In [1]:
```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.
23.5)
Requirement already satisfied: pandas>=0.25 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.5.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /usr/local/lib/python3.10/dist-packages (from seaborn)
(3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.
1,>=3.1->seaborn) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=
3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6
.1,>=3.1->seaborn) (4.42.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6
.1,>=3.1->seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1
,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>
=3.1->seaborn) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.
1,>=3.1->seaborn) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=
3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.25->seabor
n) (2023.3.post1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->ma
tplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
```

In [2]:
```
import seaborn as sns
```

In [3]:
```
print(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', 'tip
s', 'titanic']
```

In [4]:
```
df=sns.load_dataset('car_crashes')
#loading dataset car_crashes from seaborn library
```

In [5]:
```
df
```

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

Out[6]:

	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 [7]:
```
df.tail()
```

Out[7]:

	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

In [8]:
```
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 [9]:
```
df.describe()
```

Out[9]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
count	51.000000	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	134.493137
std	4.122002	2.017747	1.729133	4.508977	3.764672	178.296285	24.835922
min	5.900000	1.792000	1.593000	1.760000	5.900000	641.960000	82.750000
25%	12.750000	3.766500	3.894000	10.478000	11.348000	768.430000	114.645000
50%	15.600000	4.608000	4.554000	13.857000	13.775000	858.970000	136.050000
75%	18.500000	6.439000	5.604000	16.140000	16.755000	1007.945000	151.870000
max	23.900000	9.450000	10.038000	23.661000	21.280000	1301.520000	194.780000

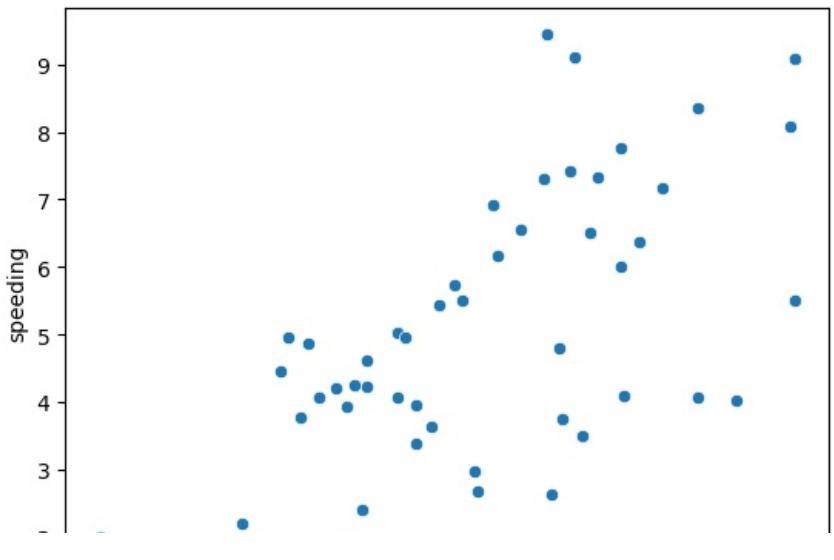
In [10]:
```
df.head()
```

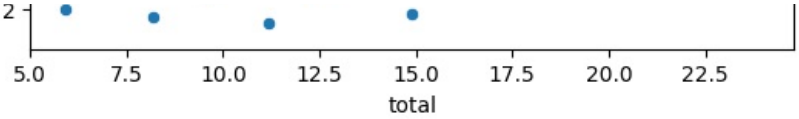
Out[10]:

	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 [11]:
```
sns.scatterplot(x="total", y="speeding", data=df)

# Inference:
# Relation between total accidents and ones involved in speeding
# From below scatter plot graph -- as speed increases, total car crashes also increases
```

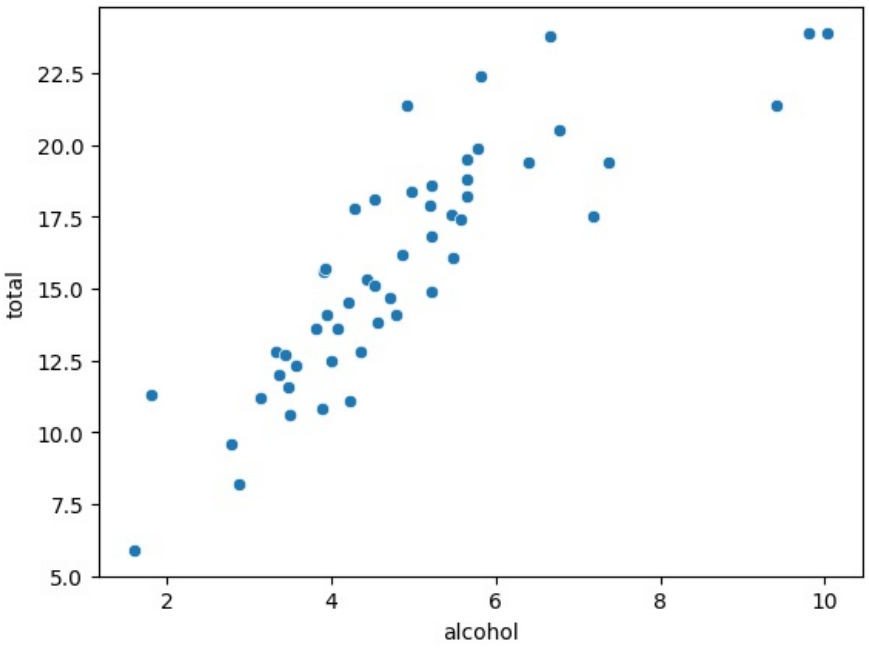
Out[11]:
```
<Axes: xlabel='total', ylabel='speeding'>
```





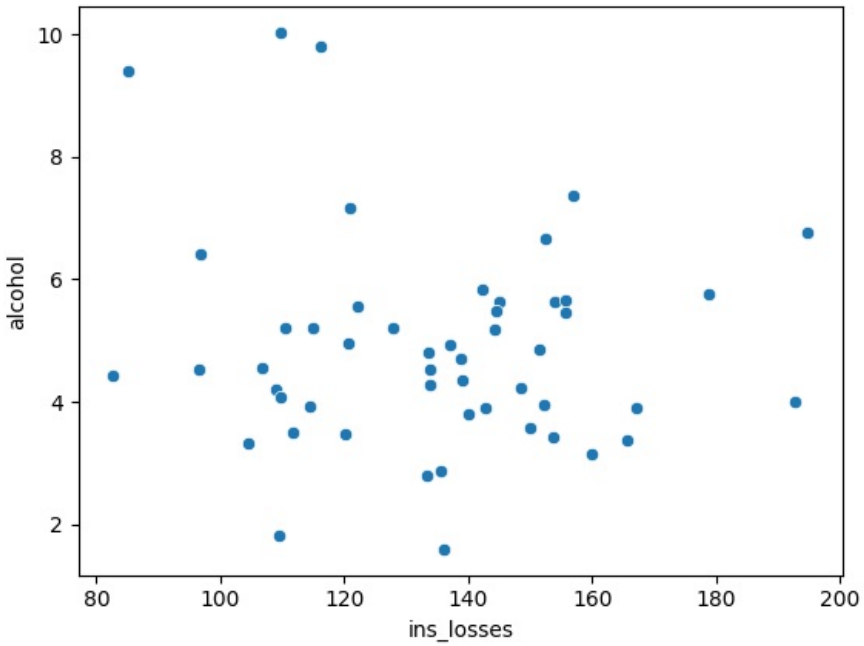
In [12]:
```
sns.scatterplot(x="alcohol",y="total",data=df)

# Relation between total accidents and ones involved in alcohol drinking
# As alcohol drinking accidents increases total accidents also increases
```

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



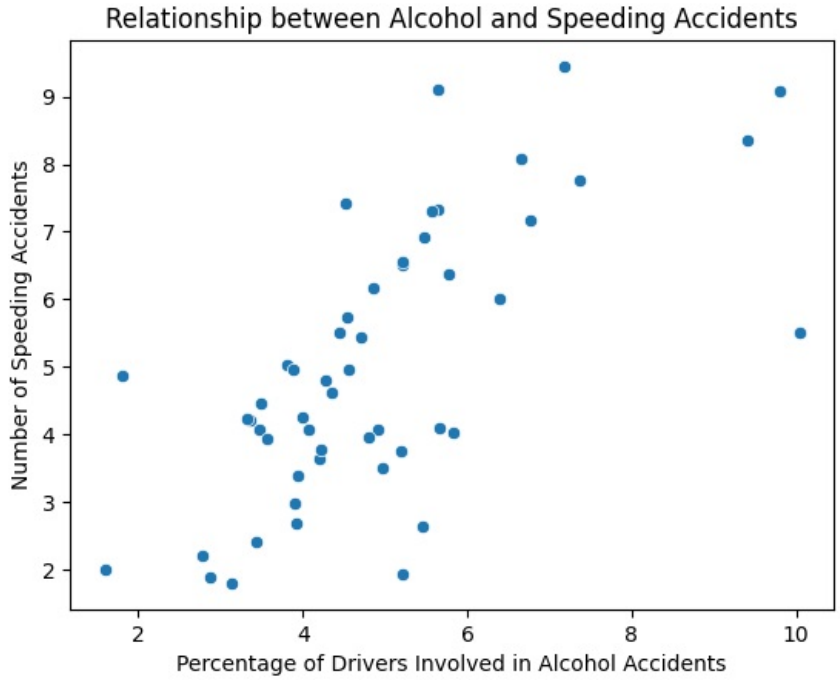
In [13]:
```
sns.scatterplot(x="ins_losses",y="alcohol",data=df)

# There is no certain relation between ins_losses and alochol sometimes it works with direct propostionality and
```

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



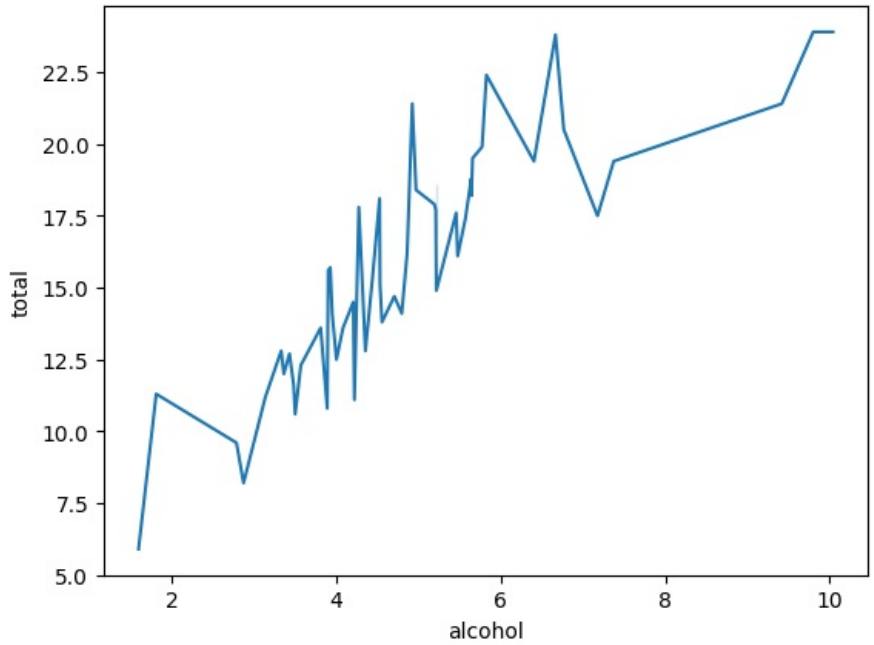
In [14]:
```
import matplotlib.pyplot as plt
```

```
sns.scatterplot(data=df, x='alcohol', y='speeding')
plt.xlabel('Percentage of Drivers Involved in Alcohol Accidents')
plt.ylabel('Number of Speeding Accidents')
plt.title('Relationship between Alcohol and Speeding Accidents')
plt.show()

# Scatter plot graph between alcohol and speeding accidents and labelled x and y axis separately as 'Percentage o
# If Percentage of Drivers Involved in Alcohol Accidents increases then Number of Speeding Accidents also increas
```



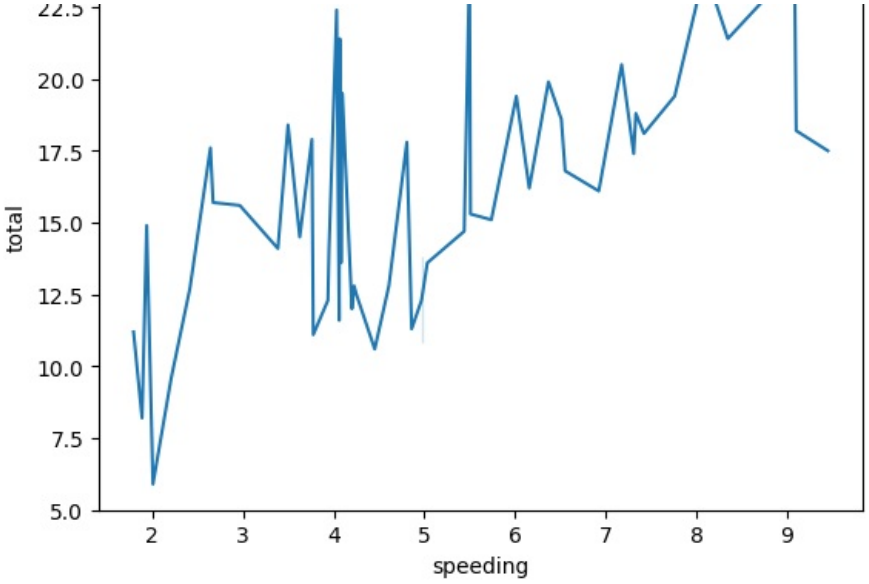
In [15]:
```
sns.lineplot(x="alcohol", y="total", data=df)

# transparent area - confidence interval(shows range)
# lineplot between total accidents and Drivers Involved in Alcohol Accidents
```

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



In [16]:
```
sns.lineplot(x="speeding", y="total", data=df)

# lineplot between total accidents and speeding Accidents
```

Out[16]: <Axes: xlabel='speeding', ylabel='total'>

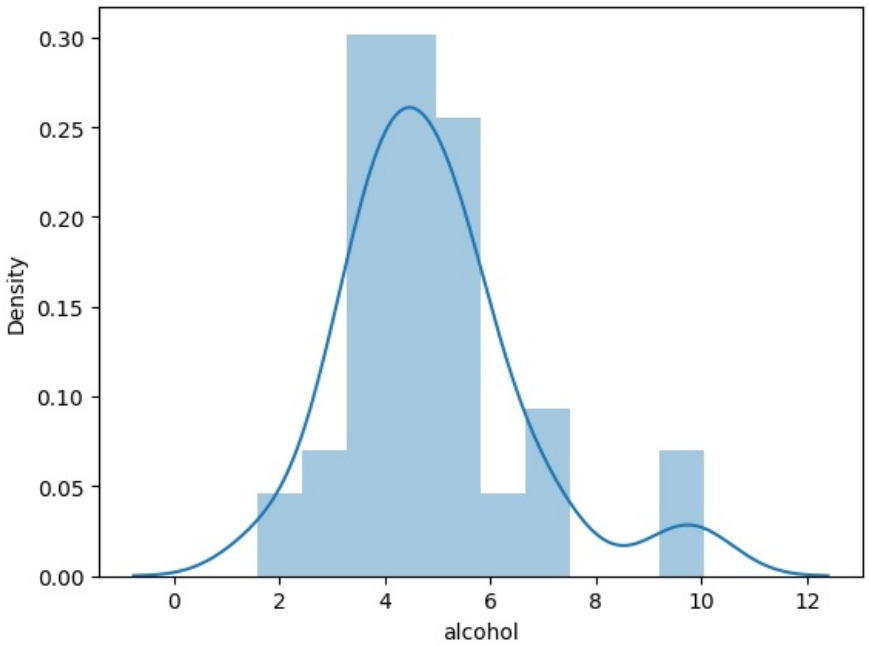


In [31]:

```
sns.distplot(df["alcohol"])

# Distplot is a univariate distribution of observations
# It combines matplotlib histogram and kernel distribution plot
# univariate distribution of accidents invloved in alcohol driving in car_crashes dataset
# involved in alcohol incident ranges from 4-6 %
```

```
<ipython-input-31-3982b821b6b0>: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

  sns.distplot(df["alcohol"])
```

Out[31]: <Axes: xlabel='alcohol', ylabel='Density'>



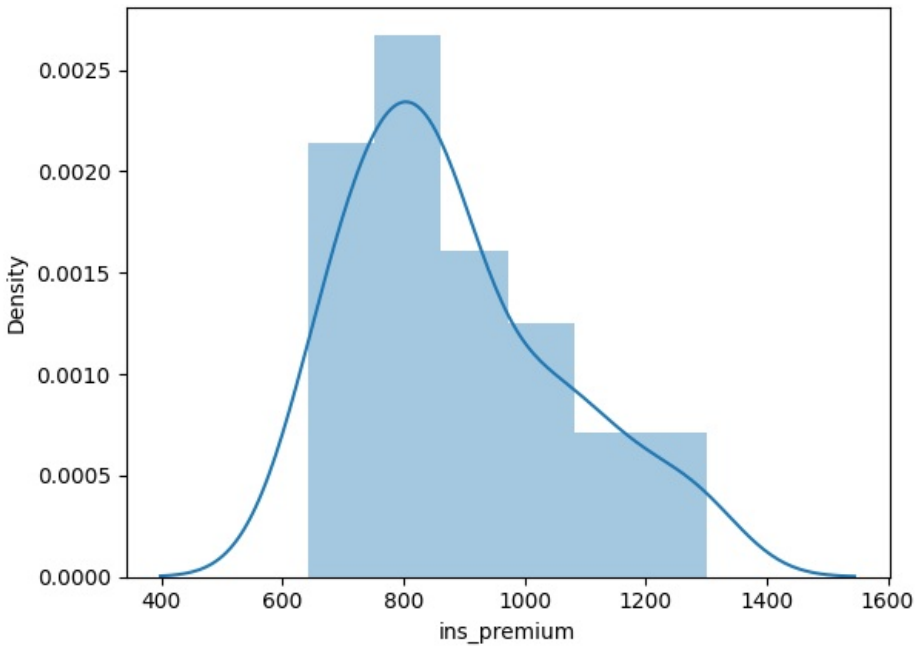
In [32]:

```
sns.distplot(df["ins_premium"])

# insurance premium ranges from 800 - 1000 dollars
```

```
<ipython-input-32-881032b4b789>: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

  sns.distplot(df["ins_premium"])
```

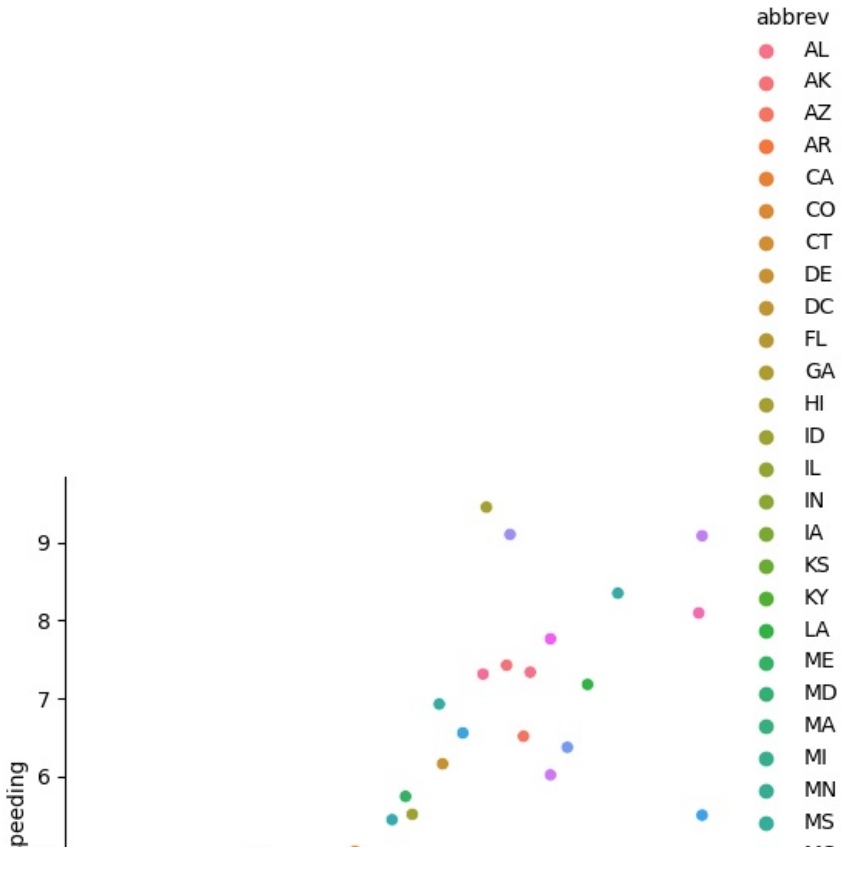
Out[32]: <Axes: xlabel='ins_premium', ylabel='Density'>

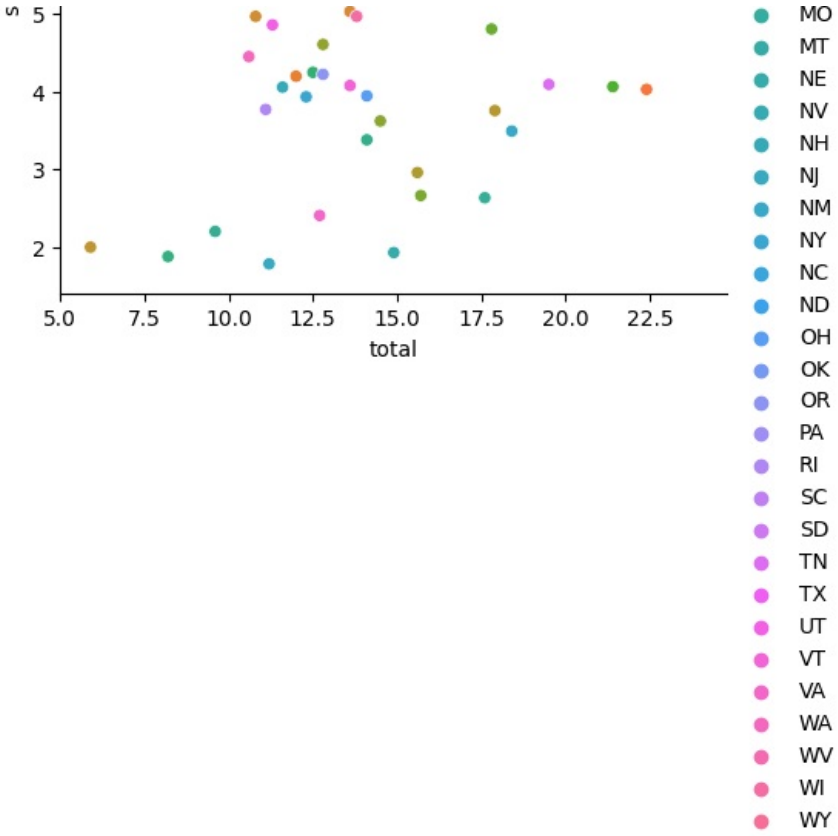


In [20]:

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

# It allows us to visualise how variables within a dataset relate to each other
# hue paramater  - says which column in dataframe is need for color encoding (categorical variable)
# scatter plot between total accidents and speeding relating to abbrev column
```

Out[20]: <seaborn.axisgrid.FacetGrid at 0x7daece619f30>





In [21]:
```
df["abbrev"].value_counts()
# checking total count of 'abbrev' feature which is a multivariate
```

Out[21]:
```
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
```

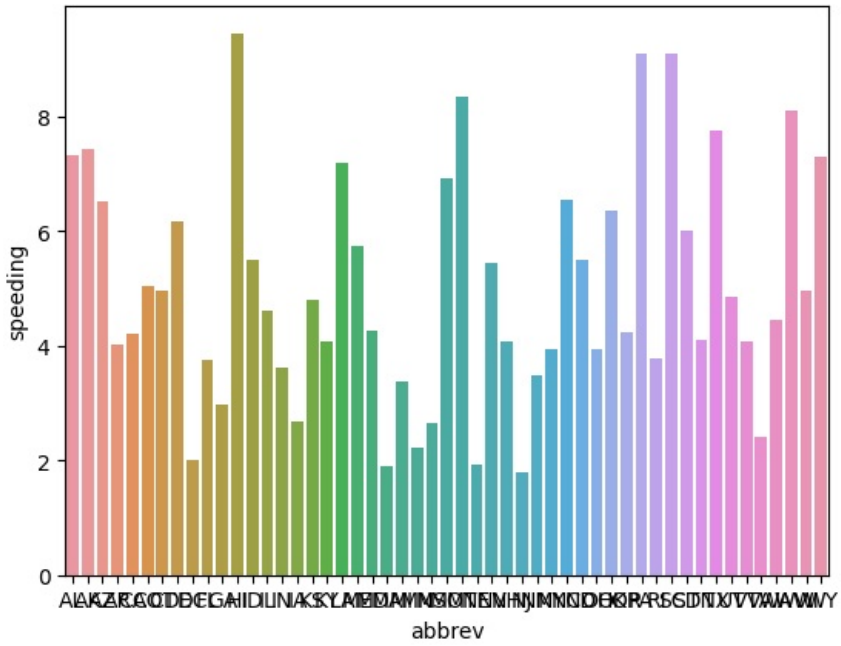
```
KY    1
LA    1
ME    1
MD    1
MA    1
MI    1
MN    1
WY    1
Name: abbrev, dtype: int64
```

In [22]:

```
sns.barplot(data=df,x="abbrev",y="speeding", ci=None)

# graph b/w categorical and numerical variable
# Bar graph plot between abbrev and speeding
# can dtect which abbrev has high speed in accidents
```

```
<ipython-input-22-5a0d96af314f>:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(data=df,x="abbrev",y="speeding", ci=None)
```

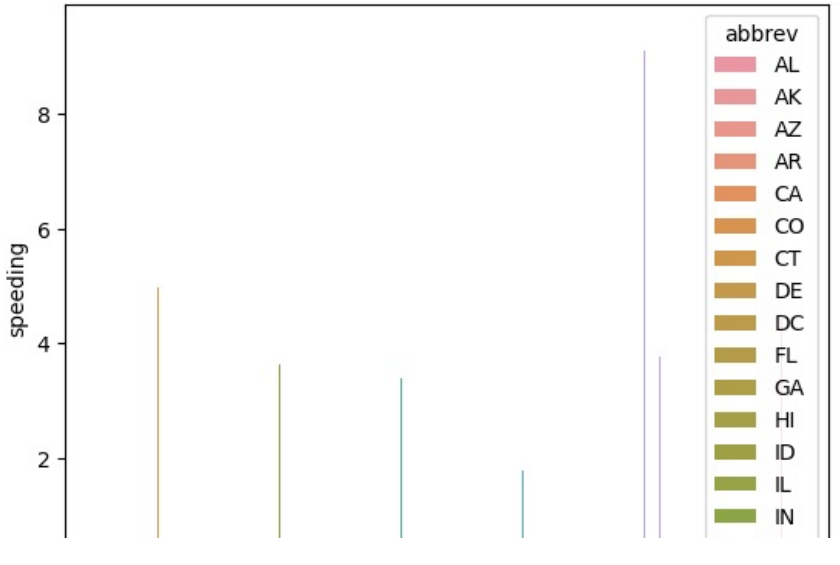
Out[22]: <Axes: xlabel='abbrev', ylabel='speeding'>



In [23]:

```
sns.barplot(data=df,x="abbrev",y="speeding", hue="abbrev")   #color coding - hue

# Bar graph plot between categorical and numerical variable i.e abbrev and speeding
```

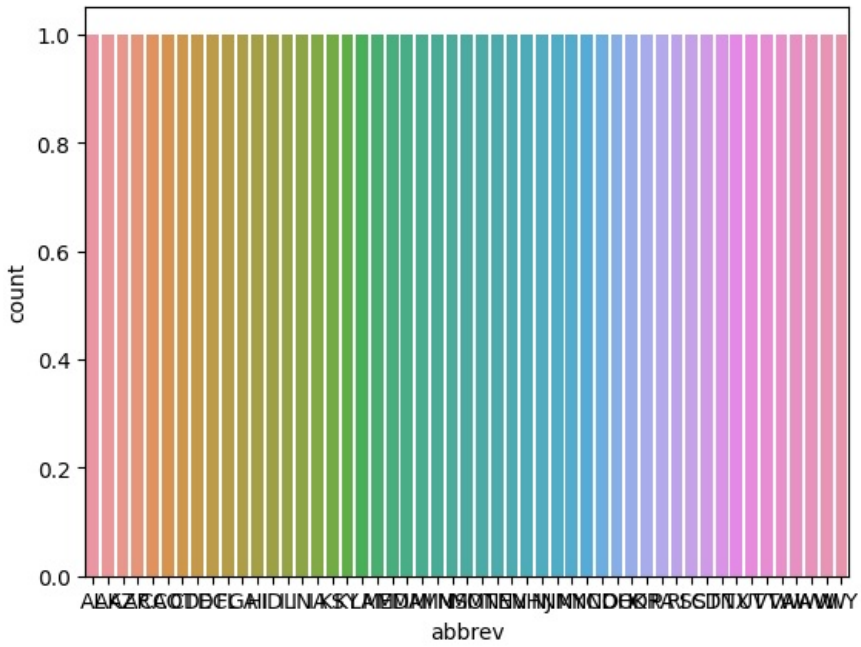
Out[23]: <Axes: xlabel='abbrev', ylabel='speeding'>



In [24]:

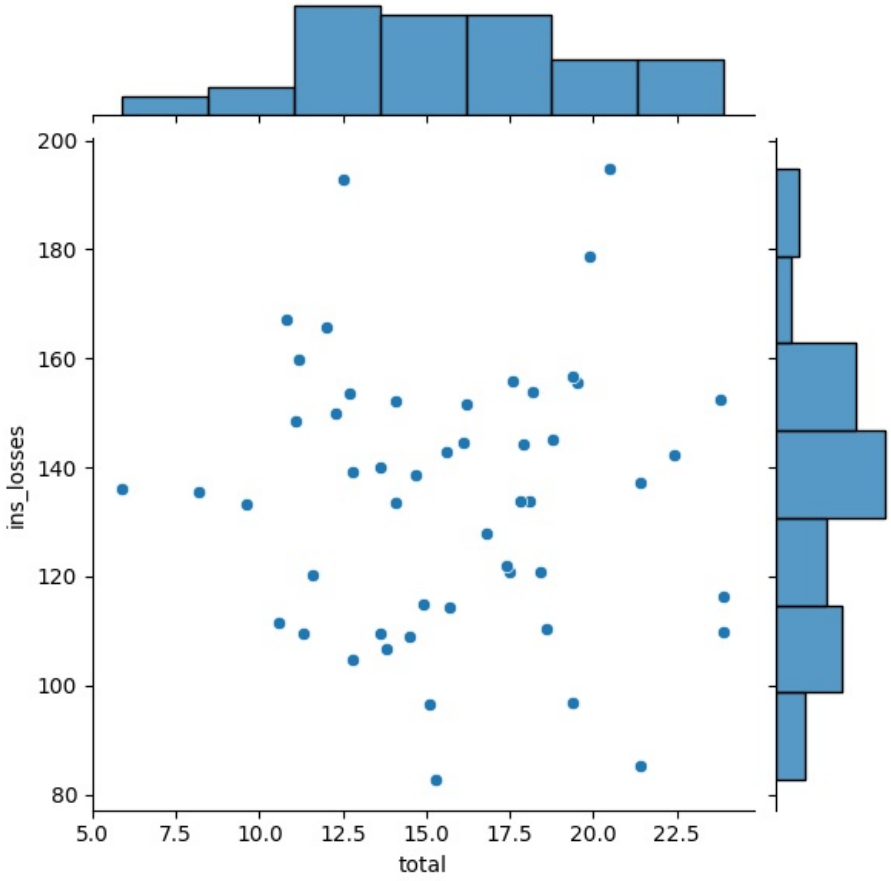
```
sns.countplot(x="abbrev",data=df)

# Used to represent the occurrence(counts) of the observation present in the categorical variable.
```

Out[24]: <Axes: xlabel='abbrev', ylabel='count'>



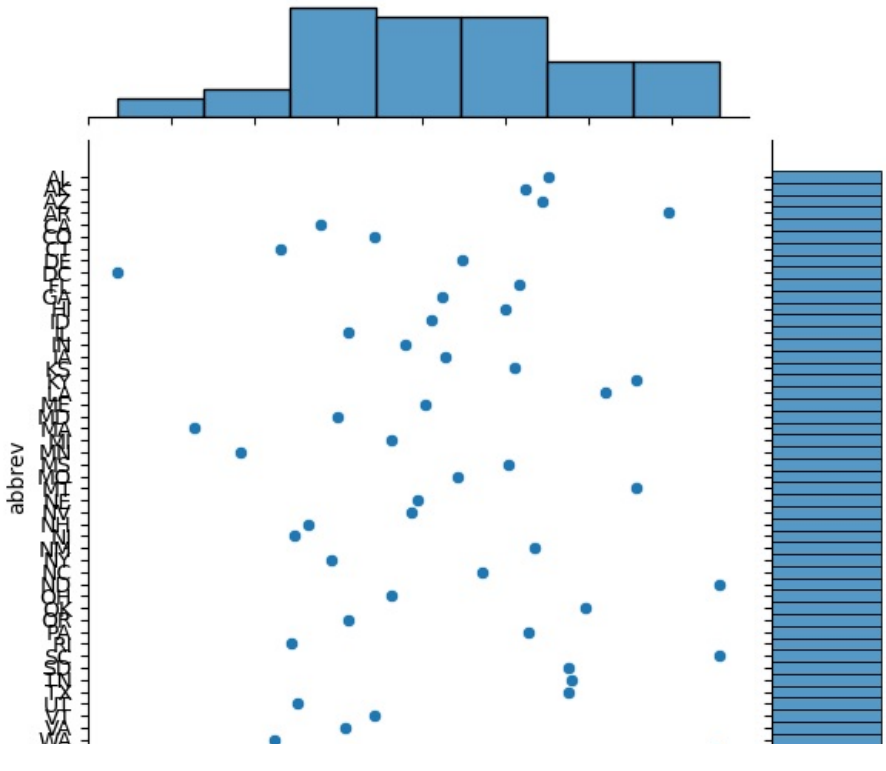
In [25]:
```
sns.jointplot(x="total",y="ins_losses",data=df)

# dispalys relationship b/w total total accidents and insurance losses
# Relationship between two variables and the distribution of individuals of each variable.
# It  is bivariate analysis -  2 var at a time
```

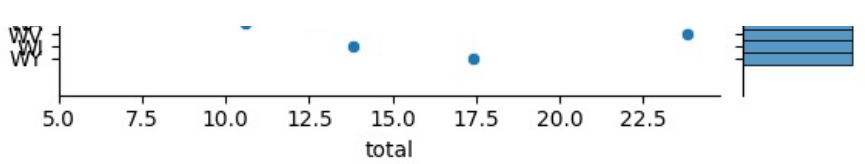
Out[25]: <seaborn.axisgrid.JointGrid at 0x7daeca5e6d70>



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

# dispalys relationship b/w total total accidents and abbrev variables
# Relationship between two variables and the distribution of individuals of each variable which is  a bivariate a
```

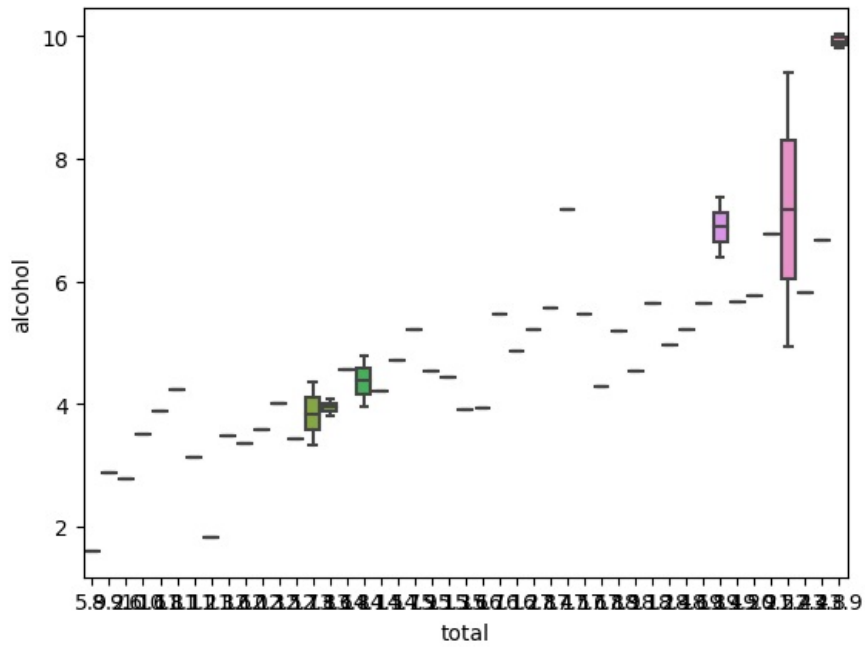
Out[26]: <seaborn.axisgrid.JointGrid at 0x7daecc142050>





In [27]:
```
sns.boxplot(x="total",y="alcohol", data=df)

# Compares the interquartile ranges (that is, the box lengths) to examine how the data is dispersed between each
```

Out[27]: <Axes: xlabel='total', ylabel='alcohol'>



In [29]:
```
corr=df.corr()
corr

# correlation --- statistical analysis relation b/w 2 variables(either directly prop or inversely prop)
# values > 0.5 --- highly correlated
# values < 0.5 --- less correlated
```

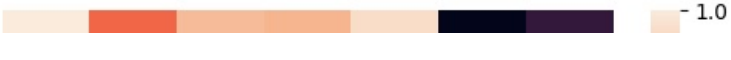
```
<ipython-input-29-604496f212db>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  corr=df.corr()
```

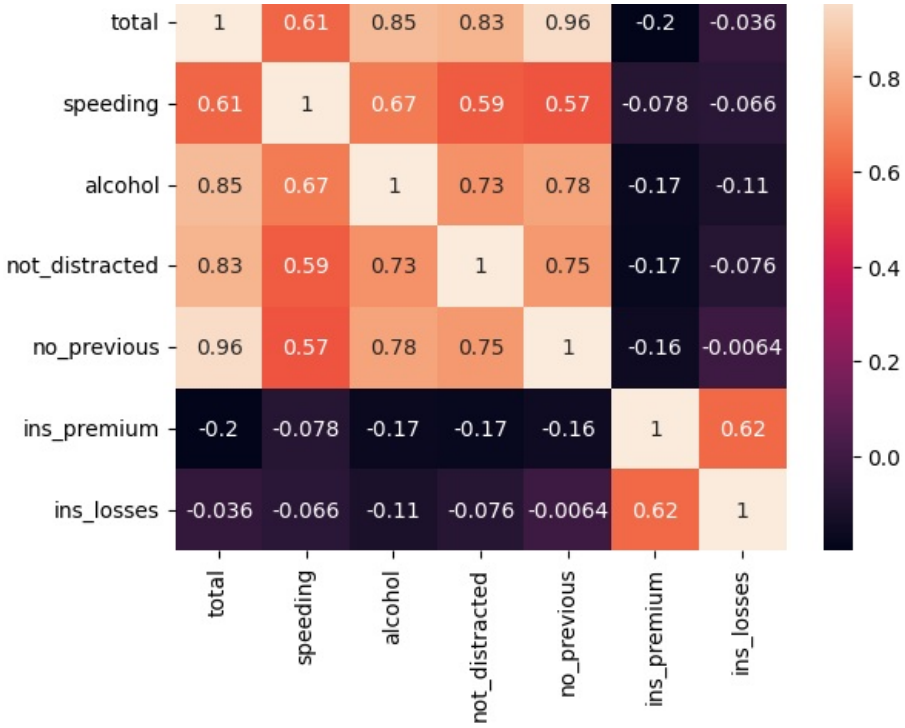
Out[29]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.199702	-0.036011
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.077675	-0.065928
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.170612	-0.112547
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	-0.174856	-0.075970
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	-0.156895	-0.006359
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856	-0.156895	1.000000	0.623116
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970	-0.006359	0.623116	1.000000

In [45]:
```
sns.heatmap(corr,annot=True)

# darker shade - less correlation
# lighter shade - highly correlated
# It represents how each feature depends on the other features
```

Out[45]: <Axes: >





In [30]: