


21BCE7139 VIT-AP

▼ Import neccessary libraries nedded

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
print(sns.get_dataset_names())
```

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



▼ Loading Dataset

+ Code

+ Text

```
df=sns.load_dataset('car_crashes')
```

```
df
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	a
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	
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	
15	15.7	2.669	3.925	15.229	13.659	649.06	114.47	
16	17.8	4.806	4.272	13.706	15.130	780.45	133.80	
17	21.4	4.066	4.922	16.692	16.264	872.51	137.13	
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	
19	15.1	5.738	4.530	13.137	12.684	661.88	96.57	

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   total                  51 non-null    float64
1   speeding               51 non-null    float64
2   alcohol                51 non-null    float64
3   not_distracted         51 non-null    float64
4   no_previous            51 non-null    float64
5   ins_premium            51 non-null    float64
6   ins_losses             51 non-null    float64
7   abbrev                 51 non-null    object  
dtypes: float64(7), object(1)
memory usage: 3.3+ KB

20   11.2      4.702      3.136      9.632      8.736      1301.52      150.85
```

```
df.shape

(51, 8)
```

```
df.head(5)

   total  speeding  alcohol  not_distracted  no_previous  ins_premium  ins_losses  abbrev
0    18.8     7.332    5.640         18.048        15.040        784.55     145.08      AL
1    18.1     7.421    4.525         16.290        17.014       1053.48     133.93      AK
2    18.6     6.510    5.208         15.624        17.856        899.47     110.35      AZ
3    22.4     4.032    5.824         21.056        21.280        827.34     142.39      AR
4    12.0     4.200    3.360         10.920        10.680        878.41     165.63      CA
```

Plotting Univariate Distribution

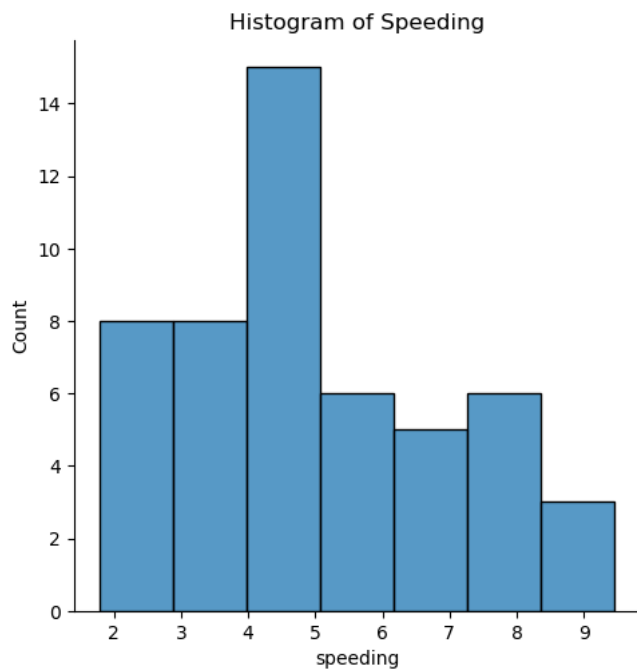
```
44   11.3      4.859      1.808      9.944      10.848      809.38      109.48

df.describe()
```

	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

Inference: it shows how many accidents took per quantile and also shows standard deviation,mean,max

```
sns.displot(df['speeding'])
plt.title("Histogram of Speeding")
plt.show()
```



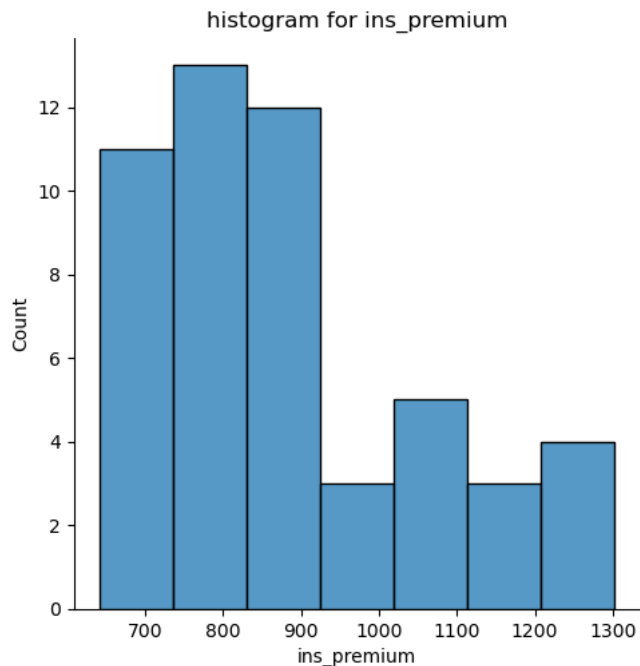
Inference: speeding ranges 4 to 5 has more count

```
sns.displot(df['alcohol'])
plt.title("Histogram of alcohol")
plt.show()
```

histogram of alcohol

Inference:alcohol range from 4 to 5 hs highest count

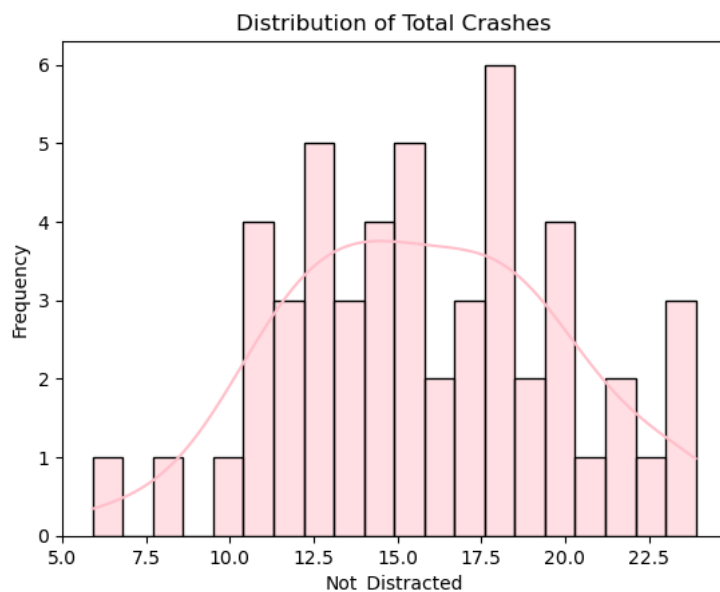
```
-- |
sns.displot(df['ins_premium'])
plt.title("Histogram for ins_premium")
plt.show()
```



Inference:car ins_premium has highest count at 800

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

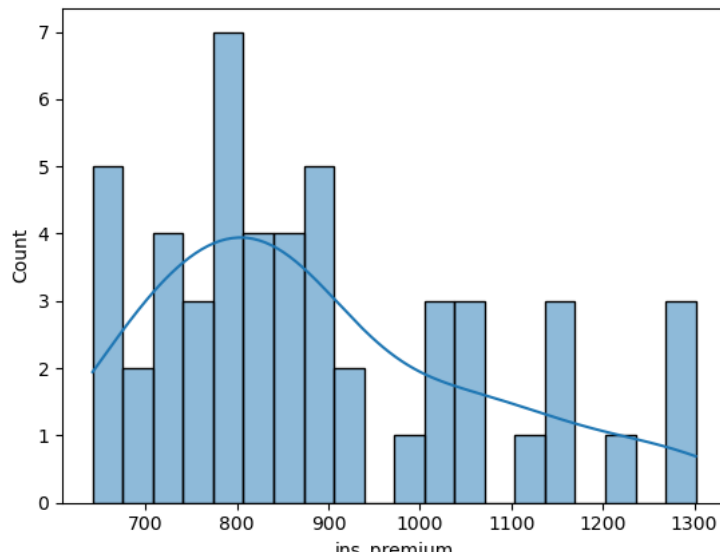
Text(0.5, 1.0, 'Distribution of Total Crashes')



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)
```

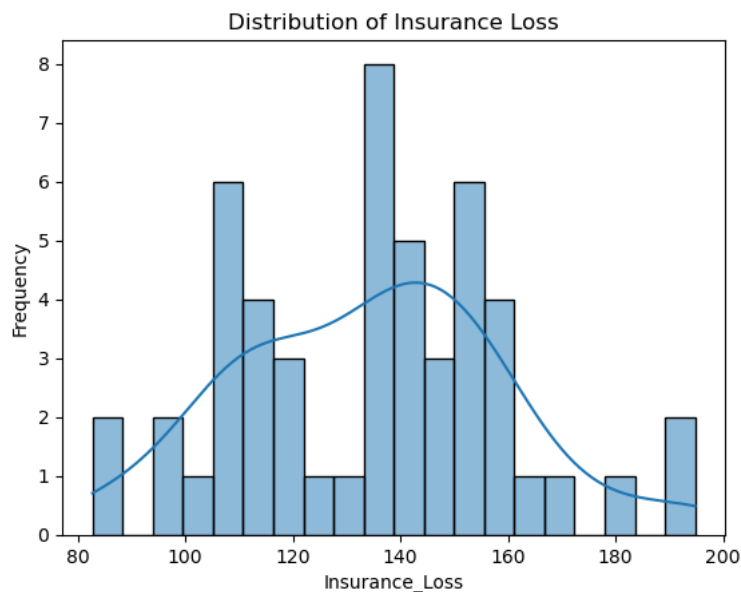
<Axes: xlabel='ins_premium', ylabel='Count'>



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')

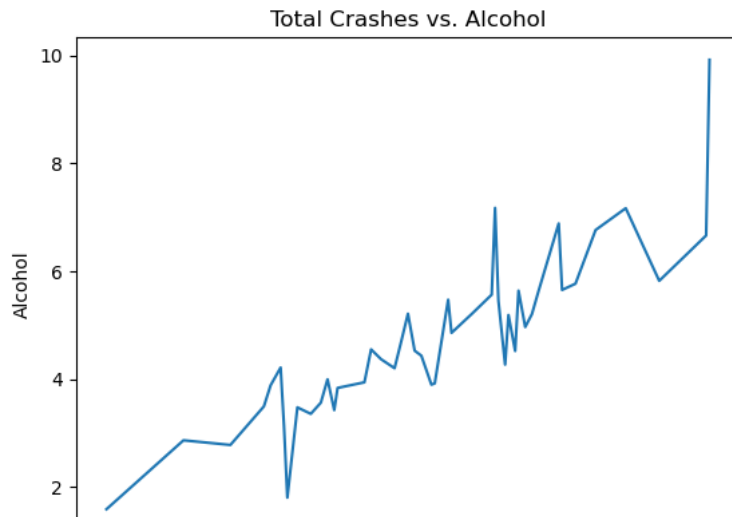


Inference : The histogram represents the distribution of insurance losses. The plot indicates that the majority of insurance losses fall within specific ranges, with peaks in frequency. The distribution appears right-skewed, indicating that a few instances have considerably higher insurance losses. This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

Lineplot

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

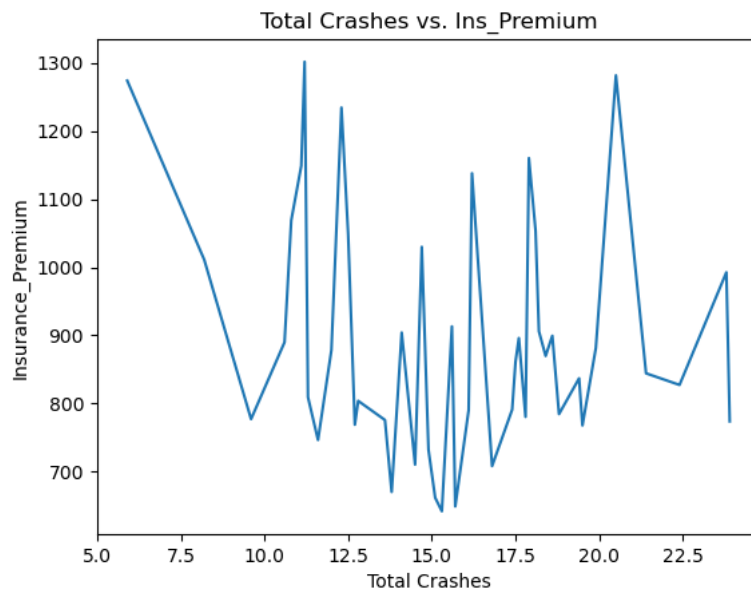
```
Text(0.5, 1.0, 'Total Crashes vs. Alcohol')
```



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. Ins_Premium')
```

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

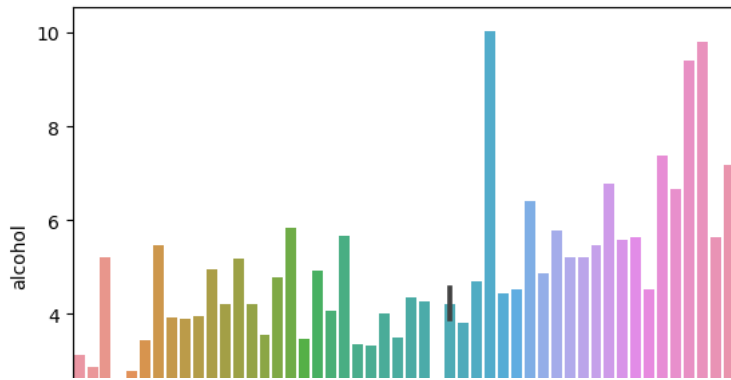


Inference : The line plot represents the relationship between total car crashes and insurance premiums. It visualizes how insurance premiums vary in relation to the total number of crashes. The plot does not show a clear linear trend; points on the line are scattered without a clear pattern. This suggests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

▼ Bivariate Distribution

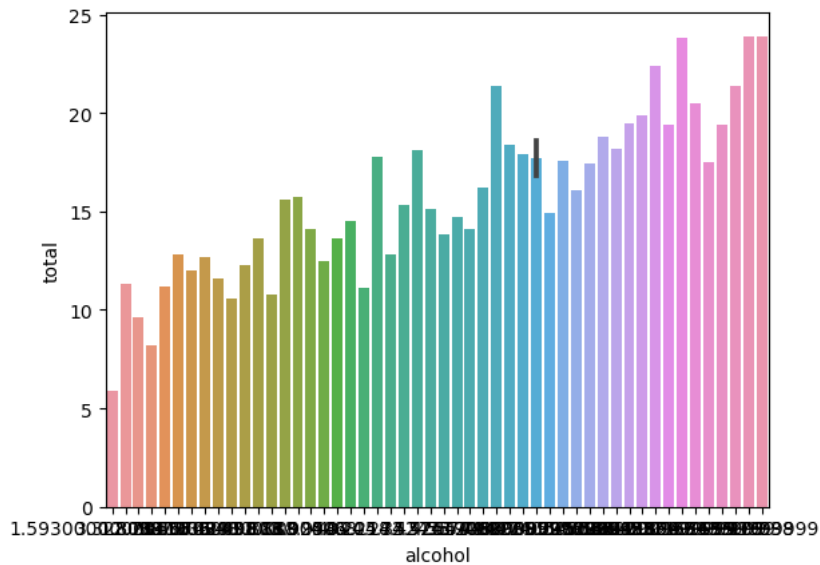
```
sns.barplot(x="speeding",y="alcohol",data=df)
```

```
<Axes: xlabel='speeding', ylabel='alcohol'>
```



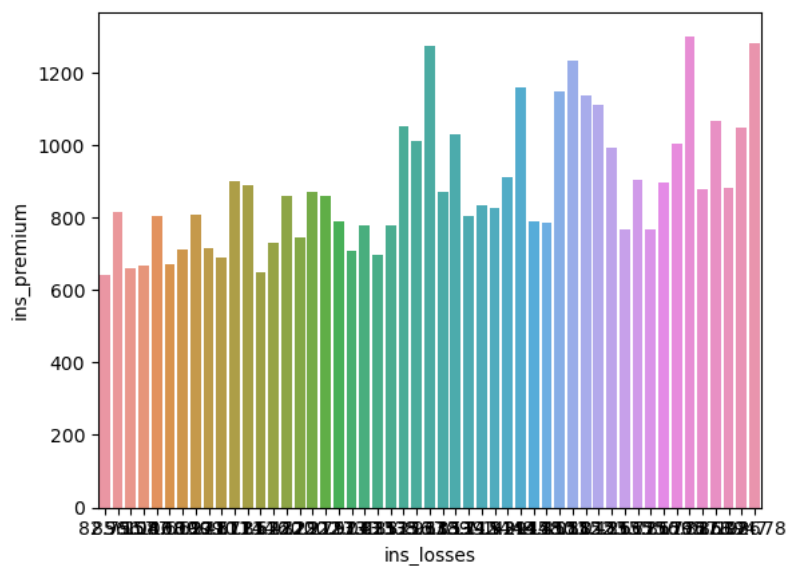
```
sns.barplot(x='alcohol',y='total',data=df)
```

```
<Axes: xlabel='alcohol', ylabel='total'>
```



```
sns.barplot(x='ins_losses',y='ins_premium',data=df)
```

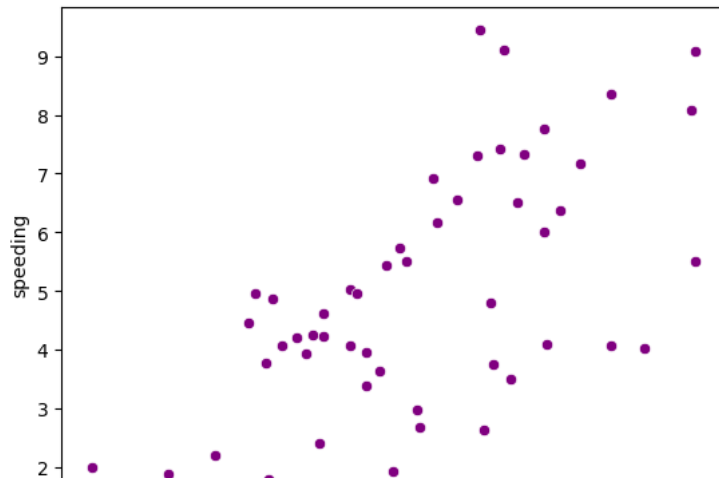
```
<Axes: xlabel='ins_losses', ylabel='ins_premium'>
```



▼ Scatter plot

```
sns.scatterplot(x="total",y="speeding",data=df,color="purple")
```

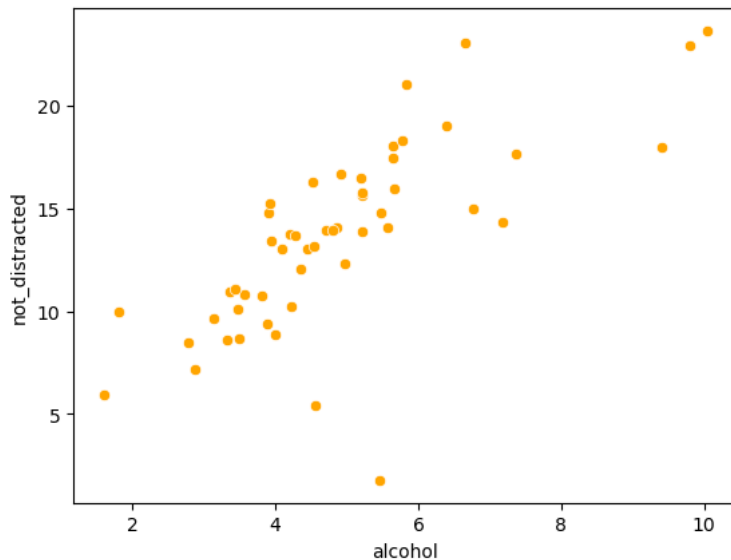
<Axes: xlabel='total', ylabel='speeding'>



Inference: from the plot we can say that as speeding increases total car_crashes accident increases and on an average accidents were occurring at position between 4 to 7

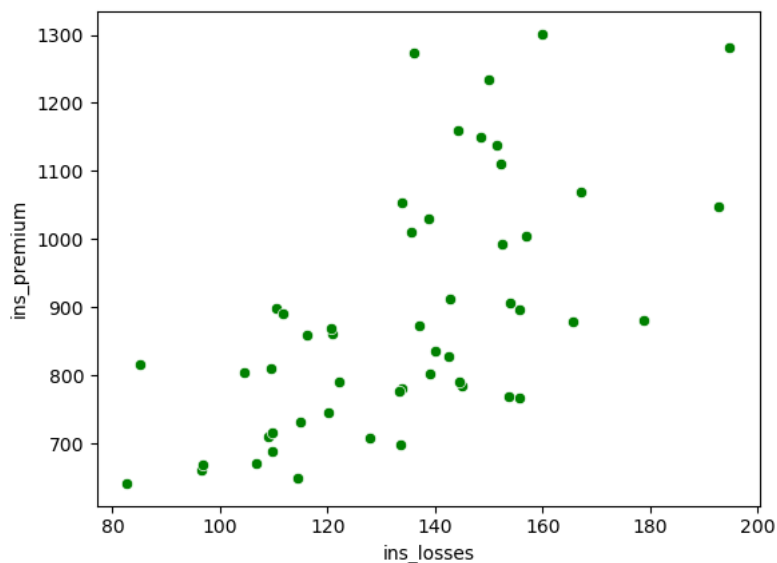
```
sns.scatterplot(x="alcohol", y="not_distracted", data=df, color="orange")
```

<Axes: xlabel='alcohol', ylabel='not_distracted'>



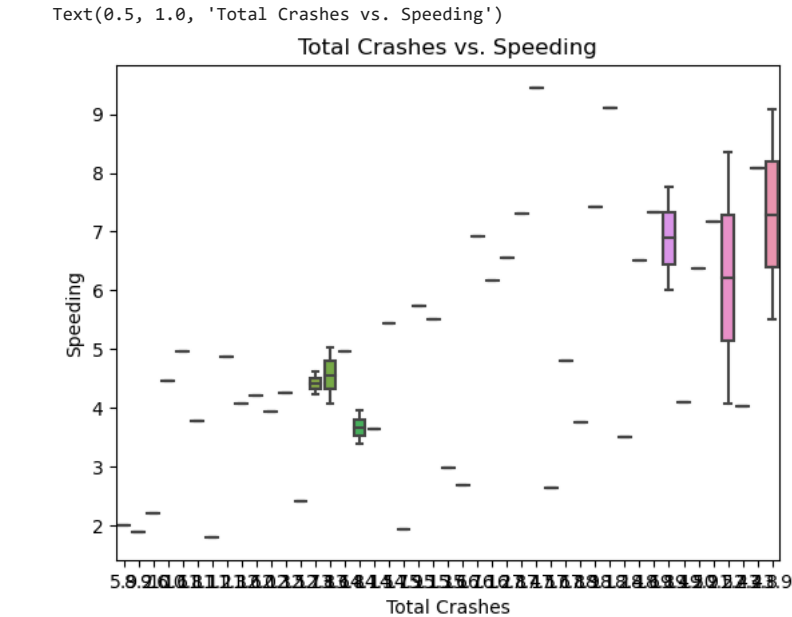
```
sns.scatterplot(x="ins_losses", y="ins_premium", data=df, color="green")
```

<Axes: xlabel='ins_losses', ylabel='ins_premium'>



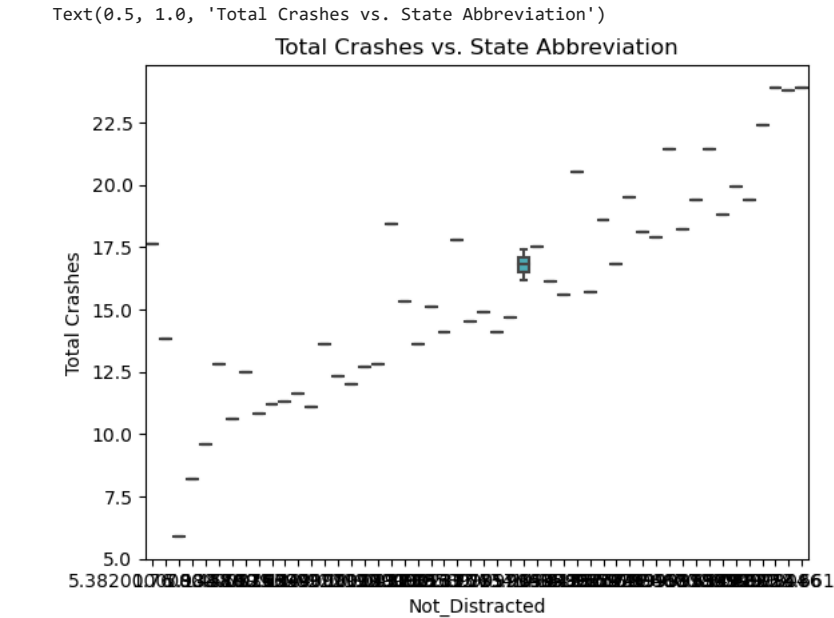
▼ Boxplot


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

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

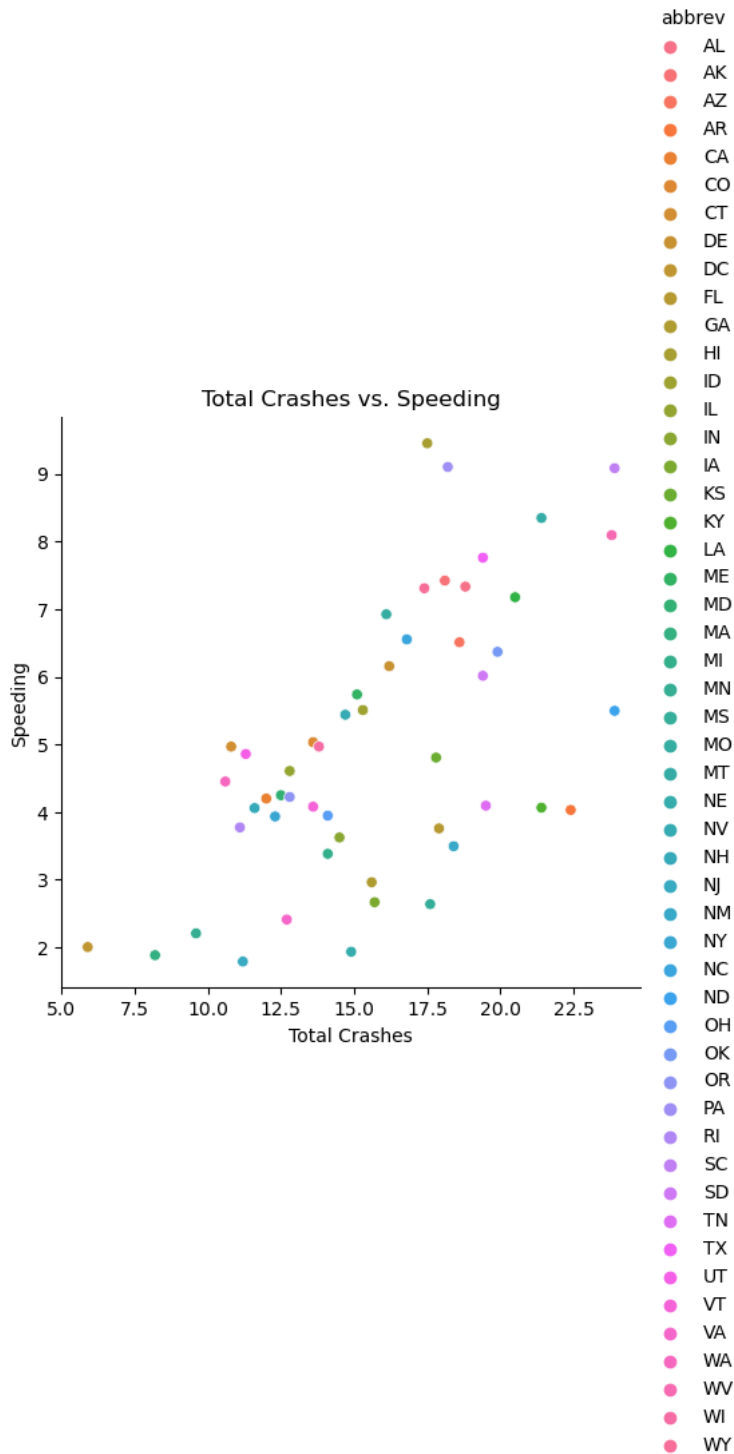


Inference : The box plot illustrates the distribution of total crashes concerning the distraction status of drivers (Not Distracted). It provides insights into how distraction affects the total number of car crashes. The plot shows varying total crash counts based on the distraction status, with potentially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

Replot

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

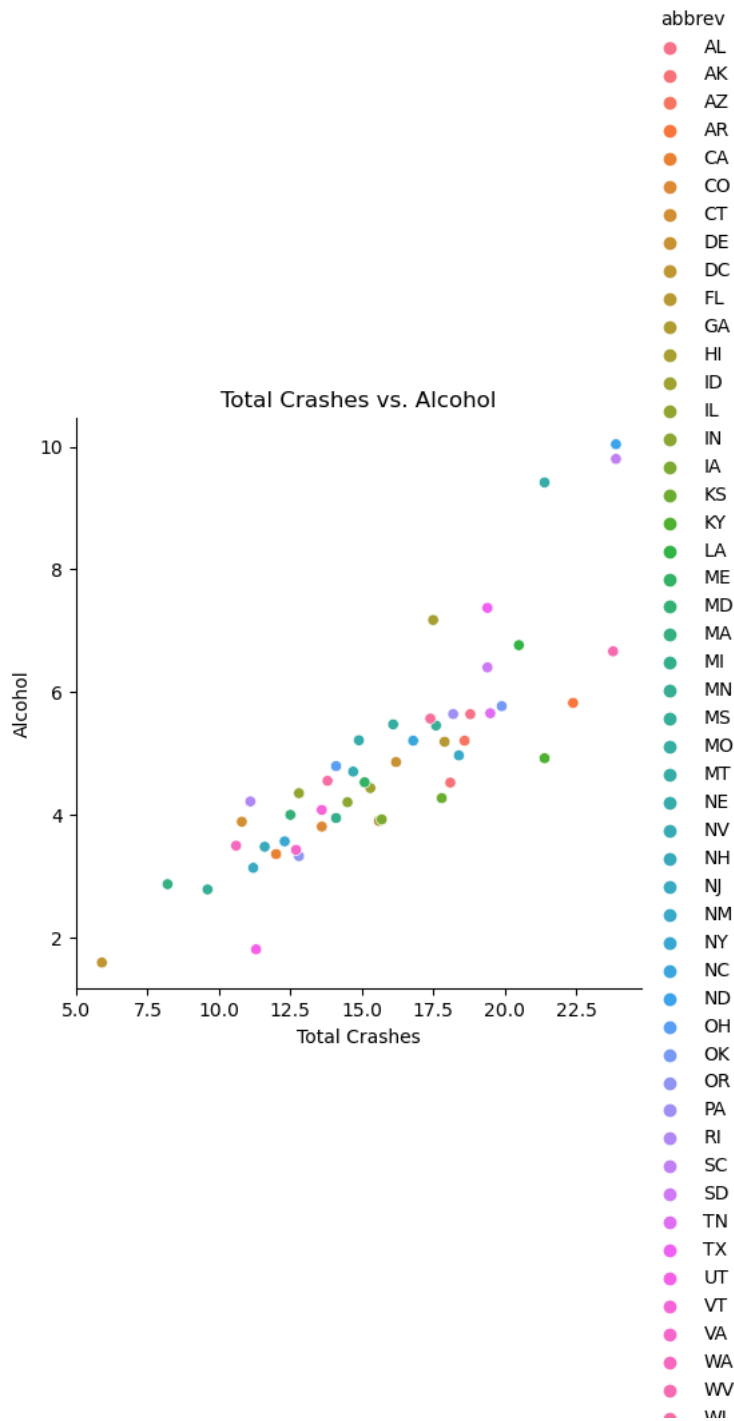
```
Text(0.5, 1.0, 'Total Crashes vs. Speeding')
```



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')
```

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

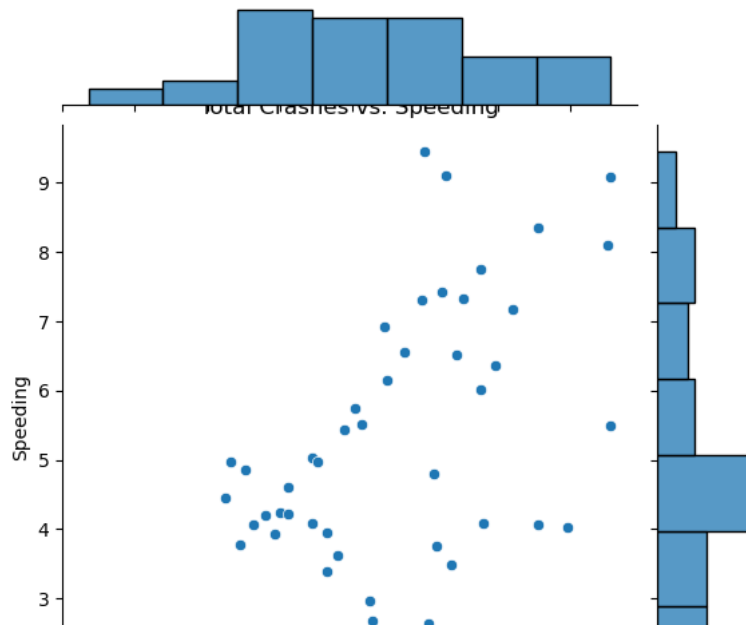


Inference : The relational plot ("relplot") illustrates the relationship between total car crashes and crashes involving alcohol. Each point on the plot represents a data point in the dataset, and different states are color-coded for comparison (hue). The plot provides a visual comparison of how alcohol-related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

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')
```

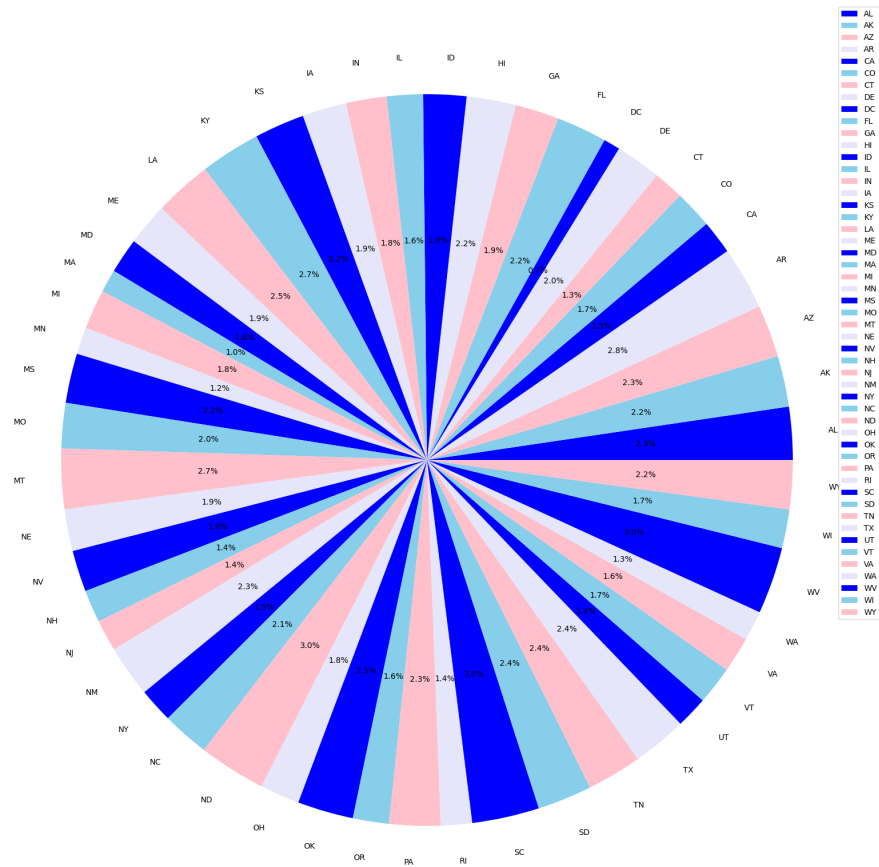


Inference : The joint plot displays the relationship between total car crashes and crashes involving speeding. It combines a scatter plot and histograms to visualize the distribution and correlation between the two variables. The scatter plot shows that there isn't a strong linear relationship between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

▼ 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 =['blue','skyblue','pink','lavender']) # %0.1f%% specifies percentage
axes1.legend()
```

<matplotlib.legend.Legend at 0x2925b2dd490>



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

▼ Finding correlation for every attribute

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

```
C:\Users\AKARSHA\AppData\Local\Temp\ipykernel_15048\3182140910.py:1: FutureWarning: The de
corr=df.corr()
```

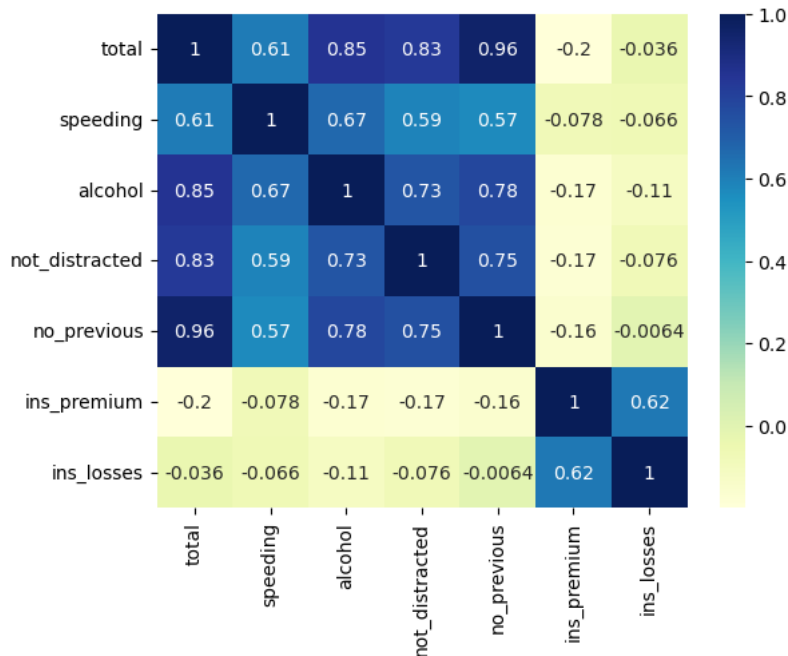
	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.199702	-0
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.077675	-0
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.170612	-0
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	-0.174856	-0
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	-0.156895	-0
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856	-0.156895	1.000000	0
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970	-0.006359	0.623116	1

Inference: It gives data from the region of -1 to 1 where greater than 0 can be considered as positively correlated and less than 0 are considered as negatively correlated. From above premium insurance and initial losses are independent variables so they were negatively correlated. Speeding and alcohol are highly positively correlated and not_distracted attribute is positively correlated.

Heatmap

```
sns.heatmap(p, annot=True, cmap="YlGnBu")
```

<Axes: >



Inference: In above heatmap blue indicates extreme values which are positively correlated and green represents negatively correlated. We can get car crashes more precisely like higher the speeding there is a chance of more likely to have accident. It also tells alcohol intake and car crashes are directly proportional. It also tells where values are drivers are not distracted but had car crash, it also tells insurance premium are not involved, similarly losses weren't involved in the similar way. In this extreme values can be seen in dark blue and minimal values are seen in light green.