NAME: Velaga sai krishna kowshik

- REG NO: 21BCE9150
- CAMPUS: VIT-AP
- Assignment 2

```
# Importing the Data Visualization libraries
import seaborn as sns # importing the seaborn library
import matplotlib.pyplot as plt # importing the matplotlib.pyplot library
```

```
print(sns.get_dataset_names()) # Finding the inbuilt datasets in seaborn library
```

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes', 'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri', 'g
```

```
df = sns.load_dataset('car_crashes') # Loading the dataset into variable 'df'
```

```
df # Printing the dataset
```

12	15.3	5.508	4.437	13.005	14.994	641.96	82.75	ID
13	12.8	4.608	4.352	12.032	12.288	803.11	139.15	IL
14	14.5	3.625	4.205	13.775	13.775	710.46	108.92	IN
15	15.7	2.669	3.925	15.229	13.659	649.06	114.47	IA
16	17.8	4.806	4.272	13.706	15.130	780.45	133.80	KS
17	21.4	4.066	4.922	16.692	16.264	872.51	137.13	KY
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
19	15.1	5.738	4.530	13.137	12.684	661.88	96.57	ME
20	12.5	4.250	4.000	8.875	12.375	1048.78	192.70	MD
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.63	MA
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26	MI
23	9.6	2.208	2.784	8.448	8.448	777.18	133.35	MN
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77	MS
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45	MO
26	21.4	8.346	9.416	17.976	18.190	816.21	85.15	MT
27	14.9	1.937	5.215	13.857	13.410	732.28	114.82	NE
28	14.7	5.439	4.704	13.965	14.553	1029.87	138.71	NV
29	11.6	4.060	3.480	10.092	9.628	746.54	120.21	NH
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85	NJ
31	18.4	3.496	4.968	12.328	18.032	869.85	120.75	NM

Handling Null Values

```
df.isnull().any() # No null values, hence no need of data manipulation
```

```
total            False
speeding         False
alcohol          False
not_distracted   False
no_previous      False
ins_premium      False
ins_losses       False
abbrev           False
dtype: bool
```

40	23.9	9.082	9.799	22.944	19.359	858.97	116.29	SC

Dataset Demographics/Statistics

```
df.describe() # describing about the df, i.e; metadat of columns with count, mean, std, min etc
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
count	51.000000	51.000000	51.000000	51.000000	51.000000	51.000000	51.000000
mean	15.790196	4.998196	4.886784	13.573176	14.004882	886.957647	134.493137
std	4.122002	2.017747	1.729133	4.508977	3.764672	178.296285	24.835922
min	5.900000	1.792000	1.593000	1.760000	5.900000	641.960000	82.750000
25%	12.750000	3.766500	3.894000	10.478000	11.348000	768.430000	114.645000
50%	15.600000	4.608000	4.554000	13.857000	13.775000	858.970000	136.050000
75%	18.500000	6.439000	5.604000	16.140000	16.755000	1007.945000	151.870000
max	23.900000	9.450000	10.038000	23.661000	21.280000	1301.520000	194.780000

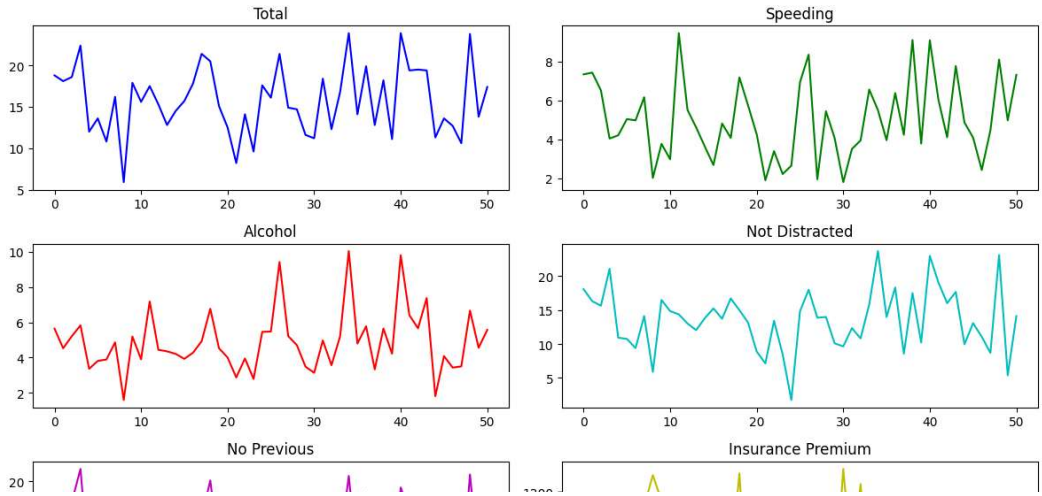
Univariate

Definition: Univariate data analysis focuses on a single variable or dataset, examining its characteristics and distribution.

Objective: The primary goal is to describe and summarize the data, understand its central tendency, and identify patterns, outliers, and potential trends within that single variable.

Methods: Common methods include histograms, bar charts, box plots, summary statistics (mean, median, mode), and measures of dispersion (variance, standard deviation)

```
plt.figure(figsize=(12, 10))

plt.subplot(4, 2, 1)
plt.plot(df['total'], 'b')
plt.title('Total')


plt.subplot(4, 2, 2)
plt.plot(df['speeding'], 'g')
plt.title('Speeding')


plt.subplot(4, 2, 3)
plt.plot(df['alcohol'], 'r')
plt.title('Alcohol')


plt.subplot(4, 2, 4)
plt.plot(df['not_distracted'], 'c')
plt.title('Not Distracted')


plt.subplot(4, 2, 5)
plt.plot(df['no_previous'], 'm')
plt.title('No Previous')



plt.subplot(4, 2, 6)
plt.plot(df['ins_premium'], 'y')
plt.title('Insurance Premium')


plt.subplot(4, 2, 7)
plt.plot(df['ins_losses'], 'k')
plt.title('Insurance Losses')


plt.tight_layout() # Used to allocate gaps between the labels and plots
```



""" Total (Blue Line): The graph shows the trend in total car crashes over the dataset. Inference: There is a noticeable variation in the total number of car crashes over time, but no specific pattern emerges. """

""" Speeding (Green Line): This graph represents the trend in car crashes caused by speeding. Inference: The number of car crashes due to speeding appears to have some fluctuations but doesn't show a consistent upward or downward trend. """

""" Alcohol (Red Line): The graph displays the trend in car crashes related to alcohol consumption. Inference: There is some variation in car crashes involving alcohol, but no clear trend is evident from the graph. """

""" Not Distracted (Cyan Line): This graph illustrates the trend in car crashes where drivers were not distracted. Inference: The number of car crashes by non-distracted drivers shows fluctuations, but no significant trend is apparent. """

""" No Previous (Magenta Line): The graph shows the trend in car crashes by drivers with no previous incidents. Inference: Car crashes by drivers with no previous incidents appear to have some fluctuations but no discernible trend. """

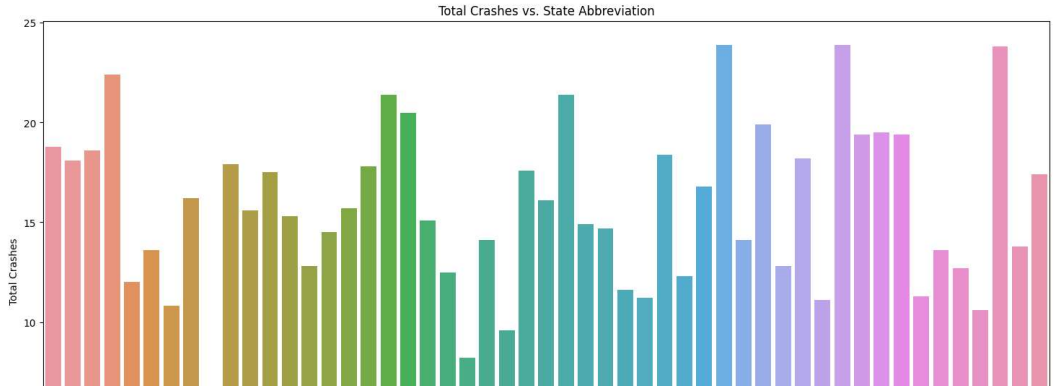
""" Insurance Premium (Yellow Line): This graph represents the trend in insurance premiums. Inference: The graph doesn't provide clear insights into the trend in insurance premiums over time, as it seems to fluctuate without a distinct pattern. """

""" Insurance Losses (Black Line): The graph displays the trend in insurance losses. Inference: Similar to insurance premiums, insurance losses also appear to fluctuate without a clear trend. """

Barplot

```
plt.figure(figsize=(18, 9))
sns.barplot(data=df,x='abbrev', y='total',errorbar=None)
plt.xlabel('State Abbreviation')
plt.ylabel('Total Crashes')
plt.title('Total Crashes vs. State Abbreviation')
```

```
Text(0.5, 1.0, 'Total Crashes vs. State Abbreviation')
```

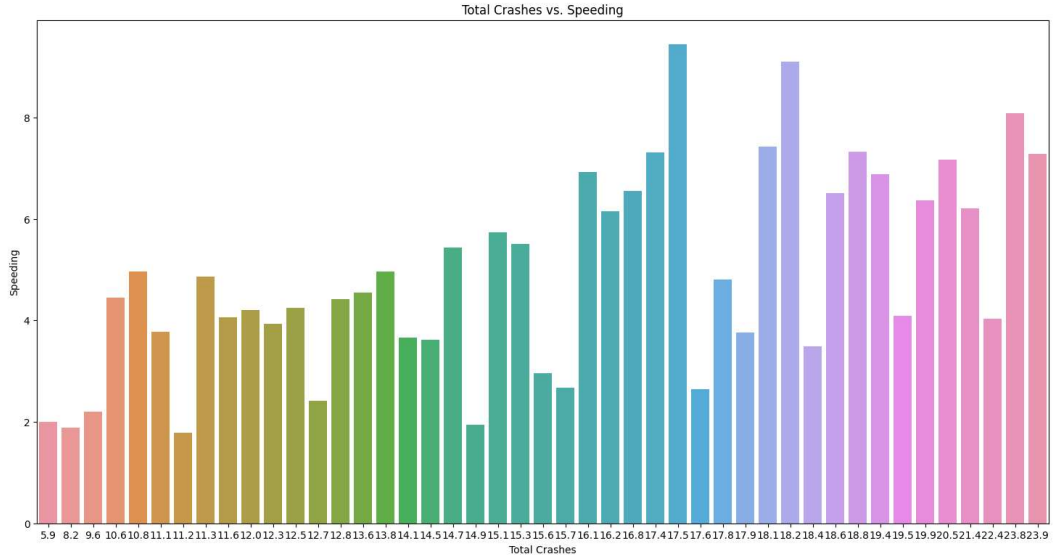


Inference: State abbreviations are on the x-axis, and the total number of crashes is on the y-axis. The plot provides a clear comparison of car crash counts between states. For example, states with abbreviations like "DC," "RI," and "NH" have relatively lower total crash counts, while "TX," "CA," and "FL" have higher crash counts. This plot is useful for identifying states with higher or lower crash rates, which can be valuable for further analysis or policy considerations.

State Abbreviation

```
plt.figure(figsize=(18, 9))
sns.barplot(data=df,x='total', y='speeding',errorbar=None)
plt.ylabel('Speeding')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Speeding')
```

```
Text(0.5, 1.0, 'Total Crashes vs. Speeding')
```

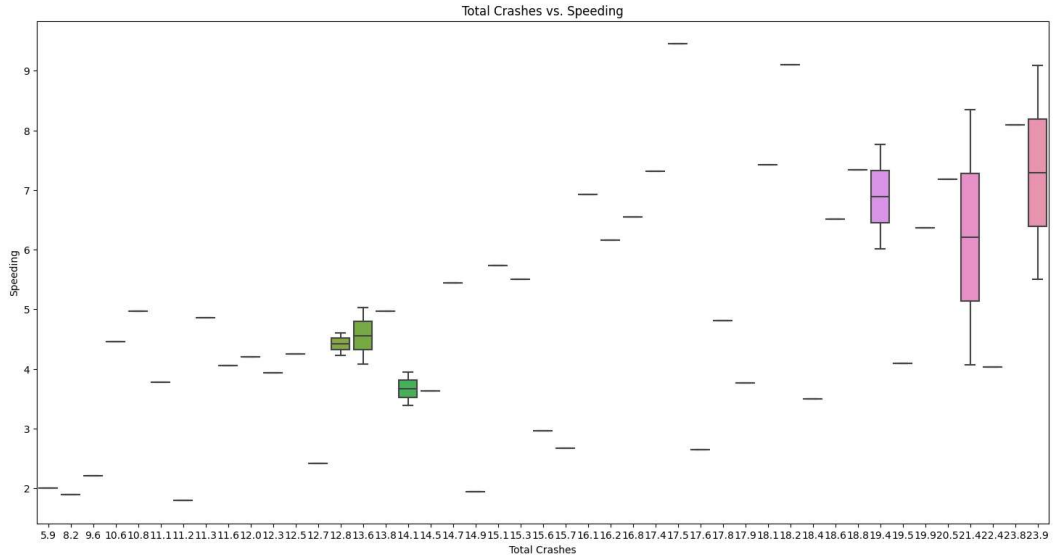


Inference: The total number of crashes is represented on the x-axis, while the number of crashes involving speeding is on the y-axis. The plot allows us to examine how speeding contributes to the overall number of car crashes. As the total number of crashes increases, there is a

general trend of an increase in the number of crashes involving speeding. This suggests that as the total number of car crashes goes up, the proportion of crashes involving speeding also tends to increase. Analyzing this relationship can help in understanding the impact of speeding on overall road safety and may inform targeted interventions to reduce speeding-related accidents.

```
plt.figure(figsize=(18,9))
sns.boxplot(x="total",y="speeding",data=df)
plt.ylabel('Speeding')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Speeding')
```

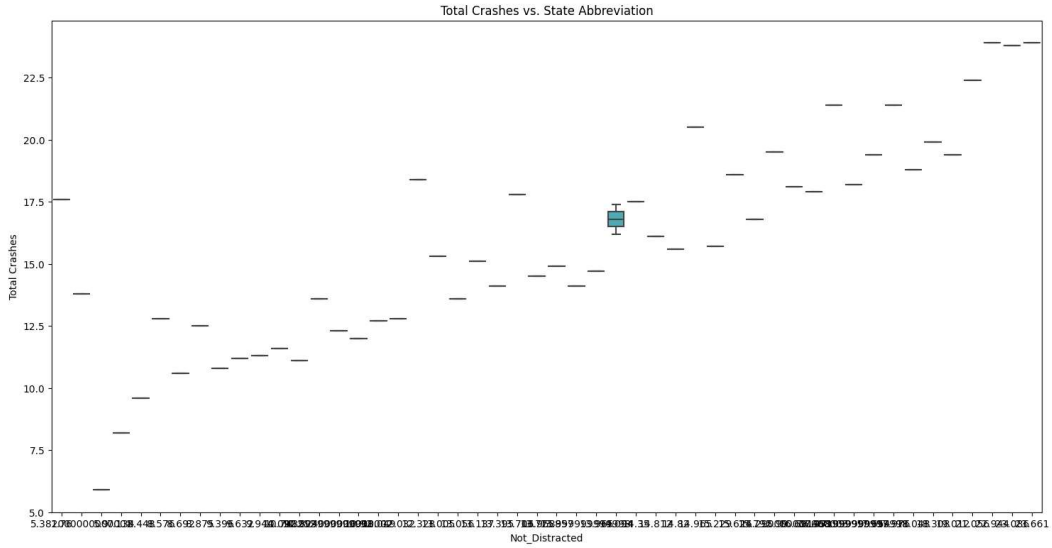
```
Text(0.5, 1.0, 'Total Crashes vs. Speeding')
```



Inference : The box plot shows the distribution of speeding-related crashes within different total crash categories. As the total number of crashes increases, there is increasing variability in the number of crashes involving speeding. This highlights the relationship between total crashes and speeding incidents, indicating the need for targeted interventions in states or situations with higher variability.

```
plt.figure(figsize=(18,9))
sns.boxplot(x="not_distracted",y="total",data=df)
plt.xlabel('Not_Distracted')
plt.ylabel('Total Crashes')
plt.title('Total Crashes vs. State Abbreviation')
```

```
Text(0.5, 1.0, 'Total Crashes vs. State Abbreviation')
```

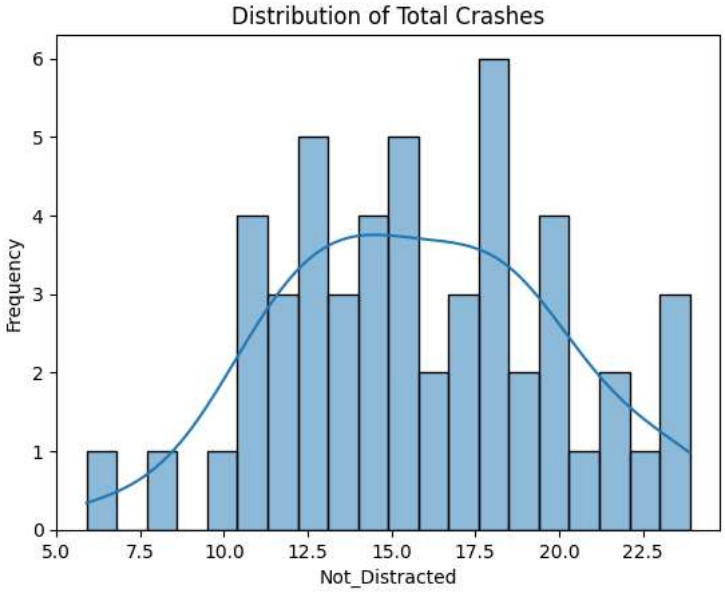


Inference : The box plot illustrates the distribution of total crashes concerning the distraction status of drivers (Not Distracted). It provides insights into how distraction affects the total number of car crashes. The plot shows varying total crash counts based on the distraction status, with potentially higher crashes when drivers are not distracted. This suggests that non-distracted drivers may be involved in more crashes, emphasizing the need for examining the causes of distraction and driving behavior to improve road safety.

Histogram

```
sns.histplot(data=df, x='total', bins=20, kde=True)
plt.xlabel('Not_Distracted')
plt.ylabel('Frequency')
plt.title('Distribution of Total Crashes')
```

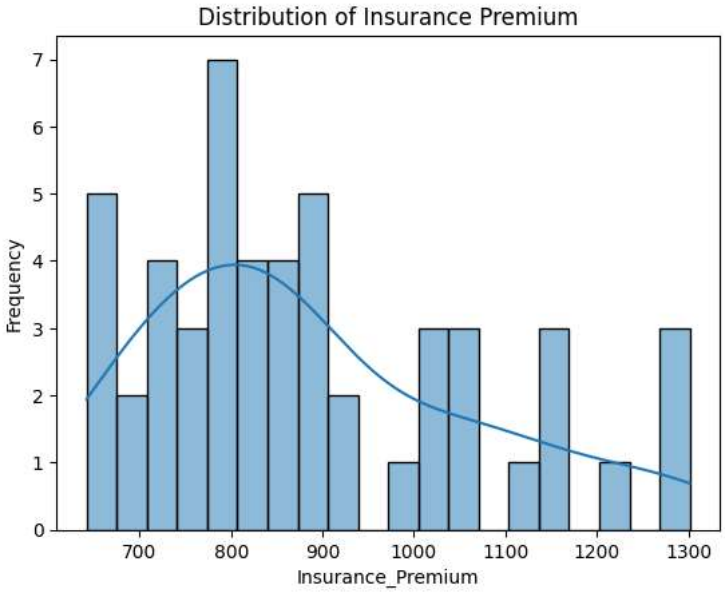
```
Text(0.5, 1.0, 'Distribution of Total Crashes')
```



Inference : The histogram displays the distribution of total car crashes.n. The plot shows that the majority of observations fall within a relatively low range of total crashes, with a peak in frequency. There is a right-skewed distribution, indicating that a few instances have significantly higher crash counts. This visualization helps understand the distribution of total crashes, which can be useful for identifying common crash count ranges and outliers in the dataset.

```
sns.histplot(data=df, x='ins_premium', bins=20, kde=True)
plt.xlabel('Insurance_Premium')
plt.ylabel('Frequency')
plt.title('Distribution of Insurance Premium')
```

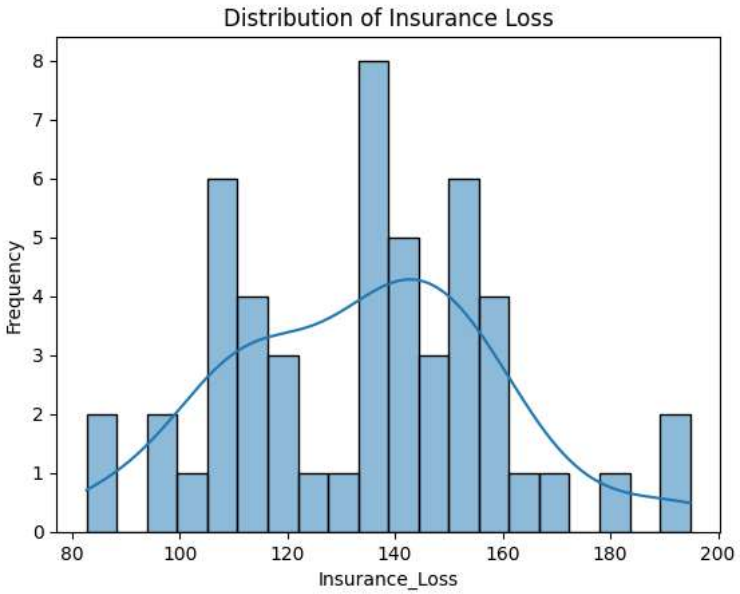
```
Text(0.5, 1.0, 'Distribution of Insurance Premium')
```



Inference : The histogram depicts the distribution of insurance premiums. The plot shows that the most common insurance premium ranges have higher frequencies, forming peaks in the distribution. The distribution appears to be right-skewed, suggesting that a few observations have exceptionally high insurance premiums. This visualization aids in understanding the distribution of insurance premiums within the dataset, providing insights into common premium ranges and potential outliers.

```
sns.histplot(data=df, x='ins_losses', bins=20, kde=True)
plt.xlabel('Insurance_Loss')
plt.ylabel('Frequency')
plt.title('Distribution of Insurance Loss')
```

```
Text(0.5, 1.0, 'Distribution of Insurance Loss')
```

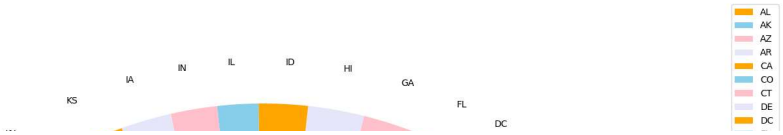


Inference : The histogram represents the distribution of insurance losses. The plot indicates that the majority of insurance losses fall within specific ranges, with peaks in frequency. The distribution appears right-skewed, indicating that a few instances have considerably higher insurance losses. This visualization helps in understanding the distribution of insurance losses within the dataset, highlighting common loss ranges and potential outliers.

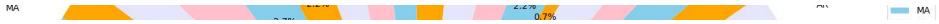
Piechart

```
fig = plt.figure(figsize=(20,20))
axes1 = fig.add_axes([0.1,0.1,0.8,0.8])  # (left,bottom,width,height)
axes1.pie(df['total'],labels=df['abbrev'],autopct='%0.1f%%',colors =['orange','skyblue','pink','lavender']) # %0.1f%% specifies percentage up
axes1.legend()
```

```
<matplotlib.legend.Legend at 0x7a55da3d3e80>
```



Inference : The pie chart visualizes the distribution of total car crashes across different states, represented by their abbreviations. Each slice of the pie represents a state, and the size of the slice corresponds to the percentage of total crashes in that state. The labels on the chart indicate the state abbreviations. The legend provides a key to identify which state each slice represents. This pie chart allows for a quick comparison of the contribution of each state to the total number of car crashes in the datase



Bivariate

Definition: Bivariate data analysis involves the analysis of two variables to explore their relationship and interactions.

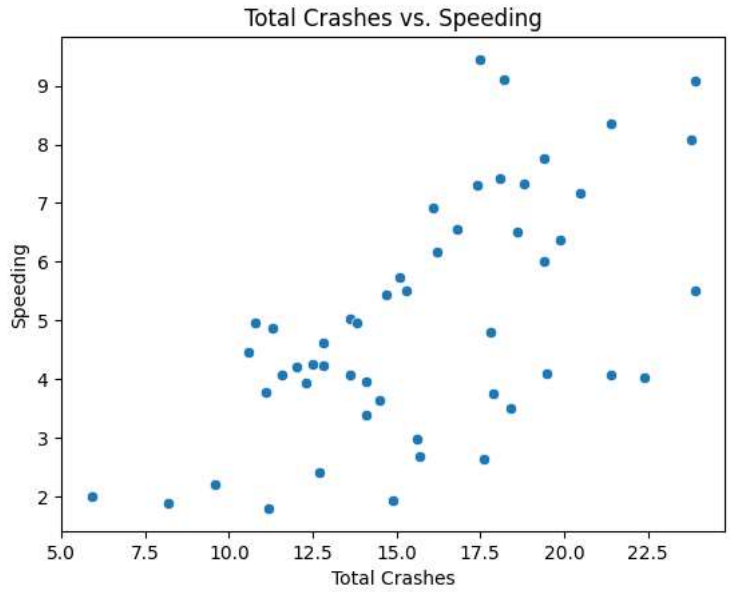
Objective: The primary goal is to understand how two variables are related, whether they exhibit correlation or causation, and to identify patterns or associations between them.

Methods: Common methods include scatter plots, line graphs, correlation coefficients (e.g., Pearson correlation), and hypothesis tests (e.g., t-tests) to determine if relationships are statistically significant.

Scatterplot


```
sns.scatterplot(x="total",y='speeding',data=df)
plt.ylabel('Speeding')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Speeding')
```

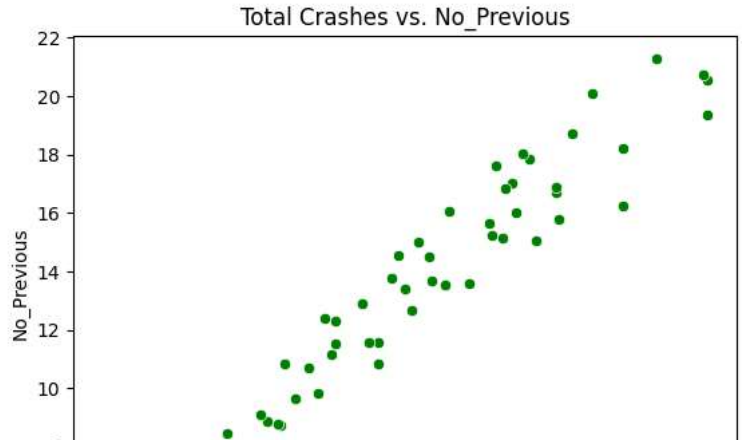
```
Text(0.5, 1.0, 'Total Crashes vs. Speeding')
```



Inference : The scatter plot visualizes the relationship between the total number of car crashes and the number of crashes involving speeding. There doesn't appear to be a strong linear relationship between total crashes and speeding incidents based on this scatter plot. The points are scattered across the plot without a clear trend, suggesting that total crashes and speeding may not be strongly correlated. Further statistical analysis may be needed to quantify the relationship between these variables accurately.

```
sns.scatterplot(x="total",y='no_previous',data=df,c='g')
plt.ylabel('No_Previous')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. No_Previous')
```

```
Text(0.5, 1.0, 'Total Crashes vs. No_Previous')
```

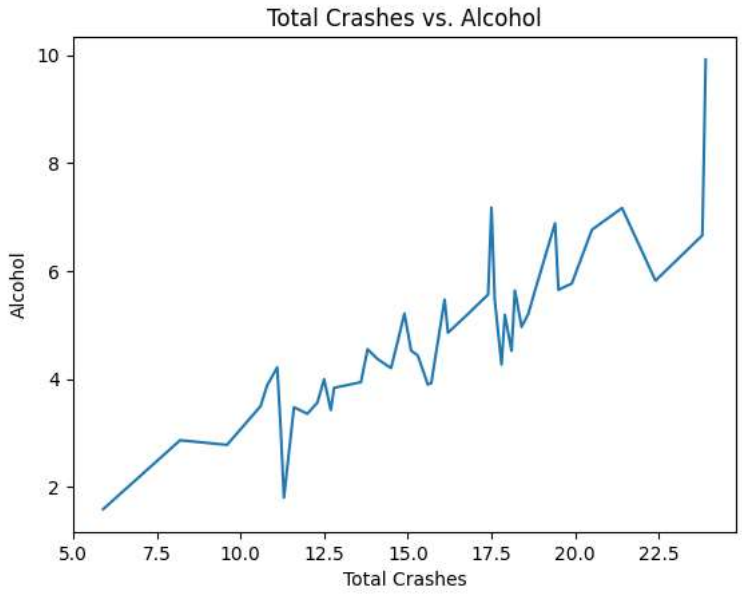


Inference : The scatter plot illustrates the relationship between the total number of car crashes and crashes involving drivers with no previous incidents. Similar to previous scatter plots, there isn't a distinct linear relationship between total crashes and crashes involving drivers with no previous incidents. The points are scattered without a clear trend, suggesting that total crashes may not directly correlate with the absence of previous incidents in drivers. Further analysis may be needed.

Lineplot

```
sns.lineplot(x="total",y="alcohol",data=df,errorbar=None)
plt.ylabel('Alcohol')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Alcohol')
```

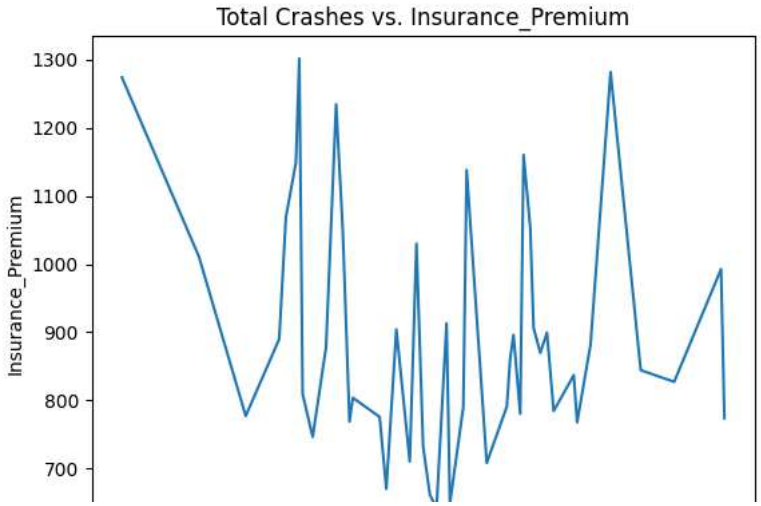
```
Text(0.5, 1.0, 'Total Crashes vs. Alcohol')
```



Inference : The line plot shows the association between total car crashes and crashes involving alcohol. It visualizes how alcohol-related crashes fluctuate concerning the total number of crashes. There isn't a clear linear relationship; the points on the line are scattered without a distinct pattern. This suggests that the total number of crashes may not have a straightforward correlation with alcohol-related incidents, warranting further analysis.

```
sns.lineplot(x="total",y="ins_premium",data=df,errorbar=None)
plt.ylabel('Insurance_Premium')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Insurance_Premium')
```

```
Text(0.5, 1.0, 'Total Crashes vs. Insurance_Premium')
```



Inference : The line plot represents the relationship between total car crashes and insurance premiums. It visualizes how insurance premiums vary in relation to the total number of crashes. The plot does not show a clear linear trend; points on the line are scattered without a clear pattern. This suggests that the total number of crashes may not have a straightforward correlation with insurance premiums, necessitating further investigation.

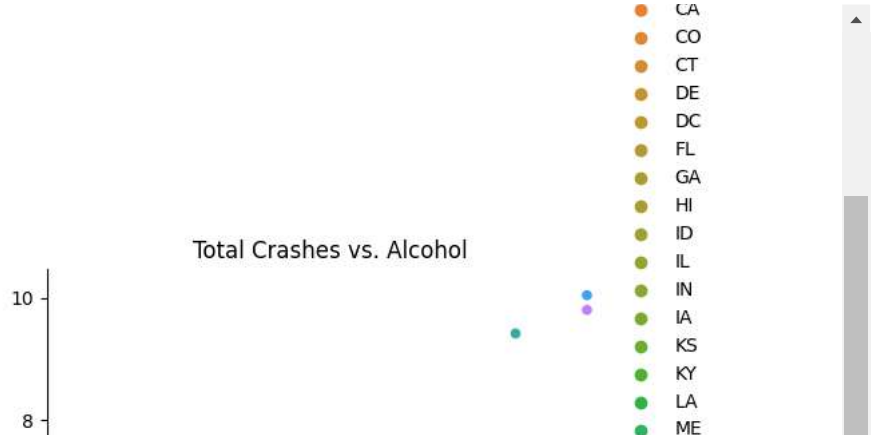
Replot

```
sns.relplot(x="total",y="speeding",data=df,hue="abbrev")
plt.ylabel('Speeding')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Speeding')
```

```
Text(0.5, 1.0, 'Total Crashes vs. Speeding')
```

Inference : The relational plot ("relplot") displays the relationship between total car crashes and crashes involving speeding. Each point represents a data point in the dataset, with different states distinguished by colors (hue). The plot allows for a quick visual assessment of how speeding-related crashes vary concerning the total number of crashes in different states. There is no clear linear trend; points are scattered without a distinct pattern, indicating that the relationship between total crashes and speeding incidents may not be straightforward and may vary by state. Further analysis may be required to explore state-specific trends.

```
sns.relplot(x="total",y="alcohol",data=df,hue="abbrev")
plt.ylabel('Alcohol')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Alcohol')
```

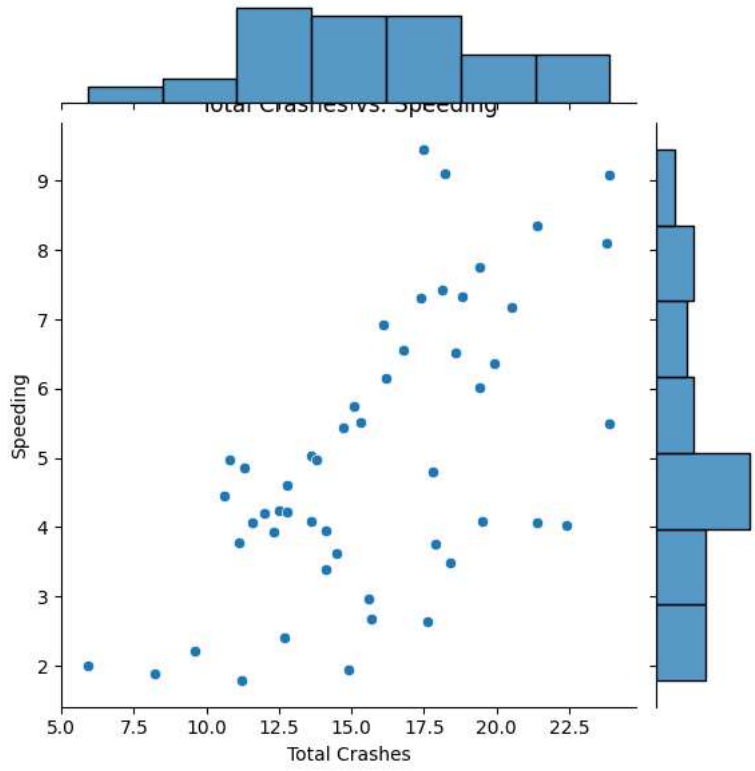


Inference : The relational plot ("relplot") illustrates the relationship between total car crashes and crashes involving alcohol. Each point on the plot represents a data point in the dataset, and different states are color-coded for comparison (hue). The plot provides a visual comparison of how alcohol-related crashes vary with the total number of crashes in different states. There isn't a clear linear trend in the relationship; points are scattered without a distinct pattern, suggesting that the association between total crashes and alcohol-related incidents may differ by state. Further state-specific analysis may be needed to explore this further.

Jointplot

```
sns.jointplot(x="total",y="speeding",data=df)
plt.ylabel('Speeding')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Speeding')
```

```
Text(0.5, 1.0, 'Total Crashes vs. Speeding')
```

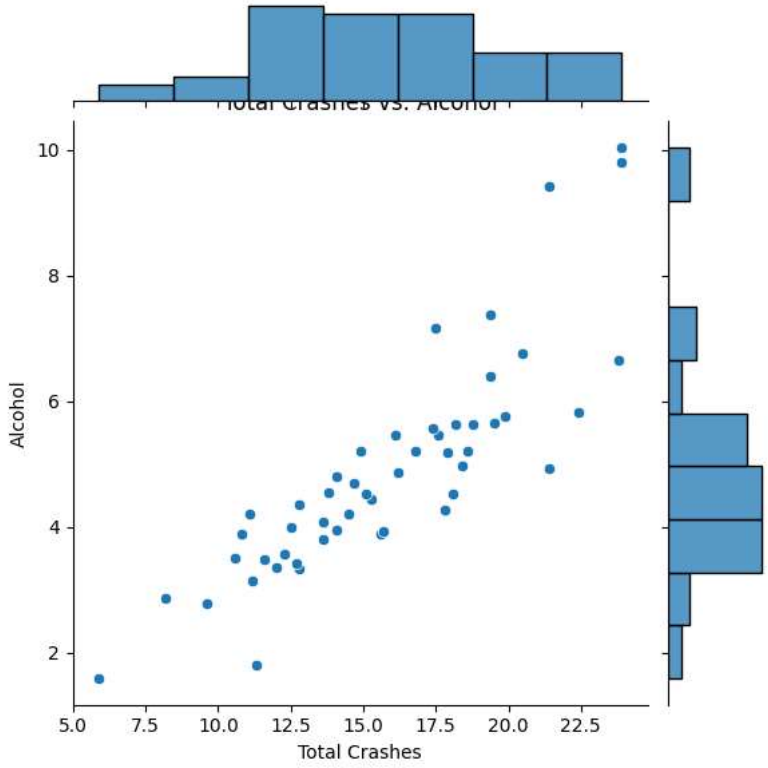


Inference : The joint plot displays the relationship between total car crashes and crashes involving speeding. It combines a scatter plot and histograms to visualize the distribution and correlation between the two variables. The scatter plot shows that there isn't a strong linear relationship between total crashes and speeding incidents. The histograms on the top and right sides provide additional information about the distributions of both variables.

```
sns.jointplot(x="total",y="alcohol",data=df)
plt.ylabel('Alcohol')
```

```
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Alcohol')
```

```
Text(0.5, 1.0, 'Total Crashes vs. Alcohol')
```



Inference : The joint plot visualizes the relationship between total car crashes and crashes involving alcohol. It combines a scatter plot and histograms to provide insights into the distribution and correlation between the two variables. The scatter plot shows that there isn't a strong linear relationship between total crashes and alcohol-related incidents. The histograms on the top and right sides offer additional information about the distributions of both variables.

Multivariate

Definition: Multivariate data analysis deals with the examination of three or more variables simultaneously, often in complex datasets.

Objective: The primary goal is to uncover intricate relationships, dependencies, and patterns involving multiple variables. It aims to explore how these variables collectively impact the outcome or phenomenon under study.

Methods: Common methods include multiple regression analysis, principal component analysis (PCA), factor analysis, cluster analysis, and machine learning techniques like decision trees, random forests, and neural networks. These methods enable the exploration of complex interactions and dependencies among multiple variables.

```
corr=df.corr() # Finding the co relation between all the fields in the dataset and storing it in the variable 'corr'.
```

```
<ipython-input-26-f8732931ad62>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future versior
  corr=df.corr() # Finding the co relation between all the fields in the dataset and storing it in the variable 'corr'.
```

```
corr # Displaying the data
```

```
                        total  speeding  alcohol  not_distracted  no_previous  ins_premium  i
plt.subplots(figsize=(18,9))
sns.heatmap(corr,annot=True)
```

```
<Axes: >
```

Inference : The heatmap visualizes the correlation between different variables in the dataset. Darker colors indicate stronger positive correlations, while lighter colors represent weaker or negative correlations. The heatmap allows for a quick assessment of which variables are strongly correlated and which are not. For example, if two variables have a dark-colored cell, it indicates a strong positive correlation between them. This visualization is valuable for identifying potential relationships and dependencies within the dataset.