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

```
columns = ['Sepal length', 'Sepal width' , 'petal length', 'petal width', 'class labels']
```

Double-click (or enter) to edit

```
df = sns.load_dataset('iris')
df.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

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

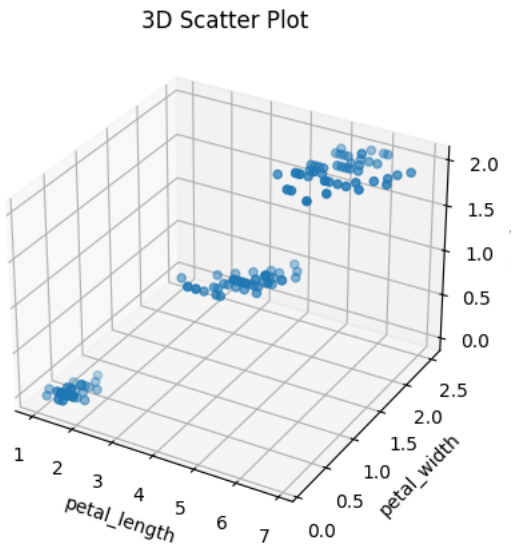
```
df.describe
```

```
pandas.core.generic.NDFrame.describe
def describe(percentiles=None, include=None, exclude=None) -> NDFrameT

Generate descriptive statistics.

Descriptive statistics include those that summarize the central
tendency, dispersion and shape of a
dataset's distribution, excluding ``NaN`` values.
```

```
df.isna().sum()
```

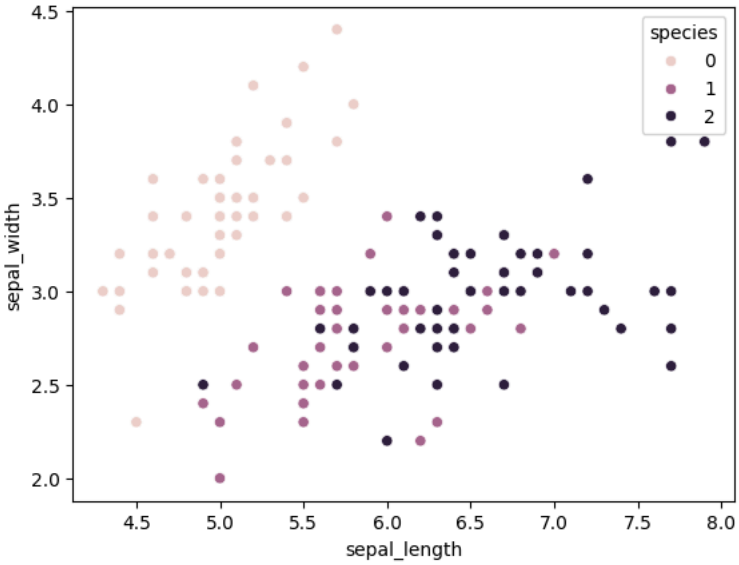
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df.petal_length, df.petal_width, df.species)
ax.set_xlabel('petal_length')
ax.set_ylabel('petal_width')
ax.set_zlabel('species')
plt.title('3D Scatter Plot')
plt.show()
```



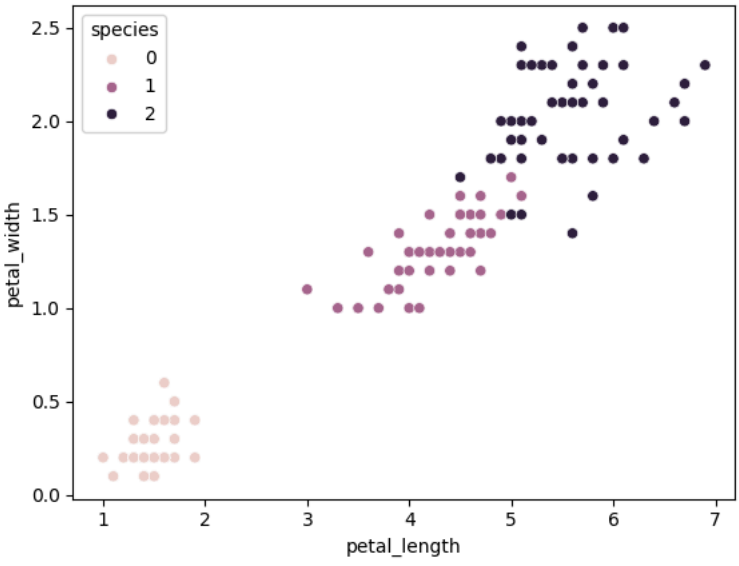

```
sns.scatterplot(data=df, x='sepal_length', y='sepal_width', hue='species')
```

```
<Axes: xlabel='sepal_length', ylabel='sepal_width'>
```




```
sns.scatterplot(data=df, x='petal_length', y='petal_width', hue='species')
```

```
<Axes: xlabel='petal_length', ylabel='petal_width'>
```



```
k_rng = range(1,10)
sse=[]

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

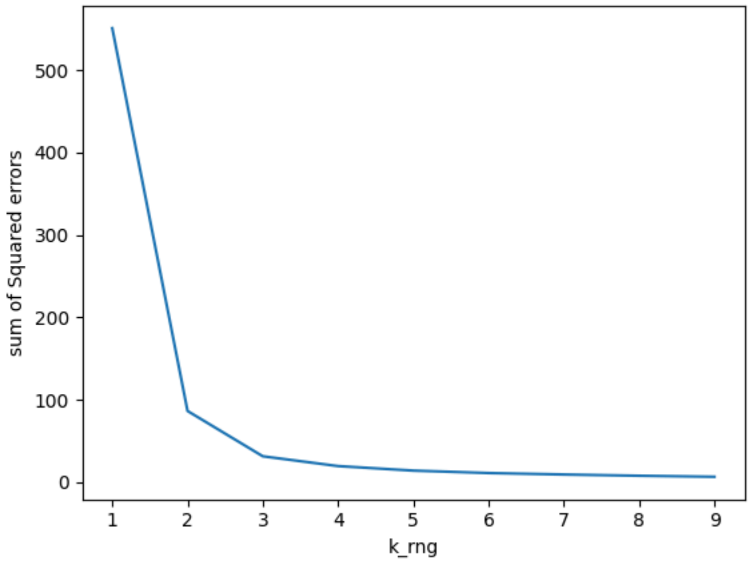
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will chang
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
```

```
sse
```

```
[550.8953333333334,
 86.39021984551397,
 31.37135897435897,
 19.465989010989013,
 13.983213141025638,
 11.051428739411808,
 9.24493855178638,
 7.7274044011544,
 6.514232636644401]
```

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

```
[<matplotlib.lines.Line2D at 0x7d6076446530>]
```




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

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
array([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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int32)
```

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

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

150 rows × 6 columns

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(df['species'],df['cluster'])
cm
```

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

```
true_labels = df['species']
predicted_labels = df['cluster']
cm = confusion_matrix(true_labels, predicted_labels)
class_labels = ['Setosa', 'versicolor','virginica']

plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar
Tick_marks = np.arange(len(class_labels))
plt.xticks(Tick_marks, class_labels, rotation=45)
plt.yticks(Tick_marks, class_labels)
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
```

```
Text(0, 0.5, 'True Labels')
```