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

sb.get_dataset_names()
```

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

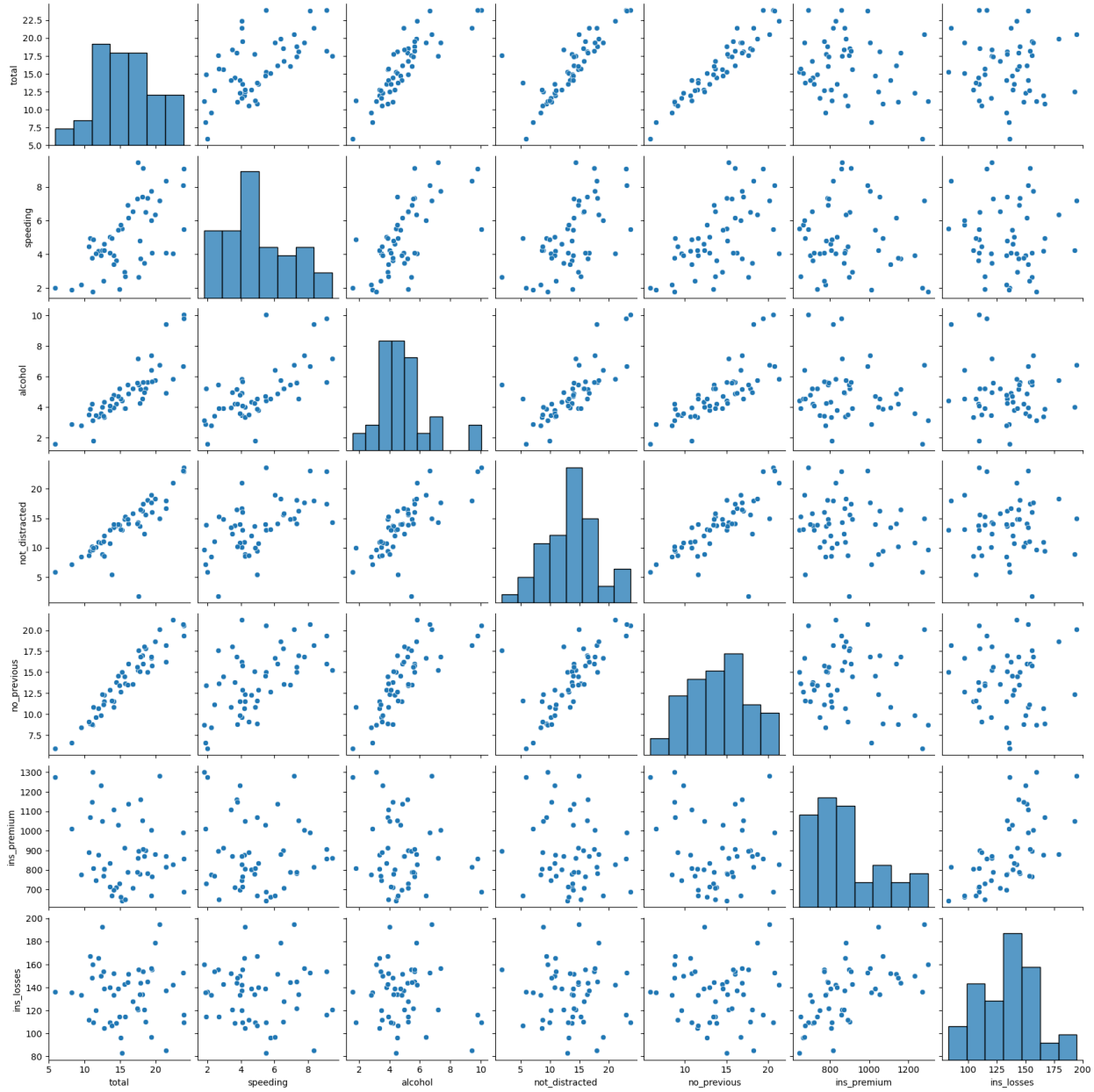
```
data=sns.load_dataset('car_crashes')

data.head()
```

```
   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


   ins_losses abbrev
0      145.08     AL
1      133.93     AK
2      110.35     AZ
```

```
3       142.39      AR
4       165.63      CA
```

```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x157d82ed0>
```

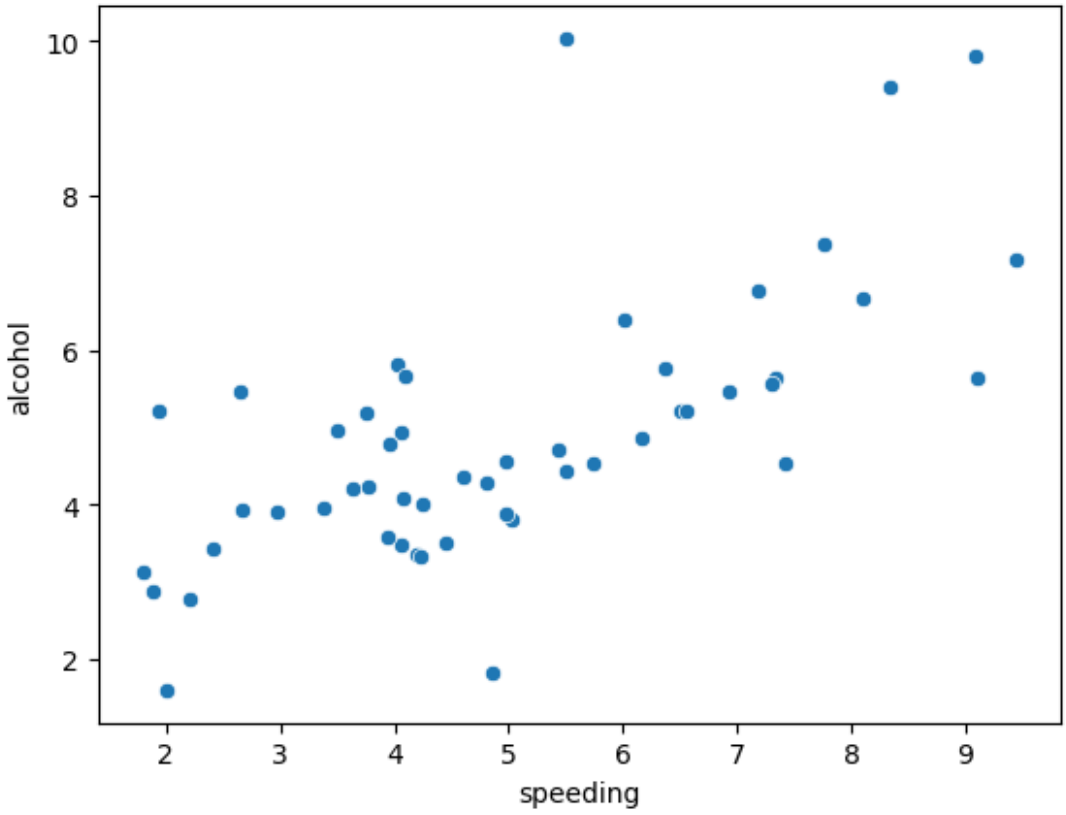


Pairplot.

Inference: You can observe relationships between different numerical variables, like "total" vs. "speeding," "total" vs. "alcohol," etc.

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

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

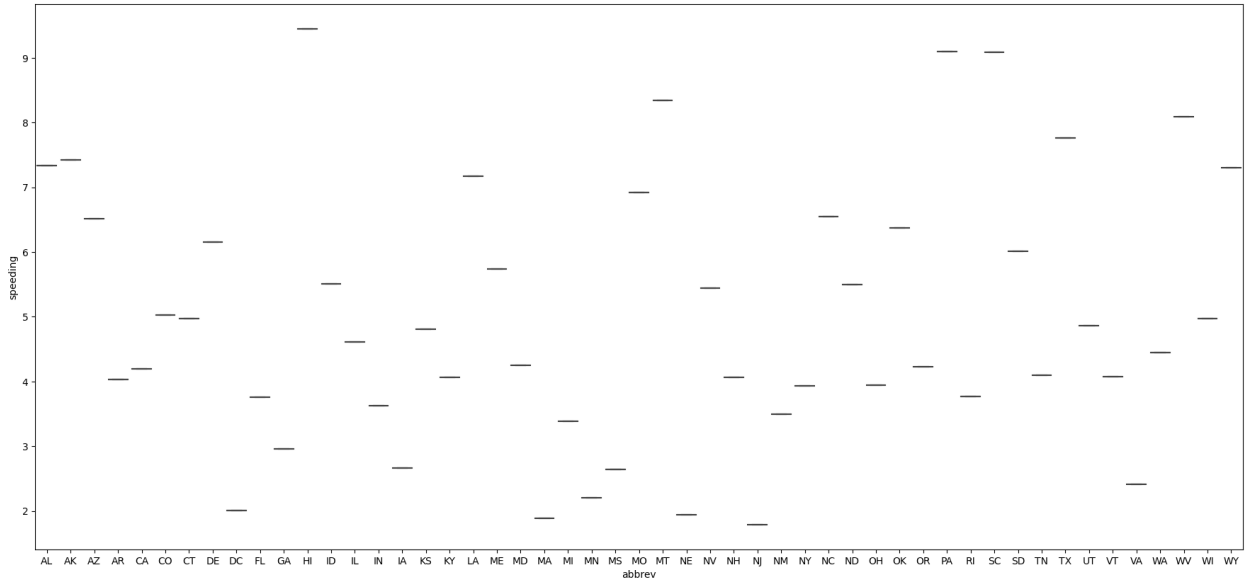


Scatter Plot

Inference: You can see if there's a correlation between speeding and alcohol-related crashes and similarly for others

```
plt.figure(figsize=(22,10))
sns.boxplot(x="abbrev",y="speeding",data=data)
```

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

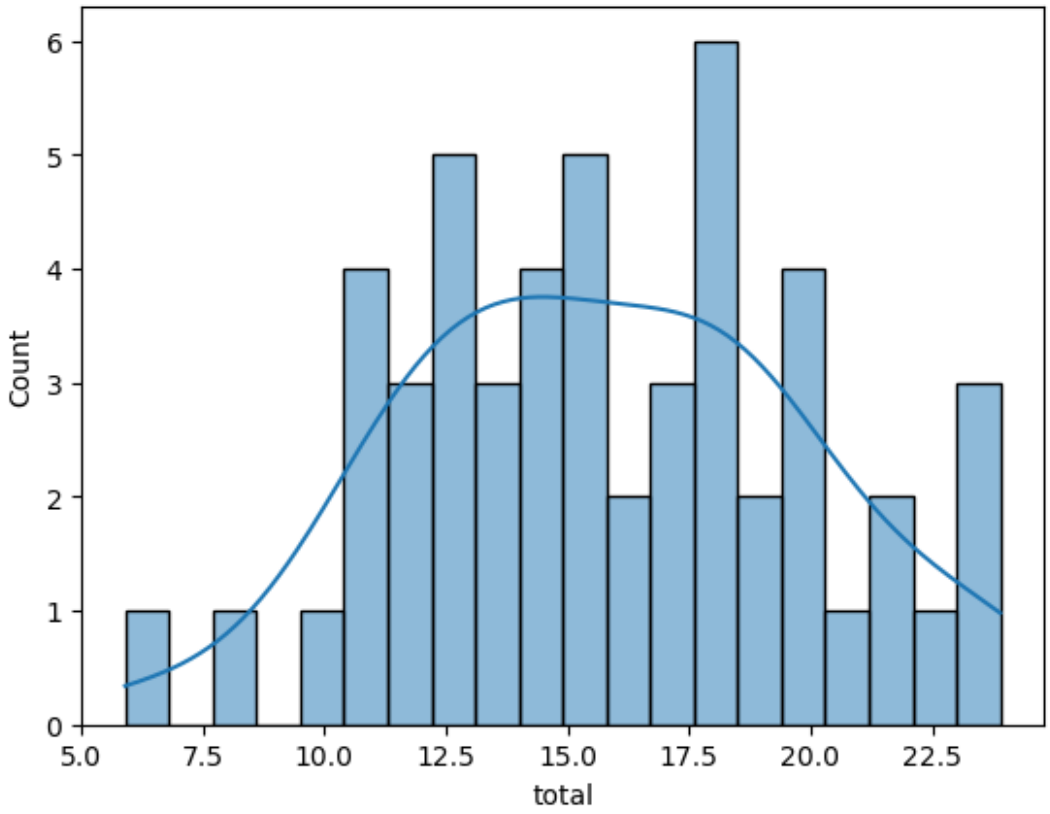


Box plot

Inference: You can compare the distribution of speeding across different states.

```
sns.histplot(data["total"],bins=20,kde=True)
```

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



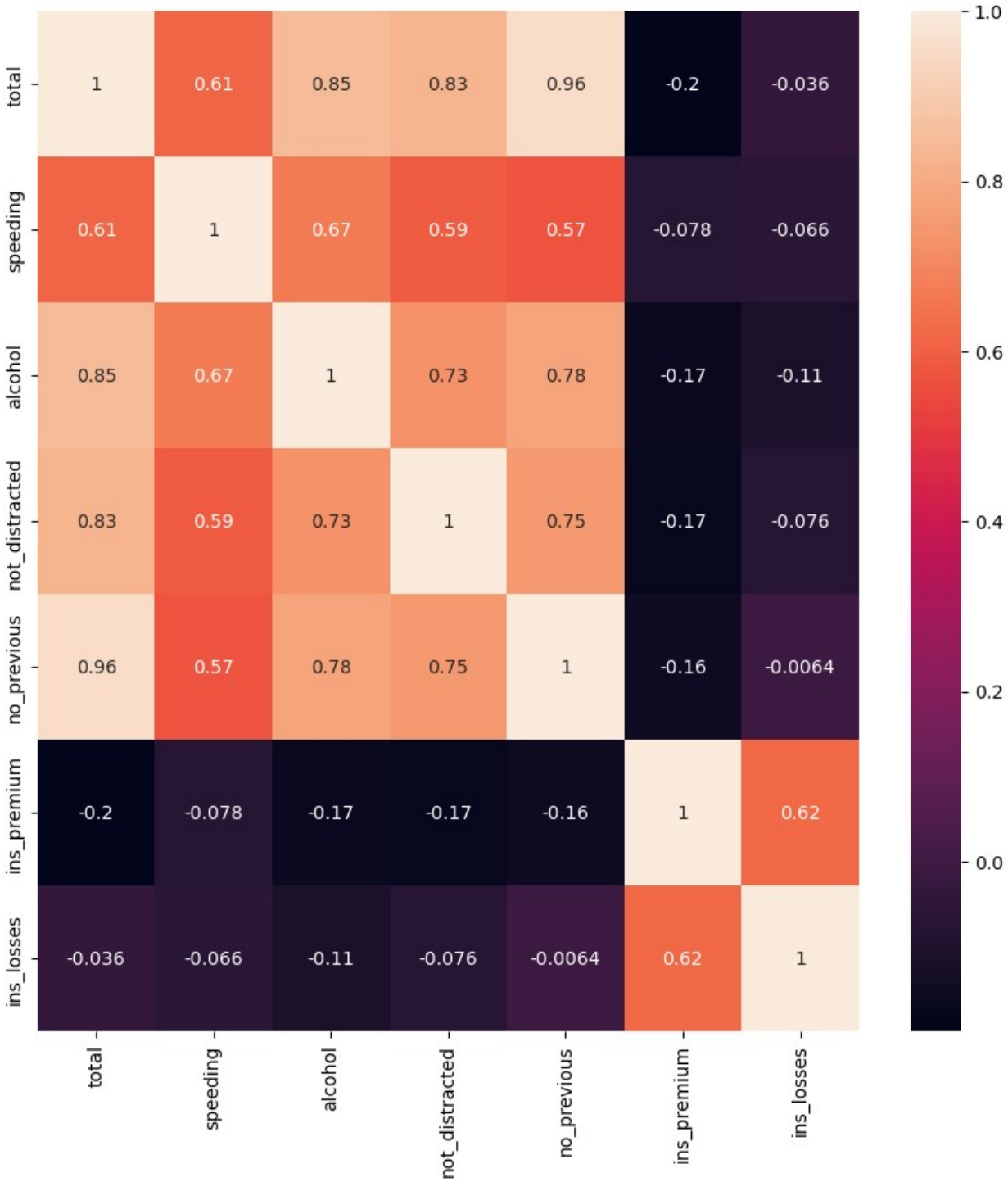
Histogram and KDE (Kernel Density Estimation)

Inference: You can visualize the distribution of total crashes and KDE overlay allows you to visualize both the raw data distribution (histogram) and the smoothed estimate of the data's probability density (KDE).

```
plt.figure(figsize=(10,10))
corr=data.corr()
sns.heatmap(corr,annot=True)
```

```
/var/folders/75/l1625v4n7qnbfp296v2f82ch0000gn/T/
ipykernel_2306/1162489340.py:2: 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=data.corr()
```

```
<Axes: >
```

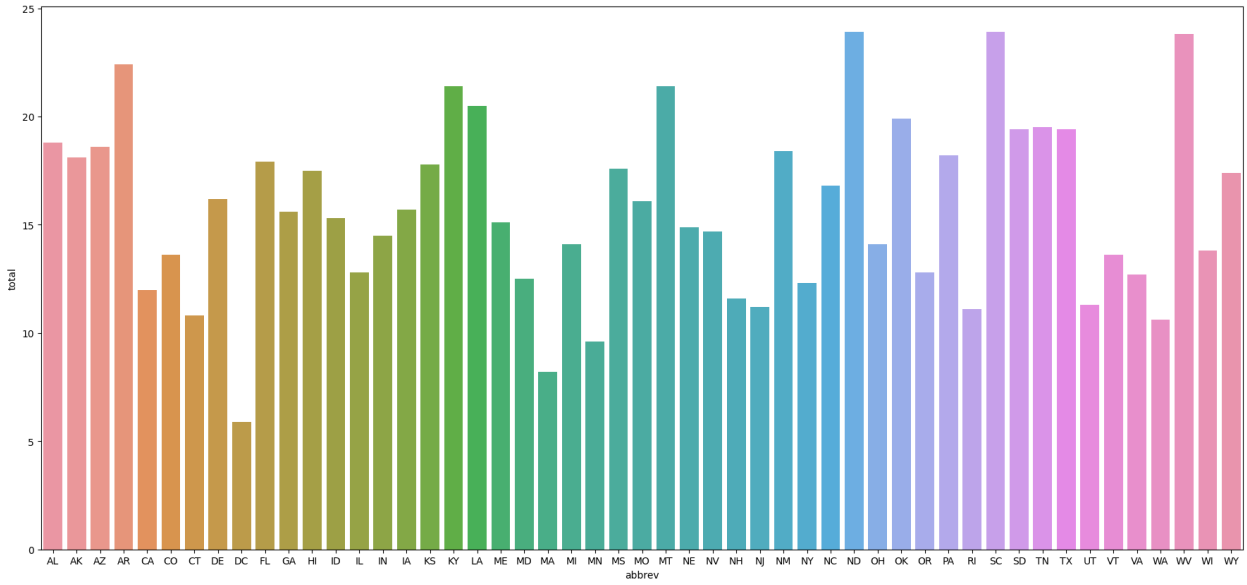

total
speeding
alcohol
not_distracted
no_previous
ins_premium
ins_losses
1 0.61 0.85 0.83 0.96 -0.2 -0.036
0.61 1 0.67 0.59 0.57 -0.078 -0.066
0.85 0.67 1 0.73 0.78 -0.17 -0.11
0.83 0.59 0.73 1 0.75 -0.17 -0.076
0.96 0.57 0.78 0.75 1 -0.16 -0.0064
-0.2 -0.078 -0.17 -0.17 -0.16 1 0.62
-0.036 -0.066 -0.11 -0.076 -0.0064 0.62 1
total
speeding
alcohol
not_distracted
no_previous
ins_premium
ins_losses
1.0
0.8
0.6
0.4
0.2
0.0

Heatmap

Inference: You can identify correlations between different numerical variables.

```
plt.figure(figsize=(22,10))
sns.barplot(x="abbrev",y="total",data=data)
```

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

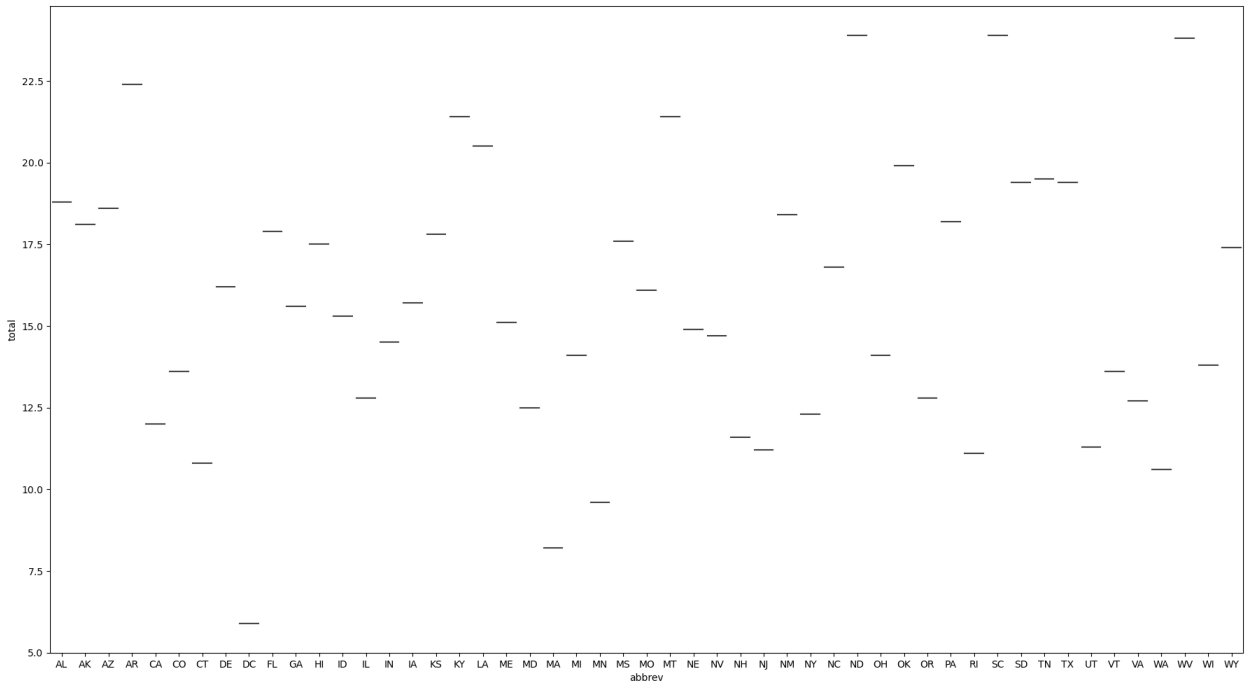


Barplot

Inference: You can compare the total crashes across different states.

```
plt.figure(figsize=(22,12))
sns.violinplot(x="abbrev",y="total",data=data)
```

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

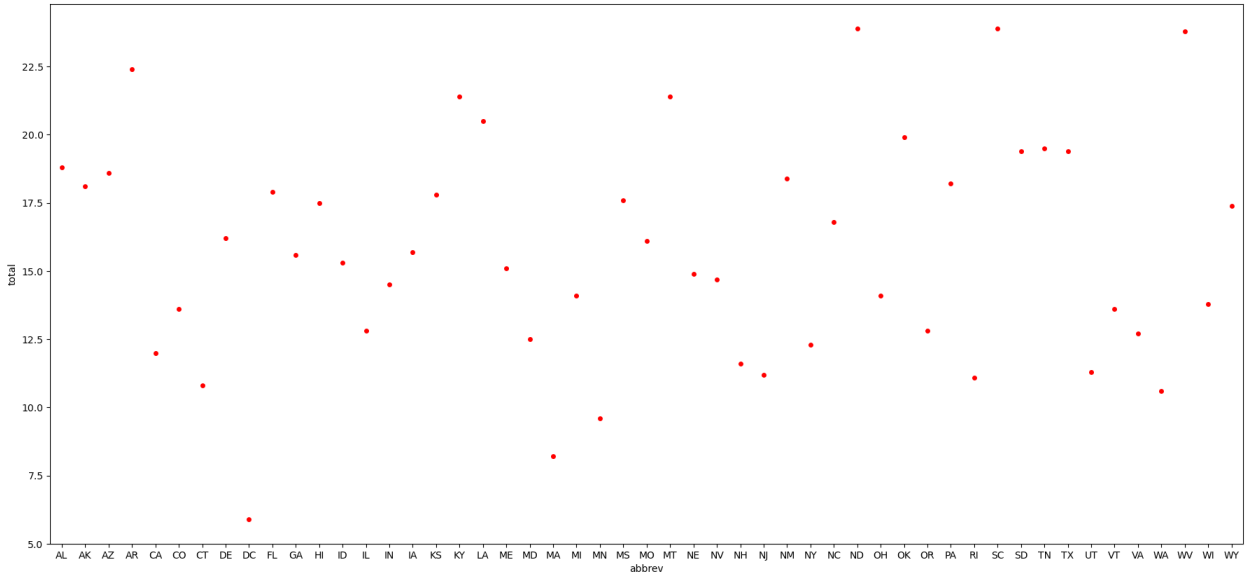


Violin Plot

Inference: You can visualize the distribution of total crashes across different states using a violin plot.

```
plt.figure(figsize=(22,10))
sns.swarmplot(x="abbrev",y="total",data=data,color='red')
```

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

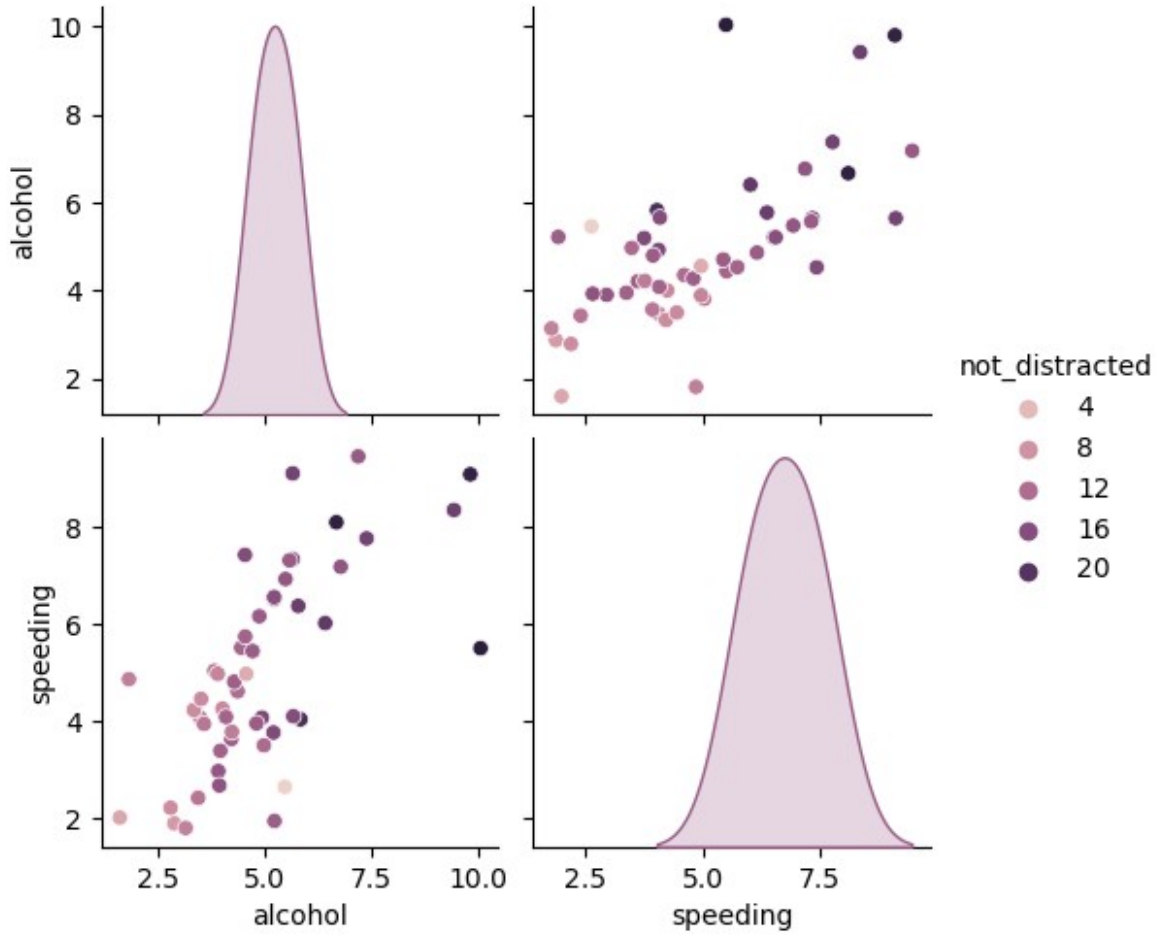


Swarmplot

Inference: A swarm plot shows the distribution of total crashes across different states, with each point representing an individual data point. It can help identify the concentration of data points and potential outliers.

```
sns.pairplot(data, x_vars=["alcohol", "speeding"], y_vars=["alcohol",
"speeding"], hue="not_distracted")
```

```
<seaborn.axisgrid.PairGrid at 0x2806d0b10>
```

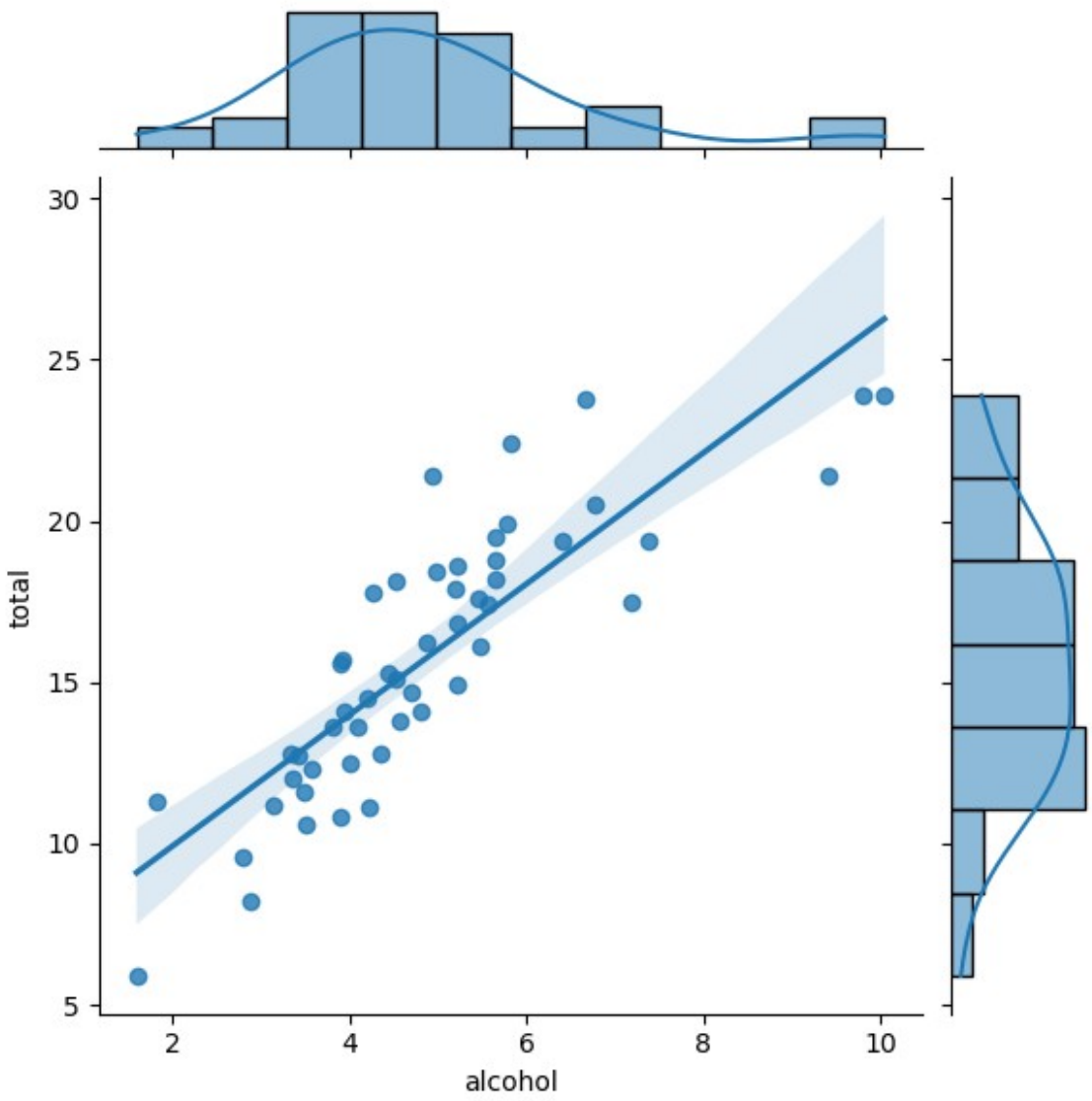


Pairwise Scatter plot with hue

Inference: This pairplot with hue allows you to see how alcohol and speeding relate to each other, with the hue indicating whether the driver was not distracted. It can help identify patterns in distracted vs. not distracted cases.

```
sns.jointplot(x="alcohol", y="total", data=data, kind="reg")
```

```
<seaborn.axisgrid.JointGrid at 0x280864e10>
```



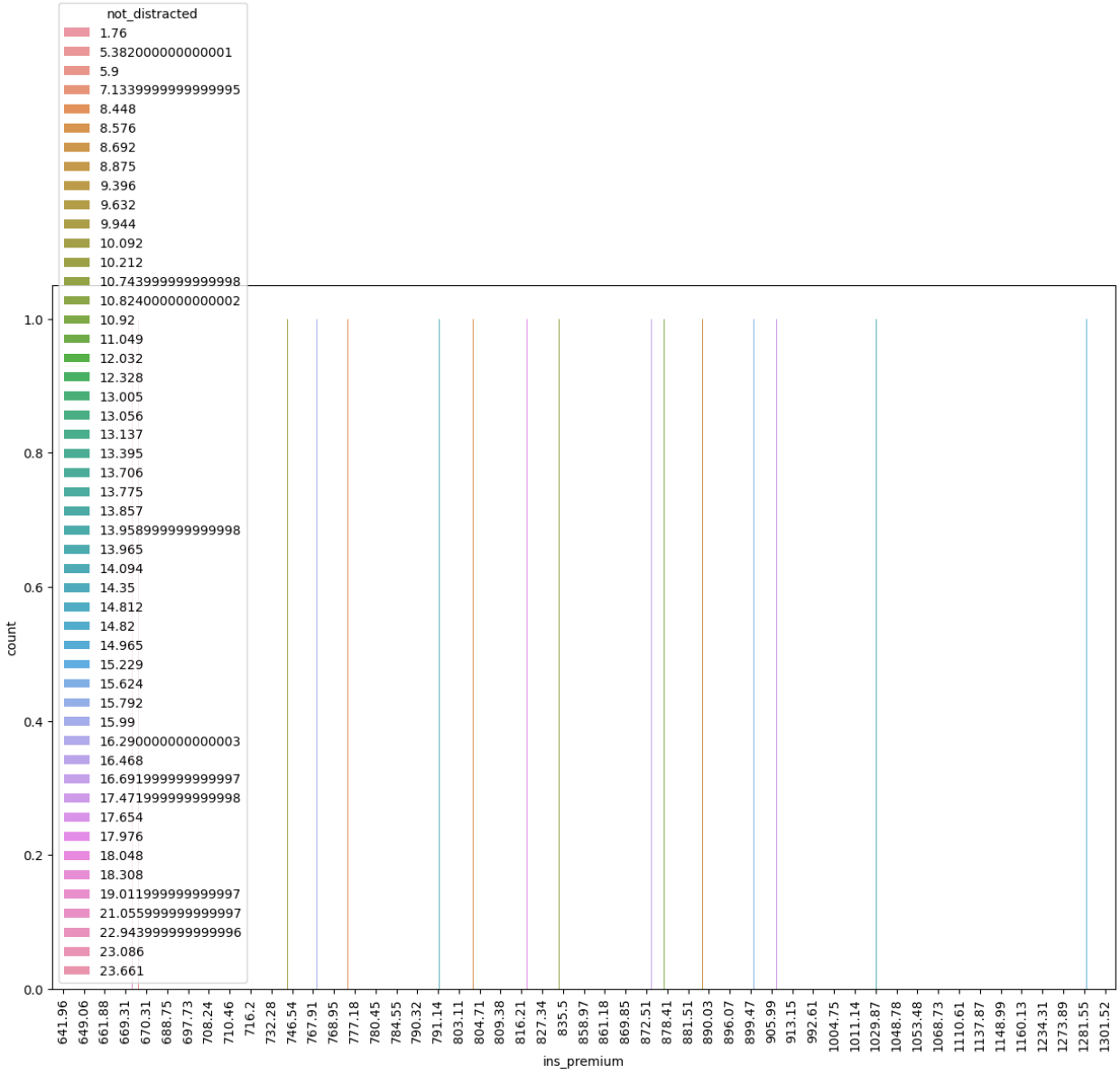
Joint Plot

Inference: The joint plot with a regression line illustrates the relationship between alcohol consumption and total crashes. The regression line provides insights into the linear association between these variables.

```
plt.figure(figsize=(15,10))
sns.countplot(x="ins_premium", hue="not_distracted", data=data)
plt.xticks(rotation=90)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14,
15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48,
49, 50]),
 [Text(0, 0, '641.96'),
  Text(1, 0, '649.06'),
  Text(2, 0, '661.88'),
  Text(3, 0, '669.31'),
  Text(4, 0, '670.31'),
  Text(5, 0, '688.75'),
  Text(6, 0, '697.73'),
  Text(7, 0, '708.24'),
  Text(8, 0, '710.46'),
  Text(9, 0, '716.2'),
  Text(10, 0, '732.28'),
  Text(11, 0, '746.54'),
  Text(12, 0, '767.91'),
  Text(13, 0, '768.95'),
  Text(14, 0, '777.18'),
  Text(15, 0, '780.45'),
  Text(16, 0, '784.55'),
  Text(17, 0, '790.32'),
  Text(18, 0, '791.14'),
  Text(19, 0, '803.11'),
  Text(20, 0, '804.71'),
  Text(21, 0, '809.38'),
  Text(22, 0, '816.21'),
  Text(23, 0, '827.34'),
  Text(24, 0, '835.5'),
  Text(25, 0, '858.97'),
  Text(26, 0, '861.18'),
  Text(27, 0, '869.85'),
  Text(28, 0, '872.51'),
  Text(29, 0, '878.41'),
  Text(30, 0, '881.51'),
  Text(31, 0, '890.03'),
  Text(32, 0, '896.07'),
  Text(33, 0, '899.47'),
  Text(34, 0, '905.99'),
  Text(35, 0, '913.15'),
  Text(36, 0, '992.61'),
  Text(37, 0, '1004.75'),
  Text(38, 0, '1011.14'),
  Text(39, 0, '1029.87'),
  Text(40, 0, '1048.78'),
  Text(41, 0, '1053.48'),
  Text(42, 0, '1068.73'),
  Text(43, 0, '1110.61'),
```

```
  Text(44, 0, '1137.87'),
  Text(45, 0, '1148.99'),
  Text(46, 0, '1160.13'),
  Text(47, 0, '1234.31'),
  Text(48, 0, '1273.89'),
  Text(49, 0, '1281.55'),
  Text(50, 0, '1301.52')])
```

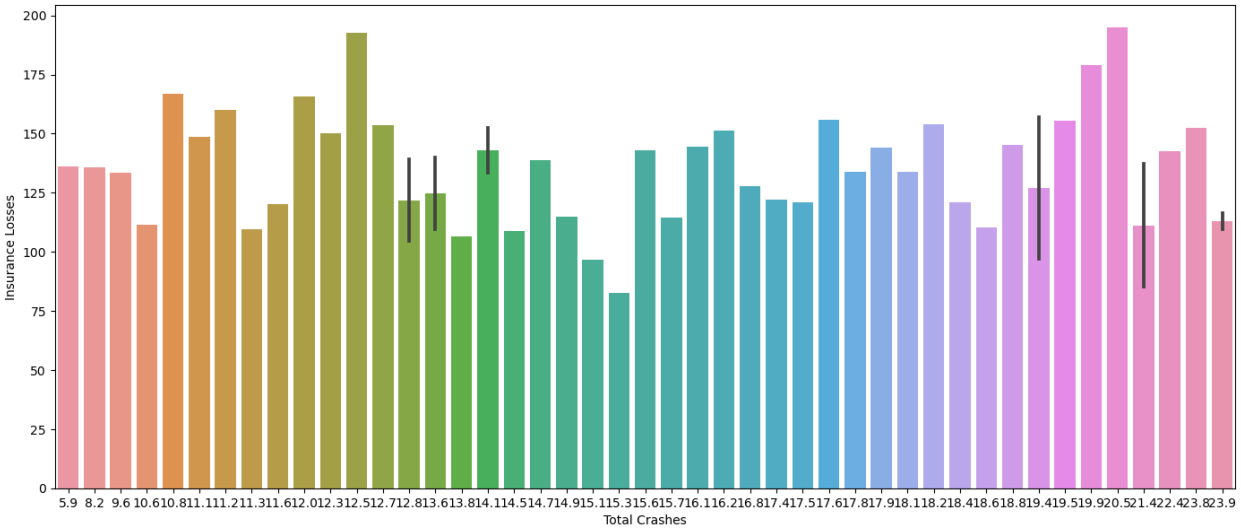


Count plot

Inference: This countplot shows the distribution of insurance premiums with a hue indicating whether the driver was not distracted. It can help identify patterns in distracted vs. not distracted drivers regarding insurance premiums.

```
plt.figure(figsize=(17,7))
sns.barplot(x="total", y="ins_losses", data=data)
plt.xlabel("Total Crashes")
plt.ylabel("Insurance Losses")
```

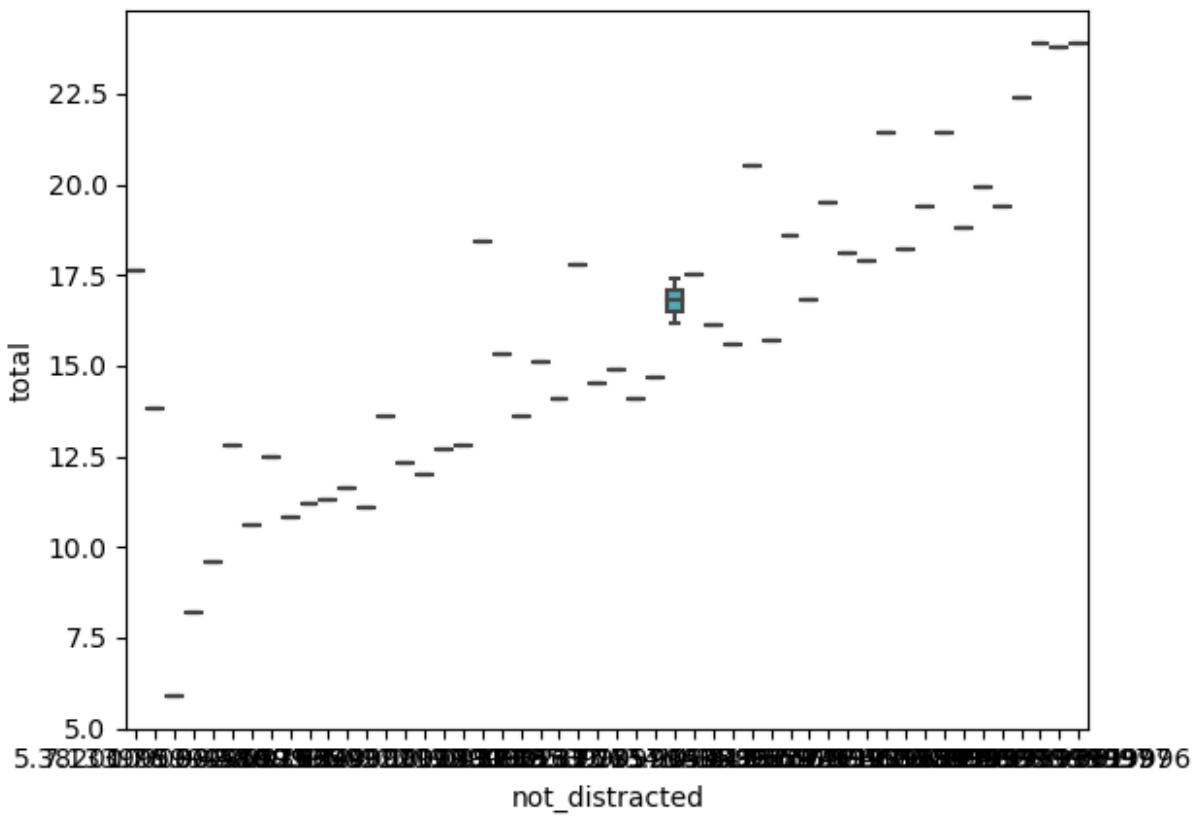
```
Text(0, 0.5, 'Insurance Losses')
```



Inference: This bar plot compares total crashes with insurance losses. It can help determine if higher total crashes are associated with higher insurance losses.

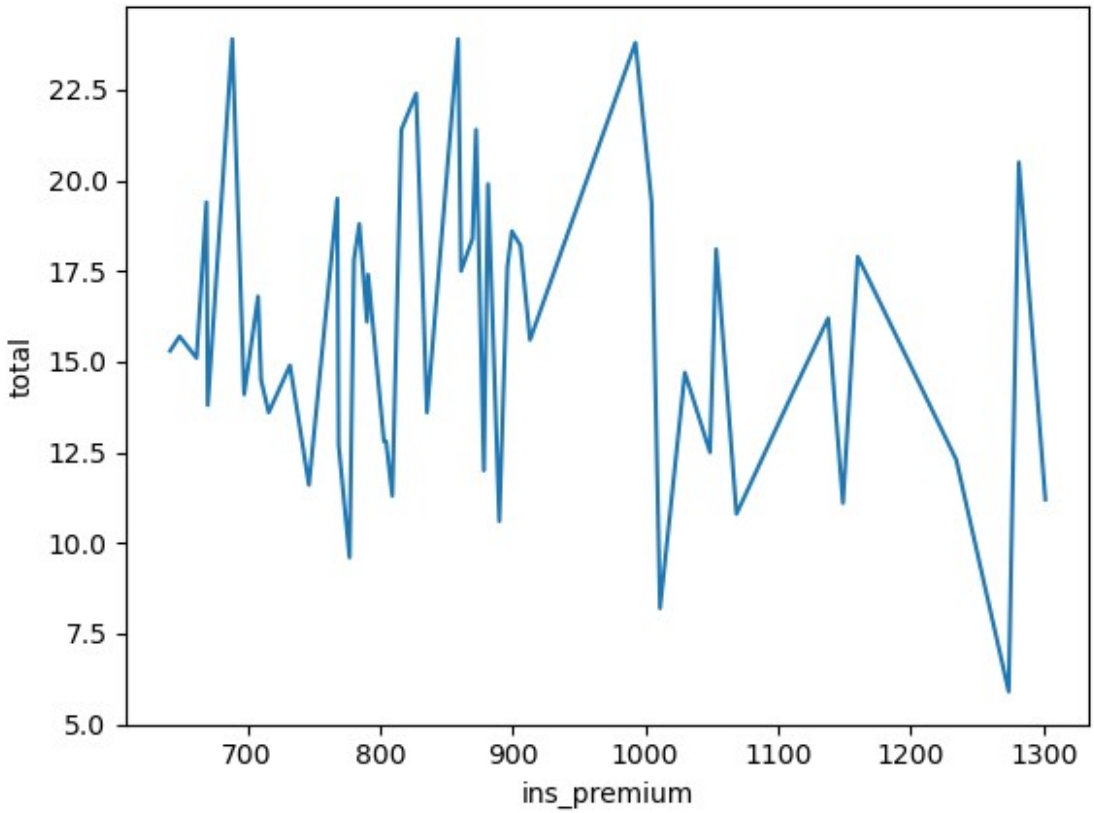
```
sns.boxplot(x="not_distracted", y="total", data=data)
```

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




```
sns.lineplot(x="ins_premium", y="total", data=data)
```

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



Lineplot

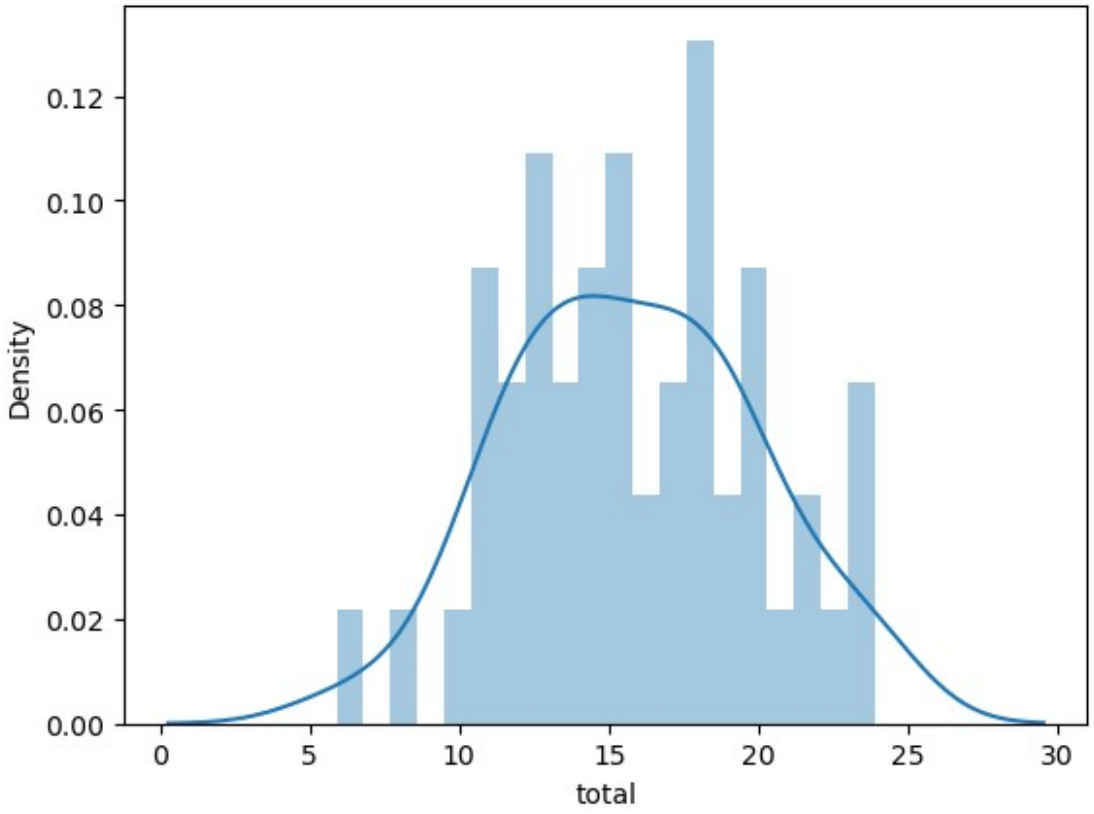
Inference: The line plot suggests a potential positive trend between insurance premiums and the total number of crashes.

```
sns.distplot(data["total"], bins=20, kde=True)
```

```
/var/folders/75/l1625v4n7qnbfp296v2f82ch0000gn/T/
ipykernel_2306/1021224044.py: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(data["total"], bins=20, kde=True)
```

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

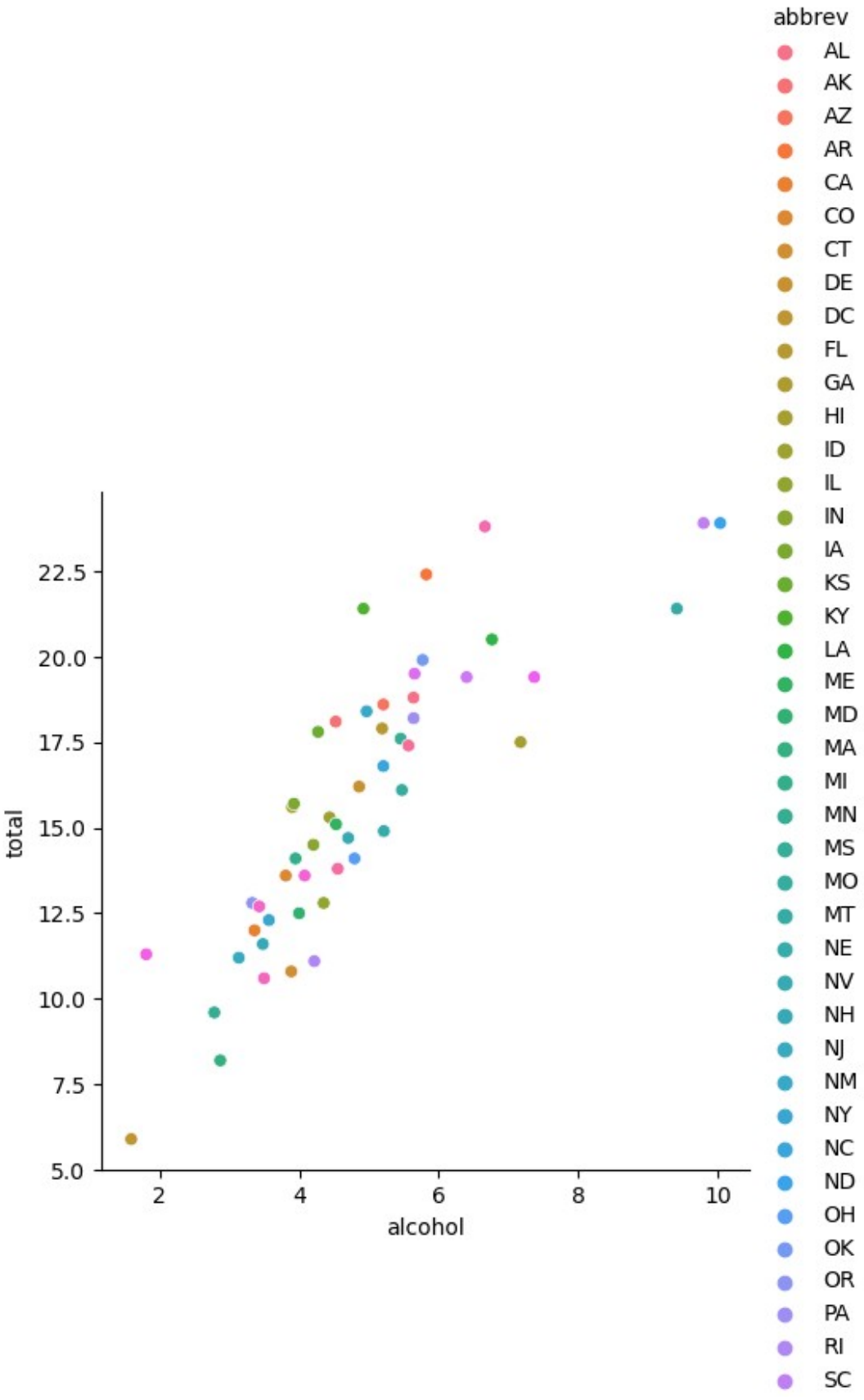


Distplot

Inference: The distribution plot reveals that most states have relatively low crash rates, with a peak around 15-20 crashes. KDE overlay allows you to visualize the smoothed estimate of the data's probability density (KDE).¶

```
sns.relplot(x="alcohol", y="total", hue="abbrev", data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x288b73c50>
```


abbrev
AL
AK
AZ
AR
CA
CO
CT
DE
DC
FL
GA
HI
ID
IL
IN
IA
KS
KY
LA
ME
MD
MA
MI
MN
MS
MO
MT
NE
NV
NH
NJ
NM
NY
NC
ND
OH
OK
OR
PA
RI
SC
total
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2
4
6
8
10
alcohol

Relplot

Inference: The scatterplot shows no clear linear relationship between alcohol consumption and total crashes, but variations exist among states.