

```
import seaborn as sns
```

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

```
df
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium
0	18.8	7.332	5.640	18.048	15.040	784.55
1	18.1	7.421	4.525	16.290	17.014	1053.48
2	18.6	6.510	5.208	15.624	17.856	899.47
3	22.4	4.032	5.824	21.056	21.280	827.34
4	12.0	4.200	3.360	10.920	10.680	878.41
5	13.6	5.032	3.808	10.744	12.920	835.50
6	10.8	4.968	3.888	9.396	8.856	1068.73
7	16.2	6.156	4.860	14.094	16.038	1137.87
8	5.9	2.006	1.593	5.900	5.900	1273.89
9	17.9	3.759	5.191	16.468	16.826	1160.13
10	15.6	2.964	3.900	14.820	14.508	913.15
11	17.5	9.450	7.175	14.350	15.225	861.18
12	15.3	5.508	4.437	13.005	14.994	641.96
13	12.8	4.608	4.352	12.032	12.288	803.11
14	14.5	3.625	4.205	13.775	13.775	710.46
15	15.7	2.669	3.925	15.229	13.659	649.06
16	17.8	4.806	4.272	13.706	15.130	780.45
17	21.4	4.066	4.922	16.692	16.264	872.51
18	20.5	7.175	6.765	14.965	20.090	1281.55
19	15.1	5.738	4.530	13.137	12.684	661.88
20	12.5	4.250	4.000	8.875	12.375	1048.78
21	8.2	1.886	2.870	7.134	6.560	1011.14
22	14.1	3.384	3.948	13.395	10.857	1110.61

23	9.6	2.208	2.784	8.448	8.448	777.18
24	17.6	2.640	5.456	1.760	17.600	896.07
25	16.1	6.923	5.474	14.812	13.524	790.32
26	21.4	8.346	9.416	17.976	18.190	816.21
27	14.9	1.937	5.215	13.857	13.410	732.28
28	14.7	5.439	4.704	13.965	14.553	1029.87
29	11.6	4.060	3.480	10.092	9.628	746.54
30	11.2	1.792	3.136	9.632	8.736	1301.52
31	18.4	3.496	4.968	12.328	18.032	869.85
32	12.3	3.936	3.567	10.824	9.840	1234.31
33	16.8	6.552	5.208	15.792	13.608	708.24
34	23.9	5.497	10.038	23.661	20.554	688.75
35	14.1	3.948	4.794	13.959	11.562	697.73
36	19.9	6.368	5.771	18.308	18.706	881.51
37	12.8	4.224	3.328	8.576	11.520	804.71
38	18.2	9.100	5.642	17.472	16.016	905.99
39	11.1	3.774	4.218	10.212	8.769	1148.99
40	23.9	9.082	9.799	22.944	19.359	858.97
41	19.4	6.014	6.402	19.012	16.684	669.31
42	19.5	4.095	5.655	15.990	15.795	767.91
43	19.4	7.760	7.372	17.654	16.878	1004.75
44	11.3	4.859	1.808	9.944	10.848	809.38
45	13.6	4.080	4.080	13.056	12.920	716.20
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

```
df.head()
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium
0	18.8	7.332	5.640	18.048	15.040	784.55
1	18.1	7.421	4.525	16.290	17.014	1053.48
2	18.6	6.510	5.208	15.624	17.856	899.47
3	22.4	4.032	5.824	21.056	21.280	827.34
4	12.0	4.200	3.360	10.920	10.680	878.41

	ins_losses	abbrev
0	145.08	AL
1	133.93	AK
2	110.35	AZ
3	142.39	AR
4	165.63	CA

```
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
```

```
df.shape
```

```
(51, 8)
```

```
###Handling null values
```

```
df.isnull().any()
```

```
total          False
speeding        False
alcohol         False
not_distracted False
no_previous     False
ins_premium     False
ins_losses      False
abbrev          False
dtype: bool
```

```
df.isnull().sum()
```

```
total          0
speeding        0
alcohol         0
not_distracted  0
no_previous     0
ins_premium     0
ins_losses      0
abbrev          0
dtype: int64
```

SEPARATION OF DEPENDENT AND INDEPENDENT VARIABLES

```
X=df.iloc[:,1]
```

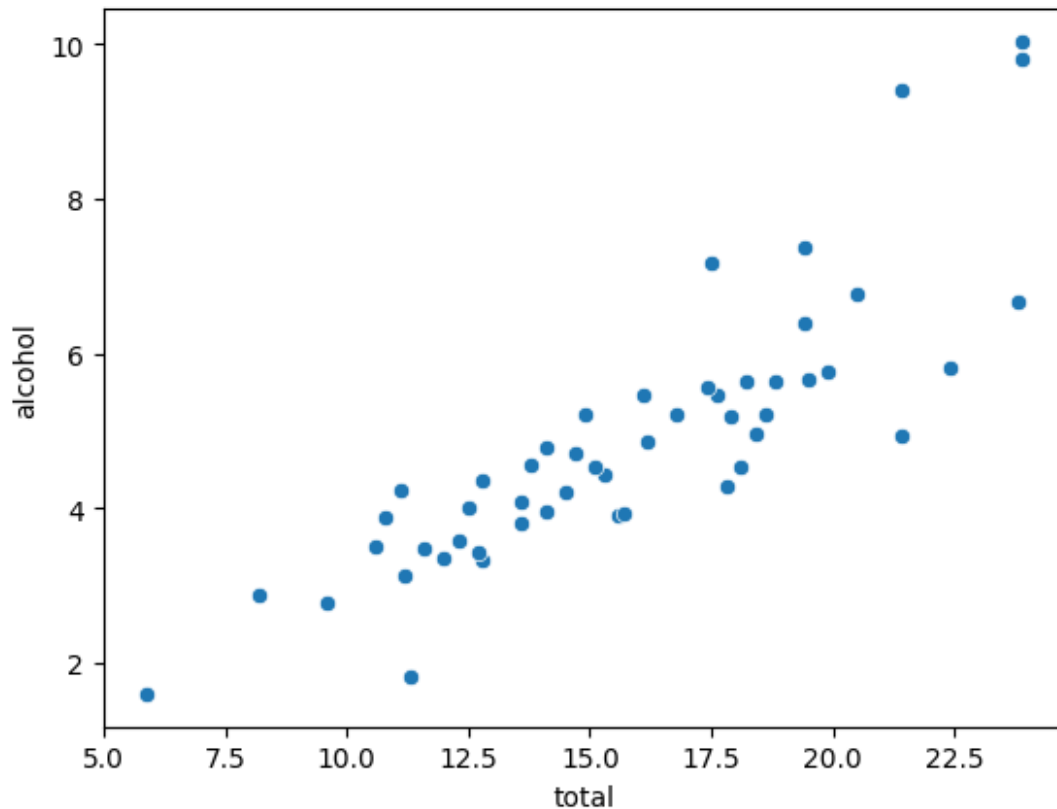
```
X
```

```
total
0    18.8
1    18.1
2    18.6
3    22.4
4    12.0
5    13.6
6    10.8
7    16.2
8     5.9
9    17.9
10   15.6
11   17.5
12   15.3
13   12.8
14   14.5
15   15.7
```

16	17.8
17	21.4
18	20.5
19	15.1
20	12.5
21	8.2
22	14.1
23	9.6
24	17.6
25	16.1
26	21.4
27	14.9
28	14.7
29	11.6
30	11.2
31	18.4
32	12.3
33	16.8
34	23.9
35	14.1
36	19.9
37	12.8
38	18.2
39	11.1
40	23.9
41	19.4
42	19.5
43	19.4
44	11.3
45	13.6
46	12.7
47	10.6
48	23.8
49	13.8
50	17.4

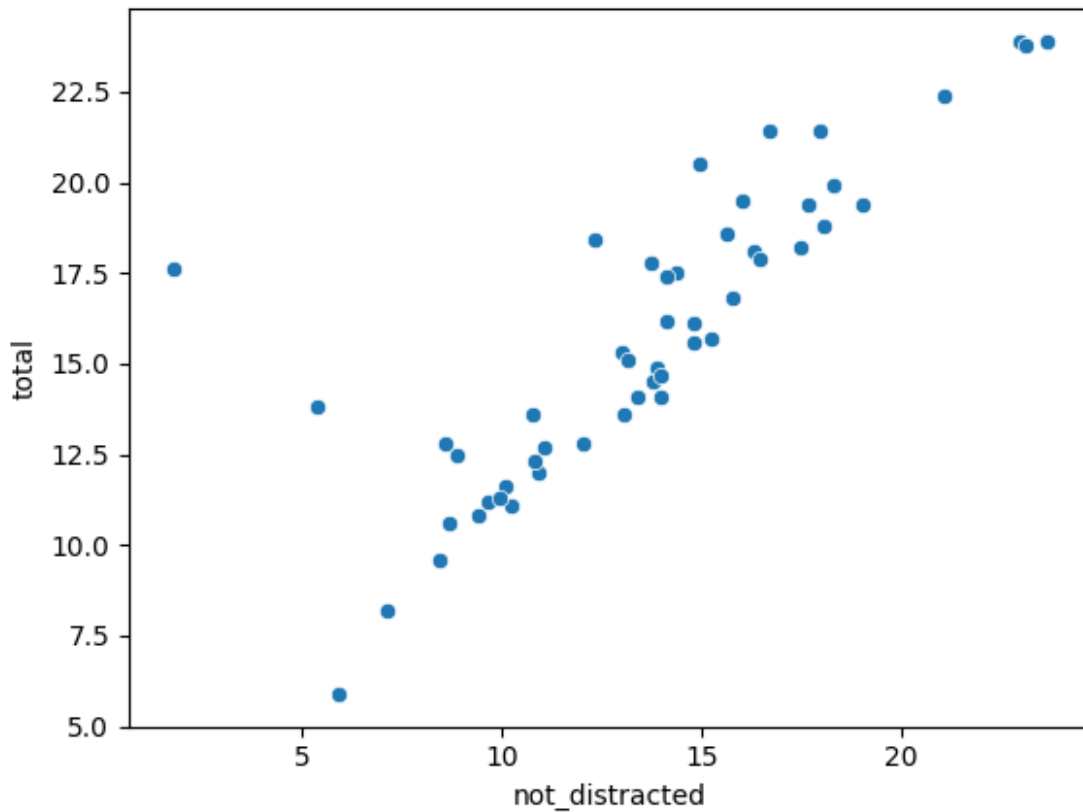
Data visualisation

```
sns.scatterplot(x="total",y="alcohol",data=df)  
<Axes: xlabel='total', ylabel='alcohol'>
```



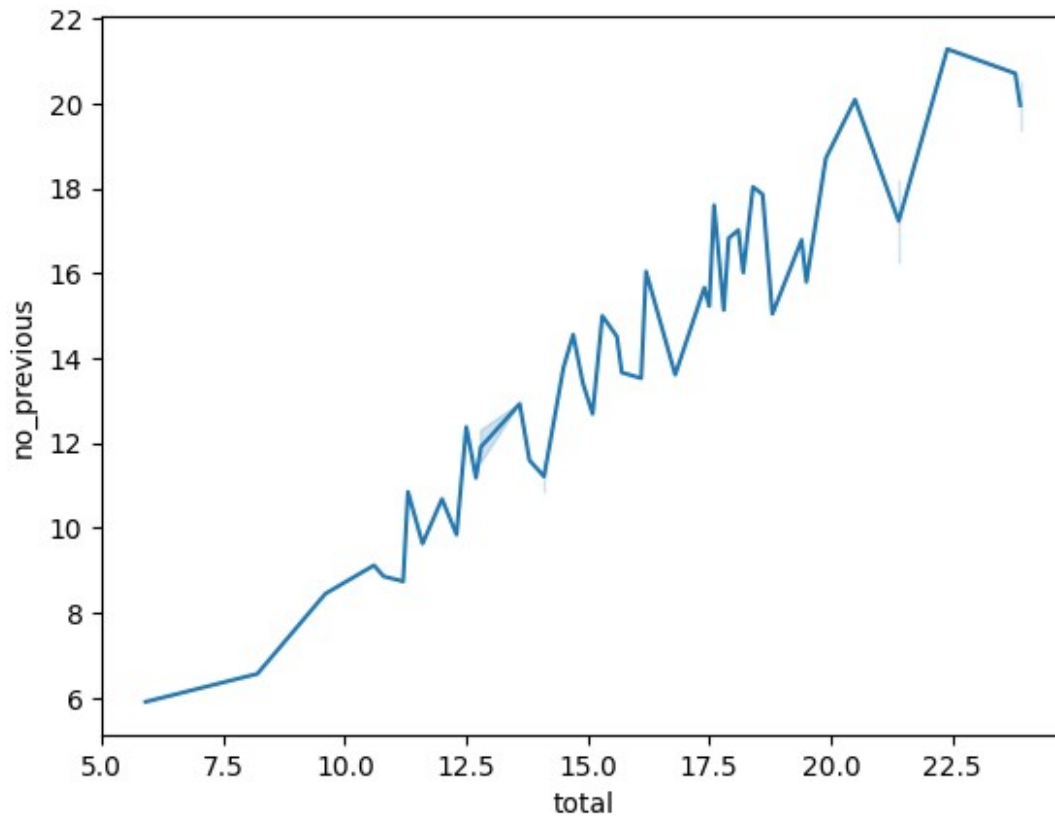
The graph infers that the total accidents that have taken place are directly proportional to the amount of alcohol consumed.

```
sns.scatterplot(y="total",x="not_distracted",data=df)  
<Axes: xlabel='not_distracted', ylabel='total'>
```



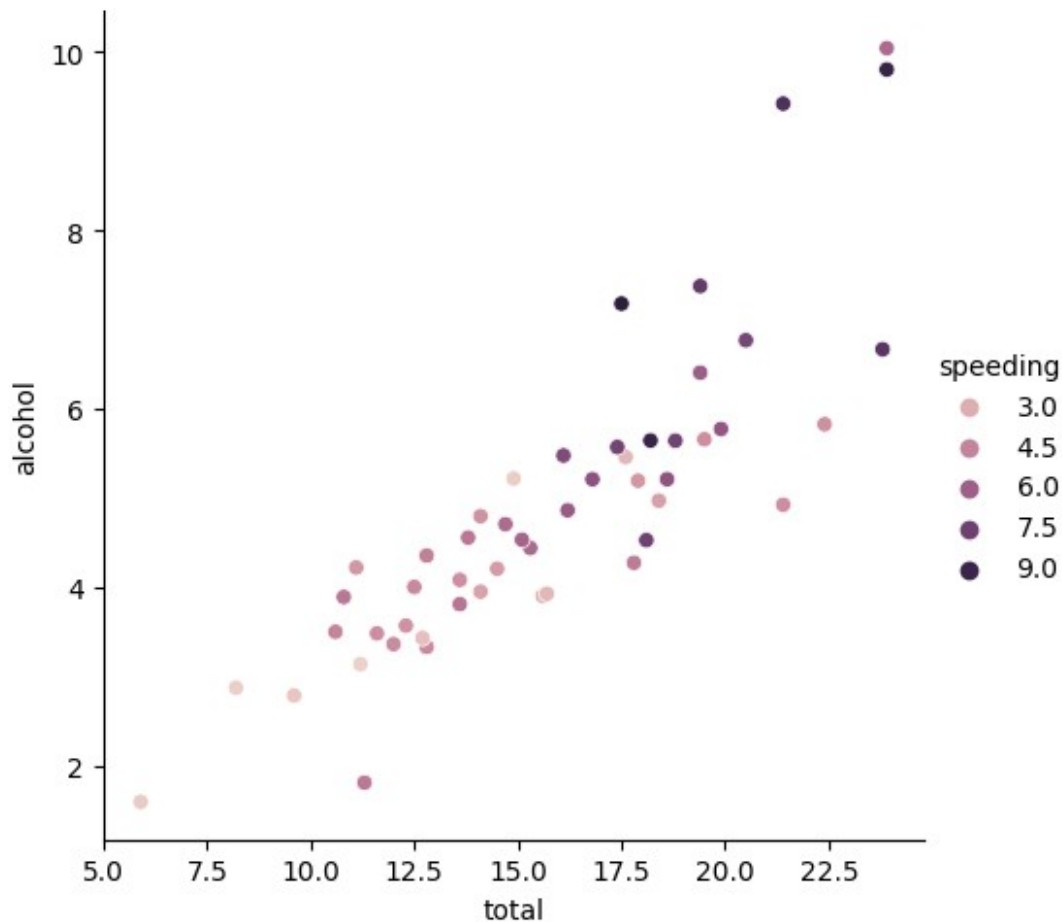
Here the graph infers that the total accidents that have taken place are directly proportional to the accidents that have taken place because of not being distracted.

```
sns.lineplot(x="total",y="no_previous",data=df)  
<Axes: xlabel='total', ylabel='no_previous'>
```

Here the graph infers that the total accidents that have taken place are directly proportional to accidents that not taken place prior.

```
sns.relplot(x="total",y="alcohol",data=df,hue="speeding")  
<seaborn.axisgrid.FacetGrid at 0x7e8085c19390>
```



Here the graph infers that the total accidents that have taken place due to consumption of alcohol are directly proportional to speed with which the car was driven

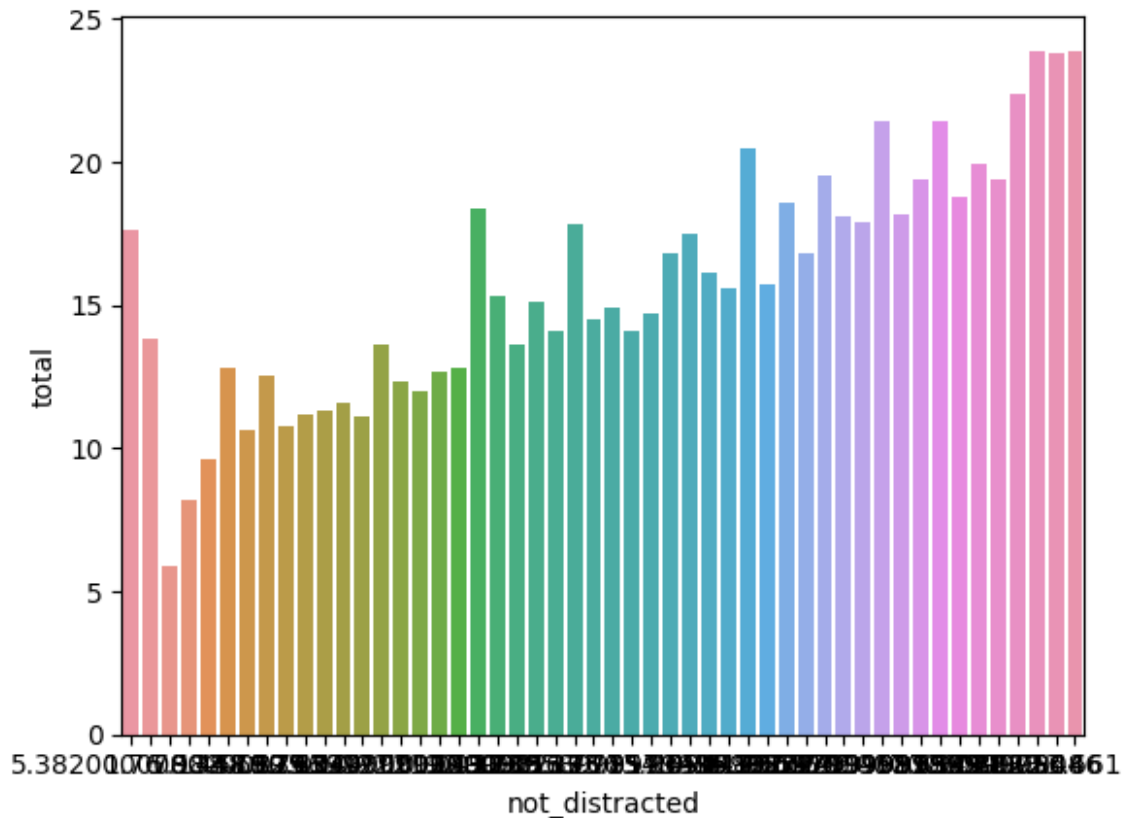
```
sns.barplot(x="not_distracted",y="total",data=df,ci=None)
```

```
<ipython-input-29-3c11a350536e>:1: FutureWarning:
```

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

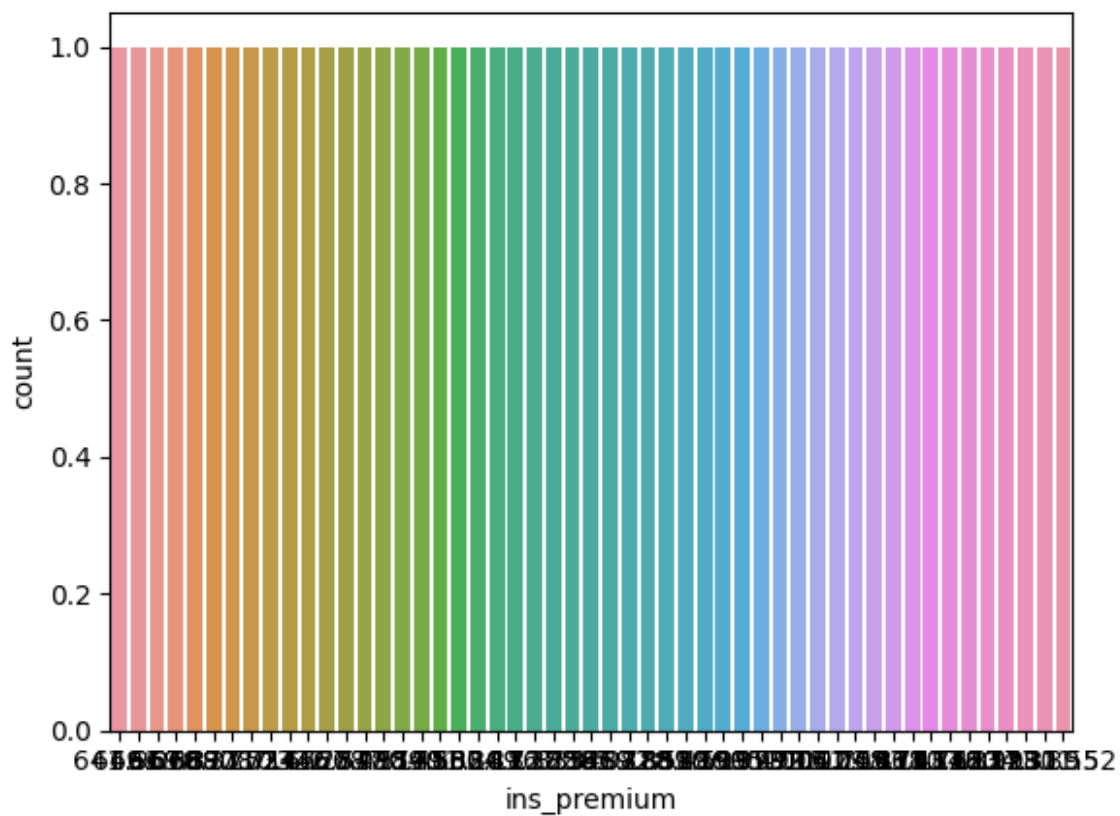
```
sns.barplot(x="not_distracted",y="total",data=df,ci=None)
```

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



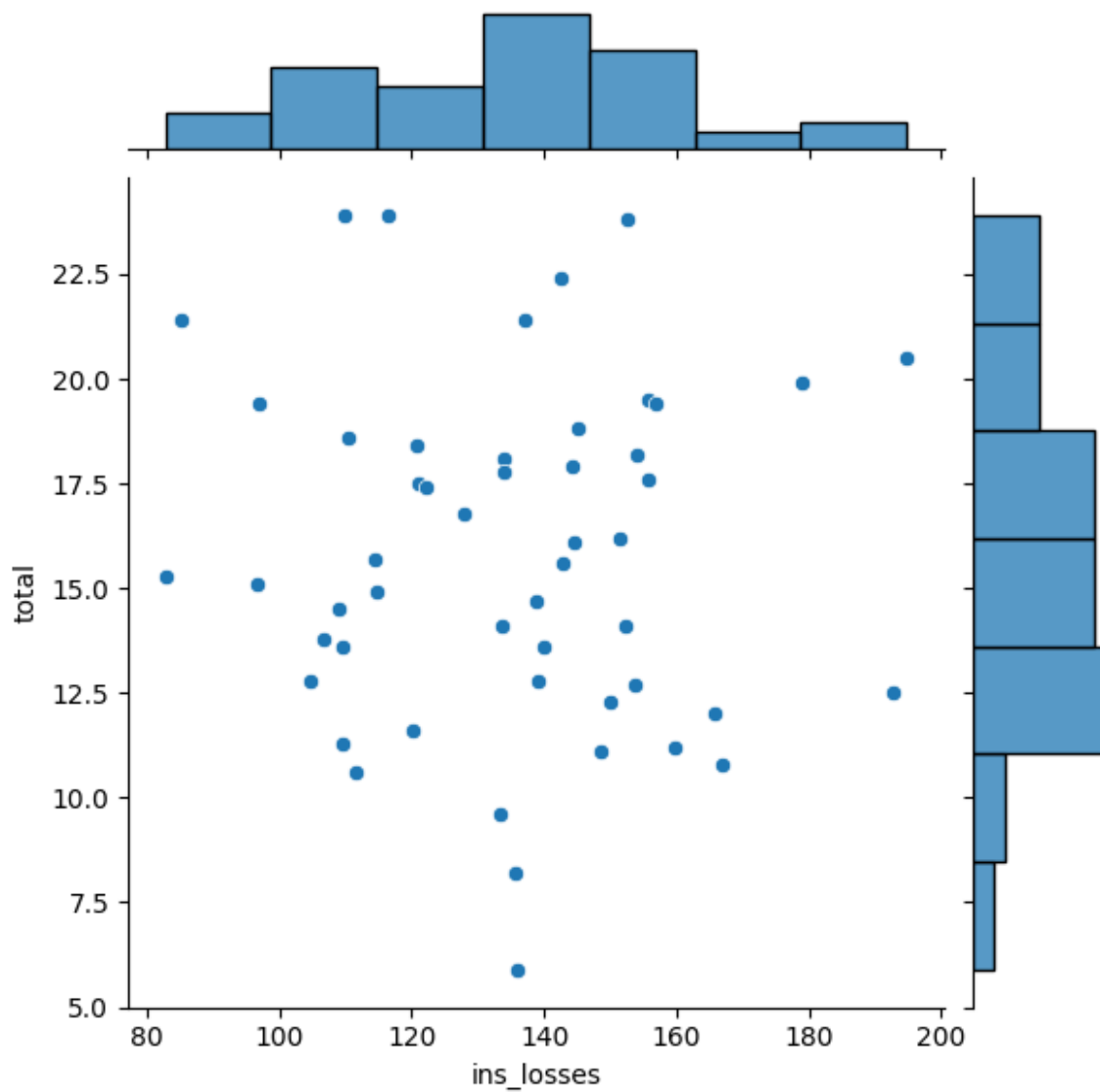
The graph infers that the total accidents that have taken place are directly proportional to the reason being that the driver was not distracted

```
sns.countplot(x="ins_premium",data=df)
<Axes: xlabel='ins_premium', ylabel='count'>
```



The graph gives us the count of total ins_premium

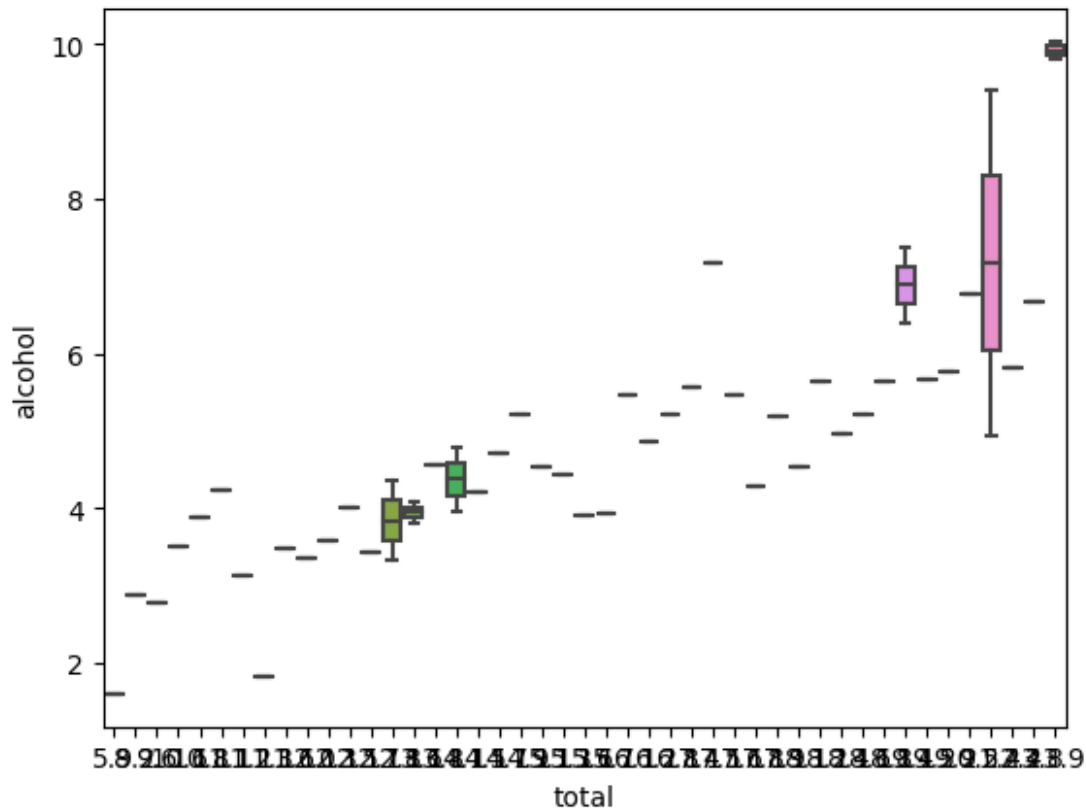
```
sns.jointplot(x="ins_losses",y="total",data=df)
<seaborn.axisgrid.JointGrid at 0x7e8088bb9870>
```



The above graph gives data plotted in two different types.

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

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



Here the graph infers that the total accidents that have taken place are directly proportional to the amount of alcohol consumed

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

```
<ipython-input-33-d9e4e726f1fe>: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.
```

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

	total	speeding	alcohol	not_distracted
no_previous \				
total	1.000000	0.611548	0.852613	0.827560
0.956179				
speeding	0.611548	1.000000	0.669719	0.588010
0.571976				
alcohol	0.852613	0.669719	1.000000	0.732816
0.783520				
not_distracted	0.827560	0.588010	0.732816	1.000000
0.747307				

no_previous	0.956179	0.571976	0.783520	0.747307
1.000000				
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856
0.156895				
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970
0.006359				

	ins_premium	ins_losses
total	-0.199702	-0.036011
speeding	-0.077675	-0.065928
alcohol	-0.170612	-0.112547
not_distracted	-0.174856	-0.075970
no_previous	-0.156895	-0.006359
ins_premium	1.000000	0.623116
ins_losses	0.623116	1.000000

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

```
<Axes: >
```



The graph shows us the intensity of variables of the above mentioned dataset

