```
from google.colab import files


uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving Iris.csv to Iris.csv
```

```
import pandas as pd

# Reading the CSV file
df = pd.read_csv("Iris.csv")

# Printing top  7rows
df.head()
```

```
   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
```

```
df.shape
```

```
(150, 6)
```

```
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
```

```
df.describe()
```

```
                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
```

```
df.isnull().sum()
```

```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

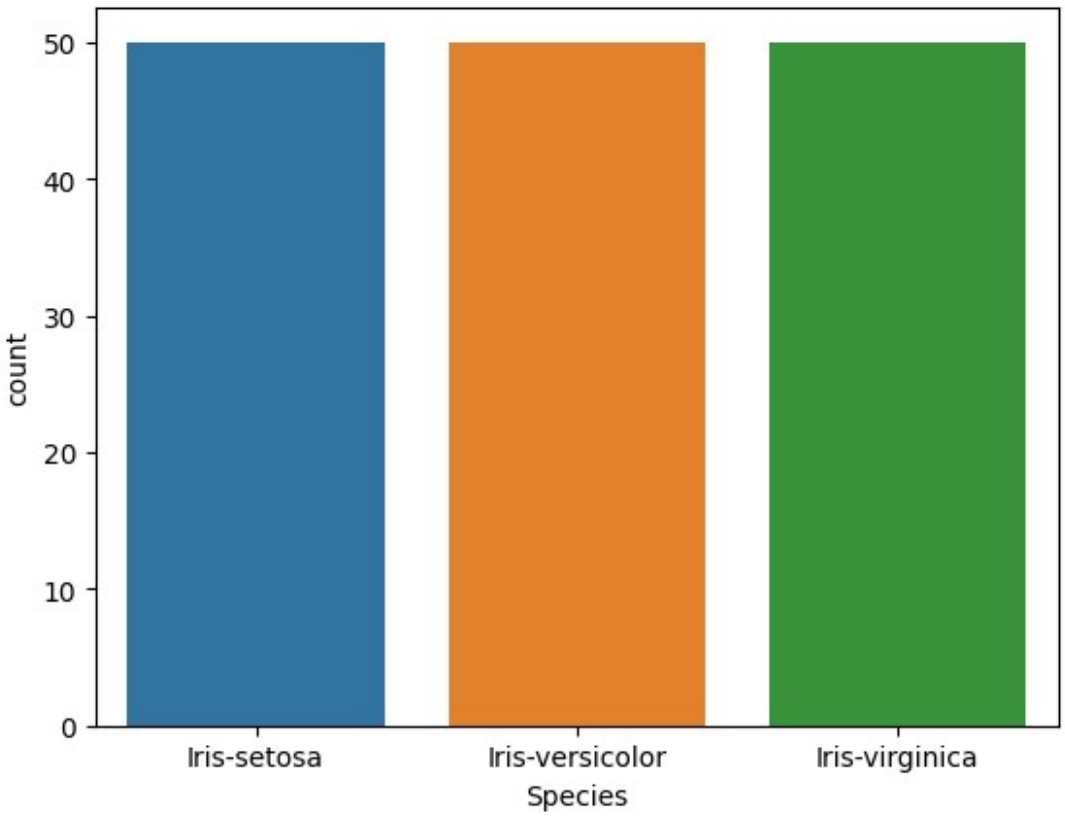
```
data = df.drop_duplicates(subset ="Species",)
data
```

```
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1            5.1           3.5            1.4           0.2
50    51            7.0           3.2            4.7           1.4
100  101            6.3           3.3            6.0           2.5

             Species
0        Iris-setosa
50   Iris-versicolor
100   Iris-virginica
```

```
df.value_counts("Species")
```

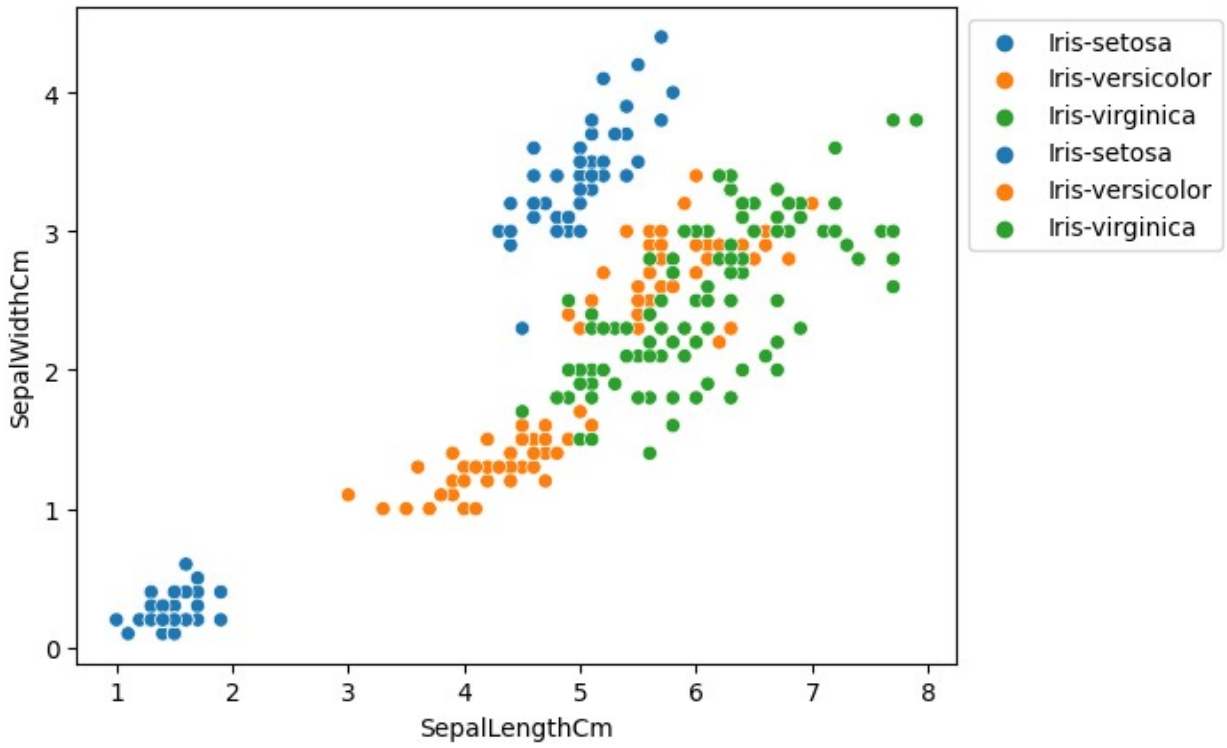
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

```
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


sns.countplot(x='Species', data=df, )
plt.show()
```

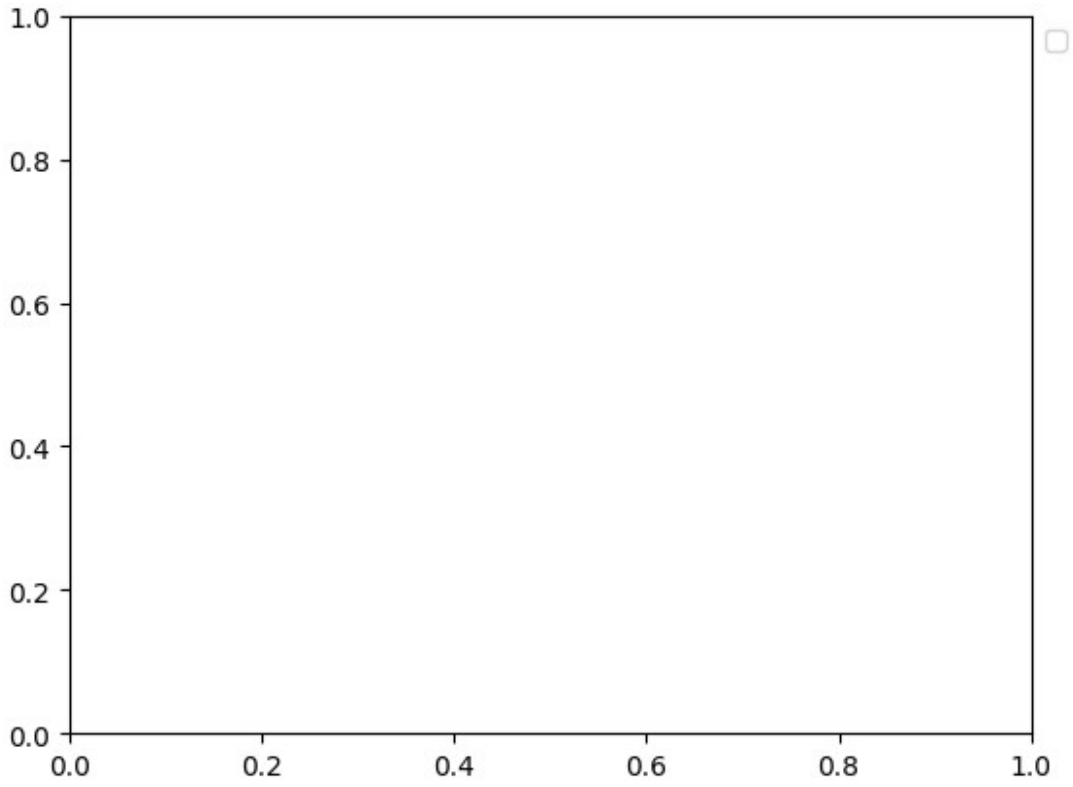



```
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm',
                      hue='Species', data=df, )

# Placing Legend outside the Figure# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


sns.scatterplot(x='PetalLengthCm', y='PetalWidthCm',
                      hue='Species', data=df, )
```

```
# Placing Legend outside the Figure
plt.legend(bbox_to_anchor=(1, 1), loc=2)

plt.show()

plt.legend(bbox_to_anchor=(1, 1), loc=2)

plt.show()
```

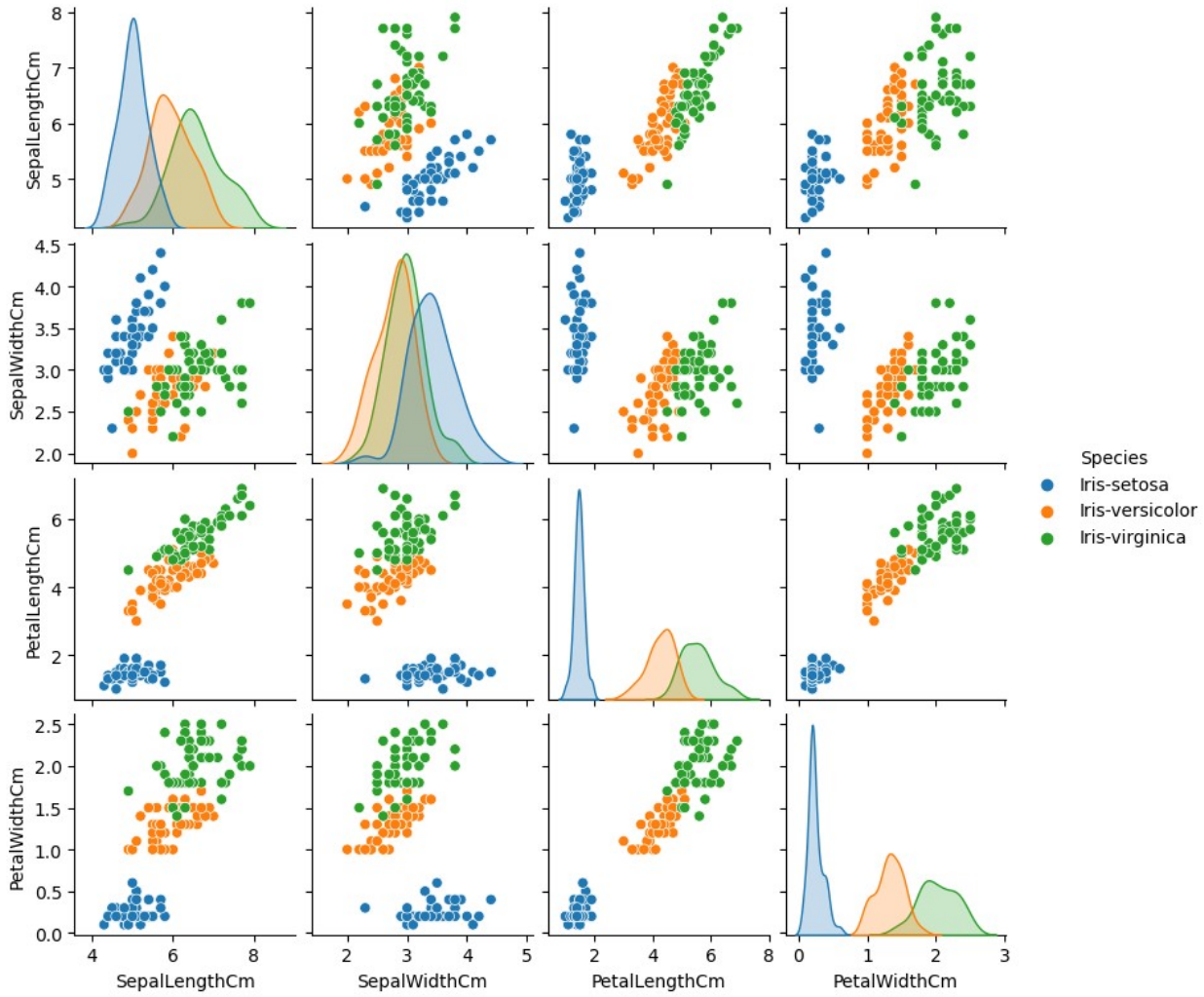



```
WARNING:matplotlib.legend:No artists with labels found to put in
legend.  Note that artists whose label start with an underscore are
ignored when legend() is called with no argument.
```

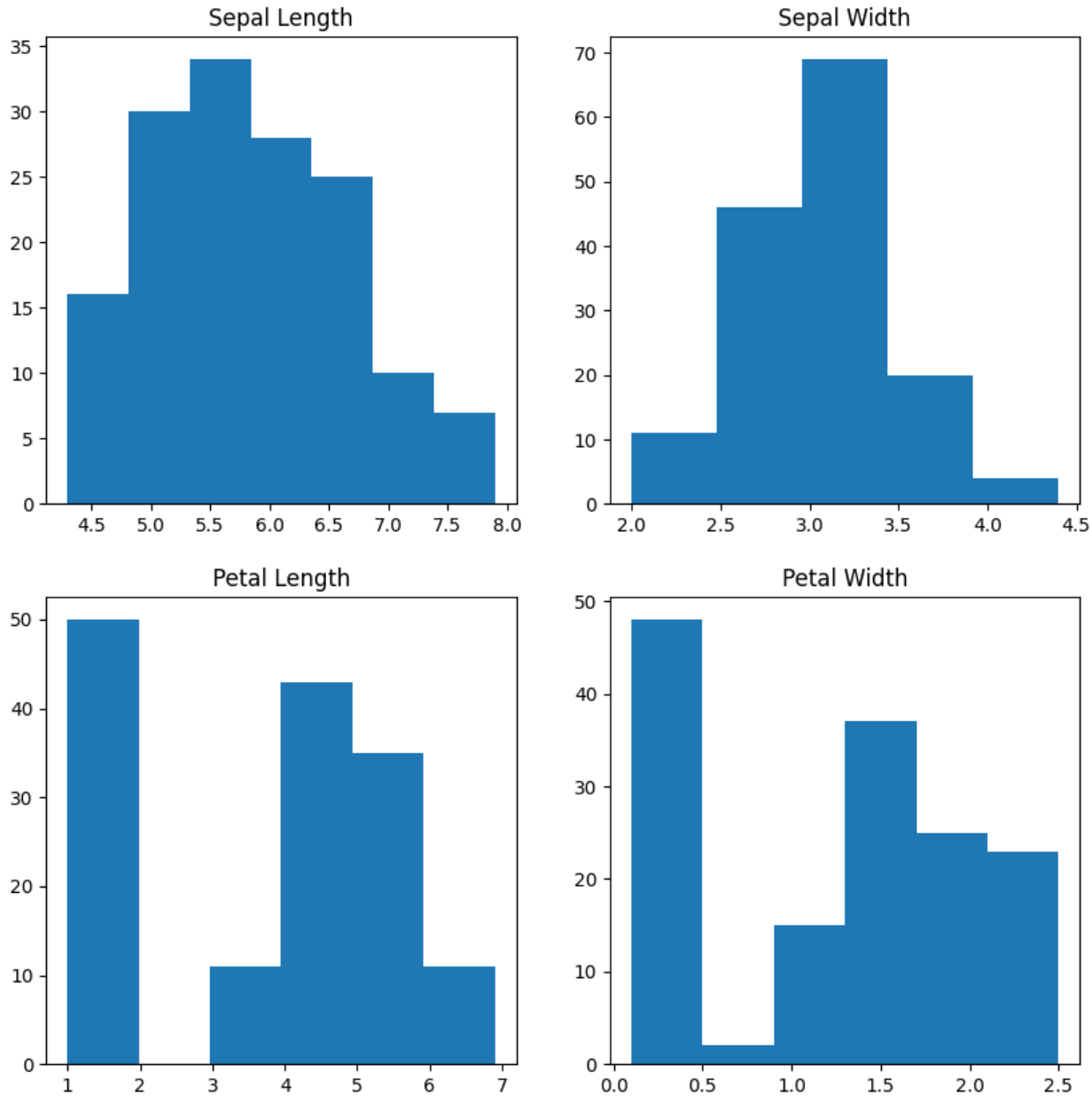



```
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


sns.pairplot(df.drop(['Id'], axis = 1),
                 hue='Species', height=2)
```

```
<seaborn.axisgrid.PairGrid at 0x7f4d0f9ce710>
```




```
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


fig, axes = plt.subplots(2, 2, figsize=(10,10))

axes[0,0].set_title("Sepal Length")
axes[0,0].hist(df['SepalLengthCm'], bins=7)

axes[0,1].set_title("Sepal Width")
axes[0,1].hist(df['SepalWidthCm'], bins=5);

axes[1,0].set_title("Petal Length")
axes[1,0].hist(df['PetalLengthCm'], bins=6);

axes[1,1].set_title("Petal Width")
axes[1,1].hist(df['PetalWidthCm'], bins=6);
```




```
data.corr(method='pearson')
```

```
<ipython-input-22-c50c7eb58c83>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it
will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  data.corr(method='pearson')
```

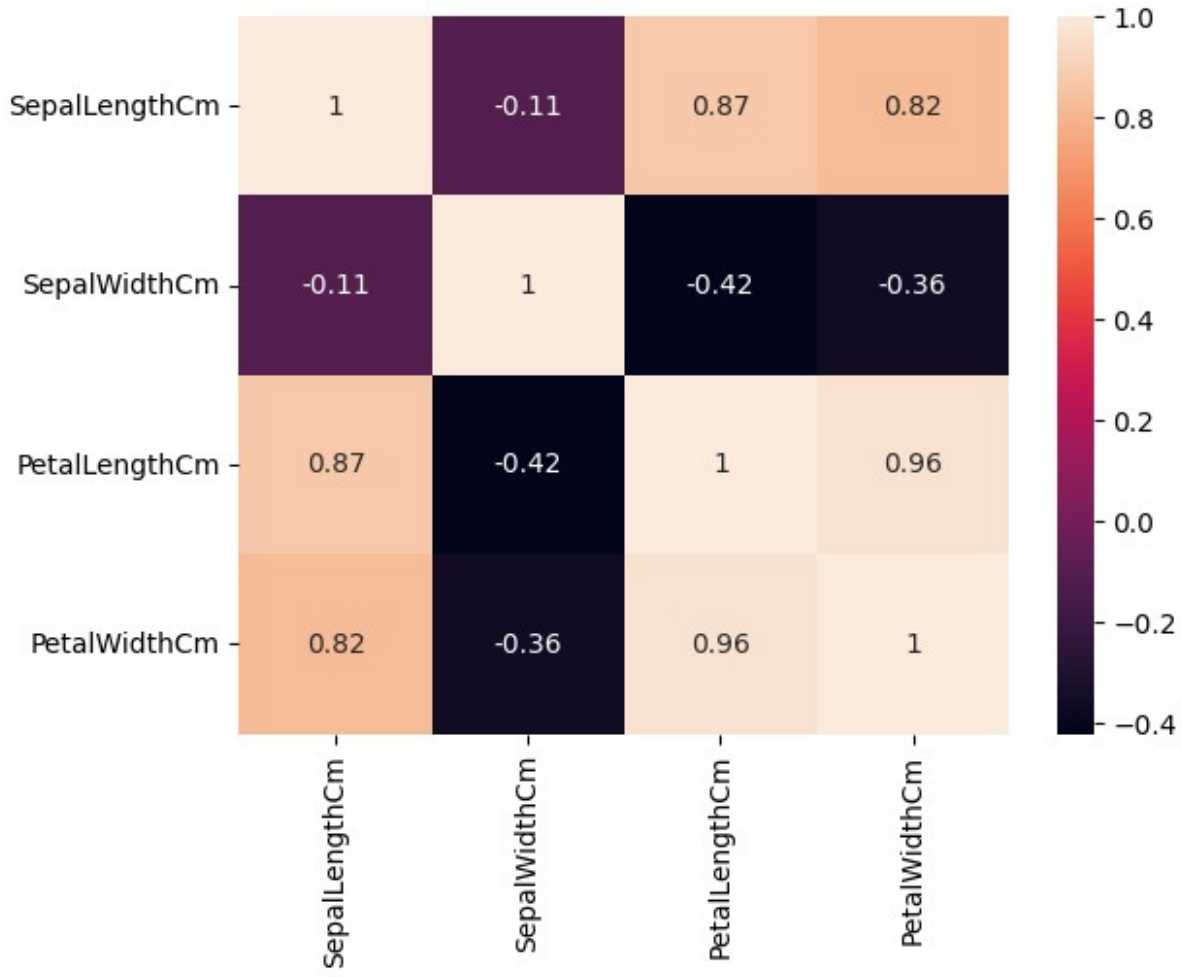
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	\
Id	1.000000	0.624413	-0.654654	0.969909	
SepalLengthCm	0.624413	1.000000	-0.999226	0.795795	
SepalWidthCm	-0.654654	-0.999226	1.000000	-0.818999	

```
PetalLengthCm  0.969909       0.795795     -0.818999       1.000000
PetalWidthCm   0.999685       0.643817     -0.673417       0.975713
```

```
                PetalWidthCm
Id                  0.999685
SepalLengthCm       0.643817
SepalWidthCm       -0.673417
PetalLengthCm       0.975713
PetalWidthCm        1.000000
```

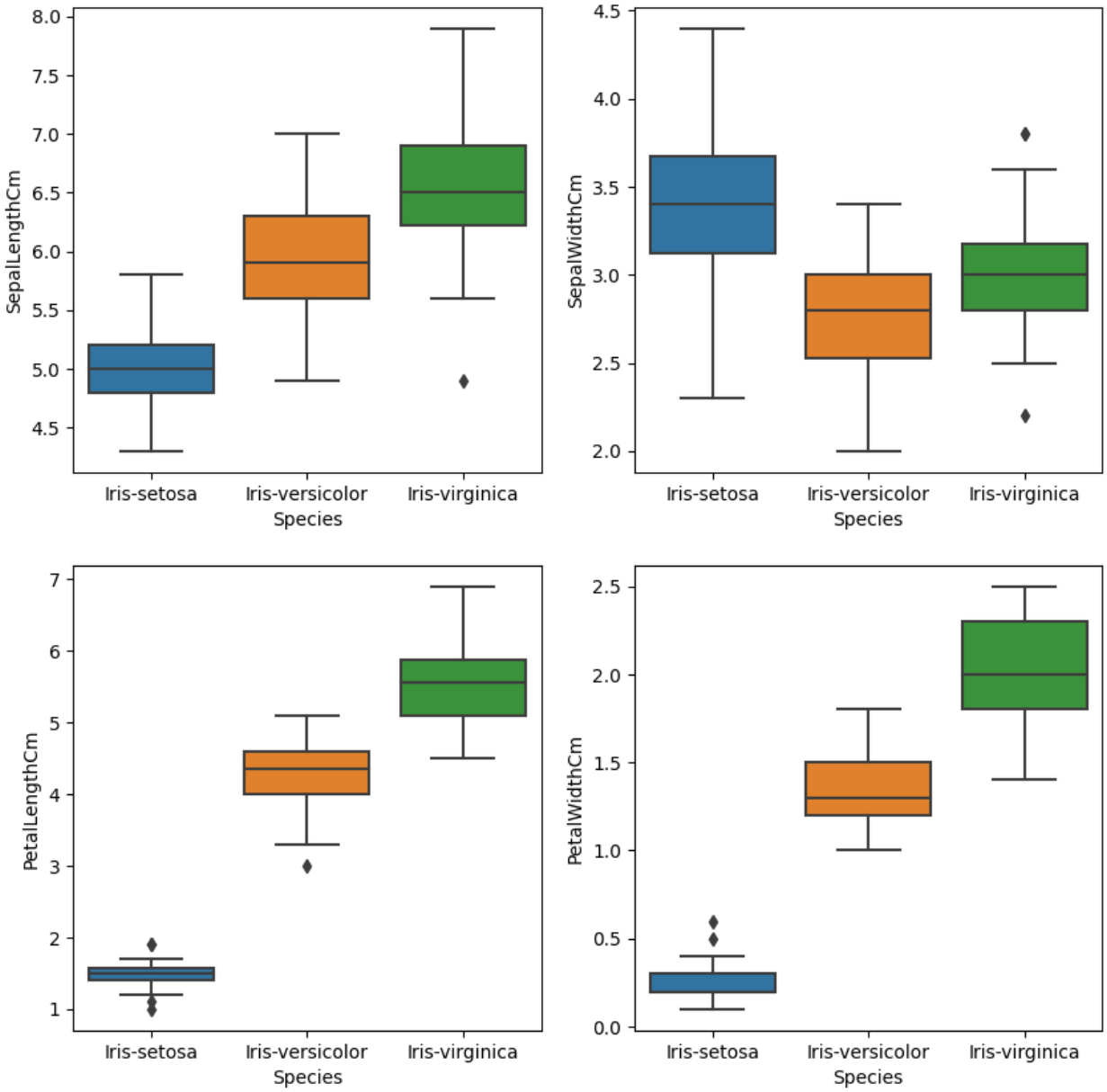
```
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt


sns.heatmap(df.corr(method='pearson').drop(
['Id'], axis=1).drop(['Id'], axis=0),
                annot = True);

plt.show()
```

```
<ipython-input-23-d09a595ab96b>:6: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it
will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  sns.heatmap(df.corr(method='pearson').drop(
```




```
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt

def graph(y):
     sns.boxplot(x="Species", y=y, data=df)

plt.figure(figsize=(10,10))

# Adding the subplot at the specified
# grid position
plt.subplot(221)
graph('SepalLengthCm')

plt.subplot(222)
graph('SepalWidthCm')

plt.subplot(223)
graph('PetalLengthCm')
```

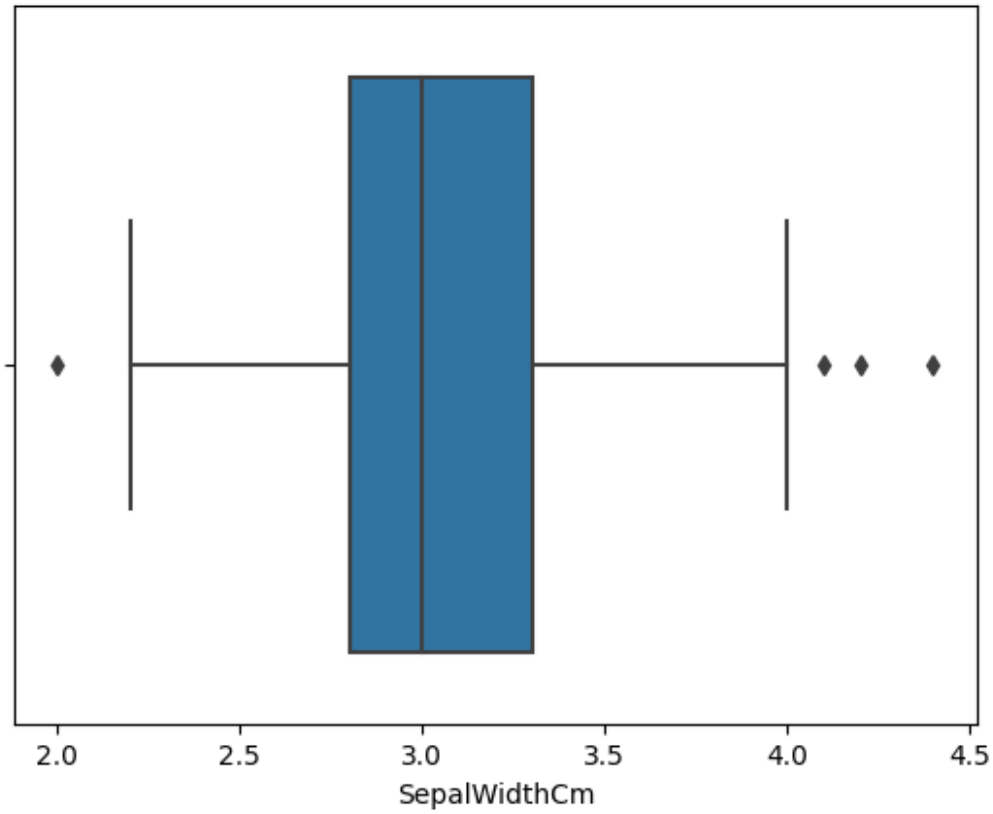
```
plt.subplot(224)
graph('PetalWidthCm')

plt.show()
```




```
# importing packages
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv('Iris.csv')
```

```
sns.boxplot(x='SepalWidthCm', data=df)
```

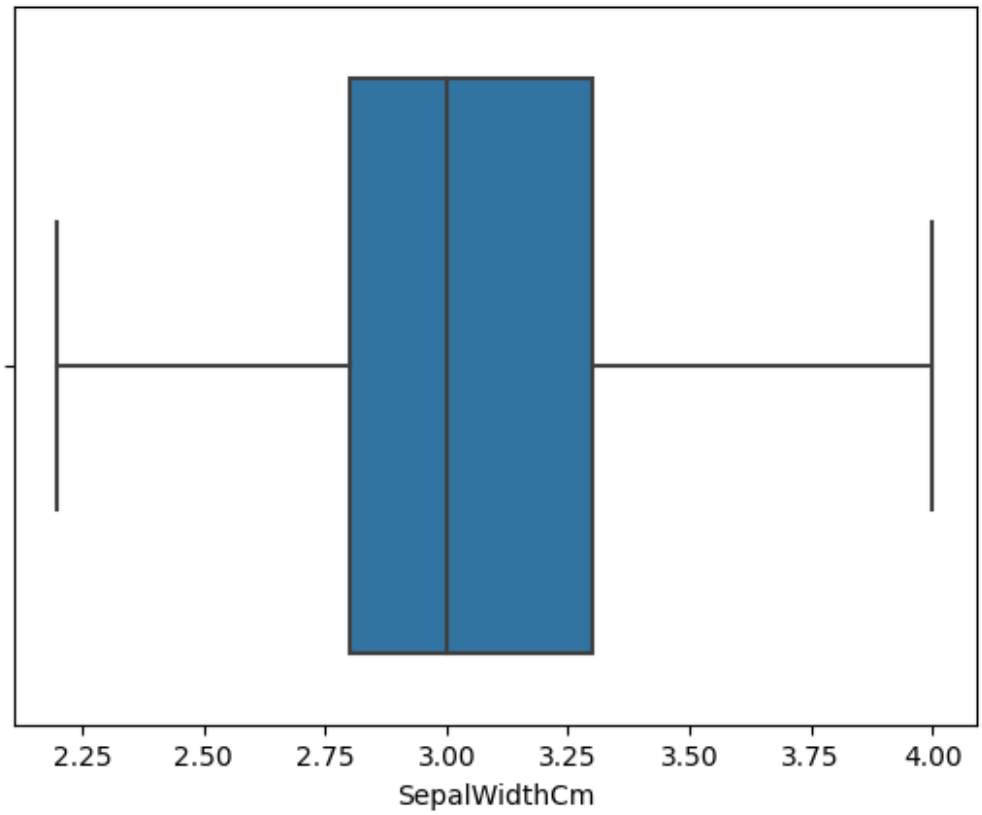
```
<Axes: xlabel='SepalWidthCm'>
```




```
# Importing necessary libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'Iris.csv' is in the current directory
df = pd.read_csv('Iris.csv')

# Calculate IQR for SepalWidthCm
Q1 = np.percentile(df['SepalWidthCm'], 25, interpolation='midpoint')
Q3 = np.percentile(df['SepalWidthCm'], 75, interpolation='midpoint')
IQR = Q3 - Q1

# Upper and lower bounds for outliers
upper_bound = Q3 + 1.5 * IQR
lower_bound = Q1 - 1.5 * IQR

# Removing outliers
```

```
df = df[(df['SepalWidthCm'] > lower_bound) & (df['SepalWidthCm'] <
upper_bound)]

# Display the old and new shape
print("Old Shape:", df.shape)
print("New Shape:", df.shape)

# Boxplot to visualize the distribution
sns.boxplot(x='SepalWidthCm', data=df)
plt.show()  # Display the plot
```

```
<ipython-input-35-4d3467ff8814>:11: DeprecationWarning: the
`interpolation=` argument to percentile was renamed to `method=`,
which has additional options.
Users of the modes 'nearest', 'lower', 'higher', or 'midpoint' are
encouraged to review the method they used. (Deprecated NumPy 1.22)
  Q1 = np.percentile(df['SepalWidthCm'], 25, interpolation='midpoint')
<ipython-input-35-4d3467ff8814>:12: DeprecationWarning: the
`interpolation=` argument to percentile was renamed to `method=`,
which has additional options.
Users of the modes 'nearest', 'lower', 'higher', or 'midpoint' are
encouraged to review the method they used. (Deprecated NumPy 1.22)
  Q3 = np.percentile(df['SepalWidthCm'], 75, interpolation='midpoint')

Old Shape: (146, 6)
New Shape: (146, 6)
```


2.25
2.50
2.75
3.00
3.25
3.50
3.75
4.00
SepalWidthCm