```
#using the seaborn library
#importing the libraries
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

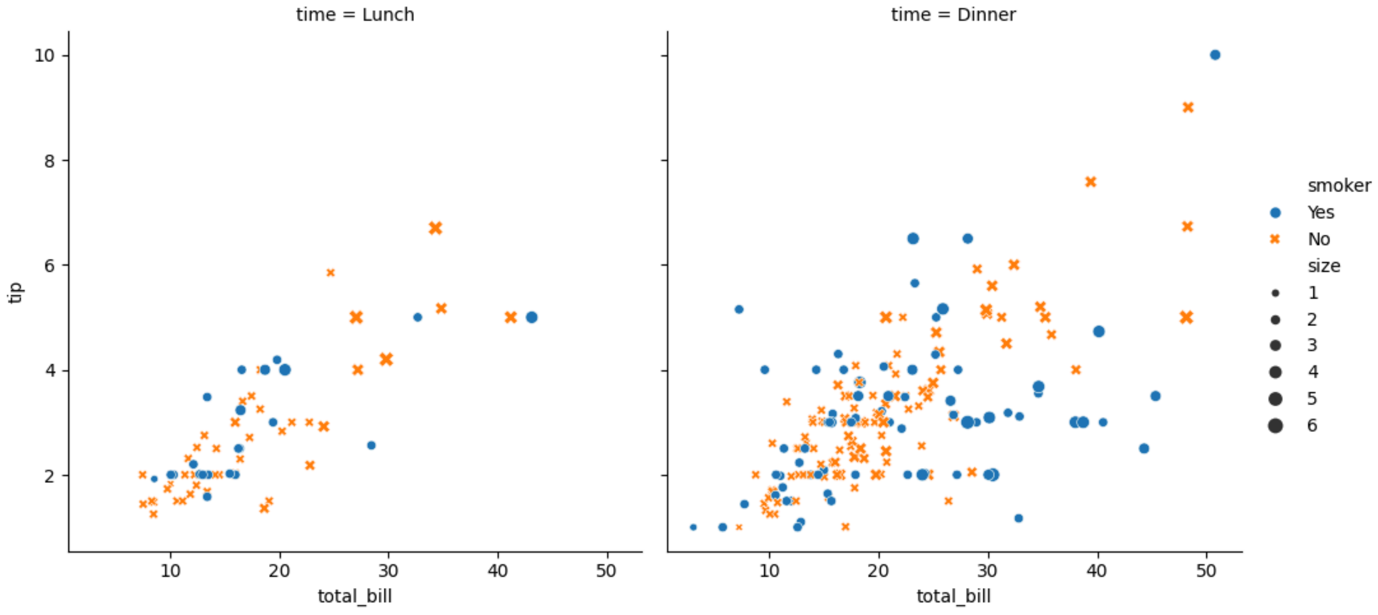
Note: seaborn has some built-in dataset

```
#total bill vs tip dataset
tips = sns.load_dataset('tips')
tips.head()
```

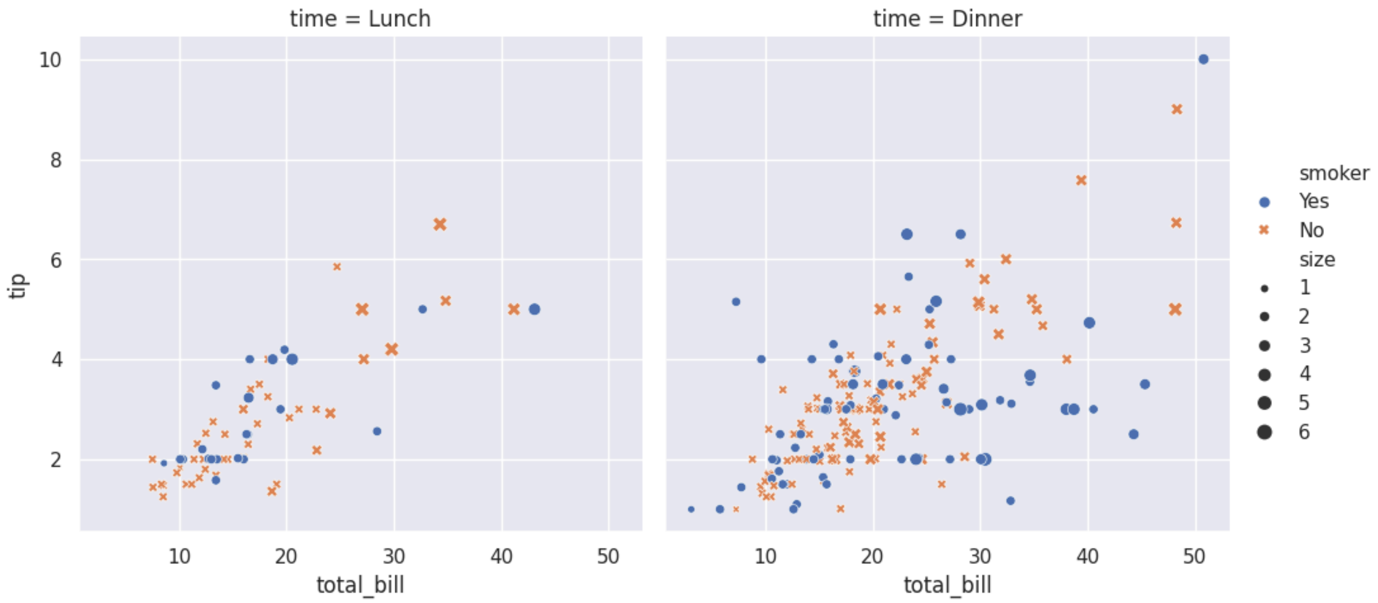
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Next steps: Generate code with `tips` View recommended plots New interactive sheet

```
#visualize the tips dataset
sns.relplot(data=tips,x='total_bill',y='tip',col='time',hue='smoker',style='smoker',size='size')
plt.show()
```



```
#visualize the tips dataset
#setting a theme for the plots
sns.set_theme(style='darkgrid')
sns.relplot(data=tips,x='total_bill',y='tip',col='time',hue='smoker',style='smoker',size='size')
plt.show()
```

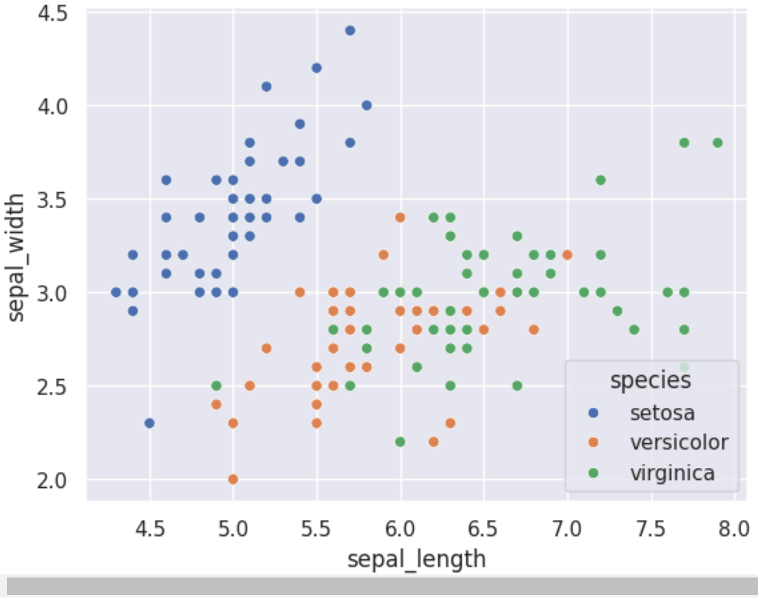


```
#load the iris dataset
iris=sns.load_dataset('iris')
iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Next steps: Generate code with `iris` View recommended plots New interactive sheet

```
sns.scatterplot(data=iris,x='sepal_length',y='sepal_width',hue='species')
plt.show()
```



```
#loading the titanic dataset
titanic=sns.load_dataset('titanic')
```

```
titanic.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

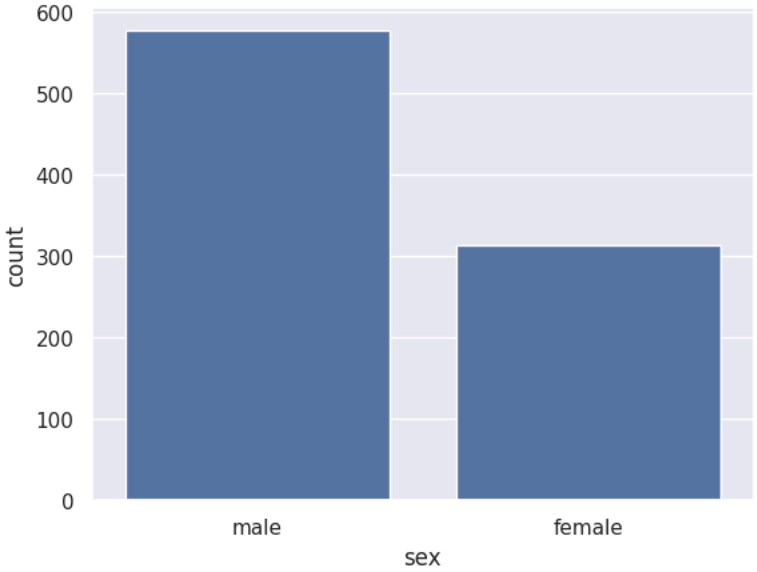
Next steps: Generate code with titanic View recommended plots New interactive sheet

```
titanic.shape
```

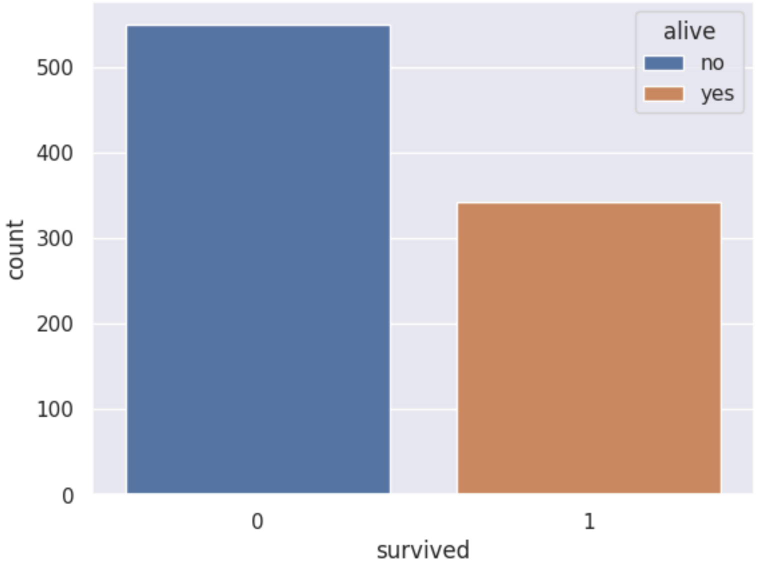
```
(891, 15)
```

Count plot

```
sns.countplot(data=titanic,x='sex')
plt.show()
```

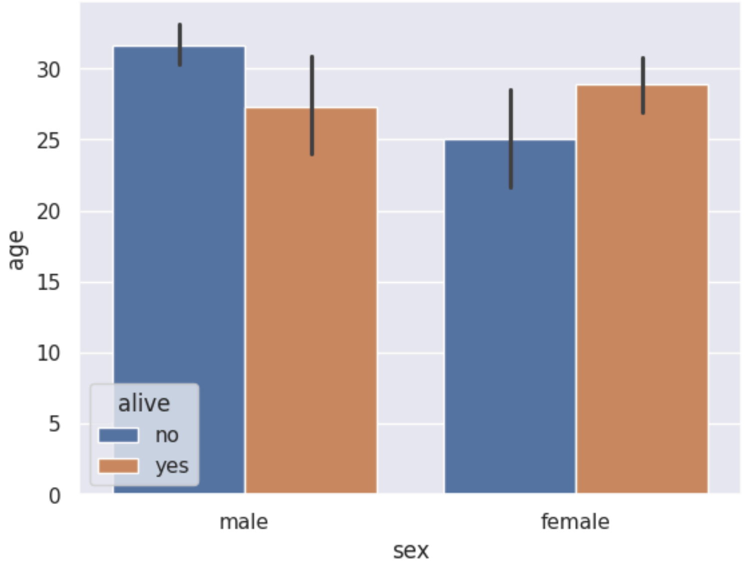


```
sns.countplot(x='survived',data=titanic,hue='alive')
plt.show()
```



Bar chart

```
sns.barplot(data=titanic,x='sex',y='age',hue='alive')
plt.show()
```



```
#house price dataset
from sklearn.datasets import fetch_california_housing
import pandas as pd
house_california=fetch_california_housing()
df=pd.DataFrame(house_california.data,columns=house_california.feature_names)
df['price']=house_california.target
print(house_california)
```

```
{'data': array([[   8.3252    ,   41.        ,    6.98412698, ...,    2.55555556,
          37.88      , -122.23      ],
       [   8.3014    ,   21.        ,    6.23813708, ...,    2.10984183,
          37.86      , -122.22      ],
       [   7.2574    ,   52.        ,    8.28813559, ...,    2.80225989,
          37.85      , -122.24      ],
       ...,
       [   1.7       ,   17.        ,    5.20554273, ...,    2.3256351 ,
          39.43      , -121.22      ],
       [   1.8672    ,   18.        ,    5.32951289, ...,    2.12320917,
          39.43      , -121.32      ],
       [   2.3886    ,   16.        ,    5.25471698, ...,    2.61698113,
          39.37      , -121.24      ]]), 'target': array([4.526, 3.585, 3.521, ..., 0.923, 0.847, 0.894]), 'frame': None, 'target_names'
```

```
df.head()
```

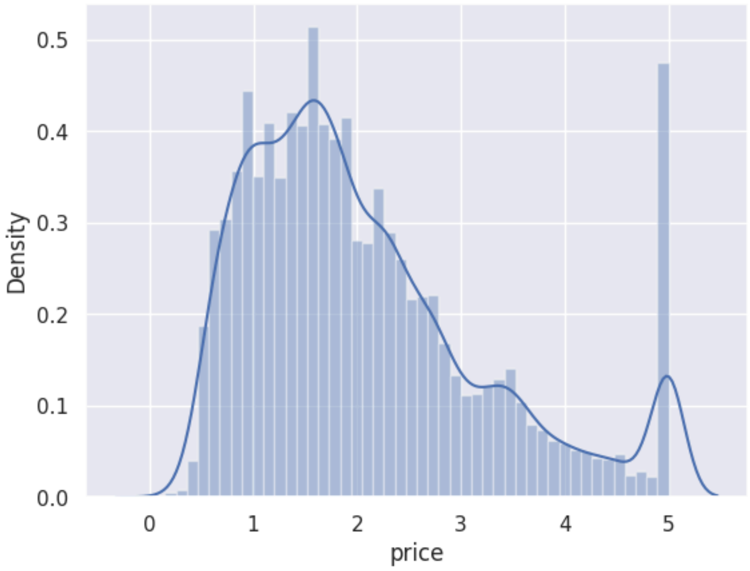
	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	price
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422

Next steps: Generate code with df | View recommended plots | New interactive sheet

Distribution plot

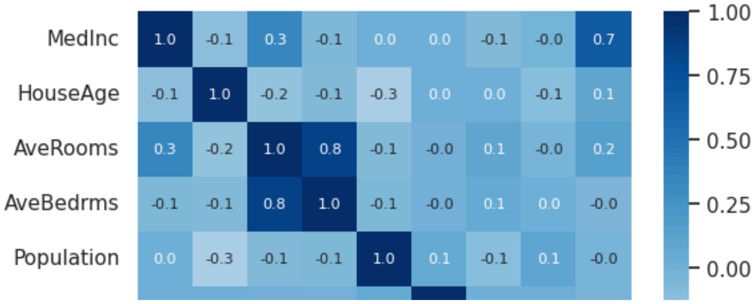
```
sns.distplot(df['price'])
plt.show()
```

```
<ipython-input-18-532aeea001af>: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['price'])
```



correlation: 1.+ve correlation 2.-ve correlation

```
correlation=df.corr()
#constructing a heat map
sns.heatmap(correlation,square=True,fmt='.1f',annot=True,annot_kws={'size':8},cmap='Blues')
plt.show()
#dark color says +vely correlated
```


MedInc
HouseAge
AveRooms
AveBedrms
Population
1.0 -0.1 0.3 -0.1 0.0 0.0 -0.1 -0.0 0.7
-0.1 1.0 -0.2 -0.1 -0.3 0.0 0.0 -0.1 0.1
0.3 -0.2 1.0 0.8 -0.1 -0.0 0.1 -0.0 0.2
-0.1 -0.1 0.8 1.0 -0.1 -0.0 0.1 0.0 -0.0
0.0 -0.3 -0.1 -0.1 1.0 0.1 -0.1 0.1 -0.0
1.00
0.75
0.50
0.25
0.00