1.Take car crashes dataset from seaborn library

```
!pip install seaborn
```

```
    Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.12.2)
    Requirement already satisfied: numpy!=1.24.0,>=1.17 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.23.5)
    Requirement already satisfied: pandas>=0.25 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.5.3)
    Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /usr/local/lib/python3.10/dist-packages (from seaborn) (3.7.1)
    Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.1.0)
    Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
    Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.42.1)
    Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.5)
    Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
    Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (9.4.0)
    Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.1.1)
    Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
    Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.25->seaborn) (2023.3.post1)
    Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
```

2. load the dataset

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
```

```
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

`df.info`

```
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  >
```

```
41  19.4  6.014  6.402  19.012  16.684  669.31  96.87  SD
```

`df.head(10)`

	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

3.Perfrom Data Visualization

1. Scatter plot

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

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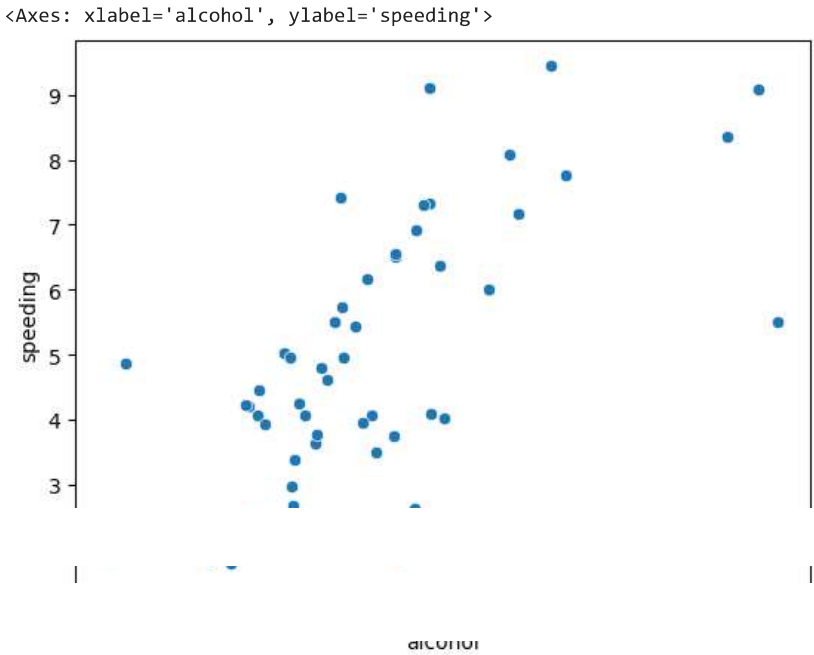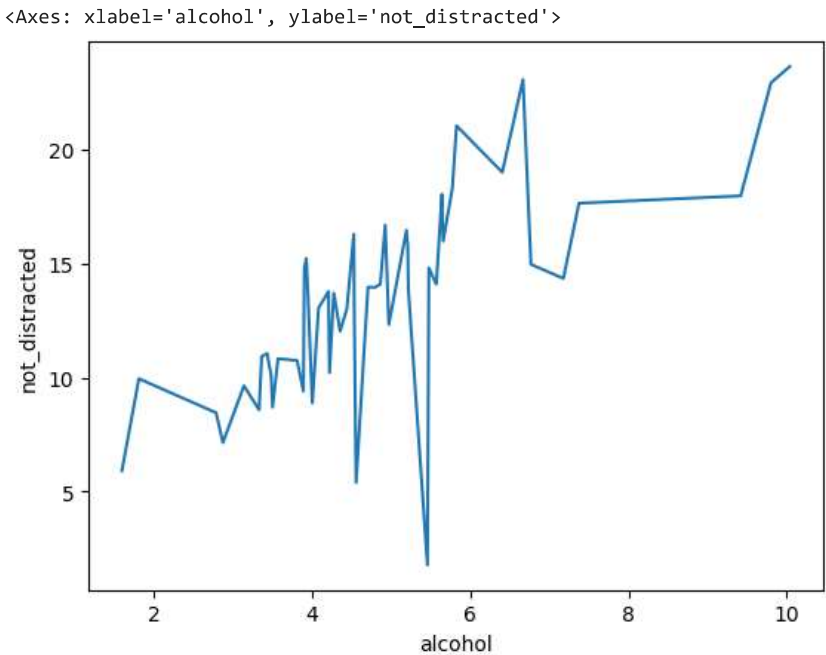

Inference: In the above graph Speeding increases as alcohol increses. Alcohol is taken on x-axis and speeding is taken on y-axis.

2. Line Plot

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

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


Inference: Line plot with alcohol on x-axis and not_distracted on y-axis is plotted. In the graph not_distracted increases as alcohol increases.

3. Distribution Plot

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

```
<ipython-input-14-f4c3808e7f5a>: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'>
```
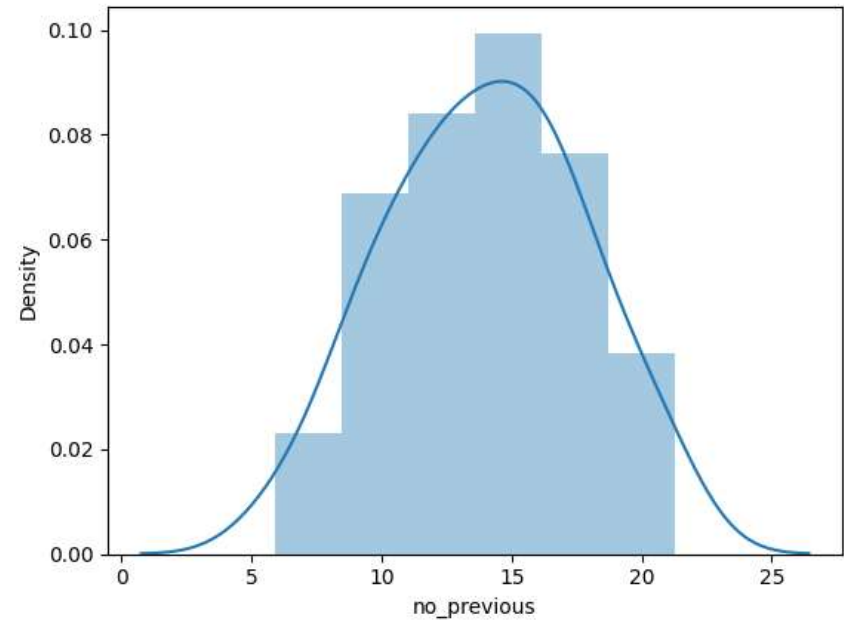
Inference: In this graph x-axis represents no_previous car crashes and y-axis represents the probability density. The density of previous car crashes increases until 15 and later decreases.

4. Relational plot

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

```
<seaborn.axisgrid.FacetGrid at 0x7d1d33a765f0>
```
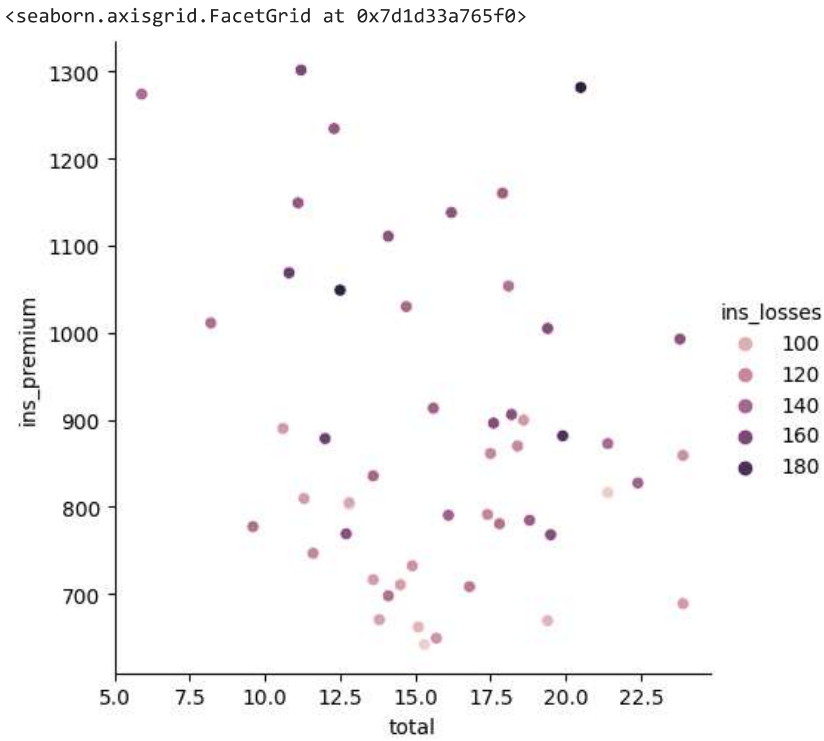

Inference: Relational plot with total on x-axis and ins_premium on y-axis with hue as ins_losses is plotted. It allows us to visualise how variables on a dataset relate to each other

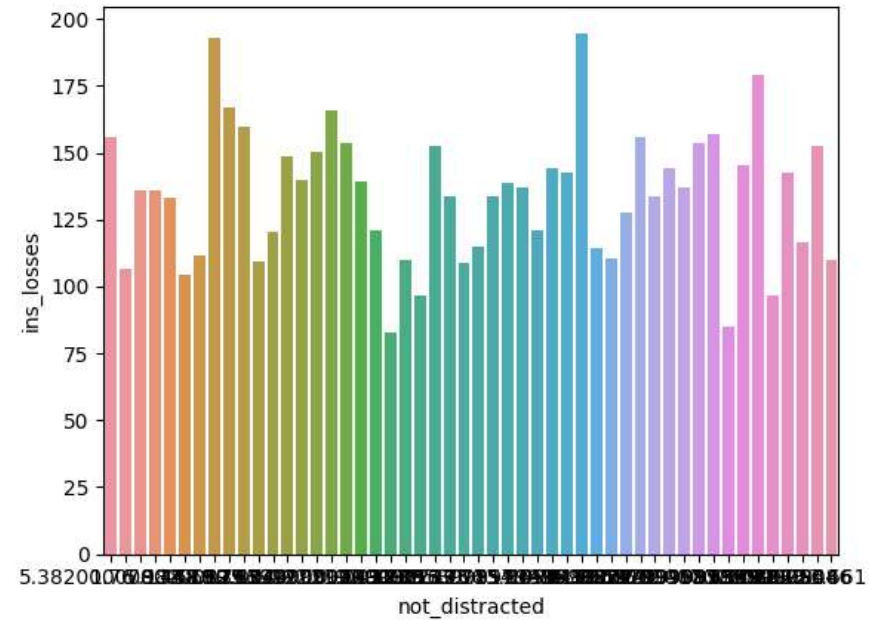5. Bar Plot

```
sns.barplot(data=df, x="not_distracted",y="ins_losses",errorbar=None)
sns.figure(figsize(17,8))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-26-56ed29c95608> in <cell line: 2>()
      1 sns.barplot(data=df, x="not_distracted",y="ins_losses",errorbar=None)
----> 2 sns.figure(figsize(17,8))

AttributeError: module 'seaborn' has no attribute 'figure'
```
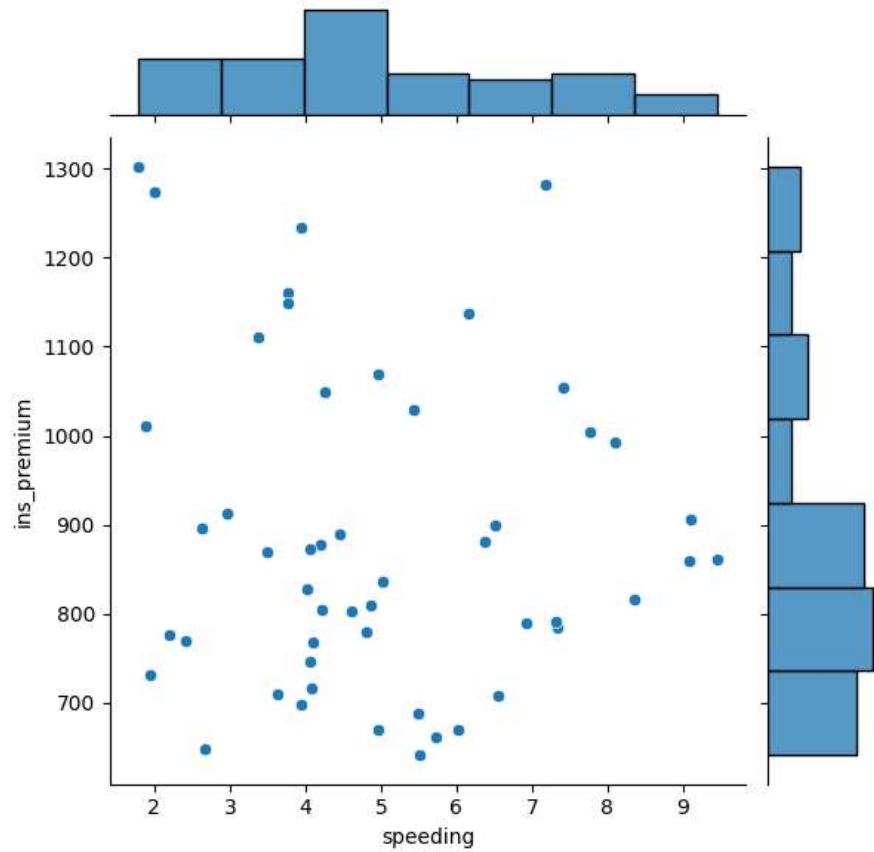

Inference: Bar graph is plotted with not_distracted on x-axis and ins_losses on y-axis.

6. Joint Plot

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

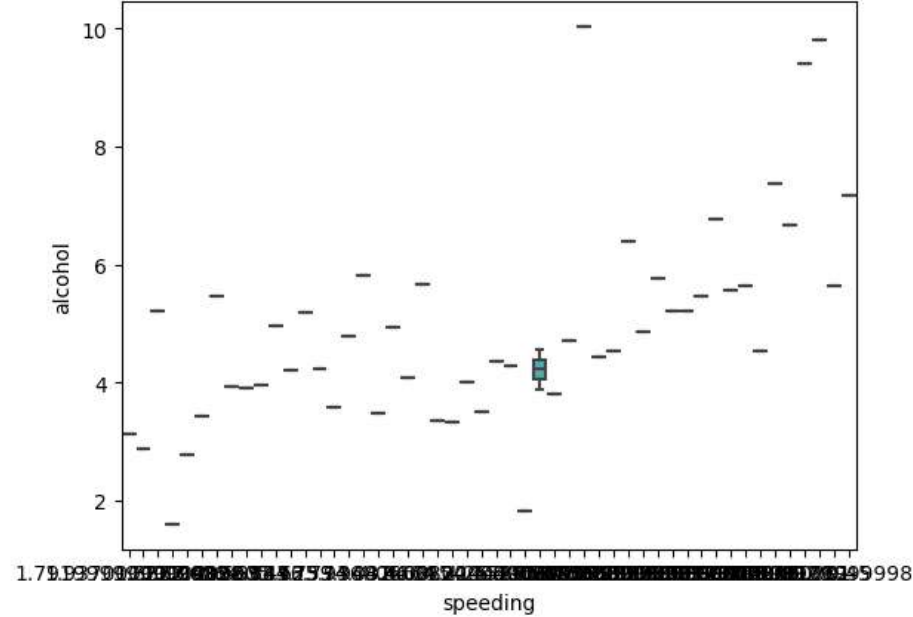```
<seaborn.axisgrid.JointGrid at 0x7d1d20ebe320>
```



Inference: Joint plot with speeding on x axis and ins_premium on y-axis is plotted.

7. Box Plot

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

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


Inference: Box plot with speeding on x axis and alcohol on y-axis is plotted.

Correlation

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

```
<ipython-input-19-5dc07b557697>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only
  cr=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

8. Heat map

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

```
<Axes: >
```
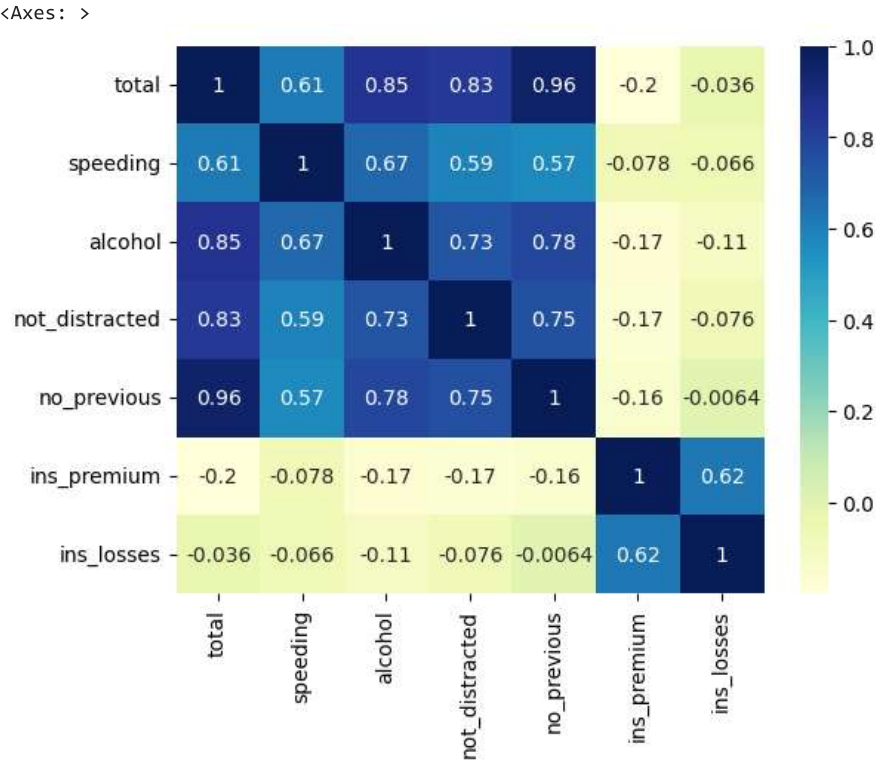

Inference: Values above 0.5 are highly correlated and below 0.5 are less correlated.

9. Scatter plot

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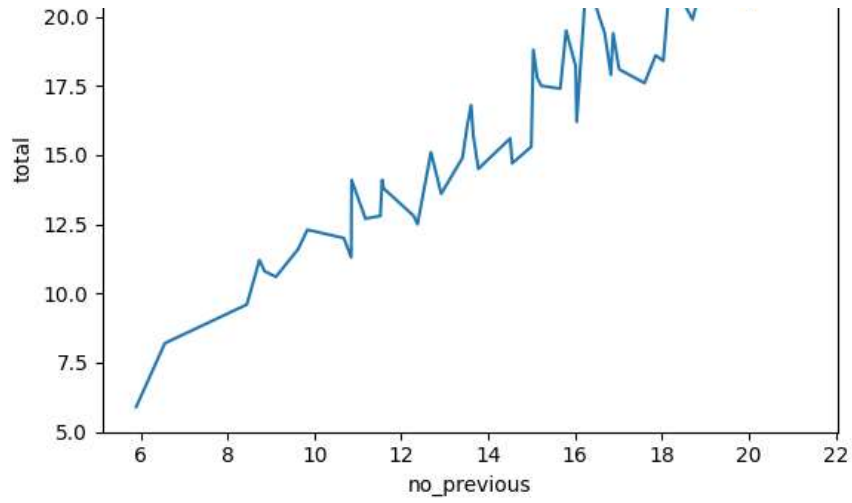
```
<Axes: xlabel='total', ylabel='ins_losses'>
```
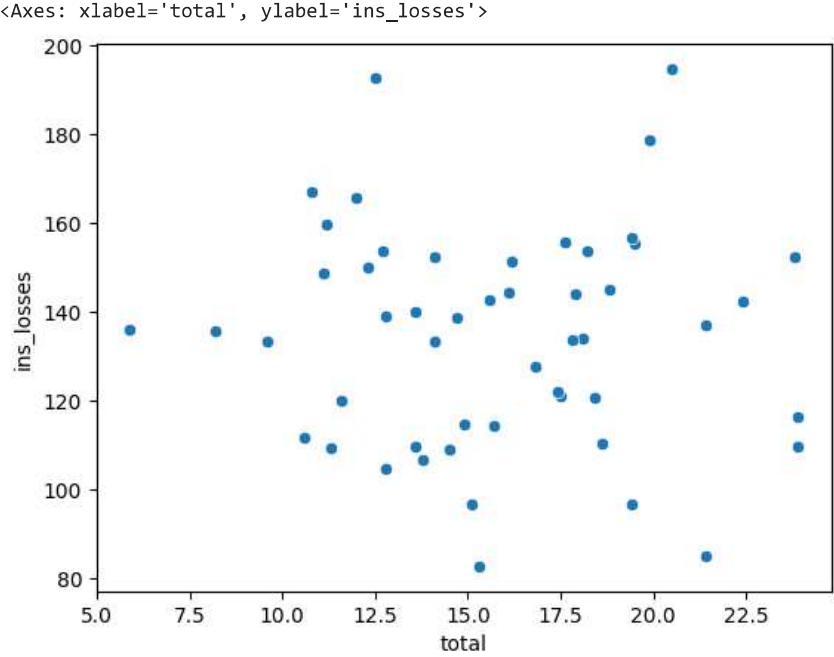

Inference: Scatter plot with total on x-axis and ins_losses on y-axis is plotted. In the graph ins_losses is maximum when total is 15.0 and 20.0.

10. Line plot

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

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

Inference: In this graph, as no_previous car crashes increases total also increases.

✓ 0s   completed at 11:00 PM