```
In [107]: import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

```
In [108]: sns.get_dataset_names()
```

```
Out[108]: ['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

```
In [109]: data=sns.load_dataset('car_crashes')
```

```
In [110]: data
```

Out[110]:

	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

```
In [111]: data.describe()
```

Out[111]:

	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

```
In [112]: data.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
```

Feature and Label Definition

· total -> Number of drivers involved in fatal collisions per billion miles (5.900–23.900)

· speeding -> Percentage Of Drivers Involved In Fatal Collisions Who Were Speeding (1.792–9.450)

· alcohol -> Percentage Of Drivers Involved In Fatal Collisions Who Were Alcohol-Impaired (1.593–10.038)

· not_distracted -> Percentage Of Drivers Involved In Fatal Collisions Who Were Not Distracted (1.760–23.661)

· no_previous -> Percentage Of Drivers Involved In Fatal Collisions Who Had Not Been Involved In Any Previous Accidents (5.900–21.280)

· ins_premium -> Car Insurance Premiums (641.960–1301.520)

· ins_losses -> Losses incurred by insurance companies for collisions per insured driver (82.75–194.780)

· abbrev -> USA states

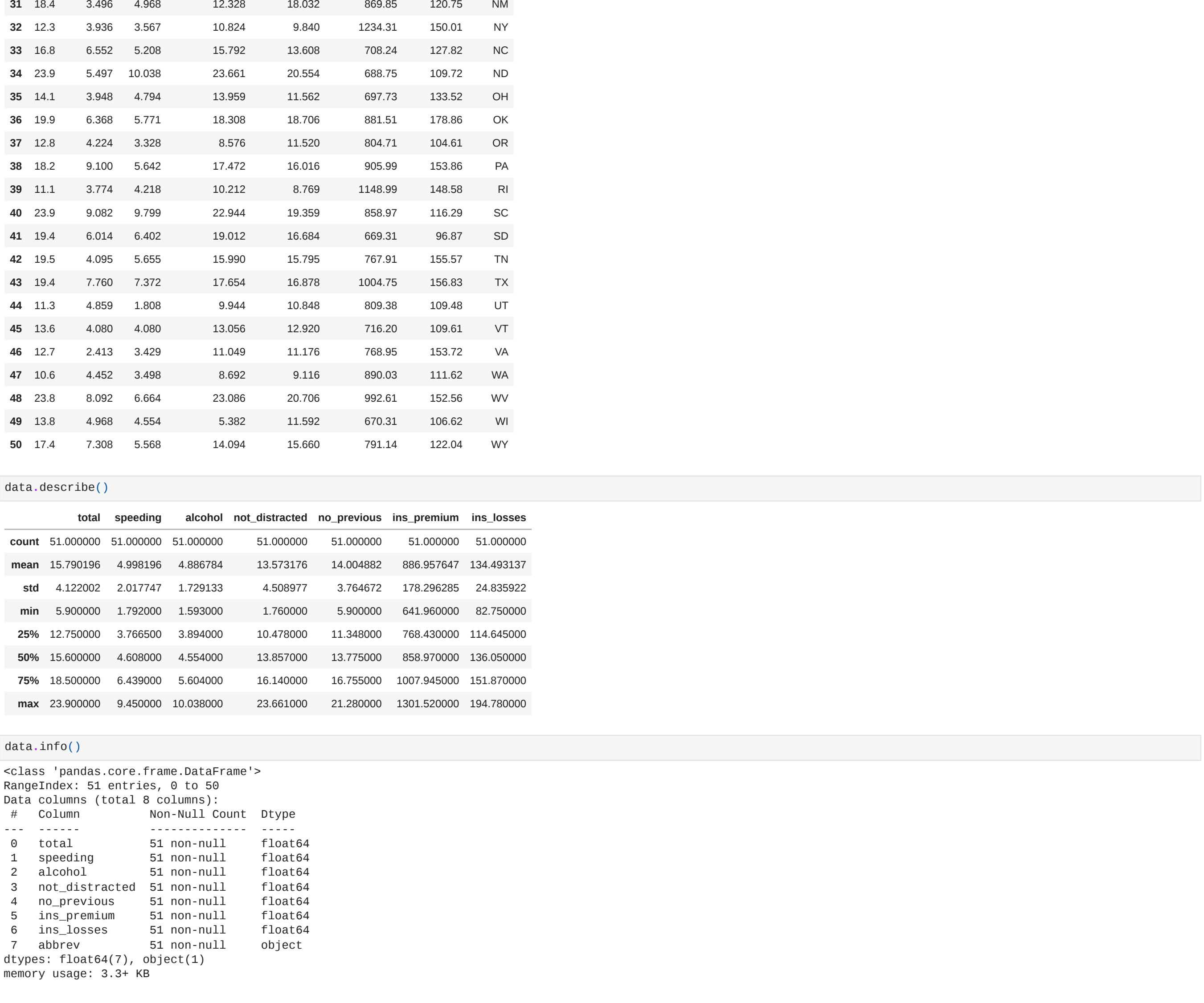
Scatter Plot

```
In [113]: sns.color_palette("rocket_r", as_cmap=True)
```

Out[113]: rocket_r

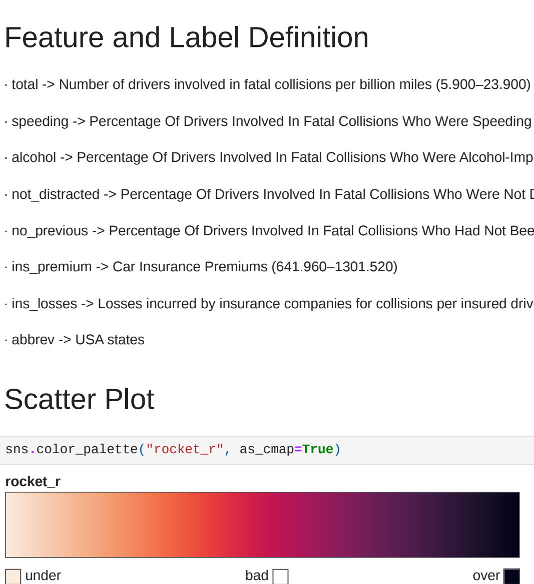

```
In [114]: sns.pairplot(data=data,hue="total",kind="scatter",palette="rocket_r")
```

Out[114]: <seaborn.axisgrid.PairGrid at 0x2aa98e320>




```
In [115]: sns.relplot(data=data,x="ins_premium",y="not_distracted")
```

Out[115]: <seaborn.axisgrid.FacetGrid at 0x2ae023a30>



Inference from Scatter Plot

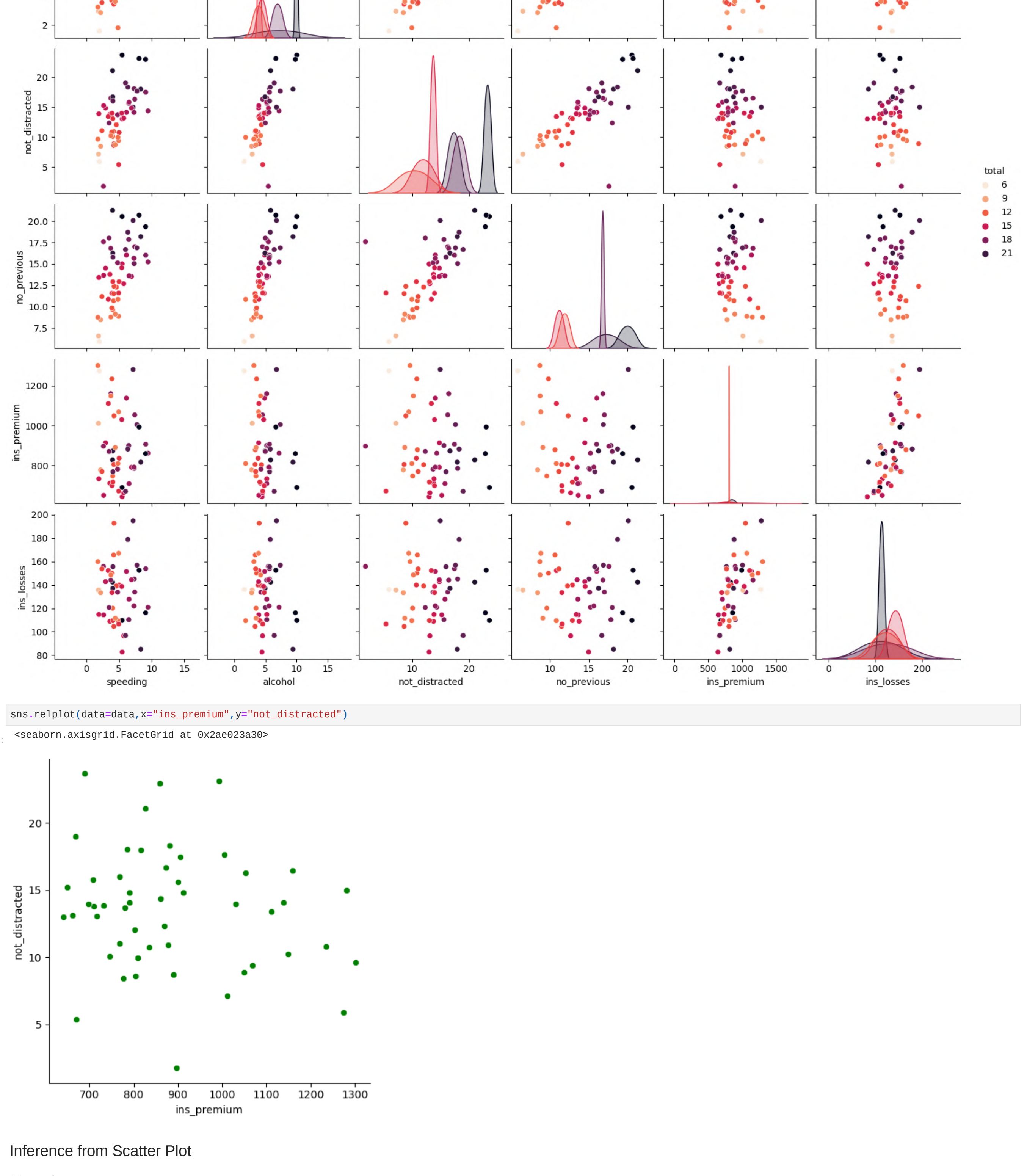
Observations are,

```
1.At higher speed and alcohol ,the total number of accidents is higher.
2.At higher speed and not_distracted ,the total number of accidents is higher.
3.At higher speed and no_previous ,the total number of accidents is higher.
4.At high alcohol and no_previous, the total number of accidents is increasing steeply.
5.Higher insurance premium means the driver will be more conscious as he has more money invested so the driver is less likely to be distracted.
6.Even if speeding is high and alcohol is high, the total accidents is less than if not_distracted is high.So the driver being distracted is the biggest reason for accidents.So we can focus on this feature more.
```

Categorical Plot

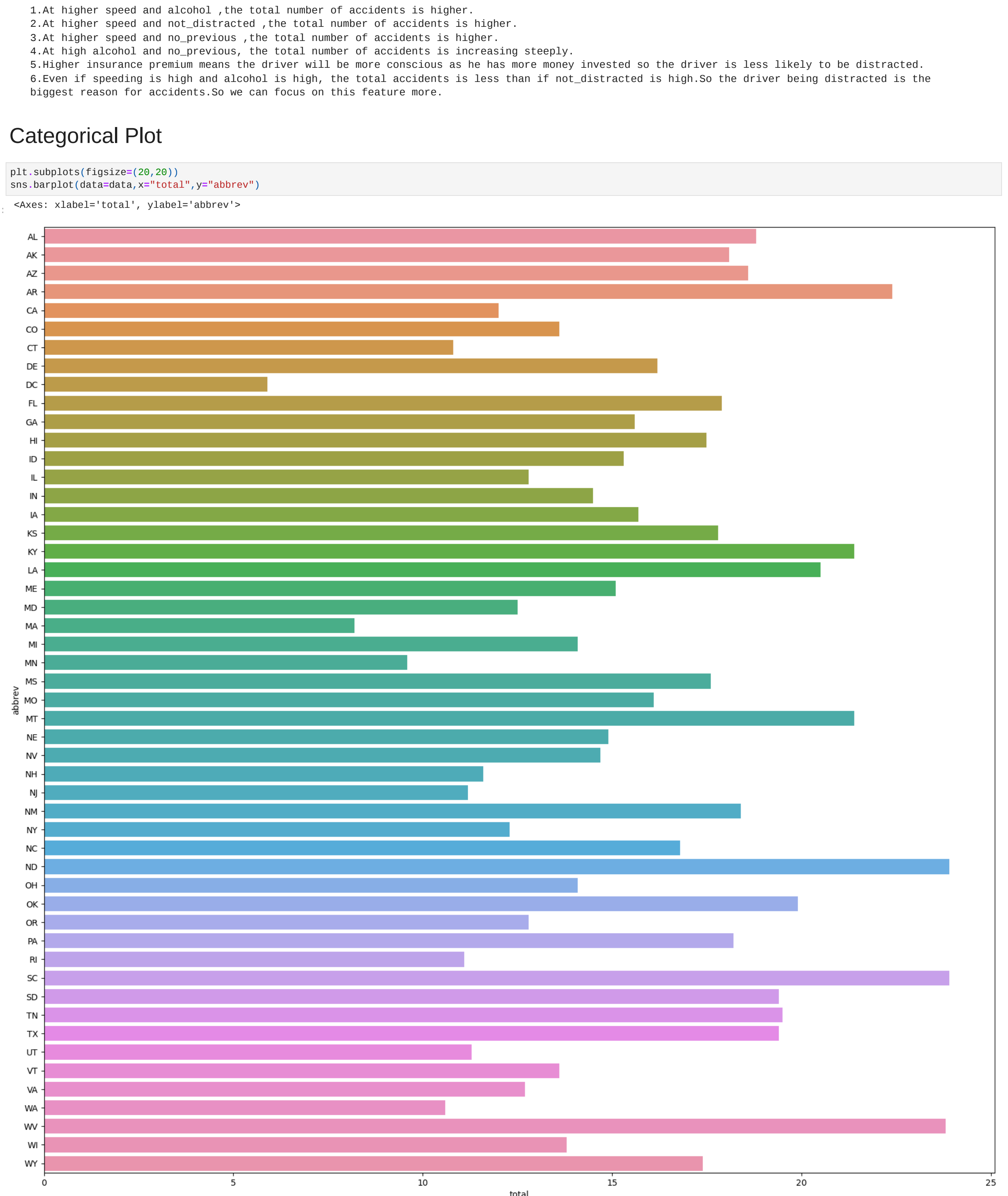
```
In [119]: plt.subplots(figsize=(20,20))
sns.barplot(data=data,x="total",y="abbrev")
```

Out[119]: <Axes: xlabel='total', ylabel='abbrev'>



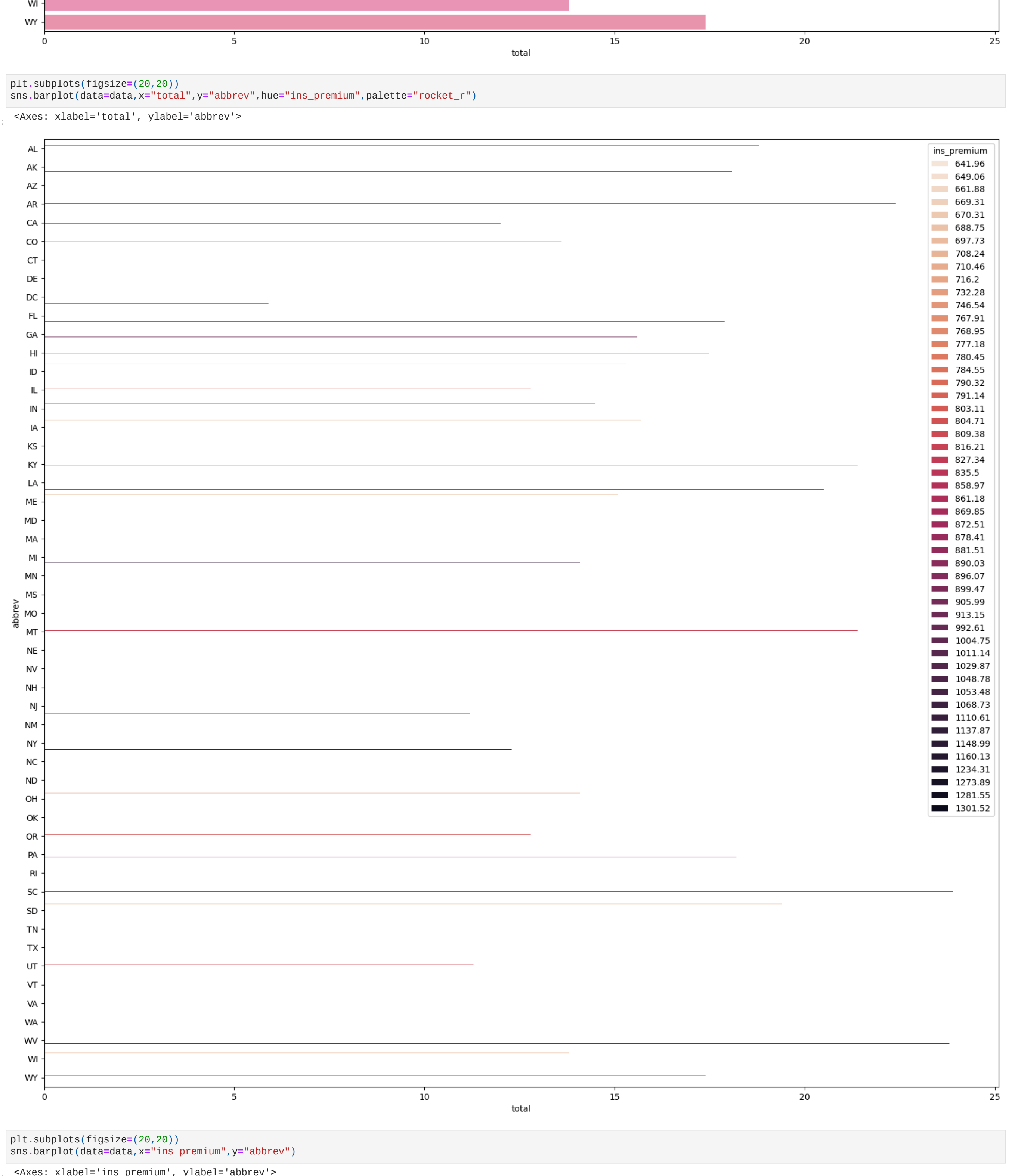

```
In [120]: plt.subplots(figsize=(20,20))
sns.barplot(data=data,x="total",y="abbrev",hue="ins_premium",palette="rocket_r")
```

Out[120]: <Axes: xlabel='total', ylabel='abbrev'>



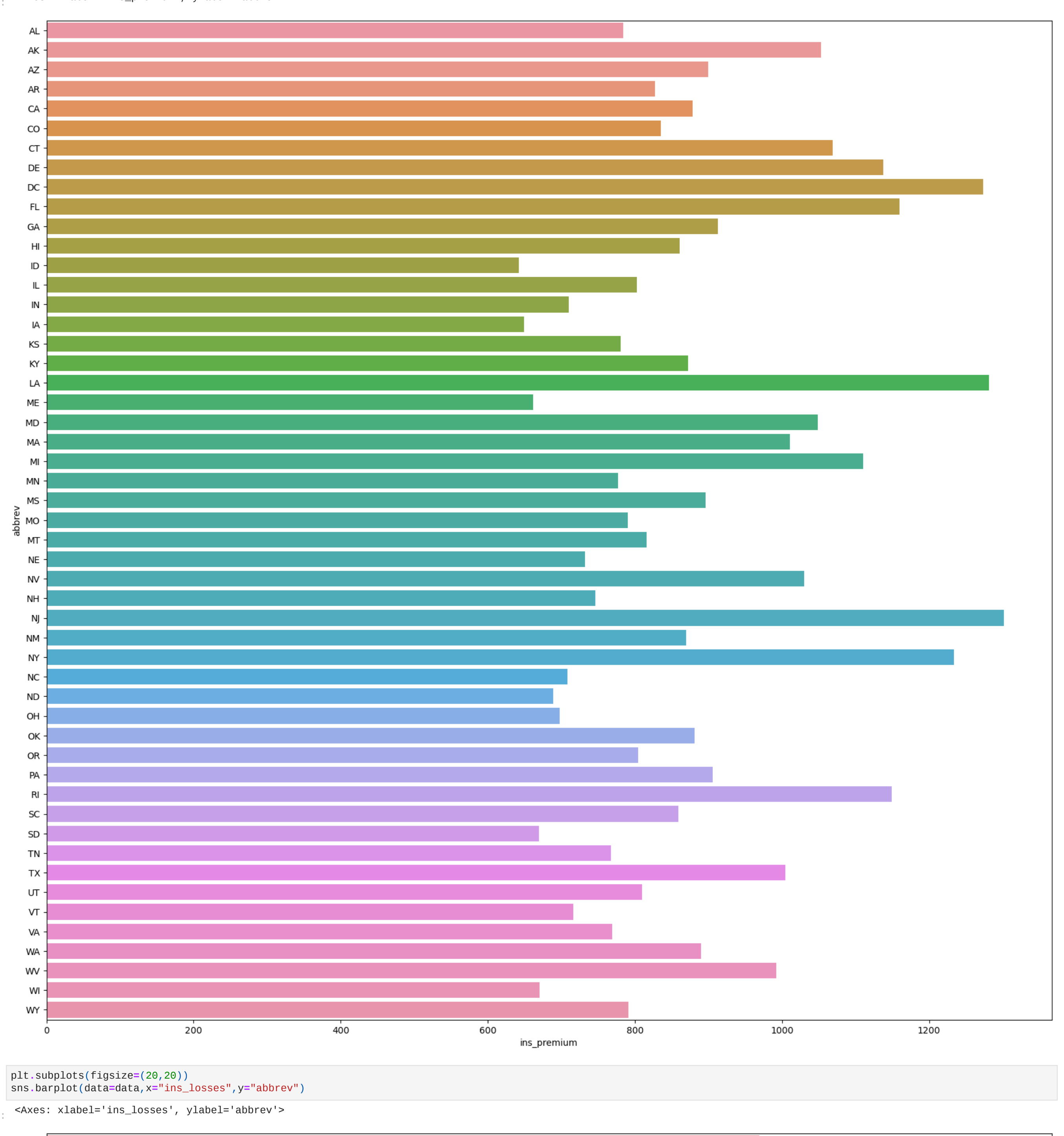

```
In [121]: plt.subplots(figsize=(20,20))
sns.barplot(data=data,x="ins_premium",y="abbrev")
```

Out[121]: <Axes: xlabel='ins_premium', ylabel='abbrev'>




```
In [122]: plt.subplots(figsize=(20,20))
sns.barplot(data=data,x="ins_losses",y="abbrev")
```

Out[122]: <Axes: xlabel='ins_losses', ylabel='abbrev'>



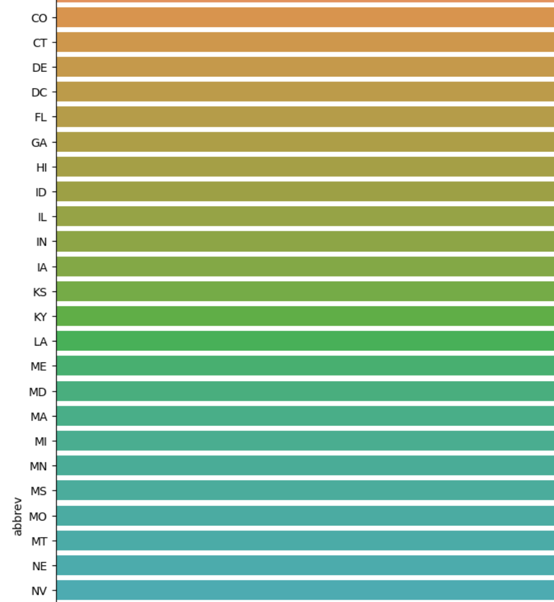
Inference from Categorical Plot

```
No conclusive data to draw observation from and relate the insurance premium or insurance losses to specific states with high or low accidents.
```

Line Plots

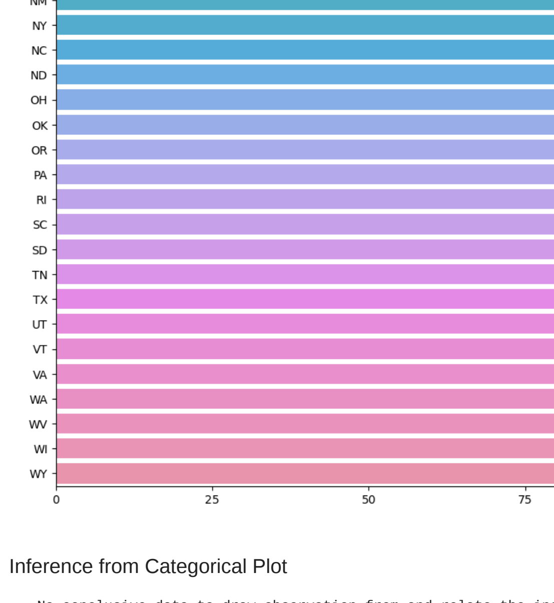
```
In [116]: sns.lineplot(data=data,x="not_distracted",y="total")
```

Out[116]: <Axes: xlabel='not_distracted', ylabel='total'>



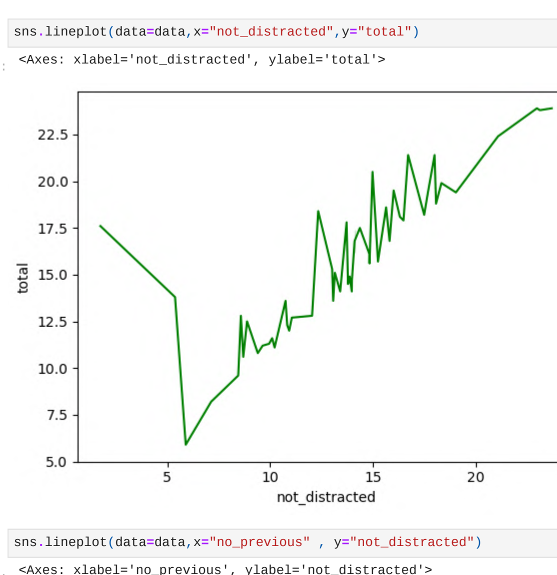

```
In [117]: sns.lineplot(data=data,x="no_previous" , y="not_distracted")
```

Out[117]: <Axes: xlabel='no_previous', ylabel='not_distracted'>




```
In [118]: sns.lineplot(data=data,x="alcohol",y="not_distracted")
```

Out[118]: <Axes: xlabel='alcohol', ylabel='not_distracted'>



Inference from Line Plot

```
1.Even if driver is not distracted to certain percentage, accidents still happen,but distracted drivers cause more accidents.
2.Drivers that have consumed alcohol before driving are more likely to be distracted.
3.Drivers that have no previous accidents are more likely to be distracted.
```

```
In [ ]:
```