In [1]:

```
import numpy as np
import pandas as pd
import matplotlib as plt
import seaborn as sns
```

```
##Assignment 8 th september
1.Take car_crashes dataset from seaborn library
2.load the dataset
3.Perfrom Data Visualization
4.Inference is must for each and every graph
5.Submit it by wednesday in html/pdf format
```

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
dataset = sns.load_dataset("car_crashes")
```

In [3]:

```
dataset.head()
```

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

In [4]:

```
corr = dataset.corr()
corr
```

```
C:\Users\pbalu\AppData\Local\Temp\ipykernel_11752\897440734.py:1: 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.
  corr = dataset.corr()
```

Out[4]:

	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 [6]:
```
df = dataset.ins_premium.isnull()
```

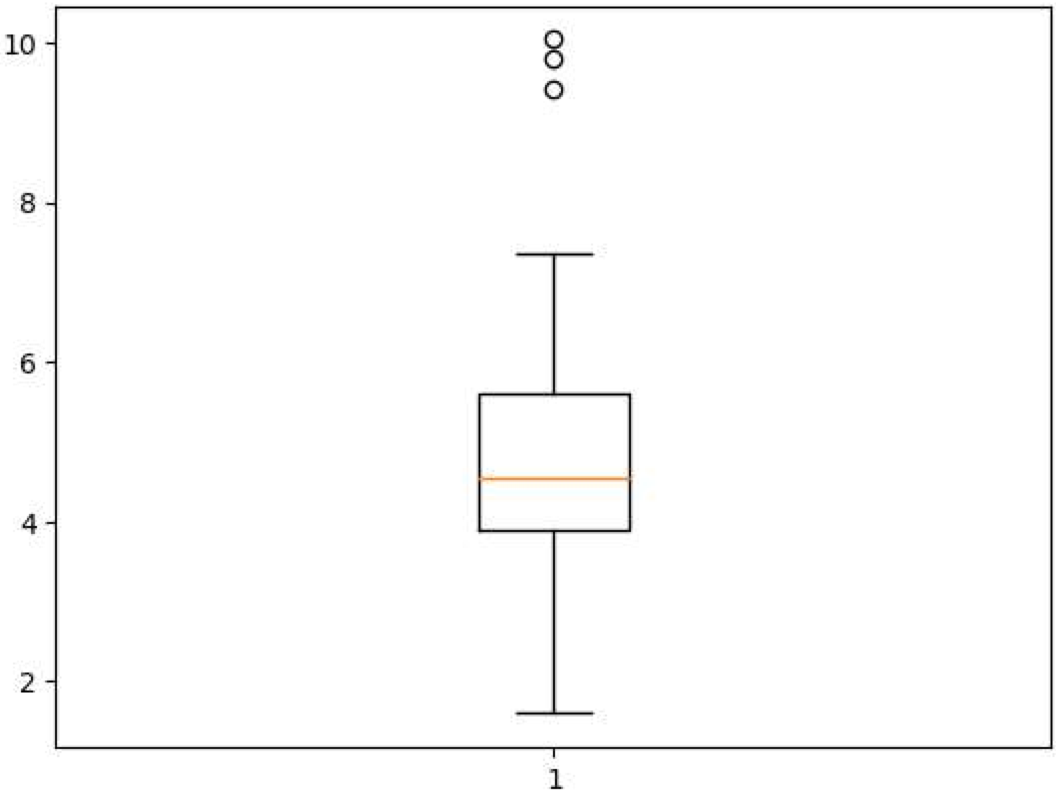
In [5]:
```
dataset.head()
```

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 [7]:
```
#for Outliers:
df = plt.boxplot(dataset.alcohol)
df
```

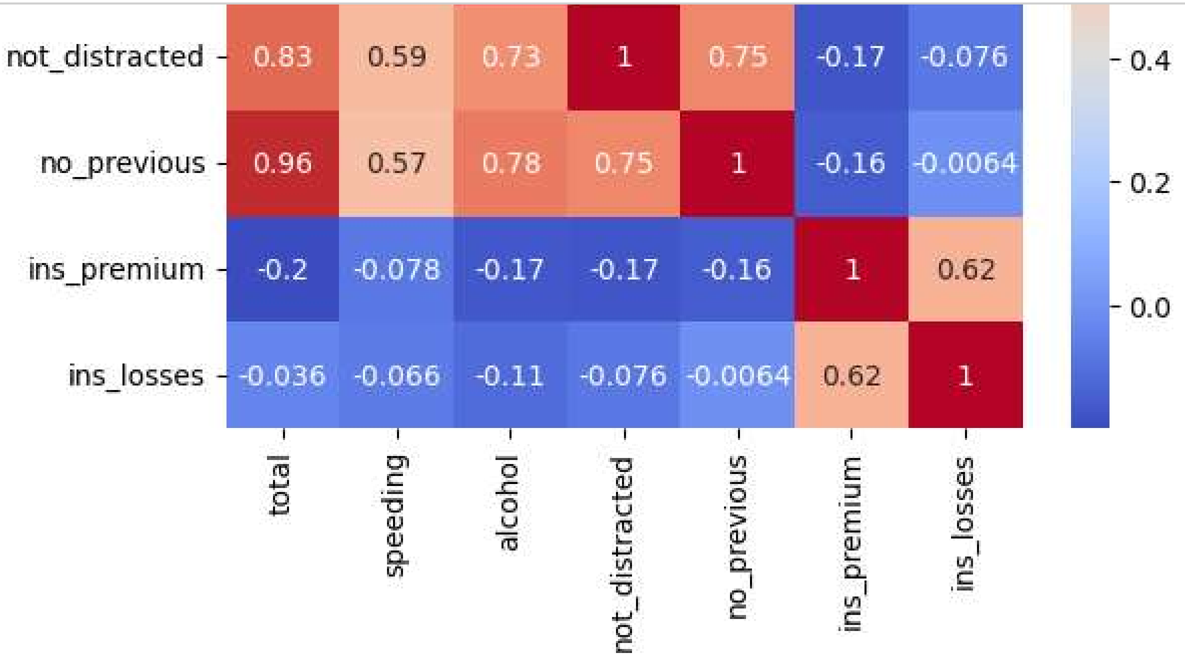
Out[7]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1f2ebc28a30>,
  <matplotlib.lines.Line2D at 0x1f2ebc28cd0>],
 'caps': [<matplotlib.lines.Line2D at 0x1f2ebc28f70>,
  <matplotlib.lines.Line2D at 0x1f2ebc29210>],
 'boxes': [<matplotlib.lines.Line2D at 0x1f2ebc28790>],
 'medians': [<matplotlib.lines.Line2D at 0x1f2ebc294b0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1f2ebc29750>],
 'means': []}
```



```
Inference:
The above graph shows that the outliers in the Alcohol cloumn which are 3values
above the average between 9 and 10
boxplots are used to give the outliers in a given feature of dataset and it shows
the flow of data direction with the horizontal line
```

In [8]:

```
#For correlational analysis going with heatmap
sns.heatmap(corr, annot=True, cmap='coolwarm')
```



	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
not_distracted	0.83	0.59	0.73	1	0.75	-0.17	-0.076
no_previous	0.96	0.57	0.78	0.75	1	-0.16	-0.0064
ins_premium	-0.2	-0.078	-0.17	-0.17	-0.16	1	0.62
ins_losses	-0.036	-0.066	-0.11	-0.076	-0.0064	0.62	1

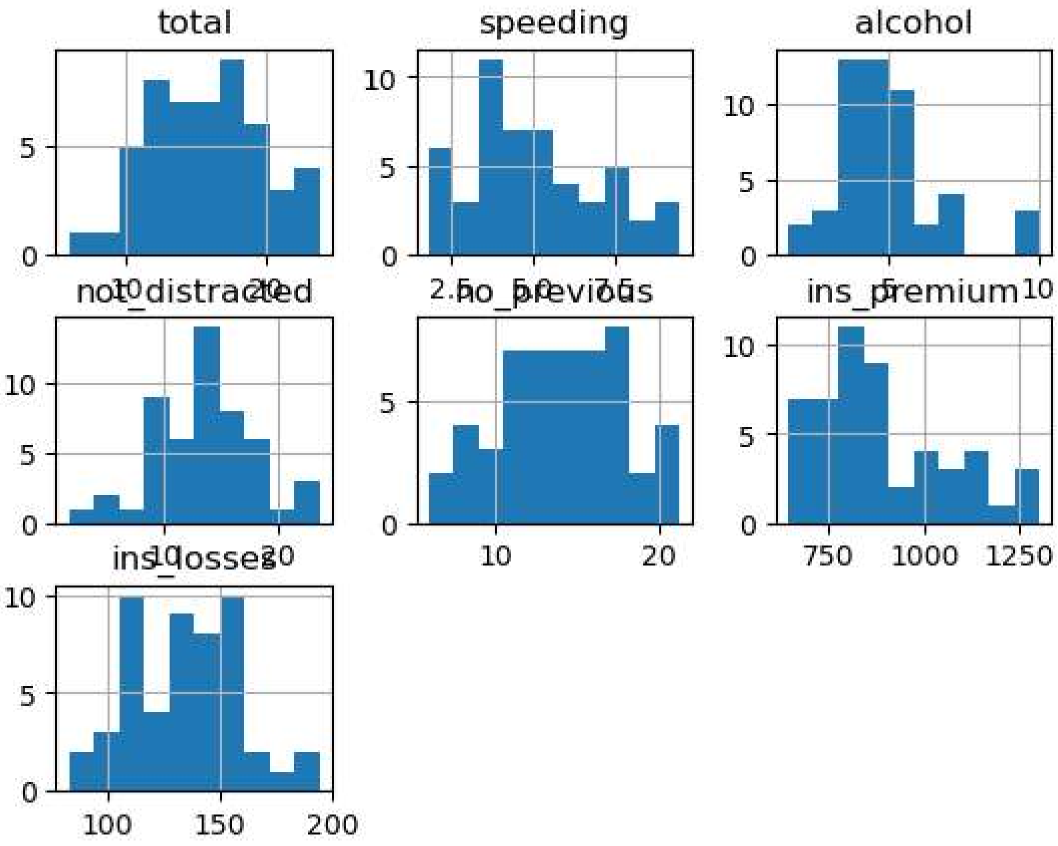
Inference: I have used 'coolwarm' for the color of heatmap and it shows the correlations between each and every variable in the dataset Here the color indicates the strength of that variable among all other features(columns) the no_previous and alcohol have the higher strength.

In [11]:

```
dataset.hist()
```

Out[11]:

```
array([[<Axes: title={'center': 'total'}>,
        <Axes: title={'center': 'speeding'}>,
        <Axes: title={'center': 'alcohol'}>],
       [<Axes: title={'center': 'not_distracted'}>,
        <Axes: title={'center': 'no_previous'}>,
        <Axes: title={'center': 'ins_premium'}>],
       [<Axes: title={'center': 'ins_losses'}>, <Axes: >, <Axes: >]],
      dtype=object)
```



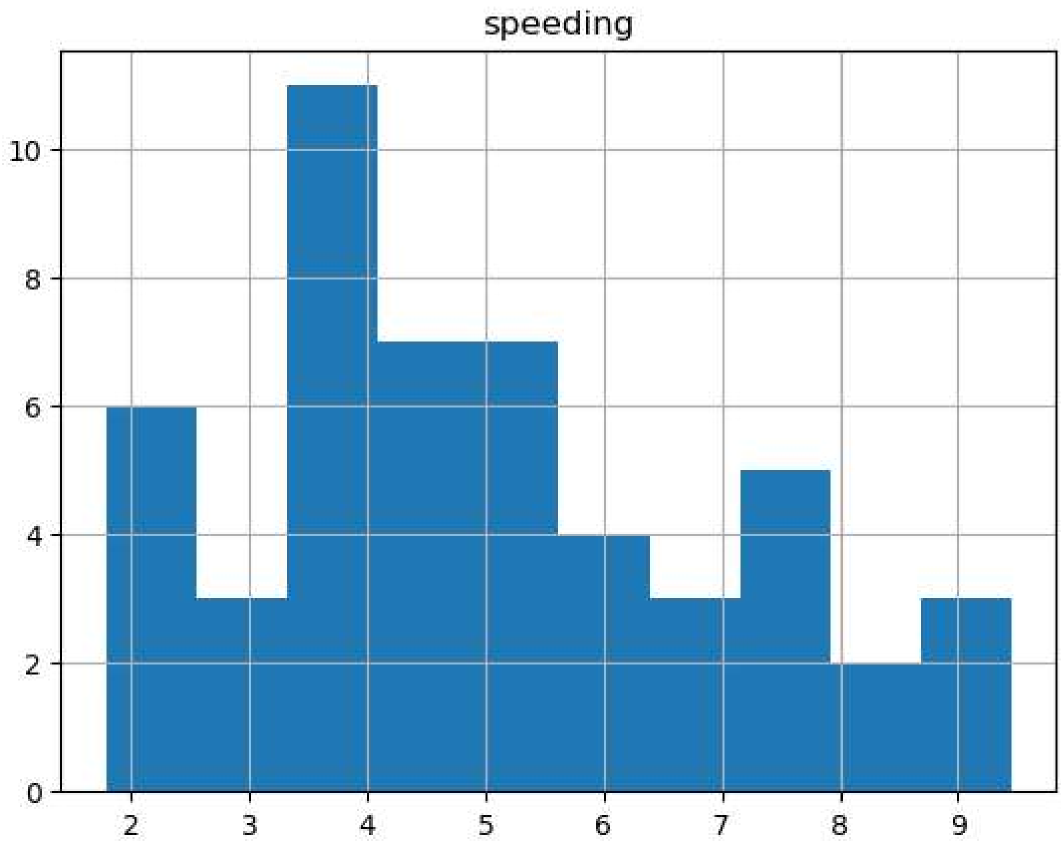
```
Inference:
If we cannot provide the feature name it will return the histogram for every
feature consisting of dataset.And Histogram explains how a the data fuluctuations
in it.
```

In [12]:

```
dataset.hist("speeding")
```

Out[12]:

```
array([[<Axes: title={'center': 'speeding'}>]], dtype=object)
```



```
Inference:

Histogram is looks like bargraph but it not like that it explains about the nature
of the one variable in a particular        dataset like the speeding feature in the
car_crashes data got rised in between the 3.5 to 5.5 at it's maximum levels.
```

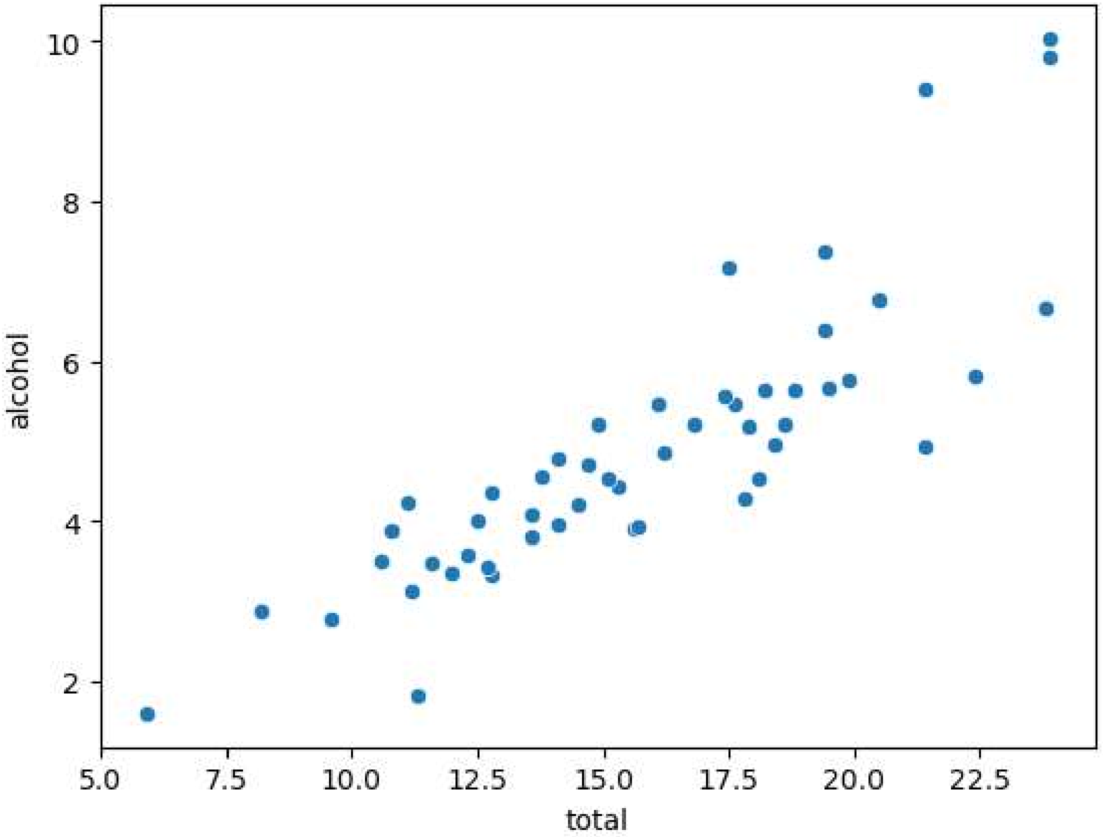
In [13]:

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

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

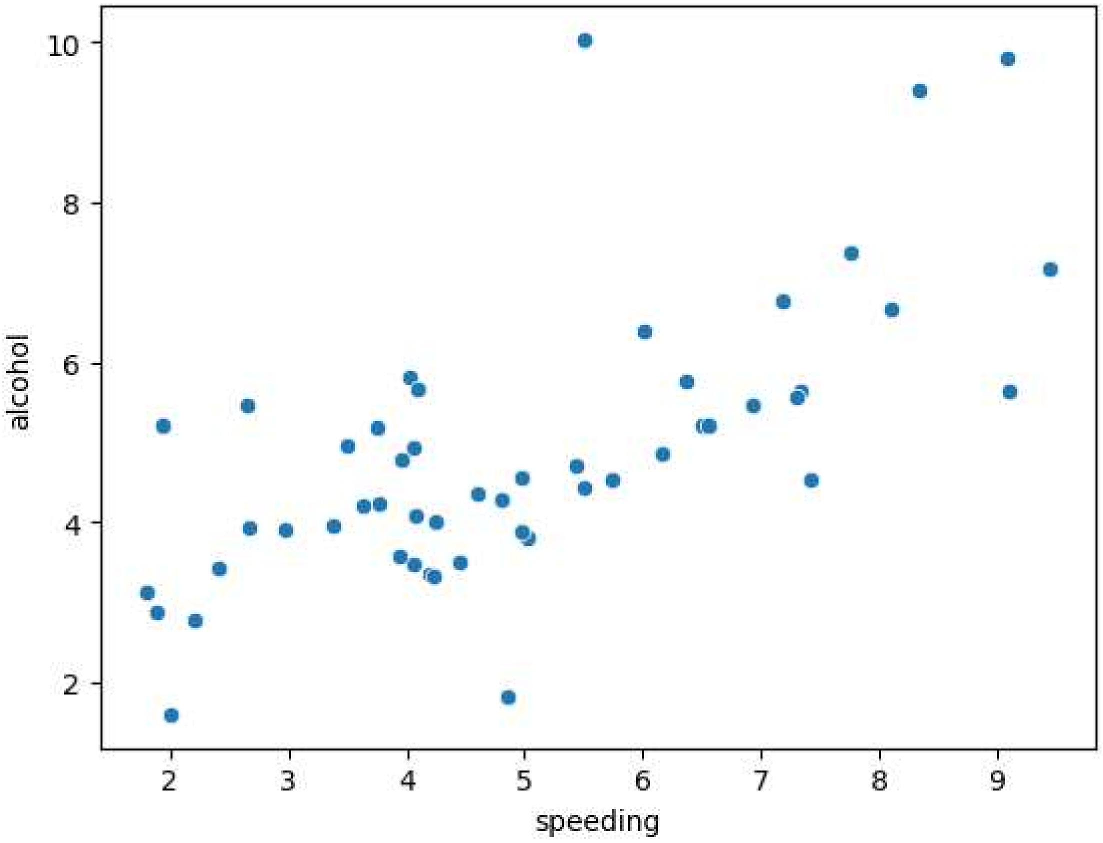


Inference:

It shows that with the rate of increase of total the alcohol levels are also increasing totally it is a positive slope

```
In [15]: sns.scatterplot(x="speeding",y="alcohol",data=dataset)
```

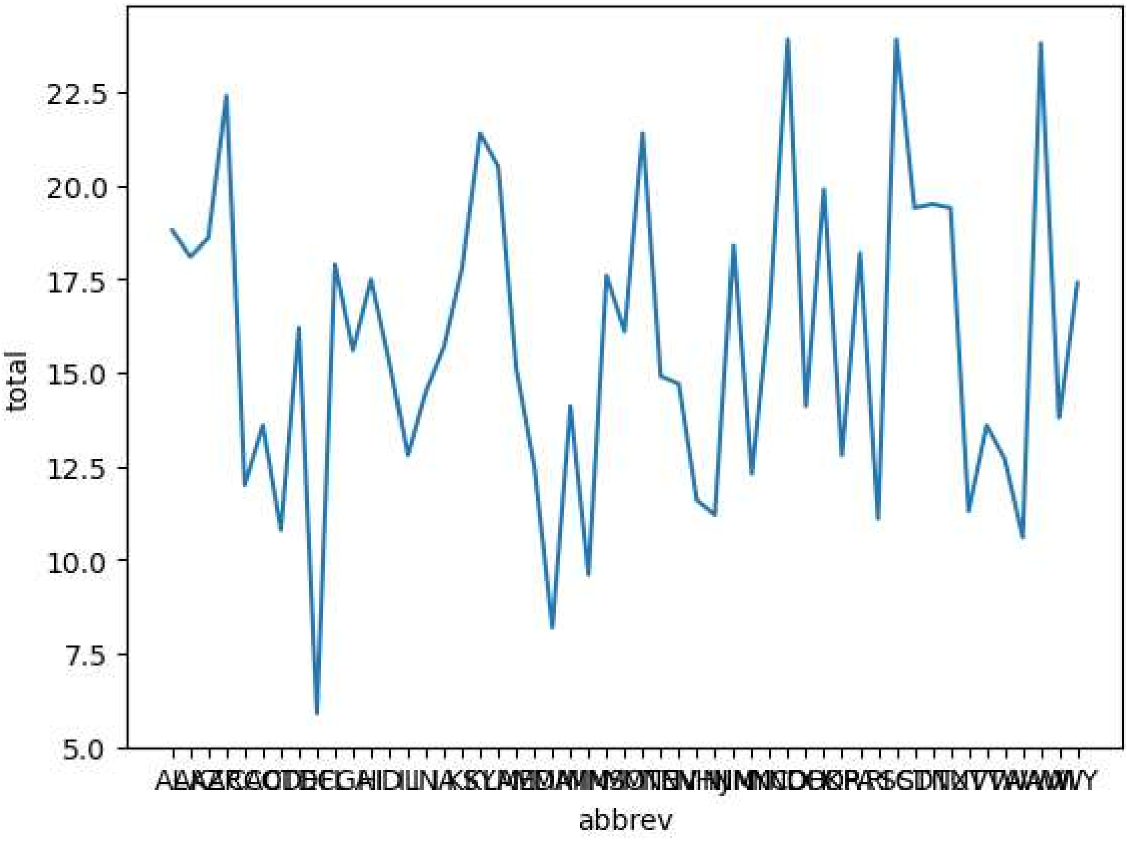
```
Out[15]: <Axes: xlabel='speeding', ylabel='alcohol'>
```



Inference:
As like the past graph it showing positive relation between the two selected variables but the intensity is weak when it compared between the pairs total - alcohol and speeding - alcohol Here there exists some outliers in the above graph

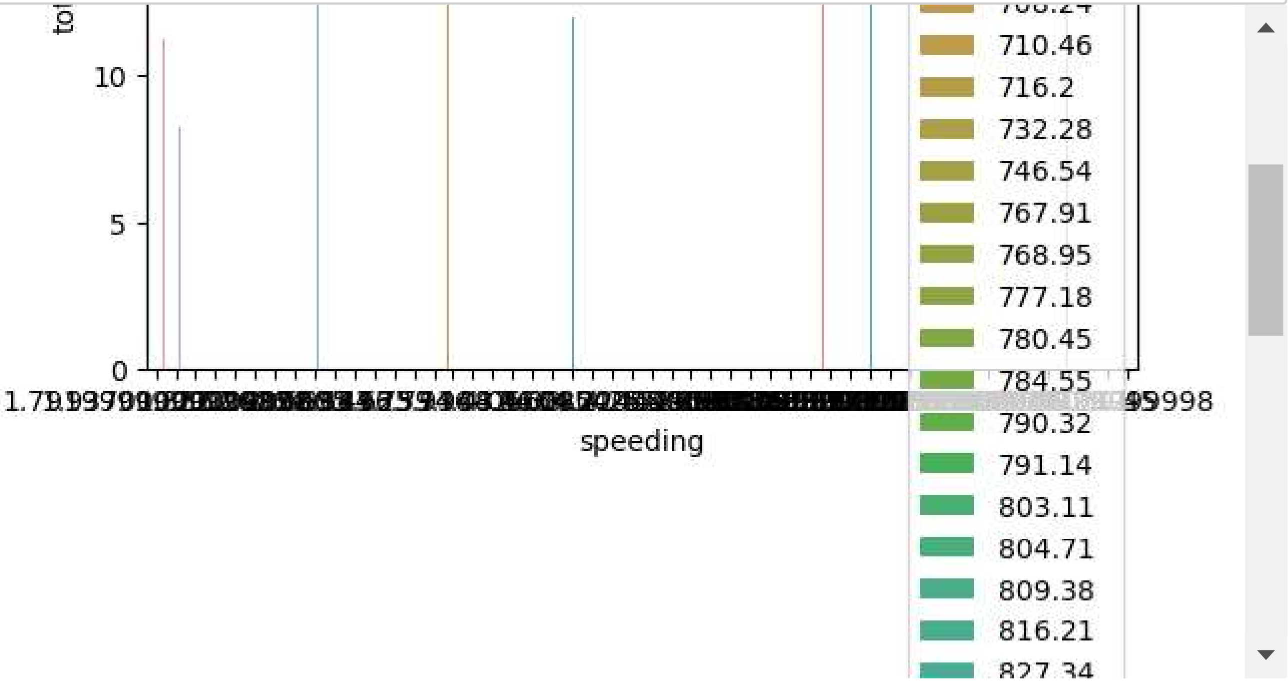
In [16]:
```
sns.lineplot(x="abbrev",y="total",data=dataset,errorbar=None)
```

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



```
It gives the lineplot for abbrev and total features in the car_crashes dataset it
shows the trend of the two features a way that starts at high level and now comes
to down and down and got rised.
```

In [17]:
```
sns.barplot(data=dataset,x="speeding",y="total",hue="ins_premium")
```

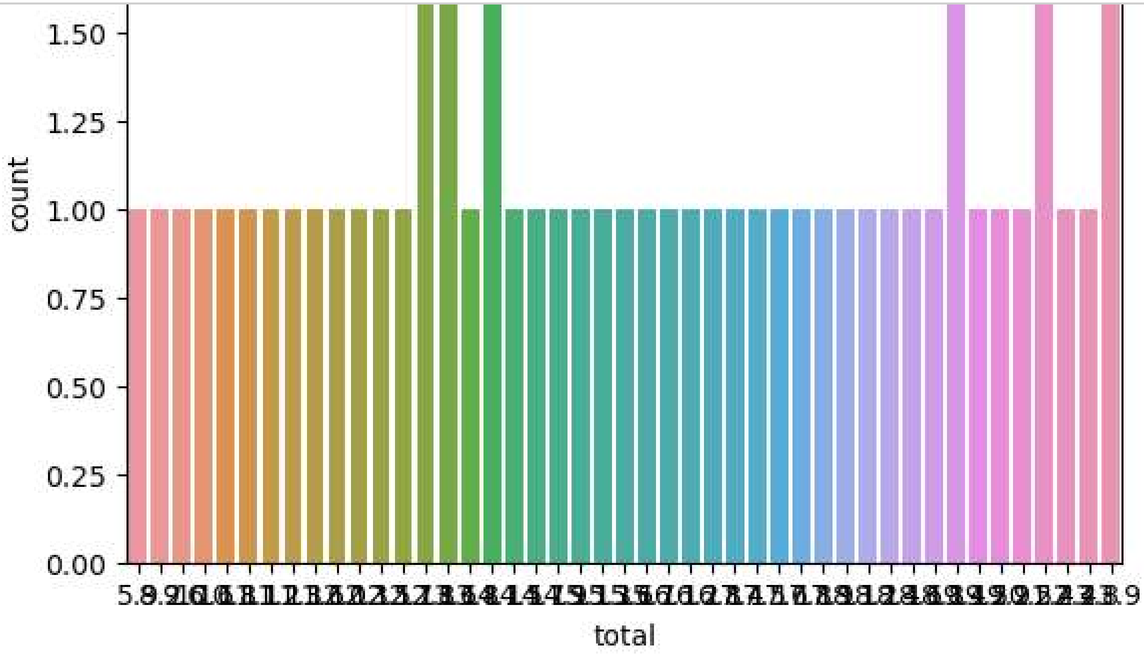


```
Inference:
```

Based upon the variable ins_premium the two variables speeding and total are got compared and it appears in different colors to get to know the values and effect of ins_premium on the other two.

In [18]:

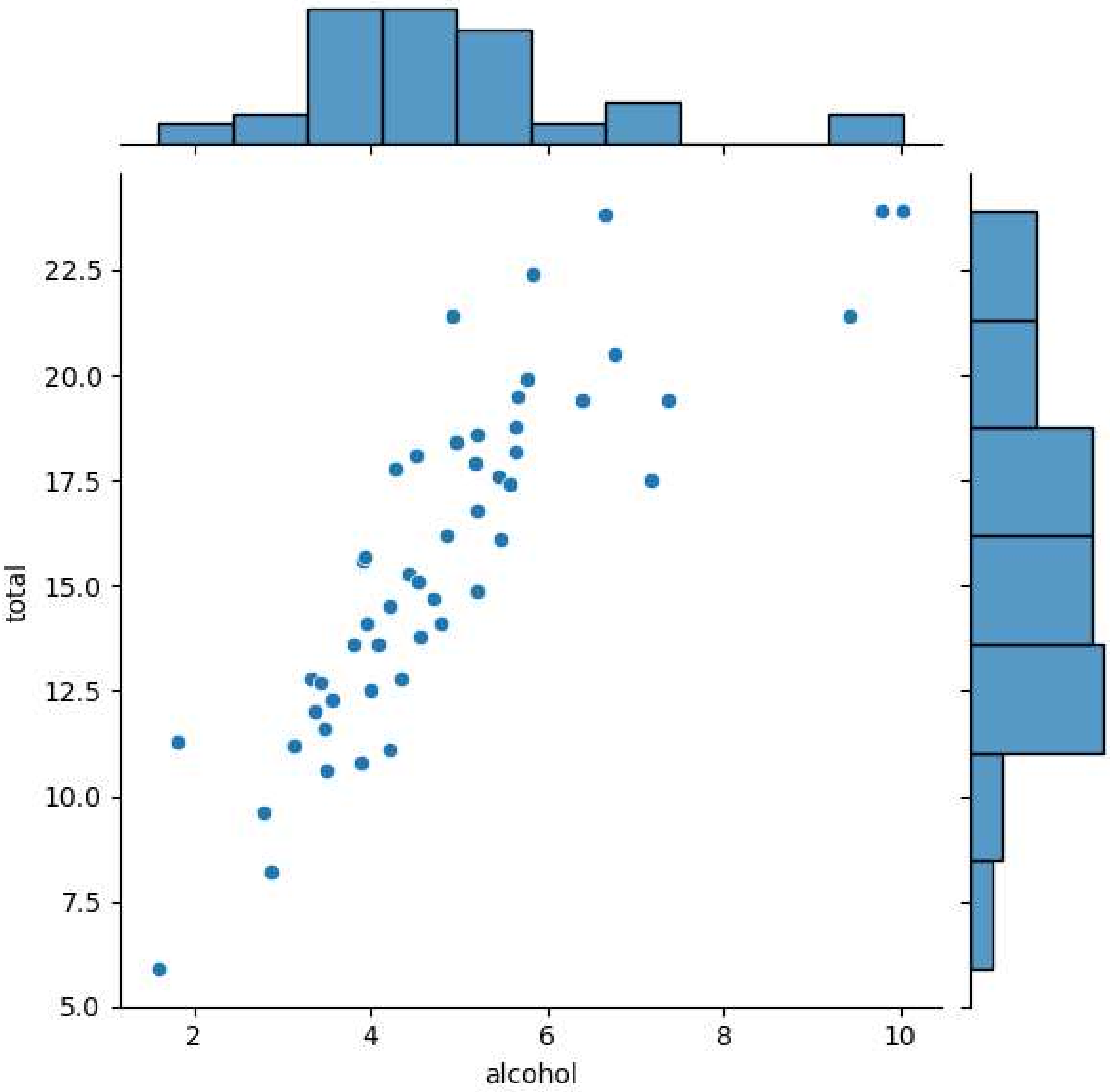
```
sns.countplot(x="total",data=dataset)
```



Inference:
It is countplot according to the count it given the frequency map

```
In [19]: sns.jointplot(x="alcohol",y="total",data=dataset)
```

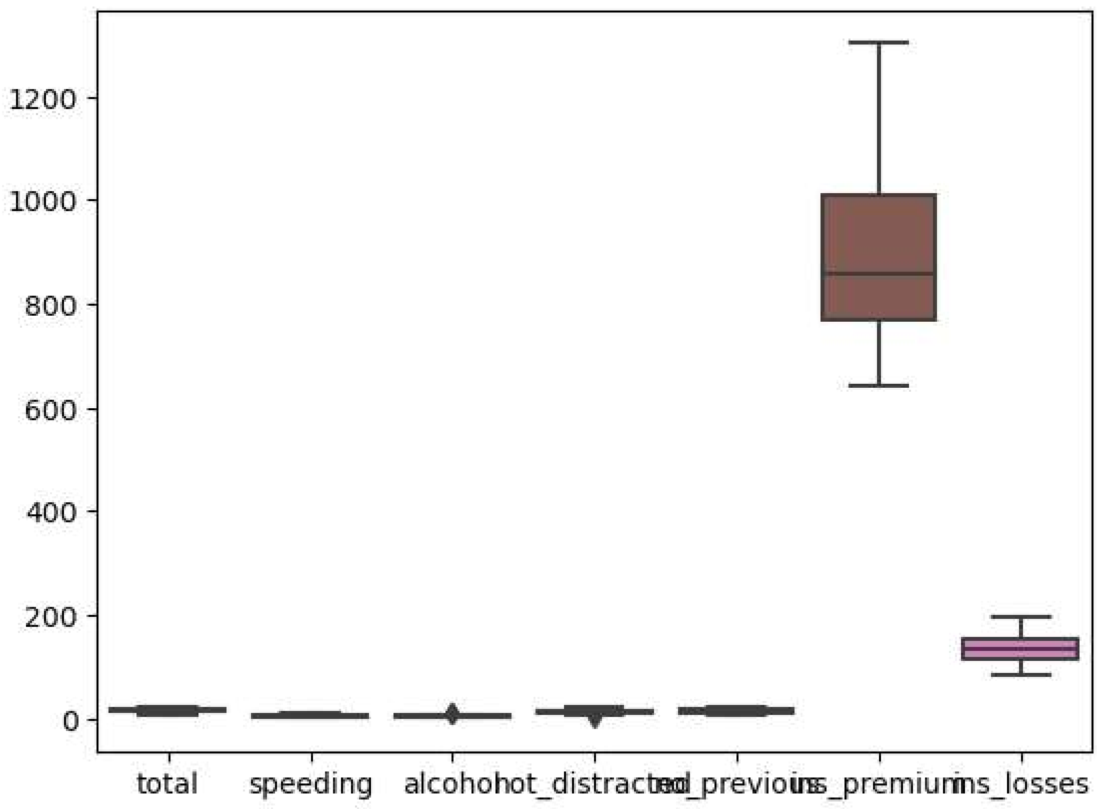
```
Out[19]: <seaborn.axisgrid.JointGrid at 0x1f2ecef3be0>
```



```
Infernece:
it shows the how two variables are interacting like total and alcohol with help of
dot and hist model visualizations
```

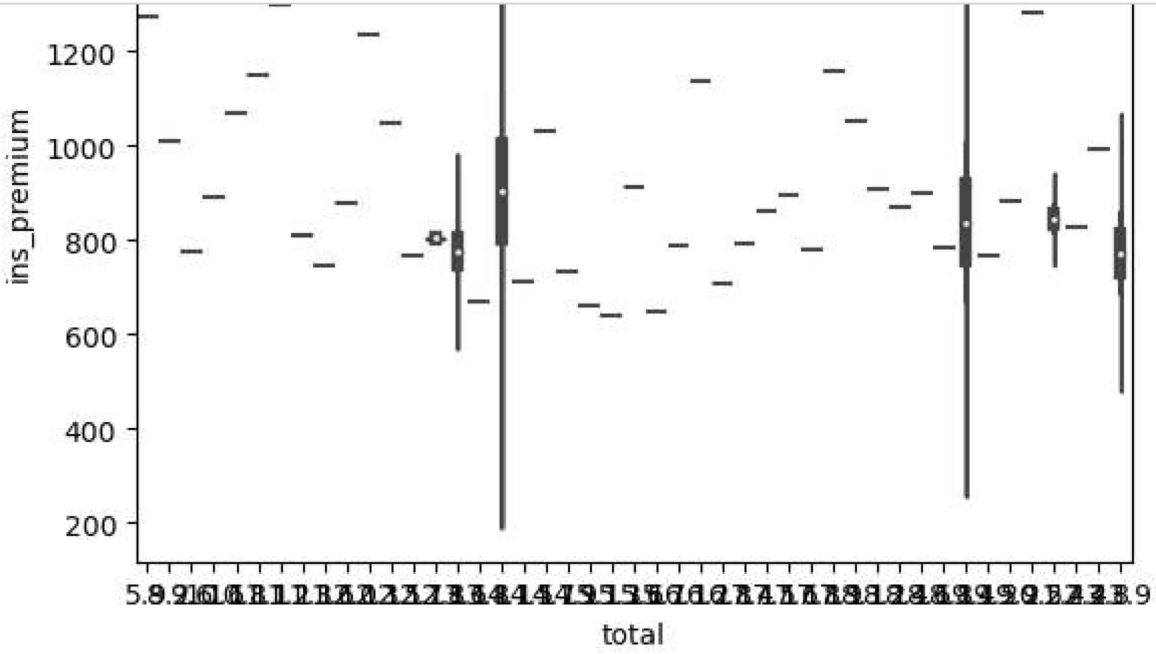
In [20]:
```
sns.boxplot(data = dataset)
```

Out[20]:
```
<Axes: >
```



```
Inference:
the lastone which is ins_perimum values are at high level in the sense of units so
that's it it is at the top when we compared towards the other.
```

In [22]:
```
sns.violinplot(x = "total", y= "ins_premium", data = dataset)
```



```
Inference:
Each violin plot shows the distribution of "total" values for a specific category
of "ins_premium."
```

The width of the violin plot represents the density of the data at different "total" values. Wider sections indicate higher data density, and narrower sections indicate lower density.

In [ ]: