IRIS DATASET VISUALIZATION(SEABORN,MATPLOTLIB)

Import libraries

In [1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')  #this will ignore the warnings.it wont display w
```

Import Dataset

In [2]:
```
iris = pd.read_csv(r"D:\Naresh IT\Python Introduction\Datasets\Iris.csv")
```

In [3]:
```
iris
```

Out[3]:

	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 [4]:
```
iris.head()
```

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

In [5]:
```
iris.drop('Id',axis=1,inplace=True)
```

In [6]:
```
iris
```

Out[6]:

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

150 rows × 5 columns

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

Out[7]:

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

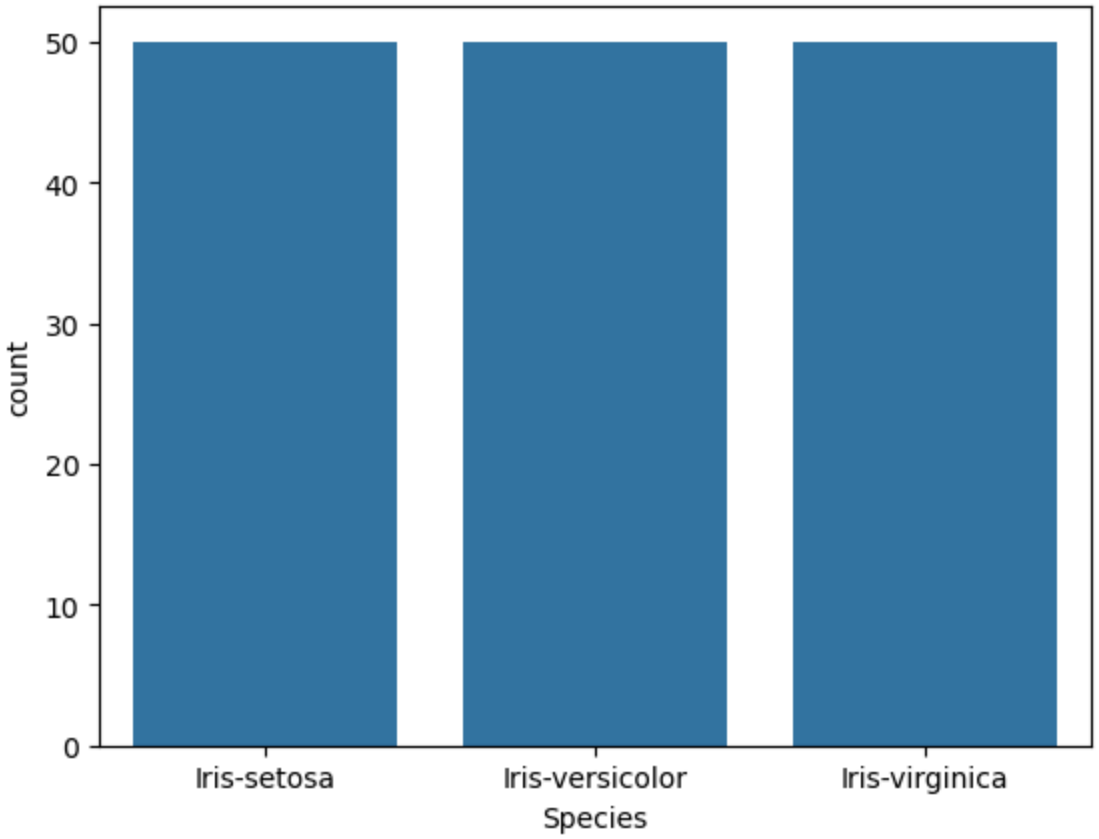
In [8]:
```
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   SepalLengthCm  150 non-null    float64
 1   SepalWidthCm   150 non-null    float64
 2   PetalLengthCm  150 non-null    float64
 3   PetalWidthCm   150 non-null    float64
 4   Species        150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [9]:
```
iris["Species"].value_counts()
```

Out[9]:
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

In [10]:
```
sns.countplot(x = 'Species', data=iris)
plt.show()
```

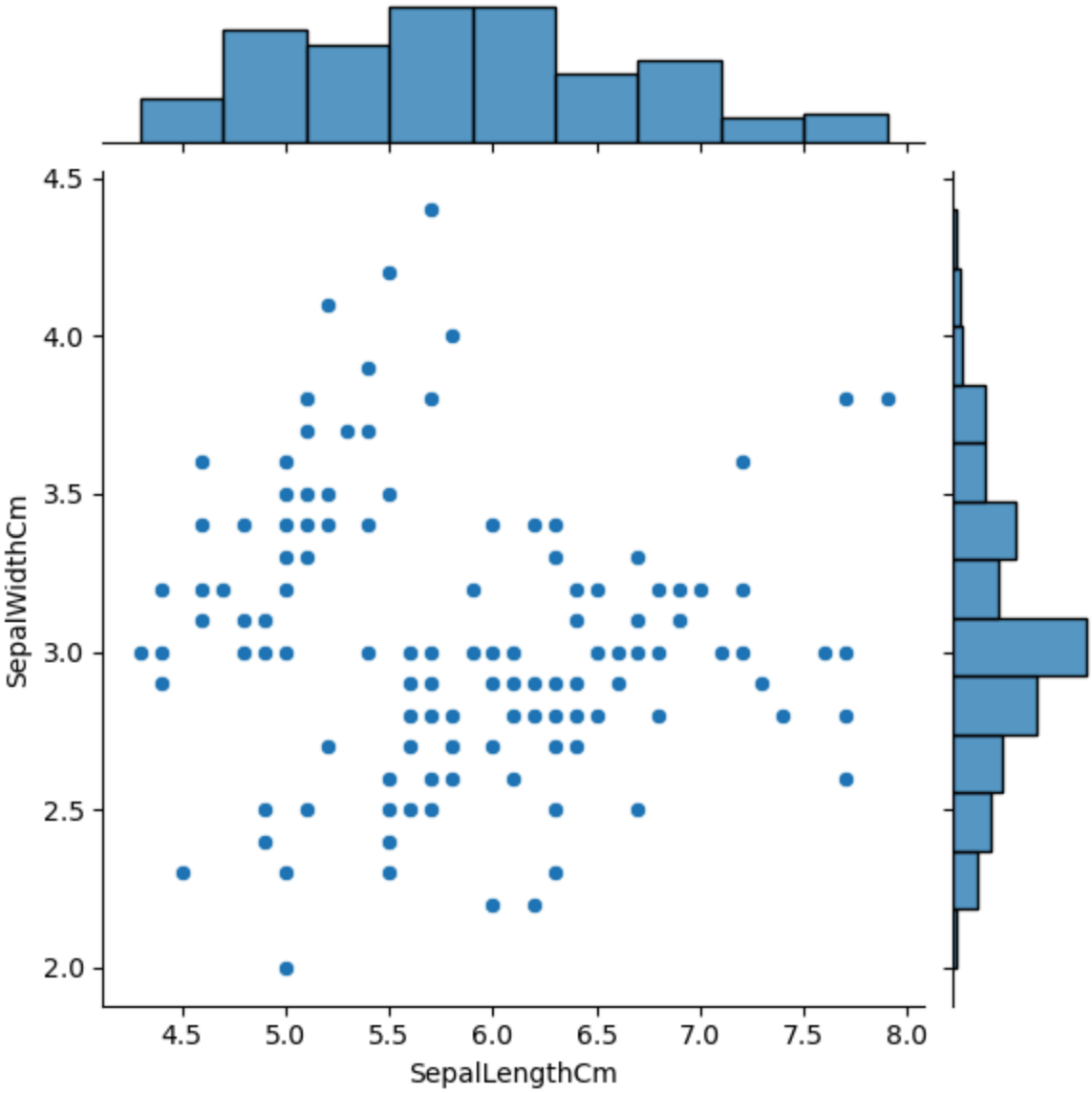


In [11]:
```
iris.head()
```

Out[11]:

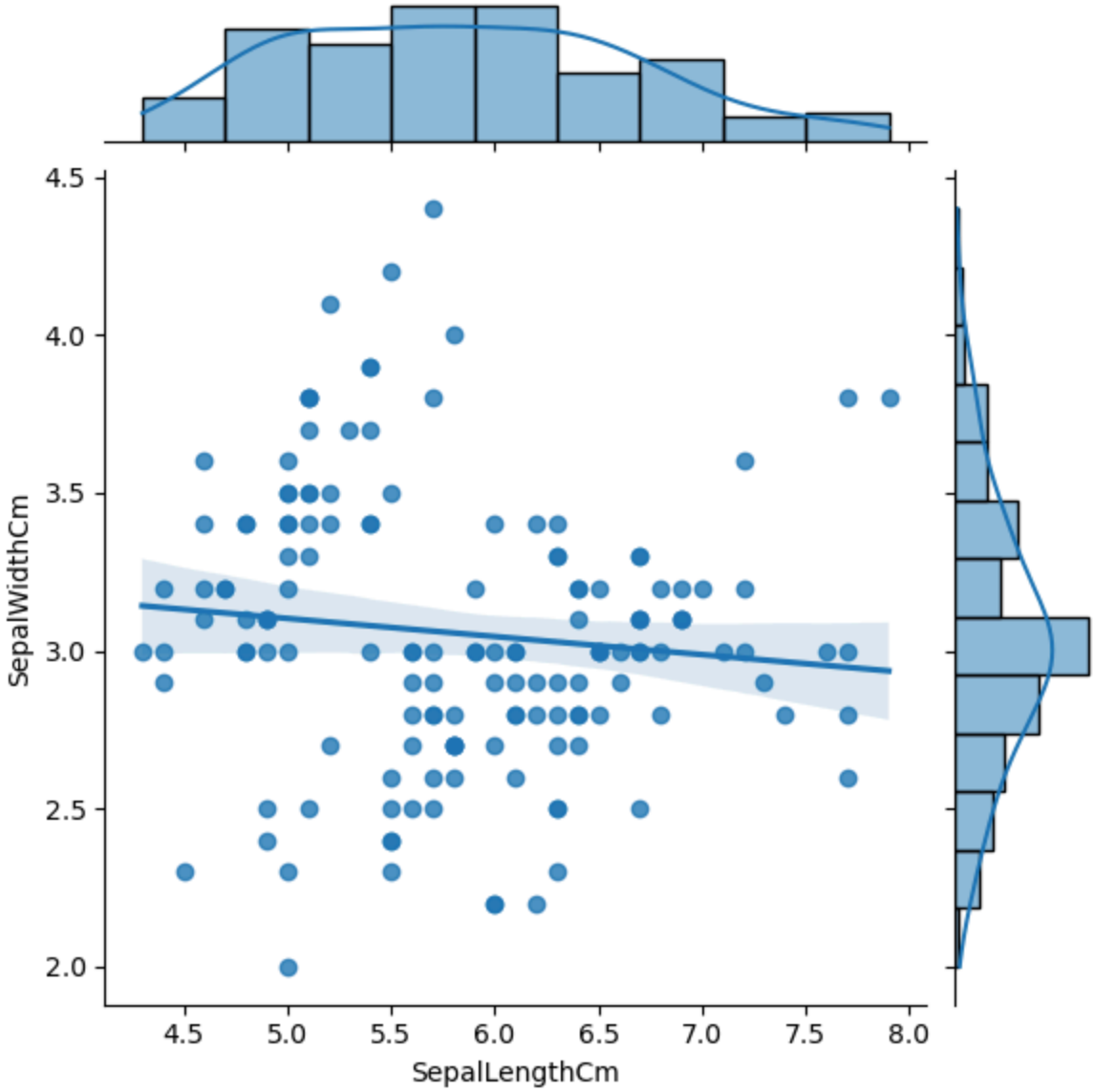
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [12]:
```
fig = sns.jointplot(x='SepalLengthCm', y='SepalWidthCm',data=iris)
```



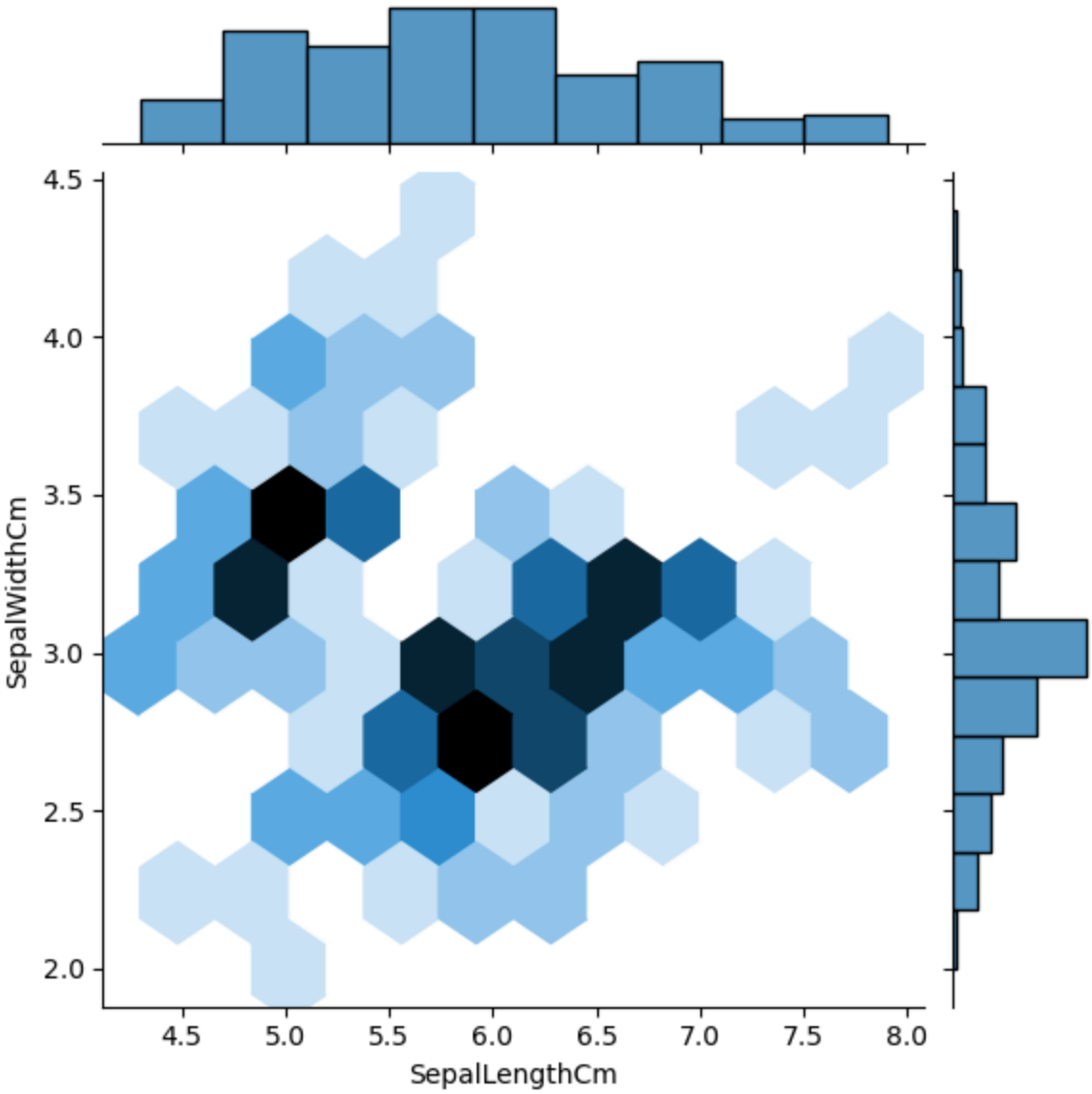
In [13]:
```
sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',data=iris,kind='reg')
```

Out[13]:
```
<seaborn.axisgrid.JointGrid at 0x21330ea9250>
```

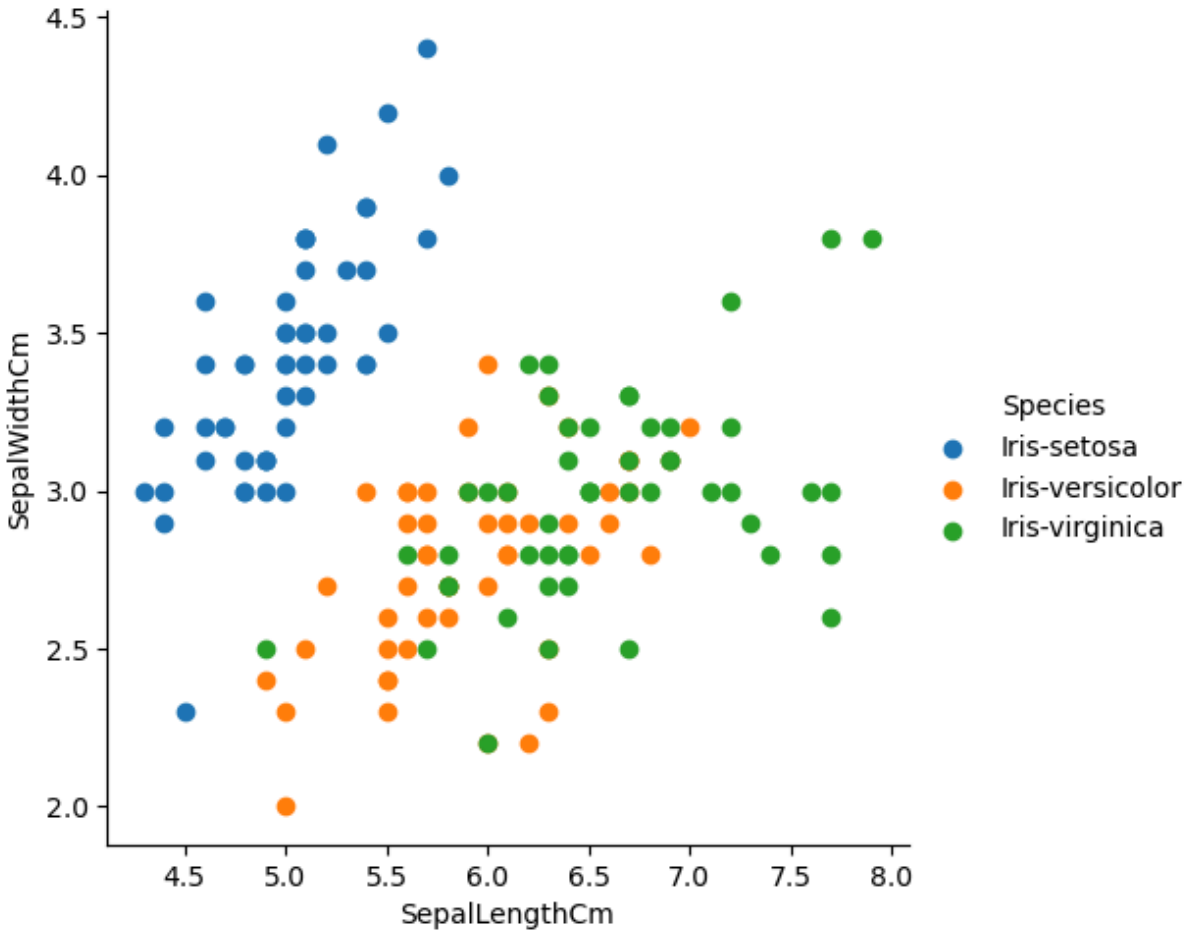


In [14]:
```
sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',data=iris,kind='hex')
```

Out[14]: <seaborn.axisgrid.JointGrid at 0x213315b0e60>



In [15]:
```
import matplotlib.pyplot as plt
%matplotlib inline

sns.FacetGrid(iris,hue='Species',height=5)\
.map(plt.scatter,'SepalLengthCm','SepalWidthCm')\
.add_legend()
plt.show()
```

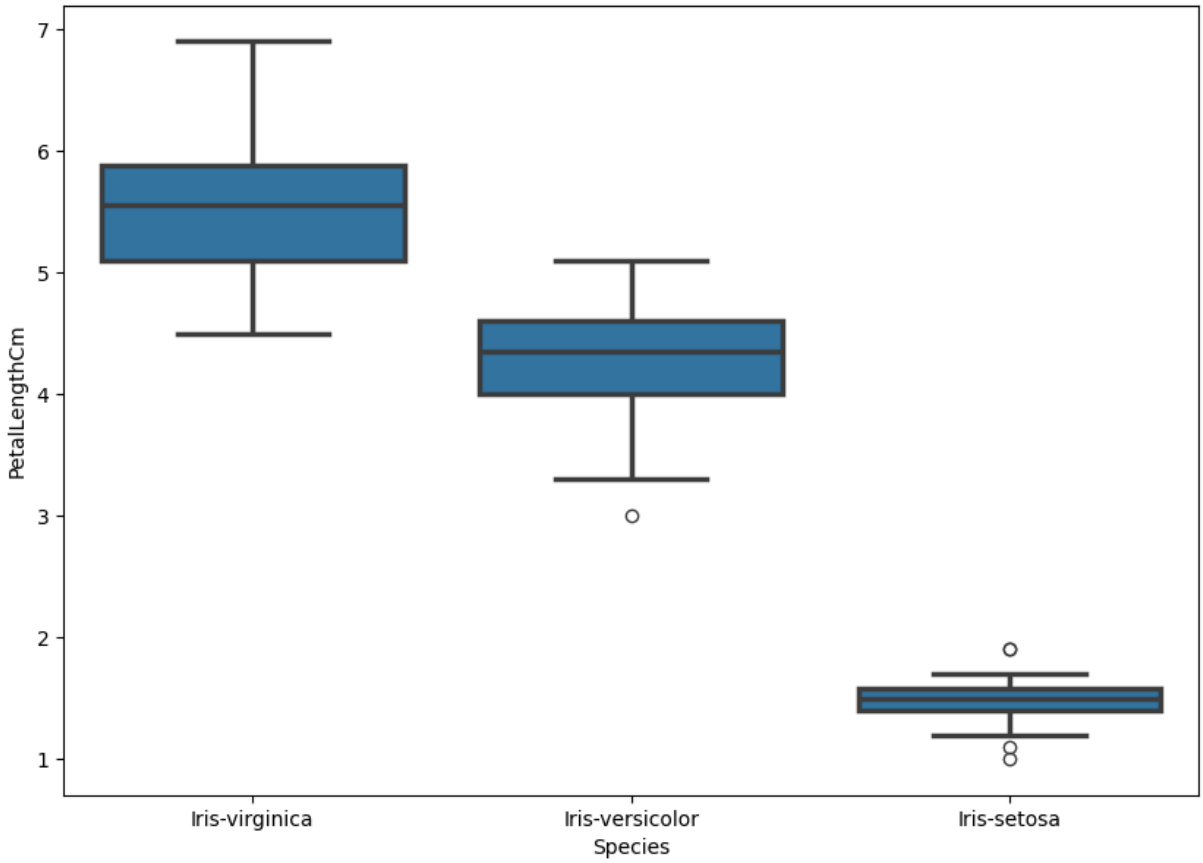


In [16]:
```
iris.head()
```

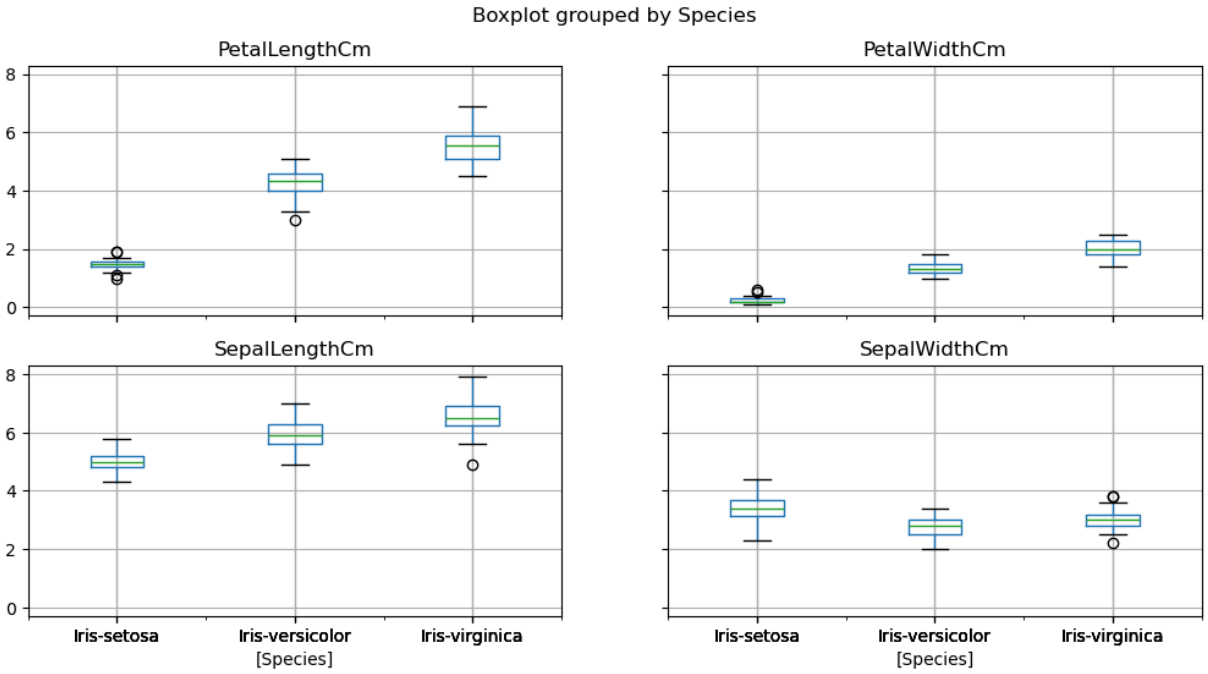
Out[16]:

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

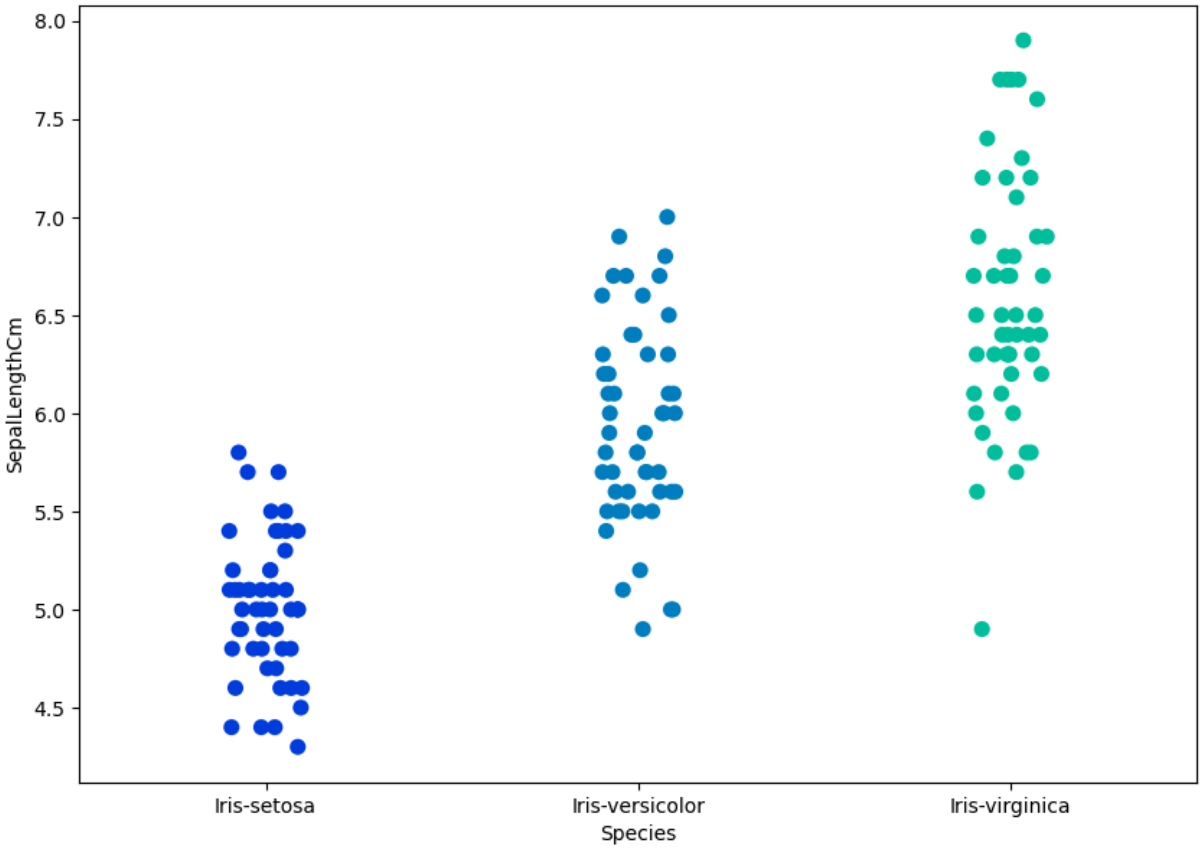
In [17]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='PetalLengthCm',data=iris,order=['Iris-virginica','Ir
plt.show()
```



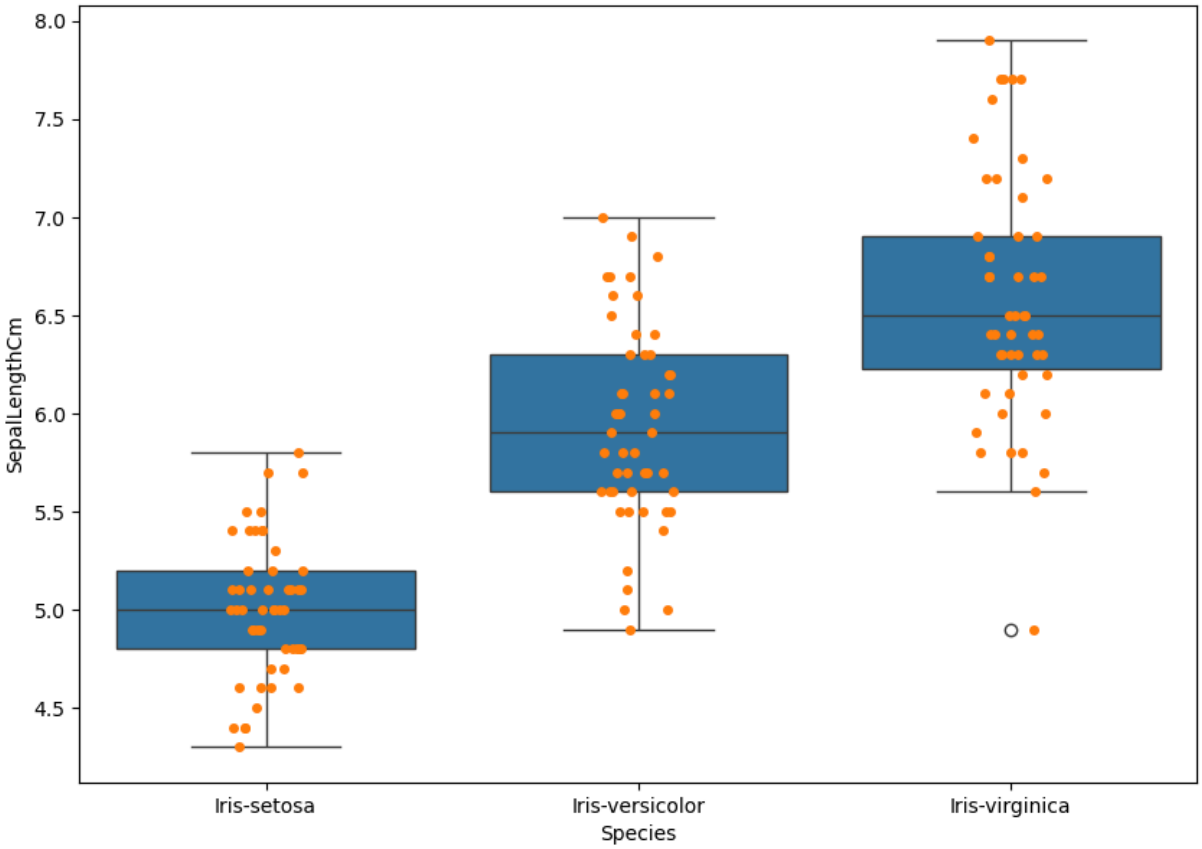
In [18]:
```
iris.boxplot(by="Species", figsize=(12, 6))
plt.show()
```



In [19]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edgecolor='gr
plt.show()
```

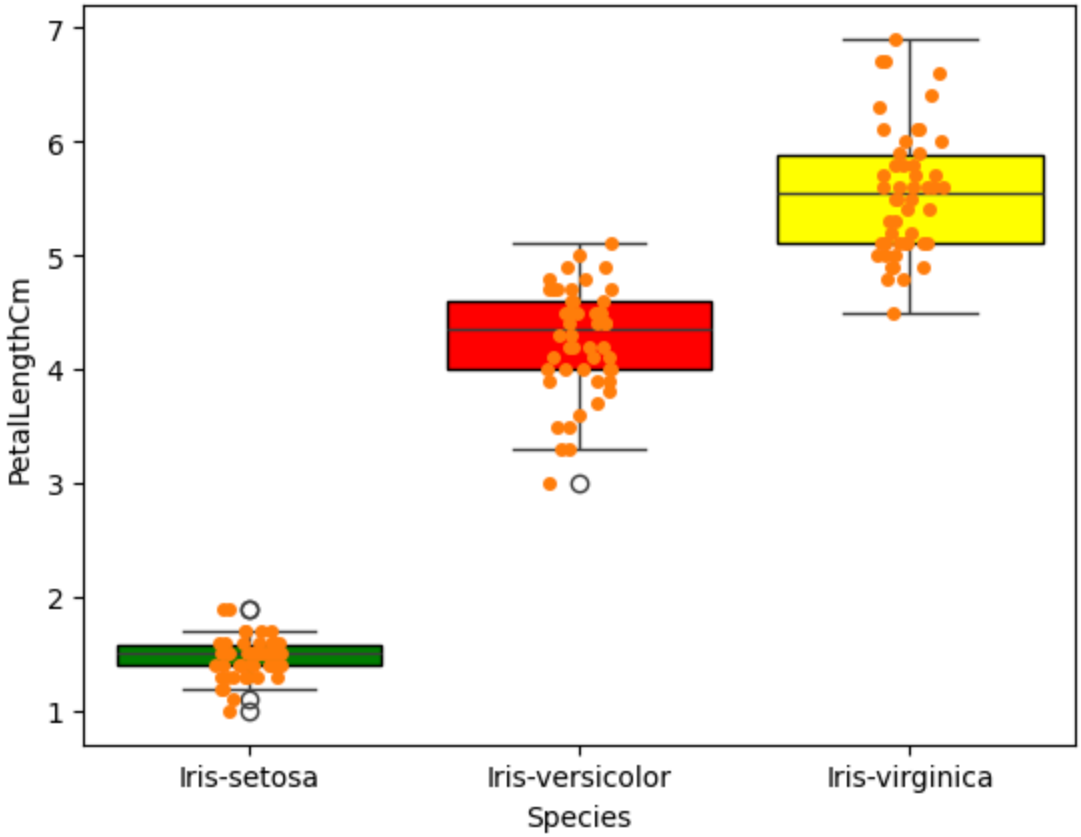


In [20]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='SepalLengthCm',data=iris)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edgecolor='gr
plt.show()
```

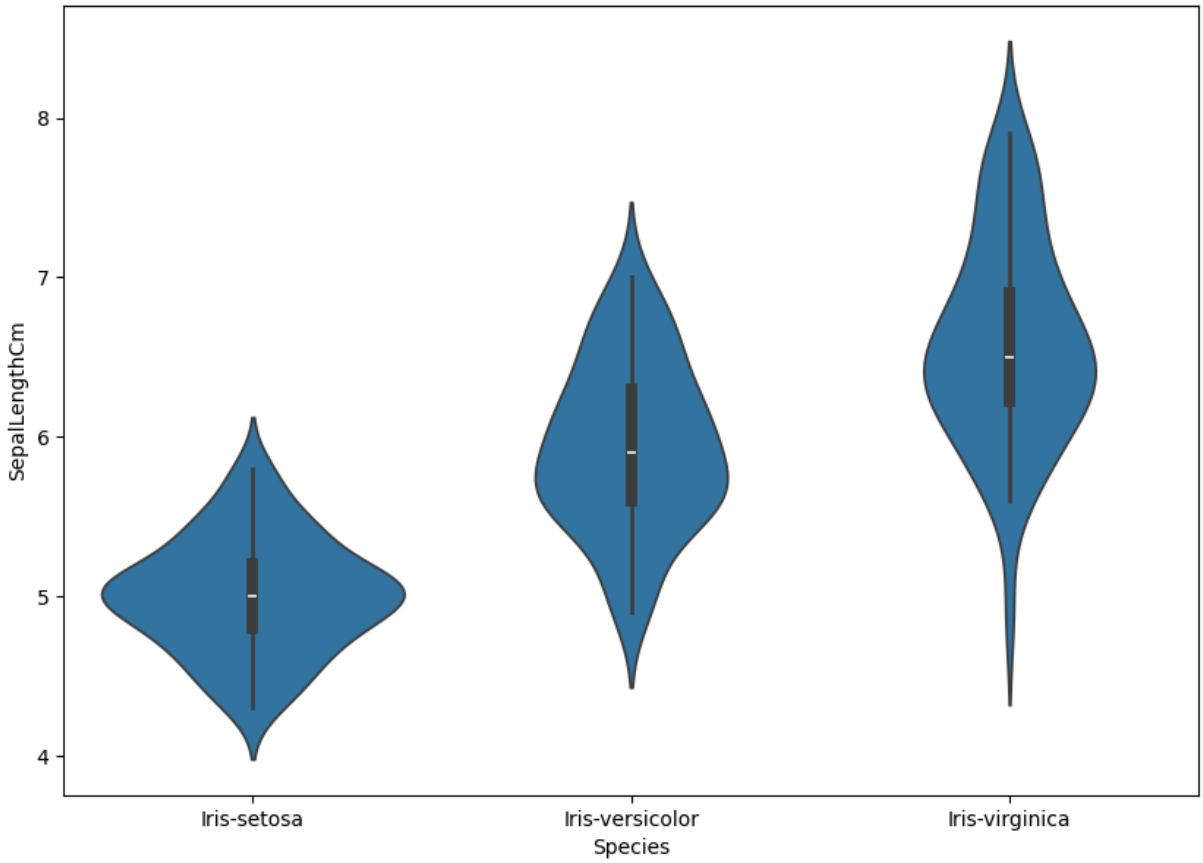


In [25]:
```
# Create the figure and axes
fig, ax = plt.subplots()

# Create the boxplot
sns.boxplot(x="Species", y="PetalLengthCm", data=iris, ax=ax)

# Create the strip plot
sns.stripplot(x="Species", y="PetalLengthCm", data=iris, jitter=True, edgecolor="gr

# Check number of boxes (patches) in the plot
print(f"Number of patches: {len(ax.patches)}")

# Modify specific boxes, if they exist
if len(ax.patches) >= 3:
    ax.patches[2].set_facecolor("yellow")
    ax.patches[2].set_edgecolor("black")

    ax.patches[1].set_facecolor("red")
    ax.patches[1].set_edgecolor("black")

    ax.patches[0].set_facecolor("green")
    ax.patches[0].set_edgecolor("black")
else:
    print("Not enough patches to modify.")

# Show the plot
plt.show()
```

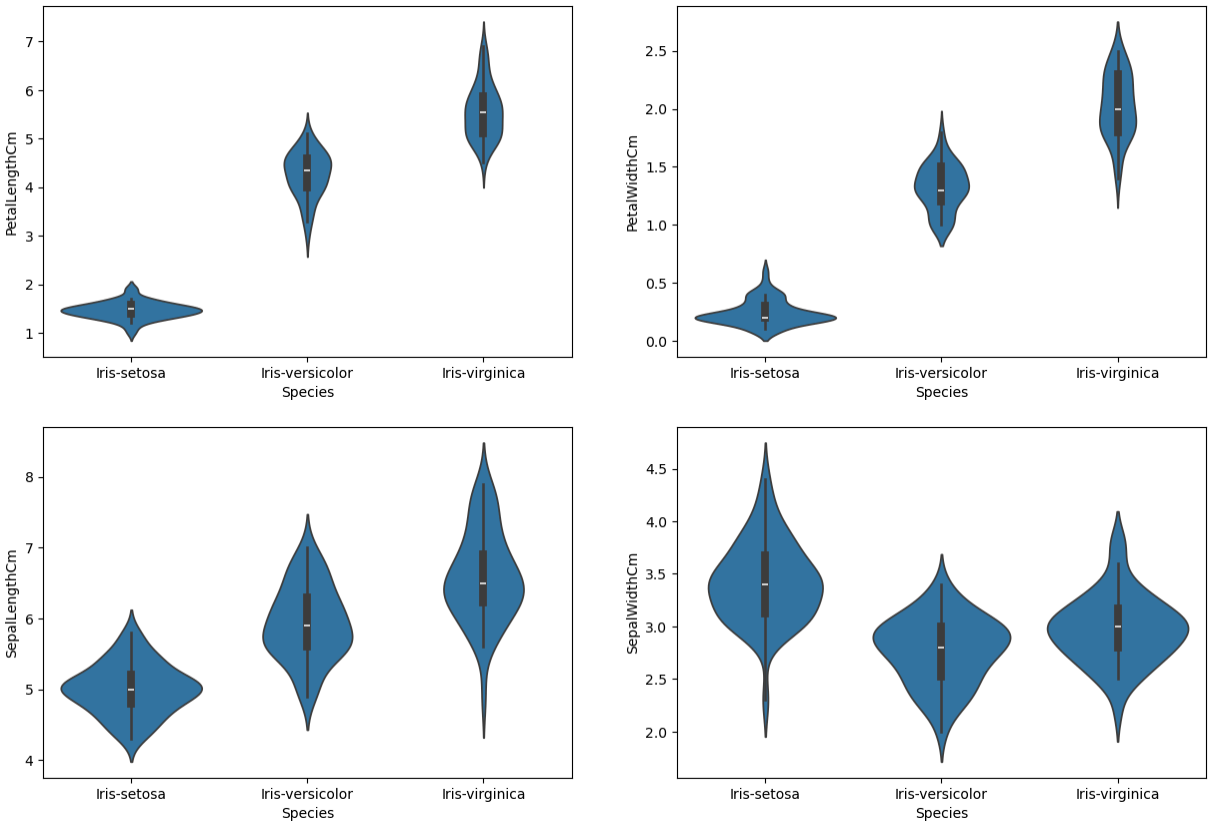
```
Number of patches: 3
```



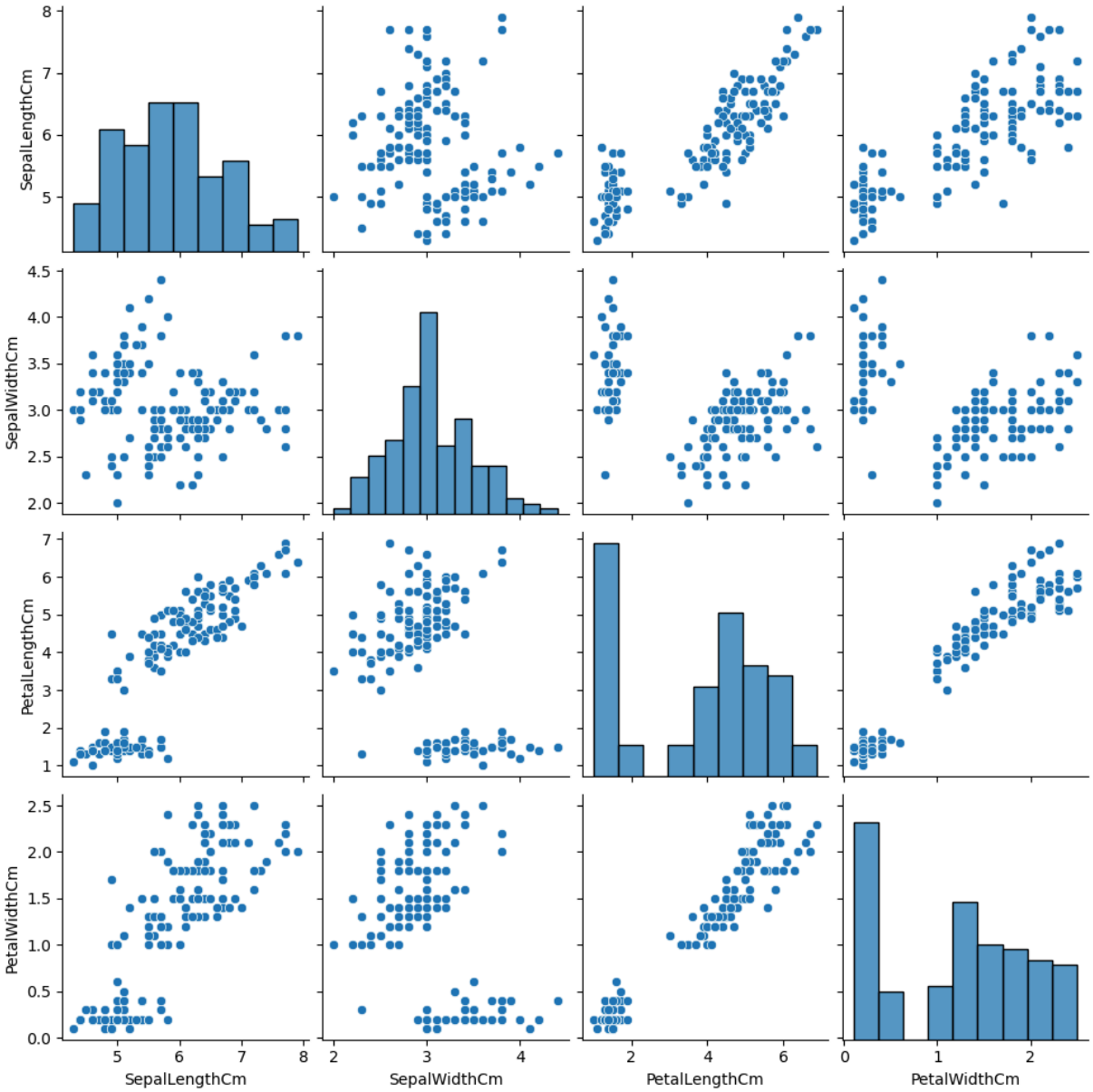
In [30]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.violinplot(x='Species',y='SepalLengthCm',data=iris)
plt.show()
```



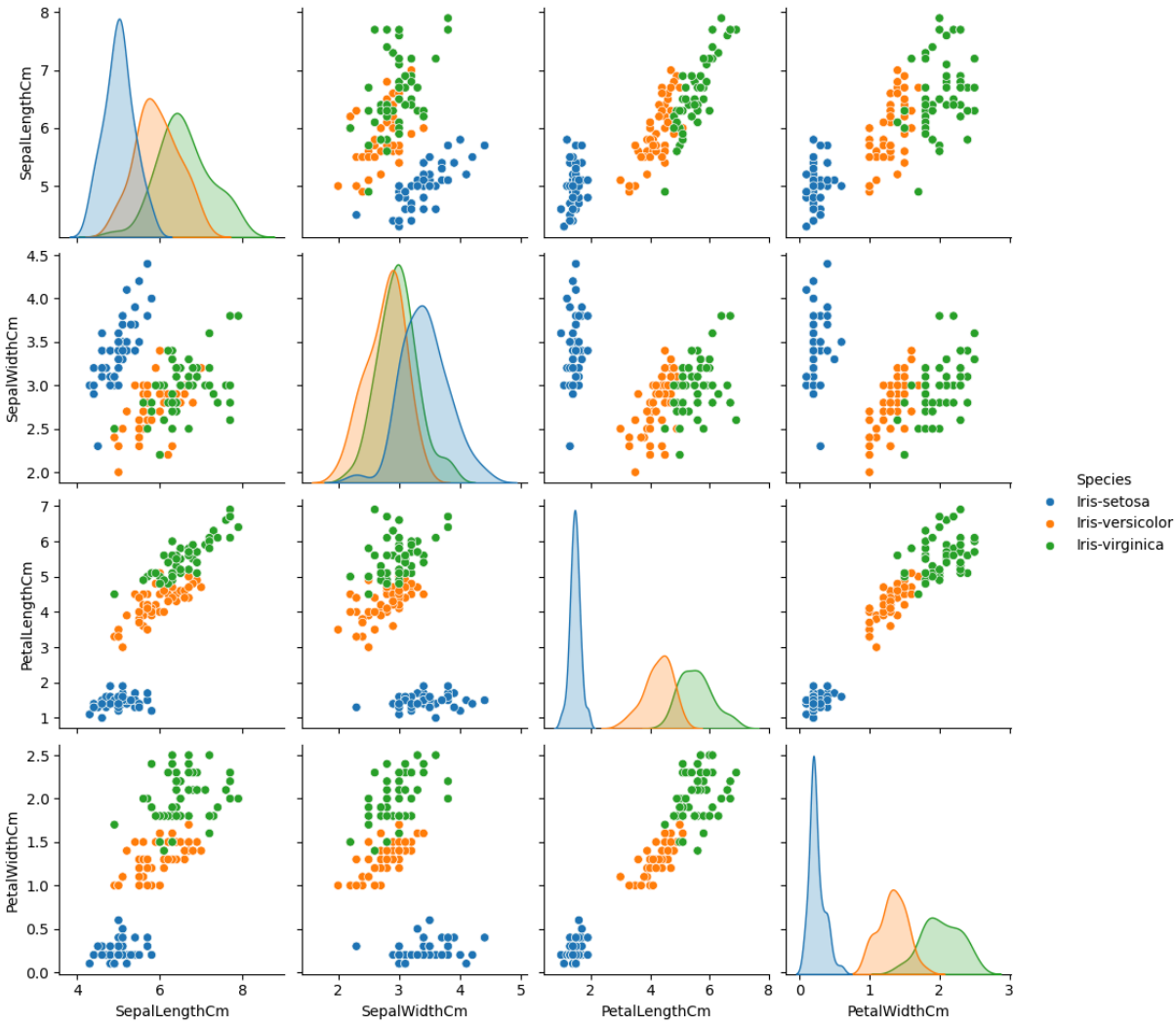
In [33]:
```
plt.figure(figsize=(15,10))
plt.subplot(2,2,1)
sns.violinplot(x='Species',y='PetalLengthCm',data=iris)
plt.subplot(2,2,2)
sns.violinplot(x='Species',y='PetalWidthCm',data=iris)
plt.subplot(2,2,3)
sns.violinplot(x='Species',y='SepalLengthCm',data=iris)
plt.subplot(2,2,4)
sns.violinplot(x='Species',y='SepalWidthCm',data=iris)
plt.show()
```



In [35]:
```
sns.pairplot(data=iris,kind='scatter')
plt.show()
```

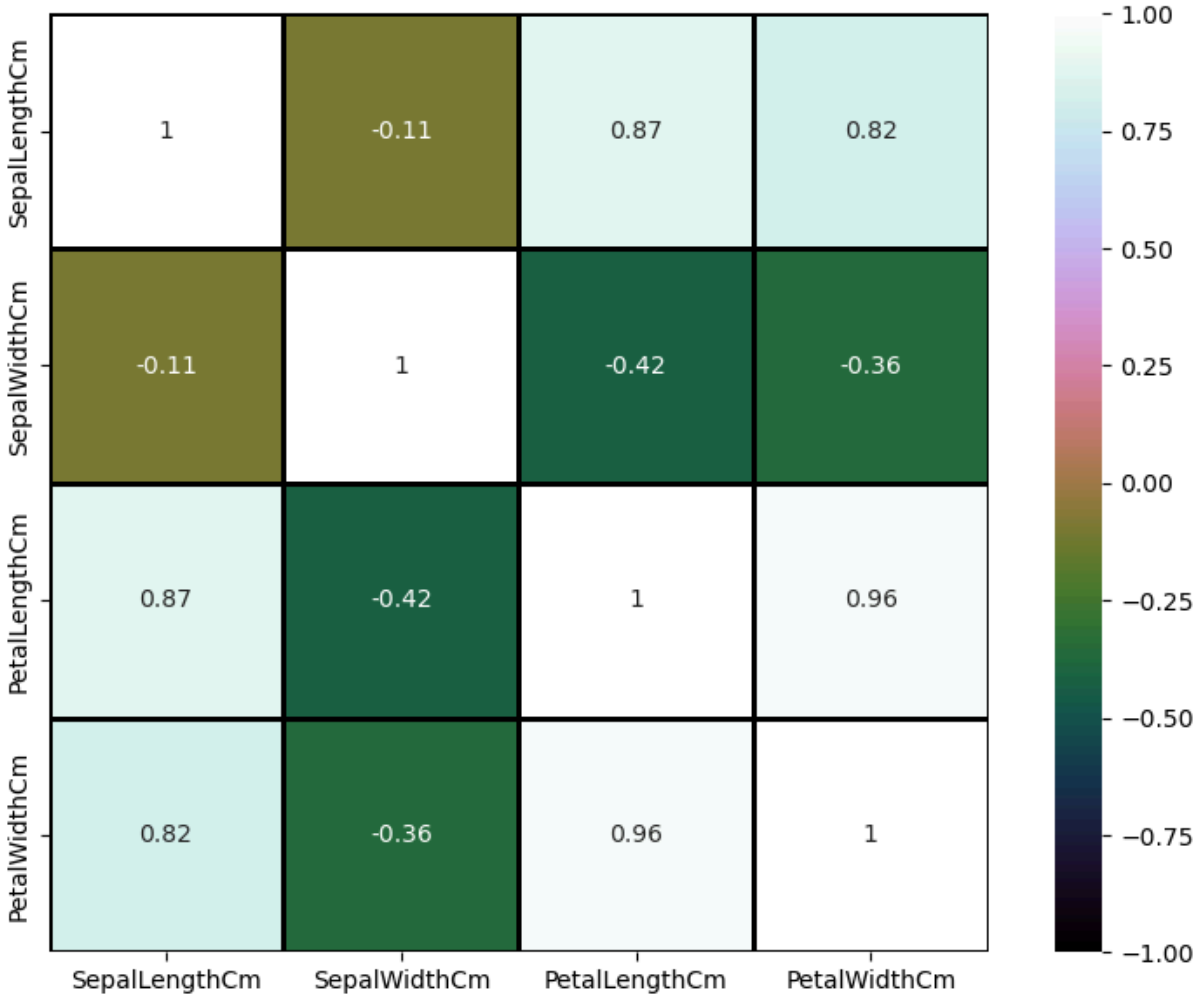


In [38]:
```
sns.pairplot(iris,hue='Species')
plt.show()
```

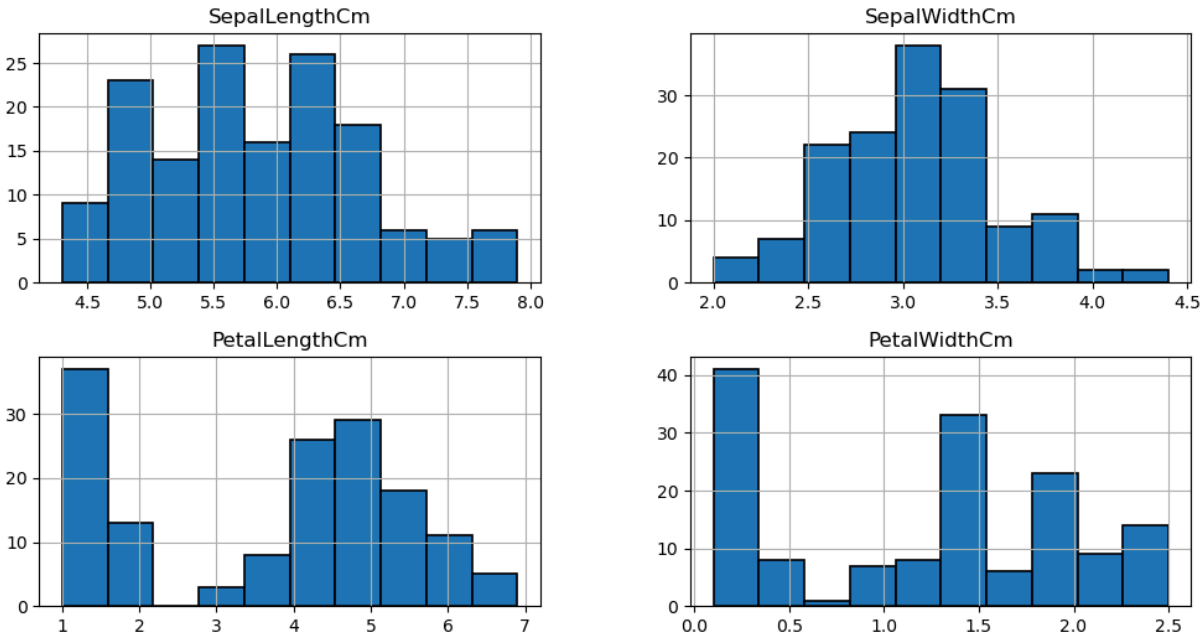


In [40]:
```
# Compute correlation (only numerical columns)
corr_matrix = iris.corr(numeric_only=True)

# Create the figure
fig, ax = plt.subplots(figsize=(10, 7))  # Set figure size correctly

# Create the heatmap and use 'ax' to specify the axis
sns.heatmap(corr_matrix, annot=True, cmap="cubehelix", linewidths=1, linecolor="k",
            square=True, mask=False, vmin=-1, vmax=1, cbar_kws={"orientation": "ver
            cbar=True, ax=ax)

# Show the plot
plt.show()
```

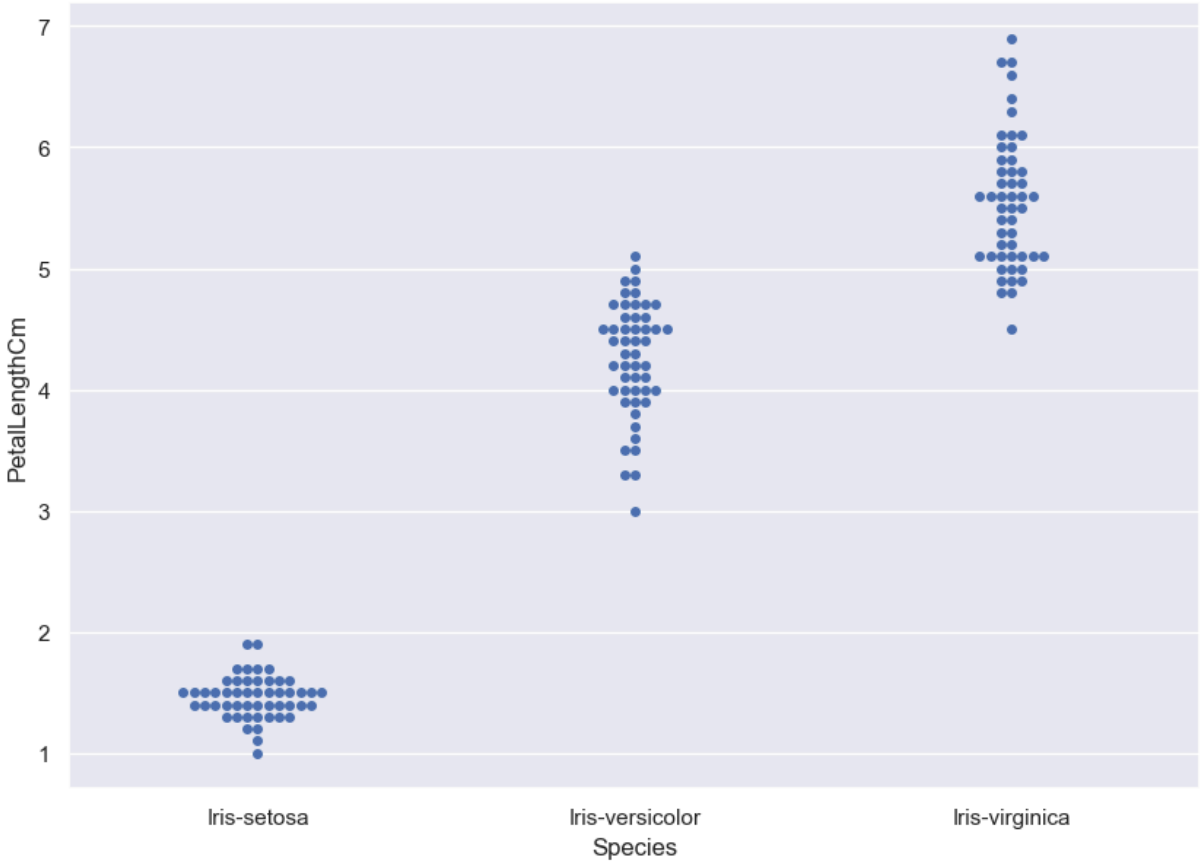
```
<Figure size 1000x700 with 0 Axes>
```



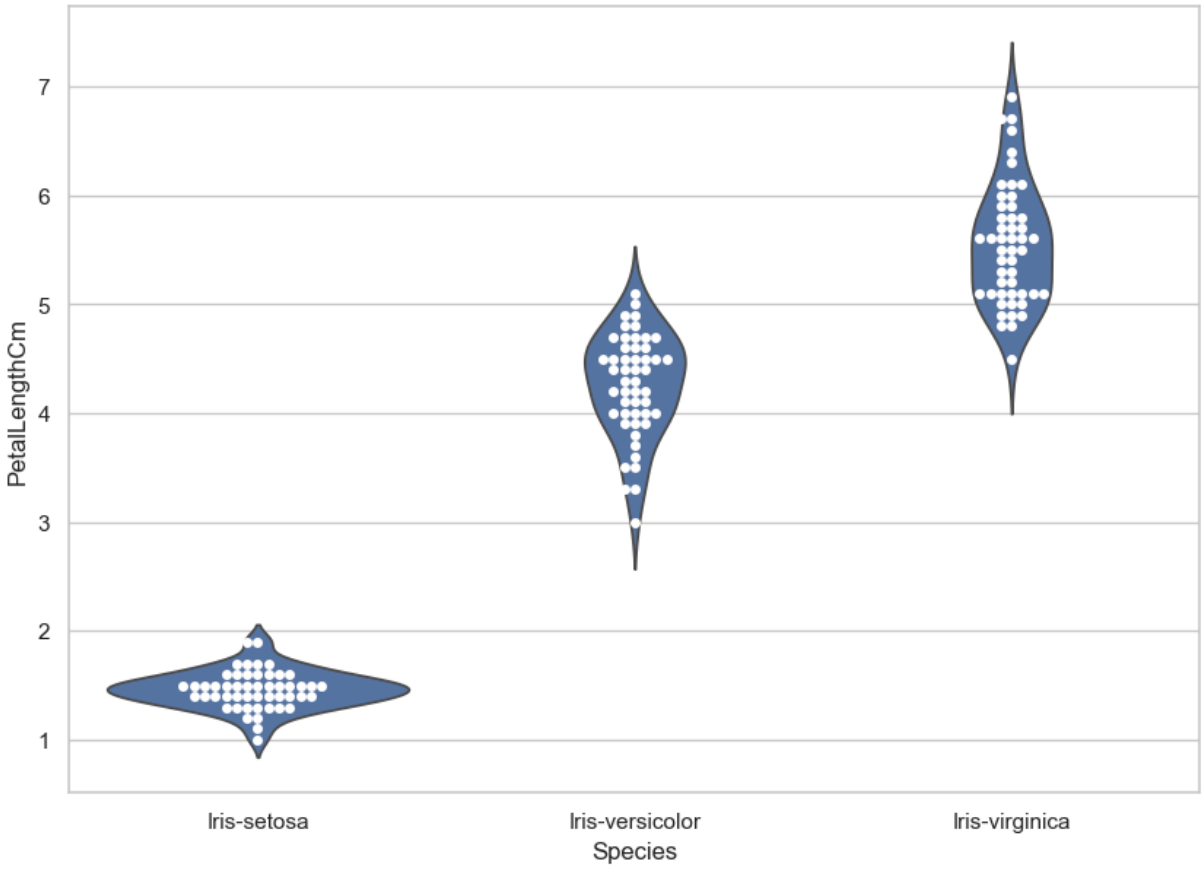
In [43]:
```
iris.hist(edgecolor='black', linewidth=1.2)
fig=plt.gcf()
fig.set_size_inches(12,6)
plt.show()
```



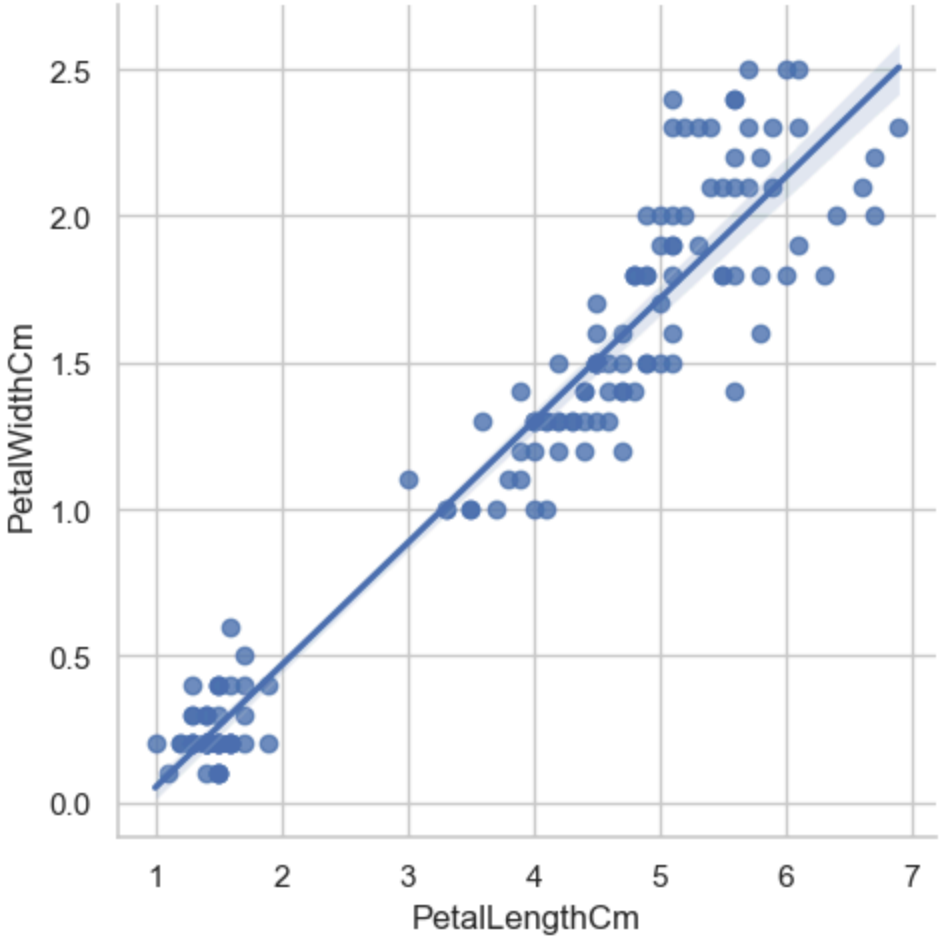
In [44]:
```
sns.set(style="darkgrid")
fig=plt.gcf()
fig.set_size_inches(10,7)
fig = sns.swarmplot(x="Species", y="PetalLengthCm", data=iris)
plt.show()
```



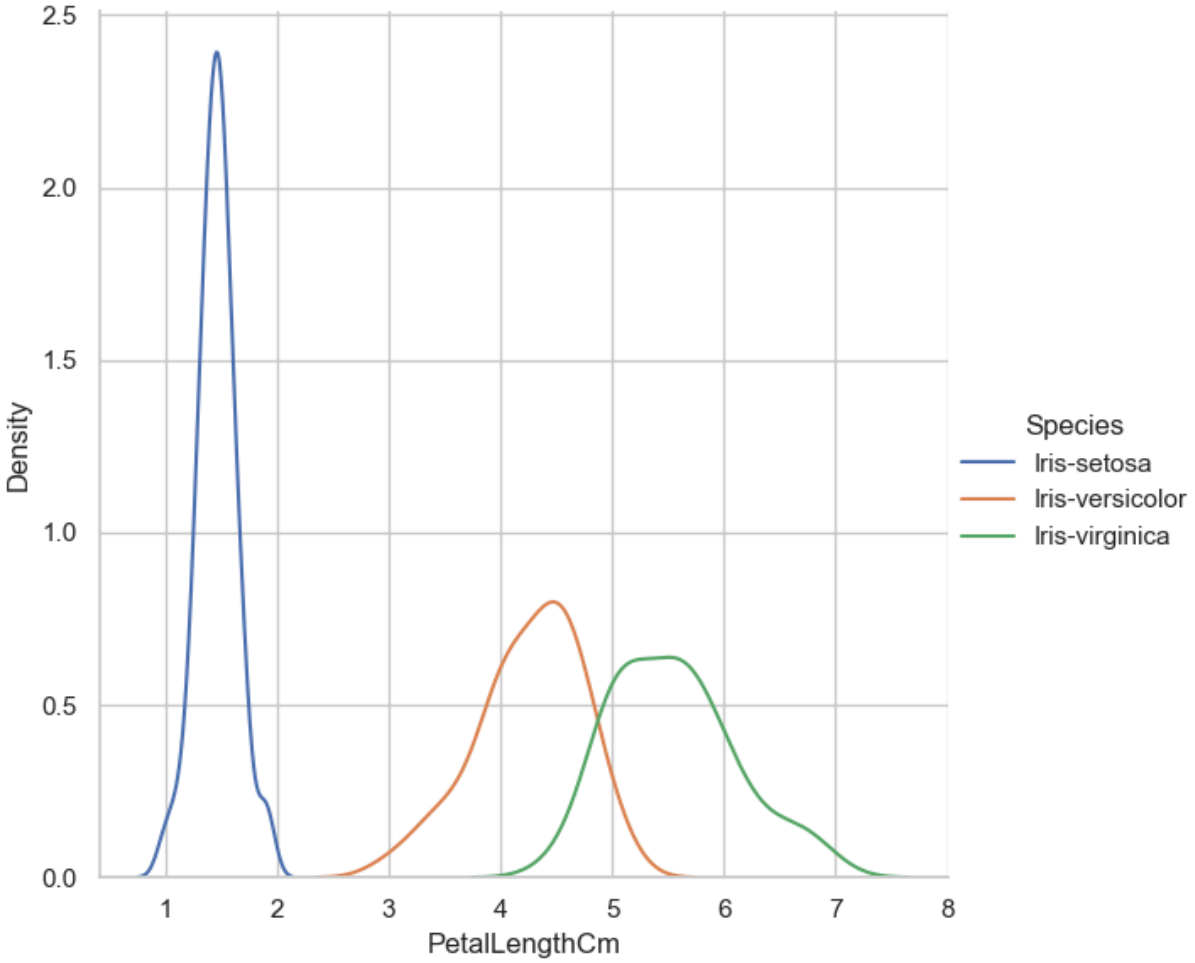
In [45]:
```
sns.set(style="whitegrid")
fig=plt.gcf()
fig.set_size_inches(10,7)
ax = sns.violinplot(x="Species", y="PetalLengthCm", data=iris, inner=None)
ax = sns.swarmplot(x="Species", y="PetalLengthCm", data=iris,color="white", edgecol
plt.show()
```



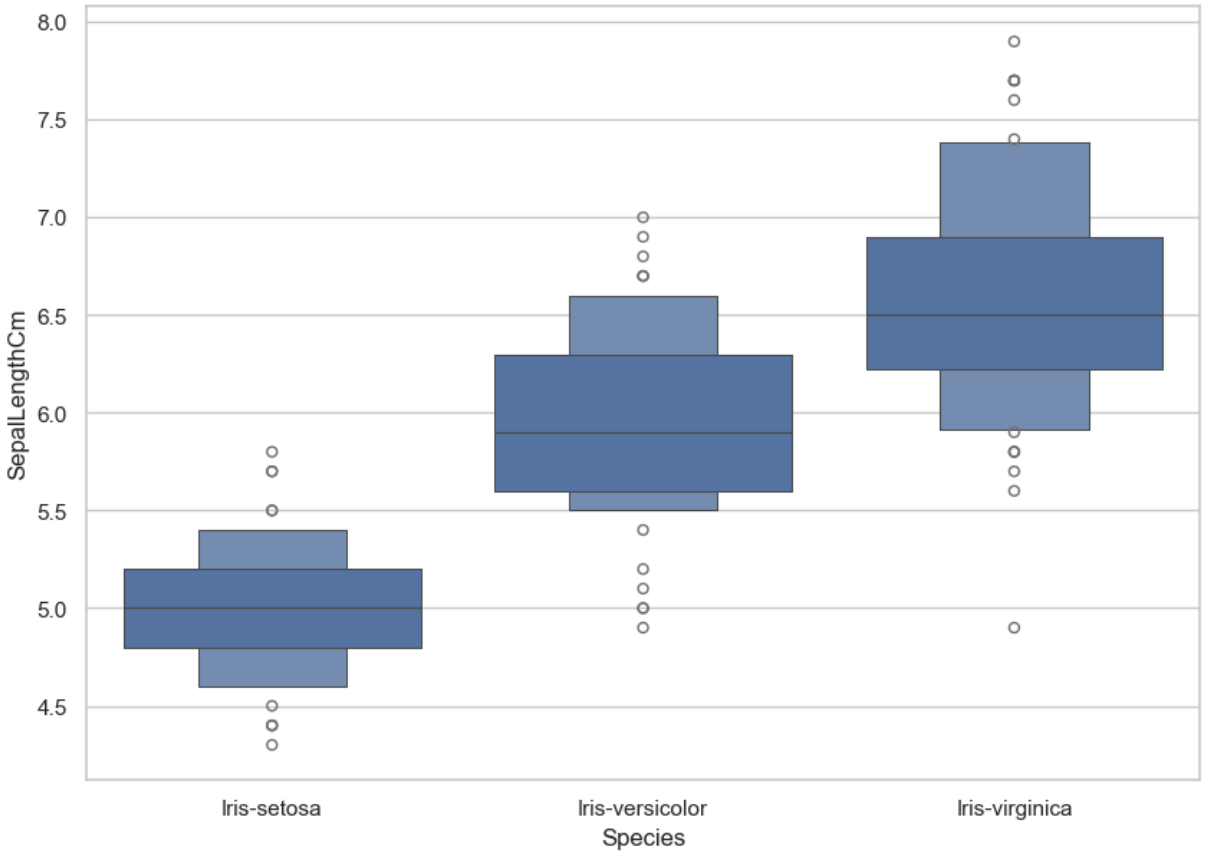
In [48]:
```
fig=sns.lmplot(x="PetalLengthCm", y="PetalWidthCm",data=iris)
plt.show()
```



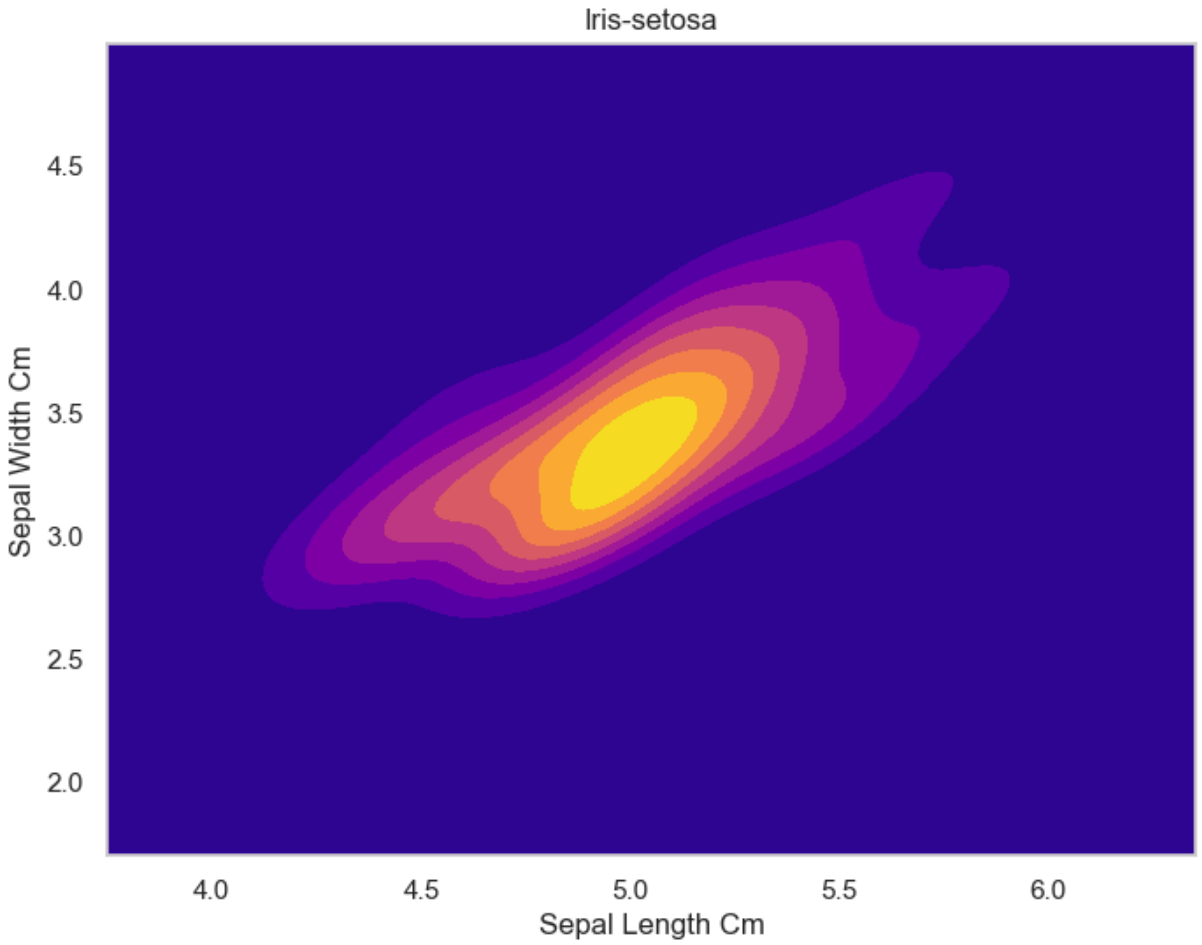
In [59]:
```
sns.FacetGrid(iris, hue="Species", height=6) \
   .map(sns.kdeplot, "PetalLengthCm") \
   .add_legend()
plt.ioff()
plt.show()
```



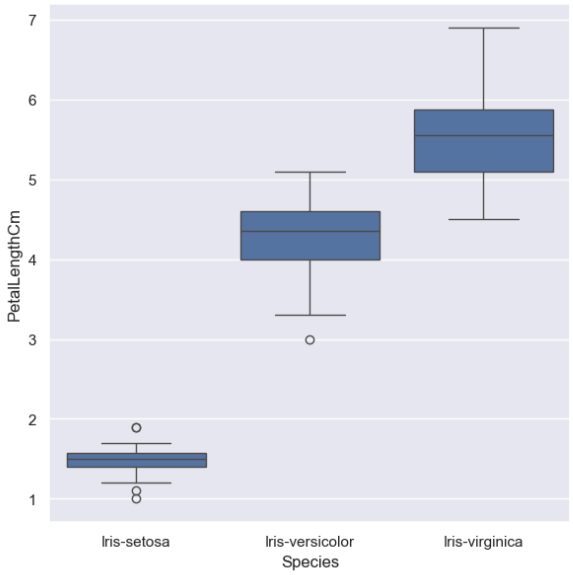
In [57]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxenplot(x='Species',y='SepalLengthCm',data=iris)
plt.show()
```

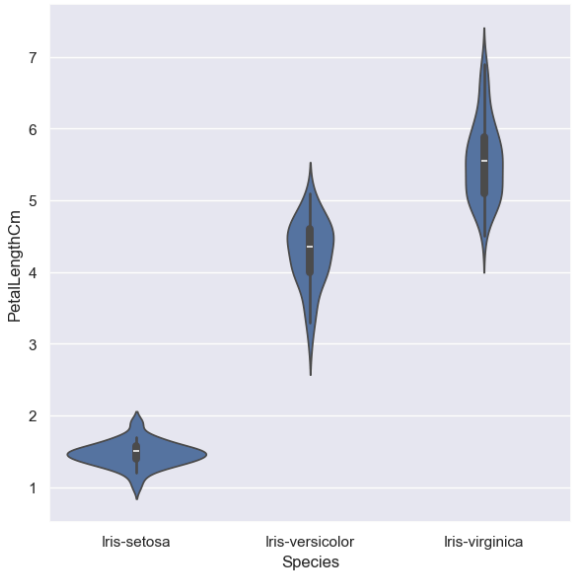


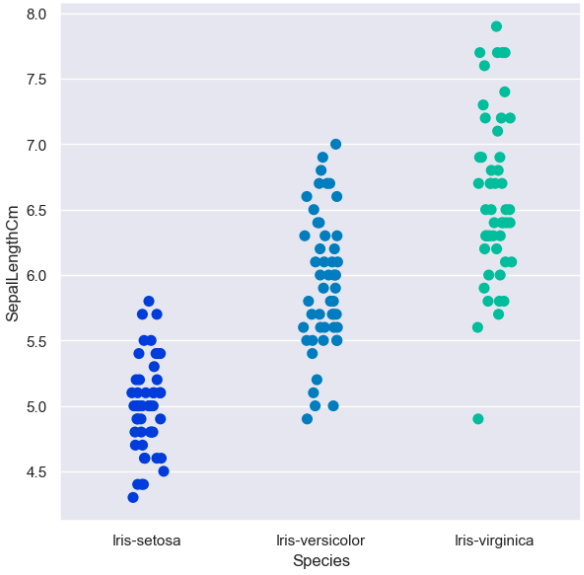
In [62]:
```
# Filter data for 'Iris-setosa'
sub = iris[iris['Species'] == 'Iris-setosa']

# Create KDE plot
plt.figure(figsize=(8,6))  # Set figure size
sns.kdeplot(x=sub["SepalLengthCm"], y=sub["SepalWidthCm"], cmap="plasma", fill=True

# Labels and title
plt.title("Iris-setosa")
plt.xlabel("Sepal Length Cm")
plt.ylabel("Sepal Width Cm")

# Show plot
plt.show()
```

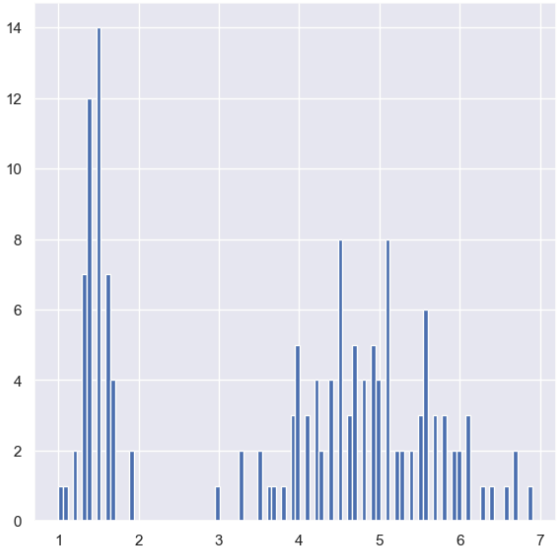


In [64]:
```
sns.set_style('darkgrid')
f,axes=plt.subplots(2,2,figsize=(15,15))

k1=sns.boxplot(x="Species", y="PetalLengthCm", data=iris,ax=axes[0,0])
k2=sns.violinplot(x='Species',y='PetalLengthCm',data=iris,ax=axes[0,1])
k3=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edgecolor='gra
axes[1,1].hist(iris.PetalLengthCm,bins=100)
plt.show()
```



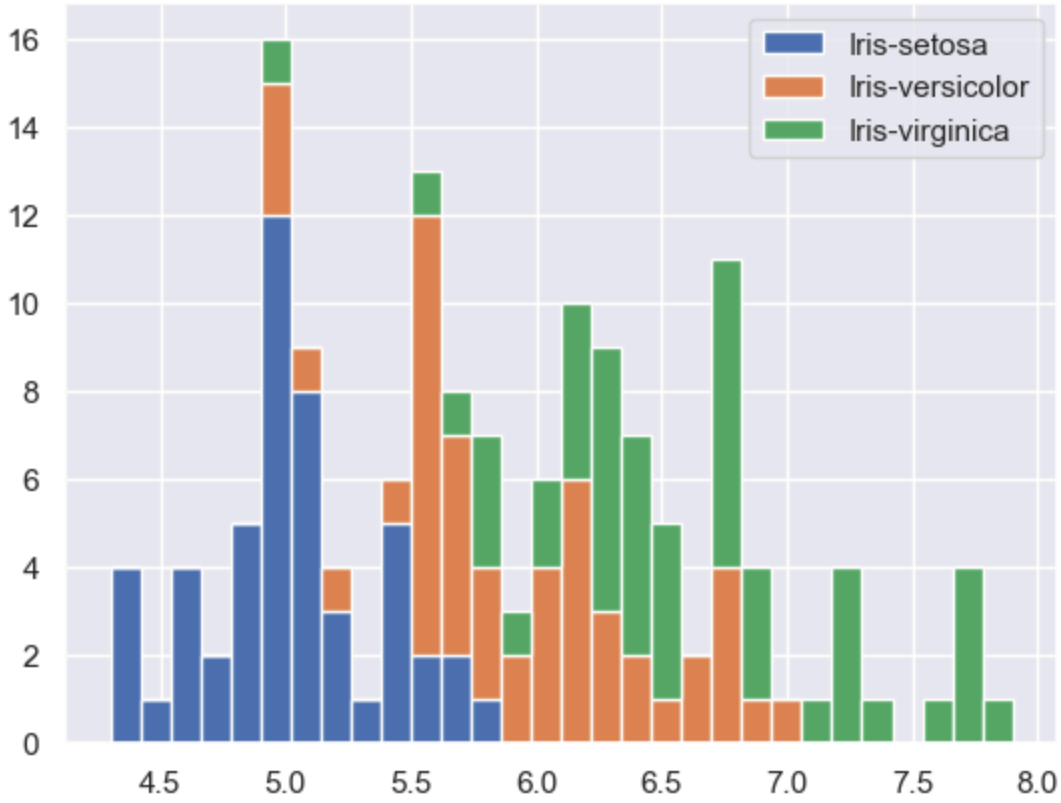




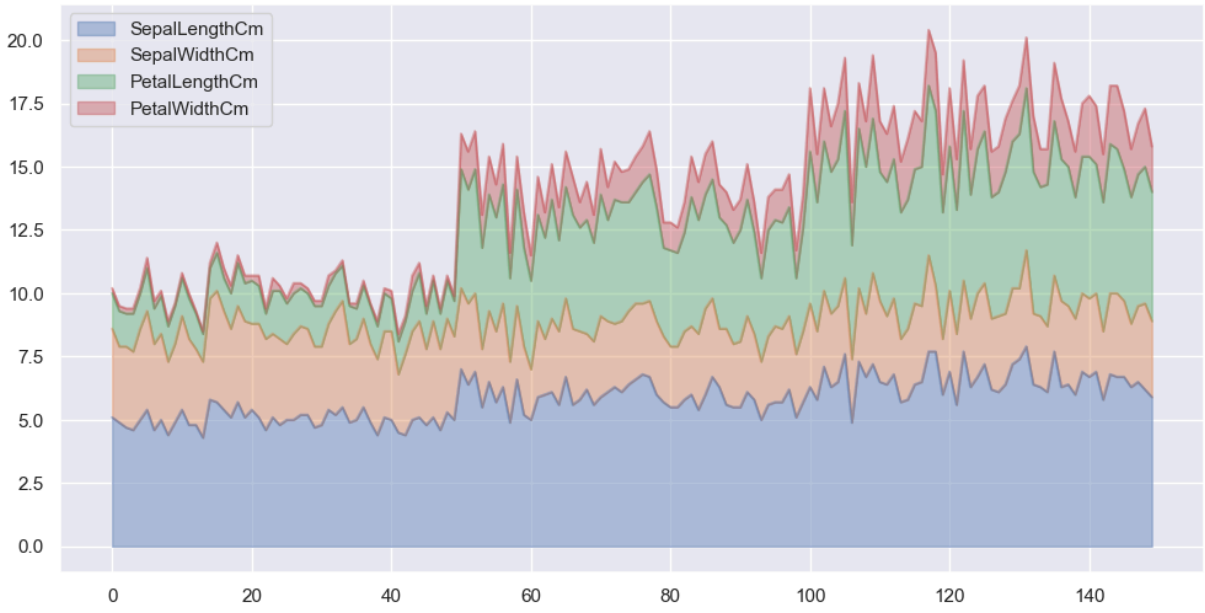


In [65]:
```
iris['Species'] = iris['Species'].astype('category')
#iris.head()
```

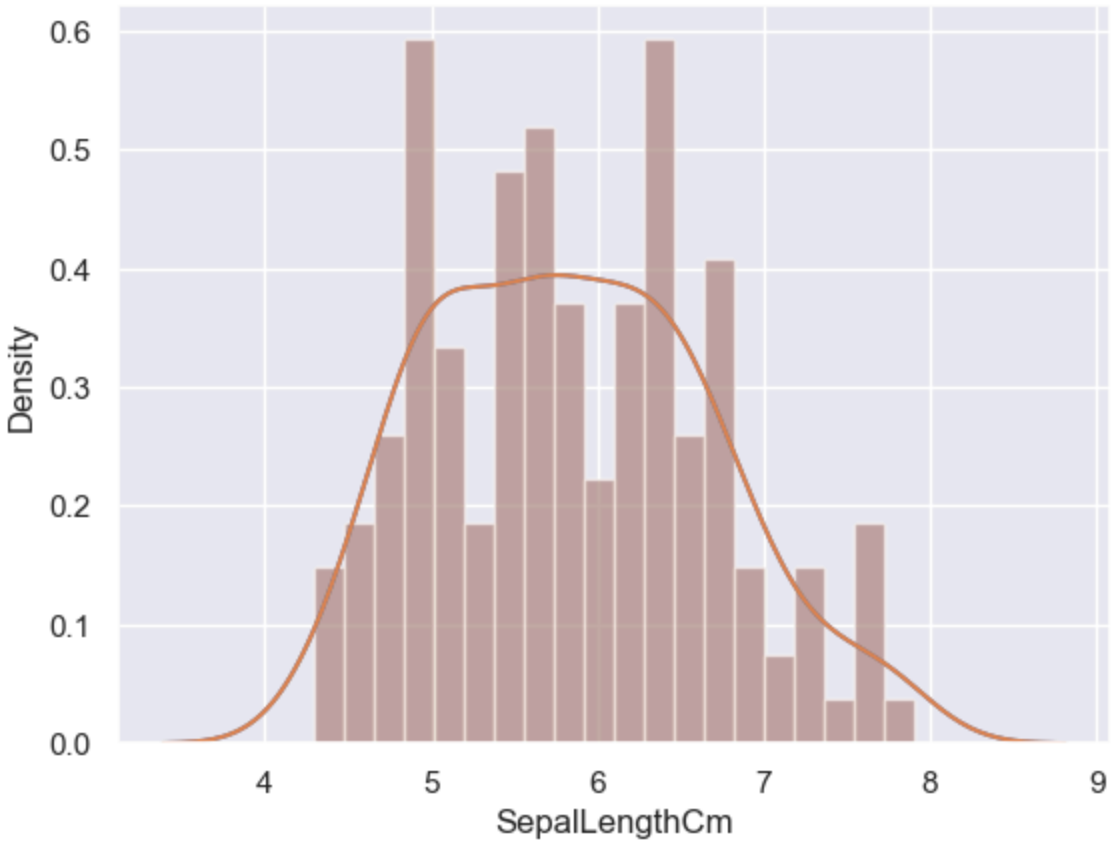
In [66]:
```
list1=list()
mylabels=list()
for gen in iris.Species.cat.categories:
    list1.append(iris[iris.Species==gen].SepalLengthCm)
    mylabels.append(gen)

h=plt.hist(list1,bins=30,stacked=True,rwidth=1,label=mylabels)
plt.legend()
plt.show()
```



In [67]:
```
iris.plot.area(y=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'],al
plt.show()
```



In [69]:
```
sns.distplot(iris['SepalLengthCm'],kde=True,bins=20)
plt.show()
```



In [ ]: