ASSIGNMENT -2

GUDLA NISHIKA - 21BCE8562

VIT-AP

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
```

In [ ]:
```
data = pd.read_csv("car_crashes.csv")
```

In [ ]:
```
data
```

Out[ ]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
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
20	12.5	4.250	4.000	8.875	12.375	1048.78	192.70
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.63
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26

23	9.6	2.208	2.784	8.448	8.448	777.18	133.35
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45
26	21.4	8.346	9.416	17.976	18.190	816.21	85.15
27	14.9	1.937	5.215	13.857	13.410	732.28	114.82
28	14.7	5.439	4.704	13.965	14.553	1029.87	138.71
29	11.6	4.060	3.480	10.092	9.628	746.54	120.21
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85
31	18.4	3.496	4.968	12.328	18.032	869.85	120.75
32	12.3	3.936	3.567	10.824	9.840	1234.31	150.01
33	16.8	6.552	5.208	15.792	13.608	708.24	127.82
34	23.9	5.497	10.038	23.661	20.554	688.75	109.72
35	14.1	3.948	4.794	13.959	11.562	697.73	133.52
36	19.9	6.368	5.771	18.308	18.706	881.51	178.86
37	12.8	4.224	3.328	8.576	11.520	804.71	104.61
38	18.2	9.100	5.642	17.472	16.016	905.99	153.86
39	11.1	3.774	4.218	10.212	8.769	1148.99	148.58
40	23.9	9.082	9.799	22.944	19.359	858.97	116.29
41	19.4	6.014	6.402	19.012	16.684	669.31	96.87
42	19.5	4.095	5.655	15.990	15.795	767.91	155.57
43	19.4	7.760	7.372	17.654	16.878	1004.75	156.83
44	11.3	4.859	1.808	9.944	10.848	809.38	109.48
45	13.6	4.080	4.080	13.056	12.920	716.20	109.61
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72
47	10.6	4.452	3.498	8.692	9.116	890.03	111.62
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56
49	13.8	4.968	4.554	5.382	11.592	670.31	106.62
50	17.4	7.308	5.568	14.094	15.660	791.14	122.04

In [ ]:

```
data.head()
```

Out[ ]:

	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	

In [ ]:

```
data.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
```

In [ ]:
```
data.describe()
```

Out[ ]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	i
count	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	1
std	4.122002	2.017747	1.729133	4.508977	3.764672	178.296285	
min	5.900000	1.792000	1.593000	1.760000	5.900000	641.960000	
25%	12.750000	3.766500	3.894000	10.478000	11.348000	768.430000	1
50%	15.600000	4.608000	4.554000	13.857000	13.775000	858.970000	1
75%	18.500000	6.439000	5.604000	16.140000	16.755000	1007.945000	1
max	23.900000	9.450000	10.038000	23.661000	21.280000	1301.520000	1

In [ ]:
```
data.shape
```

Out[ ]:
```
(51, 8)
```

HANDLING NULL VALUES

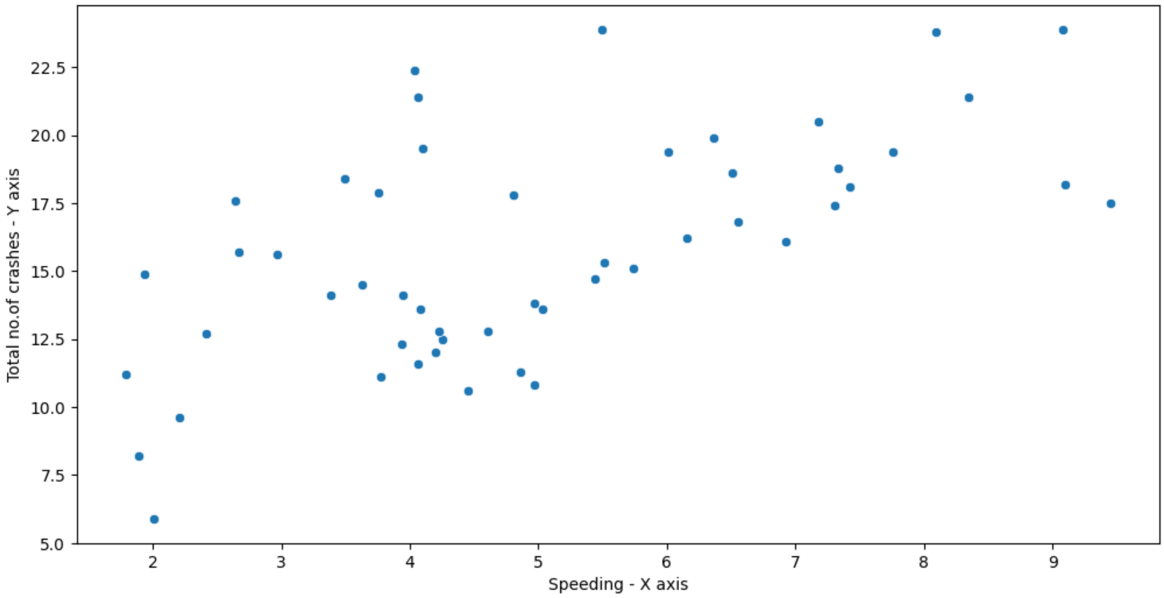
In [ ]:
```
data.isnull().any()
```

Out[ ]:
```
total             False
speeding          False
alcohol           False
not_distracted    False
no_previous       False
ins_premium       False
ins_losses        False
abbrev            False
dtype: bool
```

1. Univariate analysis - One variable is taken at a time
2. Bivariate analysis - Two variables are taken at a time
3. Multi variate analysis - Many variables are taken at a time

SCATTER PLOT

In [ ]:
```
#Speeding vs Total no.of crashes
plt.figure(figsize=(12,6))
sns.scatterplot(x='speeding', y='total', data= data)
plt.xlabel('Speeding - X axis')
plt.ylabel('Total no.of crashes - Y axis')
plt.show()
```

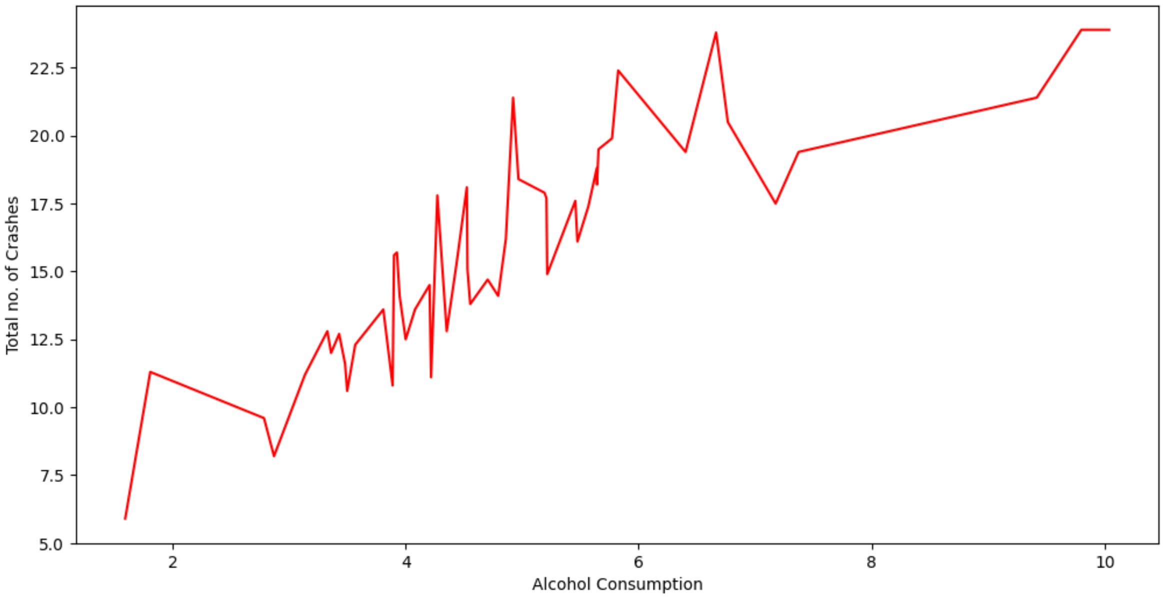


INFERENCE: The scatter plot shows a positive correlation between speeding and the total number of crashes. This means that as the percentage of accidents where the driver was speeding increases, the total number of crashes also tends to increase.

The correlation is not perfect, however, as there are some states with a high percentage of speeding accidents that have a relatively low number of total crashes, and vice versa. This suggests that other factors, also play a role in the number of car crashes.

LINE PLOT

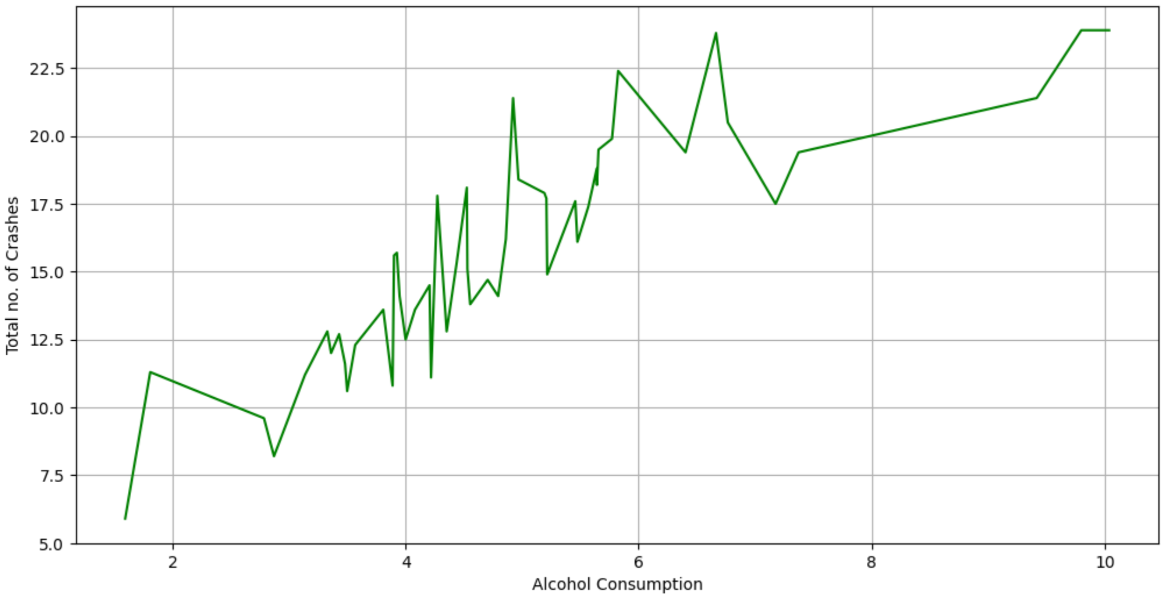
In [ ]:
```
#Alcohol Consumption vs Total no.of crashes
plt.figure(figsize=(12,6))
sns.lineplot(x='alcohol', y='total', data=data, errorbar=None, color = "r
plt.xlabel('Alcohol Consumption')
plt.ylabel('Total no. of Crashes')
plt.show()
```



INFERENCE: The line plot shows a positive correlation between alcohol consumption and the total number of crashes. This means that as the percentage of accidents where the driver was under the influence of alcohol increases, the total number of crashes also tends to increase.

The correlation is stronger than the correlation between speeding and the total number of crashes. This suggests that alcohol consumption is a more significant factor in the number of car accidents than speeding.

In [ ]:
```
plt.figure(figsize=(12,6))
sns.lineplot(x='alcohol', y='total', data=data, errorbar=None, color = "g
plt.xlabel('Alcohol Consumption')
plt.ylabel('Total no. of Crashes')
plt.grid()
```

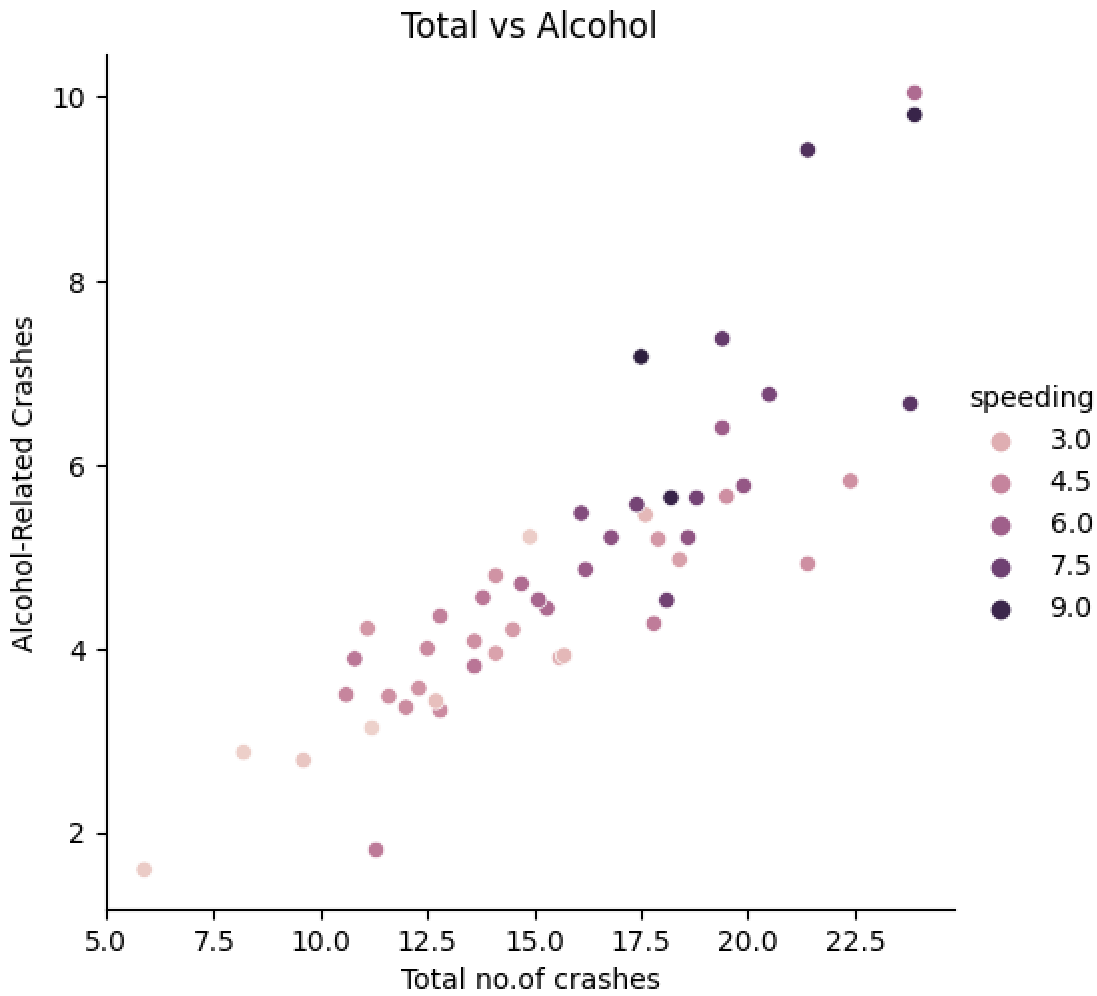


RELATIONAL PLOT

In [ ]:
```
plt.figure(figsize=(12,6))
```

```python
sns.relplot(x="total", y="alcohol", data=data, hue="speeding")
plt.title("Total vs Alcohol")
plt.xlabel("Total no.of crashes")
plt.ylabel("Alcohol-Related Crashes")
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```

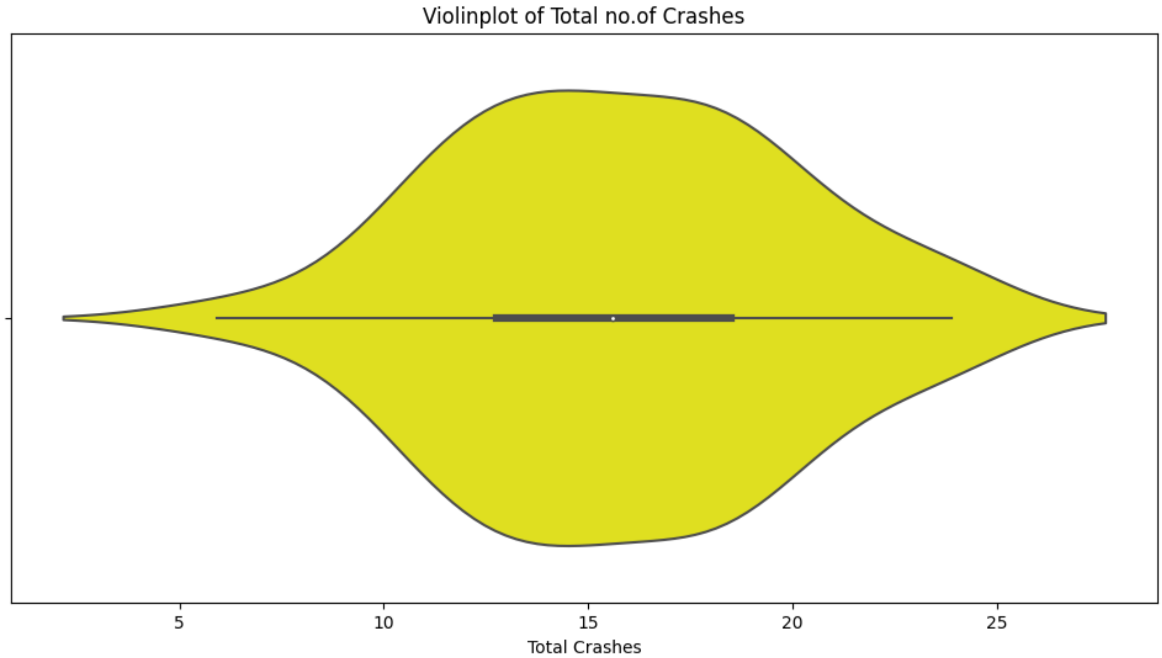


INFERENCE: The relational plot shows that there is a positive correlation between the total number of crashes and the percentage of alcohol-related crashes. This means that states with a higher total number of crashes also tend to have a higher percentage of alcohol-related crashes.

The correlation is stronger for states with a higher percentage of speeding accidents. This shows that speeding and alcohol consumption are both factors that contribute to alcohol-related crashes.

VIOLIN PLOT

In [ ]:
```python
plt.figure(figsize=(12, 6))
sns.violinplot(x="total", data= data, color= "yellow")
plt.title("Violinplot of Total no.of Crashes")
plt.xlabel("Total Crashes")
plt.show()
```

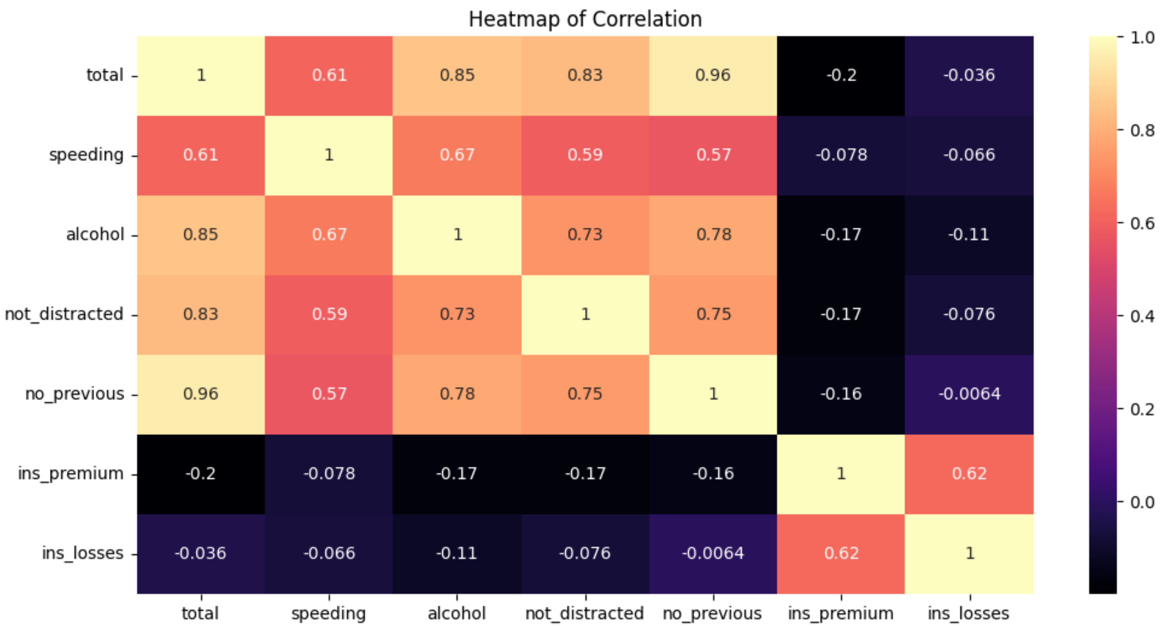


INFERENCE: The violin plot shows the distribution of the total number of crashes in each state. The thicker part of the violin represents the most common values, and the thinner parts represent less common values.

There is no clear pattern in the distribution of the total number of crashes.Some states have a relatively low number of crashes, while others have a much higher number of crashes.

HEATMAP

In [ ]:
```
plt.figure(figsize=(12, 6))
correlation_matrix = data.corr(numeric_only=True)
sns.heatmap(correlation_matrix, annot=True, cmap="magma")
plt.title("Heatmap of Correlation")
plt.show()
```

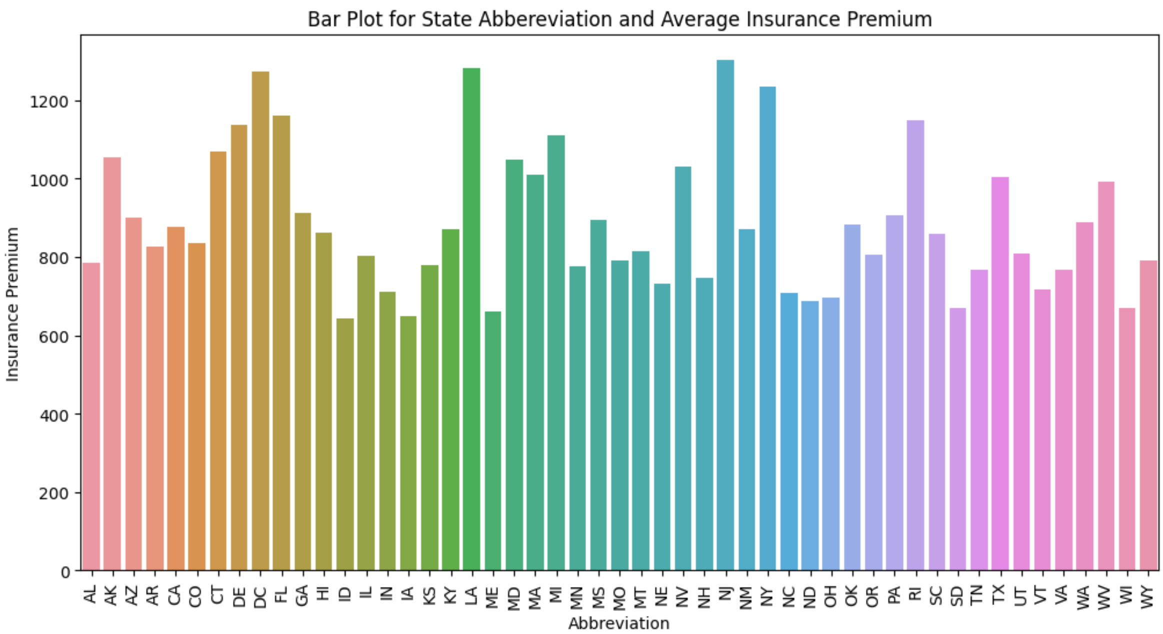


INFERENCE: The heatmap shows the correlation between all the variables in the dataset. The darker the cell, the stronger the correlation.

The strongest correlation is between speeding and alcohol consumption, followed by speeding and total number of crashes. This suggests that speeding and alcohol consumption are two of the most important factors that contribute to car accidents.

BAR PLOT

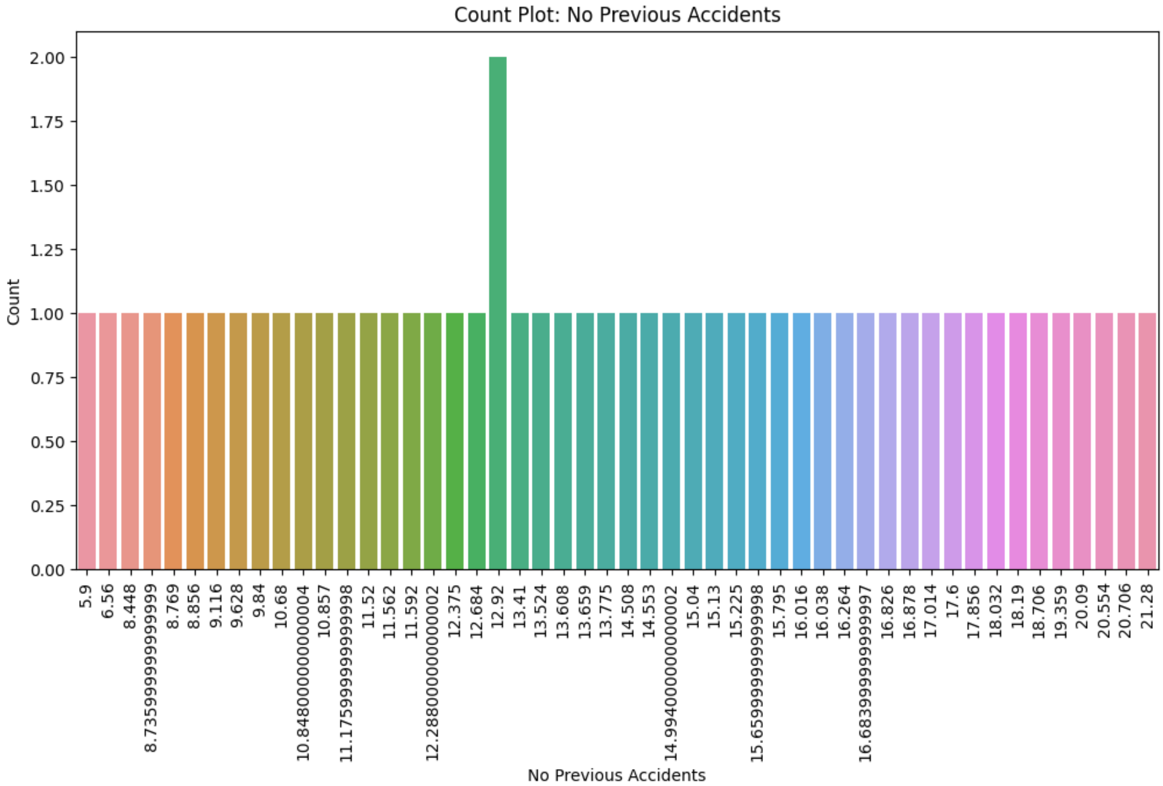
In [ ]:
```
plt.figure(figsize=(12,6))
sns.barplot(x="abbrev", y="ins_premium", data=data,errorbar=None)
plt.title("Bar Plot for State Abbereviation and Average Insurance Premium
plt.xlabel("Abbreviation")
plt.ylabel("Insurance Premium")
plt.xticks(rotation=90) #to make sure all words are visible
plt.show()
```



INFERENCE: The bar plot shows that the average insurance premium varies significantly from state to state. The states with the highest average insurance premiums tend to have stricter laws and higher rates of accidents, while the states with the lowest average insurance premiums tend to have less strict laws and lower rates of accidents.

COUNT PLOT

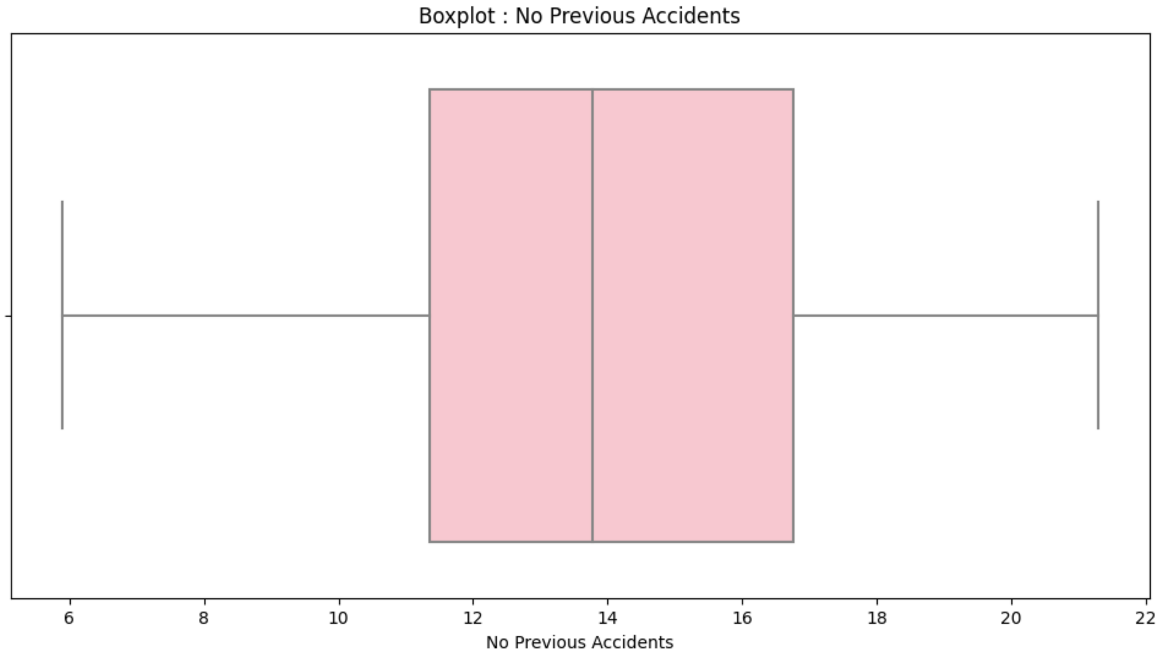
In [ ]:
```
plt.figure(figsize=(12, 6))
sns.countplot(x="no_previous", data=data)
plt.title("Count Plot: No Previous Accidents")
plt.xlabel("No Previous Accidents")
plt.ylabel("Count")
plt.xticks(rotation=90)
plt.show()
```



INFERENCE: The percentage of drivers with no previous accidents varies significantly from state to state. The states with the highest percentages tend to have lower rates of car accidents. This suggests that drivers with no previous accidents are less likely to be involved in car accidents.

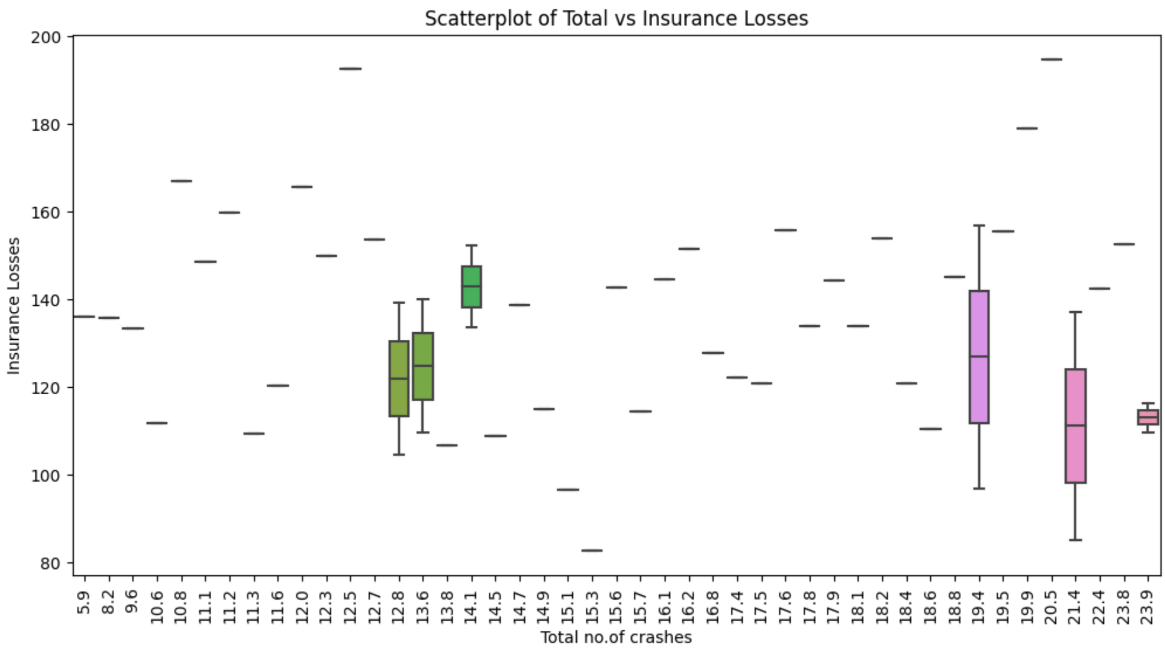
BOX PLOT

In [ ]:
```
plt.figure(figsize=(12,6))
sns.boxplot(x="no_previous", data=data, color="pink")
plt.title("Boxplot : No Previous Accidents")
plt.xlabel("No Previous Accidents")
plt.show()
```



INFERENCE: The box plot shows that the percentage of drivers with no previous accidents varies significantly. The states with the highest percentages are in the 80% range, while the states with the lowest percentages are in the 60% range. This shows that there are a number of other factors that can influence the percentage of drivers with no previous accidents.

In [ ]:
```
# Boxplot for two different colums Total and Insurance Losses
plt.figure(figsize=(12,6))
sns.boxplot(x="total", y="ins_losses", data=data)
plt.title("Scatterplot of Total vs Insurance Losses")
plt.xlabel("Total no.of crashes")
plt.ylabel("Insurance Losses")
plt.xticks(rotation=90)
plt.show()
```

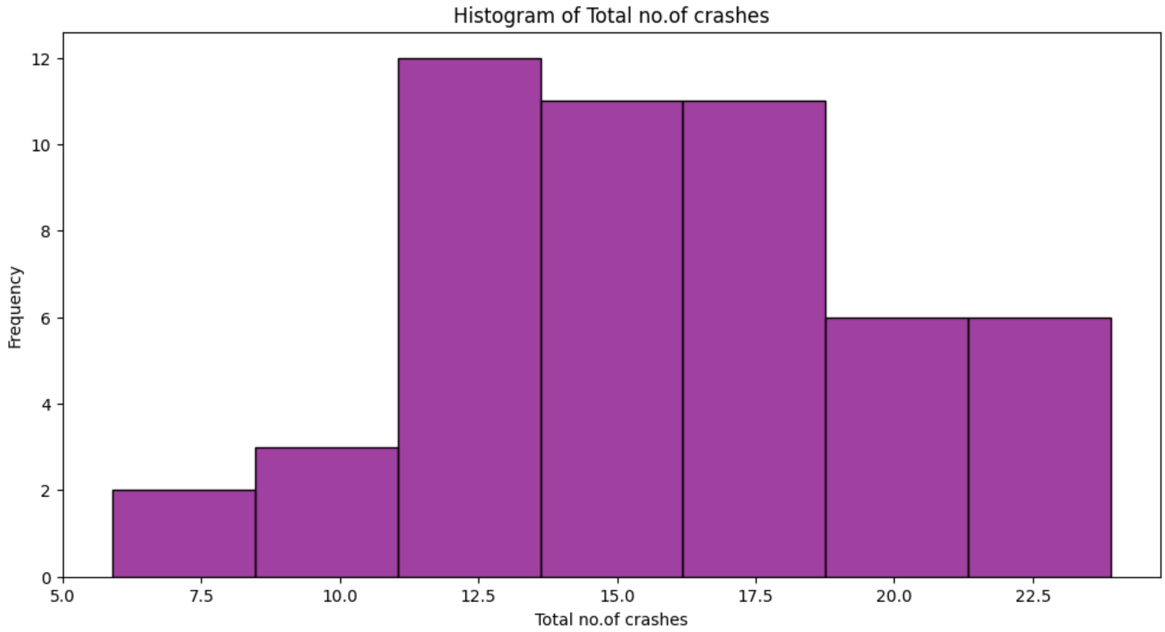


INFERENCE: The box plot shows that there is a positive correlation between the total number of crashes and insurance losses. This means that as the total number of crashes increases, the insurance losses also tend to increase.

The correlation is not perfect, however. There are some states with a high total number of crashes but low insurance losses, and vice versa. This suggests that there are other factors that can influence insurance losses.

HISTOGRAM

In [ ]:
```
plt.figure(figsize=(12, 6))
sns.histplot(data["total"], color="purple")
plt.title("Histogram of Total no.of crashes")
plt.xlabel("Total no.of crashes")
plt.ylabel("Frequency")
plt.show()
```

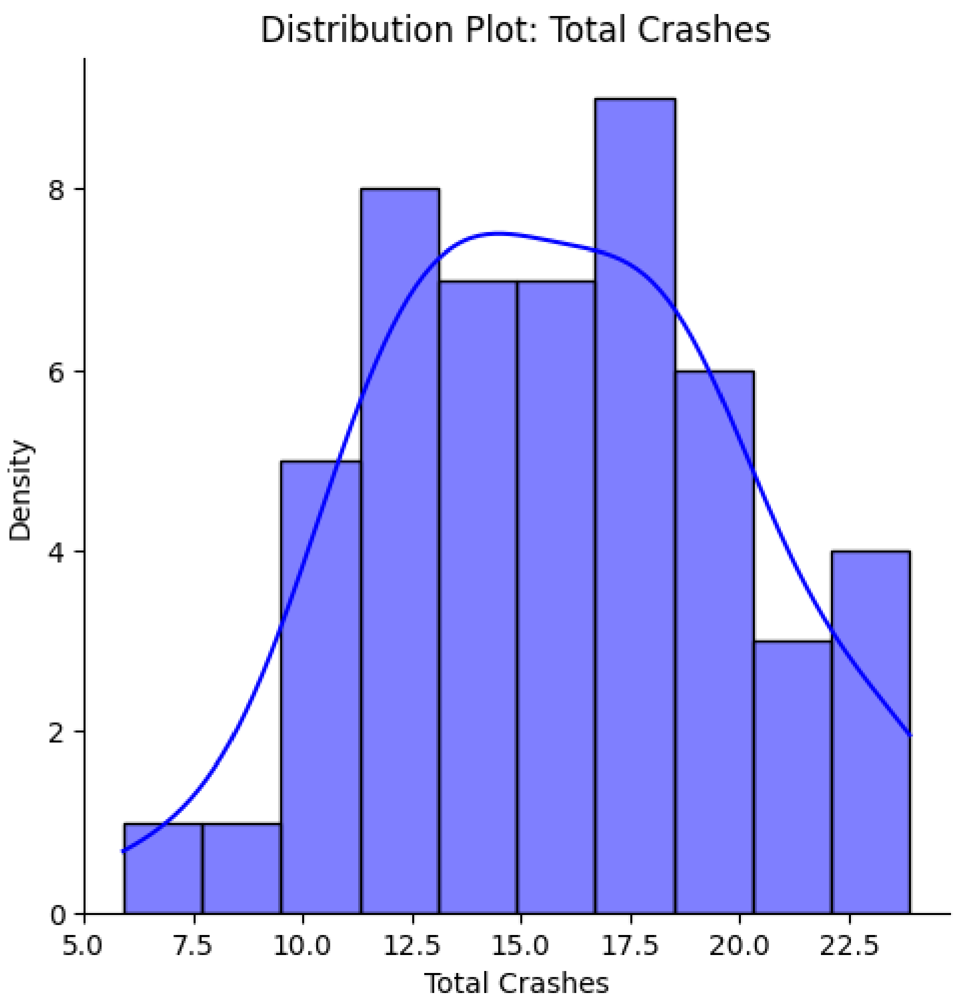


INFERENCE: The histogram shows the distribution of the total number of crashes in each state. The distribution is right-skewed, meaning that there are more states with a lower number of crashes than states with a higher number of crashes.

DISTRIBUTION PLOT

In [ ]:
```
plt.figure(figsize=(12, 6))
sns.displot(data["total"], bins=10, kde=True, color="blue")
plt.title("Distribution Plot: Total Crashes")
plt.xlabel("Total Crashes")
plt.ylabel("Density")
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```



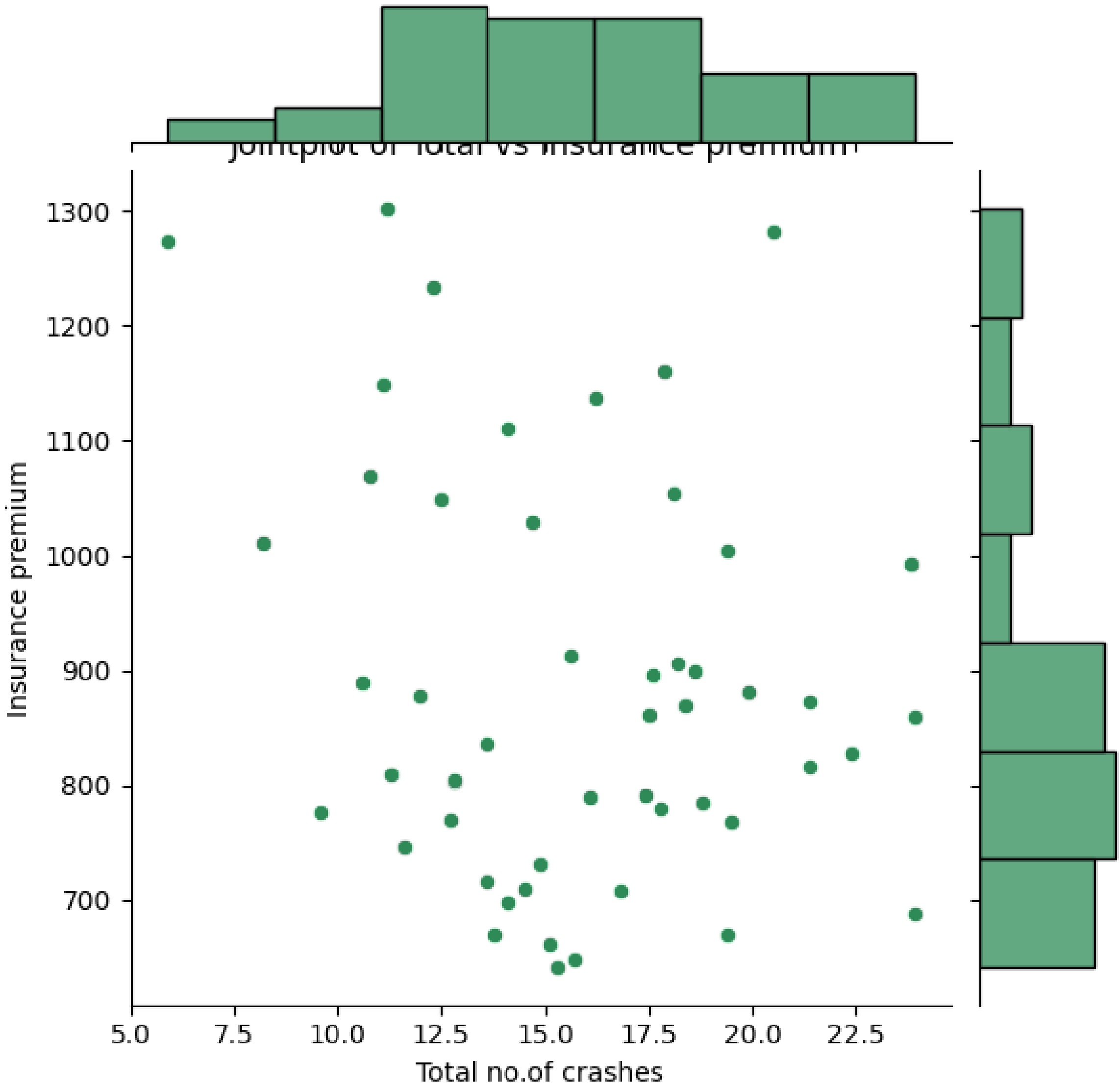
INFERENCE: The distribution plot shows that the total number of crashes in each state is normally distributed. This means that most states have a total number of crashes that is close to the average of 10. There is a small number of states with a lower number of crashes (less than 5) and a small number of states with a higher number of crashes (more than 15).

JOINT PLOT

In [ ]:

```
plt.figure(figsize=(12,6))
sns.jointplot(x="total", y="ins_premium", data=data, color="seagreen")
plt.title("Jointplot of Total vs Insurance premium")
plt.xlabel("Total no.of crashes")
plt.ylabel("Insurance premium")
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```



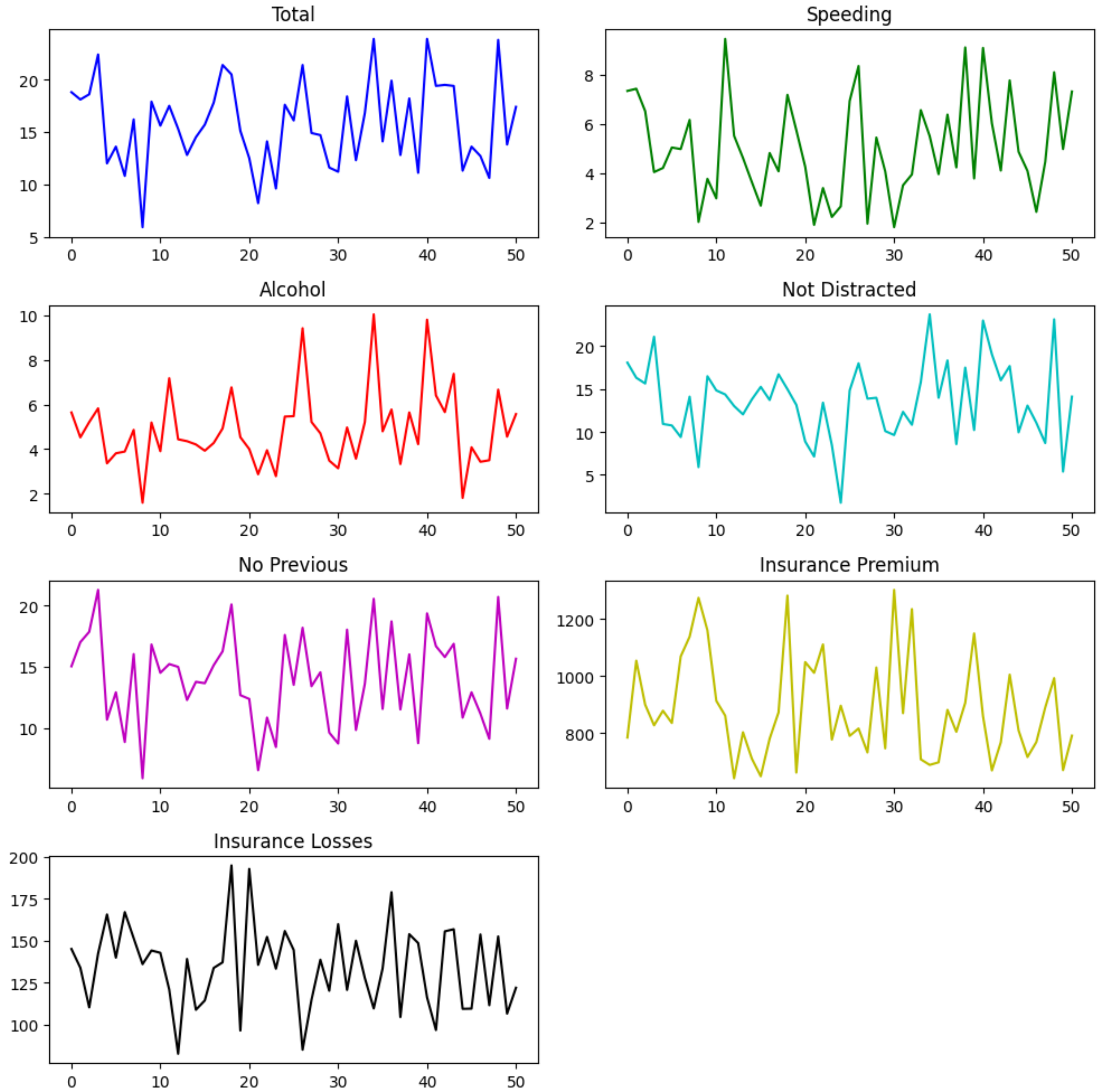
INFERENCE: The joint plot shows the relationship between the total number of crashes and insurance premium. The two variables are positively correlated, which means that as the total number of crashes increases, the insurance premium also tends to increase.

The correlation is not perfect, however, there are some states with a high total number of crashes but low insurance premiums, and vice versa. This suggests that there are other factors that can influence insurance premiums.

SUBPLOTS

In [ ]:
```
plt.figure(figsize=(10, 10))

plt.subplot(4, 2, 1)
plt.plot(data['total'], 'b')
plt.title('Total')

plt.subplot(4, 2, 2)
plt.plot(data['speeding'], 'g')
plt.title('Speeding')

plt.subplot(4, 2, 3)
```

```
plt.plot(data['alcohol'], 'r')
plt.title('Alcohol')

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

plt.subplot(4, 2, 5)
plt.plot(data['no_previous'], 'm')
plt.title('No Previous')

plt.subplot(4, 2, 6)
plt.plot(data['ins_premium'], 'y')
plt.title('Insurance Premium')

plt.subplot(4, 2, 7)
plt.plot(data['ins_losses'], 'k')
plt.title('Insurance Losses')

plt.tight_layout()
```



INFERENCE: These are subplots for all the columns in the dataset