Práctica de Laboratorio: Análisis Exploratorio de Datos - Data Wrangling 🧹📊

Docente: Ana Maria Cuadros Valdivia
Tema: Proyecto de Criminalidad en Los Ángeles

Introducción

En este análisis exploratorio de datos (EDA) abordamos un conjunto de datos de criminalidad en Los Ángeles. El objetivo principal es comprender la estructura de los datos, detectar posibles problemas de calidad, entender las relaciones entre las variables y visualizar patrones importantes que pueden influir en el comportamiento delictivo.

Paso 1: Análisis del comportamiento de los datos

In [3]:
```
# Cargar los datos
import pandas as pd

df = pd.read_csv('Crime_Data_from_2020_to_Present.csv')  # Cambiar al nombre cor
df.head()
```

Out[3]:

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm
0	190326475	03/01/2020 12:00:00 AM	03/01/2020 12:00:00 AM	2130	7	Wilshire	784	1	510	V
1	200106753	02/09/2020 12:00:00 AM	02/08/2020 12:00:00 AM	1800	1	Central	182	1	330	BU
2	200320258	11/11/2020 12:00:00 AM	11/04/2020 12:00:00 AM	1700	3	Southwest	356	1	480	BIKE -
3	200907217	05/10/2023 12:00:00 AM	03/10/2020 12:00:00 AM	2037	9	Van Nuys	964	1	343	SHOP GRAN ($
4	200412582	09/09/2020 12:00:00 AM	09/09/2020 12:00:00 AM	630	4	Hollenbeck	413	1	510	V

5 rows × 28 columns

In [4]:
```python
# Número de registros
print(f"Número de registros: {df.shape[0]}")
print(f"Número de columnas: {df.shape[1]}")

# Verificar duplicados
print(f"Registros duplicados: {df.duplicated().sum()}")
```

```
Número de registros: 1005109
Número de columnas: 28
Registros duplicados: 0
```

In [5]:
```python
# Tipos de datos
df.dtypes
```

Out[5]:

DR_NO	int64
Date Rptd	object
DATE OCC	object
TIME OCC	int64
AREA	int64
AREA NAME	object
Rpt Dist No	int64
Part 1-2	int64
Crm Cd	int64
Crm Cd Desc	object
Mocodes	object
Vict Age	int64
Vict Sex	object
Vict Descent	object
Premis Cd	float64
Premis Desc	object
Weapon Used Cd	float64
Weapon Desc	object
Status	object
Status Desc	object
Crm Cd 1	float64
Crm Cd 2	float64
Crm Cd 3	float64
Crm Cd 4	float64
LOCATION	object
Cross Street	object
LAT	float64
LON	float64

dtype: object

In [6]:
```python
# Valores únicos, mínimos y máximos por columna
df.describe(include='all')
```

Out[6]:

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	
count	1.005109e+06	1005109	1005109	1.005109e+06	1.005109e+06	1005109	1.
unique	NaN	1908	1894	NaN	NaN	21	
top	NaN	02/02/2023 12:00:00 AM	01/01/2020 12:00:00 AM	NaN	NaN	Central	
freq	NaN	929	1164	NaN	NaN	69672	
mean	2.202251e+08	NaN	NaN	1.339914e+03	1.069115e+01	NaN	1.
std	1.320042e+07	NaN	NaN	6.510476e+02	6.110394e+00	NaN	6.
min	8.170000e+02	NaN	NaN	1.000000e+00	1.000000e+00	NaN	1.
25%	2.106169e+08	NaN	NaN	9.000000e+02	5.000000e+00	NaN	5.
50%	2.209160e+08	NaN	NaN	1.420000e+03	1.100000e+01	NaN	1.
75%	2.311104e+08	NaN	NaN	1.900000e+03	1.600000e+01	NaN	1.
max	2.521041e+08	NaN	NaN	2.359000e+03	2.100000e+01	NaN	2.

11 rows × 28 columns

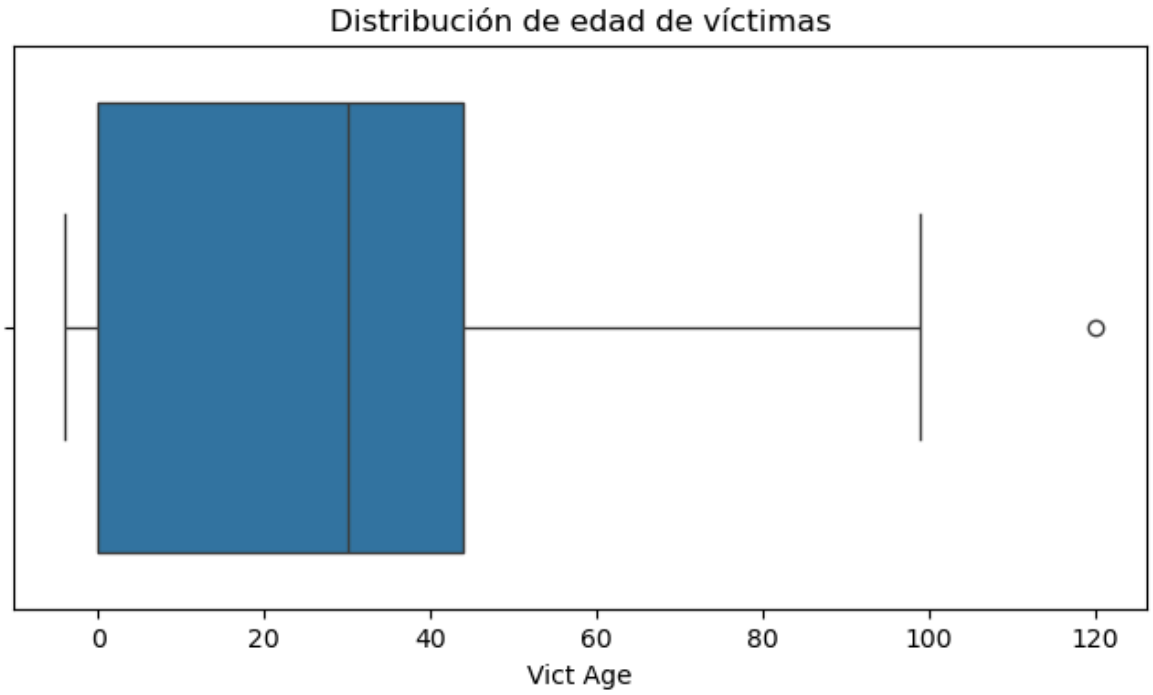
In [7]:
```
# Revisión de valores nulos
df.isnull().sum()
```

Out[7]:
```
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          151706
Vict Age              0
Vict Sex         144730
Vict Descent     144742
Premis Cd            16
Premis Desc         588
Weapon Used Cd   677850
Weapon Desc      677850
Status                1
Status Desc           0
Crm Cd 1             11
Crm Cd 2         935955
Crm Cd 3        1002795
Crm Cd 4        1005045
LOCATION              0
Cross Street     850872
LAT                   0
LON                   0
dtype: int64
```

Paso 2: Análisis de Outliers

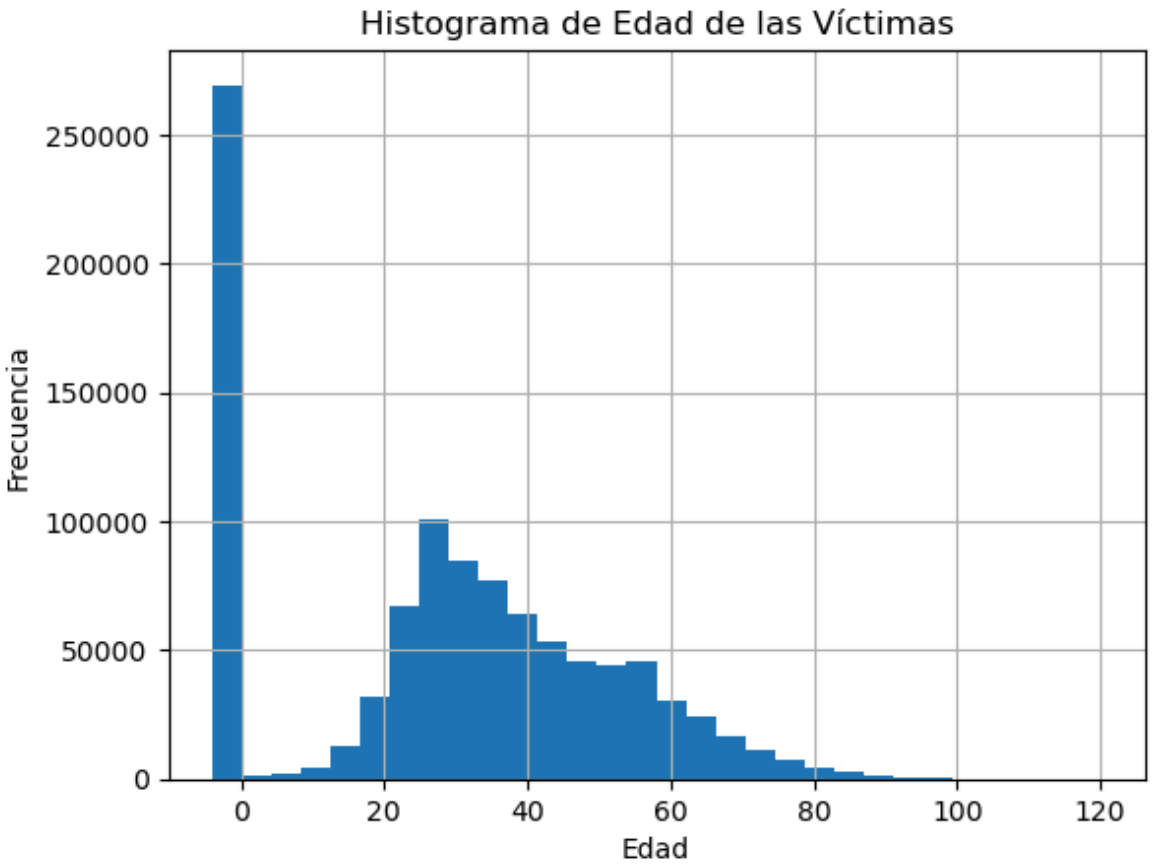
In [8]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Boxplot para detectar outliers en la variable 'Vict Age' (si existe)
if 'Vict Age' in df.columns:
    plt.figure(figsize=(8,4))
    sns.boxplot(x=df['Vict Age'])
    plt.title('Distribución de edad de víctimas')
    plt.show()
```

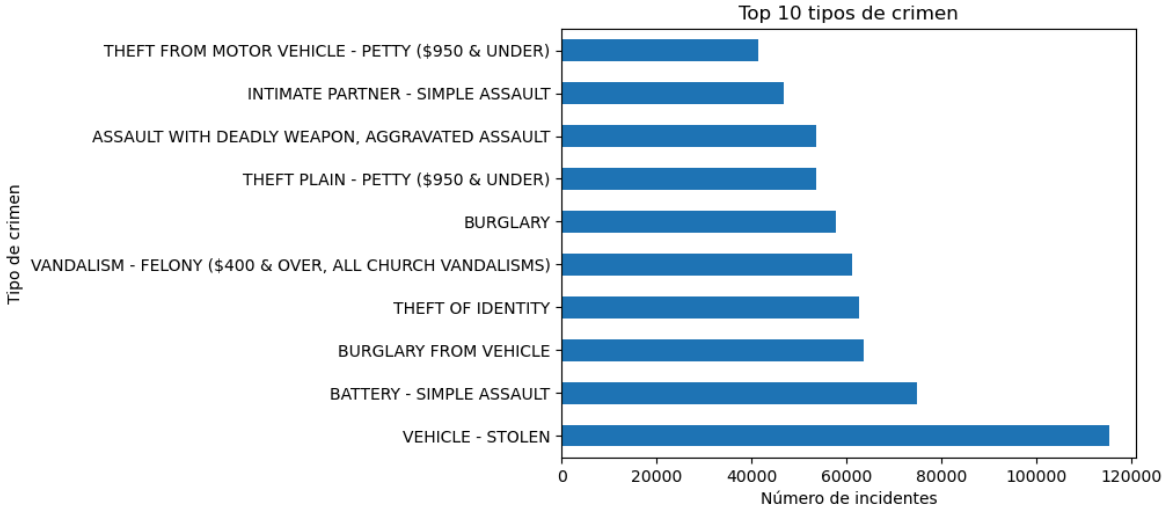


Paso 3: Visualización de variables

In [9]:
```
# Histograma de edades
if 'Vict Age' in df.columns:
    df['Vict Age'].hist(bins=30)
    plt.title("Histograma de Edad de las Víctimas")
    plt.xlabel("Edad")
    plt.ylabel("Frecuencia")
    plt.show()
```



In [10]:
```python
# Gráfico de barras para tipos de crimen
if 'Crm Cd Desc' in df.columns:
    df['Crm Cd Desc'].value_counts().nlargest(10).plot(kind='barh')
    plt.title("Top 10 tipos de crimen")
    plt.xlabel("Número de incidentes")
    plt.ylabel("Tipo de crimen")
    plt.show()
```



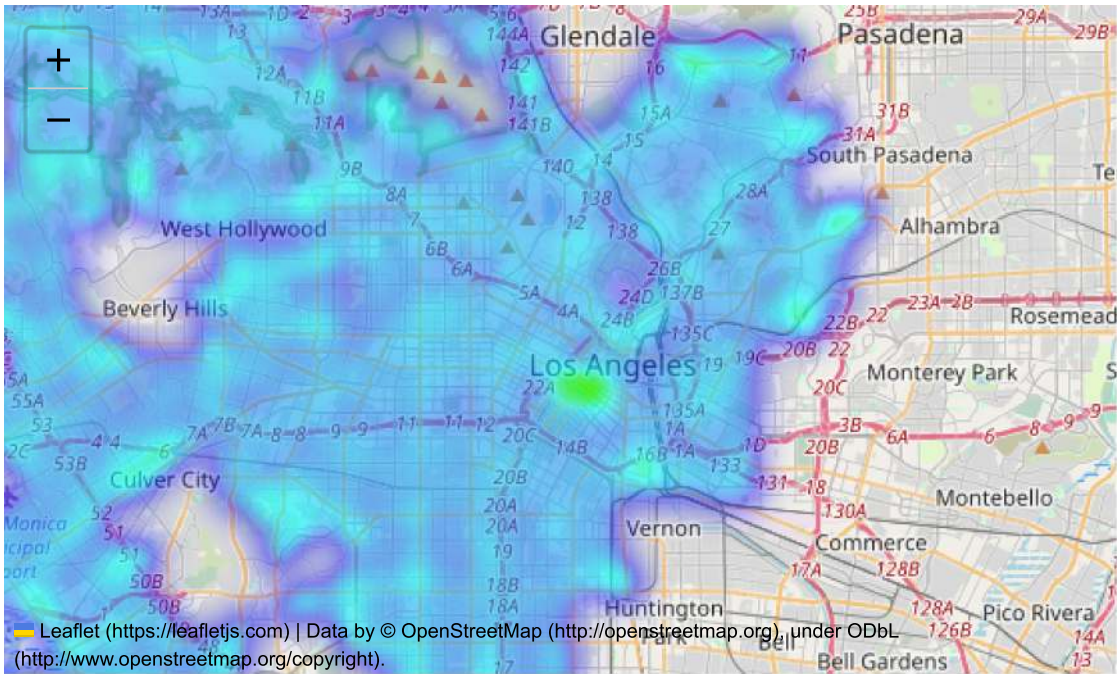
In [ ]:
```python
import folium
from folium.plugins import HeatMap

# Crear un mapa centrado en Los Ángeles
m = folium.Map(location=[34.0522, -118.2437], zoom_start=11)

# Preparar los datos para el HeatMap
if 'LAT' in df.columns and 'LON' in df.columns:
    heat_data = df[['LAT', 'LON']].dropna().values.tolist()
    HeatMap(heat_data, radius=8, blur=15, max_zoom=13).add_to(m)

m  # Mostrar el mapa en la celda
```

Out[ ]:



Paso 4: Problemas potenciales en los datos

- Si este fuera un problema supervisado, podríamos usar el tipo de crimen ('Crm Cd Desc') como variable objetivo.
- En ese caso, sería una clasificación multiclase.
- Podría haber desbalance entre clases (por ejemplo, ciertos crímenes son mucho más frecuentes).
- Algunas variables como la hora del día, fecha del crimen y localización pueden ser importantes.
- Los datos presentan granularidad por fecha (día, mes, hora) y ubicación (coordenadas).
- Es necesario revisar la calidad de datos geográficos y la consistencia temporal.

Conclusión

A través de este análisis hemos podido:

- Explorar la estructura y distribución de los datos de criminalidad.
- Identificar outliers y revisar valores nulos.
- Visualizar los tipos de crimen más comunes y sus ubicaciones.
- Detectar problemas de calidad y posibles mejoras en el preprocesamiento.

Este análisis sirve como base para desarrollar modelos predictivos o sistemas de visualización de criminalidad en la ciudad.