In [1]:

```
'''
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.
'''
```

Out[1]:

```
'\nData Visualization III\n\nDownload the Iris flower dataset or any other dataset into a DataFrame. (e.g.,\nht
tps://archive.ics.uci.edu/ml/datasets/Iris ). Scan the dataset and give the inference as:\n\n1. List down the f
eatures and their types (e.g., numeric, nominal) available in the dataset.\n\n2. Create a histogram for each fe
ature in the dataset to illustrate the feature distributions.\n\n3. Create a boxplot for each feature in the da
taset.\n\n4. Compare distributions and identify outliers.\n'
```

In [2]:

```
import pandas as pd

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

In [4]:

```
df = pd.read_csv('Iris.csv')
df
```

Out[4]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [5]:

```
df.head()
```

Out[5]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [6]:

```
df.tail()
```

Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [8]:

```
df.describe()
```

Out[8]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [28]:

```
df.dtypes
```

Out[28]:

```
Id                 int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

In [10]:

```
np.unique(df['Species'])
```

Out[10]:

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [11]:

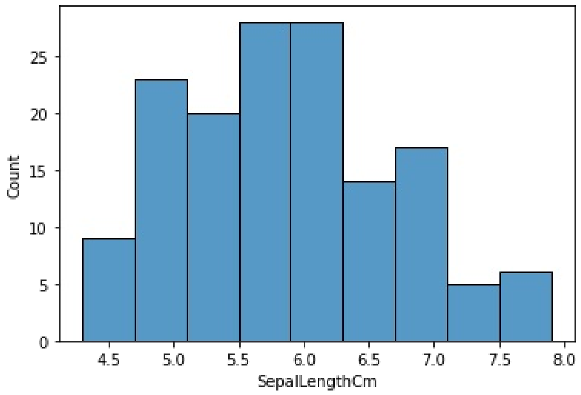
```
# Create a histogram for each feature in the dataset to illustrate the feature distributions.

# fig, axes = plt.subplots(2, 2, figsize=(16,8))   # 4 graphs plotted
```

In [13]:

```
sns.histplot(df['SepalLengthCm'])
```

Out[13]:

```
<Axes: xlabel='SepalLengthCm', ylabel='Count'>
```

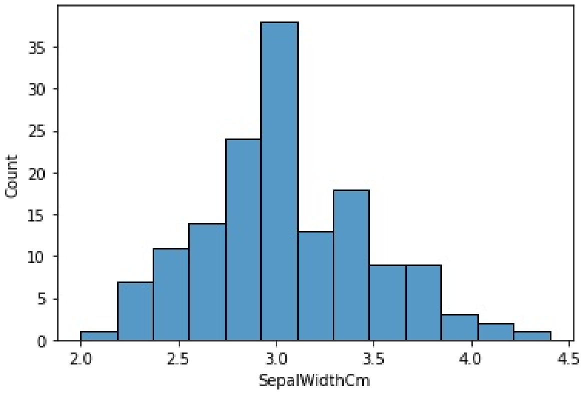


In [14]:

```
sns.histplot(df['SepalWidthCm'])
```

Out[14]:

```
<Axes: xlabel='SepalWidthCm', ylabel='Count'>
```

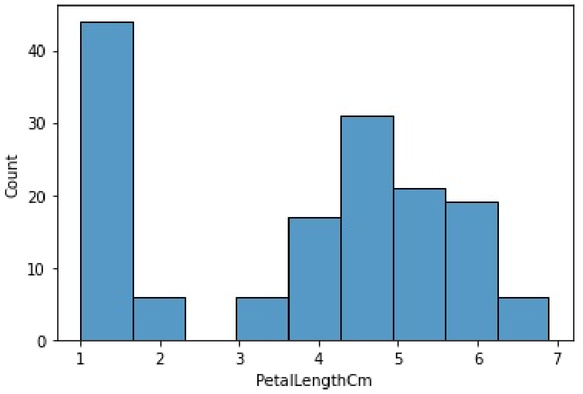


In [15]:

```
sns.histplot(df['PetalLengthCm'])
```

Out[15]:

```
<Axes: xlabel='PetalLengthCm', ylabel='Count'>
```

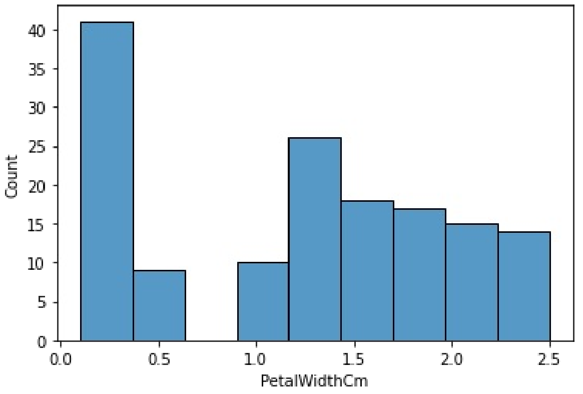


In [16]:

```
sns.histplot(df['PetalWidthCm'])
```

Out[16]:

```
<Axes: xlabel='PetalWidthCm', ylabel='Count'>
```

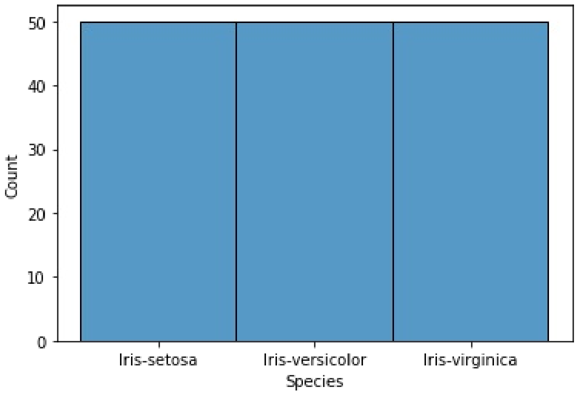


In [18]:

```
sns.histplot(df['Species'])
```

Out[18]:

```
<Axes: xlabel='Species', ylabel='Count'>
```

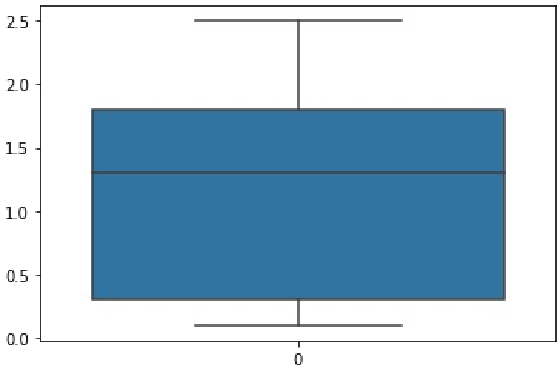


In [19]:

```
sns.boxplot(df['PetalWidthCm'])
```

Out[19]:

```
<Axes: >
```

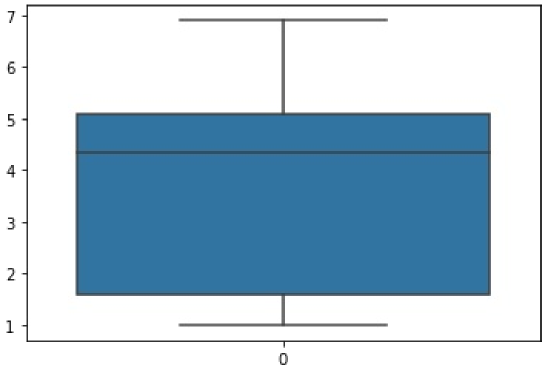


In [20]:

```
sns.boxplot(df['PetalLengthCm'])
```

Out[20]:

```
<Axes: >
```

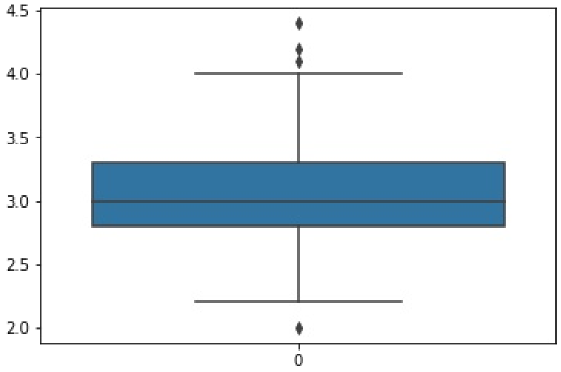


In [21]:

```
sns.boxplot(df['SepalWidthCm'])
```

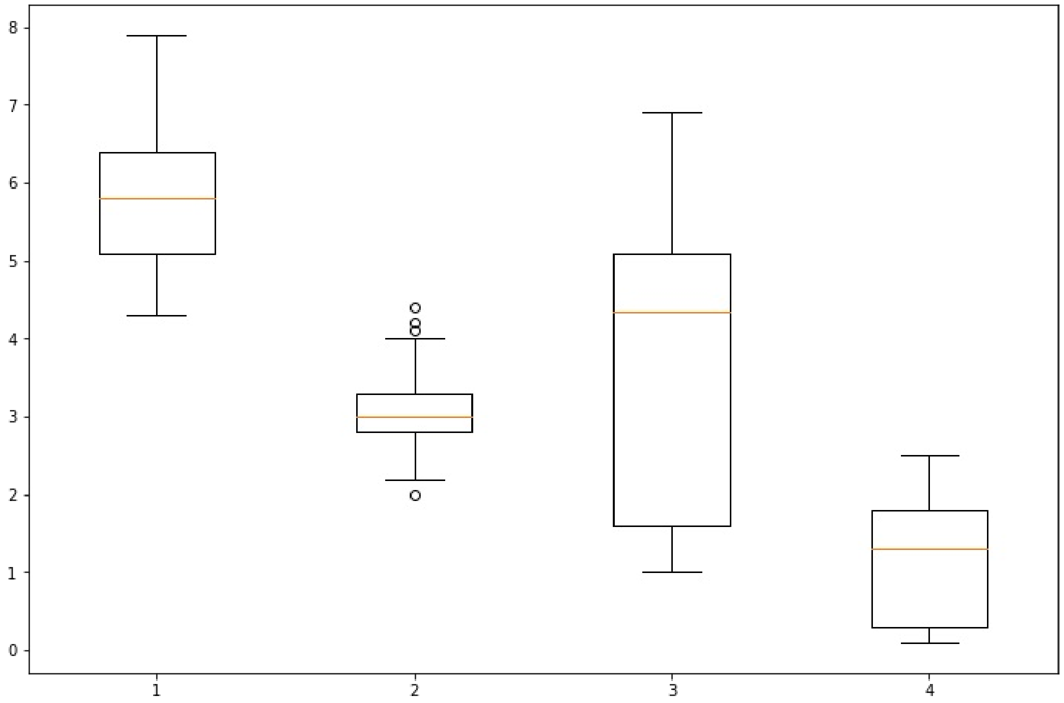
Out[21]:

```
<Axes: >
```



In [23]:

```
# comparing distributions and identifying outliers

data_to_plot = [df['SepalLengthCm'],df['SepalWidthCm'],df['PetalLengthCm'],df['PetalWidthCm']]

# Creating a figure instance
fig = plt.figure(1, figsize=(12,8)) # 1 is the unique identifier of the figure
                                    # 12,8 is the width and height in inches

# Creating an axes instance
ax = fig.add_subplot(111)           # 111 means 1x1 grid, 1st subplot

# Creating the boxplot
bp = ax.boxplot(data_to_plot);
```



In [ ]: