

# Crime data analysis using python

## **ABSTRACT:**

Crime is a significant social issue that requires in-depth analysis to identify trends, high-risk locations, and patterns that can aid law enforcement and policymakers. This project focuses on Crime Data Analysis using Python, NumPy, Pandas, and Matplotlib to derive meaningful insights from crime datasets. This analysis provides insights into crime patterns, aiding policymakers, and researchers in decision-making, crime prevention strategies.

## PROBLEM STATEMENTS:

This analysis provides valuable insights into crime trends, hotspots, and arrest probabilities. Future enhancements could include real-time crime monitoring. These findings can aid law enforcement agencies, policymakers, and researchers in designing effective crime prevention strategies. Crime rates fluctuate over time, and understanding crime patterns is essential for law enforcement and public safety. This project aims to analyze historical crime data, identify trends, and determine high-crime locations using Python-based data analysis.

- **Crime Trends Over Time:** Analyzing yearly and monthly crime variations using time-series visualization.
- **Most Common Crimes:** Identifying top crime types using bar charts.
- **Arrest Analysis:** Comparing the percentage of crimes leading to arrests versus non-arrests.
- **Crime Hotspots:** Detecting high-crime locations through geospatial analysis.

## Python code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
df = pd.read_csv("/content/crime_data.csv")
print(df.head(50))
print(df.isnull().sum())
df.dropna(inplace=True)
df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year
df['Month'] = df['Date'].dt.month
print(df.info())
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
file_path = "crime_data.csv"
df = pd.read_csv(file_path)
df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year
```

```
df['Month'] = df['Date'].dt.month
df['DayOfWeek'] = df['Date'].dt.day_name()
crime_by_year = df['Year'].value_counts().sort_index()
plt.figure(figsize=(10,5))
plt.plot(crime_by_year.index, crime_by_year.values, marker='o', linestyle='-')
plt.xlabel("Year")
plt.ylabel("Crime Count")
plt.title("Crime Trend Over the Years")
plt.grid(True)
plt.show()

top_crimes = df['Primary Type'].value_counts().head(10)
plt.figure(figsize=(12,6))
sns.barplot(x=top_crimes.index, y=top_crimes.values, palette="viridis")
plt.xlabel("Crime Type")
plt.ylabel("Count")
plt.title("Top 10 Most Common Crimes")
plt.xticks(rotation=45)
plt.show()

crime_by_month = df['Month'].value_counts().sort_index()
plt.figure(figsize=(10,5))
plt.plot(crime_by_month.index, crime_by_month.values, marker='o', linestyle='-')
plt.xlabel("Month")
plt.ylabel("Crime Count")
plt.title("Crime Trend by Month")
```

```
plt.grid(True)

plt.show()

arrest_counts = df['Arrest'].value_counts()

plt.figure(figsize=(6,6))

plt.pie(arrest_counts, labels=["No Arrest", "Arrest"], autopct="%1.1f%%",
colors=["red", "green"])

plt.title("Arrest vs Non-Arrest Cases")

plt.show()

crime_heatmap = df.groupby(["Year", "Month"]).size().unstack()

plt.figure(figsize=(12,6))

sns.heatmap(crime_heatmap, cmap="coolwarm", linewidths=0.5, annot=True,
fmt=".0f")

plt.xlabel("Month")

plt.ylabel("Year")

plt.title("Crime Heatmap (Monthly vs. Yearly)")

plt.show()

top_locations = df["Location Description"].value_counts().head(10)

plt.figure(figsize=(12,6))

sns.barplot(x=top_locations.index, y=top_locations.values, palette="magma")

plt.xlabel("Location")

plt.ylabel("Crime Count")

plt.title("Top 10 Crime Locations")

plt.xticks(rotation=45)

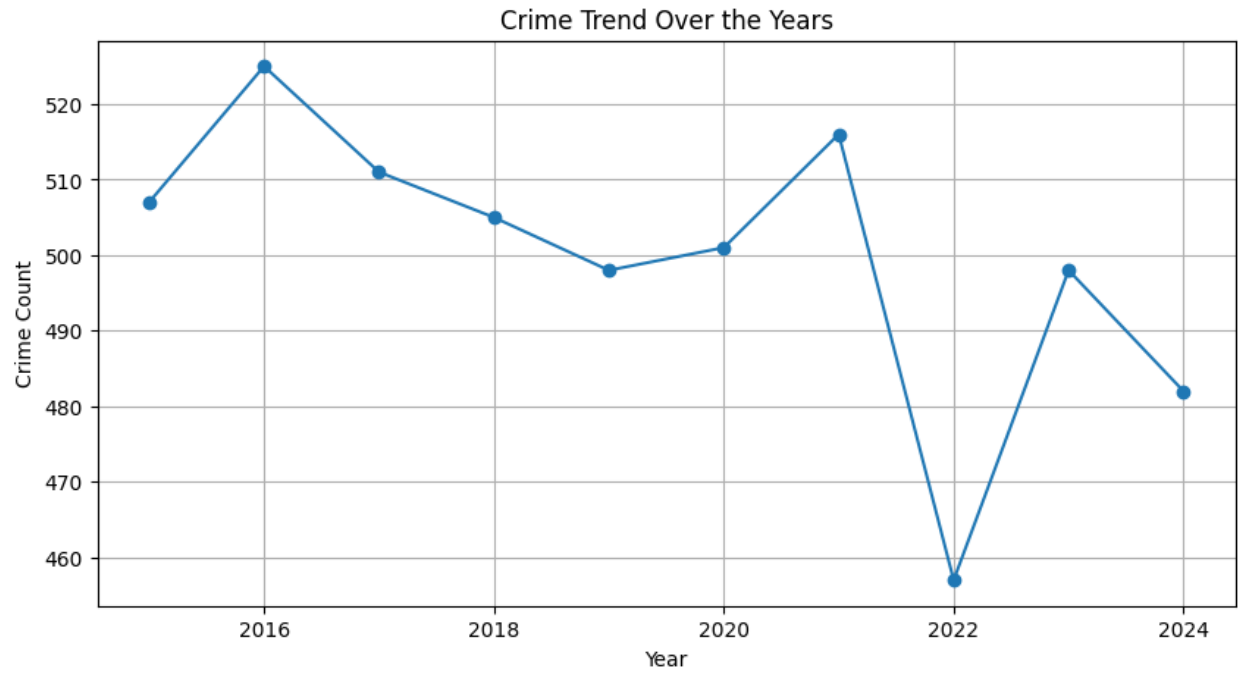
plt.show()

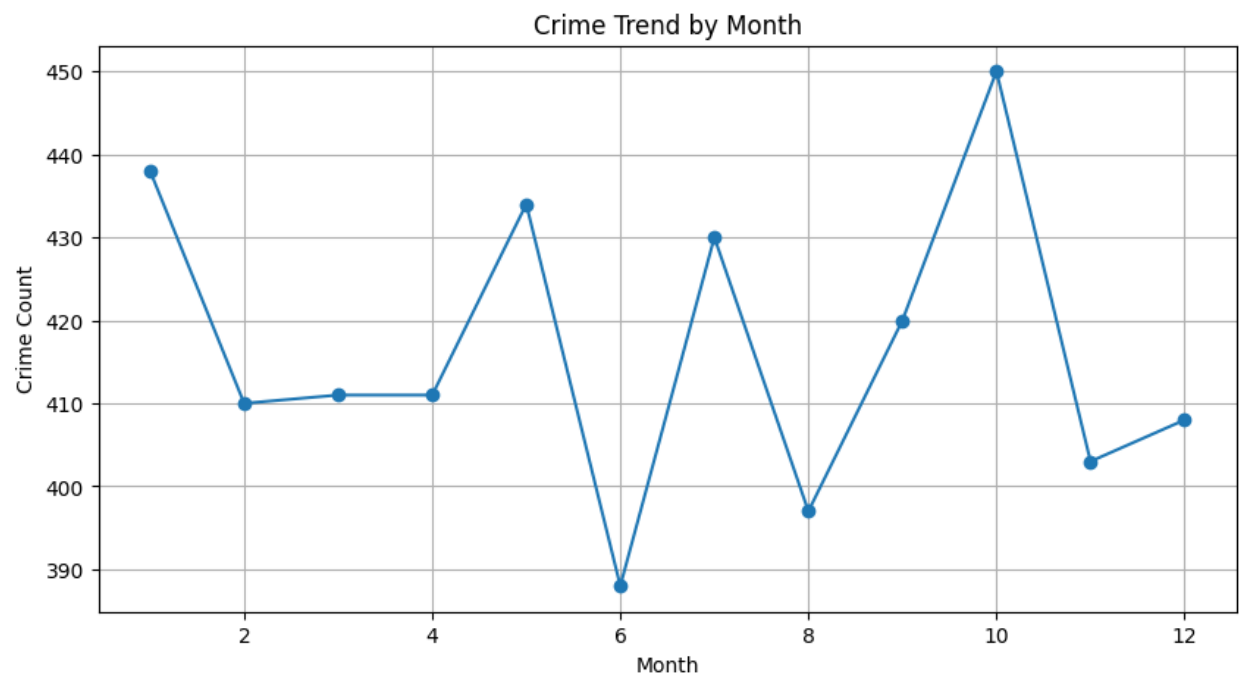
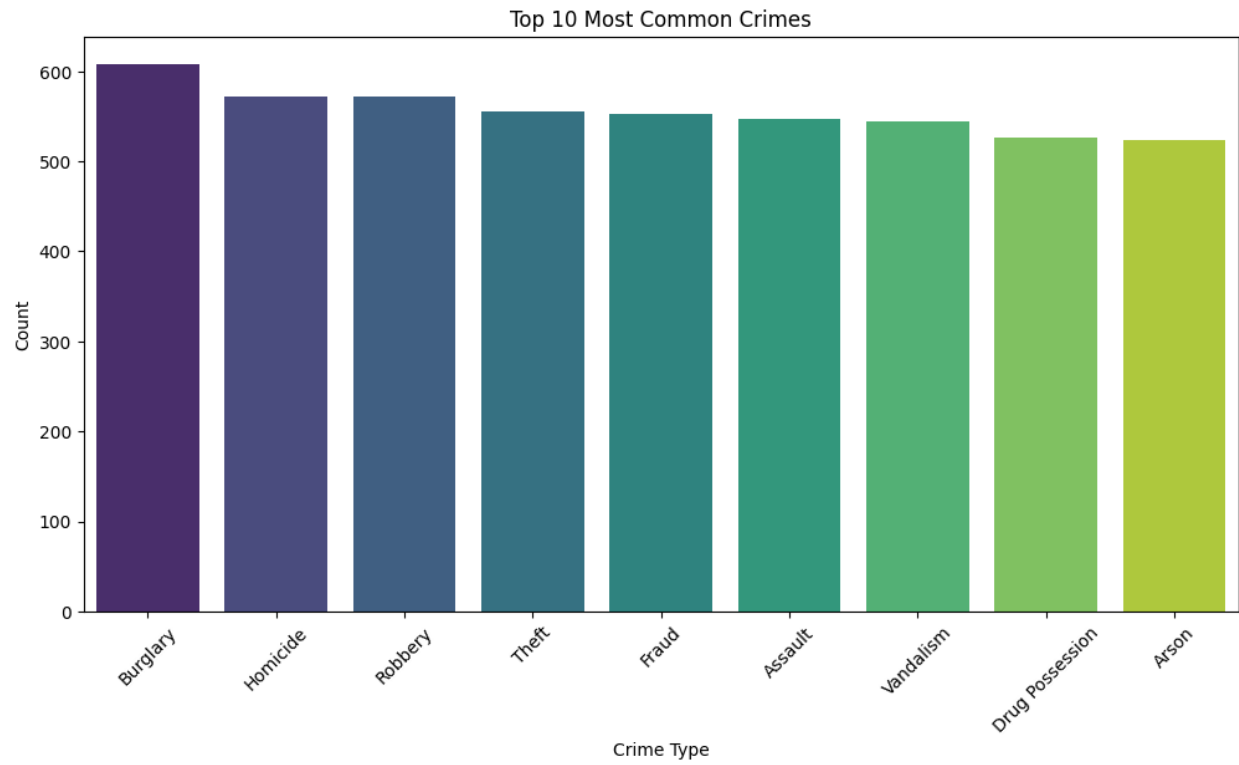
df['Crime Count'] = 1 # Create a count column
```

```
crime_trend = df.resample('M', on='Date')['Crime Count'].sum()
plt.figure(figsize=(12,6))
crime_trend.rolling(window=12).mean().plot(label='Rolling Mean (12 months)')
crime_trend.plot(alpha=0.5, linestyle='--', label='Monthly Crimes')
plt.legend()
plt.xlabel("Year")
plt.ylabel("Crime Count")
plt.title("Crime Trend Over Time (Rolling Average)")
plt.show()

df['Is Weekend'] = df['DayOfWeek'].isin(["Saturday", "Sunday"])
crime_weekend = df["Is Weekend"].value_counts()
plt.figure(figsize=(6,6))
plt.pie(crime_weekend, labels=["Weekday Crimes", "Weekend Crimes"],
autopct="%1.1f%%", colors=["blue", "orange"])
plt.title("Weekday vs. Weekend Crimes")
plt.show()
```

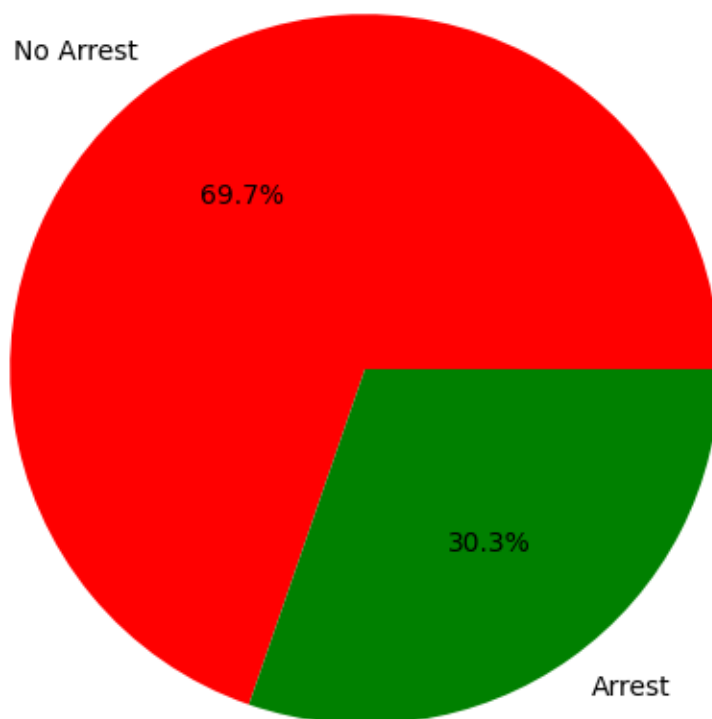
## Data visualization(output):

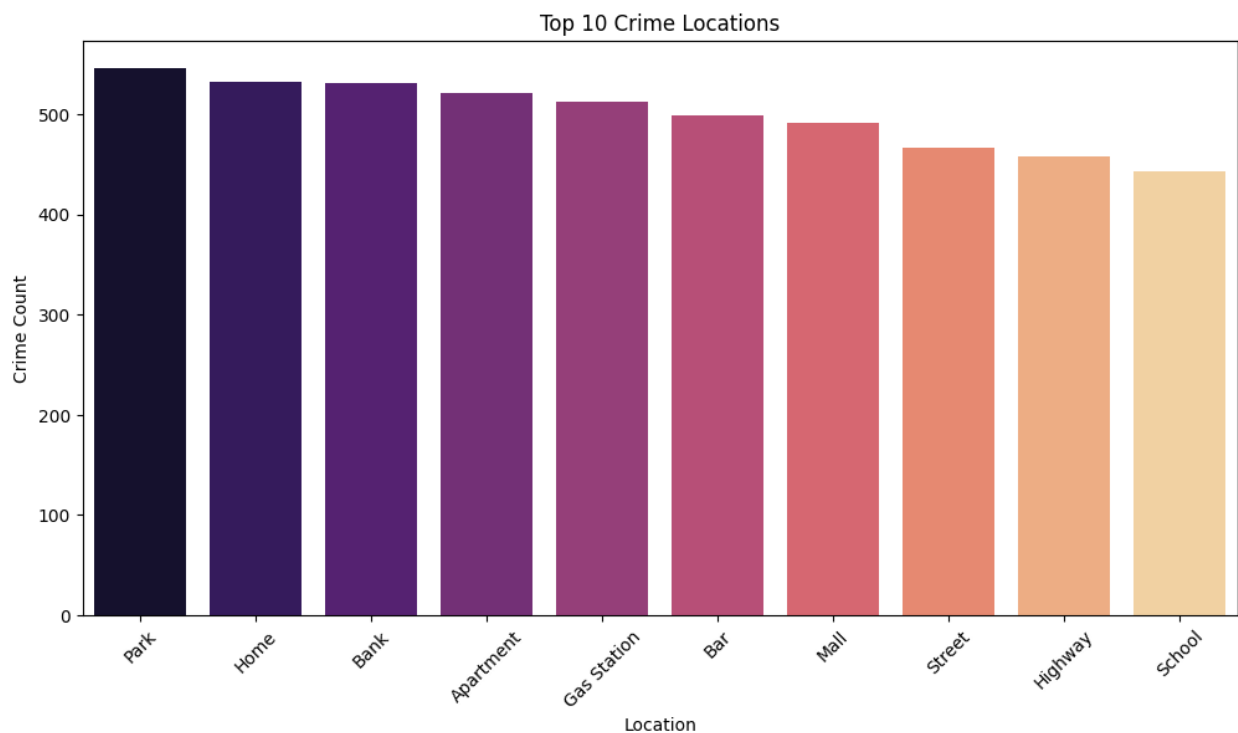


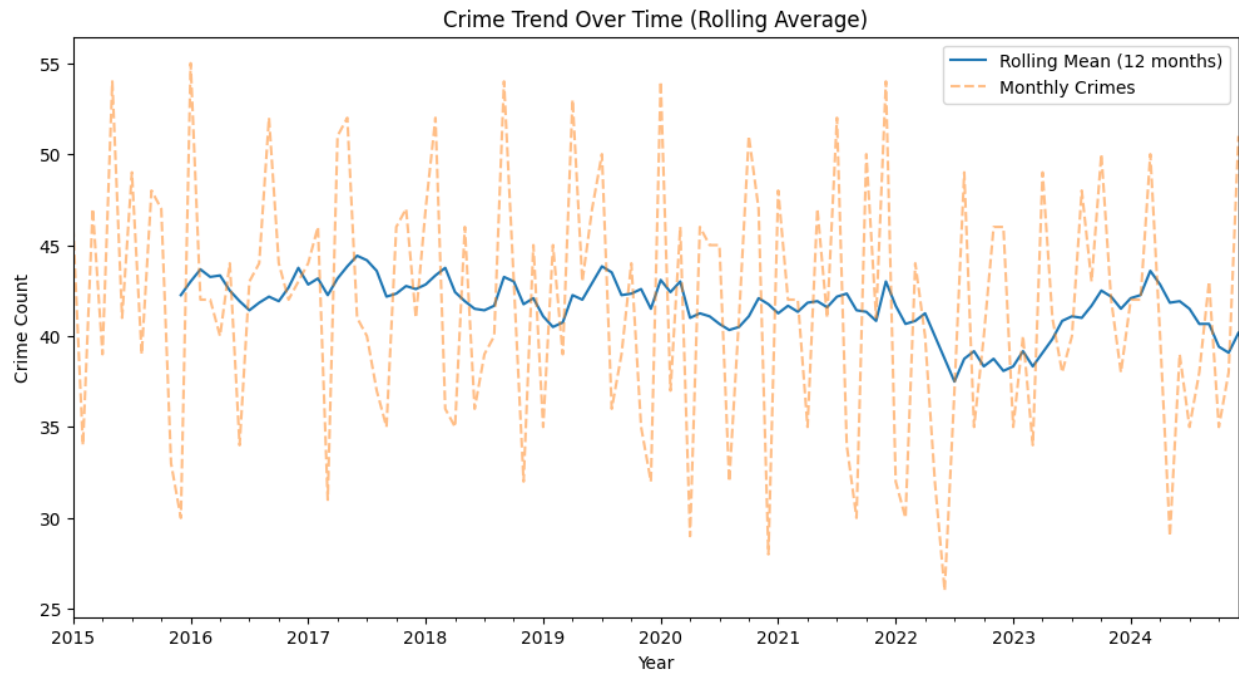




## Arrest vs Non-Arrest Cases

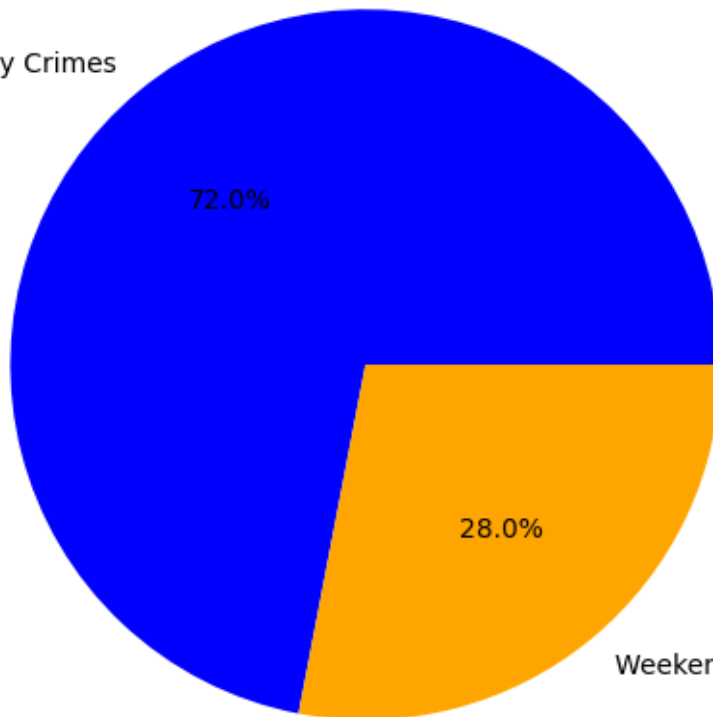






### Weekday vs. Weekend Crimes

Weekday Crimes



Weekend Crimes

## **Conclusion:**

Crime data analysis using Python provides actionable insights that help in crime prevention, resource optimization, and law enforcement decision-making. By leveraging data science techniques, we can build smarter, safer communities.