

```
In [1]: 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 [2]: data = load_iris()
```

```
In [3]: data
```

```
Out[3]: {'data': array([[5.1, 3.5, 1.4, 0.2],
 [4.9, 3. , 1.4, 0.2],
 [4.7, 3.2, 1.3, 0.2],
 [4.6, 3.1, 1.5, 0.2],
 [5. , 3.6, 1.4, 0.2],
 [5.4, 3.9, 1.7, 0.4],
 [4.6, 3.4, 1.4, 0.3],
 [5. , 3.4, 1.5, 0.2],
 [4.4, 2.9, 1.4, 0.2],
 [4.9, 3.1, 1.5, 0.1],
 [5.4, 3.7, 1.5, 0.2],
 [4.8, 3.4, 1.6, 0.2],
 [4.8, 3. , 1.4, 0.1],
 [4.3, 3. , 1.1, 0.1],
 [5.8, 4. , 1.2, 0.2],
 [5.7, 4.4, 1.5, 0.4],
 [5.4, 3.9, 1.3, 0.4],
 [5.1, 3.5, 1.4, 0.3],
 [5.7, 3.8, 1.7, 0.3],
 [5.1, 3.8, 1.5, 0.3],
 [5.4, 3.4, 1.7, 0.2],
 [5.1, 3.7, 1.5, 0.4],
 [4.6, 3.6, 1. , 0.2],
 [5.1, 3.3, 1.7, 0.5],
 [4.8, 3.4, 1.9, 0.2],
 [5. , 3. , 1.6, 0.2],
 [5. , 3.4, 1.6, 0.4],
 [5.2, 3.5, 1.5, 0.2],
 [5.2, 3.4, 1.4, 0.2],
 [4.7, 3.2, 1.6, 0.2],
 [4.8, 3.1, 1.6, 0.2],
 [5.4, 3.4, 1.5, 0.4],
 [5.2, 4.1, 1.5, 0.1],
 [5.5, 4.2, 1.4, 0.2],
 [4.9, 3.1, 1.5, 0.2],
 [5. , 3.2, 1.2, 0.2],
 [5.5, 3.5, 1.3, 0.2],
 [4.9, 3.6, 1.4, 0.1],
 [4.4, 3. , 1.3, 0.2],
 [5.1, 3.4, 1.5, 0.2],
 [5. , 3.5, 1.3, 0.3],
 [4.5, 2.3, 1.3, 0.3],
 [4.4, 3.2, 1.3, 0.2],
 [5. , 3.5, 1.6, 0.6],
 [5.1, 3.8, 1.9, 0.4],
 [4.8, 3. , 1.4, 0.3],
 [5.1, 3.8, 1.6, 0.2],
 [4.6, 3.2, 1.4, 0.2],
 [5.3, 3.7, 1.5, 0.2],
 [5. , 3.3, 1.4, 0.2],
 [7. , 3.2, 4.7, 1.4],
 [6.4, 3.2, 4.5, 1.5],
 [6.9, 3.1, 4.9, 1.5],
 [5.5, 2.3, 4. , 1.3],
 [6.5, 2.8, 4.6, 1.5],
 [5.7, 2.8, 4.5, 1.3],
 [6.3, 3.3, 4.7, 1.6],
 [4.9, 2.4, 3.3, 1. ],
 [6.6, 2.9, 4.6, 1.3],
 [5.2, 2.7, 3.9, 1.4],
 [5. , 2. , 3.5, 1. ],
 [5.9, 3. , 4.2, 1.5],
 [6. , 2.2, 4. , 1. ],
 [6.1, 2.9, 4.7, 1.4],
 [5.6, 2.9, 3.6, 1.3],
 [6.7, 3.1, 4.4, 1.4],
 [5.6, 3. , 4.5, 1.5],
 [5.8, 2.7, 4.1, 1. ],
 [6.2, 2.2, 4.5, 1.5],
 [5.6, 2.5, 3.9, 1.1],
 [5.9, 3.2, 4.8, 1.8],
 [6.1, 2.8, 4. , 1.3],
 [6.3, 2.5, 4.9, 1.5],
```

```
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
[6.7, 3. , 5. , 1.7],
[6. , 2.9, 4.5, 1.5],
[5.7, 2.6, 3.5, 1. ],
[5.5, 2.4, 3.8, 1.1],
[5.5, 2.4, 3.7, 1. ],
[5.8, 2.7, 3.9, 1.2],
[6. , 2.7, 5.1, 1.6],
[5.4, 3. , 4.5, 1.5],
[6. , 3.4, 4.5, 1.6],
[6.7, 3.1, 4.7, 1.5],
[6.3, 2.3, 4.4, 1.3],
[5.6, 3. , 4.1, 1.3],
[5.5, 2.5, 4. , 1.3],
[5.5, 2.6, 4.4, 1.2],
[6.1, 3. , 4.6, 1.4],
[5.8, 2.6, 4. , 1.2],
[5. , 2.3, 3.3, 1. ],
[5.6, 2.7, 4.2, 1.3],
[5.7, 3. , 4.2, 1.2],
[5.7, 2.9, 4.2, 1.3],
[6.2, 2.9, 4.3, 1.3],
[5.1, 2.5, 3. , 1.1],
[5.7, 2.8, 4.1, 1.3],
[6.3, 3.3, 6. , 2.5],
[5.8, 2.7, 5.1, 1.9],
[7.1, 3. , 5.9, 2.1],
[6.3, 2.9, 5.6, 1.8],
[6.5, 3. , 5.8, 2.2],
[7.6, 3. , 6.6, 2.1],
[4.9, 2.5, 4.5, 1.7],
[7.3, 2.9, 6.3, 1.8],
[6.7, 2.5, 5.8, 1.8],
[7.2, 3.6, 6.1, 2.5],
[6.5, 3.2, 5.1, 2. ],
[6.4, 2.7, 5.3, 1.9],
[6.8, 3. , 5.5, 2.1],
[5.7, 2.5, 5. , 2. ],
[5.8, 2.8, 5.1, 2.4],
[6.4, 3.2, 5.3, 2.3],
[6.5, 3. , 5.5, 1.8],
[7.7, 3.8, 6.7, 2.2],
[7.7, 2.6, 6.9, 2.3],
[6. , 2.2, 5. , 1.5],
[6.9, 3.2, 5.7, 2.3],
[5.6, 2.8, 4.9, 2. ],
[7.7, 2.8, 6.7, 2. ],
[6.3, 2.7, 4.9, 1.8],
[6.7, 3.3, 5.7, 2.1],
[7.2, 3.2, 6. , 1.8],
[6.2, 2.8, 4.8, 1.8],
[6.1, 3. , 4.9, 1.8],
[6.4, 2.8, 5.6, 2.1],
[7.2, 3. , 5.8, 1.6],
[7.4, 2.8, 6.1, 1.9],
[7.9, 3.8, 6.4, 2. ],
[6.4, 2.8, 5.6, 2.2],
[6.3, 2.8, 5.1, 1.5],
[6.1, 2.6, 5.6, 1.4],
[7.7, 3. , 6.1, 2.3],
[6.3, 3.4, 5.6, 2.4],
[6.4, 3.1, 5.5, 1.8],
[6. , 3. , 4.8, 1.8],
[6.9, 3.1, 5.4, 2.1],
[6.7, 3.1, 5.6, 2.4],
[6.9, 3.1, 5.1, 2.3],
[5.8, 2.7, 5.1, 1.9],
[6.8, 3.2, 5.9, 2.3],
[6.7, 3.3, 5.7, 2.5],
[6.7, 3. , 5.2, 2.3],
[6.3, 2.5, 5. , 1.9],
[6.5, 3. , 5.2, 2. ],
[6.2, 3.4, 5.4, 2.3],
[5.9, 3. , 5.1, 1.8]]),
'target': 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, 0,
0, 0, 0, 0, 0, 0, 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,
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,
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, 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, 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, 2, 2, 2]),
'frame': None,
'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U10'),
'DESCR': '.. _iris_dataset:\n\nIris plants dataset\n-----\n\n**Data Set Characteristics:**\n\n:Number of Instances: 150 (50 in each of three classes)\n      :Number of Attributes: 4 numeric, predictive attributes and the class\n      :Attribute Information:\n         - sepal length in cm\n         - sepal width in cm\n         - petal length in cm\n         - petal width in cm\n         - class: setosa, versicolor, virginica (all lowercase)
```

```

- petal length in cm\n      - petal width in cm\n      - class:\n      - Iris-Setosa\n
- Iris-Versicolour\n      - Iris-Virginica\n      \n      :Summary Statistics:\n\n      =====
===== \n\n      Min Max Mean SD Class Correla
tion\n      ===== \n      sepal length:  4.3 7.9  5.84  0.
83  0.7826\n      sepal width:  2.0 4.4  3.05  0.43  -0.4194\n      petal length:  1.0 6.9  3.76  1.76
0.9490 (high!)\n      petal width:  0.1 2.5  1.20  0.76  0.9565 (high!)\n      ===== \n\n
===== \n\n      :Missing Attribute Values: None\n      :Class Distribution: 33.3% for each
of 3 classes.\n      :Creator: R.A. Fisher\n      :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n      :D
ate: July, 1988\n\nThe famous Iris database, first used by Sir R.A. Fisher. The dataset is taken\nfrom Fisher\'
s paper. Note that it\'s the same as in R, but not as in the UCI\nMachine Learning Repository, which has two wr
ong data points.\n\nThis is perhaps the best known database to be found in the\npattern recognition literature.
Fisher\'s paper is a classic in the field and\nis referenced frequently to this day. (See Duda & Hart, for exa
mple.) The\ndata set contains 3 classes of 50 instances each, where each class refers to a\ntype of iris plant
. One class is linearly separable from the other 2; the\nlatter are NOT linearly separable from each other.\n\n
.. topic:: References\n\n      - Fisher, R.A. "The use of multiple measurements in taxonomic problems"\n      Annu
al Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to\n      Mathematical Statistics" (John Wiley,
NY, 1950).\n      - Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Scene Analysis.\n      (Q327.D83) Jo
hn Wiley & Sons. ISBN 0-471-22361-1. See page 218.\n      - Dasarathy, B.V. (1980) "Nosing Around the Neighborho
od: A New System\n      Structure and Classification Rule for Recognition in Partially Exposed\n      Environment
s". IEEE Transactions on Pattern Analysis and Machine\n      Intelligence, Vol. PAMI-2, No. 1, 67-71.\n      - Gat
es, G.W. (1972) "The Reduced Nearest Neighbor Rule". IEEE Transactions\n      on Information Theory, May 1972,
431-433.\n      - See also: 1988 MLC Proceedings, 54-64. Cheeseman et al\'s AUTOCLASS II\n      conceptual clusteri
ng system finds 3 classes in the data.\n      - Many, many more ...',
'feature_names': ['sepal length (cm)',
'sepal width (cm)',
'petal length (cm)',
'petal width (cm)',
'filename': 'iris.csv',
'data_module': 'sklearn.datasets.data'}

```

```

In [4]: df = pd.DataFrame()
df[data['feature_names']] = data['data']
df['label'] = data['target']

```

```

In [5]: df.head()

```

```

Out[5]:
   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 [6]: df.shape

```

```

Out[6]: (150, 5)

```

```

In [7]: 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    int32
dtypes: float64(4), int32(1)
memory usage: 5.4 KB

```

```

In [8]: df.describe()

```

```

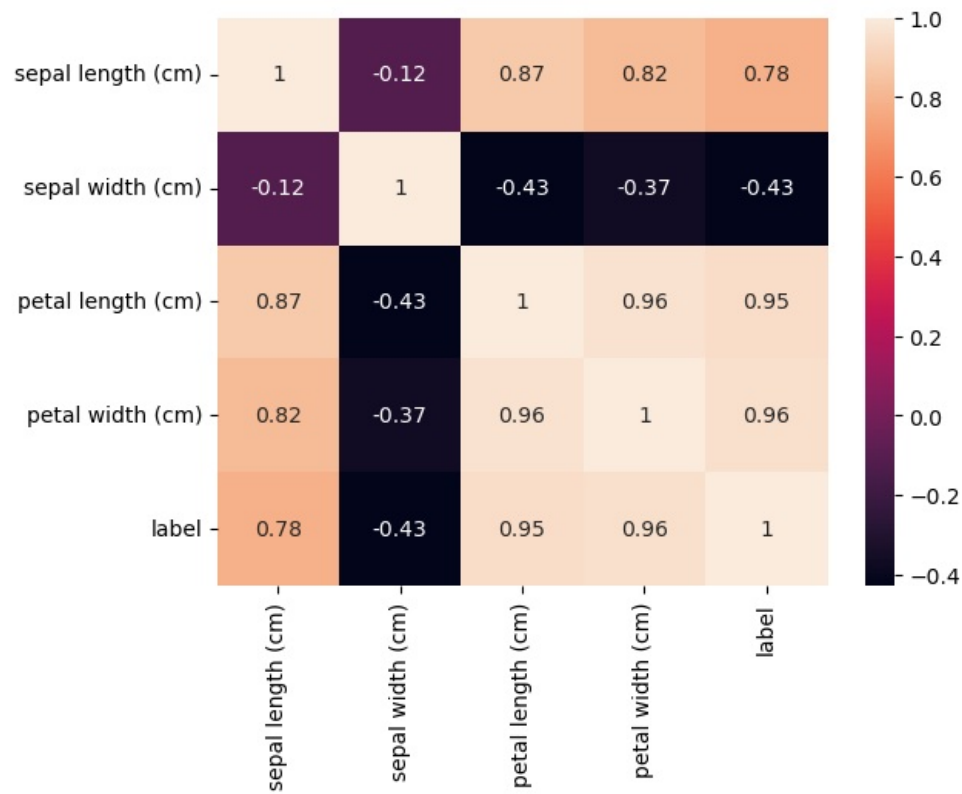
Out[8]:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  label
count      150.000000      150.000000      150.000000      150.000000      150.000000
mean         5.843333         3.057333         3.758000         1.199333         1.000000
std          0.828066         0.435866         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         3.350000         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

```

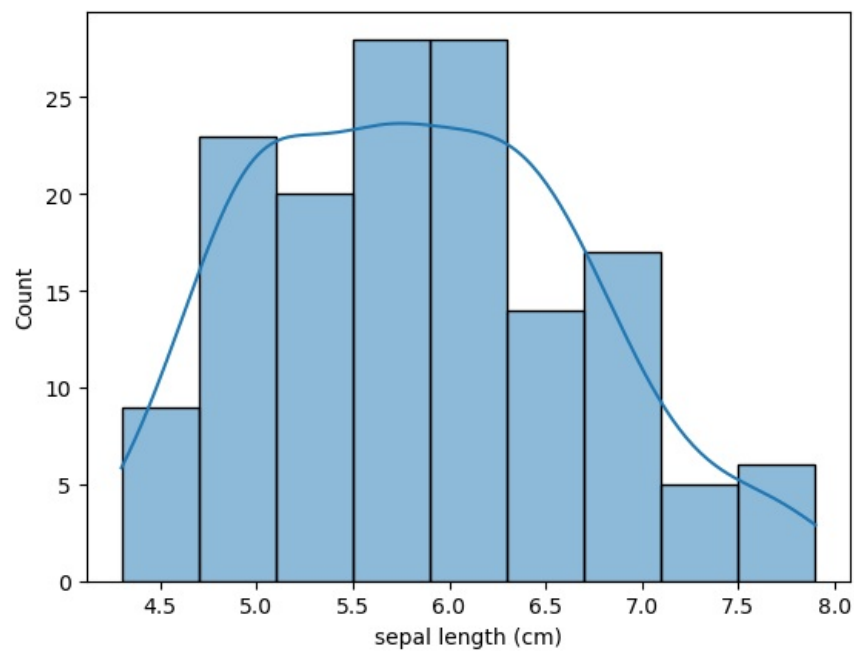
```

In [9]: sns.heatmap(df.corr(), annot=True)
plt.show()

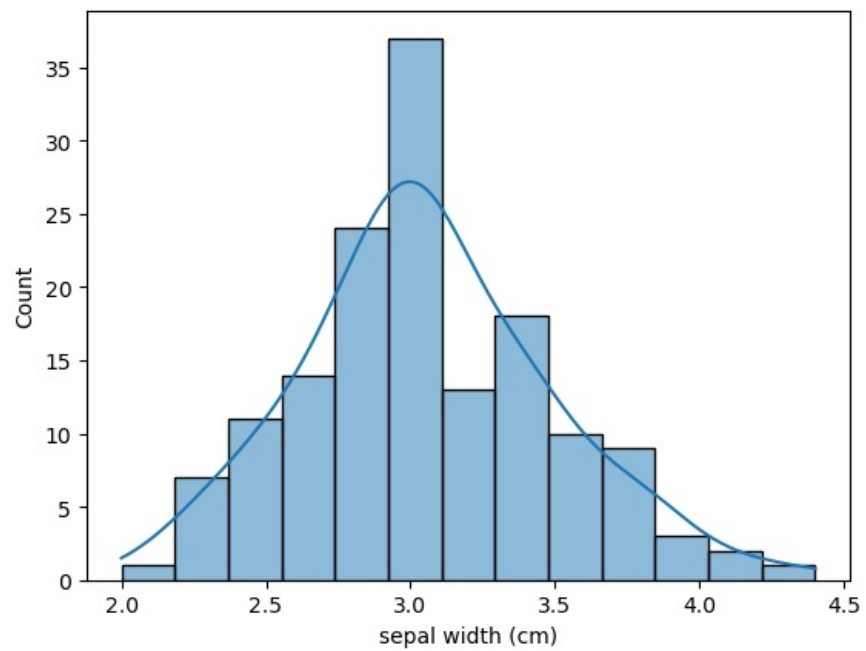
```



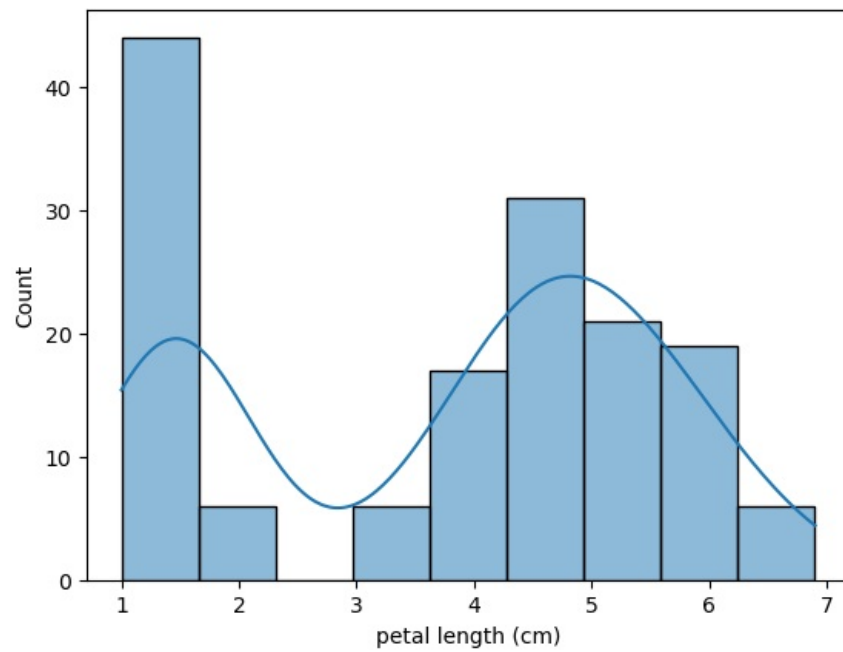
```
In [10]: sns.histplot(df["sepal length (cm)"], kde=True)
plt.show()
```



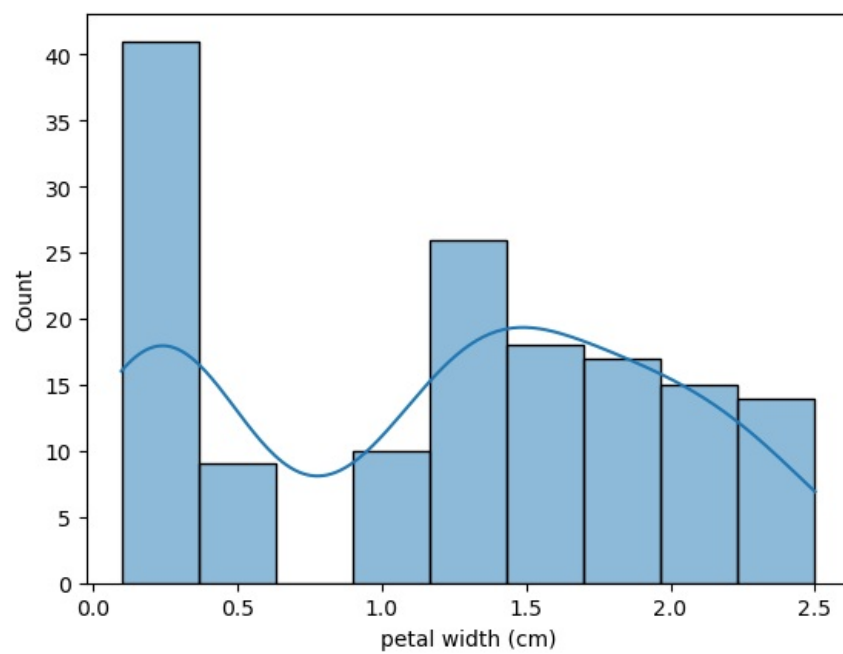
```
In [11]: sns.histplot(df["sepal width (cm)"], kde=True)
plt.show()
```



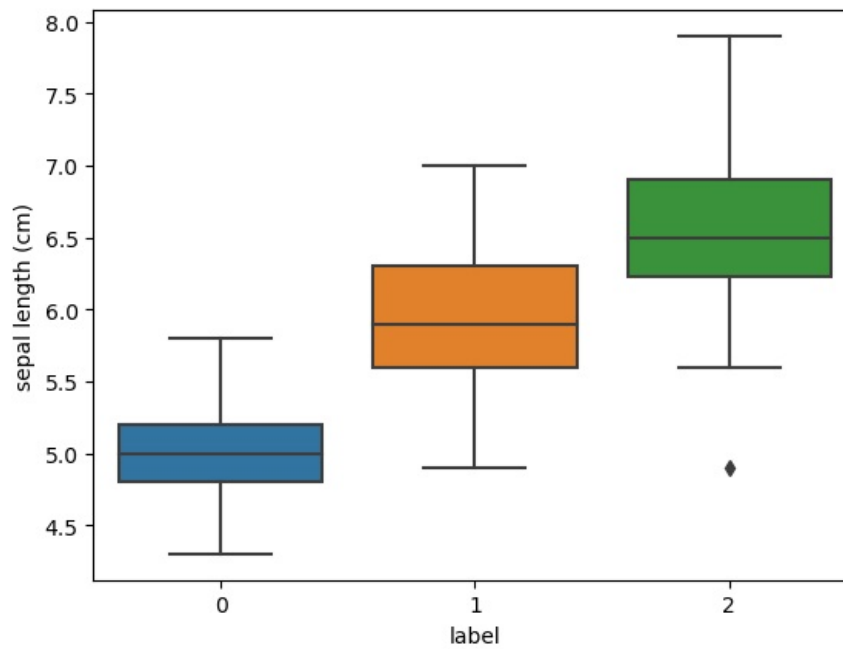
```
In [12]: sns.histplot(df["petal length (cm)"], kde=True)
plt.show()
```



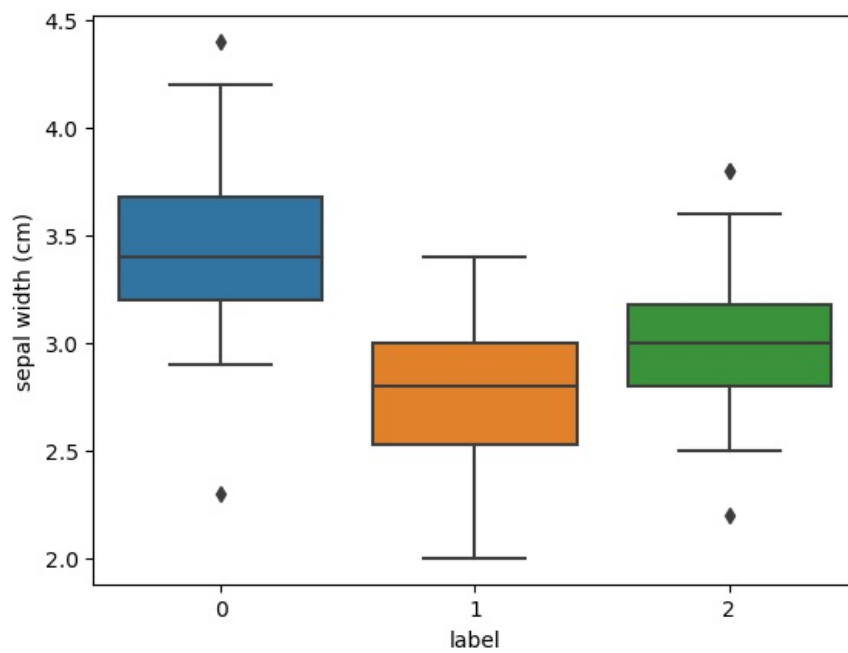
```
In [13]: sns.histplot(df["petal width (cm)"], kde=True)
plt.show()
```



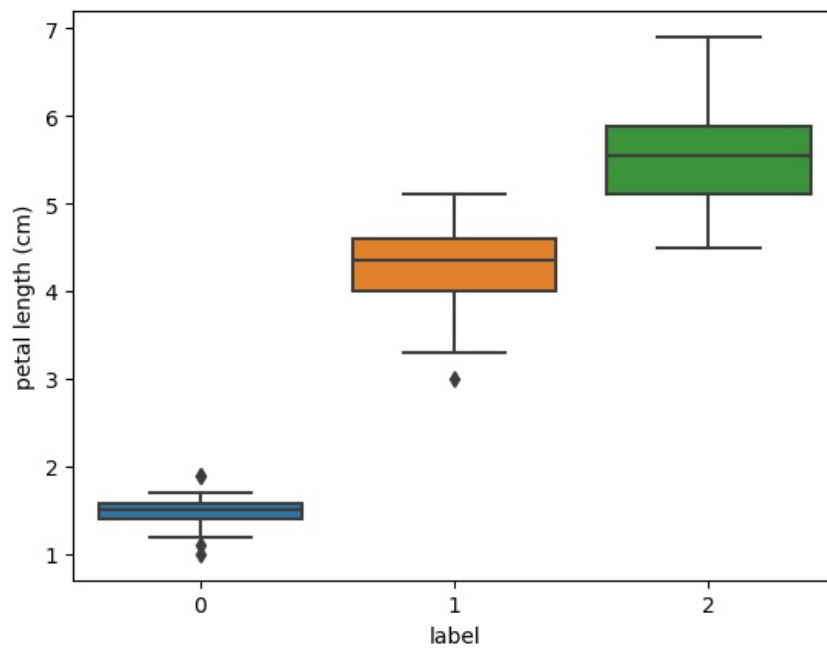
```
In [14]: sns.boxplot(x=df['label'], y=df["sepal length (cm)"])
plt.show()
```



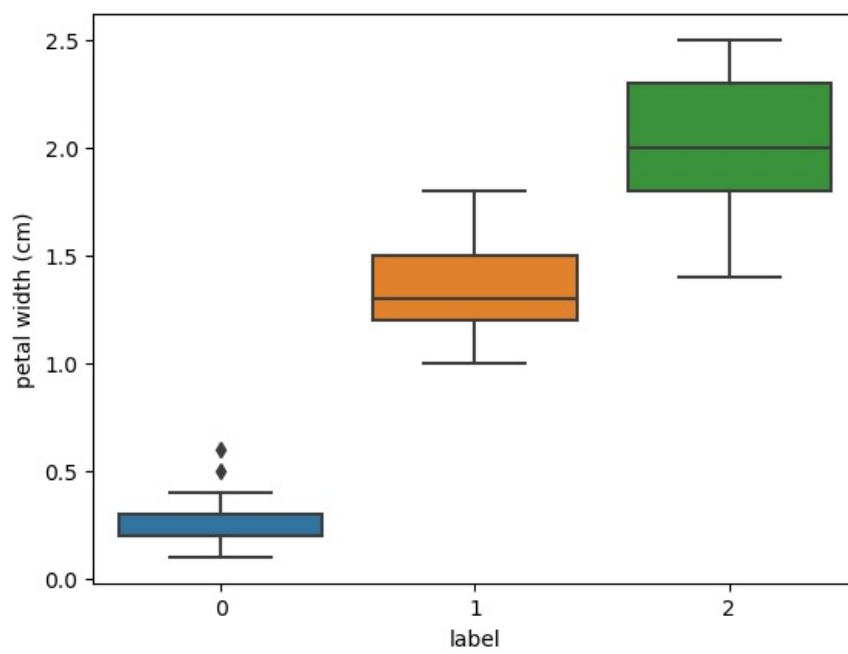
```
In [15]: sns.boxplot(x=df['label'], y=df["sepal width (cm)"])
plt.show()
```



```
In [16]: sns.boxplot(x=df["label"] ,y=df["petal length (cm)"])
plt.show()
```



```
In [17]: sns.boxplot(x=df['label'] ,y=df["petal width (cm)"])
plt.show()
```



In [ ]: # Akshay Pardeshi TE 13262

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js