

```
# 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
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbr
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	,
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	/
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	,
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	/
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	(
5	13.6	5.032	3.808	10.744	12.920	835.50	139.91	C
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.02	(
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	[
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	[
9	17.9	3.759	5.191	16.468	16.826	1160.13	144.18	
10	15.6	2.964	3.900	14.820	14.508	913.15	142.80	(
11	17.5	9.450	7.175	14.350	15.225	861.18	120.92	
12	15.3	5.508	4.437	13.005	14.994	641.96	82.75	
13	12.8	4.608	4.352	12.032	12.288	803.11	139.15	
14	14.5	3.625	4.205	13.775	13.775	710.46	108.92	

```
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
```

```
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

```
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')
```

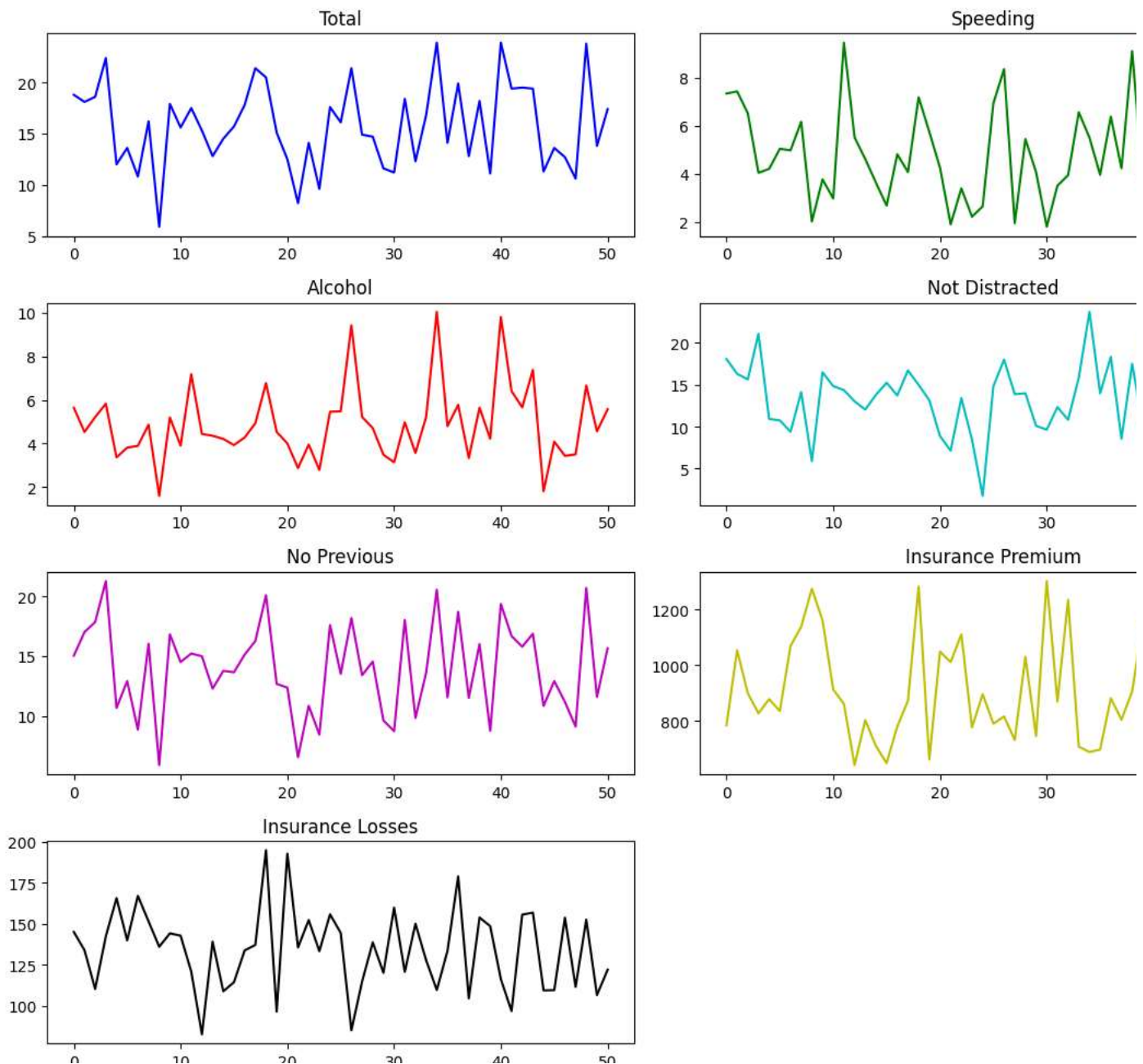
```
plt.subplot(4, 2, 4)
plt.plot(df['not_distracted'], 'c')
plt.title('Not Distracted')
```

```
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. """

```
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')
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-1-7739a7800f5d> in <cell line: 1>()
----> 1 plt.figure(figsize=(18, 9))
      2 sns.barplot(data=df,x='abbrev', y='total',errorbar=None)
      3 plt.xlabel('State Abbreviation')
      4 plt.ylabel('Total Crashes')
      5 plt.title('Total Crashes vs. State Abbreviation')

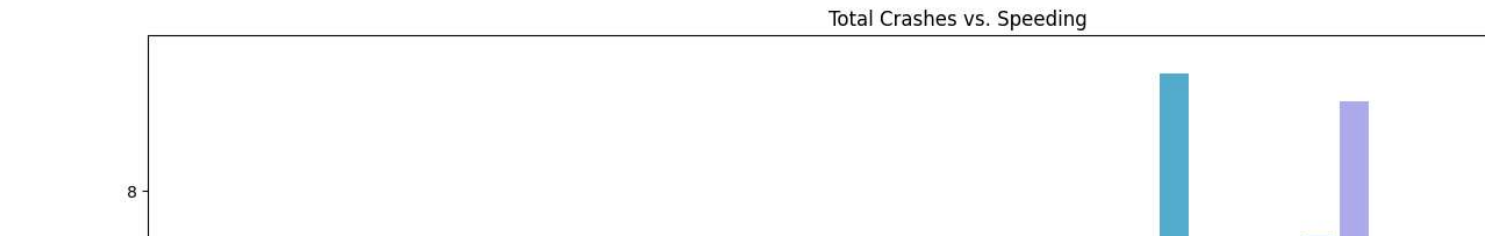
NameError: name 'plt' is not defined
```

SEARCH STACK OVERFLOW

""" 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. """

```
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')
```

'\nInference:\n\nThe 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.\n\n'



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.

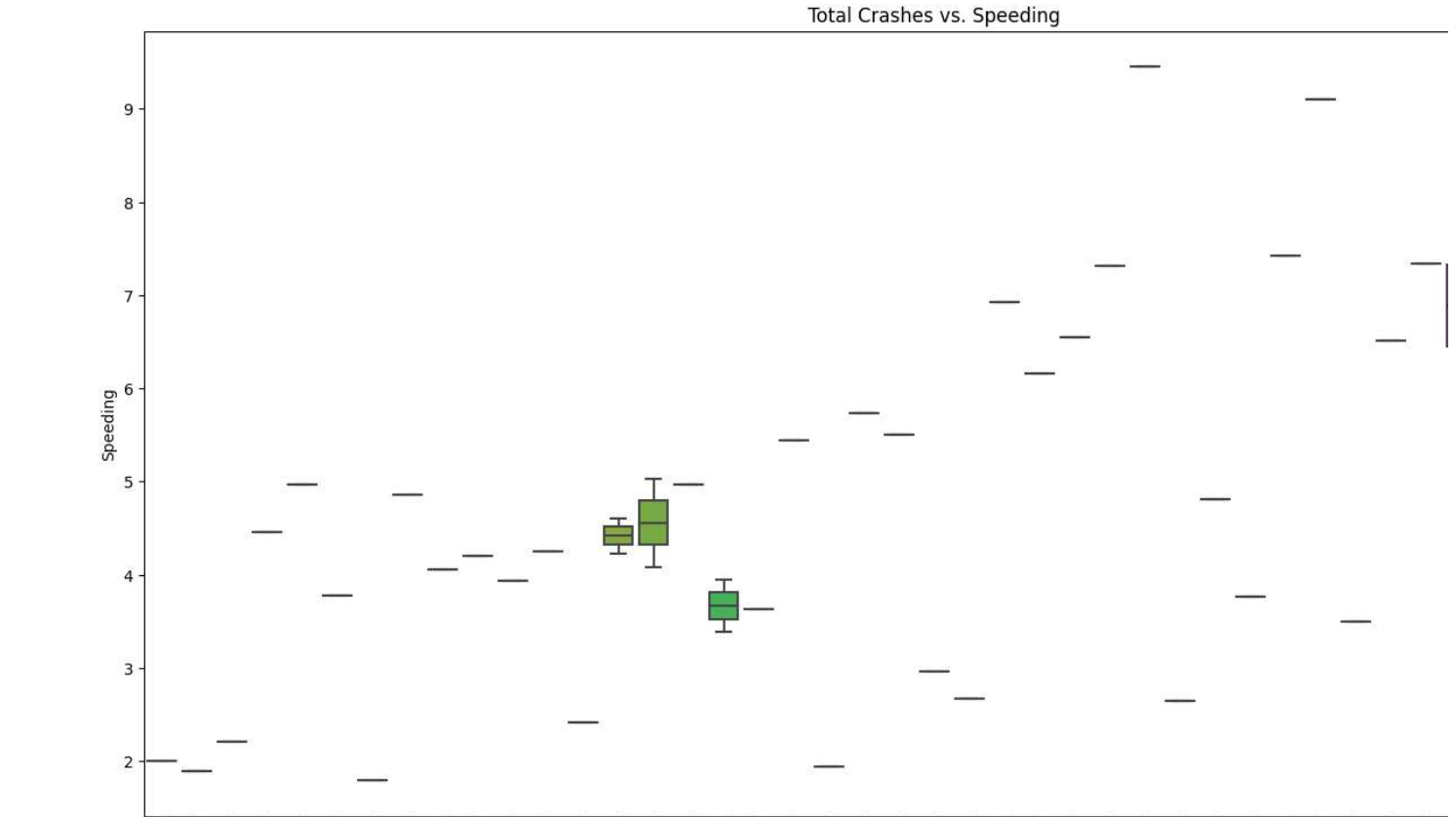


Inference: The total number of crashes is represented on the x-axis, while the number of crashes involving speeding is on the y-axis. The plot allows us to examine how speeding contributes to the overall number of car crashes. As the total number of crashes increases, there is a general trend of an increase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

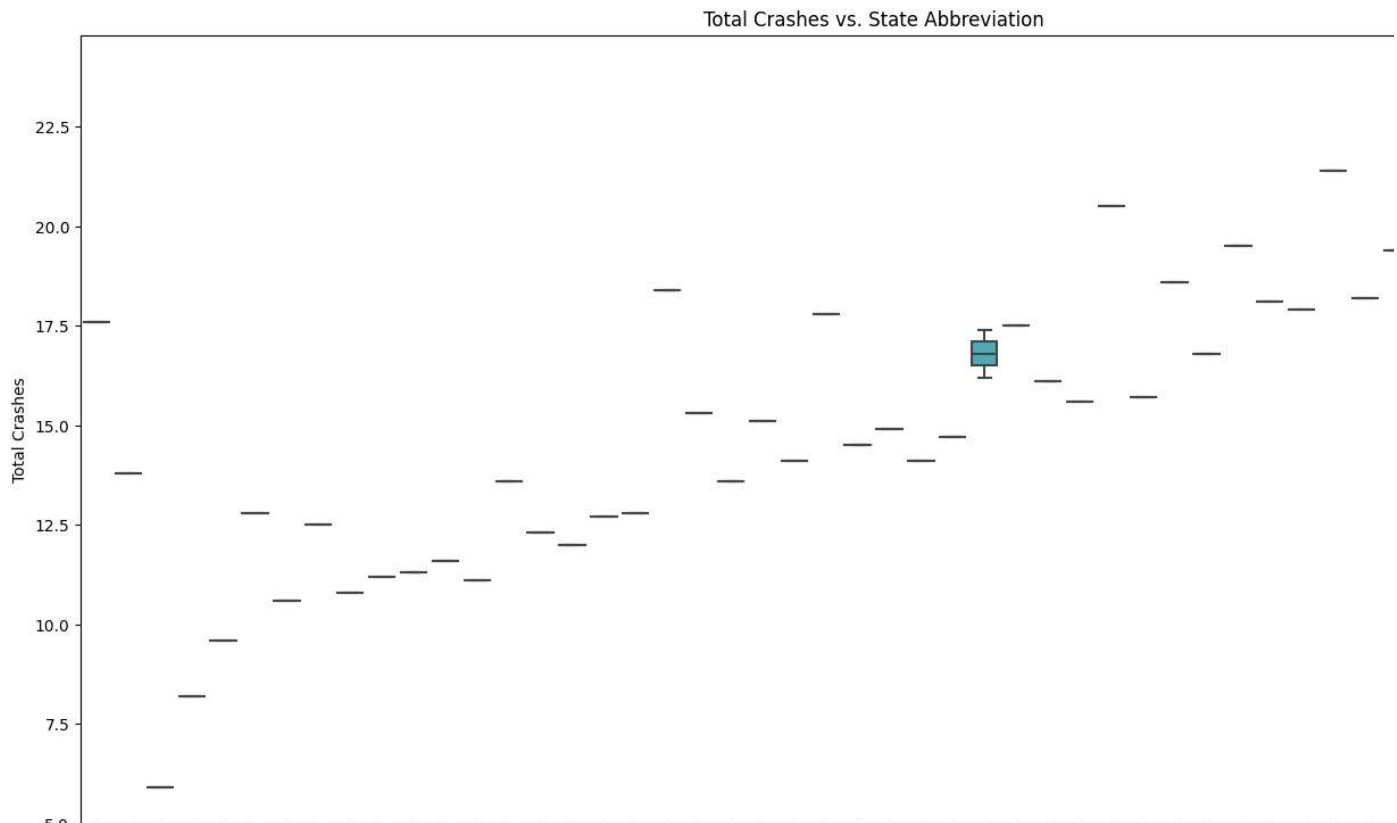
'\nInference : \n\nThe box plot shows the distribution of speeding-related crashes within different total crash categories. \n\nAs the total r there is increasing variability in the number of crashes involving speeding. \n\nThis highlights the relationship between total crashes anc ing the need for targeted interventions in states or situations with higher variability. \n\n'



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')
```

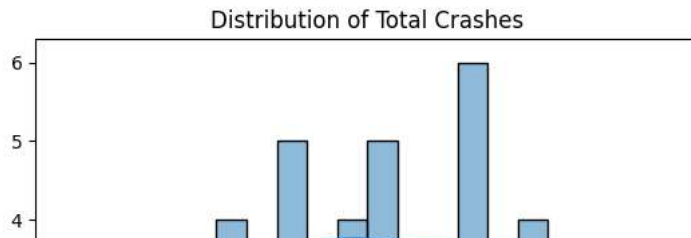
'\nInference :\n\nThe box plot illustrates the distribution of total crashes concerning the distraction status of drivers (Not Distracted). It provides insights into how distraction affects the total number of car crashes. The plot shows varying total crash counts based on the distraction status, with 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.\n'



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.

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

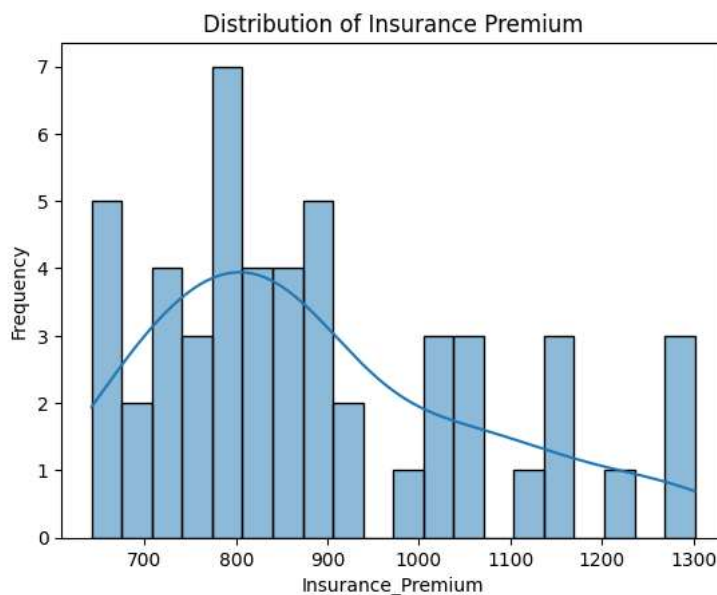
'\nInference :\n\nThe histogram displays the distribution of total car crashes.\n\nThe plot shows that the majority of observations fall within a relatively low range of total crashes, with a peak in frequency.\n\nThere is a right-skewed distribution, indicating that a few instances have significantly 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.'



Inference : The histogram displays the distribution of total car crashes. 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')
```

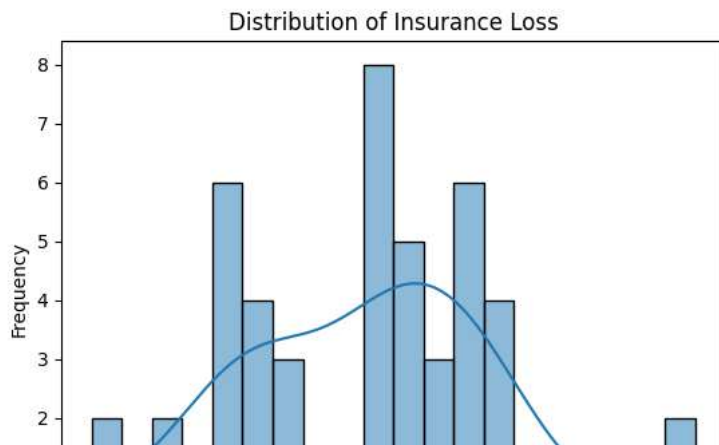
'\nInference :\n\nThe histogram depicts the distribution of insurance premiums.\n\nThe plot shows that the most common insurance premium ranges have higher frequencies, forming peaks in the distribution.\n\nThe distribution appears to be right-skewed, suggesting that a few observations have exceptionally 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.\n\n'



"" 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')
```

'\nInference :\n\nThe histogram represents the distribution of insurance losses.\n\nThe plot indicates that the majority of insurance losses s, with peaks in frequency.\n\nThe distribution appears right-skewed, indicating that a few instances have considerably higher insurance l
elps in understanding the distribution of insurance losses within the dataset, highlighting common loss ranges and potential outliers.\n\n'

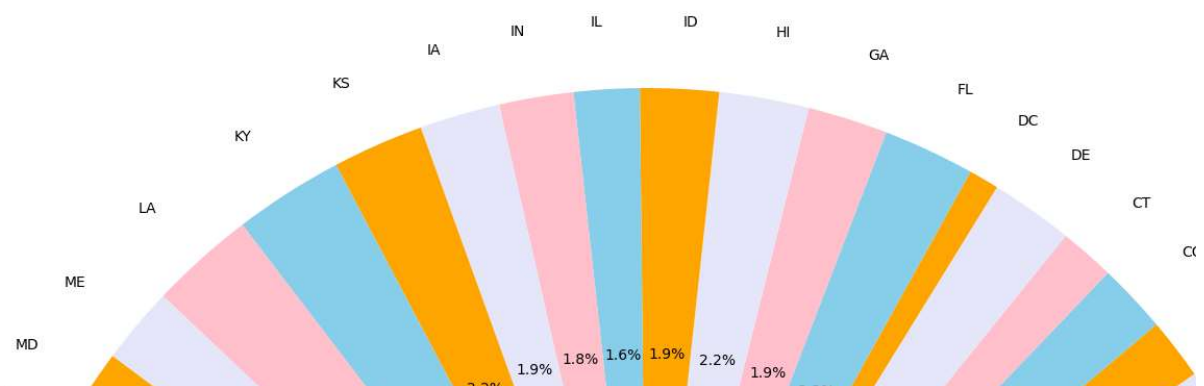


"" Inference : The histogram represents the distribution of insurance losses. The plot indicates that the majority of insurance losses fall within specific ranges, with peaks in frequency. The distribution appears right-skewed, indicating that a few instances have considerably higher insurance losses.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

```
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()
```

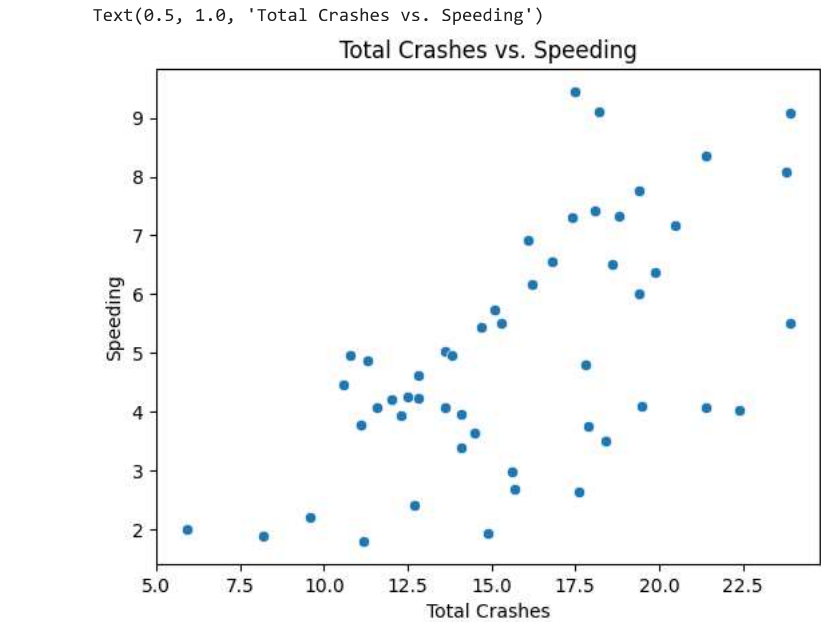


```
'''Inference :
The pie chart visualizes the distribution of total car crashes across different states, represented by their abbreviations. Each slice represents a state, and the size of the slice corresponds to the percentage of total crashes in that state. The labels on the chart indicate the state 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 dataset.'''
```



```
'''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 dataset.'''
```

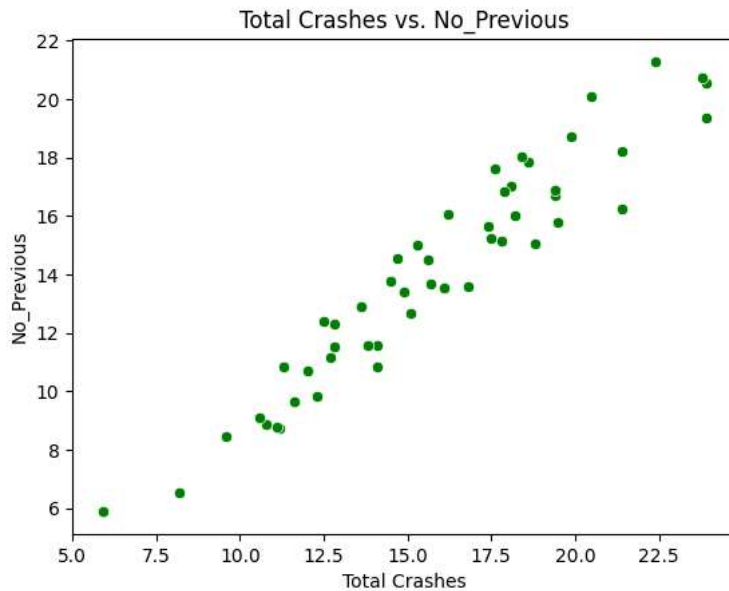
```
sns.scatterplot(x="total",y='speeding',data=df)
plt.ylabel('Speeding')
plt.xlabel('Total Crashes')
plt.title('Total Crashes vs. Speeding')
```



```
'''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')
```

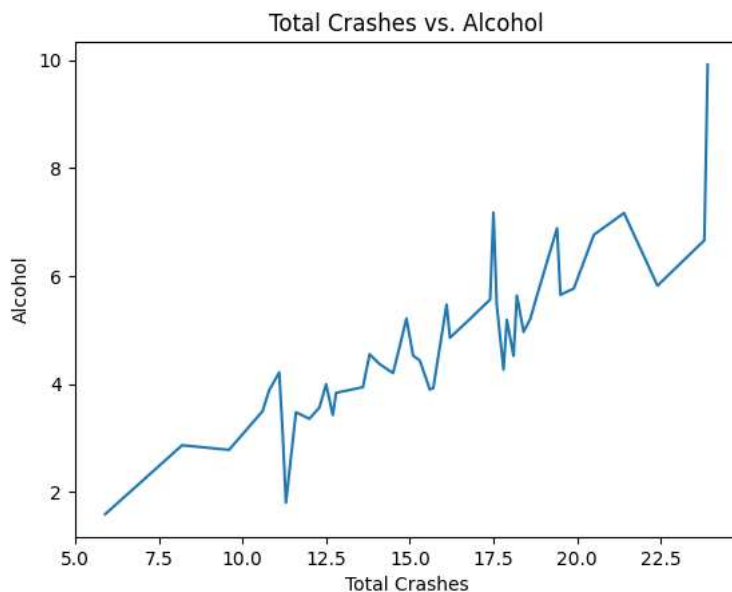
```
Text(0.5, 1.0, 'Total Crashes vs. No_Previous')
```



"" 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. ""

```
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')
```

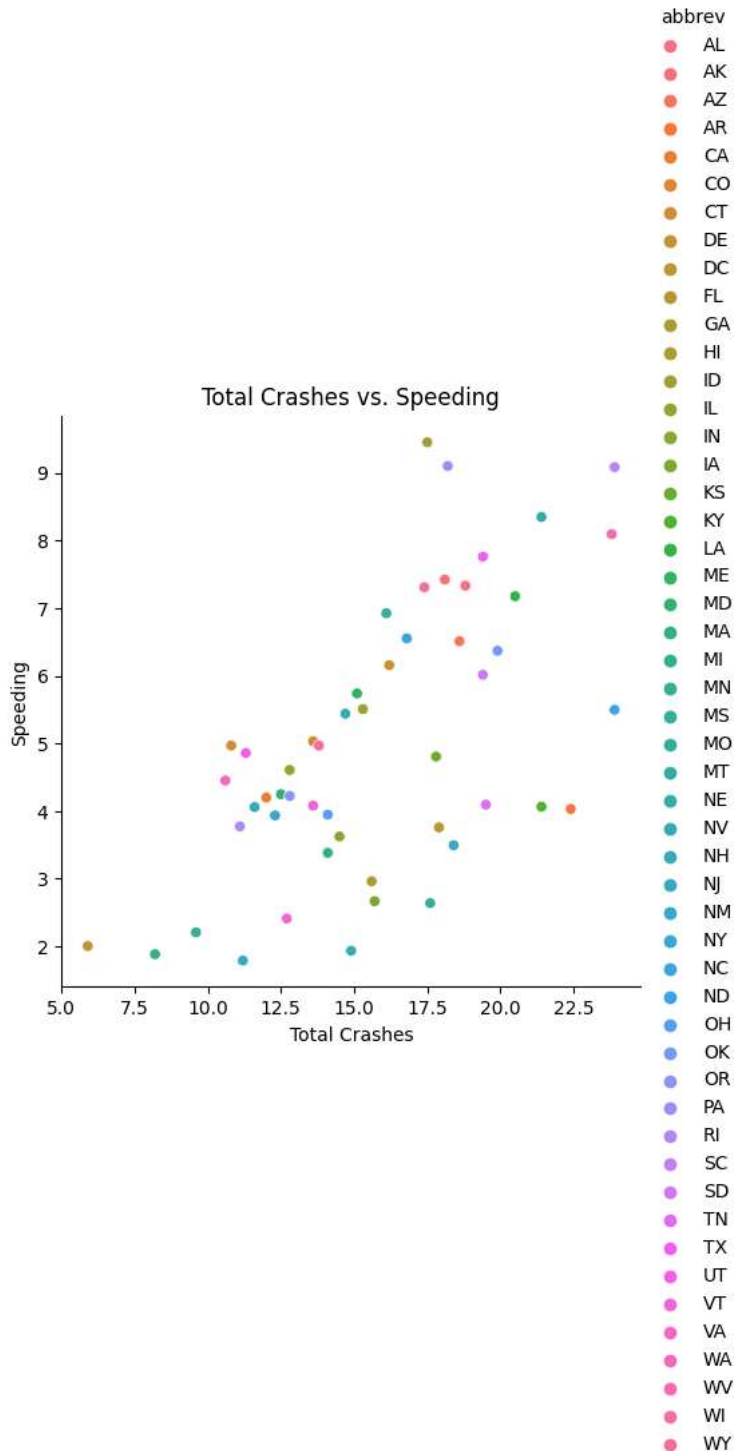


"" Inference : The line plot shows the association between total car crashes and crashes involving alcohol. It visualizes how alcohol-related crashes fluctuate concerning the total number of crashes. There isn't a clear linear relationship; the points on the line are scattered without a distinct pattern.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

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

```
plt.title('Total Crashes vs. Speeding')
```

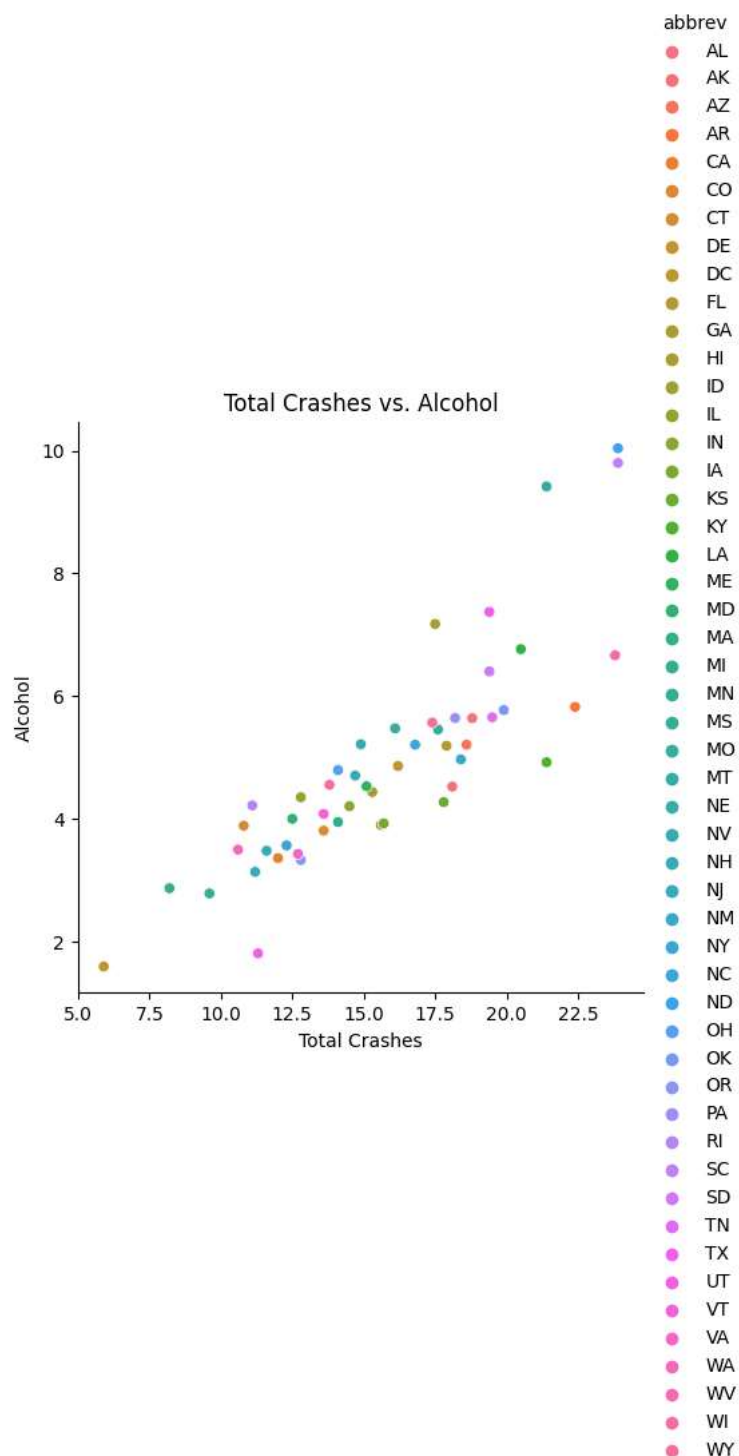
```
Text(0.5, 1.0, 'Total Crashes vs. Speeding')
```



""" 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')
```

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

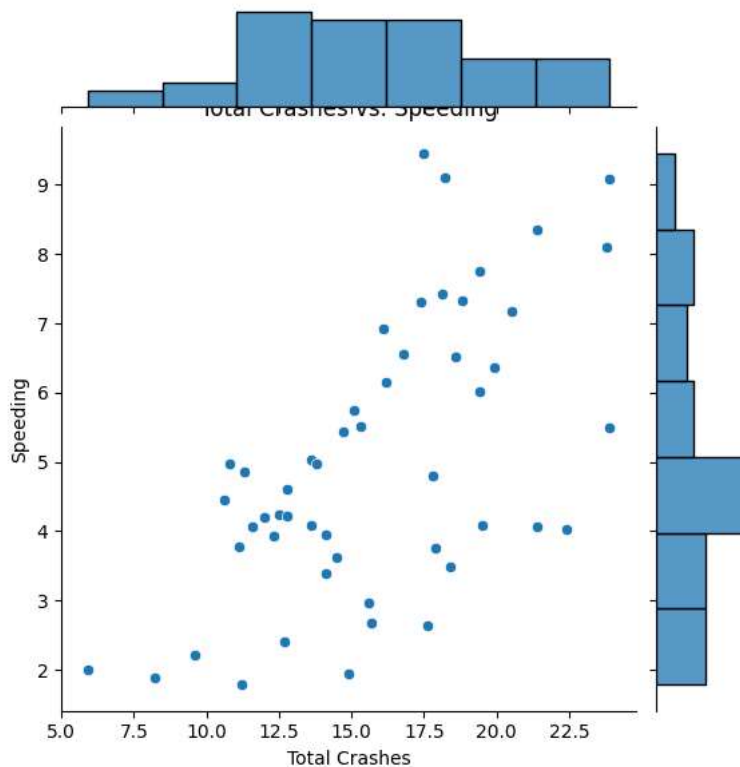


""" Inference : The relational plot ("relplot") illustrates the relationship between total car crashes and crashes involving alcohol. Each point on the plot represents a data point in the dataset, and different states are color-coded for comparison (hue). The plot provides a visual comparison of how 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. """

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

```
plt.title('Total Crashes vs. Speeding')
```

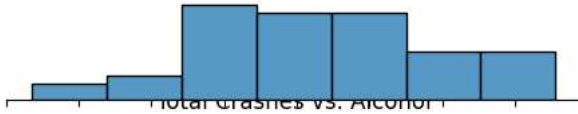
```
Text(0.5, 1.0, 'Total Crashes vs. Speeding')
```



"" 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. """

```
corr=df.corr()
```

```
corr
```

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

```
<ipython-input-44-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	ins_losses	
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.199702	-0.036011	
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.077675	-0.065928	
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.170612	-0.112547	
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	-0.174856	-0.075970	
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	-0.156895	-0.006359	
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856	-0.156895	1.000000	0.623116	
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970	-0.006359	0.623116	1.000000	

```
plt.subplots(figsize=(18,9))  
sns.heatmap(corr,annot=True)
```