Importing seaborn

```
import seaborn as sns
import matplotlib.pyplot as plt
```

Load the dataset of car crashes

```
sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

```
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.	
14	14.5	3.625	4.205	13.775	13.775	710.46	108.	
15	15.7	2.669	3.925	15.229	13.659	649.06	114.	
16	17.8	4.806	4.272	13.706	15.130	780.45	133.	
17	21.4	4.066	4.922	16.692	16.264	872.51	137.	
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
```

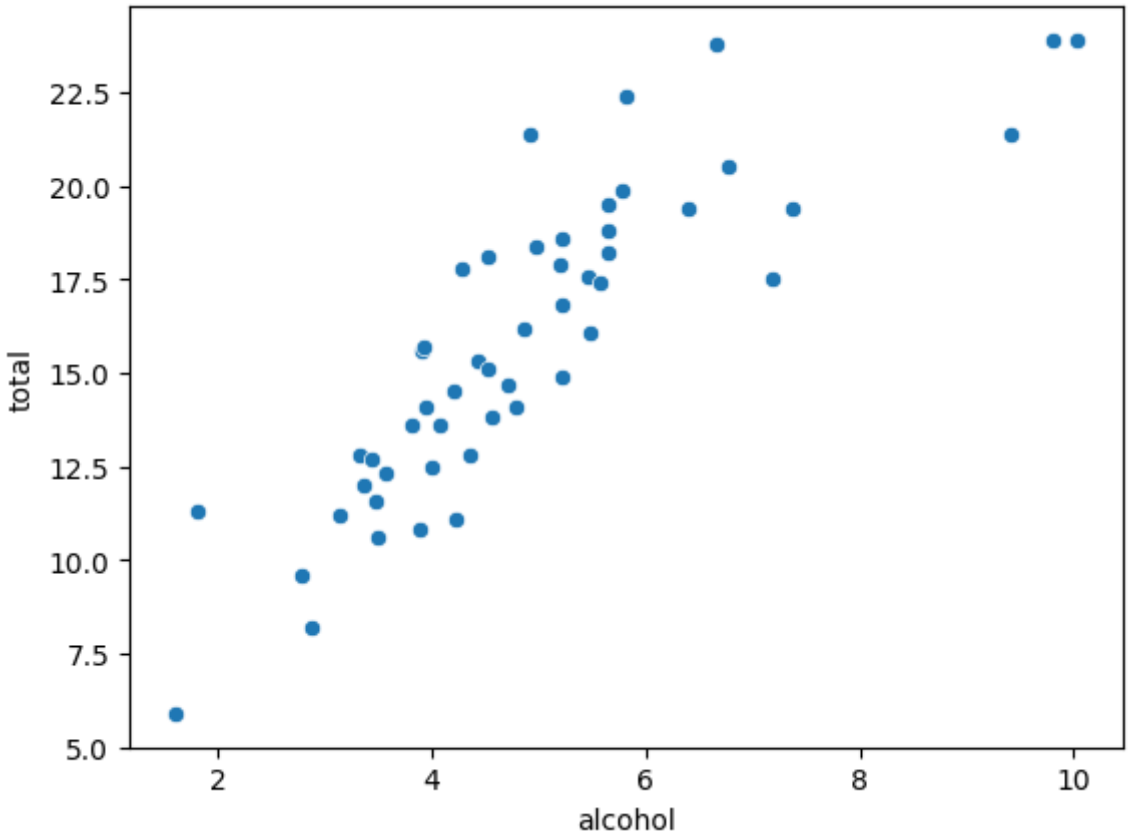
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85

```
df.head()
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	ab
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	

```
sns.scatterplot(x="alcohol",y="total",data=df)
```

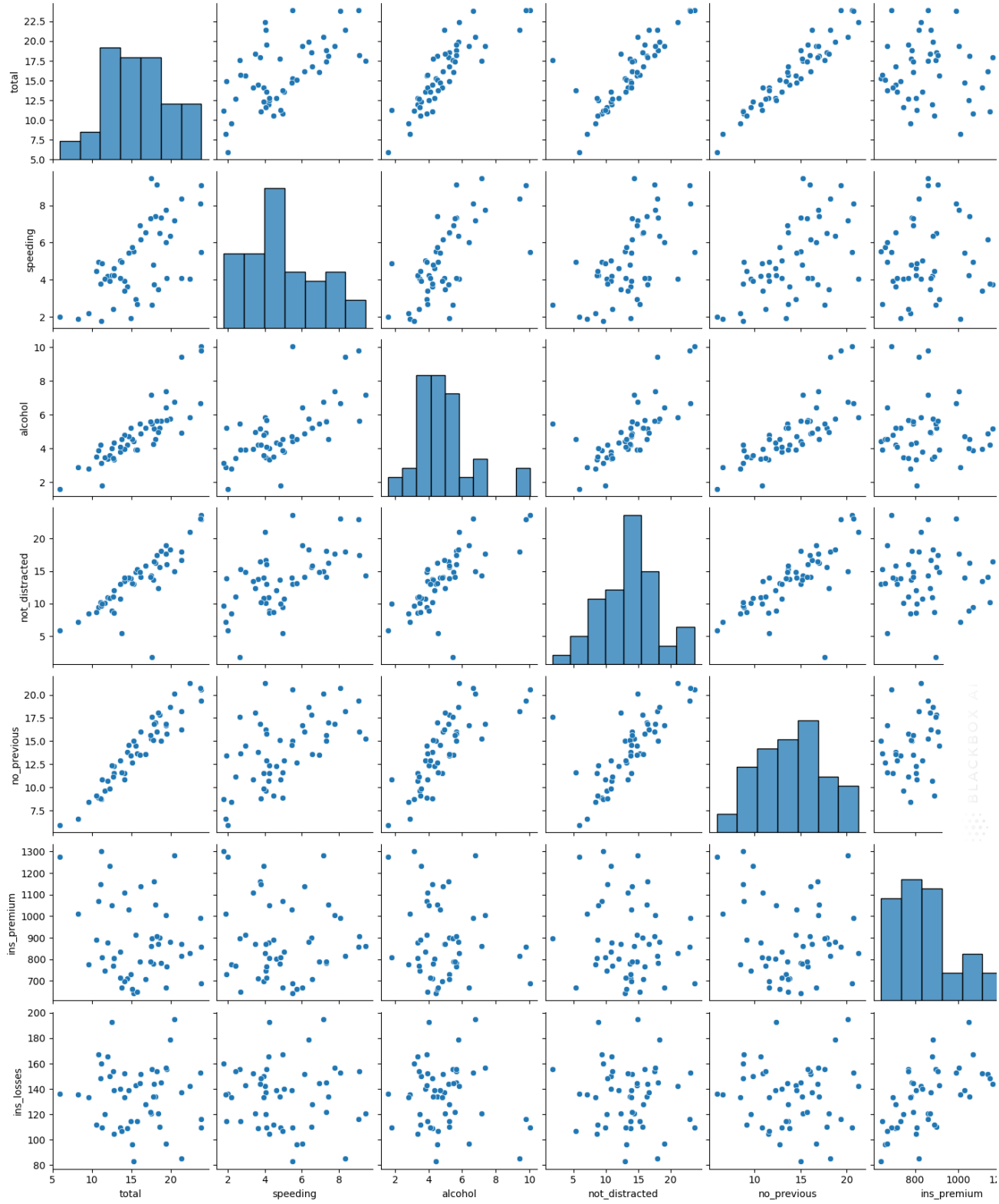
```
<Axes: xlabel='alcohol', ylabel='total'>
```



Inference: From the plot we can say that as the alcohol related crashes increases then total number of car crashes also increases

```
sns.pairplot(df)
```

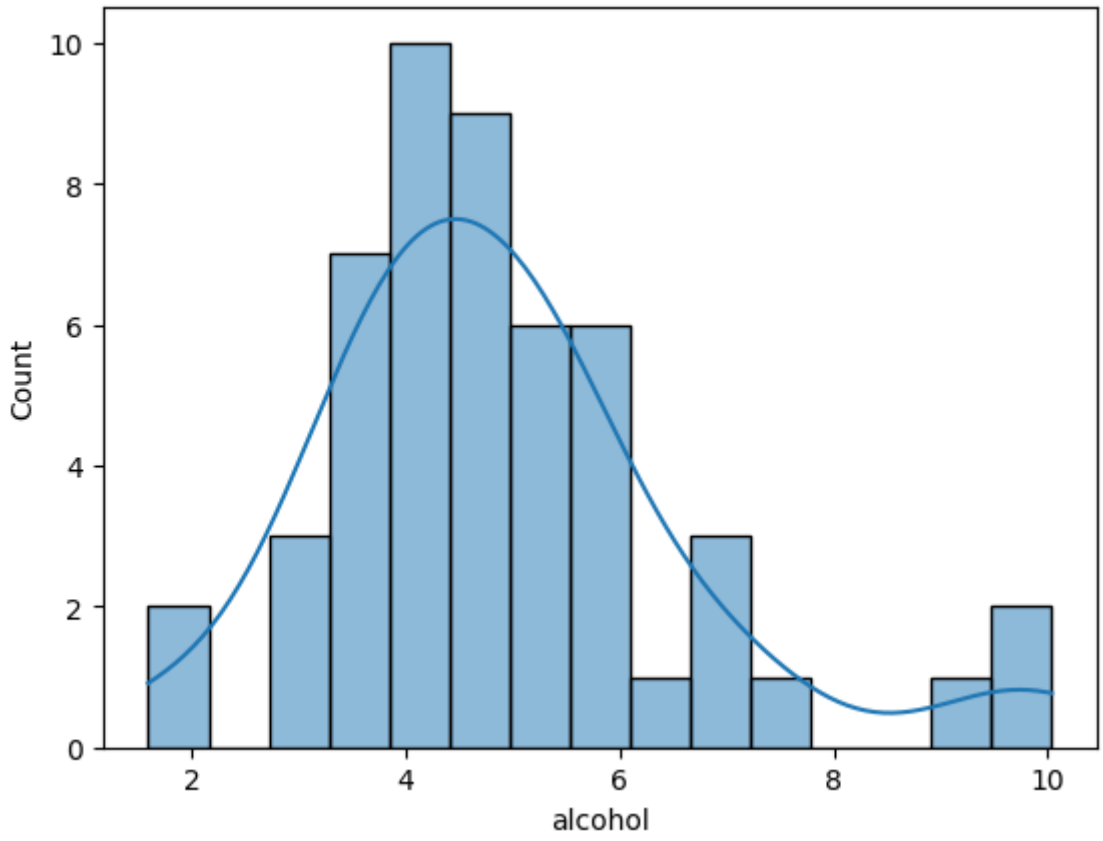
```
<seaborn.axisgrid.PairGrid at 0x7cf9e1a93e80>
```



Inference: To show the pairwise relationship in the given dataframe. It creates a matrix of scatterplots, where each variable in the dataset is plotted against every other variable.

```
sns.histplot(df["alcohol"], bins=15, kde=True)
```

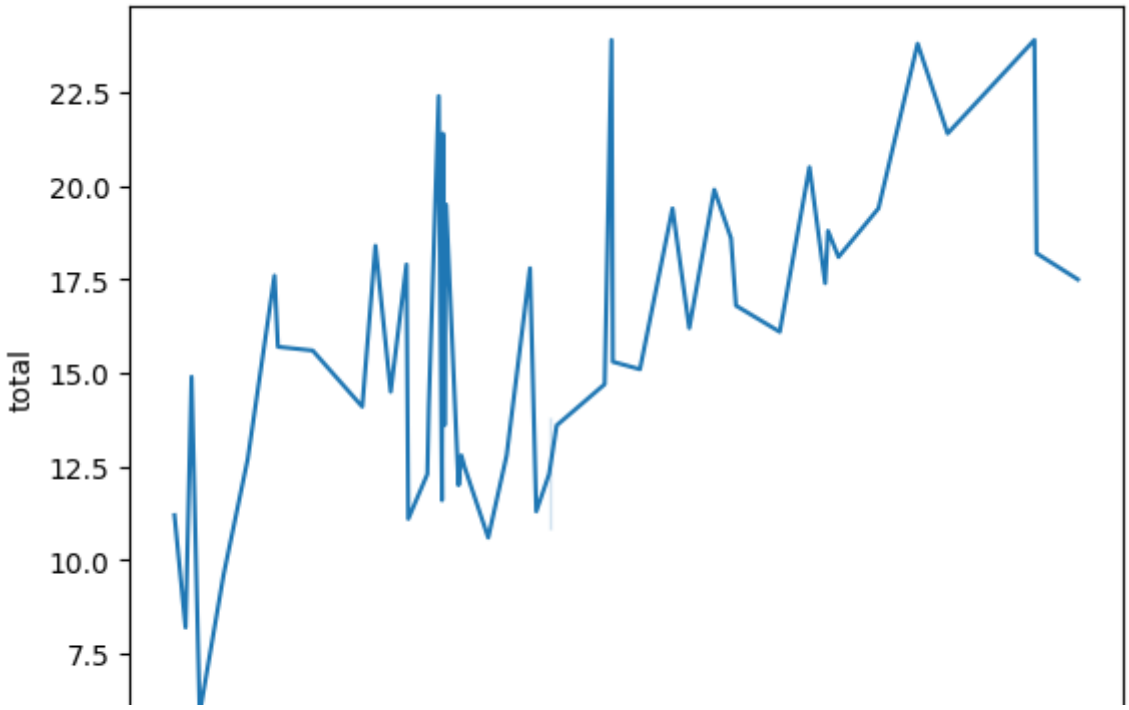
```
<Axes: xlabel='alcohol', ylabel='Count'>
```



Inference: It shows that most regions have a relatively low number of alcohol-related crashes, with a peak around 5-10 crashes.

```
sns.lineplot(x="speeding", y="total", data=df)
```

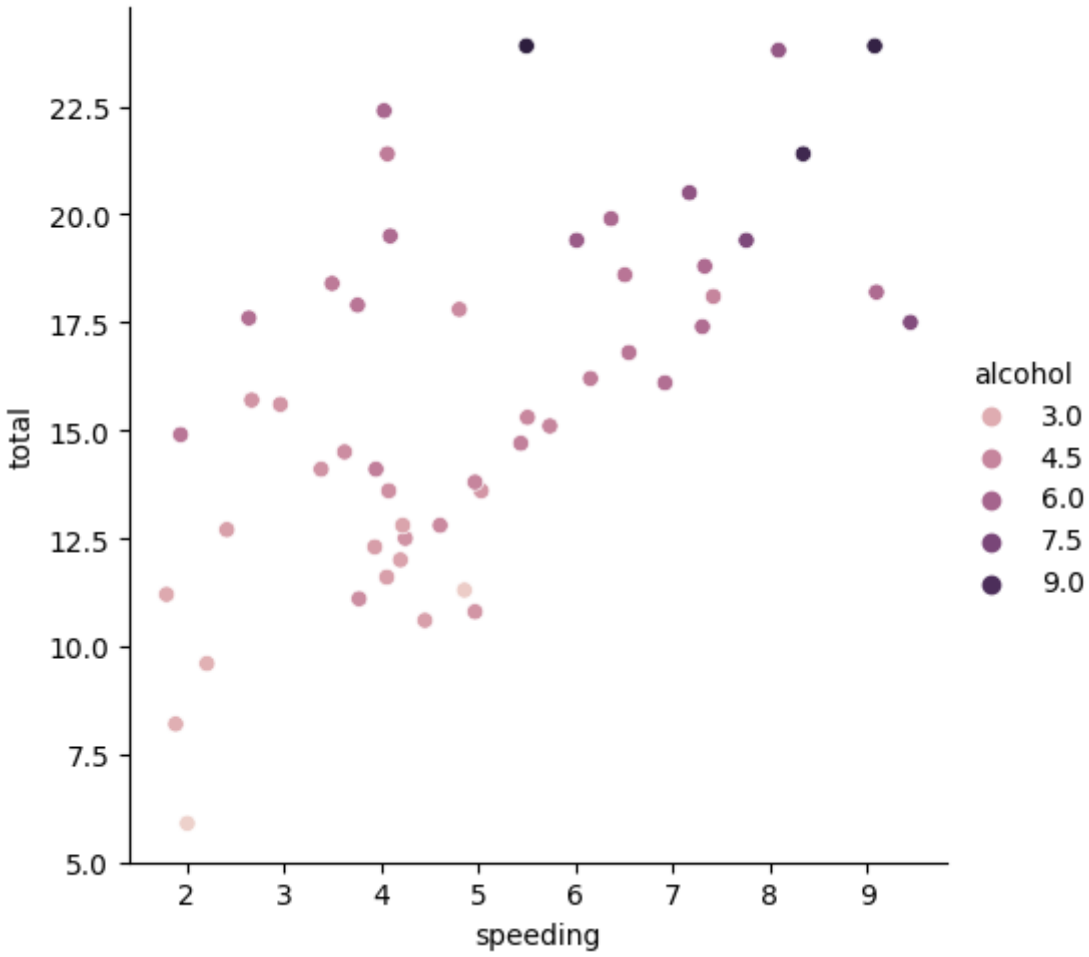
```
<Axes: xlabel='speeding', ylabel='total'>
```



Inference: As the number speeding based car crashes increases, then the total crashes has also increased

```
sns.relplot(x="speeding", y="total", hue="alcohol", data=df)
```

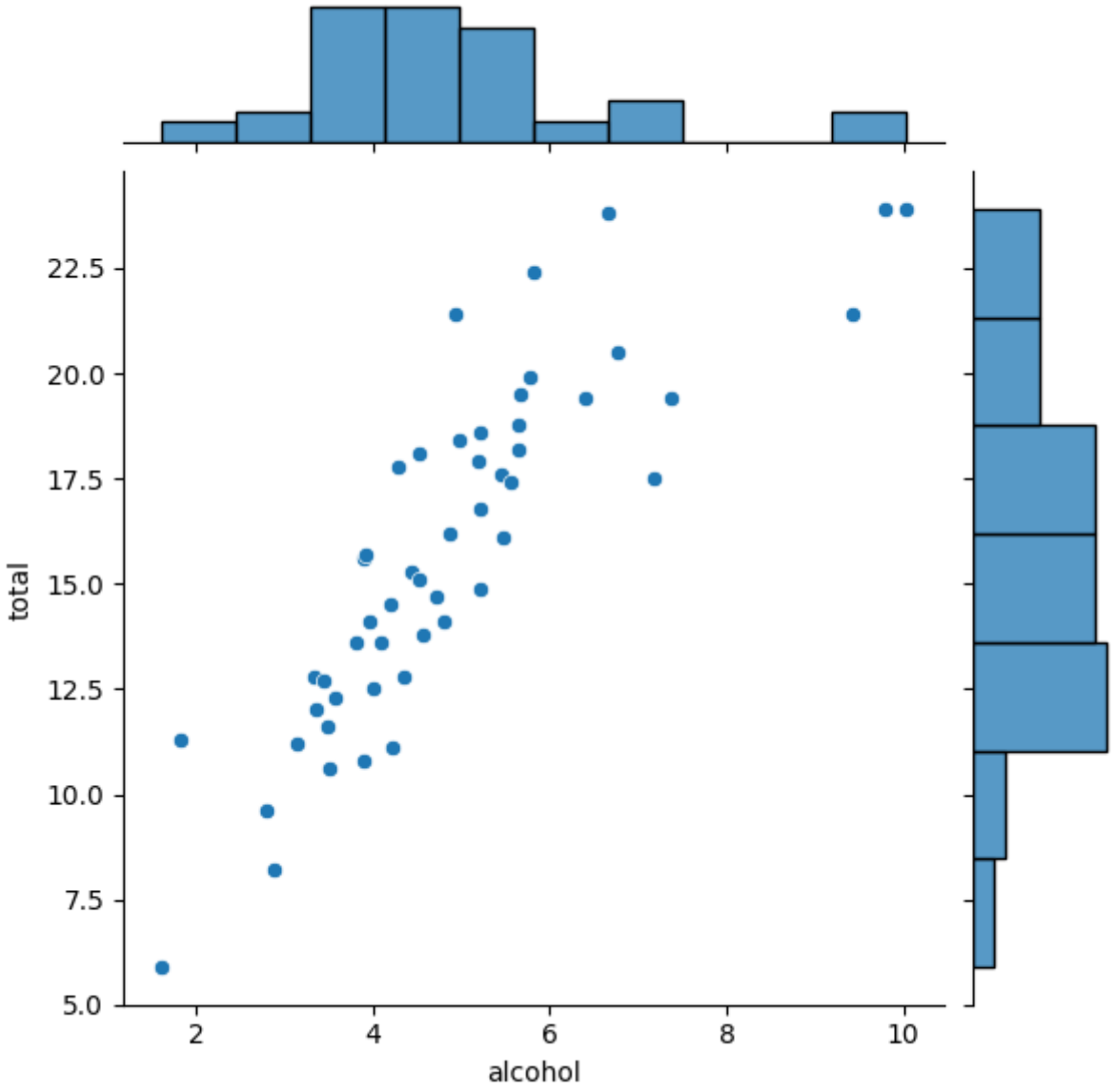
```
<seaborn.axisgrid.FacetGrid at 0x7cf9e1a91210>
```



Inference: The relplot is used to visualize the relationship between speeding-related crashes (x-axis) and total crashes (y-axis) while differentiating by alcohol-related crashes (color). This allows you to explore how alcohol-related crashes impact the speeding and total crash relationship.

```
sns.jointplot(x="alcohol", y="total", data=df, kind="scatter")
```

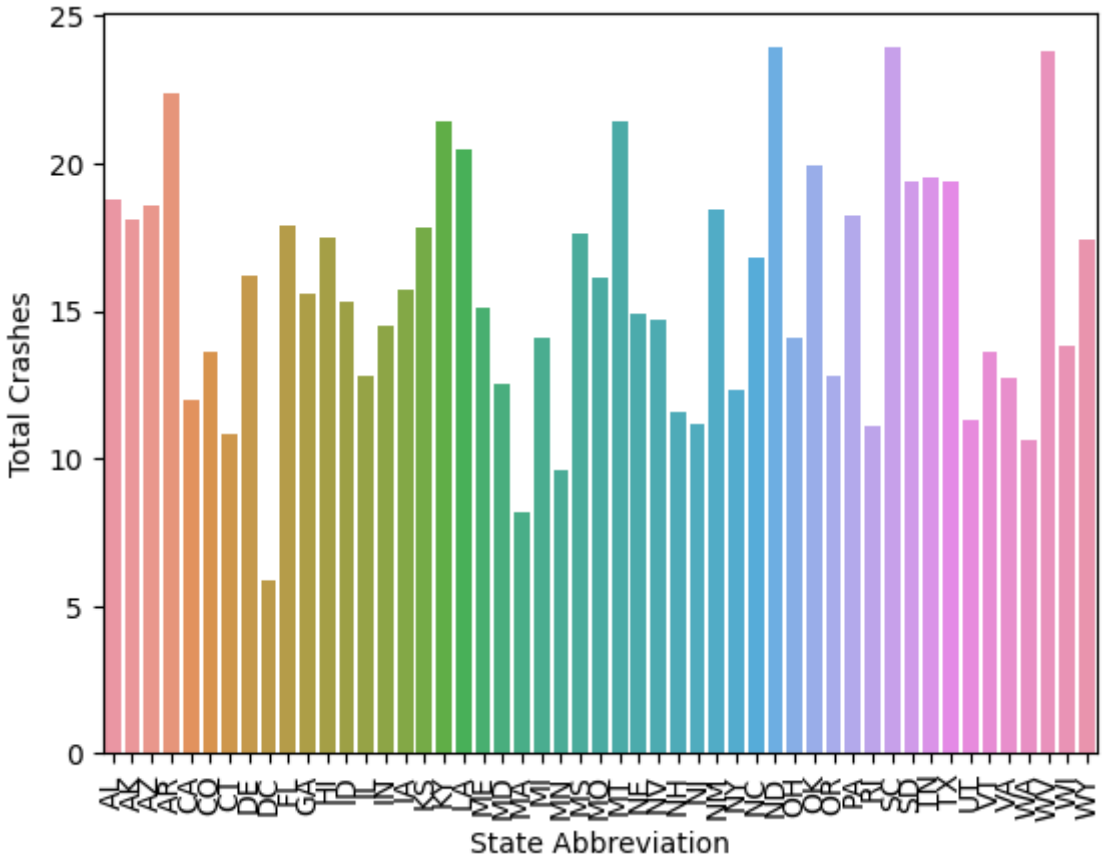
```
<seaborn.axisgrid.JointGrid at 0x7cf9d8603fd0>
```



Inference: The jointplot provides a scatter plot of alcohol-related crashes versus total crashes along with histograms for each variable. This helps in visualizing the bivariate distribution and marginal distributions of the two variables.

```
sns.barplot(x="abbrev", y="total", data=df,width=0.8)
plt.xlabel("State Abbreviation")
plt.ylabel("Total Crashes")
plt.xticks(rotation=90)
```

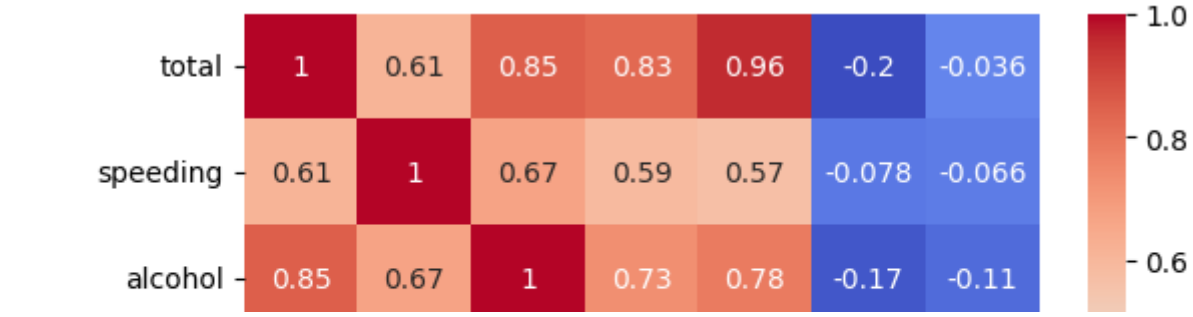
```
plt.show()
```



Inference: Shows the total crashes in different states using their abbreviations on the x-axis. can use this to compare crash counts across states.

```
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm")
```

```
<ipython-input-36-01a5e2923d95>:1: FutureWarning: The default value of numeric_only i
  corr_matrix = df.corr()
<Axes: >
```



Inference:"total" and "alcohol" have a relatively strong positive correlation.