```python
import seaborn as sns
```

```python
df = sns.load_dataset('car_crashes')
```

```
In [ ]: df
```

Out[3]:

	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 [ ]: sns.barplot(data=df.iloc[0:25,:],x='abbrev',y='total')
```
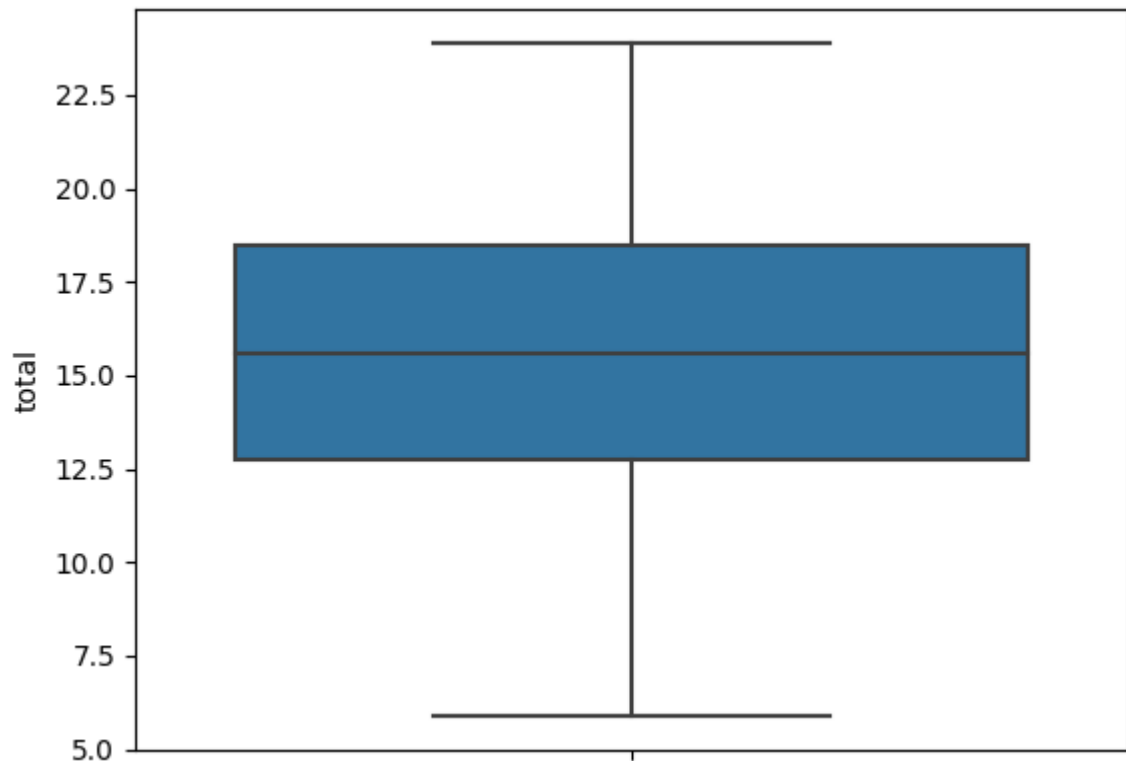
Out[16]: <Axes: xlabel='abbrev', ylabel='total'>
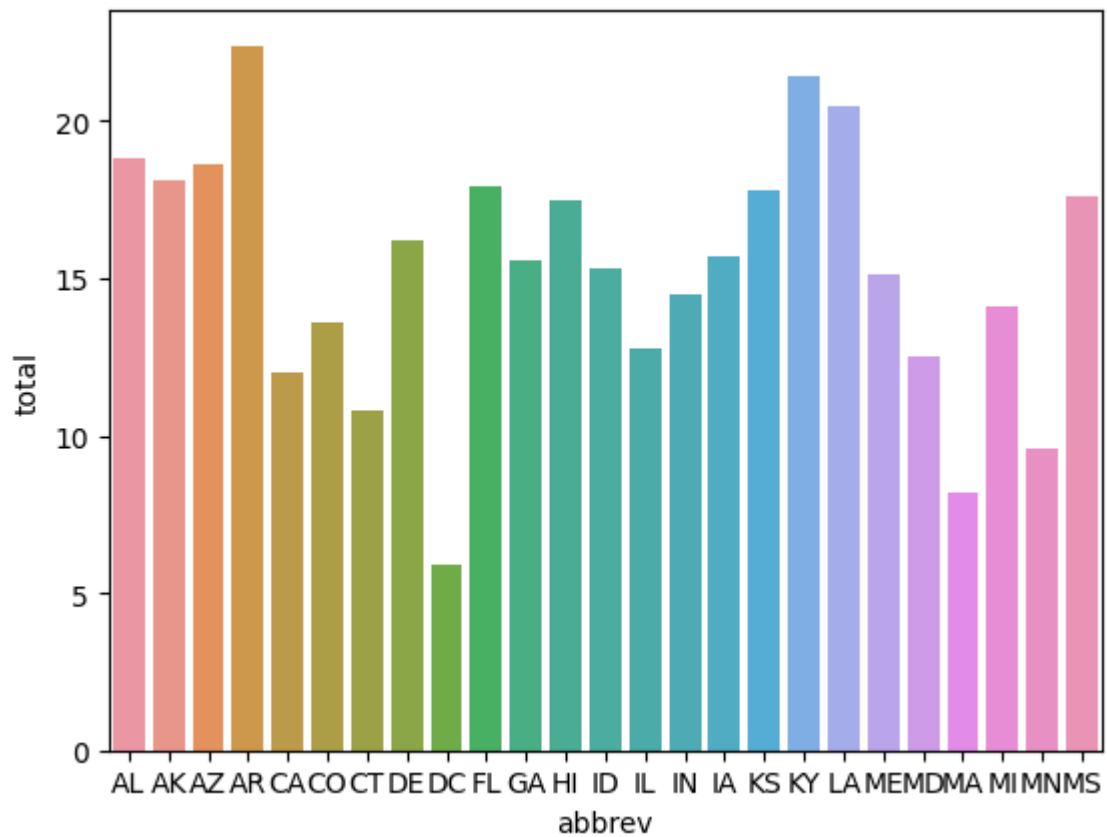
`sns.barplot(data=df.iloc[25:50,:],x='abbrev',y='total')`

Out[17]: `<Axes: xlabel='abbrev', ylabel='total'>`
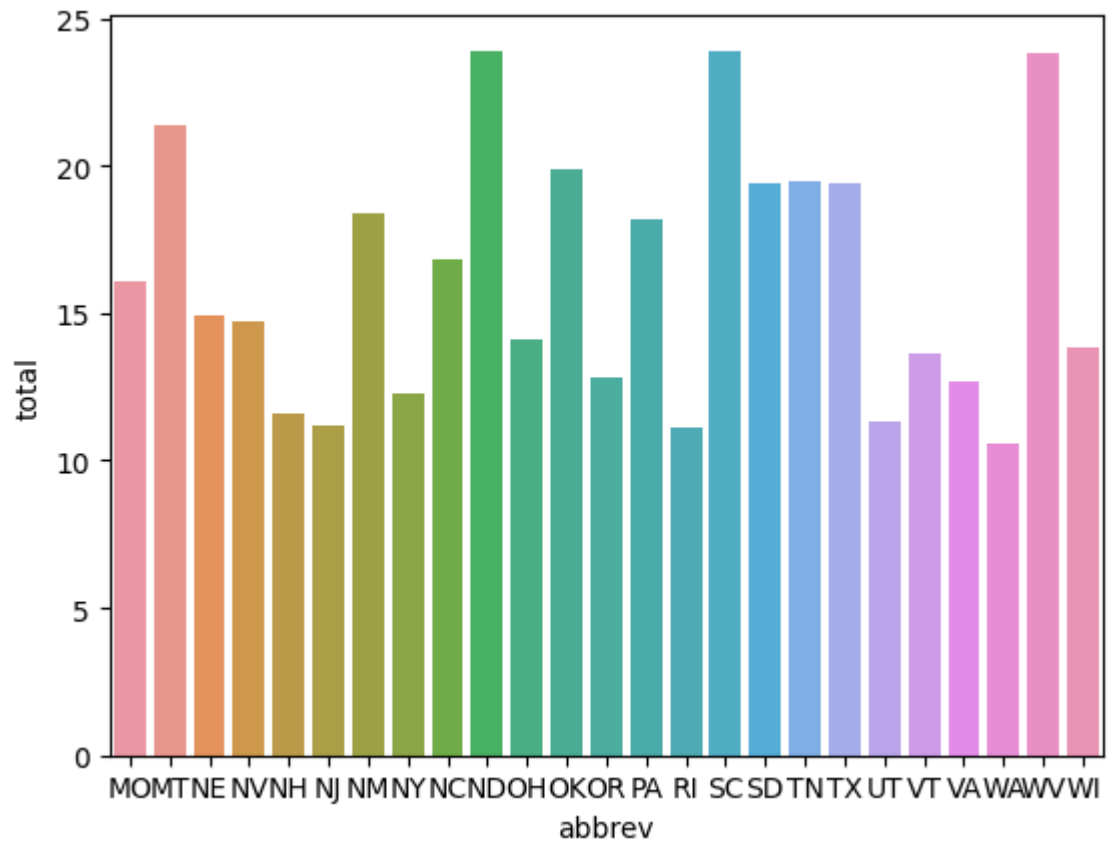
```
In [ ]: sns.boxplot(y='total',data=df)
```

Out[10]: `<Axes: ylabel='total'>`


```
In [ ]: df.describe()
```

Out[11]:

	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

Inference

The speeding and alcohol cases have averages that are far from the total mean. This implies that speeding and alcohol cases are rare.

However, the not_distracted and no_previous averages are closer to the mean, implying that they are common cases.

```
In [ ]: sns.boxplot(y='total',data=df)
```
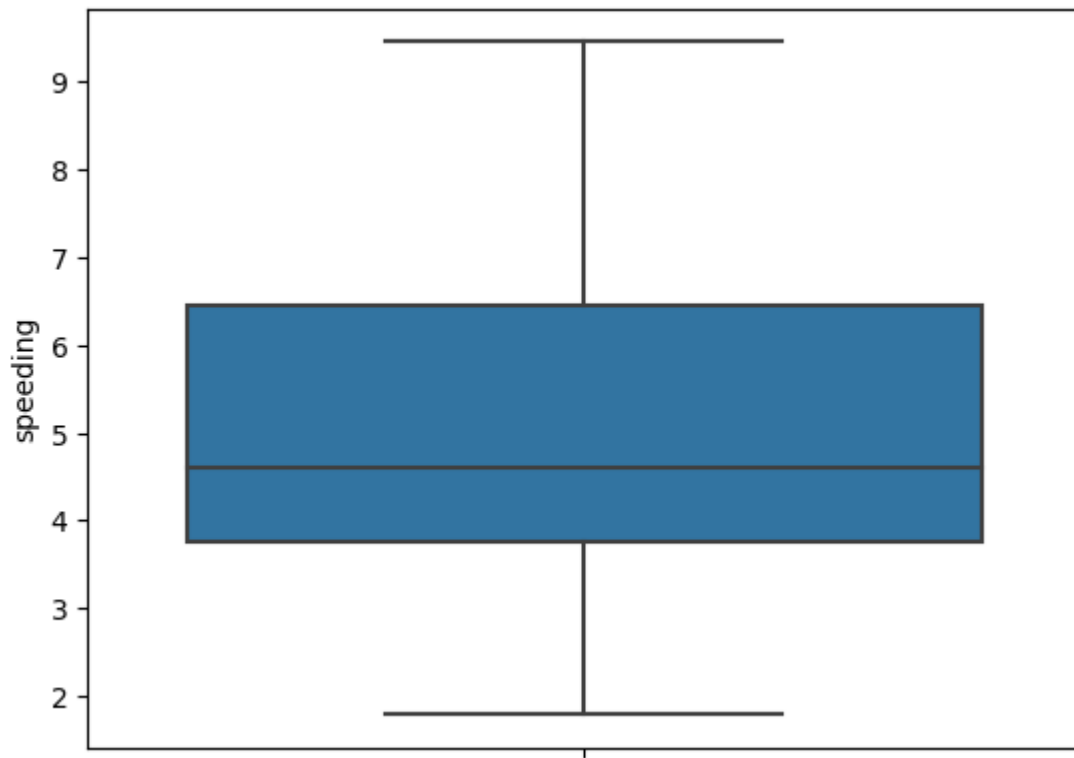
Out[4]: <Axes: ylabel='total'>
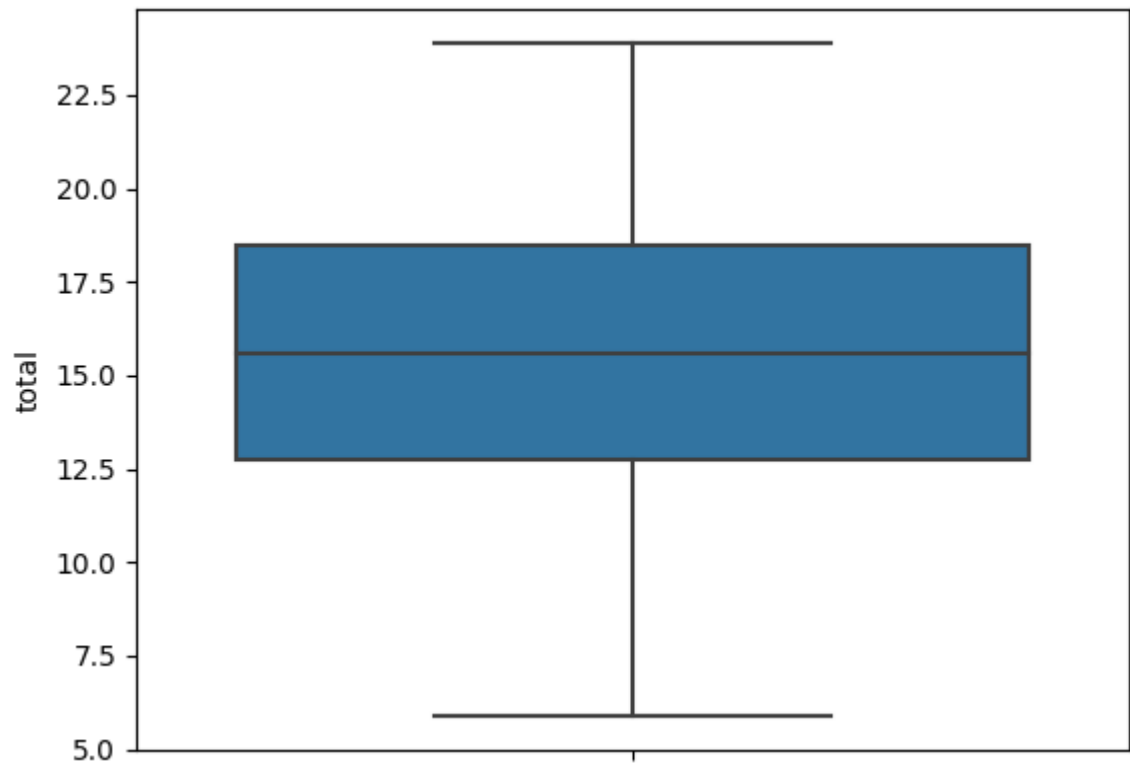
Inference

There are no outliers in the total. The maximum value is 23.9 The minimum value is 5.9 q1 = 12.75 The median is 15.6 q3 = 18.5 The IQR is 5.75
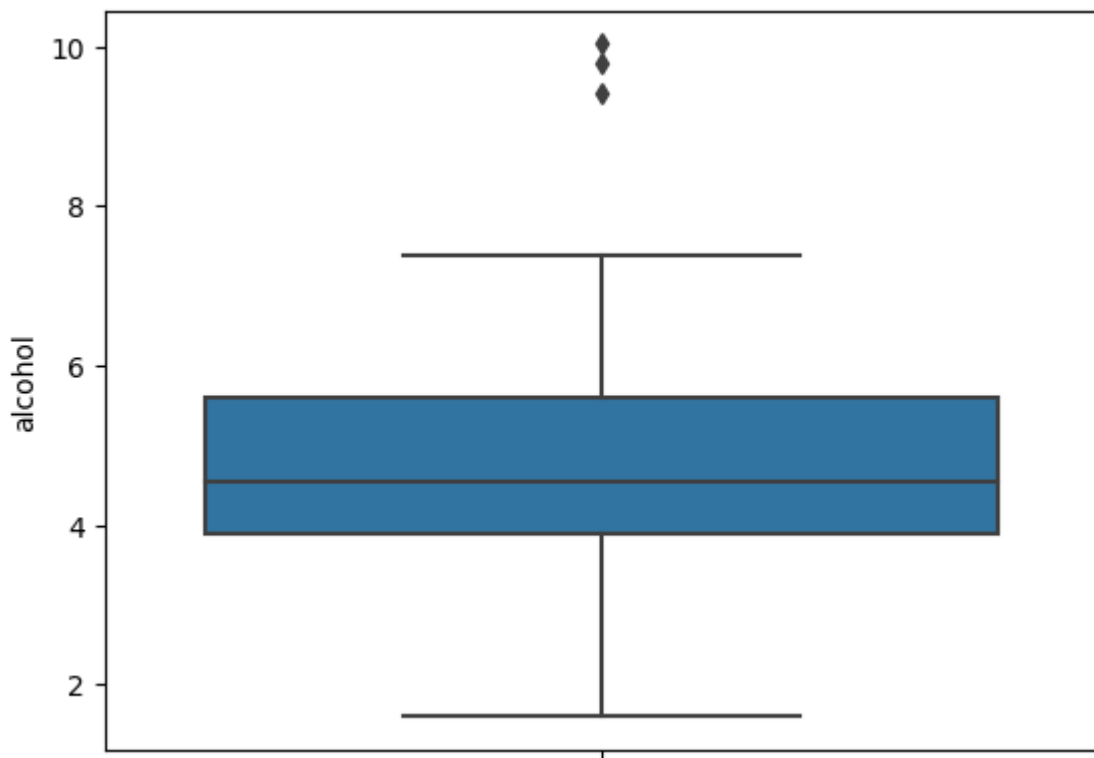
```
In [ ]:  sns.boxplot(y='speeding',data=df)
```

Out[5]: <Axes: ylabel='speeding'>


```
In [ ]:  sns.boxplot(y='alcohol',data=df)
```
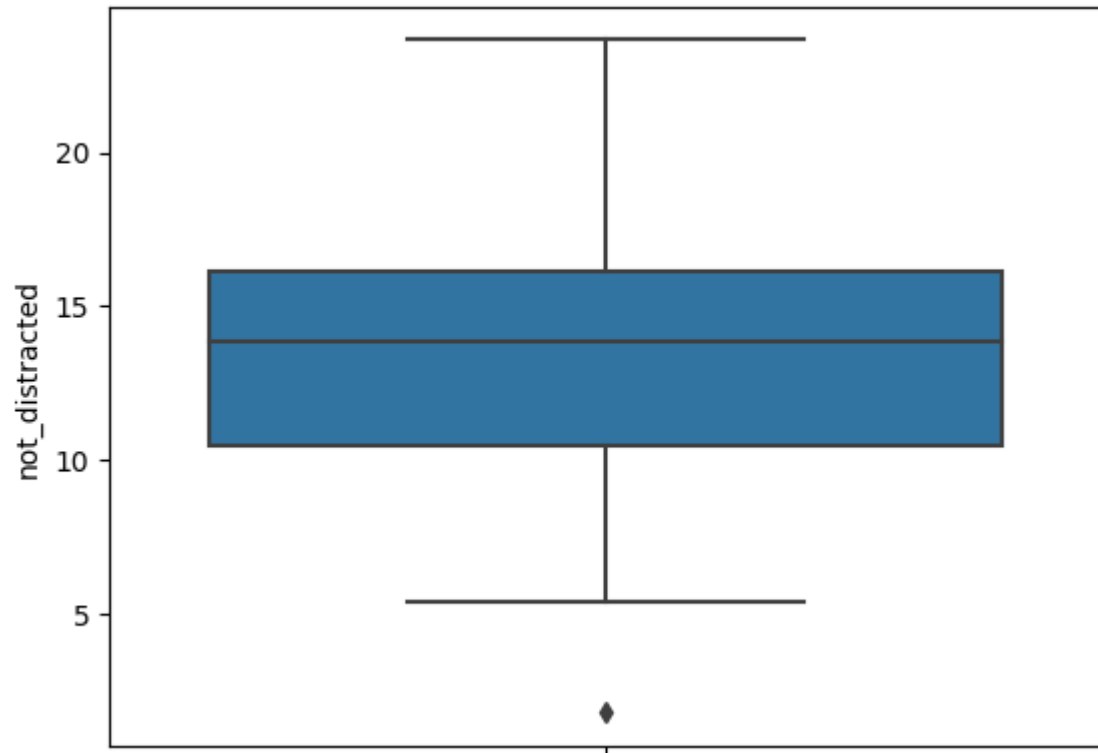
Out[6]: <Axes: ylabel='alcohol'>

Inference

The alcohol cases have three outliers which are significantly higher than the upper quartile.

```
In [ ]: sns.boxplot(y='not_distracted',data=df)
```

Out[8]: <Axes: ylabel='not_distracted'>

Inference

The is one outlier in the undistracted cases. It is lower than the lower quartile.

In [ ]: `sns.boxplot(y='no_previous',data=df)`
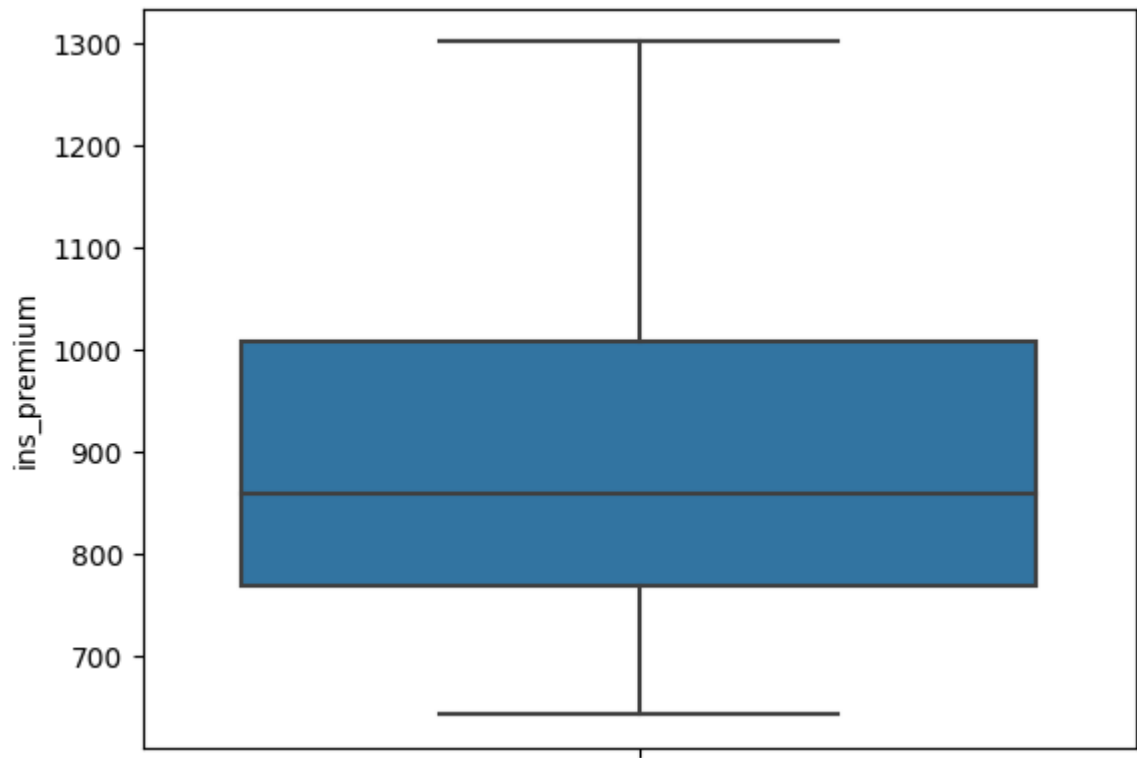
Out[9]: `<Axes: ylabel='no_previous'>`
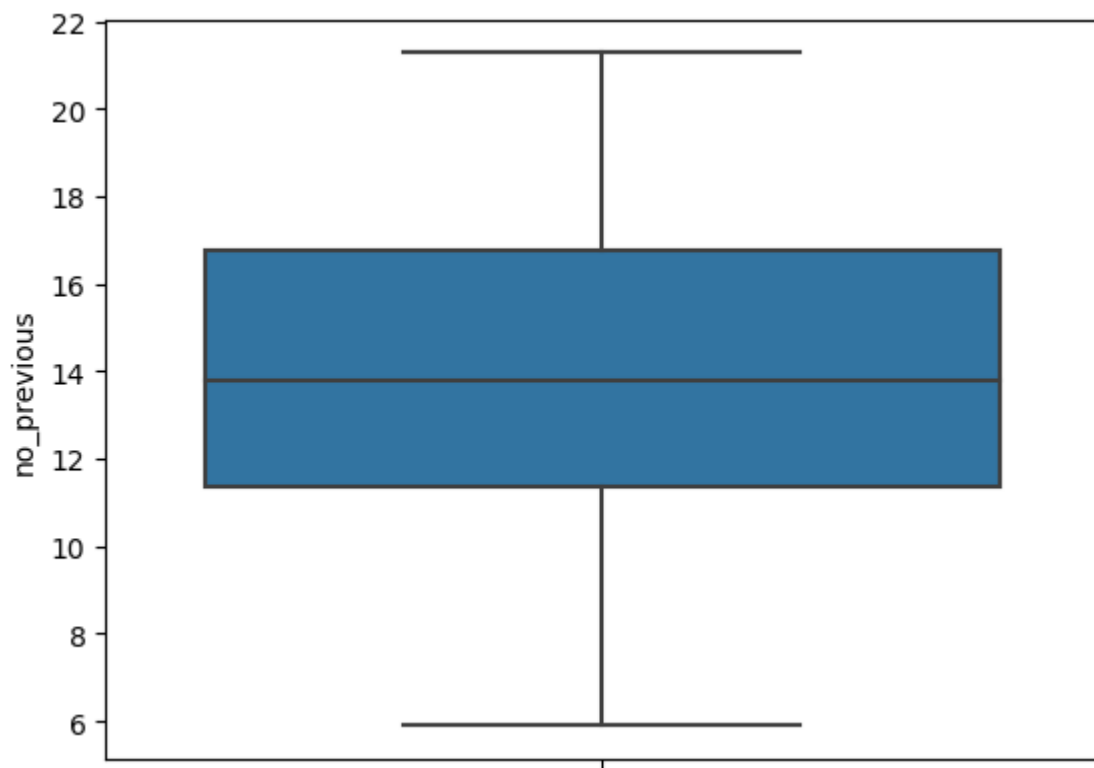
Inference

There are no outliers in the 'new car crash' cases
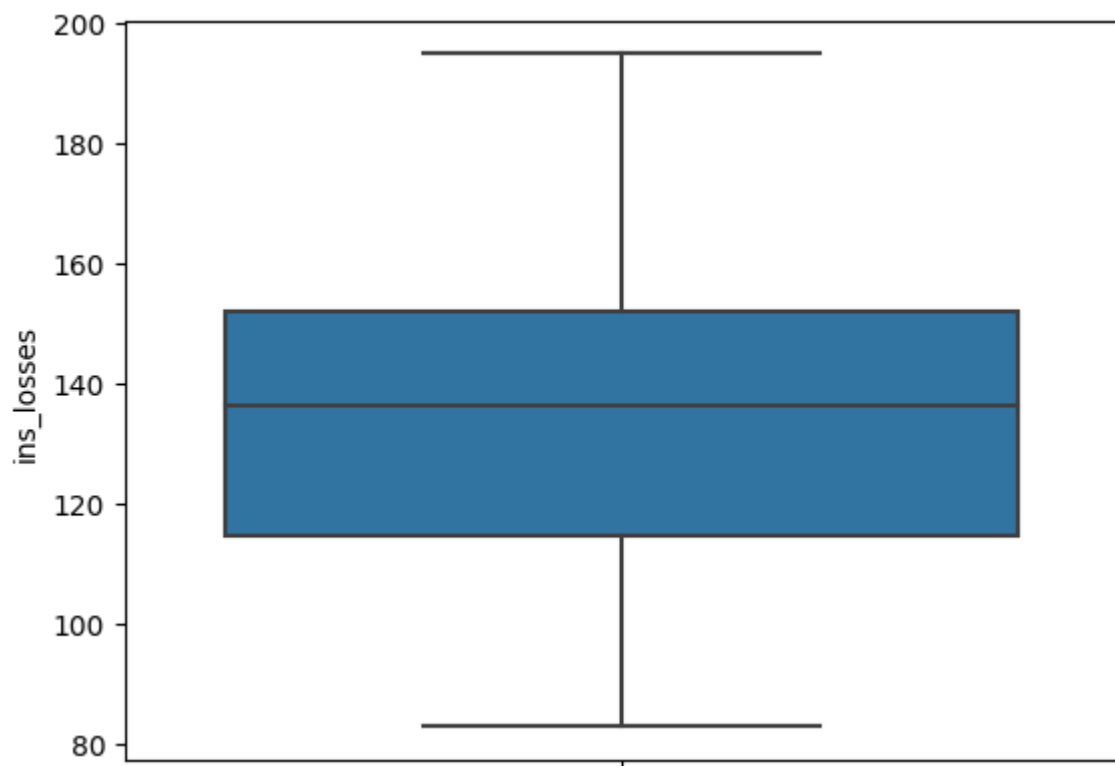
In [ ]: `sns.boxplot(y='ins_premium',data=df)`

Out[10]: `<Axes: ylabel='ins_premium'>`

In [ ]: `sns.boxplot(y='ins_losses',data=df)`
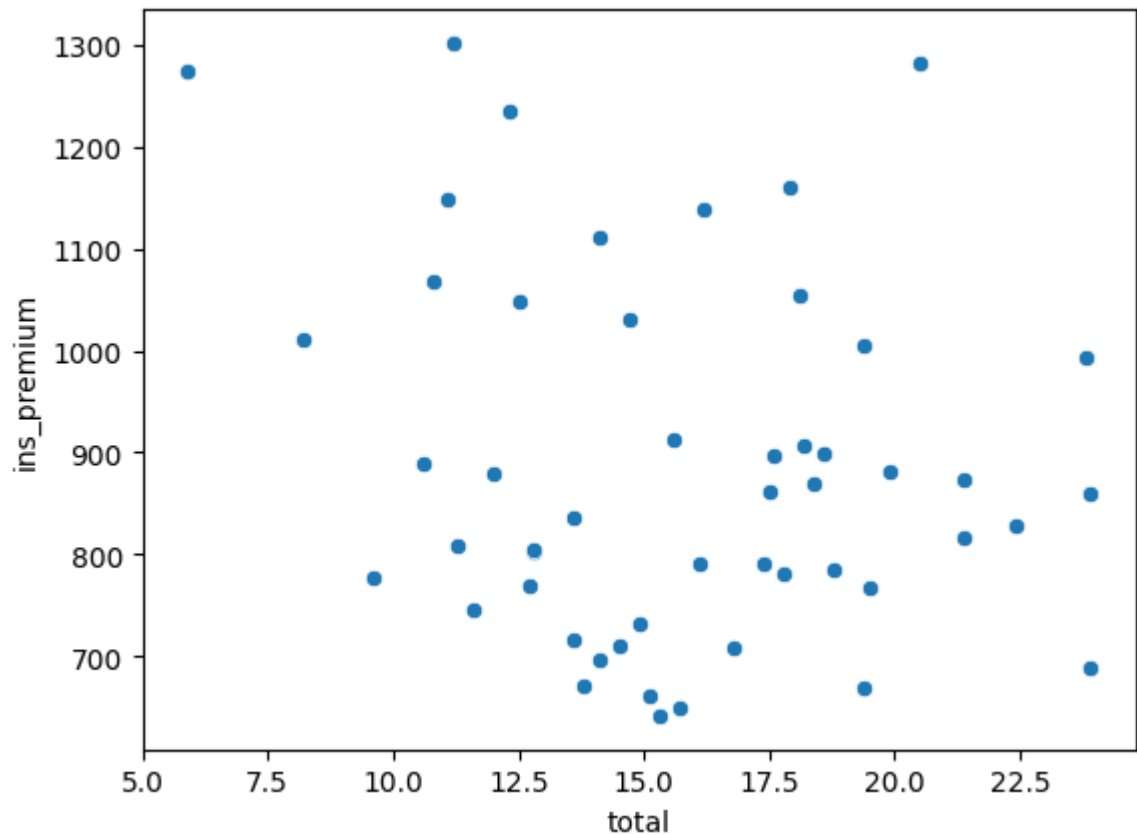
Out[11]: `<Axes: ylabel='ins_losses'>`

Inference

There are no outliers in the insurance values. The average insurance paid is 887. The average insurance loss is $25.

```
In [ ]: sns.scatterplot(x='total',y='ins_premium',data=df)
```

```
Out[12]: <Axes: xlabel='total', ylabel='ins_premium'>
```
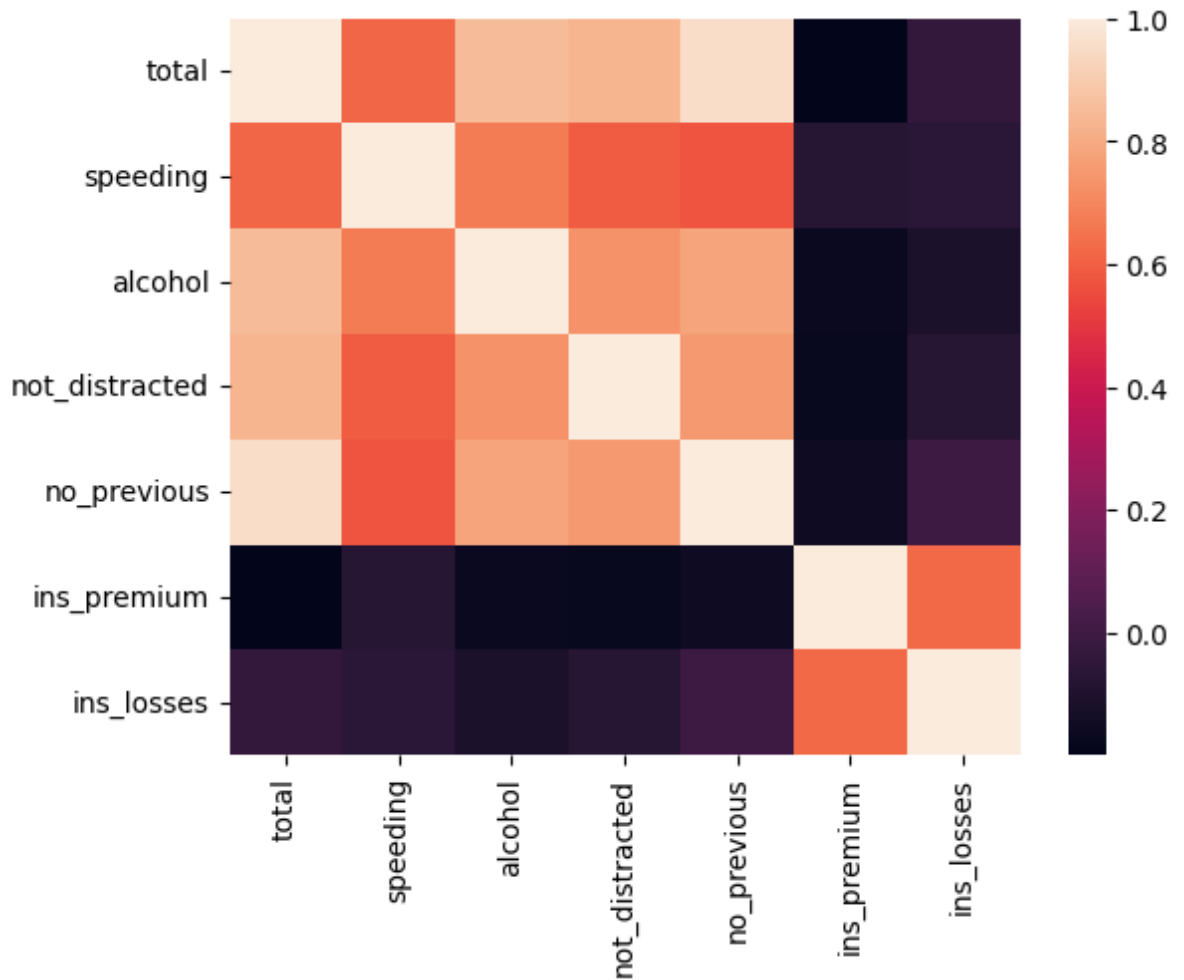

Inference

There is no correlation between the total and the insurance paid

```
In [ ]: corr= df.iloc[:,0:7].corr()
```

```
In [ ]:  sns.heatmap(corr)
```

Out[18]: <Axes: >

Inference

Insurance loss has the relatively strongest correlation with first accidents ('no_previous'). The strongest correlation premium insurance has is with speeding accidents.

```
In [ ]:    sns.distplot(df['total'])
```

```
<ipython-input-24-2ba73417f012>: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 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df['total'])
```
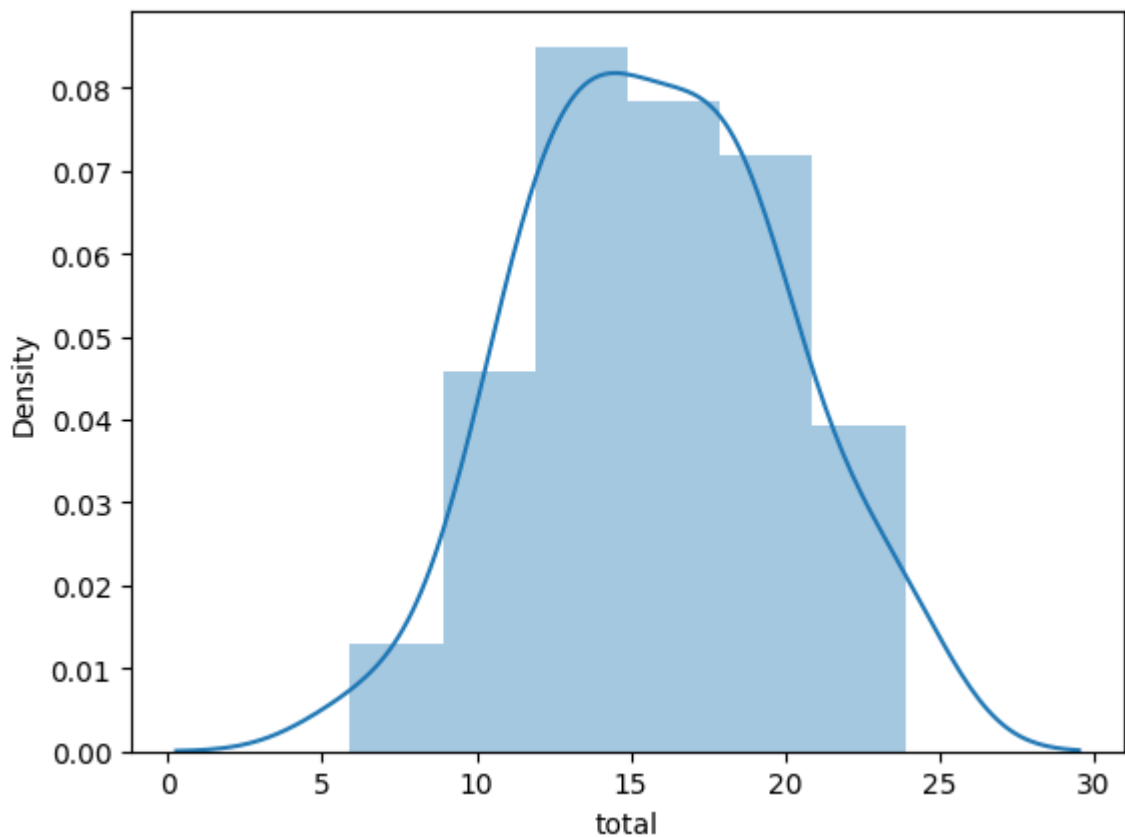
Out[24]:   <Axes: xlabel='total', ylabel='Density'>
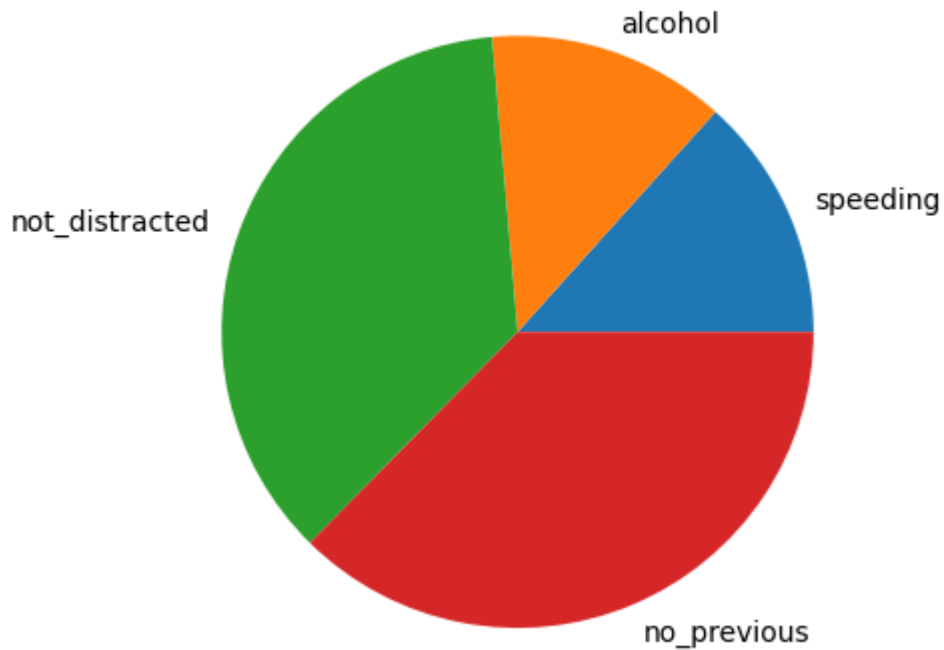

Inference

Most crash totals fall between 10 and 20.

```
In [ ]:    import matplotlib.pyplot as plt
```

```
In [ ]: plt.pie(df.iloc[:,1:5].sum(),labels=['speeding','alcohol','not_distracted','no_
```

Out[43]: ([<matplotlib.patches.Wedge at 0x7930d18f1ba0>,
          <matplotlib.patches.Wedge at 0x7930d18f1ab0>,
          <matplotlib.patches.Wedge at 0x7930d18f2470>,
          <matplotlib.patches.Wedge at 0x7930d18f28f0>],
         [Text(1.0047827521124513, 0.44767356528761915, 'speeding'),
          Text(0.3488580016798656, 1.0432152676528135, 'alcohol'),
          Text(-1.0350010951711164, 0.3725221241679345, 'not_distracted'),
          Text(0.42467804535234077, -1.0147159985905985, 'no_previous')])

Inference

Most accidents are first-time The second-largest value is for undistracted accidents Speeding accidents are the least occuring.

In [ ]:

Out[36]: [total   805.3
         dtype: float64,
         total   805.300
         speeding   254.908
         dtype: float64,
         total   805.300
         speeding   254.908
         alcohol   249.226
         dtype: float64,
         total   805.300
         speeding   254.908
         alcohol   249.226
         not_distracted   692.232
         dtype: float64]

In [ ]: