Analysis Report

Kondra Nagabhavani

September 1, 2024

Roll : BT21CSE125

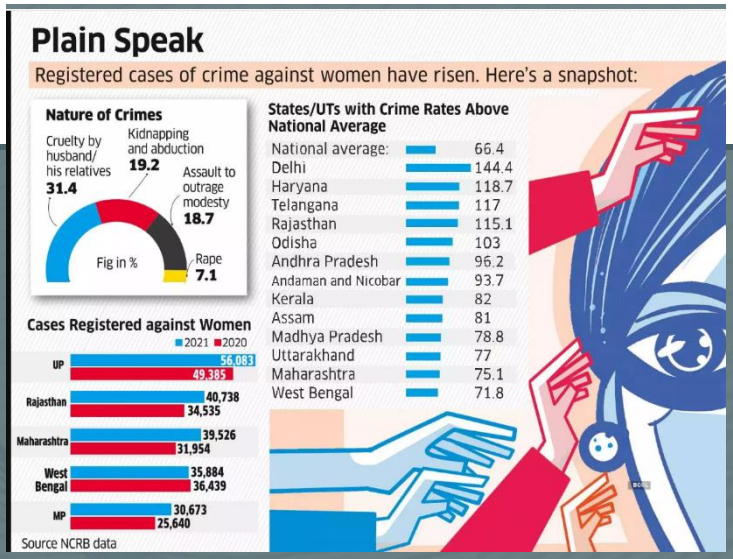
Introduction to Data Analytics & Data Mining

Assignment-1 Part-2

Professor Dr. Shital Raut

TITLE : Descriptive Analysis on Crimes Against Women in India (2001-2021)

INTRODUCTION : The safety and security of women have been a significant concern in India, with various crimes against women being reported across states over the years. This report aims to analyze state-wise data on different crimes committed against women between 2001 and 2021. The dataset includes information on crimes such as rape, kidnapping and assault, dowry deaths, assault on women, assault on modesty, domestic violence, and women trafficking. The analysis involves exploring the data, visualizing trends, and identifying key insights.



THE PROCESS

1.DATASET DESCRIPTION :

Name of the Dataset : Crimes against women in India from 2001 to 2021 [CrimesOnWomenData.csv]
You can find the Dataset here.
This data is collated from https://data.gov.in. It has state-wise data on the various crimes committed against women between 2001 to 2021. Some crimes that are included are Rape, Kidnapping and Abduction, Dowry Deaths etc.

Dataset Overview
The dataset used for this analysis contains the following columns.
State : The state where the crime was reported.
Year : The year in which the crime data was recorded.
Rape : The number of rape cases reported.
Kidnap_and_Assault : The number of kidnapping and assault cases reported.
Dowry_Deaths : The number of dowry deaths reported.
Assault_on_Women : The number of assaults on women reported.
Assault_on_Modesty : The number of assaults on the modesty of women reported.
Domestic_Violence : The number of domestic violence cases reported.
Women_Trafficking : The number of women trafficking cases reported.

2.LOADING THE DATASET

```
[9]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     # Load dataset
     df = pd.read_csv('CrimesOnWomenData.csv')
     # Summary statistics for numerical columns
     print(df.head())
```

We'll load the dataset into a pandas DataFrame for further analysis. ‘df.head()’ displays the first five rows of the DataFrame to give an initial glimpse of the data.
Output :

	Unnamed: 0	State	Year	Rape	K&A	DD	AoW	AoM	DV	WT
0	0	ANDHRA PRADESH	2001	871	765	420	3544	2271	5791	7
1	1	ARUNACHAL PRADESH	2001	33	55	0	78	3	11	0
2	2	ASSAM	2001	817	1070	59	850	4	1248	0
3	3	BIHAR	2001	888	518	859	562	21	1558	83
4	4	CHHATTISGARH	2001	959	171	70	1763	161	840	0

3. DATA EXPLORATION

The next step in the analysis is to explore the dataset to understand the distribution and trends of various crimes over the years. Summary statistics were calculated to get a sense of the central tendencies and dispersions within each crime category. The dataset revealed significant variability in the number of crimes reported across different states and years.

In this step, we'll explore the dataset to understand its structure, content, and basic statistics.

1. df.info() provides information about the DataFrame including the number of non-null entries, data types of each column, and memory usage.

```
# Get a concise summary of the DataFrame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 736 entries, 0 to 735
Data columns (total 11 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Unnamed: 0           736 non-null    int64
 1   State                736 non-null    object
 2   Year                 736 non-null    int64
 3   Rape                 736 non-null    int64
 4   Kidnap_and_Assault   736 non-null    int64
 5   Dowry_Deaths         736 non-null    int64
 6   Assault_on_Women     736 non-null    int64
 7   Assault_on_Modesty   736 non-null    int64
 8   Domestic_Violence    736 non-null    int64
 9   Women_Trafficking    736 non-null    int64
 10  Total_Crimes         736 non-null    int64
dtypes: int64(10), object(1)
memory usage: 63.4+ KB
```

```
0s
# Summary statistics for numerical columns
print(df.describe())
# Check for missing values
print(df.isnull().sum())
# Check for duplicate rows
df.duplicated().sum()
```

df.describe() provides statistical measures like mean, standard deviation, min, and max for numerical columns.
df.isnull().sum() returns the count of missing values in each column.
df.duplicated(). sum() returns the number of duplicate rows in the DataFrame.

Output :

```
0s
       Unnamed: 0         Year         Rape          K&A           DD  \
count  736.000000   736.000000   736.000000   736.000000   736.000000
mean   367.500000  2011.149457   727.855978  1134.542120   215.692935
std    212.609188     6.053453   977.024945  1993.536828   424.927334
min      0.000000  2001.000000     0.000000     0.000000     0.000000
25%    183.750000  2006.000000    35.000000    24.750000     1.000000
50%    367.500000  2011.000000   348.500000   290.000000    29.000000
75%    551.250000  2016.000000  1069.000000  1216.000000   259.000000
max    735.000000  2021.000000  6337.000000 15381.000000  2524.000000

                AoW          AoM           DV          WT
count    736.000000   736.000000   736.000000  736.000000
mean    1579.115489   332.722826  2595.078804   28.744565
std     2463.962518   806.024551  4042.004953   79.999660
min        0.000000     0.000000     0.000000    0.000000
25%       34.000000     3.000000    13.000000    0.000000
50%      387.500000    31.000000   678.500000    0.000000
75%     2122.250000   277.500000  3545.000000   15.000000
max    14853.000000  9422.000000 23278.000000  549.000000
Unnamed: 0    0
State         0
Year          0
Rape          0
K&A           0
DD            0
AoW           0
AoM           0
DV            0
```

4.DATA CLEANING

```
[13] # Fill missing values with 0 or appropriate strategy
     df.fillna(0, inplace=True)
     # Remove duplicate rows
     df.drop_duplicates(inplace=True)
     # Check data types
     df.dtypes
```

df.fillna fills any missing numerical values with 0. Depending on the context, you might choose different strategies like filling with mean/median or removing rows with missing values.
df.drop_duplicates() removes any duplicate rows from the DataFrame.

```
# Convert Year to integer if not already
df['Year'] = df['Year'].astype(int)

# Ensure all crime columns are numeric
crime_columns = ['Rape', 'K&A', 'DD', 'AoW', 'AoM', 'DV', 'WT']
for col in crime_columns:
    df[col] = pd.to_numeric(df[col], errors='coerce').fillna(0).astype(int)
```

Ensures that all columns have appropriate data types for analysis.
Converts any non-numeric entries in crime columns to numeric, replacing errors with 0.

```
[14] # Rename columns for better readability
     df.rename(columns={
         'K&A': 'Kidnap_and_Assault',
         'DD': 'Dowry_Deaths',
         'AoW': 'Assault_on_Women',
         'AoM': 'Assault_on_Modesty',
         'DV': 'Domestic_Violence',
         'WT': 'Women_Trafficking'
     }, inplace=True)
```

Renames columns to more descriptive names, improving readability in plots and analysis.

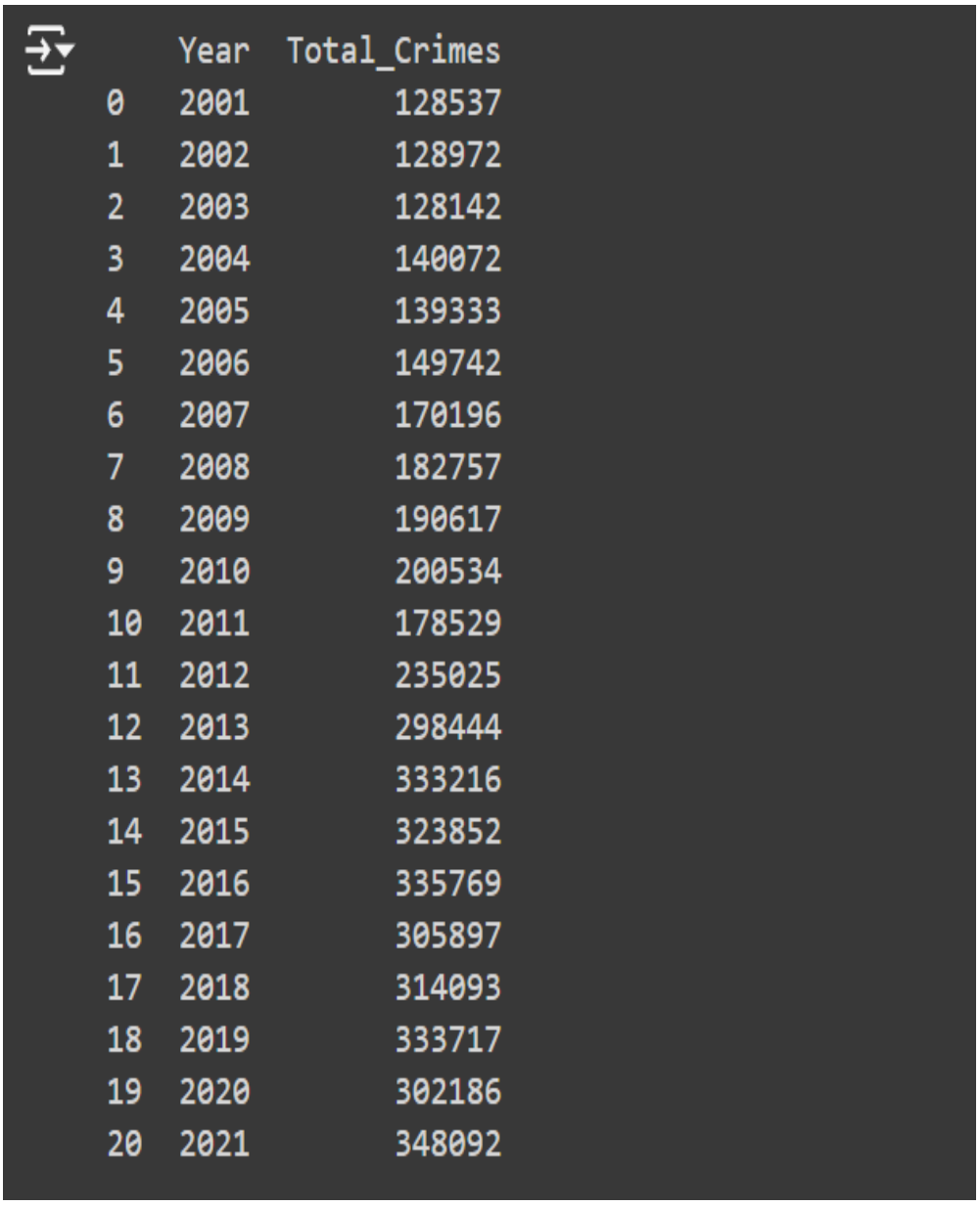
5. DESCRIPTIVE ANALYTICS & VISUALIZATION

1.Total Crimes Per Year :

```
[46] # Calculate total crimes per year
     df['Total_Crimes'] = df[['Rape', 'Kidnap_and_Assault', 'Dowry_Deaths', 'Assault_on_Women',
                              'Assault_on_Modesty', 'Domestic_Violence', 'Women_Trafficking']].sum(axis=1)

     yearly_crimes = df.groupby('Year')['Total_Crimes'].sum().reset_index()
     print (yearly_crimes)
```

Adds a new column Total_Crimes representing the sum of all crime types for each record. Aggregates total crimes for each year across all states.

Output :



	Year	Total_Crimes
0	2001	128537
1	2002	128972
2	2003	128142
3	2004	140072
4	2005	139333
5	2006	149742
6	2007	170196
7	2008	182757
8	2009	190617
9	2010	200534
10	2011	178529
11	2012	235025
12	2013	298444
13	2014	333216
14	2015	323852
15	2016	335769
16	2017	305897
17	2018	314093
18	2019	333717
19	2020	302186
20	2021	348092

Visualization Using Line Charts：

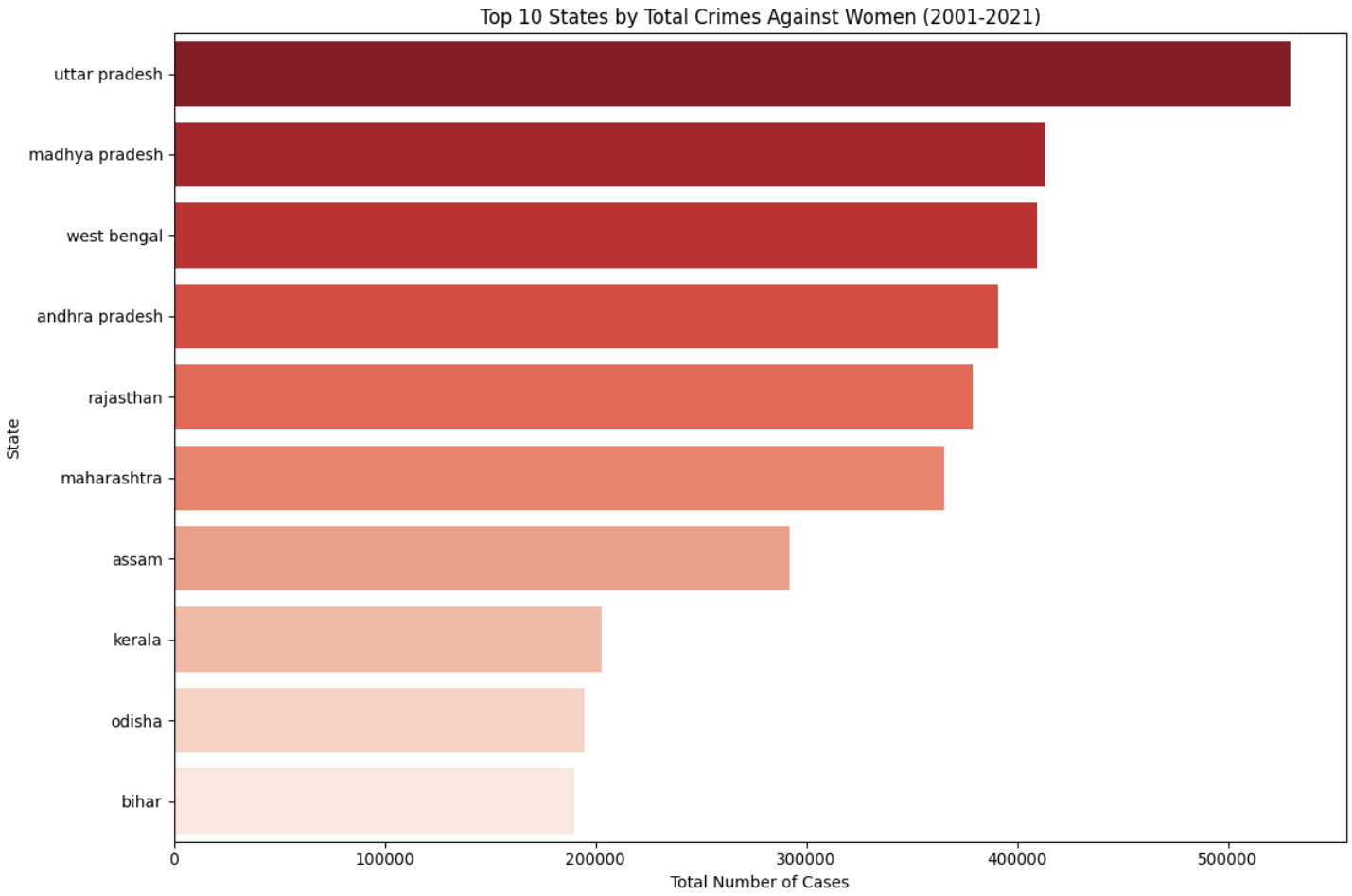
```
# Line plot for total crimes over years
plt.figure(figsize=(12, 6))
sns.lineplot(data=yearly_crimes, x='Year', y='Total_Crimes', marker='o')
plt.title('Total Crimes Against Women in India (2001-2021)')
plt.xlabel('Year')
plt.ylabel('Number of Cases')
plt.xticks(yearly_crimes['Year'], rotation=45)
plt.tight_layout()
plt.show()
```

2. Top 10 States by Total Crimes

Aggregates total crimes for each state over all years.
Displays a horizontal bar chart of the top 10 states with the highest number of crimes against women.

```
# Bar plot for top 10 states with highest crimes
# Calculate total crimes per state
state_crimes = df.groupby('State')['Total_Crimes'].sum().reset_index().sort_values(by='Total_Crimes', ascending=False)
top_10_states = state_crimes.head(10)

plt.figure(figsize=(12, 8))
sns.barplot(data=top_10_states, x='Total_Crimes', y='State', palette='Reds_r')
plt.title('Top 10 States by Total Crimes Against Women (2001-2021)')
plt.xlabel('Total Number of Cases')
plt.ylabel('State')
plt.tight_layout()
plt.show()
```

Visualization Using Horizontal Bar Chart :

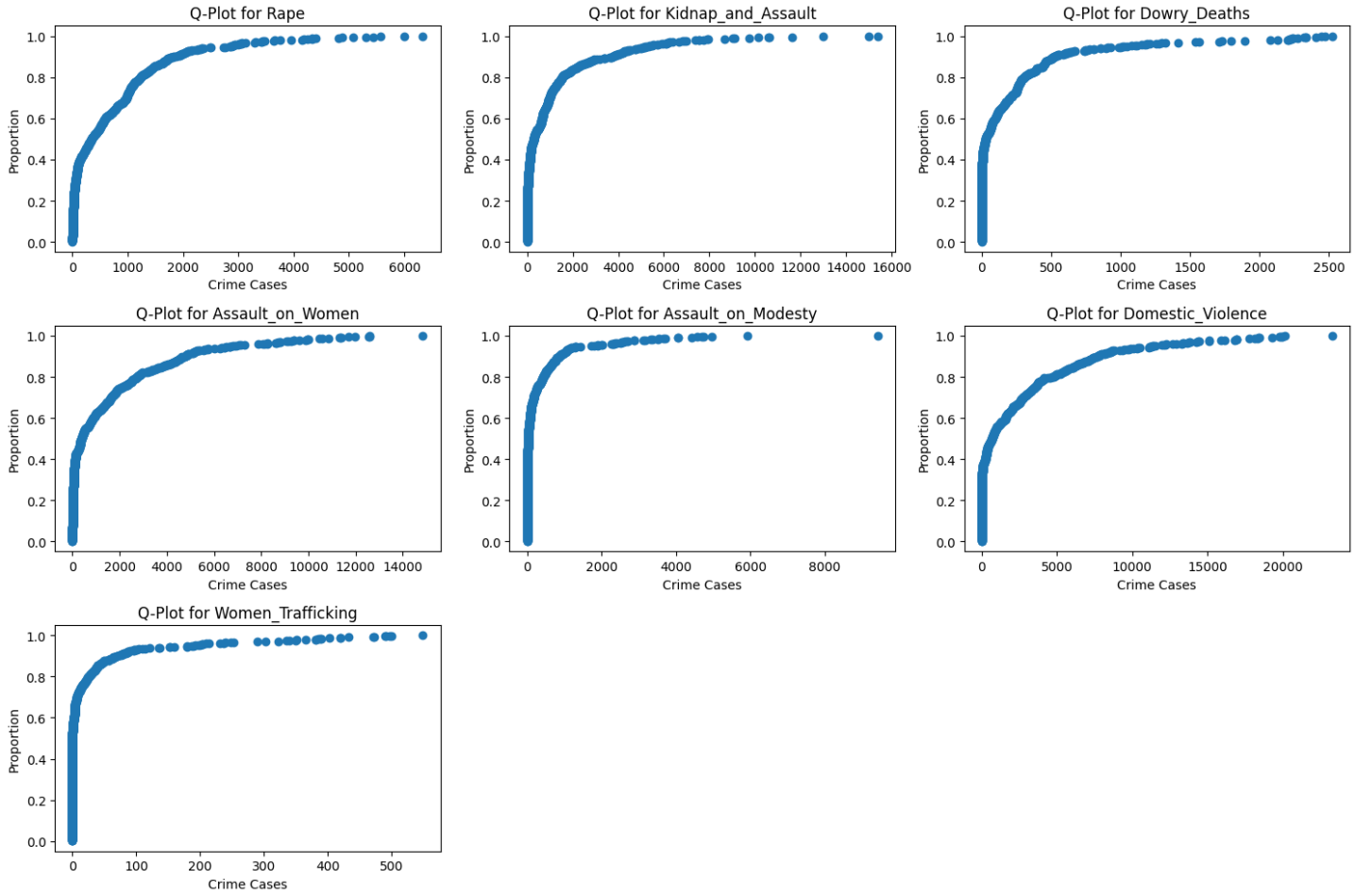


3. Q-Plots for Different Crime Columns

A Q-plot, or empirical quantile plot, visualizes the quantiles of the data distribution. This is done by sorting the data and plotting the values against their expected quantile positions.

```
# List of crime columns to plot
crime_columns = ['Rape', 'Kidnap_and_Assault', 'Dowry_Deaths',
                 'Assault_on_Women', 'Assault_on_Modesty',
                 'Domestic_Violence', 'Women_Trafficking']

# Plotting Q-Plots for each crime type
plt.figure(figsize=(15, 10))

for i, col in enumerate(crime_columns, 1):
    plt.subplot(3, 3, i)  # Adjust subplot grid size according to the number of plots
    sorted_data = np.sort(df[col])
    p = np.arange(1, len(sorted_data) + 1) / len(sorted_data)
    plt.plot(sorted_data, p, marker='o', linestyle='none')
    plt.xlabel('Crime Cases')
    plt.ylabel('Proportion')
    plt.title(f'Q-Plot for {col}')

plt.tight_layout()
plt.show()
```

We sort the data for each crime type and calculate the proportion of each data point in the distribution.
The plot shows the distribution of crime data across quantiles.

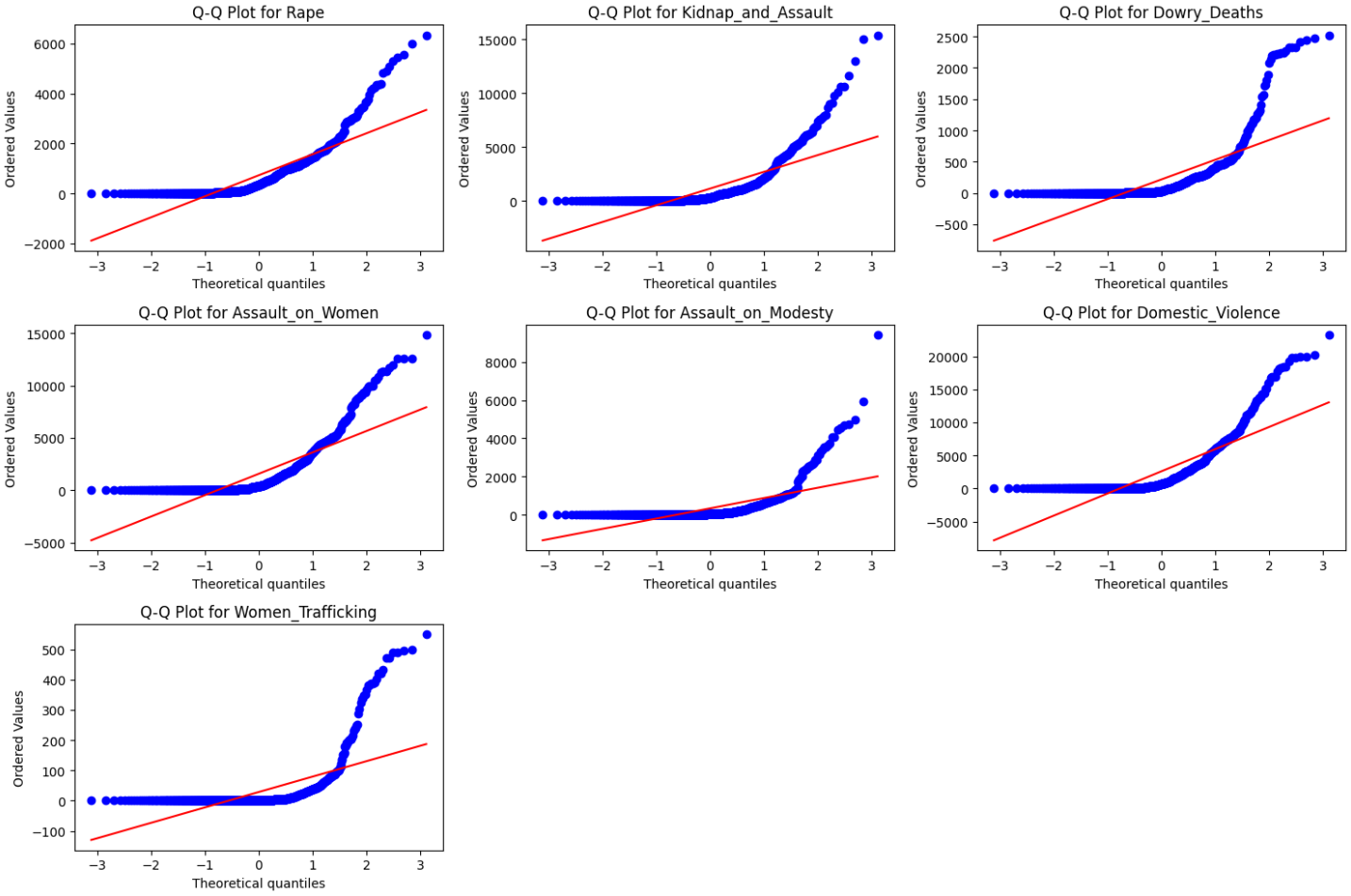


4.Q-Q Plots for Different Crime Columns

We'll generate Q-Q plots for the various crime columns to check if they follow a normal distribution.

```
[24] import scipy.stats as stats
     # List of crime columns to plot
     crime_columns = ['Rape', 'Kidnap_and_Assault', 'Dowry_Deaths',
                      'Assault_on_Women', 'Assault_on_Modesty',
                      'Domestic_Violence', 'Women_Trafficking']

     # Plotting Q-Q plots for each crime type
     plt.figure(figsize=(15, 10))

     for i, col in enumerate(crime_columns, 1):
         plt.subplot(3, 3, i)  # Adjust subplot grid size according to the number of plots
         stats.probplot(df[col], dist="norm", plot=plt)
         plt.title(f'Q-Q Plot for {col}')

     plt.tight_layout()
     plt.show()
```

stats.probplot() is used to generate the Q-Q plot, comparing the data distribution to a normal distribution.
The loop generates Q-Q plots for each specified crime column.



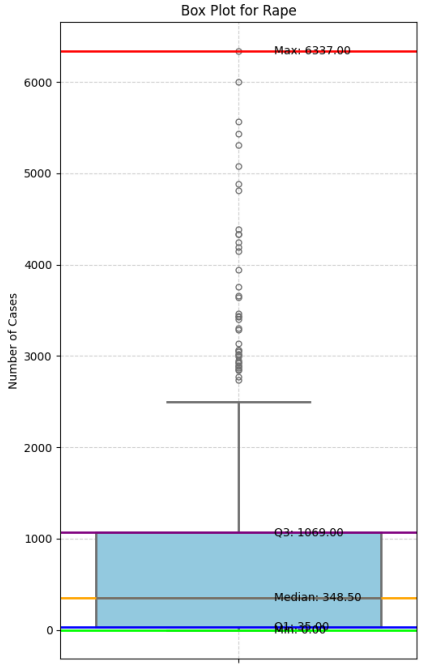
5.BOX Plots for Different Crime Columns

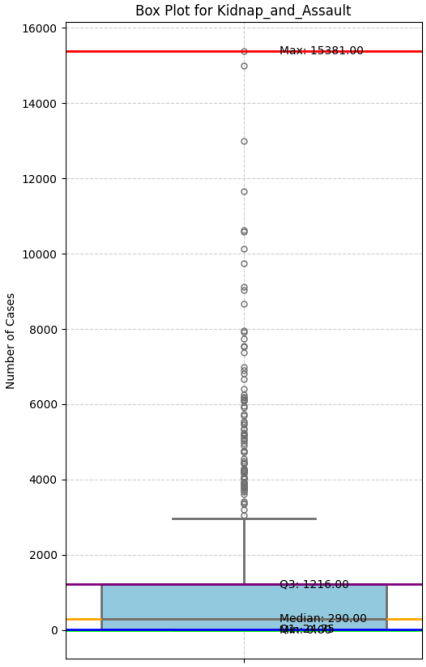
Box plots, also known as box-and-whisker plots, are used to visualize the distribution of data based on five summary statistics: minimum, first quartile (Q1), median, third quartile (Q3), and maximum. They are useful for identifying outliers and understanding the spread of the data.

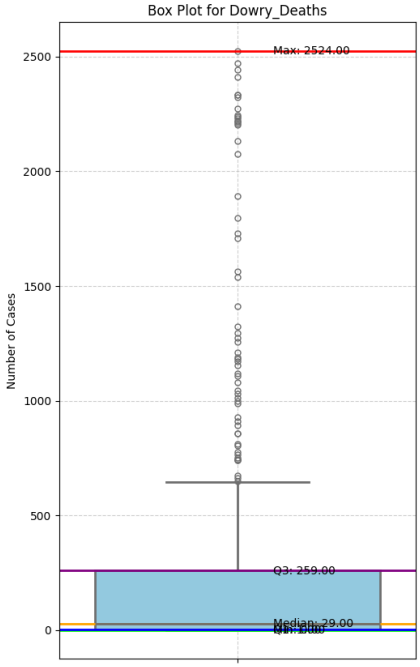
sns.boxplot() is used to create a box plot for each crime type.
The loop generates box plots for each specified crime column.

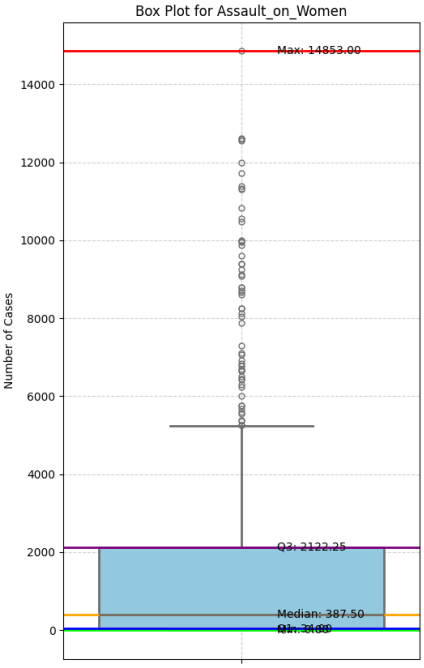
```python
# Plotting Enhanced Box Plots with annotations for each crime type
plt.figure(figsize=(16, 25))

for i, col in enumerate(crime_columns, 1):
    plt.subplot(3, 3, i)

    # Customizing the color palette
    box_color = '#87CEEB'  # Light Sky Blue
    min_color = '#00FF00'  # Green for Min
    q1_color = '#0000FF'   # Blue for Q1
    median_color = '#FFA500' # Orange for Median
    q3_color = '#800080'   # Purple for Q3
    max_color = '#FF0000'  # Red for Max
    text_color = '#000000' # Black for annotations

    # Creating box plot
    sns.boxplot(y=df[col], color=box_color, linewidth=2, fliersize=5)

    # Calculating statistics
    stats = df[col].describe()
    min_val = stats['min']
    q1 = stats['25%']
    median = stats['50%']
    q3 = stats['75%']
    max_val = stats['max']
```

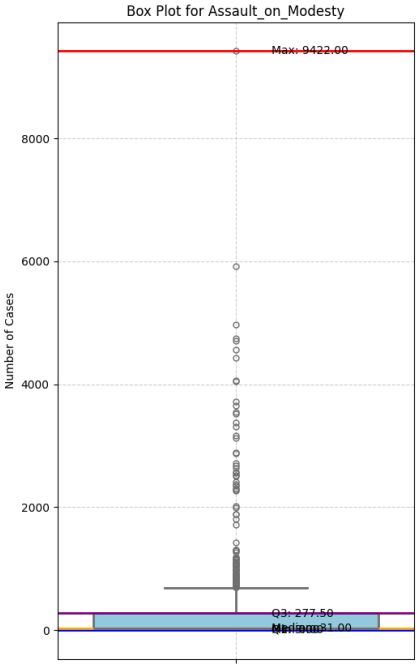
```python
    # Drawing horizontal lines for Min, Q1, Median, Q3, Max
    plt.axhline(min_val, color=min_color, linestyle='-', linewidth=2, label=f'Min: {min_val:.2f}')
    plt.axhline(q1, color=q1_color, linestyle='-', linewidth=2, label=f'Q1: {q1:.2f}')
    plt.axhline(median, color=median_color, linestyle='-', linewidth=2, label=f'Median: {median:.2f}')
    plt.axhline(q3, color=q3_color, linestyle='-', linewidth=2, label=f'Q3: {q3:.2f}')
    plt.axhline(max_val, color=max_color, linestyle='-', linewidth=2, label=f'Max: {max_val:.2f}')

    # Annotating each statistic on the boxplot
    plt.text(0.1, min_val, f'Min: {min_val:.2f}', ha='left', va='center', color=text_color, fontsize=10)
    plt.text(0.1, q1, f'Q1: {q1:.2f}', ha='left', va='center', color=text_color, fontsize=10)
    plt.text(0.1, median, f'Median: {median:.2f}', ha='left', va='center', color=text_color, fontsize=10)
    plt.text(0.1, q3, f'Q3: {q3:.2f}', ha='left', va='center', color=text_color, fontsize=10)
    plt.text(0.1, max_val, f'Max: {max_val:.2f}', ha='left', va='center', color=text_color, fontsize=10)

    plt.title(f'Box Plot for {col}', color=text_color, fontsize=12)
    plt.ylabel('Number of Cases', color=text_color, fontsize=10)

    plt.grid(True, linestyle='--', alpha=0.6)

plt.tight_layout()
plt.show()
```

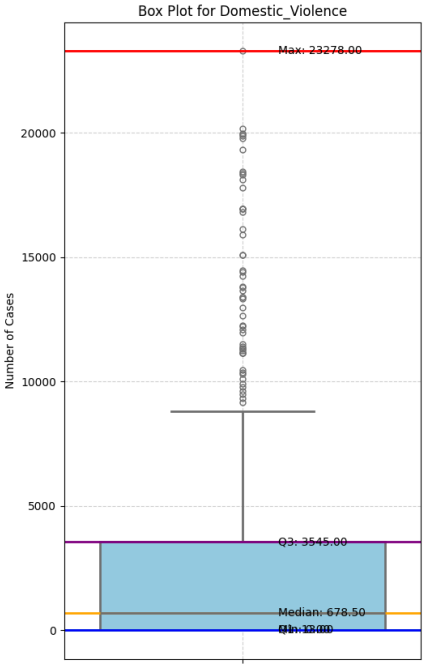

Box Plot for Rape
Number of Cases
Max: 6337.00
Q3: 1069.00
Median: 348.50
Q1: 35.00
Min: 0.00

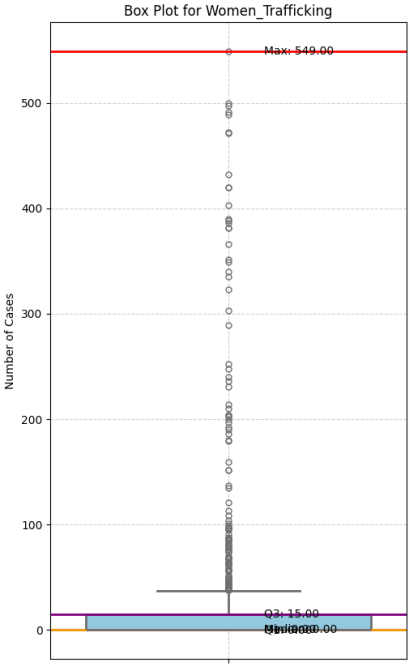

Box Plot for Kidnap_and_Assault
Number of Cases
Max: 15381.00
Q3: 1216.00
Median: 290.00


Box Plot for Dowry_Deaths
Number of Cases
Max: 2524.00
Q3: 259.00
Median: 29.00

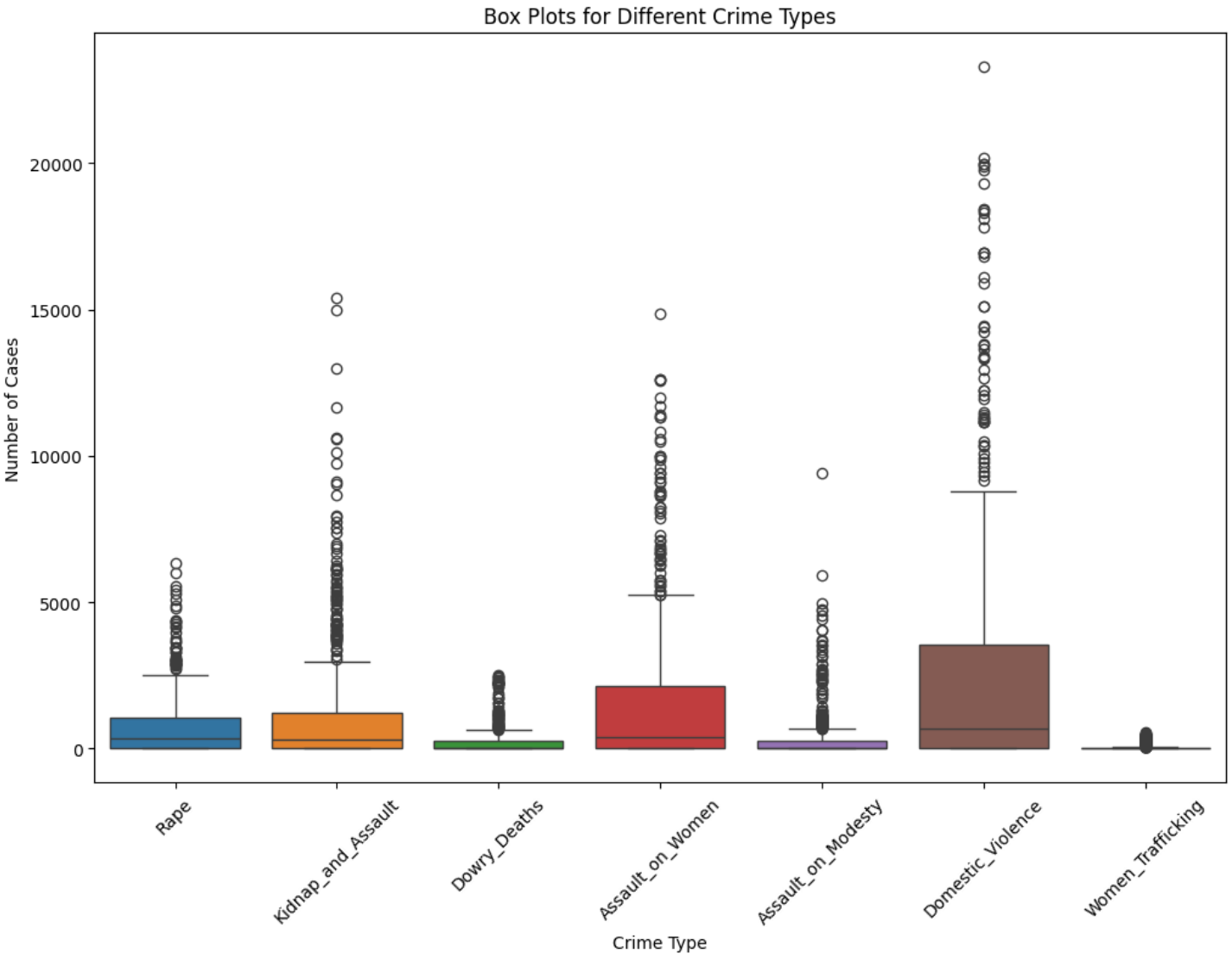

Box Plot for Assault_on_Women
Number of Cases
Max: 14853.00
Q3: 2122.25
Median: 387.50


Box Plot for Assault_on_Modesty
Number of Cases
Max: 9422.00
Q3: 277.50


Box Plot for Domestic_Violence
Number of Cases
Max: 23278.00
Q3: 3545.00
Median: 678.50


Box Plot for Women_Trafficking
Number of Cases
Max: 549.00
Q3: 15.00

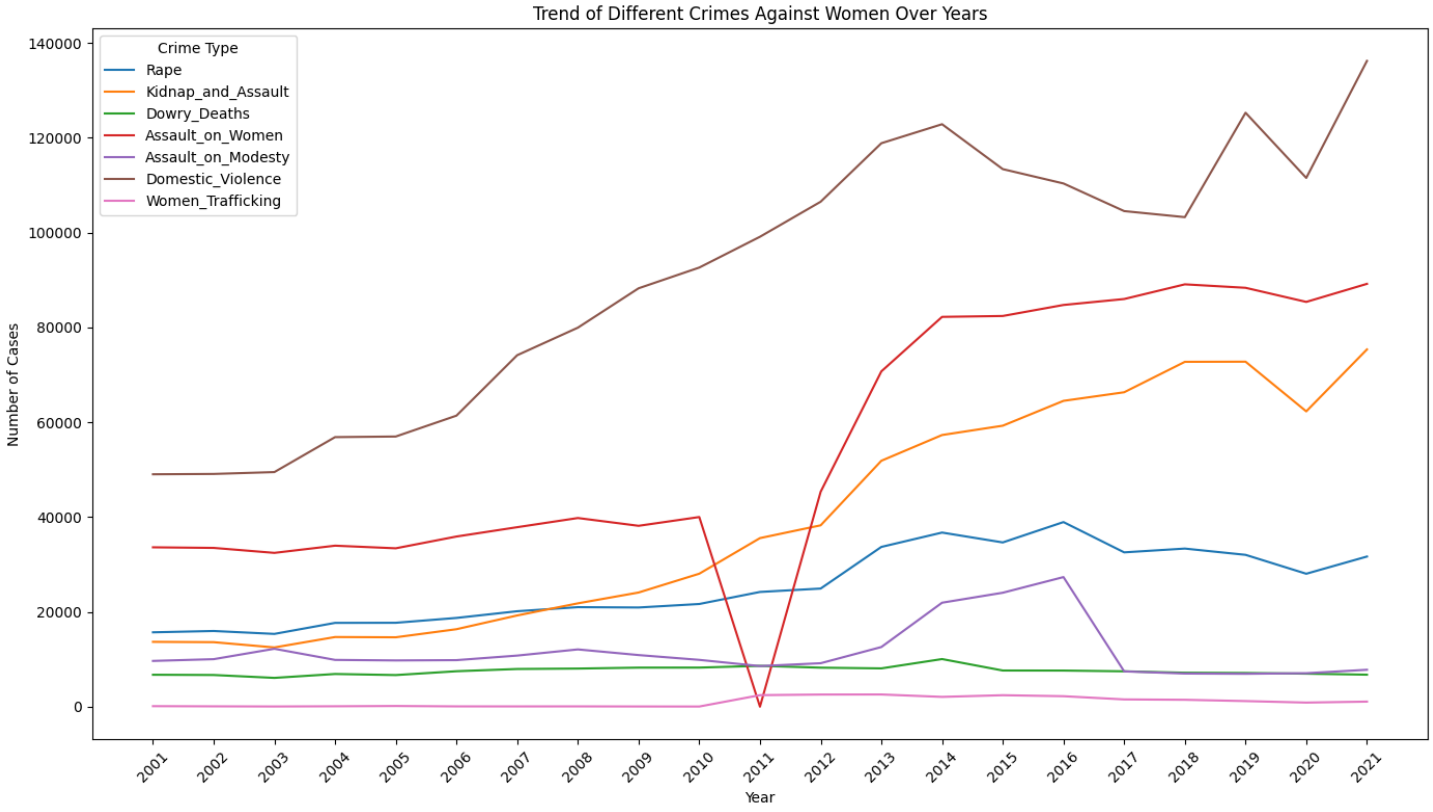
```
[52] # Box plot for all crimes combined
     plt.figure(figsize=(12, 8))
     sns.boxplot(data=df[crime_columns])
     plt.title('Box Plots for Different Crime Types')
     plt.xlabel('Crime Type')
     plt.ylabel('Number of Cases')
     plt.xticks(rotation=45)
     plt.show()
```



This shows the Box Plots for different crime types in a single plot making it easier for comparison. The smaller circles represent Outliers.

7. Trends of Different Crimes against women over years

```
# Line plots for different crime types over years
plt.figure(figsize=(14, 8))

for crime in crime_types:
    yearly_data = df.groupby('Year')[crime].sum().reset_index()
    sns.lineplot(data=yearly_data, x='Year', y=crime, label=crime)

plt.title('Trend of Different Crimes Against Women Over Years')
plt.xlabel('Year')
plt.ylabel('Number of Cases')
plt.legend(title='Crime Type')
plt.xticks(yearly_crimes['Year'], rotation=45)
plt.tight_layout()
plt.show()
```



STATISTICAL DATA ON DIFFERENT CRIME TYPES :

For each crime type, the following statistical measures are computed:

- **Mean (mean_value)**: The average number of cases reported for that crime type.
- **Median (median_value)**: The middle value when all the cases are sorted in ascending order.
- **Quartiles (q1, q2, q3)**:
  - **Q1 (1st Quartile):** Represents the 25th percentile, meaning 25% of the data falls below this value.
  - **Q2 (Median, 2nd Quartile):** Represents the 50th percentile, i.e., the median value.
  - **Q3 (3rd Quartile):** Represents the 75th percentile, meaning 75% of the data falls below this value.
- **Standard Deviation (std_dev)**: Measures how spread out the numbers are from the mean.
- **Mode (mode_value)**: The value that appears most frequently in the dataset for that crime type.
- **Variance (variance_value)**: A measure of the spread between numbers in the dataset, indicating how far each number in the set is from the mean.
- **Interquartile Range (IQR, iqr_value)**: The difference between the 3rd Quartile (Q3) and the 1st Quartile (Q1). It measures the range within which the central 50% of the data lies.
- **Coefficient of Variation (covariance_value)**: CV expresses the standard deviation as a percentage of the mean and helps in comparing variability across different datasets.

--- Statistics for Rape ---

Minimum: 0
Maximum: 6337
Mean: 727.8559782608696
Median: 348.5
1st Quartile (Q1): 35.0
2nd Quartile (Median, Q2): 348.5
3rd Quartile (Q3): 1069.0
Standard Deviation: 977.0249446635557
Mode: 0
Variance: 954577.7424948241
Interquartile Range (IQR): 1034.0
Coefficient of variation: 134.23327881403785

--- Statistics for Kidnap_and_Assault ---
Minimum: 0
Maximum: 15381
Mean: 1134.5421195652175
Median: 290.0
1st Quartile (Q1): 24.75
2nd Quartile (Median, Q2): 290.0
3rd Quartile (Q3): 1216.0
Standard Deviation: 1993.5368278358478
Mode: 0
Variance: 3974189.0839378145
Interquartile Range (IQR): 1191.25
Coefficient of variation: 175.712897164172

--- Statistics for Dowry_Deaths ---
Minimum: 0
Maximum: 2524
Mean: 215.6929347826087
Median: 29.0
1st Quartile (Q1): 1.0
2nd Quartile (Median, Q2): 29.0
3rd Quartile (Q3): 259.0
Standard Deviation: 424.9273336889511
Mode: 0
Variance: 180563.2389160012
Interquartile Range (IQR): 258.0
Coefficient of variation: 197.00568037484598

--- Statistics for Assault_on_Women ---
Minimum: 0
Maximum: 14853
Mean: 1579.1154891304348
Median: 387.5
1st Quartile (Q1): 34.0
2nd Quartile (Median, Q2): 387.5
3rd Quartile (Q3): 2122.25
Standard Deviation: 2463.962518263218
Mode: 0
Variance: 6071111.291406019
Interquartile Range (IQR): 2088.25
Coefficient of variation: 156.03434550692924

--- Statistics for Assault_on_Modesty ---
Minimum: 0
Maximum: 9422
Mean: 332.7228260869565
Median: 31.0
1st Quartile (Q1): 3.0
2nd Quartile (Median, Q2): 31.0
3rd Quartile (Q3): 277.5
Standard Deviation: 806.0245514200335
Mode: 0
Variance: 649675.5774918663
Interquartile Range (IQR): 274.5
Coefficient of variation: 242.25105349681675

--- Statistics for Domestic_Violence ---
Minimum: 0
Maximum: 23278
Mean: 2595.078804347826
Median: 678.5
1st Quartile (Q1): 13.0
2nd Quartile (Median, Q2): 678.5
3rd Quartile (Q3): 3545.0
Standard Deviation: 4042.004953332074
Mode: 3
Variance: 16337804.04276102
Interquartile Range (IQR): 3532.0
Coefficient of variation: 155.75653990006202

--- Statistics for Women_Trafficking ---
Minimum: 0
Maximum: 549
Mean: 28.744565217391305
Median: 0.0
1st Quartile (Q1): 0.0
2nd Quartile (Median, Q2): 0.0
3rd Quartile (Q3): 15.0
Standard Deviation: 79.99965967836516
Mode: 0
Variance: 6399.945548654244
Interquartile Range (IQR): 15.0
Coefficient of variation: 278.31229685799184

6.OBSERVATIONS

1. The number of crimes has been increasing continuously from 2001 to 2021 with a significant rise during the term of 2011-2012.
2. Uttar Pradesh is the state with highest number of crime cases during the time period of 2001-2021.
3.High Variability: Significant disparities exist in crime reporting across regions, with some areas showing much higher numbers than others.
4. Wide Ranges: The data shows broad variability, with maximum values indicating concentrated crime in certain regions, while others report few or no cases.
5. Outliers: Large differences between median and maximum values suggest that a few regions may have disproportionately high crime numbers.

6. Potential Underreporting: Low median and mode values, especially in categories like women trafficking, may indicate underreporting in several regions.
7.The order of type of crimes against women from 2001-2021 are as follows.
Domestic_Violence > Assault_on_Women > Rape > Kidnap_and_Assault > Assault_on_Modesty > Dowry_Deaths > Women_Trafficking.

7.KEY INSIGHTS

1. **Variability Across States:** The analysis shows significant variability in crime rates across different states. Some states report disproportionately higher numbers of certain crimes, indicating possible regional factors influencing these crime rates.
2. **High Outliers:** Certain states and years stand out as outliers with extremely high numbers of crimes reported. These outliers may require further investigation to understand the underlying causes.
3. **Rape and Kidnap_and_Assault:** The analysis of the dataset revealed a significant positive correlation between the number of rape cases and kidnapping and assault cases reported across various states. This correlation suggests that states with higher instances of rape also tend to have a higher number of kidnapping and assault cases. This trend may indicate underlying factors that contribute to the prevalence of both crimes, such as inadequate law enforcement, social norms, or regional vulnerabilities..

8.CONCLUSION

This report highlights significant variability in crimes against women across India from 2001 to 2021, with some states reporting disproportionately higher rates. A notable correlation between rape and kidnapping and assault suggests that regions with higher instances of one crime often experience elevated levels of the other. These insights underscore the need for targeted interventions and comprehensive strategies to address the root causes of violence against women, ultimately contributing to a safer and more equitable society.

CODE AVAILABLE HERE – Google Colab Notebook