In [26]:
```
#importing packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

In [4]:
```
#loading the dataset
irisdata=pd.read_csv('Iris.csv')
```

#displaying the dataset irisdata

In [6]:
```
#displaying first 5 rows of dataset
irisdata.head()
```

Out[6]:

	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 [10]:
```
irisdata.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]:
```
irisdata.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 [13]:
```
irisdata.isnull().sum()
```

Out[13]:
```
Id              0
SepalLengthCm   0
SepalWidthCm    0
PetalLengthCm   0
PetalWidthCm    0
Species         0
dtype: int64
```

In [56]:
```
irisdata.value_counts("Species")
```

Out[56]:
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

In [9]:
```
X=irisdata.drop(['Id','Species'],axis=1)
y=irisdata['Species']
```

In [16]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

In [17]:
```
knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train,y_train)
```

Out[17]:
```
KNeighborsClassifier(n_neighbors=3)
```

In [18]:
```
y_pred=knn.predict(X_test)
```

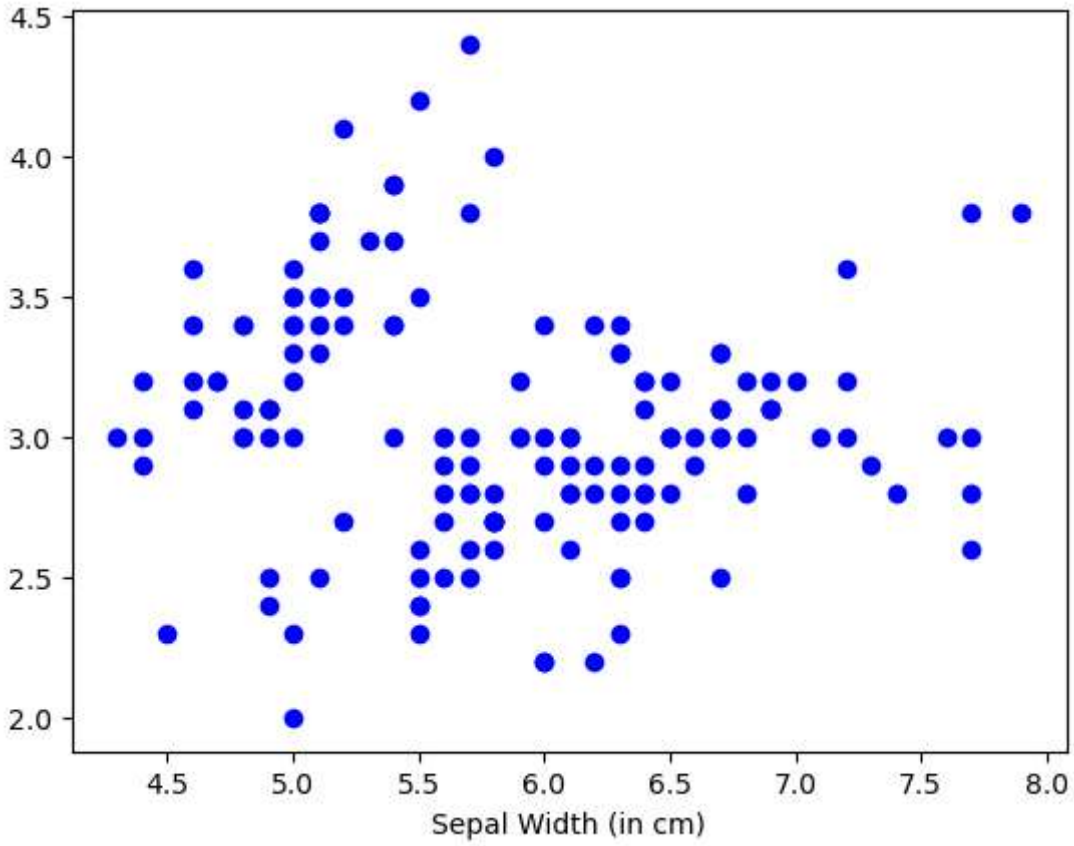
```
C:\Users\reddy\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228:
FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the defaul
t behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, thi
s behavior will change: the default value of `keepdims` will become False, the `axis`
over which the statistic is taken will be eliminated, and the value None will no long
er be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

In [20]:
```
#accuracy
accuracy=accuracy_score(y_test,y_pred)
```

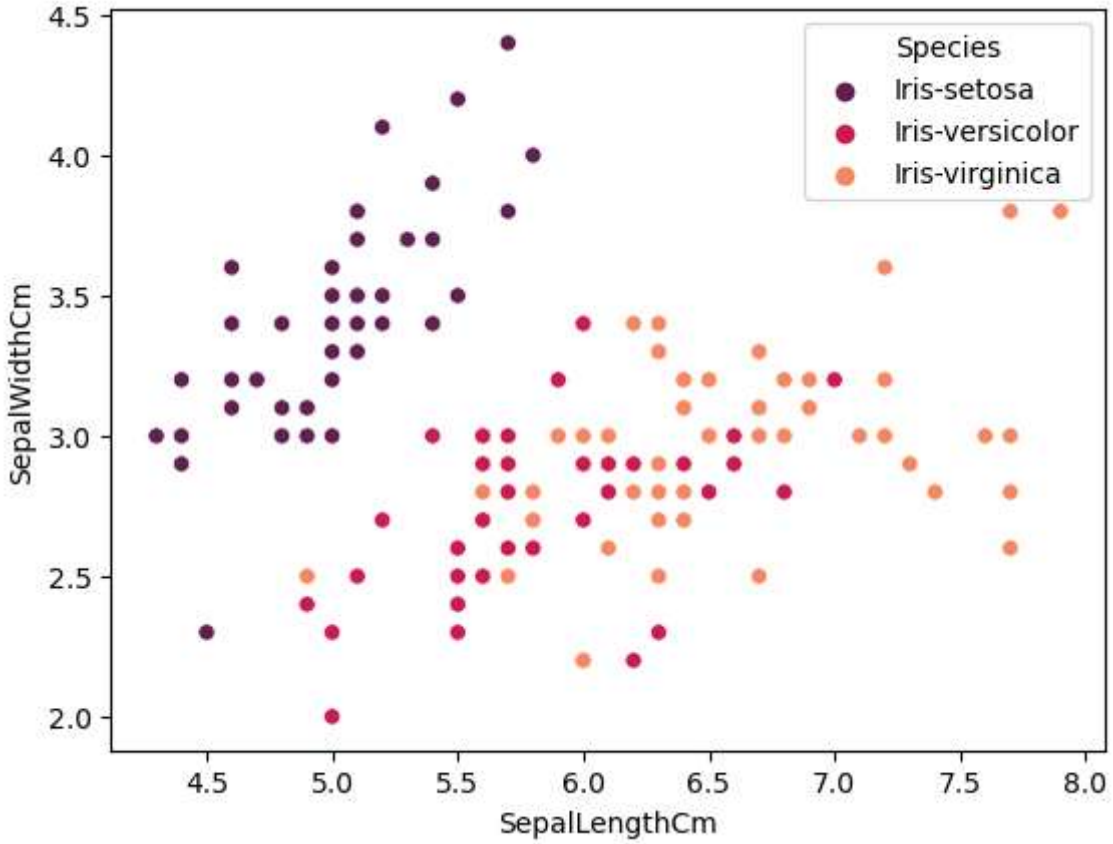
In [21]:
```
print("Accuracy is:",accuracy)
```

```
Accuracy is: 1.0
```

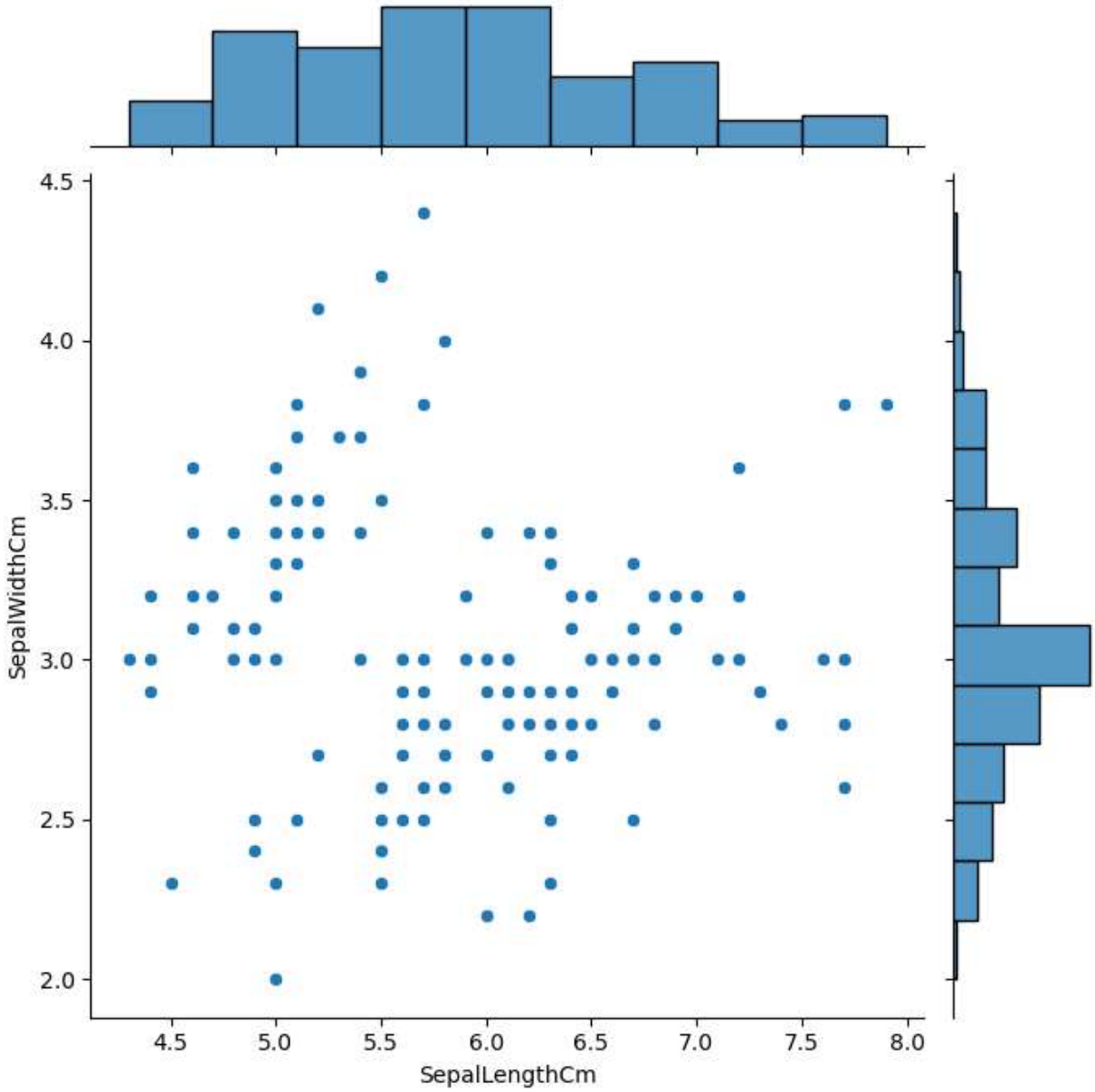
In [31]:
```
#DATA VISUALIZATION
#scatter plot
plt.scatter(irisdata["SepalLengthCm"],irisdata["SepalWidthCm"],color="b")
plt.xlabel("Sepal Length (in cm)")
plt.xlabel("Sepal Width (in cm)")
plt.show()
```



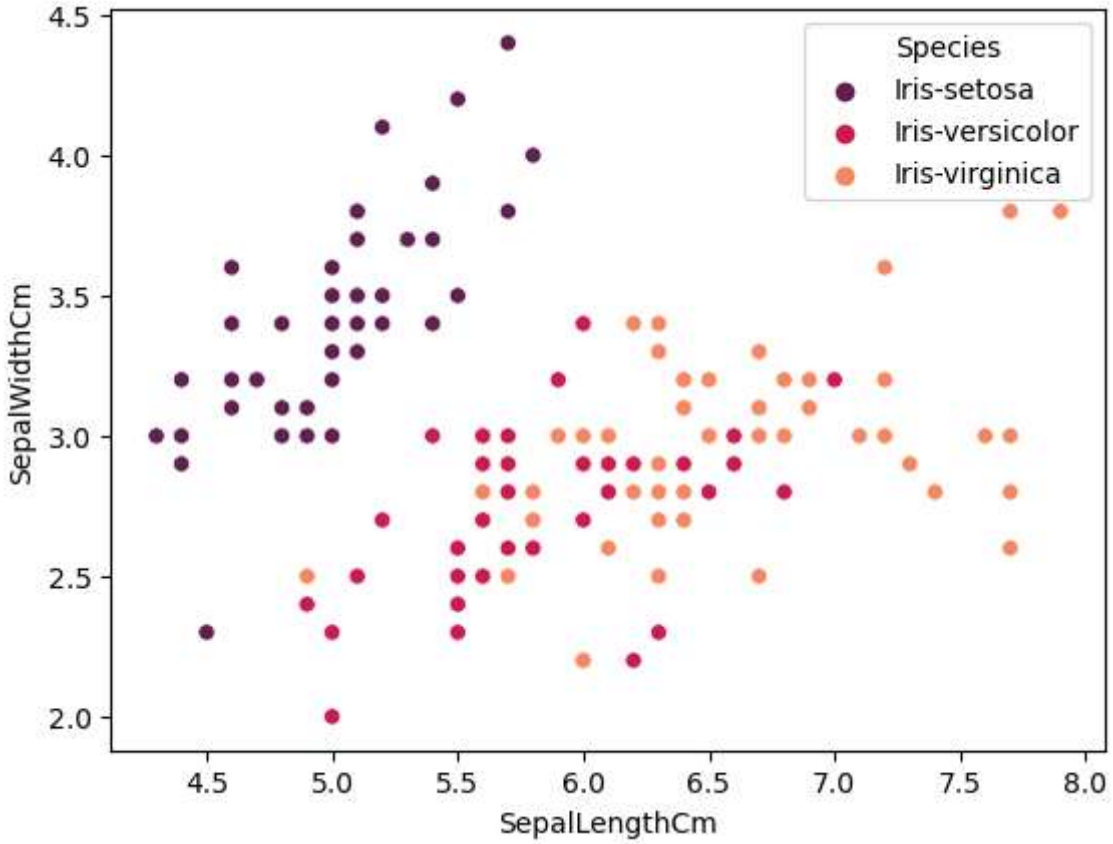
In [63]:
```
sns.scatterplot(x="SepalLengthCm",y="SepalWidthCm",data=irisdata,hue="Species",palette
plt.show()
```



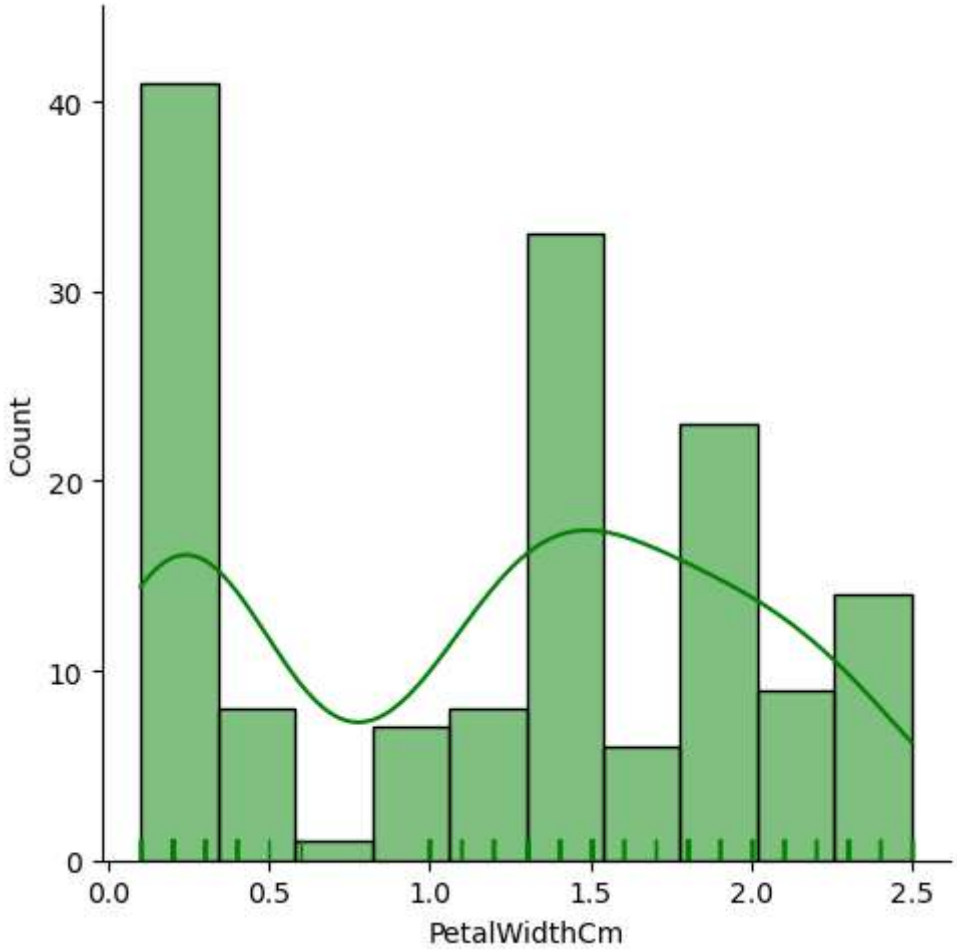
In [35]:
```
#joint plot
sns.jointplot(x="SepalLengthCm",y="SepalWidthCm",data=irisdata,height=7)
plt.show()
```



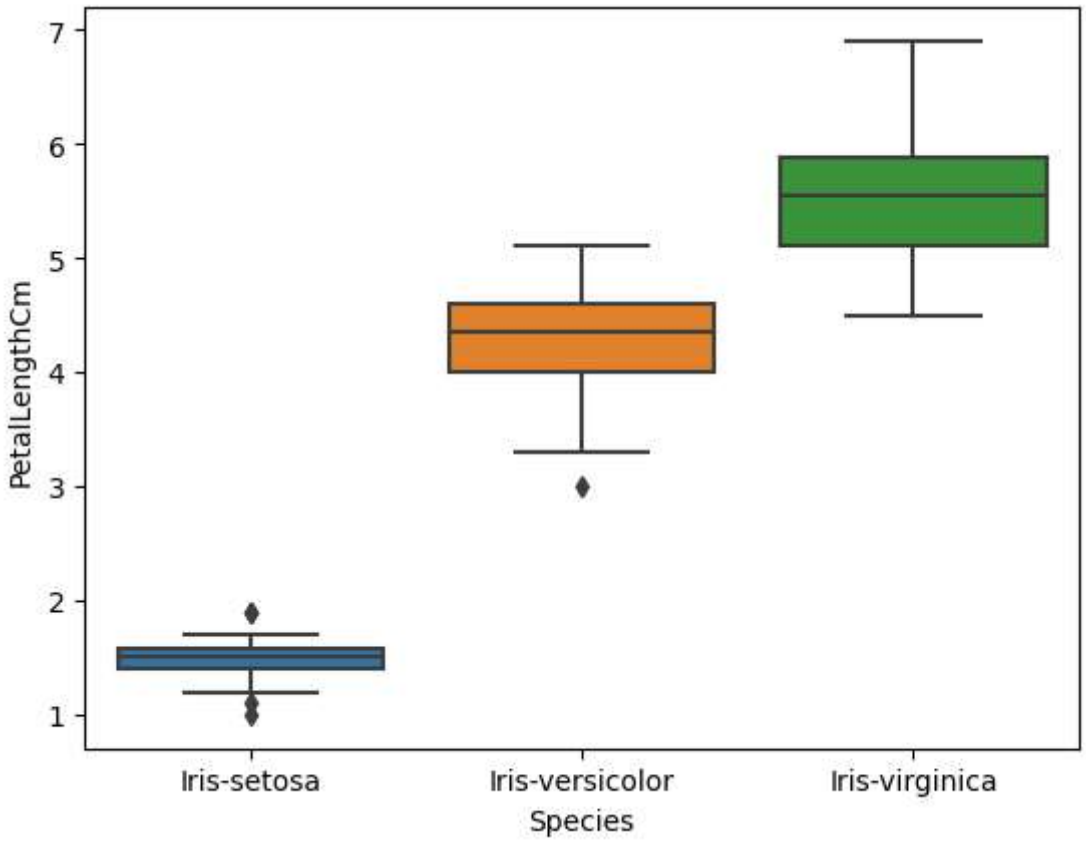
In [61]:
```
#scatter plot
sns.scatterplot(x="SepalLengthCm",y="SepalWidthCm",data=irisdata,hue="Species",palette
plt.show()
```



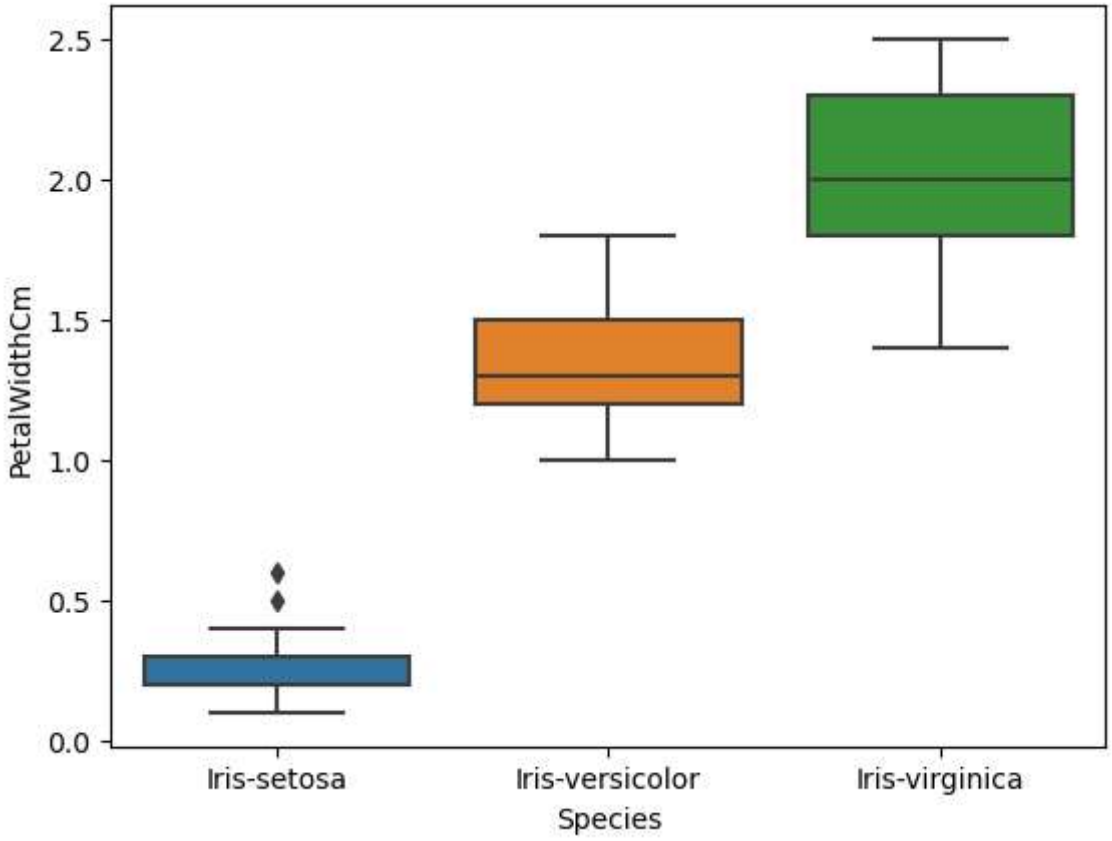
In [44]:
```
#displot
sns.displot(irisdata.PetalWidthCm,bins=10,color="green",rug="True",kde="True")
plt.show()
```



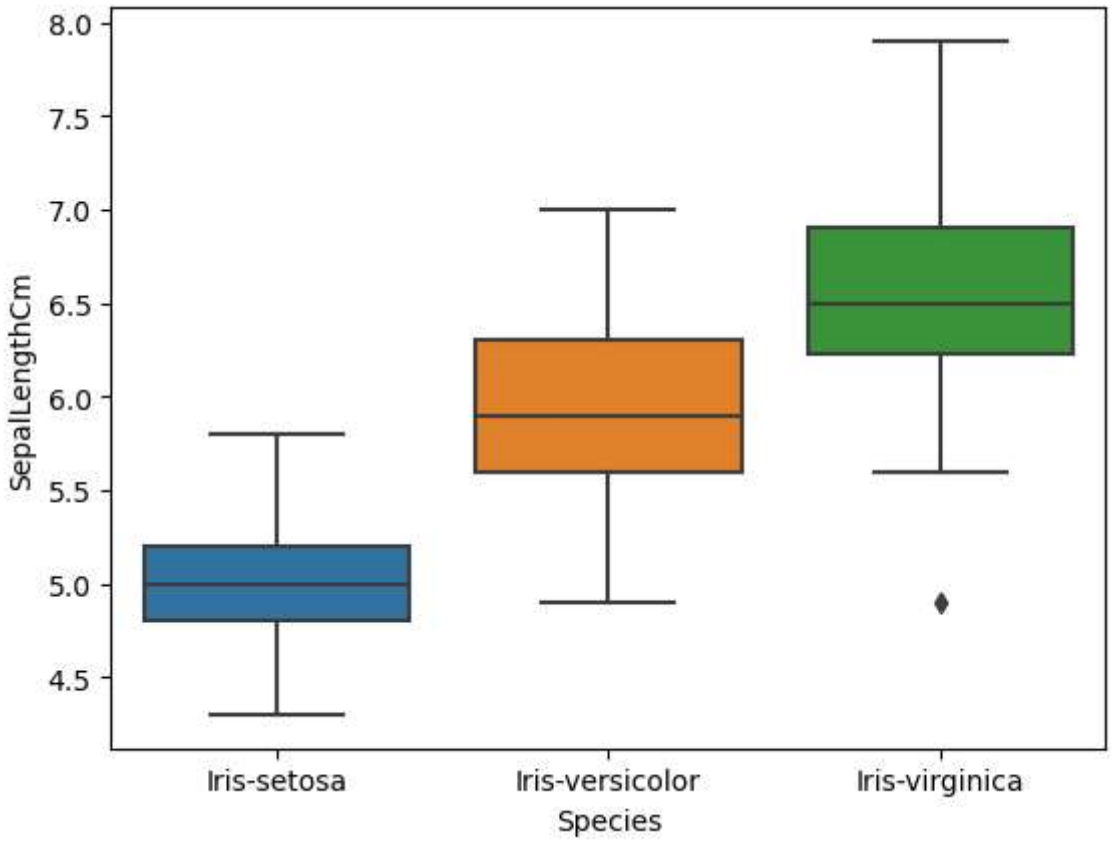
In [45]:
```
#box plots
sns.boxplot(x="Species",y="PetalLengthCm",data=irisdata)
plt.show()
```



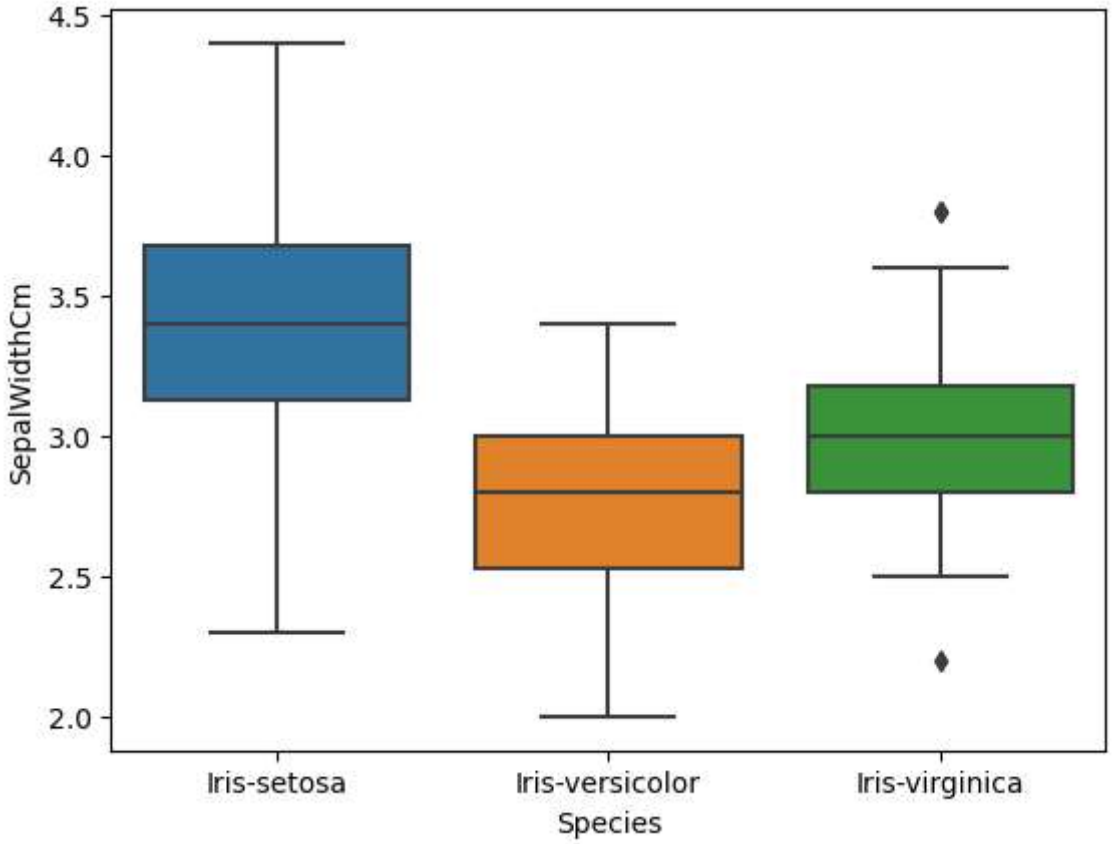
In [46]:
```
sns.boxplot(x="Species",y="PetalWidthCm",data=irisdata)
plt.show()
```



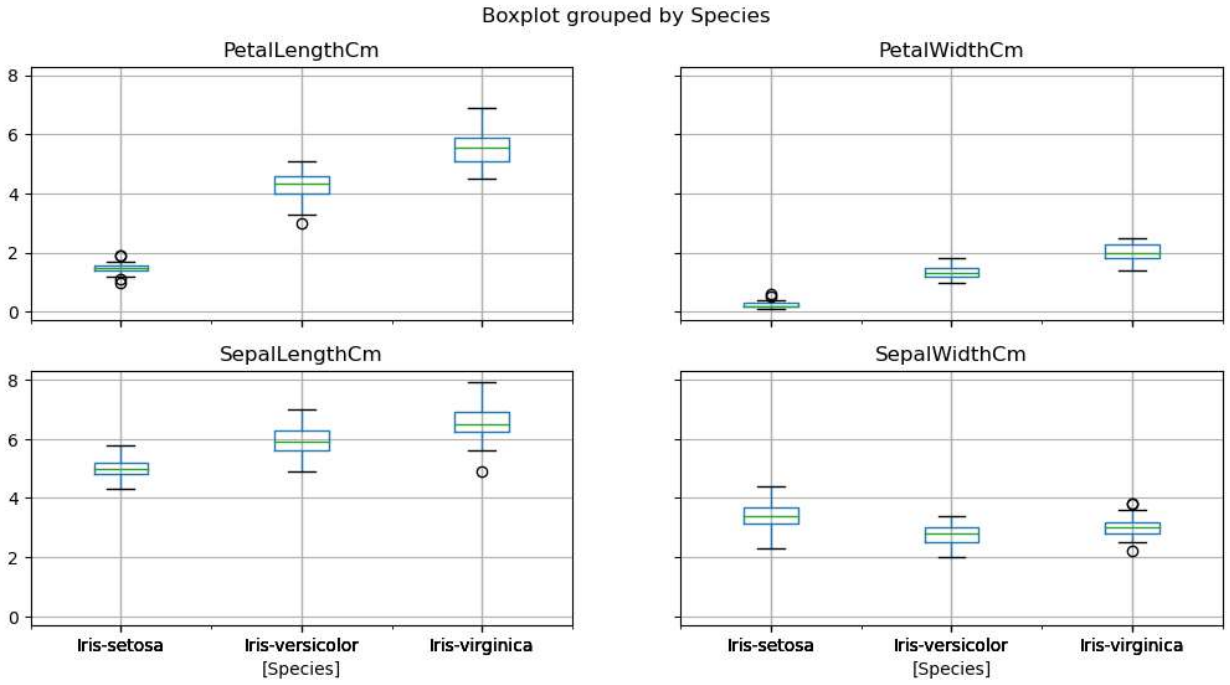
In [47]:
```
sns.boxplot(x="Species",y="SepalLengthCm",data=irisdata)
plt.show()
```



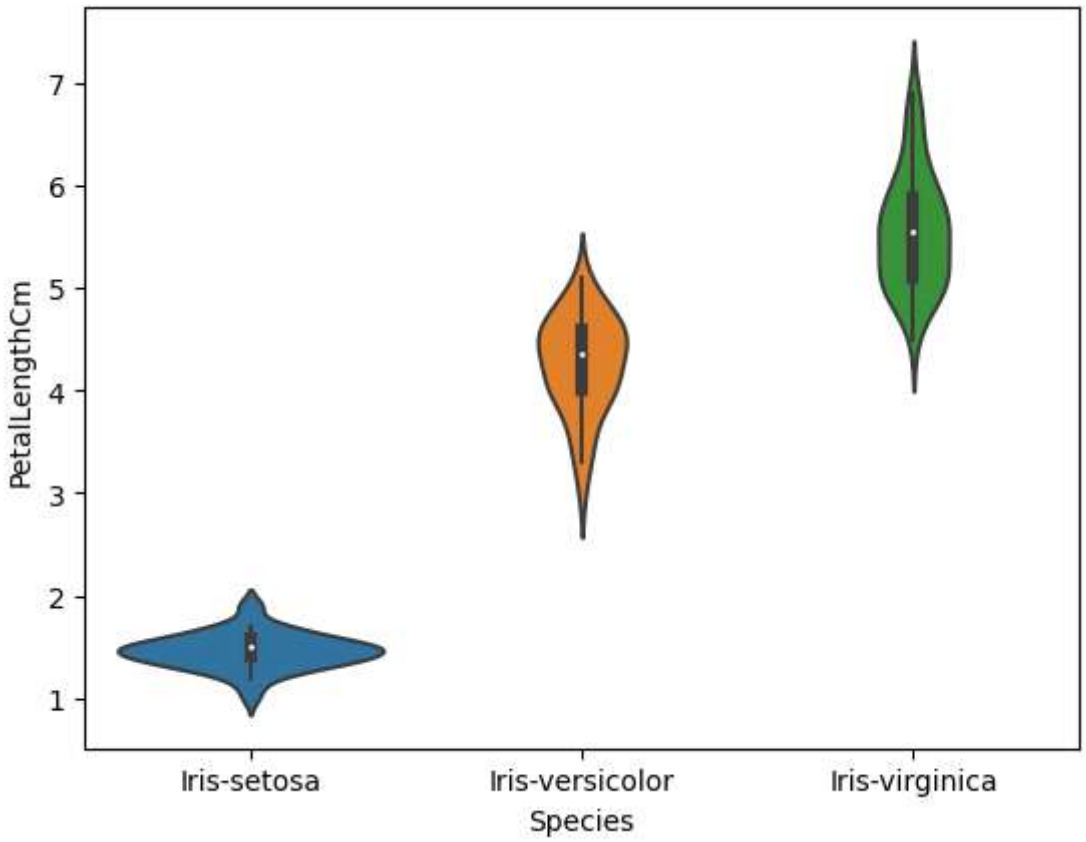
In [48]:
```
sns.boxplot(x="Species",y="SepalWidthCm",data=irisdata)
plt.show()
```



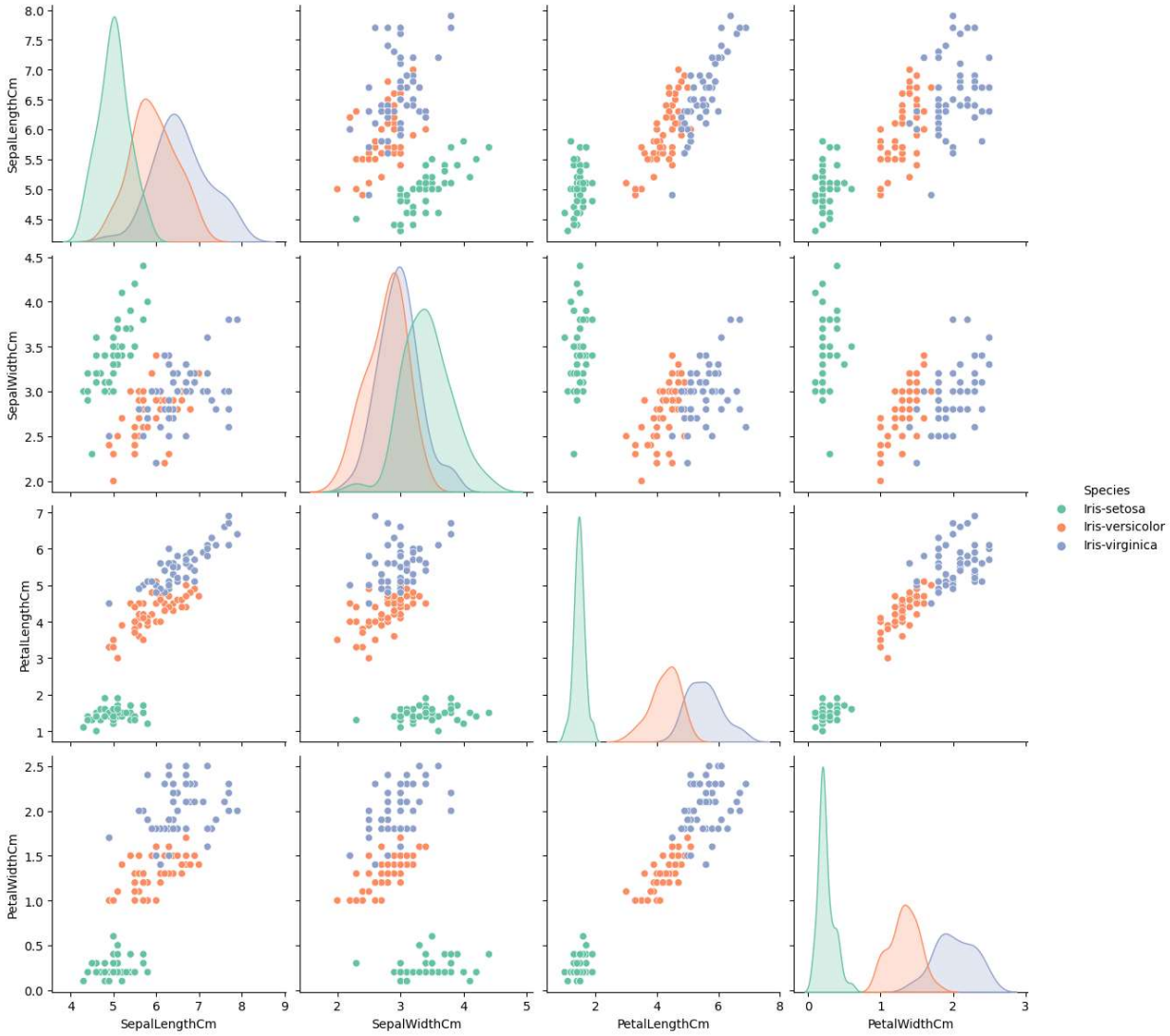
In [54]:
```
irisdata.drop("Id",axis=1).boxplot(by="Species",figsize=(12,6))
plt.show()
```



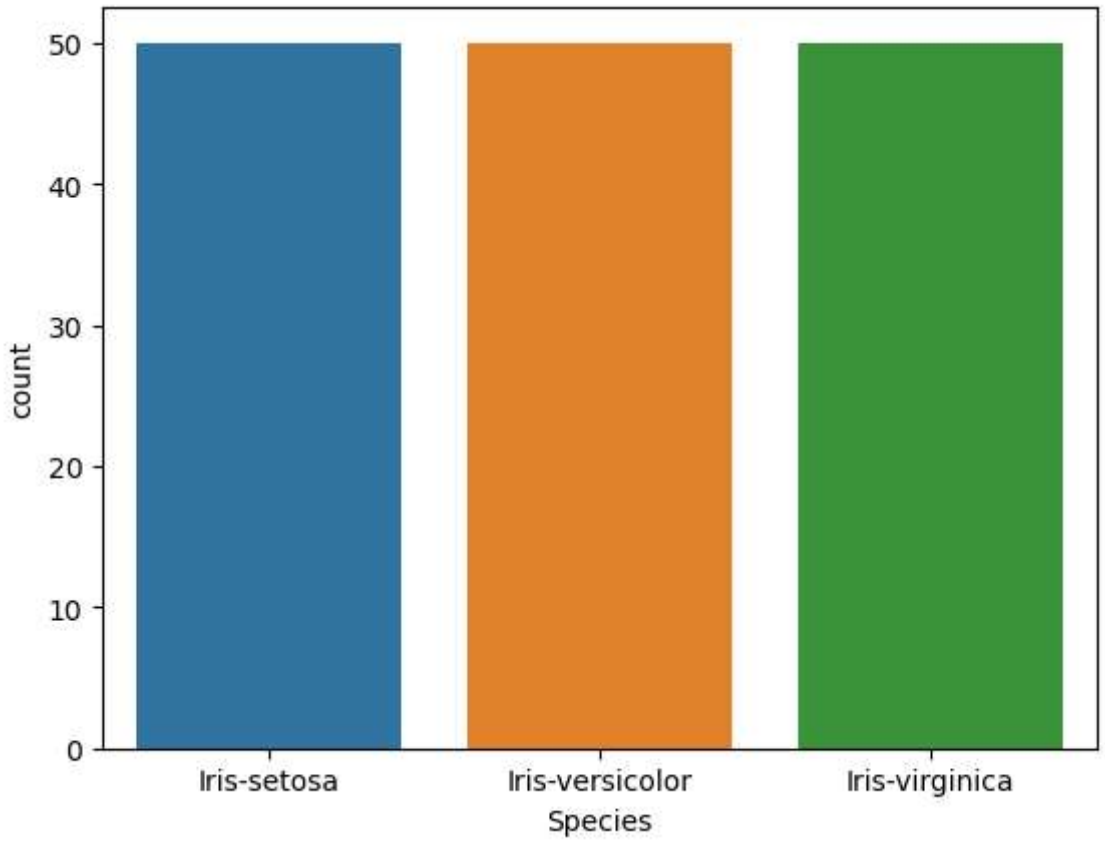
In [60]:
```
#violin plot
sns.violinplot(x="Species",y="PetalLengthCm",data=irisdata,height=10)
plt.show()
```



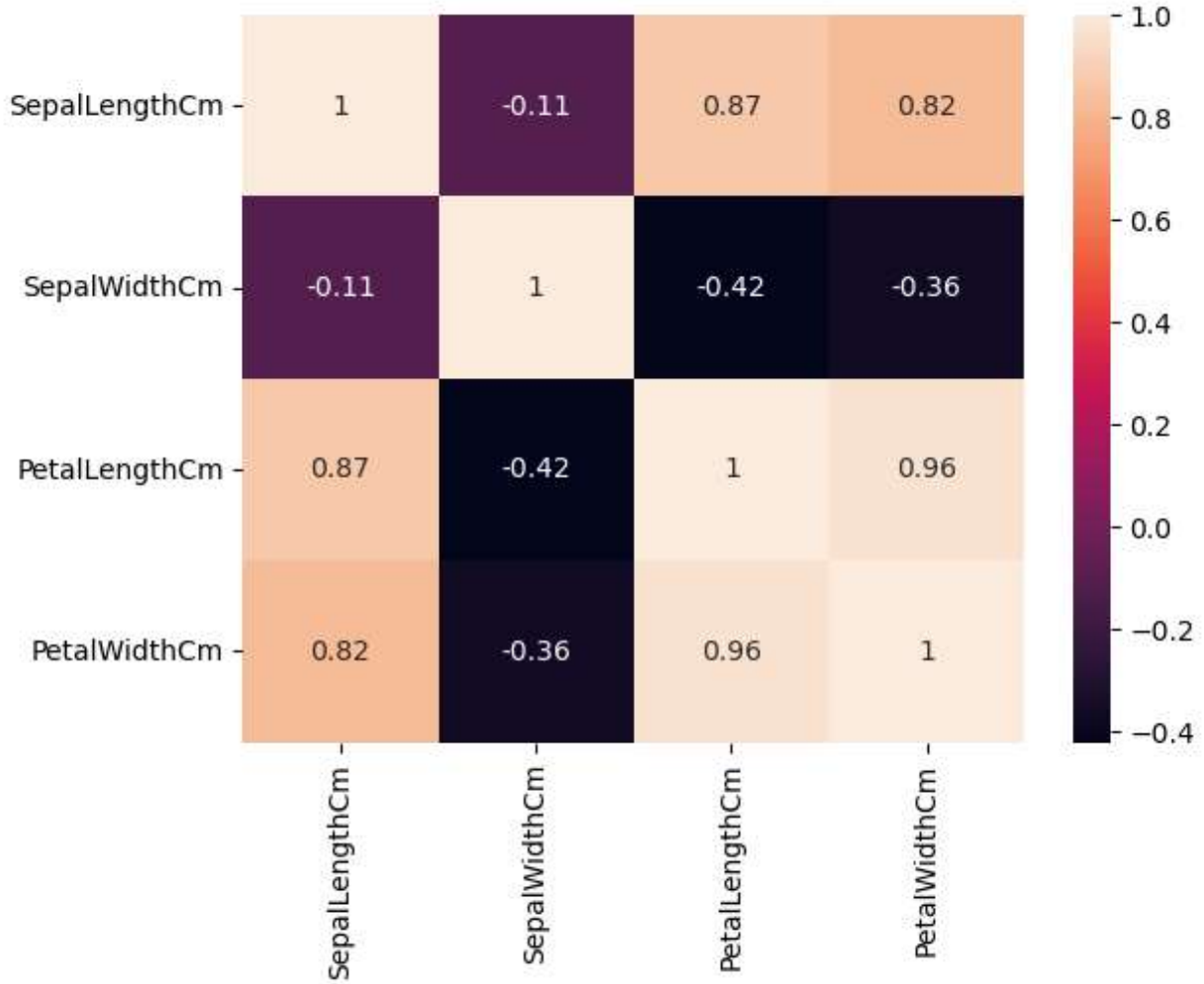
In [51]:
```
#pair plot
sns.pairplot(irisdata.drop("Id",axis=1),hue="Species",height=3,palette="Set2")
plt.show()
```



In [55]:
```
#countplot
sns.countplot(x='Species',data=irisdata)
plt.show()
```



In [59]:
```
#heatmap
sns.heatmap(irisdata.corr(method='pearson').drop(['Id'],axis=1).drop(['Id'],axis=0),ar
plt.show()
```


SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
1
-0.11
0.87
0.82
-0.11
1
-0.42
-0.36
0.87
-0.42
1
0.96
0.82
-0.36
0.96
1
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

In [ ]: