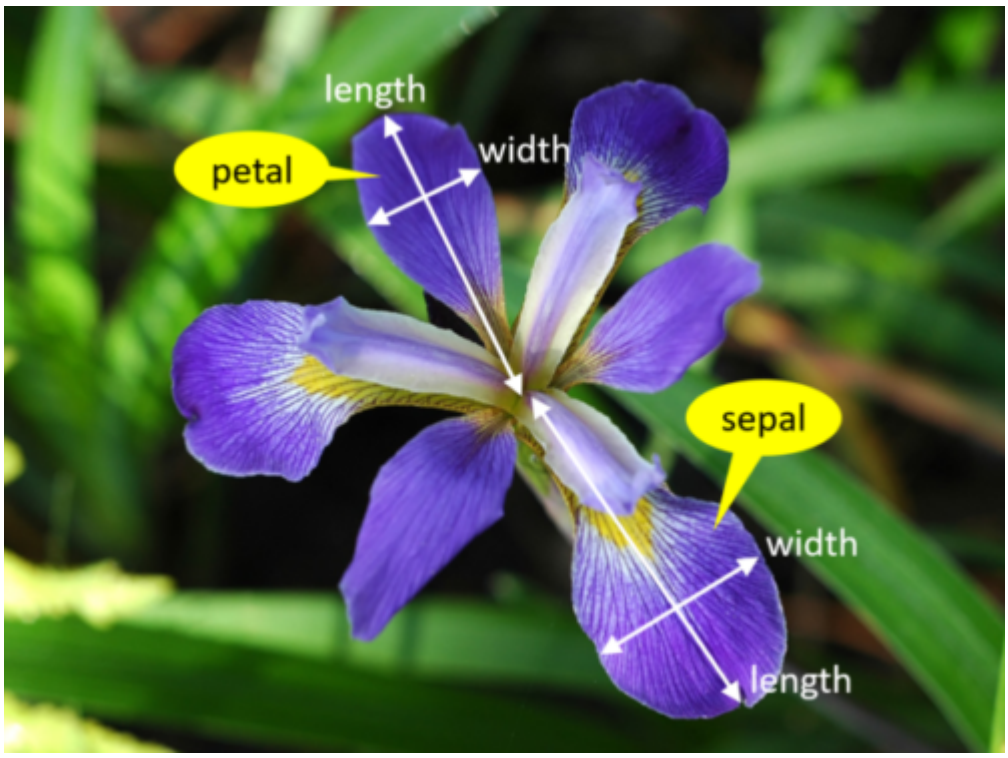
In [1]:
```
import pandas as pd
import numpy as np
from PIL import Image
```

In [2]:
```
flower_img = Image.open(r'C:\Users\lenovo\Desktop\Kaggle project\flower image.PN
```

In [3]:
```
flower_img
```

Out[3]:



In [4]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')
```

In [6]:
```
df = pd.read_csv(r'C:\Users\lenovo\Desktop\NIT FILES\5th, 6th - Sql workshop\IRI
```

In [7]:
```
df
```

Out[7]:

	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 [8]:
```
df.head()
```

Out[8]:

	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]:
```
df.drop('Id',axis=1,inplace=True)
```

In [11]:
```
df.head(2)
```

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

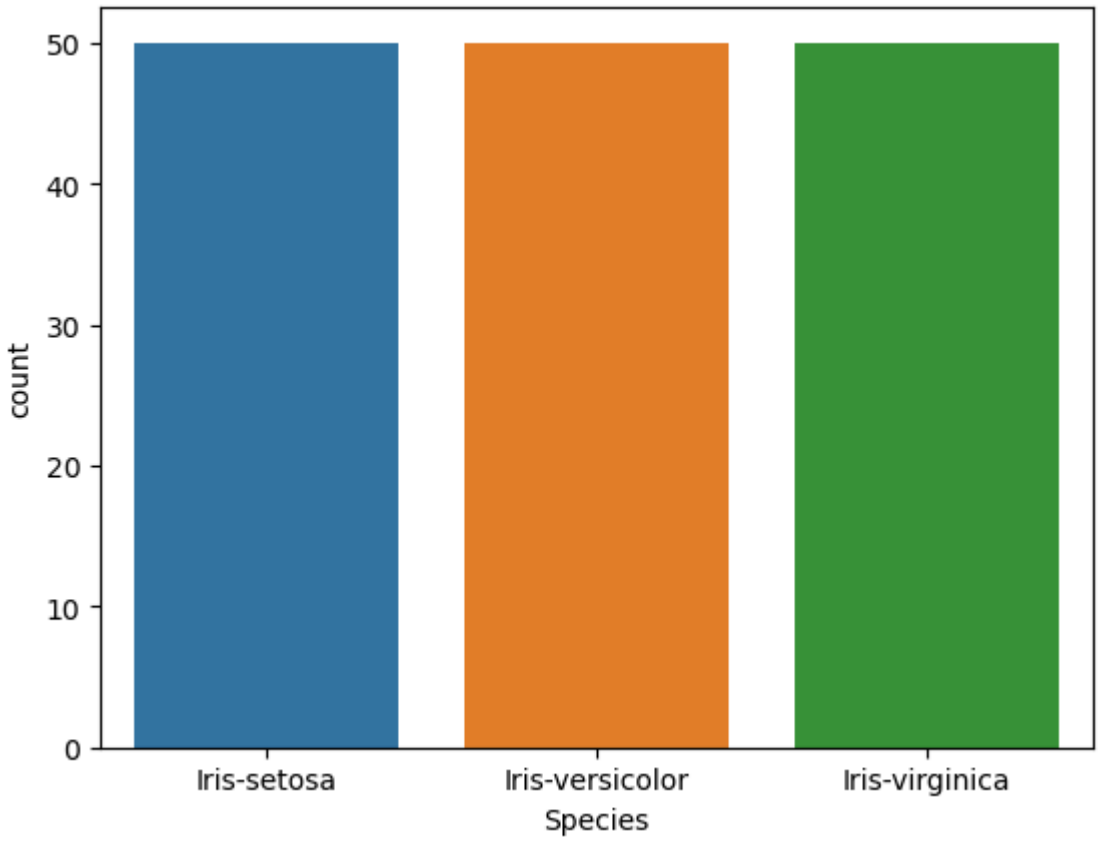
In [12]:
```
df.info()
```

```
<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 [14]:
```
df['Species'].value_counts()
```

Out[14]:
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

In [16]:
```
sns.countplot(x='Species',data=df,hue="Species")
plt.show()
```

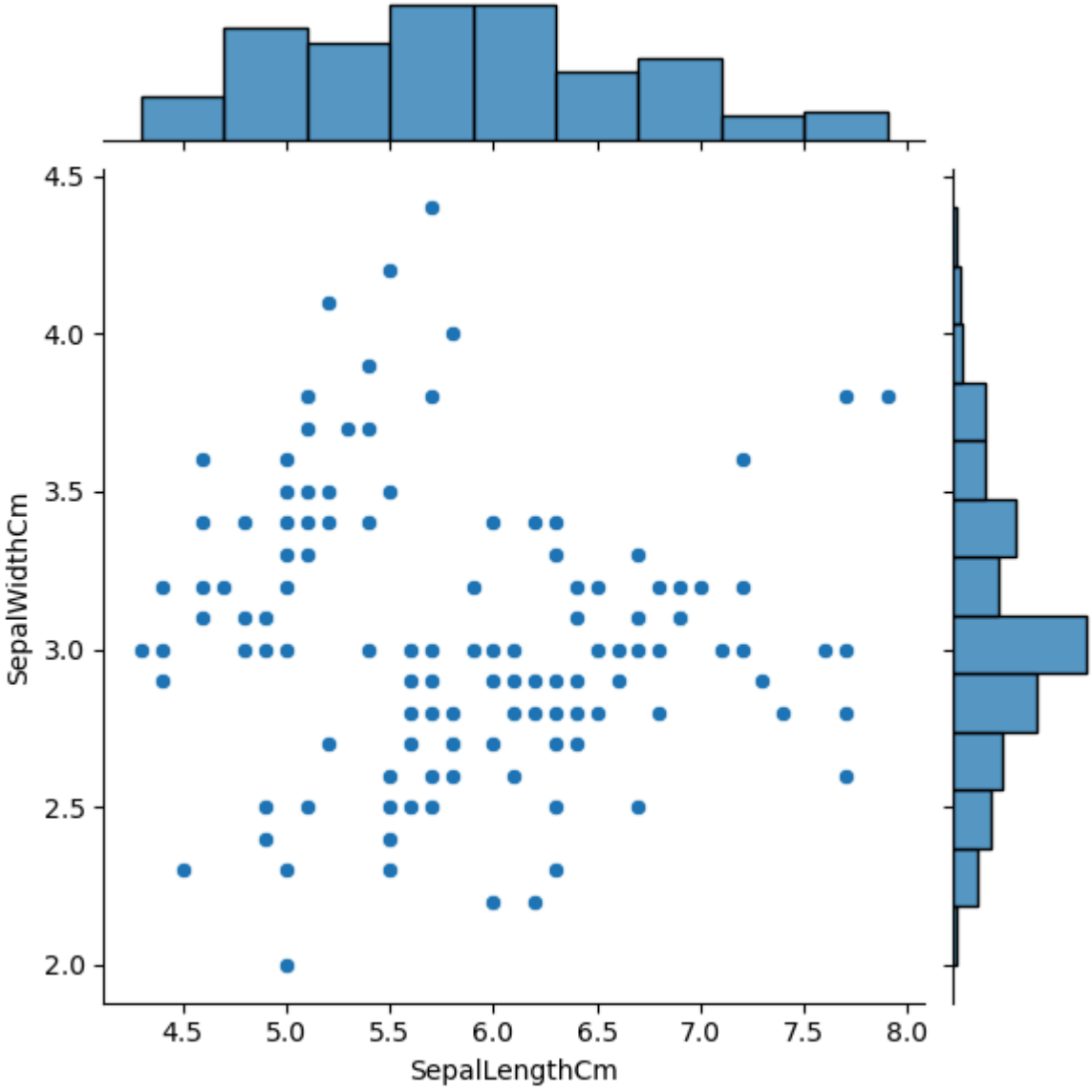


In [17]:
```
df.head()
```

Out[17]:

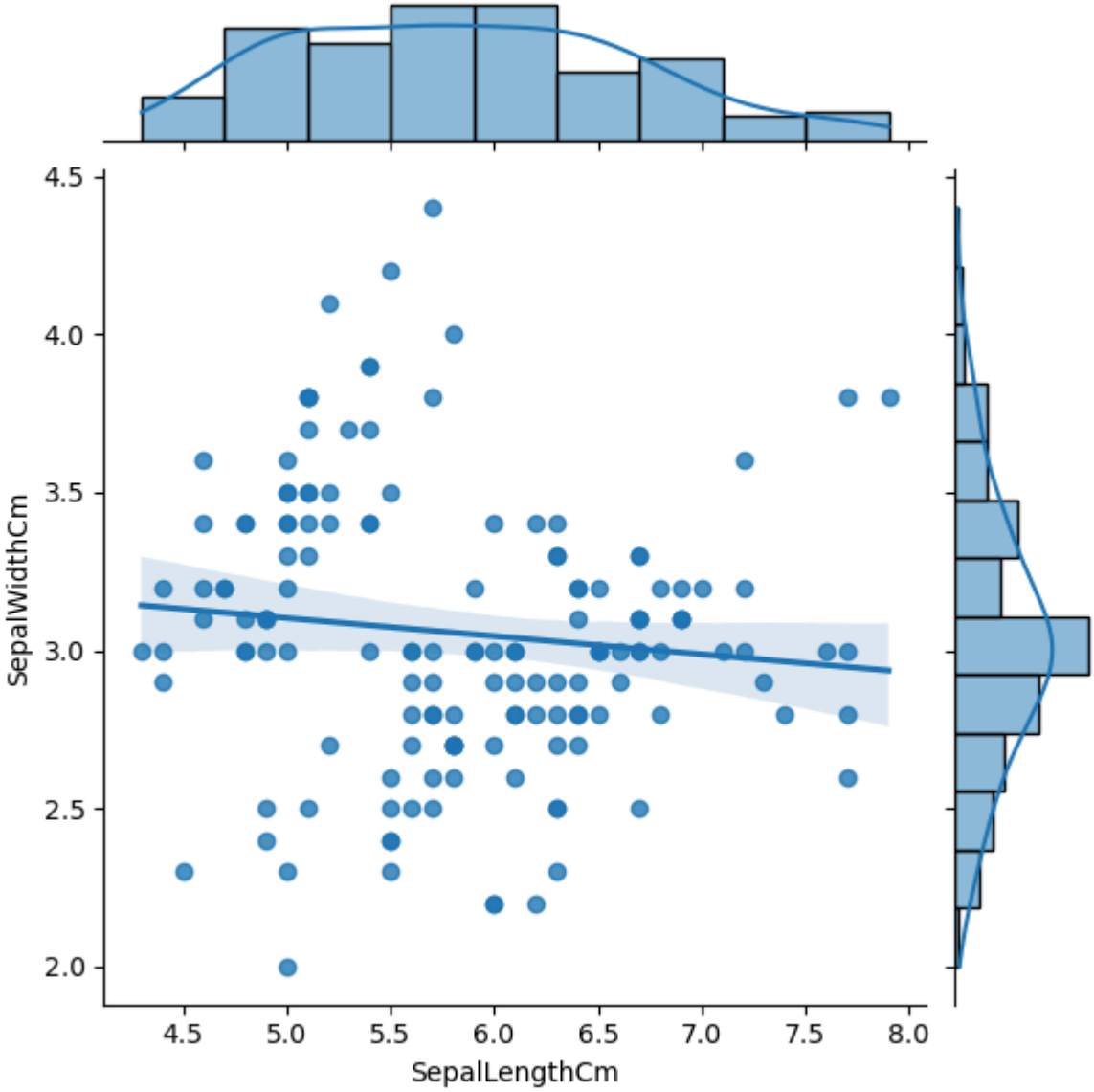
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [18]:
```
fig=sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',data=df)
```

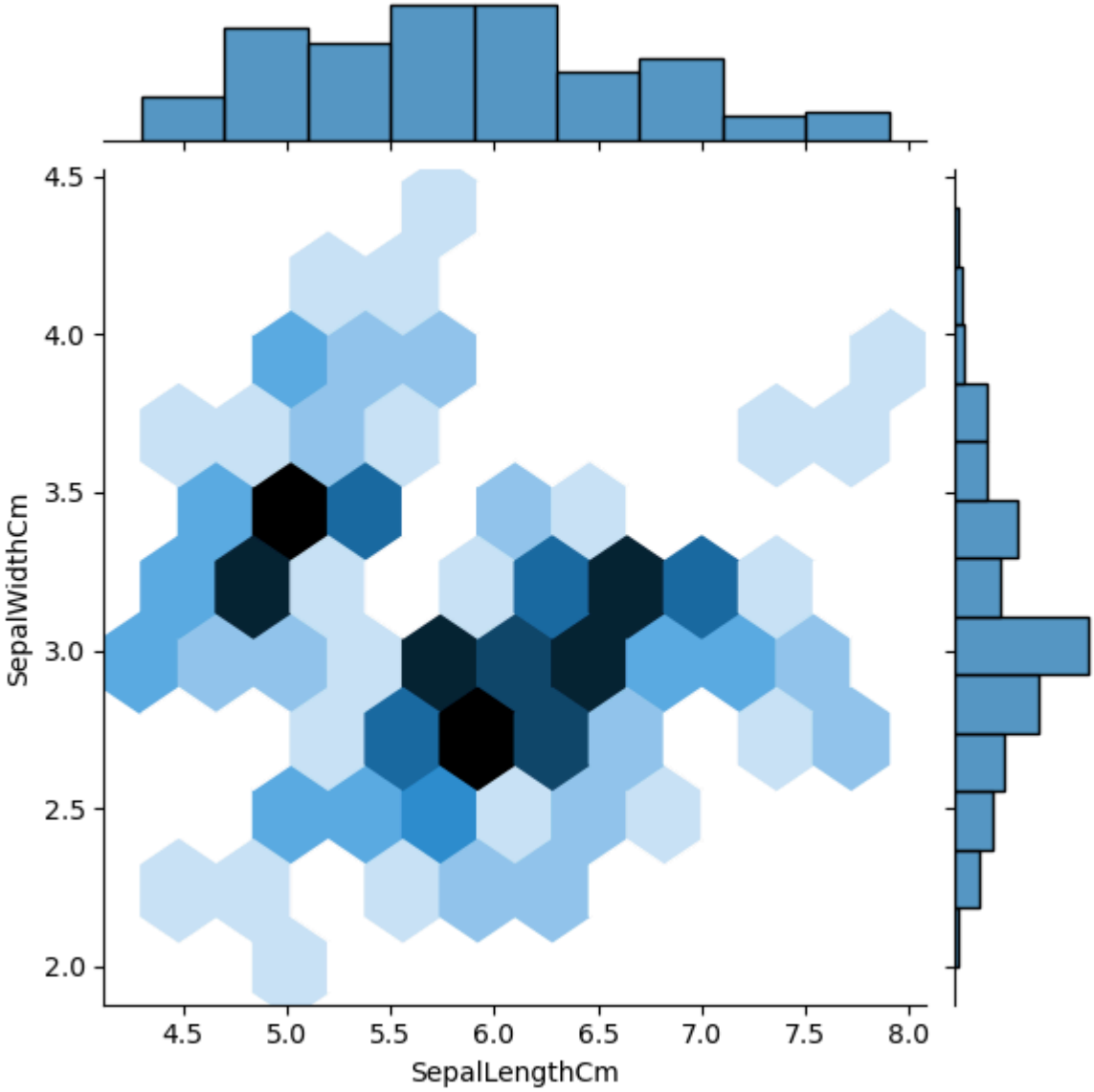


In [19]:
```
sns.jointplot(x="SepalLengthCm", y="SepalWidthCm", data=df, kind="reg")
```

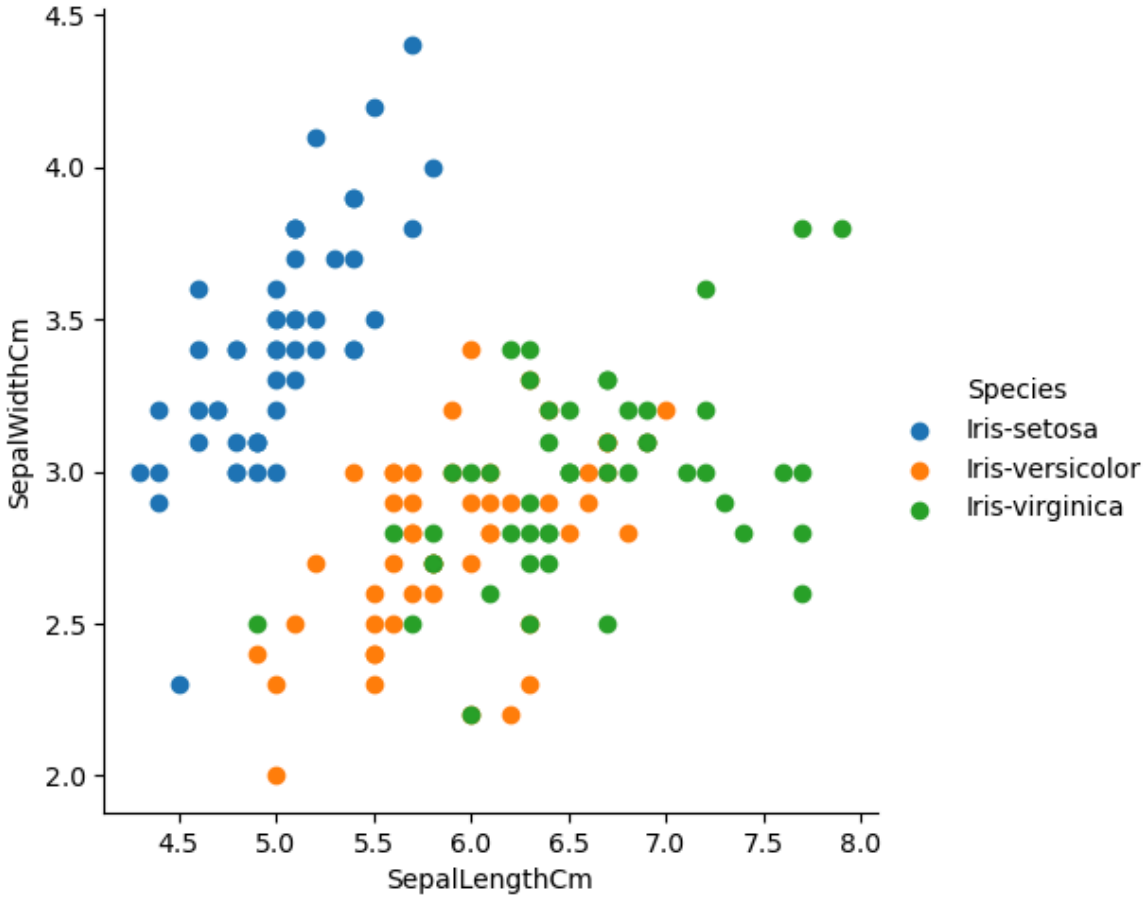
Out[19]:
```
<seaborn.axisgrid.JointGrid at 0x20500069b20>
```



In [20]:
```
fig=sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',kind='hex',data=df)
```



In [21]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'iris' is your DataFrame
sns.FacetGrid(df, hue='Species', height=5)\
    .map(plt.scatter, 'SepalLengthCm', 'SepalWidthCm')\
    .add_legend()
plt.show()
```

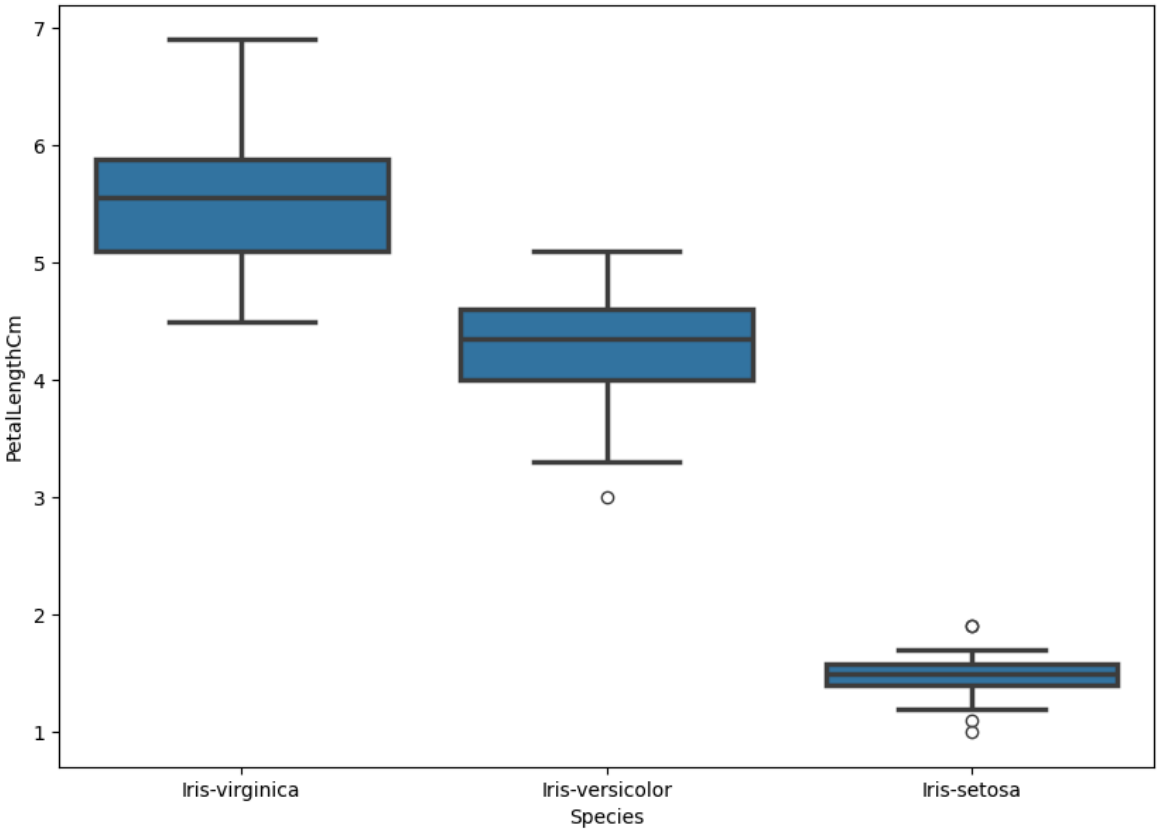


In [22]:
```
df.head()
```

Out[22]:

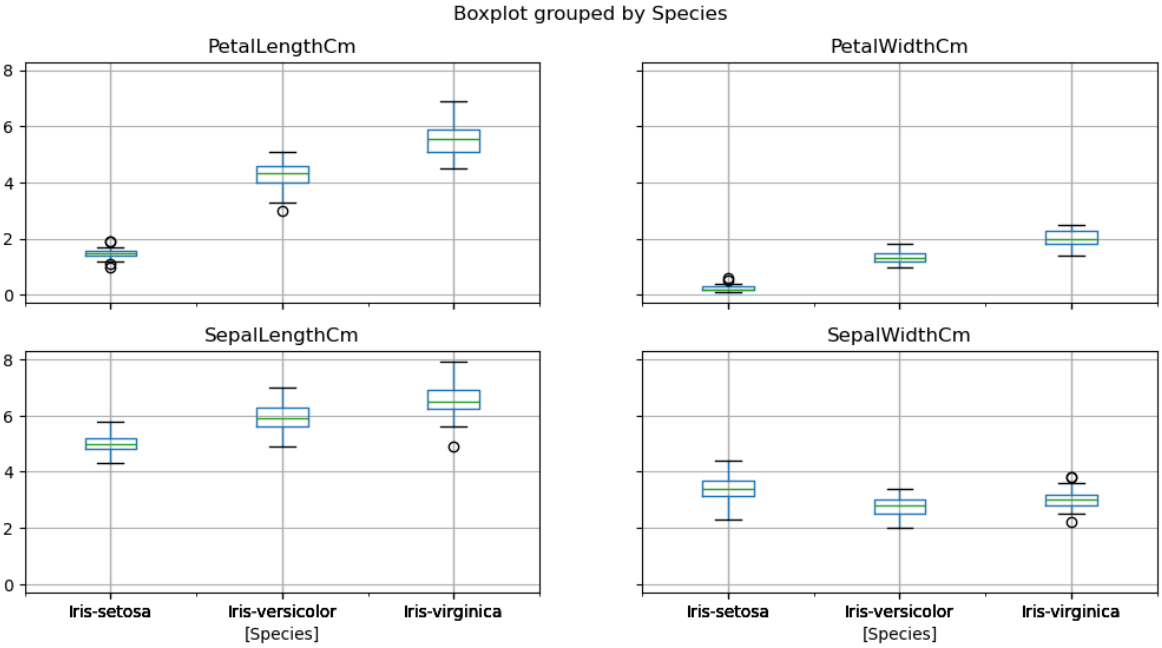
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [23]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='PetalLengthCm',data=df,order=['Iris-virginica','I
```

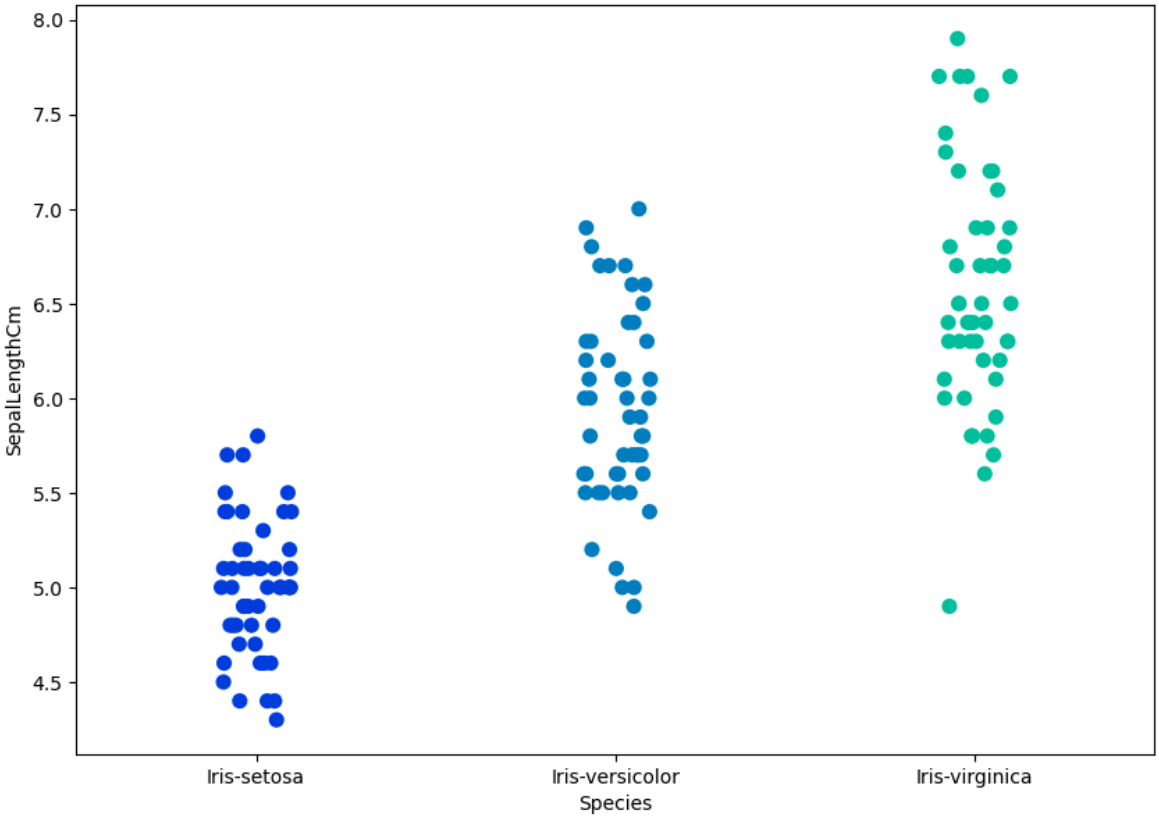


In [24]:
```
df.boxplot(by="Species", figsize=(12, 6))
```

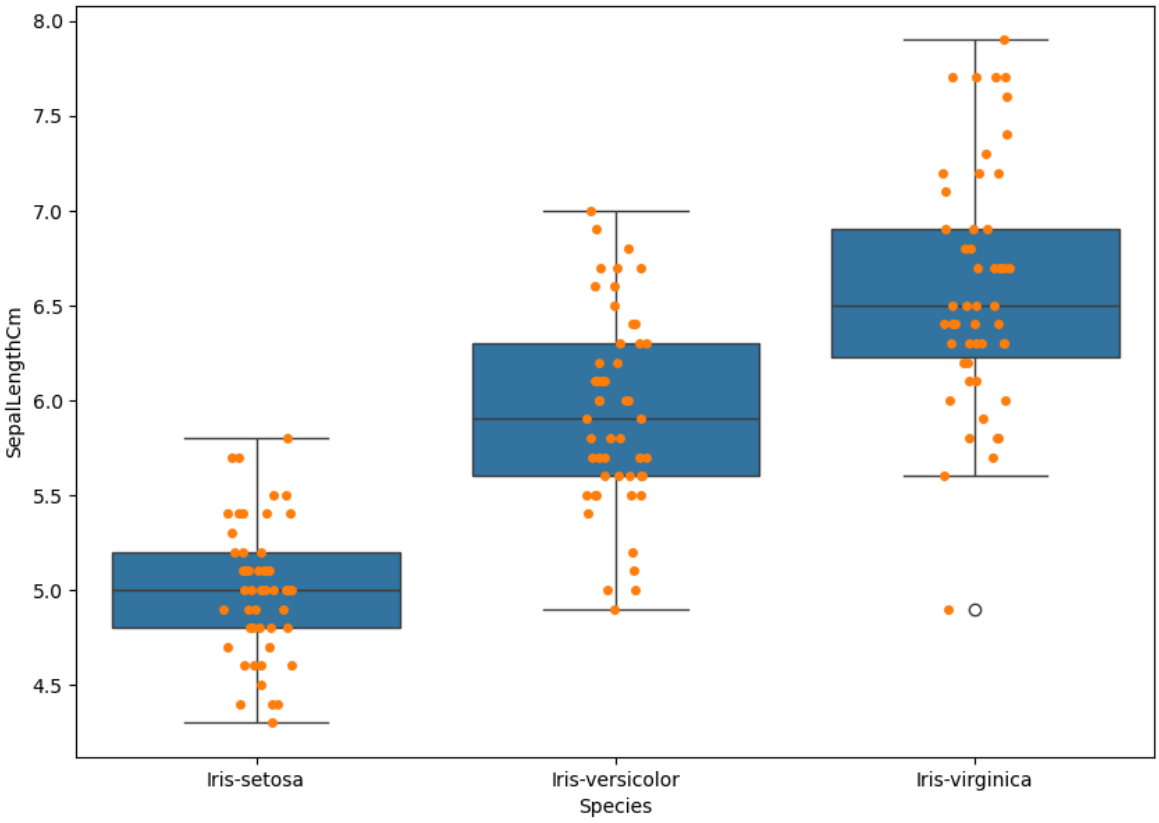
Out[24]:
```
array([[<Axes: title={'center': 'PetalLengthCm'}, xlabel='[Species]'>,
        <Axes: title={'center': 'PetalWidthCm'}, xlabel='[Species]'>],
       [<Axes: title={'center': 'SepalLengthCm'}, xlabel='[Species]'>,
        <Axes: title={'center': 'SepalWidthCm'}, xlabel='[Species]'>]],
      dtype=object)
```



In [25]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=df,jitter=True,edgecolor='g
```

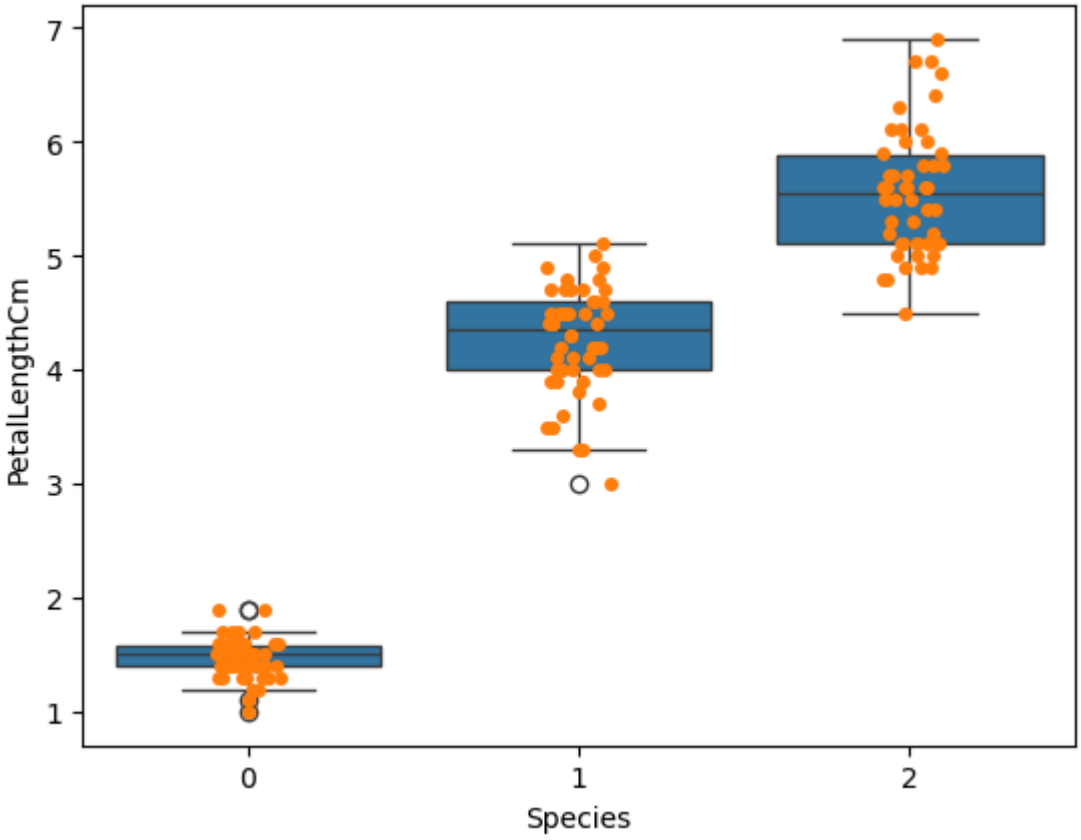


In [26]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='SepalLengthCm',data=df)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=df,jitter=True,edgecolor='g
```

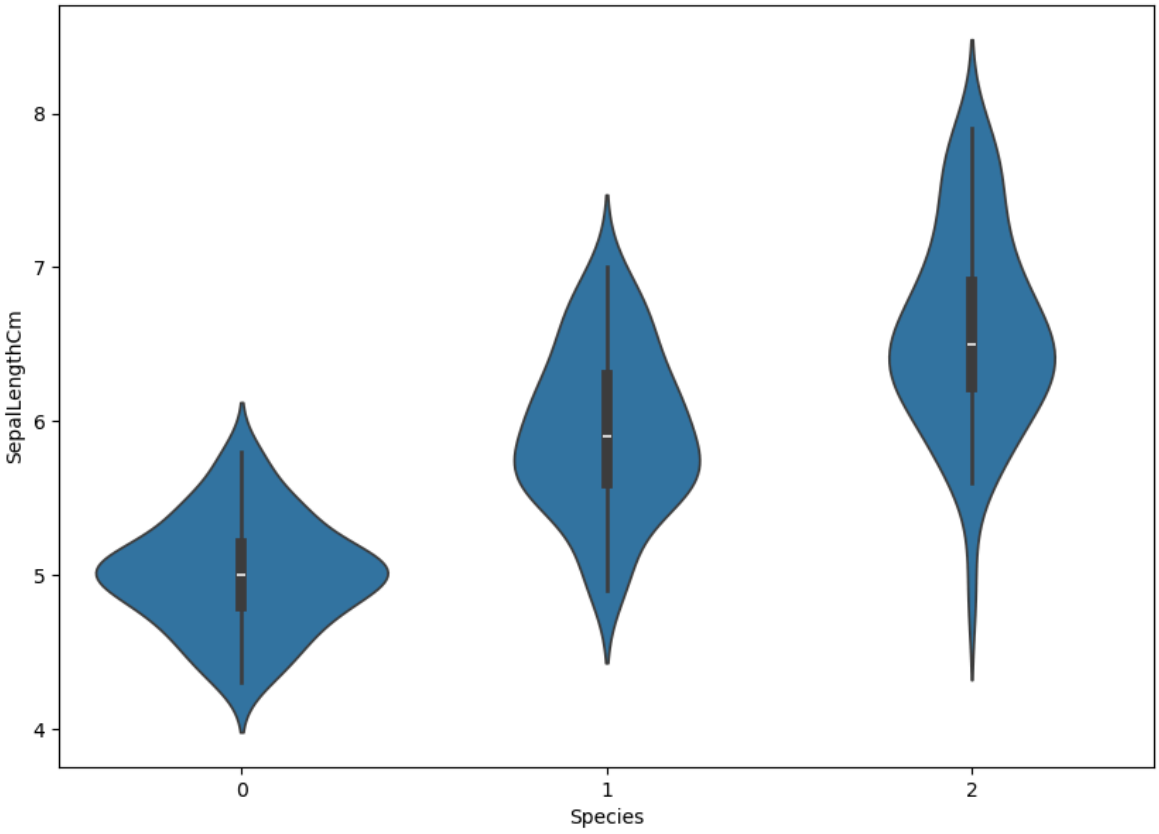


In [27]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
```

```
# Sample data loading (use your own data if different)
from sklearn.datasets import load_iris
iris_data = load_iris()
df = pd.DataFrame(data=iris_data.data, columns=iris_data.feature_names)
df['Species'] = iris_data.target

# Rename columns to match the example
df.rename(columns={
    'sepal length (cm)': 'SepalLengthCm',
    'sepal width (cm)': 'SepalWidthCm',
    'petal length (cm)': 'PetalLengthCm',
    'petal width (cm)': 'PetalWidthCm'
}, inplace=True)

# Plotting
ax = sns.boxplot(x="Species", y="PetalLengthCm", data=df)
ax = sns.stripplot(x="Species", y="PetalLengthCm", data=df, jitter=True, edgecol

plt.show()
```

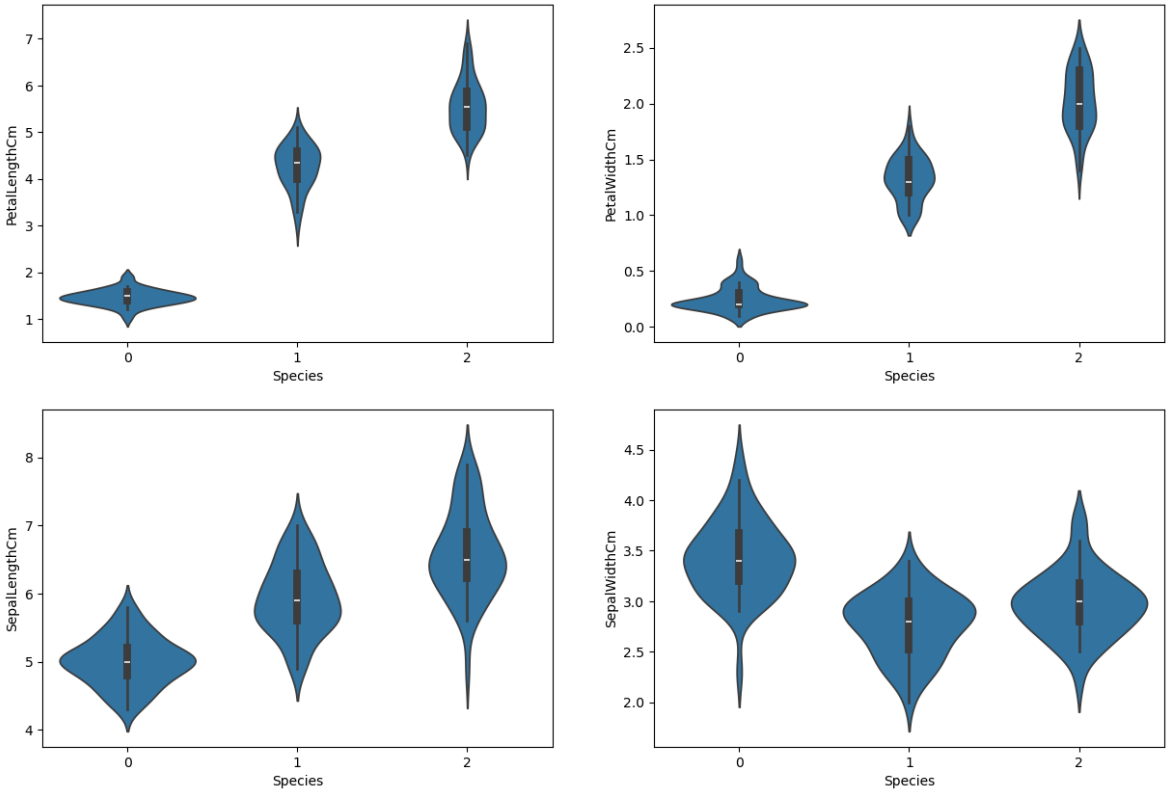


In [28]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.violinplot(x='Species',y='SepalLengthCm',data=df)
```



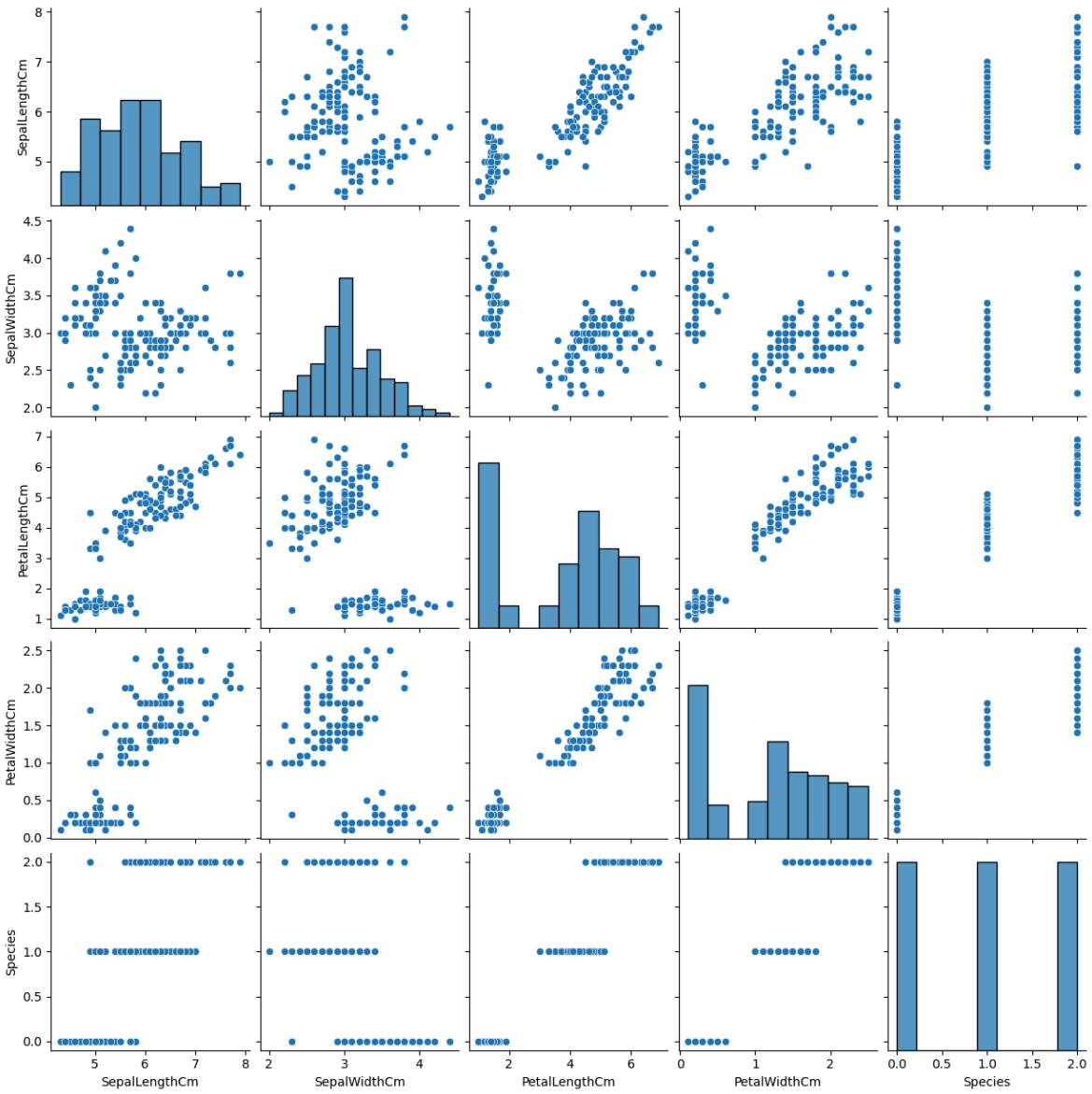
In [29]:
```
plt.figure(figsize=(15,10))
plt.subplot(2,2,1)
sns.violinplot(x='Species',y='PetalLengthCm',data=df)
plt.subplot(2,2,2)
sns.violinplot(x='Species',y='PetalWidthCm',data=df)
plt.subplot(2,2,3)
sns.violinplot(x='Species',y='SepalLengthCm',data=df)
plt.subplot(2,2,4)
sns.violinplot(x='Species',y='SepalWidthCm',data=df)
```

Out[29]:
```
<Axes: xlabel='Species', ylabel='SepalWidthCm'>
```

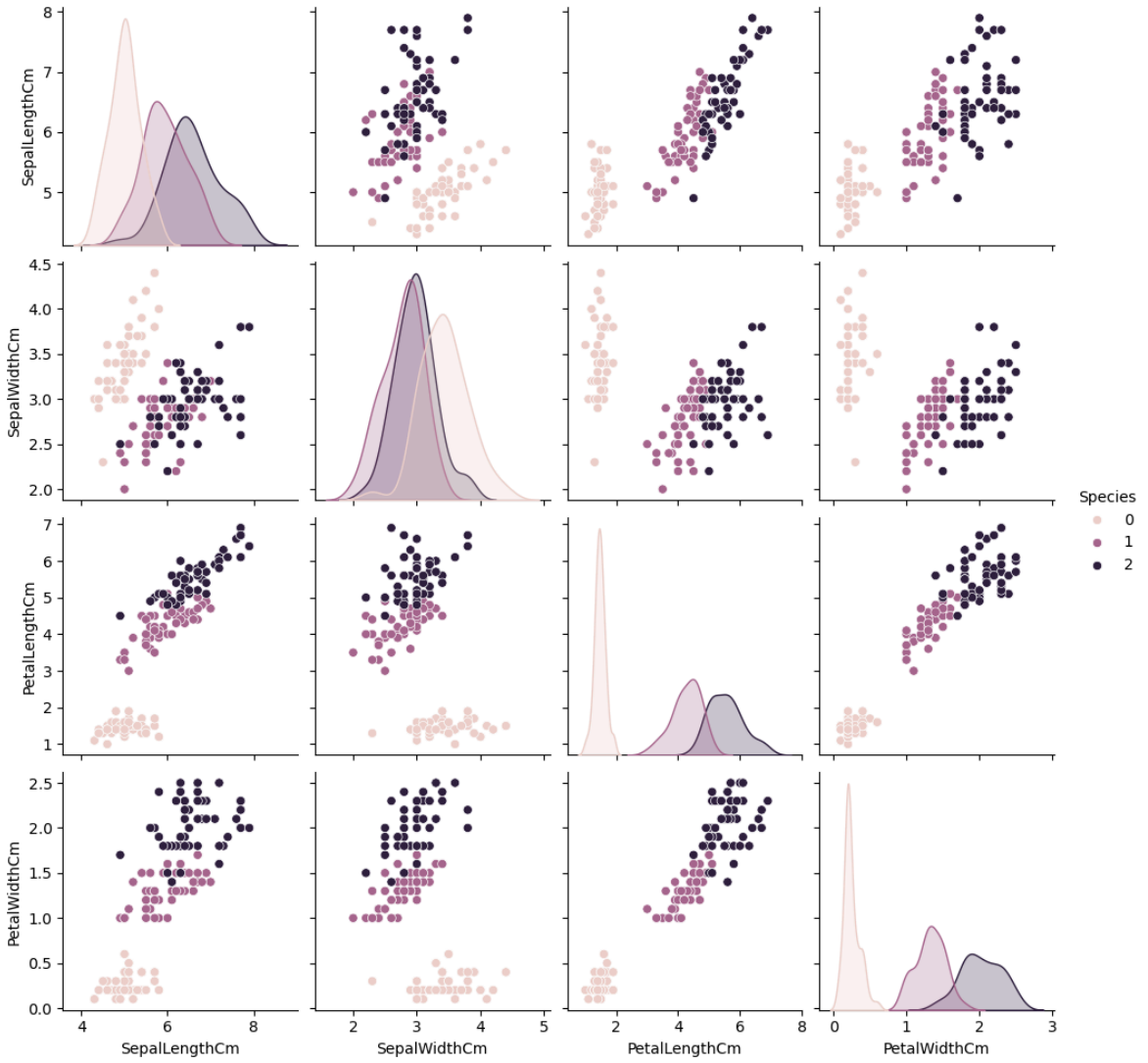


In [30]:
```
sns.pairplot(data=df,kind='scatter')
```

Out[30]:
```
<seaborn.axisgrid.PairGrid at 0x20502d24260>
```



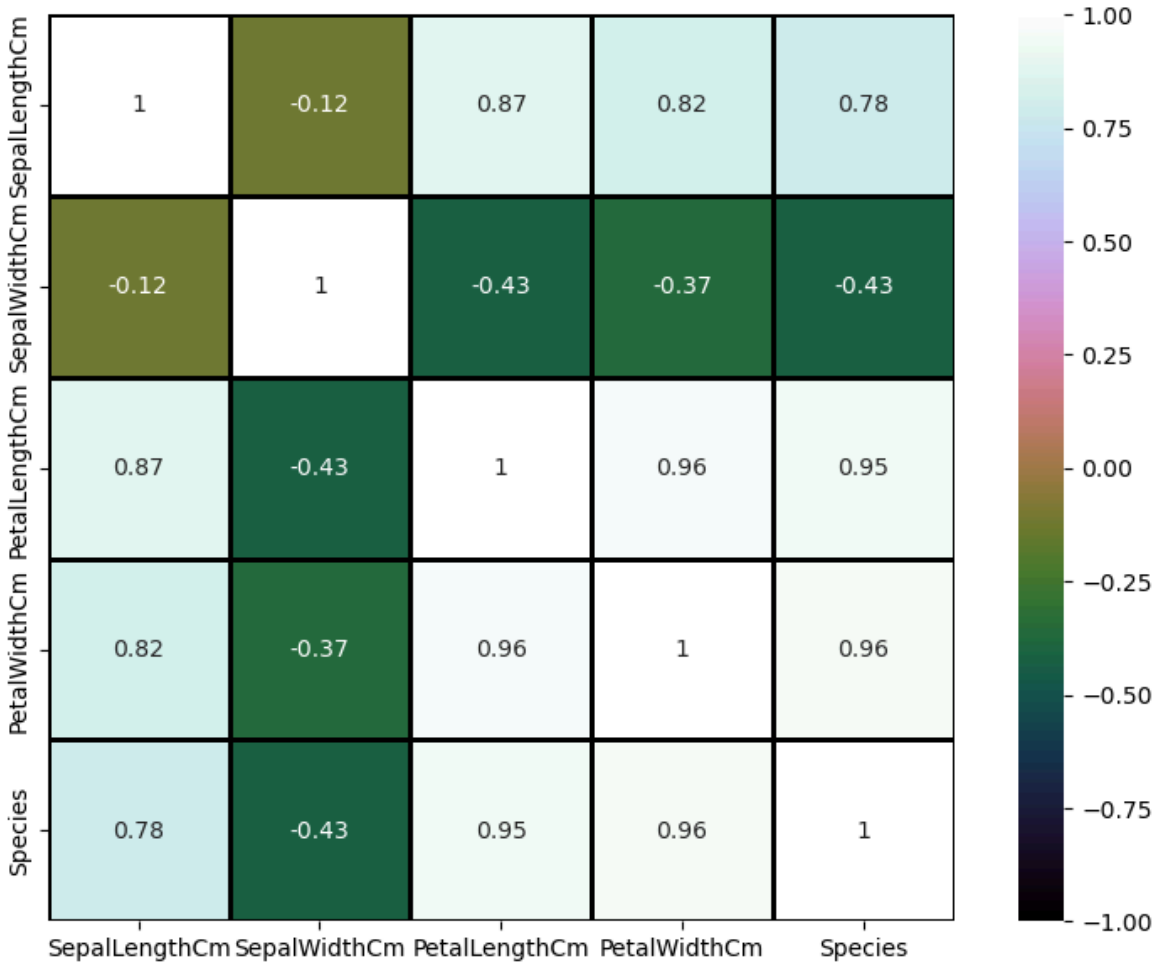
In [31]:
```
sns.pairplot(df,hue='Species');
```



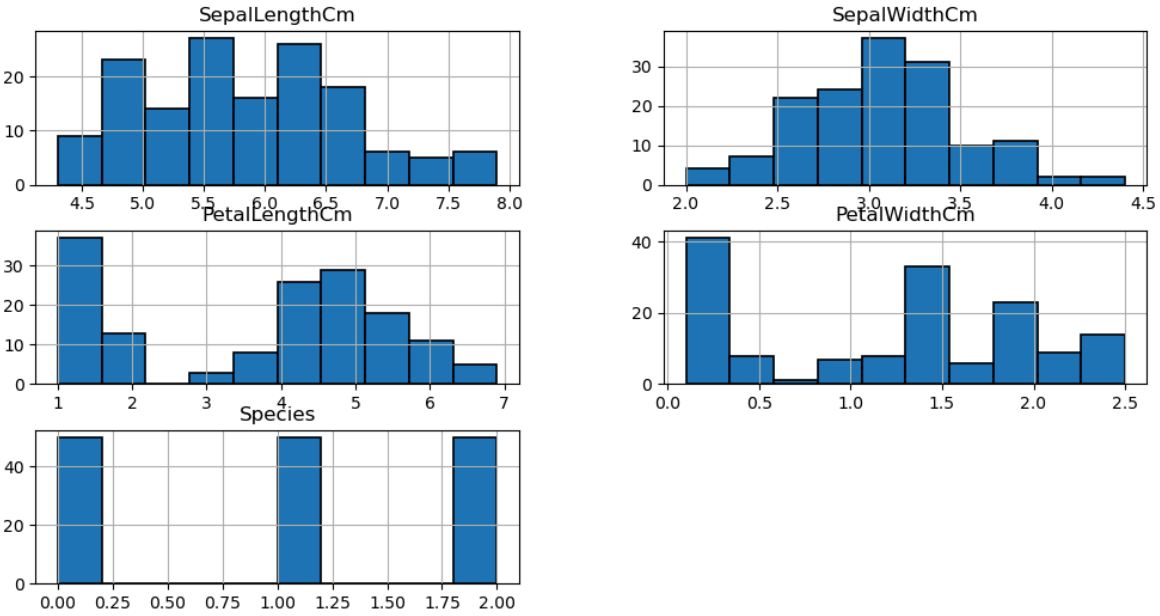
In [32]:
```
numeric_iris = df.select_dtypes(include=[np.number])
```

In [33]:
```
corr_matrix = numeric_iris.corr()
```

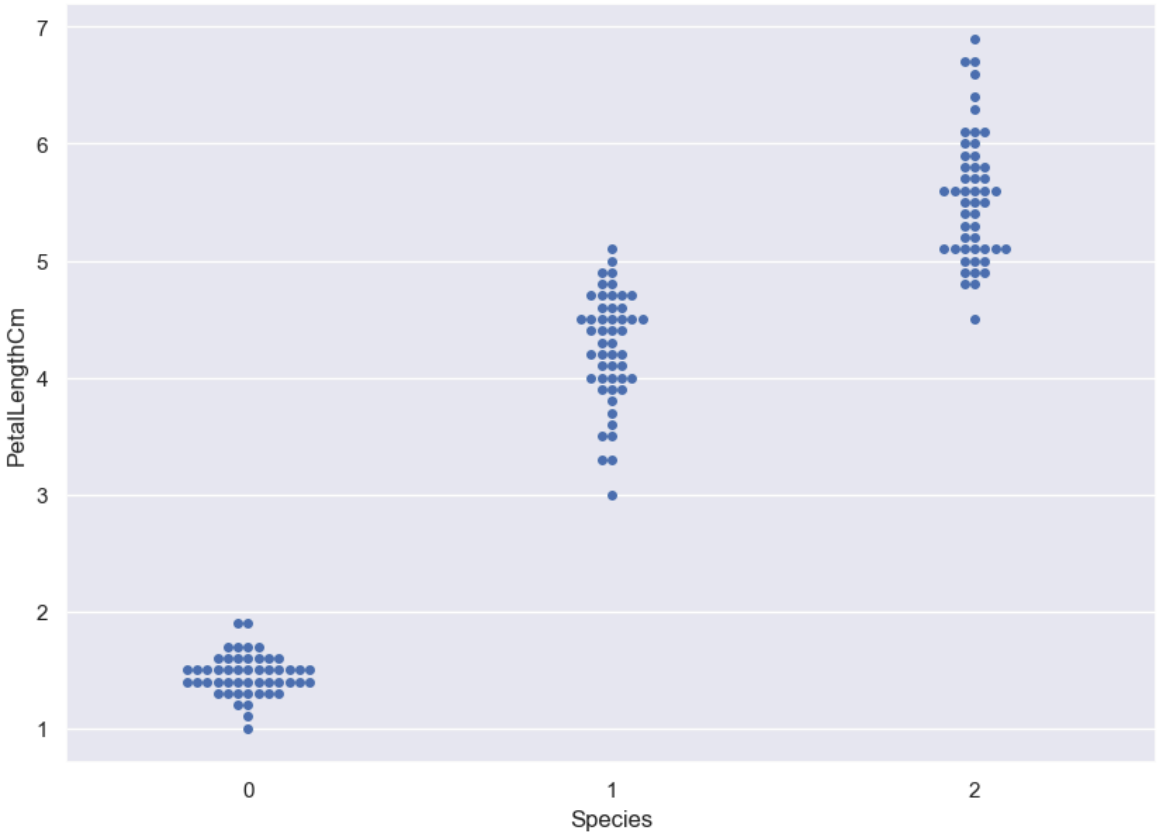
In [34]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 7))
sns.heatmap(corr_matrix, annot=True, cmap='cubehelix', linewidths=1, linecolor='
plt.show()
```



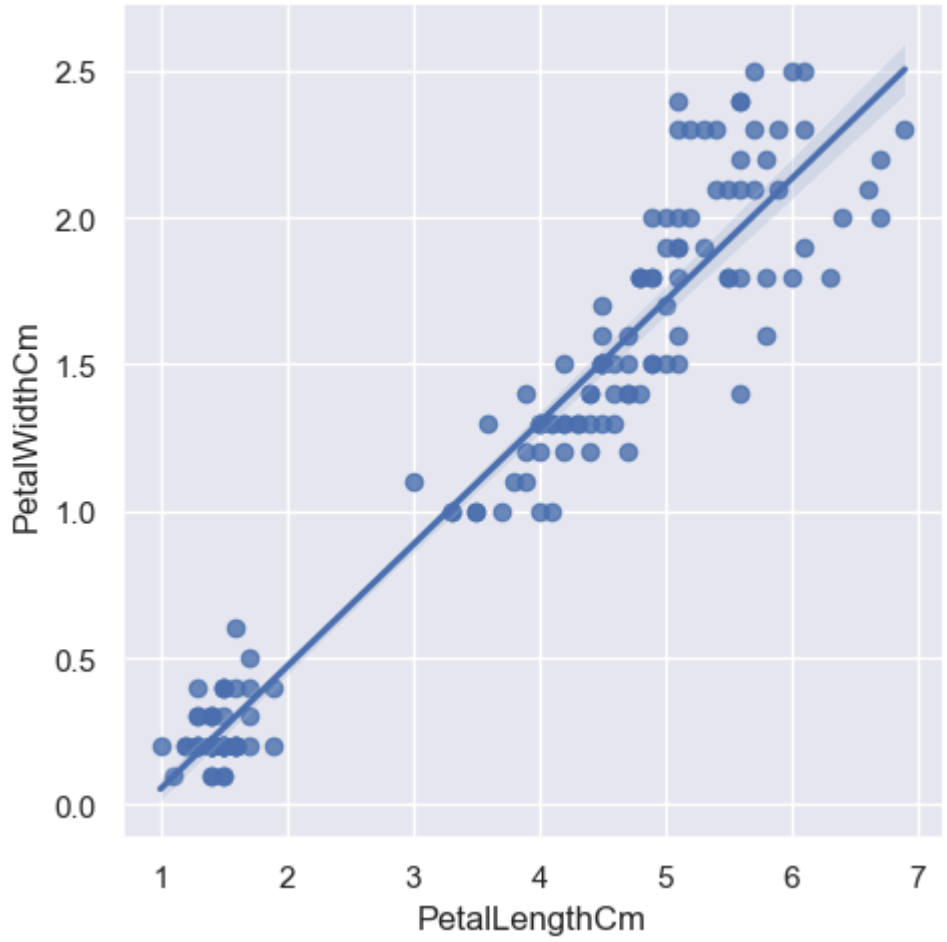
In [35]:
```
df.hist(edgecolor='black', linewidth=1.2)
fig=plt.gcf()
fig.set_size_inches(12,6)
```



In [36]:
```
sns.set(style="darkgrid")
fig=plt.gcf()
fig.set_size_inches(10,7)
fig = sns.swarmplot(x="Species", y="PetalLengthCm", data=df)
```

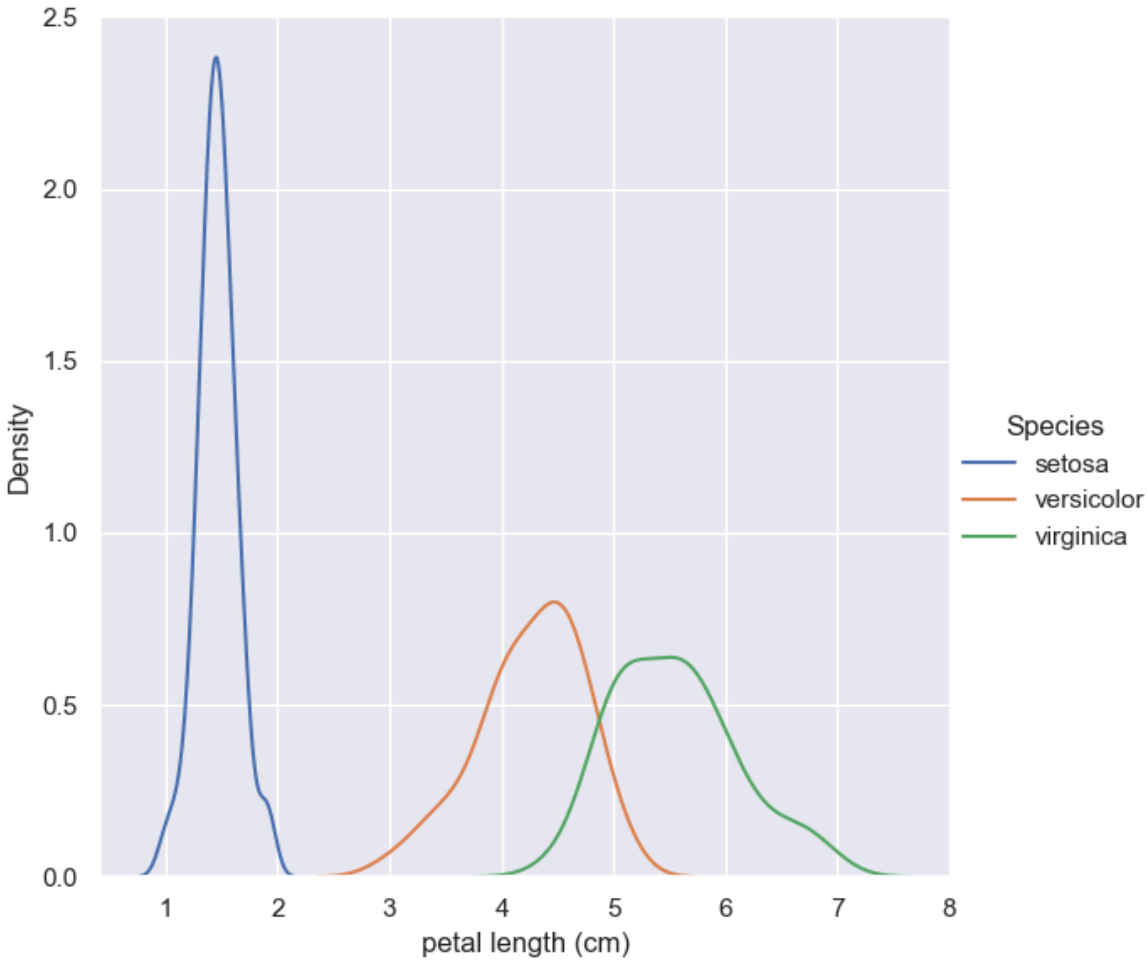


In [37]:
```
fig=sns.lmplot(x="PetalLengthCm", y="PetalWidthCm",data=df)
```



In [39]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

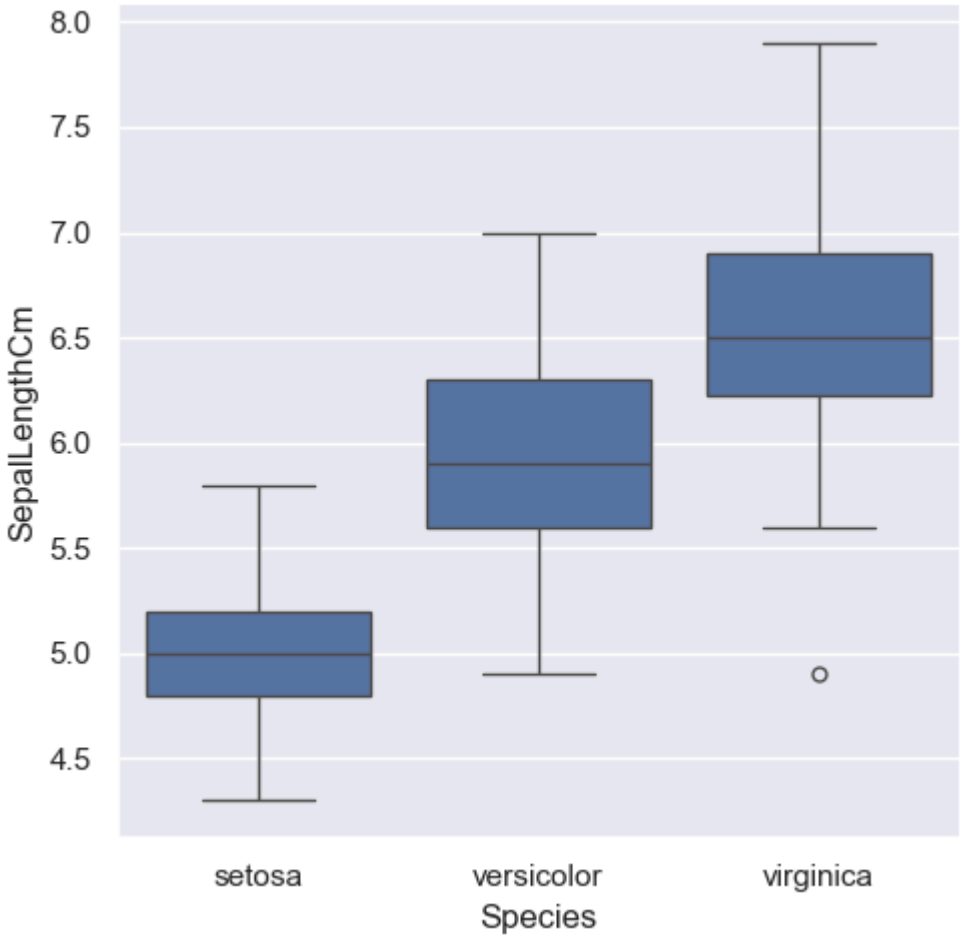
```
from sklearn.datasets import load_iris
import pandas as pd

# Load the iris dataset
iris_data = load_iris()
df = pd.DataFrame(data=iris_data.data, columns=iris_data.feature_names)
df['Species'] = iris_data.target
df['Species'] = df['Species'].map({0: 'setosa', 1: 'versicolor', 2: 'virginica'}

# Create the FacetGrid with the updated parameter
sns.FacetGrid(df, hue="Species", height=6) \
   .map(sns.kdeplot, "petal length (cm)") \
   .add_legend()

plt.show()
```



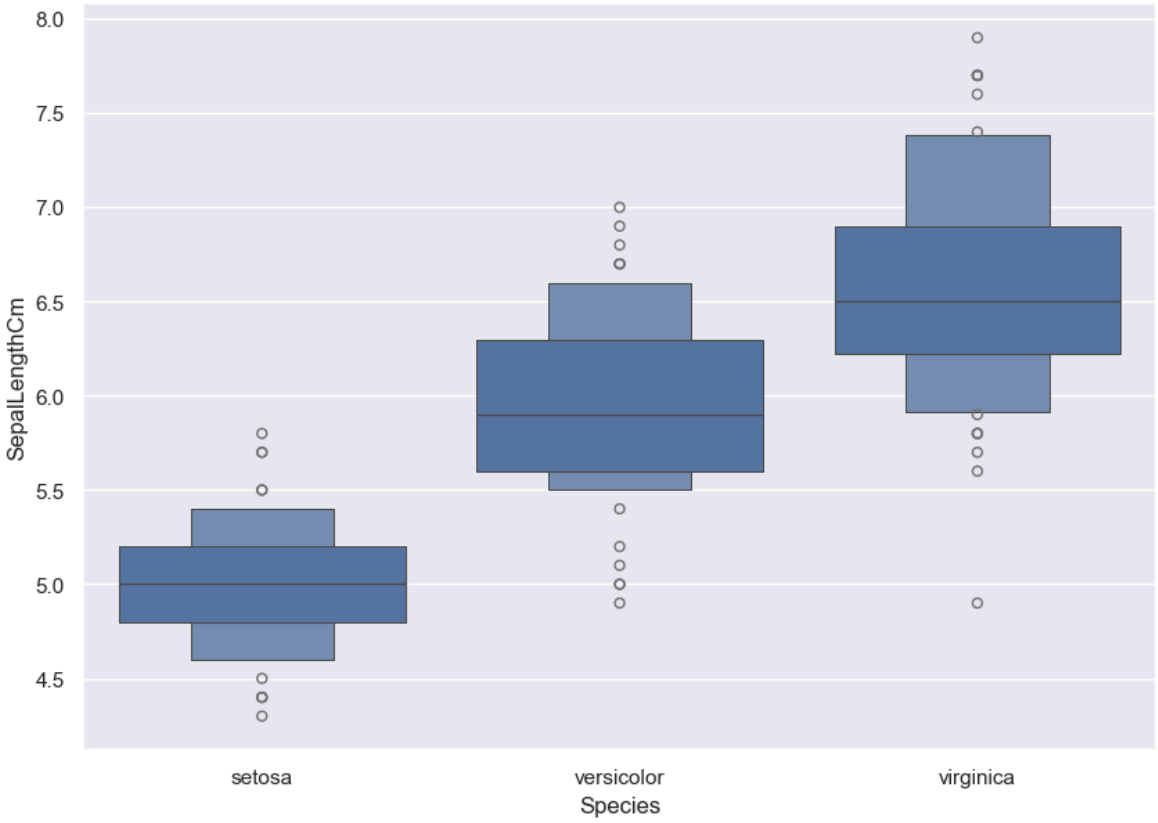
In [40]:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.datasets import load_iris

# Load and prepare the data
iris_data = load_iris()
df = pd.DataFrame(data=iris_data.data, columns=iris_data.feature_names)
df['Species'] = iris_data.target

# Rename columns to match the example
df.rename(columns={
    'sepal length (cm)': 'SepalLengthCm',
    'sepal width (cm)': 'SepalWidthCm',
```

```
    'petal length (cm)': 'PetalLengthCm',
    'petal width (cm)': 'PetalWidthCm'
}, inplace=True)

# Ensure 'Species' is in string format
species_mapping = {i: name for i, name in enumerate(iris_data.target_names)}
df['Species'] = df['Species'].map(species_mapping)

# Plotting
sns.catplot(x='Species', y='SepalLengthCm', data=df, kind='box')
plt.show()
```

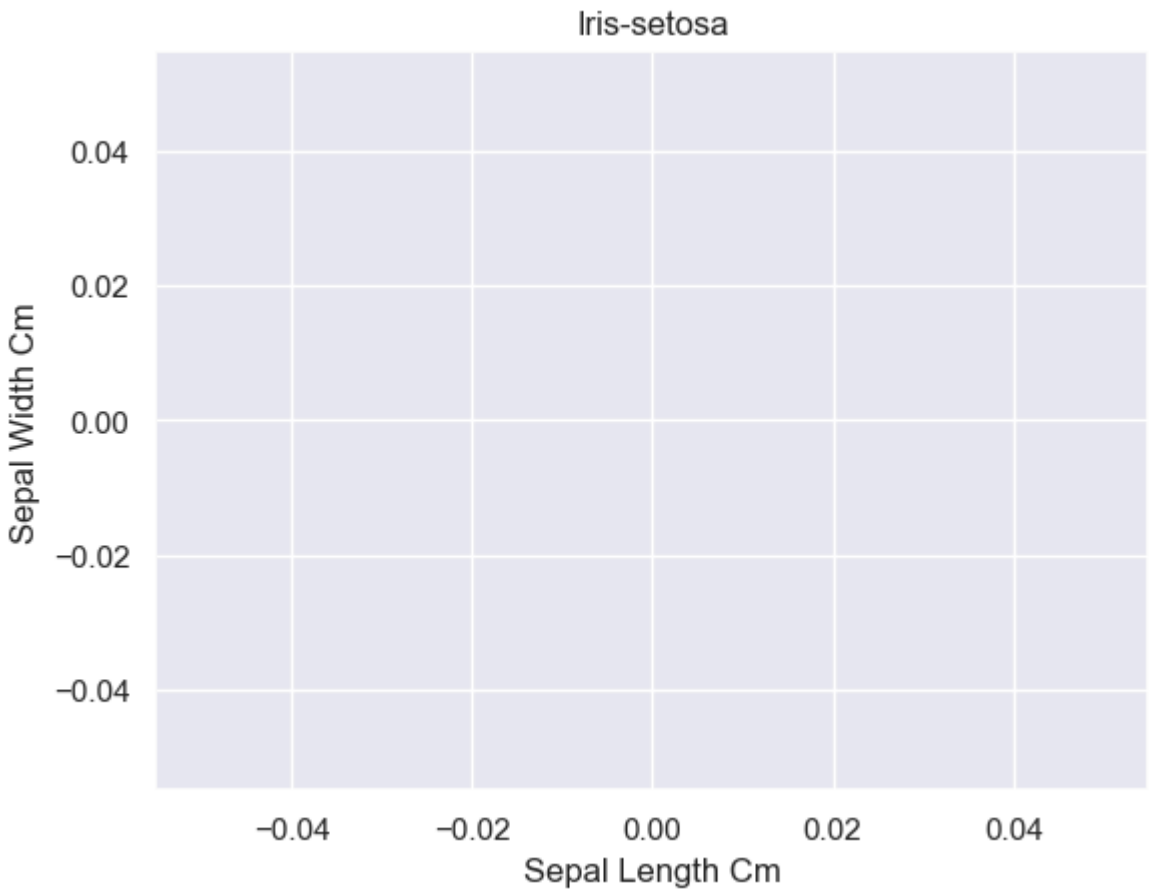


In [41]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxenplot(x='Species',y='SepalLengthCm',data=df)
```

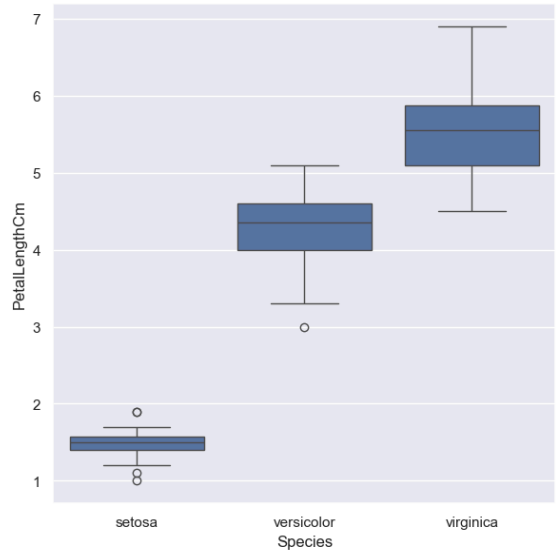


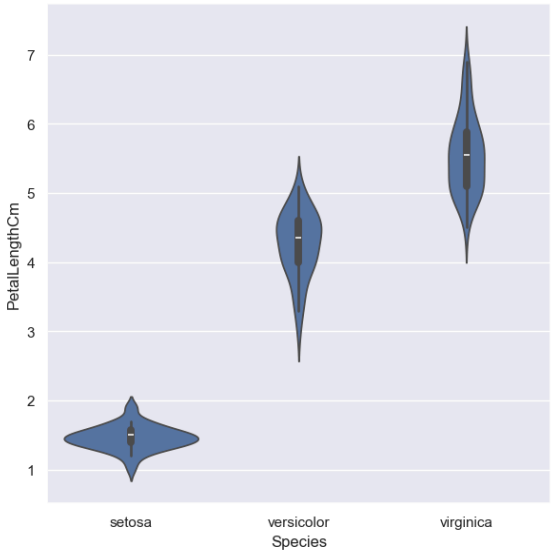
In [42]:
```
sub=df[df['Species']=='Iris-setosa']
sns.kdeplot(data=sub[['SepalLengthCm','SepalWidthCm']],cmap="plasma", shade=True
plt.title('Iris-setosa')
plt.xlabel('Sepal Length Cm')
plt.ylabel('Sepal Width Cm')
```

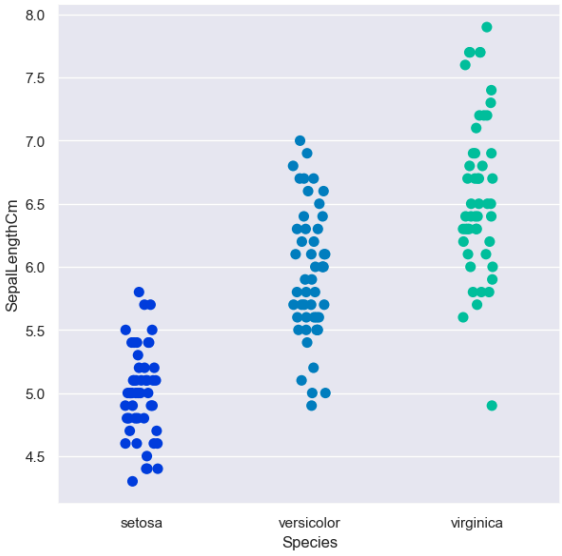
Out[42]:
```
Text(0, 0.5, 'Sepal Width Cm')
```

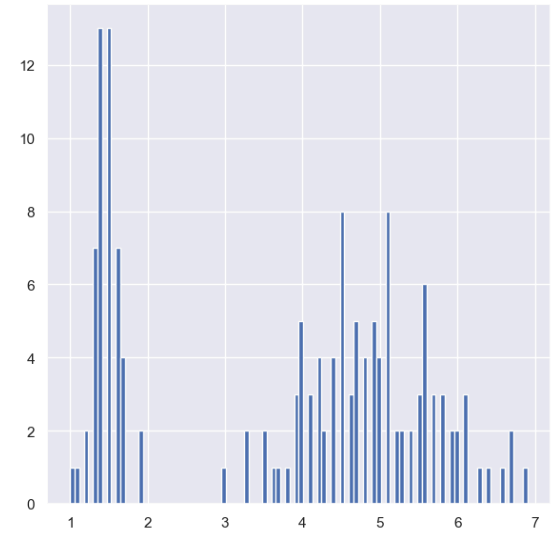


In [43]:
```
sns.set_style('darkgrid')
f,axes=plt.subplots(2,2,figsize=(15,15))

k1=sns.boxplot(x="Species", y="PetalLengthCm", data=df,ax=axes[0,0])
k2=sns.violinplot(x='Species',y='PetalLengthCm',data=df,ax=axes[0,1])
k3=sns.stripplot(x='Species',y='SepalLengthCm',data=df,jitter=True,edgecolor='gr
#axes[1,1].hist(iris.hist,bin=10)
axes[1,1].hist(df.PetalLengthCm,bins=100)
#k2.set(xlim=(-1,0.8))
plt.show()
```

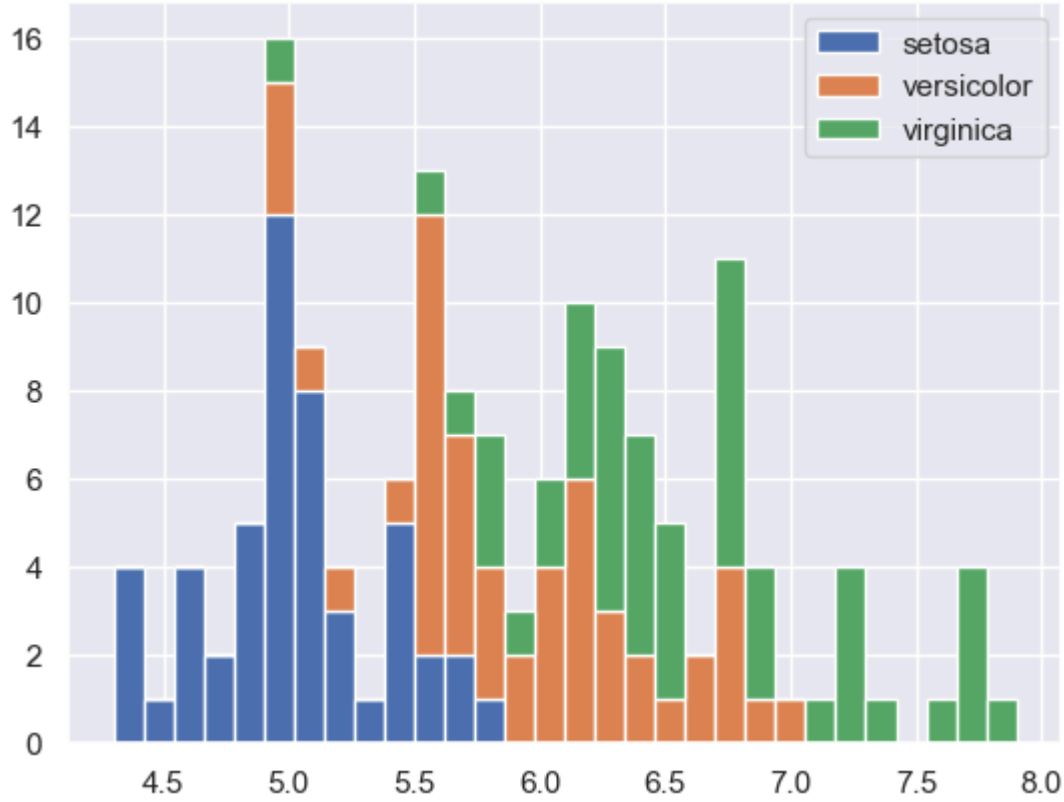








In [50]:
```
list1=list()
mylabels=list()
for gen in df.Species.cat.categories:
    list1.append(df[df.Species==gen].SepalLengthCm)
    mylabels.append(gen)

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



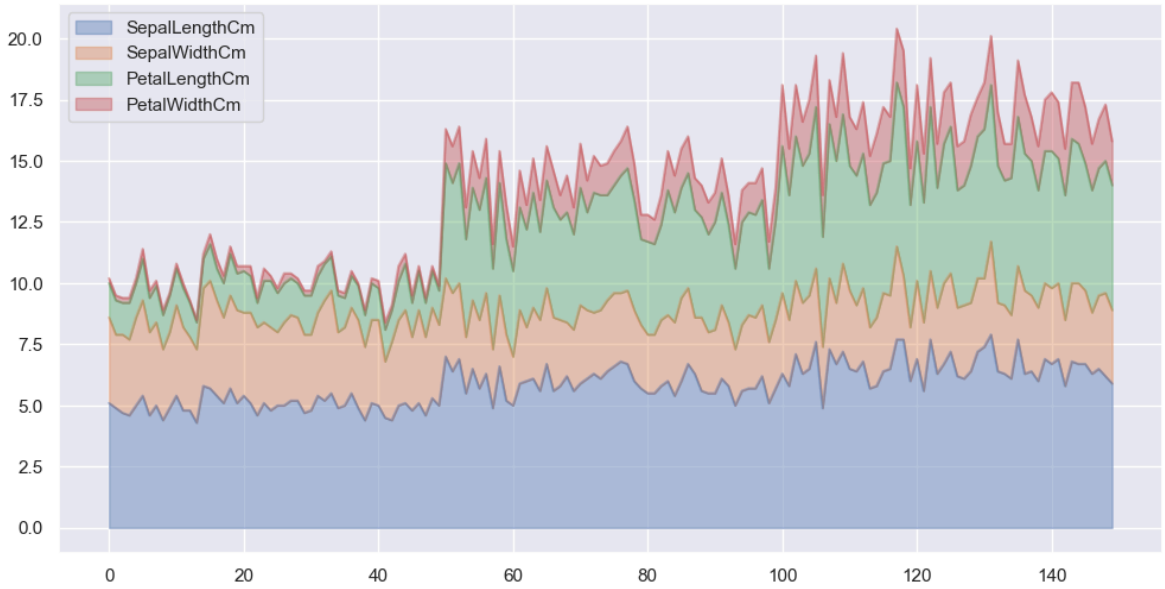
In [49]:
```
# Ensure the 'Species' column is of type 'category'
df['Species'] = df['Species'].astype('category')

# Now you can access the categorical properties
list1 = []
mylabels = []

for gen in df.Species.cat.categories:
    list1.append(df[df.Species == gen].SepalLengthCm)
    mylabels.append(gen)

print(list1, mylabels)
```

```
[0     5.1
1     4.9
2     4.7
3     4.6
4     5.0
5     5.4
6     4.6
7     5.0
8     4.4
9     4.9
10    5.4
11    4.8
12    4.8
13    4.3
14    5.8
15    5.7
16    5.4
17    5.1
18    5.7
19    5.1
20    5.4
21    5.1
22    4.6
23    5.1
24    4.8
25    5.0
26    5.0
27    5.2
28    5.2
29    4.7
30    4.8
31    5.4
32    5.2
33    5.5
34    4.9
35    5.0
36    5.5
37    4.9
38    4.4
39    5.1
40    5.0
41    4.5
42    4.4
43    5.0
44    5.1
45    4.8
46    5.1
47    4.6
48    5.3
49    5.0
Name: SepalLengthCm, dtype: float64, 50    7.0
51    6.4
52    6.9
53    5.5
54    6.5
55    5.7
56    6.3
57    4.9
58    6.6
59    5.2
```

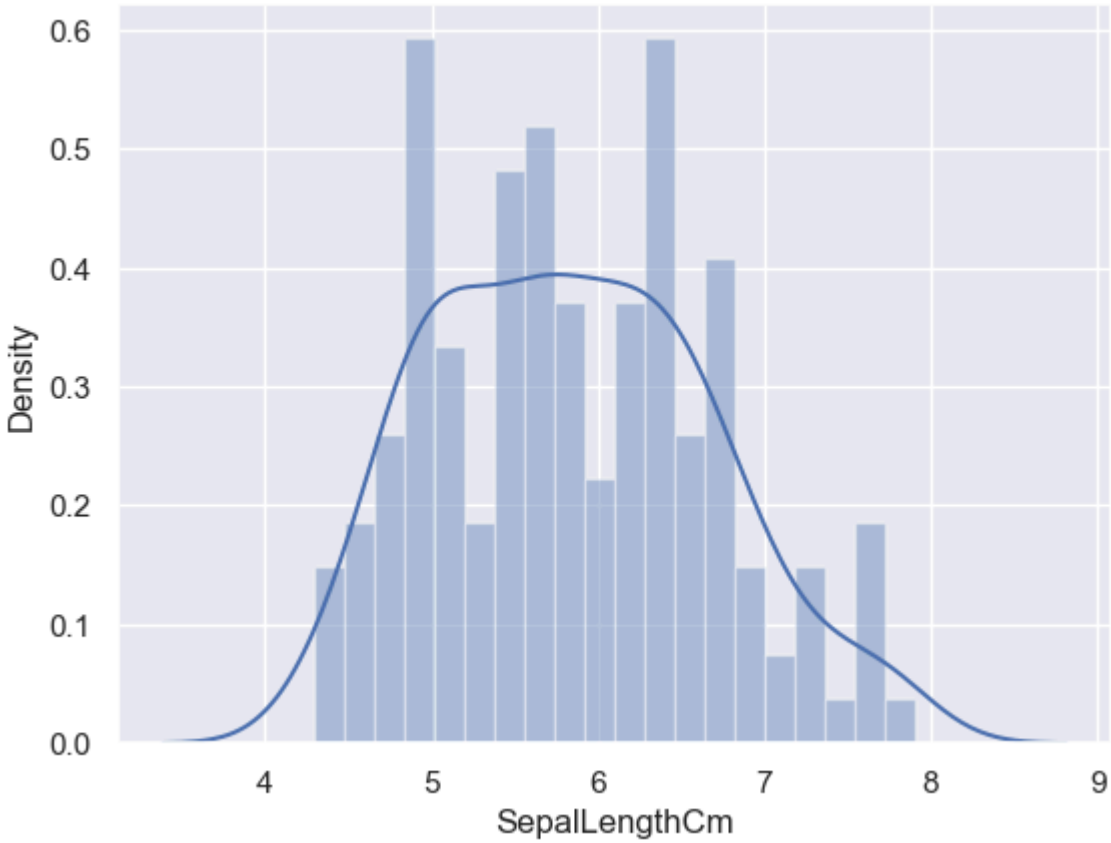
```
60    5.0
61    5.9
62    6.0
63    6.1
64    5.6
65    6.7
66    5.6
67    5.8
68    6.2
69    5.6
70    5.9
71    6.1
72    6.3
73    6.1
74    6.4
75    6.6
76    6.8
77    6.7
78    6.0
79    5.7
80    5.5
81    5.5
82    5.8
83    6.0
84    5.4
85    6.0
86    6.7
87    6.3
88    5.6
89    5.5
90    5.5
91    6.1
92    5.8
93    5.0
94    5.6
95    5.7
96    5.7
97    6.2
98    5.1
99    5.7
Name: SepalLengthCm, dtype: float64, 100    6.3
101    5.8
102    7.1
103    6.3
104    6.5
105    7.6
106    4.9
107    7.3
108    6.7
109    7.2
110    6.5
111    6.4
112    6.8
113    5.7
114    5.8
115    6.4
116    6.5
117    7.7
118    7.7
119    6.0
```

```
120    6.9
121    5.6
122    7.7
123    6.3
124    6.7
125    7.2
126    6.2
127    6.1
128    6.4
129    7.2
130    7.4
131    7.9
132    6.4
133    6.3
134    6.1
135    7.7
136    6.3
137    6.4
138    6.0
139    6.9
140    6.7
141    6.9
142    5.8
143    6.8
144    6.7
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: SepalLengthCm, dtype: float64] ['setosa', 'versicolor', 'virginica']
```

In [51]:
```
df.plot.area(y=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'],a
```



In [52]:
```
sns.distplot(df['SepalLengthCm'],kde=True,bins=20);
```


0.6
0.5
0.4
0.3
0.2
0.1
0.0
Density
4
5
6
7
8
9
SepalLengthCm