DIC Project Report (Phase 1)

1. **Akanksh Gatla (akankshg) 50465101**
2. **Gnana Abhinay Vadlamudi (gnanaabh) 50496402**
3. **Sai Teja Chittumalla (schittum) 50496091**

a. Discuss the background of the problem leading to your objectives.
Why is it a significant problem?

Background:
Crime is a pervasive social issue with far-reaching implications for the safety and well-being of communities. Analyzing crime data allows us to gain insights into the evolution of criminal activity, understand its regional disparities, and potentially inform policy decisions. The Unified Crime Reporting Statistics, compiled by the U.S. Department of Justice and the Federal Bureau of Investigation, provides a comprehensive dataset covering a wide range of years (1960 to 2019) and various crime categories.

Property crime, including burglary, larceny, and motor-related crimes, poses a significant financial and emotional burden on victims and communities. Violent crimes, such as assault, murder, rape, and robbery, endanger lives and impact the overall security of areas. Therefore, comprehensively studying these crime categories can provide valuable insights into their nature, causes, and possible preventive measures.

b. Explain the potential of your project to contribute to your problem domain. Discuss why this contribution is crucial?

The significance of this problem lies in its potential to address several critical issues:

1. Crime Prevention: By understanding the historical trends in crime rates, we can identify high-risk areas and periods, enabling law enforcement agencies to allocate resources more effectively.

2. Policy Formulation: Policymakers can use this analysis to develop evidence-based strategies to combat crime, ultimately creating safer communities.

3. Social Impact: Reducing crime rates can lead to improved quality of life, economic growth, and well-being for residents.

4. Community Awareness: Public access to crime data fosters transparency and awareness, empowering individuals to make informed decisions about their safety and contribute to crime prevention.

Things Can be done using this Dataset :

- **Predictive Analytics**: Crime data analysis can serve as a foundation for developing predictive models. Understanding historical trends in property and violent crimes can help data scientists

create models that predict future crime rates, aiding law enforcement in proactive resource allocation and response strategies.

- **Time Series Analysis**: Time series techniques can be applied to analyze temporal patterns in crime rates. Data scientists can identify seasonality, trends, and cyclical behavior, providing law enforcement with insights into when certain types of crimes are most likely to occur.

- **Machine Learning for Crime Profiling**: Data scientists can leverage machine learning algorithms to profile criminals and analyze modus operandi. These profiles can assist in criminal investigations and developing strategies to prevent future offenses.

- **Data Visualization**: Data visualization techniques can help in creating interactive dashboards for law enforcement and policymakers. These dashboards enable real-time monitoring of crime trends and facilitate data-driven decision-making.

- **Public Engagement**: Utilizing data science to create accessible crime data visualizations can foster community engagement. Citizens can gain a better understanding of crime trends in their neighborhoods, empowering them to take preventive measures and cooperate with law enforcement.

2. Data Sources :

Dataset used in this project - 'https://corgis-edu.github.io/corgis/csv/state_crime/'

From the Unified Crime Reporting Statistics and under the collaboration of the U.S. Department of Justice and the Federal Bureau of Investigation information crime statistics are available for public review. The following data set has information on the crime rates and totals for states across the United States for a wide range of years. The crime reports are divided into two main categories: property and violent crime. Property crime refers to burglary, larceny, and motor related crime while violent crime refers to assault, murder, rape, and robbery. These reports go from 1960 to 2019.

3. Data Cleaning and Preprocessing

1. We use .isna and add them to know if there any na values in our dataframe.

```
null_values = project_df.isna().sum()
null_values
```

```
State                           41
Year                            41
Data.Population                 41
Data.Rates.Property.All         41
Data.Rates.Property.Burglary    41
Data.Rates.Property.Larceny     41
Data.Rates.Property.Motor       41
Data.Rates.Violent.All          41
Data.Rates.Violent.Assault      41
Data.Rates.Violent.Murder       41
Data.Rates.Violent.Rape         41
Data.Rates.Violent.Robbery      41
Data.Totals.Property.All        41
Data.Totals.Property.Burglary   41
Data.Totals.Property.Larceny    41
Data.Totals.Property.Motor      41
Data.Totals.Violent.All         41
Data.Totals.Violent.Assault     41
Data.Totals.Violent.Murder      41
Data.Totals.Violent.Rape        41
Data.Totals.Violent.Robbery     41
dtype: int64
```

2. We use dropna() to remove rows with null values from the DataFrame.

```
[ ] project_df = project_df.dropna()
```

```
[ ] project_df.shape
```

```
(3115, 21)
```

3. We use .describe to get the maximum,mean, std,minimum and other statistical values in our dataframe.

```
project_df.describe()
```

	Year	Data.Population	Data.Rates.Property.All	Data.Rates.Property.Burglary	Data.Rates.Property.Larceny	Data.Rates.Property.Motor	Data.Rates.Violent.All	Data.Rates.Violent.As
count	3115.000000	3.115000e+03	3115.000000	3115.000000	3115.000000	3115.000000	3115.000000	3115.0
mean	1989.544141	9.708502e+06	3542.202311	876.532520	2322.659133	343.011300	397.877047	237.3
std	17.299570	3.506750e+07	1418.191397	446.531611	897.934463	221.654068	287.498896	159.3
min	1960.000000	2.261670e+05	573.100000	126.300000	293.300000	28.400000	9.500000	3.6
25%	1975.000000	1.279156e+06	2472.650000	535.000000	1663.800000	185.600000	217.200000	124.0
50%	1990.000000	3.358000e+06	3438.400000	796.600000	2275.700000	288.900000	342.200000	205.1
75%	2005.000000	6.082836e+06	4439.100000	1133.850000	2877.500000	437.200000	518.250000	319.3
max	2019.000000	3.282395e+08	9512.100000	2906.700000	5833.800000	1839.900000	2921.800000	1557.6

4. Change column Data type using convert_dtypes()

```
project_df = project_df.convert_dtypes()
project_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3115 entries, 0 to 3155
Data columns (total 21 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   State                           3115 non-null   string
 1   Year                            3115 non-null   Int64
 2   Data.Population                 3115 non-null   Int64
 3   Data.Rates.Property.All         3115 non-null   Float64
 4   Data.Rates.Property.Burglary    3115 non-null   Float64
 5   Data.Rates.Property.Larceny     3115 non-null   Float64
 6   Data.Rates.Property.Motor       3115 non-null   Float64
 7   Data.Rates.Violent.All          3115 non-null   Float64
 8   Data.Rates.Violent.Assault      3115 non-null   Float64
 9   Data.Rates.Violent.Murder       3115 non-null   Float64
 10  Data.Rates.Violent.Rape         3115 non-null   Float64
 11  Data.Rates.Violent.Robbery      3115 non-null   Float64
 12  Data.Totals.Property.All        3115 non-null   Int64
 13  Data.Totals.Property.Burglary   3115 non-null   Int64
 14  Data.Totals.Property.Larceny    3115 non-null   Int64
 15  Data.Totals.Property.Motor      3115 non-null   Int64
 16  Data.Totals.Violent.All         3115 non-null   Int64
 17  Data.Totals.Violent.Assault     3115 non-null   Int64
 18  Data.Totals.Violent.Murder      3115 non-null   Int64
 19  Data.Totals.Violent.Rape        3115 non-null   Int64
 20  Data.Totals.Violent.Robbery     3115 non-null   Int64
dtypes: Float64(9), Int64(11), string(1)
memory usage: 596.2 KB
```

5. We drop few columns which are not necessary can be added later , if needed.

```
[13] project_df.drop(['Data.Total.crime'], axis=1, inplace = True)
```

```
project_df
```

	State	Year	Data.Population	Data.Rates.Property.All	Data.Rates.Property.Burglary	Data.Rates.Property.Larceny	Data.Rates.Property.Motor	Data.Rates.Violent.All	Data.Rates.Violent.
0	Alabama	1960	3266740	1035.4	355.9	592.1	87.3	186.6	
1	Alabama	1961	3302000	985.5	339.3	569.4	76.8	168.5	
2	Alabama	1962	3358000	1067.0	349.1	634.5	83.4	157.3	
3	Alabama	1963	3347000	1150.9	376.9	683.4	90.6	182.7	
4	Alabama	1964	3407000	1358.7	466.6	784.1	108.0	213.1	
...	...	...	...	...	...	...	...	...	...
3151	Wyoming	2015	586107	1902.6	300.6	1500.9	101.0	222.1	
3152	Wyoming	2016	585501	1957.3	302.5	1518.2	136.6	244.2	
3153	Wyoming	2017	579315	1830.4	275.0	1421.0	134.5	237.5	
3154	Wyoming	2018	577737	1785.1	264.0	1375.9	145.2	212.2	
3155	Wyoming	2019	578759	1571.1	241.2	1206.7	123.2	217.4	

3115 rows × 21 columns

6. Sanity checks:

```
Check if the data present is above 0 and delete rest rows if they are found negative.
Year has negative or zeros value :  0
Data.population has negative or zeros value :  0
Data.Rates.Property.All has negative or zeros value :  0
Data.Rates.Property.Burglary has negative or zeros value :  0
Data.Rates.Property.Motor has negative or zeros value :  0
Data.Rates.Violent.All has negative or zeros value :  0
Data.Rates.Violent.Assault has negative or zeros value :  0
Data.Rates.Violent.Murder has negative or zeros value :  0
Data.Rates.Violent.Rape has negative or zeros value :  0
Data.Rates.Violent.Robbery has negative or zeros value :  0
Data.Totals.Property.All has negative or zeros value :  0
Data.Totals.Property.Larceny has negative or zeros value :  0
Data.Totals.Property.Motor has negative or zeros value :  0
Data.Totals.Violent.All has negative or zeros value :  0
Data.Totals.Violent.Assault has negative or zeros value :  0
Data.Totals.Violent.Murder has negative or zeros value :  0
Data.Totals.Violent.Rape has negative or zeros value :  0
Data.Totals.Violent.Robbery has negative or zeros value :  0
```

7. Maintaining the same case , So that the different states can have one capital word and other small word, Might be a problem when we groupby and get sub datasets. That's why we use text mining.

7. Change the String to lower case.

```
project_df['State'] = project_df['State'].str.lower()
project_df
```

	State	Year	Data.Population	Data.Rates.Property.All	Data.Rates.Property.Burglary	Data.Rates.Property.Larceny	Data.Rates.Property.Motor	Data.Rates.Violent.All	Data.Rates.Violent.A
0	alabama	1960	3266740	1035.4	355.9	592.1	87.3	186.6	
1	alabama	1961	3302000	985.5	339.3	569.4	76.8	168.5	
2	alabama	1962	3358000	1067.0	349.1	634.5	83.4	157.3	
3	alabama	1963	3347000	1150.9	376.9	683.4	90.6	182.7	
4	alabama	1964	3407000	1358.7	466.6	784.1	108.0	213.1	
...	...	...	...	...	...	...	...	...	...
3151	wyoming	2015	586107	1902.6	300.6	1500.9	101.0	222.1	
3152	wyoming	2016	585501	1957.3	302.5	1518.2	136.6	244.2	
3153	wyoming	2017	579315	1830.4	275.0	1421.0	134.5	237.5	
3154	wyoming	2018	577737	1785.1	264.0	1375.9	145.2	212.2	
3155	wyoming	2019	578759	1571.1	241.2	1206.7	123.2	217.4	

3115 rows × 21 columns

8. Rename the columns for a better idea of what columns we have and look at the features properly in rows.

8. Rename columns

```
column_mapping = {
    'Data.Population': 'Population',
    'Data.Rates.Property.All': 'Property_Rates_All',
    'Data.Rates.Property.Burglary': 'Property_Rates_Burglary',
    'Data.Rates.Property.Larceny': 'Property_Rates_Larceny',
    'Data.Rates.Property.Motor': 'Property_Rates_Motor',
    'Data.Rates.Violent.All': 'Violent_Rates_All',
    'Data.Rates.Violent.Assault': 'Violent_Rates_Assault',
    'Data.Rates.Violent.Murder': 'Violent_Rates_Murder',
    'Data.Rates.Violent.Rape': 'Violent_Rates_Rape',
    'Data.Rates.Violent.Robbery': 'Violent_Rates_Robbery',
    'Data.Totals.Property.All': 'Property_Totals_All',
    'Data.Totals.Property.Burglary': 'Property_Totals_Burglary',
    'Data.Totals.Property.Larceny': 'Property_Totals_Larceny',
    'Data.Totals.Property.Motor': 'Property_Totals_Motor',
    'Data.Totals.Violent.All': 'Violent_Totals_All',
    'Data.Totals.Violent.Assault': 'Violent_Totals_Assault',
    'Data.Totals.Violent.Murder': 'Violent_Totals_Murder',
    'Data.Totals.Violent.Rape': 'Violent_Totals_Rape',
    'Data.Totals.Violent.Robbery': 'Violent_Totals_Robbery'
}

project_df = project_df.rename(columns=column_mapping)
project_df.columns
```

```
Index(['State', 'Year', 'Population', 'Property_Rates_All',
       'Property_Rates_Burglary', 'Property_Rates_Larceny',
       'Property_Rates_Motor', 'Violent_Rates_All', 'Violent_Rates_Assault',
       'Violent_Rates_Murder', 'Violent_Rates_Rape', 'Violent_Rates_Robbery',
       'Property_Totals_All', 'Property_Totals_Burglary',
       'Property_Totals_Larceny', 'Property_Totals_Motor',
```

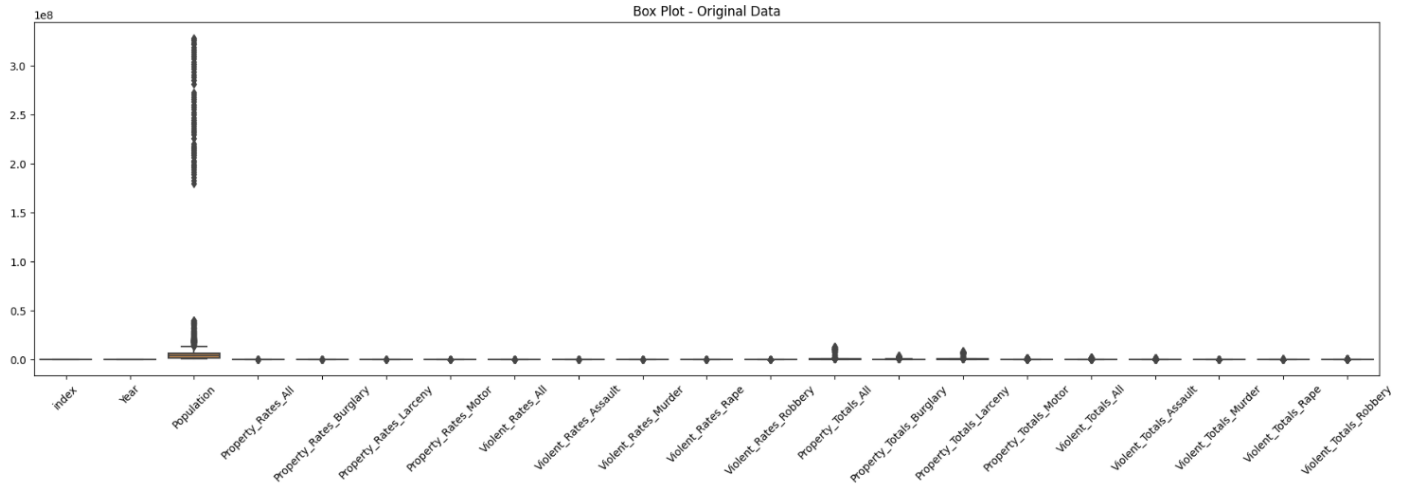
9. As we have done lots of cleaning and preprocessing of the data , we have messed up with the index , to make it proper we use reset_index.

9. Reset index

```
project_df = project_df.reset_index()
```

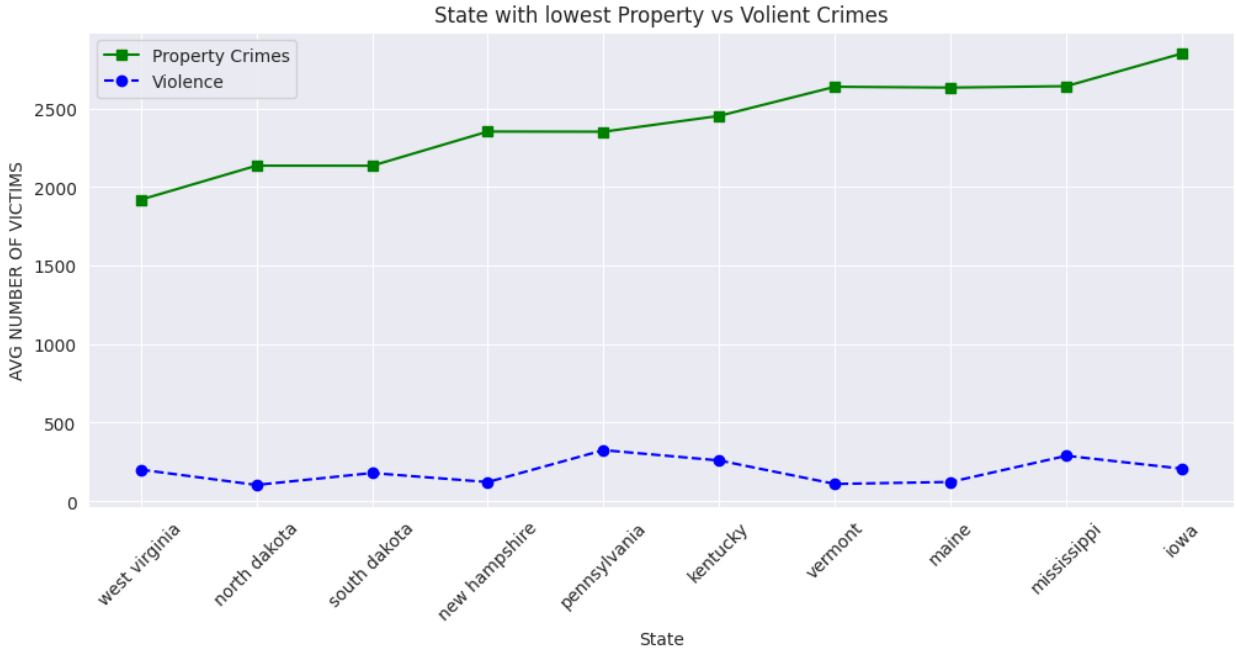
10. Outliers are the data points which are far away from the usual data and need to be ignored or mostly eliminated . In my dataset Population is the most important column , we can just eliminate that , if we remove that it simply means no live person in that year. So we ignore it .

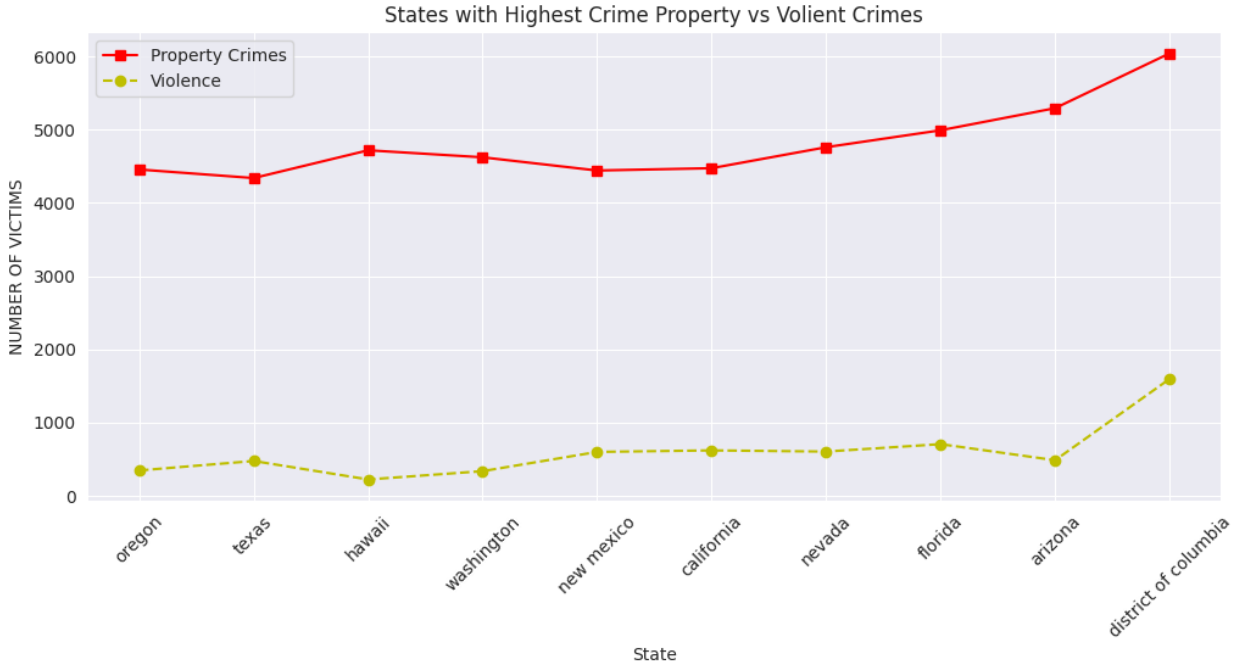
```
plt.subplot(1, 2, 1)
numeric_columns = project_df.select_dtypes(include=['int64', 'float64'])
sns.boxplot(data=numeric_columns, orient='vertical')
plt.xticks(rotation=45)
plt.title('Box Plot - Original Data');
```



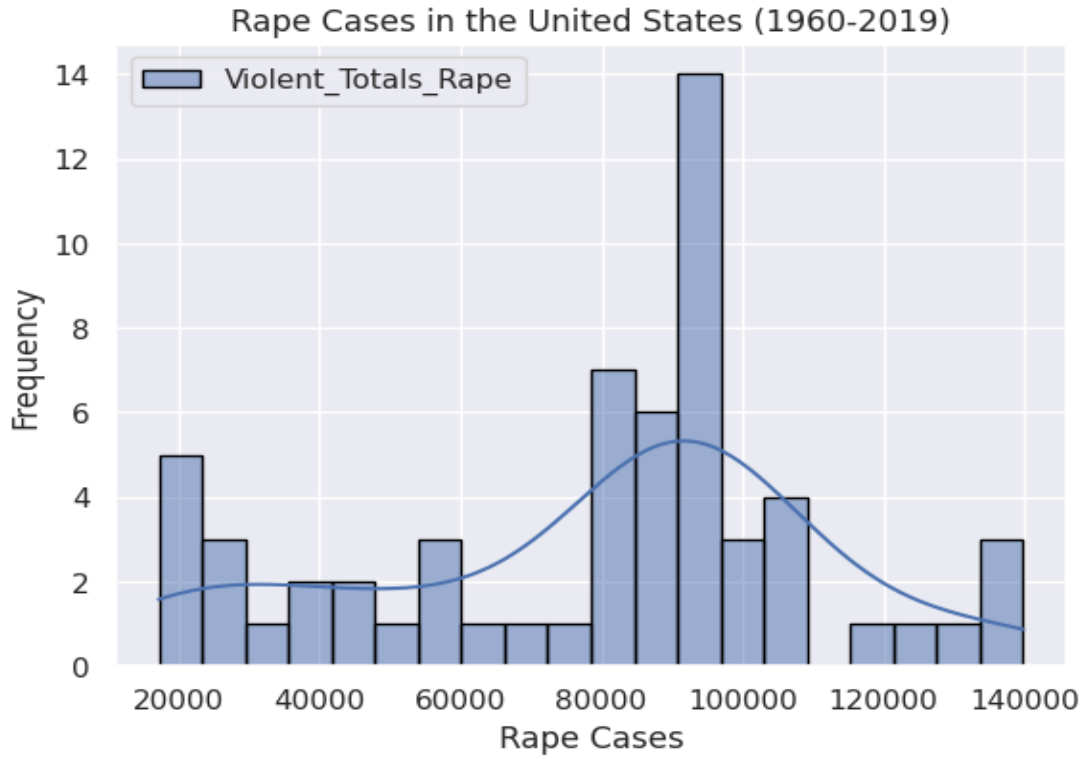
4. Exploratory Data Analysis (EDA):

a. Determine which states consistently have the highest and lowest crime rates across all years. [**LINE Graph**]



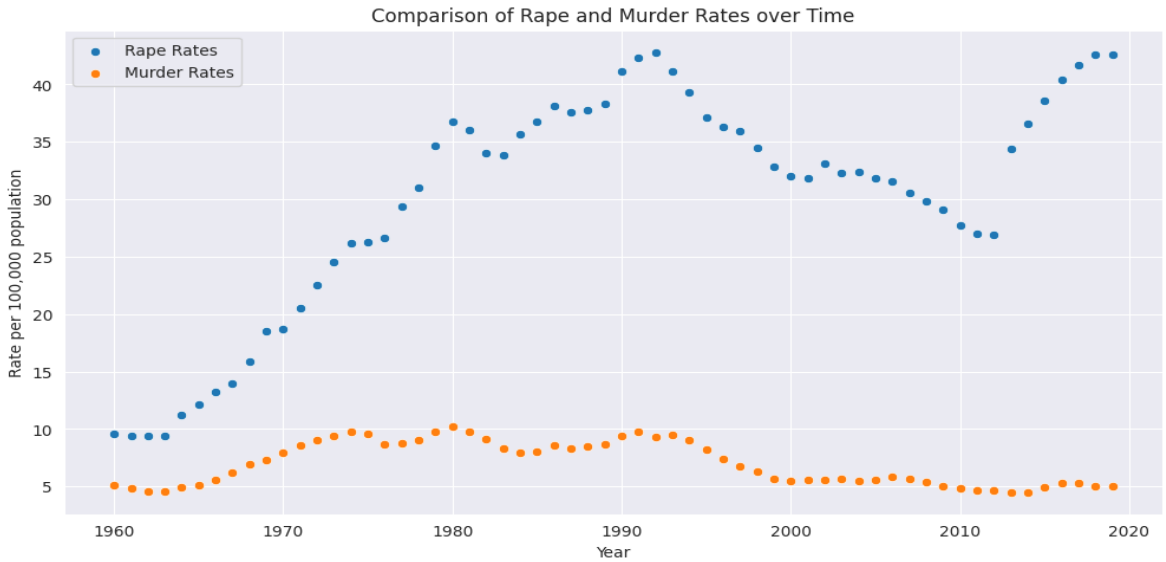


b. Time series on Rapes in United States using Histogram? (univariate analysis). [**Histogram**]



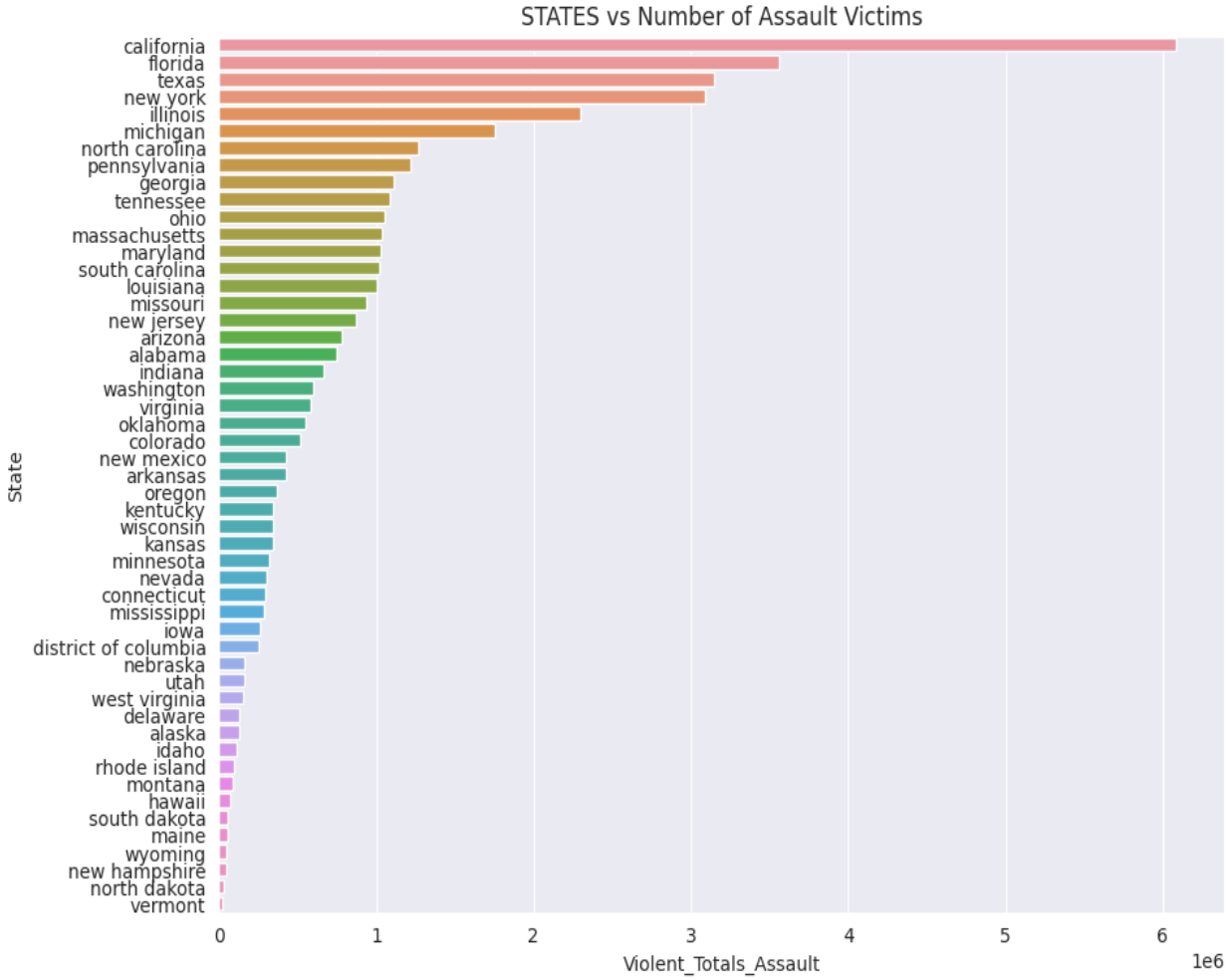
This is histogram showing the distribution of rape cases in the United States, with data grouped into 20 bins.

c. Compare Assault vs Murder cases in the United States? [**Scatter Plot**]



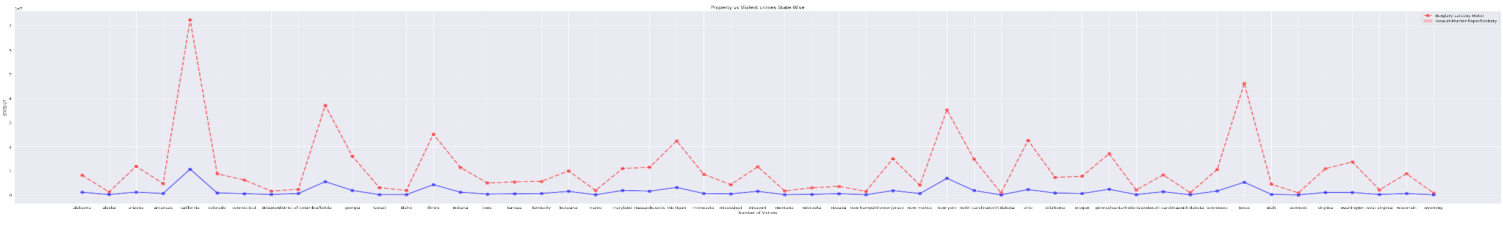
- We filter the DataFrame to select the columns related to the year, rape rates, and murder rates.
- We create a scatter plot using Seaborn, plotting rape rates and murder rates on the same chart.
- We set the figure size, style, labels, and title to make the plot informative and visually appealing.
- This will generate a scatter plot that shows the trends in rape and murder rates over time, allowing you to visually compare these two types of crimes.

d. Compare Rape cases in all States between 1960-2019? (Bi- variate) [**Bar Graph**]



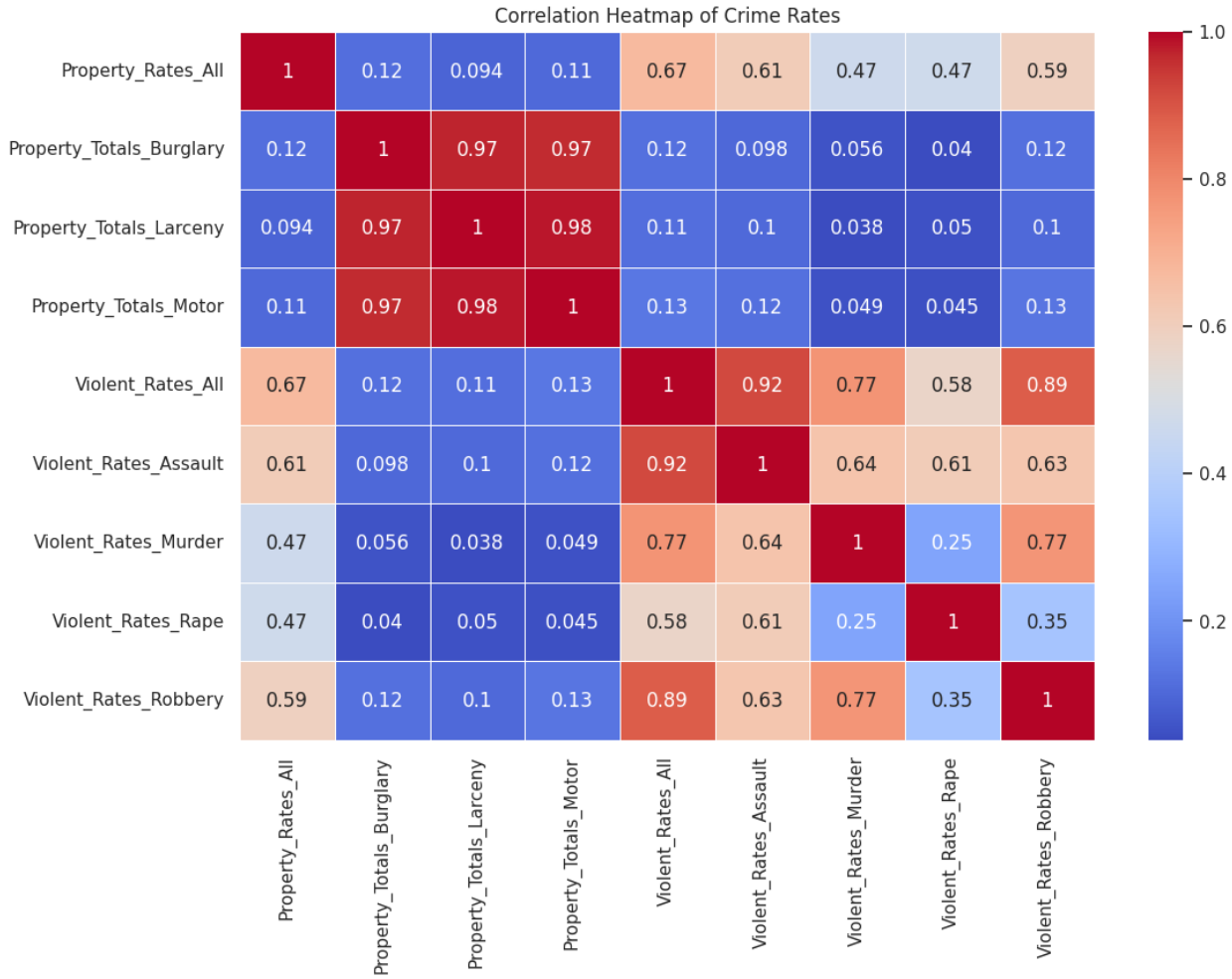
Bar plot showing the number of assault victims in different states, with states sorted in descending order based on the number of assault victims.

e. Compare Property vs Violent crimes between all the states?[**Line Plot**]



This code creates a line plot comparing the total property crime victims (Burglary, Larceny, and Motor) and total violent crime victims (Assault, Murder, Rape, and Robbery) for different states in US.

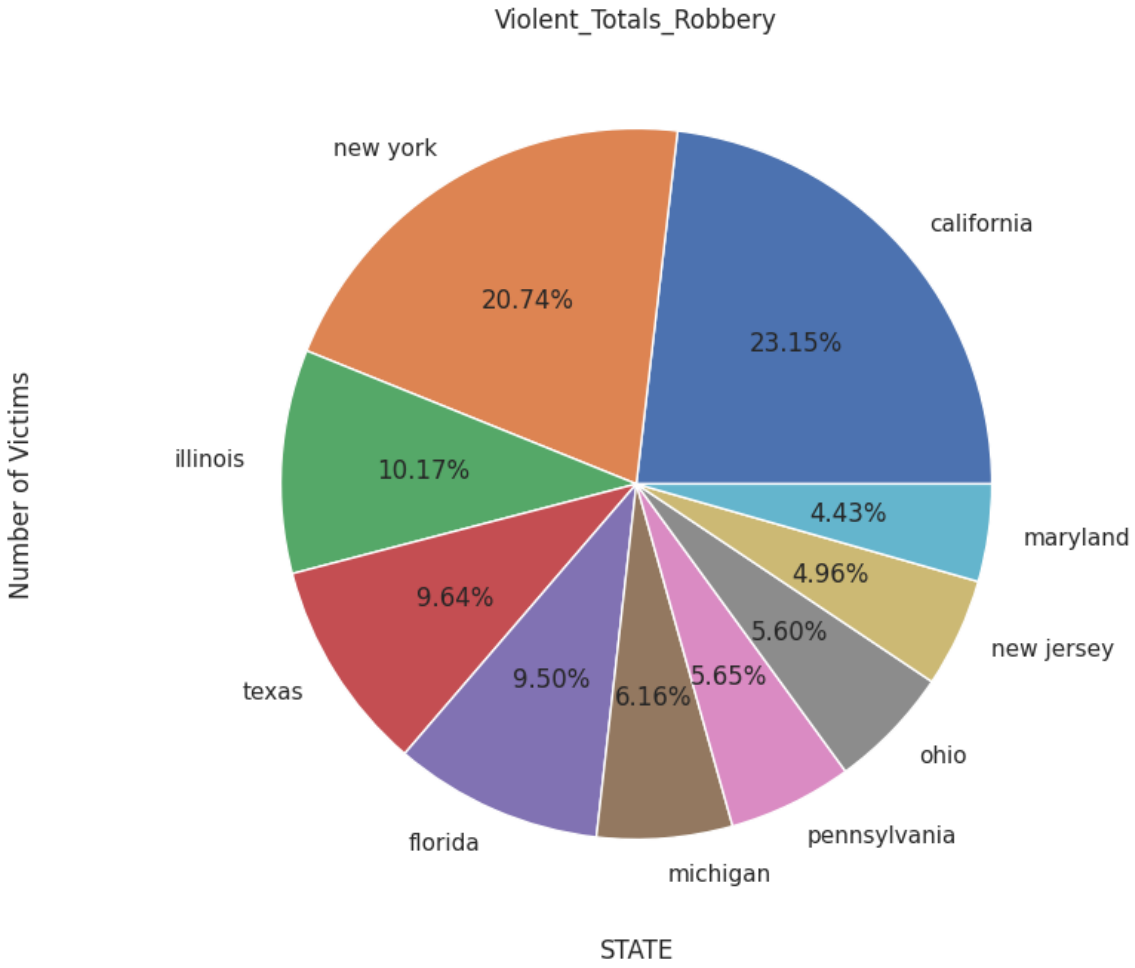
f. Correlations between different types of crime rates over the years. [**Heat Map**]



- We select the relevant columns for crime rates, including property and various violent crime rates.
- We set the 'Year' column as the index to use it as the x-axis of the heatmap.

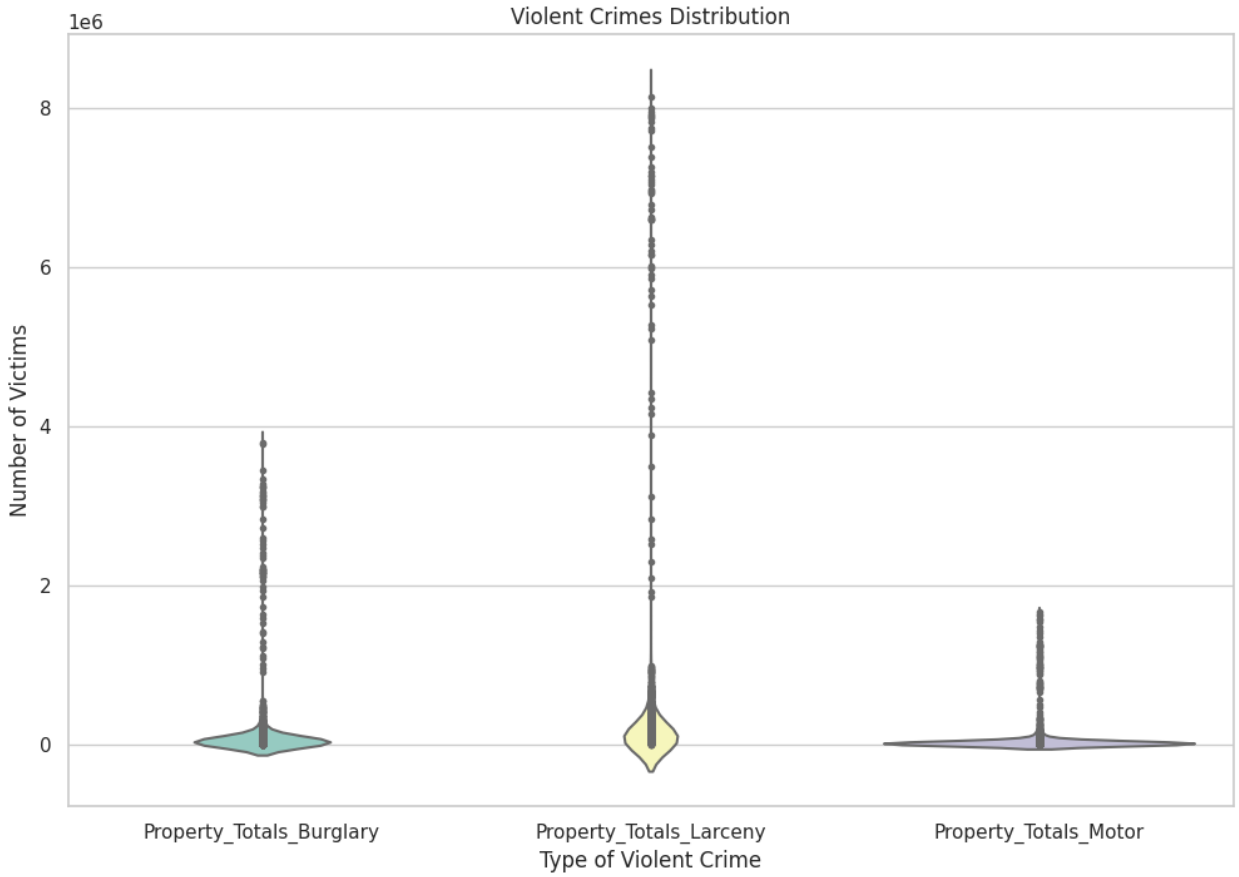
- We create a heatmap using Seaborn, displaying the correlation between different crime rates. The annot=True parameter adds numerical values to the heatmap cells to indicate the strength of the correlation.
- We use the 'coolwarm' colormap for the heatmap for better visualization of correlations.
- This will create a heatmap that shows the correlations between different types of crime rates over the years. You can adjust the columns you want to include in the analysis or use different colormaps and styling options to customize the visualization based on your specific requirements.

g. Top 10 states with Robbery Crime. [**Pie Chart**]



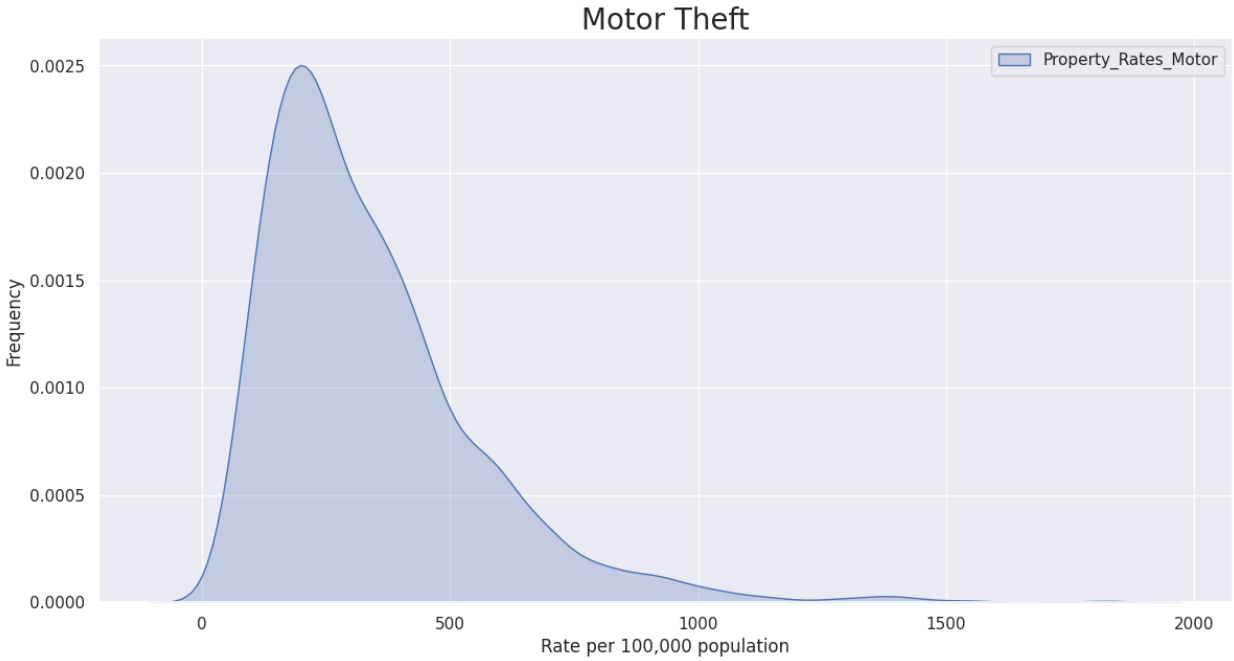
a pie chart showing the distribution of robbery victims in the top 10 states.

h. Distribution of victims for different types of Property crimes.[**Violin Plot**]



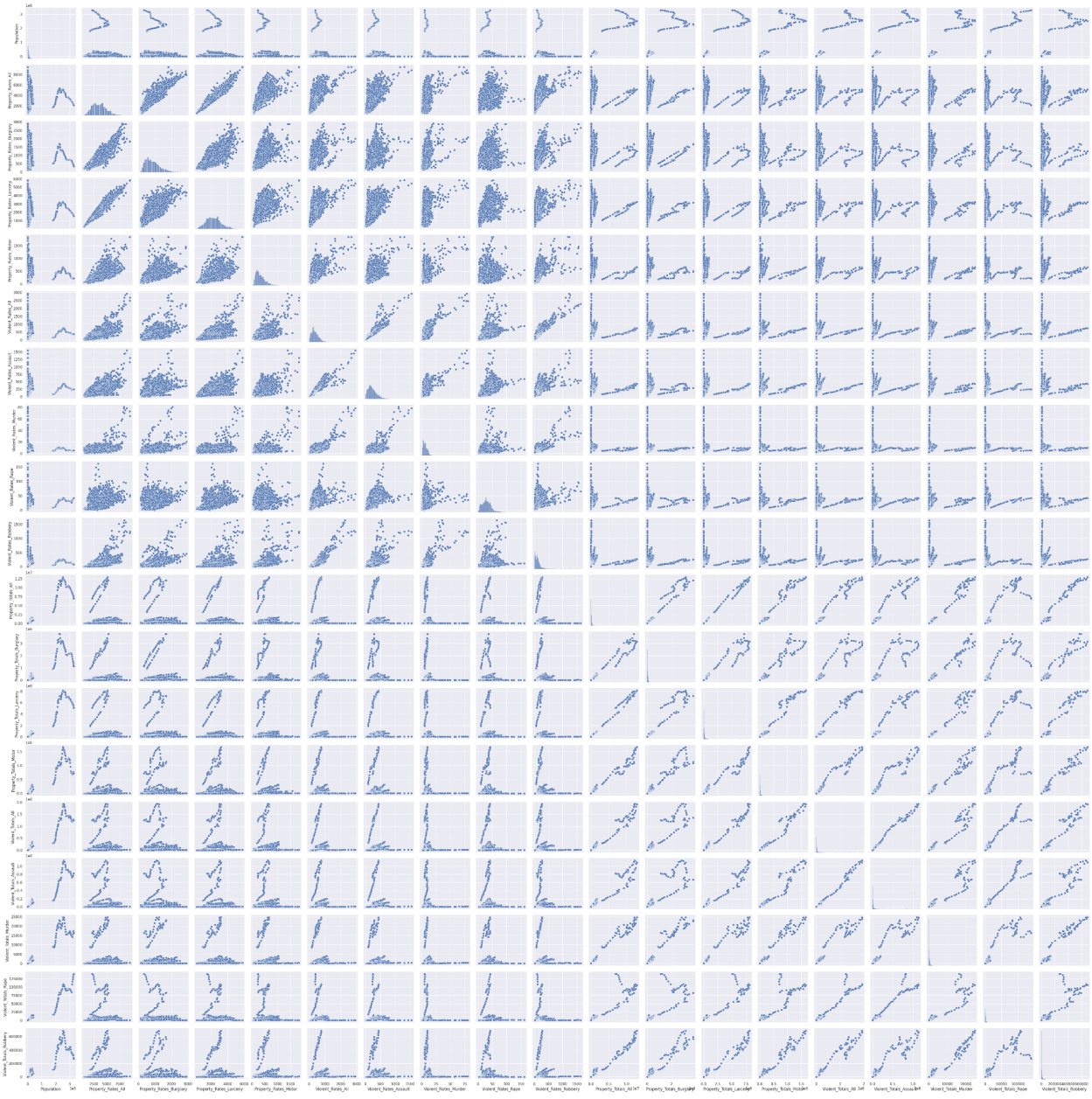
- We select the columns related to different types of violent crimes (Assault, Murder, Rape, and Robbery).
- We create a violin plot using Seaborn, specifying an "inner" parameter as "points" to display individual data points within the violin plot.
- We set the figure size, style, labels, and title for the plot.
- This code will create a violin plot that shows the distribution of victims for different types of violent crimes, allowing you to compare the distribution and the central tendencies for each crime category.

I. Motor theft across over years.[**Kde plot**]



- We've extracted the 'Property_Rates_Motor' column as a Series.
- We've used fill=True directly within the kdeplot function, which is the correct way to specify that you want to fill the area under the curve.
- This code will create a kernel density plot for the 'Property_Rates_Motor' data with the specified color and fill.

J. **Pair Plot** (Relationship and correlation between all the Variables.)



- columns you want to include in the pair plot using the DataFrame indexing.
- You create the pair plot using sns.pairplot().
- a pair plot that visualizes the relationships between the selected variables, providing insights into their pairwise correlations and distributions.