P J N D M PRAKASH

21BCE8942

durga.21bce8942@vitapstudent.ac.in

AI @ ML Morning slot

DATA - VISUALIZATION ASSIGNMENT

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data=sns.load_dataset("car_crashes")
data
```

	total	speeding	alcohol	not_distracted	no_previous	ir
0	18.8	7.332	5.640	18.048	15.040	
1	18.1	7.421	4.525	16.290	17.014	
2	18.6	6.510	5.208	15.624	17.856	
3	22.4	4.032	5.824	21.056	21.280	
4	12.0	4.200	3.360	10.920	10.680	
5	13.6	5.032	3.808	10.744	12.920	
6	10.8	4.968	3.888	9.396	8.856	
7	16.2	6.156	4.860	14.094	16.038	
8	5.9	2.006	1.593	5.900	5.900	
9	17.9	3.759	5.191	16.468	16.826	
10	15.6	2.964	3.900	14.820	14.508	
11	17.5	9.450	7.175	14.350	15.225	
12	15.3	5.508	4.437	13.005	14.994	
13	12.8	4.608	4.352	12.032	12.288	
14	14.5	3.625	4.205	13.775	13.775	
15	15.7	2.669	3.925	15.229	13.659	
16	17.8	4.806	4.272	13.706	15.130	
17	21.4	4.066	4.922	16.692	16.264	
18	20.5	7.175	6.765	14.965	20.090	
19	15.1	5.738	4.530	13.137	12.684	
20	12.5	4.250	4.000	8.875	12.375	
21	8.2	1.886	2.870	7.134	6.560	
22	14.1	3.384	3.948	13.395	10.857	
23	9.6	2.208	2.784	8.448	8.448	
24	17.6	2.640	5.456	1.760	17.600	
25	16.1	6.923	5.474	14.812	13.524	

Above data describes about the total car_crashes happened in different state , causes of the incident and details of insurance

```
# Check total info about columns in the data
data.columns
```

```
Index(['total', 'speeding', 'alcohol', 'not_distracted', 'no_previous',
       'ins_premium', 'ins_losses', 'abbrev'],
      dtype='object')
```

```
# Check is any null values present in the dataset
data.isna().sum()
```

```
total             0
speeding          0
alcohol           0
not_distracted    0
```

```
no_previous       0
ins_premium       0
ins_losses        0
abbrev            0
dtype: int64
```

There are no null values in the dataset

```
# Check the type of data in the dataset
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
```

Let us visualize the data using different tools in matplotlib and seaborn

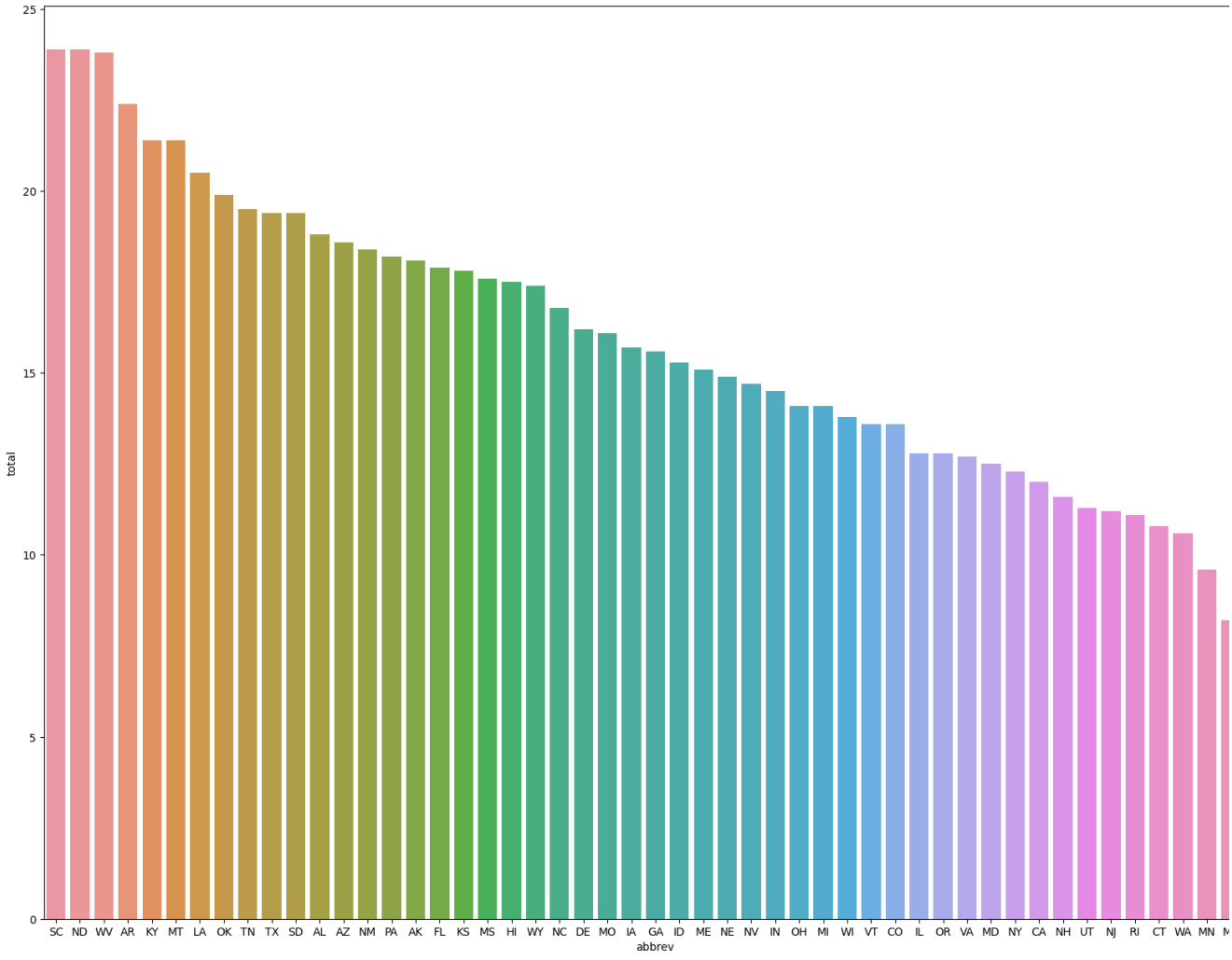
```
# Since there are 50 different states in the dataset.
# We will sort the top 10 states where more no of drivers involved in the crash from the state

top_total =data.sort_values(by="total",ascending=False)
top_total
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
40	23.9	9.082	9.799	22.944	19.359	858.97	116.29	SC
34	23.9	5.497	10.038	23.661	20.554	688.75	109.72	ND
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56	WV
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	AR
17	21.4	4.066	4.922	16.692	16.264	872.51	137.13	KY
26	21.4	8.346	9.416	17.976	18.190	816.21	85.15	MT
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
36	19.9	6.368	5.771	18.308	18.706	881.51	178.86	OK
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
41	19.4	6.014	6.402	19.012	16.684	669.31	96.87	SD
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	AL
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	AZ
31	18.4	3.496	4.968	12.328	18.032	869.85	120.75	NM
38	18.2	9.100	5.642	17.472	16.016	905.99	153.86	PA
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	AK
9	17.9	3.759	5.191	16.468	16.826	1160.13	144.18	FL
16	17.8	4.806	4.272	13.706	15.130	780.45	133.80	KS
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77	MS
11	17.5	9.450	7.175	14.350	15.225	861.18	120.92	HI
50	17.4	7.308	5.568	14.094	15.660	791.14	122.04	WY
33	16.8	6.552	5.208	15.792	13.608	708.24	127.82	NC
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	DE
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45	MO
15	15.7	2.669	3.925	15.229	13.659	649.06	114.47	IA
10	15.6	2.964	3.900	14.820	14.508	913.15	142.80	GA
12	15.3	5.508	4.437	13.005	14.994	641.96	82.75	ID
19	15.1	5.738	4.530	13.137	12.684	661.88	96.57	ME
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
14	14.5	3.625	4.205	13.775	13.775	710.46	108.92	IN
35	14.1	3.948	4.794	13.959	11.562	697.73	133.52	OH
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26	MI
49	13.8	4.968	4.554	5.382	11.592	670.31	106.62	WI
45	13.6	4.080	4.080	13.056	12.920	716.20	109.61	VT

```
plt.figure(figsize=(20,15))
sns.barplot(x="abbrev",y="total",data=top_total)
```

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



Above horizontal barplot shows the graphical representation states where total no of drivers involved in the crash from the state

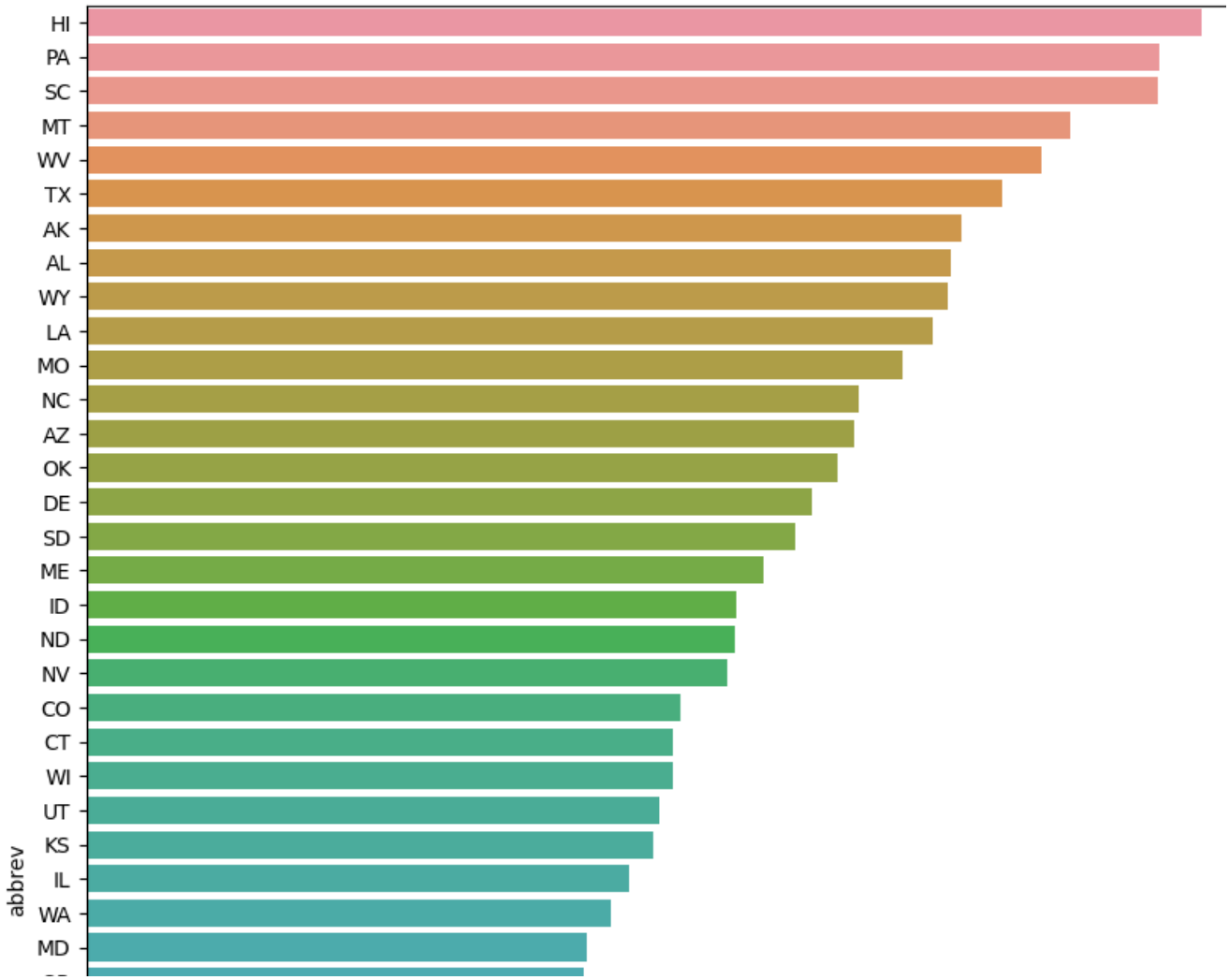
```
# Visualize the data in which crashes occured due to speeding
```

```
top15_speed =data.sort_values(by="speeding",ascending=False)
top15_speed
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
11	17.5	9.450	7.175	14.350	15.225	861.18	120.92	HI
38	18.2	9.100	5.642	17.472	16.016	905.99	153.86	PA
40	23.9	9.082	9.799	22.944	19.359	858.97	116.29	SC
26	21.4	8.346	9.416	17.976	18.190	816.21	85.15	MT
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56	WV
43	19.4	7.760	7.372	17.654	16.878	1004.75	156.83	TX
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	AK
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	AL
50	17.4	7.308	5.568	14.094	15.660	791.14	122.04	WY
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45	MO
33	16.8	6.552	5.208	15.792	13.608	708.24	127.82	NC
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	AZ
36	19.9	6.368	5.771	18.308	18.706	881.51	178.86	OK
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	DE

```
plt.figure(figsize=(10,15))
sns.barplot(x="speeding",y="abbrev",data=top15_speed)
```

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



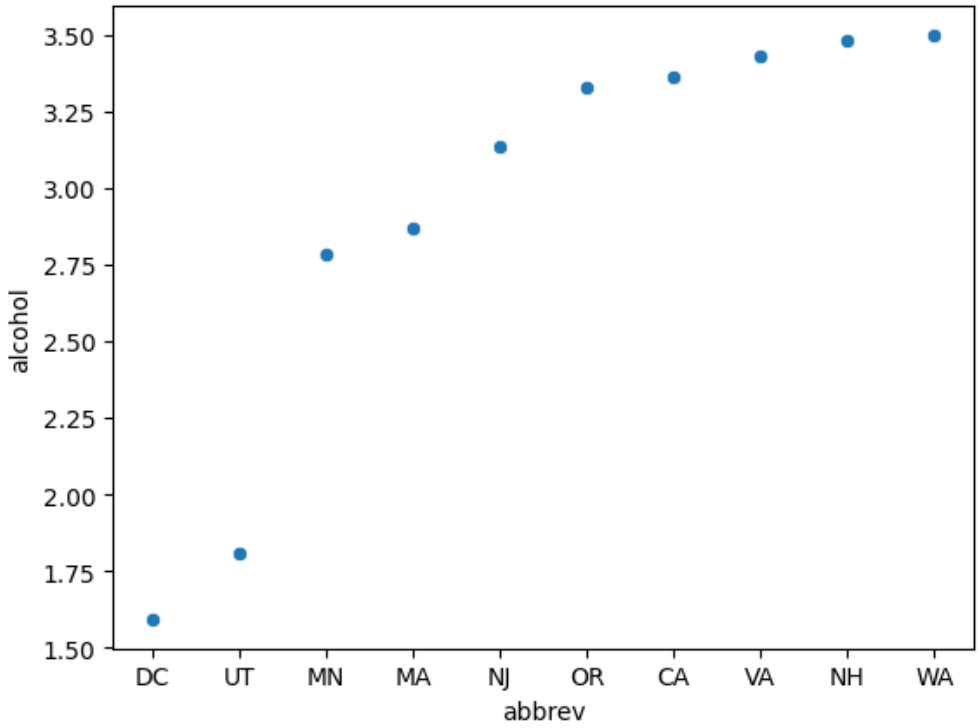
From the Vertical barplot we can analyze states where crashes occur due to speeding


```
#TO see which state less number of Crashes occured due to alcohol consumption
alcohol_crash=data.sort_values(by="alcohol",ascending=True).head(10)
alcohol_crash
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	DC
44	11.3	4.859	1.808	9.944	10.848	809.38	109.48	UT
23	9.6	2.208	2.784	8.448	8.448	777.18	133.35	MN
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.63	MA
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85	NJ
37	12.8	4.224	3.328	8.576	11.520	804.71	104.61	OR
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	CA
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72	VA
29	11.6	4.060	3.480	10.092	9.628	746.54	120.21	NH
47	10.6	4.452	3.498	8.692	9.116	890.03	111.62	WA

```
sns.scatterplot(x="abbrev",y="alcohol",data=alcohol_crash)
```

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



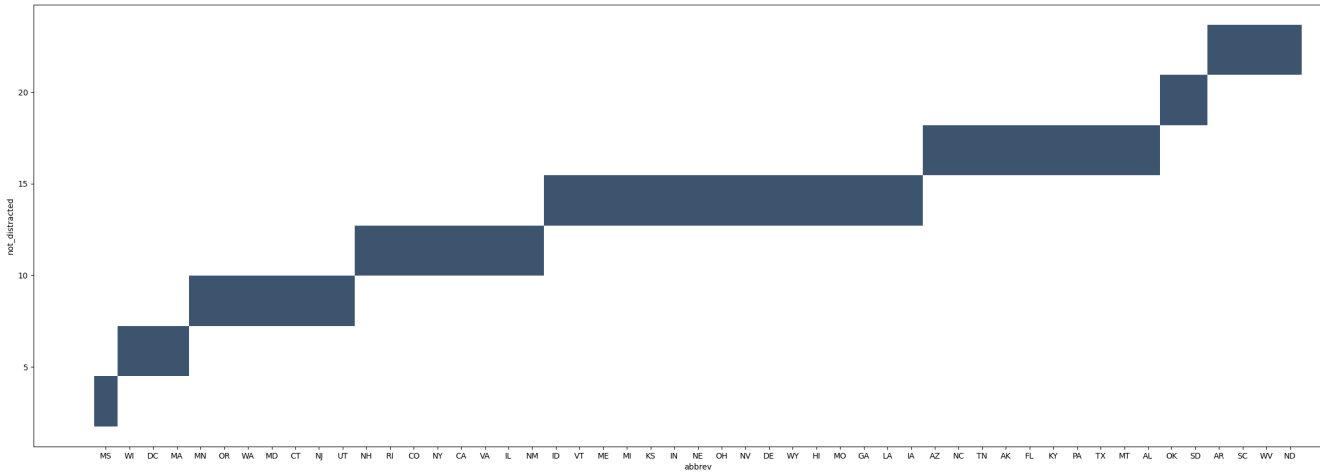
Above scatterplot can be used to analyze the states in which less number of Crashes occured due to alcohol consumption

```
# Visualize the data where crashes occured where involved In crash Who Were Not Distracted
distract=data.sort_values(by="not_distracted",ascending=True)
distract
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77	MS
49	13.8	4.968	4.554	5.382	11.592	670.31	106.62	WI
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	DC
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.63	MA
23	9.6	2.208	2.784	8.448	8.448	777.18	133.35	MN
37	12.8	4.224	3.328	8.576	11.520	804.71	104.61	OR
47	10.6	4.452	3.498	8.692	9.116	890.03	111.62	WA
20	12.5	4.250	4.000	8.875	12.375	1048.78	192.70	MD
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.02	CT
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85	NJ
44	11.3	4.859	1.808	9.944	10.848	809.38	109.48	UT
29	11.6	4.060	3.480	10.092	9.628	746.54	120.21	NH
39	11.1	3.774	4.218	10.212	8.769	1148.99	148.58	RI
5	13.6	5.032	3.808	10.744	12.920	835.50	139.91	CO
32	12.3	3.936	3.567	10.824	9.840	1234.31	150.01	NY
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	CA
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72	VA
13	12.8	4.608	4.352	12.032	12.288	803.11	139.15	IL
31	18.4	3.496	4.968	12.328	18.032	869.85	120.75	NM
12	15.3	5.508	4.437	13.005	14.994	641.96	82.75	ID
45	13.6	4.080	4.080	13.056	12.920	716.20	109.61	VT
19	15.1	5.738	4.530	13.137	12.684	661.88	96.57	ME
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26	MI
16	17.8	4.806	4.272	13.706	15.130	780.45	133.80	KS
14	14.5	3.625	4.205	13.775	13.775	710.46	108.92	IN
27	14.9	1.937	5.215	13.857	13.410	732.28	114.82	NE
35	14.1	3.948	4.794	13.959	11.562	697.73	133.52	OH
28	14.7	5.439	4.704	13.965	14.553	1029.87	138.71	NV
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	DE
50	17.4	7.308	5.568	14.094	15.660	791.14	122.04	WY
11	17.5	9.450	7.175	14.350	15.225	861.18	120.92	HI
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45	MO
10	15.6	2.964	3.900	14.820	14.508	913.15	142.80	GA

```
plt.figure(figsize=(30,10))
sns.histplot(x="abbrev",y="not_distracted",data=distract)
```

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



Above histplot will groups the similar states where not_distracted values are arranged in the ascending order and has similar ranges

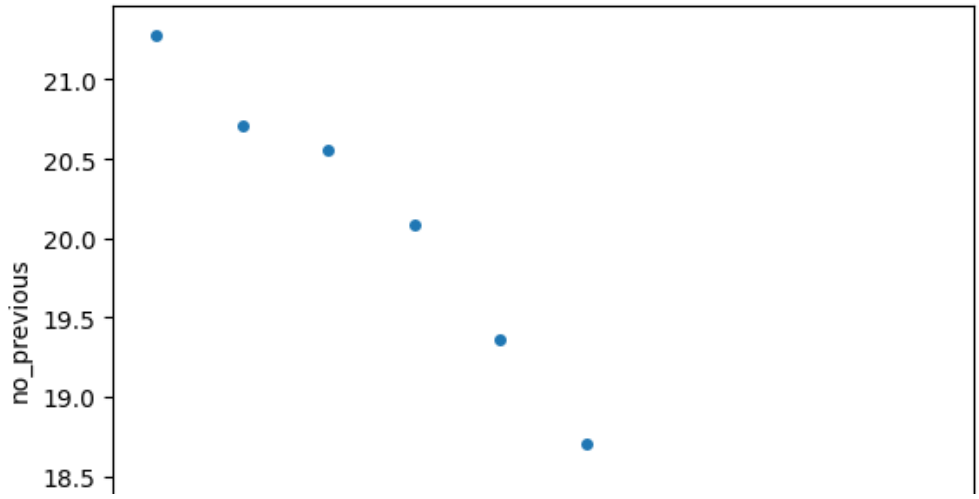
```
# To check the top 10 states where drivers were not involved in any previous crashes

prev=data.sort_values(by="no_previous",ascending=False).head(10)
prev
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	AR
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56	WV
34	23.9	5.497	10.038	23.661	20.554	688.75	109.72	ND
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
40	23.9	9.082	9.799	22.944	19.359	858.97	116.29	SC
36	19.9	6.368	5.771	18.308	18.706	881.51	178.86	OK
26	21.4	8.346	9.416	17.976	18.190	816.21	85.15	MT
31	18.4	3.496	4.968	12.328	18.032	869.85	120.75	NM
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	AZ
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77	MS

```
sns.swarmplot(x="abbrev",y="no_previous",data=prev)
```

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

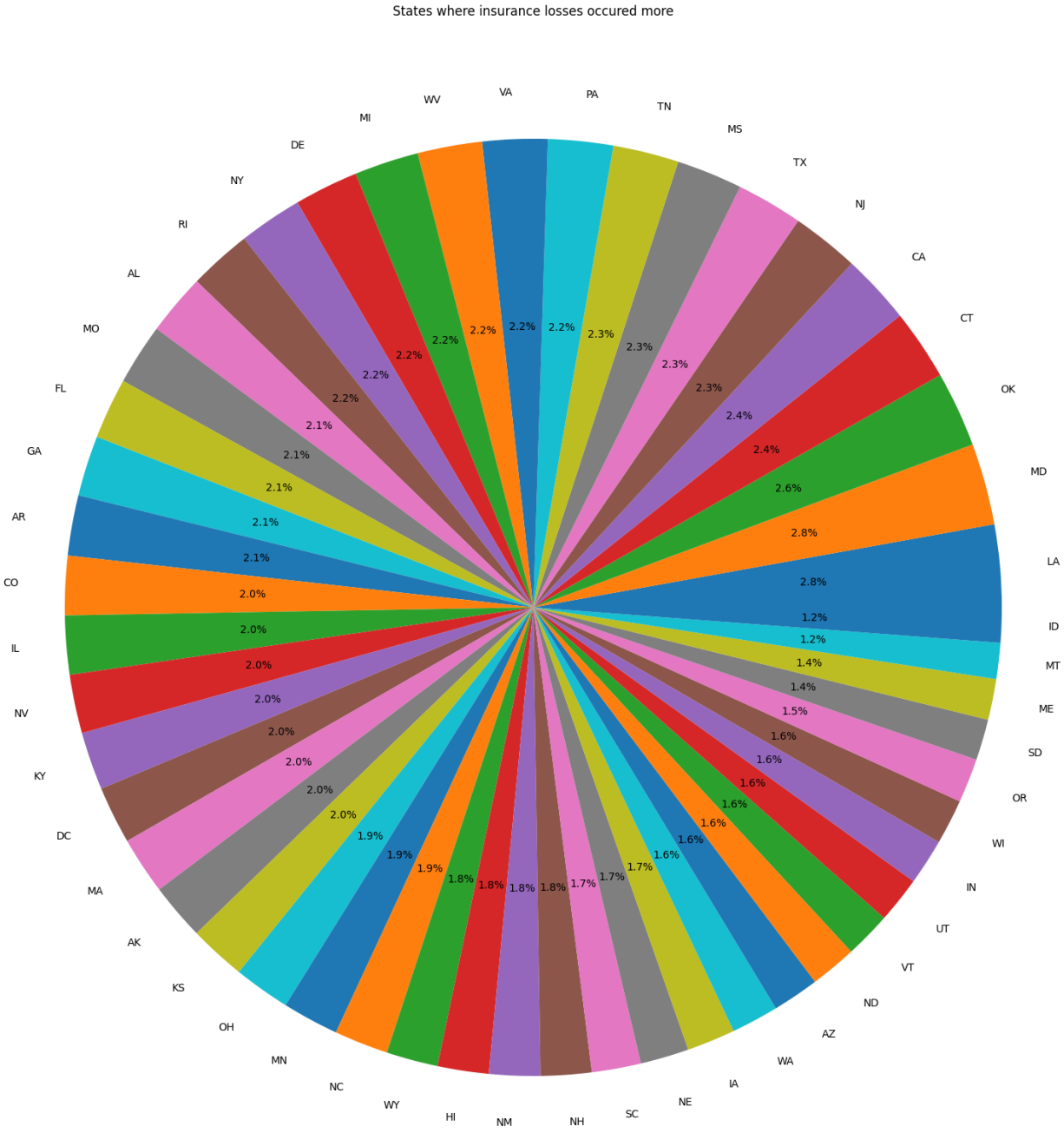


Above plot can be used to check where more no of drivers has no_previous records in the crashes

```
# States where insurance company acquired more losses
loss=data.sort_values(by="ins_losses",ascending=False)
loss
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
20	12.5	4.250	4.000	8.875	12.375	1048.78	192.70	MD
36	19.9	6.368	5.771	18.308	18.706	881.51	178.86	OK
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.02	CT
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	CA
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85	NJ
43	19.4	7.760	7.372	17.654	16.878	1004.75	156.83	TX
24	17.6	2.640	5.456	1.760	17.600	896.07	155.77	MS
42	19.5	4.095	5.655	15.990	15.795	767.91	155.57	TN
38	18.2	9.100	5.642	17.472	16.016	905.99	153.86	PA
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72	VA
48	23.8	8.092	6.664	23.086	20.706	992.61	152.56	WV
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26	MI
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	DE
32	12.3	3.936	3.567	10.824	9.840	1234.31	150.01	NY
39	11.1	3.774	4.218	10.212	8.769	1148.99	148.58	RI
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	AL
25	16.1	6.923	5.474	14.812	13.524	790.32	144.45	MO
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
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	AR
5	13.6	5.032	3.808	10.744	12.920	835.50	139.91	CO
13	12.8	4.608	4.352	12.032	12.288	803.11	139.15	IL
28	14.7	5.439	4.704	13.965	14.553	1029.87	138.71	NV
17	21.4	4.066	4.922	16.692	16.264	872.51	137.13	KY
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	DC
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.63	MA

```
plt.figure(figsize=(20,30))
plt.pie(x="ins_losses",labels="abbrev",data=loss,autopct='%.1f%%',startangle=0)
plt.title("States where insurance losses occured more")
plt.show()
```


States where insurance losses occured more
WV
VA
PA
TN
MS
TX
NJ
CA
CT
OK
MD
LA
ID
MT
ME
SD
OR
WI
IN
UT
VT
ND
AZ
WA
IA
NE
SC
NH
NM
HI
WY
NC
MN
OH
KS
AK
MA
DC
KY
NV
IL
CO
AR
GA
FL
MO
AL
RI
NY
DE
MI
2.2%
2.2%
2.2%
2.3%
2.3%
2.3%
2.3%
2.4%
2.4%
2.6%
2.8%
2.8%
1.2%
1.2%
1.4%
1.4%
1.5%
1.6%
1.6%
1.6%
1.6%
1.6%
1.6%
1.6%
1.7%
1.7%
1.8%
1.8%
1.8%
1.8%
1.9%
1.9%
2.0%
2.0%
2.0%
2.0%
2.0%
2.0%
2.0%
2.0%
2.1%
2.1%
2.1%
2.1%
2.1%
2.2%
2.2%
2.2%
2.2%

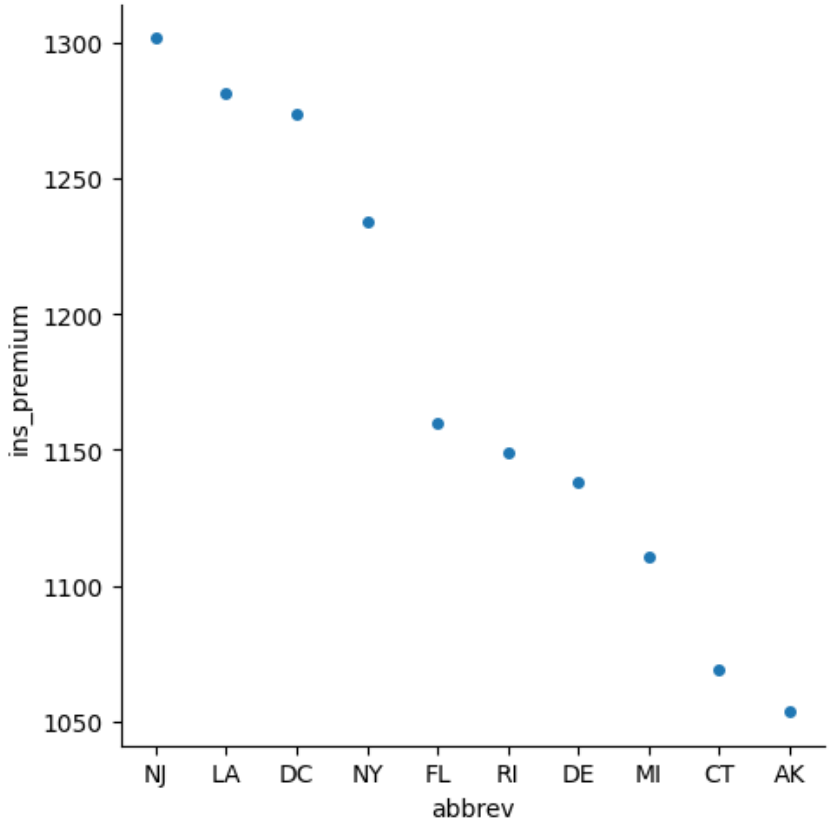
From above piechat we can analyze in which area we acquire more losses and percentage of loss acquired in the states

```
# Top States in which collided cars have more insurace premium value
premium =data.sort_values(by="ins_premium",ascending=False).head(10)
premium
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
30	11.2	1.792	3.136	9.632	8.736	1301.52	159.85	NJ
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.78	LA
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	DC
32	12.3	3.936	3.567	10.824	9.840	1234.31	150.01	NY
9	17.9	3.759	5.191	16.468	16.826	1160.13	144.18	FL
39	11.1	3.774	4.218	10.212	8.769	1148.99	148.58	RI
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	DE
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.26	MI
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.02	CT
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	AK

```
sns.catplot(x="abbrev",y="ins_premium",data=premium)
```

```
<seaborn.axisgrid.FacetGrid at 0x7a82be0f1d20>
```



Above Catplot can be used to check which state has higest premium value for the cars

```
# Analyze the relation between insurance premium amount and insurance losses
rel =data.sort_values(by="ins_premium",ascending=False)
rel
```