Chicago

August 27, 2024

1 Chicago Crime Data Exploration

The data covers daily crimes in Chicago in 2024. It's updated daily. Here is the link to the data: https://data.cityofchicago.org/Public-Safety/Crimes-2024/dqcy-ctma/about_data

[3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[11]:
```
# Load the data

file_path = '/Users/YigitAydede/Library/CloudStorage/Dropbox/Documents/Courses/
 ↪MBAN/NLPBootcamp/PythonBC/Crimes_-_2024_20240804.csv'
data = pd.read_csv(file_path)
df = pd.DataFrame(data)
# You can now work with the data variable which contains the contents of the␣
 ↪CSV file
```

1.1 1. Let's see the data

[12]:
```
# Statistical summary of the data
df.describe()
# Display the first 5 rows of the data
print(df.head())
# Display the last 5 rows of the data
print(df.tail())
# Display the shape of the data
print(df.shape)
# Display the column names
print(df.columns)
# Display the data types of the columns
print(df.dtypes)
# Display information about the data
print(df.info())
```

```
         ID Case Number                    Date                   Block  \
0  13543719    JH364008  07/27/2024 12:00:00 AM  018XX S SPRINGFIELD AVE
1  13551073    JH372867  07/27/2024 12:00:00 AM         016XX E 68TH ST
```

```
2  13548638    JH369010  07/27/2024 12:00:00 AM         034XX N CLARK ST
3  13546333    JH367190  07/27/2024 12:00:00 AM          006XX E 90TH ST
4  13544042    JH364428  07/27/2024 12:00:00 AM       001XX W HUBBARD ST

   IUCR         Primary Type               Description Location Description  \
0  0910  MOTOR VEHICLE THEFT                AUTOMOBILE               STREET
1  1130   DECEPTIVE PRACTICE  FRAUD OR CONFIDENCE GAME            APARTMENT
2  0890                THEFT             FROM BUILDING        BAR OR TAVERN
3  0810                THEFT                 OVER $500            RESIDENCE
4  0870                THEFT             POCKET-PICKING        BAR OR TAVERN

   Arrest  Domestic  …  Ward  Community Area  FBI Code  X Coordinate  \
0   False     False  …    24              29        07     1150693.0
1   False     False  …     5              43        11           NaN
2   False     False  …    44               6        06     1168850.0
3   False     False  …     8              44        06     1181979.0
4   False     False  …    42               8        06     1175290.0

  Y Coordinate  Year              Updated On   Latitude  Longitude  \
0    1890610.0  2024  08/03/2024 03:40:46 PM  41.855729 -87.722368
1          NaN  2024  08/03/2024 03:40:46 PM        NaN        NaN
2    1923293.0  2024  08/03/2024 03:40:46 PM  41.945040 -87.654775
3    1845440.0  2024  08/03/2024 03:40:46 PM  41.731111 -87.608932
4    1903292.0  2024  08/03/2024 03:40:46 PM  41.890014 -87.631705

                        Location
0  POINT (-87.722368377 41.855729314)
1                             NaN
2  POINT (-87.654774622 41.945039683)
3  POINT (-87.608931577 41.731110605)
4  POINT (-87.631705393 41.890013771)

[5 rows x 22 columns]
             ID Case Number                    Date  \
144920  13368627    JH152302  01/01/2024 12:00:00 AM
144921  13368833    JH152568  01/01/2024 12:00:00 AM
144922  13369774    JH153703  01/01/2024 12:00:00 AM
144923  13369425    JH153114  01/01/2024 12:00:00 AM
144924  13325302    JH100531  01/01/2024 12:00:00 AM

                                  Block  IUCR        Primary Type  \
144920                  028XX E 77TH ST  1153  DECEPTIVE PRACTICE
144921  058XX S DR MARTIN LUTHER KING JR DR  1540           OBSCENITY
144922              047XX W MAYPOLE AVE  2820       OTHER OFFENSE
144923                  008XX W ERIE ST  0820               THEFT
144924                038XX N DRAKE AVE  0850               THEFT

                                Description  \
144920  FINANCIAL IDENTITY THEFT OVER $ 300
144921                    OBSCENE MATTER
144922                  TELEPHONE THREAT
144923                    $500 AND UNDER
144924                     ATTEMPT THEFT

                          Location Description  Arrest  Domestic  …  Ward  \
144920                         OTHER (SPECIFY)   False     False  …     7
144921                               APARTMENT   False      True  …    20
144922                               APARTMENT   False      True  …    28
144923  PARKING LOT / GARAGE (NON RESIDENTIAL)   False     False  …    27
144924                                  STREET   False     False  …    35

        Community Area  FBI Code  X Coordinate Y Coordinate  Year  \
144920              43        11     1196407.0    1854530.0  2024
144921              40        26     1179922.0    1866345.0  2024
144922              25       08A     1144822.0    1901044.0  2024
144923              24        06     1170456.0    1904469.0  2024
144924              16        06     1152036.0    1925453.0  2024

                    Updated On   Latitude  Longitude  \
144920  02/16/2024 03:40:38 PM  41.755709 -87.555777
144921  05/02/2024 03:41:47 PM  41.788523 -87.615828
144922  02/17/2024 03:40:44 PM  41.884474 -87.743655
144923  02/17/2024 03:40:44 PM  41.893351 -87.649423
144924  01/08/2024 03:59:56 PM  41.951316 -87.716519

                                Location
144920  POINT (-87.555776662 41.755708907)
144921  POINT (-87.615828297 41.788523444)
144922   POINT (-87.74365494 41.884474129)
144923  POINT (-87.649423417 41.893350627)
144924  POINT (-87.716519304 41.951315505)

[5 rows x 22 columns]
(144925, 22)
Index(['ID', 'Case Number', 'Date', 'Block', 'IUCR', 'Primary Type',
       'Description', 'Location Description', 'Arrest', 'Domestic', 'Beat',
       'District', 'Ward', 'Community Area', 'FBI Code', 'X Coordinate',
       'Y Coordinate', 'Year', 'Updated On', 'Latitude', 'Longitude',
       'Location'],
      dtype='object')
ID                        int64
Case Number              object
Date                     object
Block                    object
IUCR                     object
Primary Type             object
```

```
Description              object
Location Description     object
Arrest                     bool
Domestic                   bool
Beat                      int64
District                  int64
Ward                      int64
Community Area            int64
FBI Code                 object
X Coordinate            float64
Y Coordinate            float64
Year                      int64
Updated On               object
Latitude                float64
Longitude               float64
Location                 object
dtype: object
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 144925 entries, 0 to 144924
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   ID                    144925 non-null  int64
 1   Case Number           144925 non-null  object
 2   Date                  144925 non-null  object
 3   Block                 144925 non-null  object
 4   IUCR                  144925 non-null  object
 5   Primary Type          144925 non-null  object
 6   Description           144925 non-null  object
 7   Location Description  144335 non-null  object
 8   Arrest                144925 non-null  bool
 9   Domestic              144925 non-null  bool
 10  Beat                  144925 non-null  int64
 11  District              144925 non-null  int64
 12  Ward                  144925 non-null  int64
 13  Community Area        144925 non-null  int64
 14  FBI Code              144925 non-null  object
 15  X Coordinate          144826 non-null  float64
 16  Y Coordinate          144826 non-null  float64
 17  Year                  144925 non-null  int64
 18  Updated On            144925 non-null  object
 19  Latitude              144826 non-null  float64
 20  Longitude             144826 non-null  float64
 21  Location              144826 non-null  object
dtypes: bool(2), float64(4), int64(6), object(10)
memory usage: 22.4+ MB
None
```

1.2 2. Selecting columns and rows

[13]:
```
print("\nSelected columns:")
selected_columns = ['ID', 'Primary Type', 'Description', 'Location␣
↪Description', 'Arrest', 'Domestic']
print(df[selected_columns].head())
```

```
Selected columns:
         ID         Primary Type               Description  \
0  13543719  MOTOR VEHICLE THEFT                AUTOMOBILE
1  13551073   DECEPTIVE PRACTICE  FRAUD OR CONFIDENCE GAME
2  13548638                THEFT             FROM BUILDING
3  13546333                THEFT                 OVER $500
4  13544042                THEFT            POCKET-PICKING

  Location Description  Arrest  Domestic
0               STREET   False     False
1            APARTMENT   False     False
2        BAR OR TAVERN   False     False
3            RESIDENCE   False     False
4        BAR OR TAVERN   False     False
```

[14]:
```
print("\nSelected rows based on condition (e.g., Arrest == True):")
arrested_crimes = df[df['Arrest'] == True]
print(arrested_crimes.head())
```

```
Selected rows based on condition (e.g., Arrest == True):
          ID Case Number                    Date                   Block  \
20  13543641    JH363989  07/27/2024 12:00:00 AM         053XX N BROADWAY
23  13543156    JH363451  07/26/2024 11:58:00 PM          009XX W LAKE ST
27  13543227    JH363420  07/26/2024 11:51:00 PM          003XX E 75TH ST
28  13543125    JH363403  07/26/2024 11:51:00 PM  006XX S SPRINGFIELD AVE
31  13543260    JH363408  07/26/2024 11:46:00 PM      034XX W BELMONT AVE

    IUCR           Primary Type                    Description  \
20  0460                BATTERY                         SIMPLE
23  0470  PUBLIC PEACE VIOLATION              RECKLESS CONDUCT
27  143A       WEAPONS VIOLATION  UNLAWFUL POSSESSION - HANDGUN
28  2024               NARCOTICS       POSSESS - HEROIN (WHITE)
31  0860                  THEFT                   RETAIL THEFT

   Location Description  Arrest  Domestic  …  Ward  Community Area  \
20   GROCERY FOOD STORE    True     False  …    48              77
23             SIDEWALK    True     False  …    27              28
27               STREET    True     False  …     6              69
28               STREET    True     False  …    24              26
31           DRUG STORE    True     False  …    35              21
```

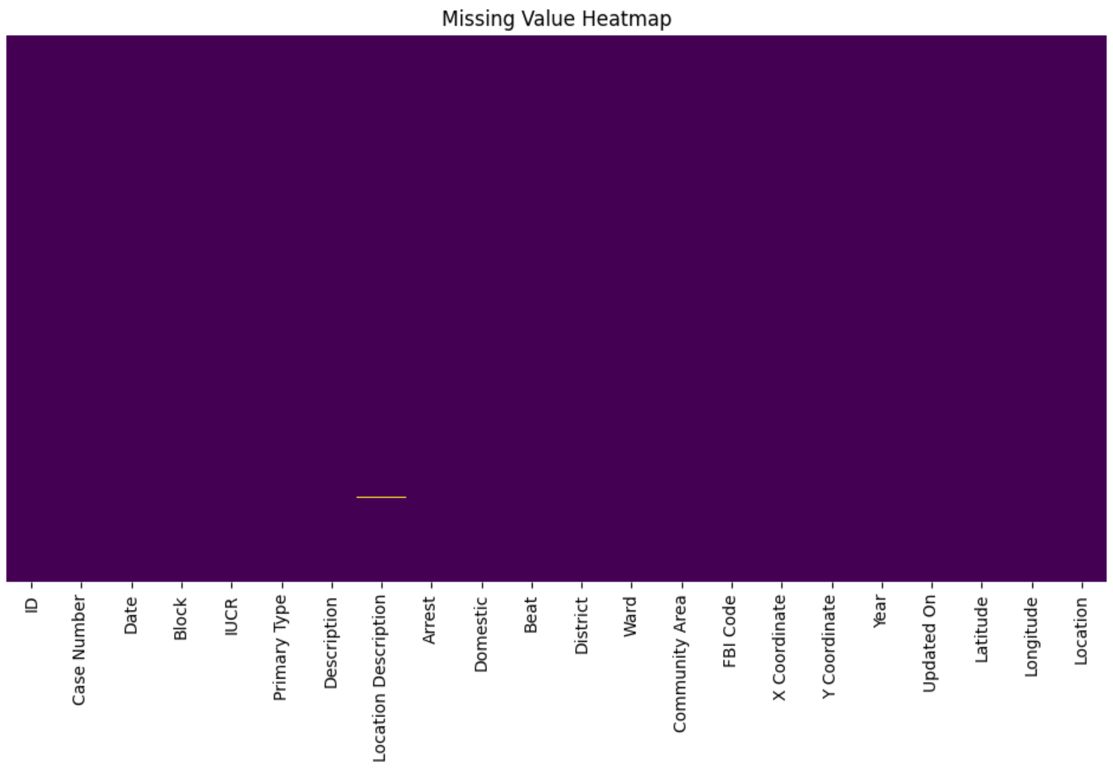
```
   FBI Code  X Coordinate Y Coordinate  Year             Updated On  \
20      08B     1167347.0    1935654.0  2024  08/03/2024 03:40:46 PM
23       24     1169954.0    1901640.0  2024  08/03/2024 03:40:46 PM
27       15     1179508.0    1855350.0  2024  08/03/2024 03:40:46 PM
28       18     1150484.0    1896982.0  2024  08/03/2024 03:40:46 PM
31       06     1152928.0    1921067.0  2024  08/03/2024 03:40:46 PM

     Latitude  Longitude                      Location
20  41.978991 -87.659942  POINT (-87.659941972 41.978991238)
23  41.885599 -87.651350  POINT (-87.651349647 41.885598628)
27  41.758362 -87.617682  POINT (-87.617681742 41.758361509)
28  41.873219 -87.722969   POINT (-87.72296923 41.873218908)
31  41.939262 -87.713357   POINT (-87.71335694 41.939262335)

[5 rows x 22 columns]
```

1.3 3. Missing values

[16]:
```
print("\nMissing values in each column:")
print(df.isnull().sum())
```

```
Missing values in each column:
ID                        0
Case Number               0
Date                      0
Block                     0
IUCR                      0
Primary Type              0
Description               0
Location Description    590
Arrest                    0
Domestic                  0
Beat                      0
District                  0
Ward                      0
Community Area            0
FBI Code                  0
X Coordinate             99
Y Coordinate             99
Year                      0
Updated On                0
Latitude                 99
Longitude                99
Location                 99
dtype: int64
```

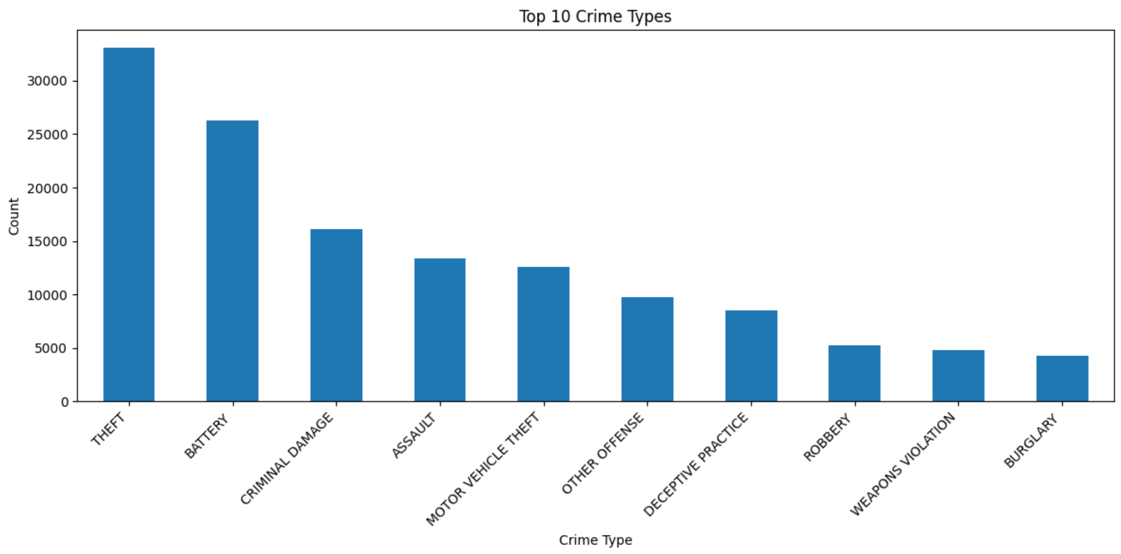
[15]:
```
# Visualize missing values
plt.figure(figsize=(12, 6))
sns.heatmap(df.isnull(), yticklabels=False, cbar=False, cmap='viridis')
plt.title('Missing Value Heatmap')
plt.show()
```



1.4 4. Explore the data

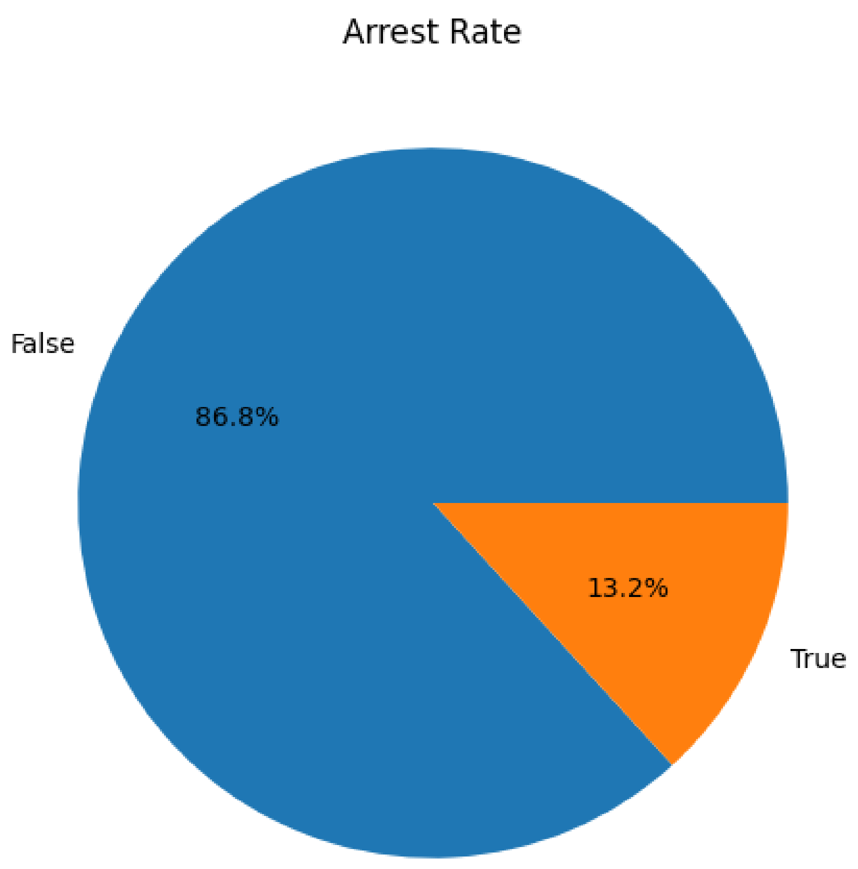
[17]:
```
# Analyzes crime type distribution with a bar plot of the top 10 crime types
print("\nCrime type distribution:")
crime_type_counts = df['Primary Type'].value_counts()
print(crime_type_counts)

plt.figure(figsize=(12, 6))
crime_type_counts[:10].plot(kind='bar')
plt.title('Top 10 Crime Types')
plt.xlabel('Crime Type')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
Crime type distribution:
Primary Type
THEFT                                33076
BATTERY                              26285
CRIMINAL DAMAGE                      16107
ASSAULT                              13363
MOTOR VEHICLE THEFT                  12620
OTHER OFFENSE                         9719
DECEPTIVE PRACTICE                    8499
ROBBERY                               5213
WEAPONS VIOLATION                     4768
BURGLARY                              4282
NARCOTICS                             3318
CRIMINAL TRESPASS                     2732
OFFENSE INVOLVING CHILDREN            1019
CRIMINAL SEXUAL ASSAULT                840
SEX OFFENSE                            716
PUBLIC PEACE VIOLATION                 553
INTERFERENCE WITH PUBLIC OFFICER       388
HOMICIDE                               335
ARSON                                  273
STALKING                               256
PROSTITUTION                           152
LIQUOR LAW VIOLATION                   112
CONCEALED CARRY LICENSE VIOLATION      110
INTIMIDATION                            86
KIDNAPPING                              51
OBSCENITY                               33
GAMBLING                                10
PUBLIC INDECENCY                         4
OTHER NARCOTIC VIOLATION                 2
HUMAN TRAFFICKING                        2
NON-CRIMINAL                             1
Name: count, dtype: int64
```



[18]:
```python
# Calculates and visualizes the arrest rate
print("\nArrest rate:")
arrest_rate = df['Arrest'].value_counts(normalize=True)
print(arrest_rate)

plt.figure(figsize=(8, 6))
arrest_rate.plot(kind='pie', autopct='%1.1f%%')
plt.title('Arrest Rate')
plt.ylabel('')
plt.show()
```

```
Arrest rate:
Arrest
False    0.86778
True     0.13222
Name: proportion, dtype: float64
```



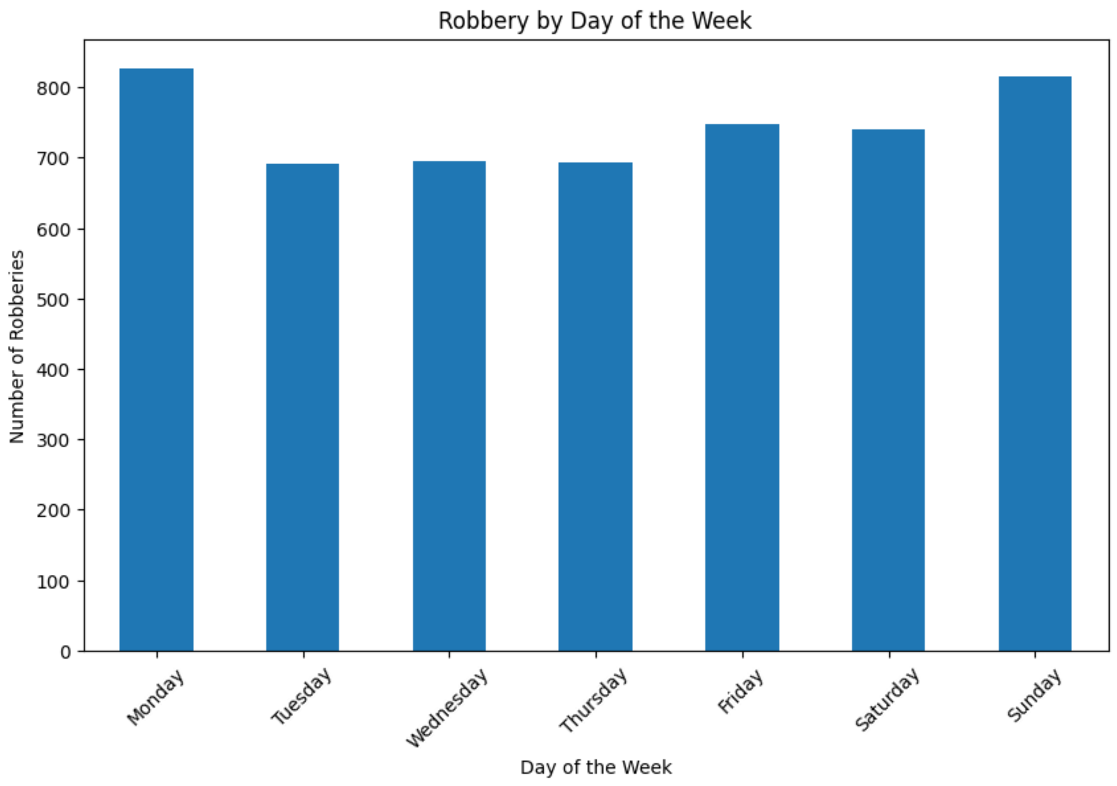
[20]:
```python
# Calculate arrest rates by crime types
arrest_rates = df.groupby('Primary Type')['Arrest'].mean()

# Display the arrest rates
print(arrest_rates.sort_values(ascending=False))
```

```
Primary Type
GAMBLING                             1.000000
PUBLIC INDECENCY                     1.000000
OTHER NARCOTIC VIOLATION             1.000000
CONCEALED CARRY LICENSE VIOLATION    0.963636
NARCOTICS                            0.946956
LIQUOR LAW VIOLATION                 0.946429
PROSTITUTION                         0.940789
INTERFERENCE WITH PUBLIC OFFICER     0.878866
WEAPONS VIOLATION                    0.584312
PUBLIC PEACE VIOLATION               0.520796
OBSCENITY                            0.424242
```

```
CRIMINAL TRESPASS                    0.286237
HOMICIDE                             0.194030
OTHER OFFENSE                        0.191069
BATTERY                              0.163477
ASSAULT                              0.101923
ARSON                                0.076923
SEX OFFENSE                          0.071229
STALKING                             0.070312
OFFENSE INVOLVING CHILDREN           0.062807
THEFT                                0.061223
ROBBERY                              0.054096
BURGLARY                             0.040168
CRIMINAL DAMAGE                      0.034705
DECEPTIVE PRACTICE                   0.032945
MOTOR VEHICLE THEFT                  0.029002
CRIMINAL SEXUAL ASSAULT              0.020238
KIDNAPPING                           0.019608
INTIMIDATION                         0.011628
NON-CRIMINAL                         0.000000
HUMAN TRAFFICKING                    0.000000
Name: Arrest, dtype: float64
```

[26]:
```python
import datetime

# Ensure the 'Date' column is in datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Filter the data for 'ROBBERY'
robbery_data = df[df['Primary Type'] == 'ROBBERY']

# Extract the day of the week
robbery_data['Day of Week'] = robbery_data['Date'].dt.day_name()

# Group the data by day of the week and count the number of robberies
robbery_by_day = robbery_data.groupby('Day of Week').size()

# Reindex to ensure all days are represented
days_of_week = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',␣
 ↪'Saturday', 'Sunday']
robbery_by_day = robbery_by_day.reindex(days_of_week, fill_value=0)

# Create the bar plot
plt.figure(figsize=(10, 6))
robbery_by_day.plot(kind='bar')
plt.title('Robbery by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Robberies')
plt.xticks(rotation=45)
plt.show()
```

```
/var/folders/b2/gpnsjh9j6bv5prtx7w5lsym80000gp/T/ipykernel_84591/3956560368.py:1
0: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  robbery_data['Day of Week'] = robbery_data['Date'].dt.day_name()
```



1.5 5. Mapping the 2024 Chicago Crime Data

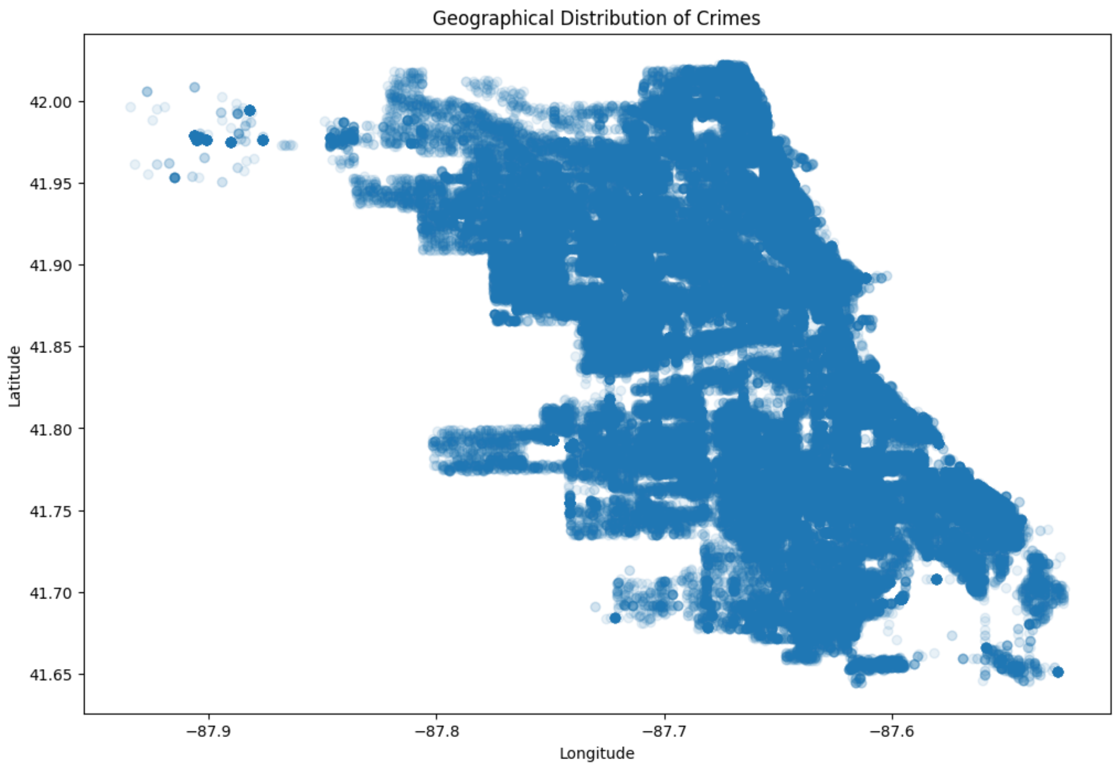
[2]:
```
%%capture
!pip install folium
```

[29]:
```
import folium
from folium.plugins import HeatMap

# Create a map centered around Chicago
chicago_map = folium.Map(location=[41.8781, -87.6298], zoom_start=10)
```

```
# Create a HeatMap layer using the crime data
heat_data = df[['Latitude', 'Longitude']].dropna()
heat_map = HeatMap(data=heat_data, radius=15)

# Add the HeatMap layer to the map
heat_map.add_to(chicago_map)

# Display the map
chicago_map
```

[29]:
```
<folium.folium.Map at 0x3634ba890>
```

[30]:
```
# 6. Geographical Distribution
plt.figure(figsize=(12, 8))
plt.scatter(df['Longitude'], df['Latitude'], alpha=0.1)
plt.title('Geographical Distribution of Crimes')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()
```

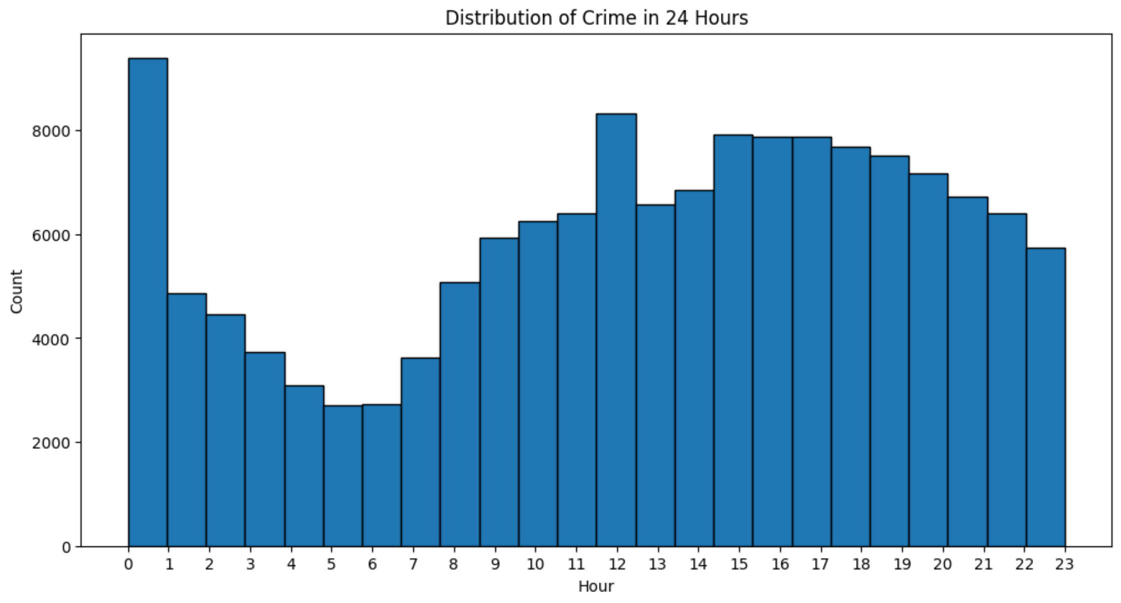


1.6 6. Time-based Analysis

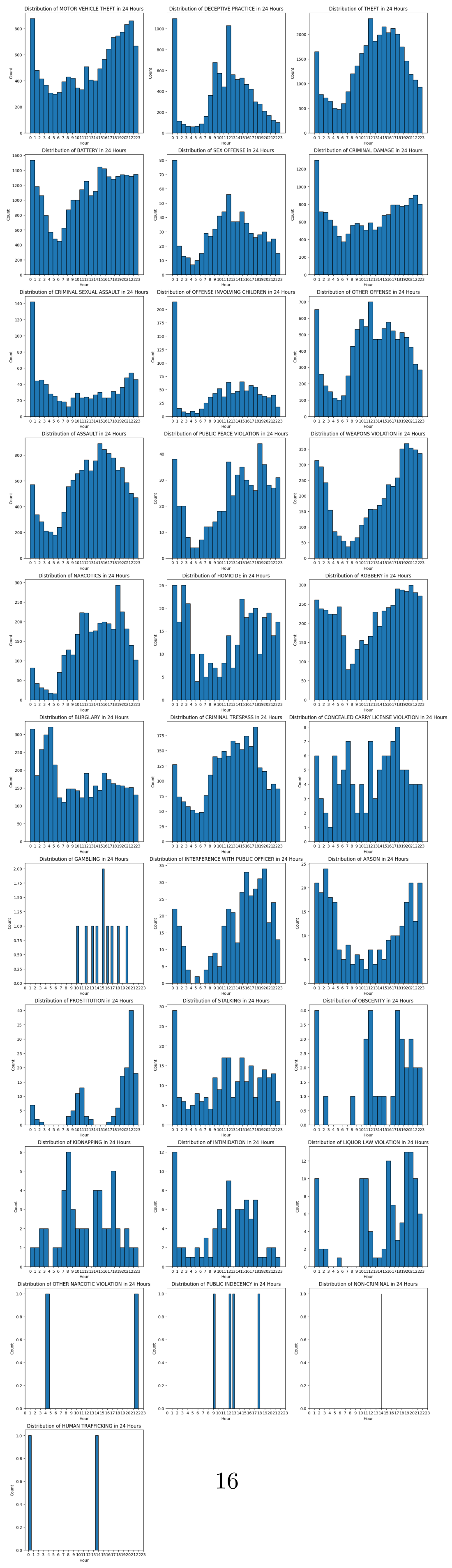
Extracts the hour from the date and analyzes hourly crime distribution

[40]:
```
file_path = '/Users/YigitAydede/Library/CloudStorage/Dropbox/Documents/Courses/
  ↪MBAN/NLPBootcamp/PythonBC/Crimes_-_2024_20240804.csv'
df = pd.read_csv(file_path, parse_dates=['Date'])

# Ensure the 'Date' column is in datetime format with AM/PM
df['Date'] = pd.to_datetime(df['Date'], format='%m/%d/%Y %I:%M:%S %p',␣
  ↪errors='coerce')

# Extract the hour from the 'Date' column
df['Hour'] = df['Date'].dt.hour

# Create a histogram of the crime distribution in 24 hours
plt.figure(figsize=(12, 6))
plt.hist(df['Hour'], bins=24, edgecolor='black')
plt.title('Distribution of Crime in 24 Hours')
plt.xlabel('Hour')
plt.ylabel('Count')
plt.xticks(range(24))
plt.show()
```

```
/var/folders/b2/gpnsjh9j6bv5prtx7w5lsym80000gp/T/ipykernel_84591/1865543446.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 = pd.read_csv(file_path, parse_dates=['Date'])
```



[42]:
```
import numpy as np

### Each crime type in 24 hours
# Get unique crime types
crime_types = df['Primary Type'].unique()

# Determine the number of rows and columns for the subplots
num_crime_types = len(crime_types)
num_cols = 3
num_rows = int(np.ceil(num_crime_types / num_cols))

# Create subplots
fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 5 * num_rows),␣
 ↪constrained_layout=True)

# Plot histograms for each crime type
for i, crime_type in enumerate(crime_types):
    row = i // num_cols
    col = i % num_cols
    ax = axes[row, col] if num_rows > 1 else axes[col]

    # Filter data for the current crime type
    crime_data = df[df['Primary Type'] == crime_type]

    # Create histogram
    ax.hist(crime_data['Hour'], bins=24, edgecolor='black')
    ax.set_title(f'Distribution of {crime_type} in 24 Hours')
    ax.set_xlabel('Hour')
    ax.set_ylabel('Count')
    ax.set_xticks(range(24))

# Remove any empty subplots
for j in range(i + 1, num_rows * num_cols):
    row = j // num_cols
    col = j % num_cols
    fig.delaxes(axes[row, col] if num_rows > 1 else axes[col])

# Display the plots
plt.show()
```


Distribution of MOTOR VEHICLE THEFT in 24 Hours
Distribution of DECEPTIVE PRACTICE in 24 Hours
Distribution of THEFT in 24 Hours
Distribution of BATTERY in 24 Hours
Distribution of SEX OFFENSE in 24 Hours
Distribution of CRIMINAL DAMAGE in 24 Hours
Distribution of CRIMINAL SEXUAL ASSAULT in 24 Hours
Distribution of OFFENSE INVOLVING CHILDREN in 24 Hours
Distribution of OTHER OFFENSE in 24 Hours
Distribution of ASSAULT in 24 Hours
Distribution of PUBLIC PEACE VIOLATION in 24 Hours
Distribution of WEAPONS VIOLATION in 24 Hours
Distribution of NARCOTICS in 24 Hours
Distribution of HOMICIDE in 24 Hours
Distribution of ROBBERY in 24 Hours
Distribution of BURGLARY in 24 Hours
Distribution of CRIMINAL TRESPASS in 24 Hours
Distribution of CONCEALED CARRY LICENSE VIOLATION in 24 Hours
Distribution of GAMBLING in 24 Hours
Distribution of INTERFERENCE WITH PUBLIC OFFICER in 24 Hours
Distribution of ARSON in 24 Hours
Distribution of PROSTITUTION in 24 Hours
Distribution of STALKING in 24 Hours
Distribution of OBSCENITY in 24 Hours
Distribution of KIDNAPPING in 24 Hours
Distribution of INTIMIDATION in 24 Hours
Distribution of LIQUOR LAW VIOLATION in 24 Hours
Distribution of OTHER NARCOTIC VIOLATION in 24 Hours
Distribution of PUBLIC INDECENCY in 24 Hours
Distribution of NON-CRIMINAL in 24 Hours
Distribution of HUMAN TRAFFICKING in 24 Hours
Count
Hour