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

```
print(sns.get_dataset_names())
```

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes', 'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri', 'g
```

```
df = sns.load_dataset('car_crashes')
print(df)
```

```
46  12.7   2.413   3.429      11.049     11.176     768.95
47  10.6   4.452   3.498       8.692      9.116     890.03
48  23.8   8.092   6.664      23.086     20.706     992.61
49  13.8   4.968   4.554       5.382     11.592     670.31
50  17.4   7.308   5.568      14.094     15.660     791.14

    ins_losses abbrev
0       145.08     AL
1       133.93     AK
2       110.35     AZ
3       142.39     AR
4       165.63     CA
5       139.91     CO
6       167.02     CT
7       151.48     DE
8       136.05     DC
9       144.18     FL
10      142.80     GA
11      120.92     HI
12       82.75     ID
13      139.15     IL
14      108.92     IN
15      114.47     IA
16      133.80     KS
17      137.13     KY
18      194.78     LA
19       96.57     ME
20      192.70     MD
21      135.63     MA
22      152.26     MI
23      133.35     MN
24      155.77     MS
25      144.45     MO
26       85.15     MT
27      114.82     NE
28      138.71     NV
29      120.21     NH
30      159.85     NJ
31      120.75     NM
32      150.01     NY
33      127.82     NC
34      109.72     ND
35      133.52     OH
36      178.86     OK
37      104.61     OR
38      153.86     PA
39      148.58     RI
40      116.29     SC
41       96.87     SD
42      155.57     TN
43      156.83     TX
44      109.48     UT
45      109.61     VT
46      153.72     VA
47      111.62     WA
48      152.56     WV
49      106.62     WI
50      122.04     WY
```

```
sns.__version__
```

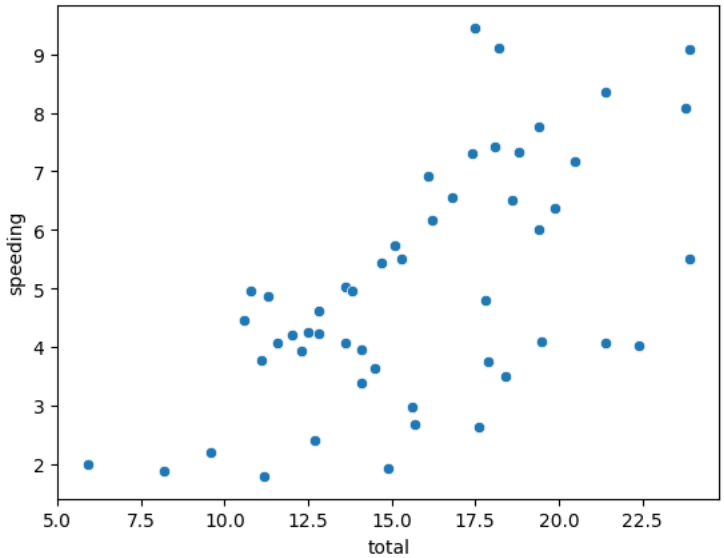
```
'0.12.2'
```

```
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
```

```
sns.scatterplot(x="total",y="speeding",data=df)
```

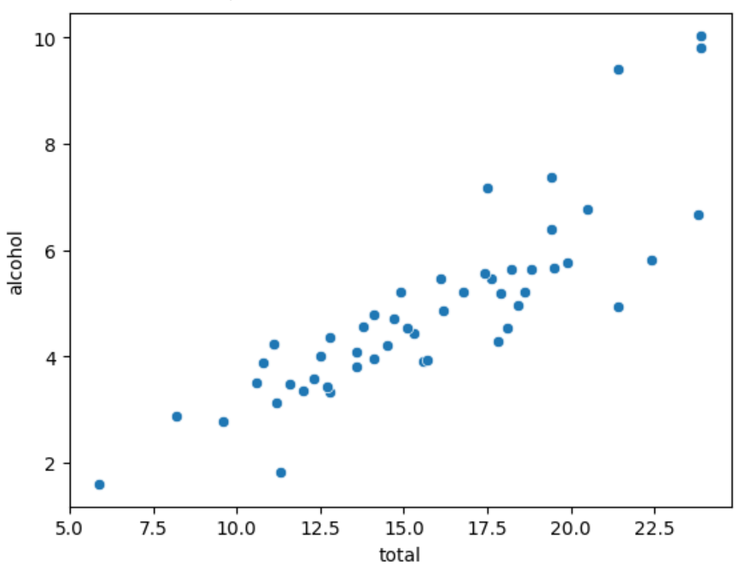
```
<Axes: xlabel='total', ylabel='speeding'>
```



Inference: In the above graph we can observe that plotting of the points with total in x-axis and speeding in y-axis. Here we can clearly observe that increase in speed has caused many car crashes. Which shows the direct proportionality in between total and speeding of car crashes.

```
sns.scatterplot(x="total",y="alcohol",data=df)
```

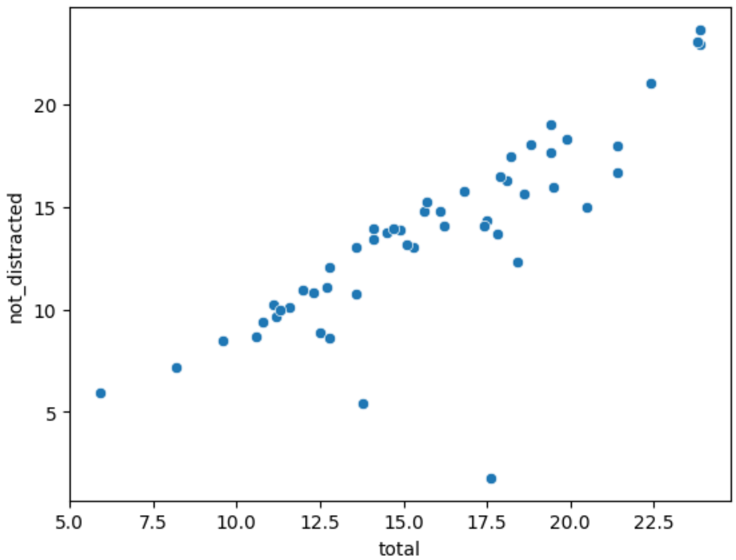
```
<Axes: xlabel='total', ylabel='alcohol'>
```



Inference: In the above graph we can observe that plotting of the points with total in x-axis and alcohol in y-axis. Here we can clearly observe that more drinking of alcohol has caused many car crashes. Which shows the direct proportionality in between total and alcohol of car crashes.

```
sns.scatterplot(x="total",y="not_distracted",data=df)
```

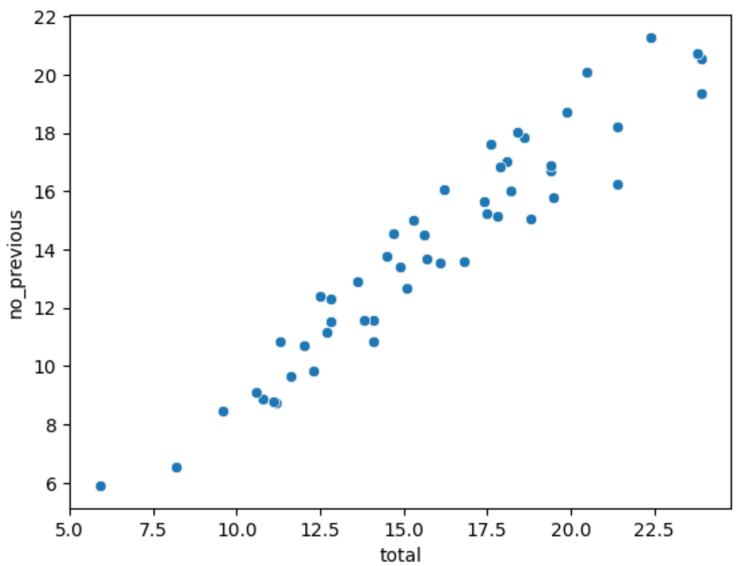
<Axes: xlabel='total', ylabel='not_distracted'>



Inference: In the above graph we can observe that plotting of the points with total in x-axis and not_distracted in y-axis. Here we can clearly observe that not_distracted has caused many car crashes. Which shows the direct proportionality in between total and not_distracted of car crashes.

```
sns.scatterplot(x="total",y="no_previous",data=df)
```

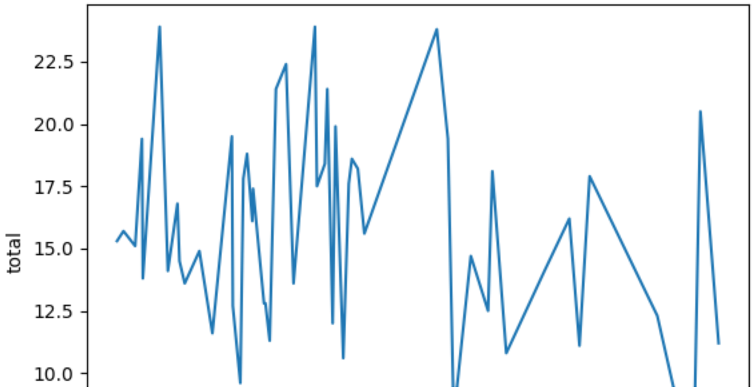
<Axes: xlabel='total', ylabel='no_previous'>



Inference: In the above graph we can observe that plotting of the points with total in x-axis and no_previous in y-axis. Here we can clearly observe that no_previous has caused many car crashes. Which shows the direct proportionality in between total and no_previous of car crashes.

```
sns.lineplot(x="ins_premium",y="total",data=df,ci=None)
```

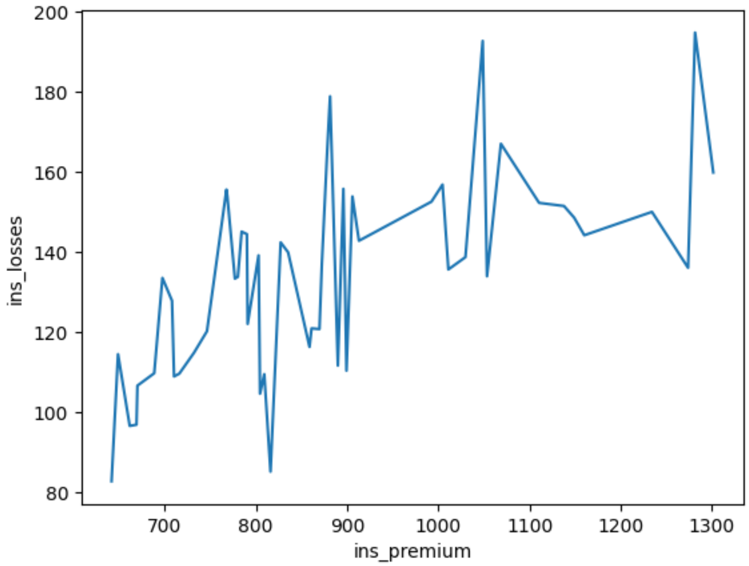
```
<ipython-input-18-eca919d1edb3>:1: FutureWarning:

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

  sns.lineplot(x="ins_premium",y="total",data=df,ci=None)
<Axes: xlabel='ins_premium', ylabel='total'>
```



Inference: In the above lineplot graph the relation between ins_premium in x=asix and total in y-asix. There is an irregular plotting of lineplot of ins_premium and total. which defines that very large number of car crash has been done their insurance premium.

```
sns.lineplot(x="ins_premium",y="ins_losses",data=df)
```

```
<Axes: xlabel='ins_premium', ylabel='ins_losses'>
```



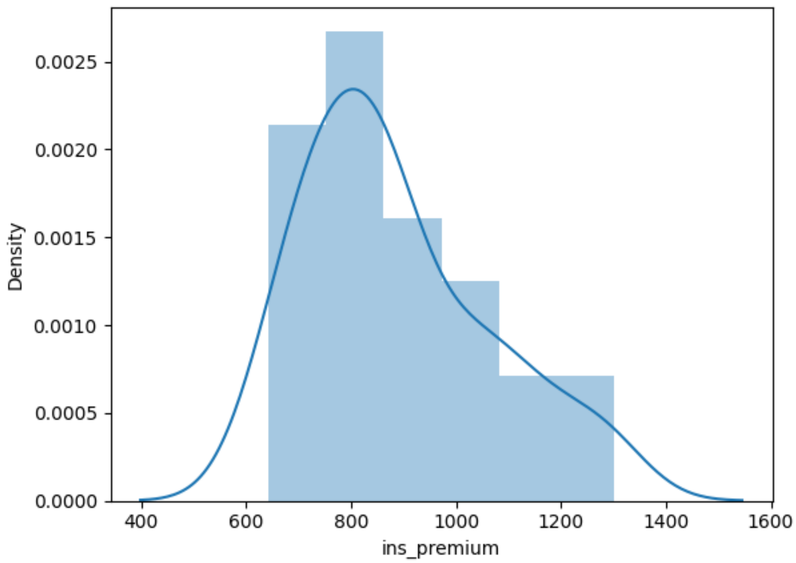

```
sns.barplot(data=df,x="ins_premium",y="ins_losses")
```

```
<Axes: xlabel='ins_premium', ylabel='ins_losses'>
```

Inference: In the above two graph the relation between ins_premium in x=asix and ins_losses in y-asix. There is an irregular plotting of lineplot of ins_premium and ins_losses. which defines that insurance losses is directly proportional to insurance premium. where increse in insurance premium also tends to increase in insurance loss.

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

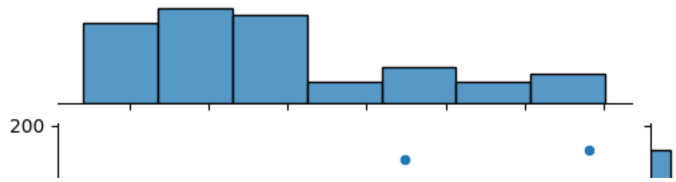
```
<ipython-input-4-2616b6ce983b>:1: UserWarning:

`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'])
<Axes: xlabel='ins_premium', ylabel='Density'>
```



Inference: The above graph is about the density of the insurance premium where there is a huge increase in the density of insurance premium at a point of 800.

```
sns.jointplot(data=df,x="ins_premium",y="ins_losses")
```

```
<seaborn.axisgrid.JointGrid at 0x7d1d5bfbbe50>
```



Inference: The above graph is about jointplot of ins_premium and ins_losses where the plotting of graph is randomly distributed through out the graph. We can observe the density of ins_premium and ins_losses where they show in great increase and decrease.

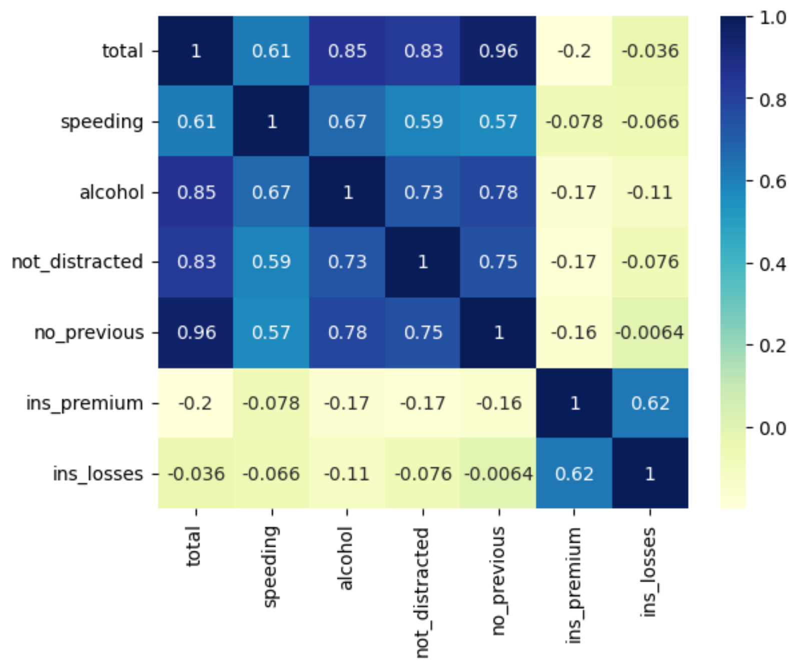
```
corr=df.corr()
corr
```

```
<ipython-input-22-7d5195e2bf4d>:1: FutureWarning: The default value of numeric_only in D
  corr=df.corr()
```

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

```
sns.heatmap(corr,annot=True,cmap="YlGnBu")
```

```
<Axes: >
```



Inferance: The above data is the correlation of the car crashs data. Where we can observe the clear correaltion between each and every info of the car crashs data.

✓ 1s completed at 9:59 AM