Crime Data Analysis Project

Submitted in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

by

Pranav Vinayak Chopdekar,

Nikhil Jaswaraj Karkera,

Mahendra Varma Vaddi,

UNIVERSITY OF COLORADO BOULDER

2024-2025

Abstract

For law enforcement organizations, legislators, and researchers to comprehend crime trends, spot patterns, and enhance public safety, crime data analysis is an essential tool. The analysis and interpretation of crime data using statistical and machine learning methods is the main emphasis of this work. We seek to find spatial and temporal patterns, pinpoint high-risk locations (hotspots), and investigate variables that can affect crime rates, such as socioeconomic status, population density, and the presence of law enforcement, by examining historical crime databases. The study offers practical insights for crime prevention tactics using data visualization, predictive modeling, and geospatial analysis. Additionally, this study assesses how different policies and initiatives affect the decrease of crime, providing evidence-based suggestions for better resource allocation and law enforcement. The results could improve community safety and bolster evidence-based policy choices.

Keywords: crime trends, crime data, high-risk locations, crime rates, evidence.

Data Collection and Preparation

- **Data Sources: -**

  For this following project, we have collected data from

  https://data.cincinnati-oh.gov/resource/k59e-2pvf.json
  The time covered in the dataset consist of the year 1992 to the year 2023.

- **Initial Data Overview: -**
  This data consists of total 28 columns, each column depicting some or the other important information about the incident. The total number or rows in the dataset are 974477. Out of all the 28 columns, the most important columns which tell us more about the data is "Offense, Location, neighbourhood, weapon used and ucrgroup".

  Offense: - This column defines the type of offense which was committed. This includes offenses such as burglary, assault, theft and many more.

  Location: - This data attribute gives the name of the place where the theft took place. One example of this is "Parking lot" Suggesting that the theft took place in a parking lot.

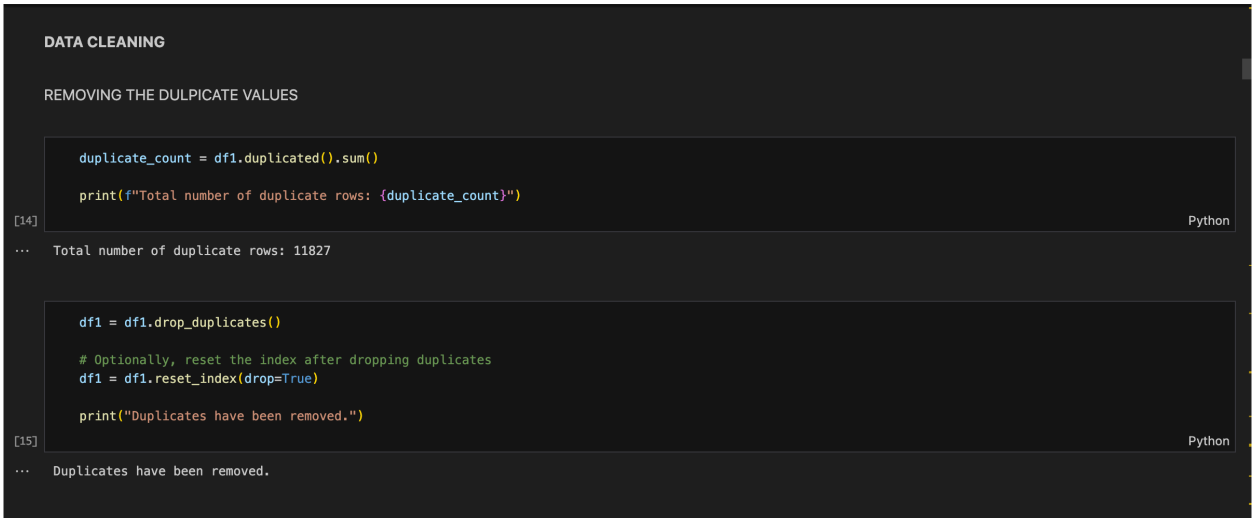
  Neighbourhood: - This attribute defines the name of the area in which the theft took place. For Example, one theft took place in the "College Hill" area, thus stating that the theft did happen in the college hill area.

  Weapon: - This attribute tells us about the type of weapon which was used to commit the crime. These also includes some description of the weapon such as whether it is a sharp weapon or dull weapon.

  Ucrgroup: - This Attribute is used to define the group to which the crime belonged to. This helps us to categories the crime based on similar crimes hence making it easier to fetch all the crimes which are like each other.

- **Data Cleaning and Transformation: -**

It is necessary to clean and transform the data as it will make it easier for further processing and get fine-tuned solutions. For our data, the first thing we did was to calculate the number of duplicate values present in whole dataset. We removed the duplicate values which were present in the dataset.



DATA CLEANING

REMOVING THE DULPICATE VALUES

```
duplicate_count = df1.duplicated().sum()

print(f"Total number of duplicate rows: {duplicate_count}")
```

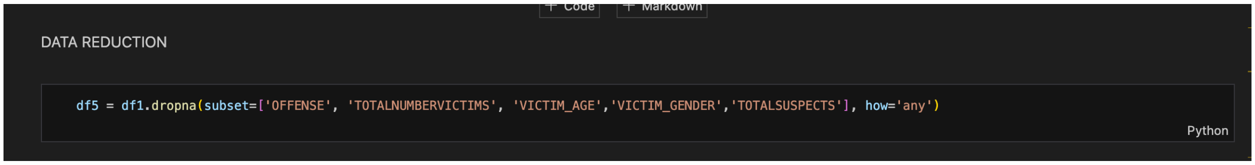
```
Total number of duplicate rows: 11827
```

```
df1 = df1.drop_duplicates()

# Optionally, reset the index after dropping duplicates
df1 = df1.reset_index(drop=True)

print("Duplicates have been removed.")
```

```
Duplicates have been removed.
```

After removing the duplicate values, we calculated the number of duplicate rows just to make sure that there are no duplicate values as well as rows in our data.

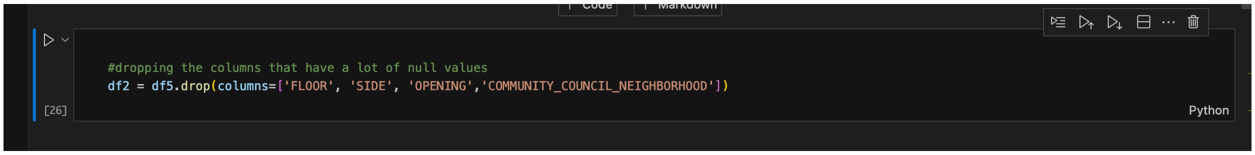
Once we have dealt with the duplicate values, we moved our focus toward dealing with NaN values. Firstly, we checked for missing values which might be present in the dataset. Later, we replaced all the NaN values with “UNKNOWN” string. It ensures that all missing values are replaced, so there are no NaNs in this specific column.



DATA REDUCTION

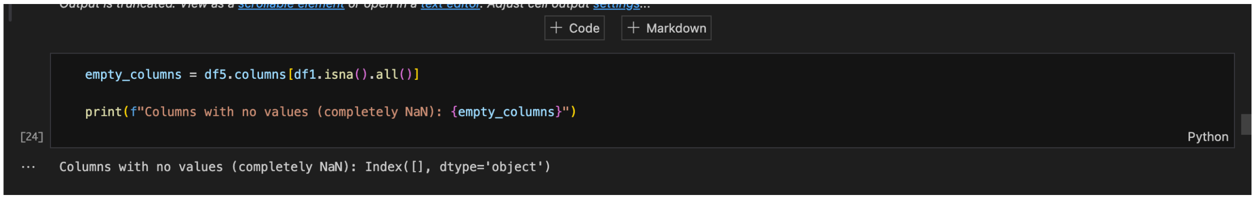
```
df5 = df1.dropna(subset=['OFFENSE', 'TOTALNUMBERVICTIMS', 'VICTIM_AGE','VICTIM_GENDER','TOTALSUSPECTS'], how='any')
```

We then checked for the columns which would have all the values as NaN but found none. Checking this, to maintain the sustainability of the data, we dropped all the columns that had the highest number of NaN values such as “Floor, side,Instanceid, rpt_area and opening”. We did check for missing values in the data after dropping columns too.



```
#dropping the columns that have a lot of null values
df2 = df5.drop(columns=['FLOOR', 'SIDE', 'OPENING','COMMUNITY_COUNCIL_NEIGHBORHOOD'])
```

[26]



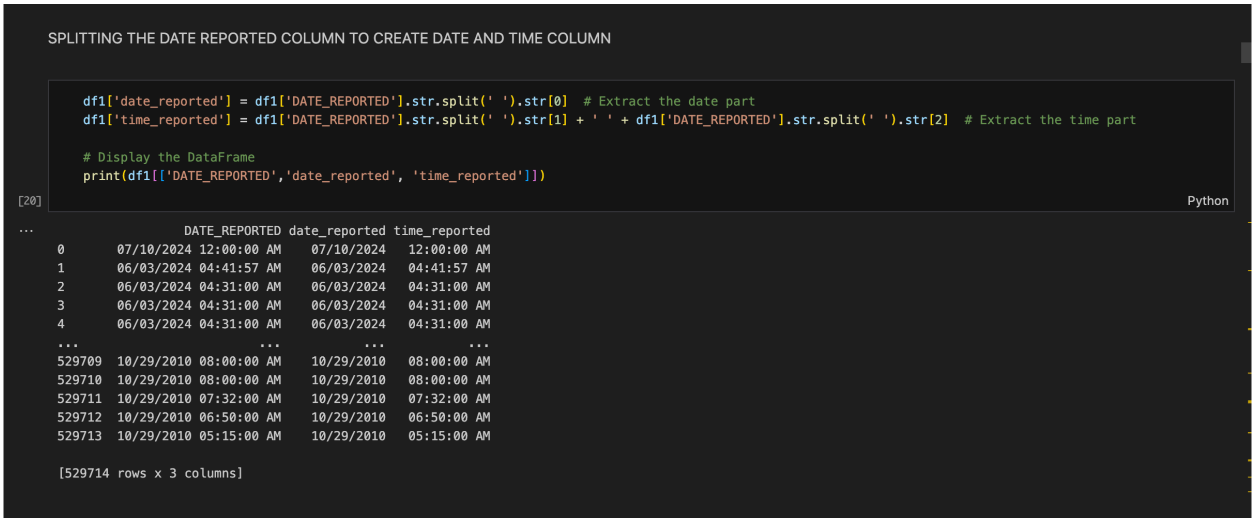
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
empty_columns = df5.columns[df1.isna().all()]

print(f"Columns with no values (completely NaN): {empty_columns}")
```

[24]

```
Columns with no values (completely NaN): Index([], dtype='object')
```

We did split the whole ‘Date reported column into two columns named “date_reported” and “time_reported”.



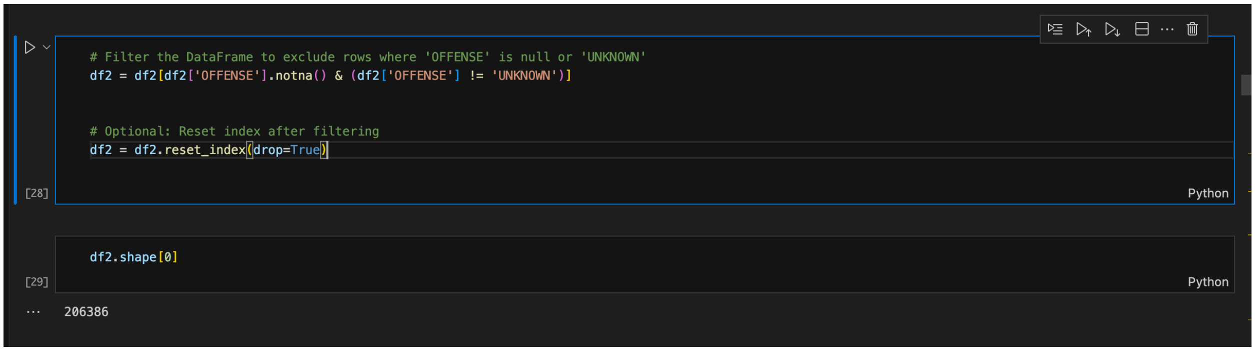
SPLITTING THE DATE REPORTED COLUMN TO CREATE DATE AND TIME COLUMN

```
df1['date_reported'] = df1['DATE_REPORTED'].str.split(' ').str[0]  # Extract the date part
df1['time_reported'] = df1['DATE_REPORTED'].str.split(' ').str[1] + ' ' + df1['DATE_REPORTED'].str.split(' ').str[2]  # Extract the time part

# Display the DataFrame
print(df1[['DATE_REPORTED','date_reported', 'time_reported']])
```

[20]

```
                 DATE_REPORTED date_reported time_reported
0       07/10/2024 12:00:00 AM    07/10/2024   12:00:00 AM
1       06/03/2024 04:41:57 AM    06/03/2024   04:41:57 AM
2       06/03/2024 04:31:00 AM    06/03/2024   04:31:00 AM
3       06/03/2024 04:31:00 AM    06/03/2024   04:31:00 AM
4       06/03/2024 04:31:00 AM    06/03/2024   04:31:00 AM
...                        ...           ...           ...
529709  10/29/2010 08:00:00 AM    10/29/2010   08:00:00 AM
529710  10/29/2010 08:00:00 AM    10/29/2010   08:00:00 AM
529711  10/29/2010 07:32:00 AM    10/29/2010   07:32:00 AM
529712  10/29/2010 06:50:00 AM    10/29/2010   06:50:00 AM
529713  10/29/2010 05:15:00 AM    10/29/2010   05:15:00 AM

[529714 rows x 3 columns]
```

removing all rows from the DataFrame df2 where the value of the OFFENSE column is either 'UNKNOWN' or null. It resets the DataFrame's index after filtering in order to preserve a clear, sequential index.



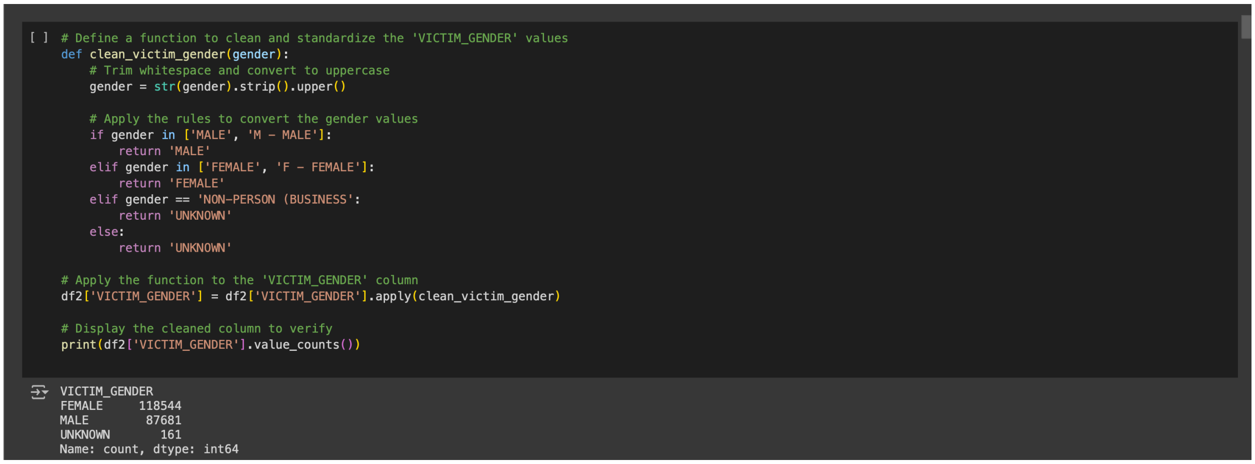
```
# Filter the DataFrame to exclude rows where 'OFFENSE' is null or 'UNKNOWN'
df2 = df2[df2['OFFENSE'].notna() & (df2['OFFENSE'] != 'UNKNOWN')]


# Optional: Reset index after filtering
df2 = df2.reset_index(drop=True)
```

```
df2.shape[0]
```

```
206386
```

Data Standardization

Removing the redundant values in the columns VICTIM_GENDER, example there are some values in the columns with the value “FEMALE” and some values are
“F – FEMALE” which are both same but are having different string values thus we converted such values to a same string value.



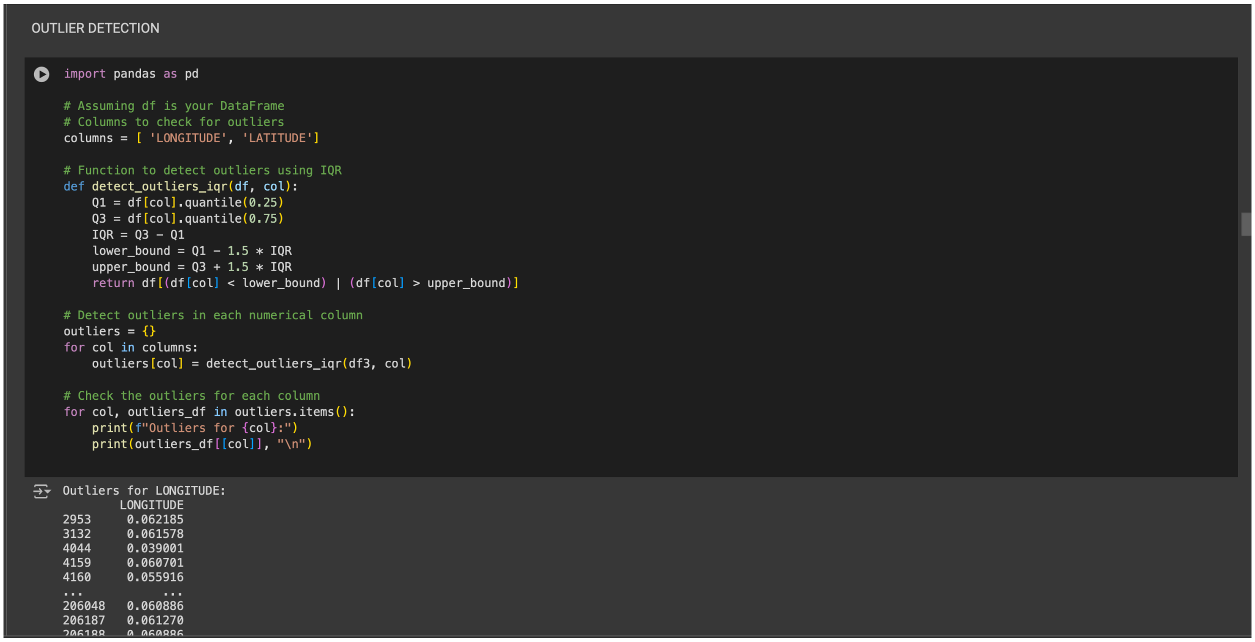
```
# Define a function to clean and standardize the 'VICTIM_GENDER' values
def clean_victim_gender(gender):
    # Trim whitespace and convert to uppercase
    gender = str(gender).strip().upper()

    # Apply the rules to convert the gender values
    if gender in ['MALE', 'M - MALE']:
        return 'MALE'
    elif gender in ['FEMALE', 'F - FEMALE']:
        return 'FEMALE'
    elif gender == 'NON-PERSON (BUSINESS':
        return 'UNKNOWN'
    else:
        return 'UNKNOWN'

# Apply the function to the 'VICTIM_GENDER' column
df2['VICTIM_GENDER'] = df2['VICTIM_GENDER'].apply(clean_victim_gender)

# Display the cleaned column to verify
print(df2['VICTIM_GENDER'].value_counts())
```

```
VICTIM_GENDER
FEMALE     118544
MALE        87681
UNKNOWN       161
Name: count, dtype: int64
```

On further investigation we found the same situation with the suspect gender as well thus we followed the same steps to correct the data for the VICTIM_GENDER as well.

Following that we dropped the column INSTANCE_ID and RPT_AREA as the instance id didn’t give much information and we also had a column INCIDENT_NO for the primary key. And the RPT_AREA column wasn’t giving any significant knowledge insight and had a lot of null values.

Moving further we removed figured out the outliers for the latitude and longitude, this information gives the co ordinates of places where crime is less likely to occur and are considered safe in general.
This information would cause a bias to the model that may usually predict the place is safe and it's less likely that the crime would occur given a certain circumstances, But it is important to know that there are significant risks in these places also.

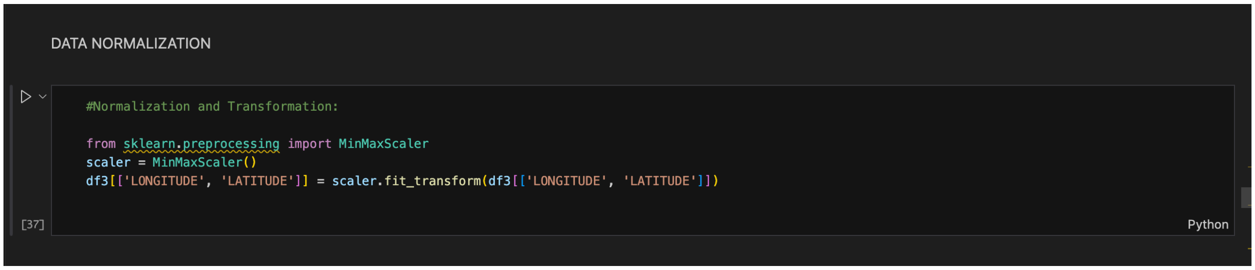


OUTLIER DETECTION

```
import pandas as pd

# Assuming df is your DataFrame
# Columns to check for outliers
columns = [ 'LONGITUDE', 'LATITUDE']

# Function to detect outliers using IQR
def detect_outliers_iqr(df, col):
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[col] < lower_bound) | (df[col] > upper_bound)]

# Detect outliers in each numerical column
outliers = {}
for col in columns:
    outliers[col] = detect_outliers_iqr(df3, col)

# Check the outliers for each column
for col, outliers_df in outliers.items():
    print(f"Outliers for {col}:")
    print(outliers_df[[col]], "\n")
```

```
Outliers for LONGITUDE:
        LONGITUDE
2953     0.062185
3132     0.061578
4044     0.039001
4159     0.060701
4160     0.055916
...           ...
206048   0.060886
206187   0.061270
206188   0.060886
```

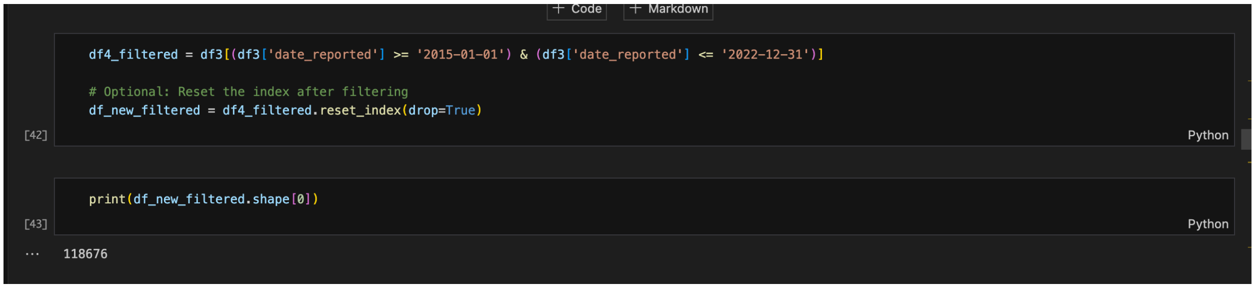
Once the data was cleaned, we then normalized the data by converting the latitude and longitude columns thus, converting their values ranging between 0 and 1. After this, we got the summary statistics for numerical columns as well as categorical columns.



DATA NORMALIZATION

```
#Normalization and Transformation:

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df3[['LONGITUDE', 'LATITUDE']] = scaler.fit_transform(df3[['LONGITUDE', 'LATITUDE']])
```

We later Reduced the data by considering only the crime reported in the recent decade, i.e, crime reported between the years 1st of Jan 2012 and 31st of December 2022 as this would ensure there is less noise in the data and the model could be built efficiently. And also as the current year's crime incident have not been completely updated in the database.

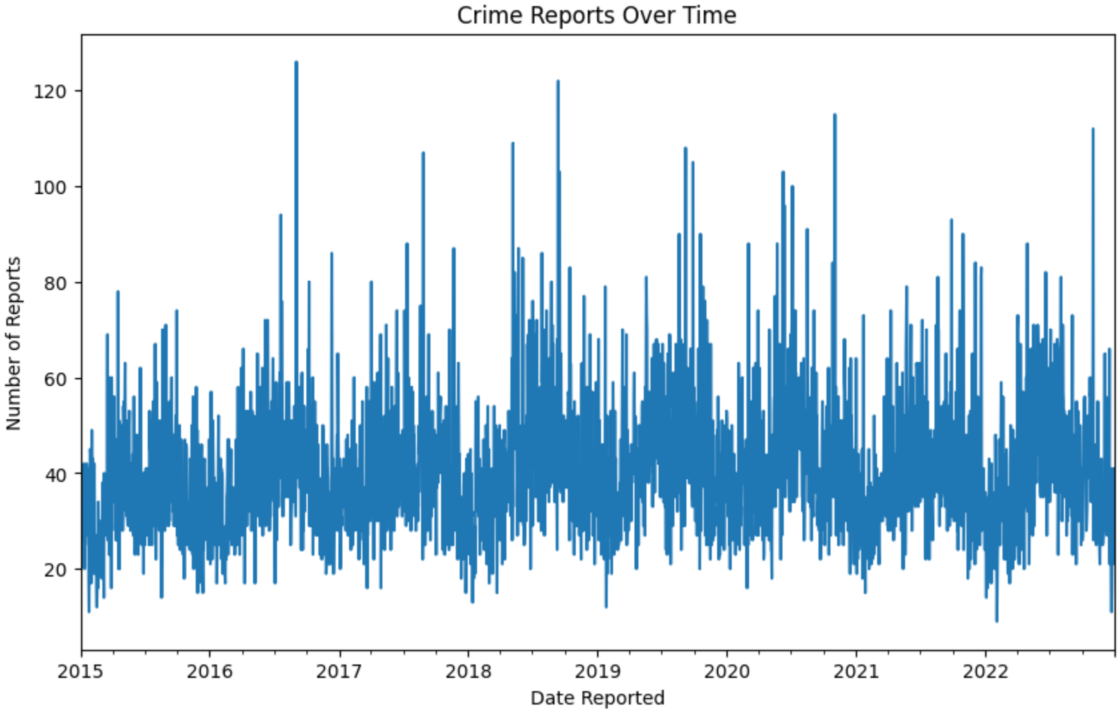


```
df4_filtered = df3[(df3['date_reported'] >= '2015-01-01') & (df3['date_reported'] <= '2022-12-31')]

# Optional: Reset the index after filtering
df_new_filtered = df4_filtered.reset_index(drop=True)
```

```
print(df_new_filtered.shape[0])
```

```
118676
```

- **VISUALIZATION**
  Plotting the crimes reported over the years



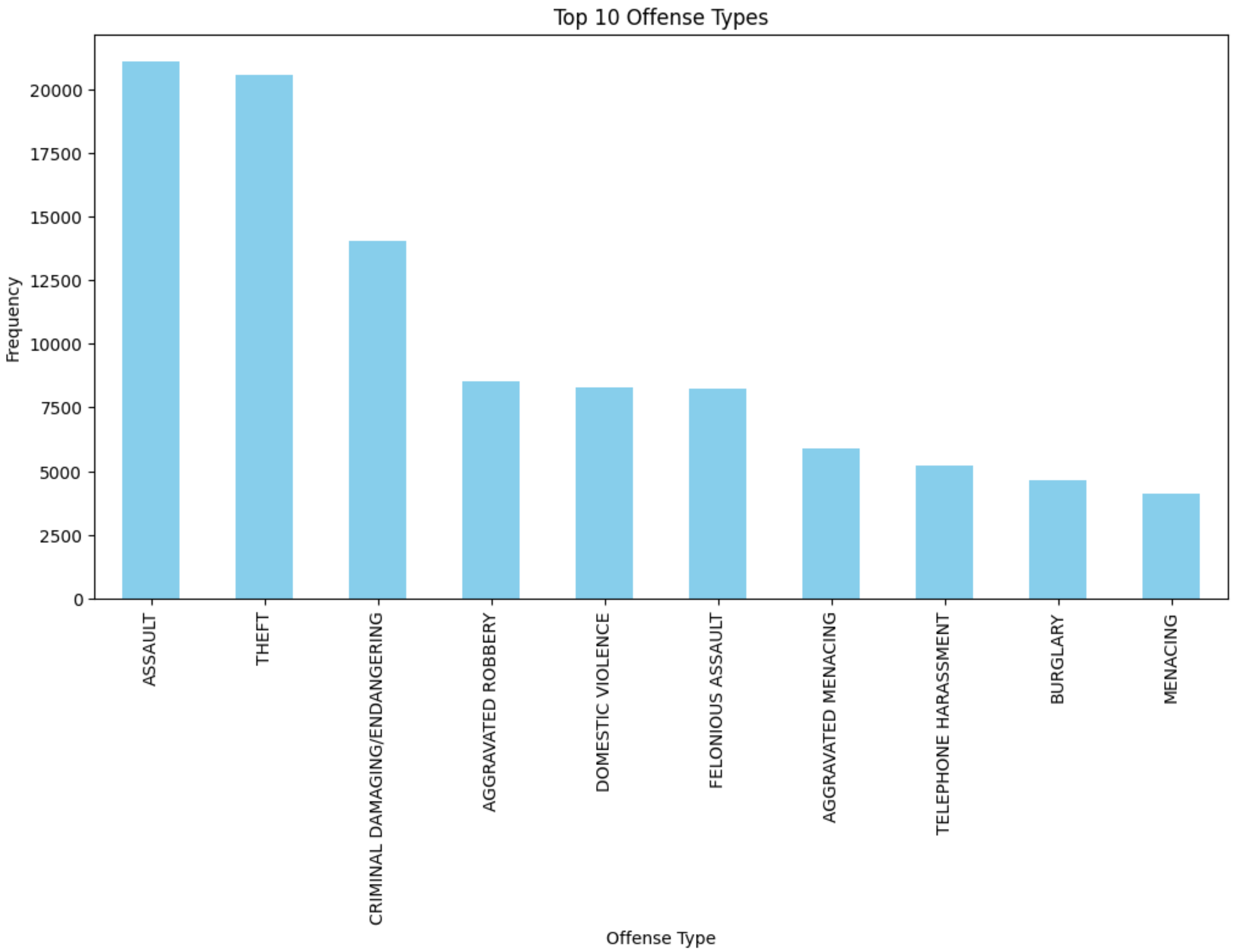
The graph indicates various variations in crime reporting over time but does not clearly illustrate an increase or declining trend.

There are times between 2017 and 2021 when the spikes are quite large; these are days when the number of crime complaints has grown dramatically.

The years before to 2017 and once more in the latter part of the period (about 2021–2022) appear to have had more constant reporting (without sharp increases).

This graph most likely depicts the daily fluctuations in crime reports over a number of years, illuminating reporting patterns that may be impacted by external variables such as the economy, the seasons, or other causes.

Plotting a bar graph for the type of crimes reported.



The following are the graph's main points:

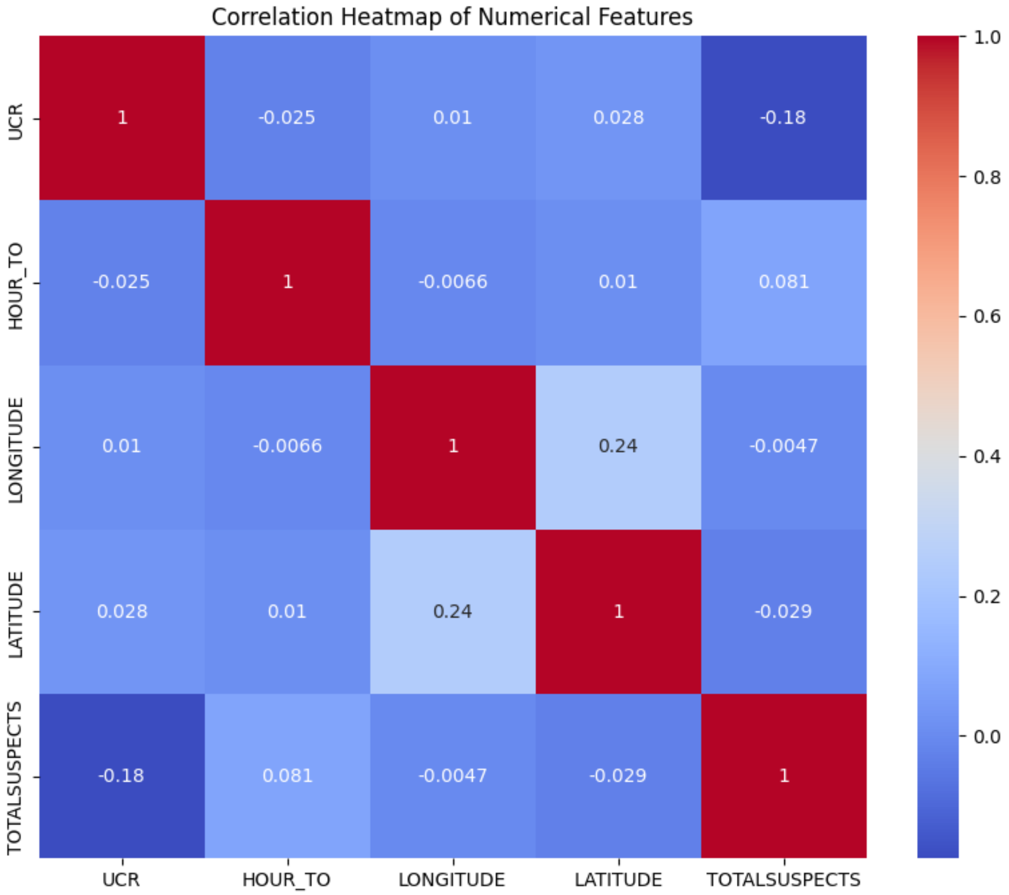
The most common offenses are theft and assault, each with a frequency of more than 20,000 incidents.

The most common offense is criminal damaging/endangering, which is followed by aggravated robbery and domestic violence.

With frequencies ranging from around 4,000 to 7,500, the offenses of Felonious Assault, Aggravated Menacing, Telephone Harassment, Burglary, and Menacing are less common but nonetheless noteworthy.

This suggests that some crime categories, such as theft and assault, are considerably more prevalent in the dataset than others, such as menacing or burglary.

Heat map



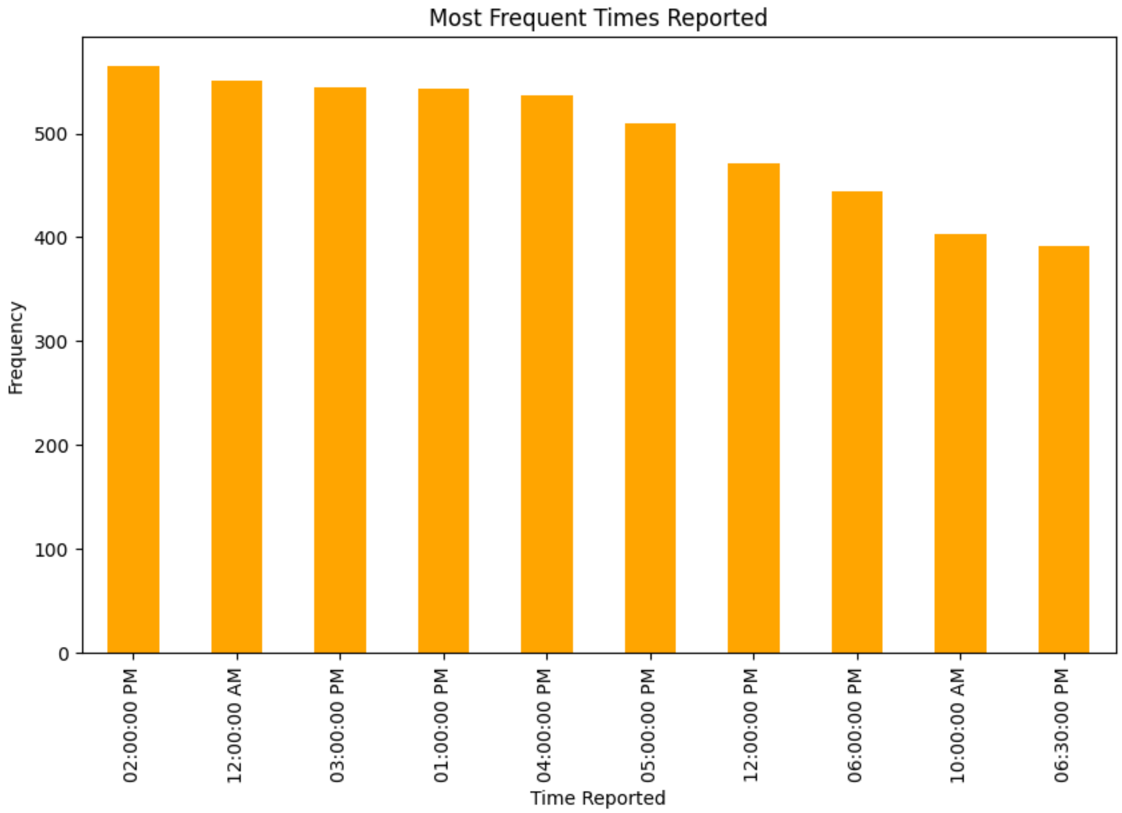
Important findings: UCR and TOTALSUSPECTS show a negative correlation (-0.18), which means that there is a minor tendency for the number of suspects to decline as UCR rises.

Geographically speaking, this moderately positive correlation (0.24) between LONGITUDE and LATITUDE makes logical because sites that are closer together will have linked coordinates.

HOUR_TO and LONGITUDE, for example, have correlations that are extremely close to zero (-0.0066), suggesting that there is little to no linear connection between these variables.

The colour gradient—darker red for positive correlations, darker blue for negative correlations, and lighter hues for weaker correlations—helps visualize the direction and intensity of these associations.

Bar Graph on crimes reported vs the time



Most Frequent Time: With about 550 occurrences, 2:00 PM has the highest frequency.

Additional High Frequencies: There are additional high frequency, at or around 550 occurrences, during times such as 12:00 AM, 3:00 PM, 1:00 PM, and 4:00 PM.

Decreasing Frequency: There is a discernible decrease in frequency after 5:00 PM. Lower frequencies are observed at times such as 6:00 PM, 10:00 AM, and 6:30 PM, with occurrences ranging from 300 to 400.

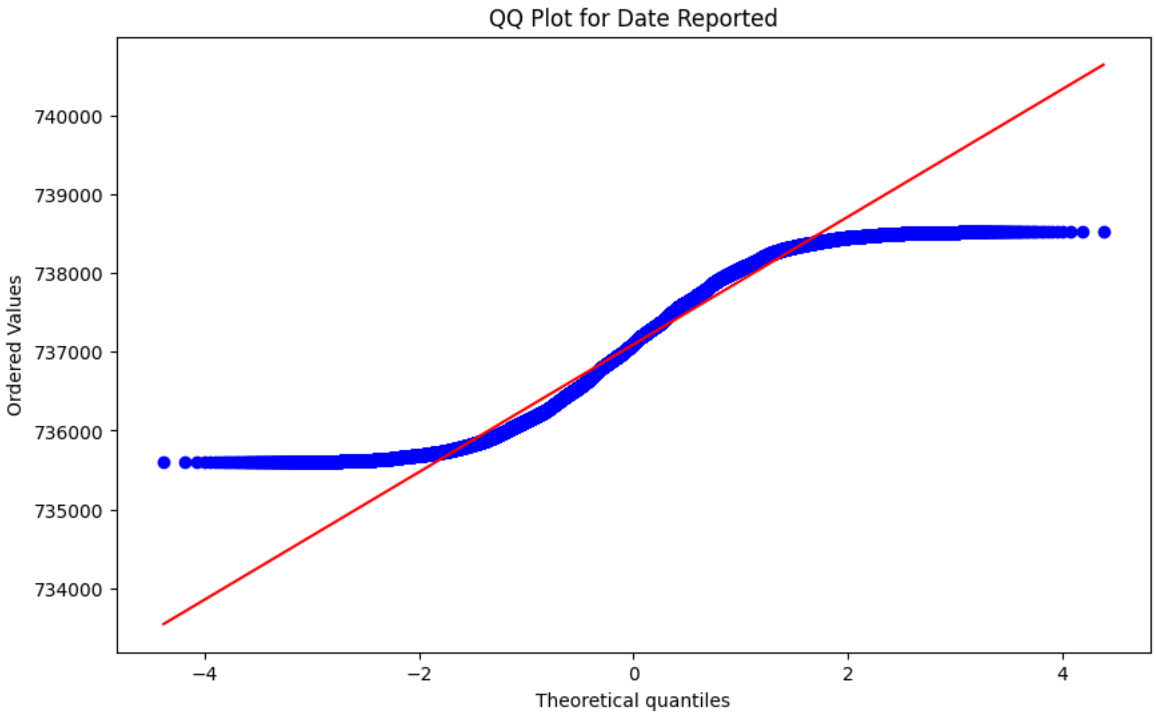
12:00 PM (Noon): Compared to the afternoon hours, noon (12:00 PM) still has a comparatively high frequency.

Night and Early Morning: 6:00 AM and 10:00 AM are less common hours.

Peak reporting hours appear to be in the afternoon and early evening (from around 12:00 PM to 5:00 PM).

Report frequencies are lower in the early morning and late at night, which may indicate that fewer occurrences or reports occur during these times.

Q-Q plot for dates reported



Graph Interpretation:

X-axis (Theoretical Quantiles): The quantiles predicted by a theoretical normal distribution are shown by the X-axis, or theoretical quantiles.

Y-axis (Ordered Values): These are the dataset's actual ordered values, in this instance those pertaining to "Date Reported."

Red Line: The data exactly follows a normal distribution in this ideal scenario, which is represented by this line. The data is said to be regularly distributed if the points lie along this line.

The data points exhibit a tiny S-shaped departure from the red line, signifying a degree of variation from the mean.

The points are below the line at the bottom end (left), indicating a left tail that is heavier than usual.

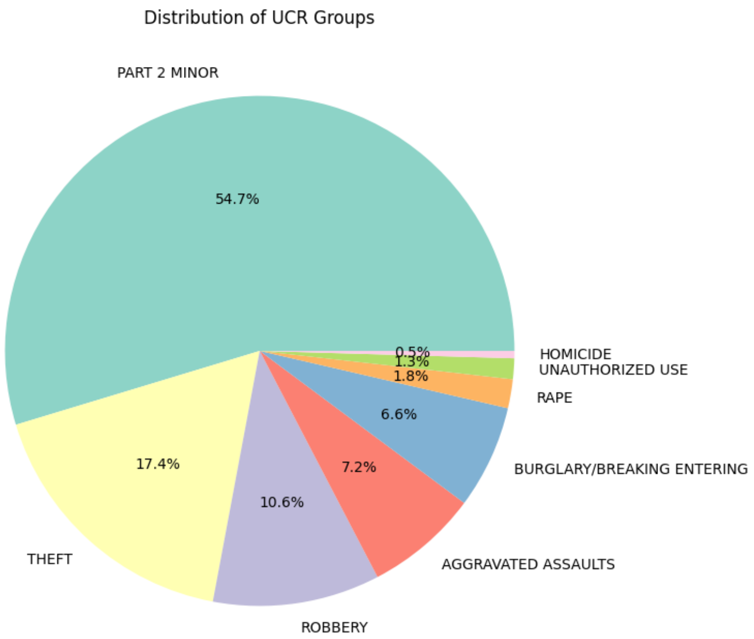
The fact that the core section of the data matches the normal distribution quite well is indicated by the points' reasonable alignment with the line in the middle.

The points are above the line at the top end (right), indicating a right tail that is heavier than usual.

A normal distribution is not exactly followed by the data. The "Date Reported" variable's distribution exhibits some skewness or heavy tails.

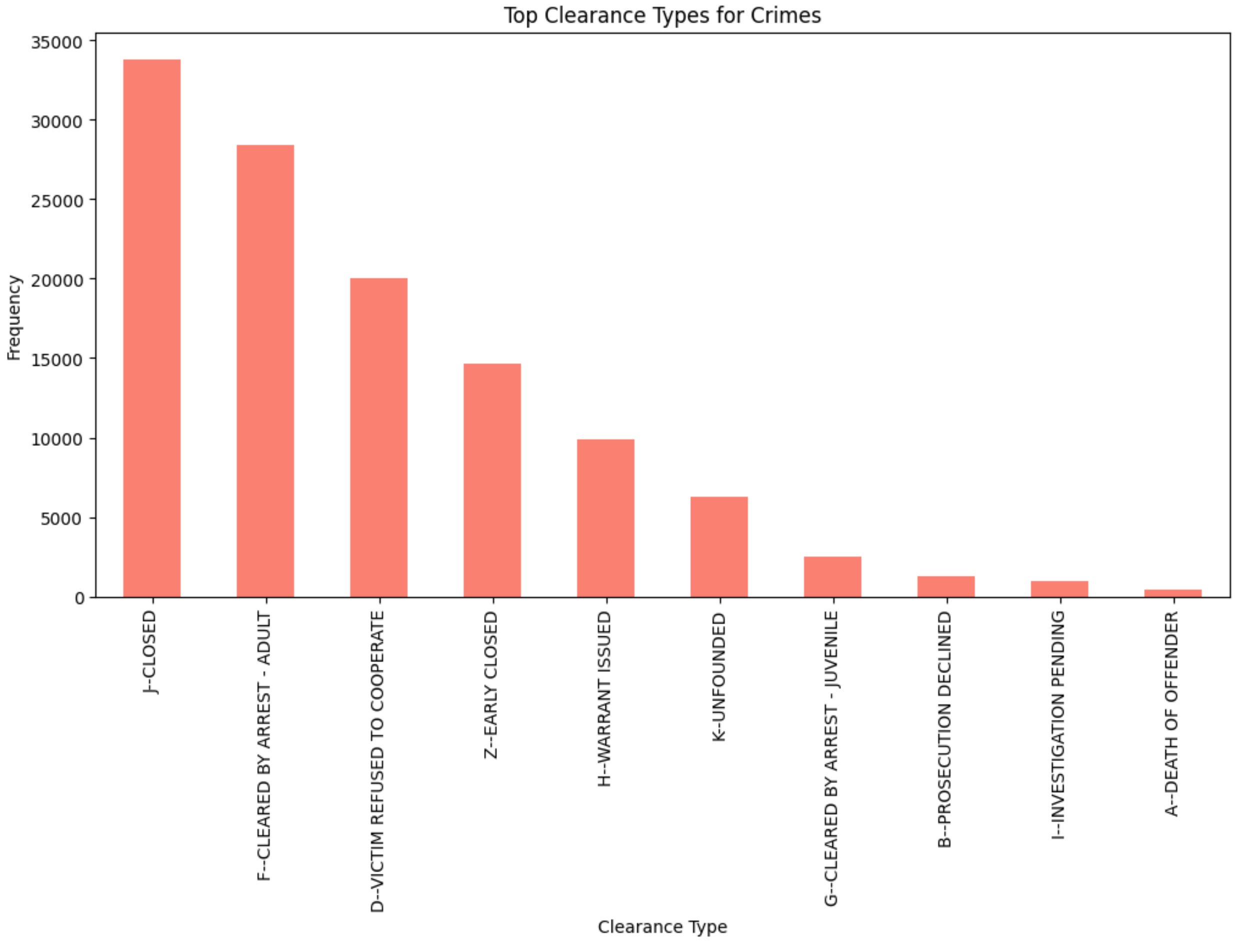
The QQ plot indicates that although the middle data points could be close to normality, there are deviations at the extremes, suggesting that the distribution's tails include non-normal behavior or outliers.

Pie chart for UCR groups



The chart illustrates that theft-related crimes make up the majority of these Part 2 Minor incidents, while violent crimes like homicide and rape represent much smaller percentages.

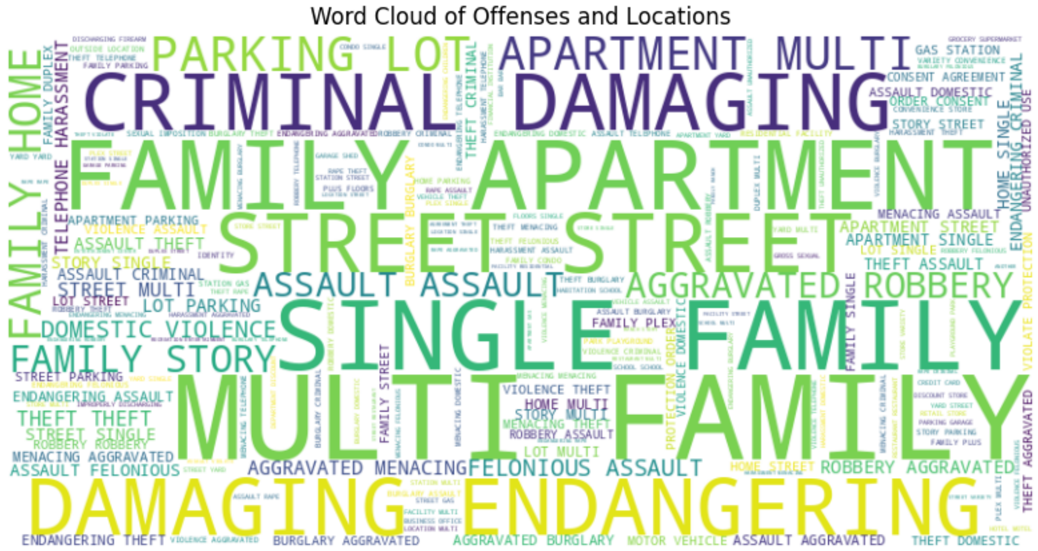
Bar Graph for the Clearance type of reported incidents



The graph exhibits a distinct downward trend from left to right, with "A-DEATH OF OFFENDER" being the least common method of case clearance and "J-CLOSED" being the most common. With adult arrests and administrative closures being the most common ways to clear cases, this graphic shows how law enforcement normally handles criminal cases.

The graph illustrates that most of the reported incident cases are in closed and followed by the cases cleared by arrest. But the third highest types of incident are victim failed to co-operate thus the cases couldn’t be investigated further.

Word Cloud: Offenses and Locations



This word cloud visualizes the most frequently occurring words related to **offenses** and **locations**.

The size of each word represents its frequency in the dataset, with larger words appearing more often.

Key terms such as "FAMILY," "DAMAGING," "MULTI," "APARTMENT," and "STREET" stand out, indicating they are frequently mentioned in crime reports.

The word cloud is useful for quickly identifying the most common terms associated with offenses and their locations.

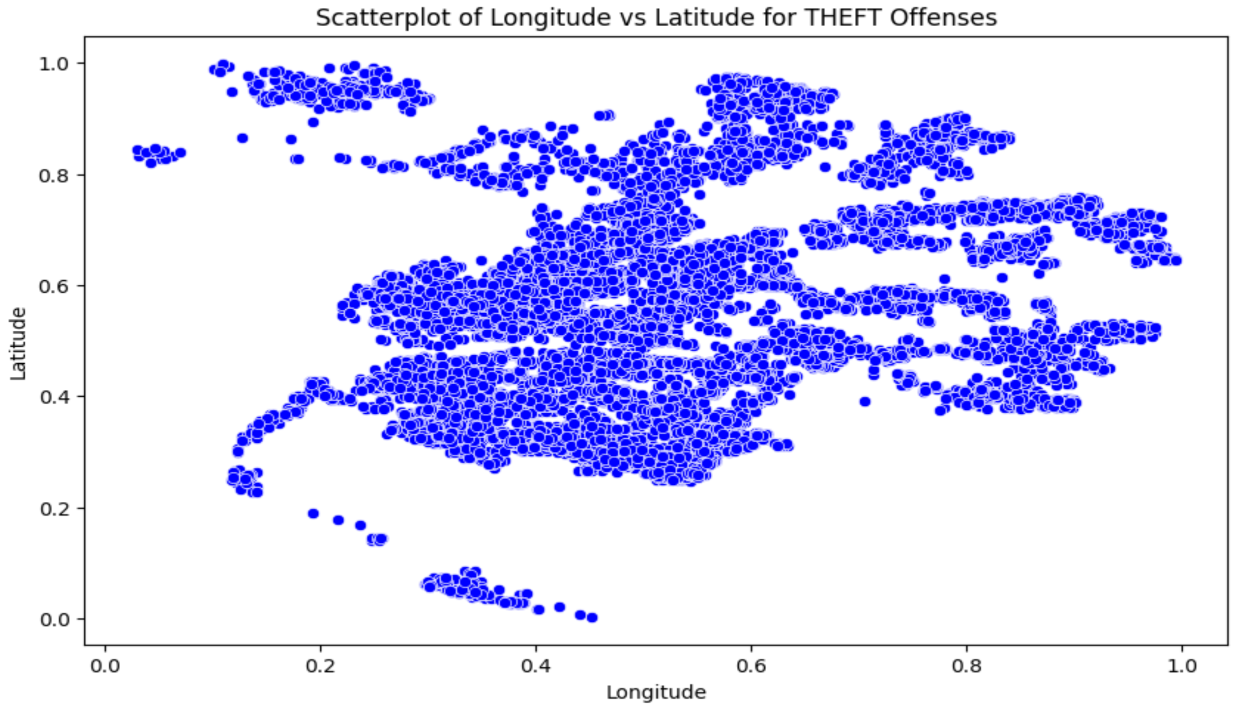
Scatter Plot: Longitude vs Latitude for Theft Offenses

This scatter plot displays the relationship between **Longitude** (x-axis) and **Latitude** (y-axis) for theft offenses.

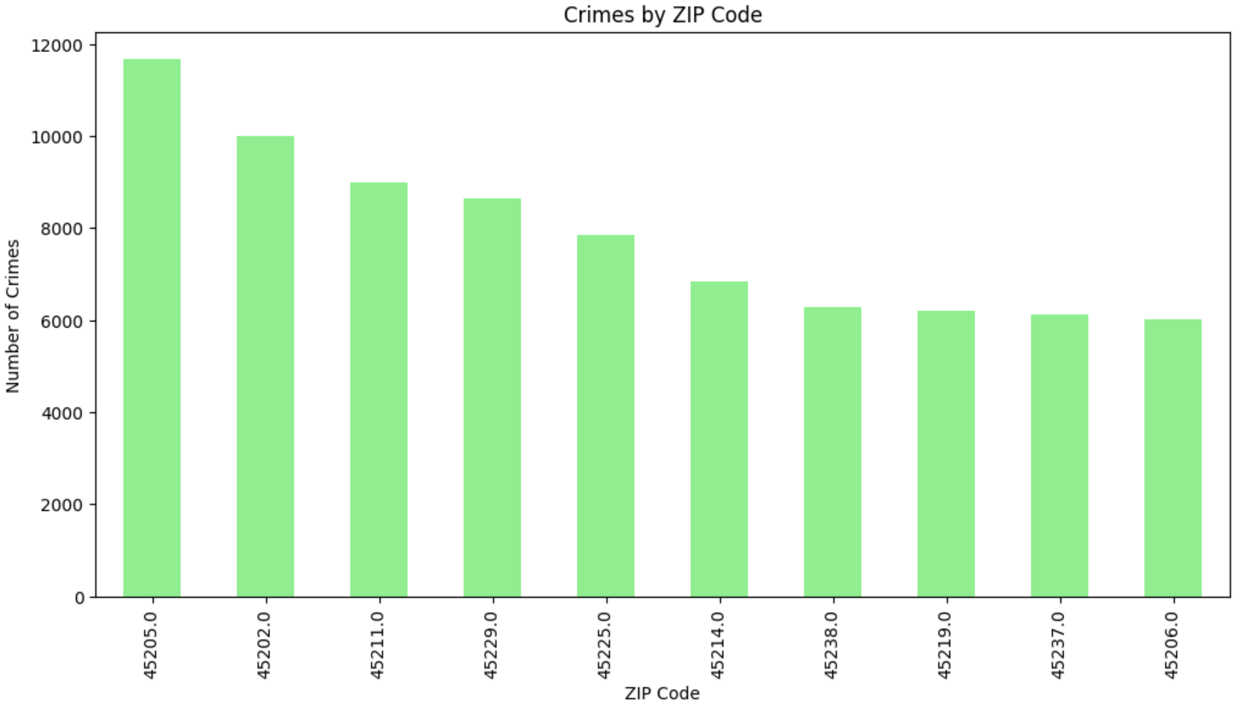
Each point represents a reported theft offense based on its geographic coordinates (longitude and latitude).

The clustering of points suggests that theft offenses are more concentrated in certain geographic regions.

The plot helps to visualize the spatial distribution of theft-related incidents.



Bar Plot: Number of Crimes by ZIP Code



The number of crimes by ZIP code is displayed in this bar chart. The ZIP codes are shown on the X-axis, while the number of offenses is shown on the Y-axis.

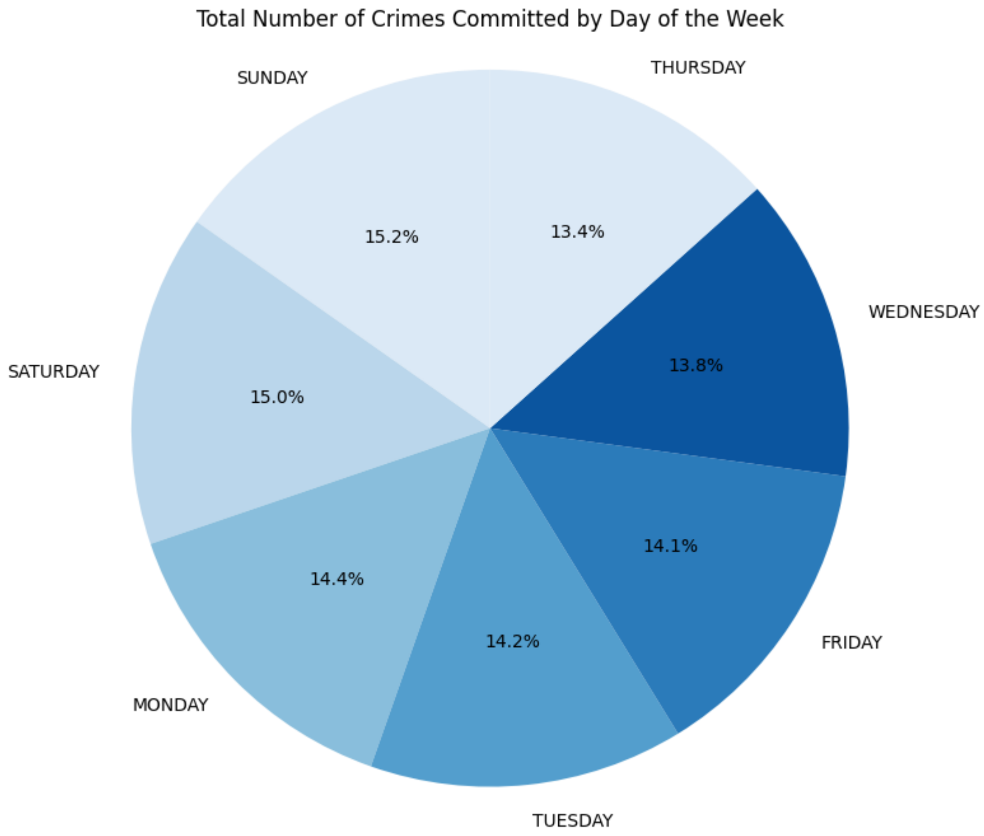
With almost 40,000 offenses, ZIP code 45202.0 has the most.

The next two ZIP codes, 45205.0 and 45211.0, each have almost 30,000 offenses.

There are between 20,000 and 25,000 fewer crimes in other ZIP codes, such as 45214.0, 45225.0, and 45206.0.

The distribution of crime in various places is seen in the chart, with 45202.0 having the most impact.

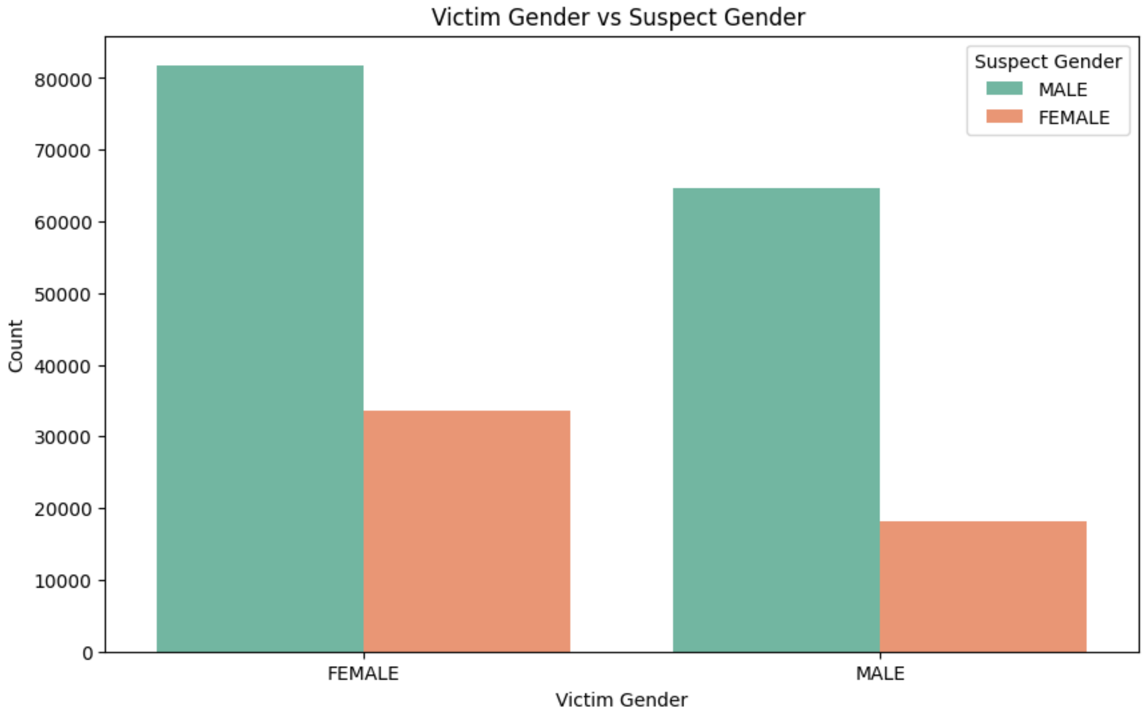
Pie Chart: Total Number of Crimes Committed by Day of the Week



- This pie chart visualizes the **total number of crimes** committed by day of the week.

- **Sunday** has the highest percentage of crimes at 15.4%, followed closely by **Saturday** at 15.0%.
- **Monday** accounts for 14.6% of the total crimes, while **Tuesday** has 14.1%, and **Friday** accounts for 14.0%.
- **Wednesday** and **Thursday** represent the lowest percentages, at 13.6% and 13.3%, respectively.
- This distribution shows that crimes occur fairly evenly throughout the week, with a slight increase during weekends.

Count Plot: Victim Gender vs Suspect Gender



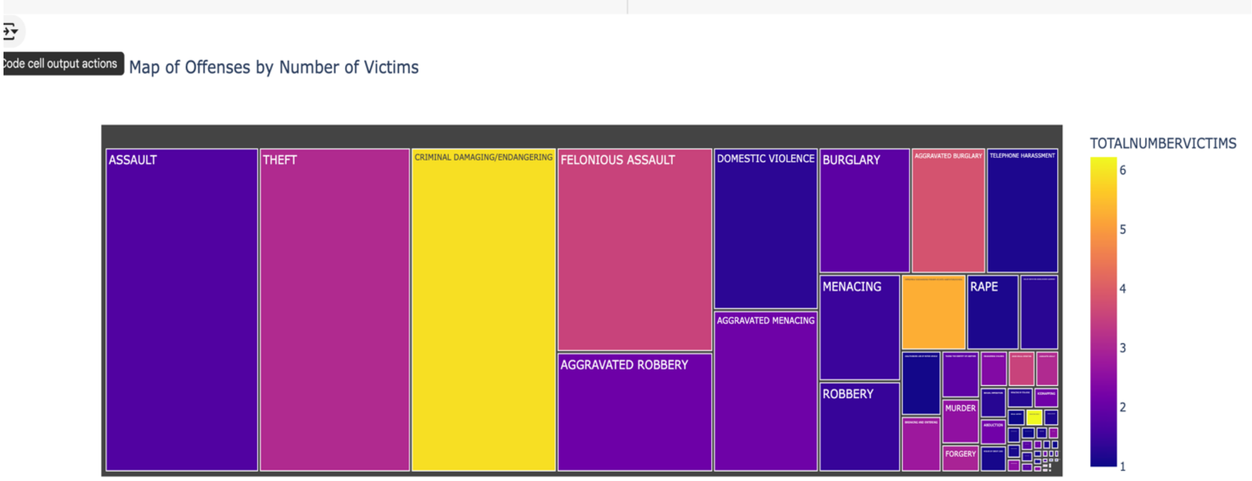
This count plot compares the **victim's gender** with the **suspect's gender** in reported crimes.

For **female victims**, the majority of suspects are **male**, with over 80,000 instances, while female suspects are less frequent.

For **male victims**, the trend is similar, with most suspects being **male**, though the count is lower compared to female victims.

This plot indicates that male suspects are more prevalent in crimes involving both male and female victims.

Treemap: Offenses by Number of Victims



This tree map visualizes the distribution of **offenses** based on the **number of victims**.

Each rectangle represents a different offense, with the size corresponding to the total number of victims.

The color scale indicates the total number of victims, ranging from yellow (higher counts) to dark purple (lower counts).

Assault and **Theft** have the largest number of victims, followed by offenses like **Criminal Damaging/Endangering** and **Felonious Assault**.

Smaller offenses, such as **Rape** and **Menacing**, have fewer victims, as shown by the smaller rectangles.

This visualization provides a clear view of the most impactful offenses in terms of victim count.

Conclusion

Cincinnati Crime Data Analysis Project Conclusion:

Distribution and Classification of Crime:

Part 2 Minor crimes account for the bulk of instances (54.7%), with stealing being the most common infraction.

Theft and burglary are examples of property crimes that greatly outweigh violent crimes.

Thefts involving motor vehicles (23F-THEFT FROM MOTOR VEHICLE) seem to be a significant issue.

Regional Trends:

There are incidents in a number of communities, such as College Hill, West End, West Price Hill, and Winton Hills.

At the same site, several occurrences frequently happen (e.g., parking lots - location code 48)

According to the statistics, crimes seem to concentrate in particular ZIP codes (e.g., 45224, 45214)

Temporal Arrangements:

Crimes happen on several days of the week.

Numerous instances occur late at night or early in the morning.

The timing of event incidence and reporting dates are frequently out of sync.

Resolution of Cases:

There are several kinds of clearances that are utilized, the most popular being:

Adult and juvenile arrests (Z--EARLY CLOSED) J-CLOSED Early closure

Case resolution times differ greatly from one another.

Demographic Perspectives:

Demographics of victims and suspects are sometimes lacking or marked as "UNKNOWN".

Age groupings that are documented vary (ranging from juveniles to 41-50)

When available, gender data demonstrates both male and female engagement.

Reporting Standard:

There are large data gaps in a number of sectors.

Unreliable reporting of demographic data Well-defined geographic data (with exact longitude and latitude coordinates)

Also it has to be noted that the Dataset might not contain all the crimes occurred.

In America, the vast majority of crimes go unreported to the police. According to the National Crime Victimization Survey (NCVS), in 2022, police were notified of just 41.5% of violent crimes and 31.8% of property crimes.

Although more accurate analysis and focused interventions might be possible with improved data collection, this study offers insightful information for the allocation of law enforcement resources and crime prevention tactics.