Assignment 2

Name: DHARMANA GNANA SAI

Reg no: 21BCE7400

Branch: CSE AI and ML

Email: gnanasai.21bce7400@vitapstudent.ac.in

Campus: VIT- AP

In [21]:

```
import pandas as pd
```

In [22]:

```
import matplotlib.pyplot as plt
```

In [23]:

```
import seaborn as sns
```

In [24]:

```
# Car crashes dataset
df = sns.load_dataset("car_crashes")
df
```

Out[24]:

	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
5	13.6	5.032	3.808	10.744	12.920	835.50	139.91	CO
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.02	CT
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	DE
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	DC
9	17.9	3.759	5.191	16.468	16.826	1160.13	144.18	FL
10	15.6	2.964	3.900	14.820	14.508	913.15	142.80	GA
11	17.5	9.450	7.175	14.350	15.225	861.18	120.92	HI
12	15.3	5.508	4.437	13.005	14.994	641.96	82.75	ID
13	12.8	4.608	4.352	12.032	12.288	803.11	139.15	IL
14	14.5	3.625	4.205	13.775	13.775	710.46	108.92	IN
15	15.7	2.669	3.925	15.229	13.659	649.06	114.47	IA
16	17.8	4.806	4.272	13.706	15.130	780.45	133.80	KS
17	21.4	4.066	4.922	16.692	16.264	872.51	137.13	KY
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
19	15.1	5.738	4.530	13.137	12.684	661.88	96.57	ME
20	12.5	4.250	4.000	8.875	12.375	1048.78	192.70	MD
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.63	MA
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26	MI
23	9.6	2.208	2.784	8.448	8.448	777.18	133.35	MN
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77	MS
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45	MO
26	21.4	8.346	9.416	17.976	18.190	816.21	85.15	MT
27	14.9	1.937	5.215	13.857	13.410	732.28	114.82	NE
28	14.7	5.439	4.704	13.965	14.553	1029.87	138.71	NV
29	11.6	4.060	3.480	10.092	9.628	746.54	120.21	NH
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85	NJ
31	18.4	3.496	4.968	12.328	18.032	869.85	120.75	NM
32	12.3	3.936	3.567	10.824	9.840	1234.31	150.01	NY

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
33	16.8	6.552	5.208	15.792	13.608	708.24	127.82	NC
34	23.9	5.497	10.038	23.661	20.554	688.75	109.72	ND
35	14.1	3.948	4.794	13.959	11.562	697.73	133.52	OH
36	19.9	6.368	5.771	18.308	18.706	881.51	178.86	OK
37	12.8	4.224	3.328	8.576	11.520	804.71	104.61	OR
38	18.2	9.100	5.642	17.472	16.016	905.99	153.86	PA
39	11.1	3.774	4.218	10.212	8.769	1148.99	148.58	RI
40	23.9	9.082	9.799	22.944	19.359	858.97	116.29	SC
41	19.4	6.014	6.402	19.012	16.684	669.31	96.87	SD
42	19.5	4.095	5.655	15.990	15.795	767.91	155.57	TN
43	19.4	7.760	7.372	17.654	16.878	1004.75	156.83	TX
44	11.3	4.859	1.808	9.944	10.848	809.38	109.48	UT
45	13.6	4.080	4.080	13.056	12.920	716.20	109.61	VT
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72	VA
47	10.6	4.452	3.498	8.692	9.116	890.03	111.62	WA
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56	WV
49	13.8	4.968	4.554	5.382	11.592	670.31	106.62	WI
50	17.4	7.308	5.568	14.094	15.660	791.14	122.04	WY

In [25]:

```
df.head(5)
```

Out[25]:

	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

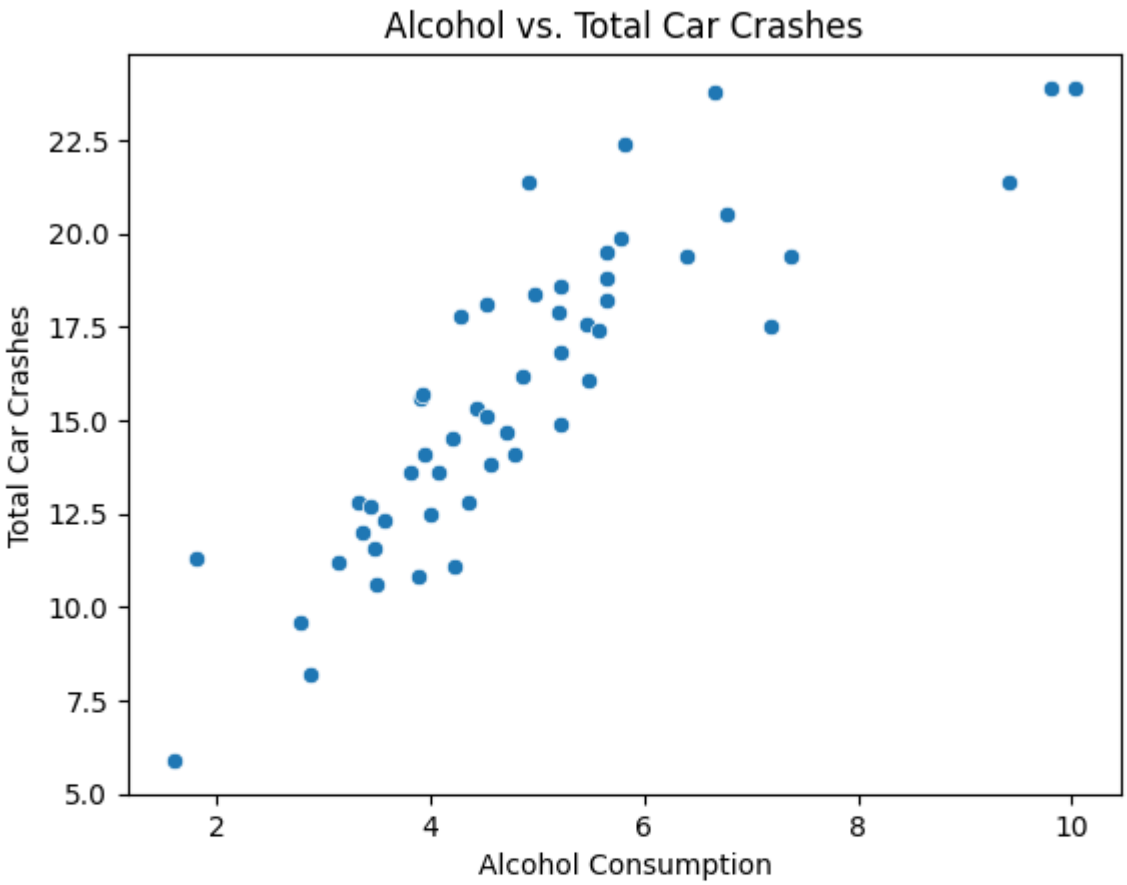
In [26]:

```
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
```

In [27]:

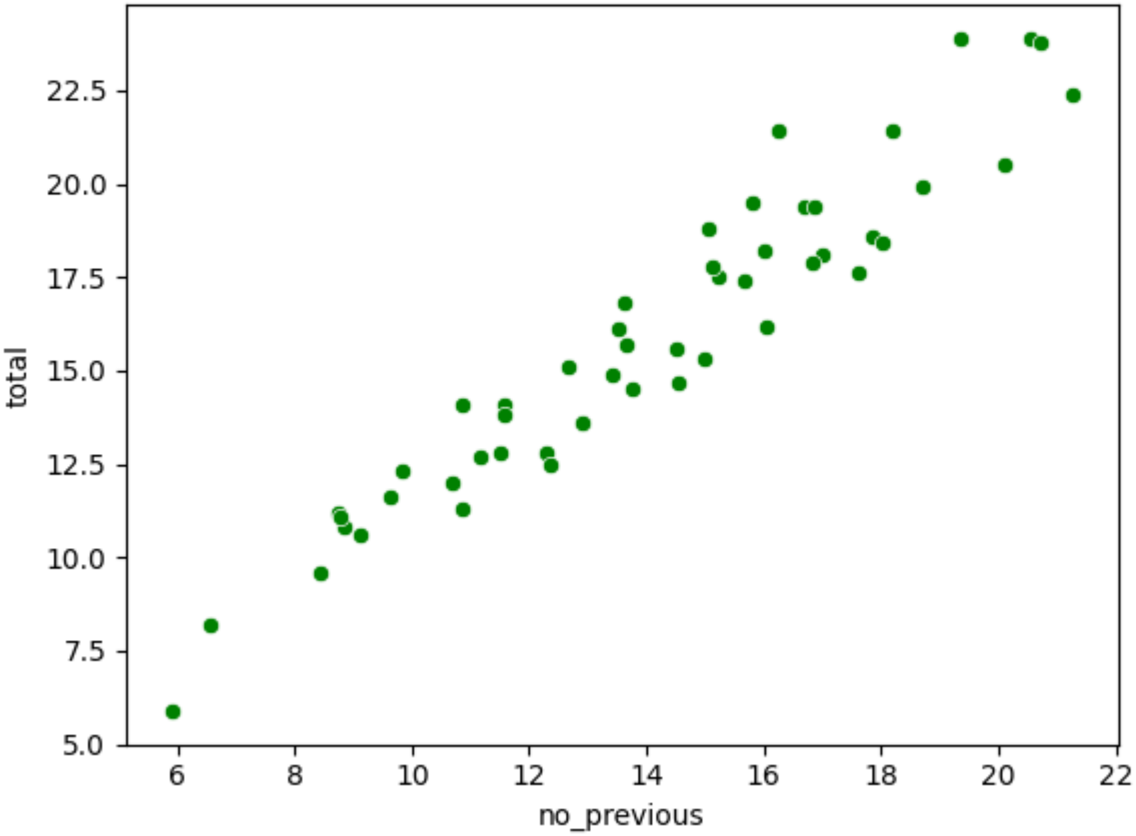
```
# Create a scatterplot
sns.scatterplot(data=df, x="alcohol", y="total")
plt.title("Alcohol vs. Total Car Crashes")
plt.xlabel("Alcohol Consumption")
plt.ylabel("Total Car Crashes")
plt.show()
```



Inference: This scatter plot shows the relationship between alcohol consumption and the total number of car crashes. You can infer whether there is a correlation between alcoholconsumption and crashes.

In [28]:

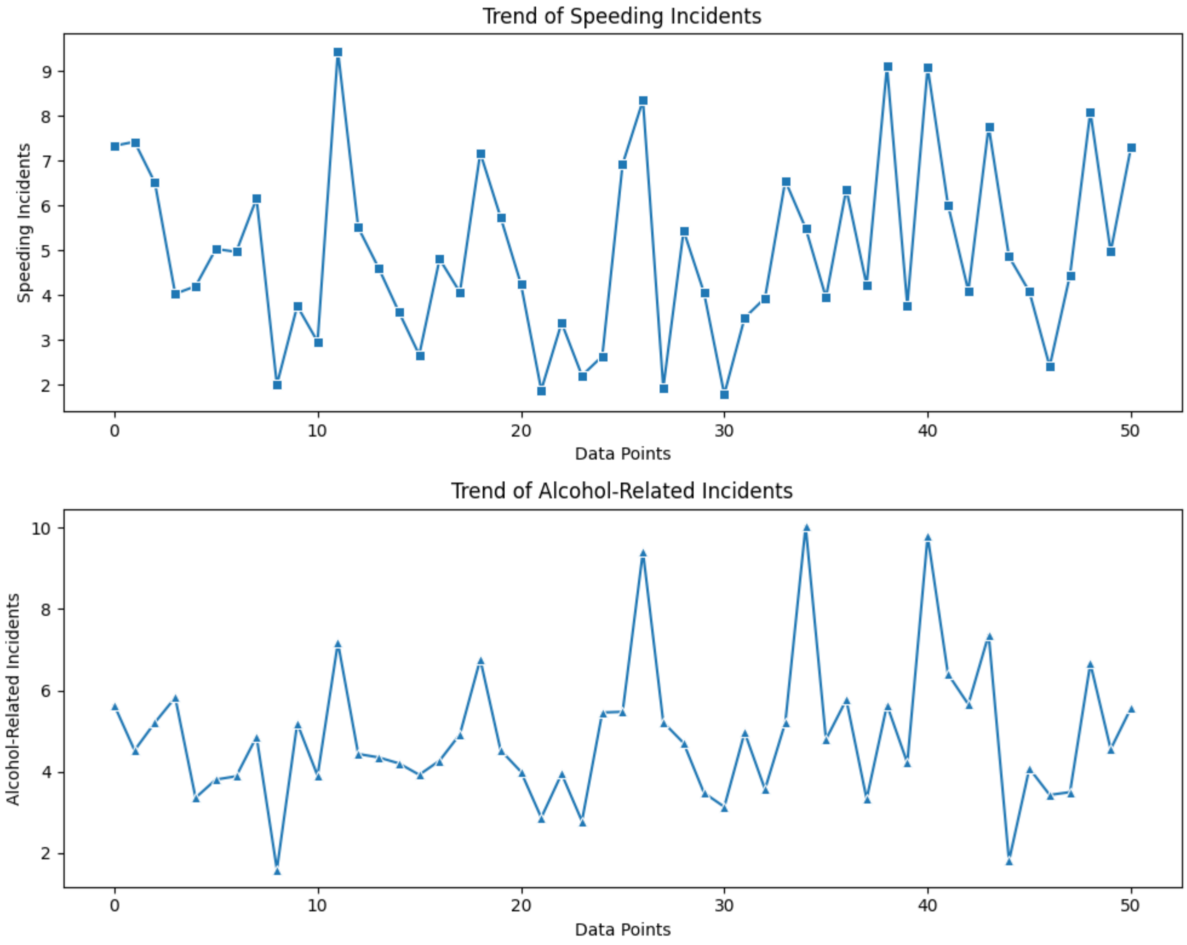
```
# Scatter plot
sns.scatterplot(x="no_previous", y="total", data=df, color = 'green')
plt.show()
```



Inference: This scatter plot shows the relationship between no_inference and the total number of car crashes. You can infer whether there is a correlation between no_previous and crashes.

In [29]:

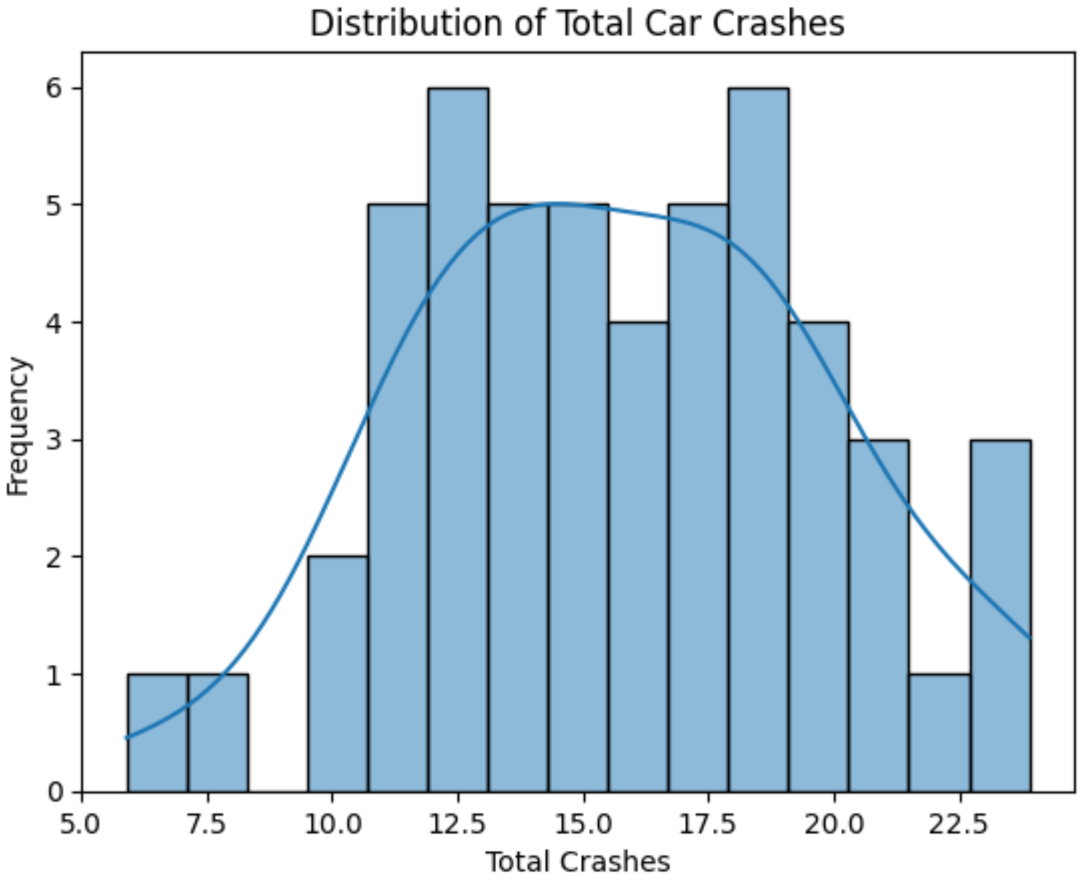
```
# Create subplots with two line graphs
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8))

# Line graph for 'speeding'
sns.lineplot(x=df.index, y='speeding',marker='s', data=df, ax=ax1)
ax1.set_title('Trend of Speeding Incidents')
ax1.set_xlabel('Data Points')
ax1.set_ylabel('Speeding Incidents')

# Line graph for 'alcohol'
sns.lineplot(x=df.index, y='alcohol',marker='^', data=df, ax=ax2)
ax2.set_title('Trend of Alcohol-Related Incidents')
ax2.set_xlabel('Data Points')
ax2.set_ylabel('Alcohol-Related Incidents')

plt.tight_layout()
plt.show()
```



Inference: The first line graph shows the trend of speeding incidents over time.The second line graph shows the trend of alcohol-related incidents over time.

In [30]:

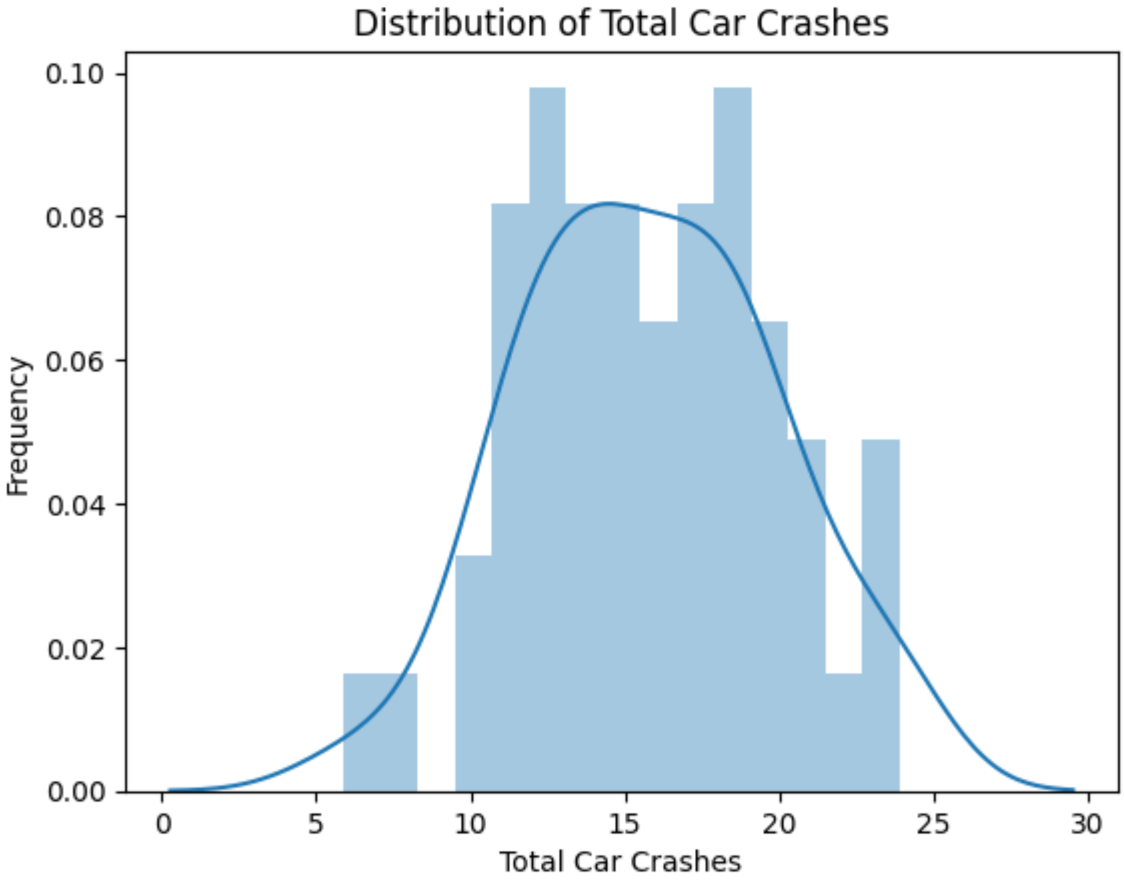
```
# Histplot
sns.histplot(df["total"], bins=15, kde=True)
plt.title("Distribution of Total Car Crashes")
plt.xlabel("Total Crashes")
plt.ylabel("Frequency")
plt.show()
```



Inference: This histogram shows the distribution of total car crashes. You can see the shape of the distribution and whether it is skewed.

```
In [31]: # Create a distplot to visualize the distribution of the "total" column
sns.distplot(df["total"], bins=15, kde=True)
plt.title('Distribution of Total Car Crashes')
plt.xlabel('Total Car Crashes')
plt.ylabel('Frequency')
plt.show()
```

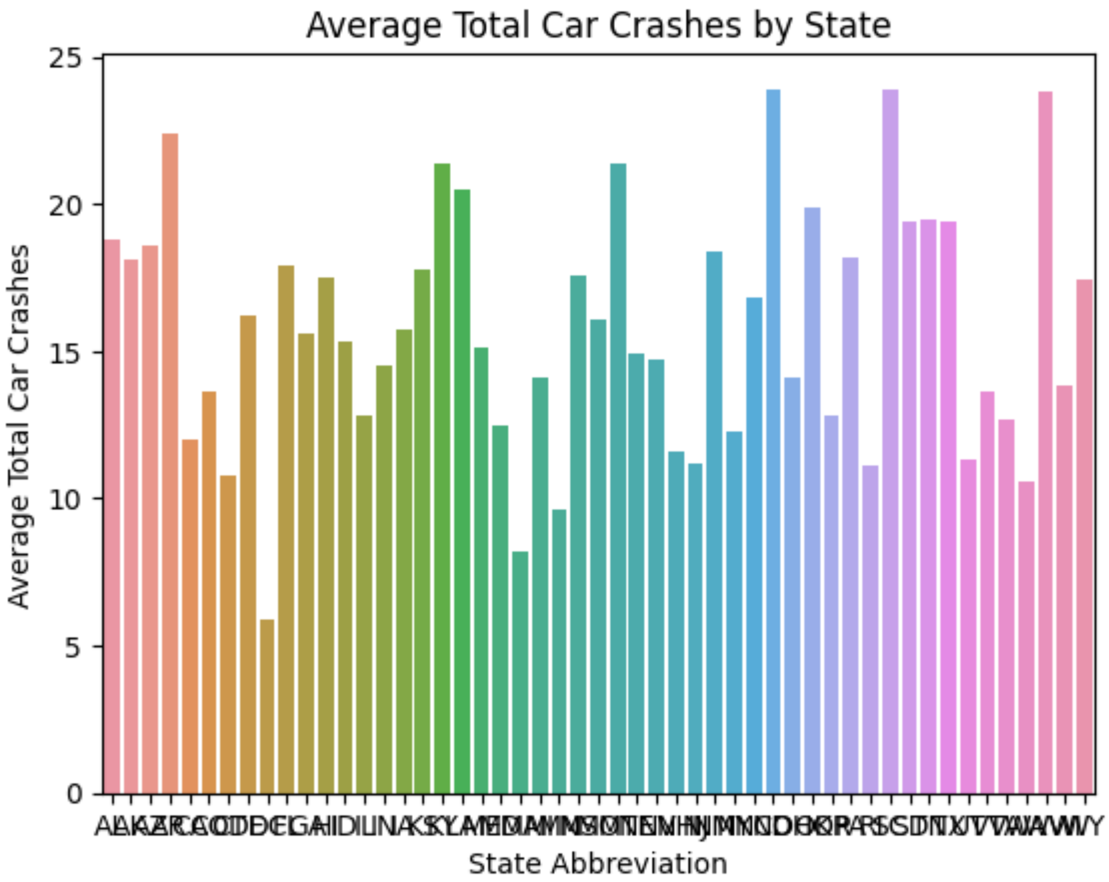
```
C:\Users\dharm\AppData\Local\Temp\ipykernel_27044\2404248515.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df["total"], bins=15, kde=True)
```



Inference: The distplot represents the distribution of "Total Car Crashes." It shows that the majority of observations cluster around a central point, creating a roughly symmetric distribution.

In [32]:

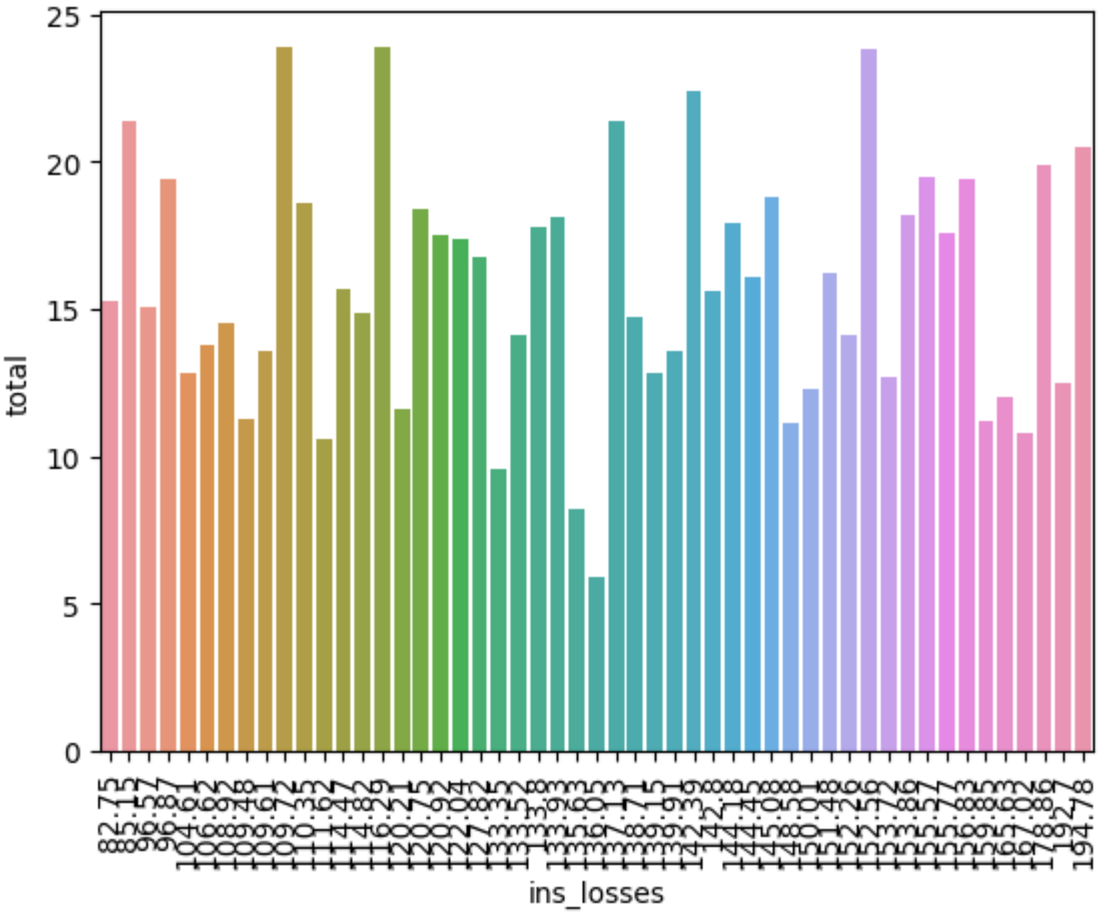
```
# Create a bar plot for average total car crashes by state
sns.barplot(data=df, x="abbrev", y="total")
plt.title("Average Total Car Crashes by State")
plt.xlabel("State Abbreviation")
plt.ylabel("Average Total Car Crashes")
plt.show()
```



Inference: bar plot shows the total number of car crashes for each state (abbrev). You can infer which states have the highest and lowest crash counts.

In [33]:

```
sns.barplot(x="ins_losses", y="total", data=df)
plt.xticks(rotation=90)
plt.show()
```

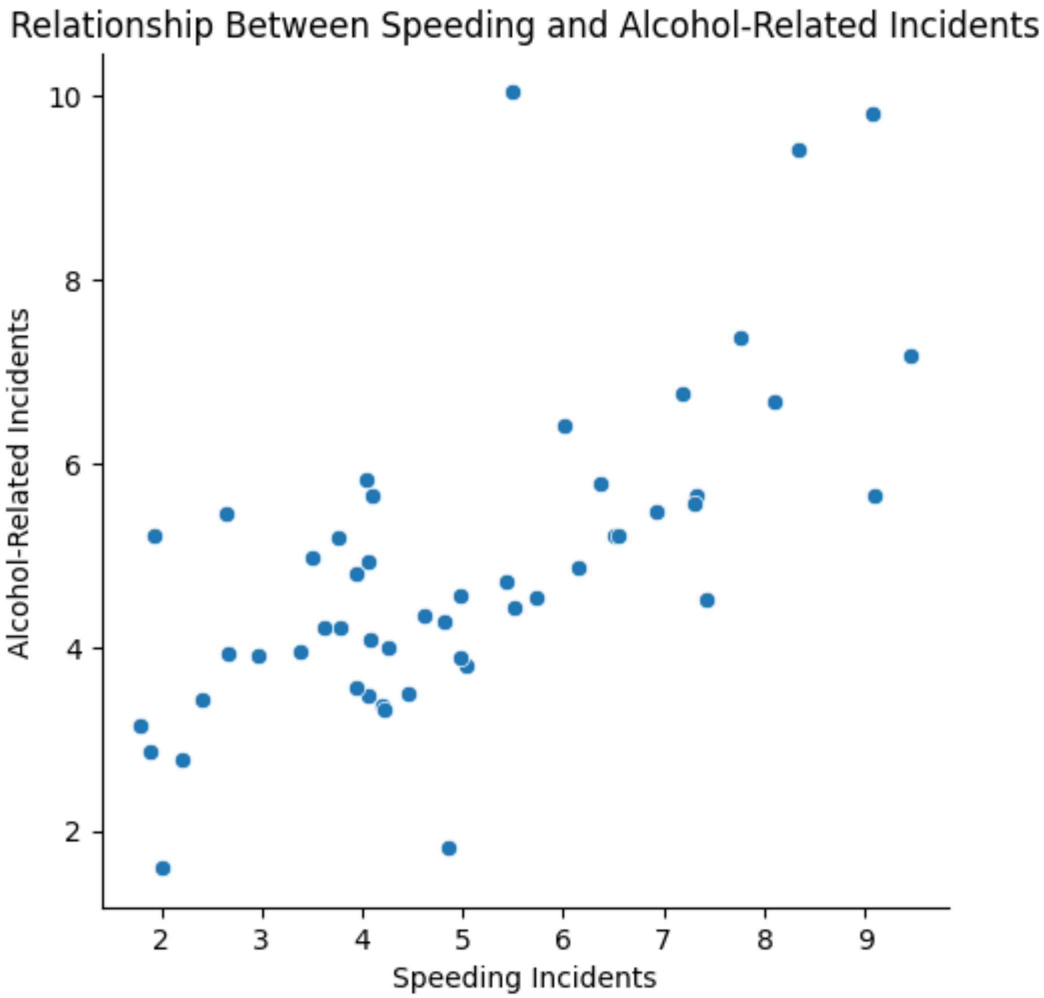


Inference: bar plot shows the total number of car crashes for each state (ins_losses). You can infer which states have the highest and lowest crash counts.

In [34]:

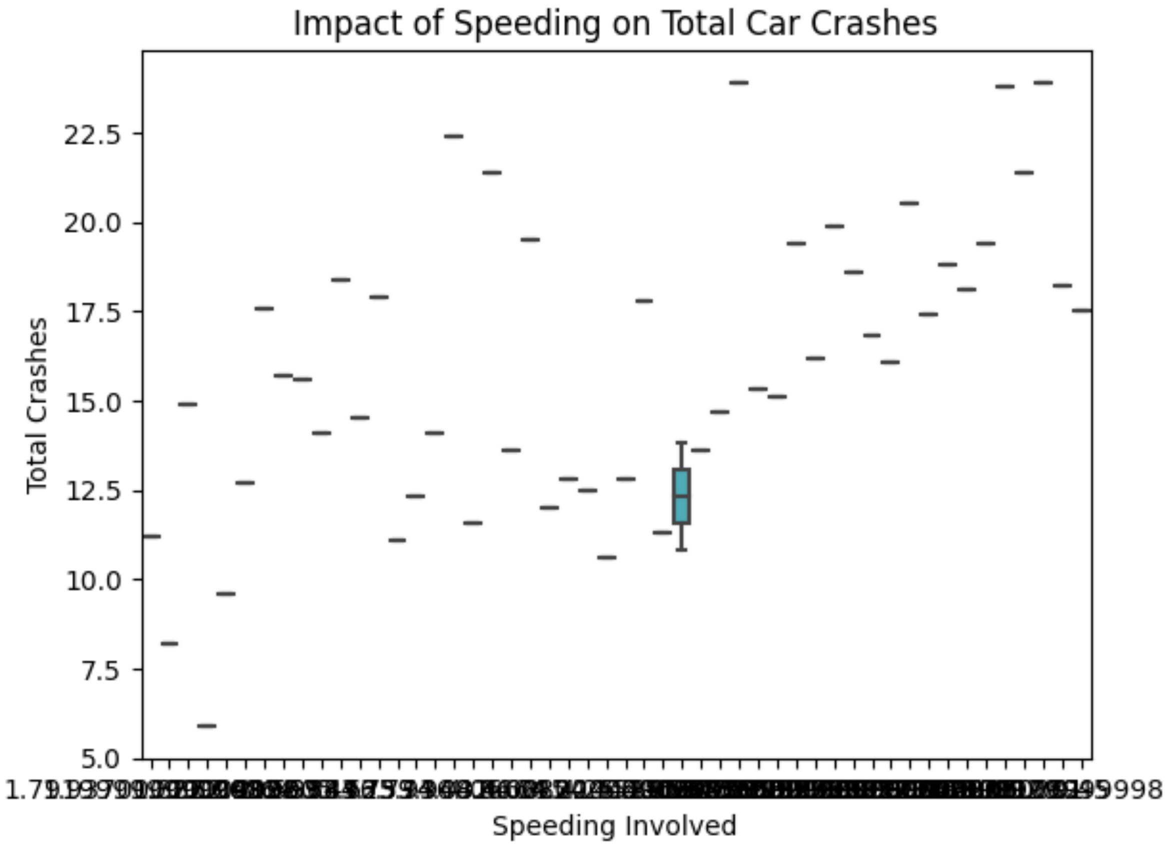
```
# Create a relational plot
sns.relplot(x="speeding", y="alcohol", data=df)
plt.title('Relationship Between Speeding and Alcohol-Related Incidents')
plt.xlabel('Speeding Incidents')
plt.ylabel('Alcohol-Related Incidents')
plt.show()
```

```
C:\Users\dharm\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\ax
isgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



Inference:The replot visualizes the relationship between "Speeding Incidents" and "Alcohol-Related Incidents" in car crashes. It appears that there is a positive correlation between these two variables, suggesting that as the number of speeding incidents increases, there is also an increase in alcohol-related incidents

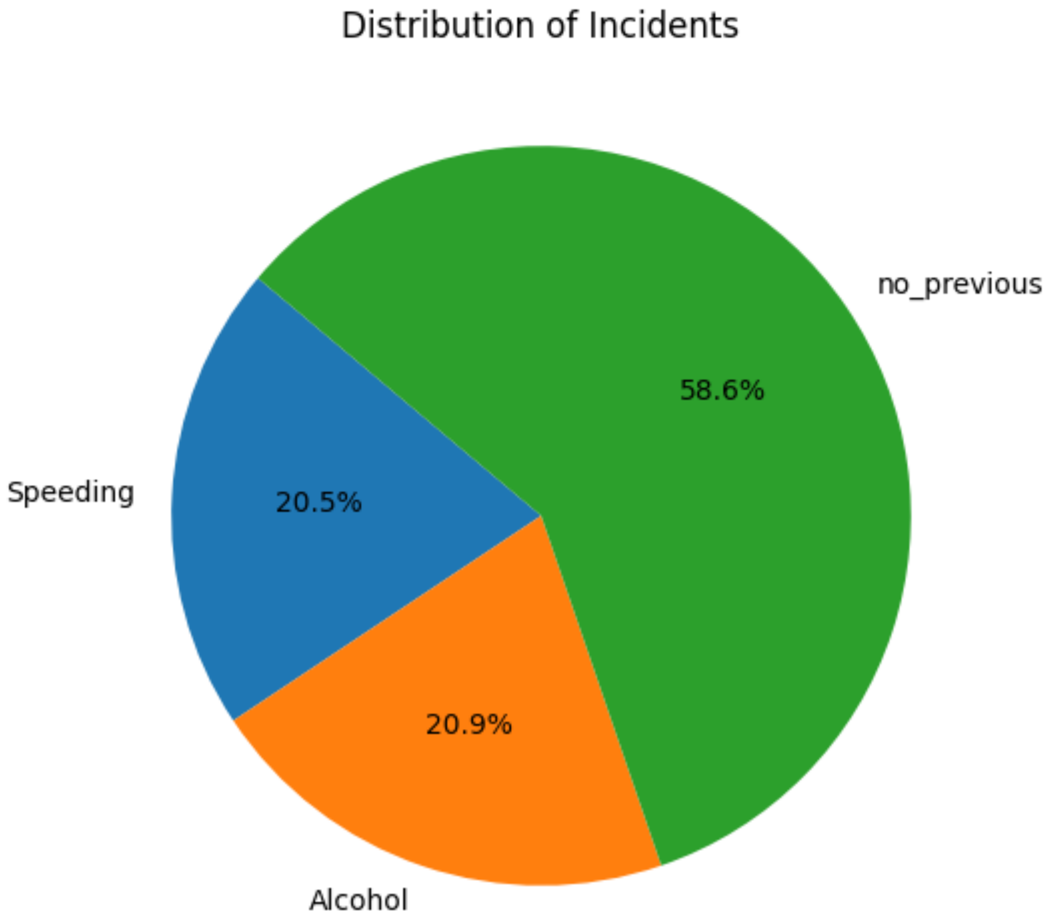
```
In [35]: #boxplot
         sns.boxplot(x="speeding", y="total", data=df)
         plt.xlabel("Speeding Involved")
         plt.ylabel("Total Crashes")
         plt.title("Impact of Speeding on Total Car Crashes")
         plt.show()
```



Inference: The box plot helps you understand the relationship between speeding involvement and total car crashes. It shows the median, quartiles, and potential outliers.

In [36]:

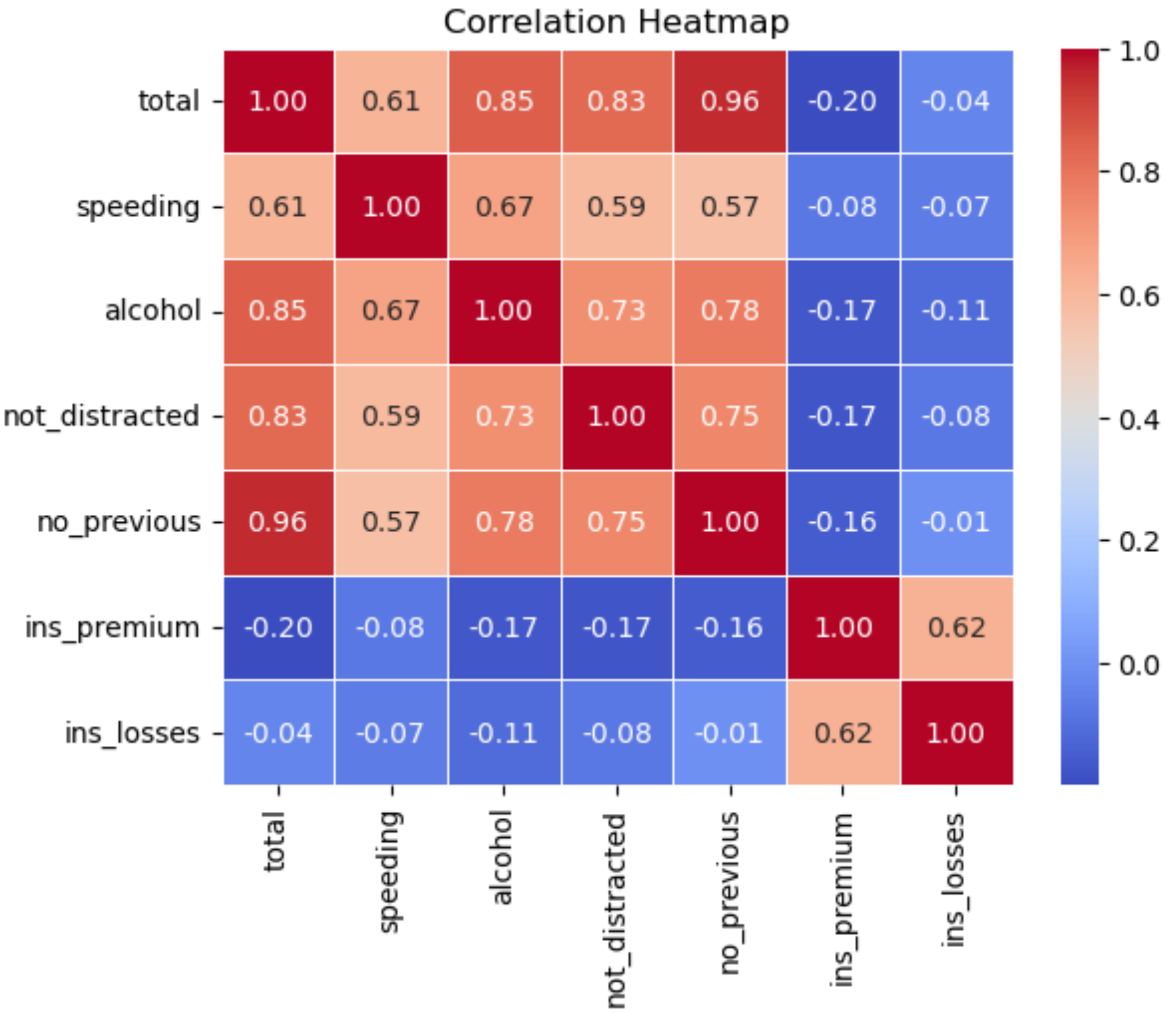
```python
alcohol_sum = df['alcohol'].sum()
speeding_sum = df['speeding'].sum()
no_previous = df['no_previous'].sum()
# Hypothetical data for pie chart
observations = [alcohol_sum, speeding_sum,no_previous]
# Hypothetical data
incident_types = ['Speeding', 'Alcohol', 'no_previous']

# Create a pie chart
plt.figure(figsize=(6, 6))
plt.pie(observations, labels=incident_types, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Incidents')
plt.show()
```



Inference:The pie chart visually summarizes the distribution of incident types within a given context. "no_previous" incidents dominate the dataset, indicating their higher prevalence compared to "Alcohol" and "Speeding" incidents. The chart simplifies the comparison of incident types, with "no_previous" clearly standing out as the most common category, making it easy to identify the primary issue in the dataset.

In [2]:
```
import seaborn as sns
import matplotlib.pyplot as plt
df = sns.load_dataset("car_crashes")
correlation_matrix=df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=
plt.title("Correlation Heatmap")
plt.show()
```

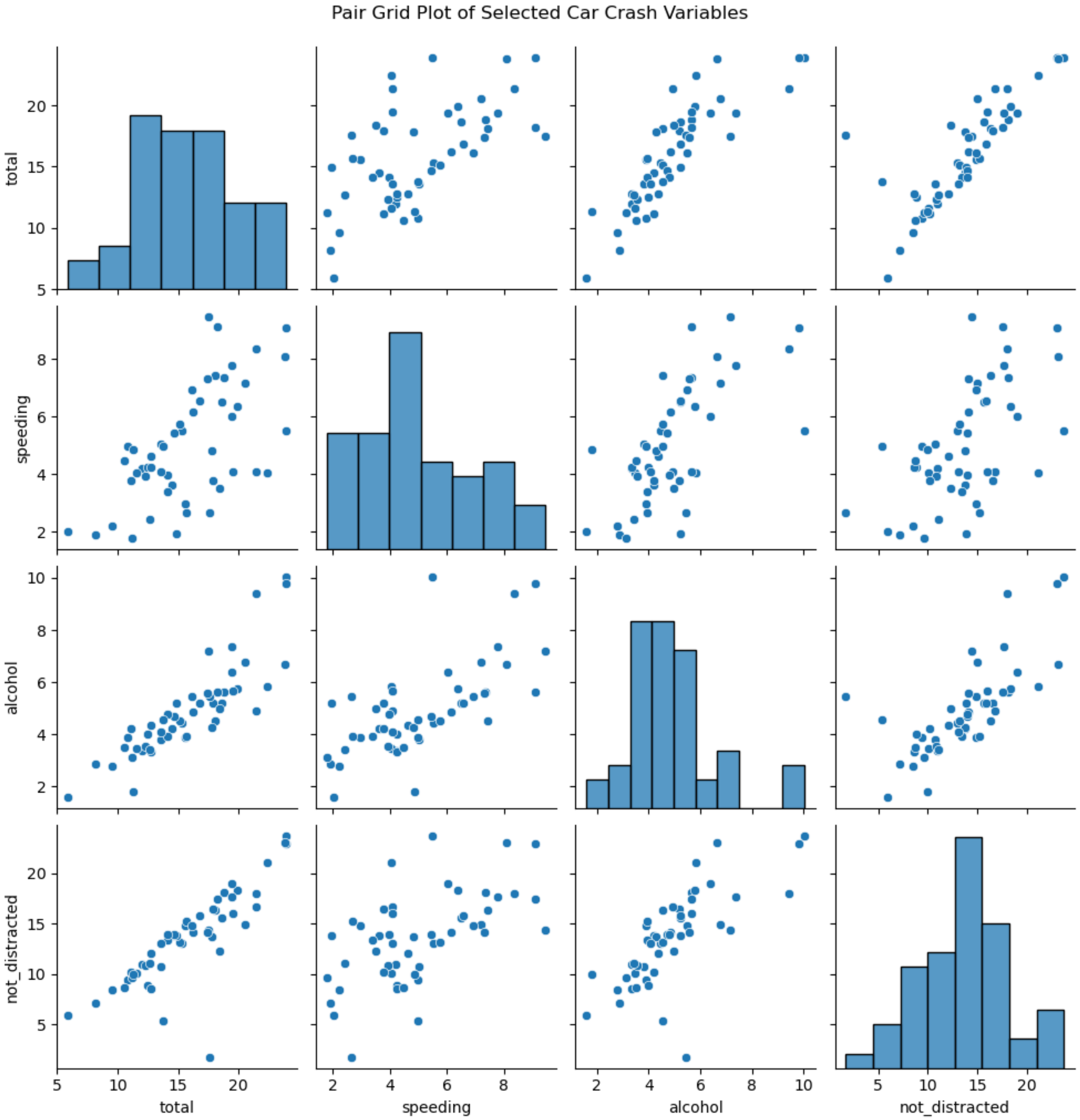
```
C:\Users\chatu\AppData\Local\Temp\ipykernel_4860\2084445547.py:4: FutureWarning: Th
e default value of numeric_only in DataFrame.corr is deprecated. In a future versio
n, it will default to False. Select only valid columns or specify the value of nume
ric_only to silence this warning.
  correlation_matrix=df.corr()
```



	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
total	1.00	0.61	0.85	0.83	0.96	-0.20	-0.04
speeding	0.61	1.00	0.67	0.59	0.57	-0.08	-0.07
alcohol	0.85	0.67	1.00	0.73	0.78	-0.17	-0.11
not_distracted	0.83	0.59	0.73	1.00	0.75	-0.17	-0.08
no_previous	0.96	0.57	0.78	0.75	1.00	-0.16	-0.01
ins_premium	-0.20	-0.08	-0.17	-0.17	-0.16	1.00	0.62
ins_losses	-0.04	-0.07	-0.11	-0.08	-0.01	0.62	1.00

Inference: Correlation heatmap provides valuable insights into the relationships between numerical variables in the dataset.

In [3]:
```
selected_columns = ['total', 'speeding', 'alcohol', 'not_distracted']
pair_grid = sns.pairplot(df[selected_columns])
plt.suptitle('Pair Grid Plot of Selected Car Crash Variables', y=1.02)
plt.show()
```



Inference:The pair plot provides scatterplots for all numeric variabels and relationship between them.