Importing seaborn libraray

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

Loading the dataset

In [2]:
```
df=sns.load_dataset('car_crashes')
```

In [3]:
```
df
```

Out[3]:

	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

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
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 [4]:
```
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 [5]:
```
df.head()
```

Out[5]:

	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 [6]:
```
df.tail()
```

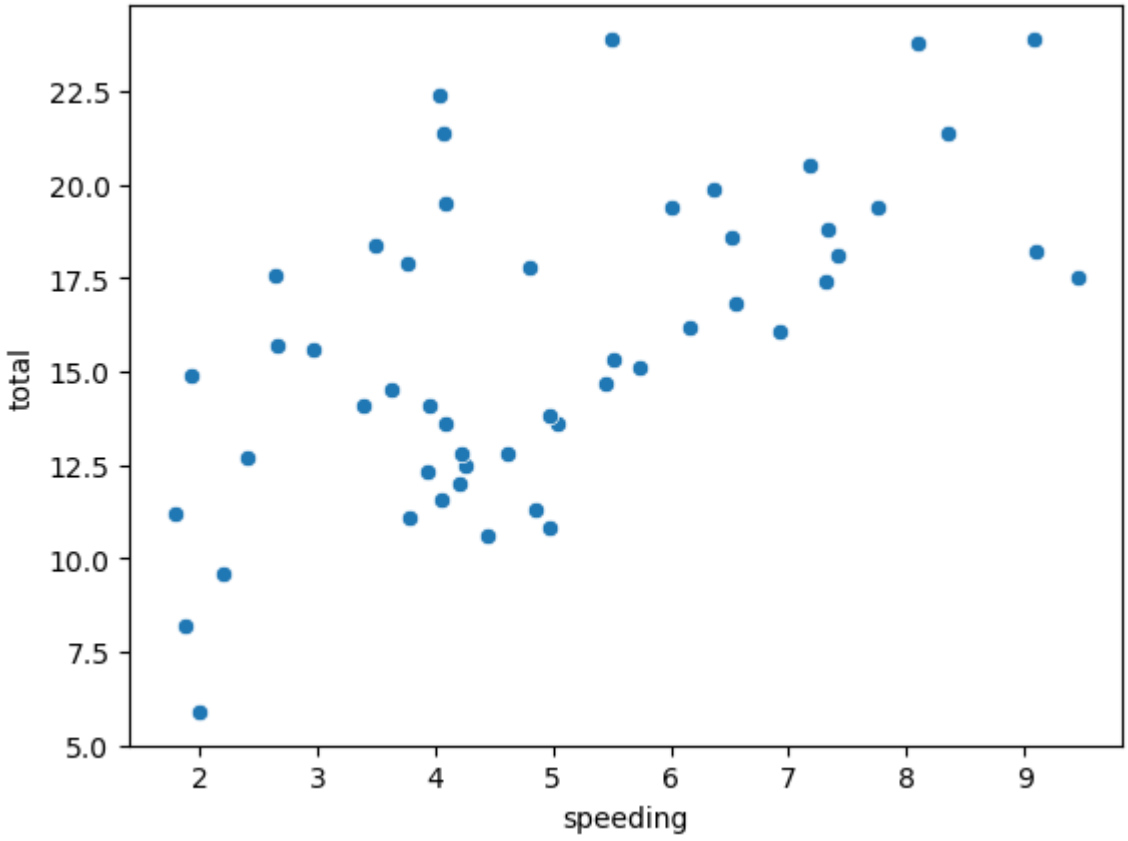
Out[6]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
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

Scatterplot

In [7]:
```
sns.scatterplot(x="speeding",y="total",data=df)
```

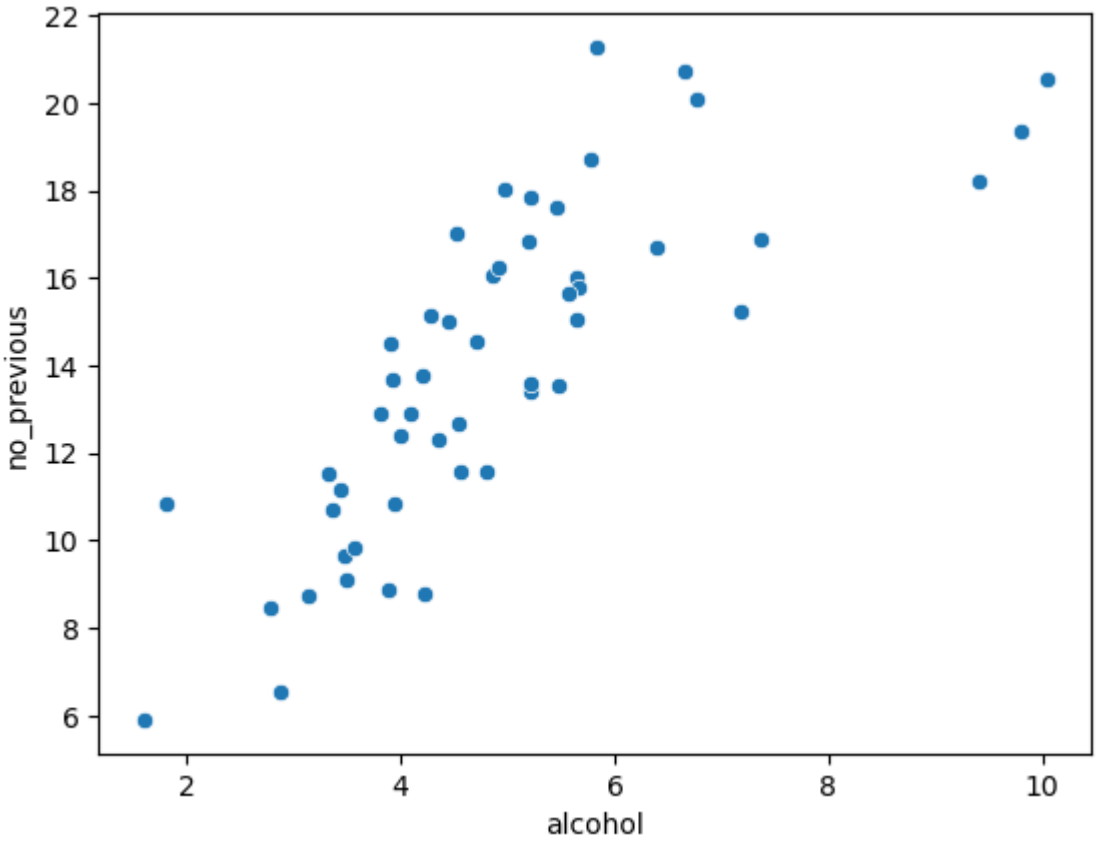
Out[7]:
```
<AxesSubplot:xlabel='speeding', ylabel='total'>
```




```
Inference : From the plot we can say that speeding can't show a linear trend.
```

In [8]:
```
sns.scatterplot(x="alcohol",y="no_previous",data=df)
```

Out[8]:
```
<AxesSubplot:xlabel='alcohol', ylabel='no_previous'>
```

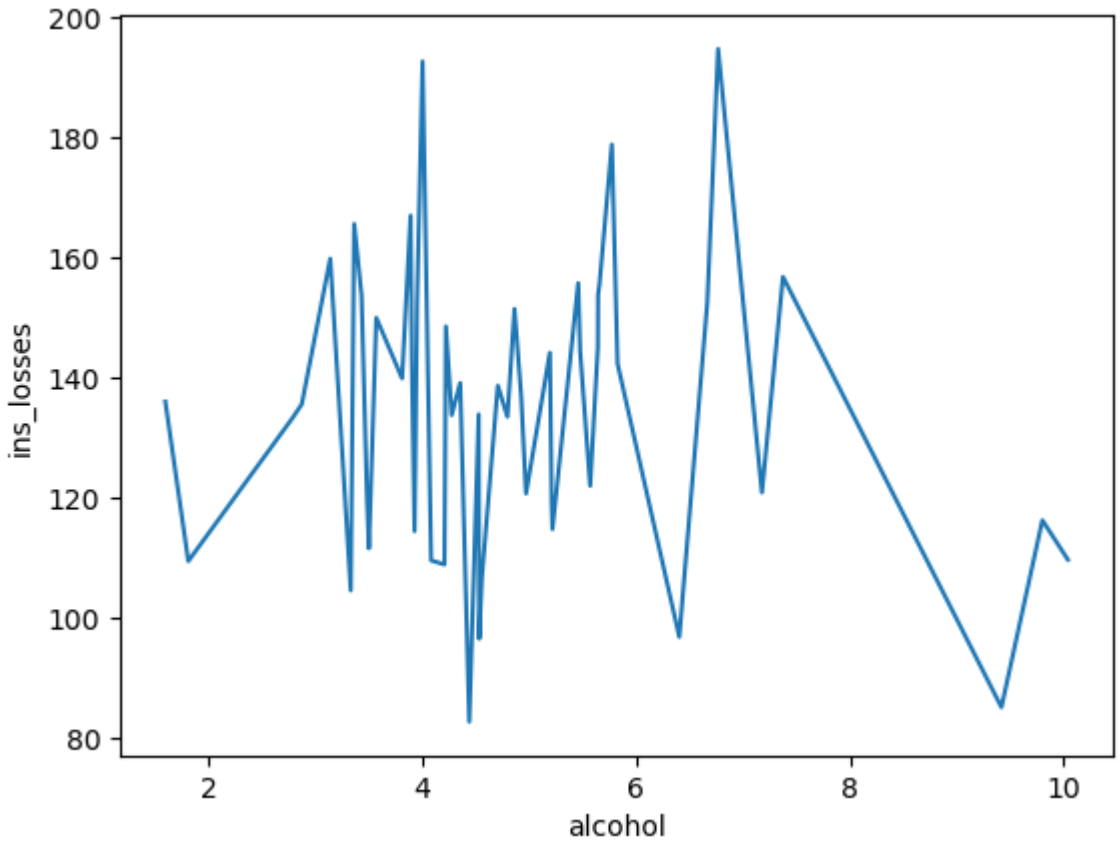



```
Inference : From the plot we can say that as alcohol increases no_previous
also increases
```

Lineplot

In [9]:
```
sns.lineplot(x="alcohol",y="ins_losses",data=df,ci=None)
```

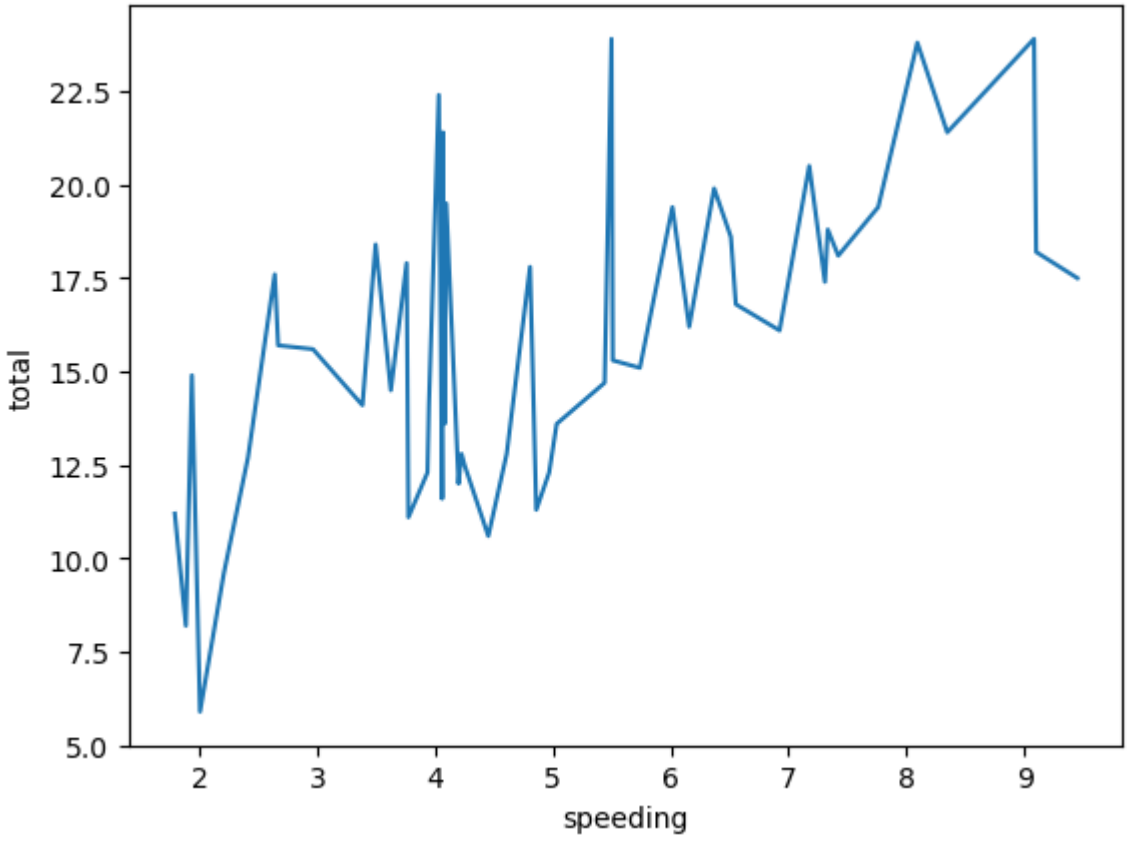
Out[9]:
```
<AxesSubplot:xlabel='alcohol', ylabel='ins_losses'>
```




```
Inference : From the plot we can say that there no straight line between
alcohol and ins_losses.
```

In [10]:
```
sns.lineplot(x="speeding",y="total",data=df,ci=None)
```

Out[10]:
```
<AxesSubplot:xlabel='speeding', ylabel='total'>
```



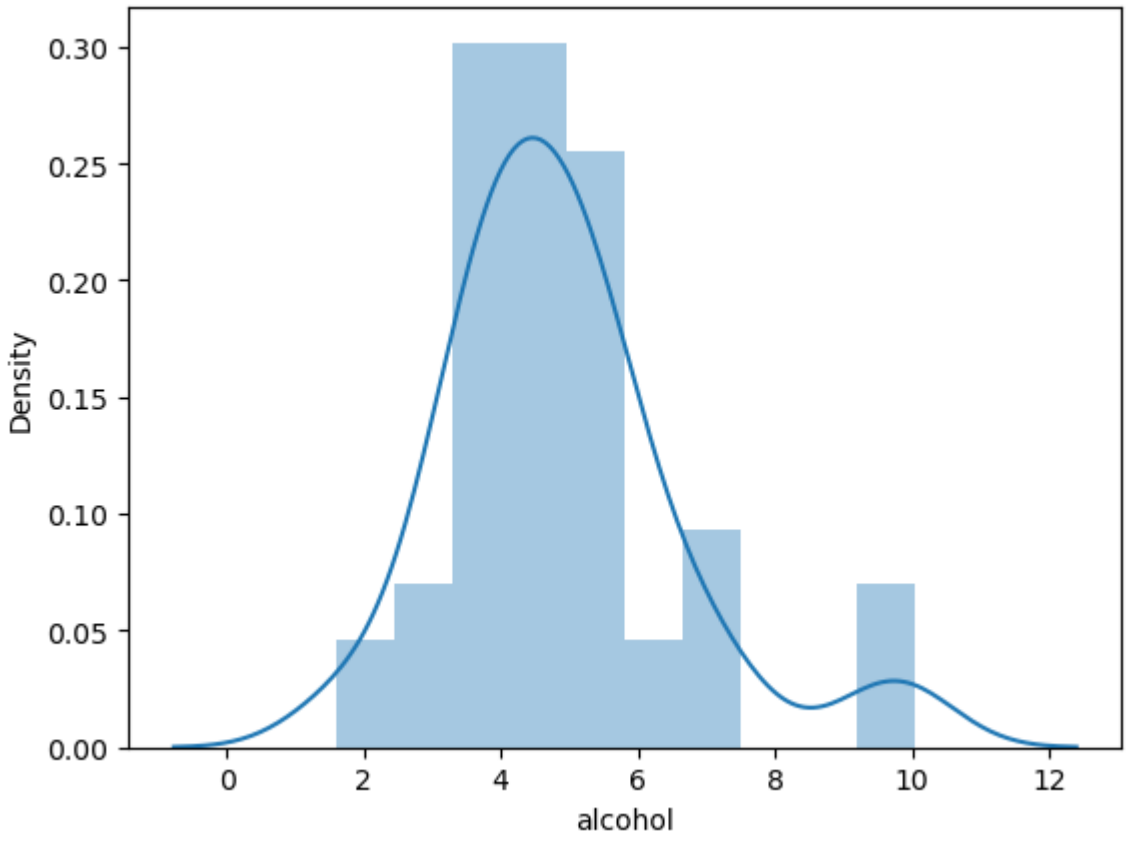

```
Inference : speeding doesn't exhibit a linear relationship with total.
```

Distributionplot

In [11]:
```
sns.distplot(df["alcohol"])
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

Out[11]: <AxesSubplot:xlabel='alcohol', ylabel='Density'>

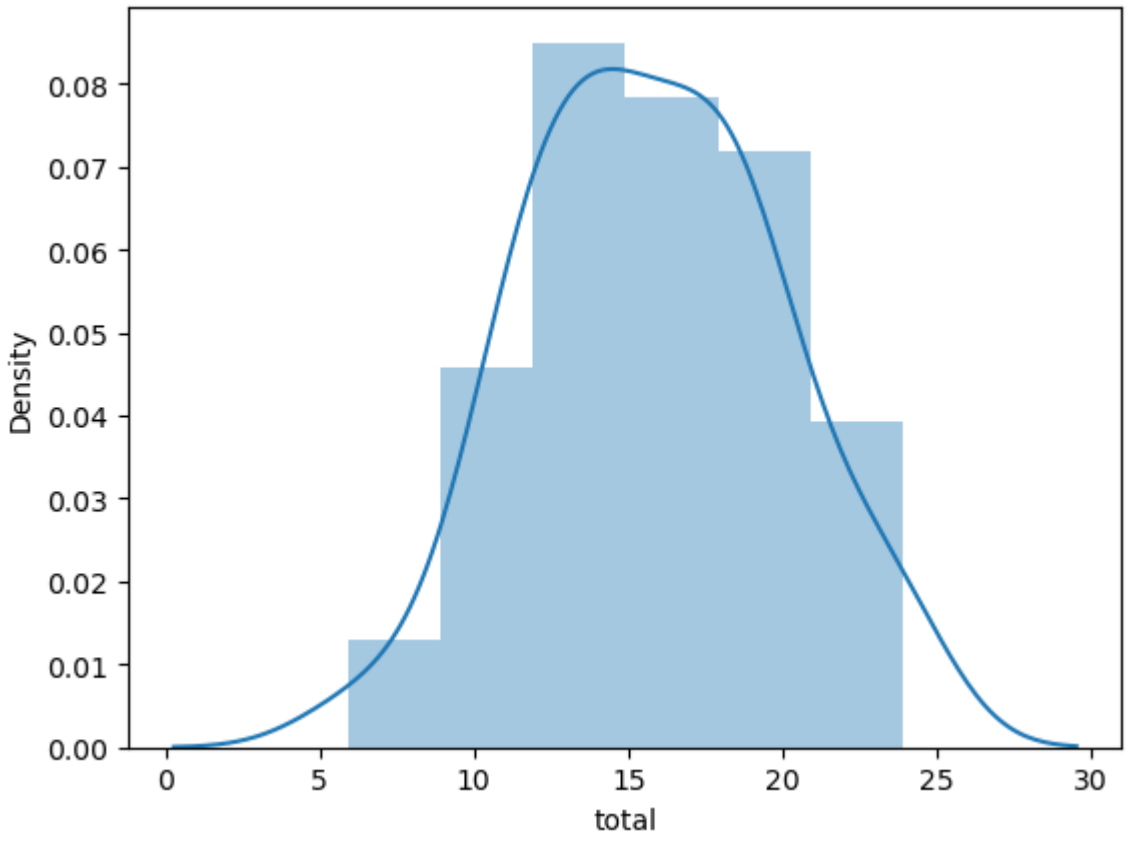



```
Inference : The distribution of alcohol is right-skewed.
```

In [12]:
```
sns.distplot(df["total"])
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

Out[12]: <AxesSubplot:xlabel='total', ylabel='Density'>

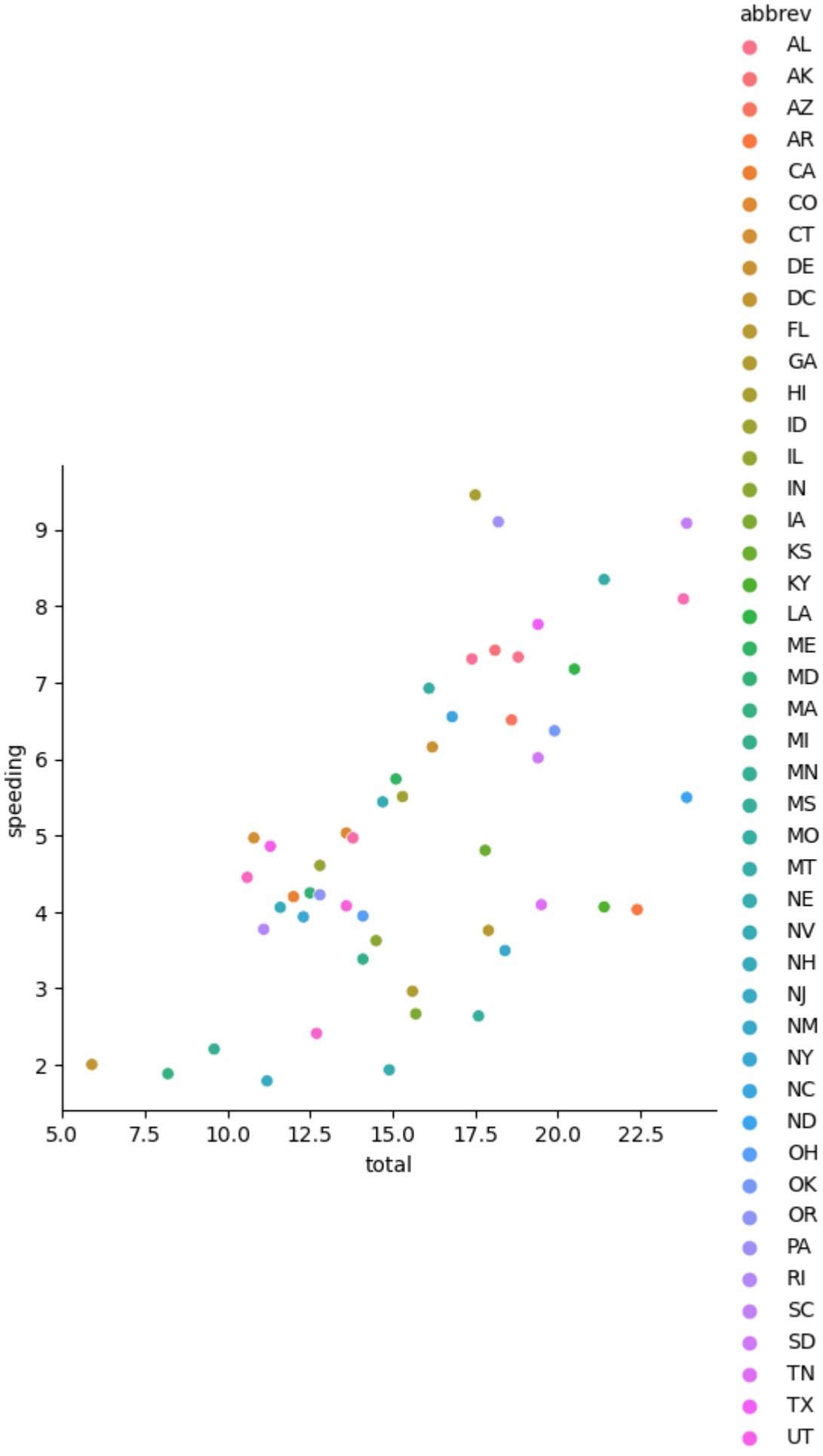



```
Inference : The total distribution is also right-skewed as well.
```

Relationplot

In [13]:
```
sns.relplot(x="total",y="speeding",data=df,hue="abbrev")
```

Out[13]:
```
<seaborn.axisgrid.FacetGrid at 0x26e3f48f670>
```


abbrev
AL
AK
AZ
AR
CA
CO
CT
DE
DC
FL
GA
HI
ID
IL
IN
IA
KS
KY
LA
ME
MD
MA
MI
MN
MS
MO
MT
NE
NV
NH
NJ
NM
NY
NC
ND
OH
OK
OR
PA
RI
SC
SD
TN
TX
UT
speeding
9
8
7
6
5
4
3
2
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
total

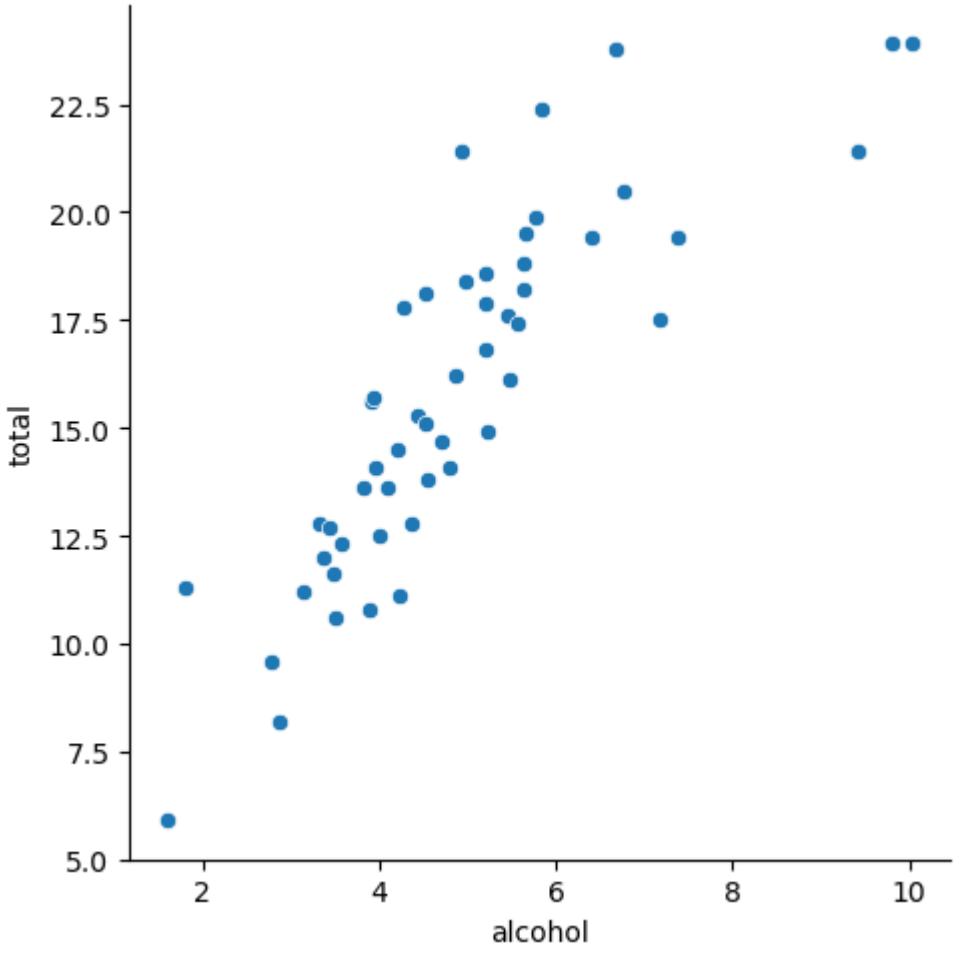

```
Inference : We get the relation plot between total and speeding and we have
different types of abbrev.
```

In [14]:
```
df["abbrev"].value_counts()
```

Out[14]:
```
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
```

In [15]:
```
sns.relplot(x="alcohol",y="total",data=df)
```

Out[15]:
```
<seaborn.axisgrid.FacetGrid at 0x26e4067b910>
```

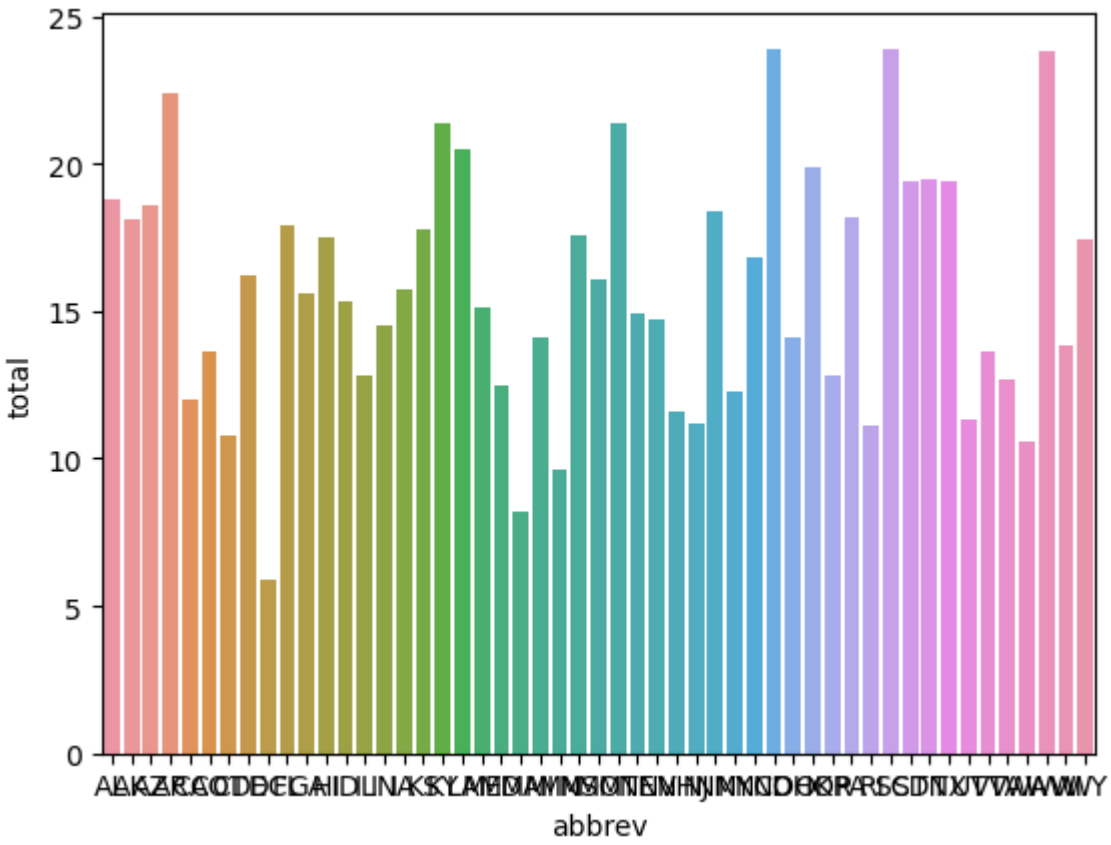



```
Inference : The relationship between alcohol and total is directly
proportional.
```

Barplot

In [16]:
```
sns.barplot(data=df,x="abbrev",y="total",ci=None)
```

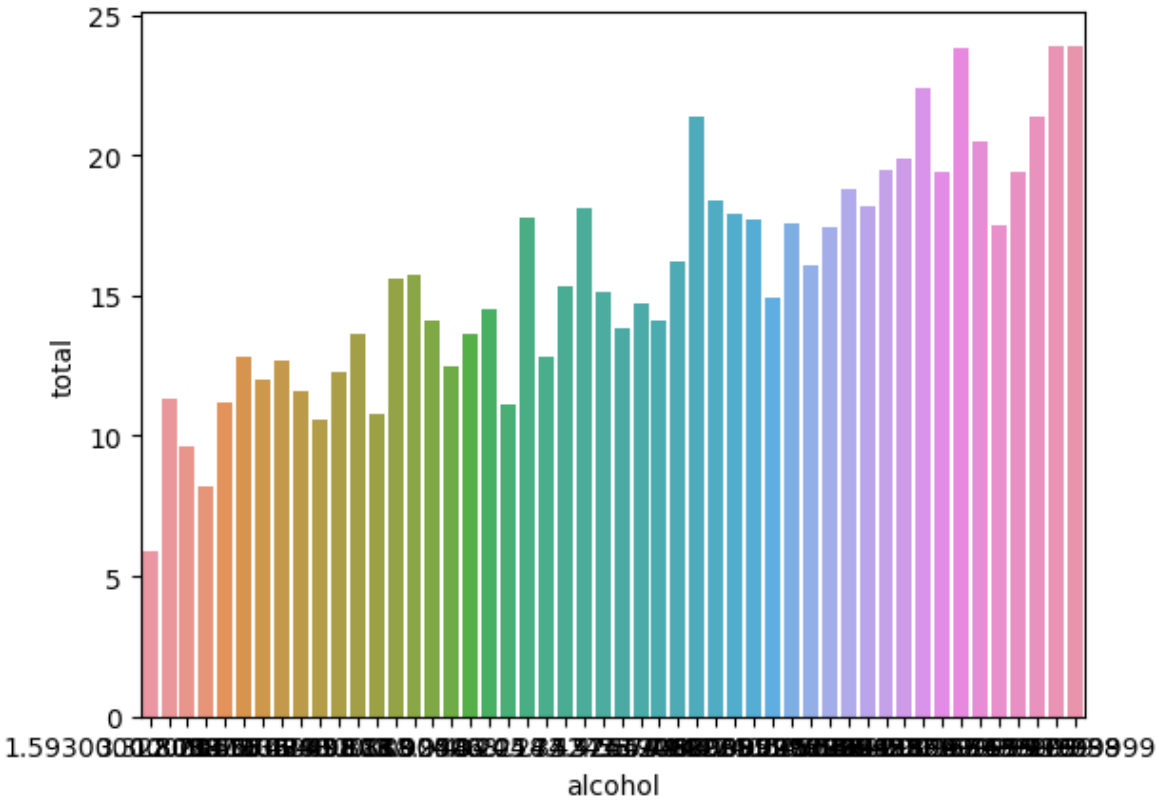
Out[16]:
```
<AxesSubplot:xlabel='abbrev', ylabel='total'>
```




```
Inference : The barplot displays the mean total crashes for different values
of abbrev.The total crashes is high at low abbrev value.
```

In [17]:
```
sns.barplot(data=df,x="alcohol",y="total",ci=None)
```

Out[17]:
```
<AxesSubplot:xlabel='alcohol', ylabel='total'>
```

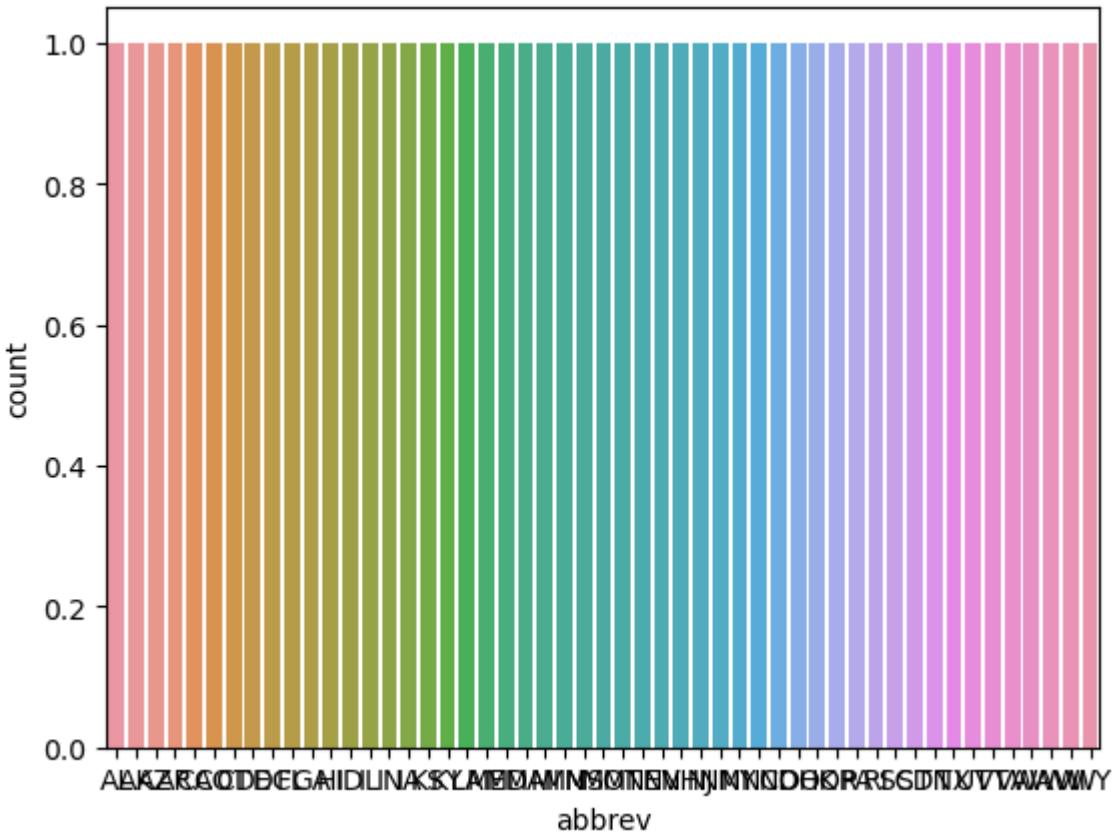



```
Inference : The barplot displays the mean total crashes for different levels
of alcohol consumption.So, if the alcohol consumption is high, then total
crashes is also high.
```

Countplot

In [18]:
```
sns.countplot(x="abbrev",data=df)
```

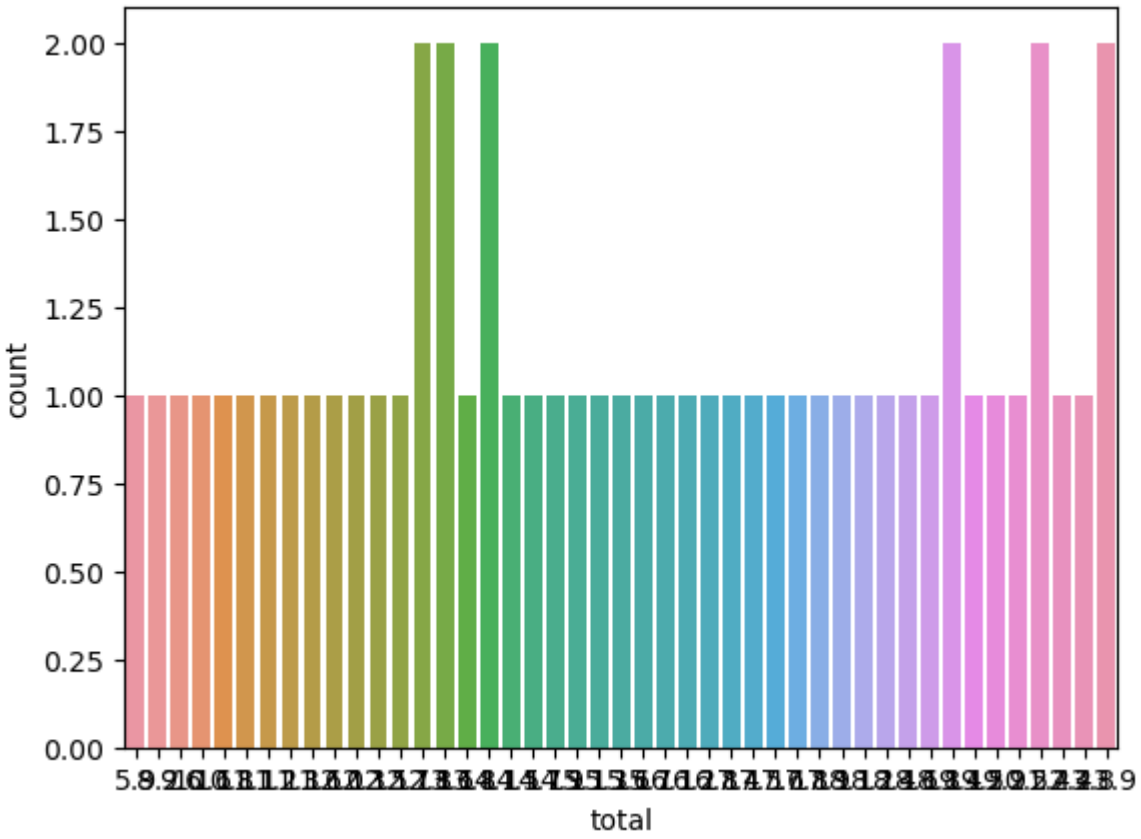
Out[18]:
```
<AxesSubplot:xlabel='abbrev', ylabel='count'>
```




```
Inference : It counts no.of abbrev as different abbrev as single value
```

In [19]:
```
sns.countplot(x="total",data=df)
```

Out[19]:
```
<AxesSubplot:xlabel='total', ylabel='count'>
```

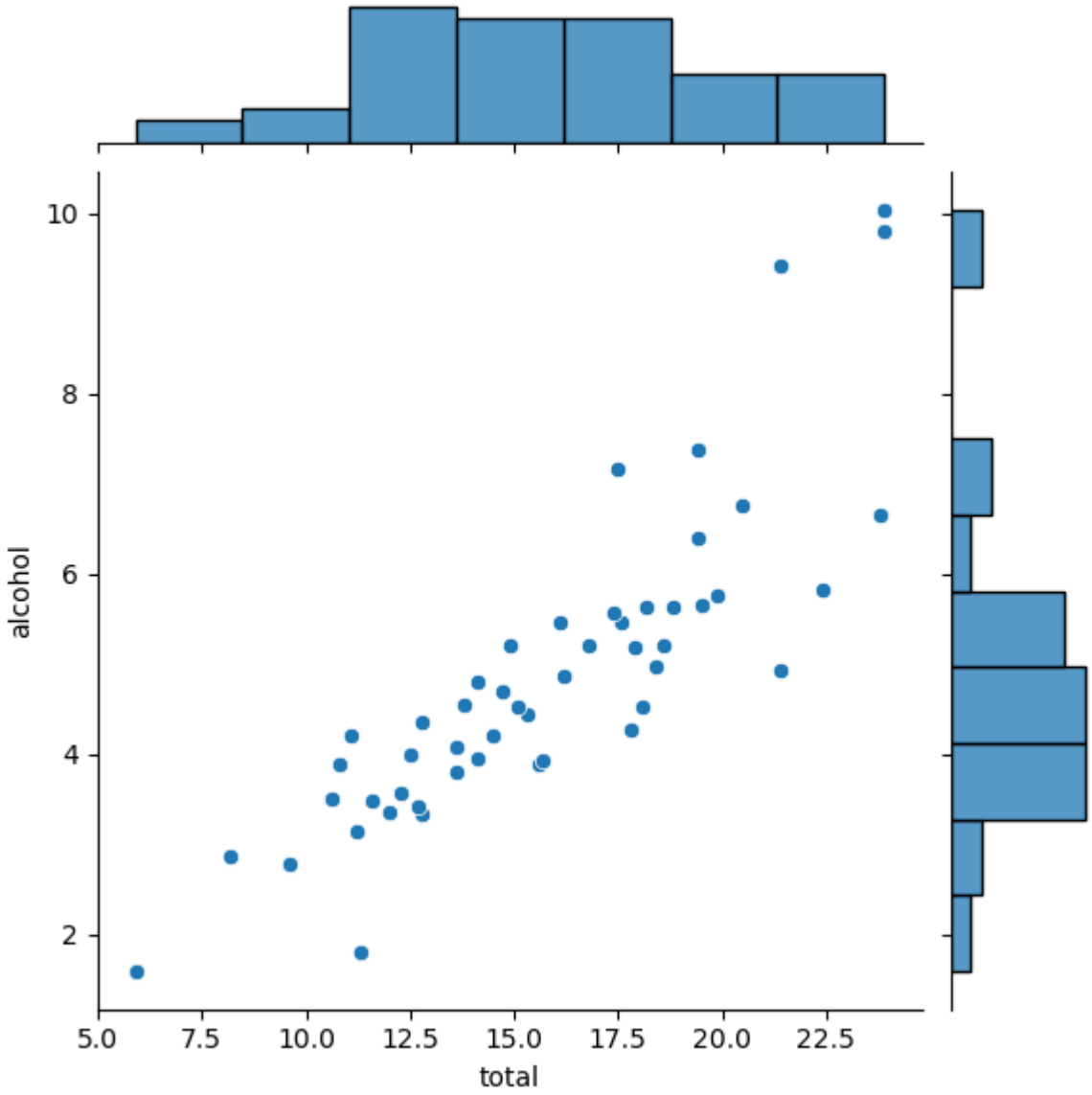



```
Inference : This countplot displays the frequency of total crashes in the
dataset and its count is high at 6 values.
```

Jointplot

In [20]:
```
sns.jointplot(x="total",y="alcohol",data=df)
```

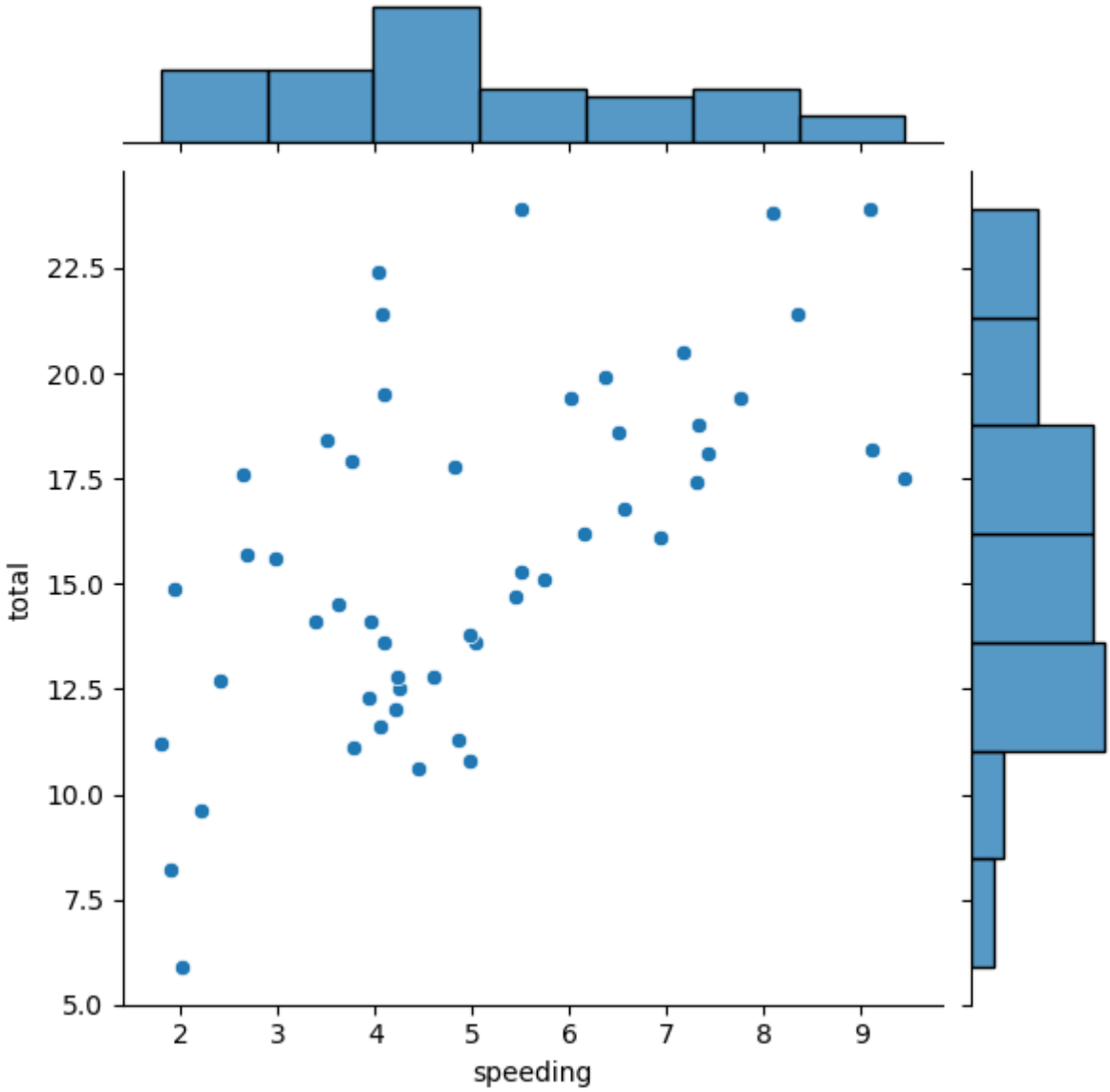
Out[20]:
```
<seaborn.axisgrid.JointGrid at 0x26e40f7e670>
```



Inference : The jointplot tells relationship between the total and alcohol.As total increases alcohol increases

In [21]:
```
sns.jointplot(x="speeding",y="total",data=df)
```

Out[21]:
```
<seaborn.axisgrid.JointGrid at 0x26e410254c0>
```

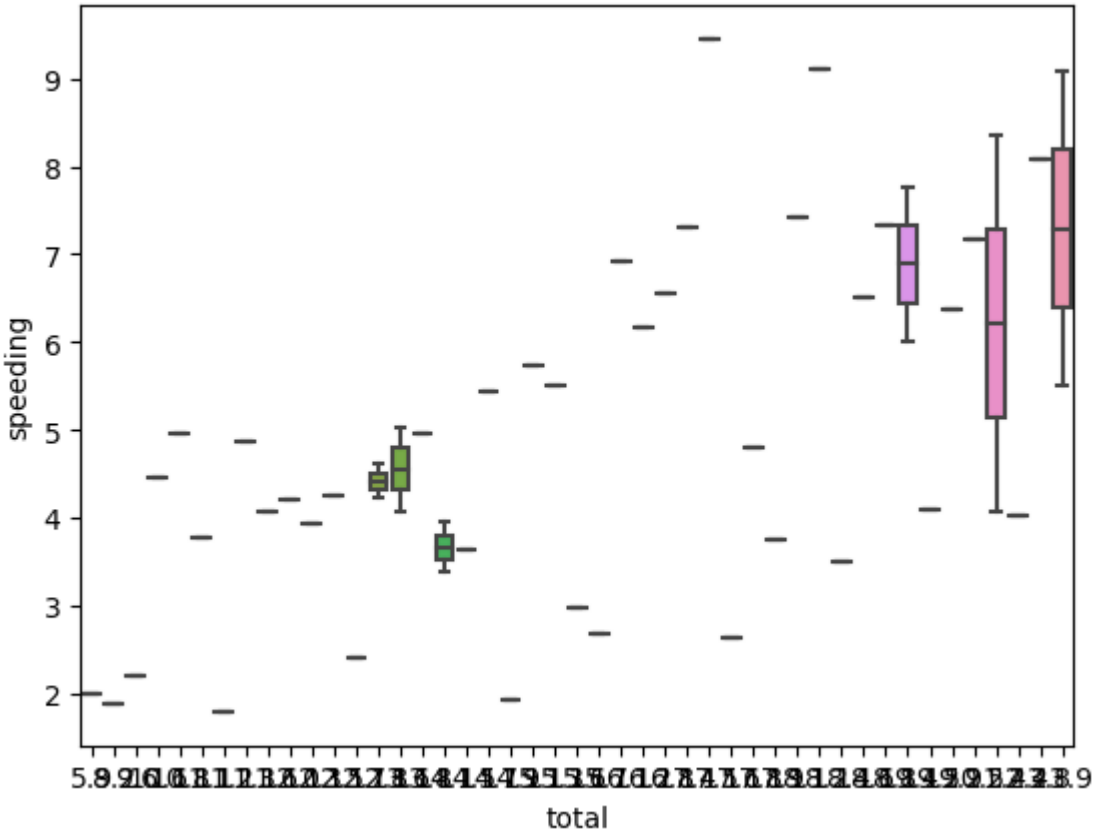



```
Inference : The jointplot tells relationship between the speeding and
total.It doesn't have spacific pattern.
```

Boxplot

In [22]:
```
sns.boxplot(x="total",y="speeding",data=df)
```

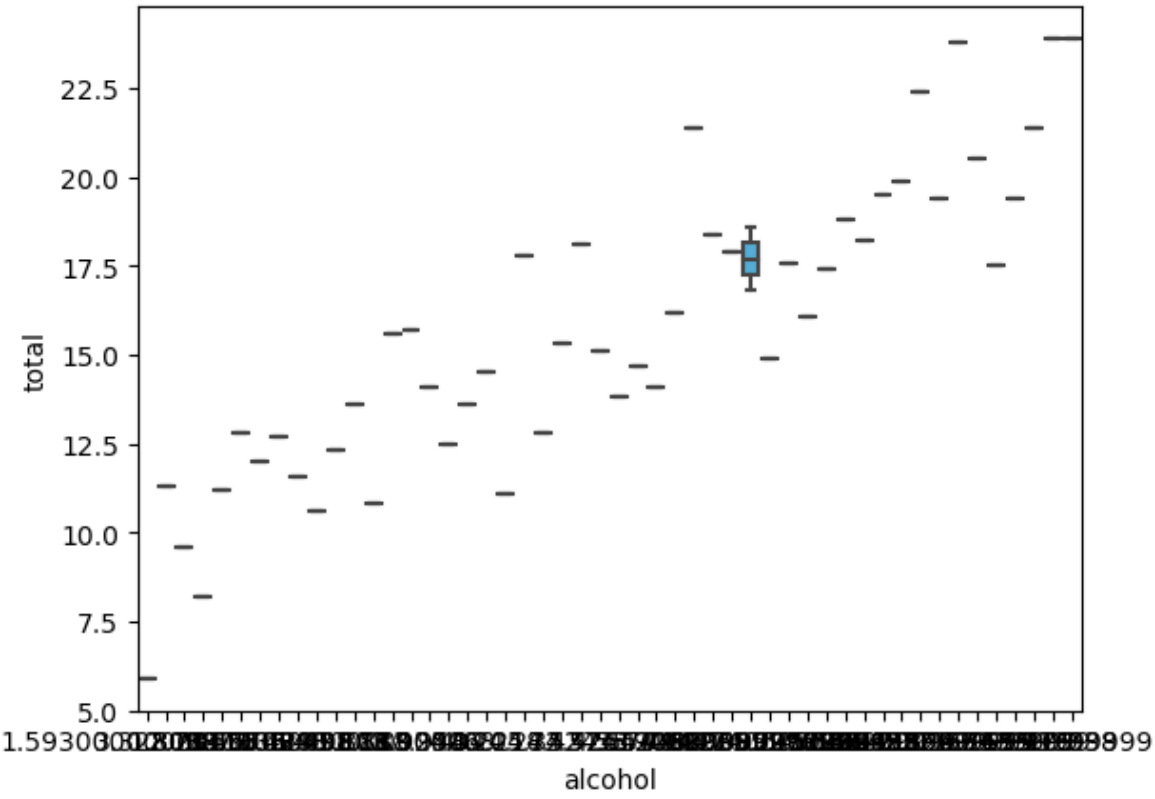
Out[22]:
```
<AxesSubplot:xlabel='total', ylabel='speeding'>
```



Inference : The boxplot shows the distribution of total crashes for different levels of speeding.These lines indicates outliers.

In [23]:
```
sns.boxplot(x="alcohol",y="total",data=df)
```

Out[23]:
```
<AxesSubplot:xlabel='alcohol', ylabel='total'>
```



Inference : The boxplot shows the distribution of total crashes for different levels of alcohol consumption.These lines indicates outliers.

Correlation

In [24]:
```
corr=df.corr()
corr
```

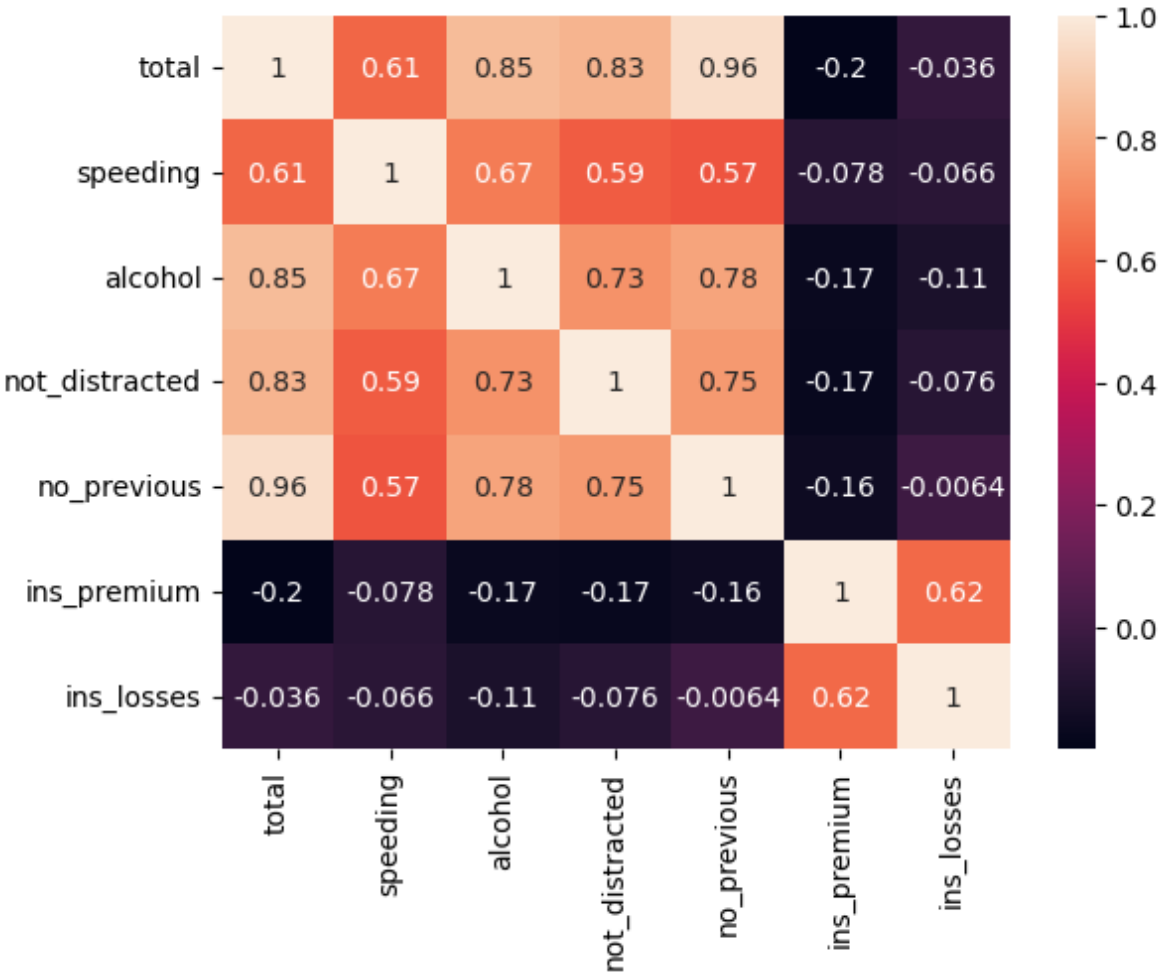
Out[24]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_los
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.199702	-0.036
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.077675	-0.065
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.170612	-0.112
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	-0.174856	-0.075
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	-0.156895	-0.006
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856	-0.156895	1.000000	0.623
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970	-0.006359	0.623116	1.000

Heatmap

In [25]:
```
sns.heatmap(corr,annot=True)   #annot used to appear values on heatmap
```

Out[25]: <AxesSubplot:>



Inference : From the above plot we can say that it is negatively,positively and neturally correlated

In [26]:
```
df.isnull().any()
```

Out[26]:
```
total             False
speeding          False
alcohol           False
not_distracted    False
no_previous       False
ins_premium       False
ins_losses        False
abbrev            False
dtype: bool
```

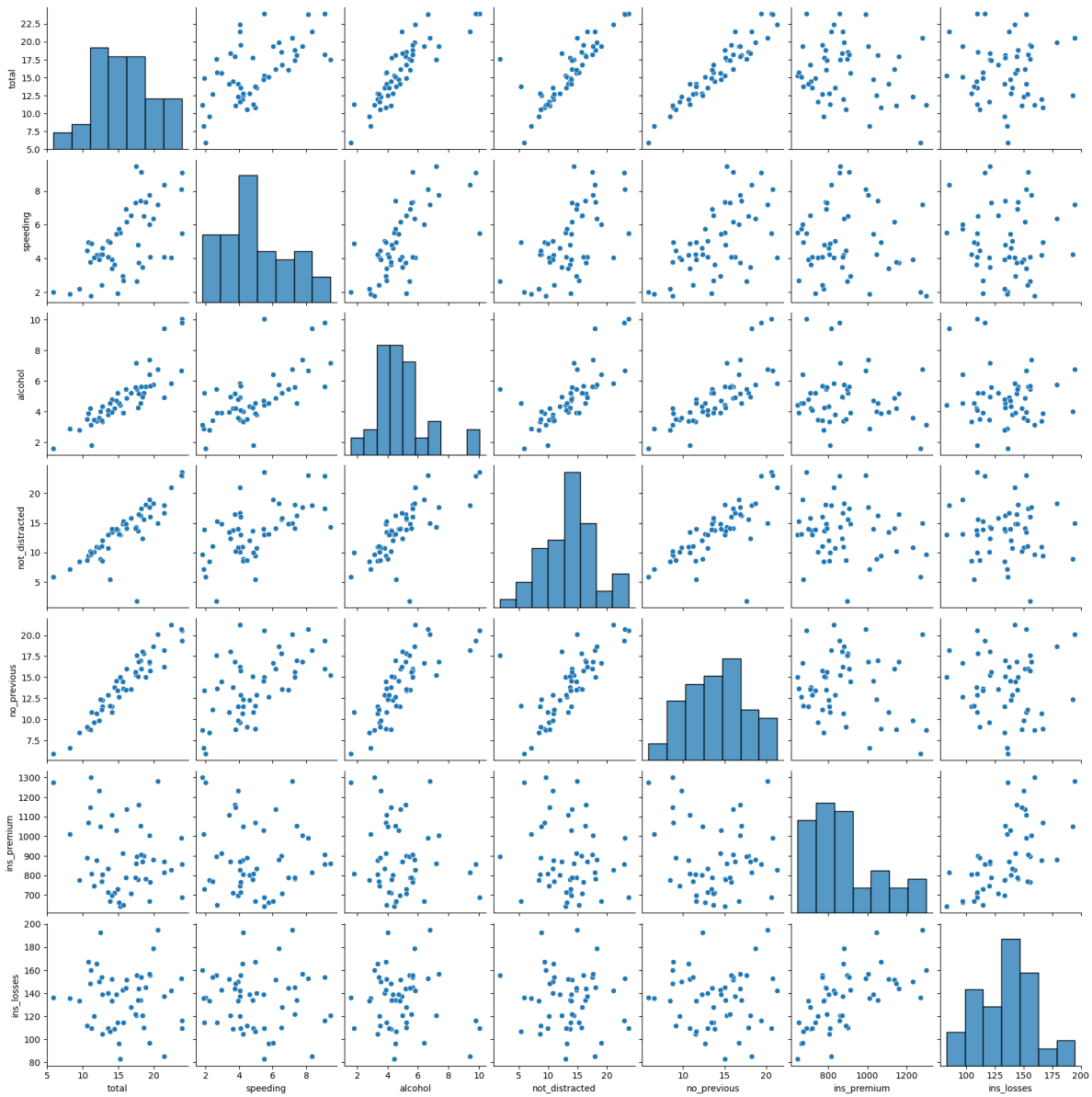
In [27]:
```
df.isnull().sum()
```

Out[27]:
```
total             0
speeding          0
alcohol           0
not_distracted    0
no_previous       0
ins_premium       0
ins_losses        0
abbrev            0
dtype: int64
```

Pairplot

In [28]:
```
sns.pairplot(data=df)
```

Out[28]:
```
<seaborn.axisgrid.PairGrid at 0x26e41821640>
```

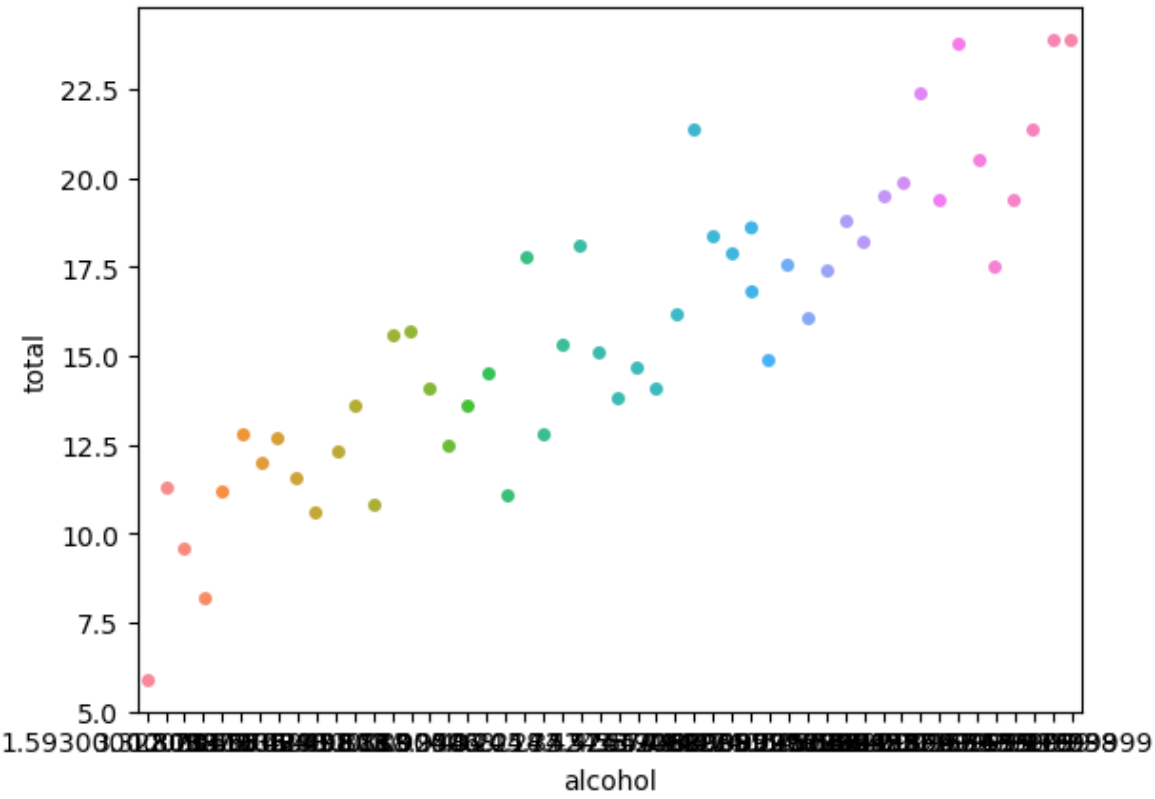


```
Inference : The pairplot distribution gives the relationship between all the values in a dataset.
```

Stripplot

In [32]:
```
sns.stripplot(x="alcohol",y="total",data=df)
```

Out[32]:
```
<AxesSubplot:xlabel='alcohol', ylabel='total'>
```

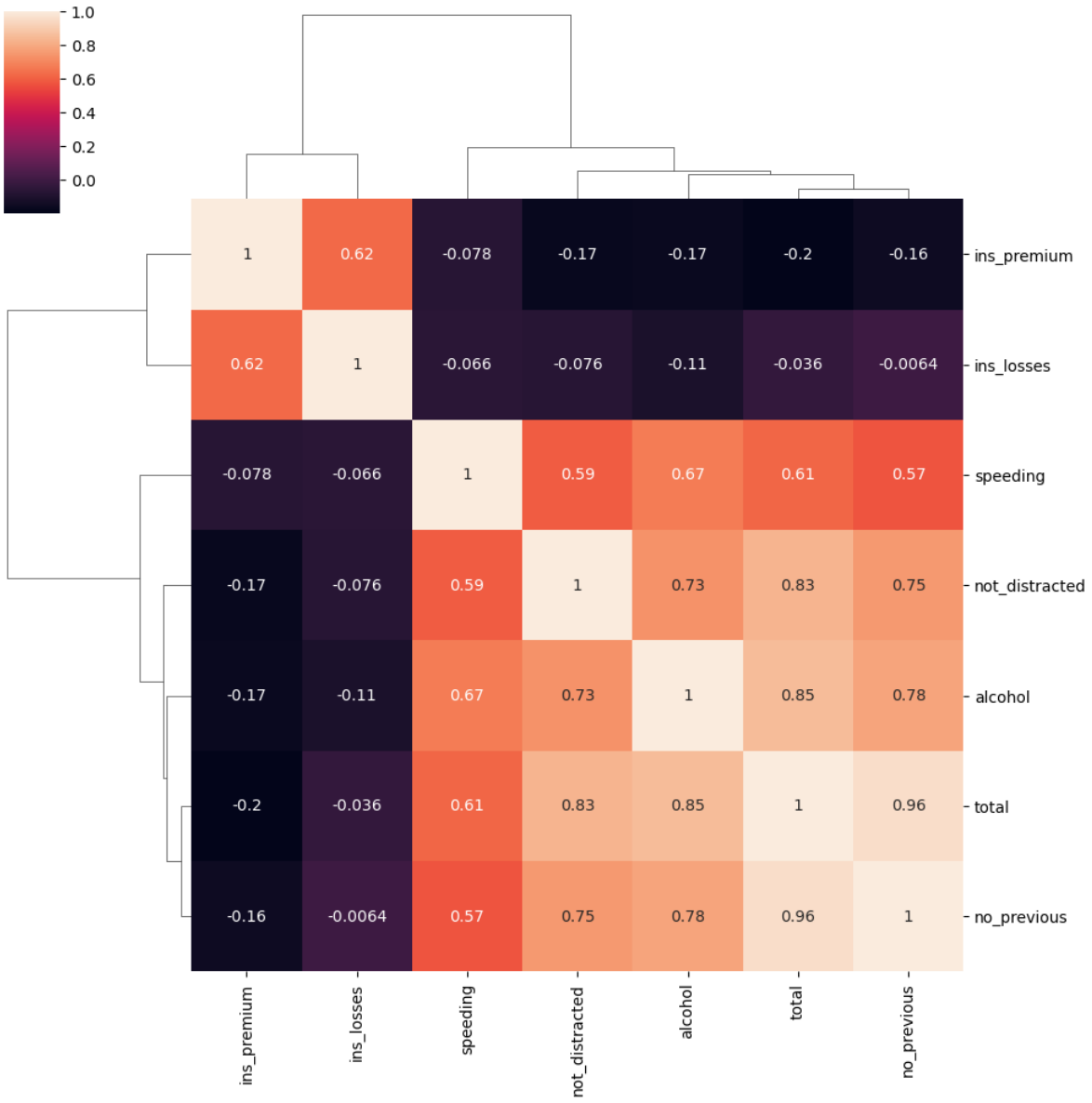



```
Inference : The stripplot shows that the total crashes increases by the increase of alcohol consumption.
```

Clustermap

In [39]:
```
sns.clustermap(corr,data=df,annot=True)
```

Out[39]: <seaborn.matrix.ClusterGrid at 0x26e483482b0>



Inference : The clustermap gives the barplot of any values in a dataset between all values in a dataset.