

```
In [51]: import seaborn as sns  
import matplotlib.pyplot as plt
```

```
In [52]: df=sns.load_dataset('car_crashes')  
df
```

Out[52]:

	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

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
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

In [53]: `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
```

In [54]: `df.head()`

Out[54]:

	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

total -> Number of drivers involved in fatal collisions per billion miles

speeding -> Percentage Of Drivers Involved In Fatal Collisions Who Were Speeding

alcohol -> Percentage Of Drivers Involved In Fatal Collisions Who Were Alcohol-Impaired

not_distracted -> Percentage Of Drivers Involved In Fatal Collisions Who Were Not Distracted

no_previous -> Percentage Of Drivers Involved In Fatal Collisions Who Had Not Been Involved In Any Previous Accidents

ins_premium -> Car Insurance Premiums

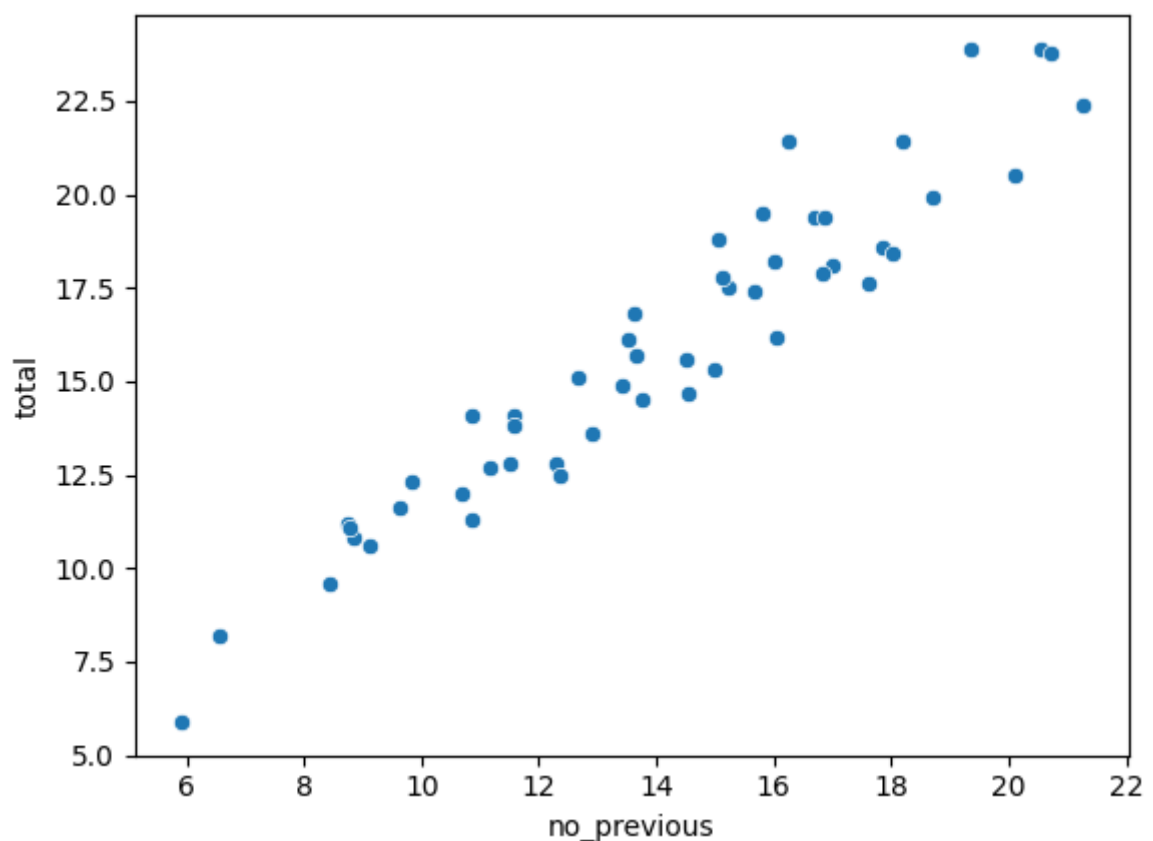
ins_losses -> Losses incurred by insurance companies for collisions per insured driver

abbrev -> USA states

ScatterPlot

```
In [55]: sns.scatterplot(x="no_previous",y="total",data=df)
```

```
Out[55]: <Axes: xlabel='no_previous', ylabel='total'>
```



Inference: from the plot we can say that as the people with no previous accident can also face accident, because of which total number of accidents also increases.

Line Graph

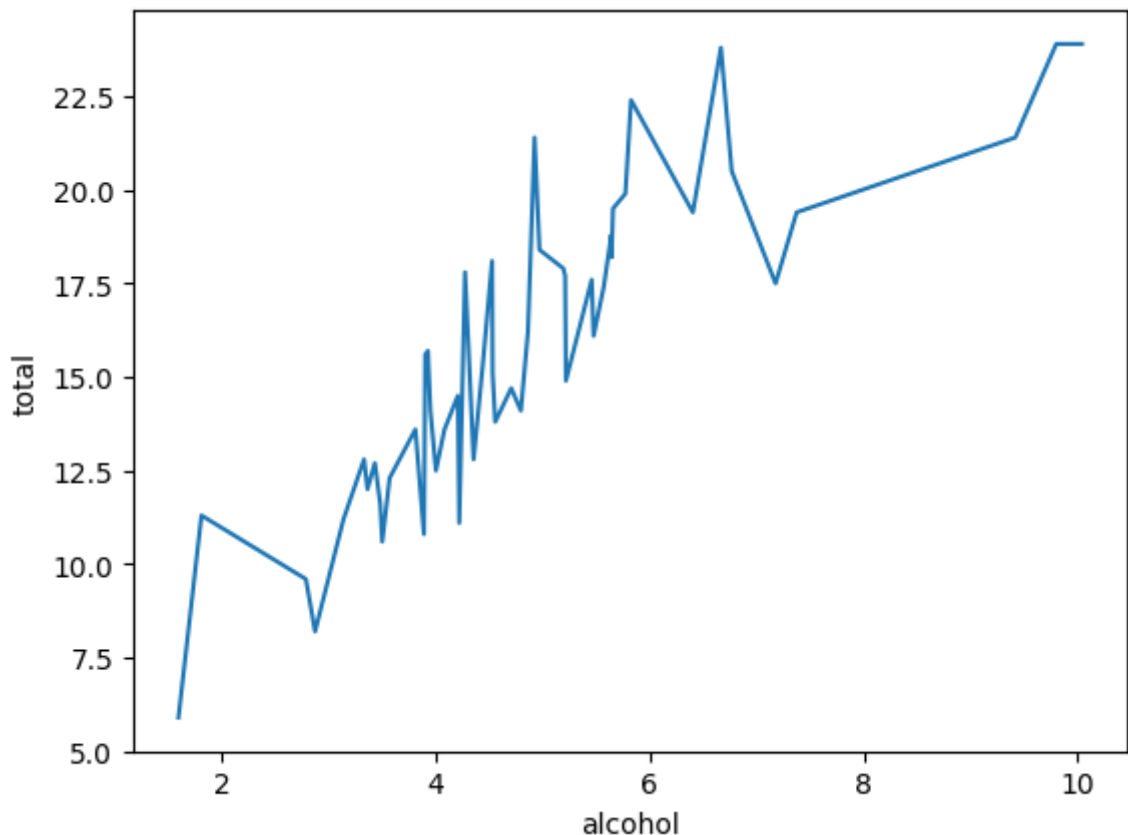
```
In [58]: sns.lineplot(x="alcohol",y="total",data=df,ci=None)
```

```
<ipython-input-58-10da8a41c1e4>:1: FutureWarning:
```

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

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

```
Out[58]: <Axes: xlabel='alcohol', ylabel='total'>
```



Inference: Overall, the total number of accidents increases with the increase in accidents caused by consumption of alcohol, but for some state the accidents caused because of that reason is less and for some state that is more. It is depicted by the zig-zag nature of the graph.

Distribution Plot

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

```
<ipython-input-18-8ecb7fd34a3c>:1: UserWarning:
```

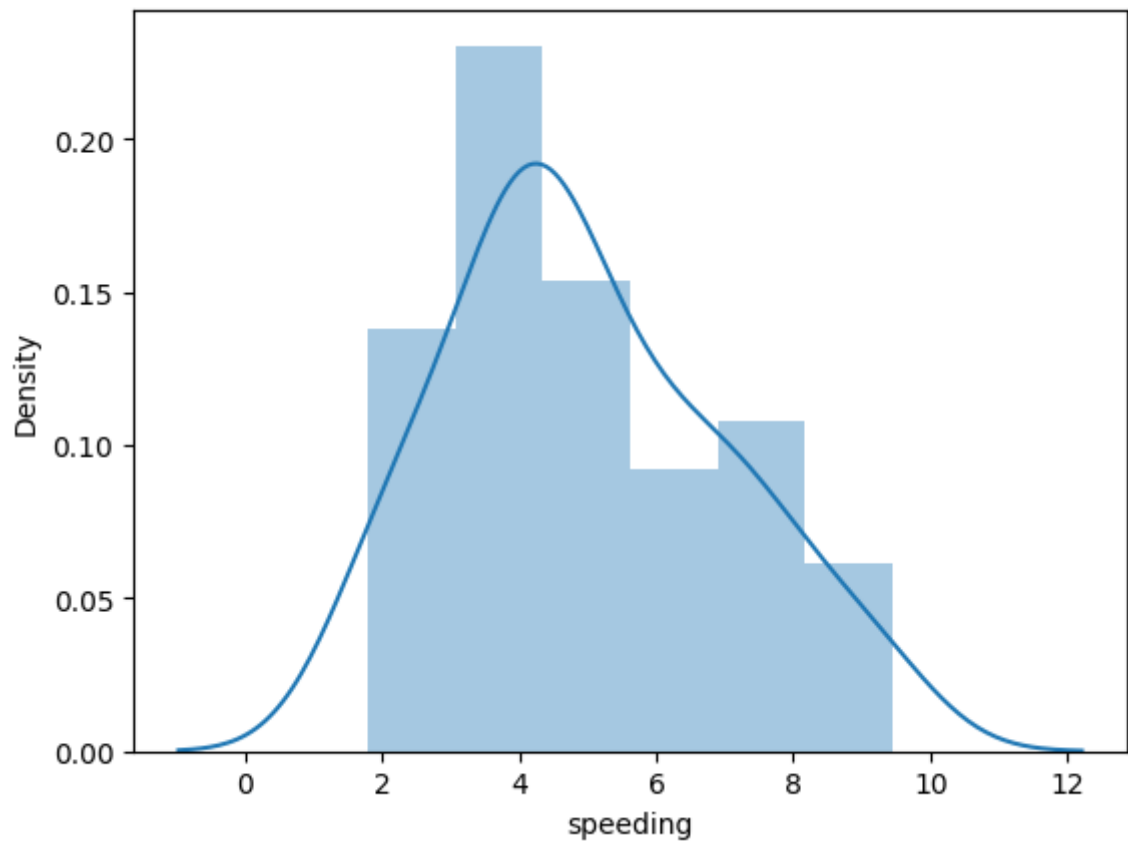
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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"])
```

```
Out[ ]: <Axes: xlabel='speeding', ylabel='Density'>
```

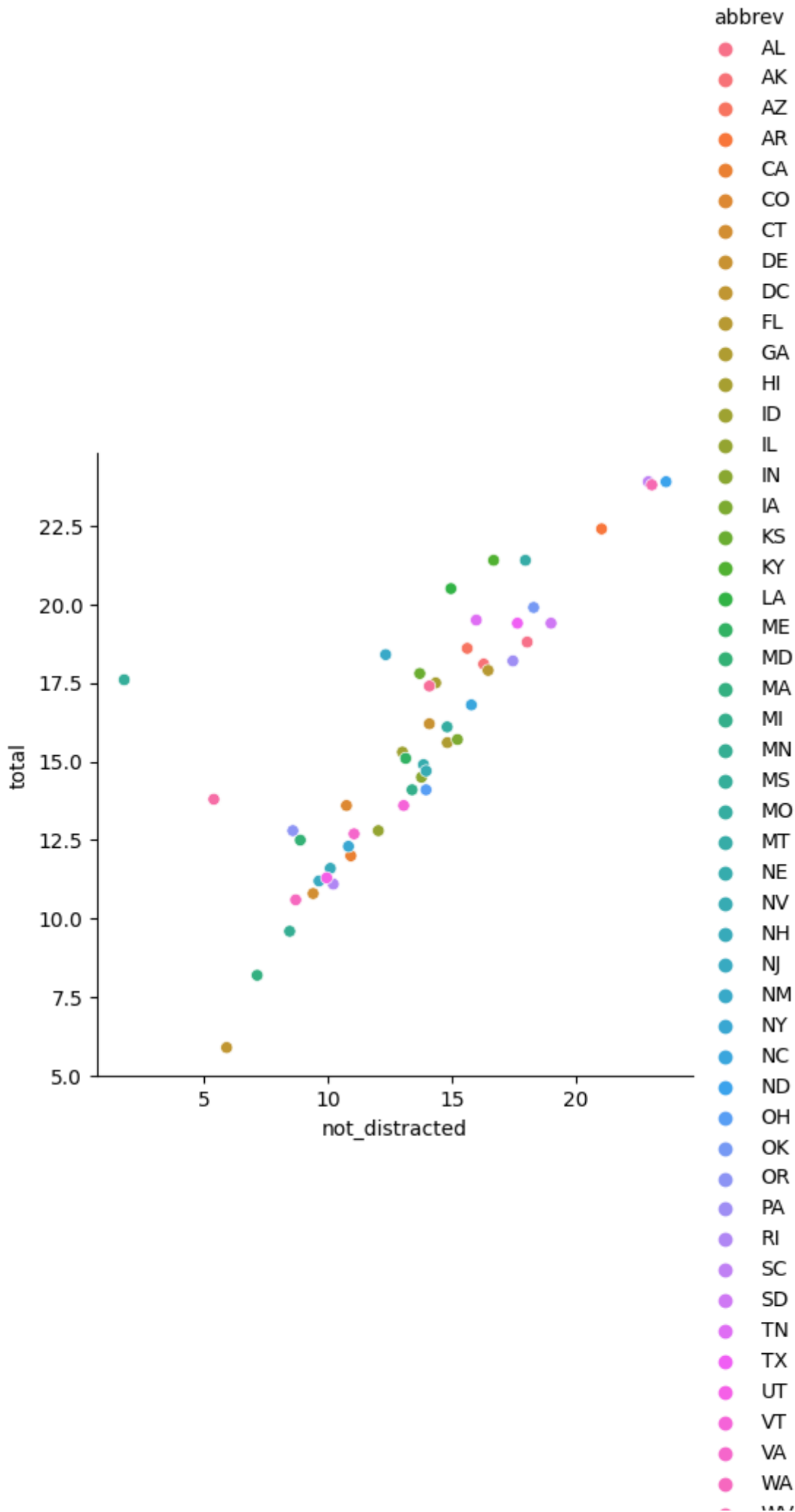


Inference: Percent of accidents caused due to speeding is ranging from 4 to 7.

Relation Plot

```
In [59]: sns.relplot(x="not_distracted",y="total",data=df,hue="abbrev")
```

```
Out[59]: <seaborn.axisgrid.FacetGrid at 0x7916967cbf10>
```



Inference: The above plot shows the relation between Total number of accidents occurred and accidents caused where the drivers were not distracted in the states of USA.

Bar Plot

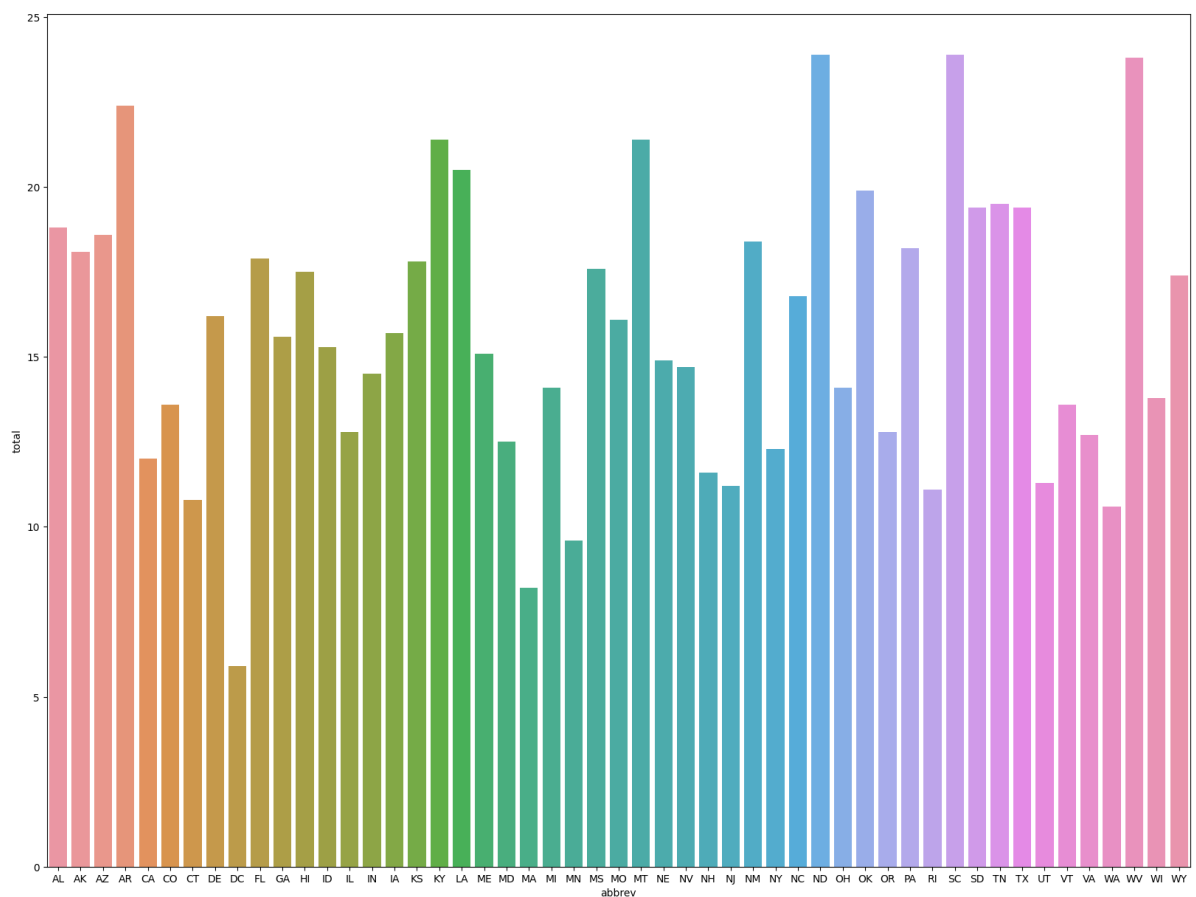
```
In [ ]: plt.subplots(figsize=(20,15))
sns.barplot(data=df,x="abbrev",y="total",ci=None)
```

<ipython-input-25-5f8b147885b5>:2: FutureWarning:

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

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

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

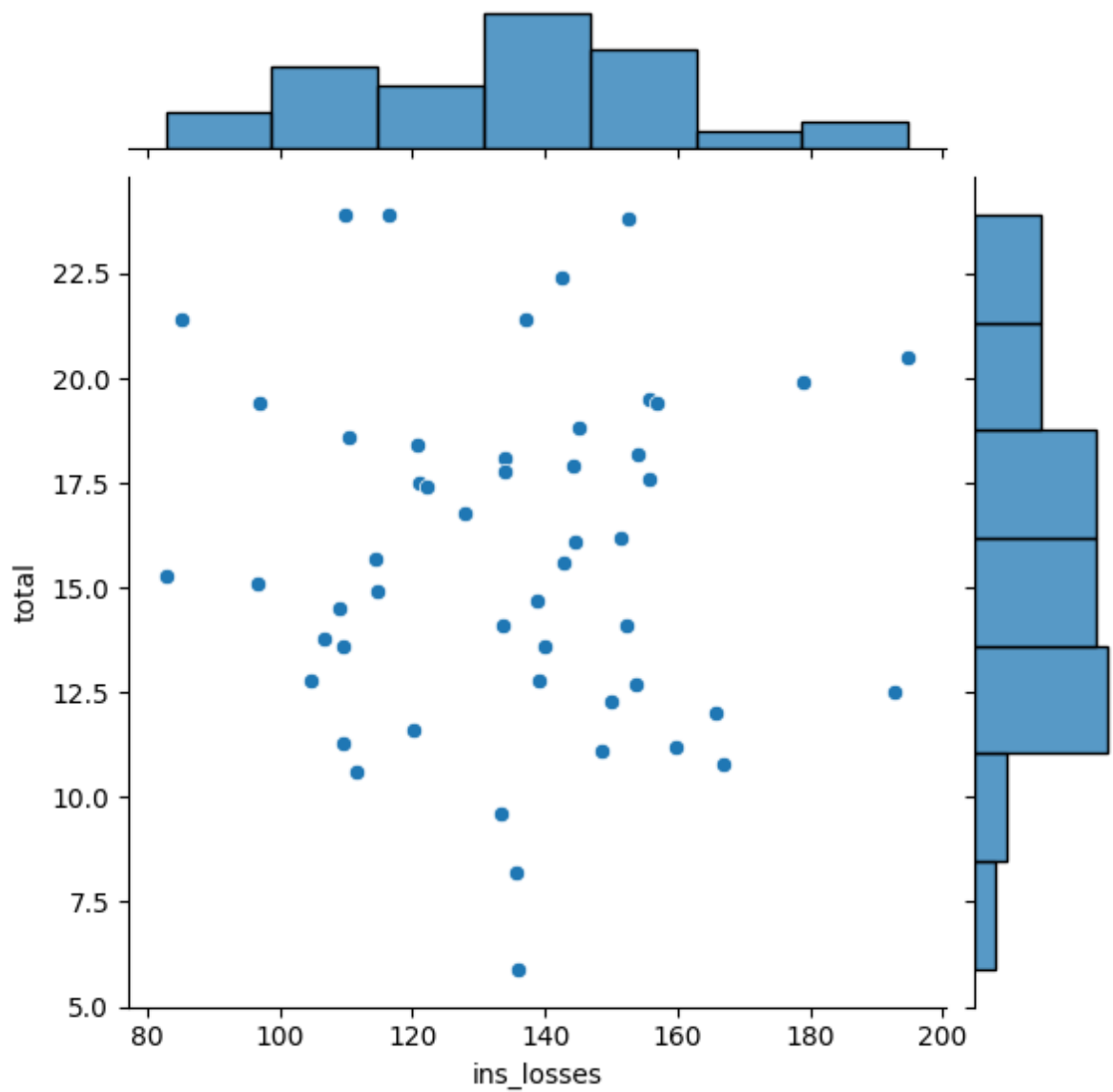


Inference: This plot shows all the states with its total number of accidents in form of bar graph.

Joint Plot

```
In [60]: sns.jointplot(x="ins_losses",y="total",data=df)
```

```
Out[60]: <seaborn.axisgrid.JointGrid at 0x7916973249a0>
```

Inference: The relation between total number of accidents taking place and insurance companies facing losses is depicted in the plot.

Correlation and Heatmap

```
In [ ]: cor=df.corr()
cor
```

<ipython-input-33-7a446f931109>: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.

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

Out[]:

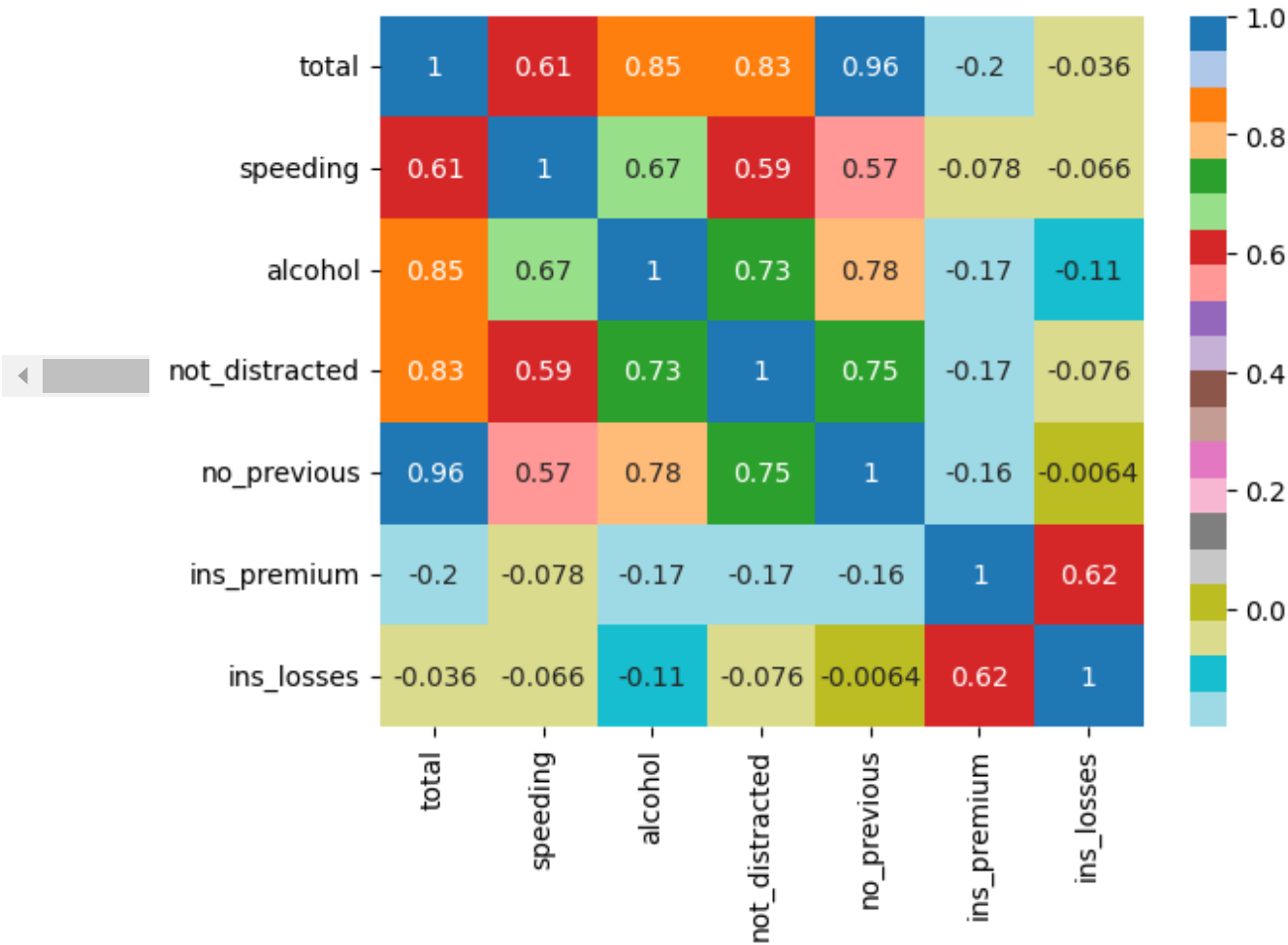
	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.199702	-0.036011
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.077675	-0.065928
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.170612	-0.112547
not_distracted	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

In []:

```
sns.heatmap(cor,annot=True,cmap="tab20_r")
```

Out[]:

<Axes: >



Inference: This shows dependencies between two variables, as shown in the charts.

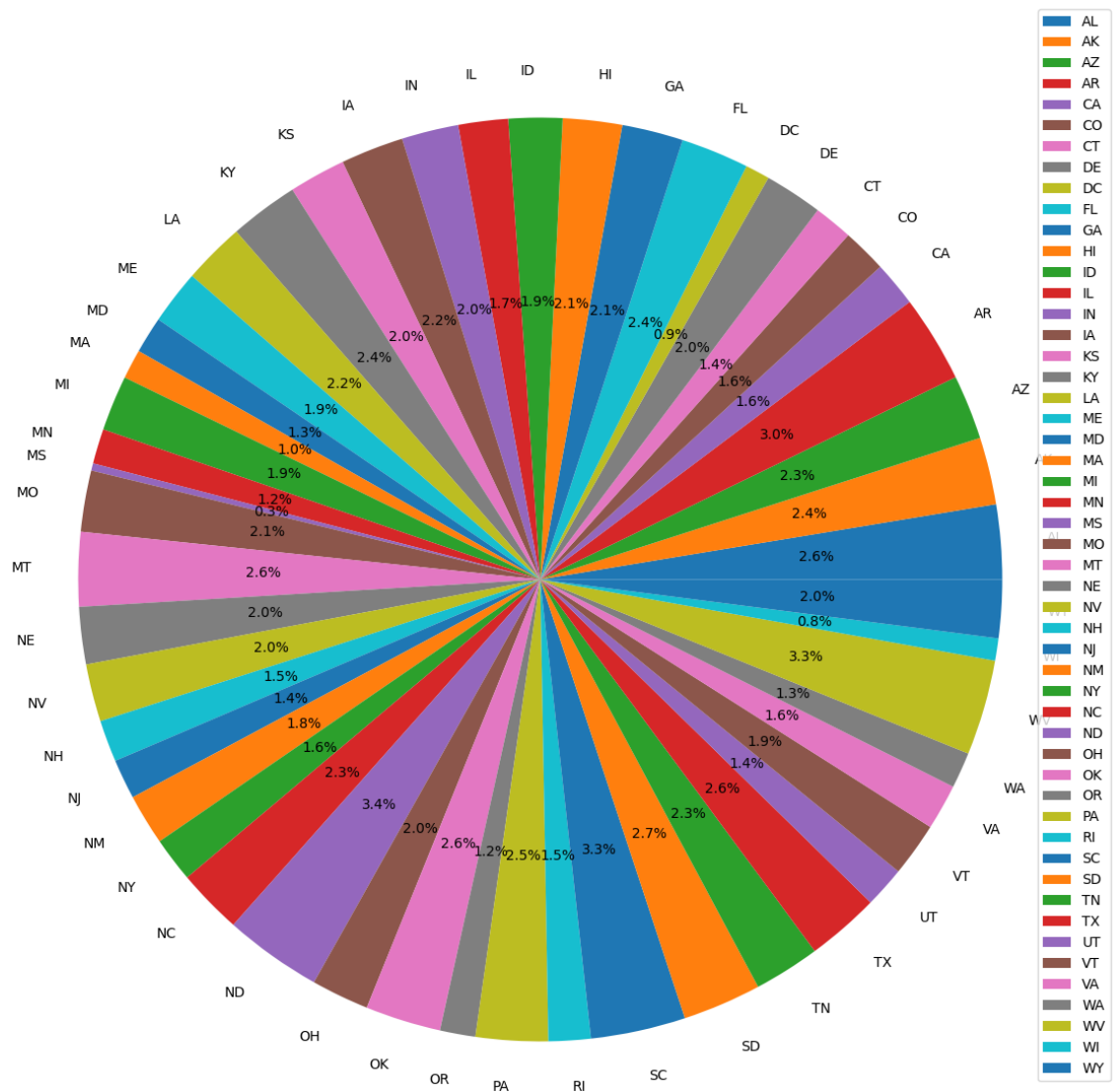
Pie Chart

In []:

```
plt.subplots(figsize=(20,15))
x=df.not_distracted
plt.pie(x,labels="abbrev",data=df,autopct="%0.1f%%")
plt.legend()
```

Out[]:

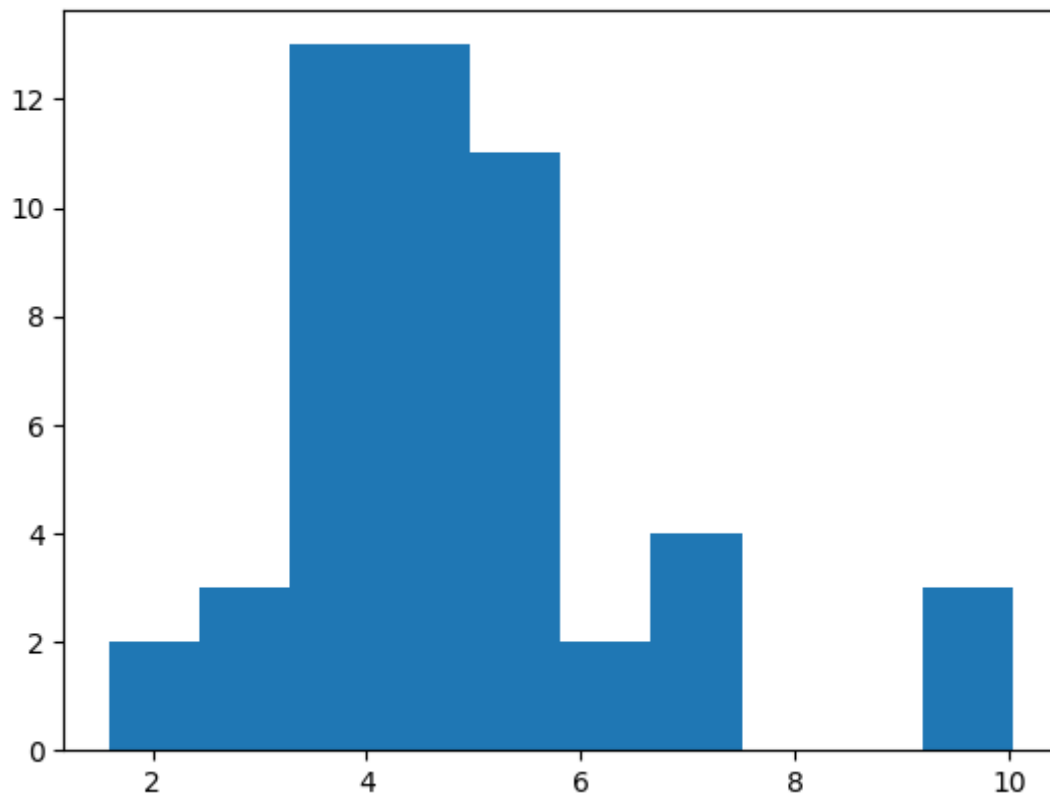
<matplotlib.legend.Legend at 0x791696eea8c0>



Inference: This plot shows all the states with its total percentage of accidents caused although the drivers were not distracted is depicted in form of pie graph.

Histogram

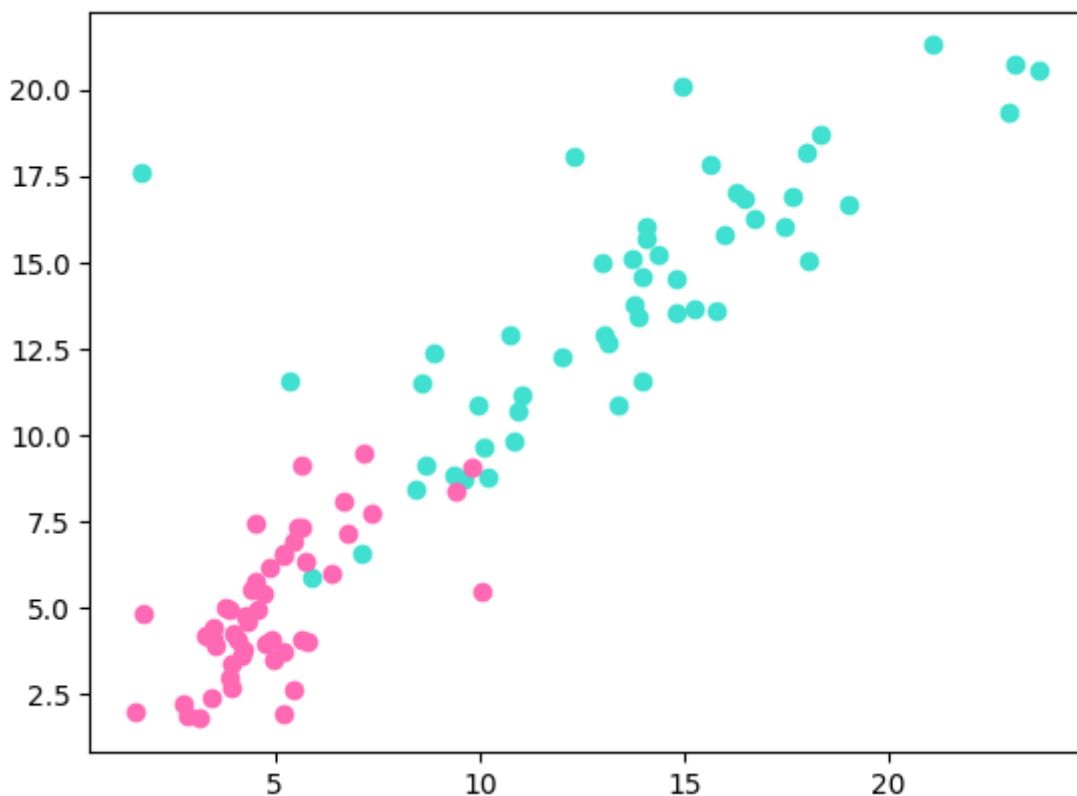
```
In [ ]: plt.hist(x="alcohol",data=df)
plt.show()
```



Inference: Shows graph of number of accidents caused due to alcohol consumption.

Comparing two Plots

```
In [ ]: plt.scatter(x="not_distracted",y="no_previous",data=df,color="turquoise")
plt.scatter(x="alcohol",y="speeding",data=df,color="hotpink")
plt.show()
```



Inference: comparing two scatterplots, one with accidents caused due to speeding and consumption of alcohol and two with accidents caused when driver was not distracted and drivers having no previous accident records.

Name- Rishima Chowdhury

Reg no-21BCE1097

Campus- VIT Chennai

E-mail- rishima.chowdhury2021@vitstudent.ac.in

Course- Machine learning and Artificial Intelligence

Slot-Morning