```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

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

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes', 'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri', 'geyser', 'glue', 'healthexp', 'iris', 'mp
```

```
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
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

```
df.size
```

```
408
```

```
df.shape
```

```
(51, 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
```

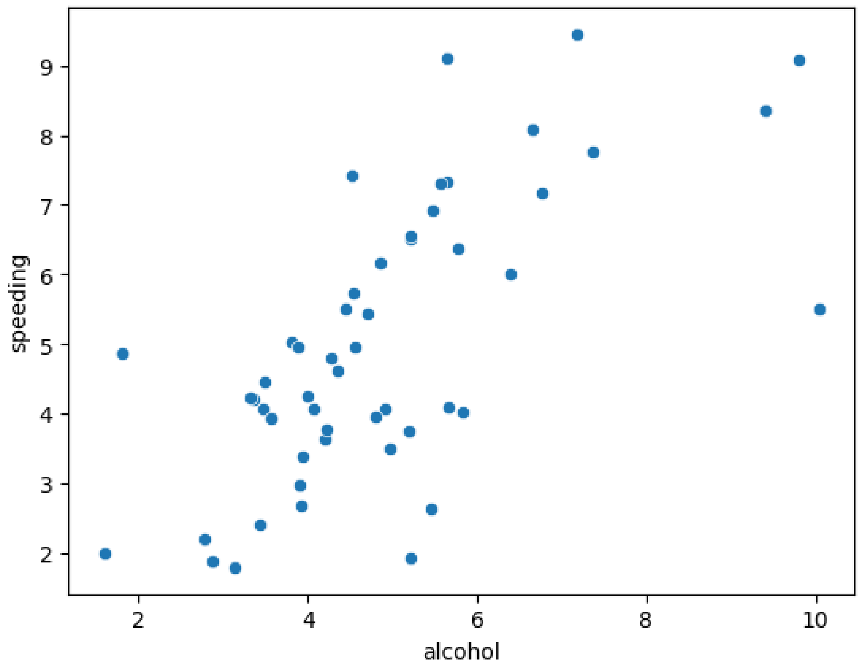
```
df.head()
```

	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

Scatter plot

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

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



Double-click (or enter) to edit

Double-click (or enter) to edit

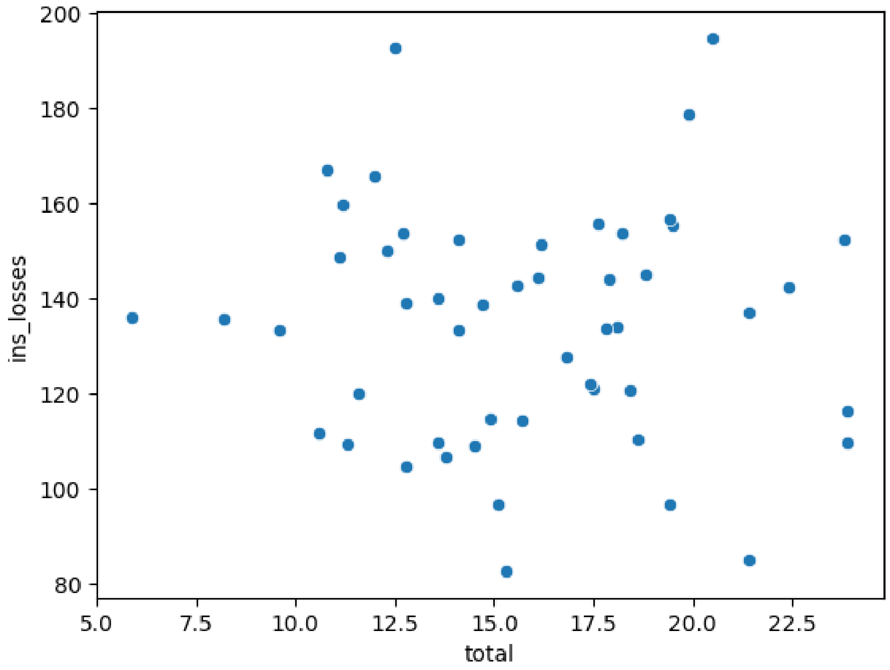
Inference:The above graph shows the scatter plot of speeding and alcohol from the car crahses data set in seaborn library.A scatterplot with "alcohol" on the x-axis and "speeding" on the y-axis.It visualizes the relationship between alcohol involvement and speeding in the dataset.Each point on the plot represents a data point, showing the combination of alcohol involvement and speeding for each observation.Alcohol is on x axis.

Double-click (or enter) to edit

Double-click (or enter) to edit

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

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



Double-click (or enter) to edit

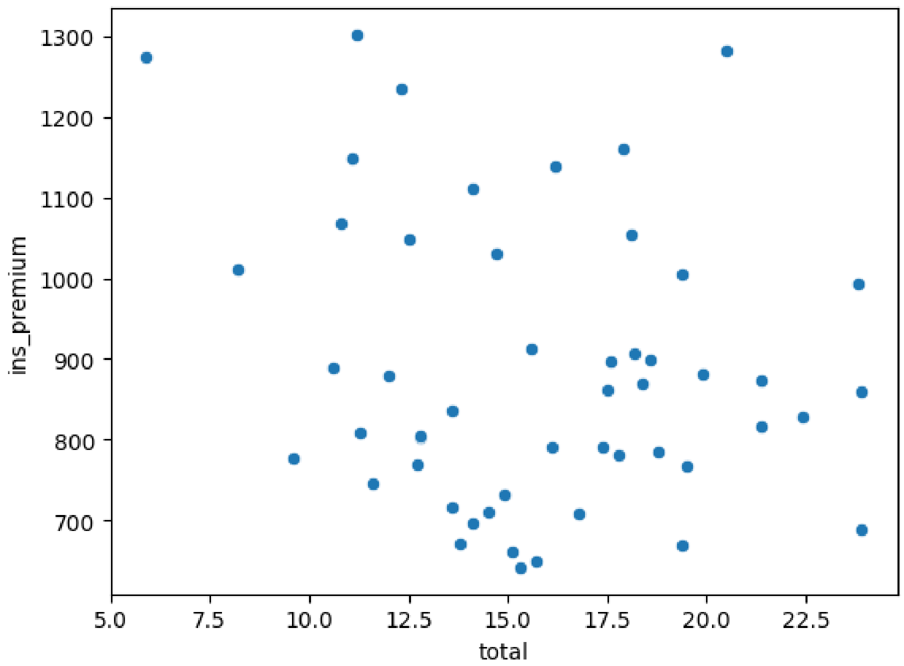
Double-click (or enter) to edit

Inference: The above graph shows the scatter plot of ins_losses and total from the car crahses data set in seaborn library.Total is on x axis ins_losses is on y axis.

Double-click (or enter) to edit

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

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



Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Inference:A scatterplot with "total" on the x-axis and "ins_premium" on the y-axis.It visualizes the relationship between the total miles driven and insurance premiums in the dataset.Each point on the plot represents an observation, showing how the total miles driven relates to the insurance premium for each data point.

Double-click (or enter) to edit

Double-click (or enter) to edit

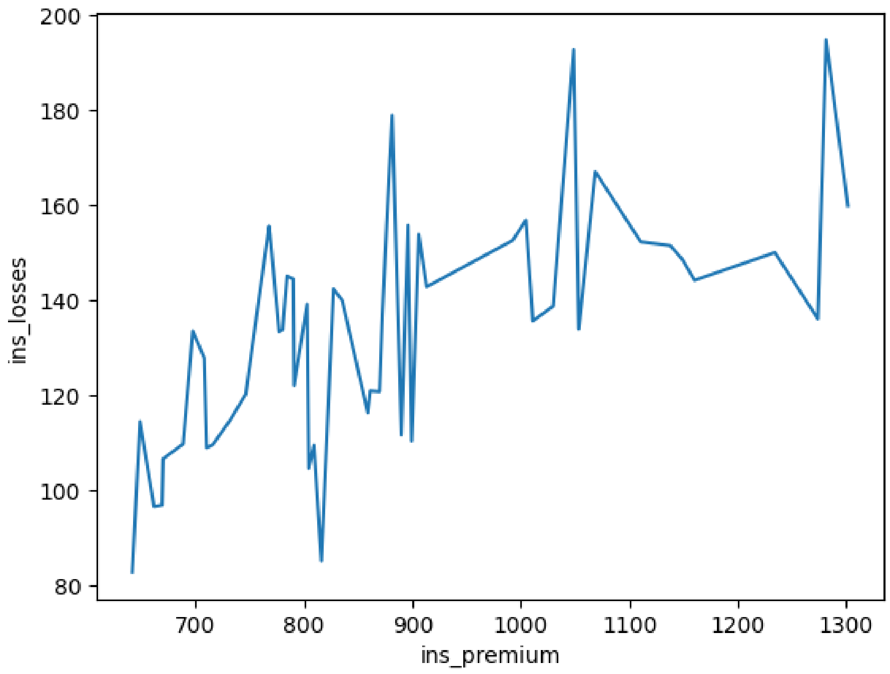
Double-click (or enter) to edit

Double-click (or enter) to edit

Line Plot

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

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



Double-click (or enter) to edit

Double-click (or enter) to edit

Inference :A lineplot with "ins_premium" on the x-axis and "ins_losses" on the y-axis.It visualizes the relationship between insurance premiums and insurance losses in the dataset. The plot shows how changes in insurance premiums are related to changes in insurance losses. The line's direction and slope can indicate whether there is a positive or negative correlation between these variables. A positive slope suggests that as insurance premiums increase, insurance losses also tend to increase, and vice versa for a negative slope.

Distribution plot

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

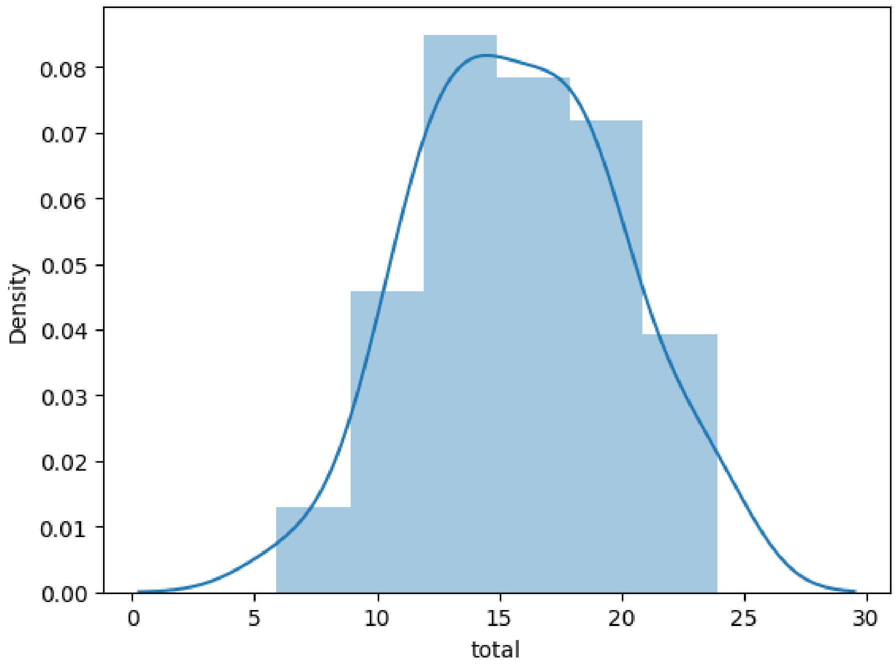
```
<ipython-input-15-0d5ead9bfd1a>:1: UserWarning:

`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'>
```



Double-click (or enter) to edit

Double-click (or enter) to edit

Inference : The above graph shows the Dist plot of total from the car crahses data set.Distplot provides insights into the car_crashes dataset's distribution shape, central tendency, spread, skewness, and potential outliers. It's a visual tool for understanding data patterns and can hint at underlying statistical properties. Total is on x axis and density is on y axis.

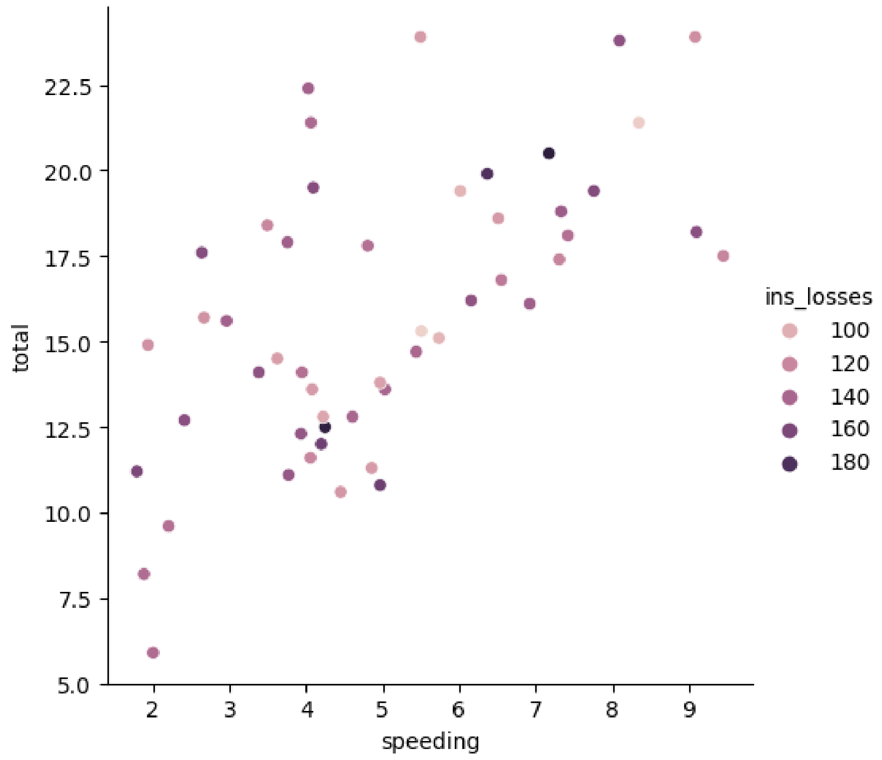
Double-click (or enter) to edit

Double-click (or enter) to edit

Relational Plot

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

```
<seaborn.axisgrid.FacetGrid at 0x7d279728fa00>
```



Double-click (or enter) to edit

Double-click (or enter) to edit

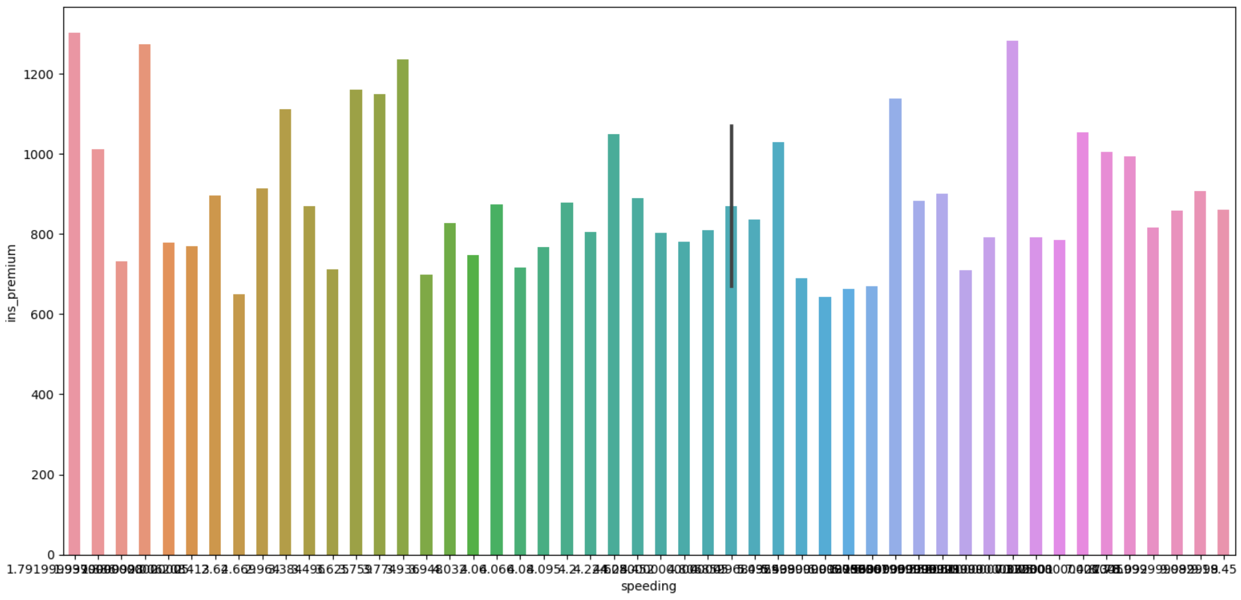
Inference: This is relplot refers to a relational plot,shows the realtion between alcohol and speeding.Identifies the patterns higher speeds and alcohol leading to more crashes.It helps in understanding correlations.Alcohol is on y axis.Speeding is on x axis.

```
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
```

BAR PLOT

```
plt.figure(figsize=(17, 8))
sns.barplot(data=df, x="speeding", y="ins_premium", width=0.5)
plt.show()
```

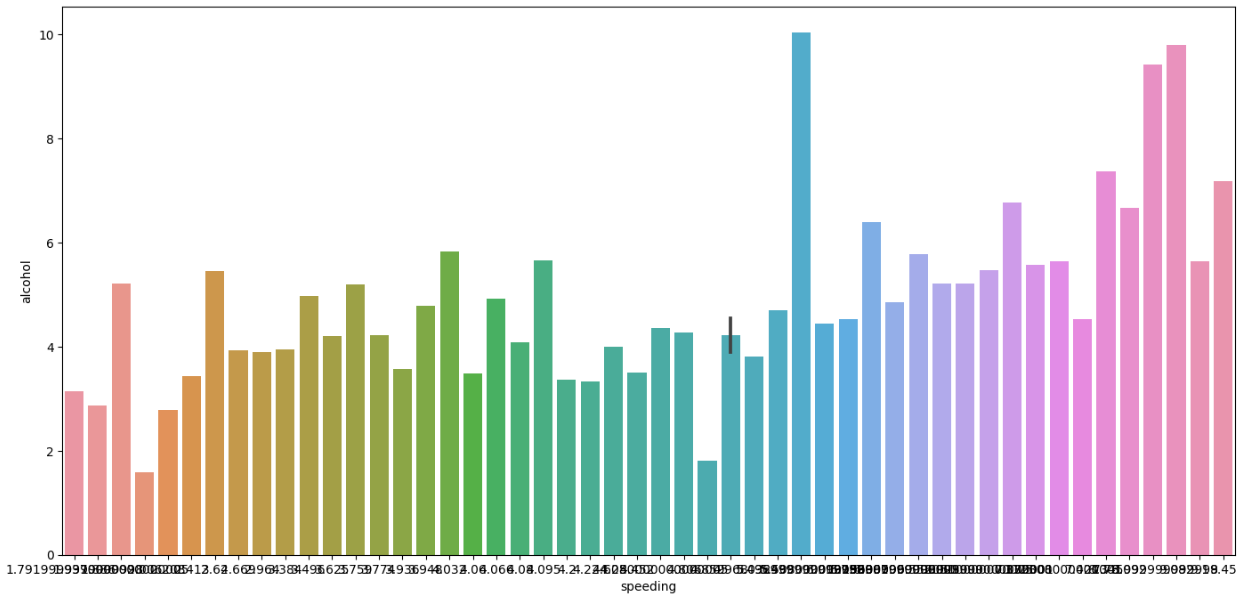


Double-click (or enter) to edit

Inference: The sns.barplot plot with "speeding" on the x-axis and "ins_premium" on the y-axis, with a bar width of 0.5.It visualizes the relationship between speeding (excessive speeding) and insurance premiums. Higher levels of speeding may be associated with higher insurance premiums. The bar width of 0.5 is used for better visibility and distinction between the bars.

```
plt.figure(figsize=(17, 8))
sns.barplot(data=df,x="speeding",y="alcohol")
```

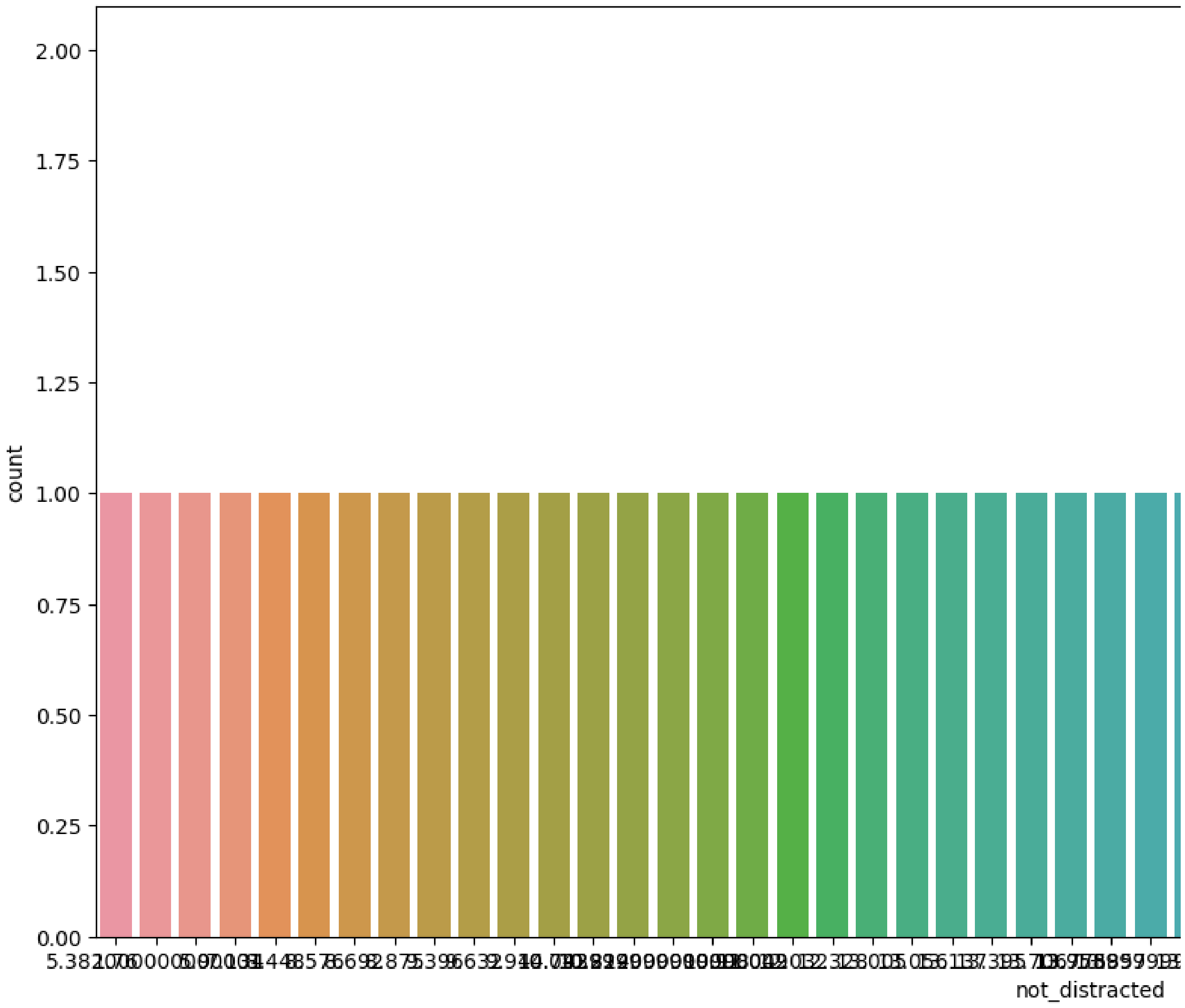
```
<Axes: xlabel='speeding', ylabel='alcohol'>
```



Double-click (or enter) to edit

```
plt.figure(figsize=(17, 8))
sns.countplot(x="not_distracted",data=df)
```

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



Double-click (or enter) to edit

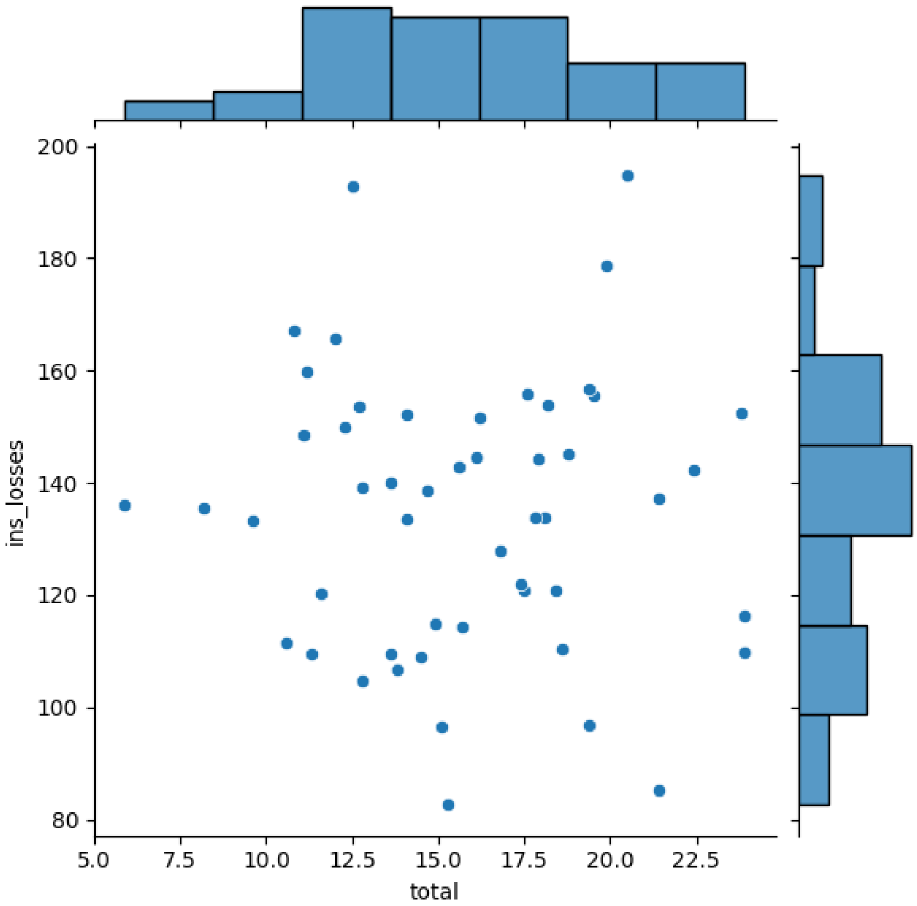
Inference: Barplot with speeding on the x-axis and alcohol on the y-axis. It visualizes the relationship between speeding and alcohol involvement in car crashes. The height of the bars represents the average or total alcohol involvement in crashes for different speeding levels. Higher bars indicate a stronger association between speeding and alcohol involvement. It can help identify if there's a significant correlation or difference in alcohol involvement based on speeding levels.

Double-click (or enter) to edit

JOINT PLOT

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

```
<seaborn.axisgrid.JointGrid at 0x7854995158a0>
```

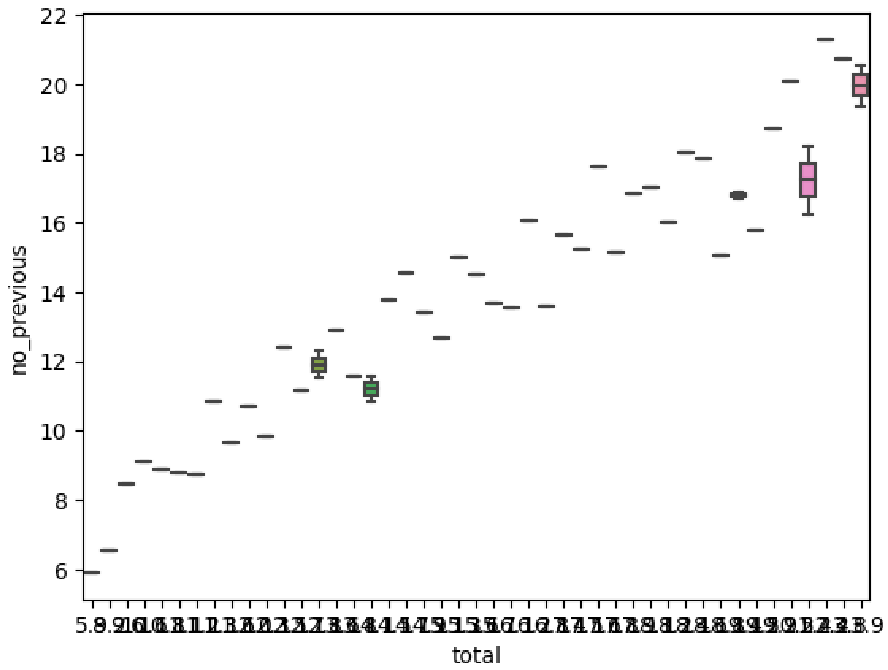


Inference:A jointplot between "total" and "ins_losses" variables in a dataset. It visualizes the relationship between total miles driven and insurance losses. The scatter plot in the center of the jointplot shows how these two variables are distributed together. The correlation between total miles driven and insurance losses can be assessed by the scatter plot's pattern. If the points cluster around a line, it indicates a linear relationship. The marginal histograms on the top and right sides provide insights into the individual distributions of each variable. It helps assess whether there's a correlation between the two variables and whether higher mileage leads to higher insurance losses or vice versa.Total is on x axis ins_losses is on y axis.

Box Plot

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

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



Inference:A boxplot with "total" on the x-axis and "no_previous" on the y-axis.It visualizes the relationship between the total miles driven and the number of previous accidents.The boxplot shows the distribution of the "no_previous" variable at different levels of "total" miles. It helps assess whether there are differences in the number of previous accidents among different mileage groups.

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

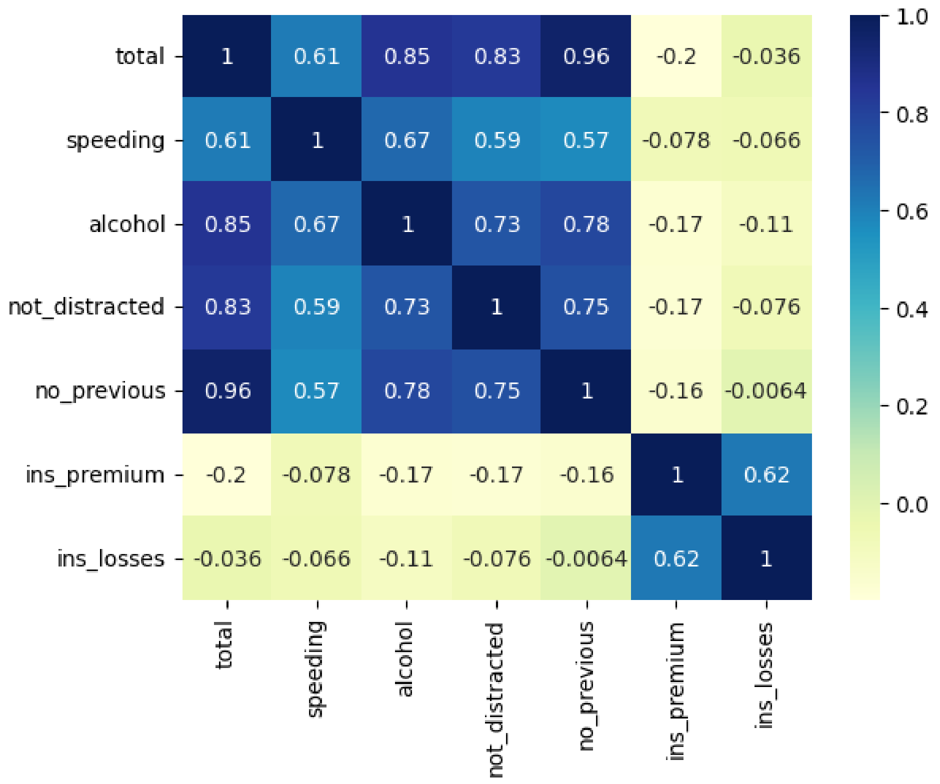
```
<ipython-input-60-7d5195e2bf4d>:1: FutureWarning: The default value of numeric_only in DataFrame.corr i
  corr=df.corr()
```

	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

Inference:A correlation matrix provides insights into the pairwise relationships between variables in a dataset.

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

<Axes: >



Inference: This heatmap visually displays the strengths and directions of relationships between variables using colors. It helps identify strong and weak correlations, enabling insights into how variables relate to each other in the dataset.

```
#NAME:HANSINI VADDEY
```

NAME:HANSINI VADDEY

REGD NO:21BDS0345

ASSIGNMENT 2

✓ 0s completed at 10:53 PM