In [ ]:

```
# Step 1: Import necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:

```
# Step 2: Load the car crashes dataset from Seaborn
car_crashes = sns.load_dataset("car_crashes")
car_crashes.head()
```

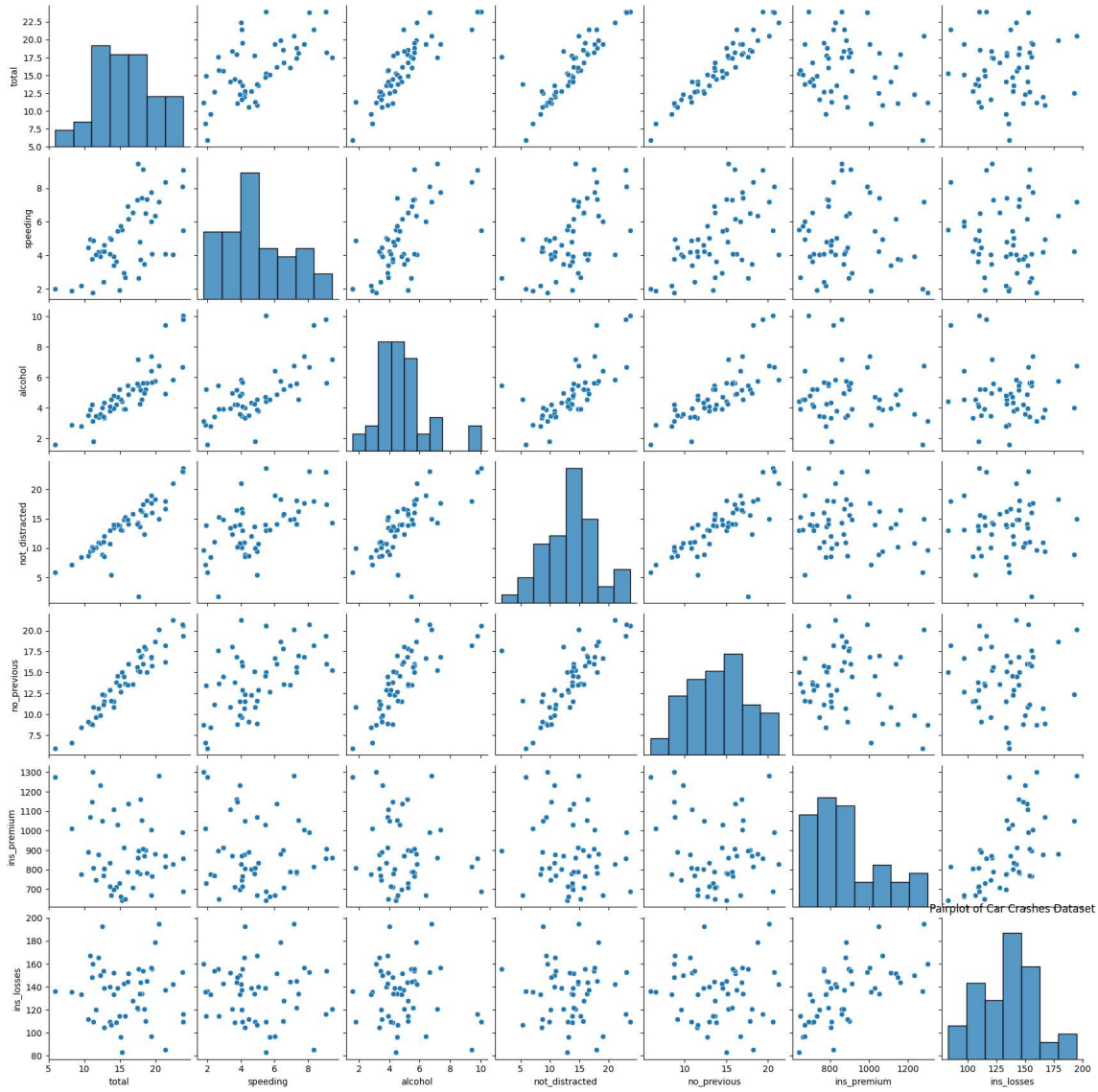
Out[16]:

	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

In [ ]:

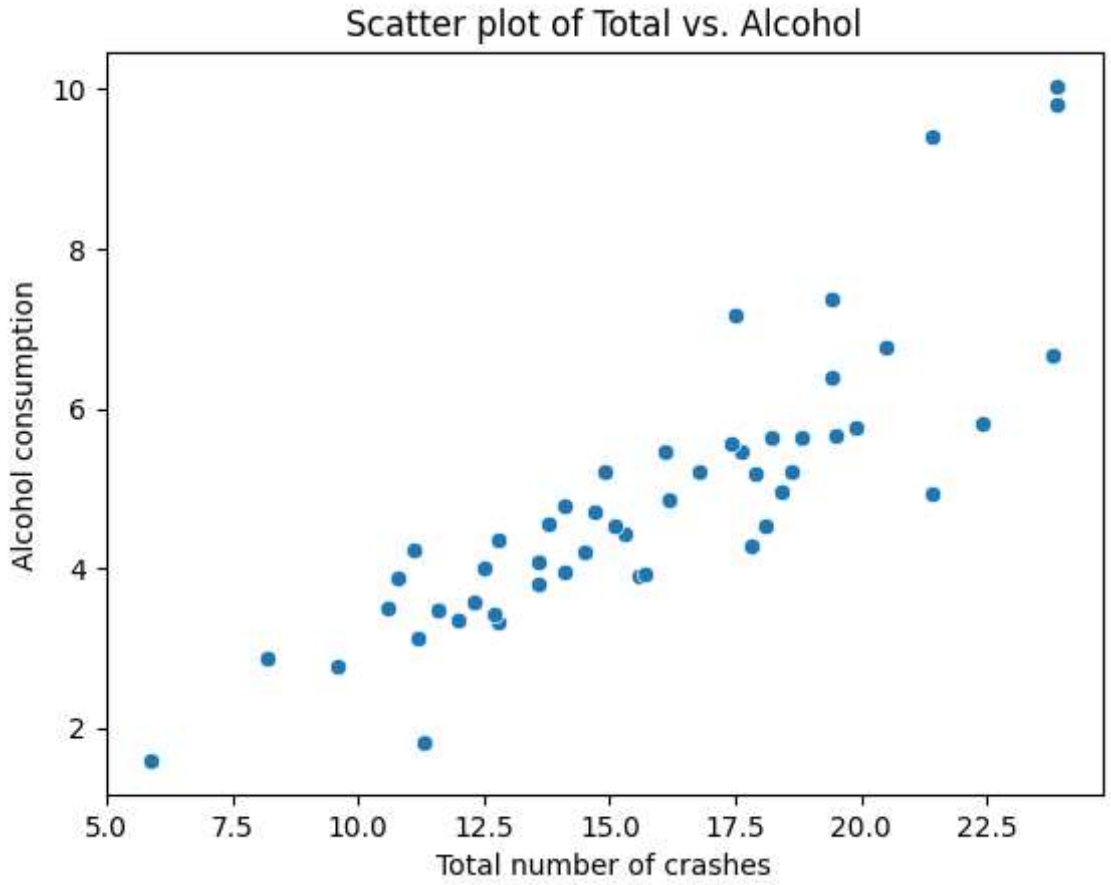
```
# Step 3: Data Visualization

# Visualization 1: Pairplot
sns.pairplot(car_crashes)
plt.title("Pairplot of Car Crashes Dataset")
plt.show()
# Inference: The pairplot displays pairwise relationships between numerical columns, whi
```



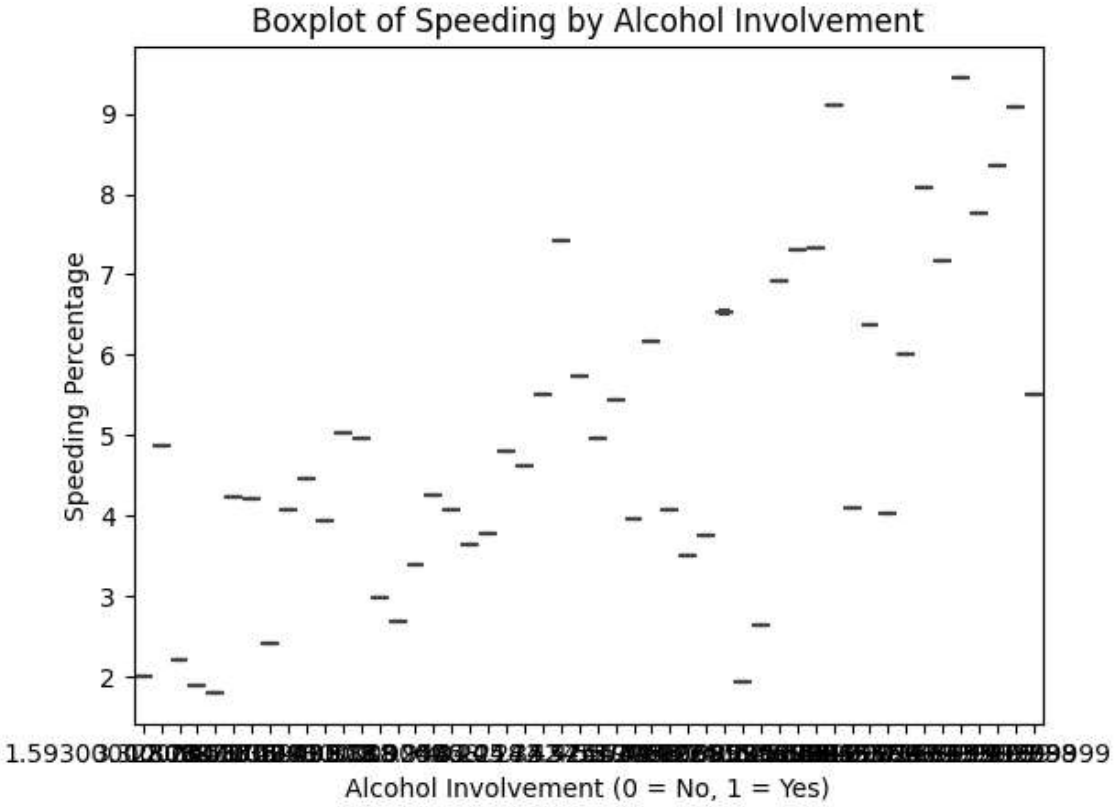
In [ ]:

```
# Visualization 2: Scatter plot of Total vs. Alcohol
sns.scatterplot(x="total", y="alcohol", data=car_crashes)
plt.title("Scatter plot of Total vs. Alcohol")
plt.xlabel("Total number of crashes")
plt.ylabel("Alcohol consumption")
plt.show()
# Inference: The scatter plot shows a positive correlation between the total number of c
```



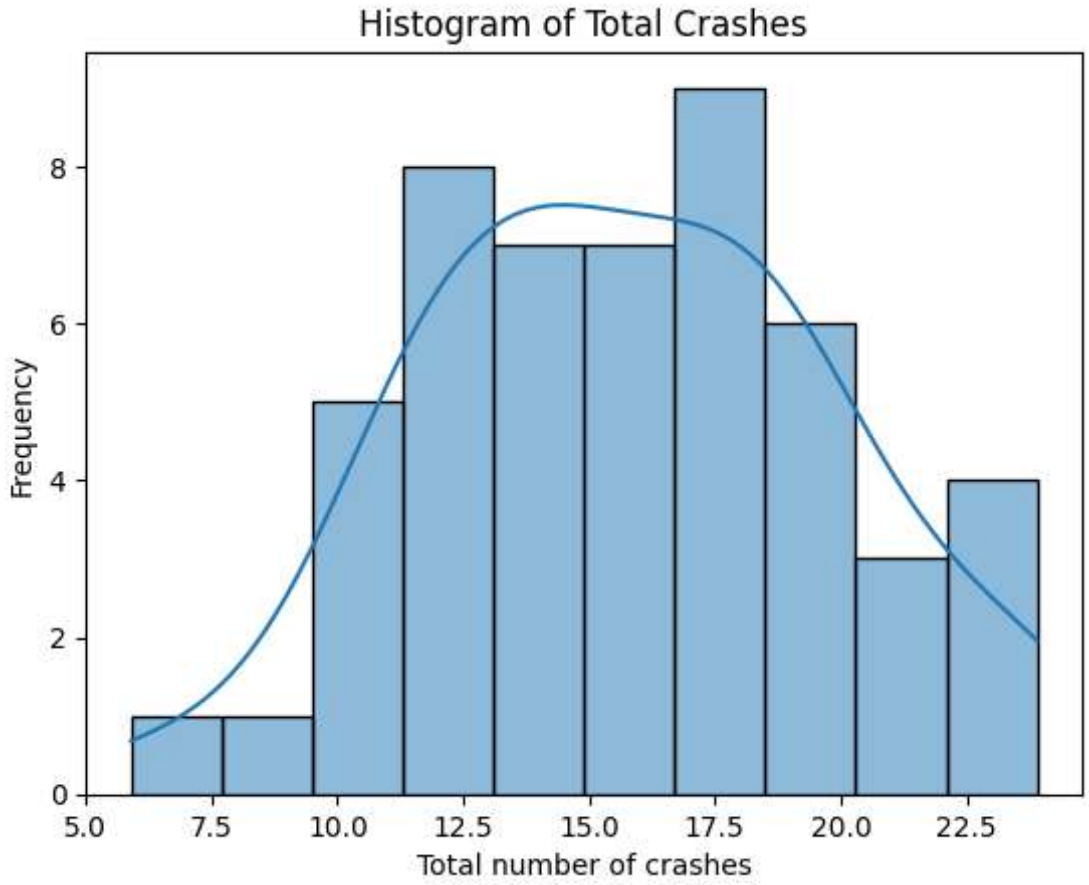
In [ ]:

```
# Visualization 3: Boxplot of Speeding by Alcohol Involvement
sns.boxplot(x="alcohol", y="speeding", data=car_crashes)
plt.title("Boxplot of Speeding by Alcohol Involvement")
plt.xlabel("Alcohol Involvement (0 = No, 1 = Yes)")
plt.ylabel("Speeding Percentage")
plt.show()
# Inference: The boxplot shows the distribution of speeding percentages for cases with a
```



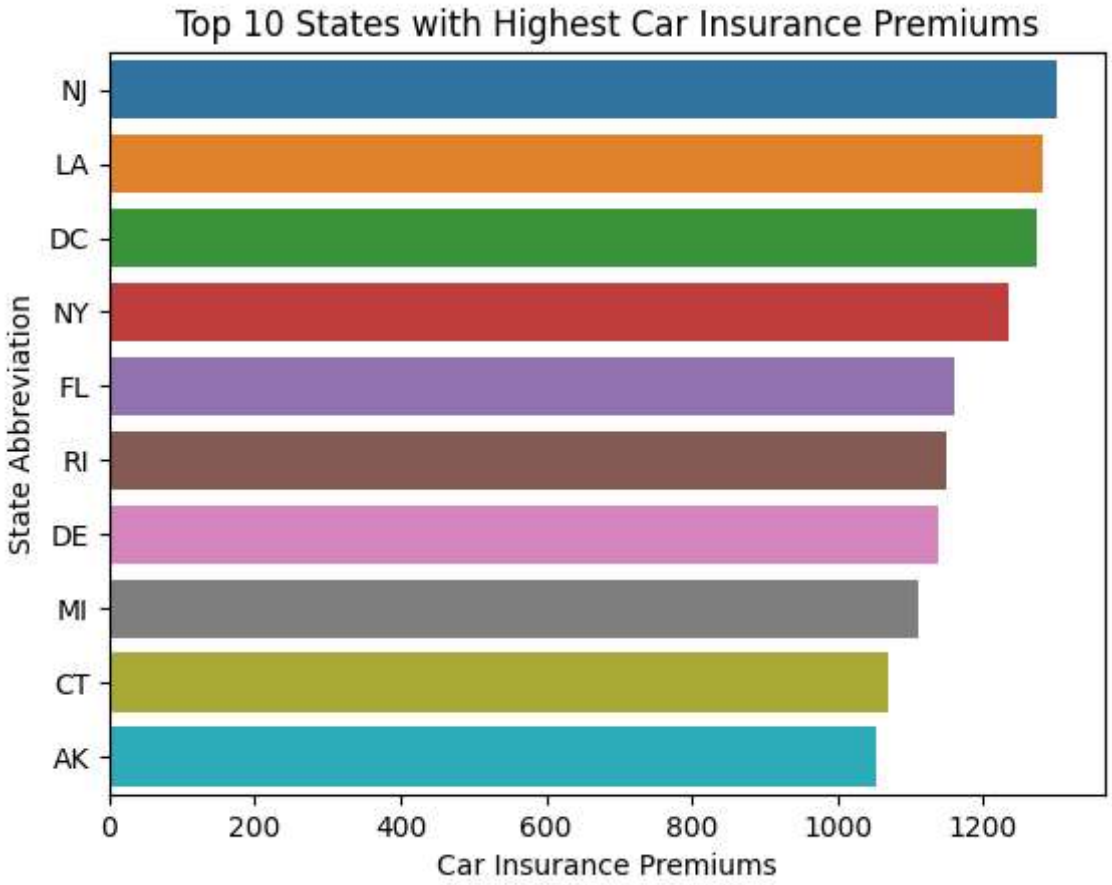
In [ ]:

```
# Visualization 4: Histogram of Total Crashes
sns.histplot(car_crashes["total"], bins=10, kde=True)
plt.title("Histogram of Total Crashes")
plt.xlabel("Total number of crashes")
plt.ylabel("Frequency")
plt.show()
# Inference: The histogram displays the distribution of the total number of crashes. It
```



In [ ]:

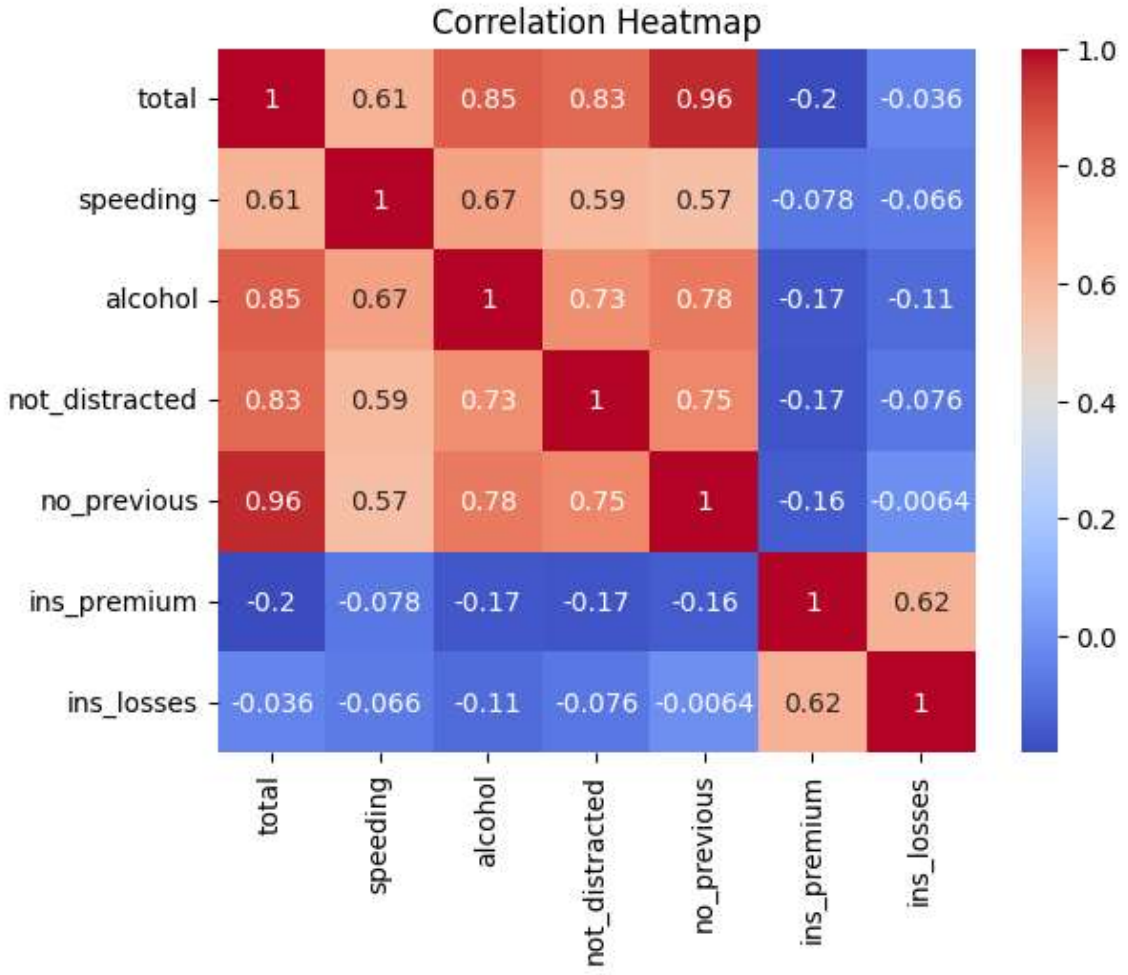
```
# Visualization 5: Bar plot of Car Insurance Premiums
sns.barplot(x="ins_premium", y="abbrev", data=car_crashes.sort_values("ins_premium", asc
plt.title("Top 10 States with Highest Car Insurance Premiums")
plt.xlabel("Car Insurance Premiums")
plt.ylabel("State Abbreviation")
plt.show()
# Inference: The bar plot highlights the top 10 states with the highest car insurance pr
```



In [ ]:

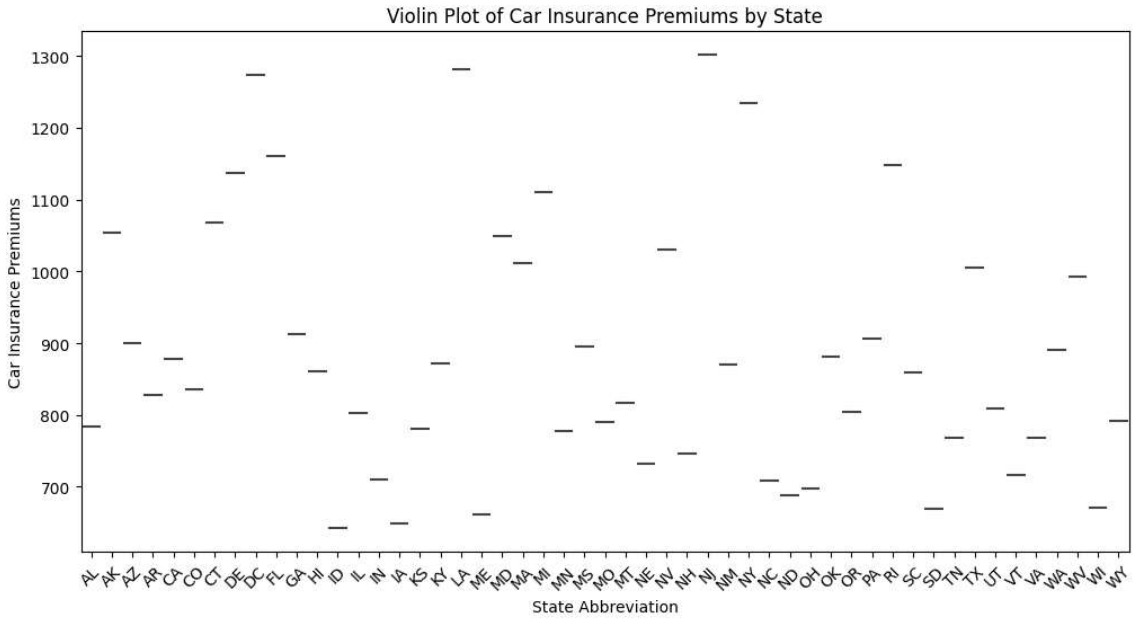
```
# Visualization 6: Heatmap of Correlation Matrix
correlation_matrix = car_crashes.corr()
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
# Inference: The heatmap displays the correlation matrix of numerical variables in the d
```

```
<ipython-input-9-1c4698ae88ff>:2: FutureWarning: The default value of nume
ric_only in DataFrame.corr is deprecated. In a future version, it will def
ault to False. Select only valid columns or specify the value of numeric_o
nly to silence this warning.
  correlation_matrix = car_crashes.corr()
```



In [ ]:

```
# Visualization 7: Violin Plot of Car Insurance Premiums by State
plt.figure(figsize=(12, 6))
sns.violinplot(x="abbrev", y="ins_premium", data=car_crashes)
plt.title("Violin Plot of Car Insurance Premiums by State")
plt.xlabel("State Abbreviation")
plt.ylabel("Car Insurance Premiums")
plt.xticks(rotation=45)
plt.show()
# Inference: The violin plot displays the distribution of car insurance premiums for dif
```



In [ ]:

```
# Visualization 8: Regression Plot of Speeding vs. Alcohol
sns.regplot(x="speeding", y="alcohol", data=car_crashes)
plt.title("Regression Plot of Speeding vs. Alcohol")
plt.xlabel("Speeding Percentage")
plt.ylabel("Alcohol Consumption")
plt.show()
# Inference: The regression plot shows the relationship between speeding percentage and
```