Data Visuvalization on the car_crashes dataset

In [1]:
```
import seaborn as sns
```

Importing the required libraries for the visualization of the car_crashes dataset

In [2]:
```
data=sns.load_dataset("car_crashes")
```

loading the car_crashes dataset

In [3]:
```
data.shape
```

Out[3]:
```
(51, 8)
```

printing the shape of the dataset

In [4]:
```
data
```

Out[4]:

	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
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

In [10]:
```
data.corr()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_7868\2627137660.py:1: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a futur
e version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  data.corr()
```

Out[10]:

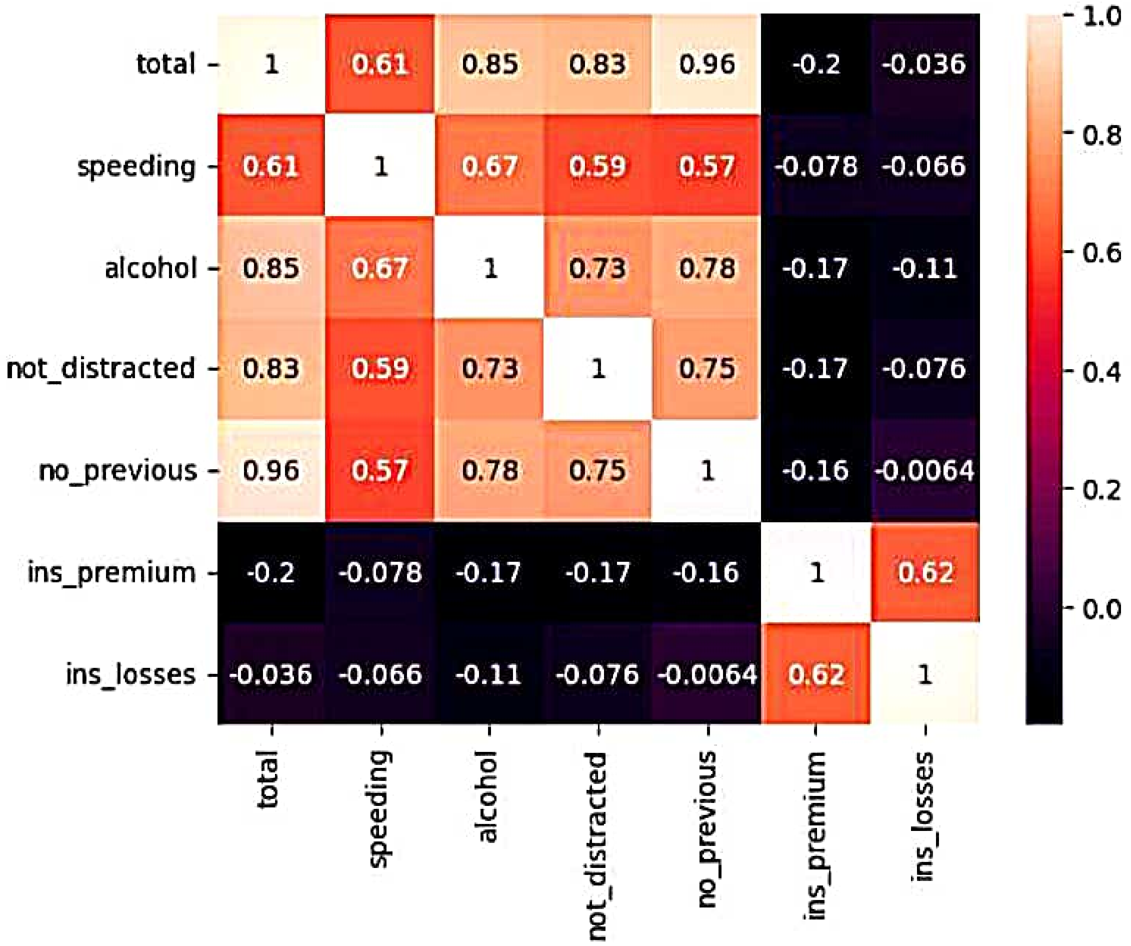
	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_lo
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.199702	-0.03
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.077675	-0.06
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.170612	-0.11
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	-0.174856	-0.07
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	-0.156895	-0.00
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856	-0.156895	1.000000	0.62
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970	-0.006359	0.623116	1.00

displayin the correlation matrix

In [11]:
```
sns.heatmap(data.corr(),annot=True)
```

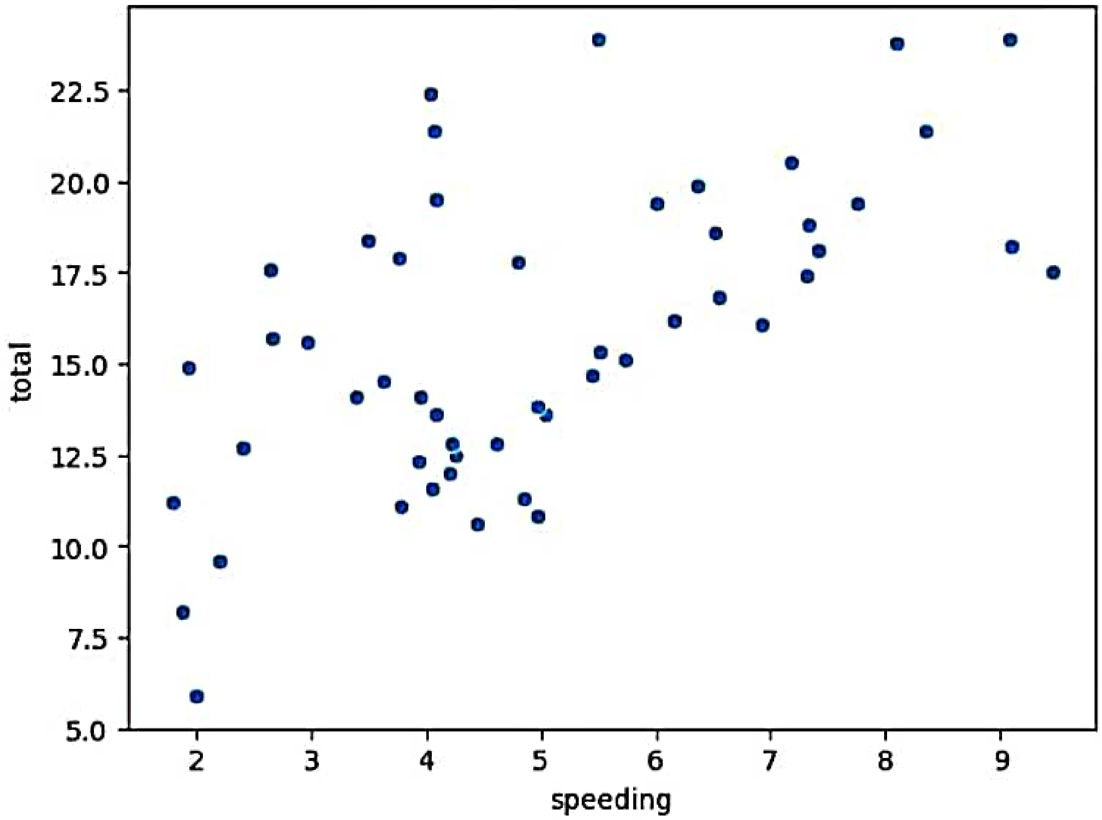
```
C:\Users\HP\AppData\Local\Temp\ipykernel_7868\2578434383.py:1: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a futur
e version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  sns.heatmap(data.corr(),annot=True)
```

Out[11]: <Axes: >



displaying the heatmap of the correlation matrix

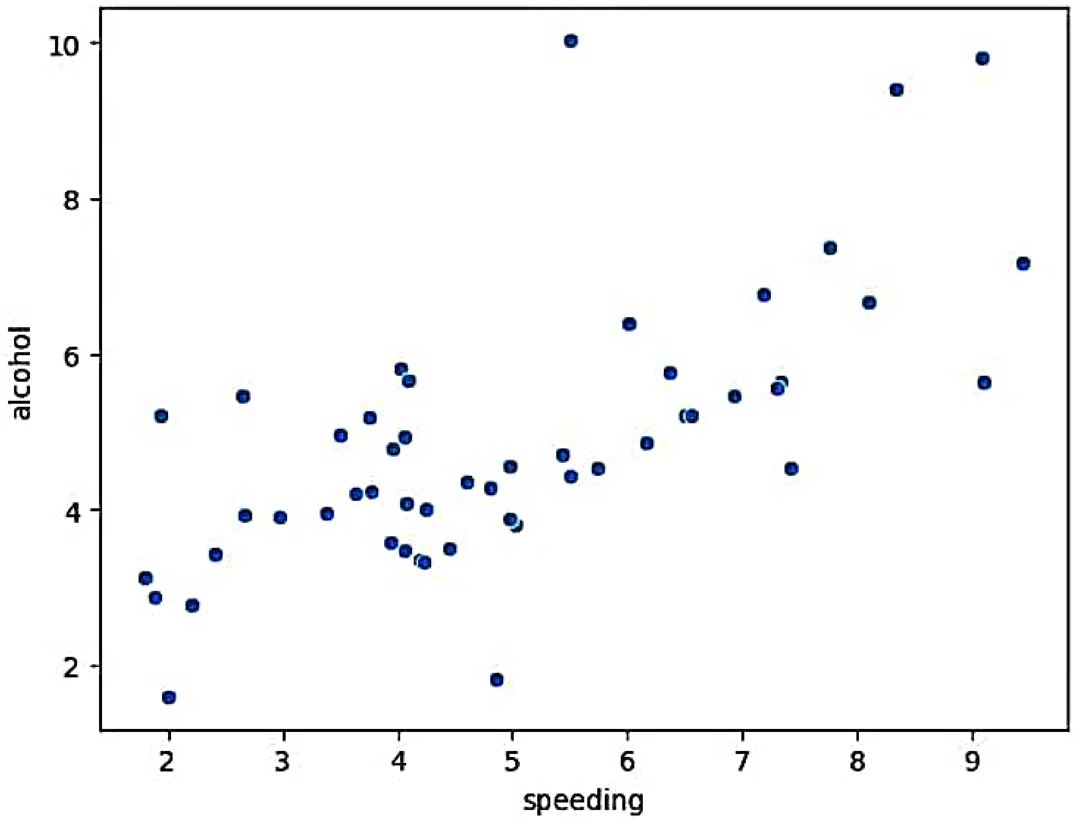
```
In [12]: sns.scatterplot(x="speeding",y="total",data=data)

Out[12]: <Axes: xlabel='speeding', ylabel='total'>
```



Bivariate analysis: From the scatterplot of total vs speeding , we can say that the total value is directly proportional to the speeding . total and speeding are strongly correlated.

In [13]:
```
sns.scatterplot(x="speeding",y="alcohol",data=data)
```

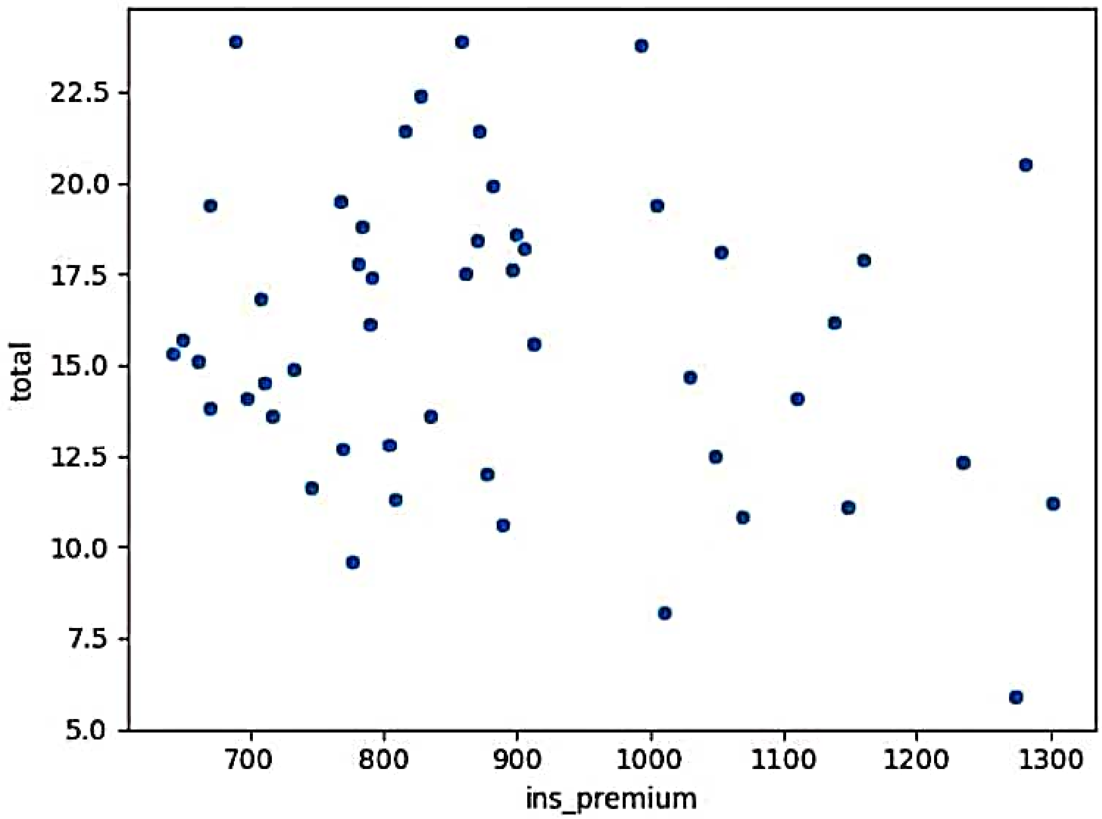
Out[13]: <Axes: xlabel='speeding', ylabel='alcohol'>



Bivariate analysis: From the scatterplot of alcohol vs speeding , we can say that the alcohol value is directly proportional to the speeding . alcohol and speeding are strongly correlated.

```
In [14]: sns.scatterplot(x="ins_premium",y="total",data=data)
```

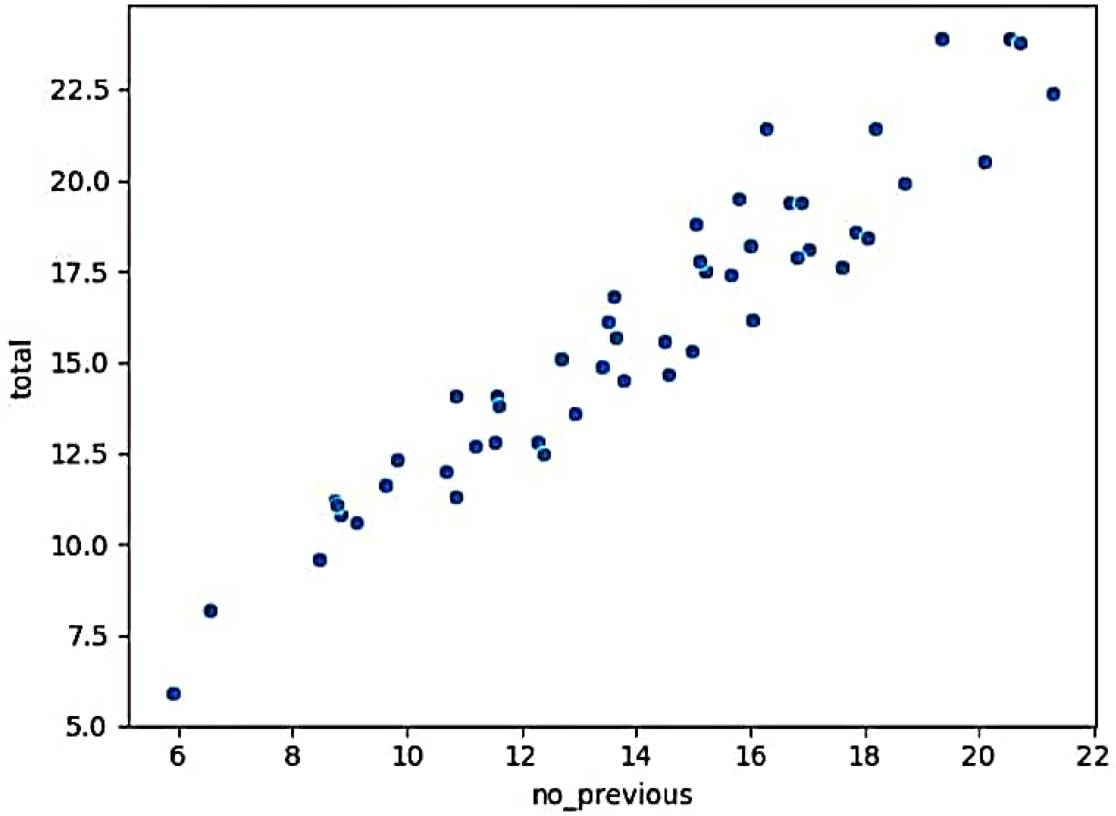
```
Out[14]: <Axes: xlabel='ins_premium', ylabel='total'>
```



Bivariate analysis: From the scatterplot of total vs ins_premium , we can say that the total value is inversly proportional to the ins_premium . total and ins_premium are weakly corrr=elated.

In [15]:
```
sns.scatterplot(x="no_previous",y="total",data=data)
```

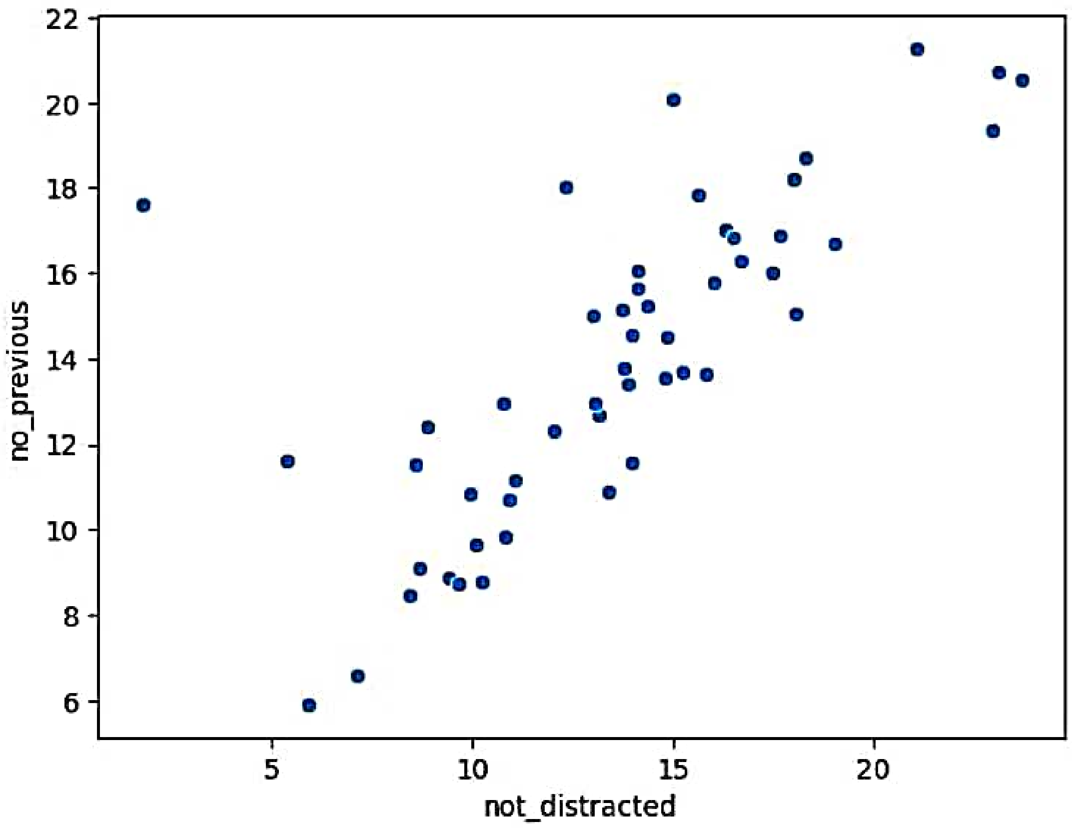
Out[15]: <Axes: xlabel='no_previous', ylabel='total'>



Bivariate analysis: From the scatterplot of total vs no_previous , we can say that the total value is directly proportional to the no_previous . total and no_previous are strongly correlated.

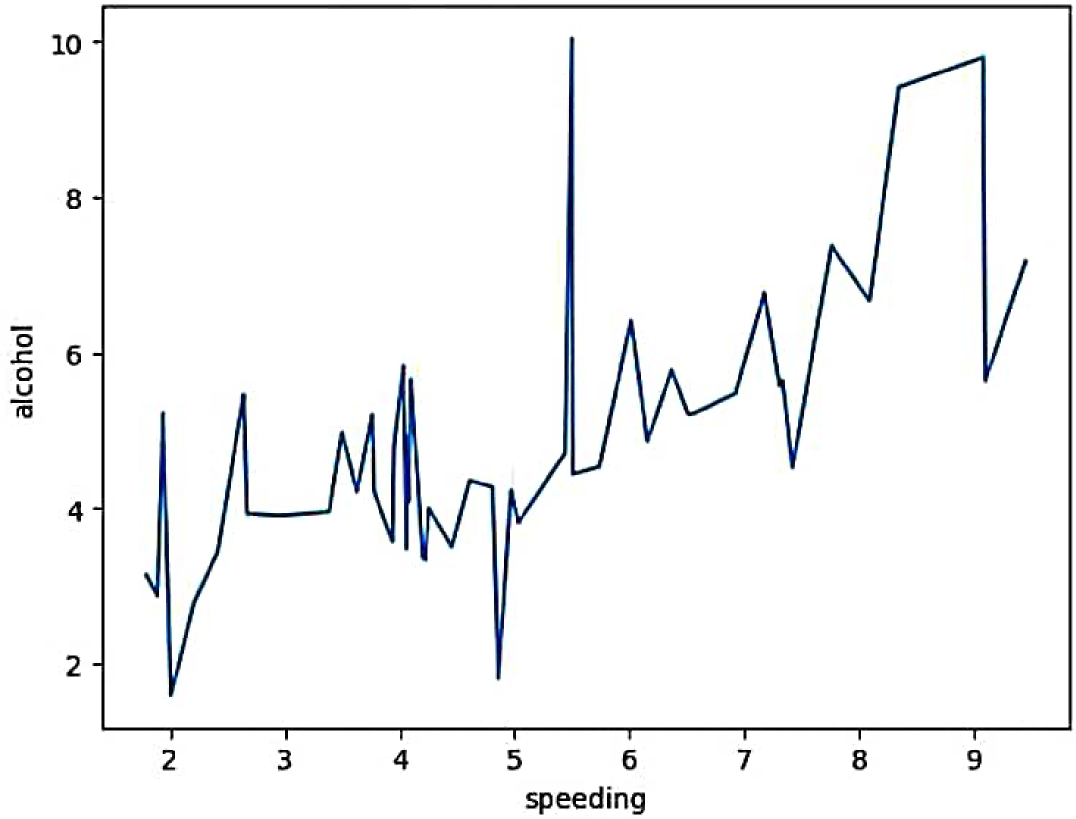
In [16]:
```
sns.scatterplot(x="not_distracted",y="no_previous",data=data)
```

Out[16]: <Axes: xlabel='not_distracted', ylabel='no_previous'>



Bivariate analysis: From the scatterplot of total vs not_distracted. , we can say that the total value is directly proportional to the not_distracted . total and not_distracted are strongly correlated.

```
In [17]: sns.lineplot(x="speeding",y="alcohol",data=data)

Out[17]: <Axes: xlabel='speeding', ylabel='alcohol'>
```

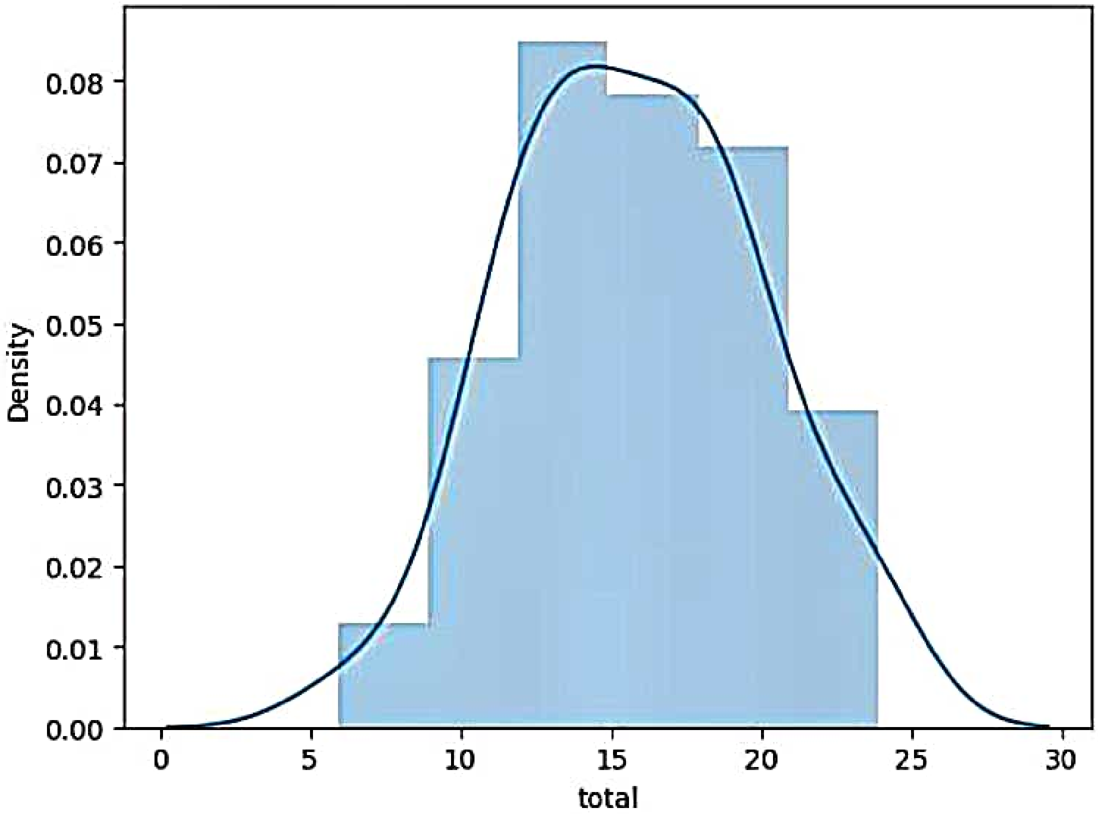


form the lineplot the we can observe the change in alcohol with respect to the speeding value.

In [18]:
```
sns.distplot(data.total)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_7868\2102236082.py:1: 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 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(data.total)
```

Out[18]: <Axes: xlabel='total', ylabel='Density'>

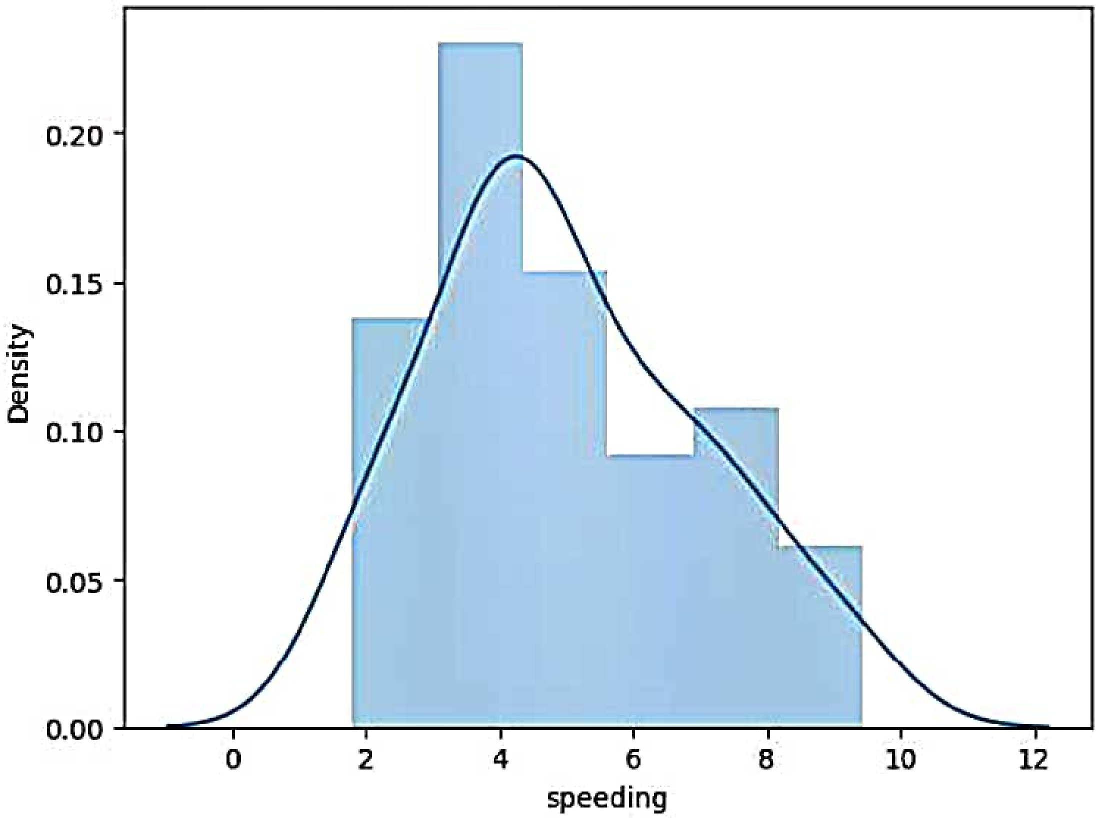


Univariate analysis : from the distplot we can display the distribution of the total values in the given car_crashes dataset.

In [19]:
```
sns.distplot(data.speeding)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_7868\2228035885.py:1: 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 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(data.speeding)
```

Out[19]: `<Axes: xlabel='speeding', ylabel='Density'>`



Univariate analysis : from the distplot we can display the distribution of the speeding values in the given car_crashes dataset.

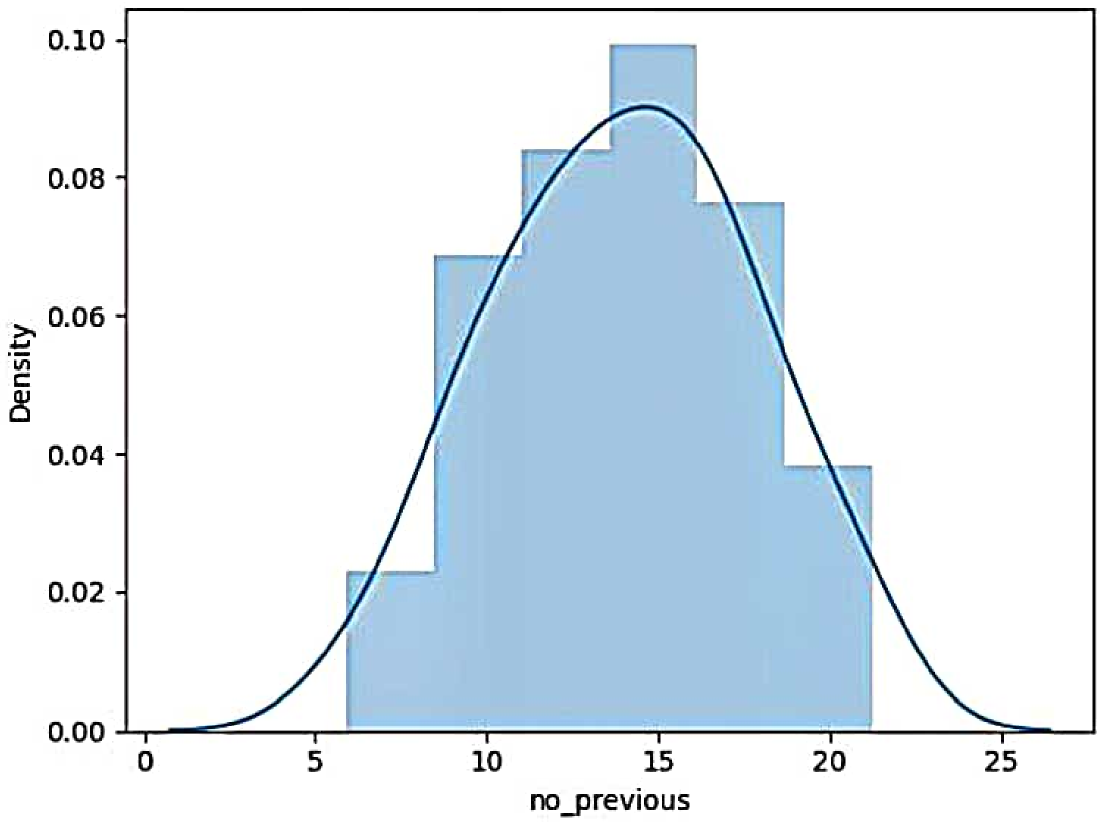
```
sns.distplot(data.alcohol)
```

Out[20]: <Axes: xlabel='alcohol', ylabel='Density'>

In [22]:
```
sns.distplot(data.no_previous)
```

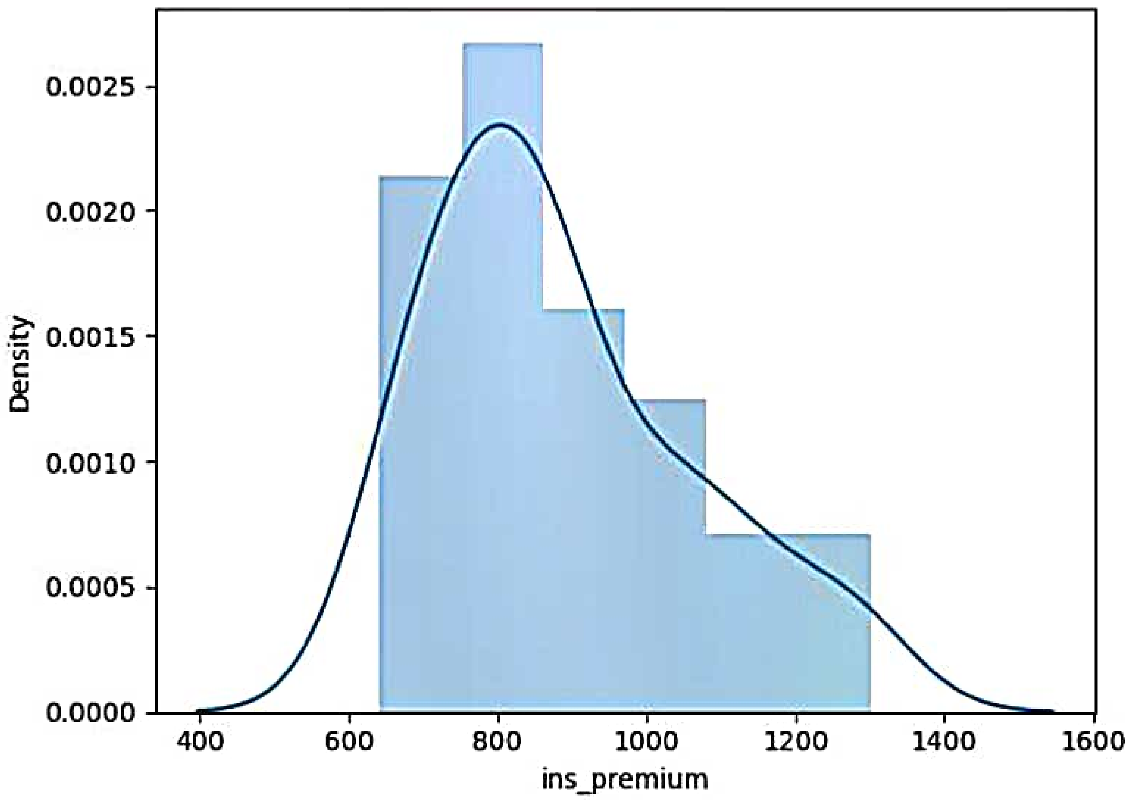
```
C:\Users\HP\AppData\Local\Temp\ipykernel_7868\1836816757.py:1: UserWarning:

`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 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(data.no_previous)
```

Out[22]: <Axes: xlabel='no_previous', ylabel='Density'>



Univariate analysis : from the distplot we can display the distribution of the no_previous values in the given car_crashes dataset.

```
In [23]: sns.distplot(data.ins_premium)

C:\Users\HP\AppData\Local\Temp\ipykernel_7868\2655982644.py:1: 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 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(data.ins_premium)

Out[23]: <Axes: xlabel='ins_premium', ylabel='Density'>
```

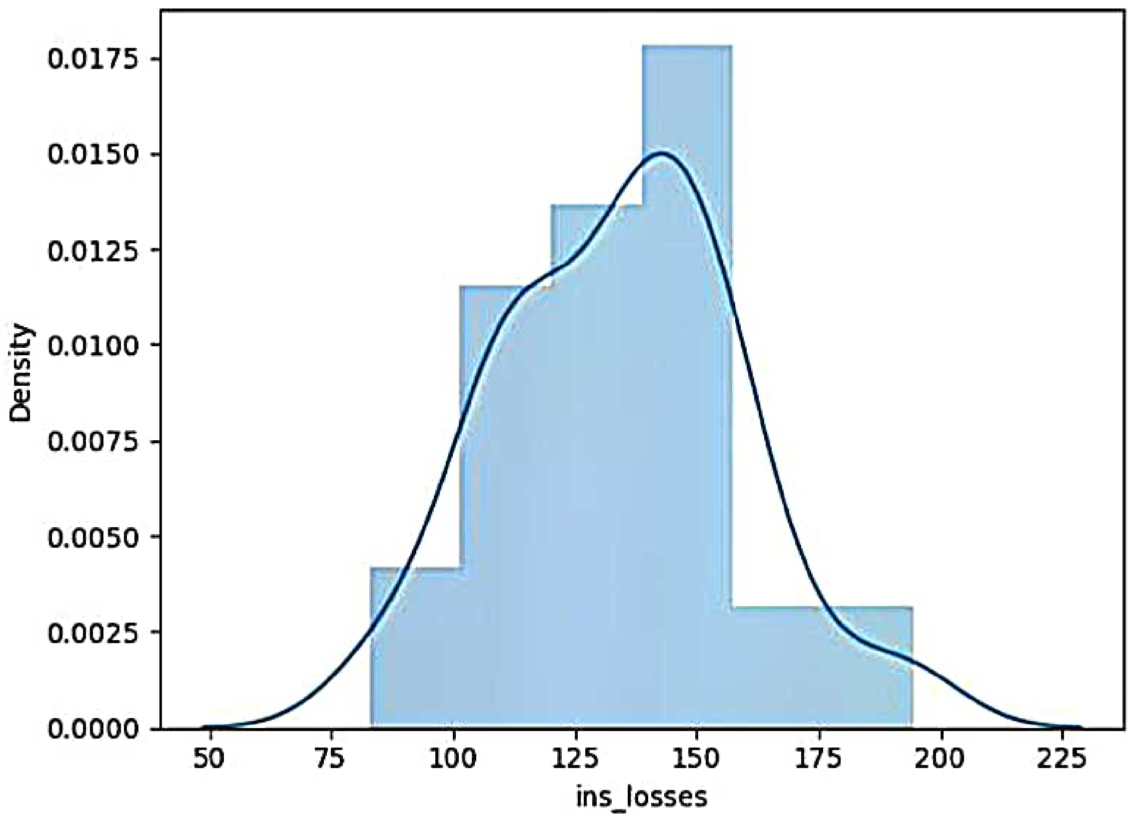


Univariate analysis : from the distplot we can display the distribution of the ins_premium values in the given car_crashes dataset.

In [24]:
```
sns.distplot(data.ins_losses)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_7868\1689502277.py:1: 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 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(data.ins_losses)
```

Out[24]:
```
<Axes: xlabel='ins_losses', ylabel='Density'>
```



Univariate analysis : from the distplot we can display the distribution of the ins_losses values in the given car_crashes dataset.

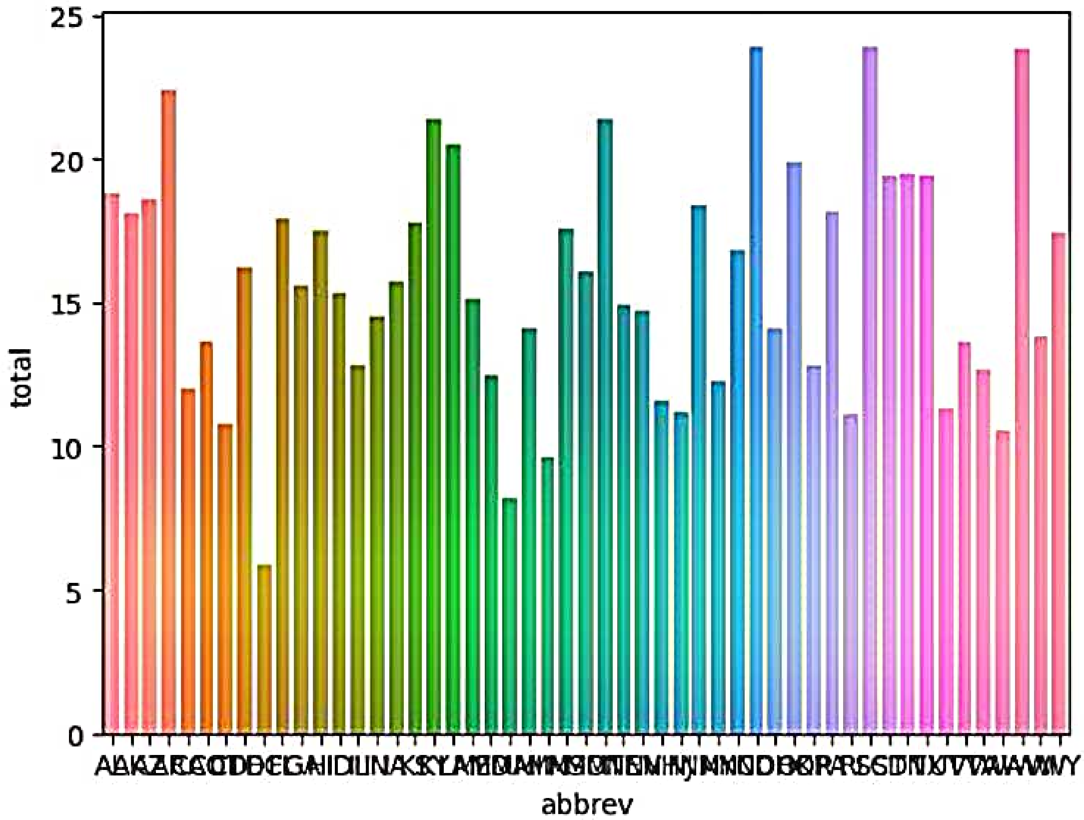
```
In [25]: data.abbrev.value_counts()
```

```
Out[25]: AL    1
PA    1
NV    1
NH    1
NJ    1
NM    1
NY    1
NC    1
ND    1
OH    1
OK    1
OR    1
RI    1
MT    1
SC    1
SD    1
TN    1
TX    1
UT    1
VT    1
VA    1
WA    1
WV    1
WI    1
NE    1
MO    1
AK    1
ID    1
AZ    1
AR    1
CA    1
CO    1
CT    1
DE    1
DC    1
FL    1
GA    1
HI    1
IL    1
MS    1
IN    1
IA    1
KS    1
KY    1
LA    1
ME    1
MD    1
MA    1
MI    1
MN    1
WY    1
Name: abbrev, dtype: int64
```

finding the different values and there respective counts in the abbrev attribute in the car_crashes dataset.

In [26]:
```
sns.barplot(data=data,x="abbrev",y="total")
```

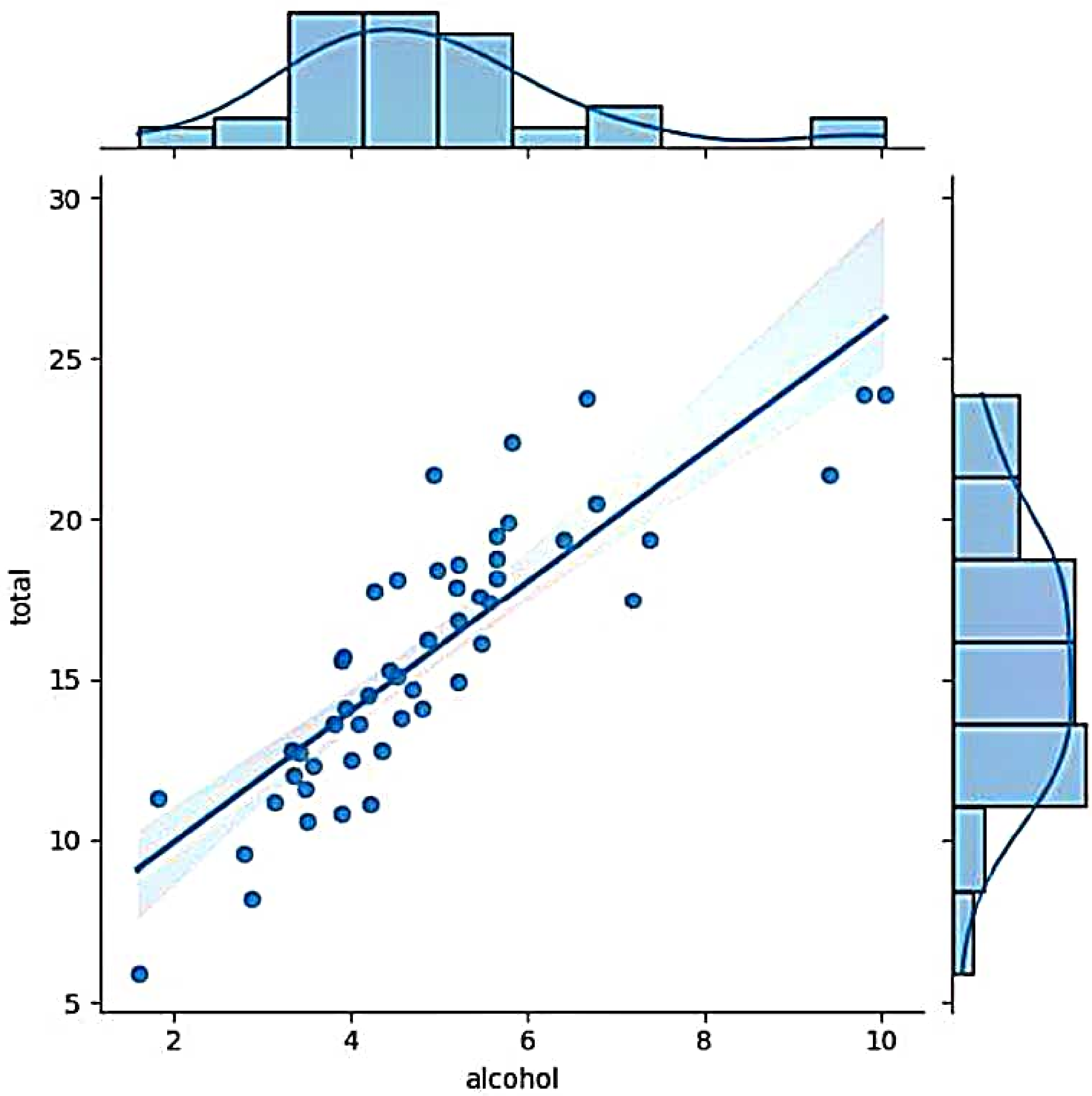
Out[26]: `<Axes: xlabel='abbrev', ylabel='total'>`



form the above graph and value_counts() we can say that there no value repeated, the bar plot was drawn between the total vs abbrev .

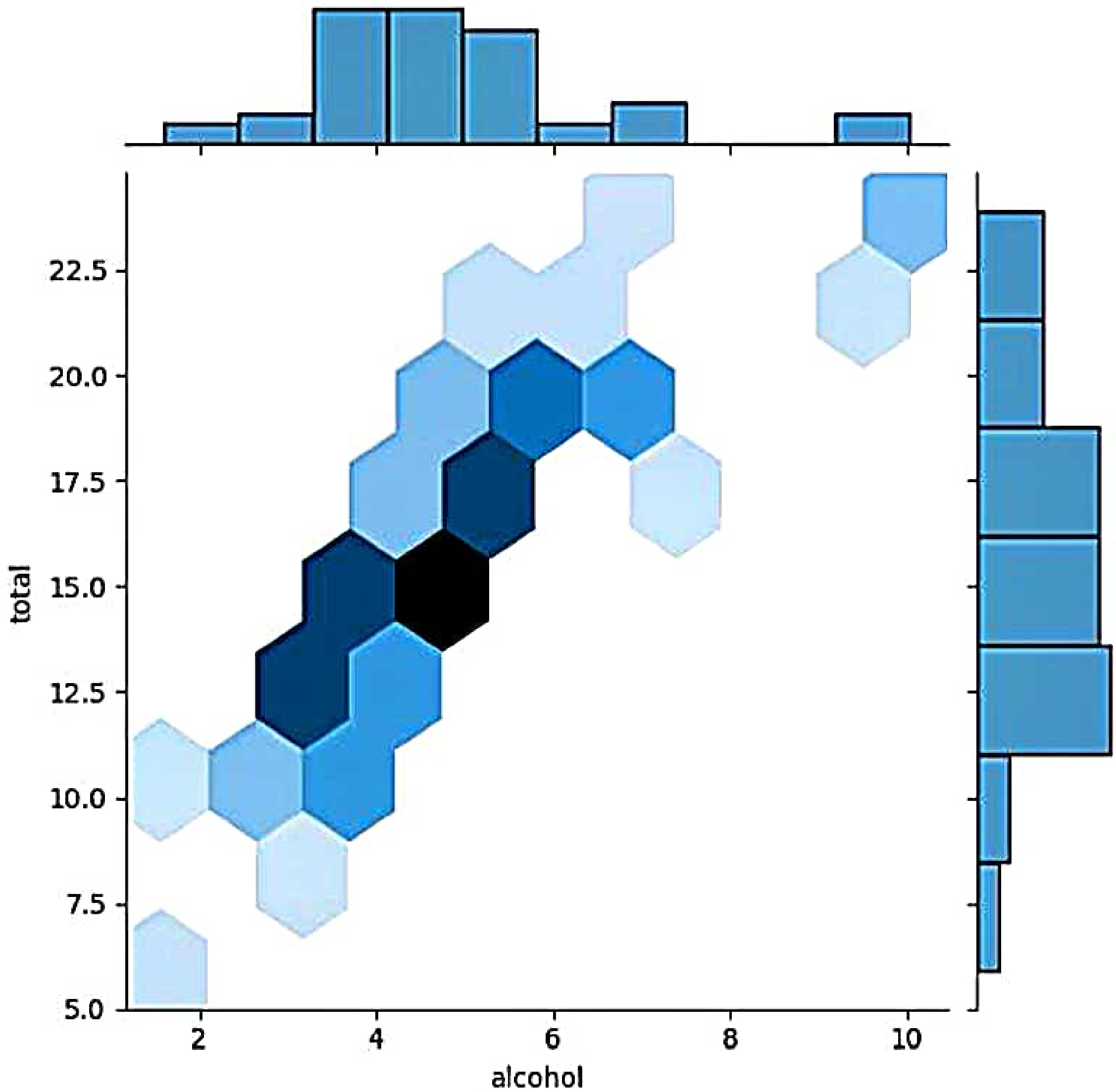
In [27]:
```
sns.jointplot(x="alcohol",y="total",data=data,kind='reg')
```

Out[27]: <seaborn.axisgrid.JointGrid at 0x267ee651cd0>



Jointplot between the total vs alcohol, we can find the distributions of the both attributes and that they are strongly correlated with each other.

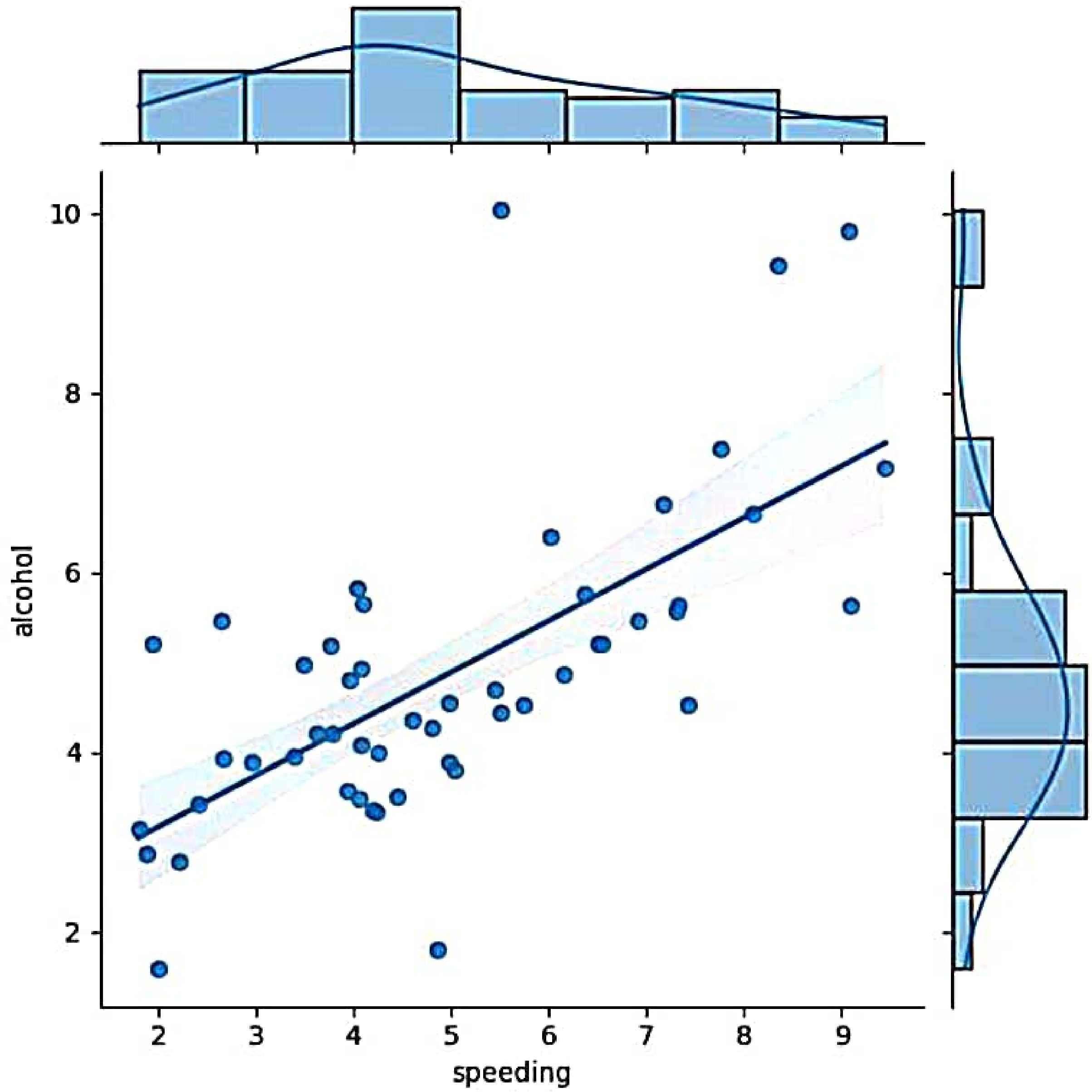
```
In [28]: sns.jointplot(x="alcohol",y="total",data=data,kind='hex')

Out[28]: <seaborn.axisgrid.JointGrid at 0x267ee8613d0>
```



Jointplot between the total vs alcohol, we can find the distributions of the both attributes . More number of cars are concentated at darkly shaded region of the hexagon

In [29]:
```
sns.jointplot(x="speeding",y="alcohol",data=data,kind='reg')
```

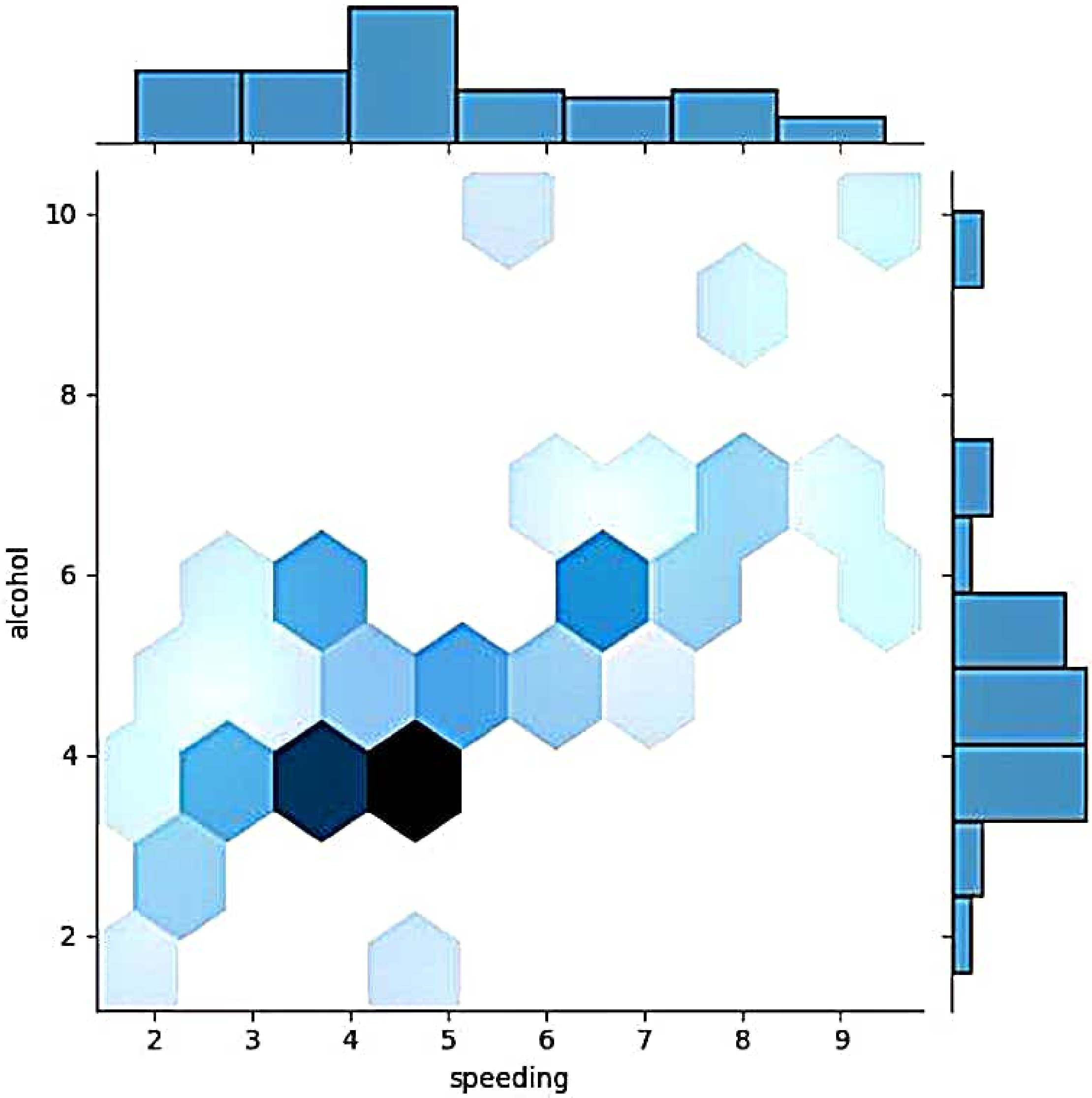
Out[29]: <seaborn.axisgrid.JointGrid at 0x267eed7ad90>



Jointplot between the alcohol vs speeding, we can find the distributions of the both attributes and that they are strongly correlated with each other.

In [30]:
```
sns.jointplot(x="speeding",y="alcohol",data=data,kind='hex')
```

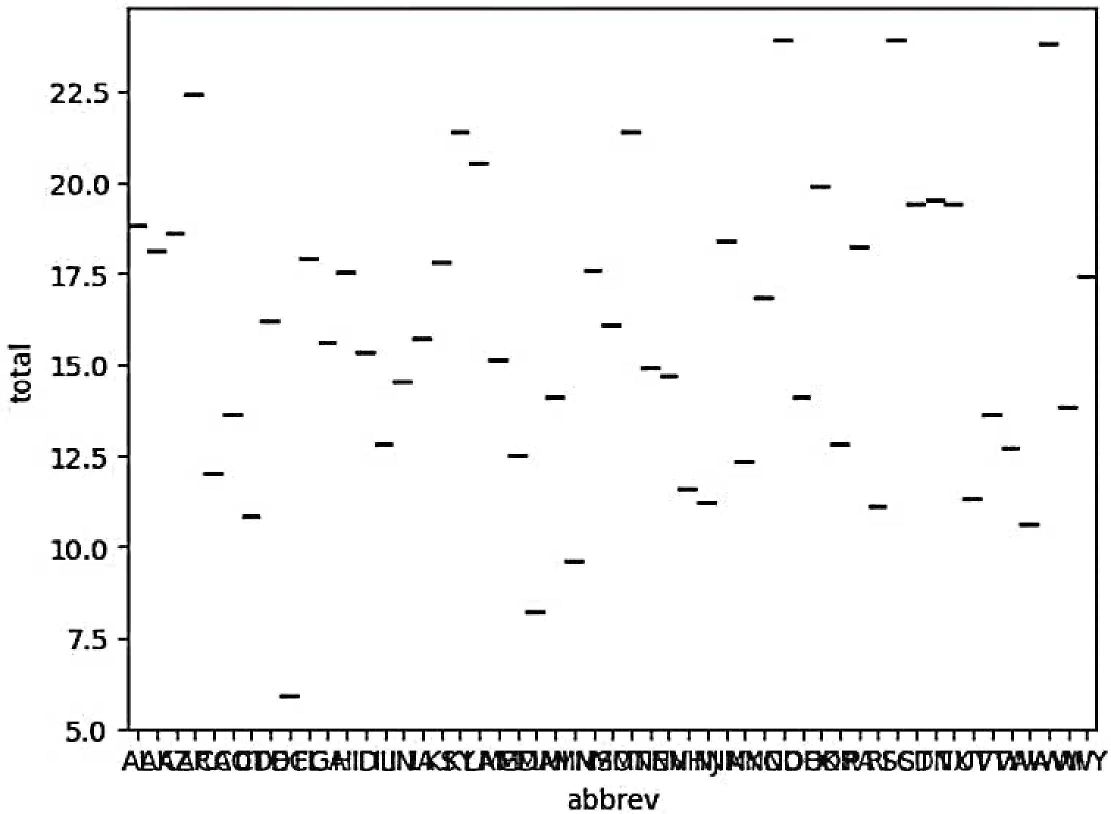
Out[30]: <seaborn.axisgrid.JointGrid at 0x267ef152810>



Jointplot between the alcohol vs speeding, we can find the distributions of the both attributes . More number of cars are concentated at darkly shaded region of the hexagon

```
In [31]: sns.boxplot(x="abbrev",y="total",data=data)
```

Out[31]: <Axes: xlabel='abbrev', ylabel='total'>



univariate analysis : we can observe the boxplot between the numerical value total and categorical value abbrev from the above plot. By finding the boxplot we can find the intervals in which the data had distributed and if there are any outliers in the data.