Hate Crime Prediction

0. Import Modules

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

1. Load Dataset

In [2]:
```
df = pd.read_csv('hate_crime.csv')
df.head()
```

Out[2]:

	incident_id	data_year	ori	pug_agency_name	pub_agency_unit	agency_type_name	state_abbr	state_name	division_
0	43	1991	AR0350100	Pine Bluff	NaN	City	AR	Arkansas	West C
1	44	1991	AR0350100	Pine Bluff	NaN	City	AR	Arkansas	West C
2	45	1991	AR0600300	North Little Rock	NaN	City	AR	Arkansas	West C
3	46	1991	AR0600300	North Little Rock	NaN	City	AR	Arkansas	West C
4	47	1991	AR0670000	Sevier	NaN	County	AR	Arkansas	West C

5 rows × 28 columns

In [3]:

```
# Check columns
df.columns
```

Out[3]:

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

In [4]:
```
df['data_year'].unique()
```

Out[4]:
```
array([1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,
       2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012,
       2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023])
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253776 entries, 0 to 253775
Data columns (total 28 columns):
 #   Column                        Non-Null Count   Dtype  
---  ------                        --------------   -----  
 0   incident_id                   253776 non-null  int64  
 1   data_year                     253776 non-null  int64  
 2   ori                           253776 non-null  object 
 3   pug_agency_name               253776 non-null  object 
 4   pub_agency_unit               7595 non-null    object 
 5   agency_type_name              253776 non-null  object 
 6   state_abbr                    253776 non-null  object 
 7   state_name                    253776 non-null  object 
 8   division_name                 253776 non-null  object 
 9   region_name                   253776 non-null  object 
 10  population_group_code         253109 non-null  object 
 11  population_group_description  253109 non-null  object 
 12  incident_date                 253776 non-null  object 
 13  adult_victim_count            82700 non-null   float64
 14  juvenile_victim_count         80063 non-null   float64
 15  total_offender_count          253776 non-null  int64  
 16  adult_offender_count          73219 non-null   float64
 17  juvenile_offender_count       73212 non-null   float64
 18  offender_race                 253776 non-null  object 
 19  offender_ethnicity            253776 non-null  object 
 20  victim_count                  253776 non-null  int64  
 21  offense_name                  253776 non-null  object 
 22  total_individual_victims      248651 non-null  float64
 23  location_name                 253776 non-null  object 
 24  bias_desc                     253776 non-null  object 
 25  victim_types                  253776 non-null  object 
 26  multiple_offense              253776 non-null  object 
 27  multiple_bias                 253776 non-null  object 
dtypes: float64(5), int64(4), object(19)
memory usage: 54.2+ MB
```

In [6]:
```
# Check shape
df.shape
```

Out[6]: (253776, 28)

In [7]:
```
# Define custom check missing values (columns on)
def check_missing_columns(df):
    index = 0
    for col in df:
        missing_count = df[col].isna().sum()
        if missing_count > 0:
            index += 1
            print(f"{col}: {missing_count}")
    print(f"\nTotal Missing Columns: {index}")
```

In [8]:
```
check_missing_columns(df)
```

```
pub_agency_unit: 246181
population_group_code: 667
population_group_description: 667
adult_victim_count: 171076
juvenile_victim_count: 173713
adult_offender_count: 180557
juvenile_offender_count: 180564
total_individual_victims: 5125

Total Missing Columns: 8
```

In [9]:
```
df.describe()
```

Out[9]:

	incident_id	data_year	adult_victim_count	juvenile_victim_count	total_offender_count	adult_offender_count	juven
count	2.537760e+05	253776.000000	82700.000000	80063.000000	253776.000000	73219.000000	
mean	4.045290e+05	2007.711320	0.749456	0.107216	0.949542	0.623090	
std	5.626399e+05	9.798864	1.089989	0.499702	1.298449	0.808085	
min	2.000000e+00	1991.000000	0.000000	0.000000	0.000000	0.000000	
25%	6.347575e+04	1999.000000	0.000000	0.000000	0.000000	0.000000	
50%	1.269305e+05	2007.000000	1.000000	0.000000	1.000000	1.000000	
75%	1.945972e+05	2017.000000	1.000000	0.000000	1.000000	1.000000	
max	1.522894e+06	2023.000000	146.000000	60.000000	99.000000	60.000000	

In [10]:
```
df.columns
```

Out[10]:
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

2. Explonatory Data Analysis (EDA)

Check data_year

In [11]:
```
df['data_year'].dtype
```

Out[11]:
```
dtype('int64')
```

In [12]:
```
df['data_year'].unique()
```

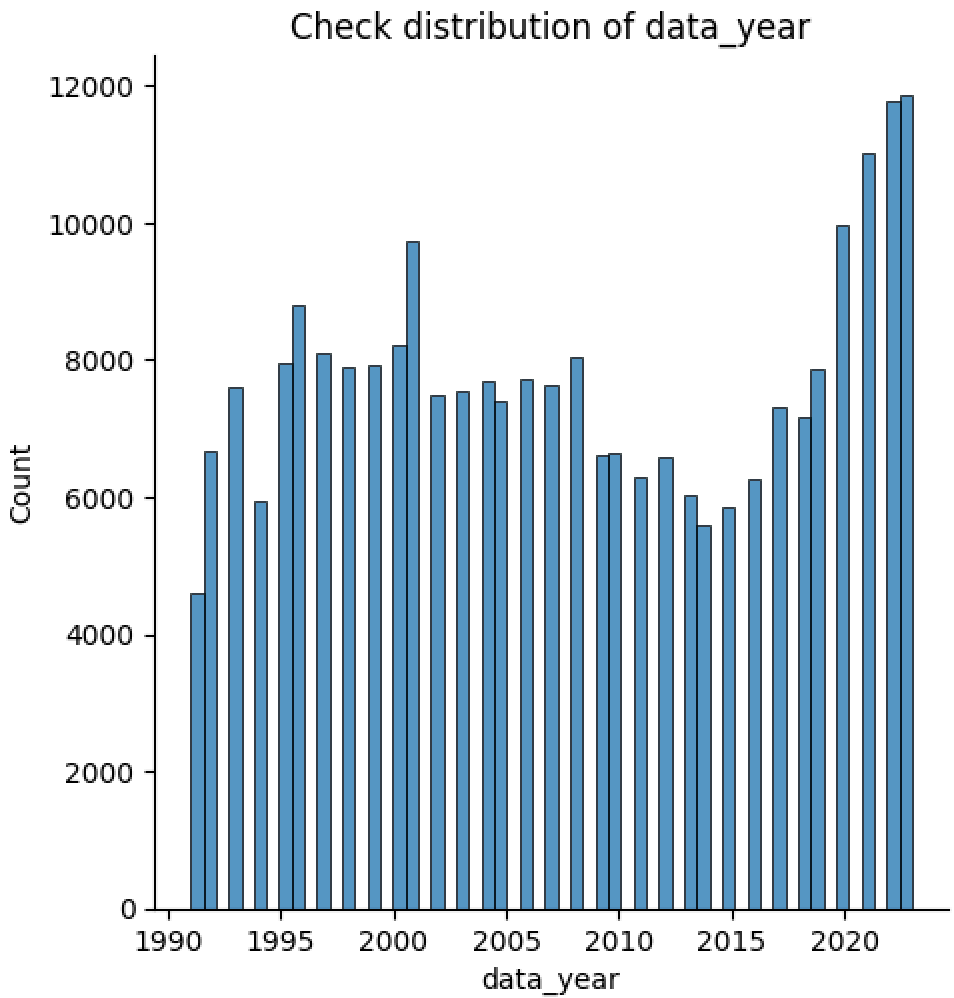
Out[12]:
```
array([1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,
       2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012,
       2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023])
```

In [13]:
```
df['data_year'].isna().sum()
```

Out[13]:
```
np.int64(0)
```

In [14]:
```
# Check distribution
plt.figure(figsize=(12,7))

sns.displot(df, x='data_year')

plt.title('Check distribution of data_year')
plt.show()

print('Likely skewed to the left: Modern years have higher chance for crime')
```

```
<Figure size 1200x700 with 0 Axes>
```



Likely skewed to the left: Modern years have higher chance for crime

In [15]:
```
# Check skewness
from scipy.stats import skew

skew_value = skew(df['data_year'])
print('-'*30)
print(f"Skewness of data_year: {round(skew_value, 4)}")
print('-'*30)
print('The skewness is being around 0.0244 which is very closed to zero')
print('This suggests that the distribution is highly normal')
```

```
------------------------------
Skewness of data_year: 0.0244
------------------------------
The skewness is being around 0.0244 which is very closed to zero
This suggests that the distribution is highly normal
```

In [16]:
```python
df.columns
```

Out[16]:
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

Check ori and state

In [17]:
```python
df['ori'].dtype
```

Out[17]:
```
dtype('O')
```

In [18]:
```python
df['ori'].unique()
```

Out[18]:
```
array(['AR0350100', 'AR0600300', 'AR0670000', ..., 'WV0440000',
       'WVWSP1100', 'WY0060300'], shape=(10710,), dtype=object)
```

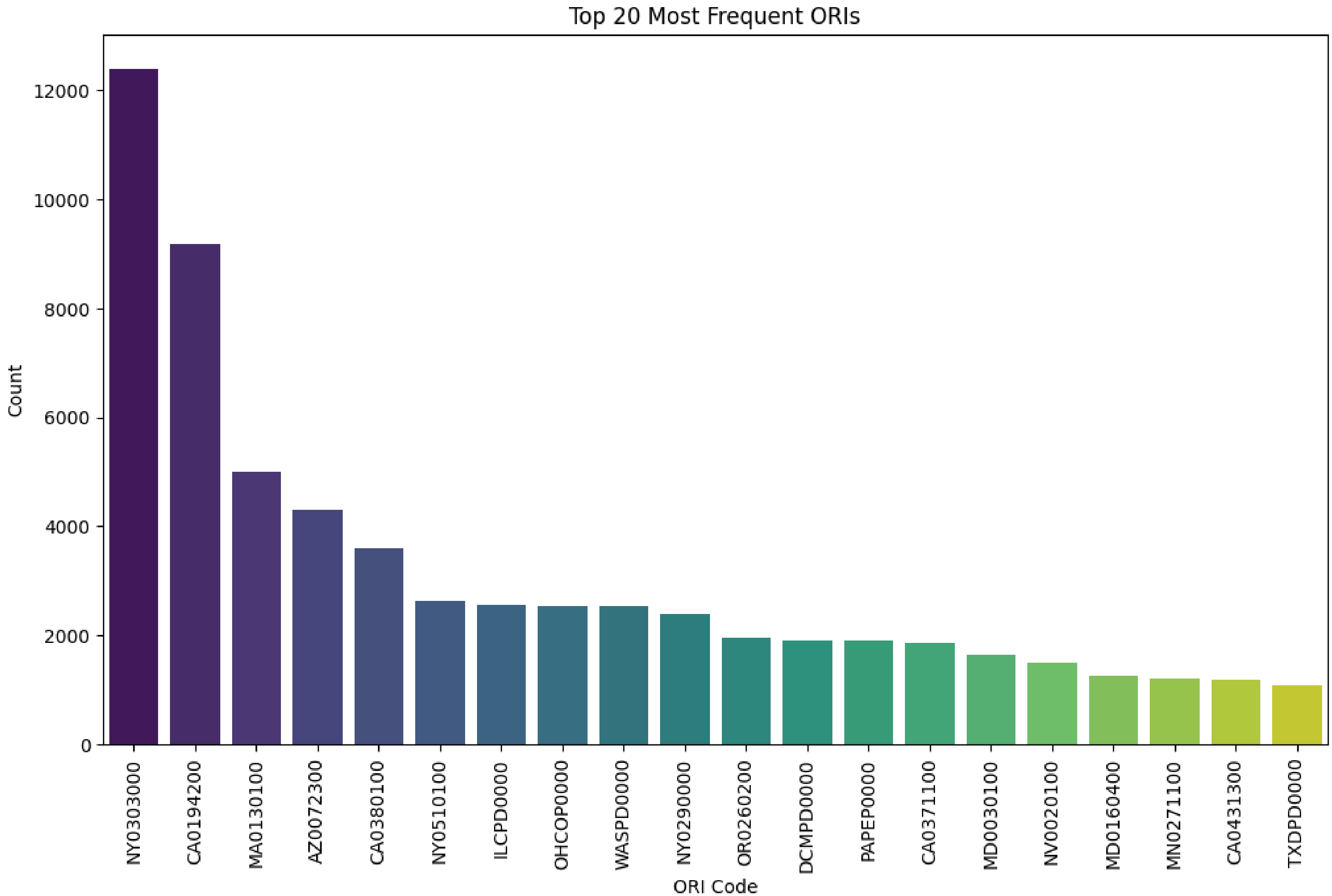
In [19]:
```python
df['ori'].isna().sum()
```

Out[19]:
```
np.int64(0)
```

In [196…
```python
# Check distribution
# Count occurrences of each ORI
top_ori = df['ori'].value_counts().head(20)  # Get top 20 most common ORI

# Plot bar chart
plt.figure(figsize=(12, 7))
sns.barplot(x=top_ori.index, y=top_ori.values, palette="viridis")
```

```
plt.xticks(rotation=90)
plt.xlabel("ORI Code")
plt.ylabel("Count")
plt.title("Top 20 Most Frequent ORIs")
plt.show()
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\3816221670.py:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(x=top_ori.index, y=top_ori.values, palette="viridis")
```

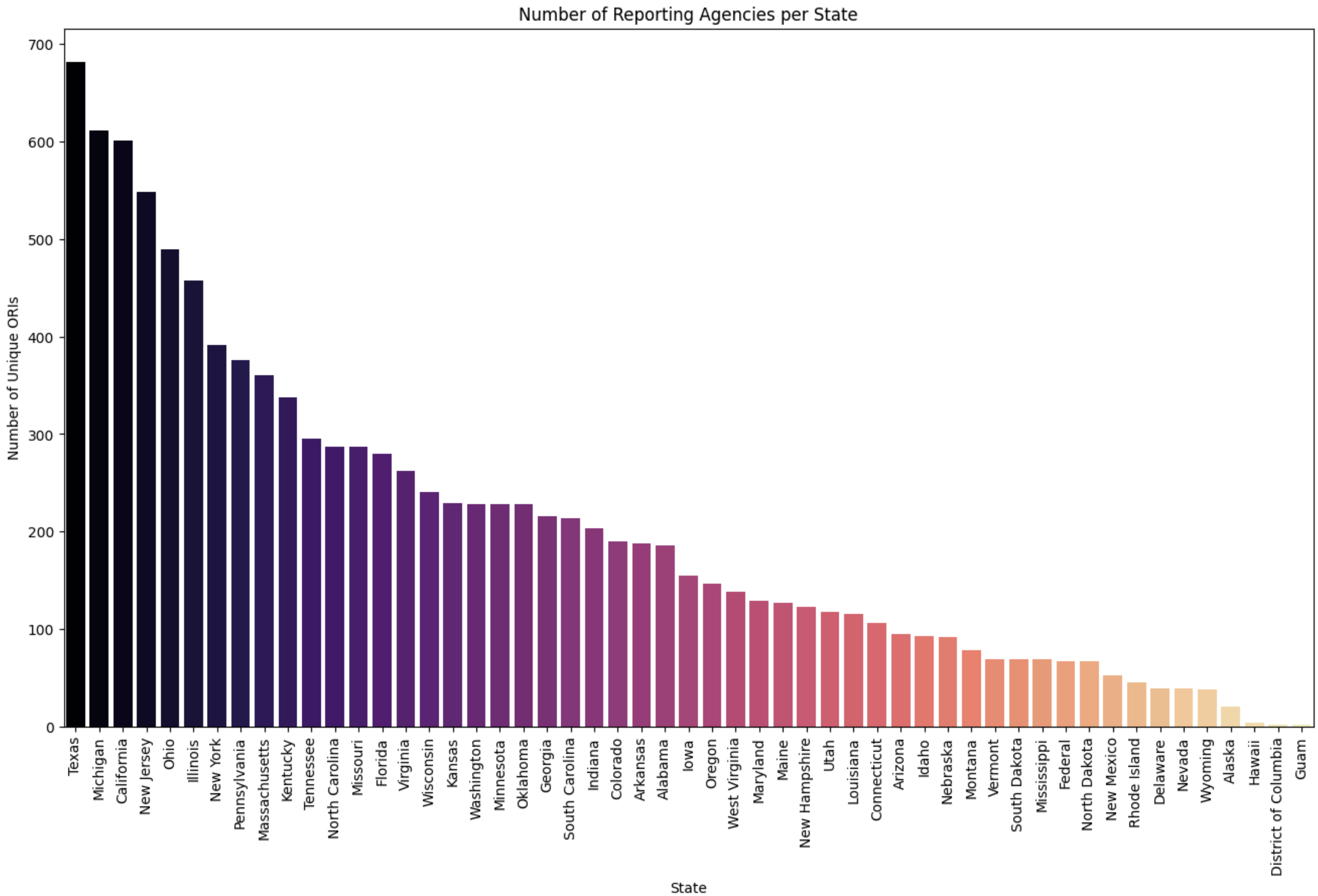


In [150…

```
# Try grouping ori by state
state_counts = df.groupby("state_name")["ori"].nunique().sort_values(ascending=False)

plt.figure(figsize=(16, 9))
sns.barplot(x=state_counts.index, y=state_counts.values, palette="magma")
plt.xticks(rotation=90)
plt.xlabel("State")
```

```
plt.ylabel("Number of Unique ORIs")
plt.title("Number of Reporting Agencies per State")
plt.show()

print('Larger states with more cities/towns tend to have more ORIs')
print('States with more ORIs tend to have more granular crime reporting, but this does not mean they have the most cr
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\3591222749.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `h
ue` and set `legend=False` for the same effect.

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



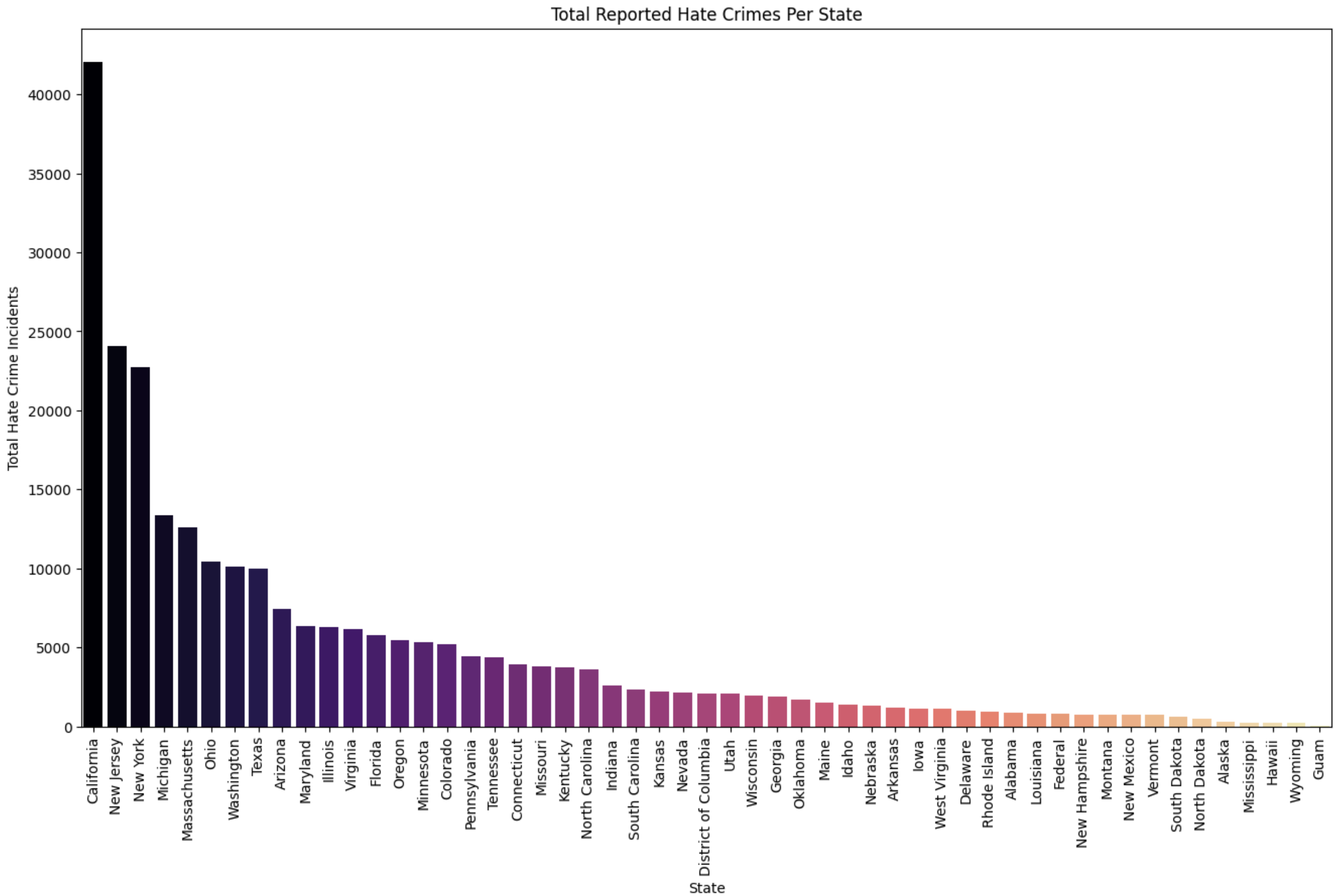
Larger states with more cities/towns tend to have more ORIs
States with more ORIs tend to have more granular crime reporting, but this does not mean they have the most crimes.

In [151…

```
# Check total hate crime per state grouped by total incident
state_crime_counts = df.groupby("state_name")["incident_id"].count().sort_values(ascending=False)

# Plot
```

```
plt.figure(figsize=(16, 9))
sns.barplot(x=state_crime_counts.index, y=state_crime_counts.values, palette="magma")
plt.xticks(rotation=90)
plt.xlabel("State")
plt.ylabel("Total Hate Crime Incidents")
plt.title("Total Reported Hate Crimes Per State")
plt.show()

print('California has the most crime detected.')
print('California is the most populous state in the U.S., so it might have more reported crimes overall.')
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\3665486666.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `h
ue` and set `legend=False` for the same effect.

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




```
California has the most crime detected.
California is the most populous state in the U.S., so it might have more reported crimes overall.
```

In [23]:
```
# Count crime types per state
crime_types = df.groupby(["state_name", "offense_name"])["incident_id"].count().reset_index()

# Get top crime types
top_crime_types = crime_types.sort_values(by="incident_id", ascending=False).groupby("state_name").head(1)
```

```
print('Most common hate crime type in California → "Destruction/Damage/Vandalism of Property" (12,598 cases!)')
print('Destruction/Vandalism is dominant in multiple states (California, New York, Maryland, Texas, Virginia, etc.).'
top_crime_types  # Show the most common hate crime type in each state
```

```
Most common hate crime type in California → "Destruction/Damage/Vandalism of Property" (12,598 cases!)
Destruction/Vandalism is dominant in multiple states (California, New York, Maryland, Texas, Virginia, etc.).
```

Out[23]:

	state_name	offense_name	incident_id
239	California	Destruction/Damage/Vandalism of Property	12598
1772	New Jersey	Intimidation	12475
1856	New York	Destruction/Damage/Vandalism of Property	9690
1380	Michigan	Intimidation	4349
1267	Massachusetts	Intimidation	4192
1166	Maryland	Destruction/Damage/Vandalism of Property	3634
2098	Ohio	Intimidation	3619
2927	Washington	Intimidation	3466
2571	Texas	Destruction/Damage/Vandalism of Property	2508
2791	Virginia	Destruction/Damage/Vandalism of Property	2444
106	Arizona	Intimidation	2165
561	Florida	Destruction/Damage/Vandalism of Property	1796
1454	Minnesota	Intimidation	1774
771	Illinois	Simple Assault	1770
358	Colorado	Intimidation	1710
2286	Pennsylvania	Intimidation	1689
2227	Oregon	Intimidation	1619
428	Connecticut	Intimidation	1382
1021	Kentucky	Intimidation	1127
1950	North Carolina	Intimidation	1075
2503	Tennessee	Intimidation	1072
1547	Missouri	Intimidation	1052

	state_name	offense_name	incident_id
816	Indiana	Intimidation	893
504	District of Columbia	Simple Assault	848
1134	Maine	Intimidation	633
2658	Utah	Destruction/Damage/Vandalism of Property	601
614	Georgia	Intimidation	598
939	Kansas	Intimidation	592
2167	Oklahoma	Intimidation	577
1685	Nevada	Destruction/Damage/Vandalism of Property	567
2363	South Carolina	Destruction/Damage/Vandalism of Property	496
3054	Wisconsin	Destruction/Damage/Vandalism of Property	474
1627	Nebraska	Destruction/Damage/Vandalism of Property	424
470	Delaware	Destruction/Damage/Vandalism of Property	423
705	Idaho	Simple Assault	368
861	Iowa	Destruction/Damage/Vandalism of Property	343
527	Federal	Intimidation	329
2316	Rhode Island	Destruction/Damage/Vandalism of Property	328
1732	New Hampshire	Destruction/Damage/Vandalism of Property	296
2725	Vermont	Destruction/Damage/Vandalism of Property	287
3025	West Virginia	Simple Assault	252
170	Arkansas	Simple Assault	250
2439	South Dakota	Simple Assault	248
1817	New Mexico	Simple Assault	233

	state_name	offense_name	incident_id
40	Alabama	Simple Assault	228
1107	Louisiana	Simple Assault	219
1587	Montana	Destruction/Damage/Vandalism of Property	200
2011	North Dakota	Simple Assault	147
656	Hawaii	Intimidation	128
3105	Wyoming	Simple Assault	79
48	Alaska	Aggravated Assault	75
1500	Mississippi	Simple Assault	69
641	Guam	Destruction/Damage/Vandalism of Property	7

In [24]:
```
df.columns
```

Out[24]:
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

Check pug_agency_name

In [25]:
```
df['pug_agency_name'].dtype
```

Out[25]:
```
dtype('O')
```

In [26]:
```
df['pug_agency_name'].unique()
```

Out[26]:
```
array(['Pine Bluff', 'North Little Rock', 'Sevier', ...,
       'Ohio Valley Drug and Violent Crime Task Force',
       'Berkeley Springs', 'Moorcroft'], shape=(7110,), dtype=object)
```

In [27]:
```
df['pug_agency_name'].isna().sum()
```

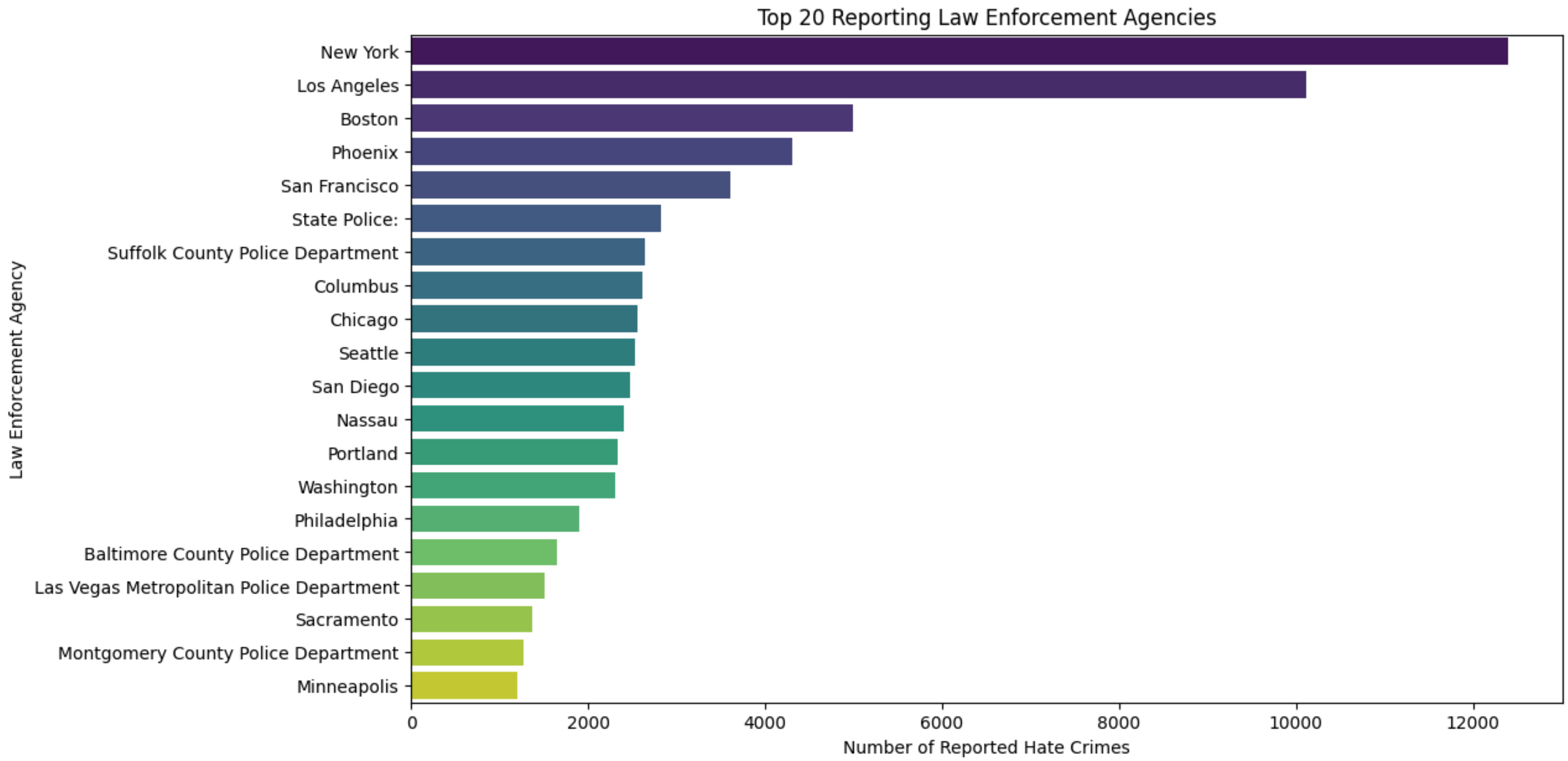
Out[27]:
```
np.int64(0)
```

In [28]:
```
df['pug_agency_name'].head()
```

Out[28]:
```
0           Pine Bluff
1           Pine Bluff
2    North Little Rock
3    North Little Rock
4               Sevier
Name: pug_agency_name, dtype: object
```

In [197…
```
# Check distribution
# Check the distribution of 'pug_agency_name' (Law Enforcement Agency Names)

# Count occurrences of each agency
agency_counts = df["pug_agency_name"].value_counts().head(20)  # Top 20 most common agencies

# Plot the distribution
plt.figure(figsize=(12, 7))
sns.barplot(x=agency_counts.values, y=agency_counts.index, palette="viridis")
plt.xlabel("Number of Reported Hate Crimes")
plt.ylabel("Law Enforcement Agency")
plt.title("Top 20 Reporting Law Enforcement Agencies")
plt.show()

print('New York has the top number of reporting hate crime via law enforement agencies')
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\3350046672.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(x=agency_counts.values, y=agency_counts.index, palette="viridis")
```



New York has the top number of reporting hate crime via law enforement agencies

In [31]:
```
df.columns
```

Out[31]:
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

Check pub_agency_unit

In [32]:
```
df['pub_agency_unit'].dtype
```
Out[32]:
```
dtype('O')
```
In [35]:
```
df['pub_agency_unit'].unique()
```

Out[35]:
```
array([nan, 'Boulder', 'Urbana', 'Montgomery County', 'College Park',
       'Anne Arundel County', 'Carroll County', 'Cecil County',
       'Charles County', 'Frederick County', 'Harford County',
       "Queen Anne's County", 'Wicomico County', 'Worcester County',
       'Baltimore County', "Prince George's County", 'Twin Cities',
       'Albany', 'Binghamton', 'Cortland', 'Buffalo State College',
       'Morrisville', 'Old Westbury', 'Oswego', 'Potsdam', 'Stony Brook',
       'New Paltz', 'Purchase', 'Dutchess County', 'Orange County',
       'Oswego County', 'Sullivan County', 'Ulster County',
       'Wayne County', 'Delhi', 'Geneseo', 'Polytechnic Institute',
       'Columbus', 'Crook County', 'Curry County', 'Jackson County',
       'Lincoln County', 'Linn County', 'Marion County', 'Chester County',
       'Lancaster County', 'Monroe County', 'Washington County',
       'Westmoreland County', 'Blair County', 'Smyth County',
       'Wythe County', 'Milwaukee', 'East Bay', 'Fullerton',
       'Kent County', 'New Castle County', 'Sussex County', 'Tampa',
       'Tallahassee', 'Chicago', 'Calvert County', 'Caroline County',
       'Orono', 'Ann Arbor', 'Chapel Hill', 'Atlantic County',
       'Cape May County', 'Newark', 'Hunterdon County', 'Monmouth County',
       'Morris County', 'Salem County', 'Warren County', 'Camden County',
       'Cumberland County', 'Middlesex County', 'Alfred', 'Buffalo',
       'Oneonta', 'Farmingdale', 'Broome County', 'Columbia County (T)',
       'Niagara County', 'Oneida County', 'Oneida County (T)',
       'St. Lawrence County', 'Saratoga County', 'Westchester County',
       'Westchester County (T)', 'Fredonia', 'Brockport', 'Canton',
       'Norman', 'Lane County', 'Multnomah County', 'Benton County',
       'Skippack', 'Killeen', 'Downtown Campus', 'Klein', 'San Marcos',
       'Lubbock', 'Midland', 'Health and Sciences Center', 'Austin',
       'Amarillo', 'Pullman', 'Eau Claire', 'Fayetteville',
       'All Campuses', 'Zone 14A', 'District 3', 'District 4', 'Zone 4',
       'District 12', 'Carbondale', 'District 14', 'Edwardsville',
       'District 11', 'Indianapolis', 'Bloomington', 'Essex County',
       'Stony Creek Metropark', 'Burlington County', 'Mercer County',
       'Ocean County', 'Passaic County', 'Somerset County',
       'Gloucester County', 'Cobleskill', 'Chenango County',
       'Delaware County', 'Otsego County', 'Plattsburgh',
       'Klamath County', 'Tillamook County', 'Altoona', 'Butler County',
       'University Park', 'Elizabethville', 'Franklin County',
       'Indiana County', 'Lawrence County', 'Lehigh County',
       'Lycoming County', 'Pike County', 'Hope Valley', 'San Antonio',
       'Beaumont', 'Conroe', 'Commerce', 'Eastern', 'St. Albans', 'West',
       'Main Campus', 'Allegany County', 'Columbia', 'Reno',
```

```
'Herkimer County', 'Onondaga County', 'Berks County',
'Clearfield County', 'Crawford County', 'Fayette County',
'Fulton County', 'Perry County', 'Knoxville',
'Health Science Center', 'Spring Branch', 'Hanover County',
'Salem', 'Rockbridge County', 'Powhatan County',
'Spotsylvania County', 'St. Johnsbury', 'Rutland', 'Derby',
'Shaftsbury', 'Brattleboro', 'Berlin', 'Alameda County',
'Northridge', 'Los Angeles', 'Irvine', 'San Bernardino',
'San Diego', 'San Luis Obispo', 'Oceano Dunes', 'San Jose',
'San Francisco County', 'San Francisco', 'Pomona',
'Contra Costa County', 'Colorado Springs', 'Health Center',
'Southeast', 'Grant County', 'Letcher County', 'Mason County',
'Union County', 'Amherst', 'Suffolk County', "St. Mary's County",
'Farmington', 'Livingston County', 'Marquette County',
'Ogemaw County', 'Saginaw County', 'Van Buren County',
'Houghton County', 'Charlotte', 'Tompkins County',
'Deschutes County', 'Cambria County', 'Jefferson County', 'Judson',
'Spring', 'Tarrant County', 'Accomack County', 'Richmond County',
'Royalton', 'Platteville', 'Oshkosh', 'Berkeley',
'North Coast Redwoods', 'Dominguez Hills', 'Long Beach',
'Riverside', 'San Diego Coast', 'Davis', 'Northern Buttes',
'Fresno', 'Hancock County', 'Ohio County', 'Edmonson County',
'Floyd County', 'Rockcastle County', 'Scott County',
'Norfolk County', 'Medical Center, Worcester',
'Harbor Campus, Boston', 'Osceola County', 'Genesee County',
'Newaygo County', 'Harrisburg', 'East Central', 'College Station',
'Central Campus', 'Goochland County', 'Pittsylvania County',
'San Diego County', 'Sacramento', 'Santa Cruz', 'Fort Collins',
'Dartmouth', 'Talbot County', 'Flint', 'Greene County', 'Okmulgee',
'Tulsa', 'Chepachet', 'Martin', 'Corpus Christi', 'Arlington',
'Tazewell County', 'Fairfax County', 'New Haven', 'Parkside',
'Humboldt', 'Bakersfield', 'Medical Center, Sacramento',
'Bristol County', 'Baltimore City', 'Bay County',
'Chippewa County', 'Iosco County', 'Roscommon County',
'Washtenaw County', 'Clinton County', 'Ionia County',
'Oakland County', 'Adams County', 'Beaver County', 'El Paso',
'Pasadena', 'Logan', 'Arlington County', 'Santa Cruz ',
'Santa Barbara', 'Chico', 'Adair County', 'Hopkins County',
'Muhlenberg County', 'Plymouth County', 'Branch County',
'Cheboygan County', 'Gogebic County', 'Montcalm County', 'Duluth',
'Erie County', 'Albany County', 'Pittsburgh', 'Bucks County',
'Columbia County', 'Chattanooga', 'Waco', 'Pan American',
```

```
'Health Science Center, San Antonio', 'Clarke County',
'Rockingham', 'Huntington', 'Moorefield', 'Daviess County',
'McLean County', 'Madison County', 'Marshall County',
'Henderson County', 'Baton Rouge', 'Barry County', 'Lapeer County',
'Oxford', 'Cattaraugus County', 'Putnam County', 'Rockland County',
'Clatsop County', 'Armstrong County', 'Potter County',
'Bicentennial Capitol Mall', 'Clearlake', 'Denton',
'Southampton County', 'Bradford', 'Elizabeth', 'Marlinton',
'Berkeley County', 'Wyoming County', 'Health Sciences Center',
'Allen County', 'Wolfe County', 'Medford', 'Howard County',
'Grand Traverse County', 'Luce County', 'Eaton County',
'Gladwin County', 'Raleigh', 'Greensboro', 'Douglas County',
'Wasco County', '3rd Judicial District', 'Madison', 'Whitewater',
'Parsons', 'Moundsville', 'Upperglade', 'Carter County',
'Oldham County', 'Monroe', 'Shiawassee County', 'Dearborn',
'Clare County', 'Asheville', 'New York County', 'Schoharie County',
'Clackamas County', 'Josephine County', 'Superior', 'Rainelle',
'Orange Coast', 'Evanston', 'Lyon County', 'Kennebec County',
'Calhoun County', 'Kansas City', 'Maritime ', 'Juniata County',
'23rd Judicial District', 'Webster County', 'St. Petersburg',
'Crittenden County', 'Johnson County', 'Berkshire County',
'Hampden County', 'Muskegon County', 'Wexford County',
'St. Clair County', 'Wilmington', 'Ontario County', 'Berks',
'Centre County', 'Northampton County', 'Pickwick Landing',
'Dallas', 'Jonesboro', 'Monticello', 'Hastings College of Law',
'Monterey Bay', 'Troop B', 'Headquarters', 'Saratoga County (T)',
'Pymatuning', 'York County', 'Lexington County', 'Newberry County',
'Alvin', 'Tyler', 'Law Enforcement Division', 'Vancouver',
'La Crosse', 'Stanislaus', 'Aroostook County', 'Waldo County',
'St. Louis', 'Lincoln', 'Dillon County', 'Upstate', 'Green Bay',
'Gold Fields District', 'San Bernardino County', 'Auburn-Washburn',
'Isabella County', 'Kalamazoo County', 'Berrien County', 'Keyser',
'Romney', 'Medical Sciences', 'Medical Center',
'Kensington Metropark', 'Rensselaer County', 'Tioga County',
'Tahlequah', 'Schuylkill County', 'Dauphin County',
'San Luis Obispo ', 'Springfield', 'Maize', 'Goddard',
'Campbellsburg', 'Harlan', 'Madisonville', 'Pikeville', 'London',
'Wooster', 'Pine Bluff', 'Channel Islands', 'Pueblo', 'Mayfield',
'Bowling Green', 'Morehead', 'Dry Ridge', 'Androscoggin County',
'St. Joseph', 'Finger Lakes Region', 'Steuben County',
'Elk County', 'Permian Basin', 'Fauquier County', 'Morgantown',
'Frankfort', 'Orleans County', 'Anderson County', 'Little Rock',
```

```
'San Mateo County', 'Kewanee', 'Richmond', 'Ashland',
'Huron County', 'Institute of Technology', 'Plumas County',
'Tehachapi District', 'Henry County', 'Knox County', 'Lake County',
'Henderson', 'Cass County', 'Lower Huron Metropark', 'Morris',
'Long Island Region', 'Nassau County', 'Humble', 'Buchanan County',
'Rockingham County', 'Decatur County', 'Vanderburgh County',
'Hazard', 'Cannabis Suppression Section', 'Gratiot County',
'New York City Region', 'Huntingdon County', 'Greenwood County',
'Internal Affairs', 'Pflugerville', 'Halifax County', 'Stout',
'Miami County', 'Ripley County', 'Rush County', 'Jefferson City',
'Meramec', 'Pembroke', 'Johnstown', 'Dallas County', 'Houston',
'West Drug Enforcement Branch', 'Suffolk', 'Allegan County',
'Ingham County', 'Manistee County', 'Mecosta County',
'Oceana County', 'Downstate Medical ', 'Chemung County',
'Scituate', 'Sweetwater', 'Surry County', 'Chesapeake',
'Newport News', 'Enforcement Division', 'Marin County',
'La Porte County', 'Posey County', 'Troop E', 'Elizabethtown',
'Alpena County', 'St. Joseph County', 'Macomb County',
'Ottawa County', 'Grafton County', 'New Brunswick', 'Lewis County',
'Hamilton County', 'Brandywine', 'Rio Grande Valley',
'Amherst County', 'Winchester', 'Harrisonburg', 'Phoenix',
'Cincinnati', 'Albuquerque', 'Baltimore', 'Boston', 'Birmingham',
'Cleveland', 'Louisville', 'Memphis', 'Mobile', 'New Orleans',
'Omaha', 'Philadelphia', 'Portland', 'Seattle', 'Washington',
'Atlanta', 'Miami', 'Salt Lake City', 'Oklahoma City', 'Detroit',
'Office of Special Investigations', 'Spencer County', 'Troop A',
'Delta County', 'Schoolcraft County', 'Alger County',
'Hillsdale County', 'Arenac County', 'Central Region',
'Eastview Precinct', 'Cayuga County', 'Seneca County',
'Snyder County', 'Shenandoah County', 'Parkersburg', 'Anchorage',
'Denver', 'Jacksonville', 'Las Vegas', 'Minneapolis', 'New York',
'Norfolk', 'Elkhart County', 'Wabash County', 'Whitley County',
'Troop C', 'Ontonagon County', 'Mackinac County', 'Sanilac County',
'St. Paul', 'Camden', 'Schenectady County', 'Malheur County',
'Bradford County', 'Venango County', 'Hutto', 'Bay City',
'Brazosport', 'Prince William County', 'Dinwiddie County',
'Russell County', 'Staunton', 'Alleghany County',
'Division of Enforcement and Licensing', 'Princeton', 'Hamlin',
'Bridgeport', 'Wayne', 'Grantsville', 'Bay Area', 'Jackson',
'Honolulu', 'San Juan', 'South Bend', 'Baraga County',
'Midland County', 'Emmet County', 'Benzie County',
'Tuscola County', 'Hudson County', 'Southern Division',
```

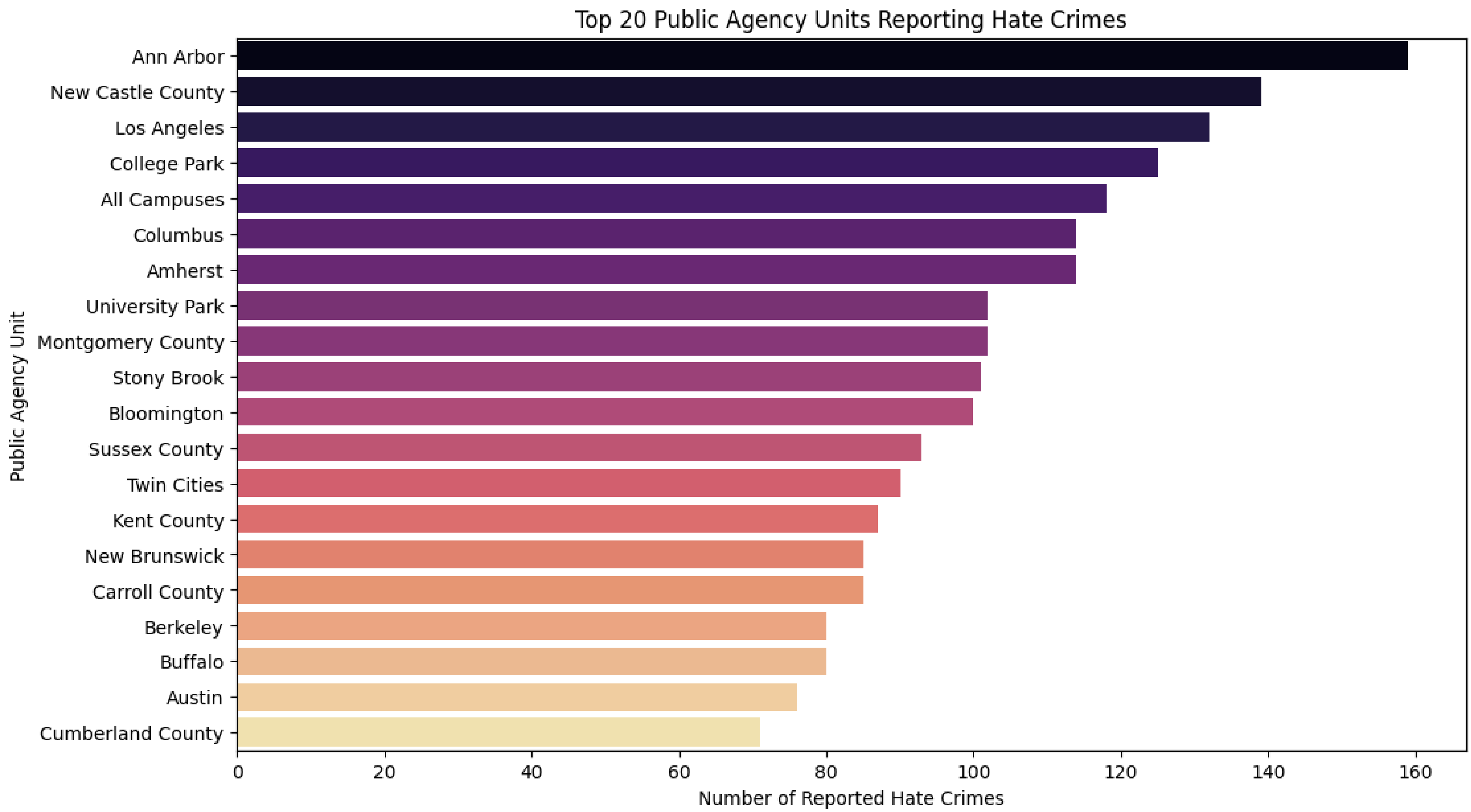
```
       'Chautauqua County', 'Chesterfield County', 'Colleton County',
       'Brunswick County', 'Roanoke', 'James City County',
       'Grayson County', 'Westminster', 'Williston', 'Beckley',
       'Berkeley Springs', 'Clay', 'Union', 'South Charleston',
       'Buckhannon', 'Elkins', 'Fire Investigation Division', 'Troop F',
       'Lowell', 'Environmental Police',
       'Internal Investigation Division', 'Prince Georges County',
       'Lenawee County', 'Cliffs of the Neuse', 'Northwestern Division',
       'Statewide', 'Genesee Region', 'Coos County', 'Allegheny County',
       'Tionesta', 'Bedford County', 'Philadelphia County',
       'Luzerne County', 'Northumberland County', 'York', 'Beaver',
       'Lackawanna County', 'Beaufort', 'Precinct 3', 'Anna',
       'Brownsville', 'Precinct 1', 'Burkburnett', 'Fort Bend',
       'Castleberry', 'Internet Crimes Against Children Unit',
       'Capitol Protection Section', 'Bishop Area Office', 'Troop G',
       'Health Sciences Center, Shreveport', 'Garrett County',
       "Queen Annes's County", 'Dorchester County', 'Leelanau County',
       'Kalkaska County', 'Hall County', 'Cheshire County',
       'Northeastern Division', 'Gilliam County', 'Clarion County',
       'Lebanon County', 'Montour County', 'Hazleton', 'Abington',
       'Susquehanna County', 'Round Rock', 'Mansfield', 'Katy', 'Hampton',
       'Culpeper County', 'Stevens Point', 'Martinsburg', 'Fairbanks',
       'Bakersfield ', 'Temecula Area Office', 'Marin Area Office',
       'Missaukee County', 'Kearney', 'Niagara Region', 'Allegany Region',
       'Umatilla County', 'Carbon County', 'Cameron County',
       'Kershaw County', 'Royal', 'Bastrop', 'Community',
       'Coldspring-Oakhurst', 'Eagle Mountain-Saginaw', 'Charlottesville',
       'Harrisville', 'Spencer', 'Grafton', 'Williamson', 'Fairmont'],
      dtype=object)
```

In [41]:
```
agency_unit_counts = df["pub_agency_unit"].value_counts().head(20)  # Top 20 most common agency units
agency_unit_counts
```

Out[41]:
```
pub_agency_unit
Ann Arbor              159
New Castle County      139
Los Angeles            132
College Park           125
All Campuses           118
Columbus               114
Amherst                114
University Park        102
Montgomery County      102
Stony Brook            101
Bloomington            100
Sussex County           93
Twin Cities             90
Kent County             87
New Brunswick           85
Carroll County          85
Berkeley                80
Buffalo                 80
Austin                  76
Cumberland County       71
Name: count, dtype: int64
```

In [198…
```
# Check distribution
plt.figure(figsize=(12, 7))
sns.barplot(x=agency_unit_counts.values, y=agency_unit_counts.index, palette="magma")
plt.xlabel("Number of Reported Hate Crimes")
plt.ylabel("Public Agency Unit")
plt.title("Top 20 Public Agency Units Reporting Hate Crimes")
plt.show()

print('Top reporting public agencies include:\n1. Ann Arbor (highest reported hate crimes!) → Diverse Student Populat
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\1034682908.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `h
ue` and set `legend=False` for the same effect.

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




```
Top reporting public agencies include:
1. Ann Arbor (highest reported hate crimes!) → Diverse Student Population → Higher diversity sometimes leads to more reported bias-motivated incidents.  Strong Local Policies → Ann Arbor is known for progressive policies and a well-documented crime reporting system.
2. New Castle County & Los Angeles → Large urban areas with higher reported incidents.
3. College Towns (e.g., College Park, University Park, Stony Brook, Bloomington) → Many universities are in this list, meaning hate crimes in academic settings might be a significant factor.
4. Counties & Urban Centers → Many counties like Montgomery, Carroll, Kent, and Sussex also appear, showing that hate crimes are not just in major cities.
```

Count the most common crime types reported by agencies unit

In [47]:
```python
# Get top 20 public agency units
top_agency_units = df["pub_agency_unit"].value_counts().head(20).index
```

```
# Filter dataset to include only these top agencies
filtered_df = df[df["pub_agency_unit"].isin(top_agency_units)]

# Count the most common crime types reported by these agencies
crime_by_agency = filtered_df.groupby(["pub_agency_unit", "offense_name"])["incident_id"].count().reset_index()

# Get top crime types for each agency
top_crime_per_agency = crime_by_agency.sort_values(by="incident_id", ascending=False).groupby("pub_agency_unit").heac

# Display the result
top_crime_per_agency
```

Out[47]:

	pub_agency_unit	offense_name	incident_id
17	Ann Arbor	Destruction/Damage/Vandalism of Property	96
63	College Park	Destruction/Damage/Vandalism of Property	90
117	Montgomery County	Destruction/Damage/Vandalism of Property	90
11	Amherst	Destruction/Damage/Vandalism of Property	89
1	All Campuses	Destruction/Damage/Vandalism of Property	68
147	Stony Brook	Intimidation	65
48	Buffalo	Destruction/Damage/Vandalism of Property	62
53	Carroll County	Destruction/Damage/Vandalism of Property	62
159	Twin Cities	Destruction/Damage/Vandalism of Property	62
125	New Brunswick	Intimidation	53
133	New Castle County	Destruction/Damage/Vandalism of Property	49
110	Los Angeles	Intimidation	48
44	Bloomington	Intimidation	45
74	Columbus	Destruction/Damage/Vandalism of Property	44
32	Berkeley	Destruction/Damage/Vandalism of Property	43
25	Austin	Destruction/Damage/Vandalism of Property	38
96	Kent County	Destruction/Damage/Vandalism of Property	37
153	Sussex County	Destruction/Damage/Vandalism of Property	37
168	University Park	Intimidation	36
87	Cumberland County	Destruction/Damage/Vandalism of Property	35

Count the top bias moltivation reported by agencies unit

In [49]:
```
# Count bias motivations (e.g., race, religion, LGBTQ+) per agency
bias_by_agency = filtered_df.groupby(["pub_agency_unit", "bias_desc"])["incident_id"].count().reset_index()

# Get top bias motivation for each agency
top_bias_per_agency = bias_by_agency.sort_values(by="incident_id", ascending=False).groupby("pub_agency_unit").head(

# Display the result
print('Most of bias moltivations are racist like Anti-Black and Anti-Jewish')
top_bias_per_agency
```

```
Most of bias moltivations are racist like Anti-Black and Anti-Jewish
```

Out[49]:

	pub_agency_unit	bias_desc	incident_id
201	New Castle County	Anti-Black or African American	75
32	Ann Arbor	Anti-Black or African American	56
227	Sussex County	Anti-Black or African American	52
150	Kent County	Anti-Black or African American	51
178	Montgomery County	Anti-Black or African American	51
257	University Park	Anti-Black or African American	47
96	Carroll County	Anti-Black or African American	44
139	Cumberland County	Anti-Black or African American	41
216	Stony Brook	Anti-Black or African American	37
16	Amherst	Anti-Black or African American	36
2	All Campuses	Anti-Black or African American	36
46	Austin	Anti-Black or African American	35
120	Columbus	Anti-Black or African American	34
110	College Park	Anti-Black or African American	33
161	Los Angeles	Anti-Black or African American	31
76	Bloomington	Anti-Jewish	30
245	Twin Cities	Anti-Black or African American	28
194	New Brunswick	Anti-Jewish	26
88	Buffalo	Anti-Jewish	25
65	Berkeley	Anti-Jewish	21

In [50]:

```
df.columns
```

Out[50]:
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

Check agency_type_name

In [51]:
```
df['agency_type_name'].dtype
```

Out[51]:
```
dtype('O')
```

In [52]:
```
df['agency_type_name'].value_counts()
```

Out[52]:
```
agency_type_name
City                        202135
County                       36551
University or College         8456
State Police                  3421
Other                         1759
Federal                        823
Other State Agency             504
Tribal                         127
Name: count, dtype: int64
```

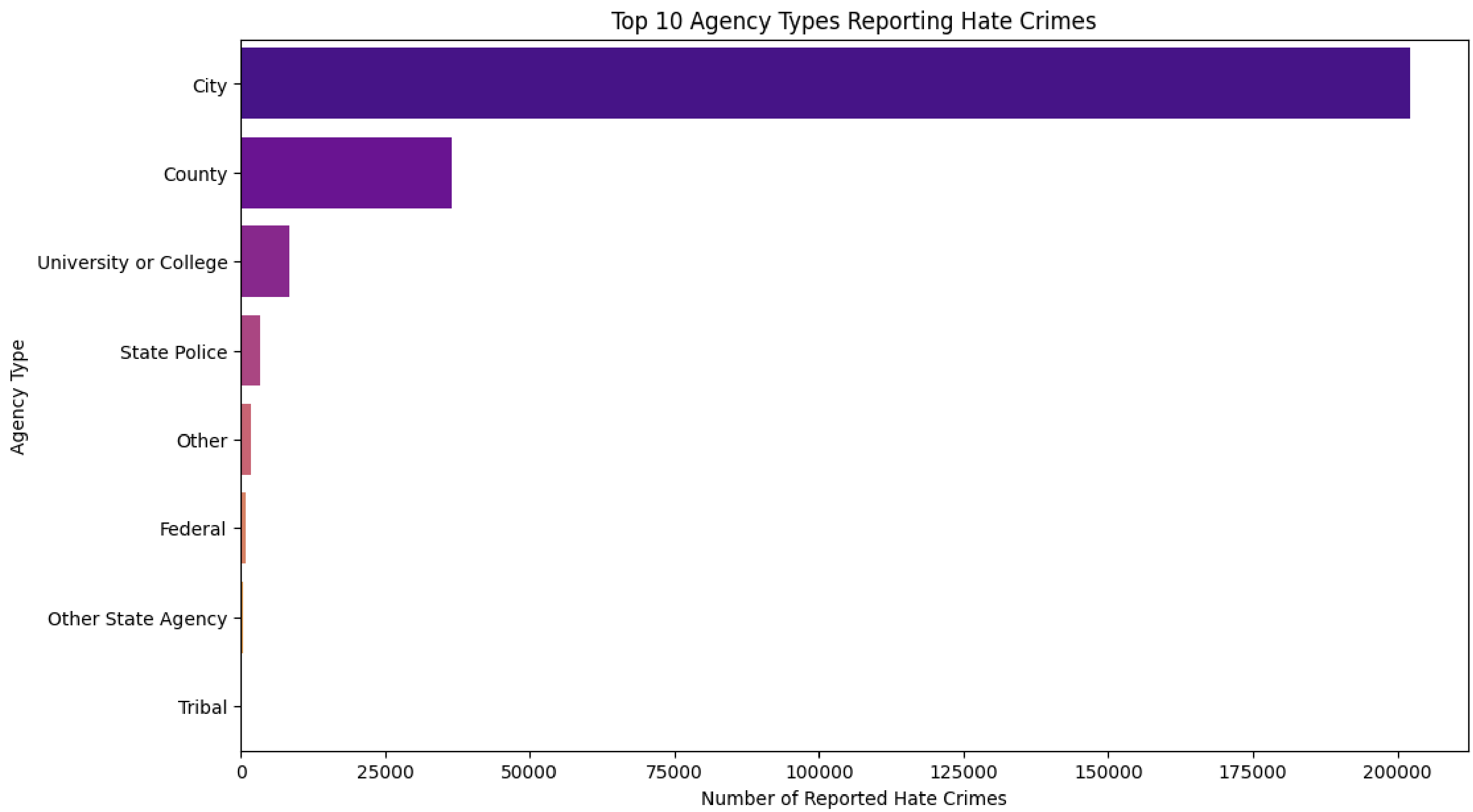
In [53]:
```
df['agency_type_name'].unique()
```

Out[53]:
```
array(['City', 'County', 'Other State Agency', 'University or College',
       'State Police', 'Other', 'Tribal', 'Federal'], dtype=object)
```

In [199…
```
# Count occurrences of each agency type
agency_type_counts = df["agency_type_name"].value_counts().head(10)  # Top 10 agency types

# Check distribution
plt.figure(figsize=(12, 7))
```

```
sns.barplot(x=agency_type_counts.values, y=agency_type_counts.index, palette="plasma")
plt.xlabel("Number of Reported Hate Crimes")
plt.ylabel("Agency Type")
plt.title("Top 10 Agency Types Reporting Hate Crimes")

plt.show()
print('City Police Departments report the most hate crimes → Likely because cities have higher population density & m
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\1508835289.py:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(x=agency_type_counts.values, y=agency_type_counts.index, palette="plasma")
```



City Police Departments report the most hate crimes → Likely because cities have higher population density & more incidents.

In [56]:
```
# Compare Hate Crime Types Across Different Agency Types

# Count most common crime types for each agency type
crime_by_agency_type = df.groupby(["agency_type_name", "offense_name"])["incident_id"].count().reset_index()

# Get top crime type per agency type
top_crime_per_agency_type = crime_by_agency_type.sort_values(by="incident_id", ascending=False).groupby("agency_type_

# Display the result
top_crime_per_agency_type
```

Out[56]:

	agency_type_name	offense_name	incident_id
277	City	Intimidation	61115
464	County	Destruction/Damage/Vandalism of Property	12539
795	University or College	Destruction/Damage/Vandalism of Property	4289
709	State Police	Destruction/Damage/Vandalism of Property	966
611	Other	Destruction/Damage/Vandalism of Property	659
585	Federal	Intimidation	330
651	Other State Agency	Destruction/Damage/Vandalism of Property	230
765	Tribal	Intimidation	33

In [57]:
```
# Most Common Bias Motivations (Race, Religion, LGBTQ) per Agency Type

# Count bias motivations for each agency type
bias_by_agency_type = df.groupby(["agency_type_name", "bias_desc"])["incident_id"].count().reset_index()

# Get top bias motivation per agency type
top_bias_per_agency_type = bias_by_agency_type.sort_values(by="incident_id", ascending=False).groupby("agency_type_na

# Display the result
top_bias_per_agency_type
```

Out[57]:

	agency_type_name	bias_desc	incident_id
94	City	Anti-Black or African American	66193
391	County	Anti-Black or African American	13245
721	University or College	Anti-Black or African American	2734
639	State Police	Anti-Black or African American	1375
551	Other	Anti-Black or African American	527
497	Federal	Anti-Black or African American	266
597	Other State Agency	Anti-Black or African American	171
688	Tribal	Anti-American Indian or Alaska Native	34

In [65]:

```
# Check if Certain Agencies Report Disproportionately High or Low Crimes

# Count total hate crimes per agency type
agency_type_totals = df.groupby("agency_type_name")["incident_id"].count().reset_index()

# Rename the column 'incident_id' to 'incident_id_count'
agency_type_totals = agency_type_totals.rename(columns={"incident_id": "incident_id_count"})

# Normalize by percentage to see distribution
agency_type_totals["percentage"] = (agency_type_totals["incident_id_count"] / agency_type_totals["incident_id_count"]

# Display the result
agency_type_totals
```

Out[65]:

	agency_type_name	incident_id_count	percentage
0	City	202135	79.650952
1	County	36551	14.402859
2	Federal	823	0.324302
3	Other	1759	0.693131
4	Other State Agency	504	0.198600
5	State Police	3421	1.348039
6	Tribal	127	0.050044
7	University or College	8456	3.332072

In [66]:
```
df.columns
```

Out[66]:
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

Check state abbr

In [67]:
```
df['state_abbr'].dtype
```

Out[67]:
```
dtype('O')
```

In [69]:
```
df['state_abbr'].unique()
```

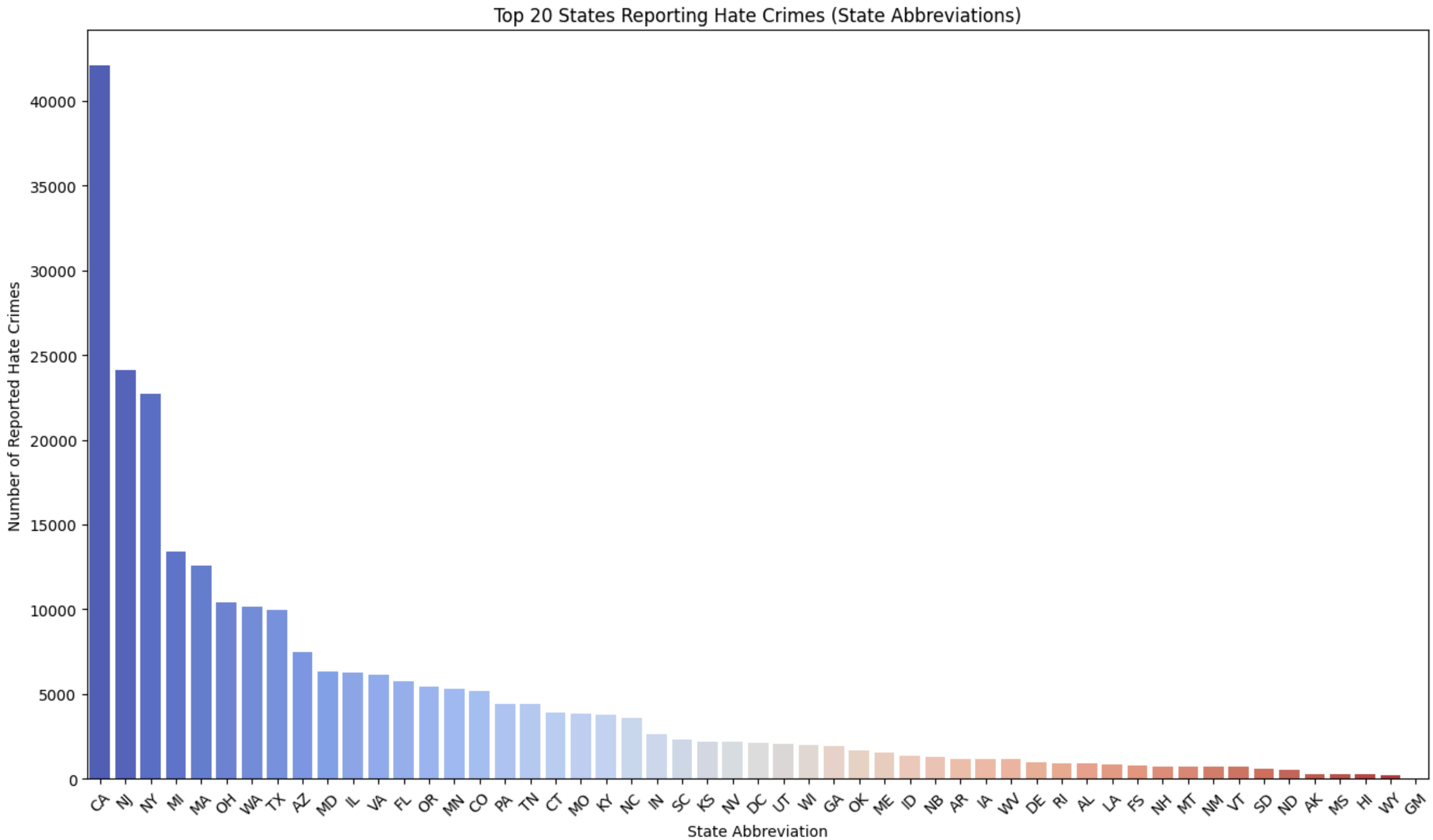
Out[69]:
```
array(['AR', 'AZ', 'CA', 'CO', 'CT', 'DC', 'DE', 'GA', 'IA', 'ID', 'IL',
       'KS', 'MA', 'MD', 'MN', 'MO', 'MS', 'NJ', 'NV', 'NY', 'OH', 'OK',
       'OR', 'PA', 'TN', 'TX', 'VA', 'WA', 'WI', 'AL', 'FL', 'IN', 'KY',
       'LA', 'ME', 'MI', 'NC', 'ND', 'RI', 'SC', 'UT', 'WY', 'AK', 'MT',
       'NM', 'SD', 'VT', 'NH', 'NB', 'WV', 'GM', 'FS', 'HI'], dtype=object)
```

In [70]:
```
df['state_abbr'].isna().sum()
```

Out[70]:
```
np.int64(0)
```

In [200…
```
# Check distribution
# Count occurrences of each state abbreviation
state_abbr_counts = df["state_abbr"].value_counts()  # Top 20 states by reported hate crimes

# Plot the distribution of state abbreviations reporting hate crimes
plt.figure(figsize=(16, 9))

sns.barplot(x=state_abbr_counts.index, y=state_abbr_counts.values, palette="coolwarm")
plt.xlabel("State Abbreviation")
plt.ylabel("Number of Reported Hate Crimes")
plt.title("Top 20 States Reporting Hate Crimes (State Abbreviations)")
plt.xticks(rotation=45)

plt.show()
print('This is the same result as total reported hate crime per state that we plotted before due to being abbreviatic
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\2006769020.py:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(x=state_abbr_counts.index, y=state_abbr_counts.values, palette="coolwarm")
```



This is the same result as total reported hate crime per state that we plotted before due to being abbreviation word from the actual state

In [76]:
```
df.columns
```

Out[76]:
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

Check division_name

In [77]:
```
df['division_name'].dtype
```

Out[77]:
```
dtype('O')
```

In [78]:
```
df['division_name'].value_counts()
```

Out[78]:
```
division_name
Pacific               58182
Middle Atlantic       51258
East North Central    34687
South Atlantic        30389
New England           20460
Mountain              19985
West North Central    14959
West South Central    13682
East South Central     9329
Other                   820
U.S. Territories         25
Name: count, dtype: int64
```

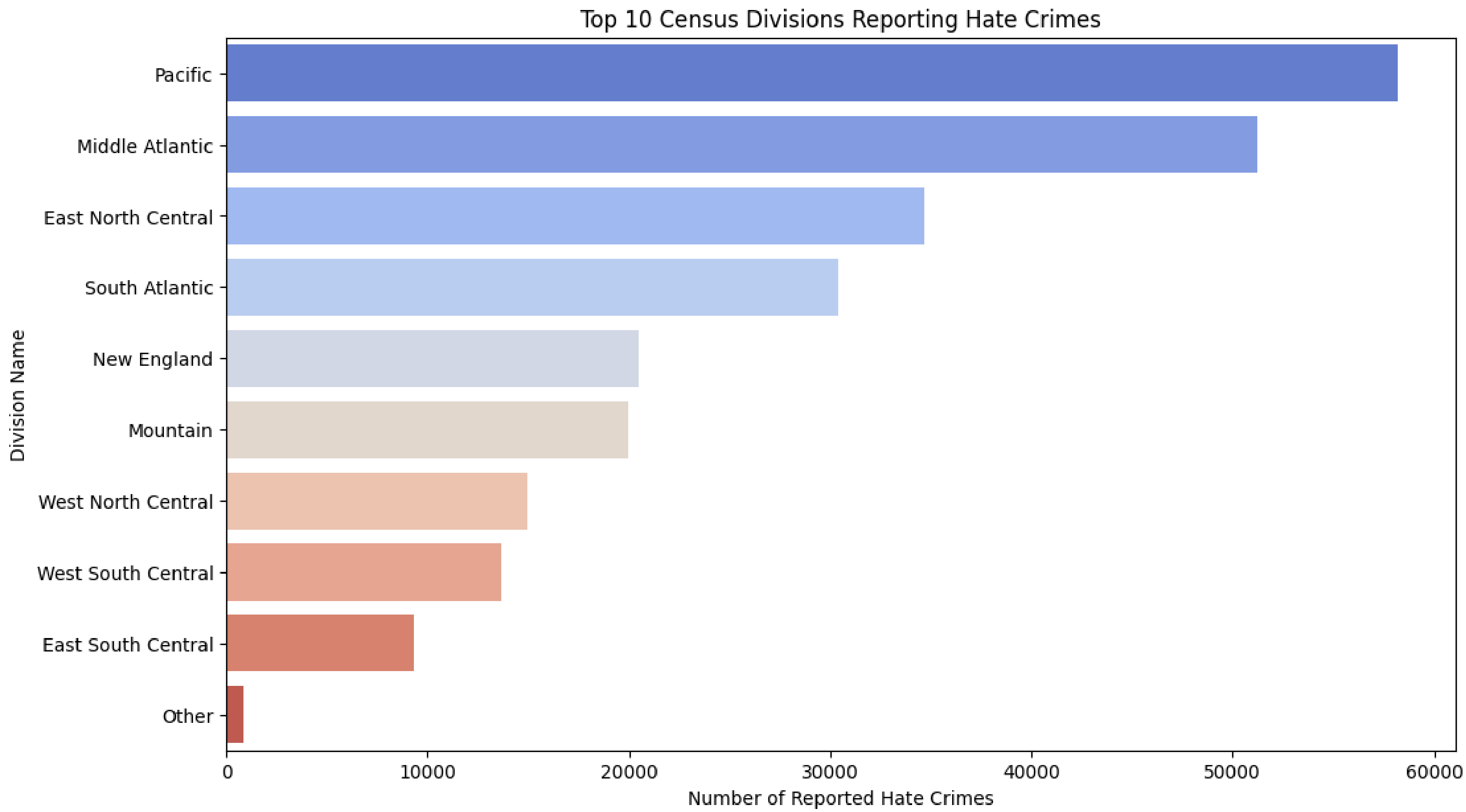
In [79]:
```
df['division_name'].isna().sum()
```

Out[79]:
```
np.int64(0)
```

In [201…
```
# Check distribution
# Count occurrences of each division name
division_counts = df["division_name"].value_counts().head(10)  # Top 10 divisions by reported hate crimes
```

```python
# Plot the distribution of hate crimes per division
plt.figure(figsize=(12, 7))

sns.barplot(x=division_counts.values, y=division_counts.index, palette="coolwarm")
plt.xlabel("Number of Reported Hate Crimes")
plt.ylabel("Division Name")
plt.title("Top 10 Census Divisions Reporting Hate Crimes")

plt.show()
print('Pacific Division reports the most hate crimes → Likely due to California, which had the highest state-level re
print('Note: California is in the Pacific census division, the geographical regions defined by the U.S. just like Cer
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\3886282386.py:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(x=division_counts.values, y=division_counts.index, palette="coolwarm")
```



Pacific Division reports the most hate crimes → Likely due to California, which had the highest state-level reports. Note: California is in the Pacific census division, the geographical regions defined by the U.S. just like Central Region and Northeastern Region in Thailand

In [84]:
```
# Compare Most Common Hate Crime Types in Each Division

# Count most common crime types for each division
crime_by_division = df.groupby(["division_name", "offense_name"])["incident_id"].count().reset_index()

# Get top crime type per division
top_crime_per_division = crime_by_division.sort_values(by="incident_id", ascending=False).groupby("division_name").he
```

```
# Display the result
top_crime_per_division
```

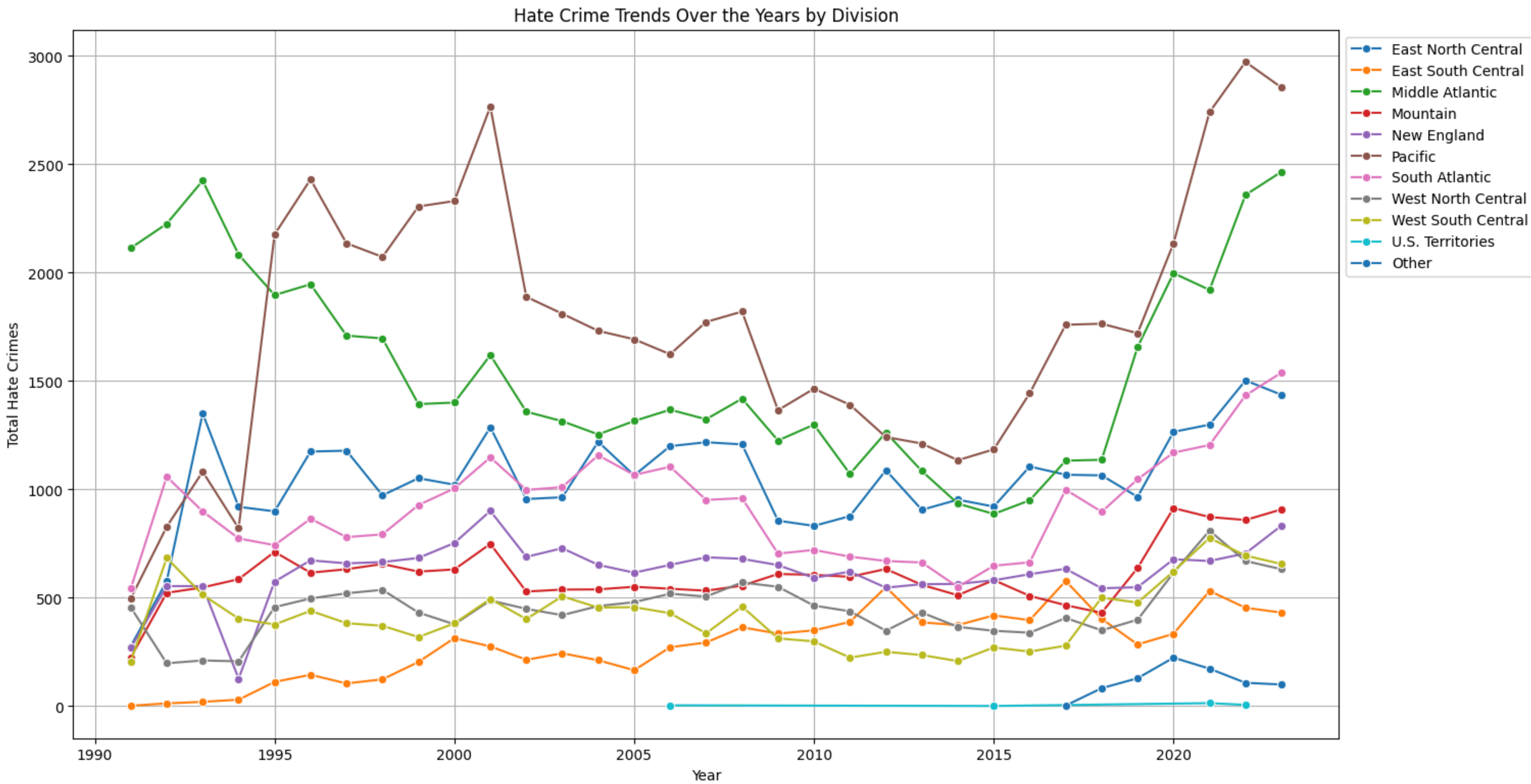
Out[84]:

	division_name	offense_name	incident_id
391	Middle Atlantic	Intimidation	20143
845	Pacific	Destruction/Damage/Vandalism of Property	16561
1023	South Atlantic	Destruction/Damage/Vandalism of Property	10691
127	East North Central	Intimidation	10531
680	New England	Intimidation	6793
539	Mountain	Intimidation	5112
1209	West North Central	Intimidation	4091
1305	West South Central	Destruction/Damage/Vandalism of Property	3230
285	East South Central	Intimidation	2423
745	Other	Intimidation	329
1112	U.S. Territories	Destruction/Damage/Vandalism of Property	7

In [88]:

```
# Trends Over Time: Hate Crime Changes in Each Division

# Aggregate crime trends per division over years
crime_trend_by_division = df.groupby(["data_year", "division_name"])["incident_id"].count().reset_index()

# Plot trends over time
plt.figure(figsize=(16, 9))

sns.lineplot(data=crime_trend_by_division, x="data_year", y="incident_id", hue="division_name", marker="o", palette='
plt.xlabel("Year")
plt.ylabel("Total Hate Crimes")
plt.title("Hate Crime Trends Over the Years by Division")
plt.legend(loc="upper left", bbox_to_anchor=(1, 1))
plt.grid(True)

plt.show()
```

```
print('Pacific & Middle Atlantic divisions show consistently higher hate crime reports.')
print('Sharp increases post-2015, suggesting changes in social dynamics or improved reporting.')
```




```
Pacific & Middle Atlantic divisions show consistently higher hate crime reports.
Sharp increases post-2015, suggesting changes in social dynamics or improved reporting.
```

In [89]:
```
df.columns
```

Out[89]:
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

Check region_name

In [90]:
```
df['region_name'].dtype
```

Out[90]:
```
dtype('O')
```

In [91]:
```
df['region_name'].value_counts()
```

Out[91]:
```
region_name
West               78167
Northeast          71718
South              53400
Midwest            49646
Other                820
U.S. Territories      25
Name: count, dtype: int64
```

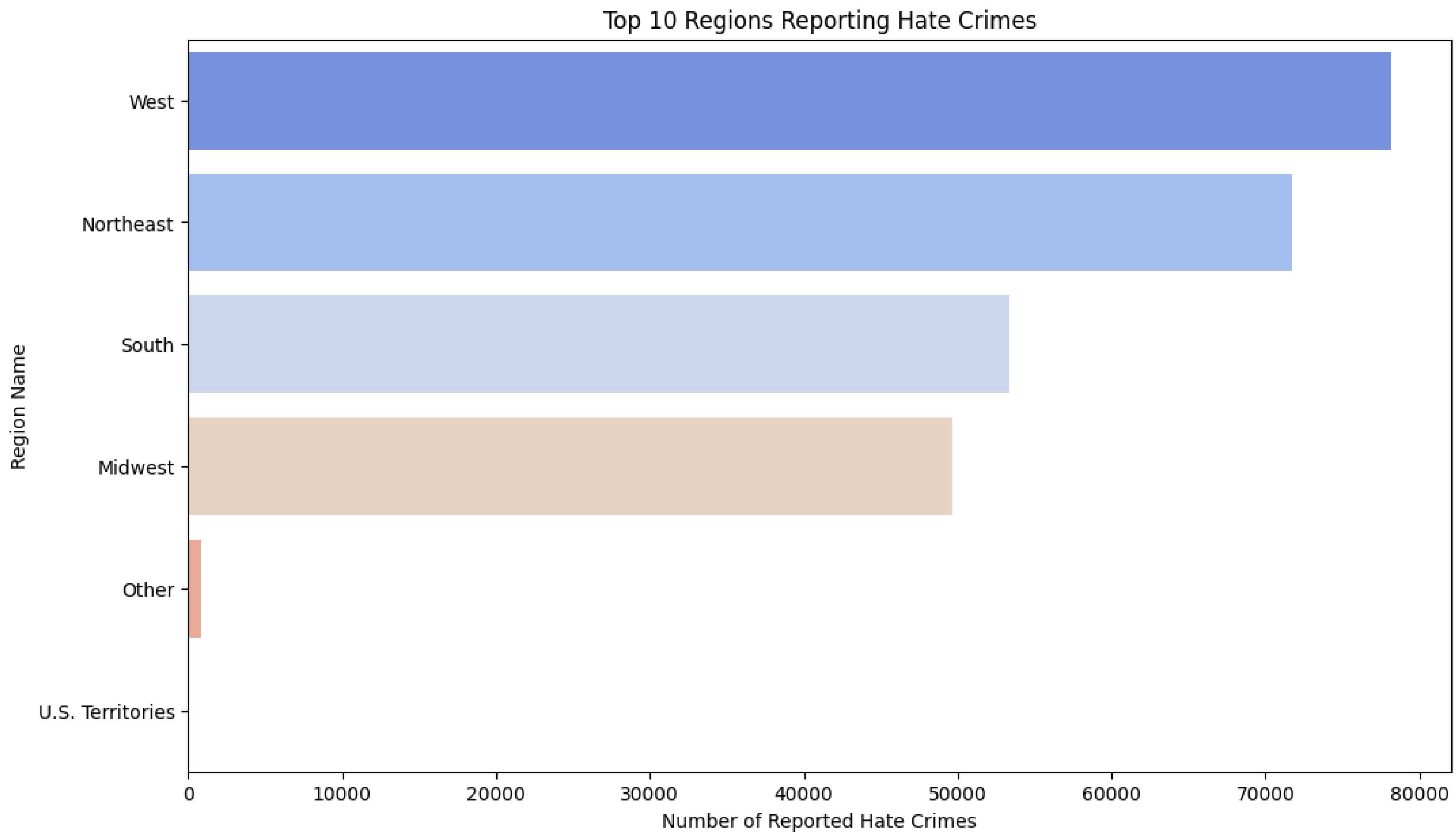
In [92]:
```
df['region_name'].isna().sum()
```

Out[92]:
```
np.int64(0)
```

In [94]:
```
# Check distribution
# Count occurrences of each region name
region_counts = df["region_name"].value_counts().head(10)  # Top 10 regions by reported hate crimes

# Plot the distribution of hate crimes per region
plt.figure(figsize=(12, 7))

sns.barplot(x=region_counts.values, y=region_counts.index, palette="coolwarm")
```

```
plt.xlabel("Number of Reported Hate Crimes")
plt.ylabel("Region Name")
plt.title("Top 10 Regions Reporting Hate Crimes")

plt.show()
print('The West reports the most hate crimes → Likely driven by California’s high reporting numbers.')
print('The Northeast follows closely → Includes states like New York and New Jersey, which have strict reporting requ
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\966949114.py:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(x=region_counts.values, y=region_counts.index, palette="coolwarm")
```



The West reports the most hate crimes → Likely driven by California's high reporting numbers.
The Northeast follows closely → Includes states like New York and New Jersey, which have strict reporting requirement s.

In [95]:
```
# Most Common Bias Motivations (Race, Religion, LGBTQ) in Each Region

# Count bias motivations per region
bias_by_region = df.groupby(["region_name", "bias_desc"])["incident_id"].count().reset_index()

# Get top bias motivation per region
top_bias_per_region = bias_by_region.sort_values(by="incident_id", ascending=False).groupby("region_name").head(1)
```

```
# Display the result
top_bias_per_region
```

Out[95]:

	region_name	bias_desc	incident_id
663	West	Anti-Black or African American	23727
183	Northeast	Anti-Black or African American	22418
497	South	Anti-Black or African American	19471
21	Midwest	Anti-Black or African American	18648
419	Other	Anti-Black or African American	266
617	U.S. Territories	Anti-White	8

In [96]:
```
df.columns
```

Out[96]:
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

Check population_group_code and population_group_description

In [97]:
```
df['population_group_code'].dtype
```

Out[97]:
```
dtype('O')
```

In [98]:
```
df['population_group_code'].unique()
```

Out[98]:
```
array(['3', '8D', '5', '6', '2', '1B', '1C', '9B', '4', '9A', '7', '8C',
       '1A', '9D', '9C', '8B', '8A', '8E', '9E', '0', nan], dtype=object)
```

In [99]:
```
df['population_group_code'].isna().sum()
```

Out[99]:
```
np.int64(667)
```

In [100…
```
df['population_group_description'].dtype
```

Out[100…
```
dtype('O')
```

In [101…
```
df['population_group_description'].unique()
```

Out[101…
```
array(['Cities from 50,000 thru 99,999', 'Non-MSA counties under 10,000',
       'Cities from 10,000 thru 24,999', 'Cities from 2,500 thru 9,999',
       'Cities from 100,000 thru 249,999',
       'Cities from 500,000 thru 999,999',
       'Cities from 250,000 thru 499,999',
       'MSA counties from 25,000 thru 99,999',
       'Cities from 25,000 thru 49,999', 'MSA counties 100,000 or over',
       'Cities under 2,500', 'Non-MSA counties from 10,000 thru 24,999',
       'Cities 1,000,000 or over', 'MSA counties under 10,000',
       'MSA counties from 10,000 thru 24,999',
       'Non-MSA counties from 25,000 thru 99,999',
       'Non-MSA counties 100,000 or over', 'Non-MSA State Police',
       'MSA State Police',
       'Possessions (Puerto Rico, Guam, Virgin Islands, and American Samoa)',
       nan], dtype=object)
```

In [102…
```
df['population_group_description'].isna().sum()
```

Out[102…
```
np.int64(667)
```

In [106…
```
# Count occurrences of each population group (Code + Description)

# Count hate crimes per population group
population_group_counts = df.groupby(["population_group_code", "population_group_description"])["incident_id"].count(

# Sort by highest reported hate crimes
population_group_counts = population_group_counts.sort_values(by="incident_id", ascending=False)

# Display the result
```

```
print('Biggest cities (1,000,000+ population) report the most hate crimes (36,210 cases).\n  - Large urban centers li
population_group_counts
```

```
Biggest cities (1,000,000+ population) report the most hate crimes (36,210 cases).
  - Large urban centers like New York, Los Angeles, and Chicago drive these numbers.
  - More people = more reported incidents & better reporting infrastructure.
```

Out[106…

	population_group_code	population_group_description	incident_id
1	1A	Cities 1,000,000 or over	36210
5	3	Cities from 50,000 thru 99,999	28356
6	4	Cities from 25,000 thru 49,999	28156
2	1B	Cities from 500,000 thru 999,999	24915
7	5	Cities from 10,000 thru 24,999	24756
4	2	Cities from 100,000 thru 249,999	24338
15	9A	MSA counties 100,000 or over	23190
3	1C	Cities from 250,000 thru 499,999	17275
8	6	Cities from 2,500 thru 9,999	15245
9	7	Cities under 2,500	13438
16	9B	MSA counties from 25,000 thru 99,999	6761
11	8B	Non-MSA counties from 25,000 thru 99,999	2441
12	8C	Non-MSA counties from 10,000 thru 24,999	2321
18	9D	MSA counties under 10,000	2005
13	8D	Non-MSA counties under 10,000	1968
17	9C	MSA counties from 10,000 thru 24,999	1154
10	8A	Non-MSA counties 100,000 or over	270
14	8E	Non-MSA State Police	238
19	9E	MSA State Police	47
0	0	Possessions (Puerto Rico, Guam, Virgin Islands…	25

In [109…

```
# Compare Most Common Hate Crime Types in Different Population Groups

# Count most common crime types for each population group
```

```
crime_by_population_group = df.groupby(["population_group_description", "offense_name"])["incident_id"].count().reset

# Get top crime type per population group
top_crime_per_population_group = crime_by_population_group.sort_values(by="incident_id", ascending=False).groupby("po

# Display the result
print('Vandalism (Destruction/Damage of Property) is the most common hate crime in large cities (1M+ people).\n  - Bi
top_crime_per_population_group
```

```
Vandalism (Destruction/Damage of Property) is the most common hate crime in large cities (1M+ people).
  - Biggest cities (New York, LA, Chicago) see more property-related hate crimes than violent offenses.
  - Higher population density + diverse communities might lead to more bias-related vandalism (graffiti, destruction, etc.).
```

Out[109…

	population_group_description	offense_name	incident_id
44	Cities 1,000,000 or over	Destruction/Damage/Vandalism of Property	11245
1287	MSA counties 100,000 or over	Destruction/Damage/Vandalism of Property	9597
654	Cities from 25,000 thru 49,999	Intimidation	9458
935	Cities from 50,000 thru 99,999	Intimidation	9003
200	Cities from 10,000 thru 24,999	Intimidation	8552
1078	Cities from 500,000 thru 999,999	Intimidation	8087
359	Cities from 100,000 thru 249,999	Intimidation	6774
1166	Cities under 2,500	Destruction/Damage/Vandalism of Property	5755
780	Cities from 250,000 thru 499,999	Intimidation	4913
505	Cities from 2,500 thru 9,999	Intimidation	4706
1491	MSA counties from 25,000 thru 99,999	Intimidation	1762
1792	Non-MSA counties from 25,000 thru 99,999	Intimidation	566
1559	MSA counties under 10,000	Destruction/Damage/Vandalism of Property	516
1728	Non-MSA counties from 10,000 thru 24,999	Simple Assault	508
1879	Non-MSA counties under 10,000	Simple Assault	419
1371	MSA counties from 10,000 thru 24,999	Destruction/Damage/Vandalism of Property	303
1646	Non-MSA counties 100,000 or over	Intimidation	99
1616	Non-MSA State Police	Intimidation	64
1230	MSA State Police	Simple Assault	12
1896	Possessions (Puerto Rico, Guam, Virgin Islands...	Simple Assault	7

In [111…

```
df.columns
```

Out[111…

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias'],
      dtype='object')
```

In [116…

```python
# Categorize Hate Crimes into Violent vs. Non-Violent Categories

# Define violent and non-violent crime categories
violent_crimes = ["Aggravated Assault", "Simple Assault", "Murder", "Rape", "Robbery"]
non_violent_crimes = ["Destruction/Damage/Vandalism of Property", "Intimidation", "Theft", "Trespassing"]

# Create a new column classifying each offense as violent or non-violent
df["crime_type"] = df["offense_name"].apply(lambda x: "Violent" if x in violent_crimes else "Non-Violent" if x in non

# Count violent vs. non-violent hate crimes by population group
crime_type_by_population = df.groupby(["population_group_description", "crime_type"])["incident_id"].count().reset_in

# Pivot for visualization
crime_type_pivot = crime_type_by_population.pivot(index="population_group_description", columns="crime_type", values=

# Display the result
print('Non-Violent Crimes (Vandalism, Intimidation) are the most common across all city sizes.\n')
crime_type_pivot
```

```
Non-Violent Crimes (Vandalism, Intimidation) are the most common across all city sizes.
```

Out[116…

crime_type	Non-Violent	Other	Violent
population_group_description			
Cities 1,000,000 or over	20066	1547	14597
Cities from 10,000 thru 24,999	16074	2732	5950
Cities from 100,000 thru 249,999	13408	2468	8462
Cities from 2,500 thru 9,999	8957	2219	4069
Cities from 25,000 thru 49,999	18234	2653	7269
Cities from 250,000 thru 499,999	8882	1484	6909
Cities from 50,000 thru 99,999	17468	2645	8243
Cities from 500,000 thru 999,999	12967	1744	10204
Cities under 2,500	9564	1274	2600
MSA State Police	18	8	21
MSA counties 100,000 or over	15275	1796	6119
MSA counties from 10,000 thru 24,999	537	255	362
MSA counties from 25,000 thru 99,999	3503	1222	2036
MSA counties under 10,000	916	414	675
Non-MSA State Police	106	93	39
Non-MSA counties 100,000 or over	167	48	55
Non-MSA counties from 10,000 thru 24,999	917	630	774
Non-MSA counties from 25,000 thru 99,999	1065	588	788
Non-MSA counties under 10,000	756	548	664
Possessions (Puerto Rico, Guam, Virgin Islands, and American Samoa)	11	5	9

In [121…

```
# Break Down Specific Violent Crime Types by Population Group

# Filter only violent crimes
violent_crime_types = df[df["crime_type"] == "Violent"]

# Count occurrences of each violent crime type per population group
violent_crime_by_population = violent_crime_types.groupby(["population_group_description", "offense_name"])["incident

# Get top violent crime per population group
top_violent_crime_per_population = violent_crime_by_population.sort_values(by="incident_id", ascending=False).groupby

# Display the result
print('"Simple Assault" is the most common violent hate crime across all city sizes.')
print('  - Largest cities (1M+ people) report the most Simple Assault cases (8,443).')
print('Violent hate crimes occur in all city sizes, not just big cities.')
print('  - Cities from 25K-49K still report 4,716 incidents of Simple Assault.')
top_violent_crime_per_population
```

```
"Simple Assault" is the most common violent hate crime across all city sizes.
  - Largest cities (1M+ people) report the most Simple Assault cases (8,443).
Violent hate crimes occur in all city sizes, not just big cities.
  - Cities from 25K-49K still report 4,716 incidents of Simple Assault.
```

Out[121…

	population_group_description	offense_name	incident_id
3	Cities 1,000,000 or over	Simple Assault	8443
31	Cities from 500,000 thru 999,999	Simple Assault	5931
27	Cities from 50,000 thru 99,999	Simple Assault	5118
11	Cities from 100,000 thru 249,999	Simple Assault	4902
19	Cities from 25,000 thru 49,999	Simple Assault	4716
7	Cities from 10,000 thru 24,999	Simple Assault	3917
23	Cities from 250,000 thru 499,999	Simple Assault	3863
41	MSA counties 100,000 or over	Simple Assault	3851
15	Cities from 2,500 thru 9,999	Simple Assault	2749
35	Cities under 2,500	Simple Assault	1886
49	MSA counties from 25,000 thru 99,999	Simple Assault	1377
63	Non-MSA counties from 10,000 thru 24,999	Simple Assault	508
67	Non-MSA counties from 25,000 thru 99,999	Simple Assault	505
53	MSA counties under 10,000	Simple Assault	461
71	Non-MSA counties under 10,000	Simple Assault	419
45	MSA counties from 10,000 thru 24,999	Simple Assault	245
59	Non-MSA counties 100,000 or over	Simple Assault	43
56	Non-MSA State Police	Simple Assault	25
37	MSA State Police	Simple Assault	12
73	Possessions (Puerto Rico, Guam, Virgin Islands...	Simple Assault	7

In [122…

```
df.columns
```

Out[122…
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type'],
      dtype='object')
```

Check incident_date

In [123…
```
df['incident_date'].dtype
```

Out[123…
```
dtype('O')
```

In [124…
```
df['incident_date'].unique()
```

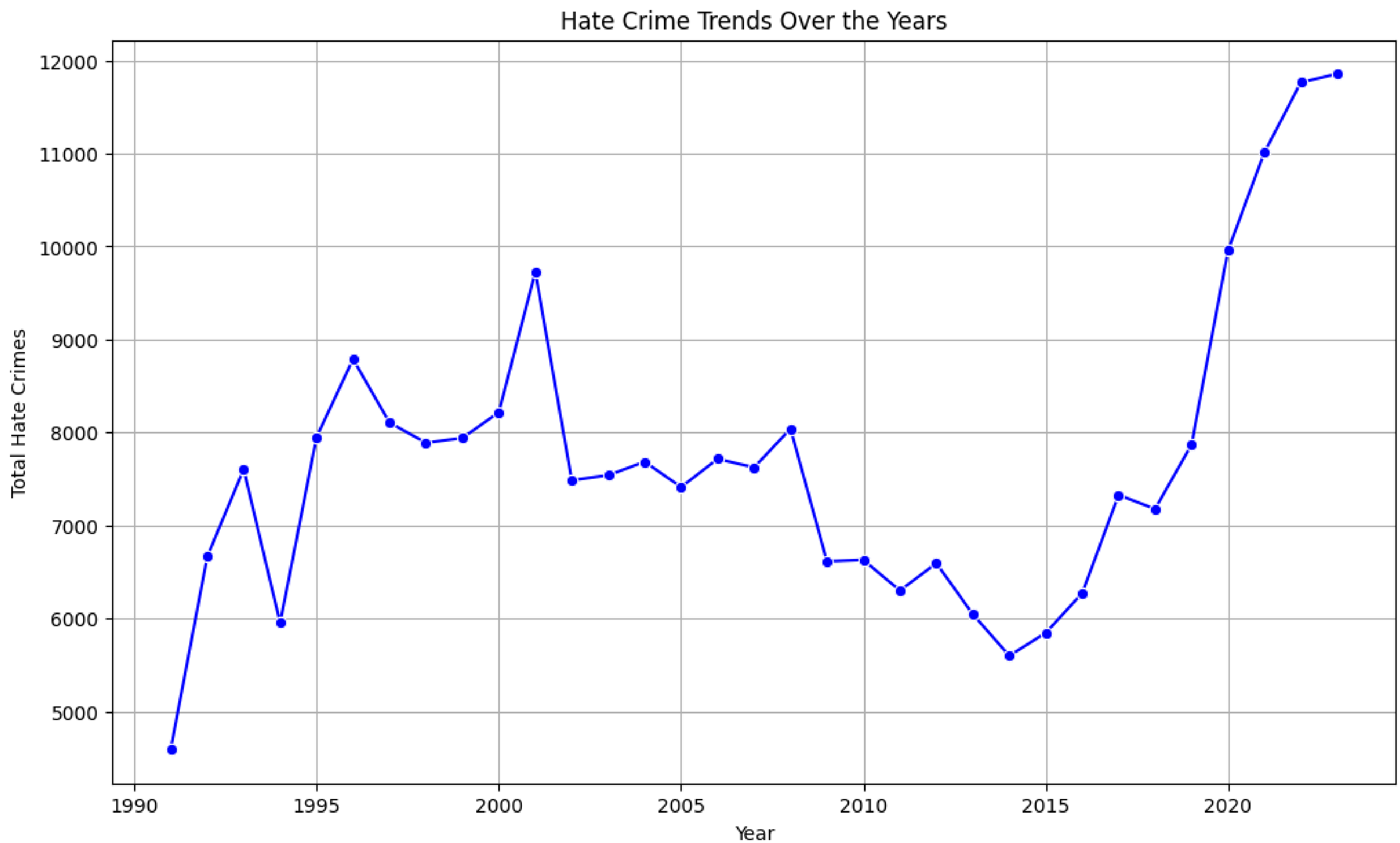
Out[124…
```
array(['1991-07-04', '1991-12-24', '1991-07-10', ..., '2023-06-30',
       '2023-04-09', '2023-04-17'], shape=(12053,), dtype=object)
```

In [125…
```
df['incident_date'].isna().sum()
```

Out[125…
```
np.int64(0)
```

In [128…
```
# Check distribution
# Convert 'incident_date' to datetime format before extracting year
df["incident_date"] = pd.to_datetime(df["incident_date"])

# Extract year from the incident_date column
df["year"] = df["incident_date"].dt.year

# Count total hate crimes per year
yearly_trends = df.groupby("year")["incident_id"].count().reset_index()

# Plot trends over years
plt.figure(figsize=(12, 7))

sns.lineplot(data=yearly_trends, x="year", y="incident_id", marker="o", color="blue")
```

```
plt.xlabel("Year")
plt.ylabel("Total Hate Crimes")
plt.title("Hate Crime Trends Over the Years")
plt.grid(True)

plt.show()
print('Sharp rise in the early 1990s')
print('  - Hate crime reports spiked quickly between 1991-1993.\n')
print('Post-2015 resurgence of hate crimes')
print('  - A sharp increase from 2016-2020, reaching new highs.')
print('  - This could be influenced by political climate, social unrest, or more active reporting.')
```

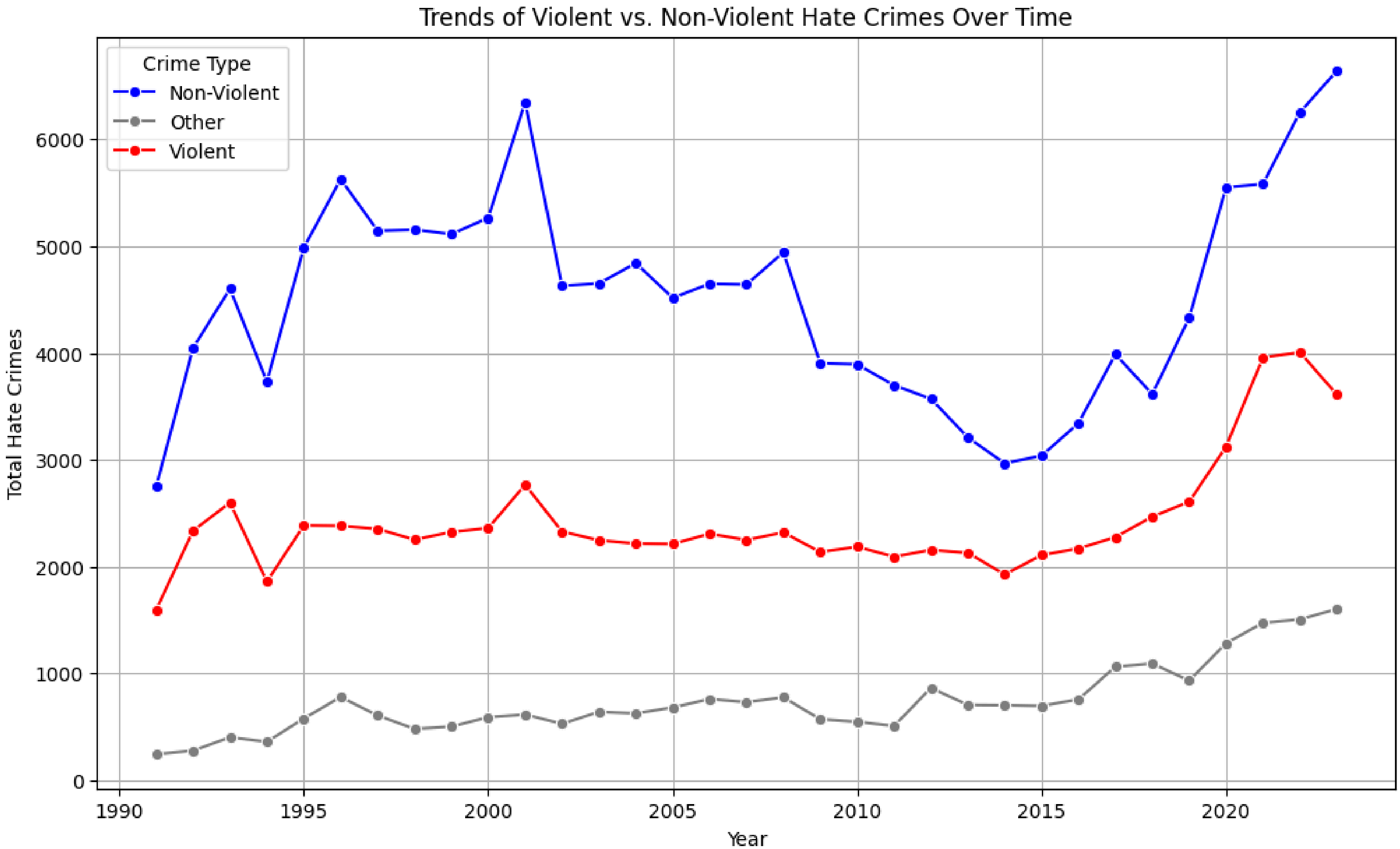



```
Sharp rise in the early 1990s
  - Hate crime reports spiked quickly between 1991-1993.

Post-2015 resurgence of hate crimes
  - A sharp increase from 2016-2020, reaching new highs.
  - This could be influenced by political climate, social unrest, or more active reporting.
```

In [130…

```python
# Compare Trends of Violent vs. Non-Violent Hate Crimes Over Time

# Aggregate crime type counts per year
crime_type_trends = df.groupby(["year", "crime_type"])["incident_id"].count().reset_index()
```

```
# Plot trends over years
plt.figure(figsize=(12, 7))

sns.lineplot(data=crime_type_trends, x="year", y="incident_id", hue="crime_type", marker="o", palette={"Violent": "re
plt.xlabel("Year")
plt.ylabel("Total Hate Crimes")
plt.title("Trends of Violent vs. Non-Violent Hate Crimes Over Time")
plt.legend(title="Crime Type")
plt.grid(True)

plt.show()
print('Non-Violent Hate Crimes (Blue Line) have always been more frequent than Violent ones.\n')
print('Violent Hate Crimes (Red Line) remained steady but increased post-2015.')
print('  - Violent offenses (e.g., assaults, murder, robbery) stayed stable from the mid-1990s to mid-2010s.')
print('  - Post-2015, violent crimes started rising again.\n')
print('Recent years (2020+) show a sharp increase in both categories.')
print('  - Potential reasons include social movements, political events, law enforcement changes, or better reporting
print('The "Other" category (Gray Line) remains consistently low.')
print('  - Other category includes Burglary/ Breaking & Entering and Shoplifting')
print('  - These are crimes that do not fall under the standard "Violent" or "Non-Violent" categories but are still r
```

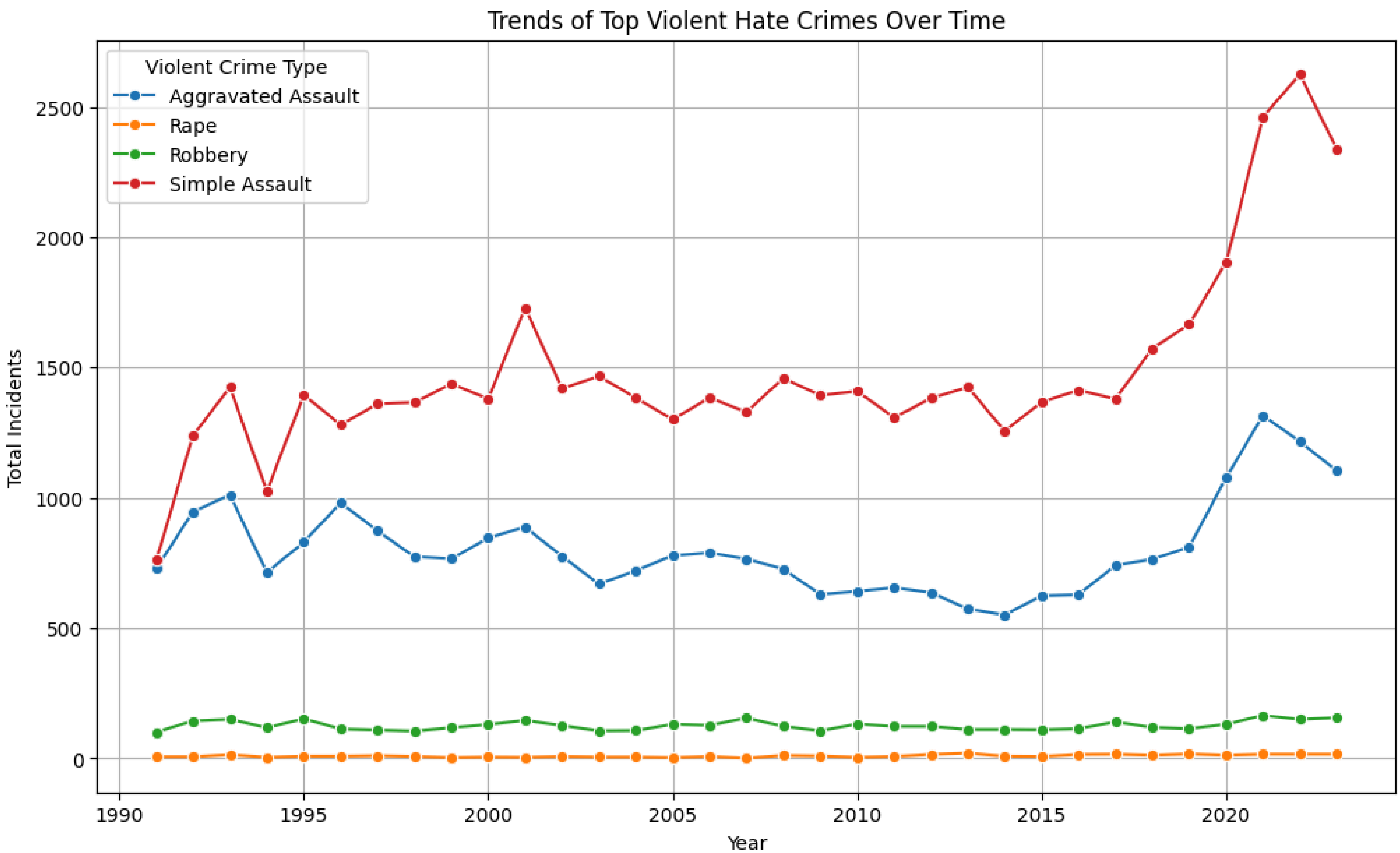

Trends of Violent vs. Non-Violent Hate Crimes Over Time
Crime Type
Non-Violent
Other
Violent
Total Hate Crimes
0
1000
2000
3000
4000
5000
6000
1990
1995
2000
2005
2010
2015
2020
Year

```
Non-Violent Hate Crimes (Blue Line) have always been more frequent than Violent ones.

Violent Hate Crimes (Red Line) remained steady but increased post-2015.
  - Violent offenses (e.g., assaults, murder, robbery) stayed stable from the mid-1990s to mid-2010s.
  - Post-2015, violent crimes started rising again.

Recent years (2020+) show a sharp increase in both categories.
  - Potential reasons include social movements, political events, law enforcement changes, or better reporting mechan
isms.

The "Other" category (Gray Line) remains consistently low.
  - Other category includes Burglary/ Breaking & Entering and Shoplifting
  - These are crimes that do not fall under the standard "Violent" or "Non-Violent" categories but are still recorded
as hate crimes.
```

In [132…

```python
# Breakdown of Violent Crime Types Over Time

# Filter only violent crimes
violent_crimes_over_time = df[df["crime_type"] == "Violent"]

# Count occurrences of each violent crime type per year
violent_crime_trends = violent_crimes_over_time.groupby(["year", "offense_name"])["incident_id"].count().reset_index(

# Get the top 5 most common violent crime types
top_violent_crimes = violent_crime_trends.groupby("offense_name")["incident_id"].sum().nlargest(5).index

# Filter data to only include the top 5 violent crime types
violent_crime_trends_filtered = violent_crime_trends[violent_crime_trends["offense_name"].isin(top_violent_crimes)]

# Plot trends over years
plt.figure(figsize=(12, 7))

sns.lineplot(data=violent_crime_trends_filtered, x="year", y="incident_id", hue="offense_name", marker="o")
plt.xlabel("Year")
plt.ylabel("Total Incidents")
plt.title("Trends of Top Violent Hate Crimes Over Time")
plt.legend(title="Violent Crime Type")
plt.grid(True)

plt.show()
print('Simple Assault (Pink Line) is the most common violent hate crime.')
print('  - Increased significantly after 2015.')
```

```
print('  - Massive spike in 2020+, reaching over 2,500 cases per year.\n')
print('Aggravated Assault (Orange Line) is the second most common.')
print('  - Generally steady from 1990s-2015 but saw a major rise post-2018.')
print('  - Highest recorded increase in 2020-2022.\n')
print('Robbery (Red Line) has remained low but steady.')
print('  - It fluctuates around the 100-200 cases per year range.\n')
print('Rape (Light Orange Line) is the least reported violent hate crime.')
print('  - Consistently low throughout all years, with very slight fluctuations.')
```



```
Simple Assault (Pink Line) is the most common violent hate crime.
  - Increased significantly after 2015.
  - Massive spike in 2020+, reaching over 2,500 cases per year.

Aggravated Assault (Orange Line) is the second most common.
  - Generally steady from 1990s-2015 but saw a major rise post-2018.
  - Highest recorded increase in 2020-2022.

Robbery (Red Line) has remained low but steady.
  - It fluctuates around the 100-200 cases per year range.

Rape (Light Orange Line) is the least reported violent hate crime.
  - Consistently low throughout all years, with very slight fluctuations.
```

In [133…

```
df.columns
```

Out[133…

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type', 'year'],
      dtype='object')
```

Check adult_victim_count

In [134…

```
df['adult_victim_count'].dtype
```

Out[134…

```
dtype('float64')
```

In [135…

```
df['adult_victim_count'].unique()
```

Out[135…

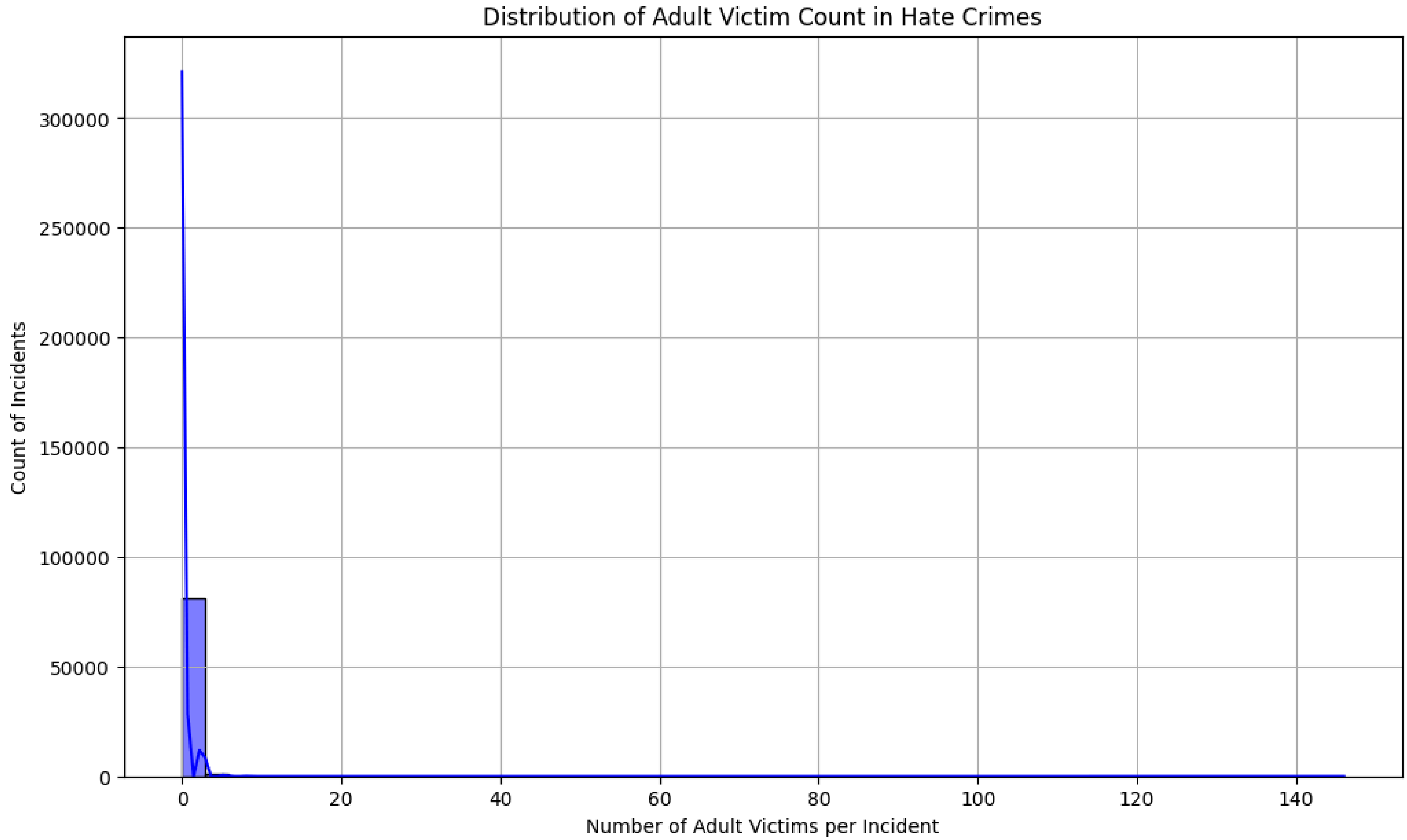
```
array([ nan,   1.,   0.,   3.,   2.,   4.,   7.,   6.,   5.,   9.,  12.,
        13.,  10.,  75.,  14.,   8.,  26.,  27.,  50.,  17.,  80.,  43.,
        15., 146.,  20.,  60.,  40.,  21.])
```

In [136…

```
df['adult_victim_count'].isna().sum()
```

Out[136…
```
np.int64(171076)
```

In [141…
```
# Check distribution

# Plot the distribution of adult victim counts
plt.figure(figsize=(12, 7))

sns.histplot(df["adult_victim_count"], bins=50, kde=True, color="blue")
plt.xlabel("Number of Adult Victims per Incident")
plt.ylabel("Count of Incidents")
plt.title("Distribution of Adult Victim Count in Hate Crimes")
plt.grid(True)

plt.show()
print('Most incidents involve 0-2 adult victims.')
print('  - The distribution is heavily skewed toward 0 or 1 adult victim per case.')
print('  - The mean (average) is ~0.75 victims per incident.\n')
print('A few extreme cases have 10+ adult victims.')
print('  - Some incidents involved dozens of victims (outliers beyond 100+).')
print('  - These might be mass hate crimes, large group attacks, or systemic discrimination cases.')
```




```
Most incidents involve 0-2 adult victims.
  - The distribution is heavily skewed toward 0 or 1 adult victim per case.
  - The mean (average) is ~0.75 victims per incident.

A few extreme cases have 10+ adult victims.
  - Some incidents involved dozens of victims (outliers beyond 100+).
  - These might be mass hate crimes, large group attacks, or systemic discrimination cases.
```

Check juvenile_victim_count

In [143…
```
df['juvenile_victim_count'].dtype
```

Out[143…
```
dtype('float64')
```

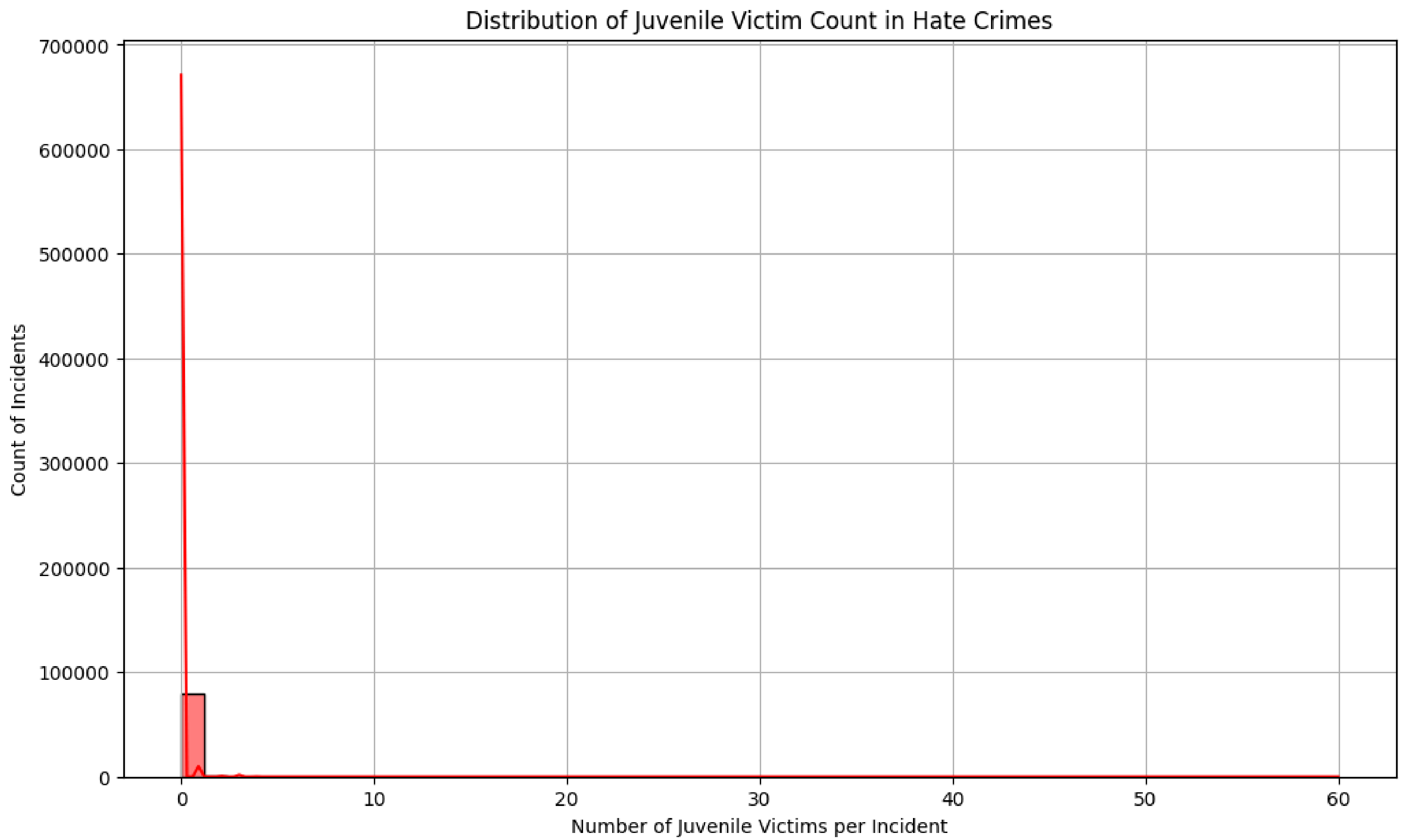
In [144…
```
df['juvenile_victim_count'].unique()
```

Out[144…
```
array([nan,  0.,  4.,  1.,  2.,  3.,  5., 40.,  9.,  6., 10.,  7., 20.,
        8., 29., 60., 12., 18.])
```

In [145…
```
df['juvenile_victim_count'].isna().sum()
```

Out[145…
```
np.int64(173713)
```

In [146…
```
# Check distribution

# Plot the distribution of juvenile victim counts
plt.figure(figsize=(12, 7))

sns.histplot(df["juvenile_victim_count"], bins=50, kde=True, color="red")
plt.xlabel("Number of Juvenile Victims per Incident")
plt.ylabel("Count of Incidents")
plt.title("Distribution of Juvenile Victim Count in Hate Crimes")
plt.grid(True)

plt.show()
```



In [148…

```
# Compare Victim Counts Across Different Crime Types

# Aggregate average number of victims per crime type
victim_by_crime_type = df.groupby("offense_name")[["adult_victim_count", "juvenile_victim_count"]].mean().reset_index

# Display the result
print('Hate crimes overwhelmingly target adults.')
print('  - Most crimes involve 0-2 adult victims, with very few cases affecting large groups.\n')
print('Juvenile victim counts are extremely low.')
print('  - Most hate crimes do not involve juvenile victims at all.\n')
```

```
print('Certain violent crimes (Aggravated Assault) involve more adult victims.')
print('  - Aggravated Assault = ~0.92 adult victims per case.\n')
victim_by_crime_type
```

```
Hate crimes overwhelmingly target adults.
  - Most crimes involve 0-2 adult victims, with very few cases affecting large groups.

Juvenile victim counts are extremely low.
  - Most hate crimes do not involve juvenile victims at all.

Certain violent crimes (Aggravated Assault) involve more adult victims.
  - Aggravated Assault = ~0.92 adult victims per case.
```

Out[148…

	offense_name	adult_victim_count	juvenile_victim_count
0	Aggravated Assault	0.923640	0.108435
1	Aggravated Assault;All Other Larceny	0.923077	0.000000
2	Aggravated Assault;All Other Larceny;Burglary/...	0.000000	0.000000
3	Aggravated Assault;All Other Larceny;Destructi...	1.000000	0.000000
4	Aggravated Assault;All Other Larceny;Extortion...	4.000000	0.000000
...	...	...	...
418	Theft of Motor Vehicle Parts or Accessories	0.705357	0.008929
419	Treason	0.000000	0.000000
420	Weapon Law Violations	0.000000	0.000000
421	Welfare Fraud	1.000000	0.000000
422	Wire Fraud	0.833333	0.055556

423 rows × 3 columns

In [149…

```
df.columns
```

Out[149…

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type', 'year'],
      dtype='object')
```

Check total_offender_count

In [152…

```
df['total_offender_count'].dtype
```

Out[152…

```
dtype('int64')
```

In [153…

```
df['total_offender_count'].unique()
```

Out[153…

```
array([ 1,  2, 10,  0,  5,  4,  6,  3, 11, 12, 26, 25,  8,  9, 40,  7, 35,
       17, 16, 20, 13, 30, 15, 14, 50, 29, 22, 99, 75, 18, 21, 23, 60, 36])
```

In [154…

```
df['total_offender_count'].isna().sum()
```

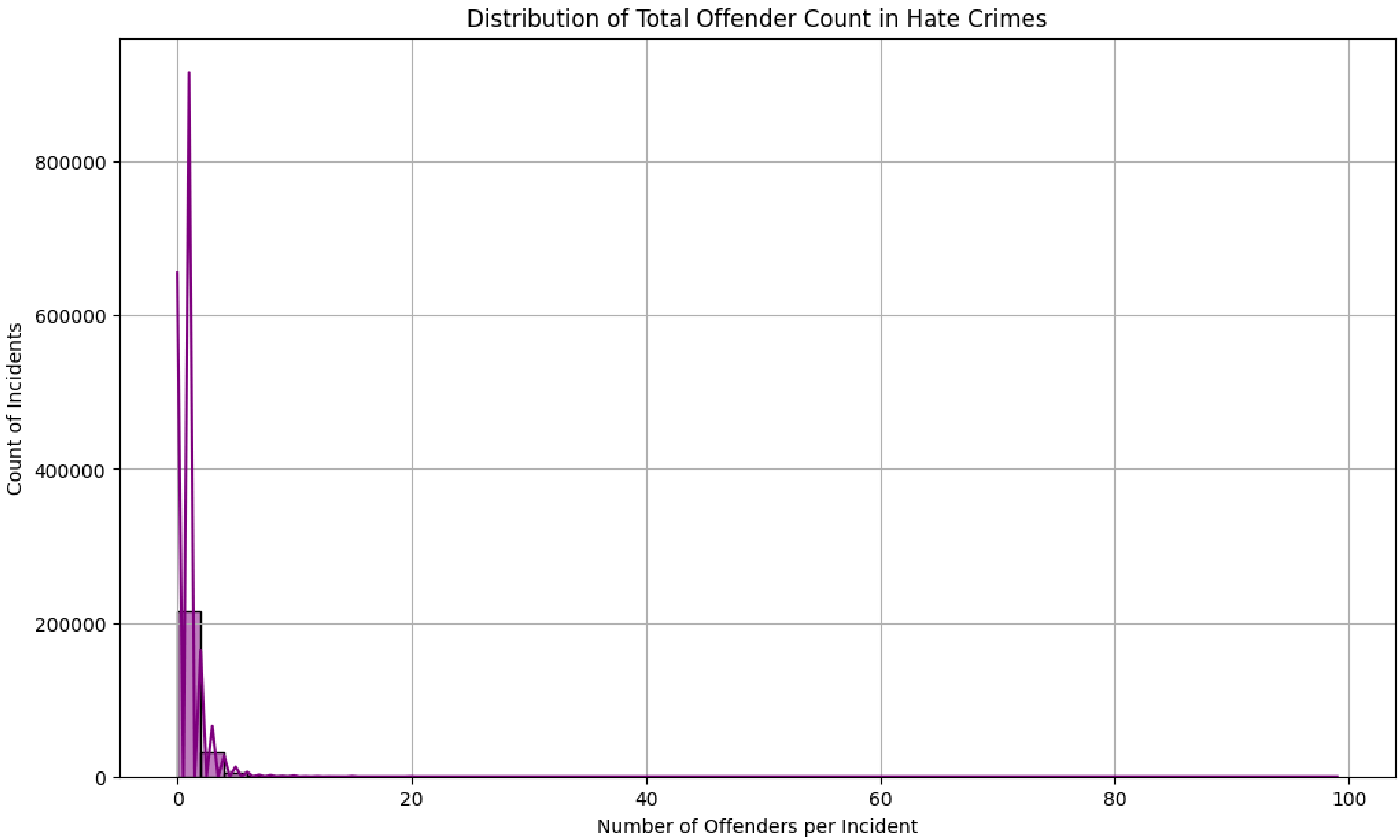
Out[154…

```
np.int64(0)
```

In [161…

```
# Check distribution

# Plot the distribution of total offender counts
plt.figure(figsize=(12, 7))

sns.histplot(df["total_offender_count"], bins=50, kde=True, color="purple")
plt.xlabel("Number of Offenders per Incident")
plt.ylabel("Count of Incidents")
plt.title("Distribution of Total Offender Count in Hate Crimes")
plt.grid(True)

plt.show()

# Summary statistics
```

```
print(f"Summary statistic of total_offender_count:\n\n{df["total_offender_count"].describe()}\n\n")
print('Most incidents involve only 1 offender.')
print('  - The mean (average) is ~0.95 offenders per incident, which means most cases involve a single perpetrator.')
print('  - A huge spike at 0 and 1 offenders, suggesting some cases lack a known offender (reported but unsolved).\n
print('Some incidents involve multiple offenders.')
print('  - A small number of cases involve more than 5-10 offenders.')
print('  - This could indicate group-based hate crimes, extremist activities, or gang-related incidents.\n')
print('Outliers exist (cases with 20+ offenders).')
print('  - Some hate crimes involve organized attacks or large groups acting together.')
print('  - These cases could be riots, mob violence, or targeted mass hate crimes.')
```



```
Summary statistic of total_offender_count:

count    253776.000000
mean          0.949542
std           1.298449
min           0.000000
25%           0.000000
50%           1.000000
75%           1.000000
max          99.000000
Name: total_offender_count, dtype: float64


Most incidents involve only 1 offender.
  - The mean (average) is ~0.95 offenders per incident, which means most cases involve a single perpetrator.
  - A huge spike at 0 and 1 offenders, suggesting some cases lack a known offender (reported but unsolved).

Some incidents involve multiple offenders.
  - A small number of cases involve more than 5-10 offenders.
  - This could indicate group-based hate crimes, extremist activities, or gang-related incidents.

Outliers exist (cases with 20+ offenders).
  - Some hate crimes involve organized attacks or large groups acting together.
  - These cases could be riots, mob violence, or targeted mass hate crimes.
```

In [162…

```
df.columns
```

Out[162…

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type', 'year'],
      dtype='object')
```

Check adult_offender_count

In [163…

```
df['adult_offender_count'].dtype
```

Out[163…
```
dtype('float64')
```

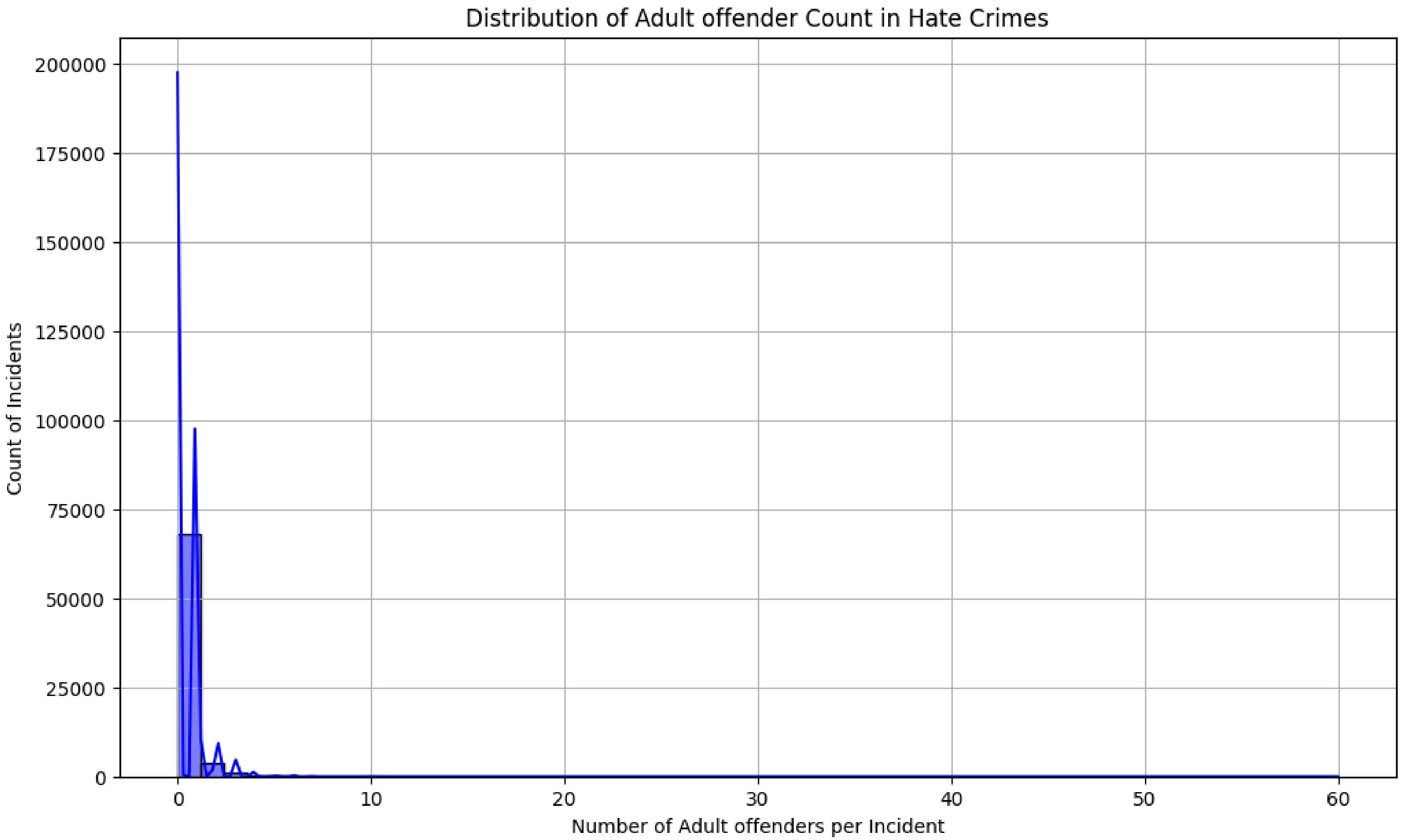
In [164…
```
df['adult_offender_count'].unique()
```

Out[164…
```
array([nan,  1.,  0.,  4.,  2.,  3.,  5.,  9.,  6.,  7.,  8., 20., 13.,
       19., 10., 30., 60., 15.])
```

In [165…
```
df['adult_offender_count'].isna().sum()
```

Out[165…
```
np.int64(180557)
```

In [166…
```
# Check distribution
# Plot the distribution of adult offender counts
plt.figure(figsize=(12, 7))

sns.histplot(df["adult_offender_count"], bins=50, kde=True, color="blue")
plt.xlabel("Number of Adult offenders per Incident")
plt.ylabel("Count of Incidents")
plt.title("Distribution of Adult offender Count in Hate Crimes")
plt.grid(True)

plt.show()
```



Check juvenile_offender_count

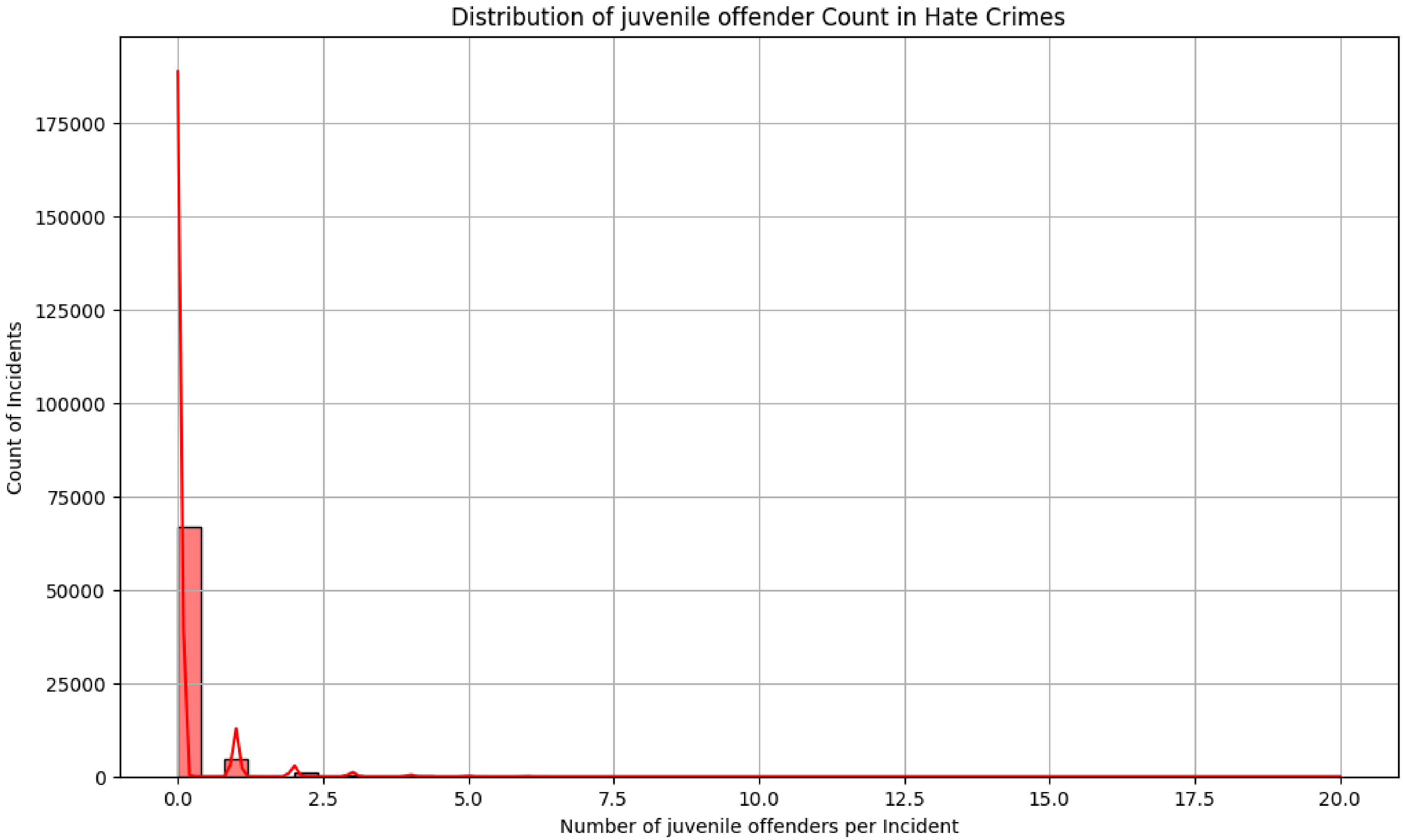
In [167…
```
df['juvenile_offender_count'].dtype
```
Out[167…
```
dtype('float64')
```
In [168…
```
df['juvenile_offender_count'].unique()
```

Out[168…
```
array([nan,  0.,  1.,  3.,  2.,  5.,  6.,  4., 13.,  7., 12., 10.,  9.,
       15., 11., 20.,  8.])
```

In [169…
```python
df['juvenile_offender_count'].isna().sum()
```

Out[169…
```
np.int64(180564)
```

In [171…
```python
# Check distribution
# Plot the distribution of juvenile offender counts
plt.figure(figsize=(12, 7))

sns.histplot(df["juvenile_offender_count"], bins=50, kde=True, color="red")
plt.xlabel("Number of juvenile offenders per Incident")
plt.ylabel("Count of Incidents")
plt.title("Distribution of juvenile offender Count in Hate Crimes")
plt.grid(True)

plt.show()
```



In [174…

```
# Compare Adult vs. Juvenile Offender Counts

# Summary statistics for juvenile offender count
juvenile_offender_summary = df["juvenile_offender_count"].describe()

# Display comparison summary for both adult and juvenile offender counts
offender_summary = pd.DataFrame({
    "Total Offender Count": df["total_offender_count"].describe(),
    "Adult Offender Count": df["adult_offender_count"].describe(),
    "Juvenile Offender Count": df["juvenile_offender_count"].describe()
```

```
})

# Display the result
print('Hate crimes are overwhelmingly committed by adult offenders.')
print('  - The mean adult offender count is ~0.89 per incident, while juvenile offenders are much lower (~0.09 per ir
print('  - Most hate crimes involve only one adult offender.')
offender_summary
```

```
Hate crimes are overwhelmingly committed by adult offenders.
  - The mean adult offender count is ~0.89 per incident, while juvenile offenders are much lower (~0.09 per inciden
t).
  - Most hate crimes involve only one adult offender.
```

Out[174…

	Total Offender Count	Adult Offender Count	Juvenile Offender Count
count	253776.000000	73219.000000	73212.000000
mean	0.949542	0.623090	0.128804
std	1.298449	0.808085	0.531138
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	1.000000	1.000000	0.000000
75%	1.000000	1.000000	0.000000
max	99.000000	60.000000	20.000000

In [176…

```
# Compare Offender Counts Across Different Crime Types

# Aggregate average number of offenders per crime type
offender_by_crime_type = df.groupby("offense_name")[["total_offender_count", "adult_offender_count", "juvenile_offend

# Display the result
print('Aggravated Assault involves the most offenders (~1.64 per case).')
print('  - Hate-motivated physical violence is more likely to involve multiple attackers.\n')
print('Combination Crimes (multiple offenses at once) tend to have the highest offender counts.')
print('  - Cases where multiple crimes occur at once (e.g., assault + robbery + vandalism) often involve more than or
print('Some crimes involve mostly juvenile offenders.')
print('  - A few rare cases had juvenile-only offender groups.')
```

```
print('  - These could be related to gang-related activity or school-based hate crimes.\n')
offender_by_crime_type
```

```
Aggravated Assault involves the most offenders (~1.64 per case).
  - Hate-motivated physical violence is more likely to involve multiple attackers.

Combination Crimes (multiple offenses at once) tend to have the highest offender counts.
  - Cases where multiple crimes occur at once (e.g., assault + robbery + vandalism) often involve more than one offen
der.

Some crimes involve mostly juvenile offenders.
  - A few rare cases had juvenile-only offender groups.
  - These could be related to gang-related activity or school-based hate crimes.
```

Out[176…

	offense_name	total_offender_count	adult_offender_count	juvenile_offender_count
0	Aggravated Assault	1.637726	0.892724	0.094683
1	Aggravated Assault;All Other Larceny	1.478261	0.600000	0.500000
2	Aggravated Assault;All Other Larceny;Burglary/...	2.000000	0.000000	0.000000
3	Aggravated Assault;All Other Larceny;Destructi...	1.500000	2.000000	0.000000
4	Aggravated Assault;All Other Larceny;Extortion...	4.000000	0.000000	4.000000
...	...	...	...	...
418	Theft of Motor Vehicle Parts or Accessories	0.354701	0.333333	0.060606
419	Treason	1.000000	1.000000	0.000000
420	Weapon Law Violations	1.182013	0.946360	0.114943
421	Welfare Fraud	0.200000	1.000000	0.000000
422	Wire Fraud	0.615385	0.250000	0.000000

423 rows × 4 columns

In [177…

```
df.columns
```

Out[177…

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type', 'year'],
      dtype='object')
```

Check offender_race

In [178…

```
df['offender_race'].dtype
```

Out[178…

```
dtype('O')
```

In [179…

```
df['offender_race'].value_counts()
```

Out[179…

```
offender_race
White                                        99689
Unknown                                      98890
Black or African American                    33599
Not Specified                                12734
Multiple                                      5243
Asian                                         1979
American Indian or Alaska Native              1467
Native Hawaiian or Other Pacific Islander      175
Name: count, dtype: int64
```

In [180…

```
df['offender_race'].isna().sum()
```

Out[180…

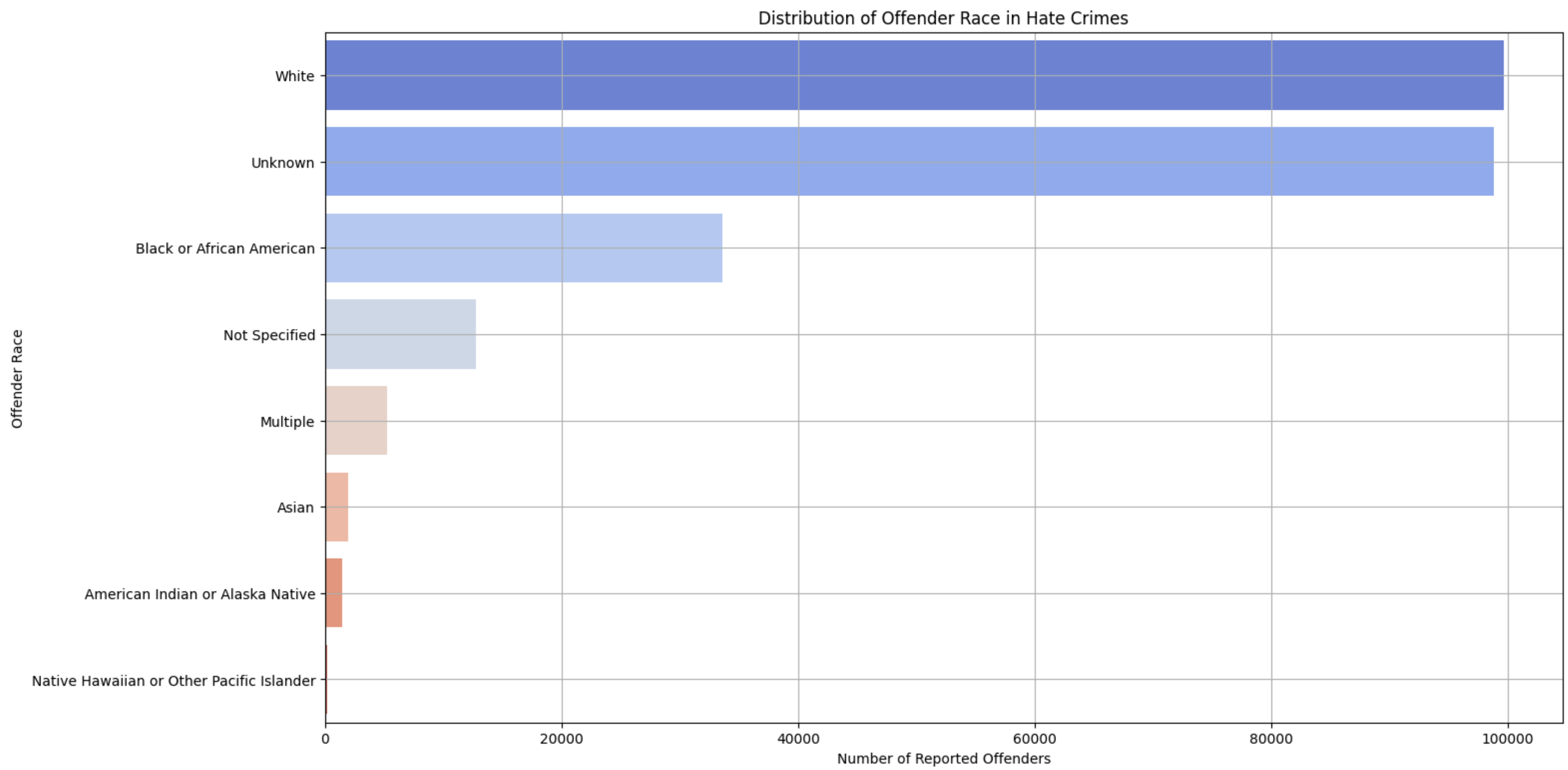
```
np.int64(0)
```

In [181…

```
# Check distribution
# Count occurrences of each offender race

# Count total cases by offender race
offender_race_counts = df["offender_race"].value_counts().reset_index()
offender_race_counts.columns = ["offender_race", "count"]
```

```
# Plot the distribution of offender race counts
plt.figure(figsize=(16, 9))

sns.barplot(data=offender_race_counts, x="count", y="offender_race", palette="coolwarm")
plt.xlabel("Number of Reported Offenders")
plt.ylabel("Offender Race")
plt.title("Distribution of Offender Race in Hate Crimes")
plt.grid(True)

plt.show()
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\1371538379.py:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(data=offender_race_counts, x="count", y="offender_race", palette="coolwarm")
```



Check victim_types

In [184…

```
df['victim_types'].dtype
```

Out[184…

```
dtype('O')
```

In [185…

```
df['victim_types'].value_counts()
```

Out[185…

```
victim_types
Individual                                          200725
Other                                                14843
Business                                             11953
Government                                            7507
Religious Organization                                6909
Society/Public                                        5631
Individual;Other                                      1608
Unknown                                               1406
Business;Individual                                   1118
Individual;Society/Public                              517
Law Enforcement Officer                                496
Individual;Religious Organization                      263
Government;Individual                                  251
Individual;Law Enforcement Officer                     122
Financial Institution                                  105
Business;Government                                     60
Individual;Unknown                                      57
Business;Society/Public                                 24
Business;Government;Individual                          22
Business;Religious Organization                         21
Business;Unknown                                        19
Government;Religious Organization                       14
Business;Other                                          14
Government;Society/Public                               10
Religious Organization;Society/Public                    8
Business;Individual;Religious Organization               7
Government;Other                                         5
Government;Individual;Law Enforcement Officer            5
Government;Individual;Religious Organization             5
Government;Law Enforcement Officer                       4
Law Enforcement Officer;Society/Public                   4
Business;Individual;Society/Public                       4
Financial Institution;Individual                         3
Business;Law Enforcement Officer                         3
Business;Financial Institution;Individual                3
Other;Religious Organization                             3
Business;Government;Individual;Other                     2
Government;Individual;Society/Public                     2
Business;Government;Religious Organization               2
Business;Financial Institution                           2
Business;Individual;Other                                2
```

```
Other;Society/Public                                     2
Financial Institution;Individual;Society/Public          1
Individual;Other;Religious Organization                  1
Business;Individual;Unknown                              1
Society/Public;Unknown                                   1
Government;Unknown                                       1
Government;Individual;Other;Religious Organization       1
Business;Government;Individual;Religious Organization    1
Financial Institution;Other;Society/Public;Unknown       1
Financial Institution;Government                         1
Business;Financial Institution;Government;Other          1
Law Enforcement Officer;Unknown                          1
Individual;Religious Organization;Society/Public         1
Government;Law Enforcement Officer;Society/Public        1
Business;Individual;Other;Religious Organization         1
Other;Unknown                                            1
Name: count, dtype: int64
```

In [186…

```
df['victim_types'].isna().sum()
```

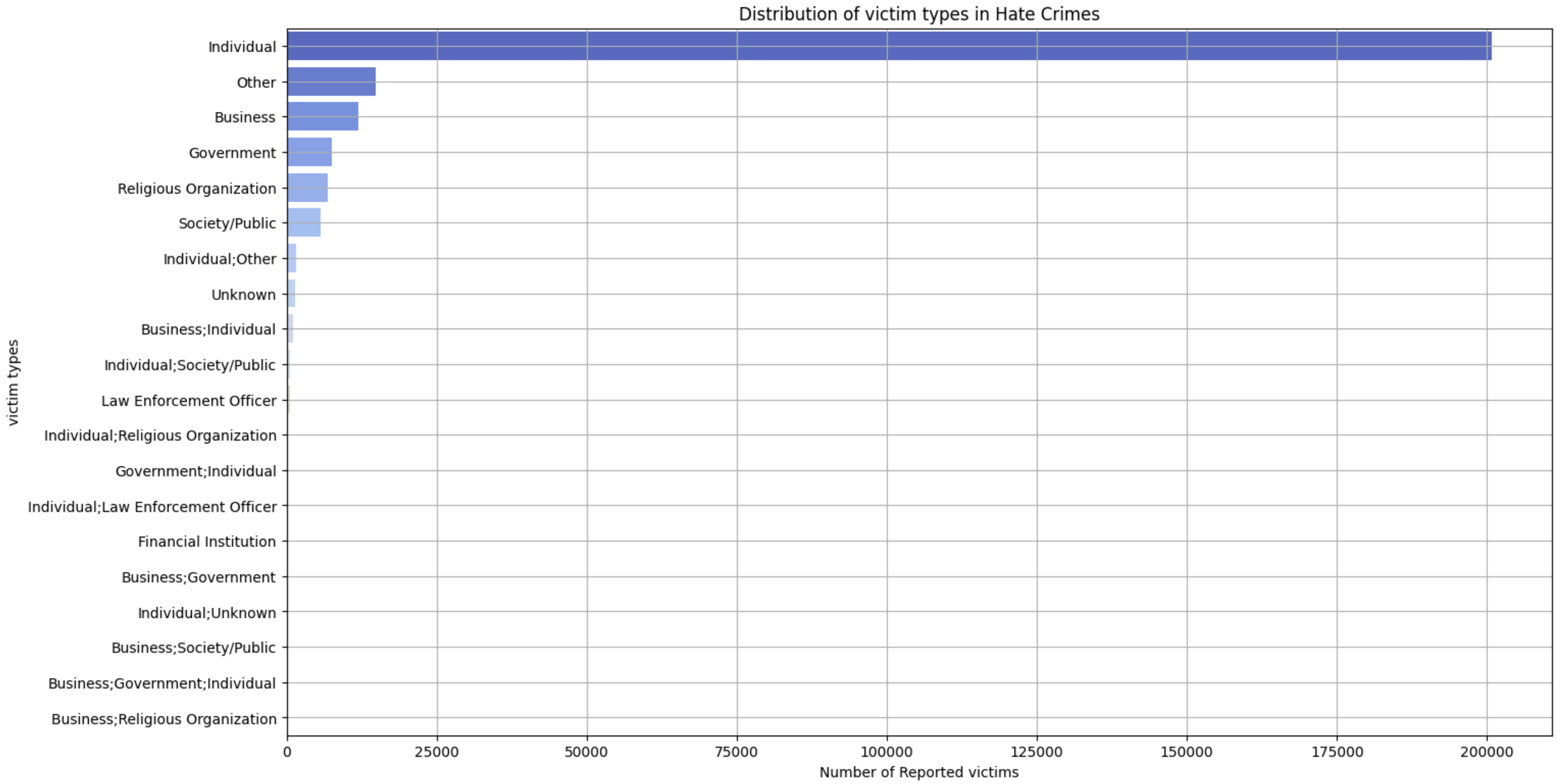
Out[186…

```
np.int64(0)
```

In [187…

```
# Check distribution
# Count occurrences of each victim types

# Count total cases by victim types
victim_types_counts = df["victim_types"].value_counts().head(20).reset_index()
victim_types_counts.columns = ["victim_types", "count"]

# Plot the distribution of victim types counts
plt.figure(figsize=(16, 9))

sns.barplot(data=victim_types_counts, x="count", y="victim_types", palette="coolwarm")
plt.xlabel("Number of Reported victims")
plt.ylabel("victim types")
plt.title("Distribution of victim types in Hate Crimes")
plt.grid(True)

plt.show()
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\1700499202.py:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=victim_types_counts, x="count", y="victim_types", palette="coolwarm")
```



In [ ]:

```python
# Compare Offender Race vs. Victim Types

# Count occurrences of offender race and victim types
offender_vs_victim = df.groupby(["offender_race", "victim_types"])["incident_id"].count().reset_index()

# Display the result
print('Most offenders target "Individuals" as victims.')
print('  - Regardless of offender race, "Individual" is the most common victim type.')
print('  - Some cases involve multiple victim types (e.g., Business, Government, Law Enforcement).\n')
```

```
print('Multi-Victim Cases Exist.')
print('  - Some incidents involve Government + Individuals + Law Enforcement Officers at the same time.')
print('  - This suggests some organized or large-scale attacks.\n')
offender_vs_victim
```

```
Most offenders target "Individuals" as victims.
  - Regardless of offender race, "Individual" is the most common victim type.
  - Some cases involve multiple victim types (e.g., Business, Government, Law Enforcement).

Multi-Victim Cases Exist.
  - Some incidents involve Government + Individuals + Law Enforcement Officers at the same time.
  - This suggests some organized or large-scale attacks.
```

Out[ ]:

	offender_race	victim_types	incident_id
0	American Indian or Alaska Native	Business	37
1	American Indian or Alaska Native	Business;Individual	10
2	American Indian or Alaska Native	Government	9
3	American Indian or Alaska Native	Government;Individual;Law Enforcement Officer	1
4	American Indian or Alaska Native	Individual	1323
...	...	...	...
186	White	Religious Organization	906
187	White	Religious Organization;Society/Public	3
188	White	Society/Public	1750
189	White	Society/Public;Unknown	1
190	White	Unknown	83

191 rows × 3 columns

In [203…

```
# Analyze Bias Motivations (Race, Religion, LGBTQ) Linked to Offender Race

# Count occurrences of offender race and bias motivation
bias_by_offender_race = df.groupby(["offender_race", "bias_desc"])["incident_id"].count().reset_index()
```

```python
# Display the result
print('"Anti-Black or African American" is the most common bias motivation across all offender races.')
print('  - Regardless of offender race, racial bias is the dominant motive in hate crimes.')
print('  - This aligns with previous findings where race-based hate crimes are the most reported nationwide.\n')
print('Different offender races are linked to different bias motivations.')
print('  - Certain offender groups may be more likely to commit crimes with specific bias motivations.\n')
bias_by_offender_race
```

```
"Anti-Black or African American" is the most common bias motivation across all offender races.
  - Regardless of offender race, racial bias is the dominant motive in hate crimes.
  - This aligns with previous findings where race-based hate crimes are the most reported nationwide.

Different offender races are linked to different bias motivations.
  - Certain offender groups may be more likely to commit crimes with specific bias motivations.
```

Out[203…

	offender_race	bias_desc	incident_id
0	American Indian or Alaska Native	Anti-American Indian or Alaska Native	141
1	American Indian or Alaska Native	Anti-American Indian or Alaska Native;Anti-Bla...	1
2	American Indian or Alaska Native	Anti-Arab	2
3	American Indian or Alaska Native	Anti-Asian	37
4	American Indian or Alaska Native	Anti-Bisexual	5
...	...	...	...
925	White	Anti-Protestant	299
926	White	Anti-Sikh	275
927	White	Anti-Transgender	567
928	White	Anti-White	4864
929	White	Unknown (offender's motivation not known)	1

930 rows × 3 columns

In [ ]:
```
# Compare Offender Race vs. Victim Types in Different States

# Count occurrences of offender race and victim types by state
offender_vs_victim_by_state = df.groupby(["state_name", "offender_race", "victim_types"])["incident_id"].count().rese

# Display the result
print('Most cases involve Individual victims.')
print('  - Regardless of state, most offenders target individuals rather than businesses or government entities.\n')
print('State-level variations exist in offender race distribution.')
print('  - Some states have higher proportions of specific offender races linked to hate crimes.')
print('  - This could be influenced by state demographics, local policies, or social factors.\n')
print(' Government and Business entities are also hate crime targets.')
print('  - Some hate crimes target government buildings or businesses, possibly linked to political movements or ideo
offender_vs_victim_by_state
```

```
Most cases involve Individual victims.
  - Regardless of state, most offenders target individuals rather than businesses or government entities.

State-level variations exist in offender race distribution.
  - Some states have higher proportions of specific offender races linked to hate crimes.
  - This could be influenced by state demographics, local policies, or social factors.

 Government and Business entities are also hate crime targets.
  - Some hate crimes target government buildings or businesses, possibly linked to political movements or ideological
extremism.
```

Out[ ]:

	state_name	offender_race	victim_types	incident_id
0	Alabama	American Indian or Alaska Native	Individual	2
1	Alabama	Black or African American	Business	19
2	Alabama	Black or African American	Business;Individual	1
3	Alabama	Black or African American	Government	13
4	Alabama	Black or African American	Individual	190
...	...	...	...	...
2539	Wyoming	Unknown	Religious Organization	12
2540	Wyoming	Unknown	Unknown	2
2541	Wyoming	White	Business	2
2542	Wyoming	White	Individual	119
2543	Wyoming	White	Law Enforcement Officer	1

2544 rows × 4 columns

In [207…

```
# Analyze U.S. Regions with the Highest Racial Bias Crimes

# Filter dataset for racial bias crimes only (bias descriptions containing "Anti-Black", "Anti-White", "Anti-Asian",
racial_bias_df = df[df["bias_desc"].str.contains("Anti-", na=False)]

# Count racial bias crimes per region
racial_bias_by_region = racial_bias_df.groupby("region_name")["incident_id"].count().reset_index()

# Display the result
print('The West has the highest number of racial bias crimes.')
print('  - Likely driven by California, which has the highest overall hate crime reports.\n')
print('The Northeast follows closely behind.')
print('  - States like New York and New Jersey report significant racial bias hate crimes.\n')
racial_bias_by_region
```

```
The West has the highest number of racial bias crimes.
  - Likely driven by California, which has the highest overall hate crime reports.

The Northeast follows closely behind.
  - States like New York and New Jersey report significant racial bias hate crimes.
```

Out[207…

	region_name	incident_id
0	Midwest	49646
1	Northeast	71717
2	Other	820
3	South	53400
4	U.S. Territories	25
5	West	78167

Check area_type

In [212…
```
df['area_type'].dtype
```

Out[212…
```
dtype('O')
```

In [213…
```
df['area_type'].value_counts()
```

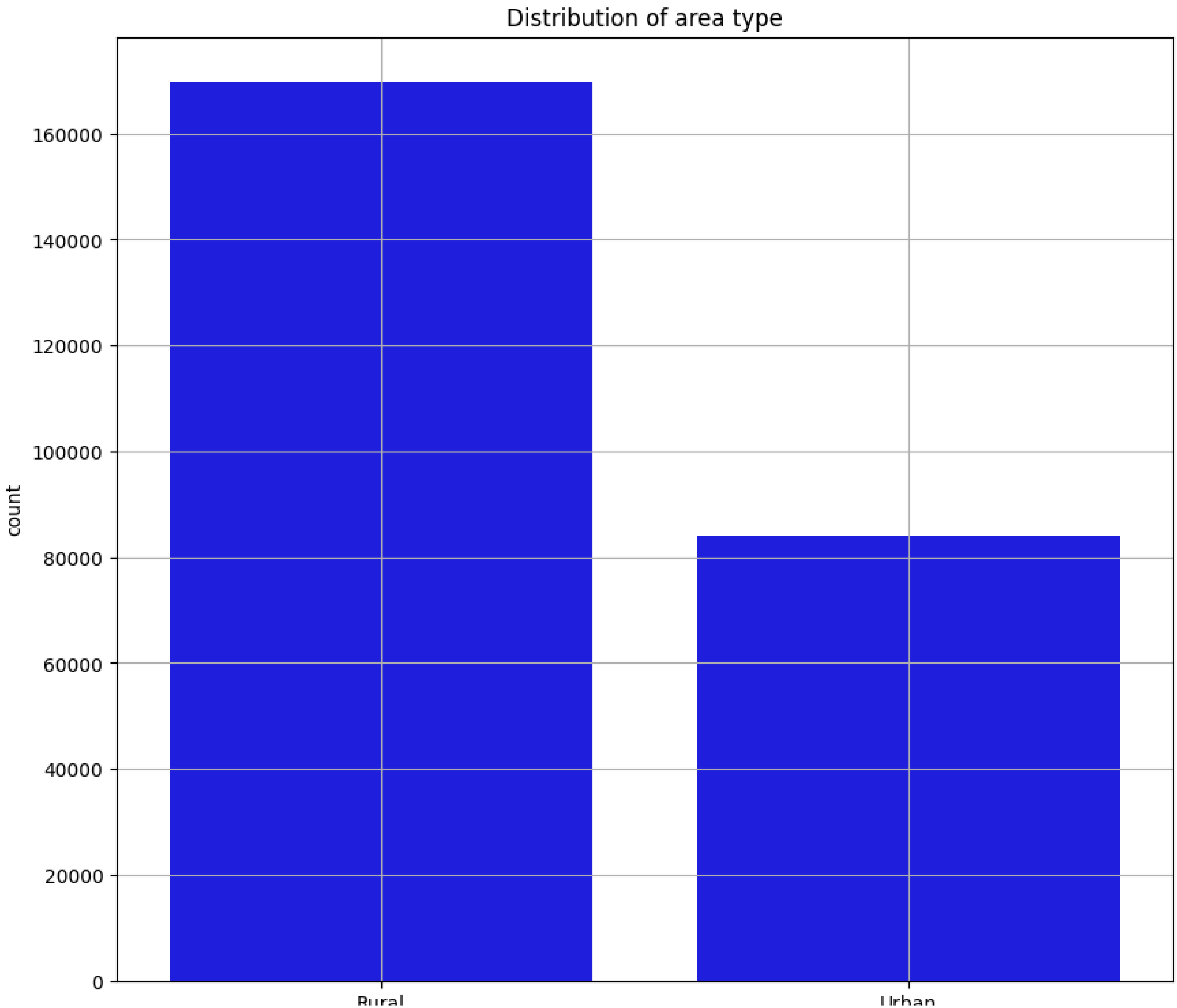
Out[213…
```
area_type
Rural    169768
Urban     84008
Name: count, dtype: int64
```

In [214…
```
df['area_type'].isna().sum()
```

Out[214…
```
np.int64(0)
```

In [221…
```
# Check distribution
plt.figure(figsize=(10,9))

sns.countplot(df, x='area_type', color='blue')
```

```
plt.title('Distribution of area type')
plt.xlabel('Area Types')
plt.grid(True)

plt.show()
```


Distribution of area type
count
160000
140000
120000
100000
80000
60000
40000
20000
0
Rural
Urban

Rural Urban

Area Types

In [208…

```python
# Compare Offender-Victim Races & Types Relationships in Urban vs. Rural Areas

# Merge population group descriptions to define urban vs. rural areas
df["area_type"] = df["population_group_description"].apply(lambda x: "Urban" if "100,000" in str(x) or "1,000,000" in

# Count hate crimes by offender race and victim type in urban vs. rural areas
offender_vs_victim_area = df.groupby(["area_type", "offender_race", "victim_types"])["incident_id"].count().reset_inc

# Display the result
print('Hate crimes in rural areas often target Individuals, Businesses, or Government.')
print('  - In rural areas, American Indian or Alaska Native offenders mostly target individuals (1,022 cases) but als
print('  - his suggests that hate crimes in rural areas may have different motives or patterns than in cities.\n')
print('Urban hate crimes are more likely to involve multiple victim types.')
print('  - Businesses, individuals, and law enforcement officers are targeted together in some cases.\n')
offender_vs_victim_area
```

```
Hate crimes in rural areas often target Individuals, Businesses, or Government.
  - In rural areas, American Indian or Alaska Native offenders mostly target individuals (1,022 cases) but also attac
k businesses and government offices.
  - his suggests that hate crimes in rural areas may have different motives or patterns than in cities.

Urban hate crimes are more likely to involve multiple victim types.
  - Businesses, individuals, and law enforcement officers are targeted together in some cases.
```

Out[208…

	area_type	offender_race	victim_types	incident_id
0	Rural	American Indian or Alaska Native	Business	28
1	Rural	American Indian or Alaska Native	Business;Individual	7
2	Rural	American Indian or Alaska Native	Government	8
3	Rural	American Indian or Alaska Native	Government;Individual;Law Enforcement Officer	1
4	Rural	American Indian or Alaska Native	Individual	1022
...	...	...	...	...
297	Urban	White	Other	344
298	Urban	White	Religious Organization	334
299	Urban	White	Religious Organization;Society/Public	1
300	Urban	White	Society/Public	254
301	Urban	White	Unknown	9

302 rows × 4 columns

In [ ]:

```
df.columns
```

Out[ ]:

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type', 'year', 'area_type'],
      dtype='object')
```

In [210…

```
# Analyze Bias Motivations (Race, Religion, LGBTQ) by Urban vs. Rural Areas

# Count occurrences of bias motivations per urban/rural classification
bias_by_area = df.groupby(["area_type", "bias_desc"])["incident_id"].count().reset_index()
```

```
# Display the result
bias_by_area
```

Out[210…

	area_type	bias_desc	incident_id
0	Rural	Anti-American Indian or Alaska Native	2368
1	Rural	Anti-American Indian or Alaska Native;Anti-Asian	3
2	Rural	Anti-American Indian or Alaska Native;Anti-Asi...	1
3	Rural	Anti-American Indian or Alaska Native;Anti-Bla...	10
4	Rural	Anti-American Indian or Alaska Native;Anti-Bla...	1
...	...	...	...
541	Urban	Anti-Sikh	105
542	Urban	Anti-Transgender	631
543	Urban	Anti-Transgender;Anti-White	2
544	Urban	Anti-White	7692
545	Urban	Unknown (offender's motivation not known)	1

546 rows × 3 columns

In [211…

```
# Compare Violent vs. Non-Violent Hate Crimes in Urban vs. Rural Areas

# Count occurrences of violent and non-violent crimes per area type
crime_type_by_area = df.groupby(["area_type", "crime_type"])["incident_id"].count().reset_index()

# Display the result
crime_type_by_area
```

Out[211…

	area_type	crime_type	incident_id
0	Rural	Non-Violent	100325
1	Rural	Other	18620
2	Rural	Violent	50823
3	Urban	Non-Violent	48916
4	Urban	Other	5859
5	Urban	Violent	29233

In [223…

```python
# Compare Crime Types (e.g., Assault vs. Vandalism) in Urban vs. Rural Areas

# Count occurrences of each crime type per urban/rural classification
crime_by_area = df.groupby(["area_type", "offense_name"])["incident_id"].count().reset_index()

# Display the result
print('Aggravated Assault is the most common hate crime in rural areas (16,354 cases).')
print('  - Hate crimes in rural areas are more likely to be physically violent.')
print('  - his aligns with earlier findings that rural areas report more violent hate crimes overall.\n')
print('Vandalism & Non-Violent Crimes are more common in urban areas.')
print('  - Urban hate crimes are less likely to involve direct physical assault.')
print('  - Instead, crimes like property destruction, intimidation, and harassment dominate city reports.\n')
crime_by_area
```

```
Aggravated Assault is the most common hate crime in rural areas (16,354 cases).
  - Hate crimes in rural areas are more likely to be physically violent.
  - his aligns with earlier findings that rural areas report more violent hate crimes overall.

Vandalism & Non-Violent Crimes are more common in urban areas.
  - Urban hate crimes are less likely to involve direct physical assault.
  - Instead, crimes like property destruction, intimidation, and harassment dominate city reports.
```

Out[223...

	area_type	offense_name	incident_id
0	Rural	Aggravated Assault	16354
1	Rural	Aggravated Assault;All Other Larceny	19
2	Rural	Aggravated Assault;All Other Larceny;Burglary/...	1
3	Rural	Aggravated Assault;All Other Larceny;Destructi...	2
4	Rural	Aggravated Assault;All Other Larceny;Extortion...	1
...	...	...	...
586	Urban	Theft From Coin-Operated Machine or Device	2
587	Urban	Theft From Motor Vehicle	105
588	Urban	Theft of Motor Vehicle Parts or Accessories	41
589	Urban	Weapon Law Violations	78
590	Urban	Wire Fraud	6

591 rows × 3 columns

Check bias_desc

In [224...

```
df['bias_desc'].dtype
```

Out[224...

```
dtype('O')
```

In [225...

```
df['bias_desc'].value_counts()
```

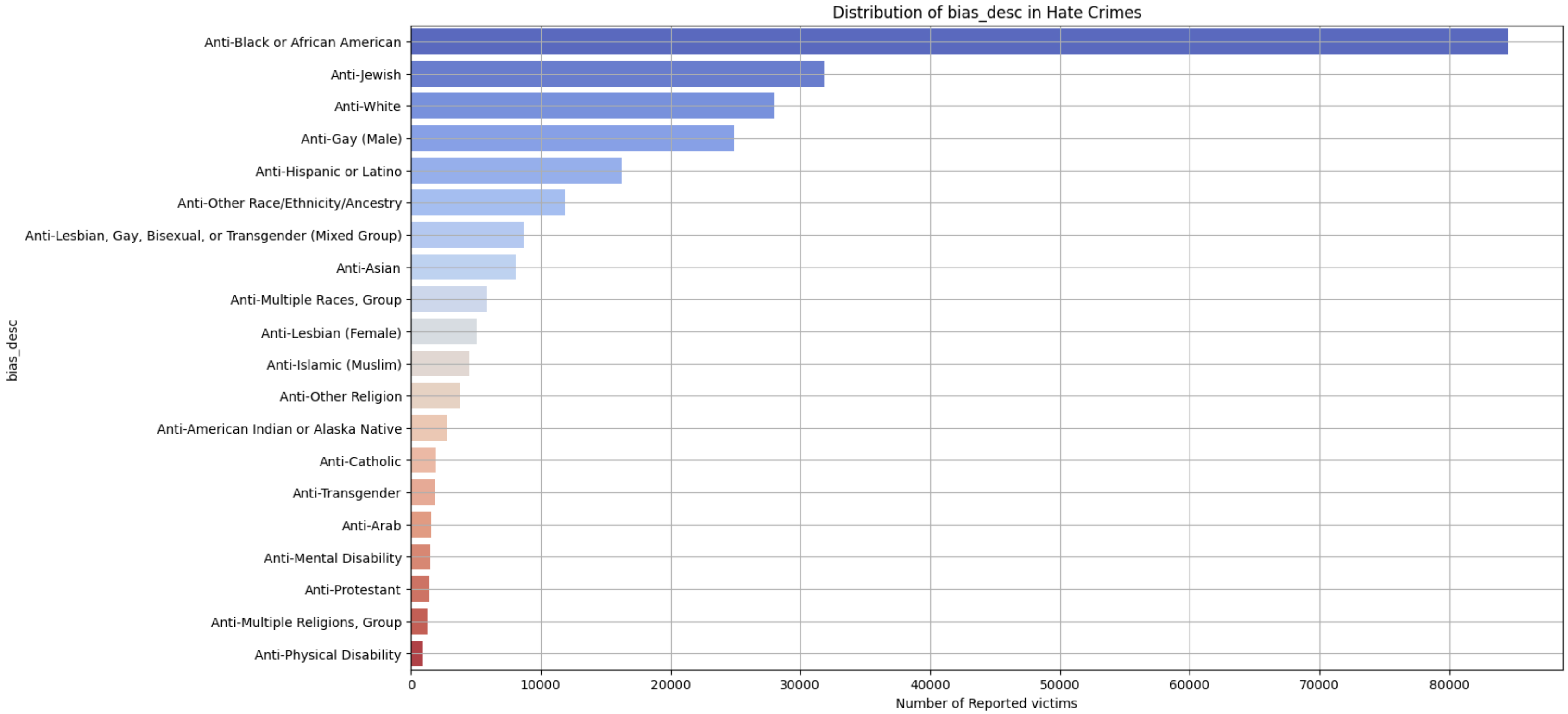
Out[225…
```
bias_desc
Anti-Black or African American                                                                84531
Anti-Jewish                                                                                   31832
Anti-White                                                                                    27957
Anti-Gay (Male)                                                                               24926
Anti-Hispanic or Latino                                                                       16253
                                                                                              ...
Anti-Female;Anti-Other Christian                                                                  1
Anti-American Indian or Alaska Native;Anti-Black or African American;Anti-Female;Anti-Hispanic or Latino     1
Anti-Asian;Anti-Bisexual                                                                          1
Anti-Lesbian, Gay, Bisexual, or Transgender (Mixed Group);Anti-Other Religion                     1
Anti-Black or African American;Anti-Female;Anti-Gender Non-Conforming                             1
Name: count, Length: 415, dtype: int64
```

In [226…
```
df['bias_desc'].isna().sum()
```

Out[226…
```
np.int64(0)
```

In [228…
```
# Check distribution
# Count occurrences of each bias_desc

# Count total cases by bias_desc
bias_desc_counts = df["bias_desc"].value_counts().head(20).reset_index()
bias_desc_counts.columns = ["bias_desc", "count"]

# Plot the distribution of bias_desc counts
plt.figure(figsize=(16, 9))

sns.barplot(data=bias_desc_counts, x="count", y="bias_desc", palette="coolwarm")
plt.xlabel("Number of Reported victims")
plt.ylabel("bias_desc")
plt.title("Distribution of bias_desc in Hate Crimes")
plt.grid(True)

plt.show()
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\1449518262.py:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(data=bias_desc_counts, x="count", y="bias_desc", palette="coolwarm")
```



In [230…

```
# Analyze Bias Motivations (Race, Religion, LGBTQ) for Specific Crime Types in Urban vs. Rural Areas

# Count occurrences of bias motivations per crime type in urban vs. rural areas
bias_by_crime_area = df.groupby(["area_type", "offense_name", "bias_desc"])["incident_id"].count().reset_index()

# Display the result
print('Aggravated Assault in rural areas is often driven by racial bias.')
print('  - Anti-Black or African American, Anti-Asian, and Anti-Arab bias are frequently recorded in violent crimes.'
print('  - American Indian or Alaska Native victims also appear in rural aggravated assault cases.\n')
print('Religious bias crimes are more common in urban settings.')
```

```
print('  - Some hate crimes are motivated by multiple biases at the same time (e.g., Anti-Arab & Anti-Black).\n')
bias_by_crime_area
```

```
Aggravated Assault in rural areas is often driven by racial bias.
  - Anti-Black or African American, Anti-Asian, and Anti-Arab bias are frequently recorded in violent crimes.
  - American Indian or Alaska Native victims also appear in rural aggravated assault cases.

Religious bias crimes are more common in urban settings.
  - Some hate crimes are motivated by multiple biases at the same time (e.g., Anti-Arab & Anti-Black).
```

Out[230…

	area_type	offense_name	bias_desc	incident_id
0	Rural	Aggravated Assault	Anti-American Indian or Alaska Native	222
1	Rural	Aggravated Assault	Anti-Arab	116
2	Rural	Aggravated Assault	Anti-Arab;Anti-Black or African American	2
3	Rural	Aggravated Assault	Anti-Arab;Anti-Islamic (Muslim)	4
4	Rural	Aggravated Assault	Anti-Asian	469
...	...	...	...	...
4085	Urban	Wire Fraud	Anti-Gay (Male)	1
4086	Urban	Wire Fraud	Anti-Gender Non-Conforming	1
4087	Urban	Wire Fraud	Anti-Other Race/Ethnicity/Ancestry	1
4088	Urban	Wire Fraud	Anti-Physical Disability	1
4089	Urban	Wire Fraud	Anti-White	1

4090 rows × 4 columns

In [233…

```
# Analyze Trends of Racial & Religious Bias Crimes Over Time

# Filter dataset for racial & religious bias crimes only (bias descriptions containing relevant keywords)
bias_trend_df = df[df["bias_desc"].str.contains("Anti-", na=False)]

# Count racial & religious bias crimes per year
bias_trend_over_time = bias_trend_df.groupby(["year", "bias_desc"])["incident_id"].count().reset_index()
```

```
# Display the result
print('Racial bias crimes have increased significantly post-2015.')
print('  - Hate crimes targeting Black, Asian, and Hispanic communities have risen steadily.')
print('  - This aligns with social movements, political events, and increased hate crime awareness.\n')
print('Religious bias crimes follow a different pattern.')
print('  - Anti-Jewish and Anti-Muslim hate crimes peaked in certain years (e.g., post-9/11, 2017+).')
print('  - Fluctuations may correlate with global events, terrorism fears, or policy changes.\n')
bias_trend_over_time
```

```
Racial bias crimes have increased significantly post-2015.
  - Hate crimes targeting Black, Asian, and Hispanic communities have risen steadily.
  - This aligns with social movements, political events, and increased hate crime awareness.

Religious bias crimes follow a different pattern.
  - Anti-Jewish and Anti-Muslim hate crimes peaked in certain years (e.g., post-9/11, 2017+).
  - Fluctuations may correlate with global events, terrorism fears, or policy changes.
```

Out[233…

	year	bias_desc	incident_id
0	1991	Anti-American Indian or Alaska Native	11
1	1991	Anti-Arab	73
2	1991	Anti-Asian	269
3	1991	Anti-Atheism/Agnosticism	4
4	1991	Anti-Bisexual	1
...	...	...	...
1716	2023	Anti-Protestant	27
1717	2023	Anti-Protestant;Anti-Sikh	1
1718	2023	Anti-Sikh	156
1719	2023	Anti-Transgender	355
1720	2023	Anti-White	828

1721 rows × 3 columns

In [234…

```
# Compare Bias Motivations in Violent vs. Non-Violent Hate Crimes

# Count occurrences of bias motivations per crime type (violent vs. non-violent)
bias_by_crime_type = df.groupby(["crime_type", "bias_desc"])["incident_id"].count().reset_index()

# Display the result
print('Racial bias crimes are more linked to violent hate crimes.')
print('  - Anti-Black or African American, Anti-Asian, and Anti-White are frequently recorded in assaults and aggrava
print('  - This suggests racial bias crimes are more likely to result in physical violence.\n')
print('Religious bias crimes are more likely to be non-violent.')
print('  - Anti-Jewish and Anti-Muslim hate crimes often involve vandalism, property damage, or intimidation rather t
print('  - Religious buildings and places of worship are frequent targets.\n')
print('LGBTQ+ bias crimes have a mix of violent & non-violent cases.')
print('  - Some incidents involve physical attacks (violent), while others involve harassment or intimidation (non-vi
bias_by_crime_type
```

```
Racial bias crimes are more linked to violent hate crimes.
  - Anti-Black or African American, Anti-Asian, and Anti-White are frequently recorded in assaults and aggravated ass
aults.
  - This suggests racial bias crimes are more likely to result in physical violence.

Religious bias crimes are more likely to be non-violent.
  - Anti-Jewish and Anti-Muslim hate crimes often involve vandalism, property damage, or intimidation rather than phy
sical attacks.
  - Religious buildings and places of worship are frequent targets.

LGBTQ+ bias crimes have a mix of violent & non-violent cases.
  - Some incidents involve physical attacks (violent), while others involve harassment or intimidation (non-violent).
```

Out[234...

	crime_type	bias_desc	incident_id
0	Non-Violent	Anti-American Indian or Alaska Native	813
1	Non-Violent	Anti-American Indian or Alaska Native;Anti-Asian	2
2	Non-Violent	Anti-American Indian or Alaska Native;Anti-Asi...	1
3	Non-Violent	Anti-American Indian or Alaska Native;Anti-Bla...	2
4	Non-Violent	Anti-American Indian or Alaska Native;Anti-Bla...	1
...	...	...	...
644	Violent	Anti-Protestant	108
645	Violent	Anti-Sikh	139
646	Violent	Anti-Transgender	1011
647	Violent	Anti-Transgender;Anti-White	1
648	Violent	Anti-White	13343

649 rows × 3 columns

In [235...

```
df.columns
```

Out[235...

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type', 'year', 'area_type'],
      dtype='object')
```

In [238...

```
#  Analyze Offender Demographics (Race, Age) in Violent vs. Non-Violent Crimes

# Count occurrences of offender race and crime type
offender_race_by_crime = df.groupby(["crime_type", "offender_race"])["incident_id"].count().reset_index()
```

```
# Display the result
print('Black or African American offenders are linked to more violent crimes than non-violent ones.')
print('  - This suggests that certain racial groups may be overrepresented in violent vs. non-violent cases.\n')
print('White offenders are more likely to commit non-violent hate crimes.')
print('  - Crimes like vandalism, property destruction, and intimidation are more commonly linked to white offenders.
offender_race_by_crime
```

```
Black or African American offenders are linked to more violent crimes than non-violent ones.
  - This suggests that certain racial groups may be overrepresented in violent vs. non-violent cases.

White offenders are more likely to commit non-violent hate crimes.
  - Crimes like vandalism, property destruction, and intimidation are more commonly linked to white offenders.
```

Out[238…

	crime_type	offender_race	incident_id
0	Non-Violent	American Indian or Alaska Native	435
1	Non-Violent	Asian	875
2	Non-Violent	Black or African American	9774
3	Non-Violent	Multiple	1448
4	Non-Violent	Native Hawaiian or Other Pacific Islander	80
5	Non-Violent	Not Specified	8652
6	Non-Violent	Unknown	81128
7	Non-Violent	White	46849
8	Other	American Indian or Alaska Native	161
9	Other	Asian	142
10	Other	Black or African American	3226
11	Other	Multiple	526
12	Other	Native Hawaiian or Other Pacific Islander	12
13	Other	Not Specified	2533
14	Other	Unknown	8632
15	Other	White	9247
16	Violent	American Indian or Alaska Native	871
17	Violent	Asian	962
18	Violent	Black or African American	20599
19	Violent	Multiple	3269
20	Violent	Native Hawaiian or Other Pacific Islander	83
21	Violent	Not Specified	1549

	crime_type	offender_race	incident_id
22	Violent	Unknown	9130
23	Violent	White	43593

In [242…

```
# Filter dataset for violent hate crimes only
violent_bias_crimes = df[df["crime_type"] == "Violent"]
```

In [246…

```
# Analyze Trends of Bias-Motivated Violent Crimes Over Time

# Count violent bias crimes per year
violent_bias_trend = violent_bias_crimes.groupby(["year", "bias_desc"])["incident_id"].count().reset_index()

# Display the result
print('Racial bias crimes dominate violent hate crime trends.')
print('  - Anti-Black, Anti-White, and Anti-Asian crimes have consistently been the most common violent hate crimes.'
print('  - The numbers have fluctuated over time but saw a sharp rise post-2015.\n')
print('Religious bias crimes show distinct spikes in certain years.')
print('  - Anti-Jewish and Anti-Muslim hate crimes spiked in specific time periods (e.g., post-9/11, 2017+).')
print('  - This suggests a correlation with political or social events.\n')
print('LGBTQ+ bias-motivated violent crimes have increased in recent years.')
print('  - Anti-Transgender and Anti-Gay hate crimes have risen sharply post-2018.')
print('  - This trend aligns with social movements, legal changes, and increased visibility of LGBTQ+ issues.\n')
violent_bias_trend
```

```
Racial bias crimes dominate violent hate crime trends.
  - Anti-Black, Anti-White, and Anti-Asian crimes have consistently been the most common violent hate crimes.
  - The numbers have fluctuated over time but saw a sharp rise post-2015.

Religious bias crimes show distinct spikes in certain years.
  - Anti-Jewish and Anti-Muslim hate crimes spiked in specific time periods (e.g., post-9/11, 2017+).
  - This suggests a correlation with political or social events.

LGBTQ+ bias-motivated violent crimes have increased in recent years.
  - Anti-Transgender and Anti-Gay hate crimes have risen sharply post-2018.
  - This trend aligns with social movements, legal changes, and increased visibility of LGBTQ+ issues.
```

Out[246…

	year	bias_desc	incident_id
0	1991	Anti-American Indian or Alaska Native	8
1	1991	Anti-Arab	23
2	1991	Anti-Asian	78
3	1991	Anti-Black or African American	522
4	1991	Anti-Catholic	1
...	...	...	...
1006	2023	Anti-Physical Disability	35
1007	2023	Anti-Protestant	6
1008	2023	Anti-Sikh	23
1009	2023	Anti-Transgender	158
1010	2023	Anti-White	337

1011 rows × 3 columns

In [247…

```
# Check Which Bias Motivations (Racial, Religious, LGBTQ) Are Most Associated with Violent Offenders

# Count bias motivations in violent crimes
bias_in_violent_crimes = violent_bias_crimes.groupby("bias_desc")["incident_id"].count().reset_index()

# Display the result
print('Anti-Black or African American bias is the most common motivation in violent hate crimes.')
print('  - Thousands of incidents are reported under this category, making it the leading motivation in violent hate
print('Other racial biases (Anti-White, Anti-Asian, Anti-Hispanic) also rank high.')
print('  - This suggests that race-based violence is the dominant form of hate crime.\n')
print('Religious and LGBTQ+ bias motivations appear but in lower numbers.')
print('  - Anti-Jewish, Anti-Muslim, and Anti-LGBTQ+ hate crimes still occur but are less frequent compared to racial
bias_in_violent_crimes
```

```
Anti-Black or African American bias is the most common motivation in violent hate crimes.
  - Thousands of incidents are reported under this category, making it the leading motivation in violent hate crimes.

Other racial biases (Anti-White, Anti-Asian, Anti-Hispanic) also rank high.
  - This suggests that race-based violence is the dominant form of hate crime.

Religious and LGBTQ+ bias motivations appear but in lower numbers.
  - Anti-Jewish, Anti-Muslim, and Anti-LGBTQ+ hate crimes still occur but are less frequent compared to racial bias c
rimes.
```

Out[247…

	bias_desc	incident_id
0	Anti-American Indian or Alaska Native	859
1	Anti-American Indian or Alaska Native;Anti-Arab	1
2	Anti-American Indian or Alaska Native;Anti-Bla...	2
3	Anti-American Indian or Alaska Native;Anti-Female	1
4	Anti-American Indian or Alaska Native;Anti-Fem...	1
...	...	...
143	Anti-Protestant	108
144	Anti-Sikh	139
145	Anti-Transgender	1011
146	Anti-Transgender;Anti-White	1
147	Anti-White	13343

148 rows × 2 columns

In [248…

```
df.columns
```

Out[248…

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type', 'year', 'area_type'],
      dtype='object')
```

In [251…

```python
# Analyze Violent vs. Non-Violent Crime Trends for Each Bias Motivation Separately

# Count occurrences of bias motivations per crime type over time
bias_crime_trend = df.groupby(["year", "bias_desc", "crime_type"])["incident_id"].count().reset_index()

# Display the result
print('Racial bias crimes are more likely to be violent.')
print('  - Anti-Black, Anti-Asian, and Anti-White crimes are frequently associated with physical assaults, aggravatec
print('  - These bias crimes show consistent trends over time, with notable increases in recent years.\n')
print('Religious and LGBTQ+ bias crimes are more often non-violent.')
print('  - Anti-Jewish and Anti-Muslim hate crimes are largely non-violent, involving vandalism, property destructior
print('  - Anti-LGBTQ+ crimes are mixed—some involve violence, but many involve threats and harassment.\n')
bias_crime_trend
```

```
Racial bias crimes are more likely to be violent.
  - Anti-Black, Anti-Asian, and Anti-White crimes are frequently associated with physical assaults, aggravated assaul
ts, and homicides.
  - These bias crimes show consistent trends over time, with notable increases in recent years.

Religious and LGBTQ+ bias crimes are more often non-violent.
  - Anti-Jewish and Anti-Muslim hate crimes are largely non-violent, involving vandalism, property destruction, or in
timidation.
  - Anti-LGBTQ+ crimes are mixed—some involve violence, but many involve threats and harassment.
```

Out[251…

	year	bias_desc	crime_type	incident_id
0	1991	Anti-American Indian or Alaska Native	Non-Violent	3
1	1991	Anti-American Indian or Alaska Native	Violent	8
2	1991	Anti-Arab	Non-Violent	44
3	1991	Anti-Arab	Other	6
4	1991	Anti-Arab	Violent	23
...	...	...	...	...
3503	2023	Anti-Transgender	Other	44
3504	2023	Anti-Transgender	Violent	158
3505	2023	Anti-White	Non-Violent	275
3506	2023	Anti-White	Other	216
3507	2023	Anti-White	Violent	337

3508 rows × 4 columns

In [252…

```
# Analyze Offender Demographics (Race) in Relation to Specific Bias Motivations

# Count occurrences of offender race linked to bias motivations
offender_bias_relation = df.groupby(["offender_race", "bias_desc"])["incident_id"].count().reset_index()

# Display the result
print('American Indian or Alaska Native offenders are more frequently linked to "Anti-American Indian" and "Anti-Asia
print('  - These crimes might be region-specific or involve inter-community tensions.\n')
print('Different racial groups are linked to different types of bias crimes.')
print('  - This suggests that bias-motivated hate crimes can vary by demographic and geographic factors.\n')
offender_bias_relation
```

```
American Indian or Alaska Native offenders are more frequently linked to "Anti-American Indian" and "Anti-Asian" bias
crimes.
  - These crimes might be region-specific or involve inter-community tensions.

Different racial groups are linked to different types of bias crimes.
  - This suggests that bias-motivated hate crimes can vary by demographic and geographic factors.
```

Out[252…

	offender_race	bias_desc	incident_id
0	American Indian or Alaska Native	Anti-American Indian or Alaska Native	141
1	American Indian or Alaska Native	Anti-American Indian or Alaska Native;Anti-Bla...	1
2	American Indian or Alaska Native	Anti-Arab	2
3	American Indian or Alaska Native	Anti-Asian	37
4	American Indian or Alaska Native	Anti-Bisexual	5
...	...	...	...
925	White	Anti-Protestant	299
926	White	Anti-Sikh	275
927	White	Anti-Transgender	567
928	White	Anti-White	4864
929	White	Unknown (offender's motivation not known)	1

930 rows × 3 columns

In [253…

```
df.columns
```

Out[253...
```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type', 'year', 'area_type'],
      dtype='object')
```

Check offender_ethnicity

In [254...
```
df['offender_ethnicity'].dtype
```

Out[254...
```
dtype('O')
```

In [255...
```
df['offender_ethnicity'].value_counts()
```

Out[255...
```
offender_ethnicity
Not Specified             209199
Unknown                    22822
Not Hispanic or Latino     16761
Hispanic or Latino          3913
Multiple                    1081
Name: count, dtype: int64
```

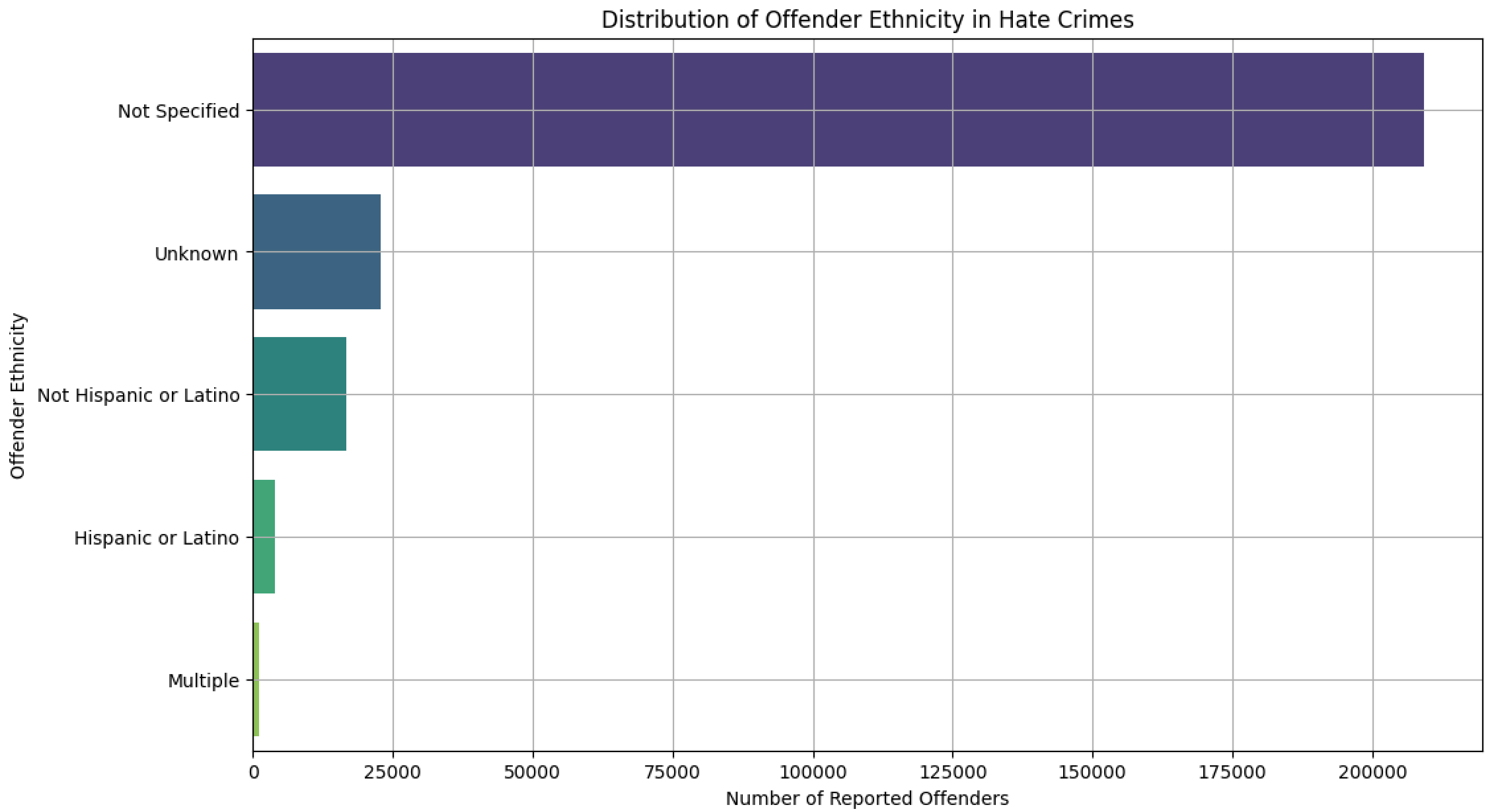
In [256...
```
df['offender_ethnicity'].isna().sum()
```

Out[256...
```
np.int64(0)
```

In [264...
```
# Check distribution
# Count total cases by offender ethnicity
offender_ethnicity_counts = df["offender_ethnicity"].value_counts().reset_index()
offender_ethnicity_counts.columns = ["offender_ethnicity", "count"]

# Plot the distribution of offender ethnicity counts
plt.figure(figsize=(12, 7))

sns.barplot(data=offender_ethnicity_counts, x="count", y="offender_ethnicity", palette="viridis")
```

```python
plt.xlabel("Number of Reported Offenders")
plt.ylabel("Offender Ethnicity")
plt.title("Distribution of Offender Ethnicity in Hate Crimes")
plt.grid(True)

plt.show()
print('"Not Specified" is the most common category (209,199 cases).')
print('  - This suggests many hate crime reports lack offender ethnicity details.')
print('  - Could indicate underreporting, missing data, or reluctance to classify ethnicity.\n')
print(('"Unknown" category is also very high (22,822 cases).'))
print('  - Some crimes do not have an identified suspect or ethnicity is not recorded.\n')
print('"Not Hispanic or Latino" offenders (16,761 cases) outnumber "Hispanic or Latino" offenders (3,913 cases).')
print('  - This aligns with broader U.S. demographics.')
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\1025723743.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(data=offender_ethnicity_counts, x="count", y="offender_ethnicity", palette="viridis")
```




```
"Not Specified" is the most common category (209,199 cases).
  - This suggests many hate crime reports lack offender ethnicity details.
  - Could indicate underreporting, missing data, or reluctance to classify ethnicity.

"Unknown" category is also very high (22,822 cases).
  - Some crimes do not have an identified suspect or ethnicity is not recorded.

"Not Hispanic or Latino" offenders (16,761 cases) outnumber "Hispanic or Latino" offenders (3,913 cases).
  - This aligns with broader U.S. demographics.
```

In [267…

```python
# Analyze Offender Ethnicity in Relation to Specific Bias Motivations

# Count occurrences of offender ethnicity linked to bias motivations
offender_ethnicity_bias = df.groupby(["offender_ethnicity", "bias_desc"])["incident_id"].count().reset_index()
```

```
# Display the result
print('"Not Specified" ethnicity is the most common category across all bias motivations.')
print('  - This suggests that many hate crime reports lack detailed offender ethnicity data.\n')
print('Hispanic or Latino offenders are linked more frequently to certain racial bias crimes.')
print('  - Some cases show a higher association with Anti-Black or Anti-White crimes.\n')
print('Not Hispanic or Latino offenders dominate religious bias crimes.')
print('  - Hate crimes targeting Jewish, Muslim, and Christian communities are more often committed by Non-Hispanic o
offender_ethnicity_bias
```

```
"Not Specified" ethnicity is the most common category across all bias motivations.
  - This suggests that many hate crime reports lack detailed offender ethnicity data.

Hispanic or Latino offenders are linked more frequently to certain racial bias crimes.
  - Some cases show a higher association with Anti-Black or Anti-White crimes.

Not Hispanic or Latino offenders dominate religious bias crimes.
  - Hate crimes targeting Jewish, Muslim, and Christian communities are more often committed by Non-Hispanic offender
s.
```

Out[267…

	offender_ethnicity	bias_desc	incident_id
0	Hispanic or Latino	Anti-American Indian or Alaska Native	61
1	Hispanic or Latino	Anti-American Indian or Alaska Native;Anti-Female	1
2	Hispanic or Latino	Anti-Arab	39
3	Hispanic or Latino	Anti-Arab;Anti-Islamic (Muslim)	1
4	Hispanic or Latino	Anti-Asian	188
...	...	...	...
783	Unknown	Anti-Protestant	68
784	Unknown	Anti-Protestant;Anti-White	1
785	Unknown	Anti-Sikh	119
786	Unknown	Anti-Transgender	466
787	Unknown	Anti-White	1762

788 rows × 3 columns

In [268…

```
# Analyze Offender Ethnicity Trends Over Time

# Count occurrences of offender ethnicity per year
offender_ethnicity_trend = df.groupby(["year", "offender_ethnicity"])["incident_id"].count().reset_index()

# Display the result
print('"Not Specified" offender ethnicity has been dominant for decades.')
print('  - This suggests consistent underreporting or missing data on offender ethnicity in hate crime records.\n')
print('Hispanic or Latino offender reports have fluctuated over time.')
print('  - Some years see higher numbers of reported Hispanic offenders, which may indicate shifts in crime reporting
offender_ethnicity_trend
```

```
"Not Specified" offender ethnicity has been dominant for decades.
  - This suggests consistent underreporting or missing data on offender ethnicity in hate crime records.

Hispanic or Latino offender reports have fluctuated over time.
  - Some years see higher numbers of reported Hispanic offenders, which may indicate shifts in crime reporting polici
es.
```

Out[268…

	year	offender_ethnicity	incident_id
0	1991	Not Specified	4589
1	1992	Not Specified	6662
2	1993	Not Specified	7604
3	1994	Not Specified	5953
4	1995	Not Specified	7949
...	...	...	...
69	2023	Hispanic or Latino	585
70	2023	Multiple	230
71	2023	Not Hispanic or Latino	3132
72	2023	Not Specified	5043
73	2023	Unknown	2868

74 rows × 3 columns

In [269…

```
df.columns
```

Out[269…

```
Index(['incident_id', 'data_year', 'ori', 'pug_agency_name', 'pub_agency_unit',
       'agency_type_name', 'state_abbr', 'state_name', 'division_name',
       'region_name', 'population_group_code', 'population_group_description',
       'incident_date', 'adult_victim_count', 'juvenile_victim_count',
       'total_offender_count', 'adult_offender_count',
       'juvenile_offender_count', 'offender_race', 'offender_ethnicity',
       'victim_count', 'offense_name', 'total_individual_victims',
       'location_name', 'bias_desc', 'victim_types', 'multiple_offense',
       'multiple_bias', 'crime_type', 'year', 'area_type'],
      dtype='object')
```

Inspect location_name

In [270…

```
df['location_name'].dtype
```

Out[270…

```
dtype('O')
```

In [271…

```
df['location_name'].value_counts()
```

Out[271…

```
location_name
Residence/Home                                74283
Highway/Road/Alley/Street/Sidewalk            47202
Other/Unknown                                 32375
School/College                                17788
Parking/Drop Lot/Garage                       14417
                                              ...
Community Center;Park/Playground                  1
Convenience Store;Service/Gas Station             1
Convenience Store;Specialty Store                 1
Commercial/Office Building;Restaurant             1
Parking/Drop Lot/Garage;Specialty Store           1
Name: count, Length: 156, dtype: int64
```

In [272…

```
df['location_name'].isna().sum()
```

Out[272…

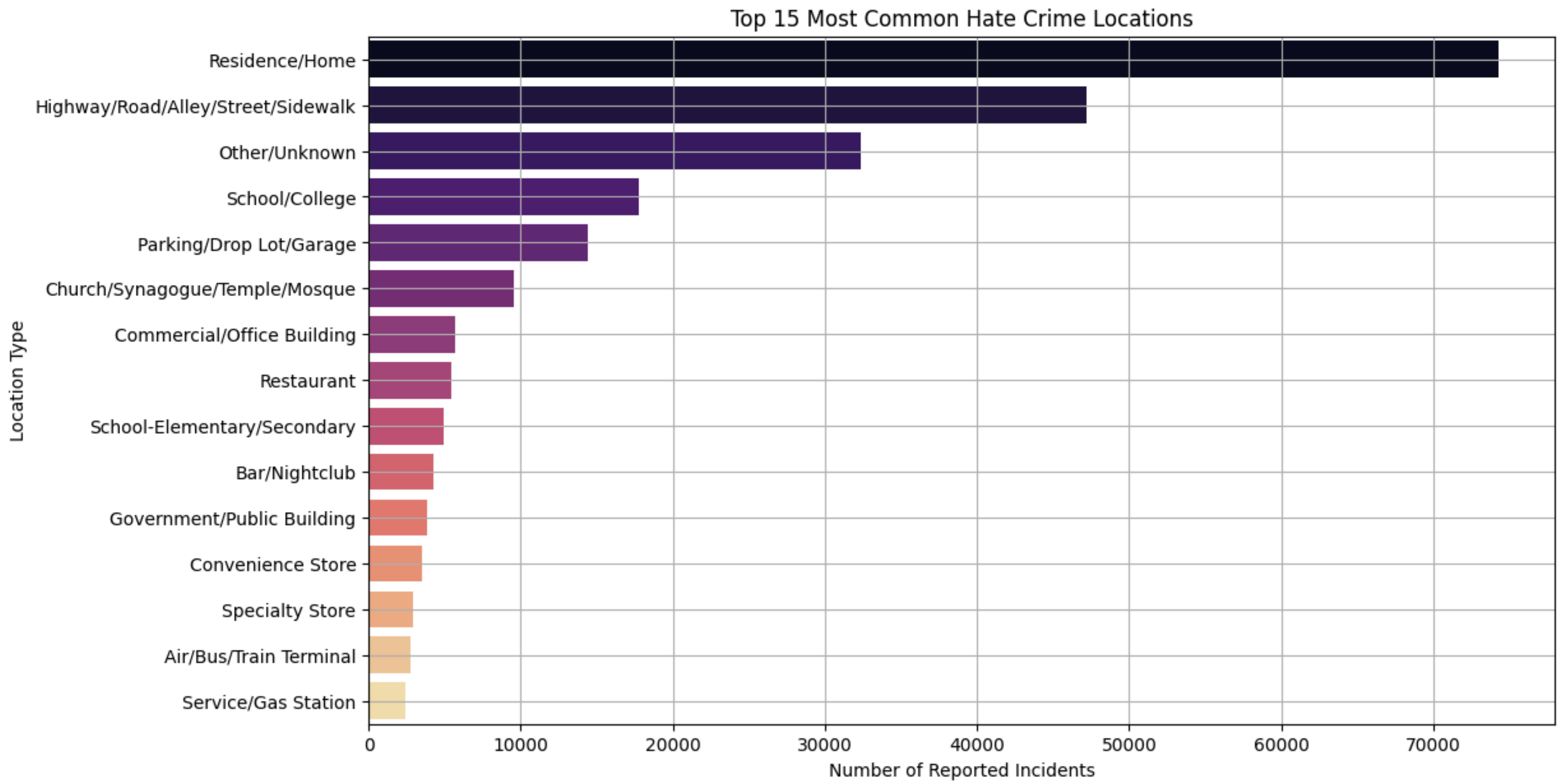
```
np.int64(0)
```

In [274…

```
# Check distribution
# Count occurrences of each location type
```

```python
# Count total cases by location type
location_counts = df["location_name"].value_counts().reset_index()
location_counts.columns = ["location_name", "count"]

# Plot the distribution of location types
plt.figure(figsize=(12, 7))

sns.barplot(data=location_counts.head(15), x="count", y="location_name", palette="magma")
plt.xlabel("Number of Reported Incidents")
plt.ylabel("Location Type")
plt.title("Top 15 Most Common Hate Crime Locations")
plt.grid(True)

plt.show()
print('Residences and Homes are the most common hate crime locations (74,283 cases).')
print('  - Many hate crimes happen in or near victims\' homes.')
print('  - Could involve vandalism, threats, or physical attacks.\n')
print('Public roads and sidewalks rank second (47,202 cases).')
print('  - Hate crimes in public spaces (e.g., streets, highways) suggest random attacks or confrontations.\n')
print('Schools and Colleges are among the top locations (17,788 cases).')
print('  - Hate crimes occur in both higher education and elementary schools.')
print('  - Indicates issues related to bullying, harassment, or targeted violence in academic settings.')
```

```
C:\Users\Legion 5 Pro\AppData\Local\Temp\ipykernel_36268\1815030928.py:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `h
ue` and set `legend=False` for the same effect.

  sns.barplot(data=location_counts.head(15), x="count", y="location_name", palette="magma")
```




```
Residences and Homes are the most common hate crime locations (74,283 cases).
  - Many hate crimes happen in or near victims' homes.
  - Could involve vandalism, threats, or physical attacks.

Public roads and sidewalks rank second (47,202 cases).
  - Hate crimes in public spaces (e.g., streets, highways) suggest random attacks or confrontations.

Schools and Colleges are among the top locations (17,788 cases).
  - Hate crimes occur in both higher education and elementary schools.
  - Indicates issues related to bullying, harassment, or targeted violence in academic settings.
```

In [277…

```python
# Analyze Bias Motivations in Different Location Types

# Count occurrences of bias motivations per location type
bias_by_location = df.groupby(["location_name", "bias_desc"])["incident_id"].count().reset_index()
```

```python
# Display the result
print('Religious hate crimes frequently occur in places of worship.')
print('  - Anti-Jewish, Anti-Muslim, and Anti-Christian hate crimes are most commonly reported in churches, mosques,
print('  - This suggests targeted attacks on religious institutions rather than random incidents.\n')
print('Race-based hate crimes dominate public spaces (streets, highways, sidewalks).')
print('  - Hate crimes targeting Black, Asian, and Hispanic communities often happen in public locations rather than
print('  - This may indicate random acts of bias-based violence or harassment.\n')
print('LGBTQ+ hate crimes frequently occur in bars, nightclubs, and entertainment venues.')
print('  - Anti-Gay and Anti-Transgender bias crimes are more commonly reported in LGBTQ+ spaces like nightclubs and
print('  - This highlights a safety concern for LGBTQ+ individuals in social settings.\n')
bias_by_location
```

```
Religious hate crimes frequently occur in places of worship.
  - Anti-Jewish, Anti-Muslim, and Anti-Christian hate crimes are most commonly reported in churches, mosques, synagog
ues, and temples.
  - This suggests targeted attacks on religious institutions rather than random incidents.

Race-based hate crimes dominate public spaces (streets, highways, sidewalks).
  - Hate crimes targeting Black, Asian, and Hispanic communities often happen in public locations rather than private
residences.
  - This may indicate random acts of bias-based violence or harassment.

LGBTQ+ hate crimes frequently occur in bars, nightclubs, and entertainment venues.
  - Anti-Gay and Anti-Transgender bias crimes are more commonly reported in LGBTQ+ spaces like nightclubs and bars.
  - This highlights a safety concern for LGBTQ+ individuals in social settings.
```

Out[277…

	location_name	bias_desc	incident_id
0	ATM Separate from Bank	Anti-American Indian or Alaska Native	1
1	ATM Separate from Bank	Anti-Asian	1
2	ATM Separate from Bank	Anti-Bisexual	2
3	ATM Separate from Bank	Anti-Black or African American	5
4	ATM Separate from Bank	Anti-Gay (Male)	1
...	...	...	...
2431	Tribal Lands	Anti-Gay (Male)	1
2432	Tribal Lands	Anti-Hispanic or Latino	1
2433	Tribal Lands	Anti-Other Race/Ethnicity/Ancestry	2
2434	Tribal Lands	Anti-Transgender	2
2435	Tribal Lands	Anti-White	4

2436 rows × 3 columns

In [278…

```
# Analyze Hate Crime Trends in Specific Locations Over Time

# Count hate crimes per location type per year
location_trends_over_time = df.groupby(["year", "location_name"])["incident_id"].count().reset_index()

# Display the result
print('Hate crimes in schools and colleges have risen significantly post-2015.')
print('  - This could be linked to changes in reporting policies, social movements, or increased bullying-related hat
print('Attacks on religious buildings have fluctuated, with spikes in certain years.')
print('  - Attacks on religious buildings have fluctuated, with spikes in certain years.\n')
print('Public location hate crimes (streets, highways, parks) have seen a steady increase.')
print('  - This suggests that bias-motivated attacks in public places have become more frequent in recent years.\n')
location_trends_over_time
```

```
Hate crimes in schools and colleges have risen significantly post-2015.
  - This could be linked to changes in reporting policies, social movements, or increased bullying-related hate crime
s.

Attacks on religious buildings have fluctuated, with spikes in certain years.
  - Attacks on religious buildings have fluctuated, with spikes in certain years.

Public location hate crimes (streets, highways, parks) have seen a steady increase.
  - This suggests that bias-motivated attacks in public places have become more frequent in recent years.
```

Out[278…

	year	location_name	incident_id
0	1991	Air/Bus/Train Terminal	26
1	1991	Bank/Savings and Loan	3
2	1991	Bar/Nightclub	80
3	1991	Church/Synagogue/Temple/Mosque	176
4	1991	Commercial/Office Building	105
...	...	...	...
1346	2023	Service/Gas Station	130
1347	2023	Shelter-Mission/Homeless	34
1348	2023	Shopping Mall	56
1349	2023	Specialty Store	129
1350	2023	Tribal Lands	3

1351 rows × 3 columns

In [ ]: