```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data= sns.load_dataset('iris')
data.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 data View recommended plots New interactive sheet

```
data['species'],categories=pd.factorize(data['species'])
data.head()
```

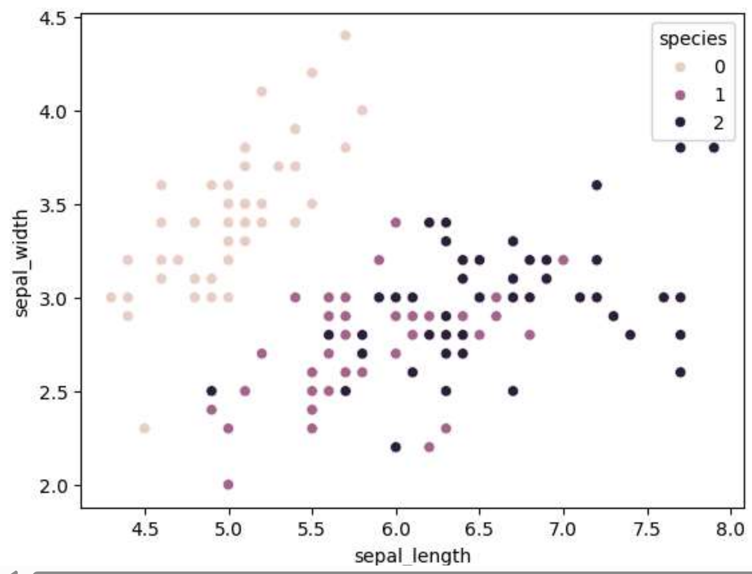
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

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

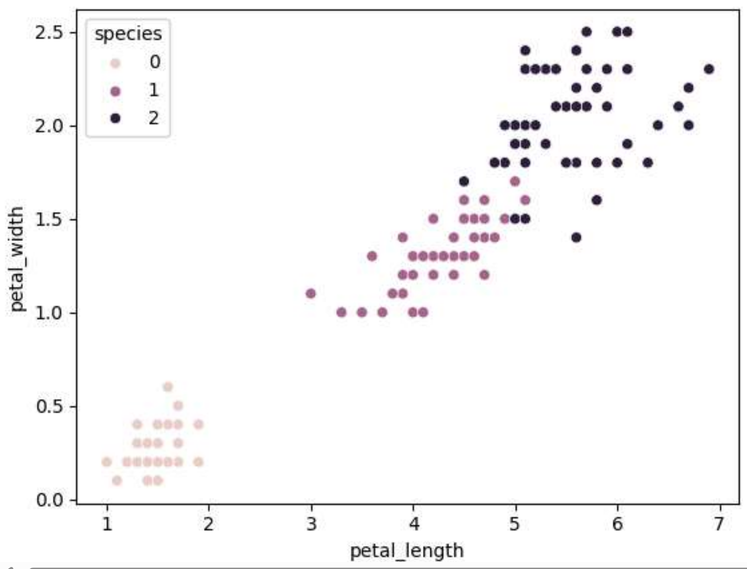
```
data.isna().sum()
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

```
sns.scatterplot(data=data,x="sepal_length",y="sepal_width",hue="species");
```



```
sns.scatterplot(data=data,x="petal_length",y="petal_width",hue="species");
```



```
k_rng=range(1,10)
sse=[]
for k in k_rng:
  km =KMeans(n_clusters=k)
  km.fit(data[['petal_length','petal_width']])
  sse.append(km.inertia_)
```

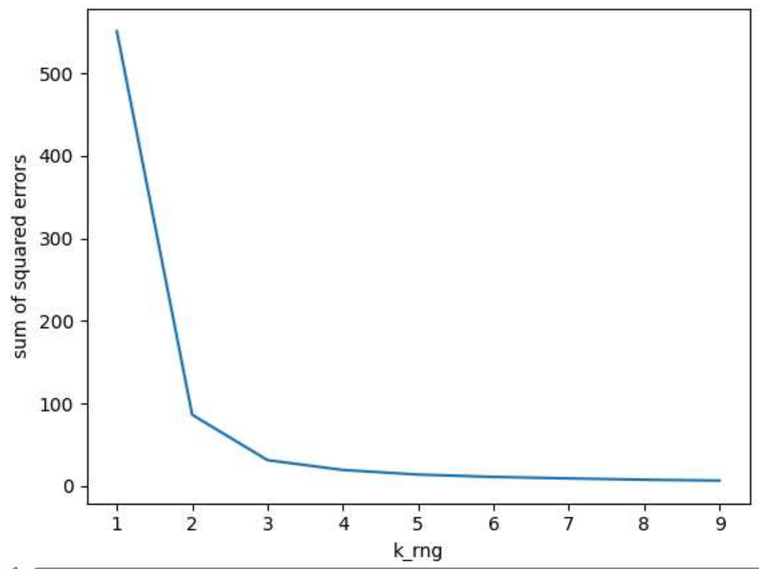
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
```

```
sse
```

```
[550.8953333333333,
 86.39021984551391,
 31.371358974358966,
 19.46598901098901,
 13.91690875790876,
 11.036333877751735,
 9.242108730158728,
 7.6723624030431825,
 6.576538396386222]
```

```
plt.xlabel('k_rng')
plt.ylabel("sum of squared errors")
plt.plot(k_rng,sse)
```

```
[<matplotlib.lines.Line2D at 0x7fac4ded4880>]
```



```
km = KMeans(n_clusters=3,random_state=0,)
y_predicted = km.fit_predict(data[['petal_length','petal_width']])
y_predicted
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will change from 1
  super()._check_params_vs_input(X, default_n_init=10)
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2], dtype=int32)
```

```
data['cluster']=y_predicted
data.head(150)
```

	sepal_length	sepal_width	petal_length	petal_width	species	cluster
0	5.1	3.5	1.4	0.2	0	1
1	4.9	3.0	1.4	0.2	0	1
2	4.7	3.2	1.3	0.2	0	1
3	4.6	3.1	1.5	0.2	0	1
4	5.0	3.6	1.4	0.2	0	1
...	...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2	2
146	6.3	2.5	5.0	1.9	2	2
147	6.5	3.0	5.2	2.0	2	2
148	6.2	3.4	5.4	2.3	2	2
149	5.9	3.0	5.1	1.8	2	2

150 rows × 6 columns

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

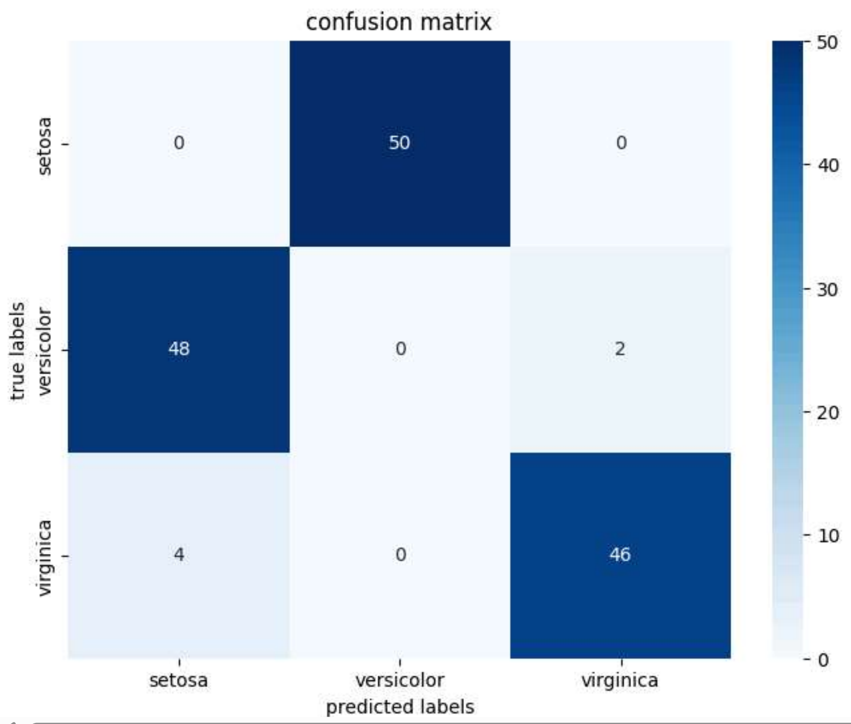
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(data.species,data.cluster)
cm
```

```
array([[ 0, 50,  0],
       [48,  0,  2],
       [ 4,  0, 46]])
```

```
true_labels=data.species
predicted_labels=data.cluster


cm = confusion_matrix(true_labels,predicted_labels)
class_labels=['setosa','versicolor','virginica']


plt.figure(figsize=(8,6))
sns.heatmap(cm,annot=True,fmt='d',cmap='Blues',xticklabels=class_labels,yticklabels=class_labels)
plt.title('confusion matrix')
plt.xlabel('predicted labels')
plt.ylabel('true labels')
plt.show()
```


confusion matrix
true labels
predicted labels
setosa
versicolor
virginica