Importing the required modules and dataset

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
dataset = sns.load_dataset("car_crashes")

dataset
```

	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

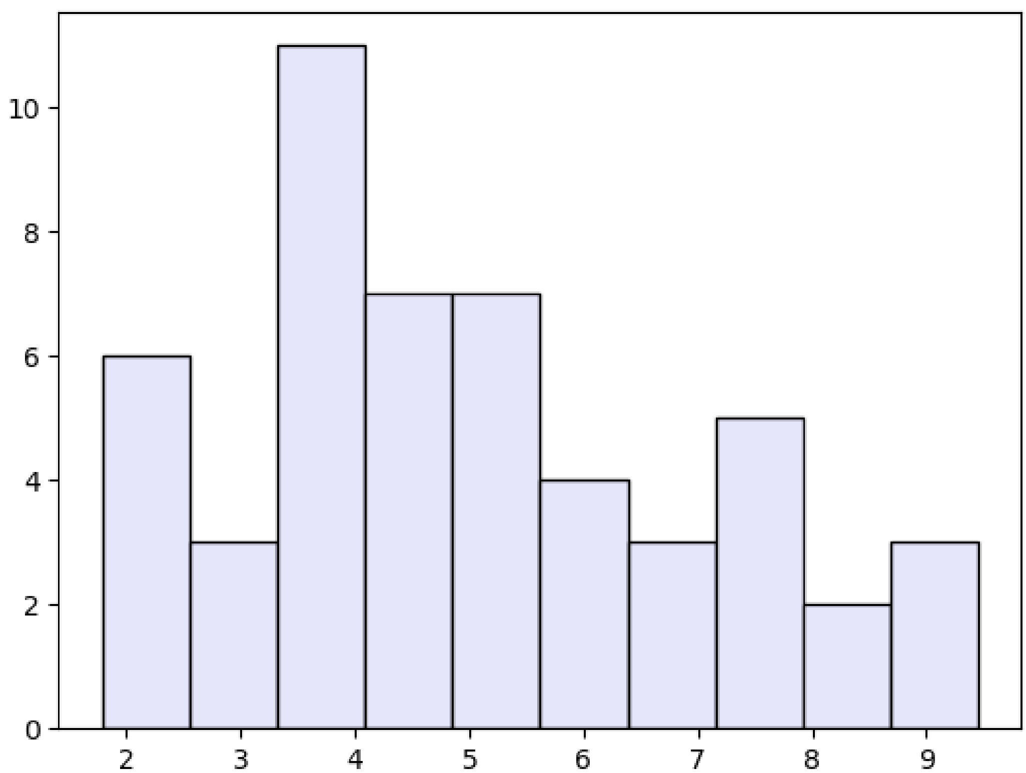
	ins_losses	abbrev
0	145.08	AL
1	133.93	AK
2	110.35	AZ
3	142.39	AR
4	165.63	CA
5	139.91	CO
6	167.02	CT
7	151.48	DE
8	136.05	DC
9	144.18	FL
10	142.80	GA
11	120.92	HI
12	82.75	ID
13	139.15	IL
14	108.92	IN
15	114.47	IA
16	133.80	KS
17	137.13	KY
18	194.78	LA
19	96.57	ME
20	192.70	MD
21	135.63	MA
22	152.26	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
```

Univariate Data

Histogram

```
x = dataset["speeding"]
plt.hist(x,color = 'lavender', edgecolor = 'black')
plt.show()
```



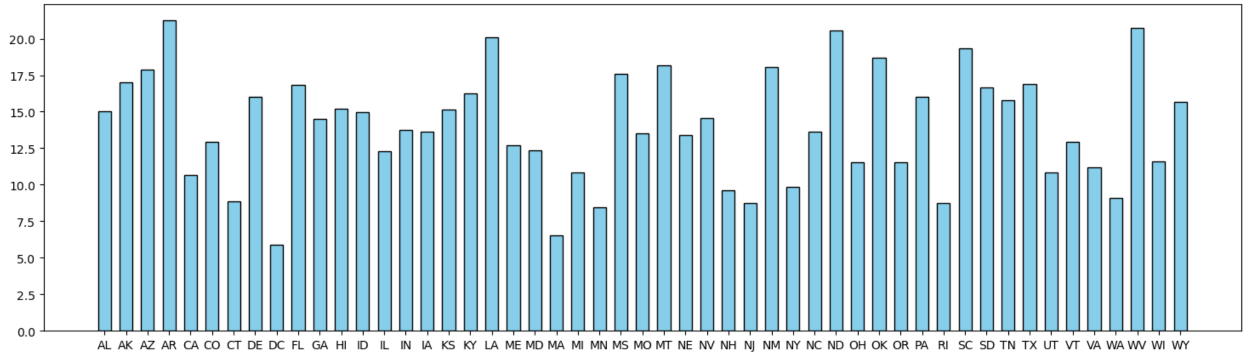
This graph indicates that most of the people drive in the speed values b/w 3 to 4

Barcharts

```python
fig = plt.figure()
ax1 = fig.add_axes([0.1,0.1,2.2,0.8])

x = dataset["abbrev"]
y = dataset["no_previous"]
ax1.bar(x,y,color = 'skyblue',width = 0.6, edgecolor = 'black')
```

```
<BarContainer object of 51 artists>
```



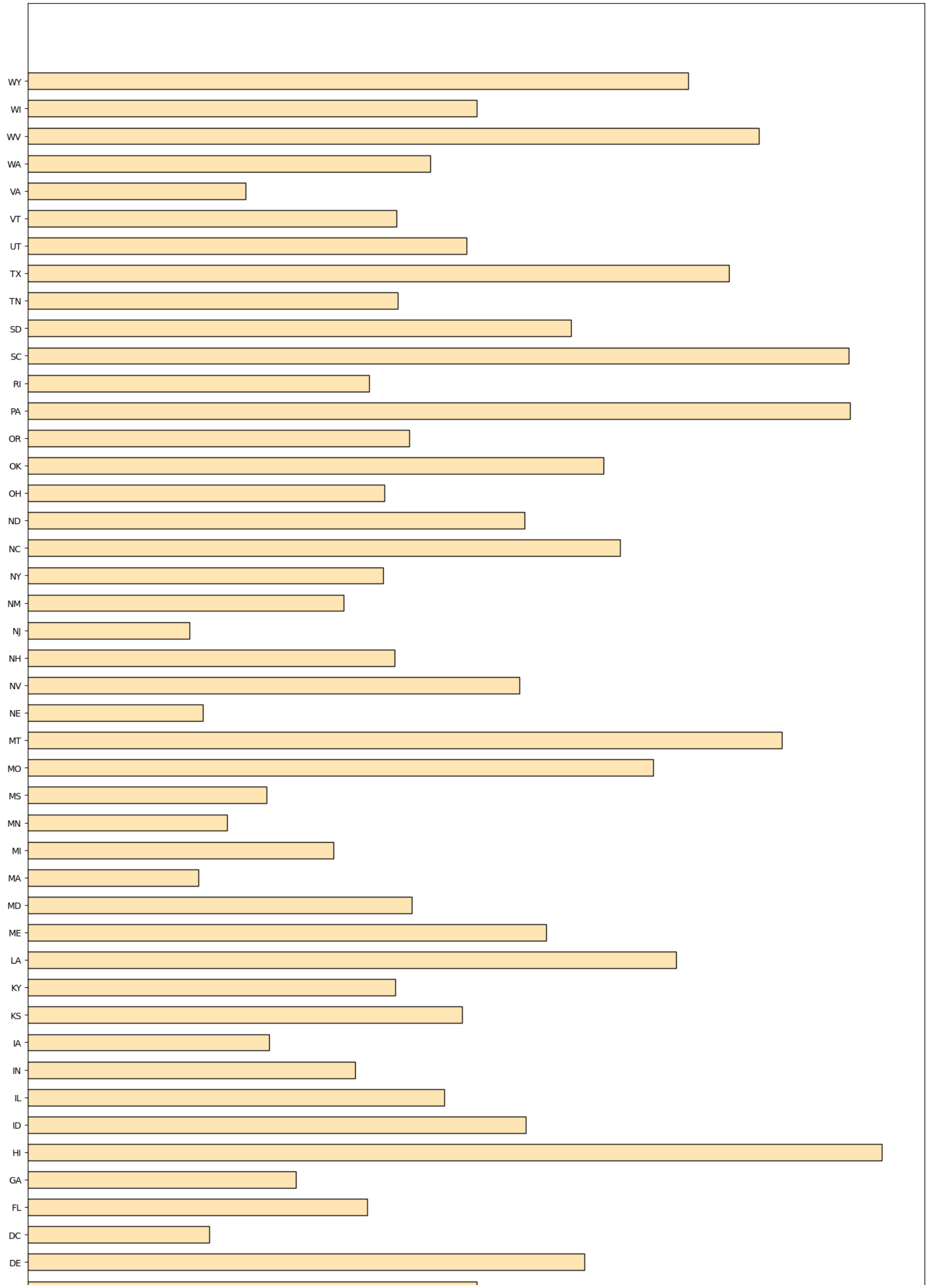
This graph indicates that AR has the highest number of people who did not have a previous accident

Barcharts Horizontal

```python
fig = plt.figure()
ax1 = fig.add_axes([0.1,0.1,2.2,5])

x = dataset["abbrev"]
y = dataset["speeding"]
ax1.barh(x,y,color = "#FFE5B4",height = 0.6, edgecolor = "black")
```

```
<BarContainer object of 51 artists>
```


WY
WI
WV
WA
VA
VT
UT
TX
TN
SD
SC
RI
PA
OR
OK
OH
ND
NC
NY
NM
NJ
NH
NV
NE
MT
MO
MS
MN
MI
MA
MD
ME
LA
KY
KS
IA
IN
IL
ID
HI
GA
FL
DC
DE

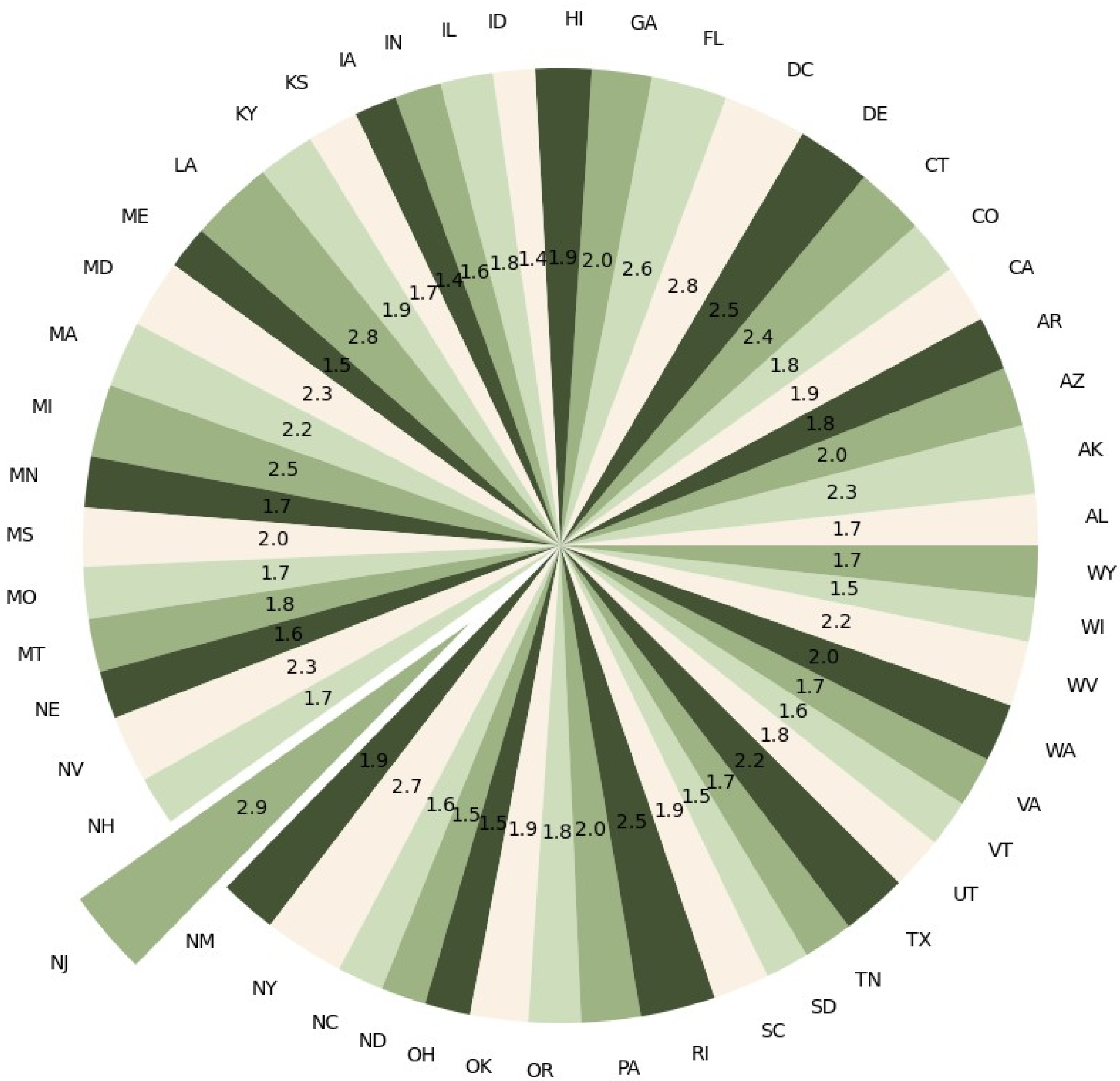
This graph indicates that people from HI drive the fastest

Piecharts

```
fig = plt.figure()
axes1 = fig.add_axes([0.1,0.1,1.75,1.75])
labels = dataset["abbrev"]
numbers = dataset["ins_premium"]

explode = np.zeros(numbers.count())
explode[numbers.argmax()] = 0.25

axes1.pie(numbers, labels = labels,explode = explode, autopct =
"%0.1f", colors = ["#FAF1E4", "#CEDEBD", "#9EB384", "#435334", ])
plt.show()
```



This graph indicates that NJ has pays the highest premium

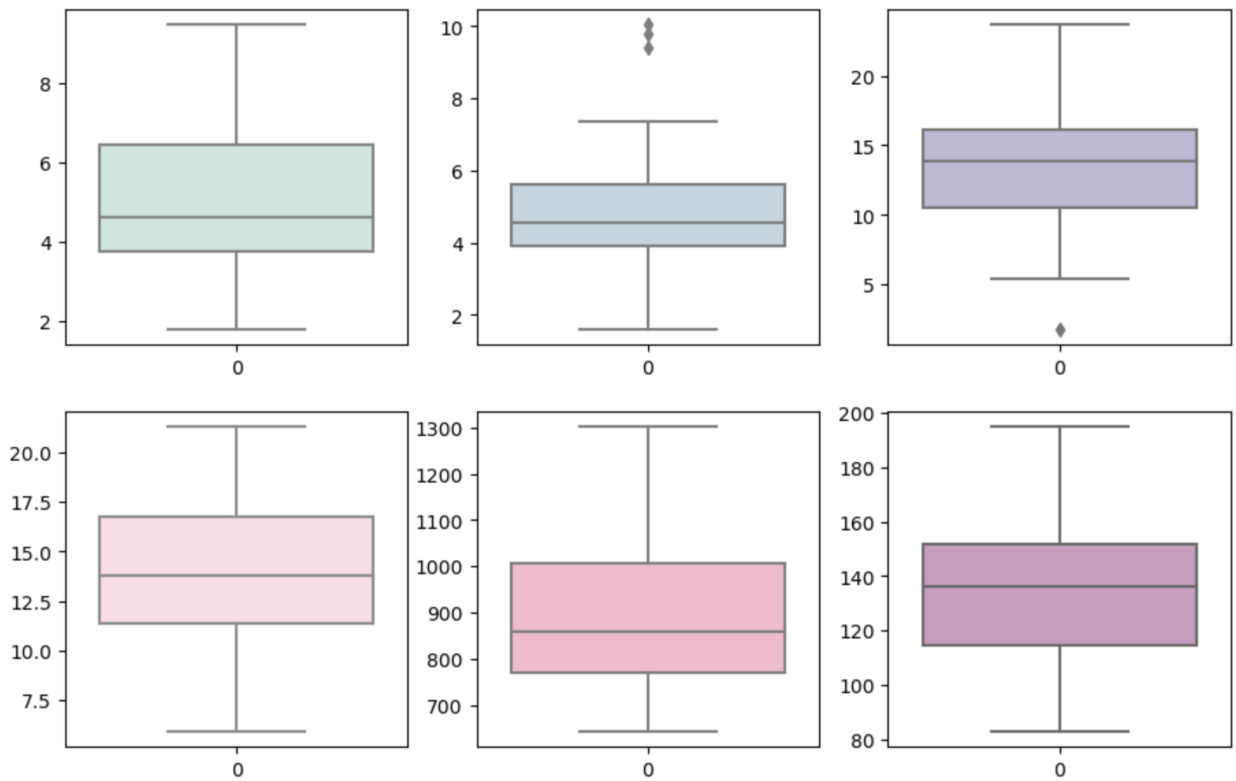
Boxplots

```
plt.subplots(figsize = [11,7])
plt.subplot(2,3,1)
sns.boxplot(dataset["speeding"], color = "#CCE8DB")
plt.subplot(2,3,2)
sns.boxplot(dataset["alcohol"], color = "#C1D4E3")
plt.subplot(2,3,3)
sns.boxplot(dataset["not_distracted"], color = "#BBB4D6")
plt.subplot(2,3,4)
sns.boxplot(dataset["no_previous"], color = "#FADAE2")
```

```
plt.subplot(2,3,5)
sns.boxplot(dataset["ins_premium"], color = "#F8B3CA")
plt.subplot(2,3,6)
sns.boxplot(dataset["ins_losses"], color = "#CC97C1")
```

```
C:\Users\raman\AppData\Local\Temp\ipykernel_9616\3332801330.py:2: MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two minor releases later; explicitly call ax.remove() as needed.
  plt.subplot(2,3,1)

<Axes: >
```



We can incur from this graph that Alcohol has the highest number of outliers

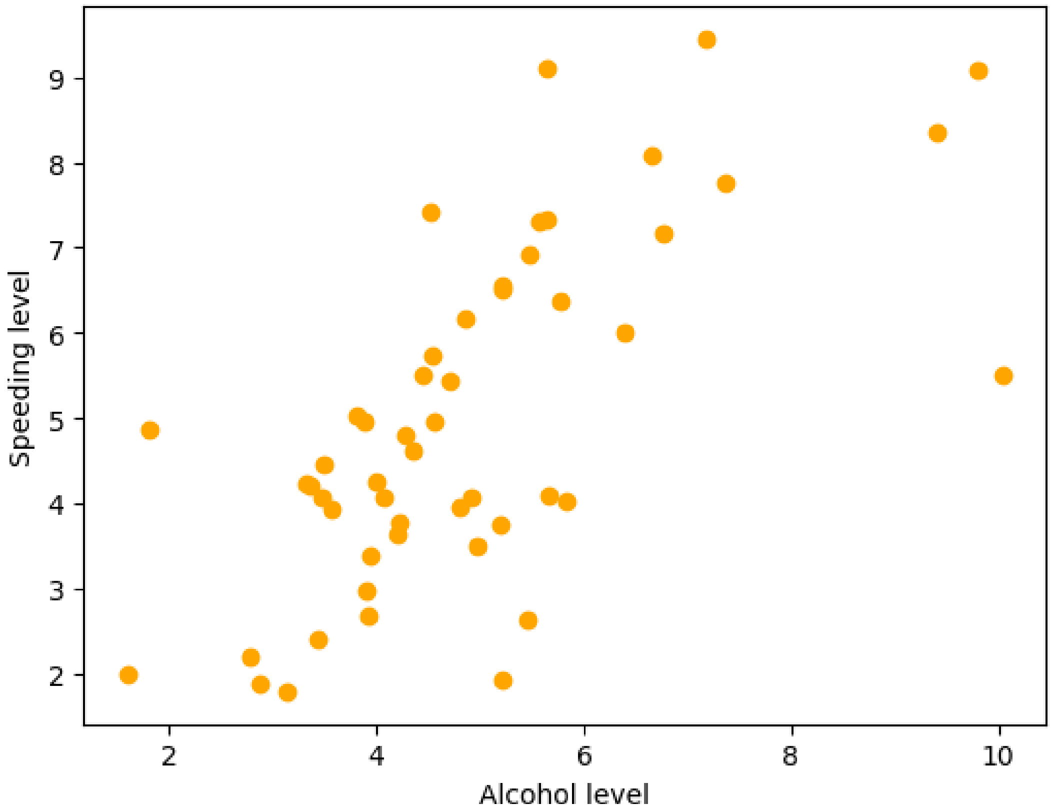
Bivariate Data

Scatterplot

```
x = dataset["alcohol"]
y = dataset["speeding"]
plt.scatter(x, y, color = 'orange')

plt.xlabel('Alcohol level')
plt.ylabel('Speeding level')
```

```
Text(0, 0.5, 'Speeding level')
```



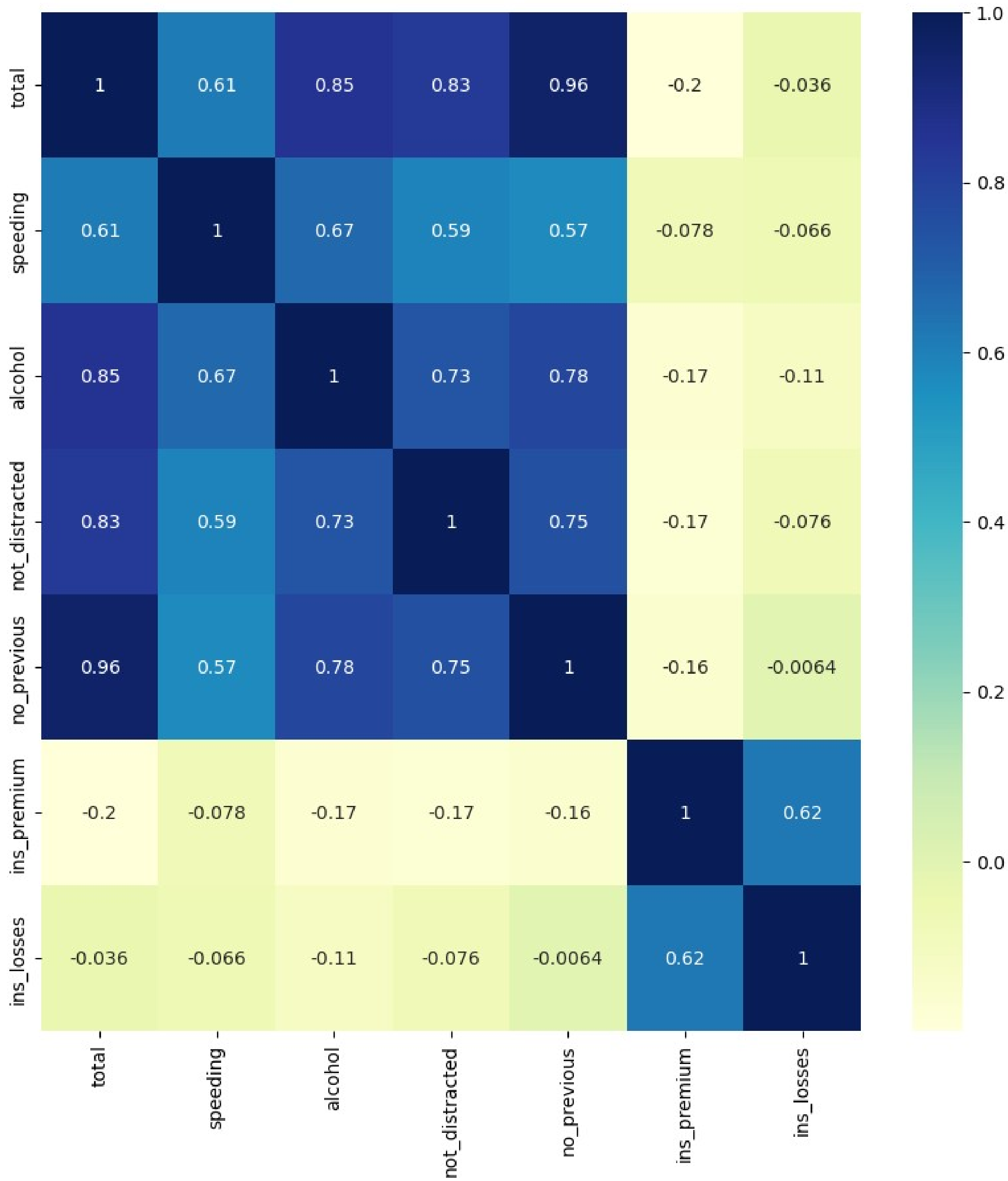
From this graph, we can understand that higher the alcohol level of a person, there is a higher chance of speeding level as well

Correlation Heatmap

```
corr = dataset.corr()
plt.subplots(figsize = [10,10])
sns.heatmap(corr, annot = True, cmap= "YlGnBu")
```

```
C:\Users\raman\AppData\Local\Temp\ipykernel_9616\1155716378.py: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.
  corr = dataset.corr()
```

```
<Axes: >
```



From this data we can understand that there is a very high correlation between multiple variables. Insurance premium and insurance loss are having a good correlation whereas these both have very weak correlation with the rest of the categories