ASSIGNMENT 10

Data Visualization III

Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., https://archive.ics.uci.edu/ml/datasets/Iris). Scan the dataset and give the inference as:

1. List down the features and their types (e.g., numeric, nominal) available in the dataset.
2. Create a histogram for each feature in the dataset to illustrate the feature distributions.
3. Create a boxplot for each feature in the dataset.
4. Compare distributions and identify outliers.

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn.datasets import load_iris
import warnings
warnings.filterwarnings("ignore")
```

In [5]:
```
data = load_iris()
```

In [6]:
```
df = pd.DataFrame()
df[data['feature_names']] = data['data']
df['label'] = data['target']
```

In [7]:
```
df.head()
```

Out[7]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
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

In [8]:
```
df.shape
```

Out[8]:
```
(150, 5)
```

In [9]:
```
df.info()
```

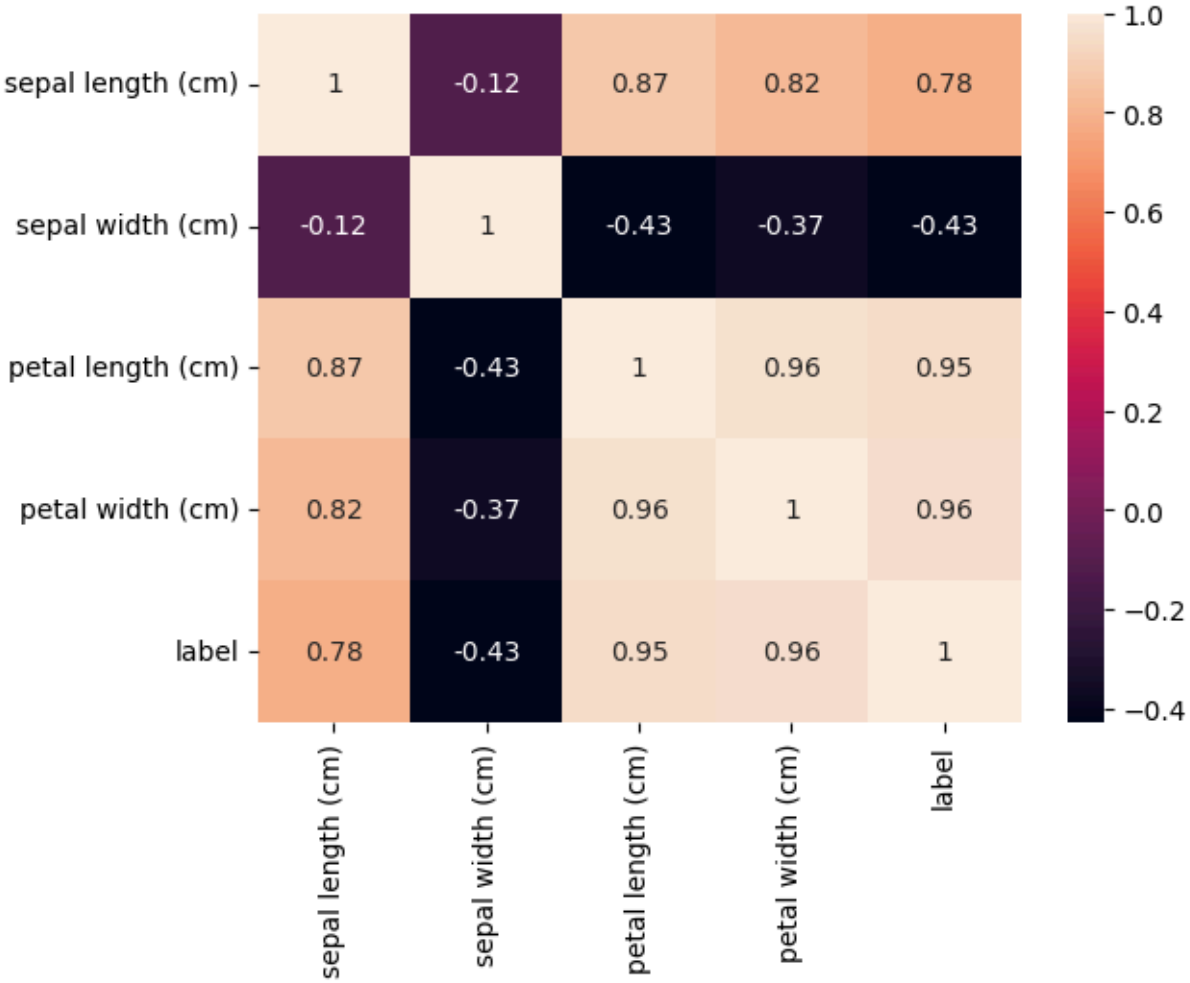
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   label              150 non-null    int64  
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

In [10]:
```
df.describe()
```

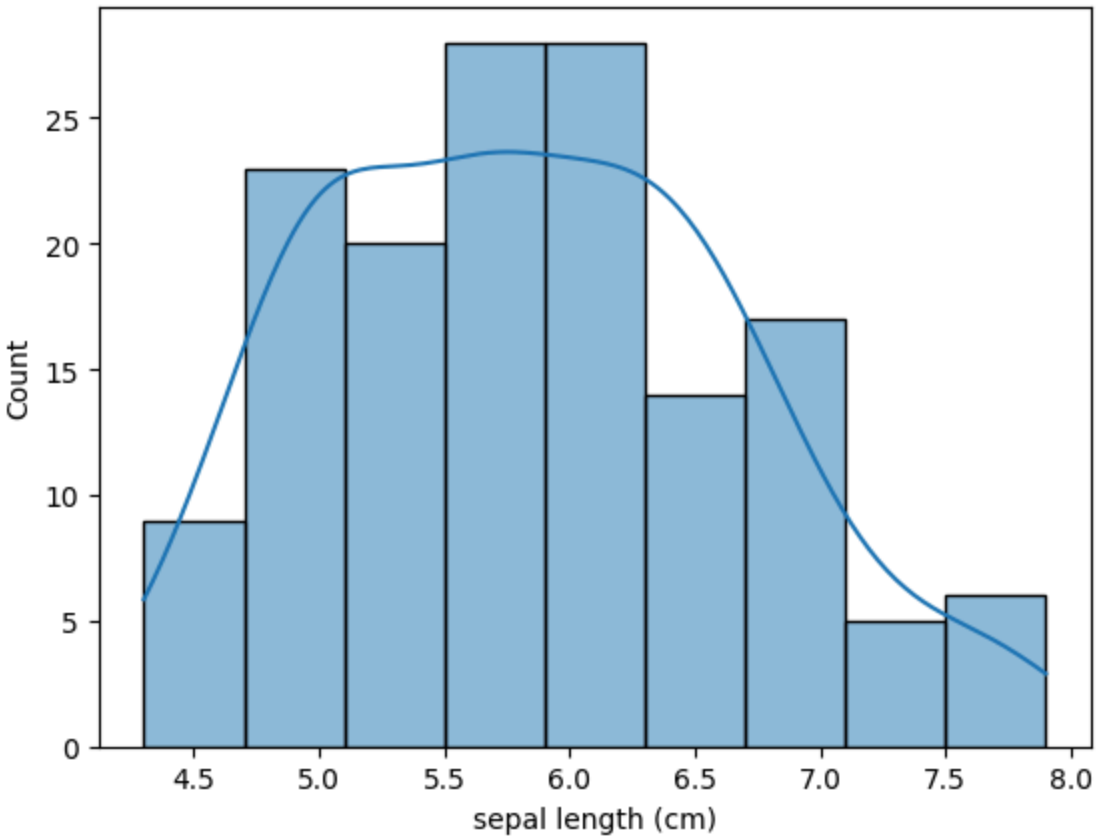
Out[10]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

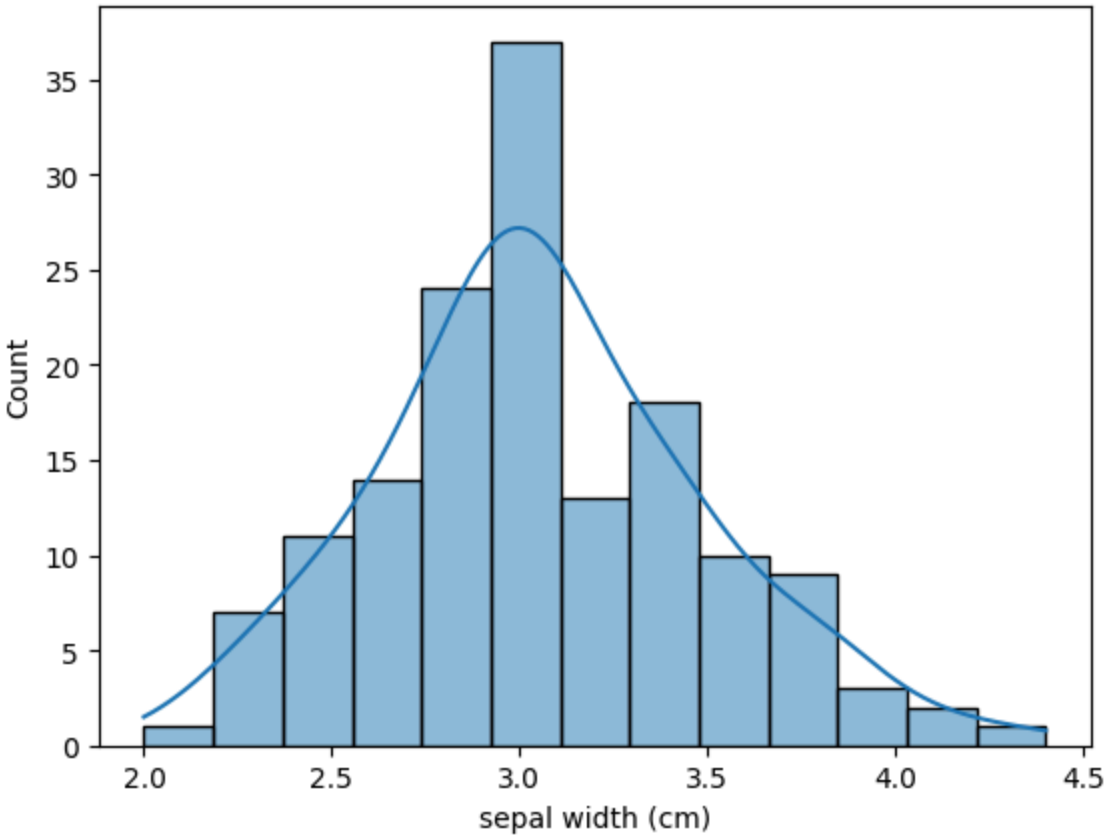
In [11]:
```
sns.heatmap(df.corr(), annot=True)
plt.show()
```



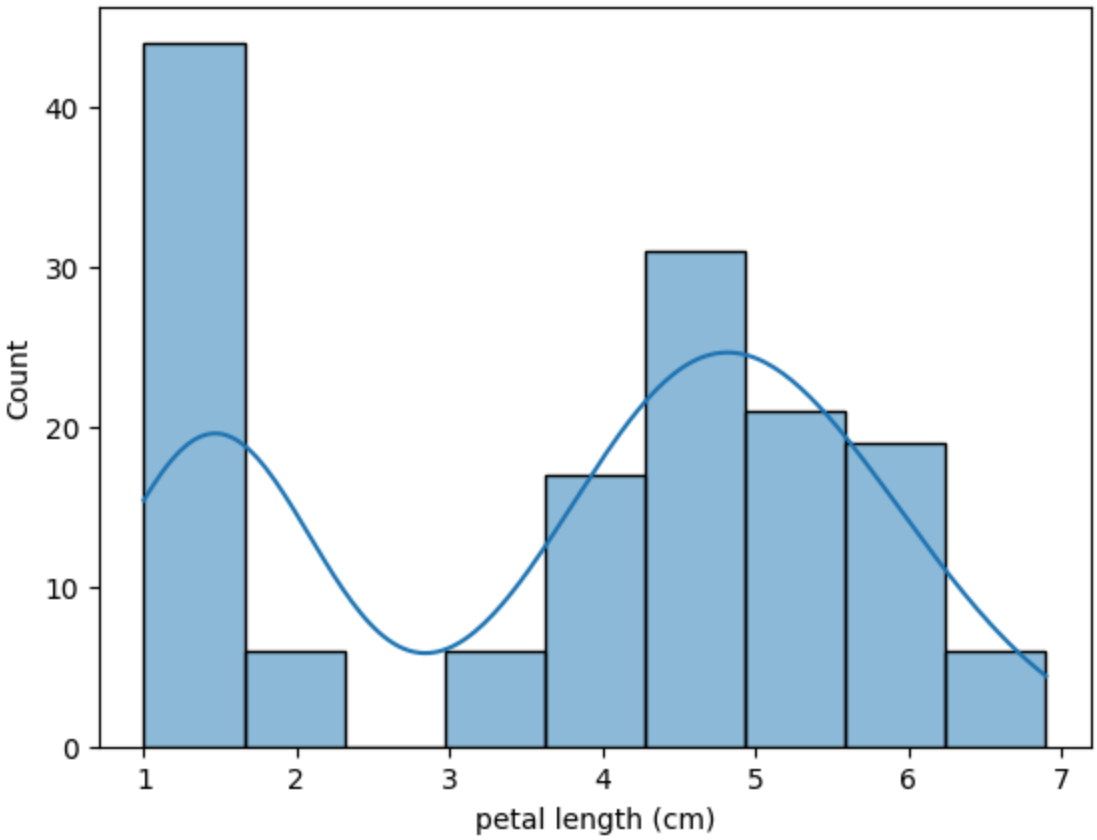
In [12]:
```
sns.histplot(df["sepal length (cm)"], kde=True)
plt.show()
```



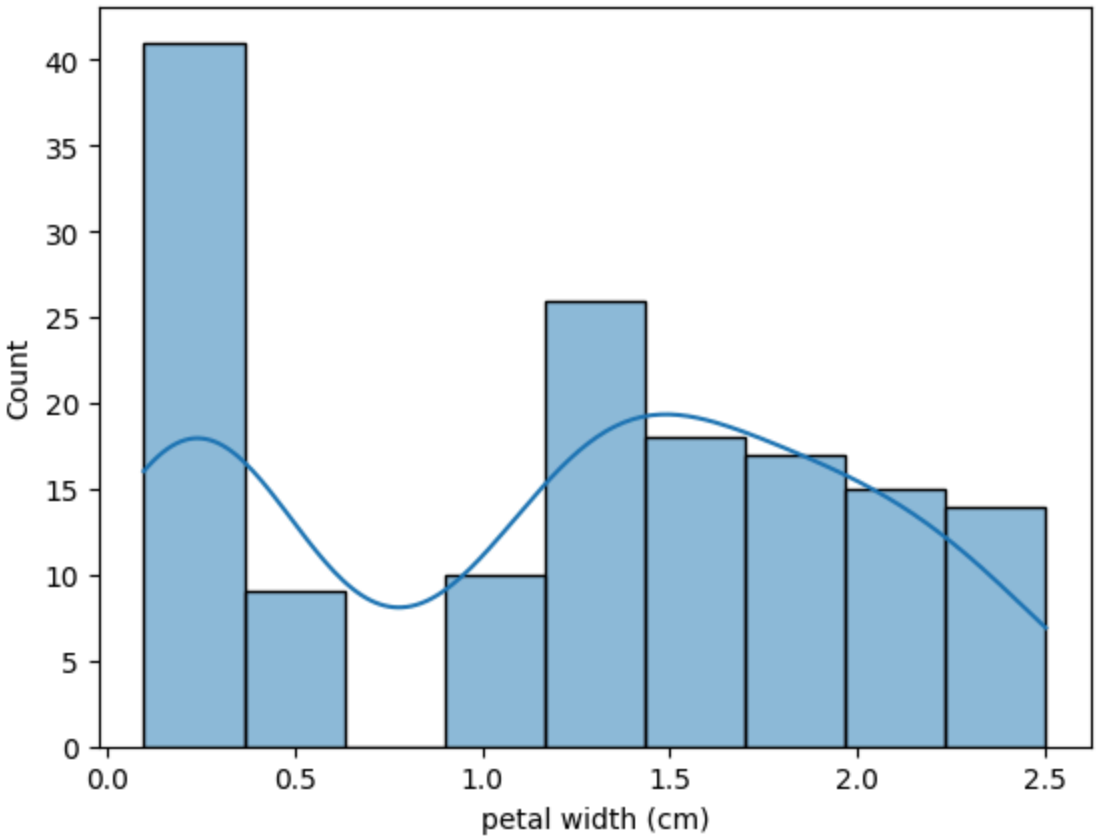
In [13]:
```
sns.histplot(df["sepal width (cm)"], kde=True)
plt.show()
```



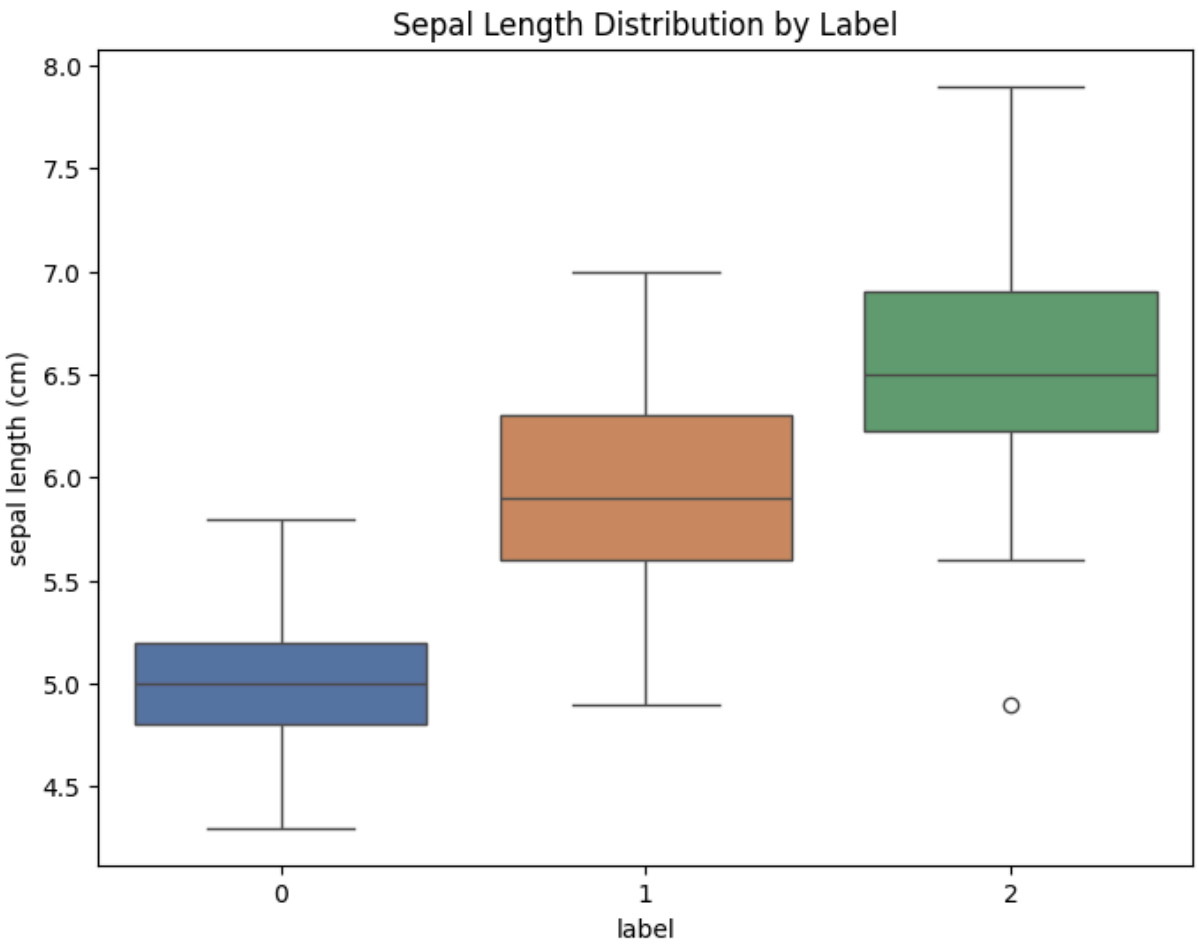
In [14]:
```
sns.histplot(df["petal length (cm)"], kde=True)
plt.show()
```



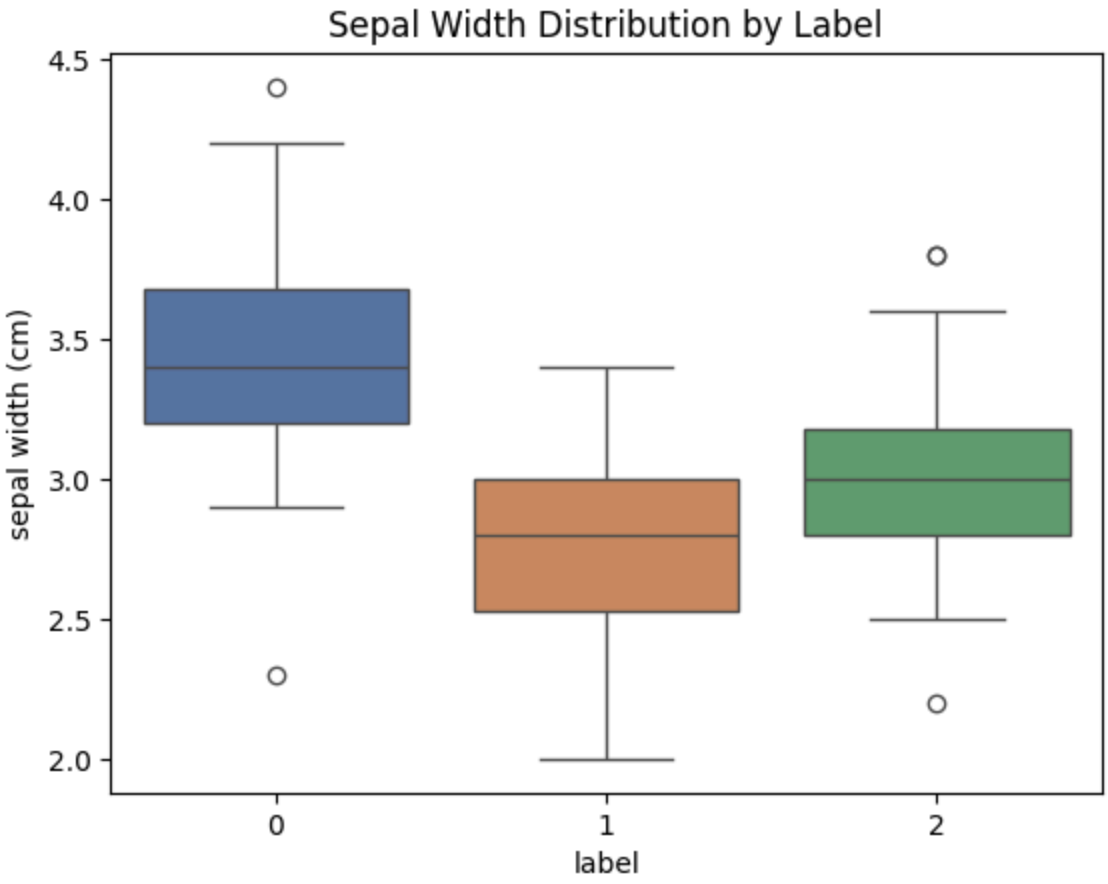
In [15]:
```
sns.histplot(df["petal width (cm)"], kde=True)
plt.show()
```



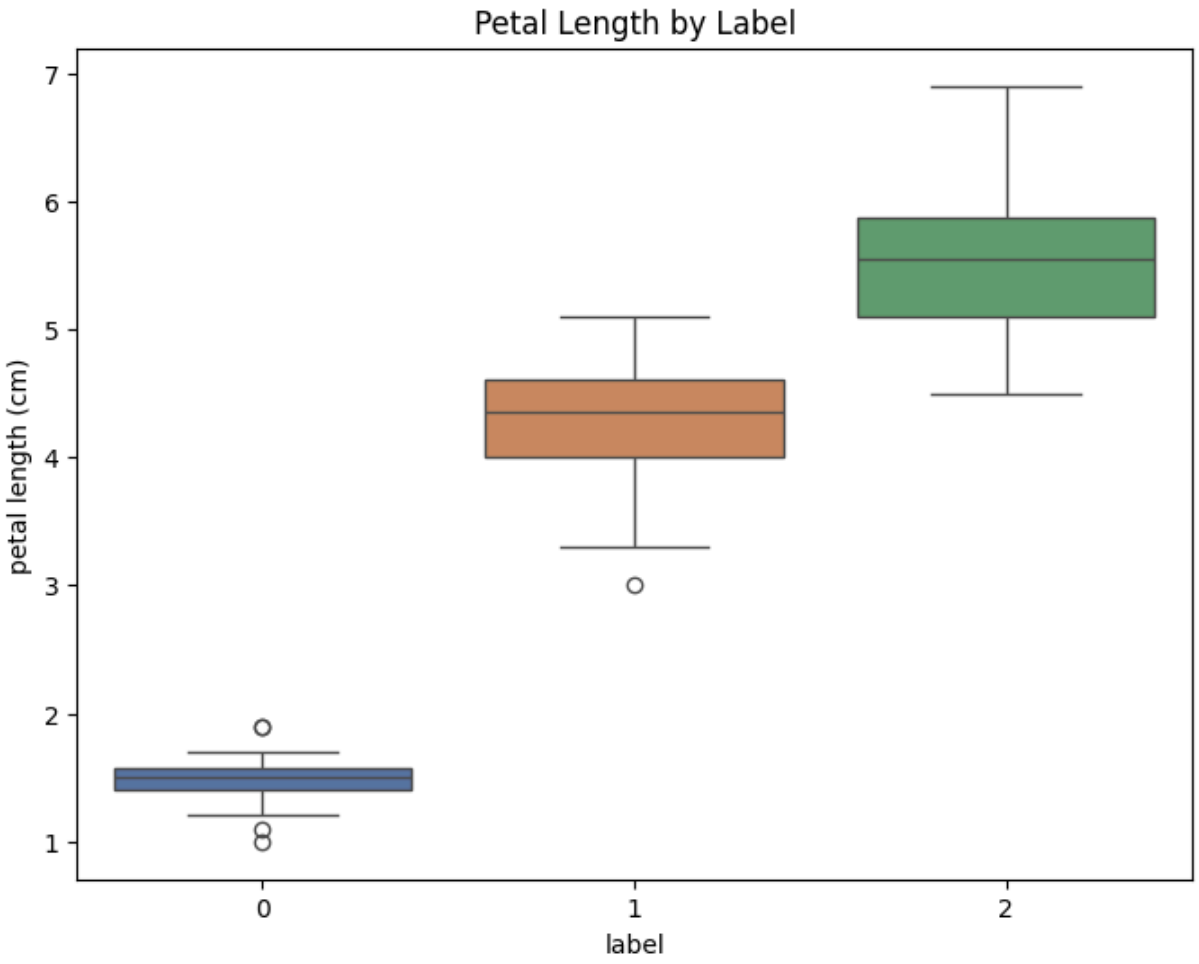
In [22]:
```
plt.figure(figsize=(8, 6))
sns.boxplot(x='label', y='sepal length (cm)', data=df, palette='deep')
plt.title("Sepal Length Distribution by Label")
plt.show()
```



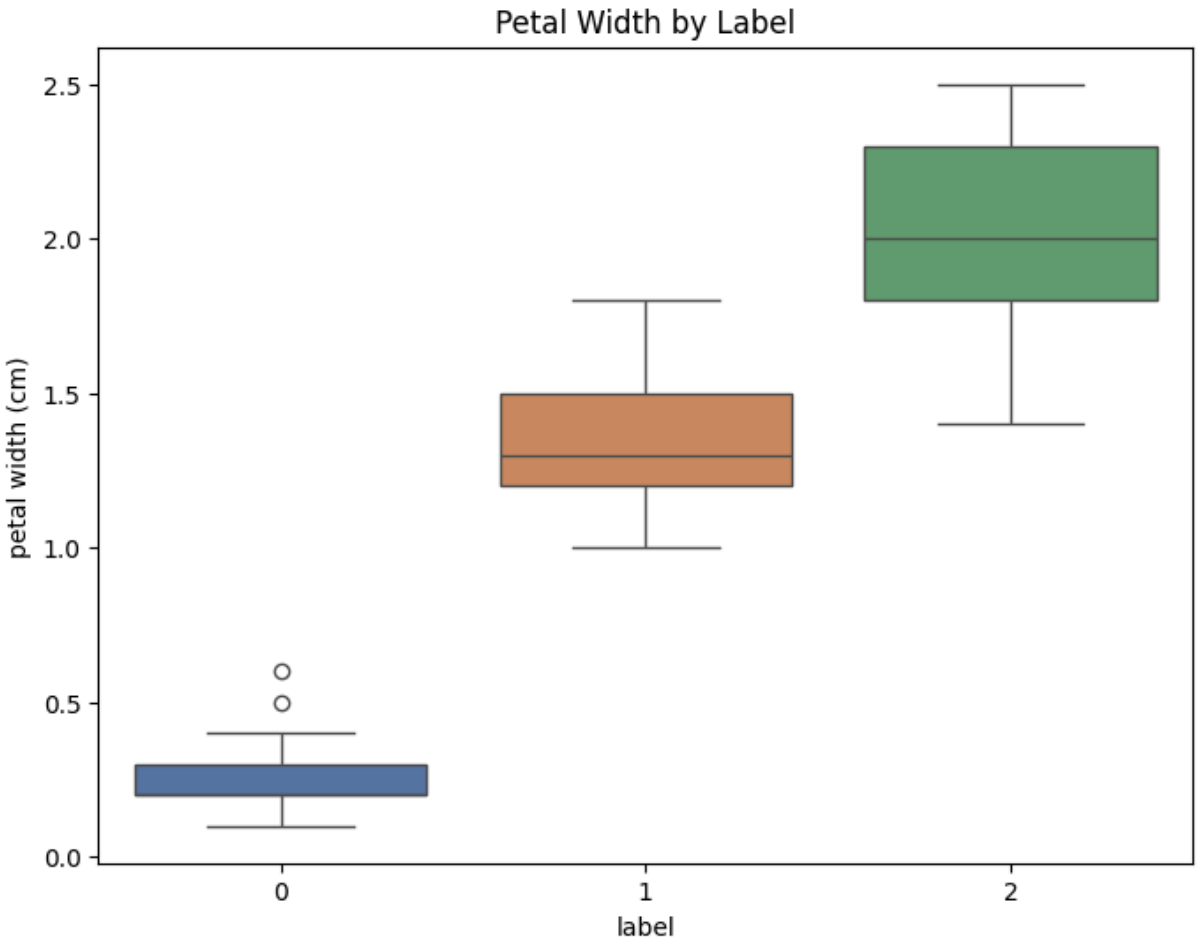
In [19]:
```
sns.boxplot(x='label', y='sepal width (cm)', data=df, palette='deep')
plt.title("Sepal Width Distribution by Label")
plt.show()
```



In [24]:
```
plt.figure(figsize=(8, 6))
sns.boxplot(x='label', y='petal length (cm)', data=df, palette='deep')
plt.title("Petal Length by Label")
plt.show()
```



In [26]:
```
plt.figure(figsize=(8, 6))
sns.boxplot(x='label', y='petal width (cm)', data=df, palette='deep')
plt.title("Petal Width by Label")
plt.show()
```


Petal Width by Label
petal width (cm)
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
label