

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

```
In [2]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#plt.style.use('fivethirtyeight')
import warnings
warnings.filterwarnings('ignore') #this will ignore the warnings.it wont display w
```

```
In [3]: iris=pd.read_csv(r'C:\DS & GEN AI\Seaborn\Class Notes\Iris.csv')
```

```
In [4]: iris
```

```
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]: len(iris)
```

```
Out[5]: 150
```

```
In [6]: iris.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 [7]: iris.drop('Id',axis=1,inplace=True)
```

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

```
Out[8]:
```

	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 [9]: iris.tail()
```

```
Out[9]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
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

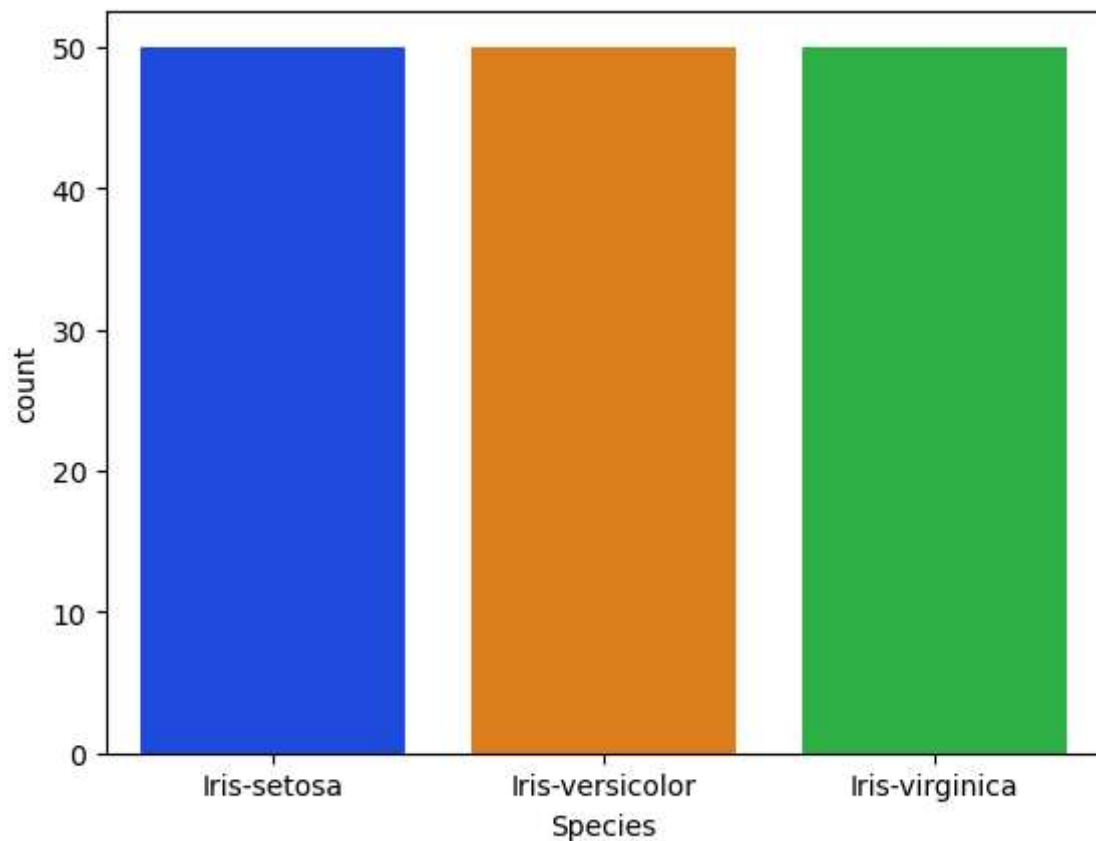
```
In [10]: iris.info()    # Checking if there are any missing values
```

```
<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 [11]: iris['Species'].value_counts()
```

```
Out[11]: Species  
Iris-setosa      50  
Iris-versicolor  50  
Iris-virginica   50  
Name: count, dtype: int64
```

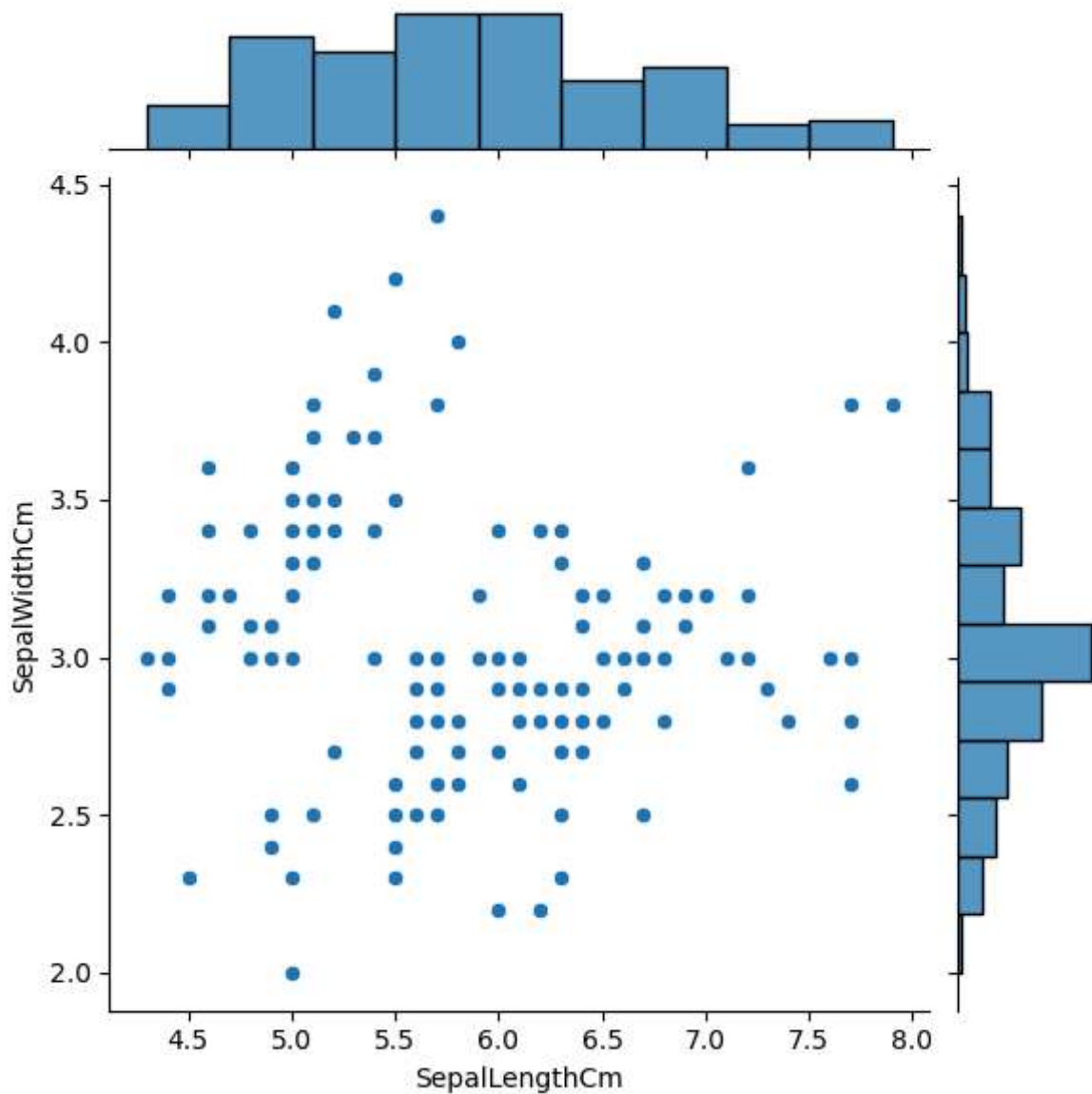
```
In [12]: sns.countplot(x='Species',data=iris,palette='bright')  
plt.show()
```

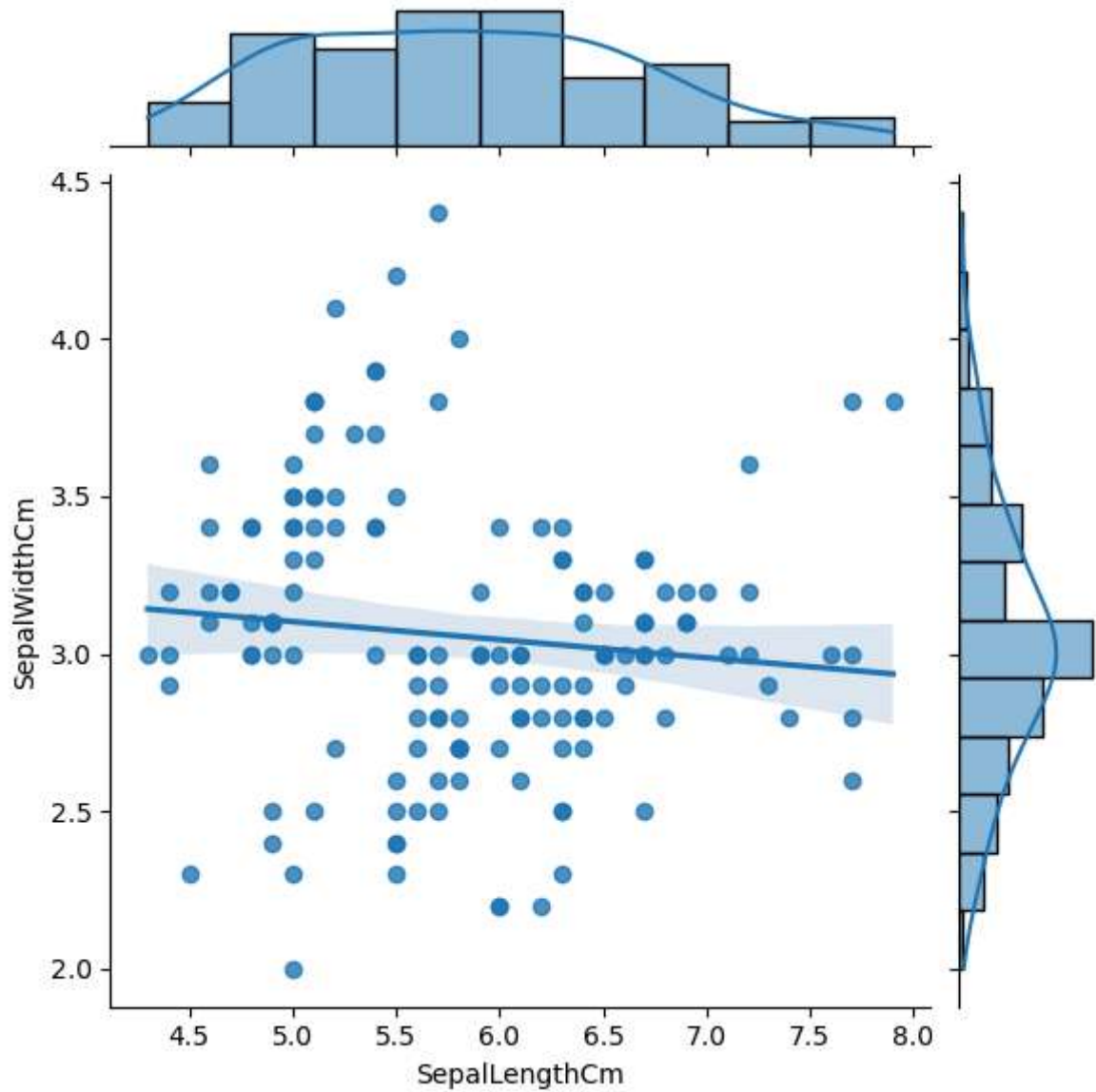


```
In [13]: iris.head()
```

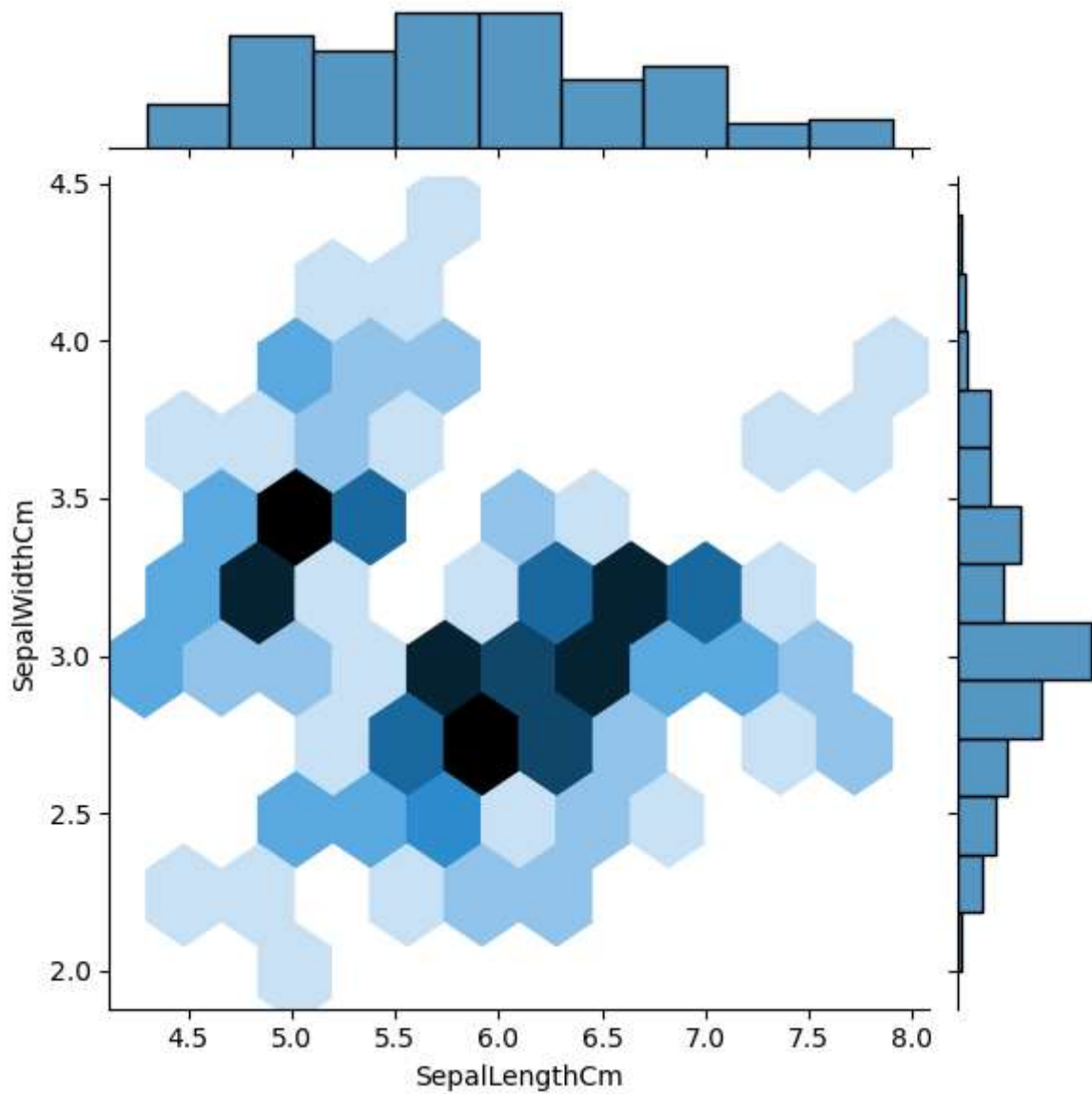
Out[13]:

	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 [14]: `fig=sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',data=iris)`In [15]: `sns.jointplot(x="SepalLengthCm", y="SepalWidthCm", data=iris, kind="reg")`Out[15]: `<seaborn.axisgrid.JointGrid at 0x276e96cdbc0>`

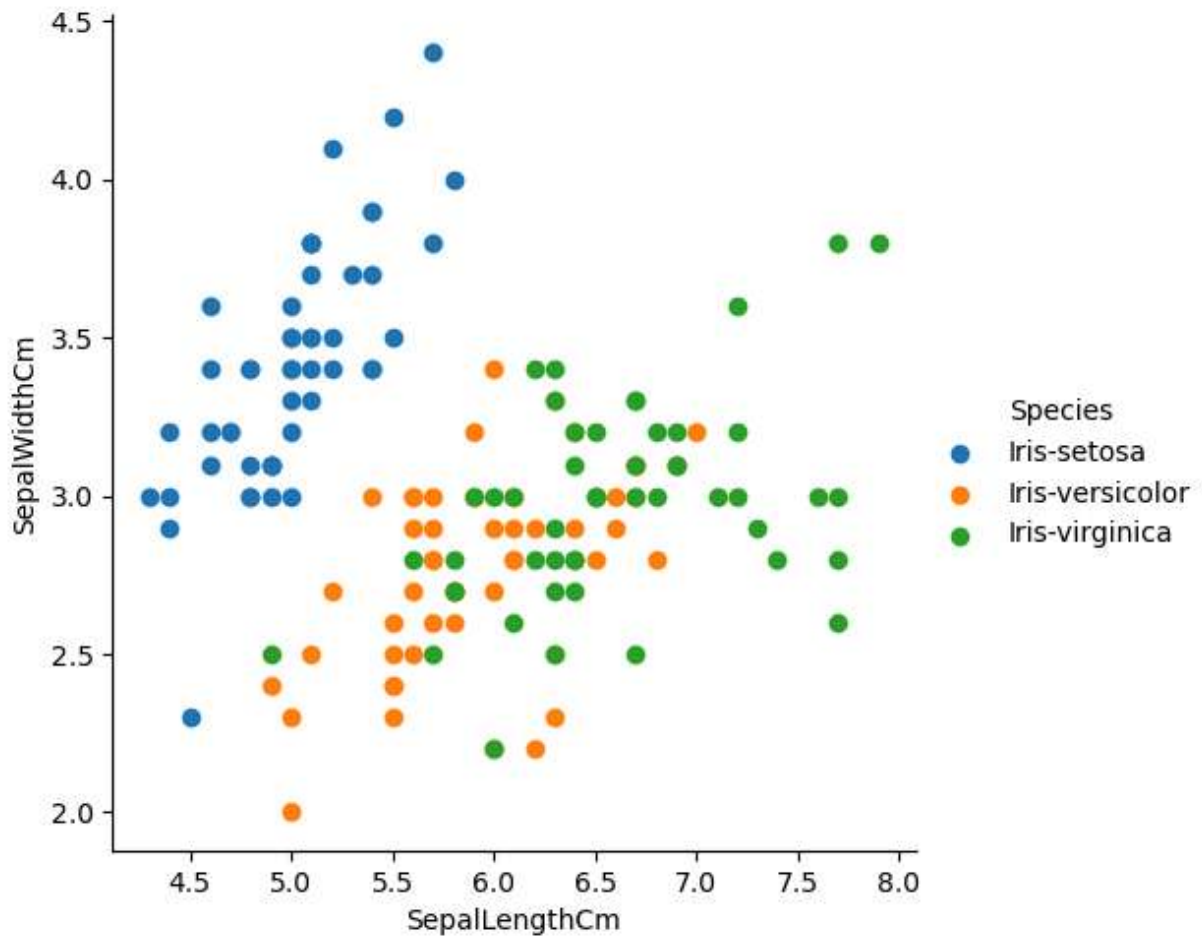


```
In [16]: fig=sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',kind='hex',data=iris)
```



```
In [17]: import matplotlib.pyplot as plt
%matplotlib inline

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

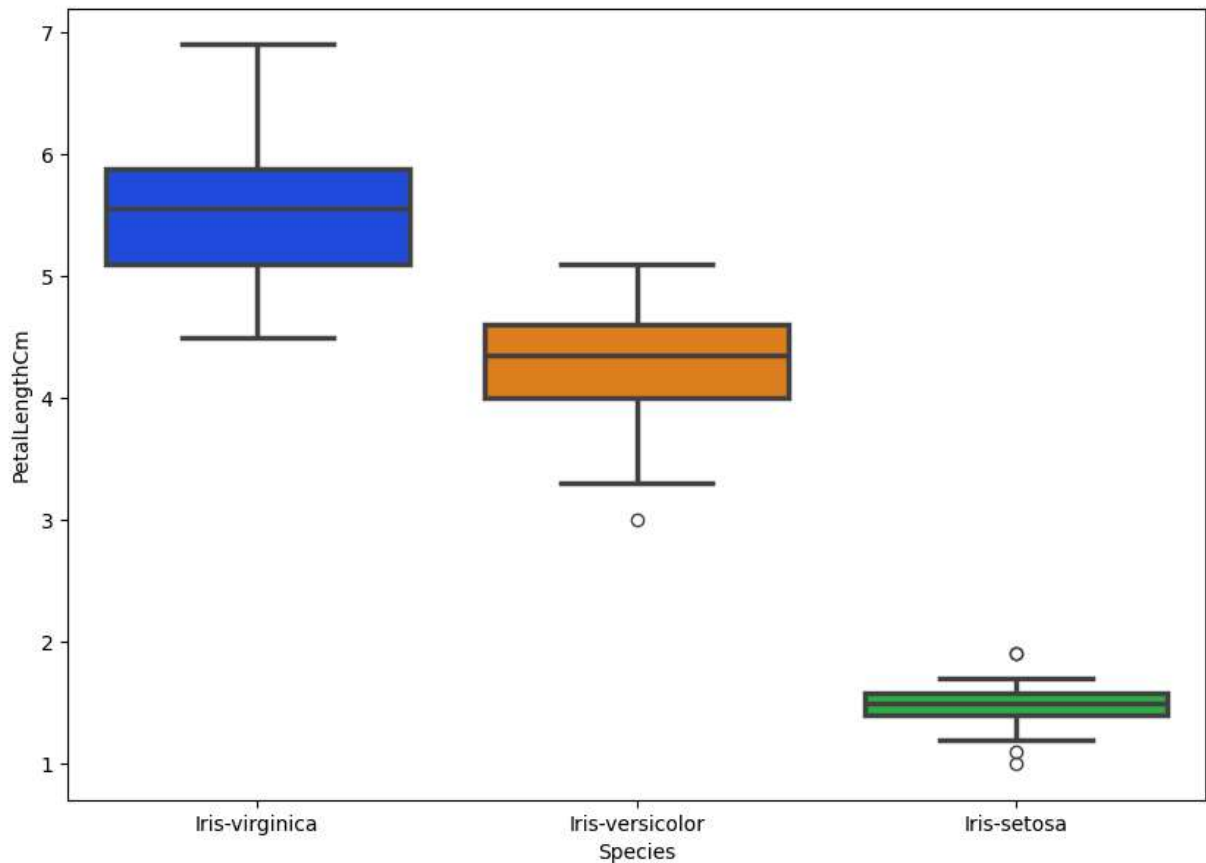


```
In [18]: iris.head()
```

```
Out[18]:
```

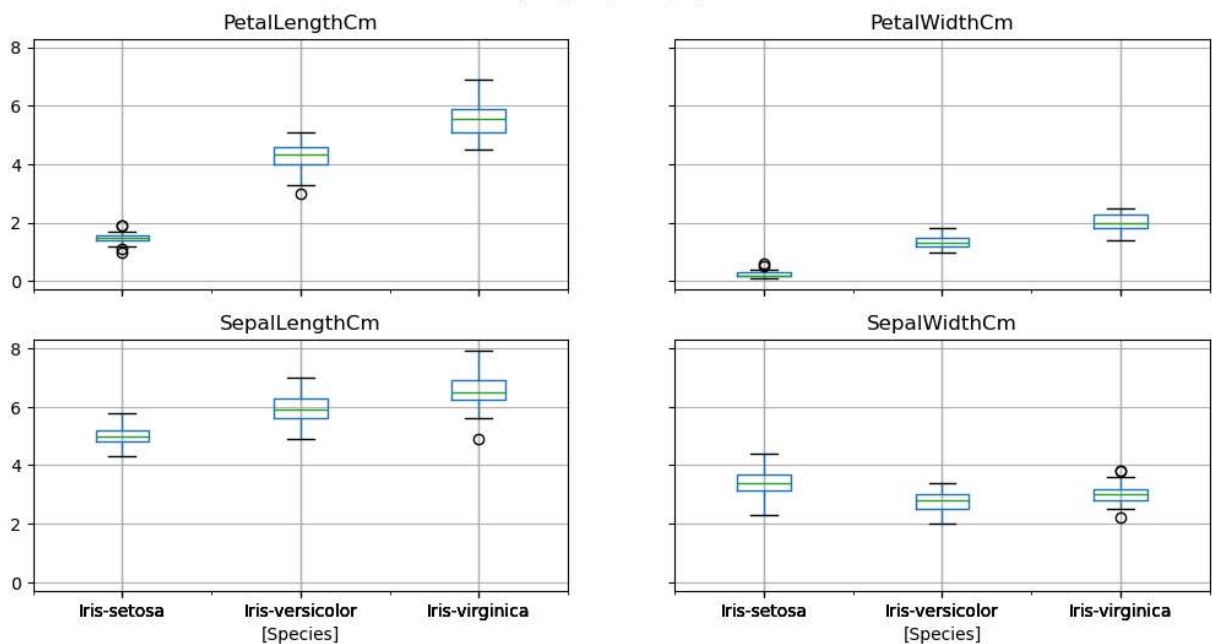
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
In [19]: fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='PetalLengthCm',data=iris,order=['Iris-virginica','Iris-versicolor','Iris-setosa'])
plt.show()
```

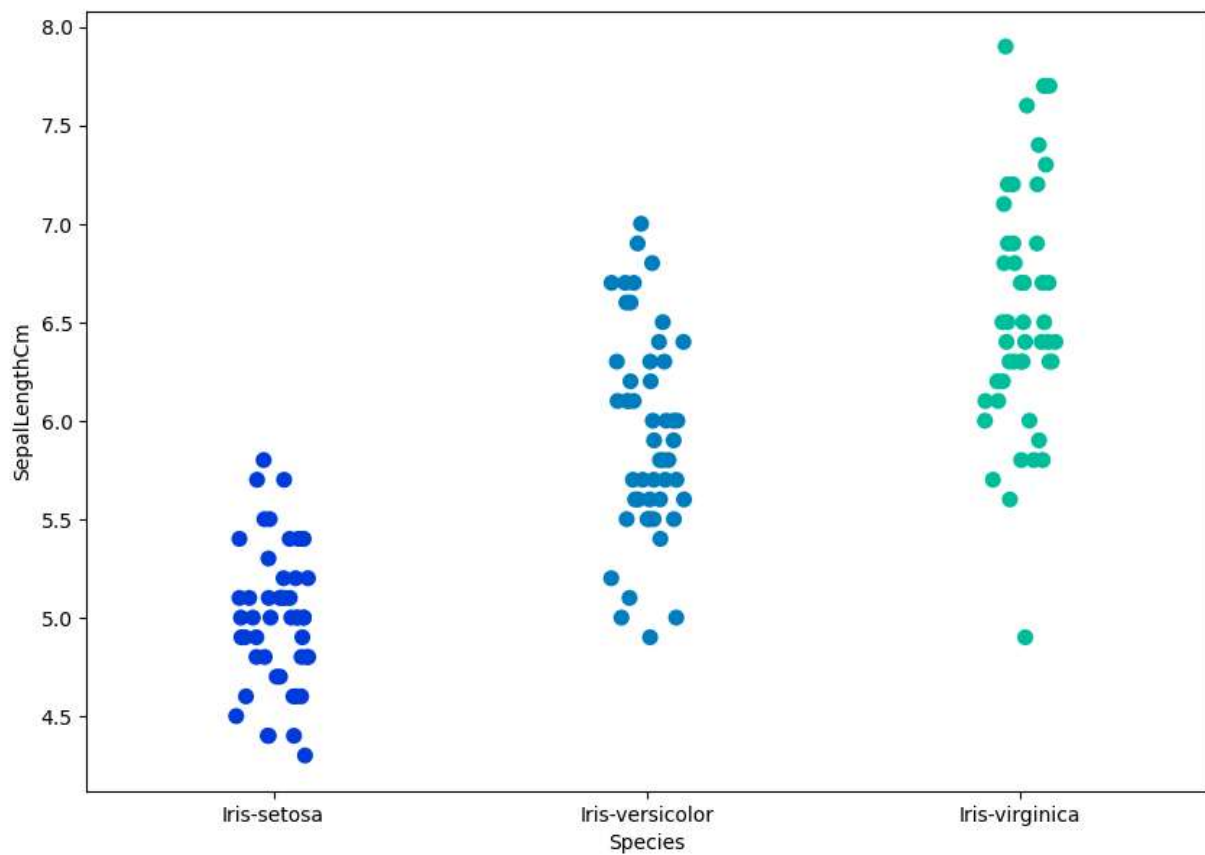


```
In [20]: #iris.drop("Id", axis=1).boxplot(by="Species", figsize=(12, 6))
iris.boxplot(by="Species", figsize=(12, 6))
plt.show()
```

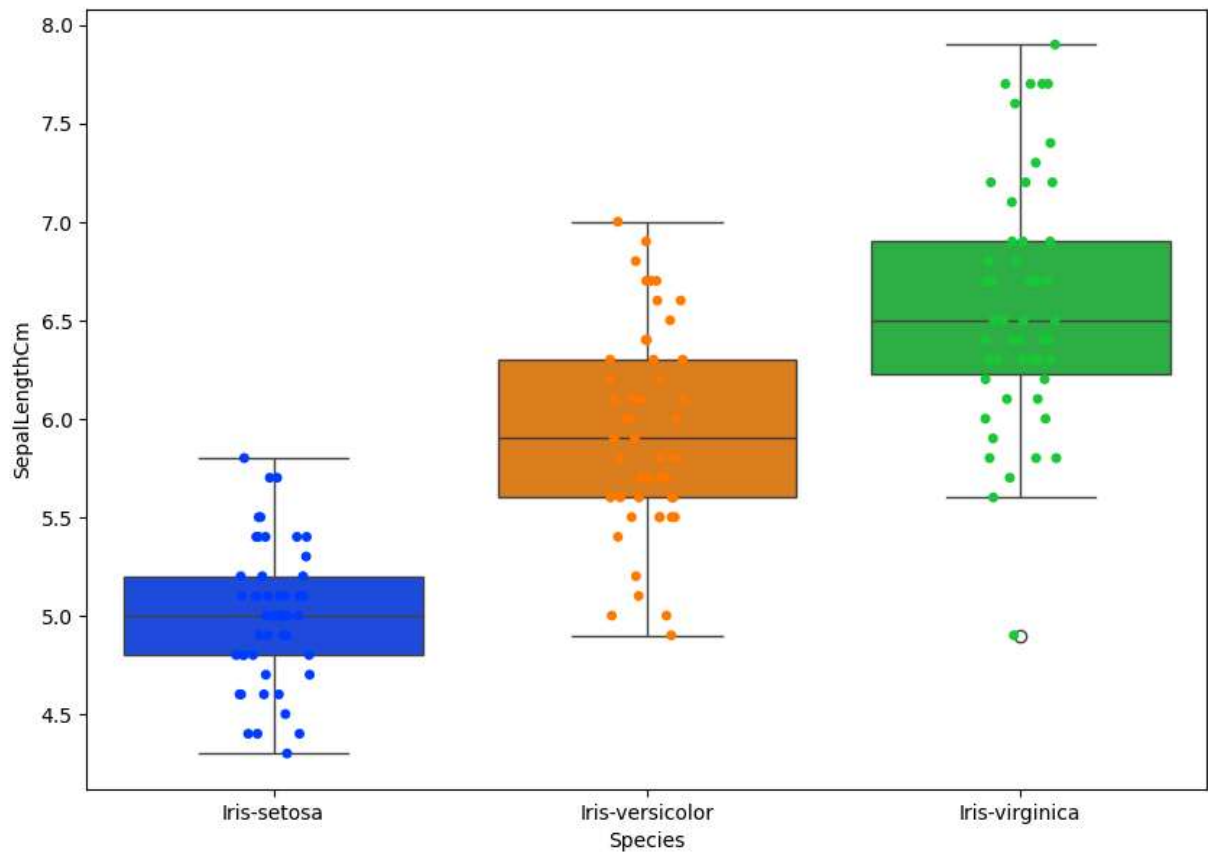
Boxplot grouped by Species



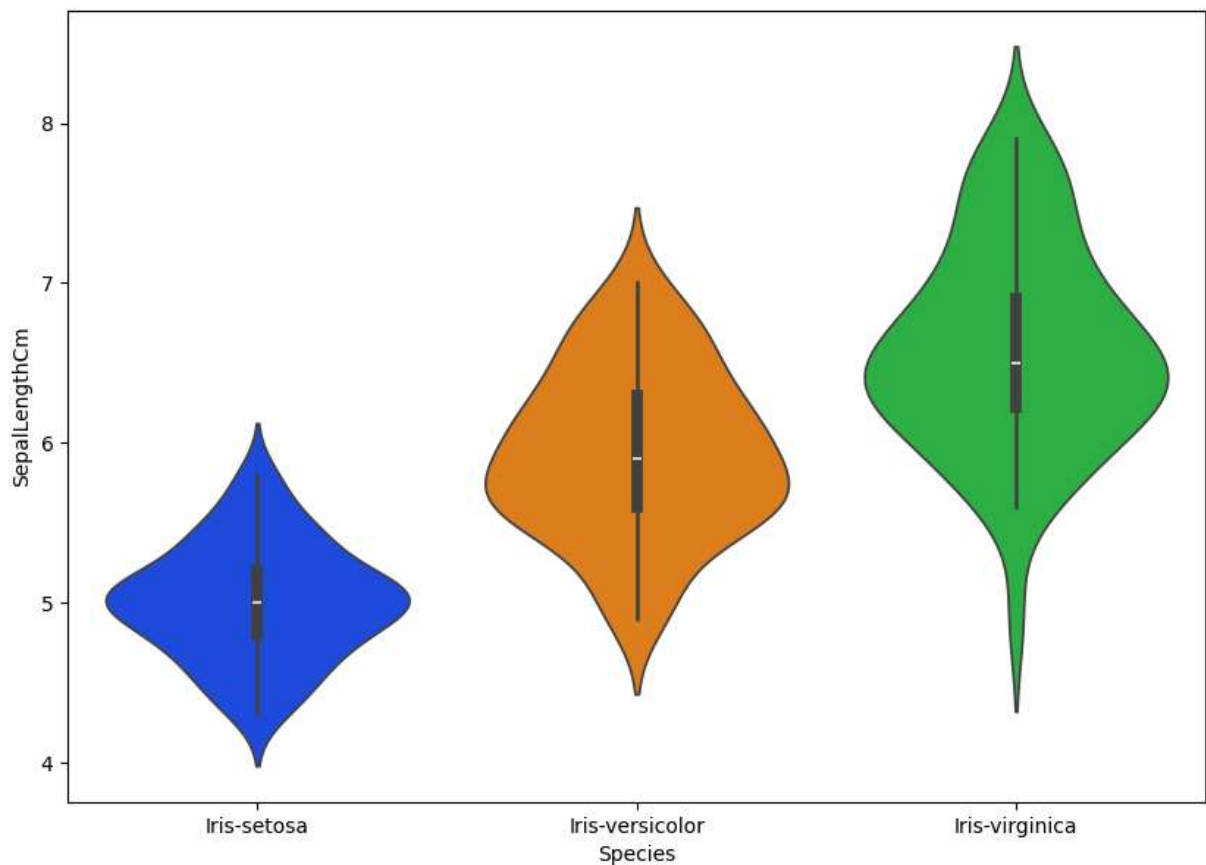
```
In [21]: fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edgecolor='gr
plt.show()
```

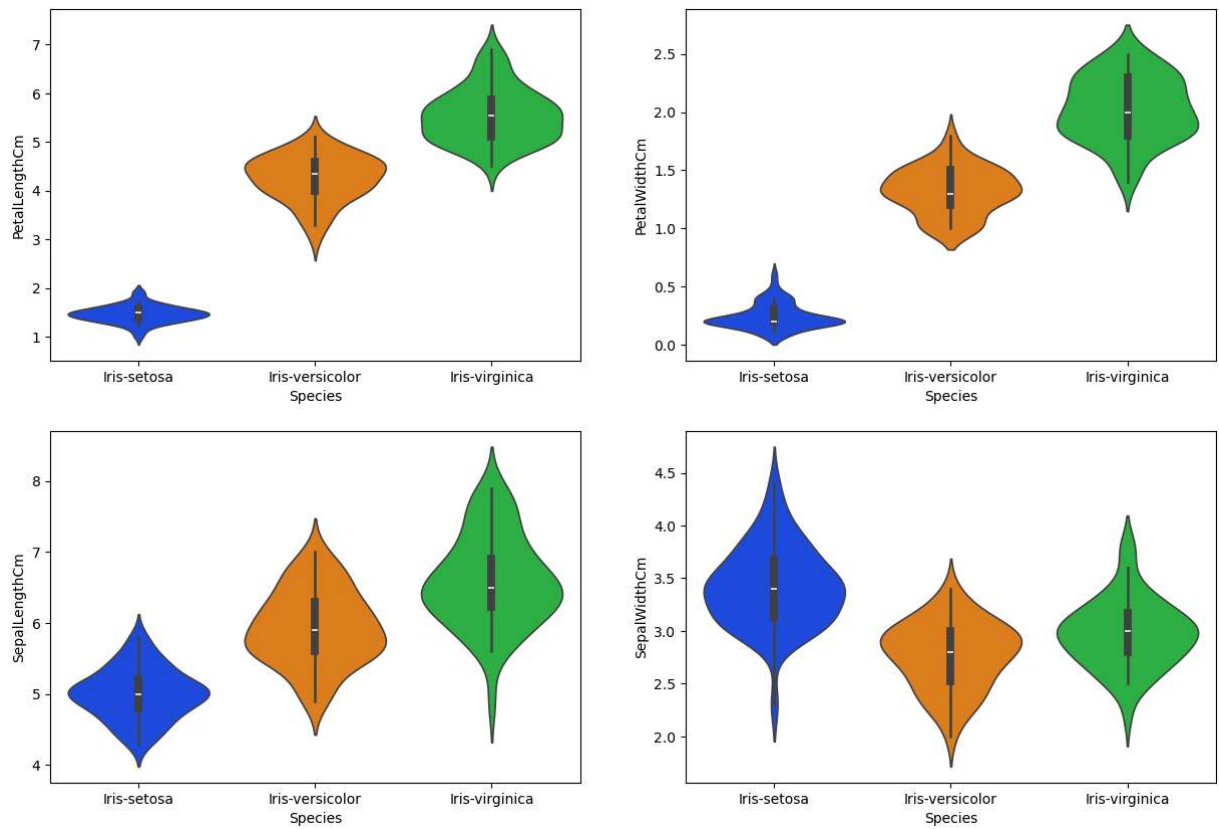
```
In [22]: fig=plt.gcf()                                # Combining box and strip plots
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='SepalLengthCm',data=iris,palette='bright')
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,palette='bright',jitter=True)
plt.show()
```



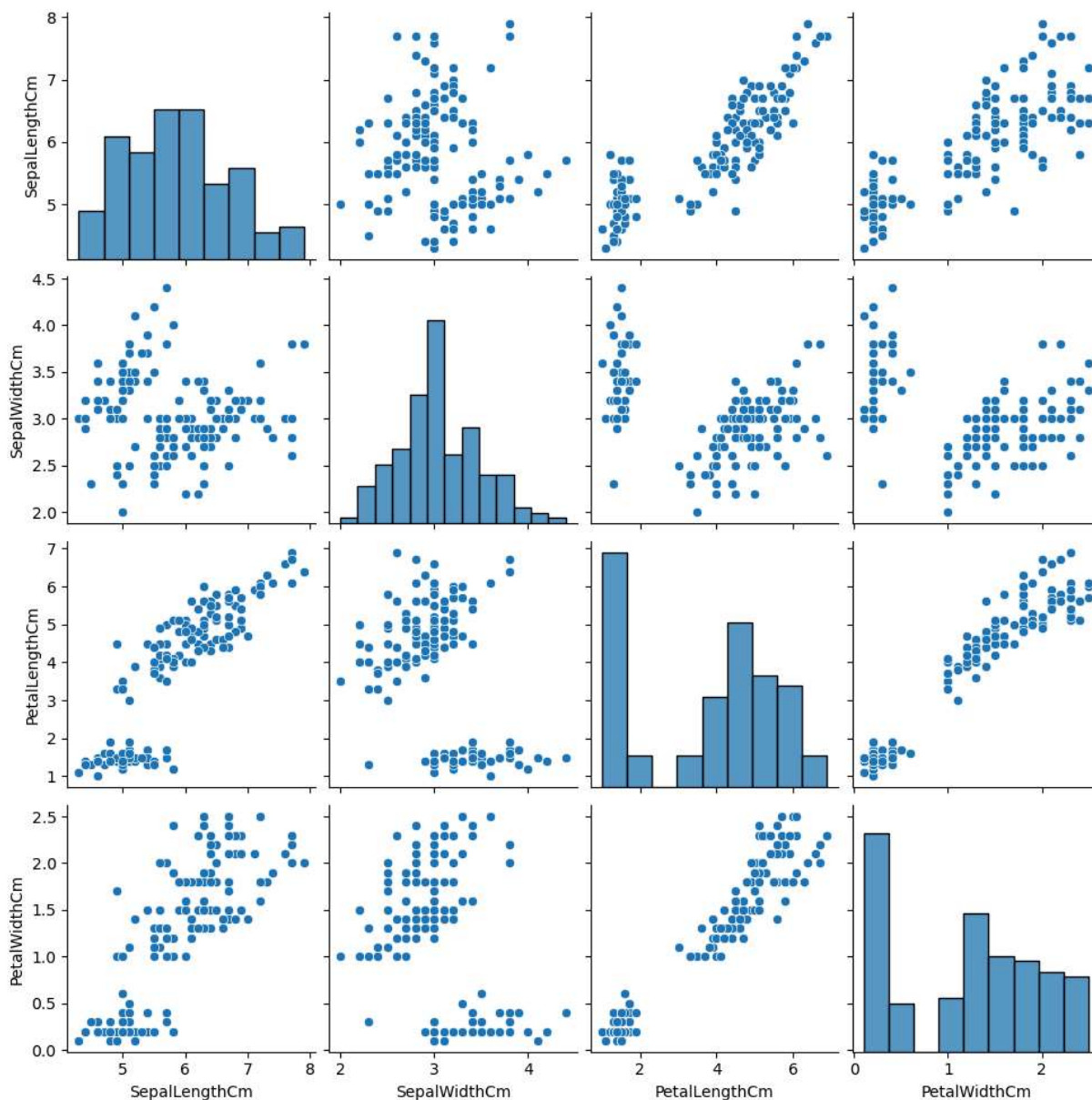
```
In [23]: fig=plt.gcf()                # Violin plot
fig.set_size_inches(10,7)
fig=sns.violinplot(x='Species',y='SepalLengthCm',palette='bright',data=iris)
plt.show()
```



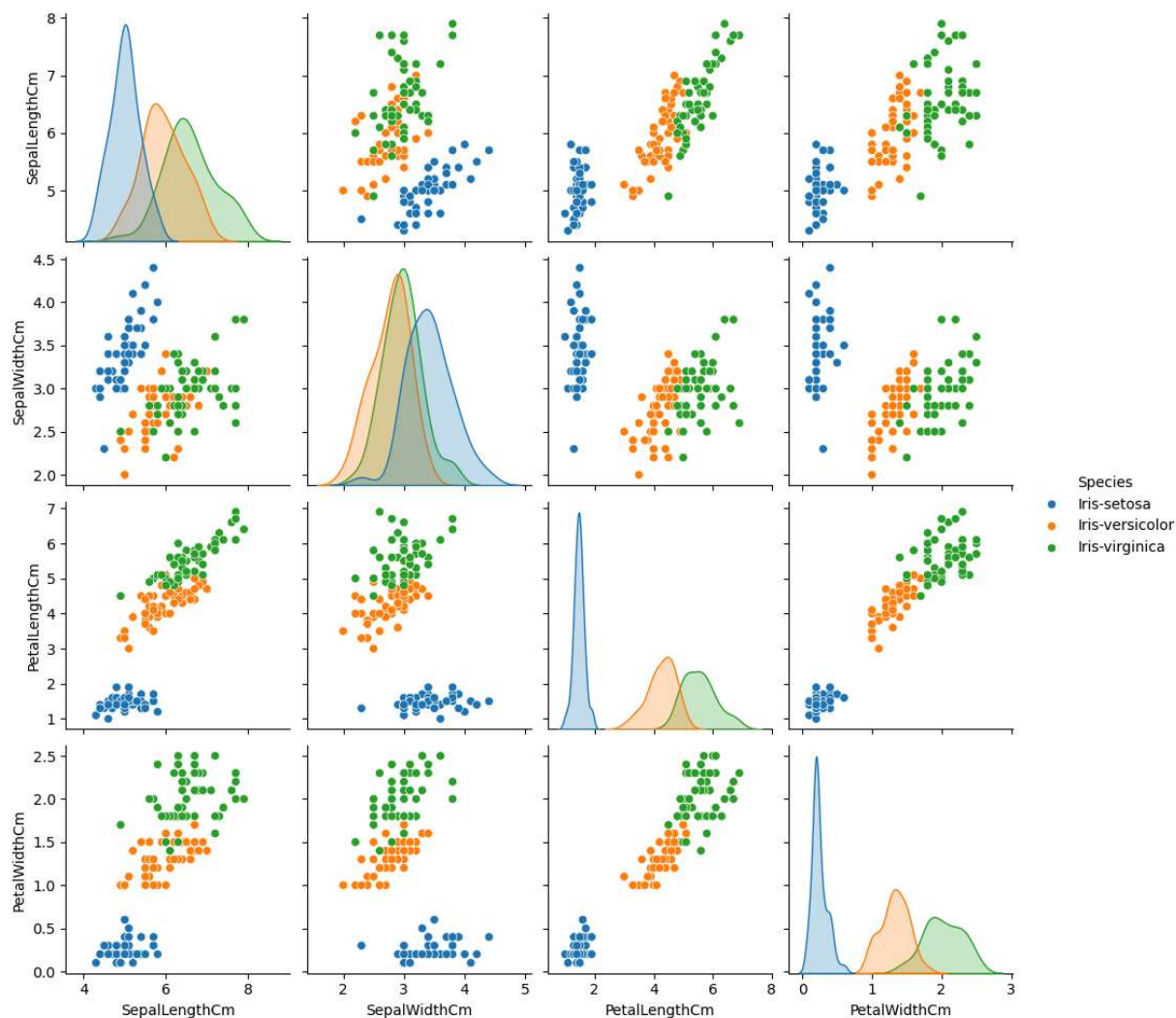
```
In [24]: plt.figure(figsize=(15,10))
plt.subplot(2,2,1)
sns.violinplot(x='Species',y='PetalLengthCm',data=iris,palette='bright')
plt.subplot(2,2,2)
sns.violinplot(x='Species',y='PetalWidthCm',data=iris,palette='bright')
plt.subplot(2,2,3)
sns.violinplot(x='Species',y='SepalLengthCm',data=iris,palette='bright')
plt.subplot(2,2,4)
sns.violinplot(x='Species',y='SepalWidthCm',data=iris,palette='bright')
plt.show()
```



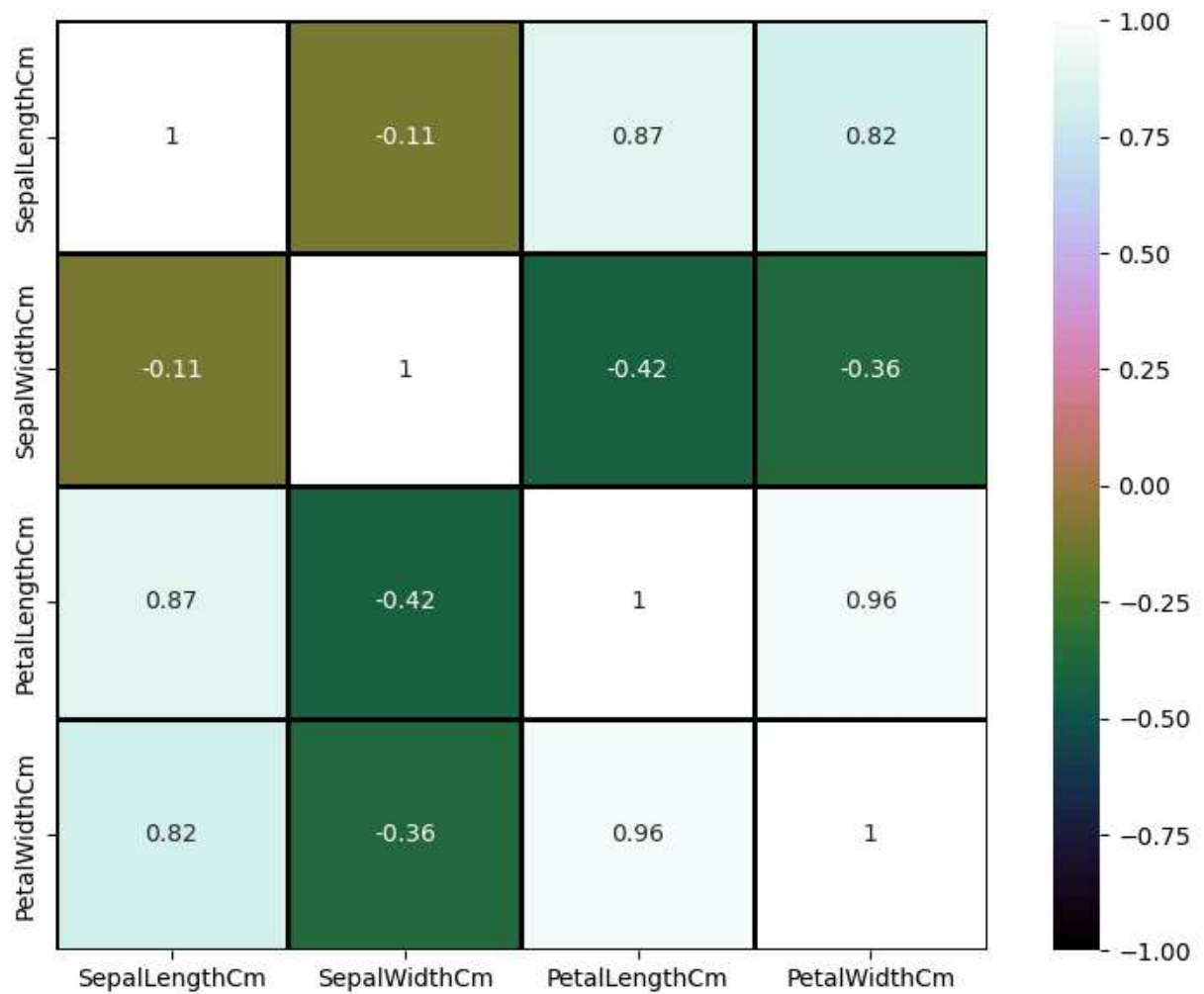
```
In [25]: sns.pairplot(data=iris,kind='scatter')  
plt.show()
```



```
In [26]: sns.pairplot(iris,hue='Species');  
plt.show()
```



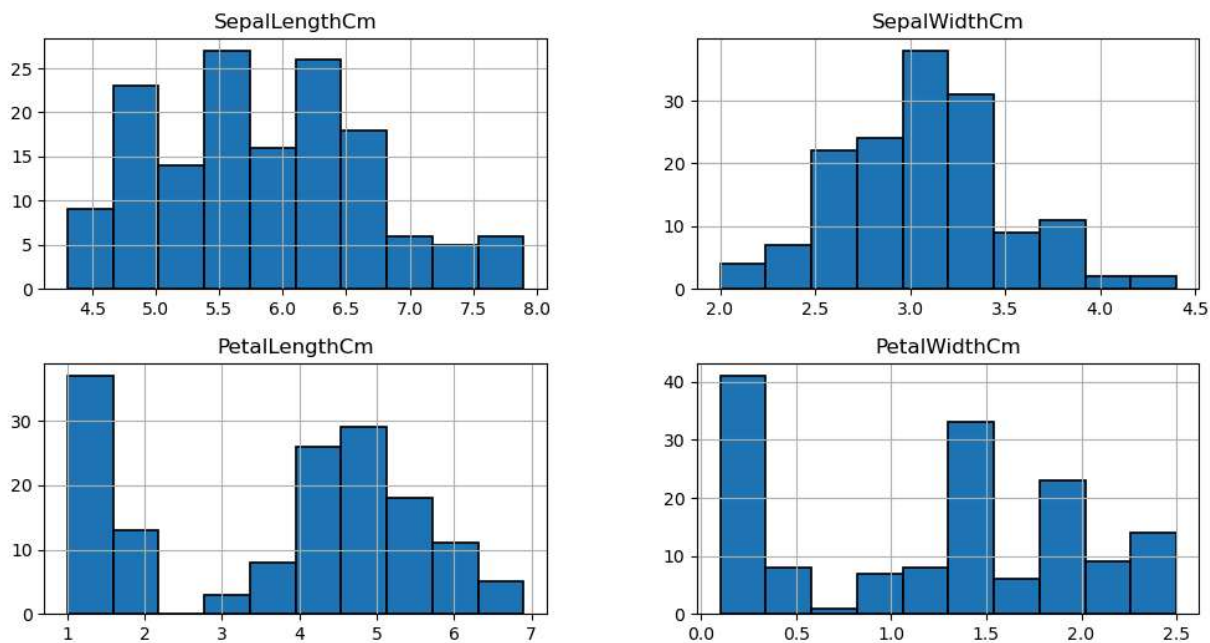
```
In [27]: fig=plt.gcf()
fig.set_size_inches(10,7)
corr = iris[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']].corr
fig=sns.heatmap(corr,annot=True,cmap='cubehelix',linewidths=1,linecolor='k',square=
plt.show())
```



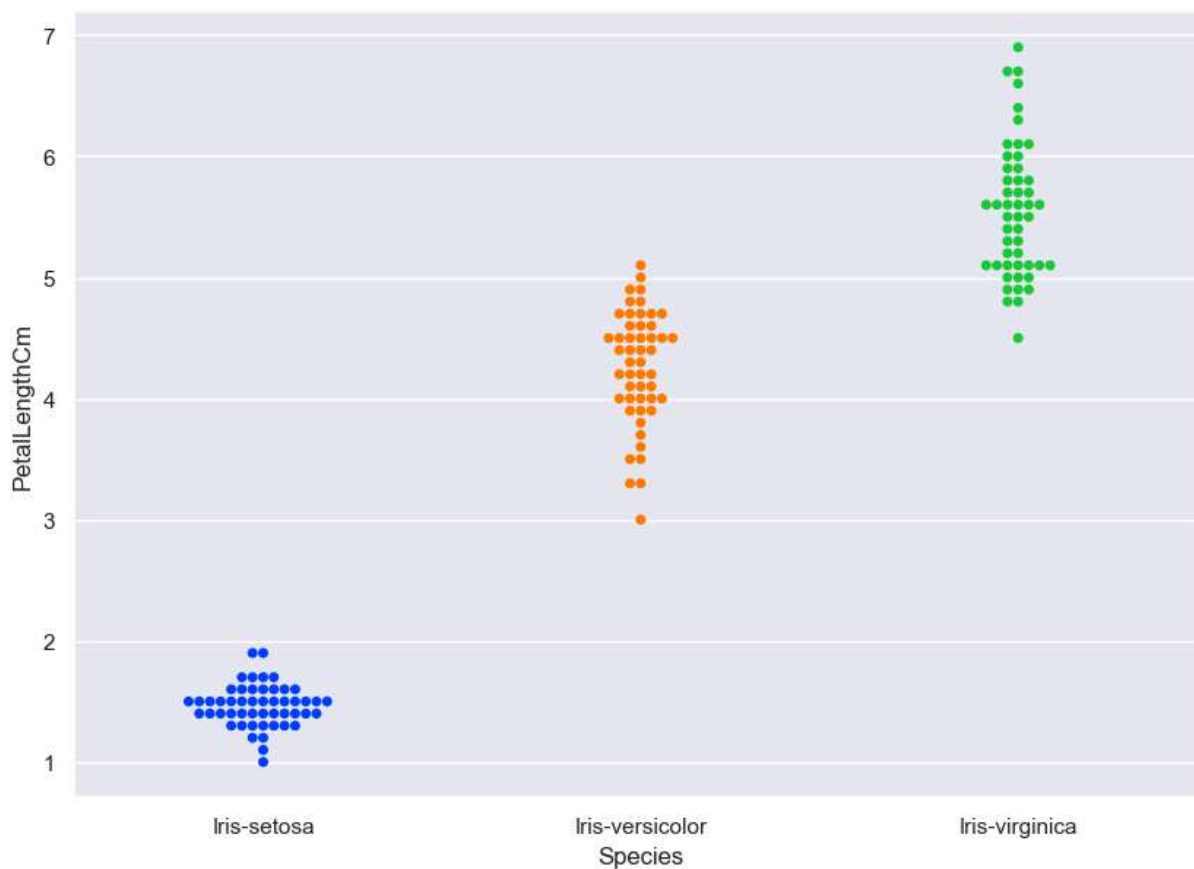
```
In [89]: iris['Species'].value_counts()
```

```
Out[89]: Species
Iris-setosa      50
Iris-versicolor  50
Iris-virginica   50
Name: count, dtype: int64
```

```
In [28]: iris.hist(edgecolor='black', linewidth=1.2)      # Distribution plot
fig=plt.gcf()
fig.set_size_inches(12,6)
plt.show()
```



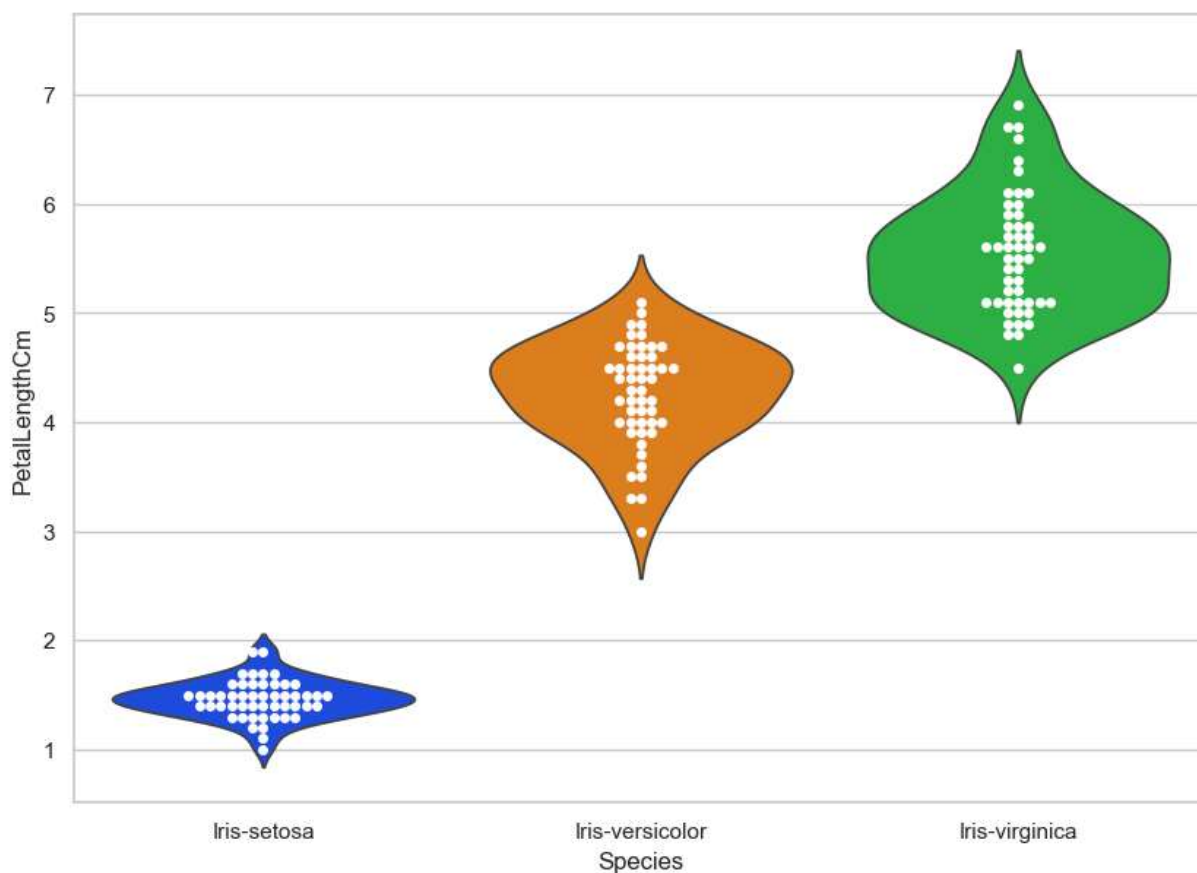
```
In [29]: sns.set(style="darkgrid")           # Swarm plot
fig=plt.gcf()
fig.set_size_inches(10,7)
fig = sns.swarmplot(x="Species", y="PetalLengthCm", data=iris,palette='bright')
plt.show()
```



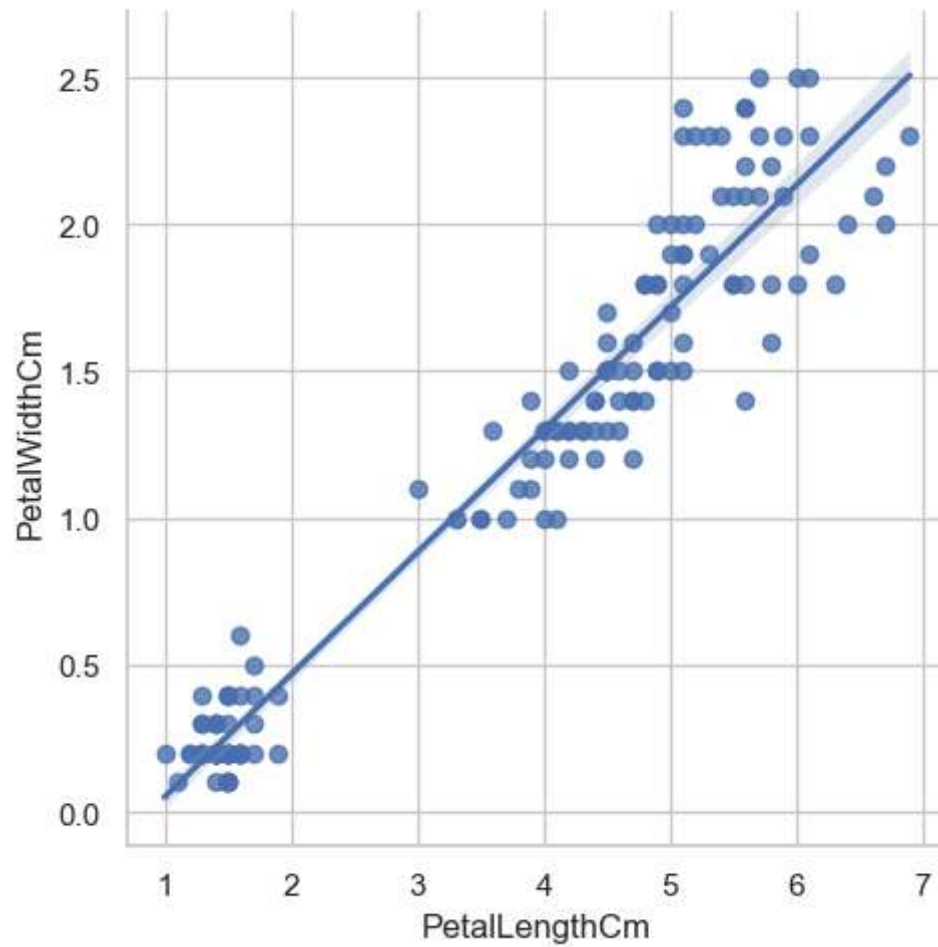
```
In [30]: sns.set(style="whitegrid")
fig=plt.gcf()
fig.set_size_inches(10,7)
```



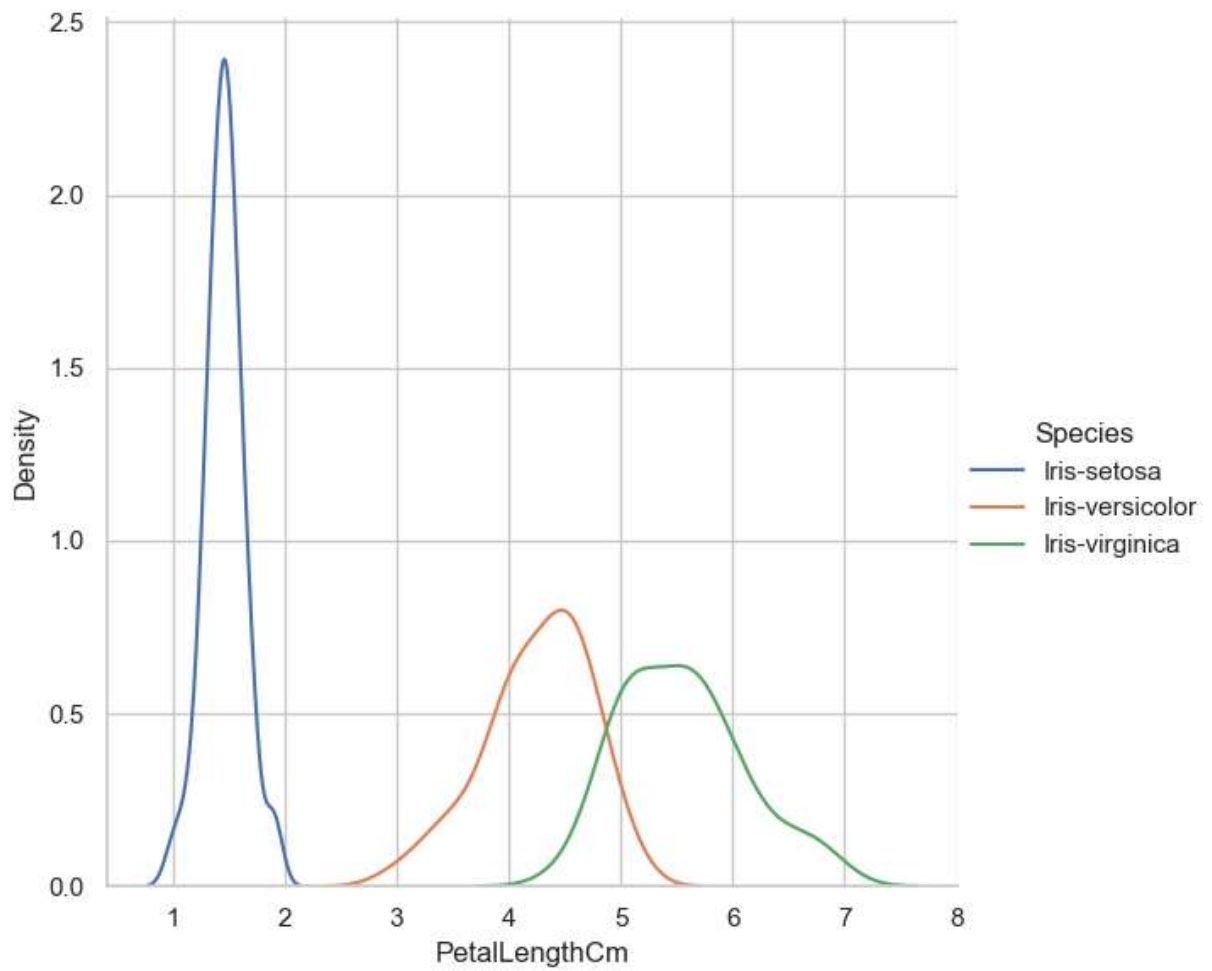
```
ax = sns.violinplot(x="Species", y="PetalLengthCm", data=iris,palette='bright',inner=True)
ax = sns.swarmplot(x="Species", y="PetalLengthCm", data=iris,color="white", edgecolor="black")
plt.show()
```



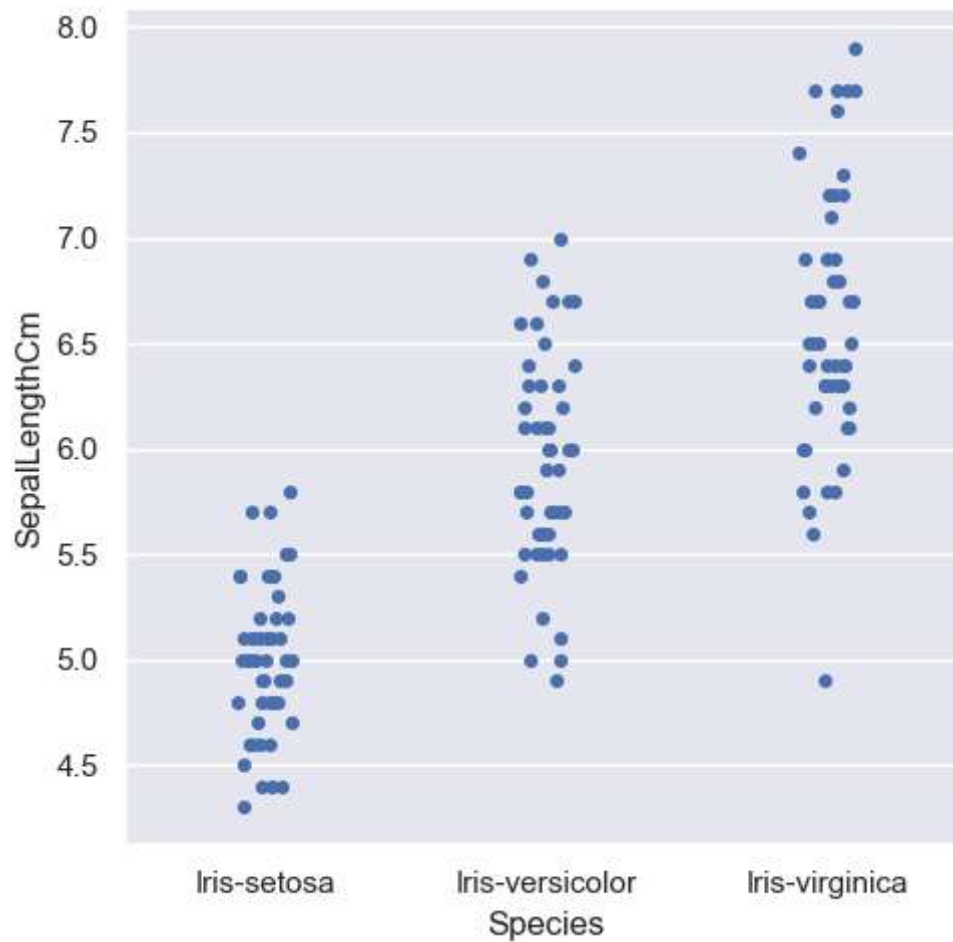
```
In [31]: fig=sns.lmplot(x="PetalLengthCm", y="PetalWidthCm",data=iris) # LM plot
plt.show()
```



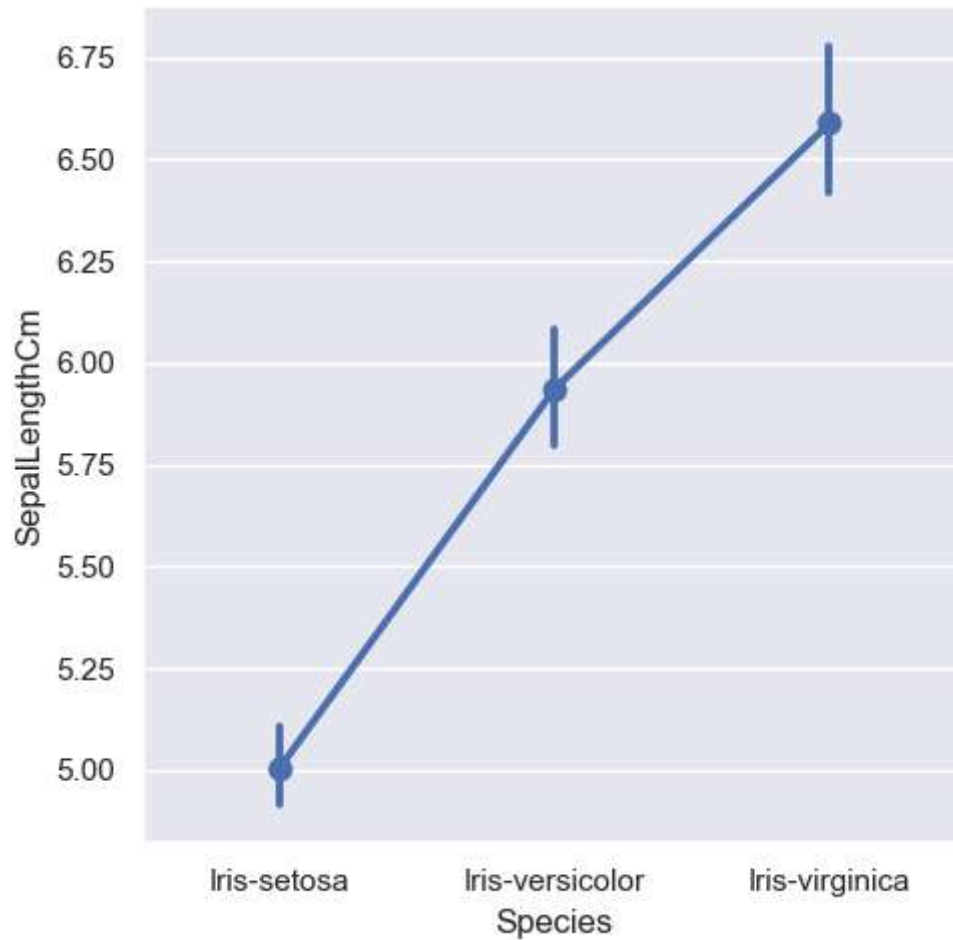
```
In [32]: g = sns.FacetGrid(iris, hue="Species",height=6) #FacetGrid
g.map(sns.kdeplot, "PetalLengthCm")
g.add_legend()
plt.show()
```



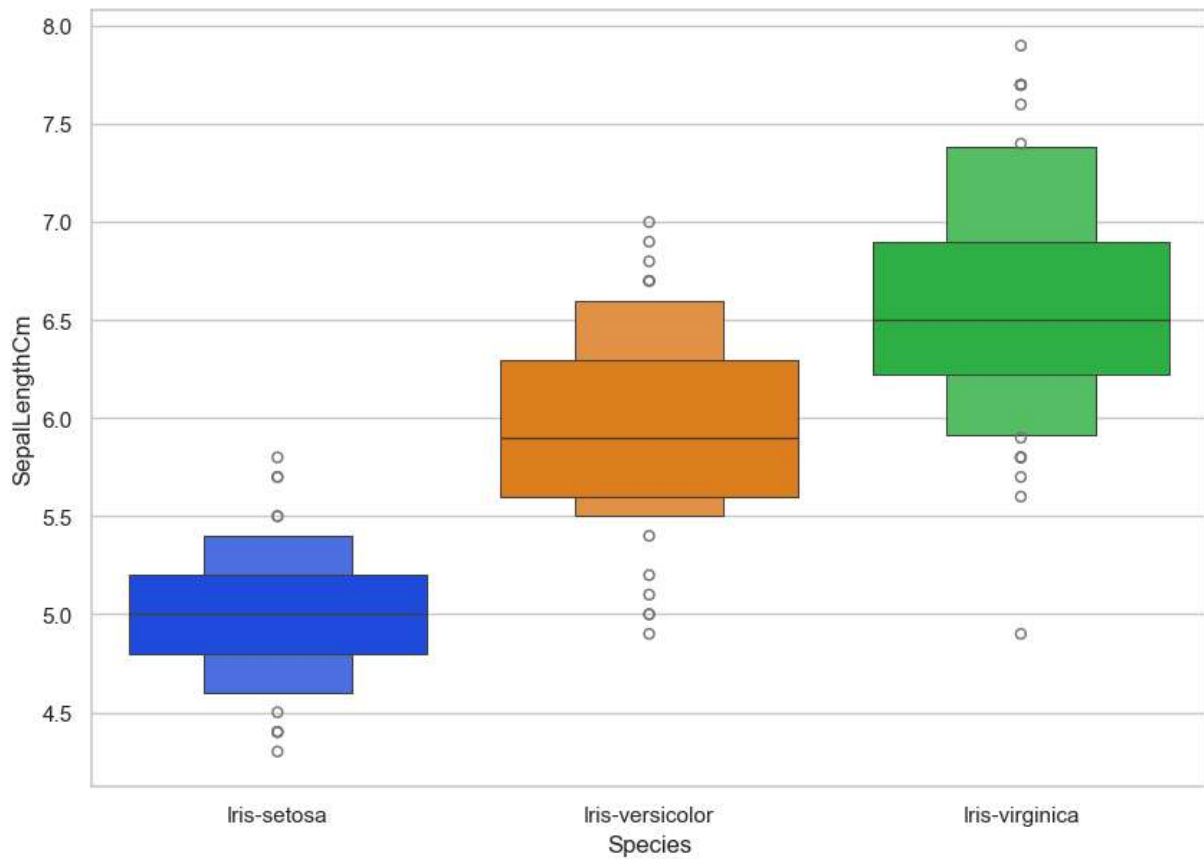
```
In [85]: sns.catplot(x='Species',y='SepalLengthCm', data=iris,)  
plt.show()
```



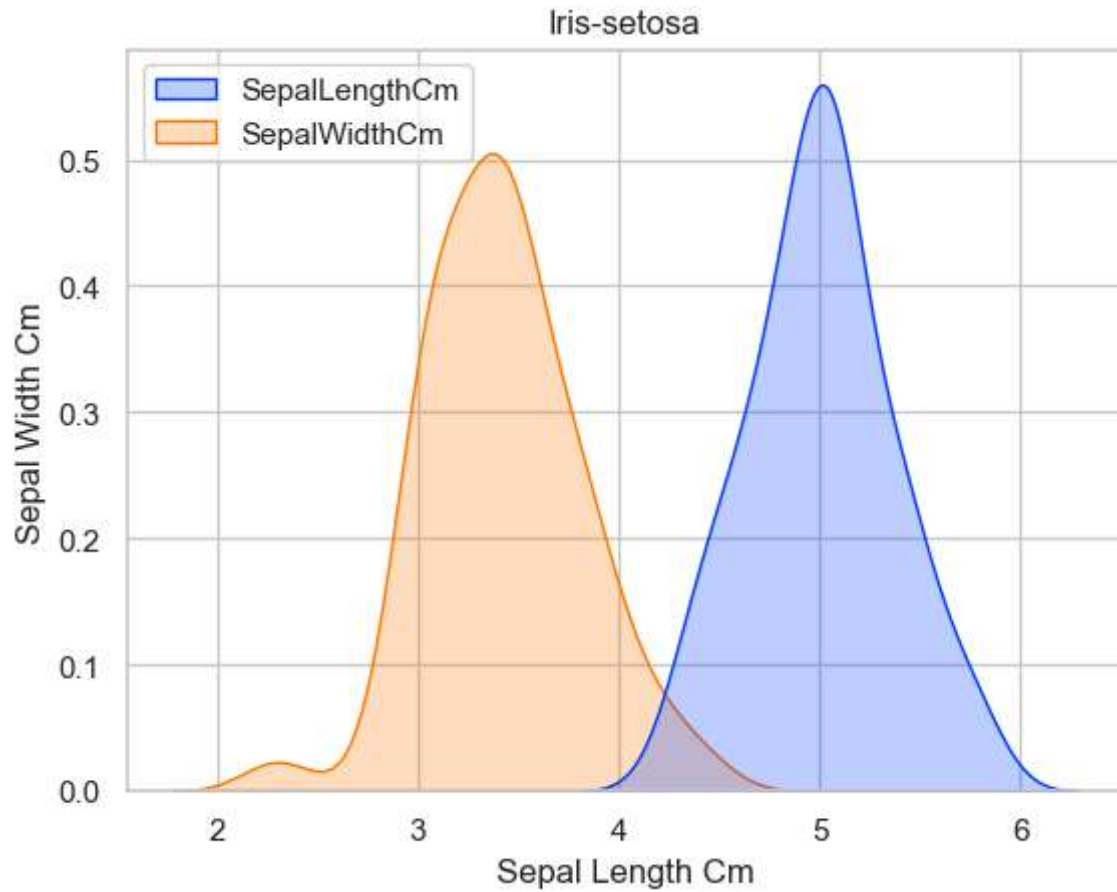
```
In [87]: sns.catplot(x='Species',y='SepalLengthCm', data=iris, kind='point')  
plt.show()
```



```
In [33]: fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxenplot(x='Species',y='SepalLengthCm',data=iris,palette='bright')
plt.show()
```

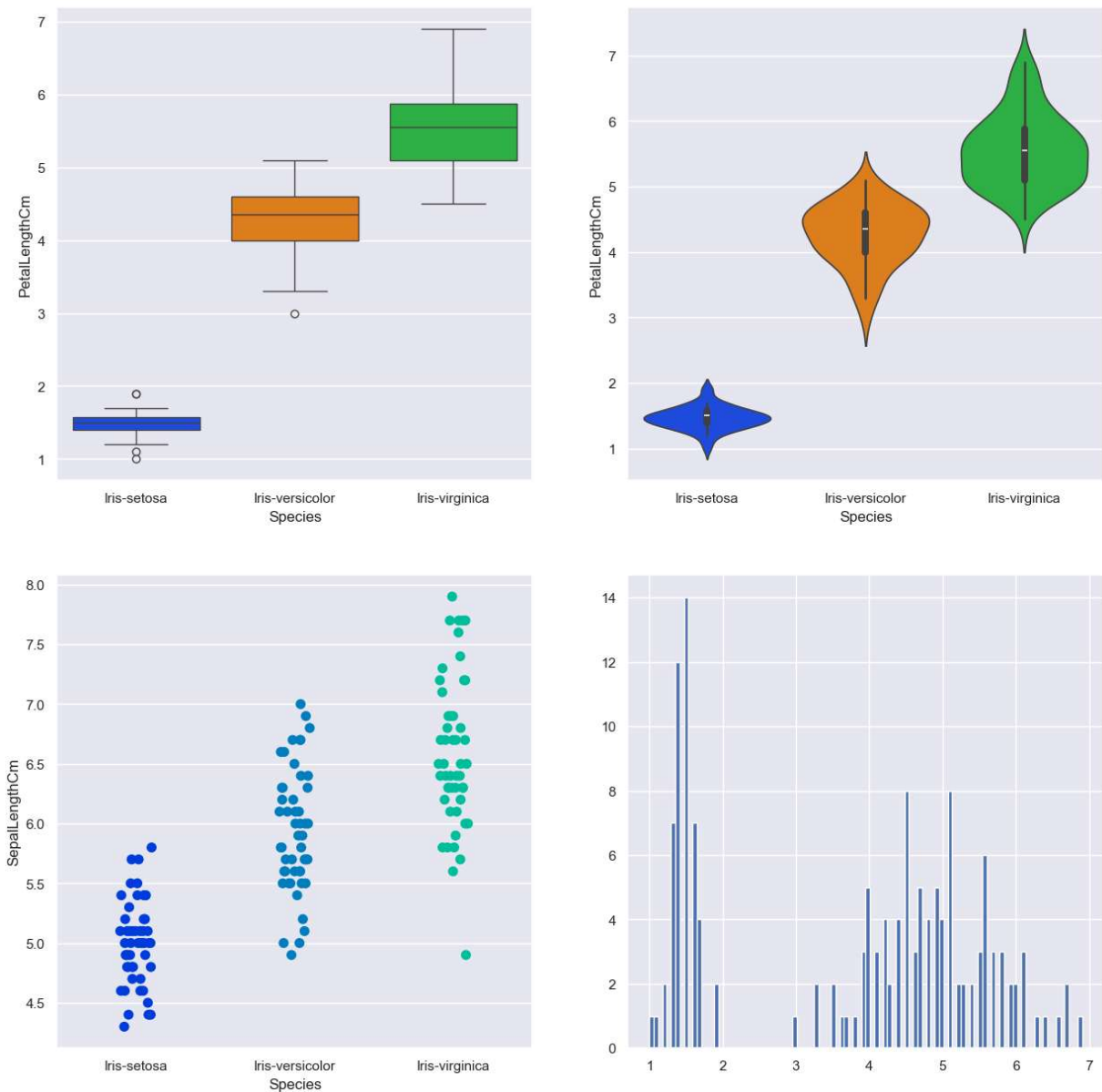


```
In [34]: # Create a kde plot of sepal_length versus sepal width for setosa species of flower
sub=iris[iris['Species']=='Iris-setosa']
sns.kdeplot(data=sub[['SepalLengthCm', 'SepalWidthCm']],palette='bright',shade=True,
plt.title('Iris-setosa')
plt.xlabel('Sepal Length Cm')
plt.ylabel('Sepal Width Cm')
plt.show()
```



```
In [35]: sns.set_style('darkgrid')
f, axes = plt.subplots(2, 2, figsize=(15, 15))

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



```
In [36]: iris['Species'] = iris['Species'].astype('category')
#iris.head()
```

```
In [91]: iris.dtypes
```

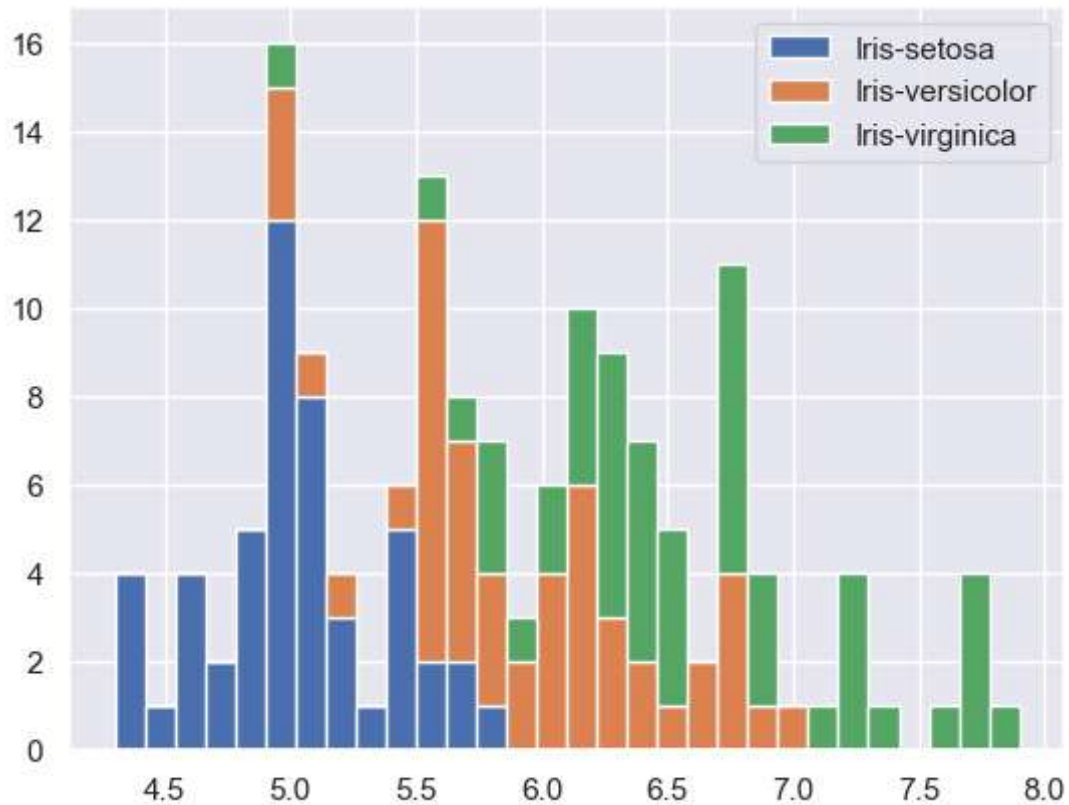
```
Out[91]: Sepal.LengthCm    float64
Sepal.WidthCm           float64
Petal.LengthCm          float64
Petal.WidthCm           float64
Species                 category
dtype: object
```

```
In [37]: list1=list()
mylabels=list()
for gen in iris.Species.cat.categories:
    list1.append(iris[iris.Species==gen].Sepal.LengthCm)
    mylabels.append(gen)

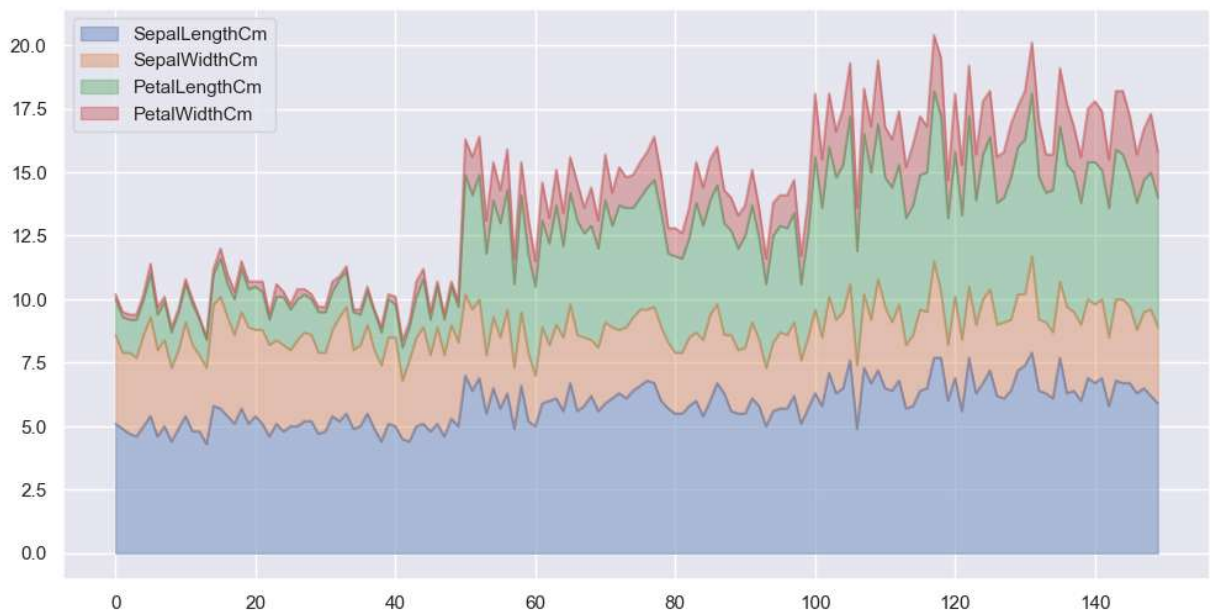
h=plt.hist(list1,bins=30,stacked=True,rwidth=1,label=mylabels)
```



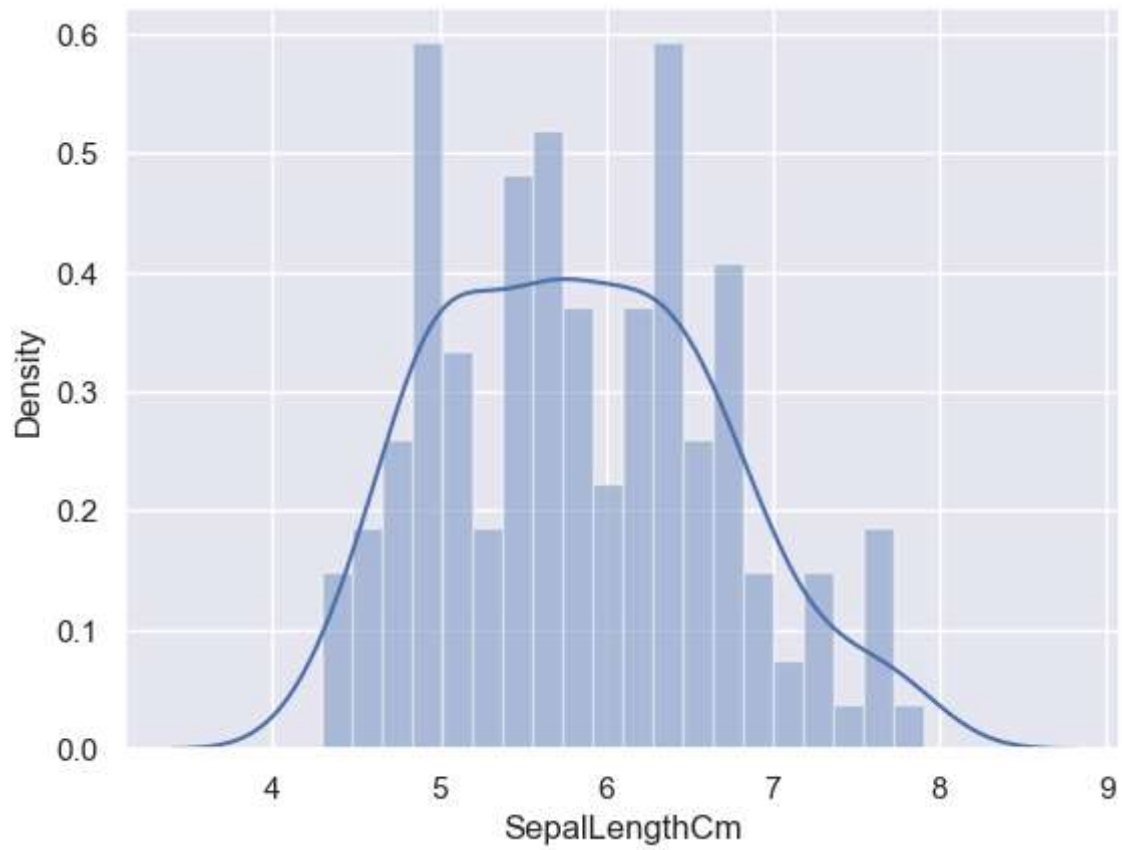
```
plt.legend()
plt.show()
```



```
In [38]: #iris['SepalLengthCm'] = iris['SepalLengthCm'].astype('category')
#iris.head()
#iris.plot.area(y='SepalLengthCm',alpha=0.4,figsize=(12, 6));
iris.plot.area(y=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'],alpha=0.4,figsize=(12, 6))
plt.show()
```



```
In [39]: sns.distplot(iris['SepalLengthCm'],kde=True,bins=20);
plt.show()
```



In []:

In []: