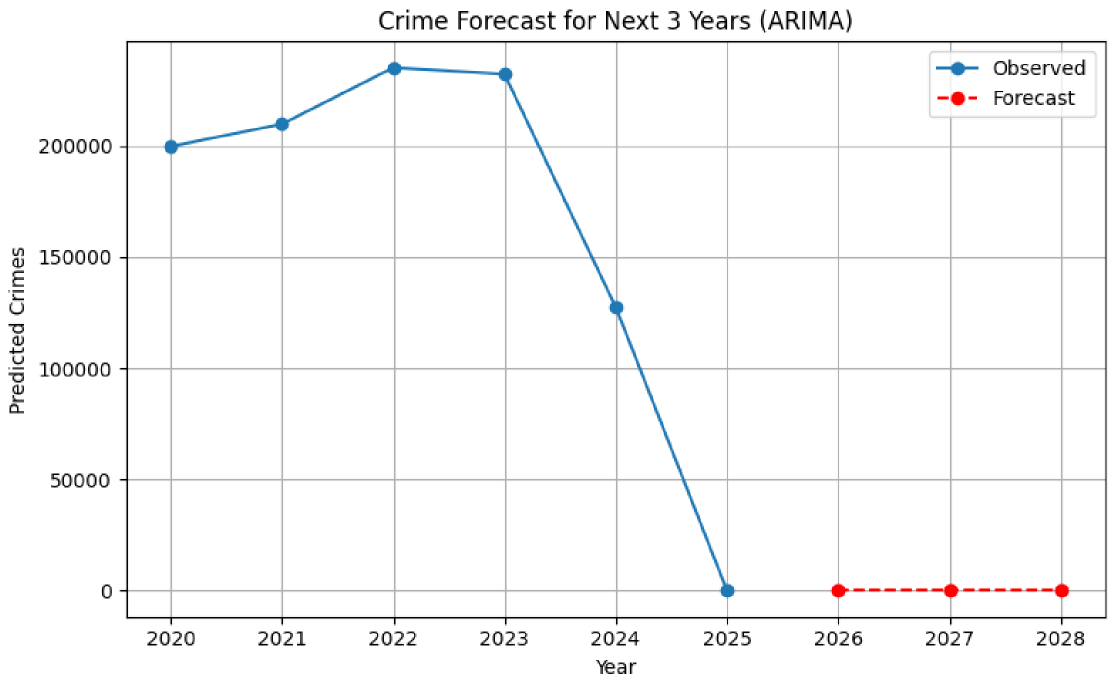


Females had a higher average victim age than males. A large majority of the category probably represent crimes where children or babies are the victim, where gender may be less important.

[82]:
```python
# 10. Predicting Future Trends (ARIMA Forecast)

from statsmodels.tsa.arima.model import ARIMA

yearly = cleaned_df['Year'].value_counts().sort_index()
model = ARIMA(yearly, order=(1,1,1))
model_fit = model.fit()
forecast = model_fit.forecast(steps=3)

plt.figure(figsize=(8,5))
plt.plot(yearly.index, yearly.values, label='Observed', marker='o')
plt.plot(range(yearly.index[-1]+1, yearly.index[-1]+4), forecast,␣
 ↪label='Forecast', marker='o', linestyle='--', color='red')
plt.title("Crime Forecast for Next 3 Years (ARIMA)")
plt.xlabel("Year")
plt.ylabel("Predicted Crimes")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided. As a result, forecasts cannot
```

```
be generated. To use the model for forecasting, use one of the supported classes
of index.
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided. As a result, forecasts cannot
be generated. To use the model for forecasting, use one of the supported classes
of index.
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided. As a result, forecasts cannot
be generated. To use the model for forecasting, use one of the supported classes
of index.
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.12/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
```

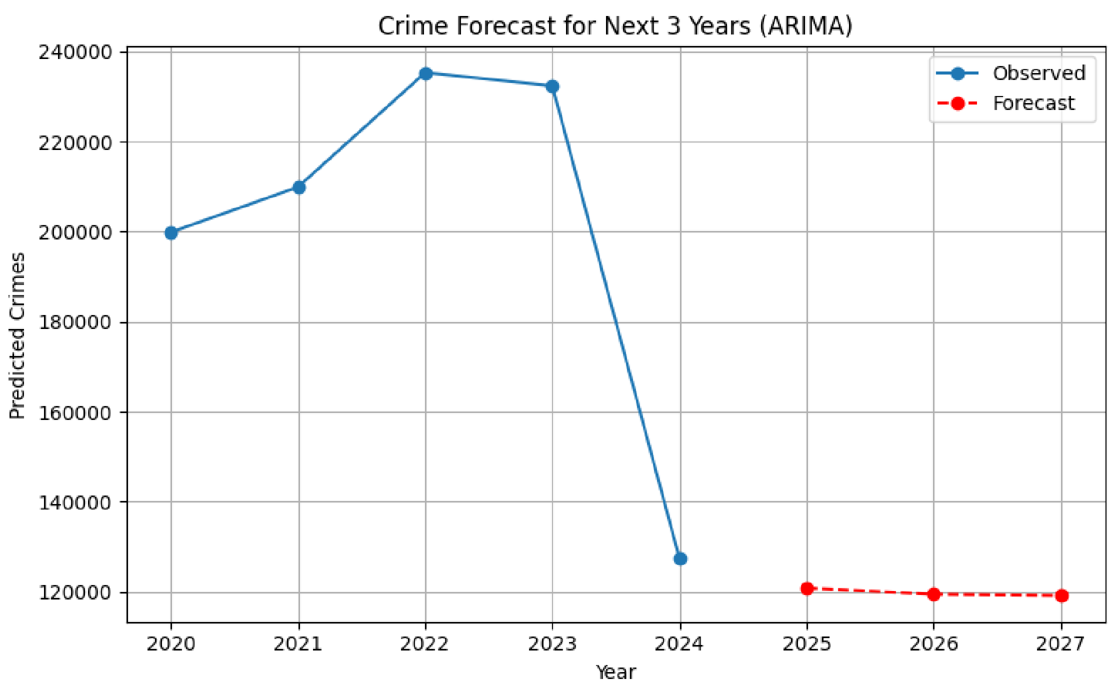


Since 2025 is not yet completed, and may be missing values, let us eliminate 2025 and run the predictions again.

[83]:
```
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.pyplot as plt

filtered_df = cleaned_df[cleaned_df['Year'] < 2025]
yearly = filtered_df['Year'].value_counts().sort_index()
model = ARIMA(yearly, order=(1,1,1))
model_fit = model.fit()
forecast = model_fit.forecast(steps=3)
plt.figure(figsize=(8,5))
plt.plot(yearly.index, yearly.values, label='Observed', marker='o')
plt.plot(range(yearly.index[-1]+1, yearly.index[-1]+4), forecast,␣
↪label='Forecast', marker='o', linestyle='--', color='red')
plt.title("Crime Forecast for Next 3 Years (ARIMA)")
plt.xlabel("Year")
plt.ylabel("Predicted Crimes")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided. As a result, forecasts cannot
be generated. To use the model for forecasting, use one of the supported classes
```

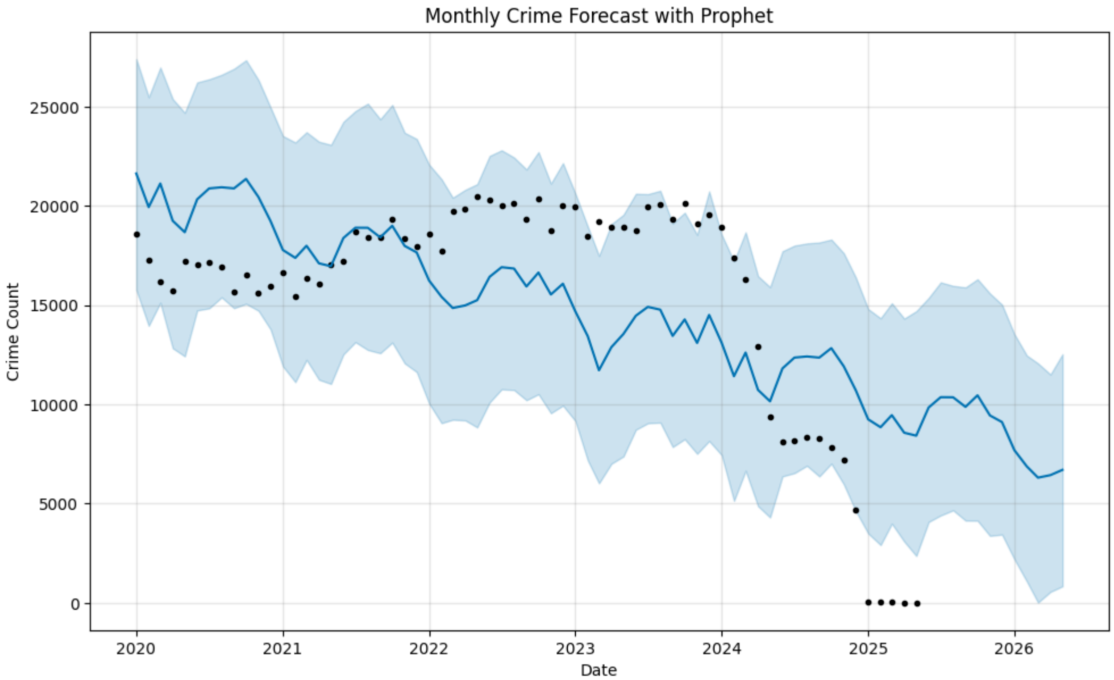
```
of index.
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided. As a result, forecasts cannot
be generated. To use the model for forecasting, use one of the supported classes
of index.
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided. As a result, forecasts cannot
be generated. To use the model for forecasting, use one of the supported classes
of index.
  self._init_dates(dates, freq)
/usr/local/lib/python3.12/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.12/dist-packages/statsmodels/tsa/base/tsa_model.py:837:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
```

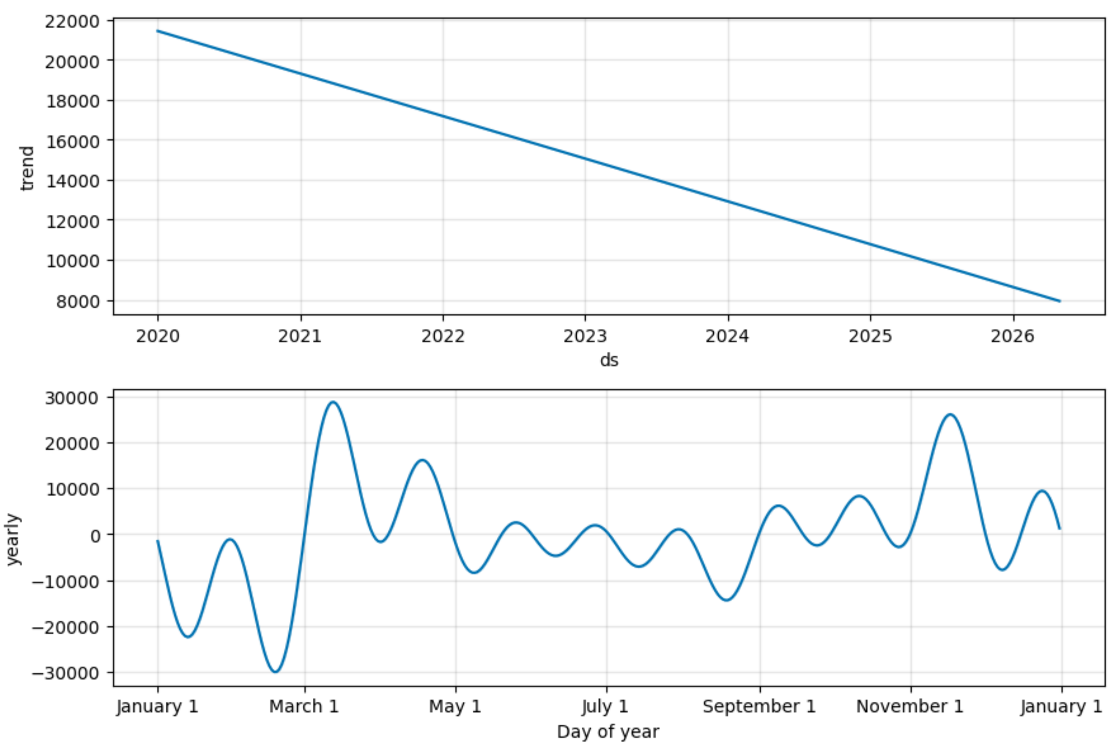


According to PRIMA, crime is supposed to continue its decrease over the next three years but plateau around 120,000 crimes per year.

[84]:
```
from prophet import Prophet
import matplotlib.pyplot as plt

df_prophet = monthly_crime_counts.rename(columns={'Monthly_count': 'ds',␣
 ↪'crime_count': 'y'})

model = Prophet(yearly_seasonality=True, daily_seasonality=False,␣
 ↪weekly_seasonality=False)
model.fit(df_prophet)
future = model.make_future_dataframe(periods=12, freq='MS')
forecast = model.predict(future)

fig1 = model.plot(forecast)
plt.title('Monthly Crime Forecast with Prophet')
plt.xlabel('Date')
plt.ylabel('Crime Count')
plt.show()

fig2 = model.plot_components(forecast)
plt.show()
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmp3qay19t9/wtp0kk1a.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmp3qay19t9/i31u87ap.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.12/dist-
packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=74137', 'data',
'file=/tmp/tmp3qay19t9/wtp0kk1a.json', 'init=/tmp/tmp3qay19t9/i31u87ap.json',
'output',
'file=/tmp/tmp3qay19t9/prophet_model2li8v8op/prophet_model-20251015154048.csv',
'method=optimize', 'algorithm=newton', 'iter=10000']
15:40:48 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
15:40:49 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
```





In the top graph, the black dots display the actual historical monthly crime data points. The blue line running through it shows the Prophet model's forecast of the predicted trend of the data, including the model's forecasts. The light blue shaded area shows the model's confidence interval for the line. Given the limited number of data points (especially the single data point for 2021), the forecast might have a high degree of uncertainty, which is reflected in the wide shaded area.

The second graph shows the trend of crimes over time (a negative correlation) and the last graph shows how crime counts can vary within a given year.

[84]:

```

```