

```
import seaborn as sns

print(sns.get_dataset_names())

['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes', 'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri', 'g
```

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

```
df.head()
```

	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

```
df.info()
```

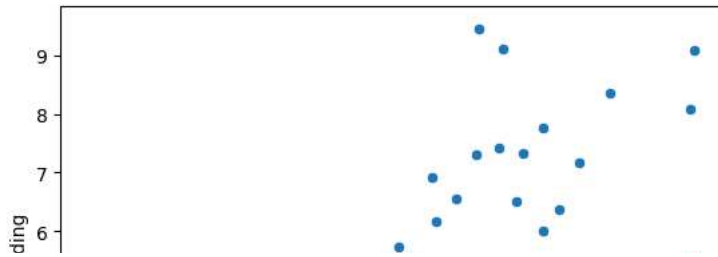
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 8 columns):
#   Column              Non-Null Count  Dtype
---  -
0    total              51 non-null    float64
1    speeding           51 non-null    float64
2    alcohol            51 non-null    float64
3    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
```

```
df.describe()
```

	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

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

<Axes: xlabel='total', ylabel='speeding'>



"""

Inference from above plot:

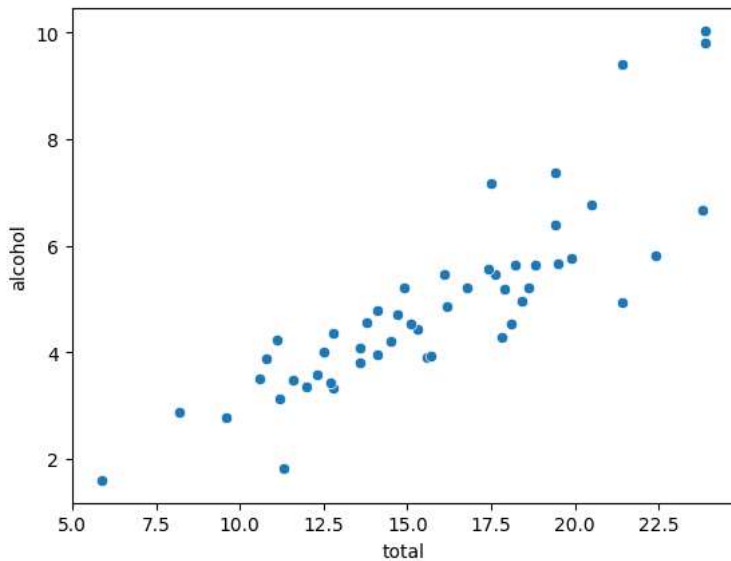
as the value of speeding increases total no of accidents also increases
from the plot we can say, total and speeding are directly proportional

"""

'\nInference from above plot:\nas speeding increases total no of accidents also increases\nfrom the plot we can say, total and speeding are directly proportional\n'

sns.scatterplot(x = "total", y = "alcohol", data = df)

<Axes: xlabel='total', ylabel='alcohol'>



"""

Inference from above plot:

as the value alcohol increases total no of accidents also increases
from the plot we can say, total and alcohol are directly proportional

"""

'\nInference from above plot:\nas alcohol increases total no of accidents also increases\nfrom the plot we can say, total and alcohol are directly proportional\n'

sns.lineplot(x = "total", y = "not_distracted", data = df, ci = None)

```
<ipython-input-12-b4aaa992b3bd>:1: FutureWarning:
```

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
sns.lineplot(x = "total", y = "not_distracted", data = df, ci = None)
<Axes: xlabel='total', ylabel='not_distracted'>
```



```
"""
```

Inference from above plot:

from the plot we can say that there is a uniform increase in total as not_distracted increases and at some points in the graph there are steep falls and rises

```
"""
```

```
sns.distplot(df["ins_losses"])
```

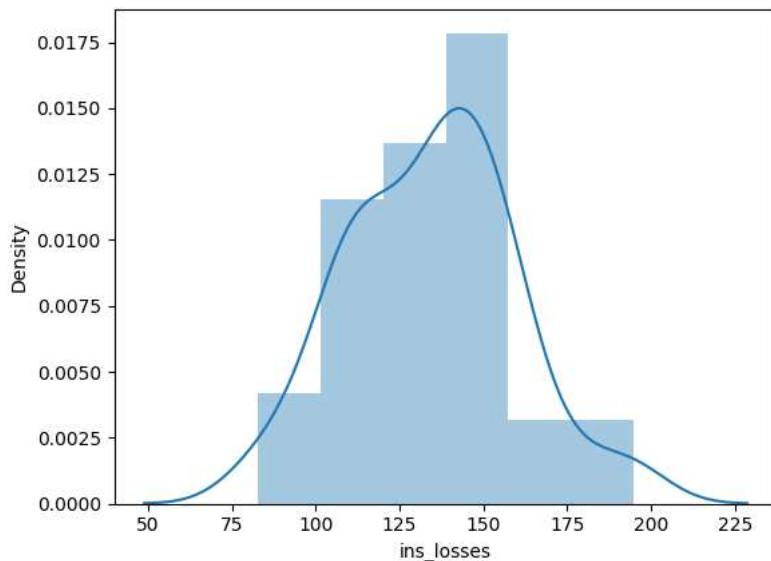
```
<ipython-input-13-46fdb0fb15ea>:1: UserWarning:
```

`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(df["ins_losses"])
<Axes: xlabel='ins_losses', ylabel='Density'>
```



```
"""
```

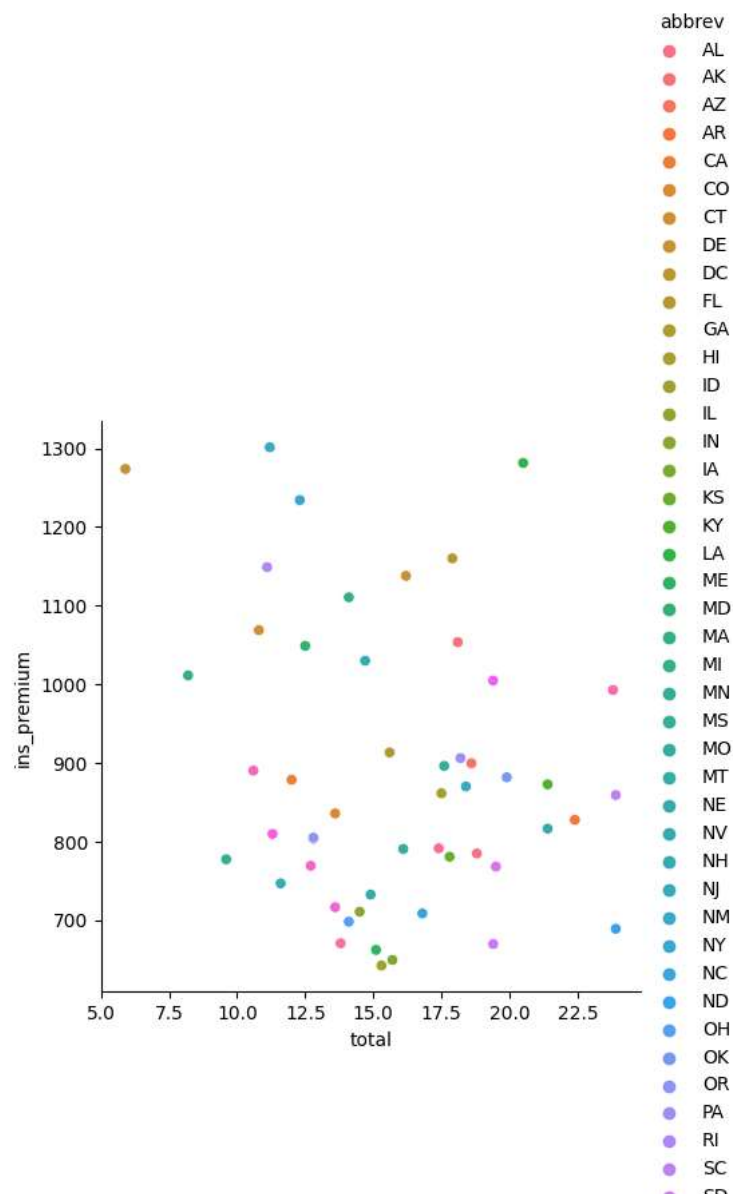
Inference from above plot:

from the plot we can say, that the max values of ins_losses lies between 125 and 150 or the mean of ins_losses lies between 125 and 150

```
"""
```

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

<seaborn.axisgrid.FacetGrid at 0x7b578b3cbc70>



"""

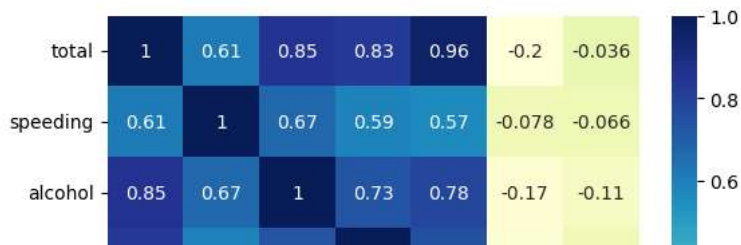
Inference from above plot:

from the plot we can visualise how variables (total and ins_premium) that are within a dataset(car_crash) relate to each other on the basis of

"""

```
corr = df.corr()
from __future__ import annotations
sns.heatmap(corr, annot = True, cmap = "YlGnBu")
```

```
<ipython-input-17-352ffc32d0d8>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a fu
corr = df.corr()
<Axes: >
```



```
"""
```

Inference from above plot:

from the plot we can say that total and no_previous are highly dependent on each other
and total and ins_premium are not at all dependent on each other

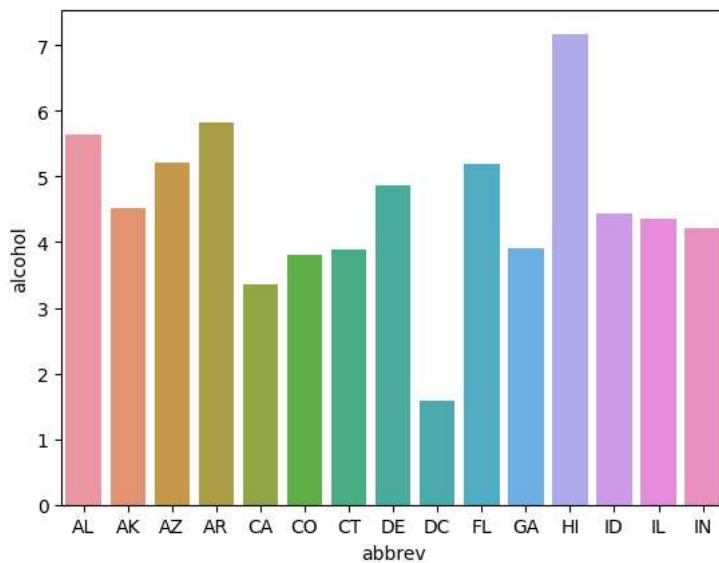
```
"""
```

'\nInference from above plot:\nas alcohol increases total no of accidents also increases\nfrom the plot we can say, total and alcohol are directly proportional\n'

```
ins_losses -0.036 -0.066 -0.11 -0.076 -0.0064 0.62 1
```

```
sns.barplot(data = df.head(15), x = "abbrev", y = "alcohol")
```

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



```
"""
```

Inference from above plot:

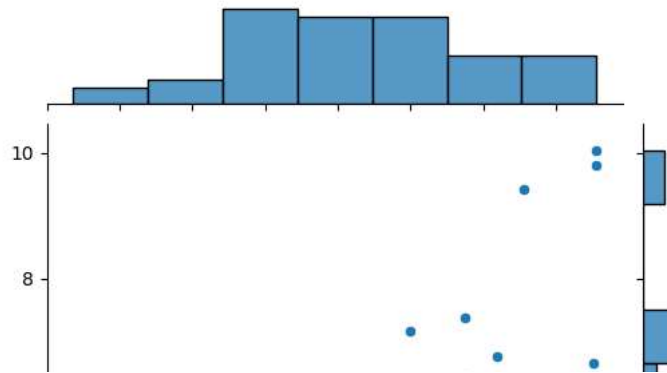
from the plot we can say that from the starting 15 observations HI have the highest no of car crashes due to alcohol

```
"""
```

'\nInference from above plot:\nas alcohol increases total no of accidents also increases\nfrom the plot we can say, total and alcohol are directly proportional\n'

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

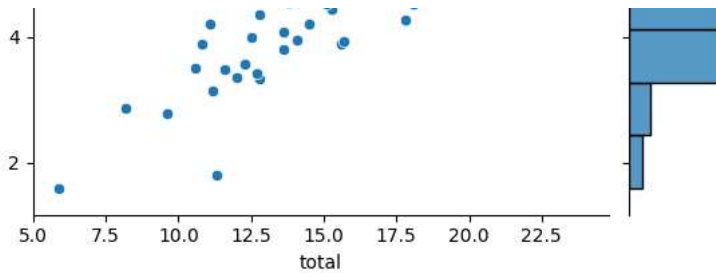
```
<seaborn.axisgrid.JointGrid at 0x7b578538d240>
```



```
"""
```

Inference from above plot:
from the plot we can say, total and alcohol are directly proportional

```
"""
```



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 10:50 PM

● ×