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


df = pd.read_csv('car_crashes.csv')


# Check for missing values
print(df.isnull().sum())
```

```
total             0
speeding          0
alcohol           0
not_distracted    0
no_previous       0
ins_premium       0
ins_losses        0
abbrev            0
dtype: int64
```

There are no null values in the dataset

```
X = df[['speeding', 'alcohol', 'not_distracted', 'no_previous', 'ins_premium', 'ins_losses', 'abbrev']]
y = df['total']
```

Separate independent and dependent variables

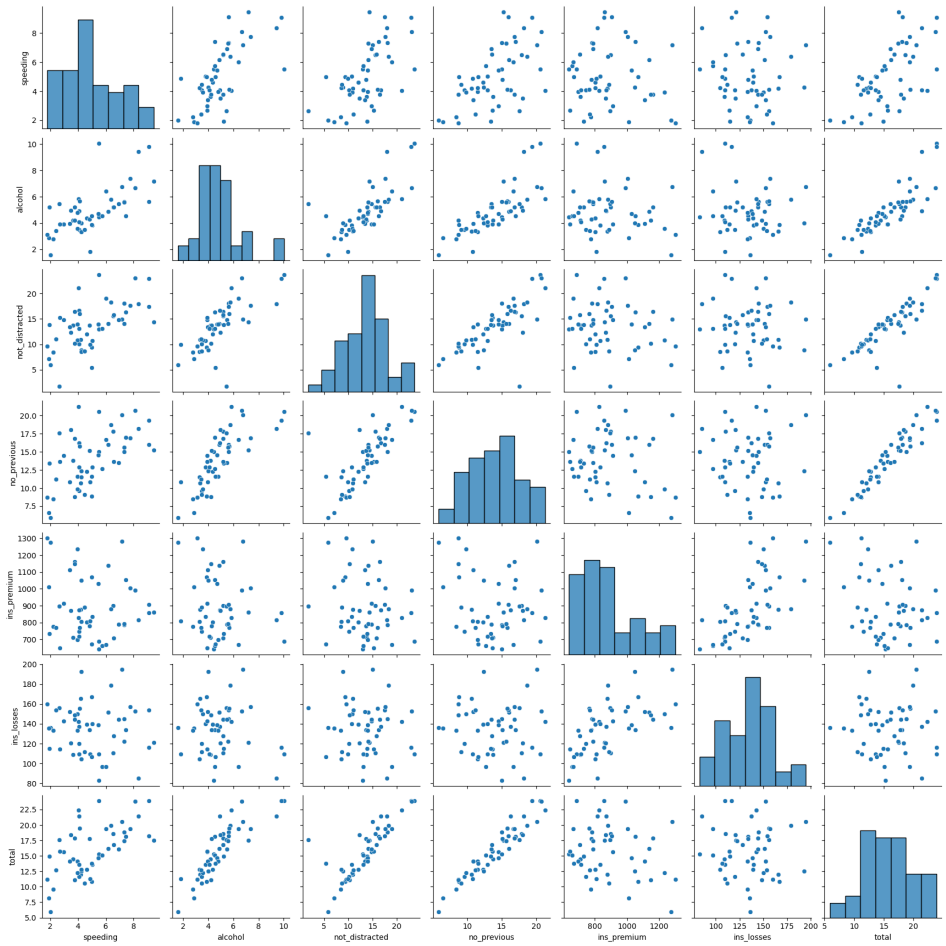
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
X_train = X_train.drop(columns=['abbrev'])
X_test = X_test.drop(columns=['abbrev'])
```

splitting into training and testing data

```
from sklearn.preprocessing import StandardScaler
# Standard Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Feature scaling

```
# Here's an example of a pair plot using Seaborn:
sns.pairplot(df, vars=['speeding', 'alcohol', 'not_distracted', 'no_previous', 'ins_premium', 'ins_losses', 'total'])
plt.show()
```



INFERENCE: The scatterplots in the upper diagonal of the grid show the relationships between pairs of continuous variables.

Correlation: The general trend and direction of the points can give you insights into the correlation between variables. If points slope upwards from left to right, it suggests a positive correlation; if they slope downwards, it suggests a negative correlation.

Strength of Association: The density of points and the tightness of the scatter can give you an idea of the strength of the relationship between variables. A dense cluster of points suggests a strong association, while a scattered plot suggests a weaker association.

Outliers: Outliers, if present, can be identified as points that are far from the main cluster. These outliers might represent data points that behave differently from the majority.

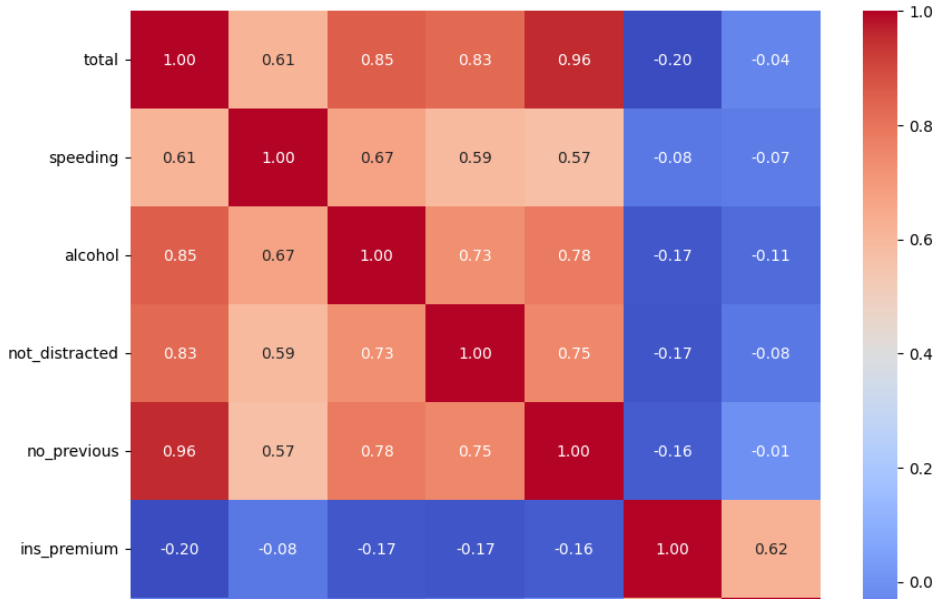
Histograms: The histograms on the diagonal of the grid show the distribution of each variable individually. You can infer the following from histograms:

1) **Data Distribution**: You can see how each variable is distributed, whether it follows a normal distribution, is skewed, or has multiple peaks.

2) **Range**: The range of values for each variable can be observed.

```
# Correlation Heatmap
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.show()
```

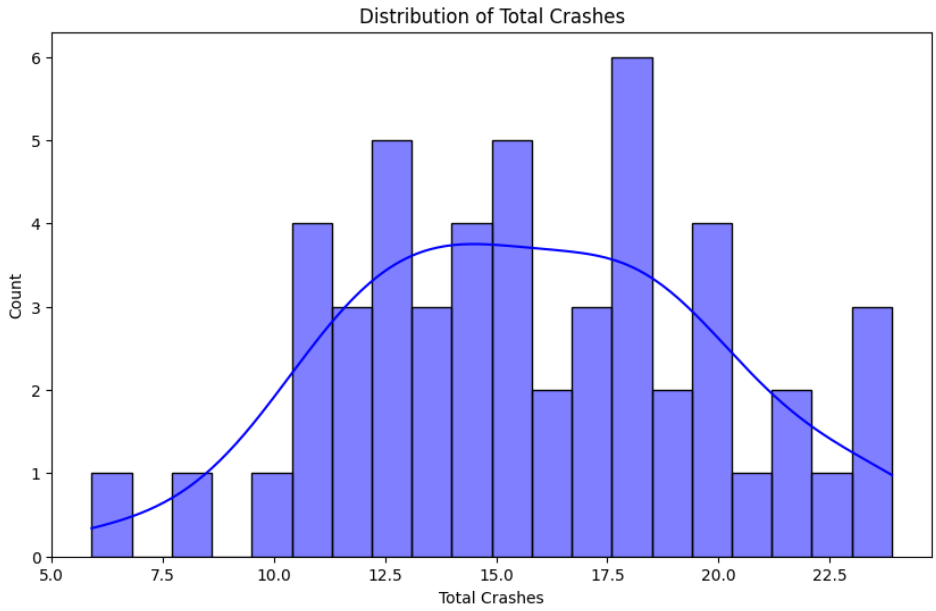
```
<ipython-input-8-55ed4af2f009>:2: FutureWarning: The default value of numeric_only in DataFr
  correlation_matrix = df.corr()
```



INFERENCE: The heatmap allows you to quickly identify which pairs of variables are strongly correlated, either positively or negatively. It provides insights into which variables may have redundant or similar information. In feature selection or dimensionality reduction, you might consider removing one of two highly correlated variables.

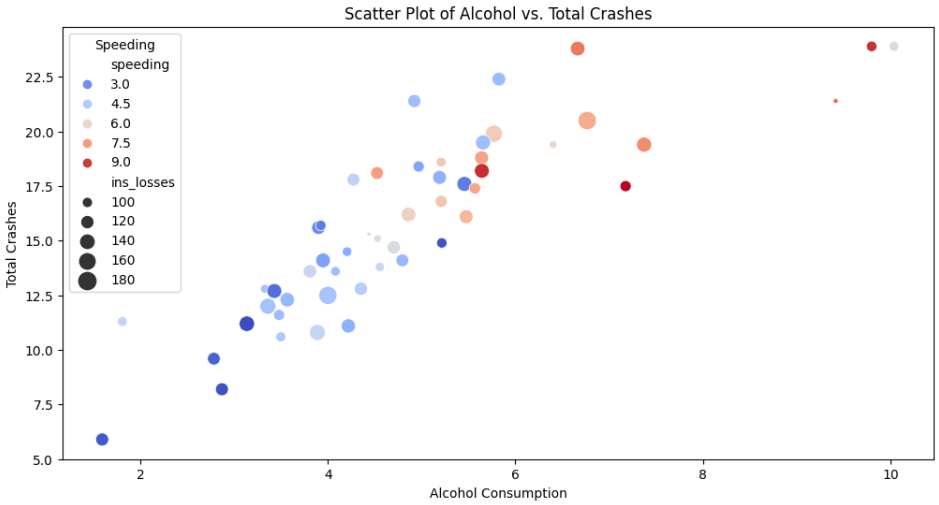
Here, we can see that the total accidents have a high positive correlation with no_previous accidents. Alcohol and total also has a high correlation (0.85) which implies that % of total accidents are directly related to % of alcohol-related incidents. There aren't any pairs of variables with considerable negative correlation, so there's no feature that is inversely related to another.

```
# Distribution Plots
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='total', bins=20, kde=True, color='blue')
plt.xlabel('Total Crashes')
plt.title('Distribution of Total Crashes')
plt.show()
```



INFERENCE: The histogram has multiple peaks, which suggests that the data may have multiple modes or subpopulations with distinct characteristics. The histogram can help us identify the central tendency of the data. For example, the peak of the histogram represents the mode, which is the most common value or range of values in the distribution. Here, ~18 is the value that reoccurs the most in total crashes. The width and spread of the histogram provide insights into the variability of the "total crashes" variable. A wider histogram suggests greater variability, while a narrower histogram indicates less variability.

```
# Scatter Plots
plt.figure(figsize=(12, 6))
sns.scatterplot(data=df, x='alcohol', y='total', hue='speeding', palette='coolwarm', size='ins_losses', sizes=(10, 200))
plt.xlabel('Alcohol Consumption')
plt.ylabel('Total Crashes')
plt.title('Scatter Plot of Alcohol vs. Total Crashes')
plt.legend(title='Speeding')
plt.show()
```



INFERENCE: The points form an upward-sloping pattern from left to right, it suggests a positive correlation. In this context, it means that as alcohol involvement increases, the total number of crashes tends to increase as well. The points cluster closely around a clear pattern, which suggests a strong correlation.

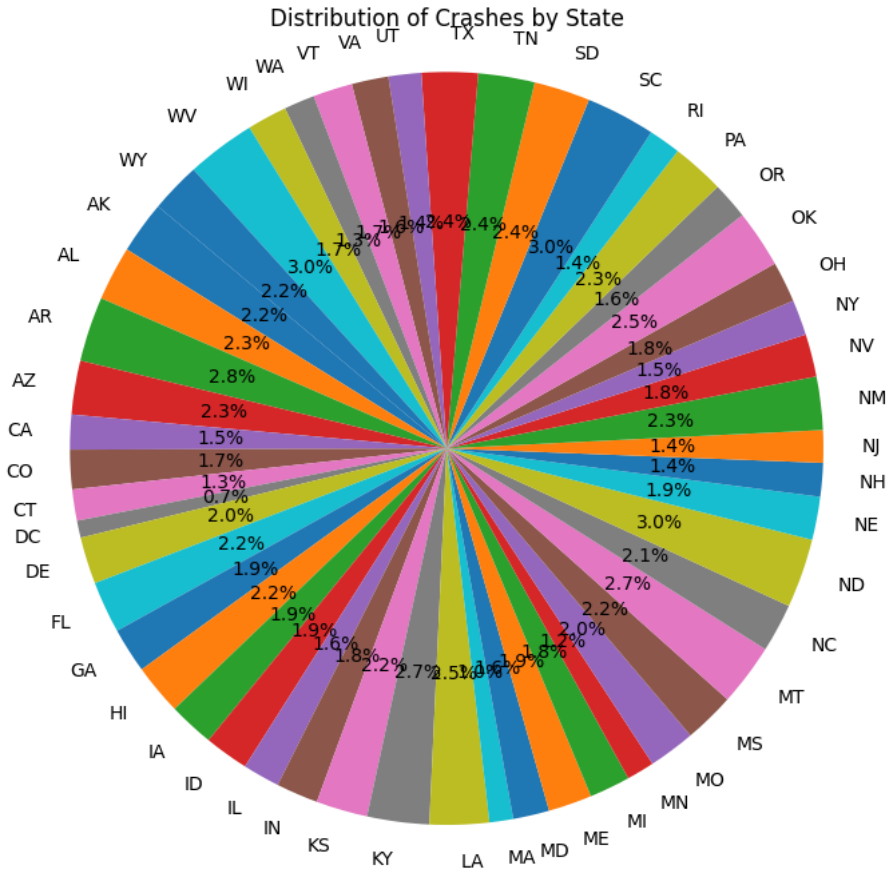
```
# Box Plots
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='abbrev', y='total', palette='Set2', showfliers=True)
plt.xlabel('State Abbreviation')
plt.ylabel('Total Crashes')
plt.title('Box Plot of Total Crashes by State')
plt.xticks(rotation=45)
plt.show()
```

INFERENCE: From the box plot, it is evident that the median crash count varies among states, with some states having a higher central tendency than others. Additionally, the interquartile range (IQR) indicates the spread of crash counts within each state, highlighting states with wider IQRs as having greater variability in crash counts. Overall, this box plot serves as a concise visual summary of the regional variation in total crashes, helping identify states with distinct patterns and potential areas for further investigation or intervention in road safety measures. We can see that states with the abbreviations ND, SC and WV have the highest amount of crashes, and DC has the lowest.

```
df1 = df.drop(columns=['abbrev'])
```

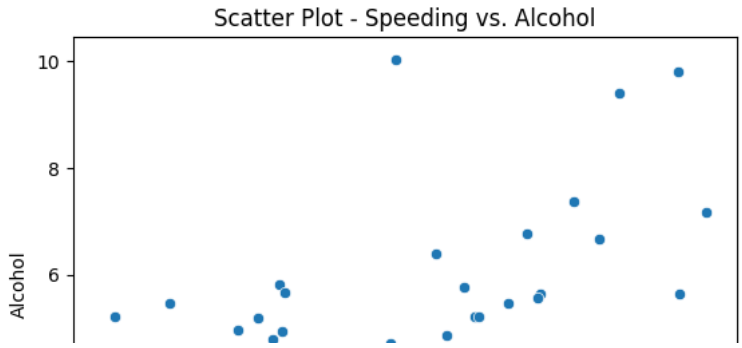
```
# Calculate the total crashes by state
crashes_by_state = df.groupby('abbrev')['total'].sum()

# Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(crashes_by_state, labels=crashes_by_state.index, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Distribution of Crashes by State')
plt.show()
```



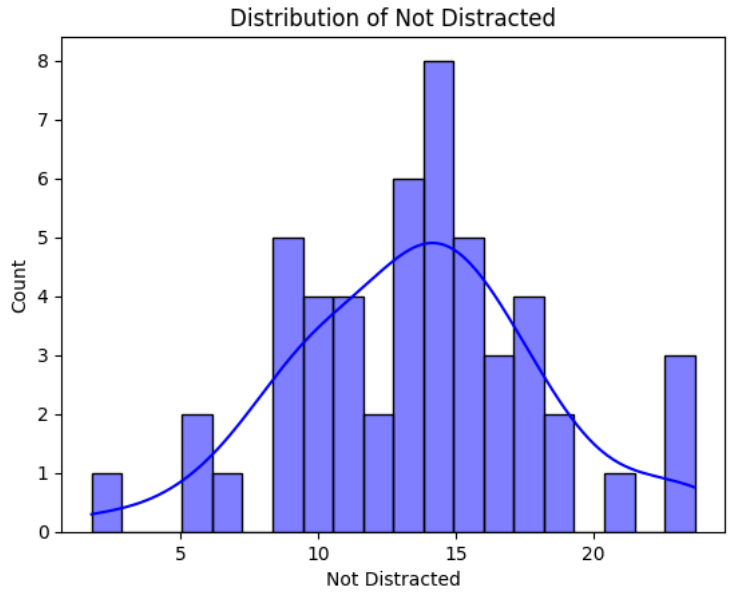
INFERENCE: The above pie chart shows the distribution of crashes statewise as a part of total accidents. As observed in the box-plot, ND, SC and WV have the highest percentage of crashes.

```
# Visualization 3: Scatter Plot - Speeding vs. Alcohol
sns.scatterplot(data=df, x='speeding', y='alcohol')
plt.xlabel('Speeding')
plt.ylabel('Alcohol')
plt.title('Scatter Plot - Speeding vs. Alcohol')
plt.show()
```



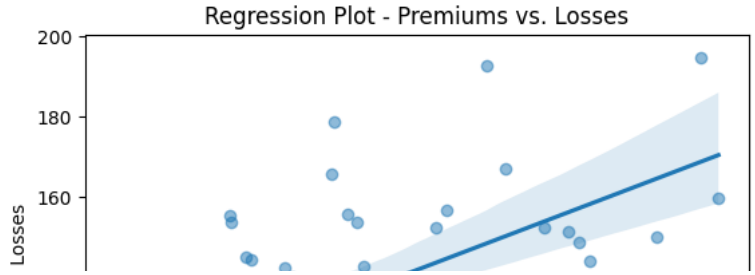
INFERENCE: A scatter plot of speeding vs. alcohol involvement reveals an almost direct relationship between these two variables. As the points cluster in an upward-sloping pattern from left to right, it suggests a positive correlation between speeding and alcohol involvement in car crashes. This correlation raises concerns, as it implies that higher instances of speeding are associated with a greater likelihood of alcohol involvement in accidents. Such insights from this scatter plot underscore the importance of addressing both speeding and alcohol-related factors in road safety initiatives to reduce the risk of accidents.

```
sns.histplot(data=df, x='not_distracted', bins=20, kde=True, color='blue')
plt.xlabel('Not Distracted')
plt.title('Distribution of Not Distracted')
plt.show()
```



The histogram distribution of "not_distracted" indicates a notable trend in driver behavior. The majority of drivers appear to be concentrated in the lower range of "not_distracted" values, suggesting that a significant proportion of drivers in the dataset are typically not distracted while driving. However, there is a tail towards higher "not_distracted" values, indicating that some drivers report being less distracted than the majority. This histogram provides valuable insights into the distribution of driver attentiveness and serves as a basis for understanding the prevalence of distracted driving within the dataset.

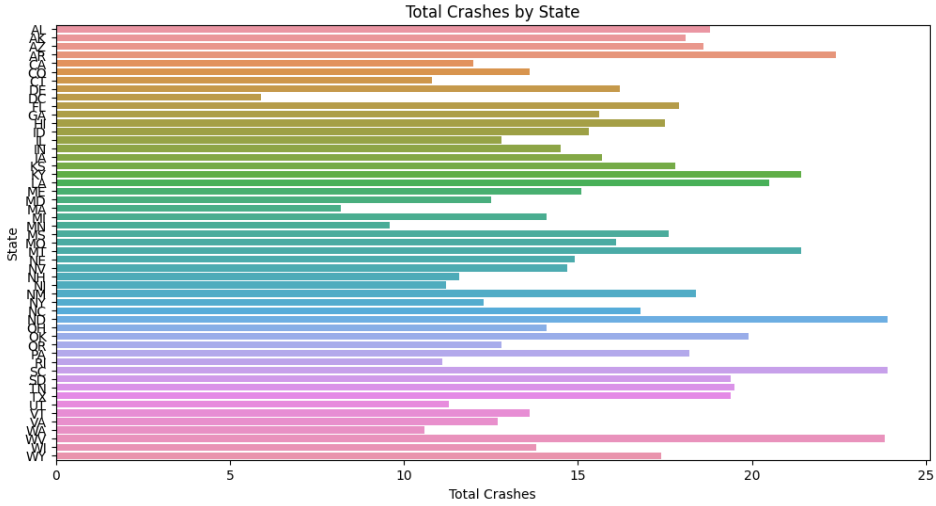
```
# Visualization 7: Regression Plot - Premiums vs. Losses
sns.regplot(data=df, x='ins_premium', y='ins_losses', scatter_kws={"alpha":0.5})
plt.xlabel('Insurance Premiums')
plt.ylabel('Insurance Losses')
plt.title('Regression Plot - Premiums vs. Losses')
plt.show()
```



INFERENCE: The plot displays a positively sloped line that suggests a positive correlation between insurance premiums and insurance losses. As premiums increase, the plot indicates a tendency for higher insurance losses, implying that insurance companies may charge higher premiums to compensate for increased claims. This observation underscores the economic dynamics involved in the insurance industry and highlights the importance of accurately assessing risks and premiums to maintain profitability while adequately covering policyholders' losses.

100 +

```
# Bar Graph - Total Crashes by State
plt.figure(figsize=(12, 6))
sns.barplot(data=df, x='total', y='abbrev')
plt.xlabel('Total Crashes')
plt.ylabel('State')
plt.title('Total Crashes by State')
plt.show()
```



INFERENCE: This is another visualization method that tells us about the total number of crashes, state-wise

✓ 0s completed at 11:33 AM