CAR CRASHES

ASSIGNMENT-2

NAME :- goli revanth krishna_ redg no: 21bce7852_ email: revanth.21bce7852@vitapstudent.ac.in.

Importing Libraries

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

Load the dataset

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

	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

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

32	12.3	3.936	3.567	10.824	9.840	1234.31	150.01	NY

```
df.head(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
40	23.9	9.082	9.799	22.944	19.359	858.97	116.29	SC

```
df.tail()
```

	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

```
df.shape
```

```
(51, 8)
```

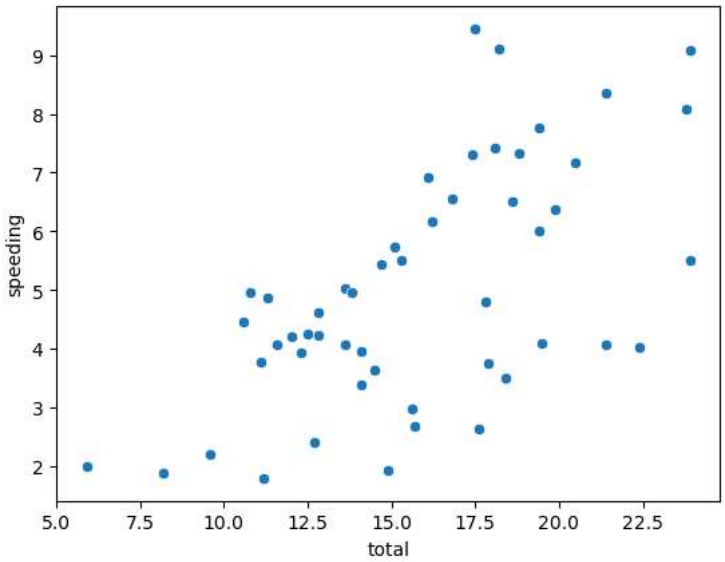
```
df.describe()
```

	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

1.Scatterplot

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

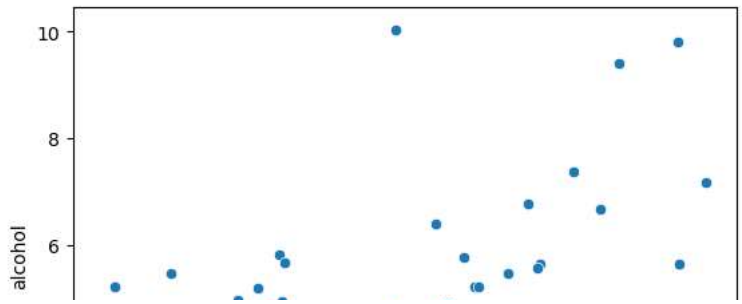
```
<Axes: xlabel='total', ylabel='speeding'>
```



Inference : The plot labels between total and speeding. Here total increases with respective to speeding

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

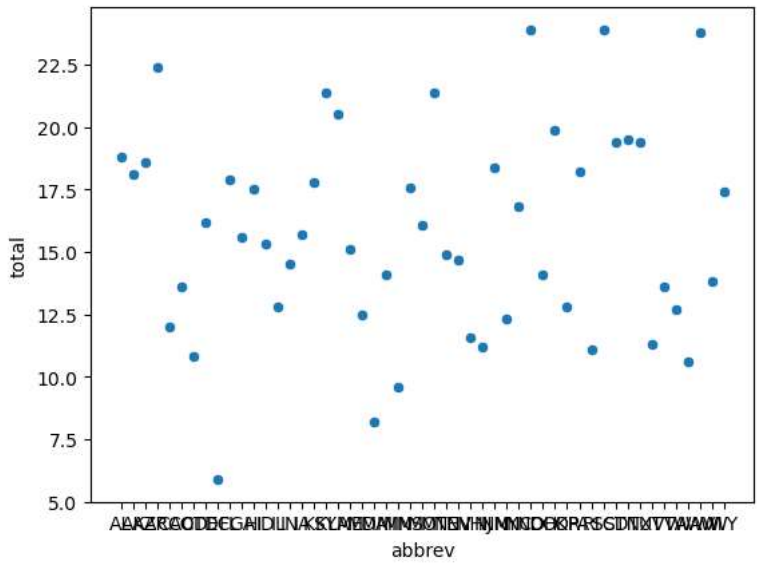
```
<Axes: xlabel='speeding', ylabel='alcohol'>
```



Inference : The plot labels between speeding and alcohol. Here speeding increases with respective to alcohol

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

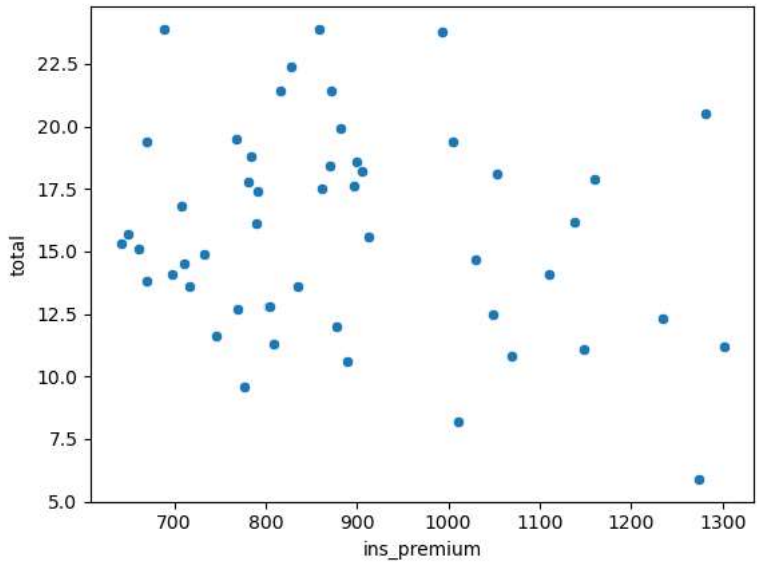
```
<Axes: xlabel='abbrev', ylabel='total'>
```



Inference : The plot labels between abbrev and total. Here abbrev increases with respective to total

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

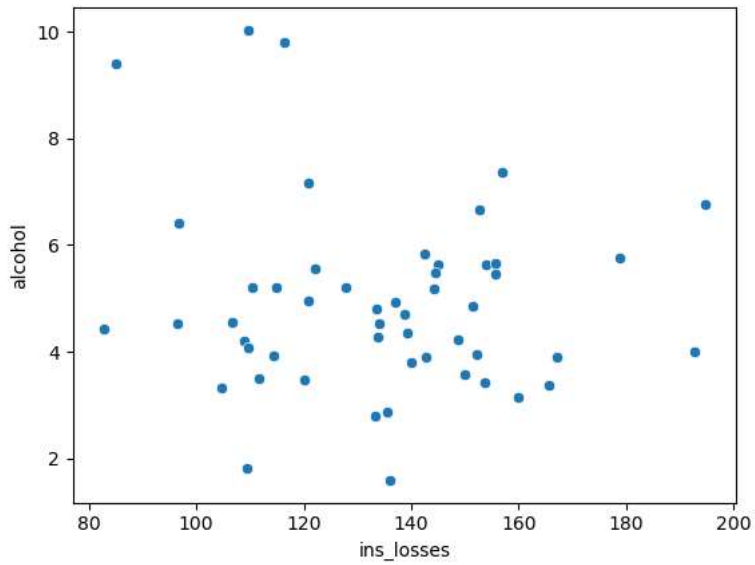
```
<Axes: xlabel='ins_premium', ylabel='total'>
```



Inference : The plot labels between ins_premium and total. Here ins_premium increases with respective to total

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

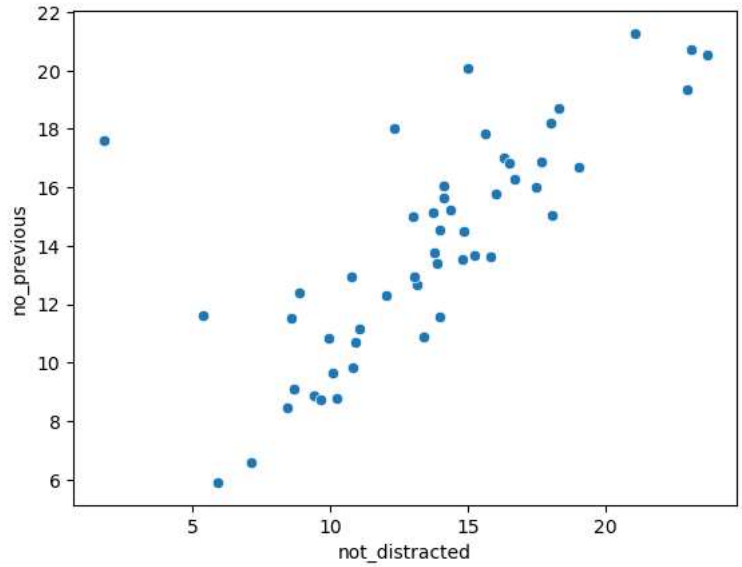
```
<Axes: xlabel='ins_losses', ylabel='alcohol'>
```



Inference : The plot labels between ins_losses and alcohol. Here ins_losses increases with respective to alcohol

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

```
<Axes: xlabel='not_distracted', ylabel='no_previous'>
```

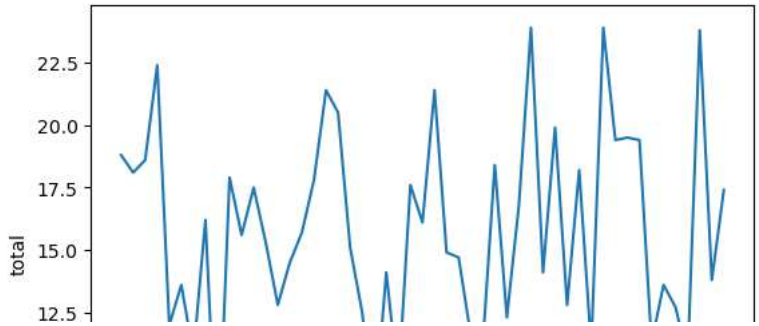


Inference : The plot labels between not_distracted and no_previous. Here not_distracted increases with respective to no_previous

2.Line plot

```
sns.lineplot(x="abbrev",y="total",data=df)
```

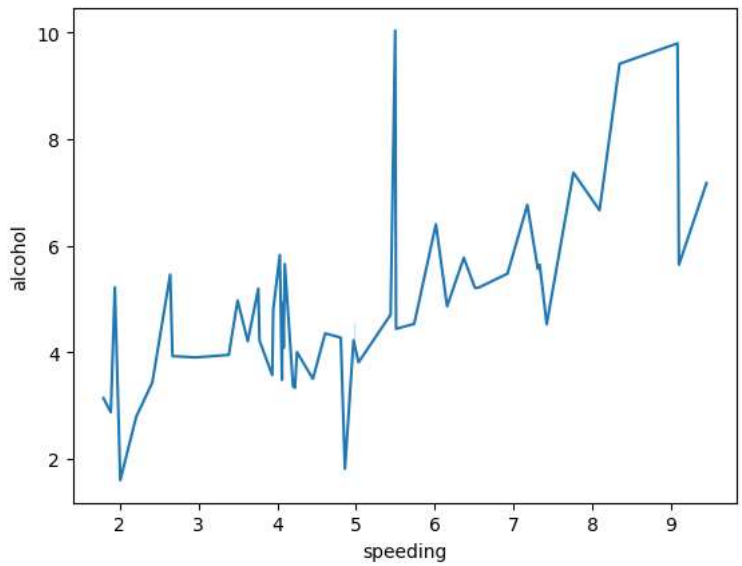
```
<Axes: xlabel='abbrev', ylabel='total'>
```



Inference : The plot labels between abbrev and total. Here the plot is linear

```
sns.lineplot(x="speeding",y="alcohol",data=df)
```

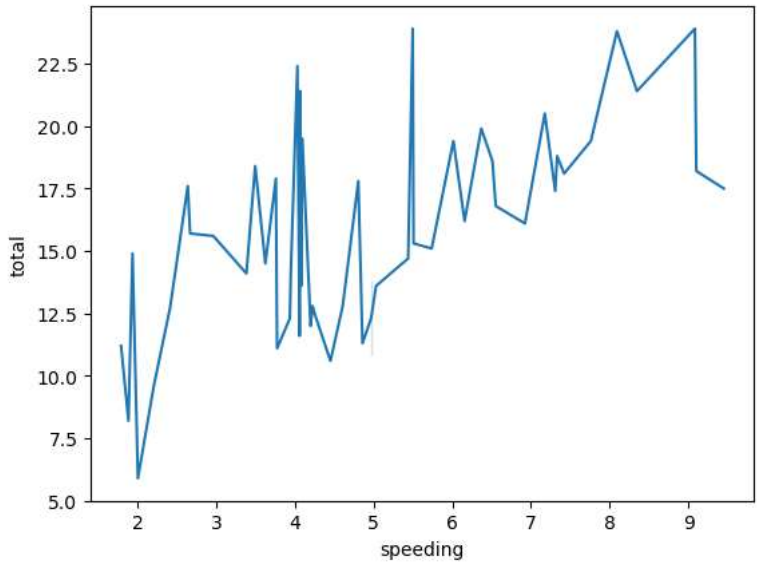
```
<Axes: xlabel='speeding', ylabel='alcohol'>
```



Inference : The plot labels between speeding and alcohol. Here the plot is linear

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

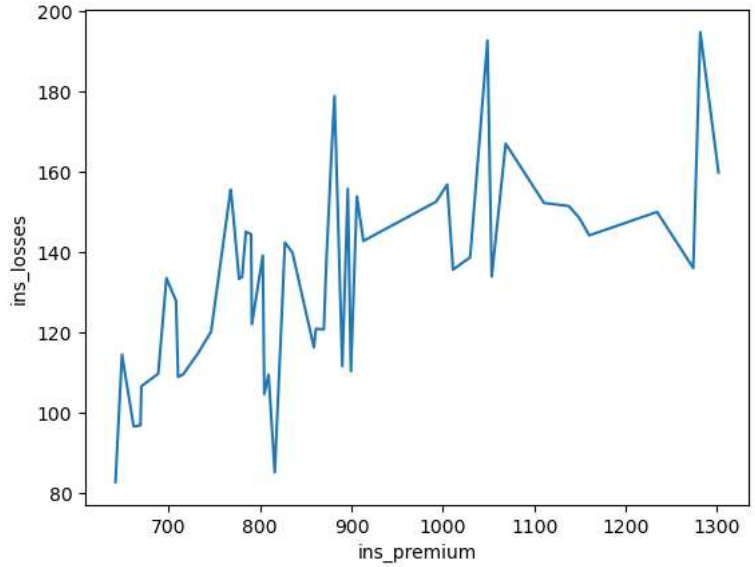
```
<Axes: xlabel='speeding', ylabel='total'>
```



Inference : The plot labels between speeding and alcohol. Here the plot is linear

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

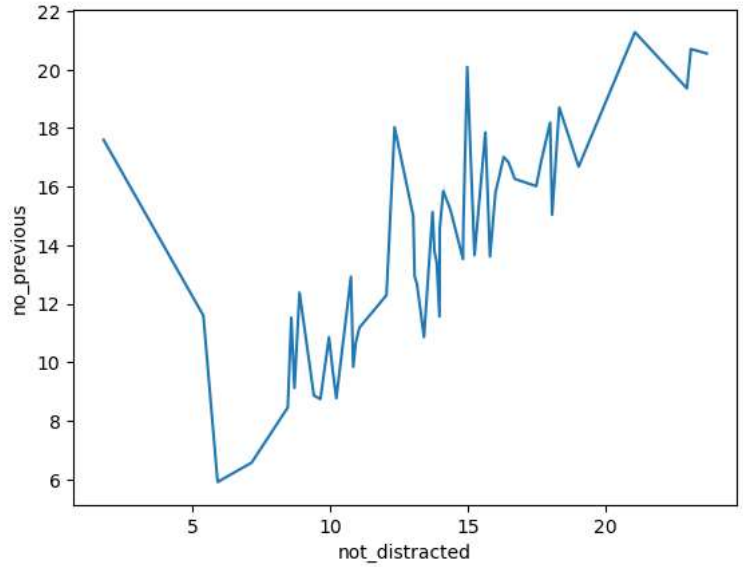
```
<Axes: xlabel='ins_premium', ylabel='ins_losses'>
```



Inference : The plot labels between speeding and alcohol. Here the plot is linear

```
sns.lineplot(x="not_distracted",y="no_previous",data=df)
```

```
<Axes: xlabel='not_distracted', ylabel='no_previous'>
```

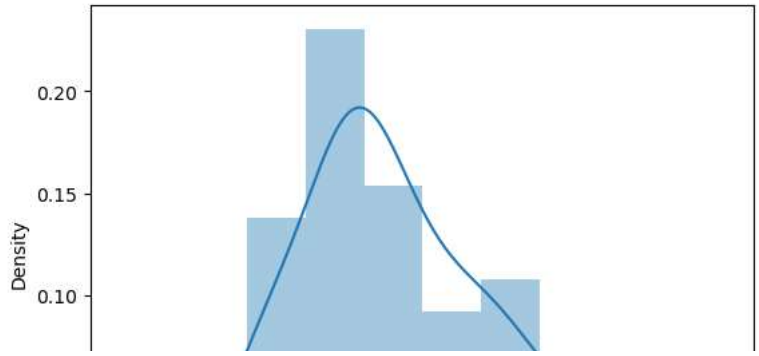


Inference : The plot labels between not_distracted and no_previous. Here the plot is not linear

3.Distribution Plot

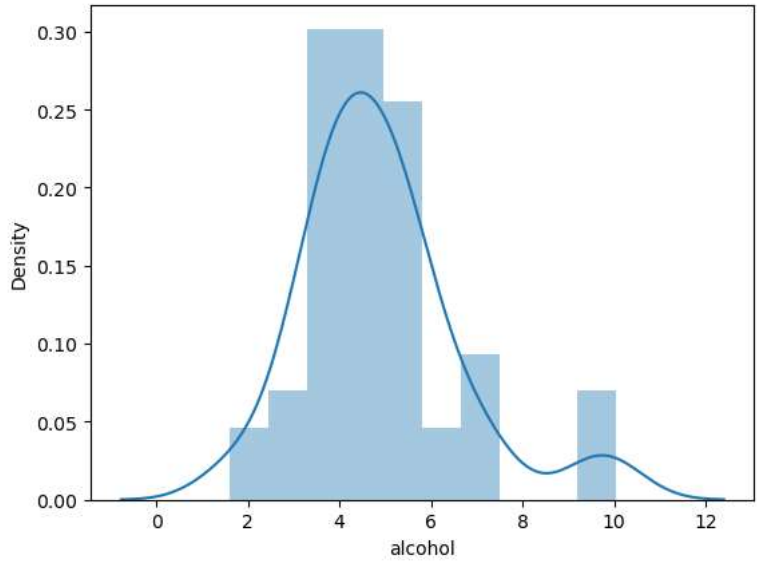
```
sns.distplot(df["speeding"])
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_16988\2127910581.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

  sns.distplot(df["speeding"])
<Axes: xlabel='speeding', ylabel='Density'>
```



Inference : From the distplot displays a histogram overlaid with the density curve of speeding

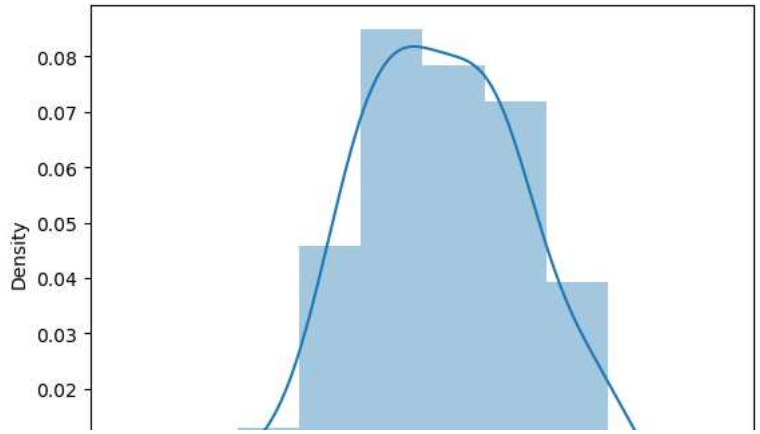
```
sns.distplot(df["alcohol"])

C:\Users\Asus\AppData\Local\Temp\ipykernel_16988\3201832786.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

  sns.distplot(df["alcohol"])
<Axes: xlabel='alcohol', ylabel='Density'>
```



Inference : From the distplot displays a histogram overlaid with the density curve of alcohol

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

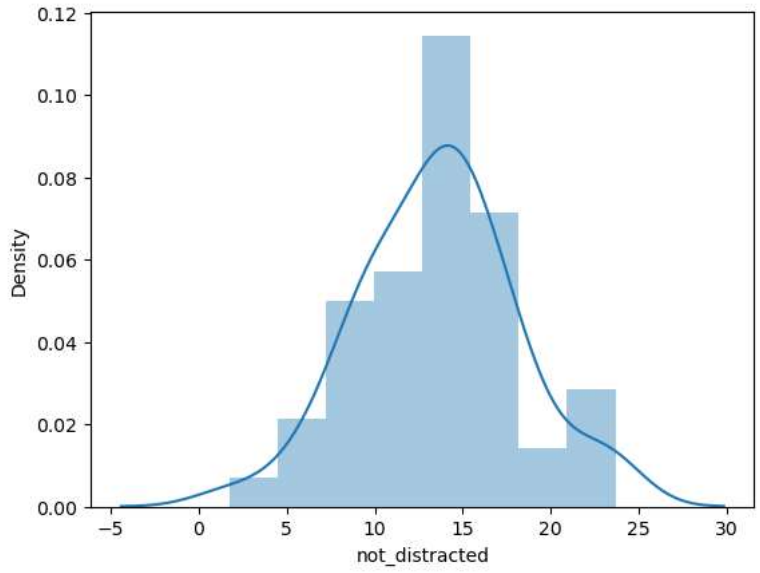
```
C:\Users\Asus\AppData\Local\Temp\ipykernel_16988\1102674835.py:1: UserWarning:

`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

  sns.distplot(df["total"])
<Axes: xlabel='total', ylabel='Density'>
```



Inference : From the distplot displays a histogram overlaid with the density curve of total

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

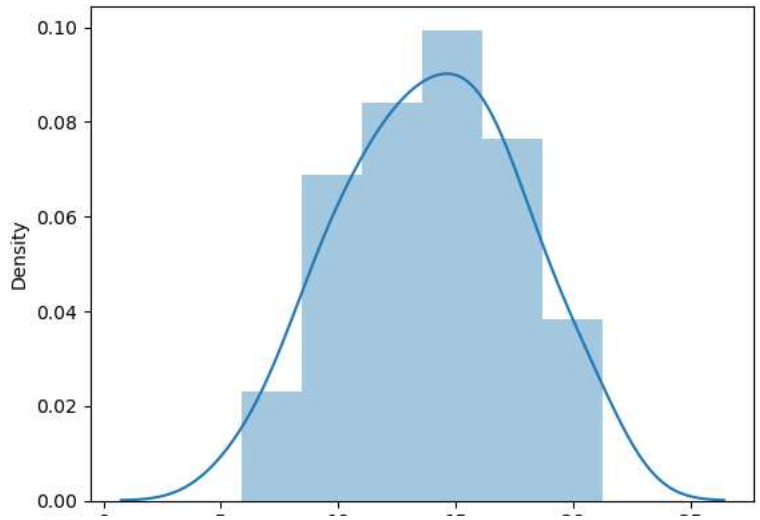
```
C:\Users\Asus\AppData\Local\Temp\ipykernel_16988\743952441.py:1: UserWarning:

`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

  sns.distplot(df["not_distracted"])
<Axes: xlabel='not_distracted', ylabel='Density'>
```



Inference : From the distplot displays a histogram overlaid with the density curve of not_distracted

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

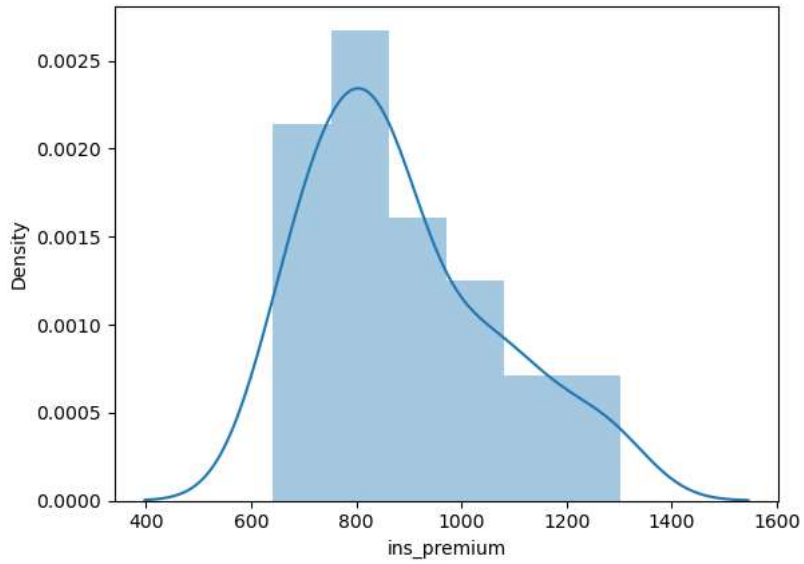
```
C:\Users\Asus\AppData\Local\Temp\ipykernel_16988\1806622040.py:1: UserWarning:

`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

  sns.distplot(df["no_previous"])
<Axes: xlabel='no_previous', ylabel='Density'>
```



Inference : From the distplot displays a histogram overlaid with the density curve of not_distracted

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

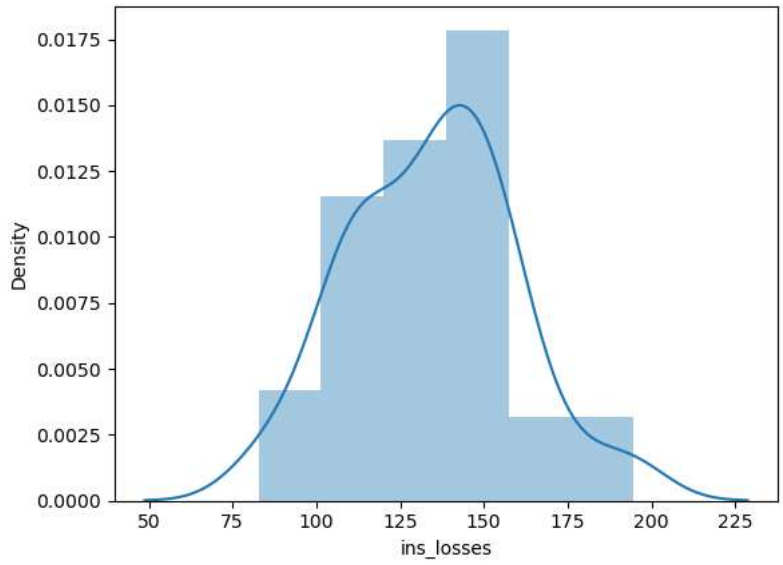
```
C:\Users\Asus\AppData\Local\Temp\ipykernel_16988\3028344525.py:1: UserWarning:

`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

  sns.distplot(df["ins_premium"])
<Axes: xlabel='ins_premium', ylabel='Density'>
```



Inference : From the distplot displays a histogram overlaid with the density curve of ins_premium

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

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_16988\4078262684.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

  sns.distplot(df["ins_losses"])
<Axes: xlabel='ins_losses', ylabel='Density'>
```

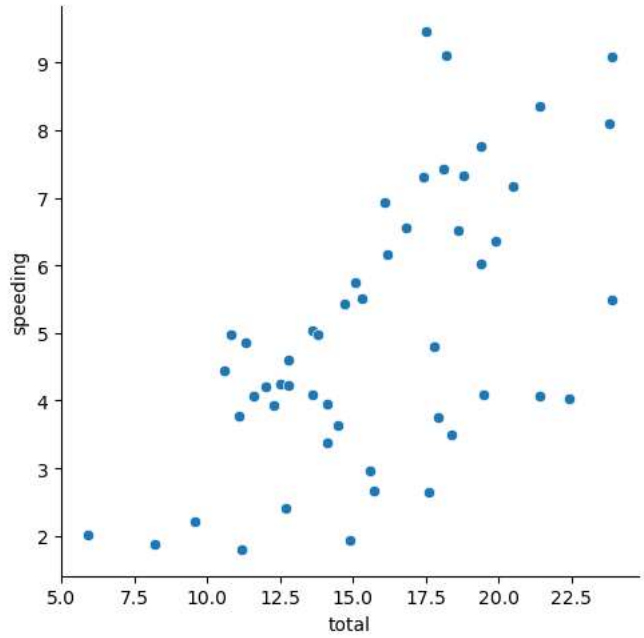


Inference : From the distplot displays a histogram overlaid with the density curve of ins_losses

4.Relational Plot

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

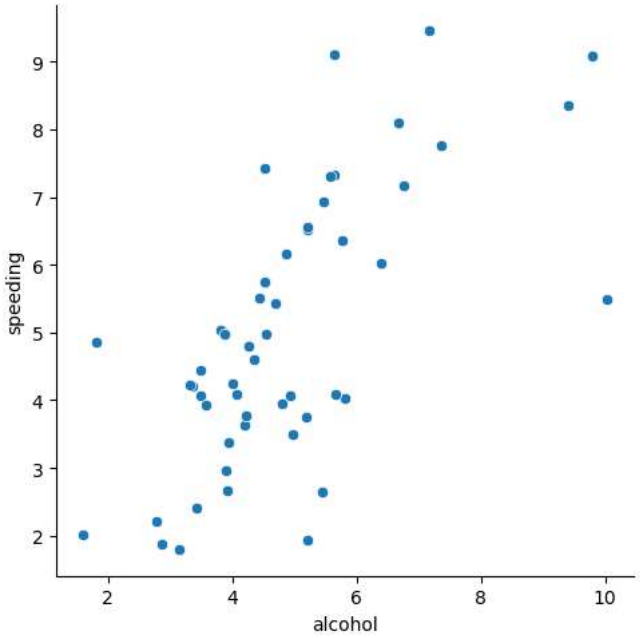
```
<seaborn.axisgrid.FacetGrid at 0x203e7dcb190>
```



Inference : From the relplot it shows the relation between the total and speeding in the form of scatter plots

```
sns.relplot(x="alcohol",y="speeding",data=df)
```

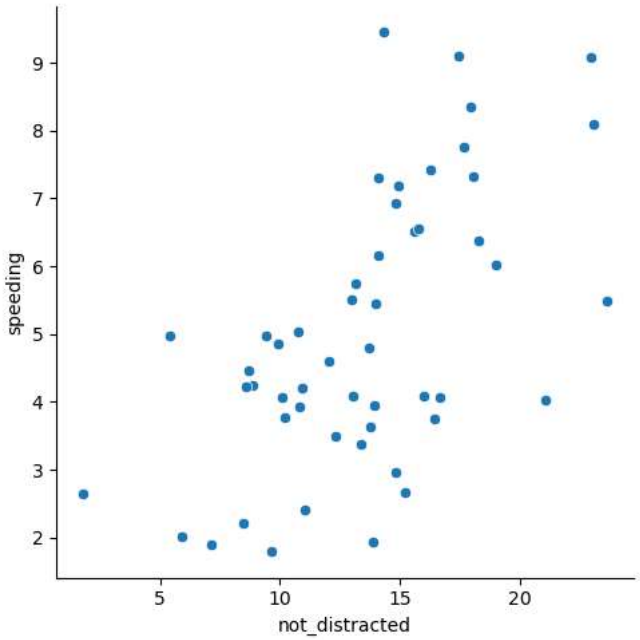
```
<seaborn.axisgrid.FacetGrid at 0x203e7fc9d50>
```



Inference : From the relplot it shows the relation between the alcohol and speeding in the form of scatter plots

```
sns.relplot(x="not_distracted",y="speeding",data=df)
```

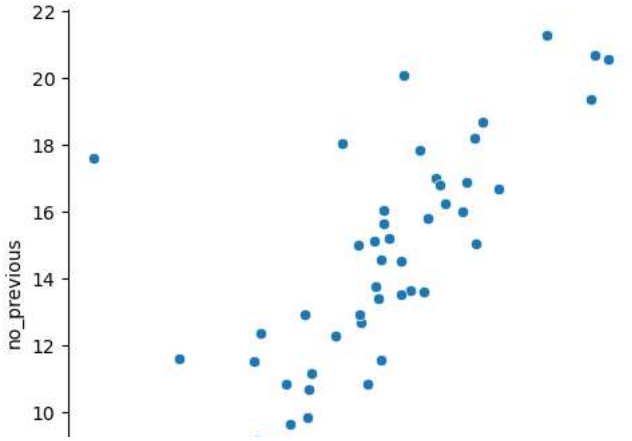
```
<seaborn.axisgrid.FacetGrid at 0x203e7fe2690>
```



Inference : From the relplot it shows the relation between the not_distracted and speeding in the form of scatter plots

```
sns.relplot(x="not_distracted",y="no_previous",data=df)
```

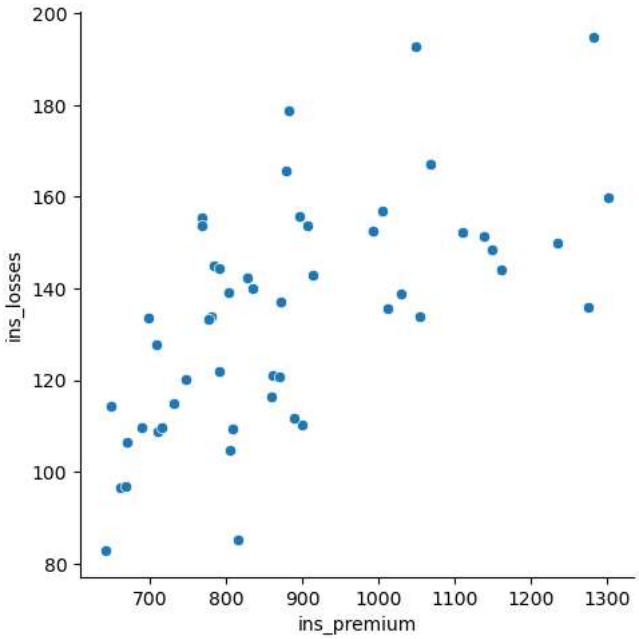
```
<seaborn.axisgrid.FacetGrid at 0x203e7b0d210>
```



Inference : From the relplot it shows the relation between the not_distracted and no_previous in the form of scatter plots

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

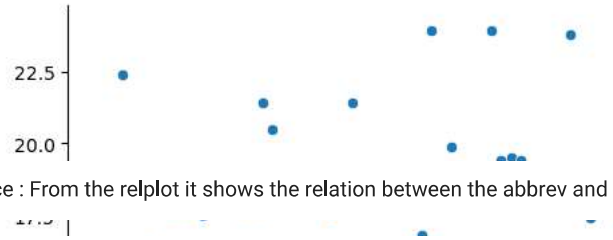
```
<seaborn.axisgrid.FacetGrid at 0x203e7b7d0d0>
```



Inference : From the relplot it shows the relation between the ins_premium and ins_losses in the form of scatter plots

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

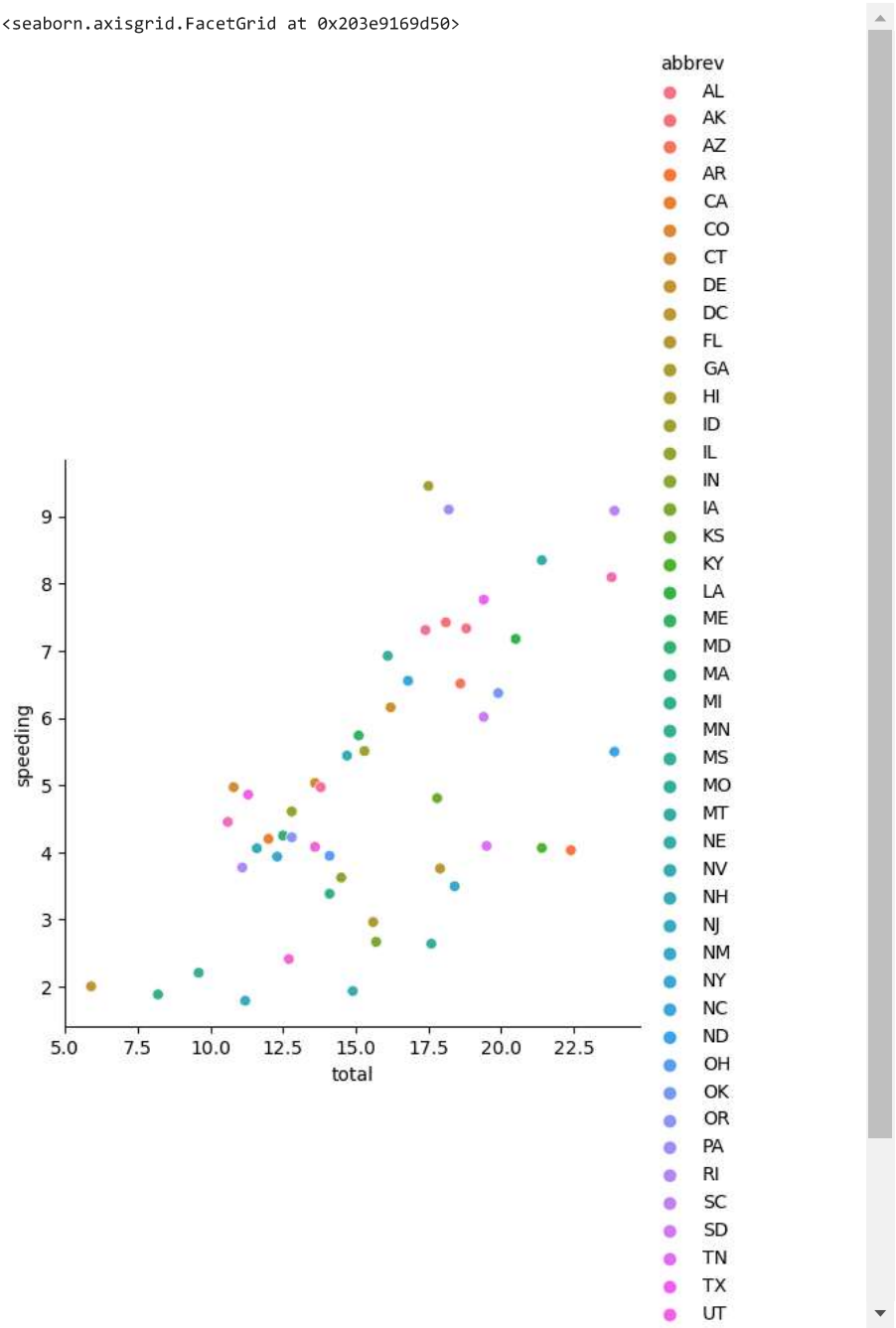
```
<seaborn.axisgrid.FacetGrid at 0x203e902a510>
```



Inference : From the relplot it shows the relation between the abbrev and total in the form of scatter plots

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

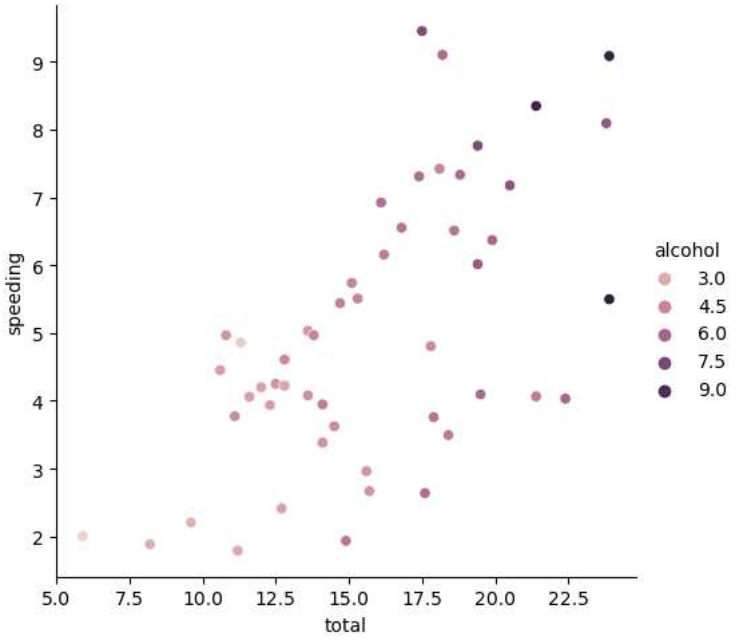
```
<seaborn.axisgrid.FacetGrid at 0x203e9169d50>
```



Inference : From the relplot it shows the relation between the total and speeding in the form of scatter plots and also shows the different colours of variables in abbrev

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

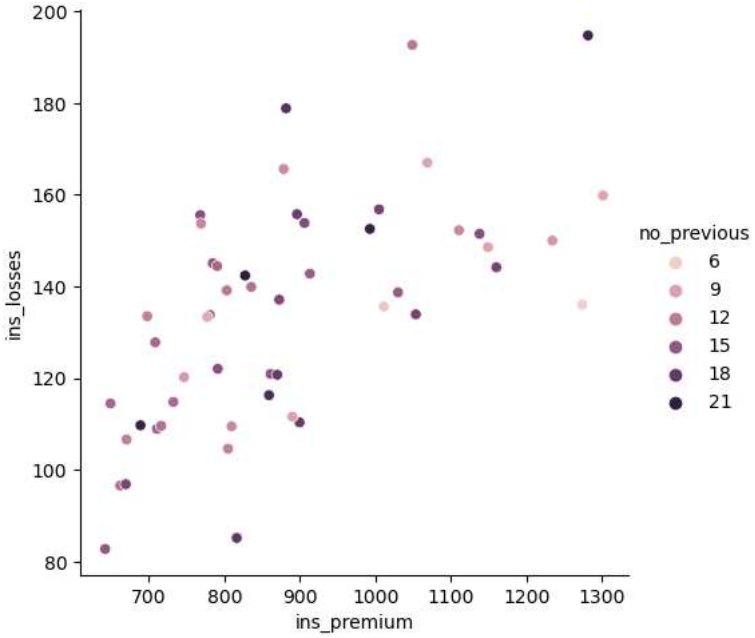
```
<seaborn.axisgrid.FacetGrid at 0x203e920a690>
```



Inference : From the relplot it shows the relation between the total and speeding in the form of scatter plots and also shows the different colours of variables in alcohol

```
sns.relplot(x="ins_premium",y="ins_losses",data=df,hue="no_previous")
```

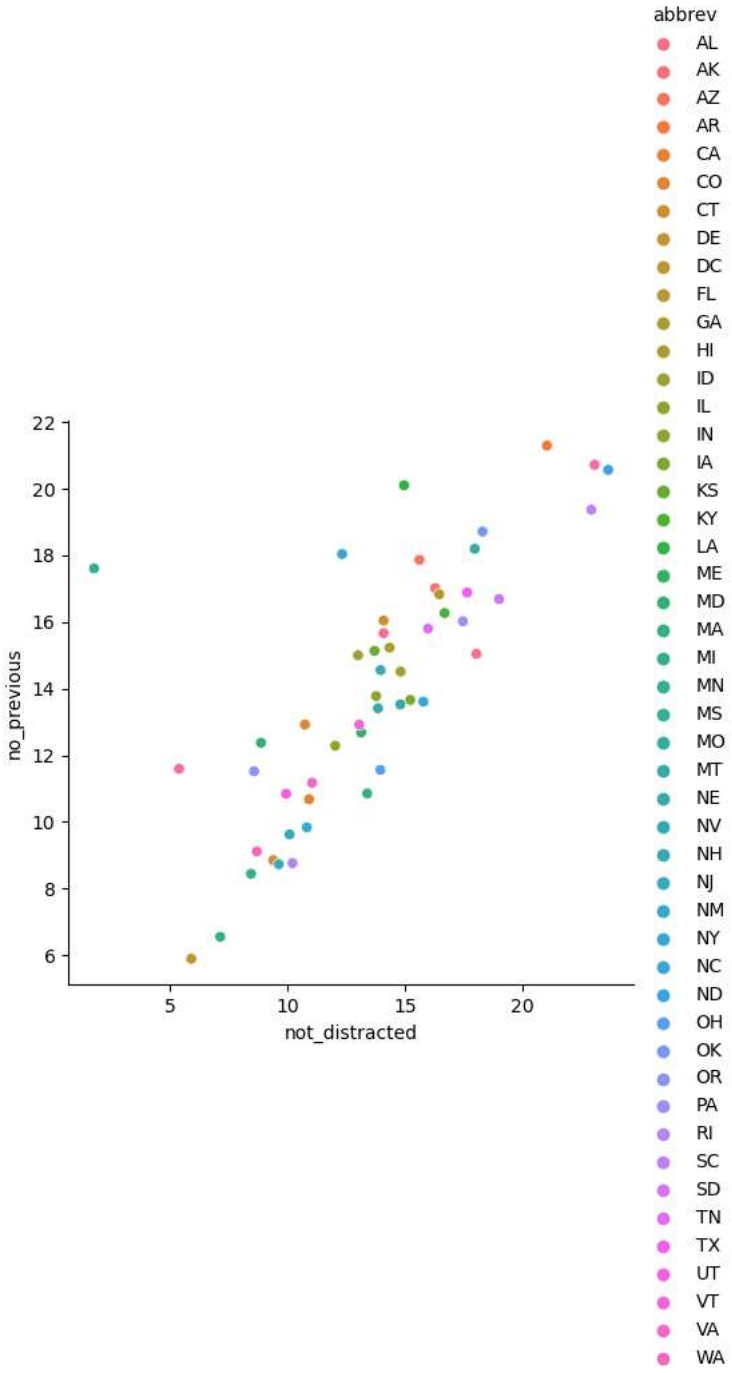
```
<seaborn.axisgrid.FacetGrid at 0x203e936d210>
```



Inference : From the relplot it shows the relation between the ins_premium and ins_losses in the form of scatter plots and also shows the different colours of variables in no_previous

```
sns.relplot(x="not_distracted",y="no_previous",data=df,hue="abbrev")
```

```
<seaborn.axisgrid.FacetGrid at 0x203e94a2090>
```



Inference : From the relplot it shows the relation between the not_distracted and no_previous in the form of scatter plots and also shows the different colours of variables in abbrev

```
df["abbrev"].value_counts()
```

```
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
```

```
df["speeding"].value_counts()
```

```
4.968    2
7.332    1
9.100    1
5.439    1
4.060    1
1.792    1
3.496    1
3.936    1
6.552    1
5.497    1
3.948    1
6.368    1
4.224    1
3.774    1
8.346    1
9.082    1
6.014    1
4.095    1
7.760    1
4.859    1
4.080    1
2.413    1
4.452    1
8.092    1
1.937    1
6.923    1
7.421    1
2.640    1
6.510    1
4.032    1
4.200    1
5.032    1
6.156    1
2.006    1
3.759    1
2.964    1
9.450    1
5.508    1
4.608    1
3.625    1
2.669    1
4.806    1
4.066    1
7.175    1
```

```
5.738    1
4.250    1
1.886    1
3.384    1
2.208    1
7.308    1
Name: speeding, dtype: int64
```

```
df["total"].value_counts()
```

```
14.1    2
12.8    2
13.6    2
21.4    2
19.4    2
23.9    2
14.9    1
14.7    1
11.6    1
11.2    1
18.4    1
12.3    1
16.8    1
19.9    1
17.6    1
18.2    1
11.1    1
19.5    1
11.3    1
12.7    1
10.6    1
23.8    1
13.8    1
16.1    1
18.8    1
9.6     1
18.1    1
18.6    1
22.4    1
12.0    1
10.8    1
16.2    1
5.9     1
17.9    1
15.6    1
17.5    1
15.3    1
14.5    1
15.7    1
17.8    1
20.5    1
15.1    1
12.5    1
8.2     1
17.4    1
Name: total, dtype: int64
```

```
df["not_distracted"].value_counts()
```

```
14.094    2
18.048    1
17.472    1
13.965    1
10.092    1
9.632     1
12.328    1
10.824    1
15.792    1
23.661    1
13.959    1
18.308    1
8.576     1
10.212    1
17.976    1
22.944    1
19.012    1
15.990    1
17.654    1
9.944     1
13.056    1
11.049    1
8.692     1
```

```
23.086    1
13.857    1
14.812    1
16.290    1
1.760     1
15.624    1
21.056    1
10.920    1
10.744    1
9.396     1
5.900     1
16.468    1
14.820    1
14.350    1
13.005    1
12.032    1
13.775    1
15.229    1
13.706    1
16.692    1
14.965    1
13.137    1
8.875     1
7.134     1
13.395    1
8.448     1
5.382     1
Name: not_distracted, dtype: int64
```

```
df["no_previous"].value_counts()
```

```
12.920    2
15.040    1
16.016    1
14.553    1
9.628     1
8.736     1
18.032    1
9.840     1
13.608    1
20.554    1
11.562    1
18.706    1
11.520    1
8.769     1
18.190    1
19.359    1
16.684    1
15.795    1
16.878    1
10.848    1
11.176    1
9.116     1
20.706    1
11.592    1
13.410    1
13.524    1
17.014    1
17.600    1
17.856    1
21.280    1
10.680    1
8.856     1
16.038    1
5.900     1
16.826    1
14.508    1
15.225    1
14.994    1
12.288    1
13.775    1
13.659    1
15.130    1
16.264    1
20.090    1
12.684    1
12.375    1
6.560     1
10.857    1
8.448     1
15.660    1
Name: no_previous, dtype: int64
```

```
df["alcohol"].value_counts()
```

```
5.208     2
5.640     1
4.218     1
4.704     1
3.480     1
3.136     1
4.968     1
3.567     1
10.038    1
4.794     1
5.771     1
3.328     1
5.642     1
9.799     1
9.416     1
6.402     1
5.655     1
7.372     1
1.808     1
4.080     1
3.429     1
3.498     1
6.664     1
4.554     1
5.215     1
5.474     1
4.525     1
5.456     1
5.824     1
3.360     1
3.808     1
3.888     1
4.860     1
1.593     1
5.191     1
3.900     1
7.175     1
4.437     1
4.352     1
4.205     1
3.925     1
4.272     1
4.922     1
6.765     1
4.530     1
4.000     1
2.870     1
3.948     1
2.784     1
5.568     1
Name: alcohol, dtype: int64
```

```
df["ins_premium"].value_counts()
```

```
784.55     1
905.99     1
1029.87    1
746.54     1
1301.52    1
869.85     1
1234.31    1
708.24     1
688.75     1
697.73     1
881.51     1
804.71     1
1148.99    1
816.21     1
858.97     1
669.31     1
767.91     1
1004.75    1
809.38     1
716.20     1
768.95     1
890.03     1
992.61     1
670.31     1
732.28     1
790.32     1
1053.48    1
```

```
641.96     1
899.47     1
827.34     1
878.41     1
835.50     1
1068.73    1
1137.87    1
1273.89    1
1160.13    1
913.15     1
861.18     1
803.11     1
896.07     1
710.46     1
649.06     1
780.45     1
872.51     1
1281.55    1
661.88     1
1048.78    1
1011.14    1
1110.61    1
777.18     1
791.14     1
Name: ins_premium, dtype: int64
```

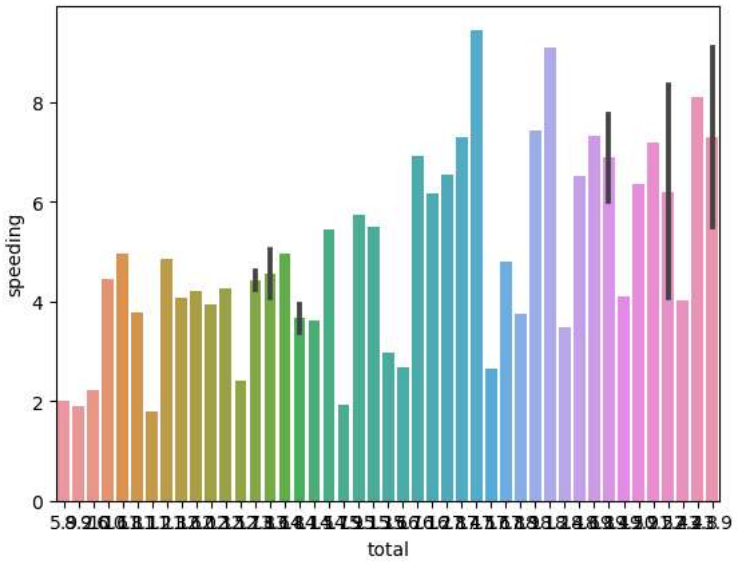
```
df["ins_losses"].value_counts()
```

```
145.08    1
153.86    1
138.71    1
120.21    1
159.85    1
120.75    1
150.01    1
127.82    1
109.72    1
133.52    1
178.86    1
104.61    1
148.58    1
85.15     1
116.29    1
96.87     1
155.57    1
156.83    1
109.48    1
109.61    1
153.72    1
111.62    1
152.56    1
106.62    1
114.82    1
144.45    1
133.93    1
82.75     1
110.35    1
142.39    1
165.63    1
139.91    1
167.02    1
151.48    1
136.05    1
144.18    1
142.80    1
120.92    1
139.15    1
155.77    1
108.92    1
114.47    1
133.80    1
137.13    1
194.78    1
96.57     1
192.70    1
135.63    1
152.26    1
133.35    1
122.04    1
Name: ins_losses, dtype: int64
```

5.Bar Plot

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

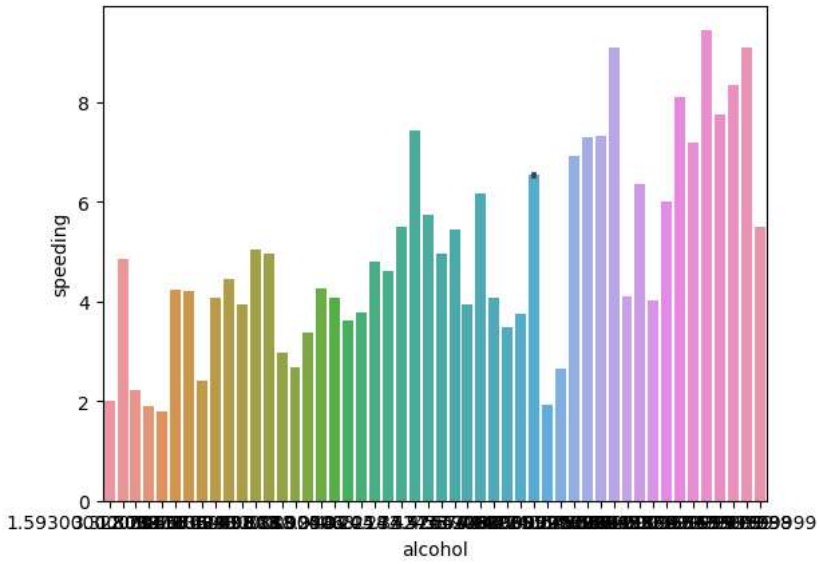
```
<Axes: xlabel='total', ylabel='speeding'>
```



Inference : From the barplot displays a plot between total and speeding with a error bars

```
sns.barplot(data=df,x="alcohol",y="speeding")
```

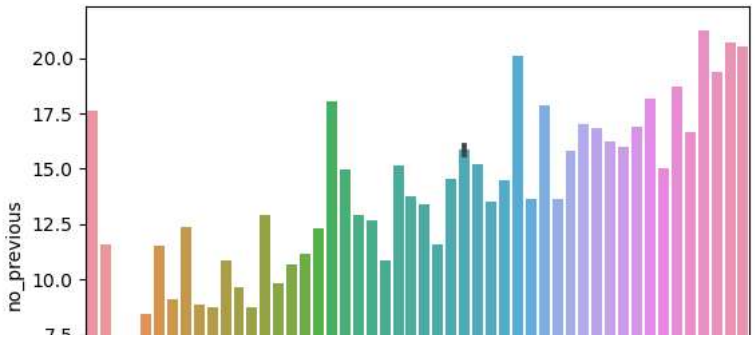
```
<Axes: xlabel='alcohol', ylabel='speeding'>
```



Inference : From the barplot displays a plot between alcohol and speeding with a small error bars

```
sns.barplot(data=df,x="not_distracted",y="no_previous")
```

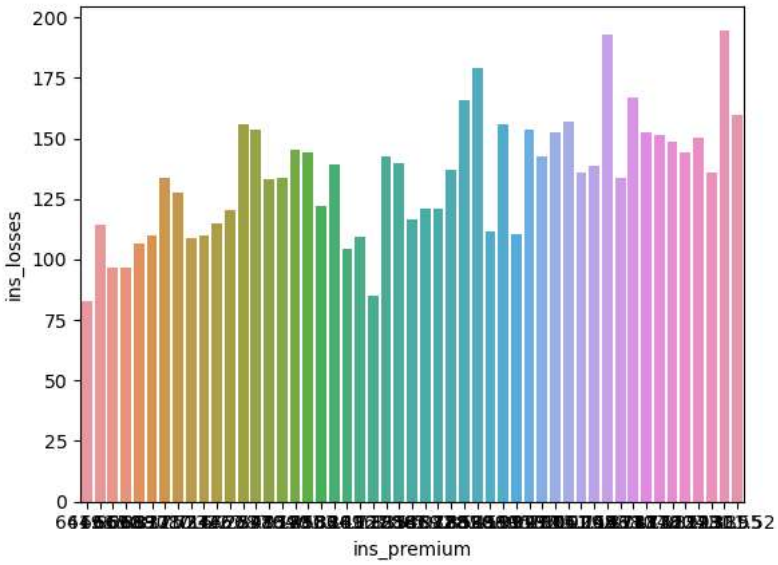
```
<Axes: xlabel='not_distracted', ylabel='no_previous'>
```



Inference : From the barplot displays a plot between not_distracted and no_previous with a small error bars

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

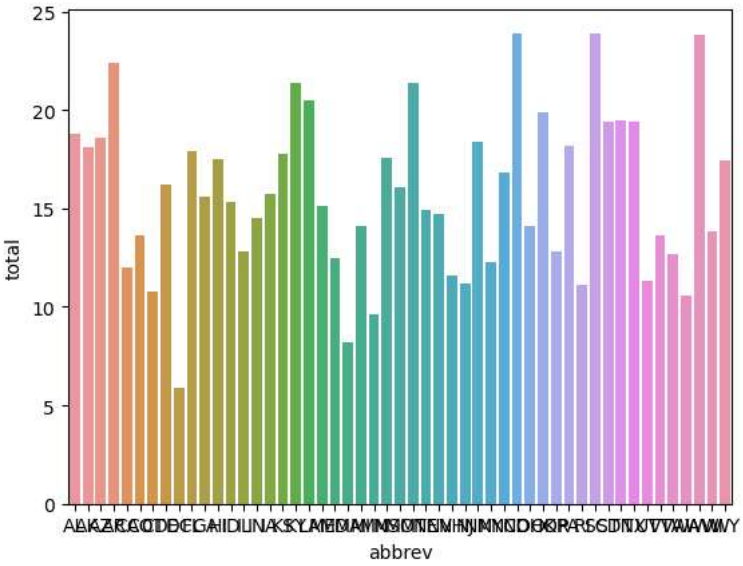
```
<Axes: xlabel='ins_premium', ylabel='ins_losses'>
```



Inference : From the barplot displays a plot between ins_premium and ins_losses without a error bars

```
sns.barplot(data=df,x="abbrev",y="total")
```

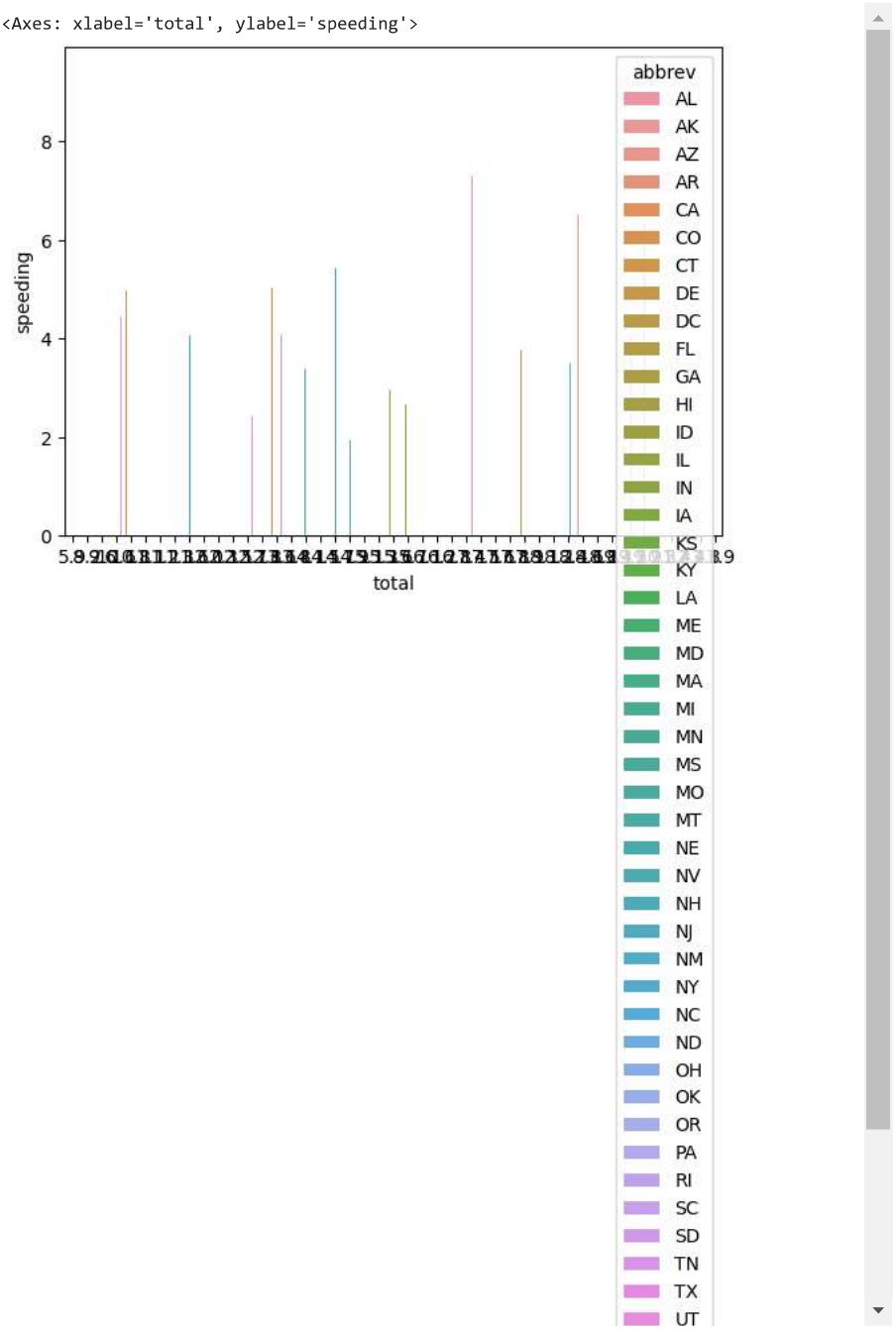
```
<Axes: xlabel='abbrev', ylabel='total'>
```



Inference : From the barplot displays a plot between ins_premium and ins_losses without a error bars

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

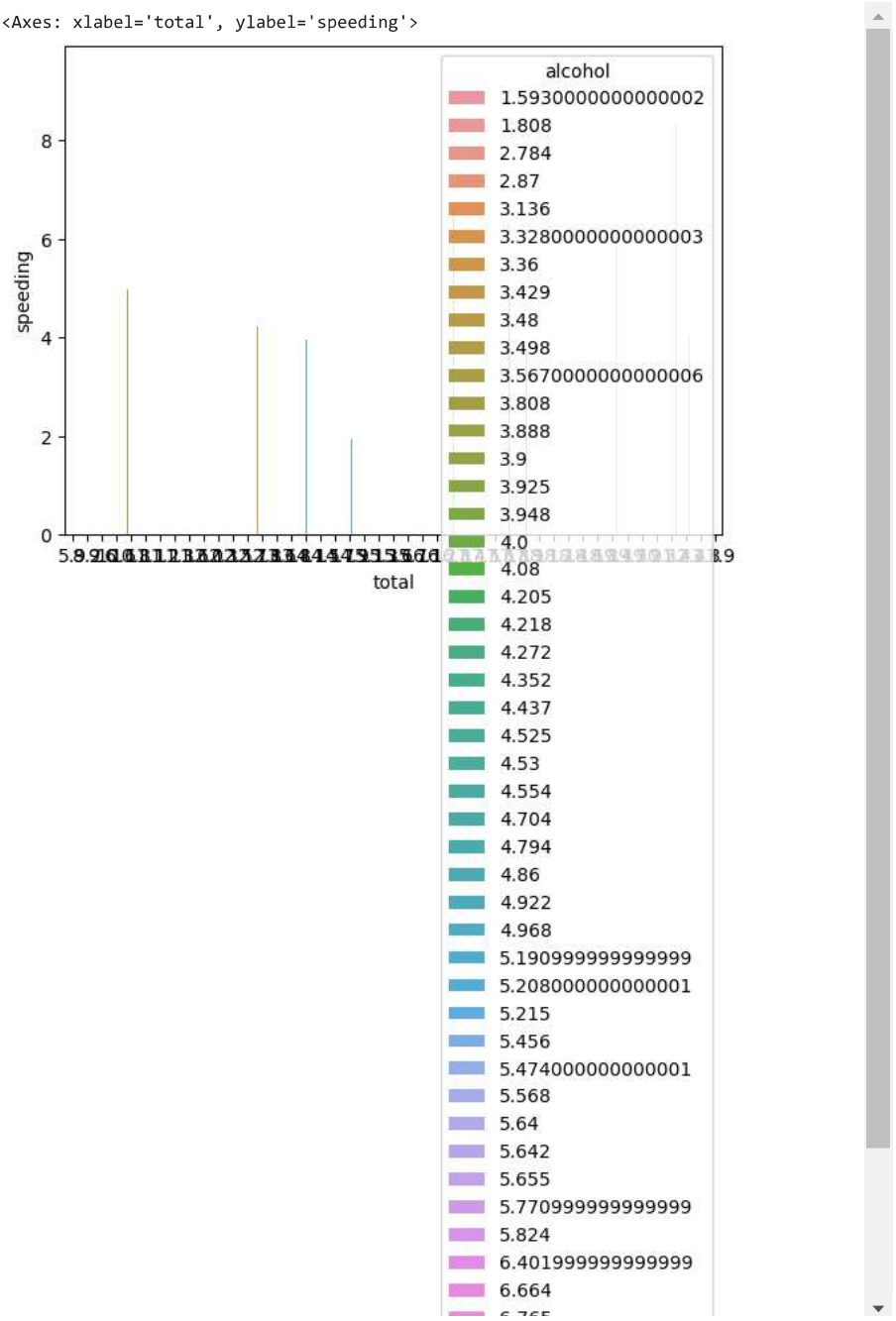
```
<Axes: xlabel='total', ylabel='speeding'>
```



Inference : From the barplot displays a plot between total and speeding with a different colours of variables of abbrev

```
sns.barplot(data=df,x="total",y="speeding",hue="alcohol")
```

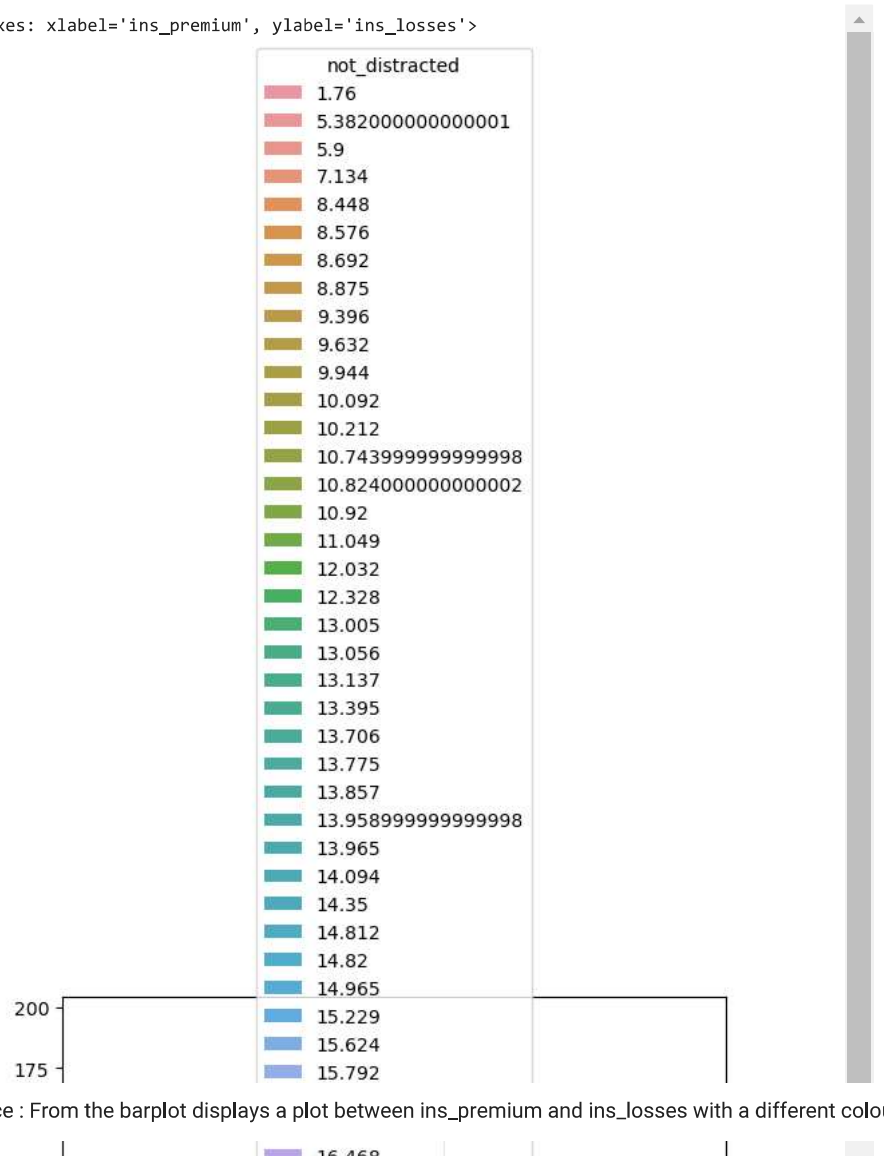
```
<Axes: xlabel='total', ylabel='speeding'>
```



Inference : From the barplot displays a plot between total and speeding with a different colours of variables of alcohol

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

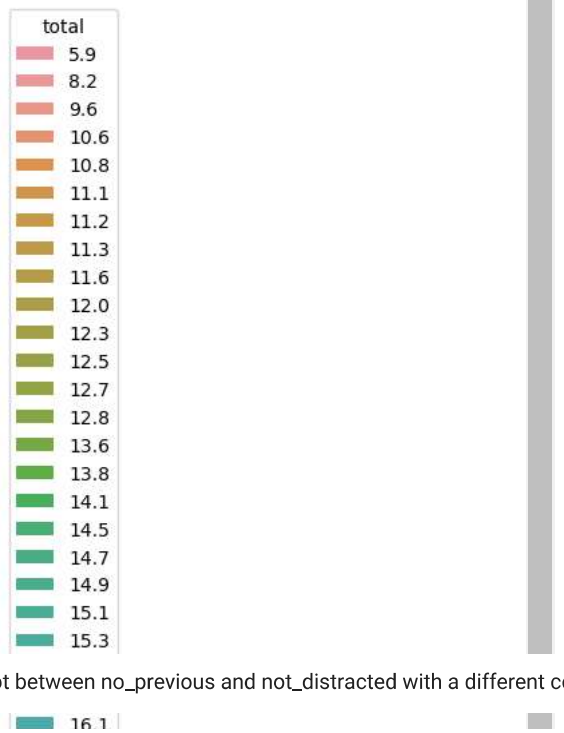
```
<Axes: xlabel='ins_premium', ylabel='ins_losses'>
```



Inference : From the barplot displays a plot between ins_premium and ins_losses with a different colours of variables of not_distracted

```
sns.barplot(data=df,x="no_previous",y="not_distracted",hue="total")
```

```
<Axes: xlabel='no_previous', ylabel='not_distracted'>
```

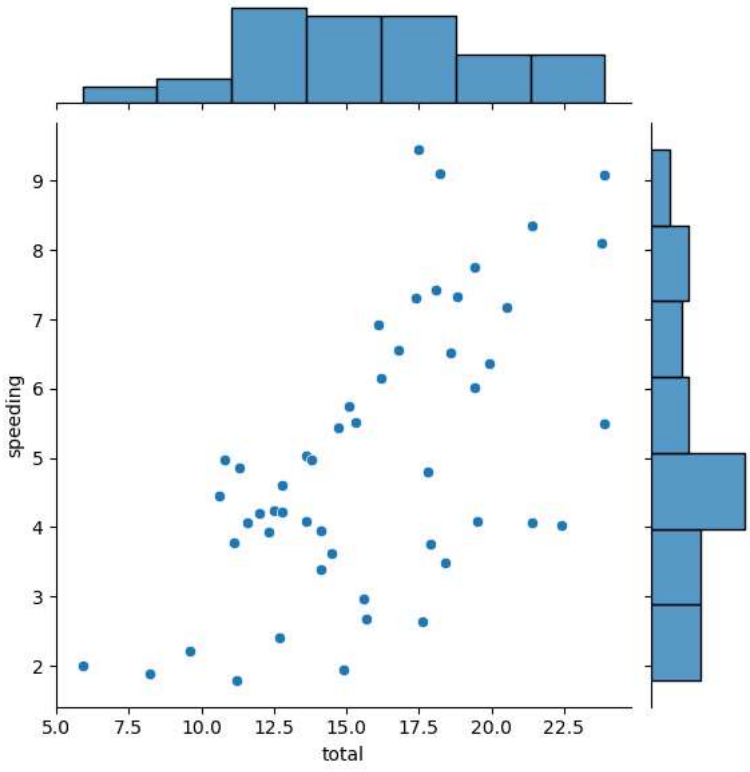


Inference : From the barplot displays a plot between no_previous and not_distracted with a different colours of variables of total

6.Joint Plot

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

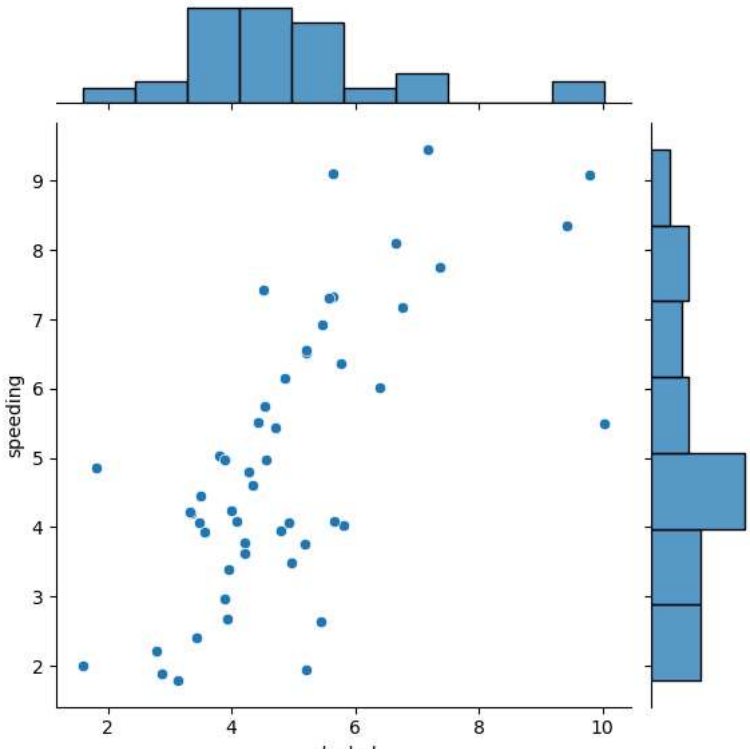
```
<seaborn.axisgrid.JointGrid at 0x203fa413c10>
```



Inference : From JointPlot the relationship between the total and speeding display the individual distributions

```
sns.jointplot(x="alcohol",y="speeding",data=df)
```

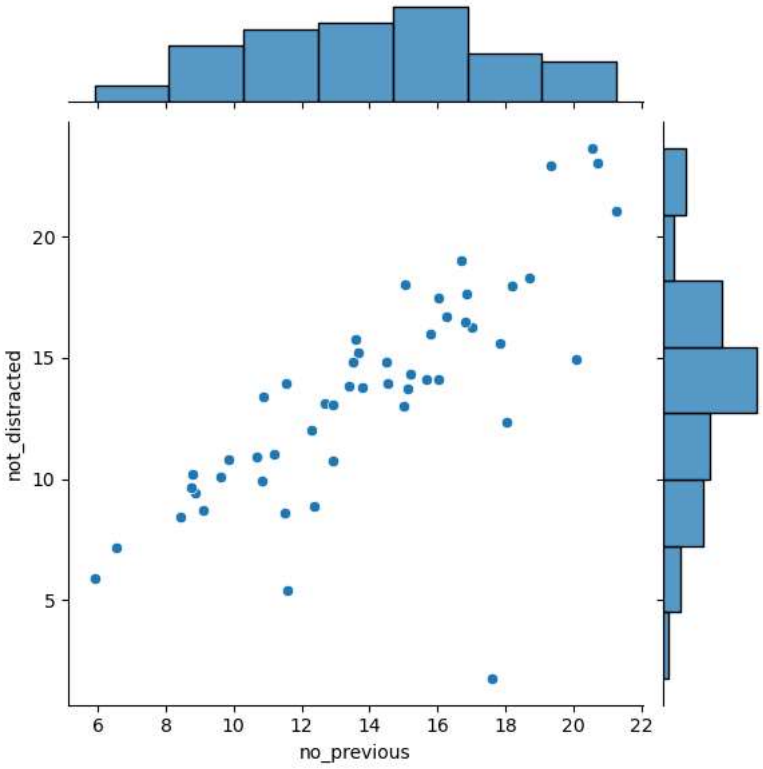
```
<seaborn.axisgrid.JointGrid at 0x203fab3b150>
```



Inference : From JointPlot the relationship between the alcohol and speeding display the individual distributions

```
sns.jointplot(x="no_previous",y="not_distracted",data=df)
```

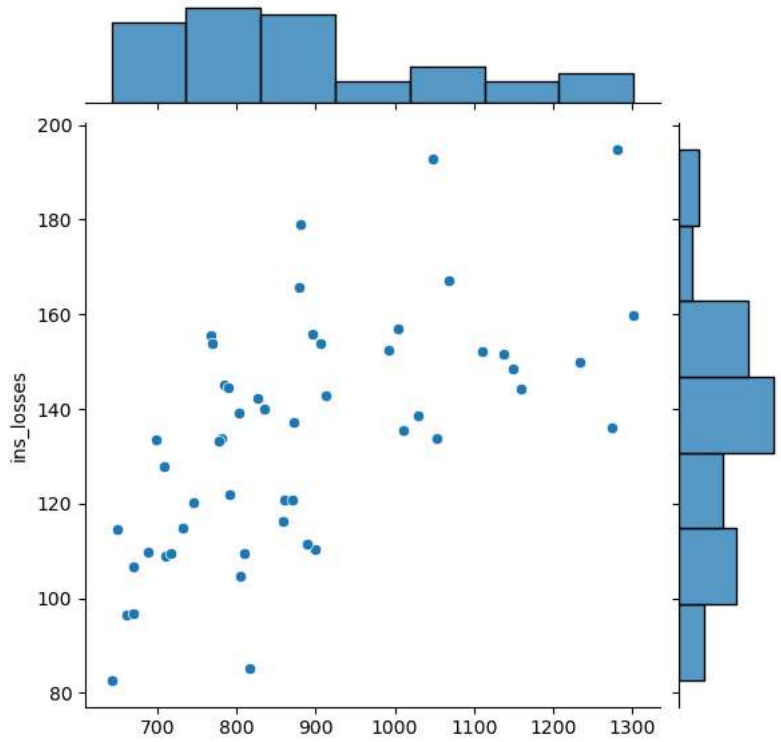
```
<seaborn.axisgrid.JointGrid at 0x203fa55ae50>
```



Inference : From JointPlot the relationship between the no_previous and not_distracted display the individual distributions

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

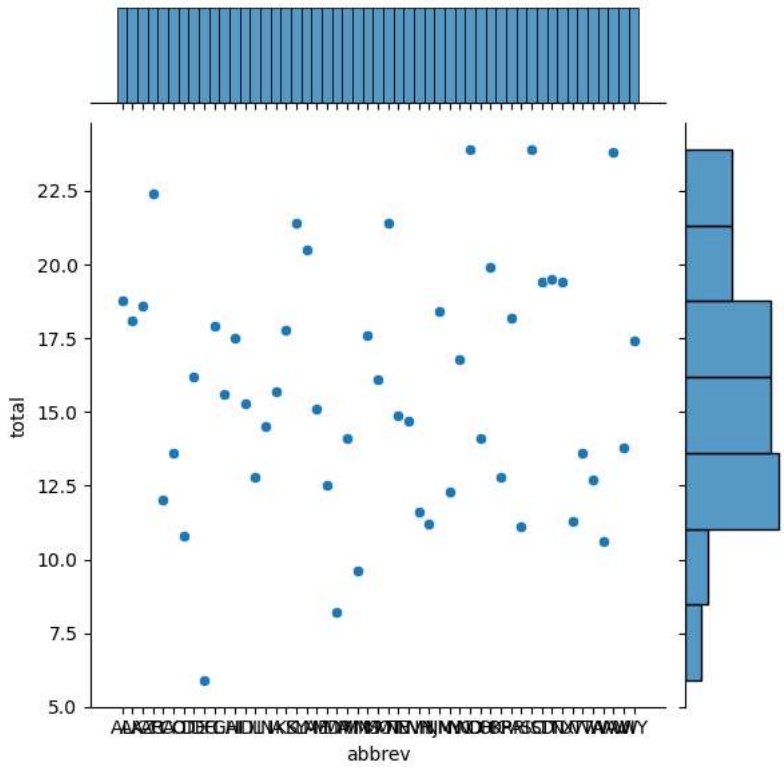
```
<seaborn.axisgrid.JointGrid at 0x203fb3bb6d0>
```



Inference : From JointPlot the relationship between the ins_premium and ins_losses display the individual distributions

```
sns.jointplot(x="abbrev",y="total",data=df)
```

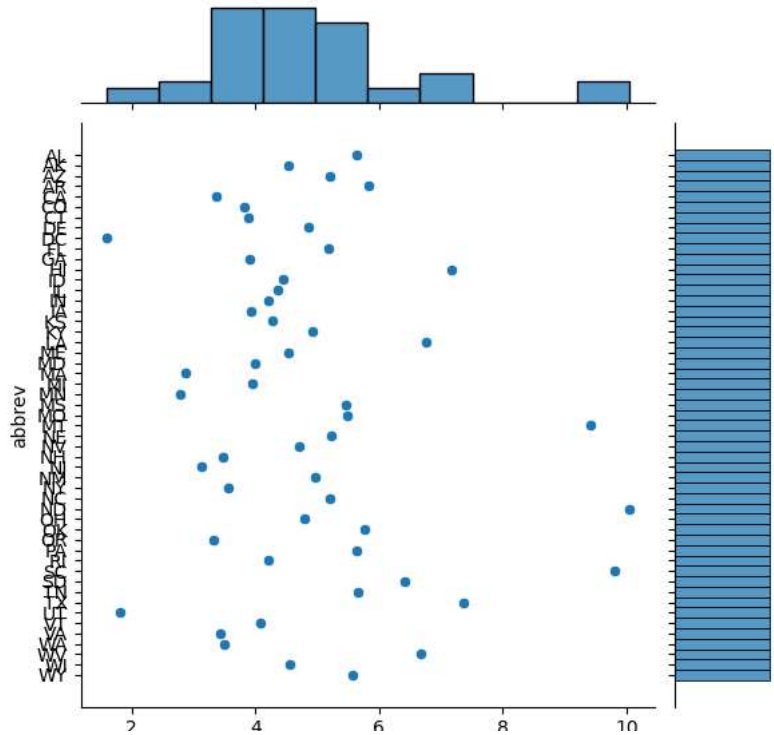
```
<seaborn.axisgrid.JointGrid at 0x203fa9e5cd0>
```



Inference : From JointPlot the relationship between the abbrev and total display the individual distributions

```
sns.jointplot(x="alcohol",y="abbrev",data=df)
```

```
<seaborn.axisgrid.JointGrid at 0x203fbe1bed0>
```

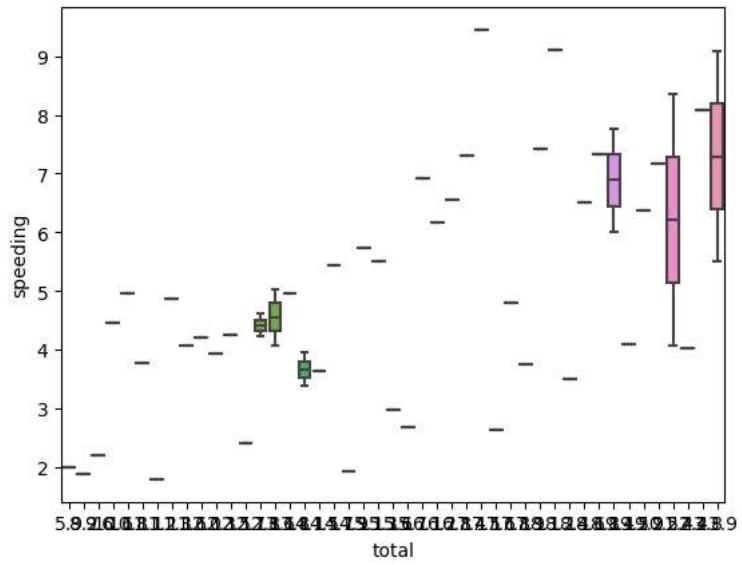


Inference : From JointPlot the relationship between the alcohol and abbrev display the individual distributions

7.Box Plot

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

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



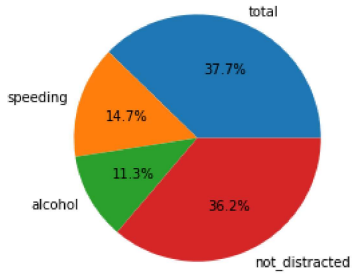
Inference : From the boxplot the plot between total and speeding and it contains outliers too

8.Pie Chart

```
import matplotlib.pyplot as plt
labels = ["total","speeding","alcohol","not_distracted"]
numbers = [18.8,7.332,5.640,18.048]
fig = plt.figure()
```

```
axes1=fig.add_axes([0.1,0.1,0.8,0.8])  #[Left,bottom,width,height]
axes1.pie(numbers,labels = labels,autopct="%0.1f%%")
```

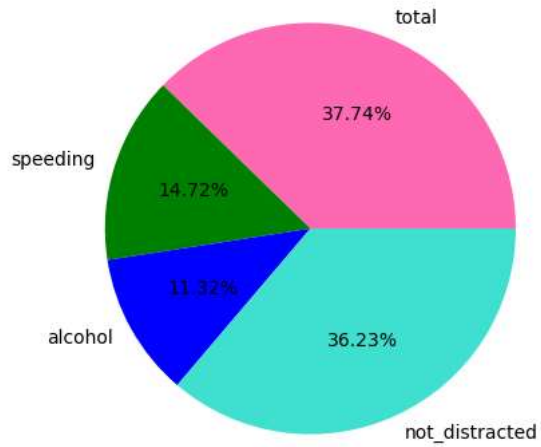
```
([<matplotlib.patches.Wedge at 0x26a9a5a57c0>,
  <matplotlib.patches.Wedge at 0x26a9a5a5dc0>,
  <matplotlib.patches.Wedge at 0x26a9a5ba520>,
  <matplotlib.patches.Wedge at 0x26a9a5bac40>],
 [Text(0.41341041775749476, 1.019358536771814, 'total'),
  Text(-1.048158608365565, 0.33371174943229454, 'speeding'),
  Text(-0.9601436594221328, -0.5367719751174382, 'alcohol'),
  Text(0.46126564360326483, -0.9986160453503969, 'not_distracted')],
 [Text(0.22549659150408802, 0.5560137473300804, '37.7%'),
  Text(-0.571722877290308, 0.18202459059943335, '14.7%'),
  Text(-0.5237147233211632, -0.2927847137004208, '11.3%'),
  Text(0.25159944196541717, -0.5446996611002164, '36.2%')])
```



Inference : From the piechart the labels are not_distracted,alcohol,speeding,total with a default colour

```
import matplotlib.pyplot as plt
labels = ["total","speeding","alcohol","not_distracted"]
numbers = [18.8,7.332,5.640,18.048]
fig = plt.figure()
axes1=fig.add_axes([0.1,0.1,0.8,0.8])  #[Left,bottom,width,height]
axes1.pie(numbers,labels = labels,autopct="%0.2f%%",colors=["Hotpink","green","blue","turquoise"])
```

```
([<matplotlib.patches.Wedge at 0x203fce6a150>,
  <matplotlib.patches.Wedge at 0x203fcedfe90>,
  <matplotlib.patches.Wedge at 0x203fceed390>,
  <matplotlib.patches.Wedge at 0x203fceec990>],
 [Text(0.41341041775749476, 1.019358536771814, 'total'),
  Text(-1.048158608365565, 0.33371174943229454, 'speeding'),
  Text(-0.9601436594221328, -0.5367719751174382, 'alcohol'),
  Text(0.46126564360326483, -0.9986160453503969, 'not_distracted')],
 [Text(0.22549659150408802, 0.5560137473300804, '37.74%'),
  Text(-0.571722877290308, 0.18202459059943335, '14.72%'),
  Text(-0.5237147233211632, -0.2927847137004208, '11.32%'),
  Text(0.25159944196541717, -0.5446996611002164, '36.23%')])
```

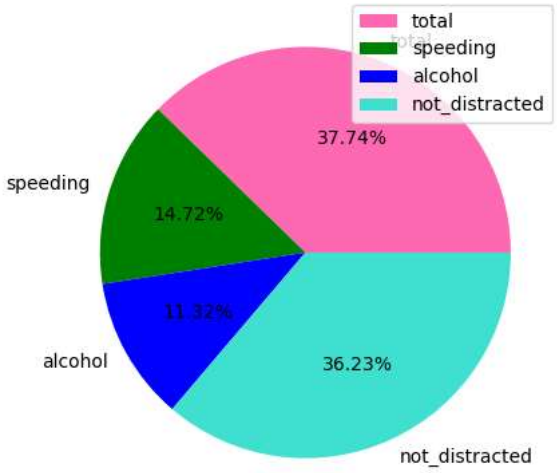


Inference : From the piechart the labels are not_distracted,alcohol,speeding,total with a selected colour

```
import matplotlib.pyplot as plt
labels = ["total","speeding","alcohol","not_distracted"]
```

```
numbers = [18.8,7.332,5.640,18.048]
fig = plt.figure()
axes1=fig.add_axes([0.1,0.1,0.8,0.8])  #[Left,bottom,width,height]
axes1.pie(numbers,labels = labels,autopct="%0.2f%%",colors=["Hotpink","green","blue","turquoise"])
axes1.legend()
```

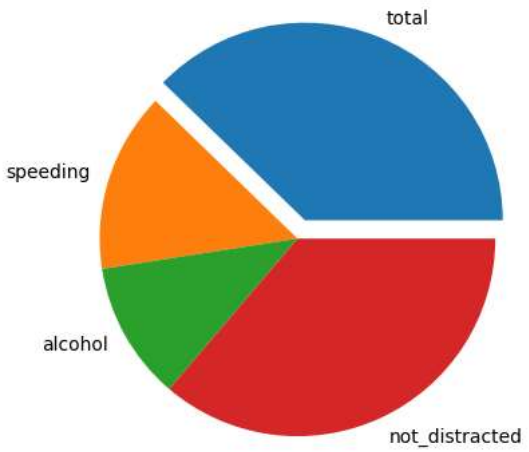
```
<matplotlib.legend.Legend at 0x203fccd01d0>
```



Inference : From the piechart the labels are not_distracted,alcohol,speeding,total with a selected colour and has a legend

```
Explode = [0.1,0,0,0]
plt.pie(numbers,labels=labels,explode=Explode)
```

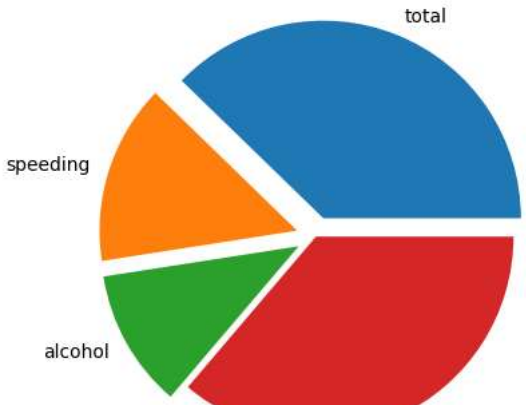
```
([<matplotlib.patches.Wedge at 0x203fcd4b850>,
  <matplotlib.patches.Wedge at 0x203fcd60150>,
  <matplotlib.patches.Wedge at 0x203fcd61050>,
  <matplotlib.patches.Wedge at 0x203fcd61fd0>],
 [Text(0.4509931830081761, 1.1120274946601607, 'total'),
  Text(-1.048158608365565, 0.33371174943229454, 'speeding'),
  Text(-0.9601436594221328, -0.5367719751174382, 'alcohol'),
  Text(0.46126564360326483, -0.9986160453503969, 'not_distracted')])
```



Inference : From the piechart the labels are not_distracted,alcohol,speeding,total with a default colour using explode

```
Explode = [0.1,0.1,0.1,0]
plt.pie(numbers,labels=labels,explode=Explode)
```

```
([<matplotlib.patches.Wedge at 0x203fcda0590>,
  <matplotlib.patches.Wedge at 0x203fcda1490>,
  <matplotlib.patches.Wedge at 0x203fcda2710>,
  <matplotlib.patches.Wedge at 0x203fcda3bd0>],
 [Text(0.4509931830081761, 1.1120274946601607, 'total'),
  Text(-1.1434457545806163, 0.36404918119886676, 'speeding'),
  Text(-1.0474294466423266, -0.5855694274008417, 'alcohol'),
  Text(0.46126564360326483, -0.9986160453503969, 'not_distracted')])
```



Inference : From the piechart the labels are not_distracted,alcohol,speeding,total with a default colour using explode with a sample explode points

9.HeatMap

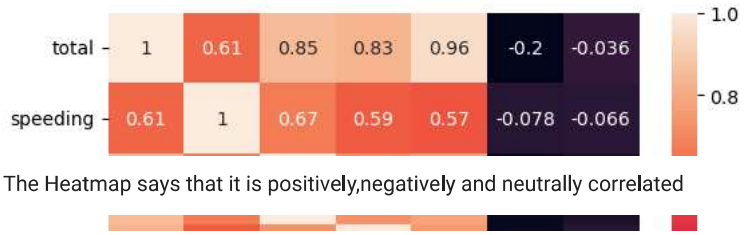
```
corr=df.corr()
corr
```

```
C:\Users\Asus\AppData\Local\Temp\ipykernel_16988\3182140910.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is
  corr=df.corr()
```

	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

```
sns.heatmap(corr,annot=True)
```

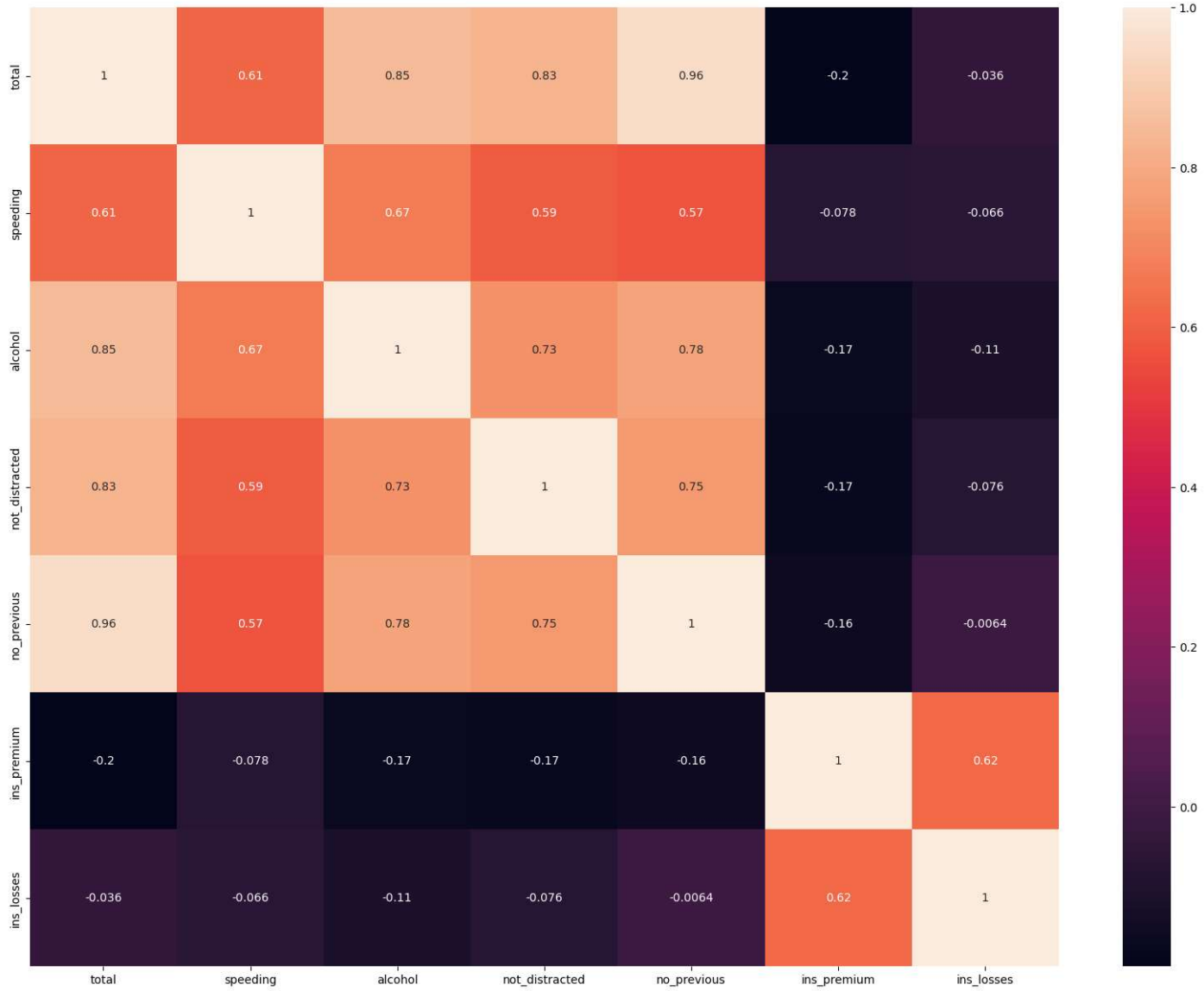
```
<Axes: >
```



Inference : The Heatmap says that it is positively,negatively and neutrally correlated

```
plt.subplots(figsize=(20,15))
sns.heatmap(corr,annot=True)
```

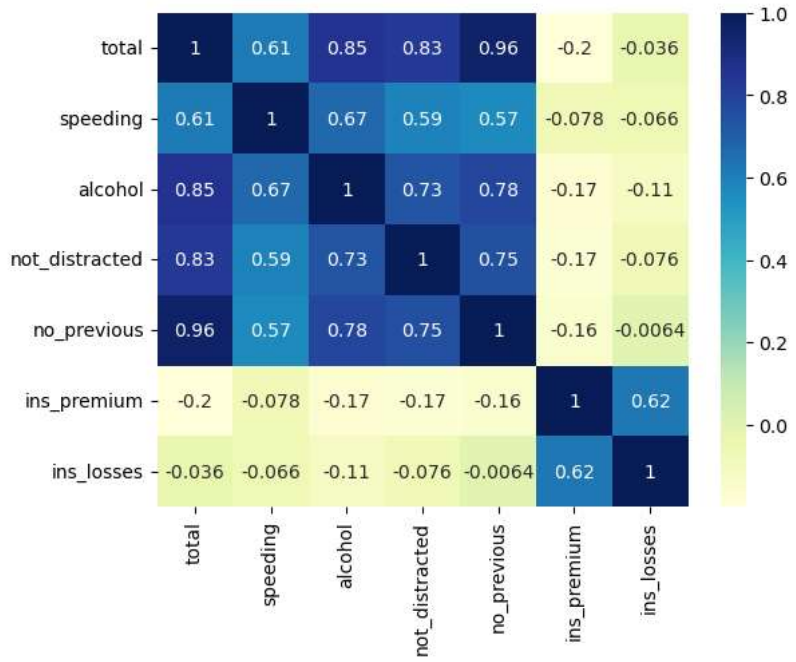
```
<Axes: >
```



Inference : The Heatmap says that it is positively,negatively and neutrally correlated with a sample of figuresize numbers

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

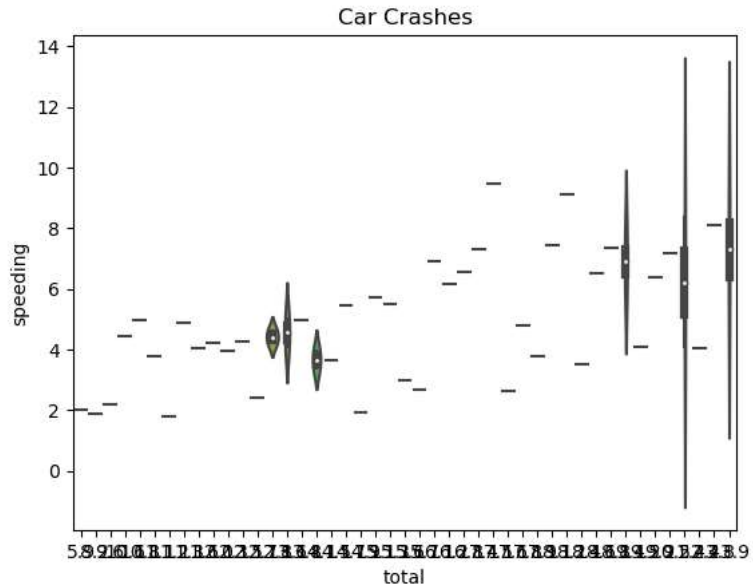
```
<Axes: >
```



Inference : The Heatmap says that it is positively,negatively and neutrally correlated with a selected colour

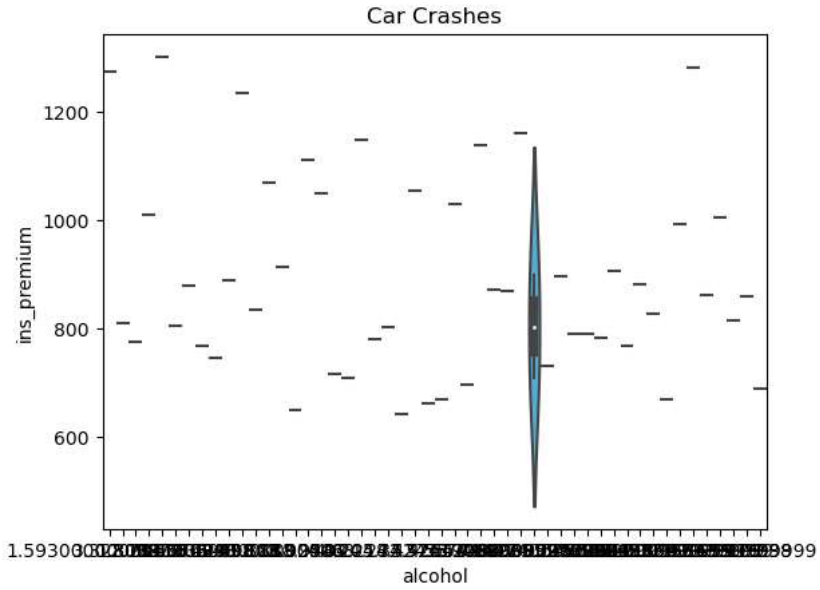
10.Violin Plot

```
sns.violinplot(x='total', y='speeding', data=df)

# Customize the plot
plt.title('Car Crashes')
plt.xlabel('total')
plt.ylabel('speeding')

# Display the plot
plt.show()
```



Inference : The Violin plot shows the distribution of data with the combination of boxplot and density between the total and speeding

```
sns.violinplot(x='alcohol', y='ins_premium', data=df)

# Customize the plot
plt.title('Car Crashes')
plt.xlabel('alcohol')
plt.ylabel('ins_premium')

# Display the plot
plt.show()
```

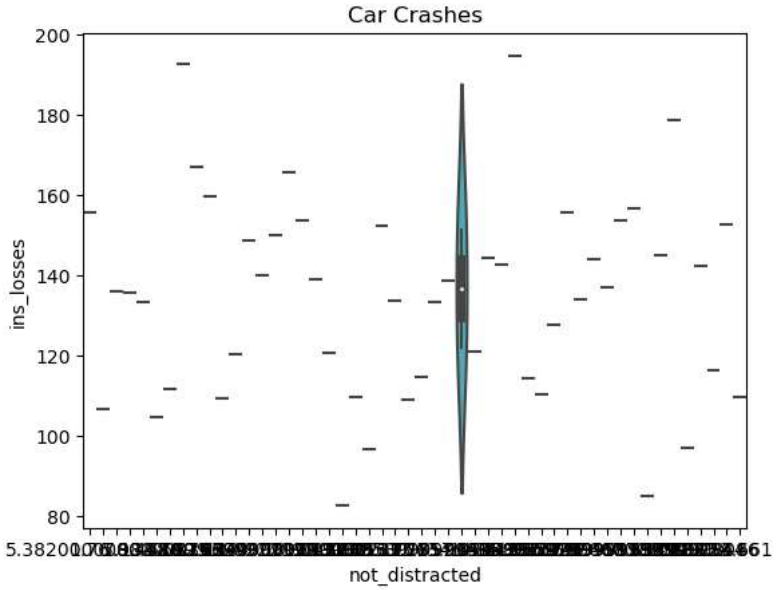


Inference : The Violin plot shows the distribution of data with the combination of boxplot and density between the alcohol and ins_premium

```
sns.violinplot(x='no_previous', y='ins_premium', data=df)

# Customize the plot
plt.title('Car Crashes')
plt.xlabel('no_previous')
plt.ylabel('ins_premium')

# Display the plot
plt.show()
```

Inference : The Violin plot shows the distribution of data with the combination of boxplot and density between the no_previous and ins_premium

```
sns.violinplot(x='not_distracted', y='ins_losses', data=df)

# Customize the plot
plt.title('Car Crashes')
plt.xlabel('not_distracted')
plt.ylabel('ins_losses')

# Display the plot
plt.show()
```

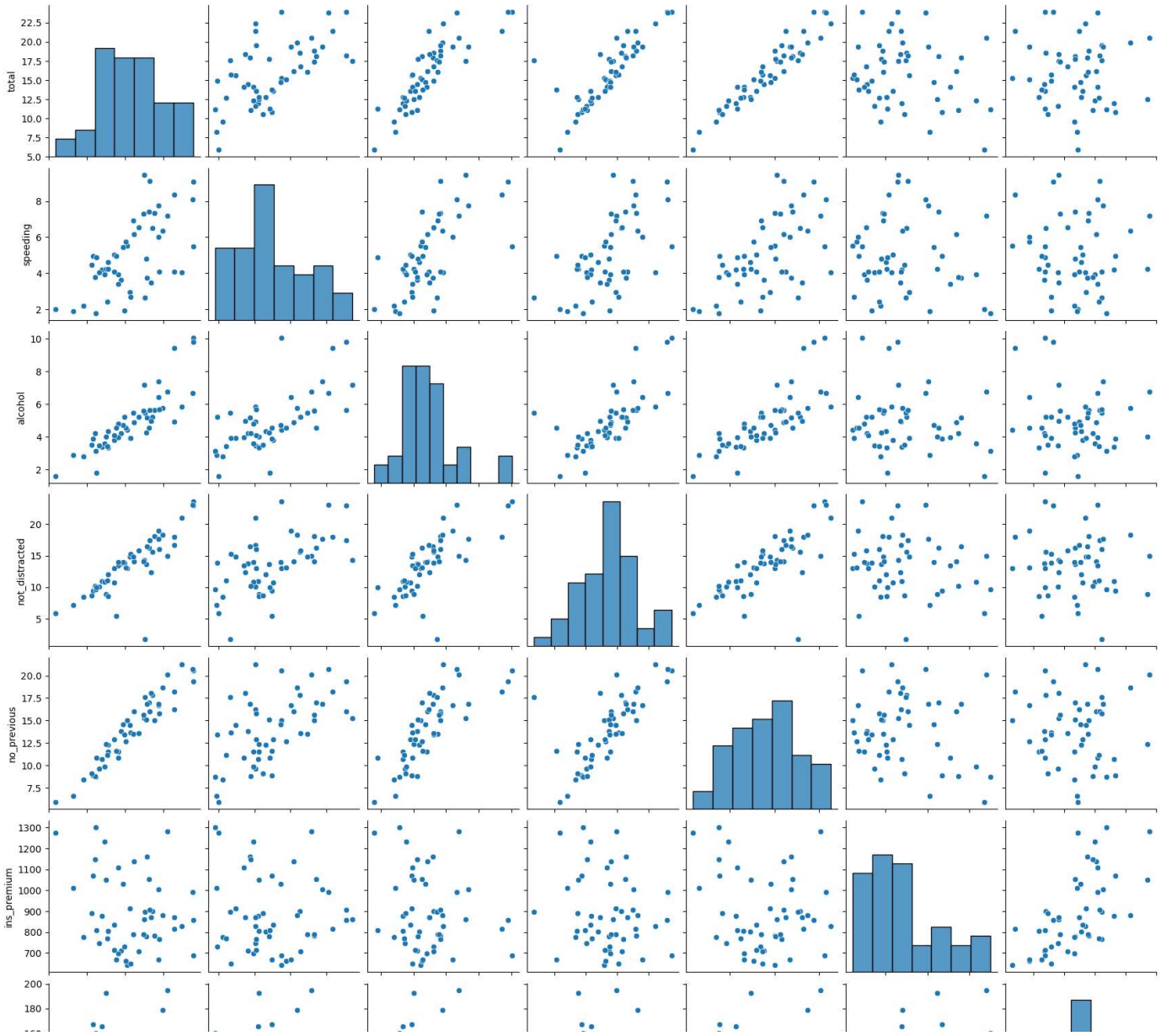


Inference : The Violin plot shows the distribution of data with the combination of boxplot and density between the not_distracted and ins_losses

11.Pair Plot

```
sns.pairplot(df)
```

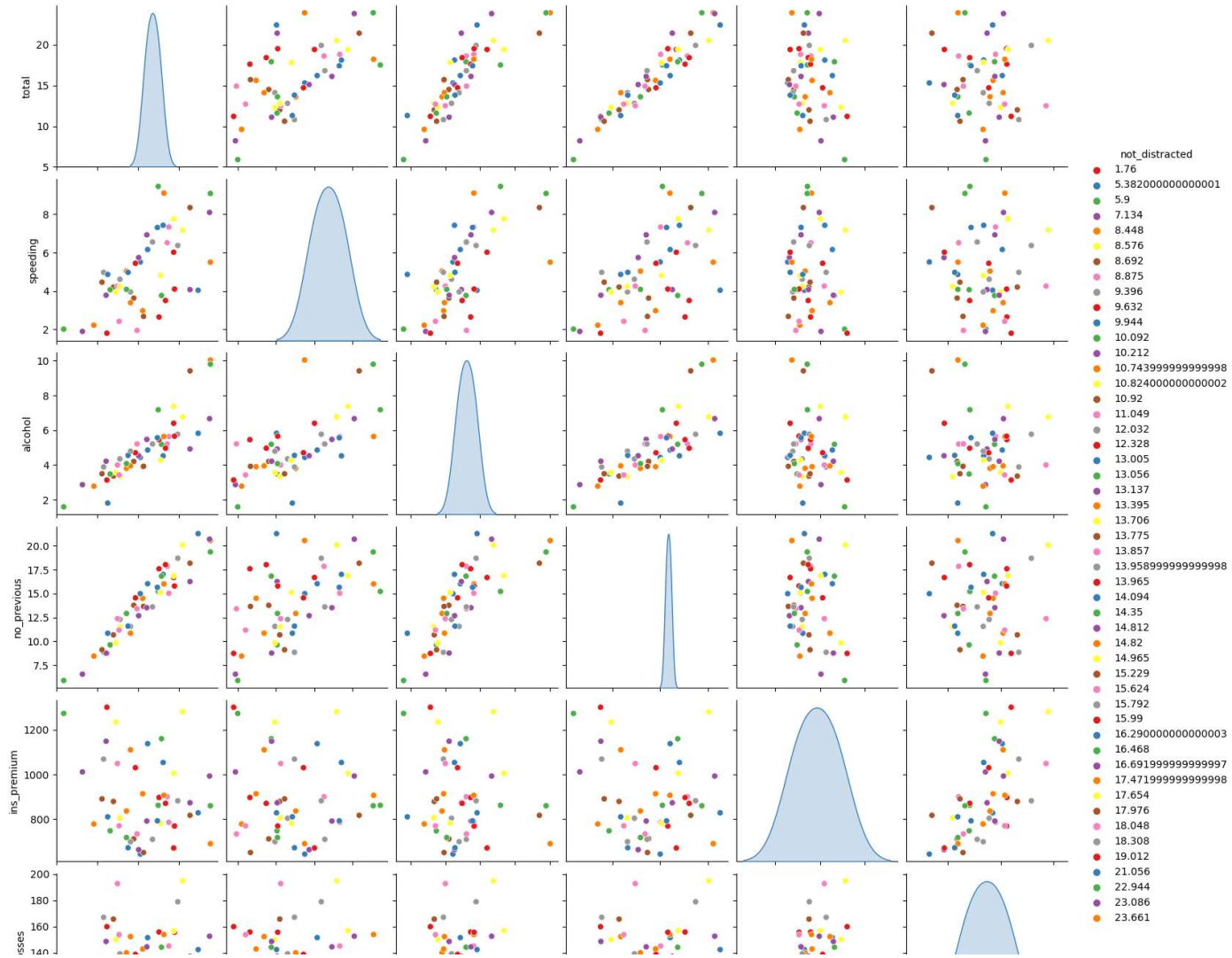
```
<seaborn.axisgrid.PairGrid at 0x203845ab690>
```



Inference : The Pair Plot shows the pair grid for every column in car_crashes dataset

```
sns.pairplot(df,hue="not_distracted",palette = "Set1")
```

```
<seaborn.axisgrid.PairGrid at 0x203e60f6a90>
```



Inference : The Pair Plot shows the pair grid for every column in car_crashes dataset with a different colours using not_distracted

12.Linear Regression Plot

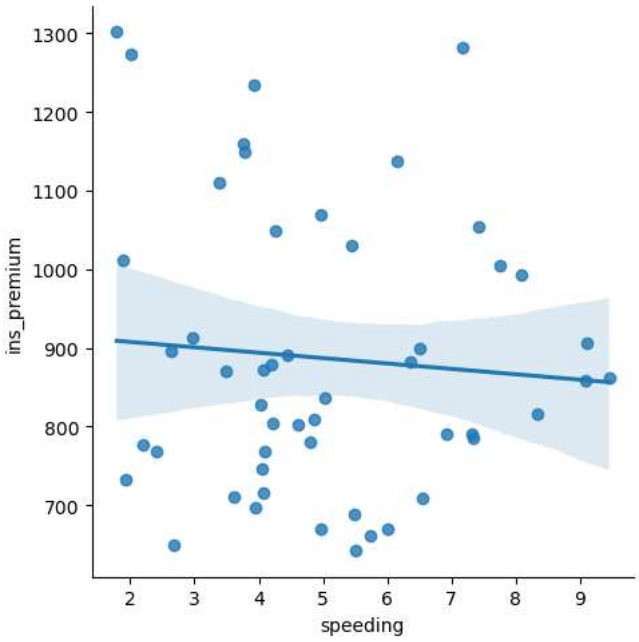
```
sns.lmplot(x="alcohol",y="ins_losses",data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x20391de30d0>
```

Inference : The Linear Regression plot between alcohol and ins_losses shows linear lin

```
sns.lmplot(x="speeding",y="ins_premium",data=df)
```

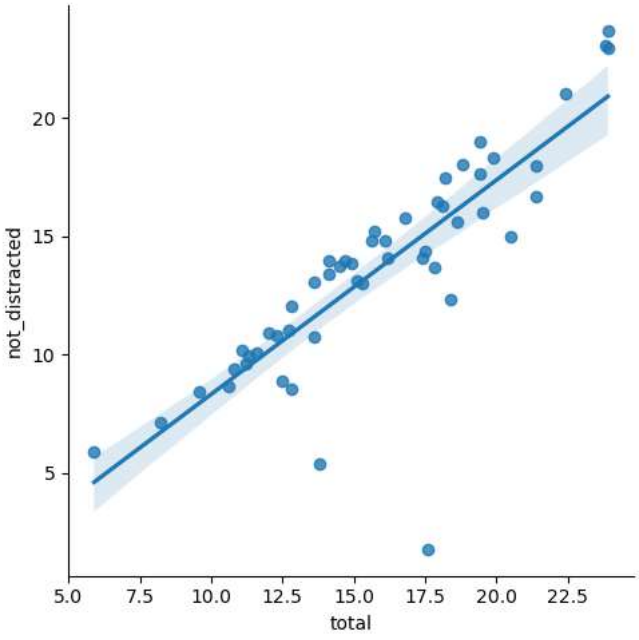
```
<seaborn.axisgrid.FacetGrid at 0x2039259e3d0>
```



Inference : The Linear Regression plot between speeding and ins_premium shows linear line

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

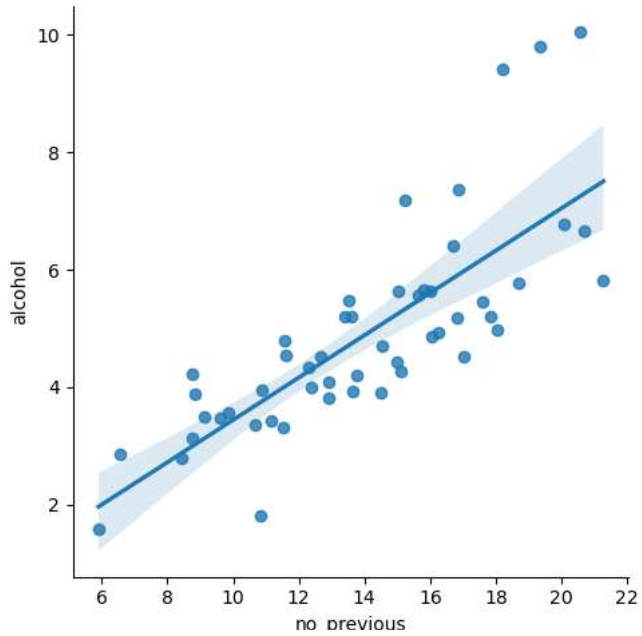
```
<seaborn.axisgrid.FacetGrid at 0x2039283a590>
```



Inference : The Linear Regression plot between total and not_distracted shows linear line

```
sns.lmplot(x="no_previous",y="alcohol",data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x2038d694690>
```

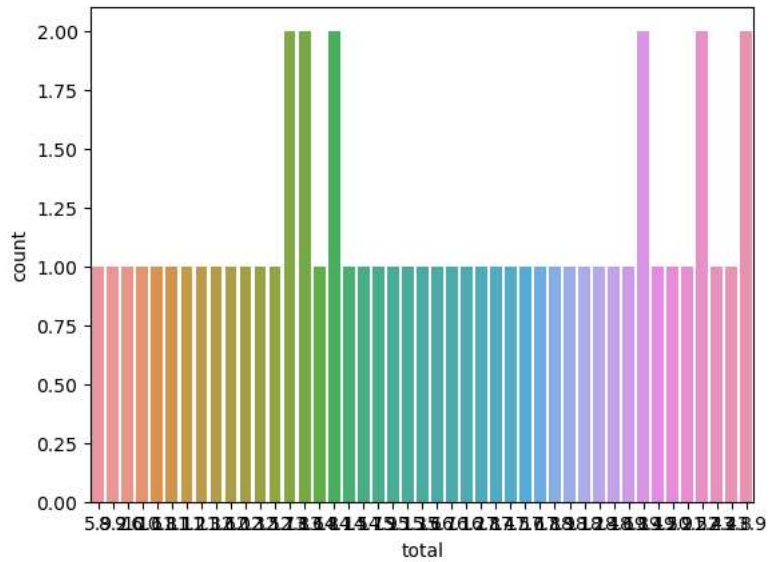


Inference : The Linear Regression plot between no_previous and alcohol shows linear line

13.Count Plot

```
sns.countplot(x='total',data=df)
```

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



Inference : From the countplot displays the count of observation in each line of total

```
sns.countplot(x='total', hue = 'alcohol',data=df)
```