IRIS DATASET Analysis and Visualization

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
%matplotlib inline

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

In [2]:
```
iris=pd.read_csv(r'C:\Users\Affan\OneDrive\Desktop\FSDS Course NIT\Prakash Sir S
iris
```

Out[2]:

	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 [3]:
```
iris.head()
```

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

In [4]:
```
iris.tail()
```

Out[4]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
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

In [5]:
```
iris.shape
```

Out[5]:
```
(150, 6)
```

In [6]:
```
iris.columns
```

Out[6]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

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

Out[7]:

	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 [8]:
```
iris.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
```

Comment

-- from the above info we get to know that there are 6 attributes from which 1 object dtype, 1 int dtype and rest 4 are float dtype

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

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

the Species attribute has three values

In [10]:
```
iris['Species'].unique()
```

Out[10]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

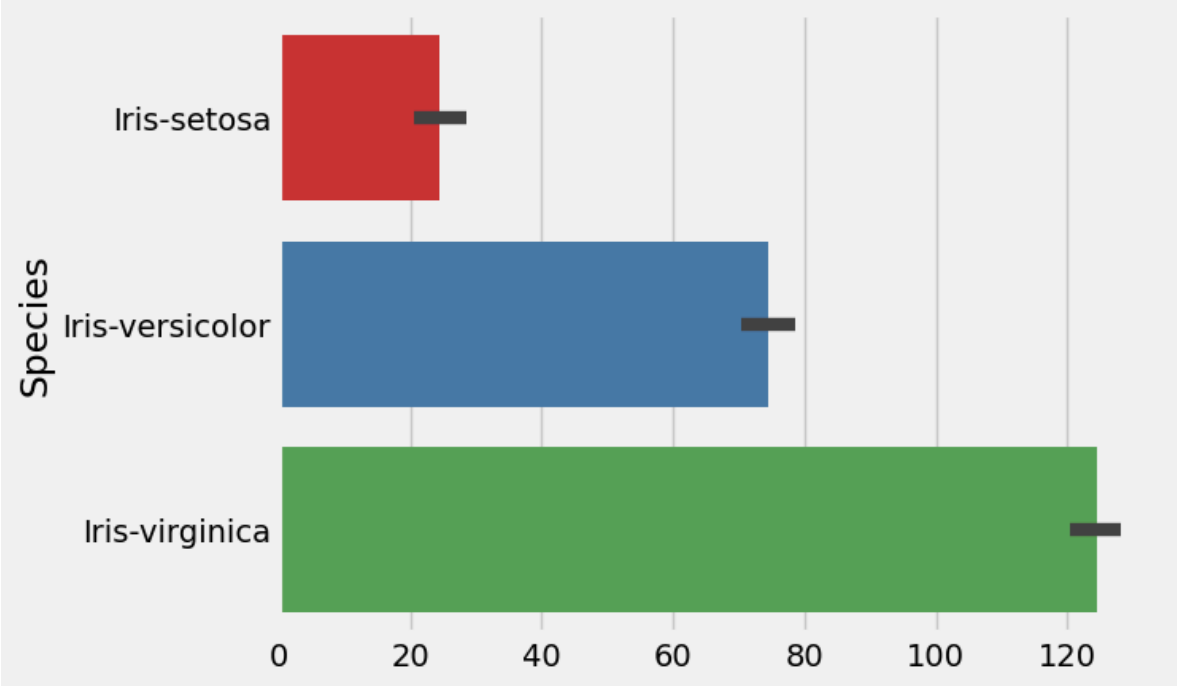
In [11]:
```
iris['Species'].nunique()
```

Out[11]:
```
3
```

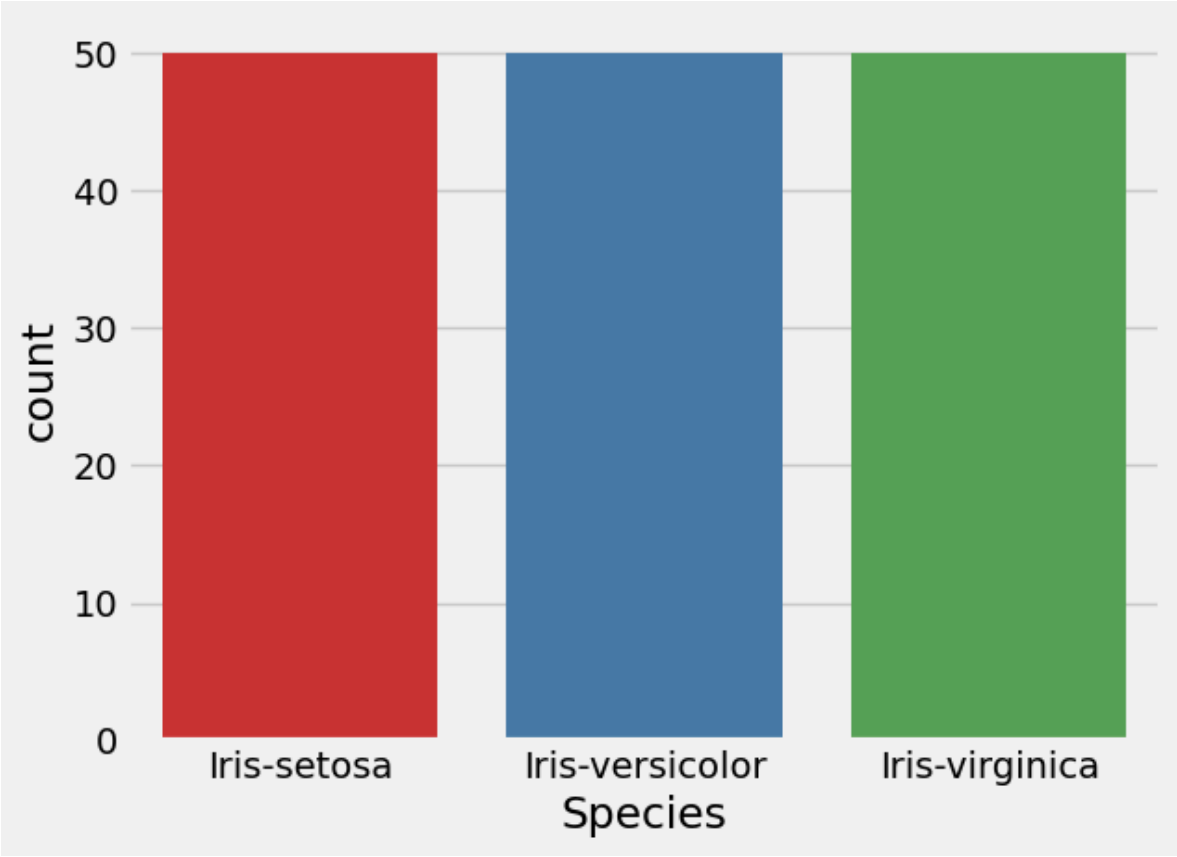
Now it's time for Visualization with Matplotlib/Seaborn

The following is the visualization of Dataset with different types of Matplotlib/Seaborn Graphs---

In [12]:
```
sns.barplot(iris['Species'],palette='Set1')
plt.show()
```

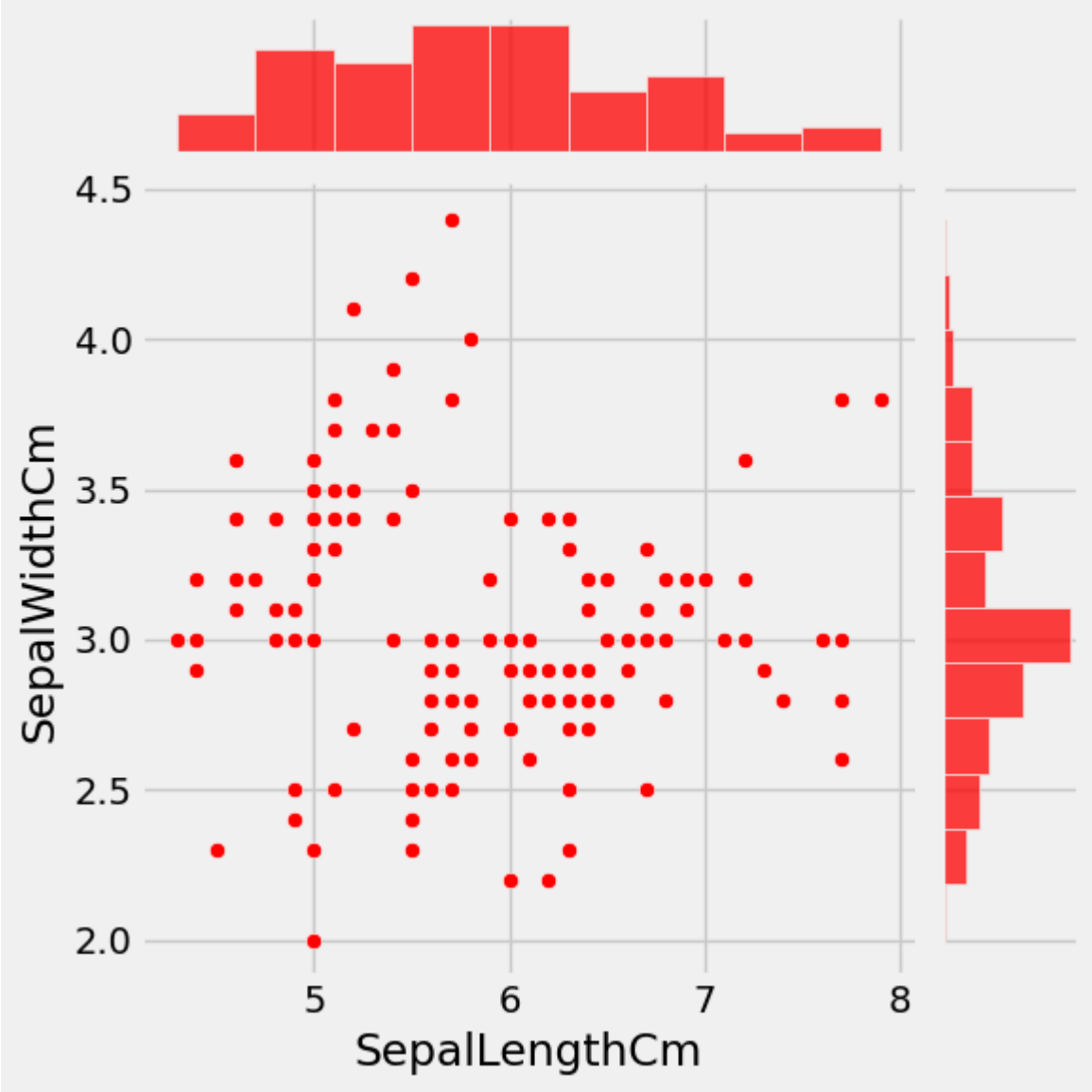


In [13]:
```
sns.countplot(data=iris,x='Species',palette='Set1')
plt.show()
```

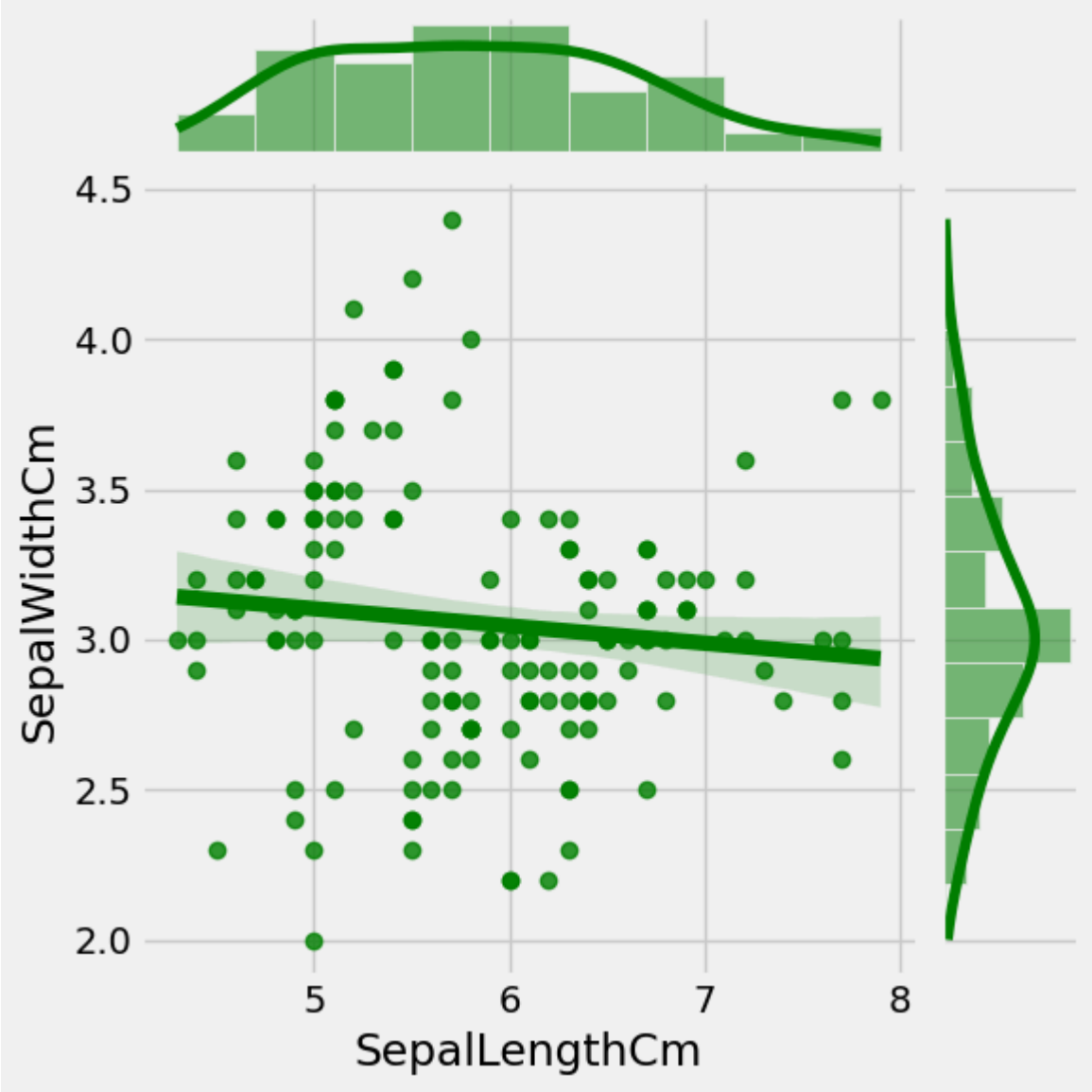


from above graph we can say that the all the samples of each specie of the 3 is 50

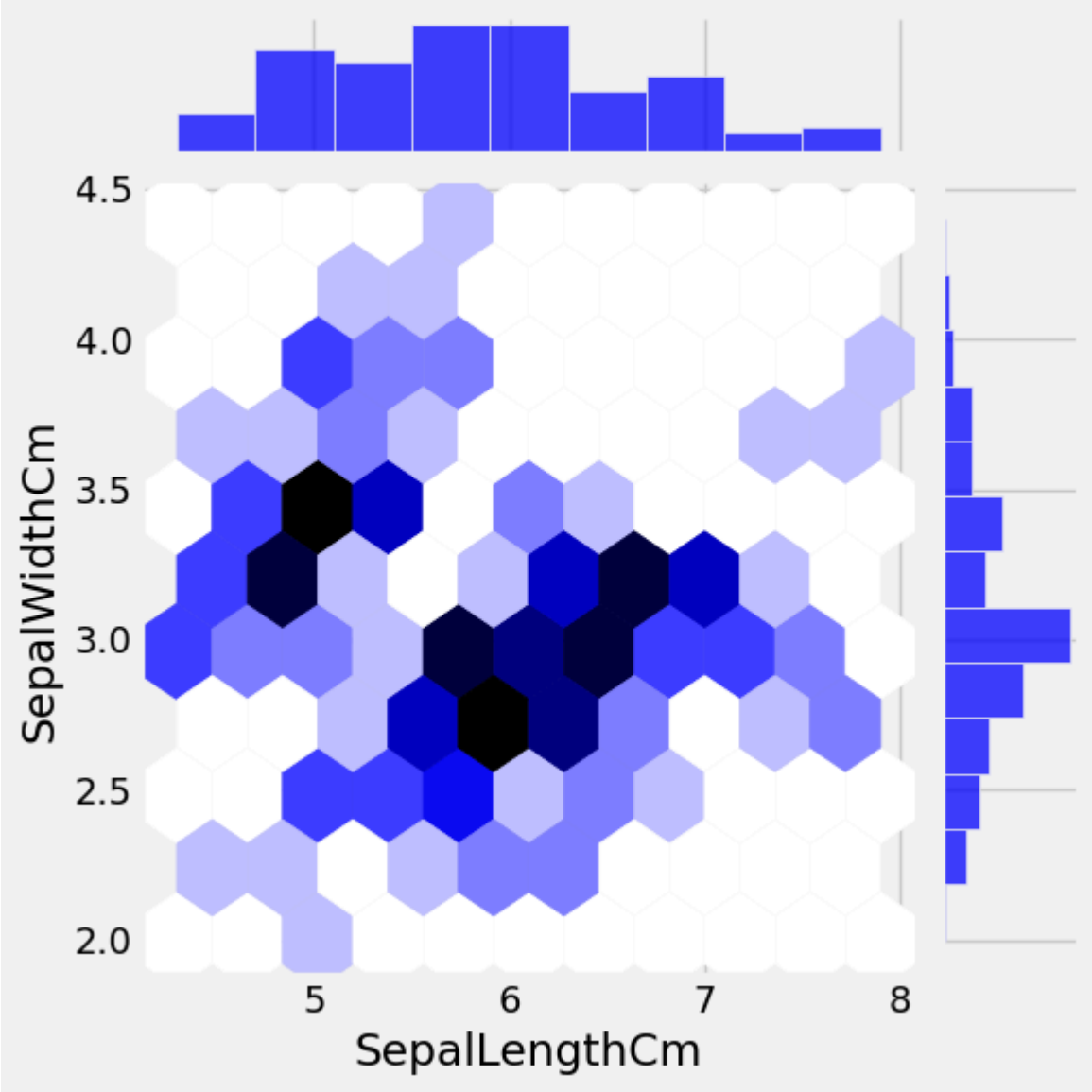
In [14]:
```
sns.jointplot(x=iris['SepalLengthCm'],y=iris['SepalWidthCm'],color='r')
plt.show()
```



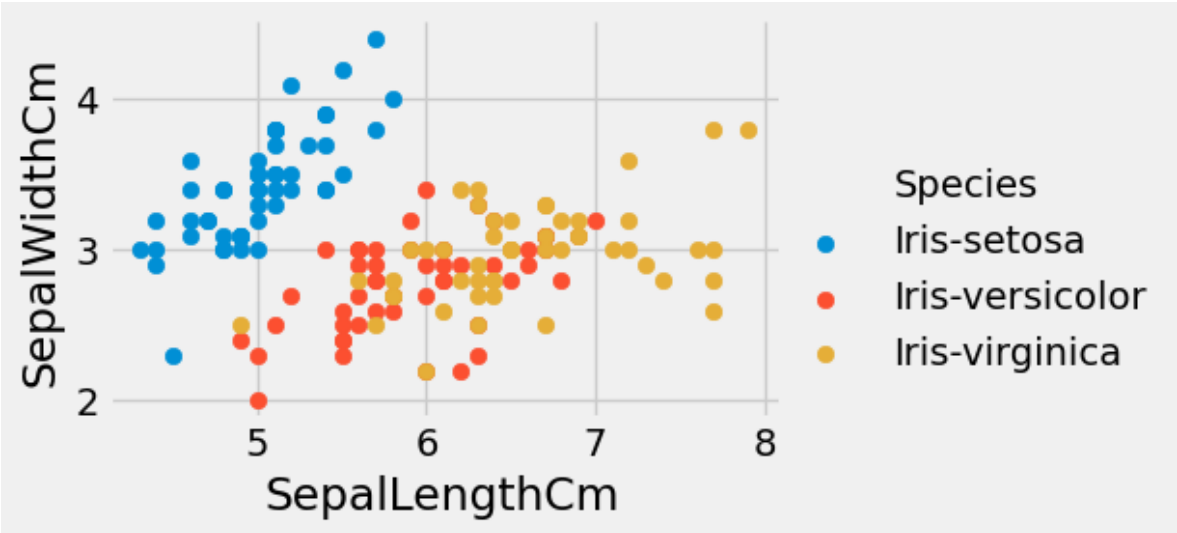
In [15]:
```
sns.jointplot(x=iris['SepalLengthCm'],y=iris['SepalWidthCm'],color='g',kind='reg
plt.show()
```



In [16]:
```
sns.jointplot(x=iris['SepalLengthCm'],y=iris['SepalWidthCm'],color='b',kind='hex
plt.show()
```

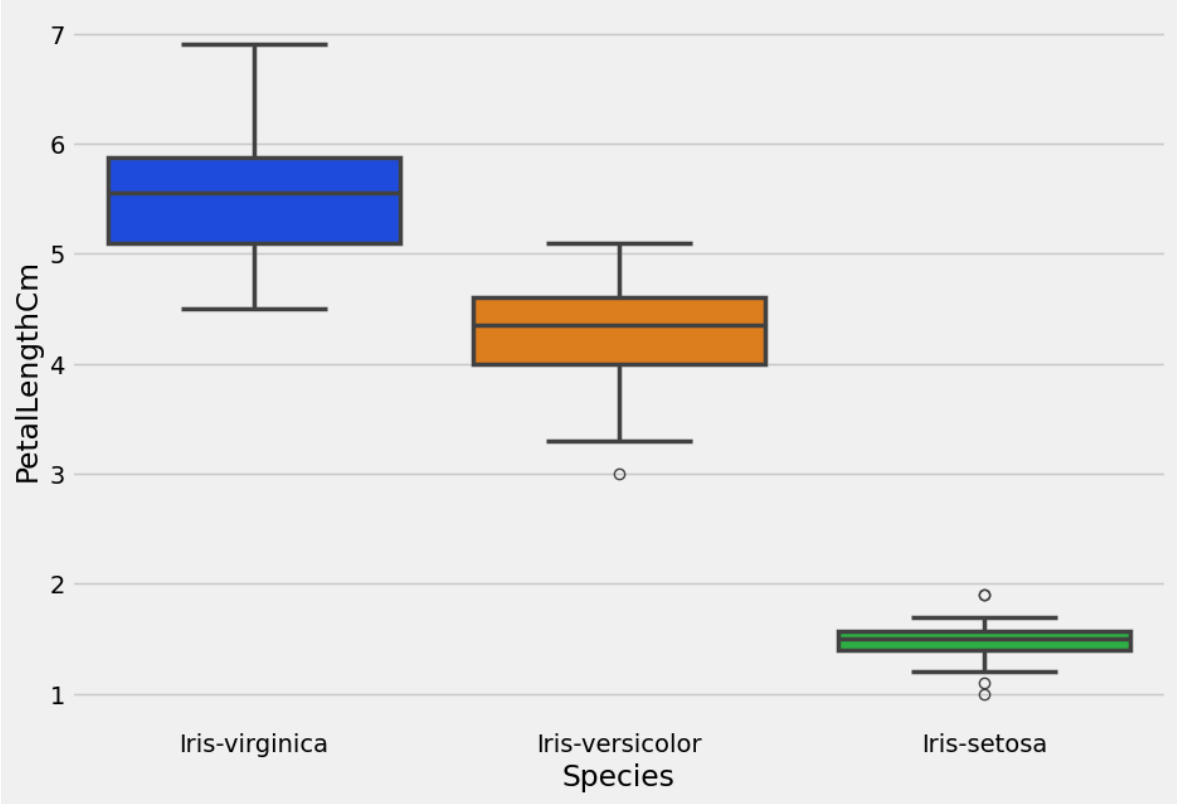


In [17]:
```
sns.FacetGrid(iris,hue='Species',aspect=1.5)\
.map(plt.scatter,'SepalLengthCm','SepalWidthCm')\
.add_legend()
plt.show()
```

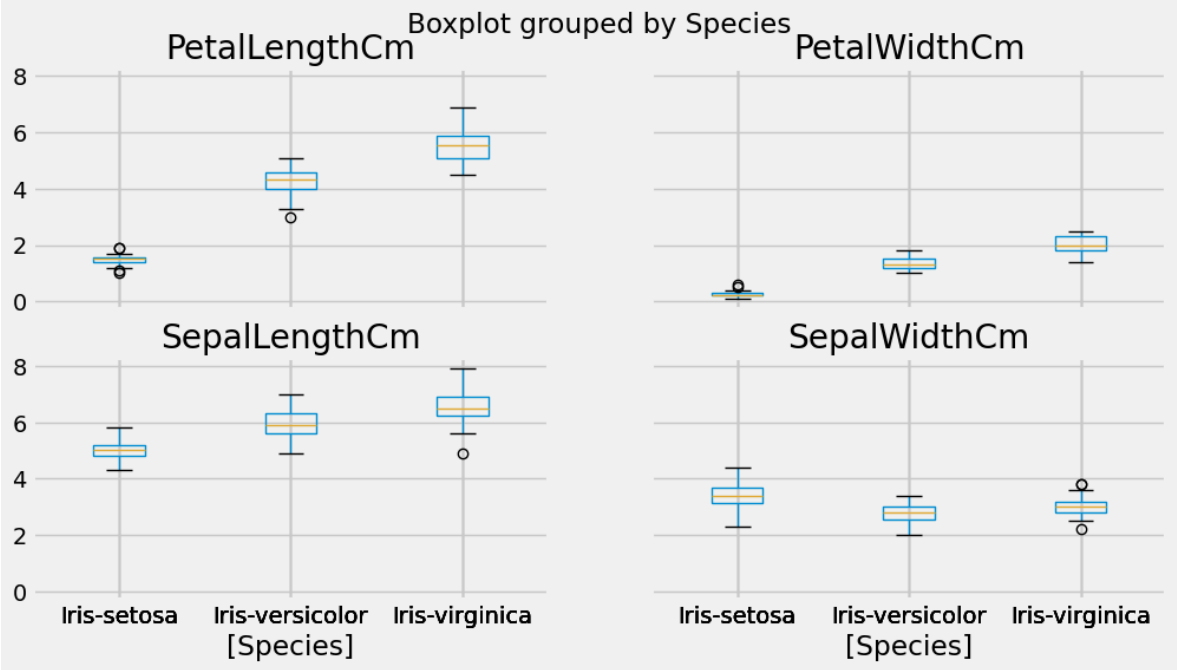


In [18]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
```

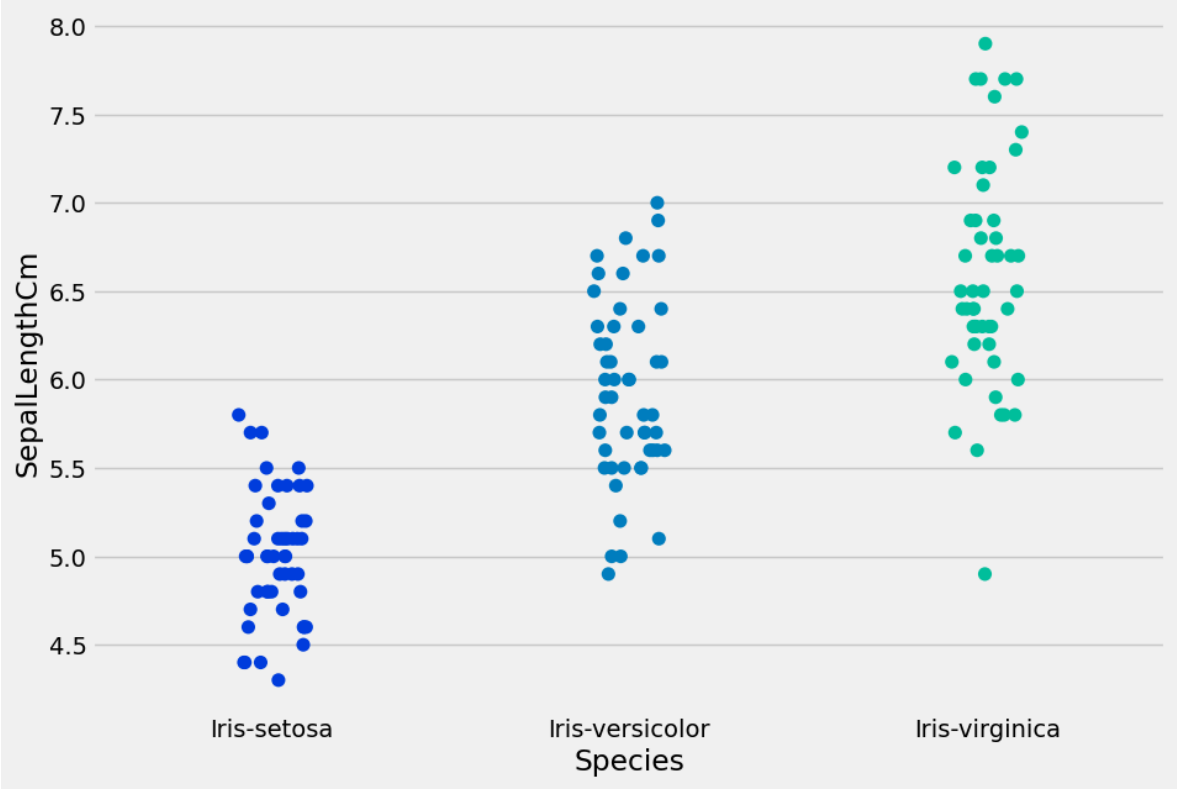
```
fig=sns.boxplot(x='Species',y='PetalLengthCm',data=iris,order=['Iris-virginica',
plt.show()
```



In [19]:
```
iris.drop("Id", axis=1).boxplot(by="Species", figsize=(12, 6))
plt.show()
```

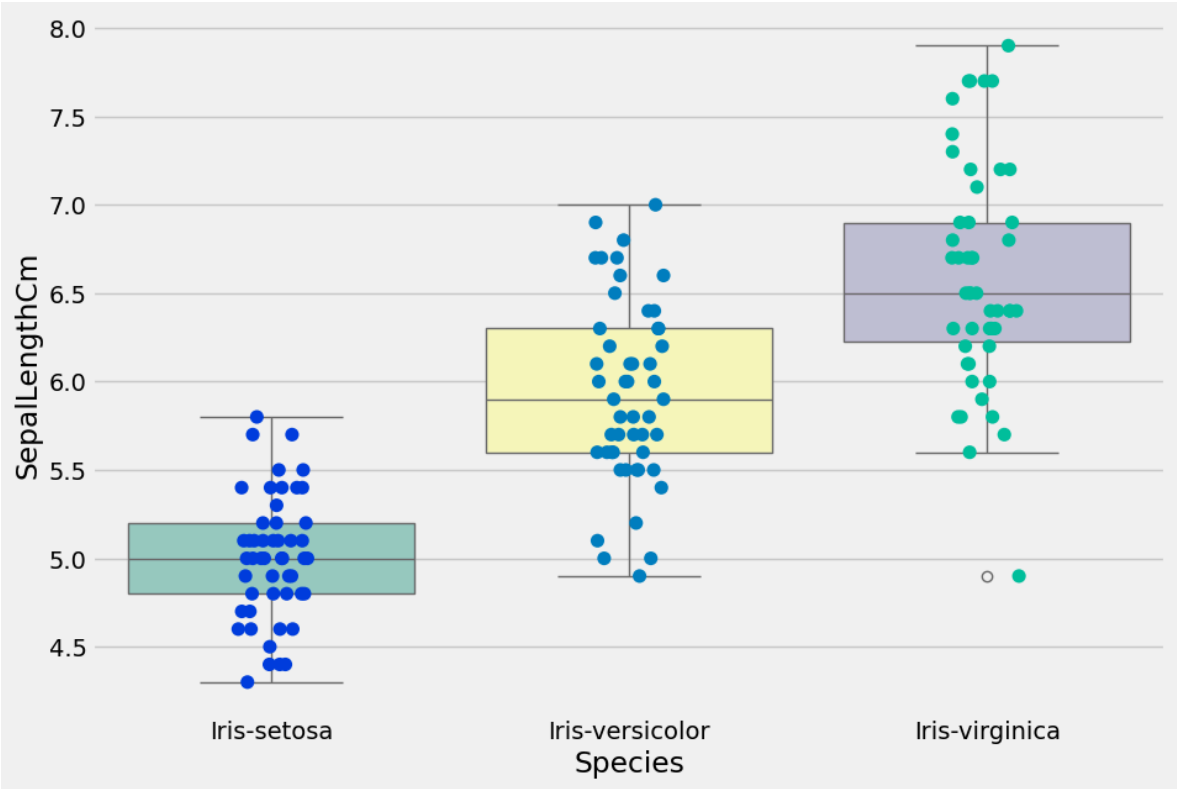


In [20]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edgecolor=

plt.show()
```



combing box and strip plot

In [21]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edgecolor=
fig=sns.boxplot(x='Species',y='SepalLengthCm',data=iris,palette='Set3')

plt.show()
```



In [32]:
```
iris.columns
```

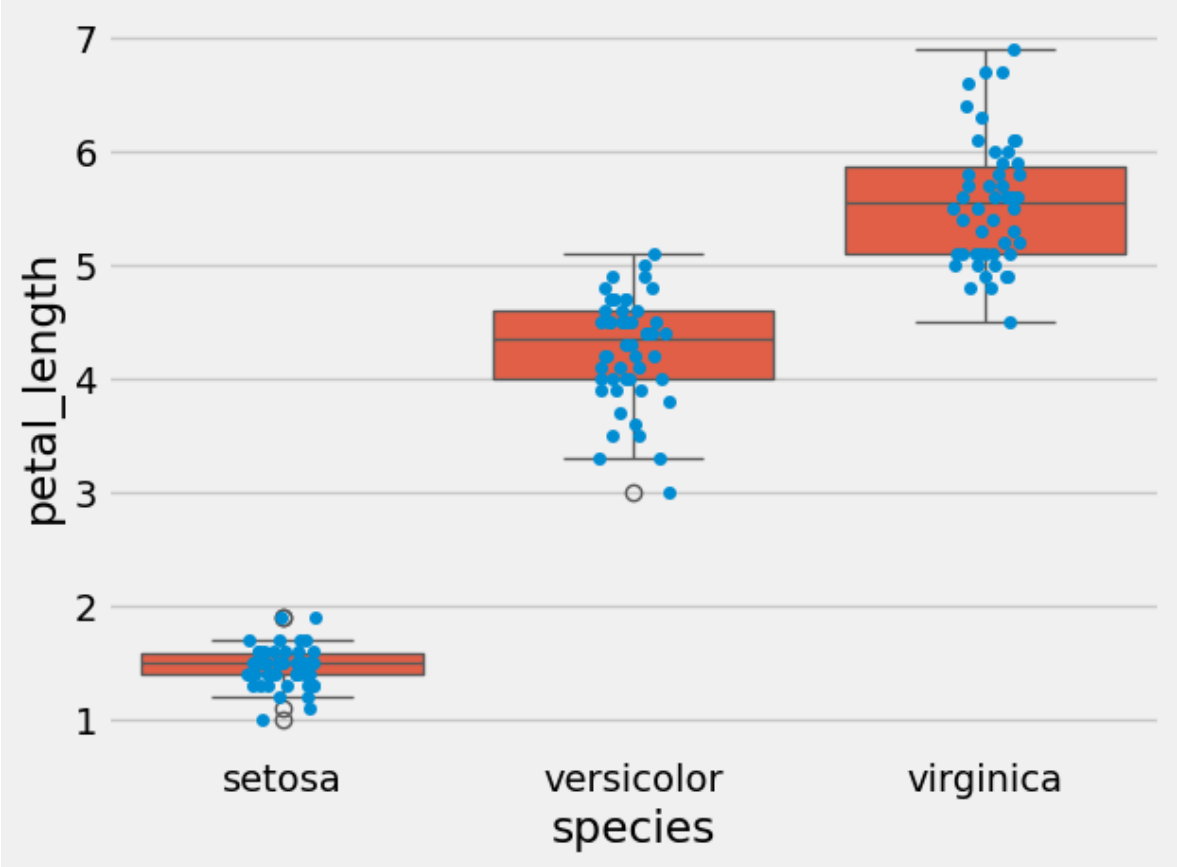
Out[32]:
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [35]:
```
iris = sns.load_dataset("iris")

ax=sns.boxplot(x='species',y='petal_length',data=iris)
ax=sns.stripplot(x='species',y='petal_length',data=iris,jitter=True,edgecolor='g

boxtwo = ax.artists[2]
boxtwo.set_facecolor('yellow')
boxtwo.set_edgecolor('black')
boxthree=ax.artists[1]
boxthree.set_facecolor('red')
boxthree.set_edgecolor('black')
boxthree=ax.artists[0]
boxthree.set_facecolor('green')
boxthree.set_edgecolor('black')

plt.show()
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[35], line 6
      3 ax=sns.boxplot(x='species',y='petal_length',data=iris)
      4 ax=sns.stripplot(x='species',y='petal_length',data=iris,jitter=True,edgec
olor='gray')
----> 6 boxtwo = ax.artists[2]
      7 boxtwo.set_facecolor('yellow')
      8 boxtwo.set_edgecolor('black')

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:1453, in _AxesBase.Ar
tistList.__getitem__(self, key)
   1452 def __getitem__(self, key):
-> 1453     return [artist
   1454             for artist in self._axes._children
   1455             if self._type_check(artist)][key]

IndexError: list index out of range
```

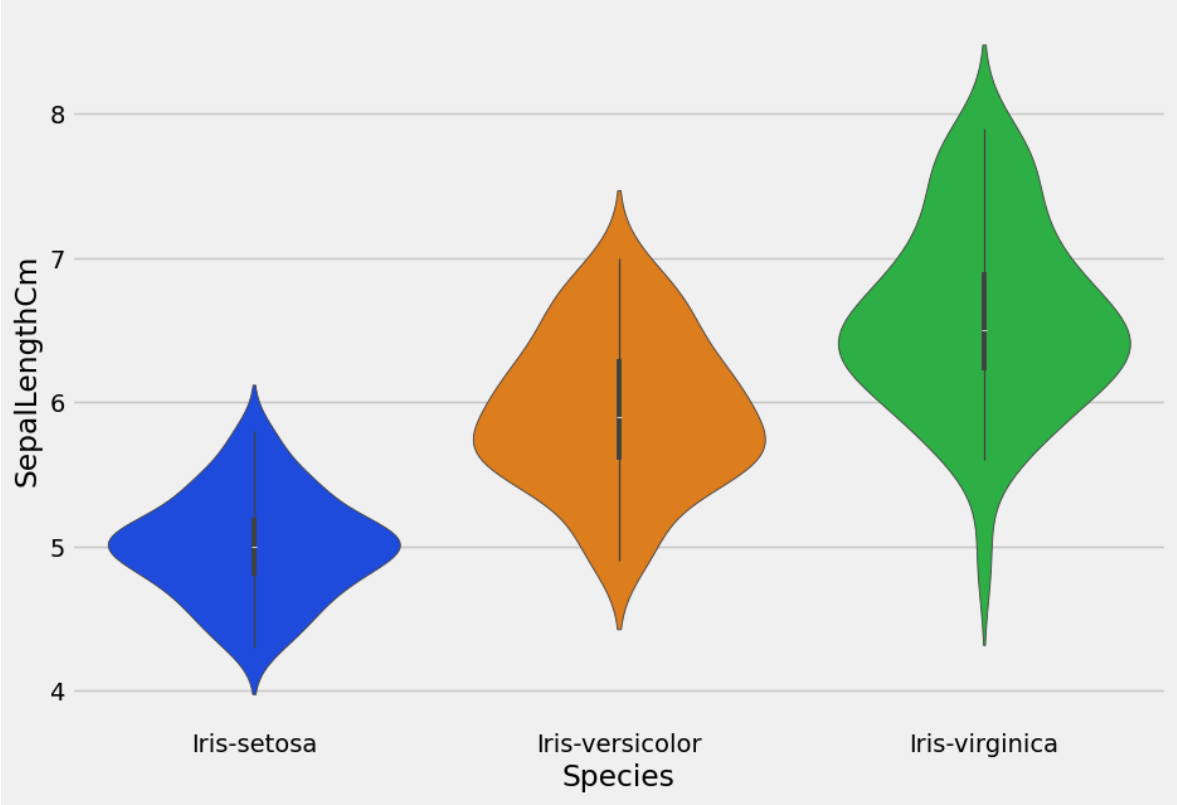
In [37]:
```
iris = sns.load_dataset("iris")

# Plot
fig, ax = plt.subplots()
sns.stripplot(x='species', y='petal_length', data=iris, jitter=True, edgecolor='
sns.boxplot(x='species', y='petal_length', data=iris, ax=ax)

# Get all rectangles (box elements)
boxes = [patch for patch in ax.patches if isinstance(patch, plt.Rectangle)]

# Print how many boxes found
print(f"Found {len(boxes)} boxes")

# Example: color the second box if it exists
if len(boxes) > 1:
    boxes[1].set_facecolor('yellow')
    boxes[1].set_edgecolor('black')

plt.show()
```

```
Found 0 boxes
```



In [ ]:

In [44]:
```
iris=pd.read_csv(r'C:\Users\Affan\OneDrive\Desktop\FSDS Course NIT\Prakash Sir S
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.violinplot(x='Species',y='SepalLengthCm',data=iris,palette='bright')

plt.show()
```


8
7
6
5
4
SepalLengthCm
Iris-setosa
Iris-versicolor
Iris-virginica
Species

In [45]:
```
iris
```

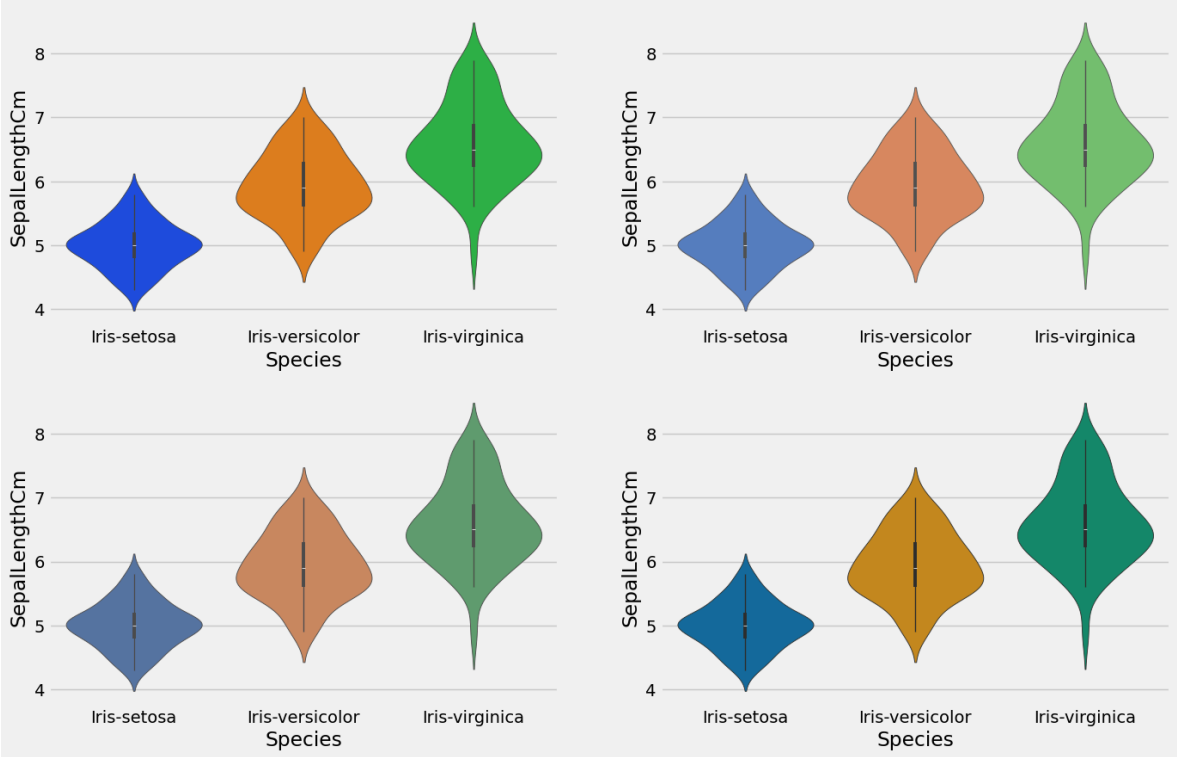
Out[45]:

	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

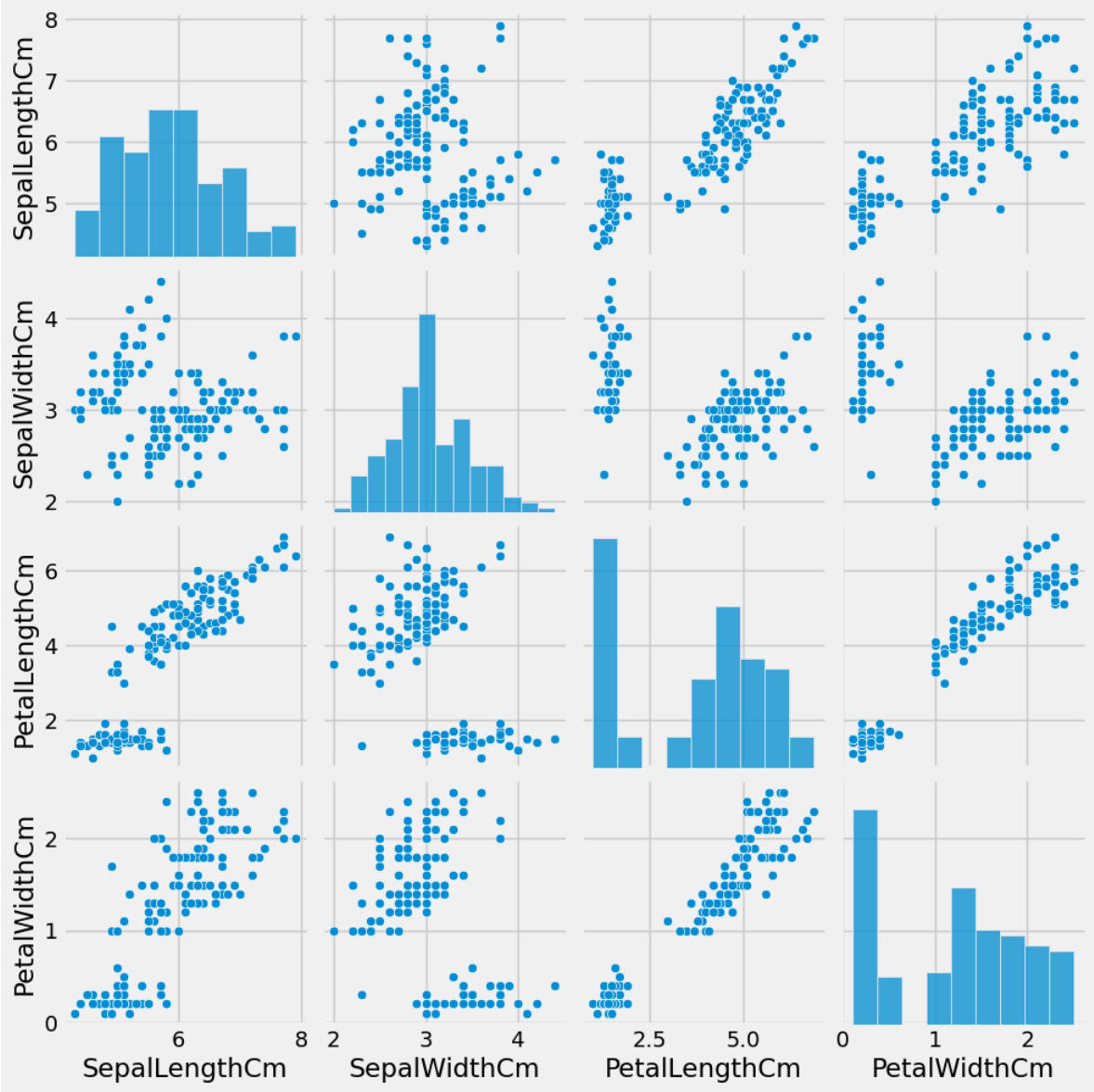
150 rows × 6 columns

In [49]:

```
plt.figure(figsize=(15,10))
plt.subplot(2,2,1)
sns.violinplot(x='Species',y='SepalLengthCm',data=iris,palette='bright')
plt.subplot(2,2,2)
sns.violinplot(x='Species',y='SepalLengthCm',data=iris,palette='muted')
plt.subplot(2,2,3)
sns.violinplot(x='Species',y='SepalLengthCm',data=iris,palette='deep')
plt.subplot(2,2,4)
sns.violinplot(x='Species',y='SepalLengthCm',data=iris,palette='colorblind')

plt.show()
```



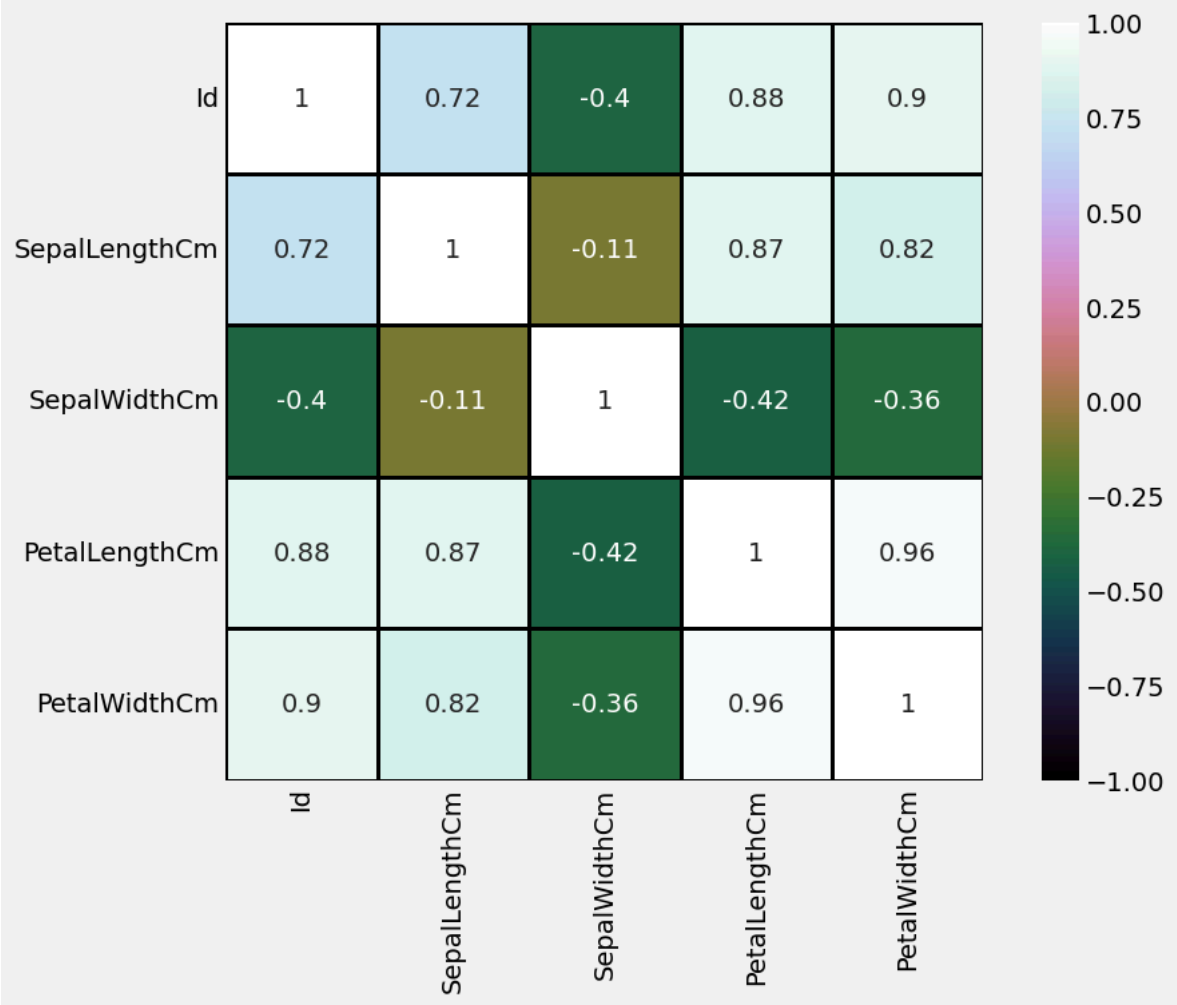
In [60]:
```
iris_cleaned = iris.drop("Id", axis=1)

# Now use seaborn's pairplot function
sns.pairplot(data=iris_cleaned, kind='scatter')

plt.show()
```



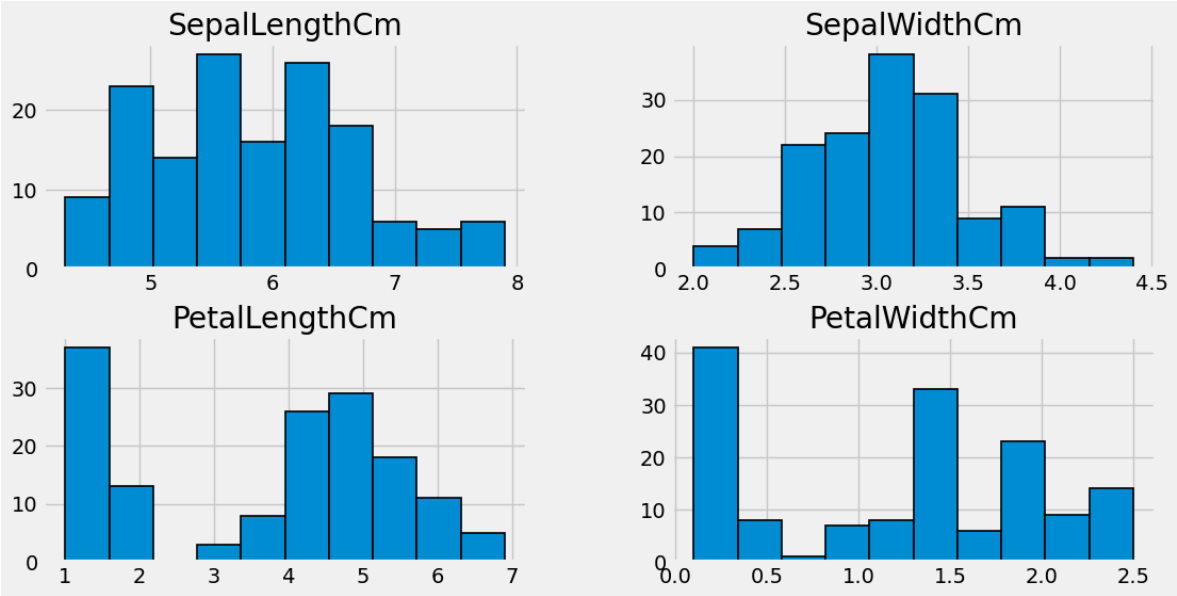
In [62]:
```
sns.pairplot(iris,hue='Species')

plt.show()
```

In [71]:

```
fig=plt.gcf()
fig.set_size_inches(10,7)
# Drop non-numeric columns like 'Species' (based on your dataset)
iris_numeric = iris.select_dtypes(include=['float64', 'int64'])

fig=sns.heatmap(iris_numeric.corr(),annot=True,cmap='cubehelix',linewidths=1,lin

plt.show()
```

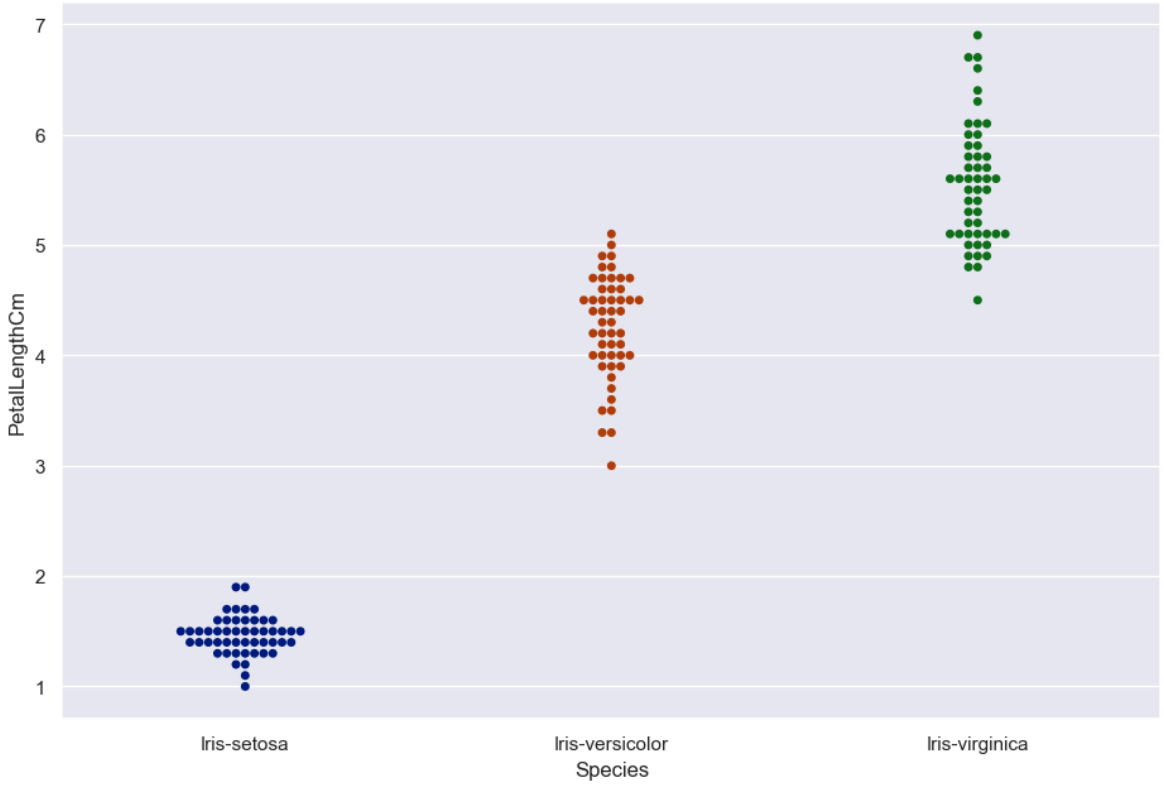


In [75]:
```
iris_cleaned.hist(edgecolor='black',linewidth=1.2)
fig=plt.gcf()
fig.set_size_inches(12,6)

plt.show()
```

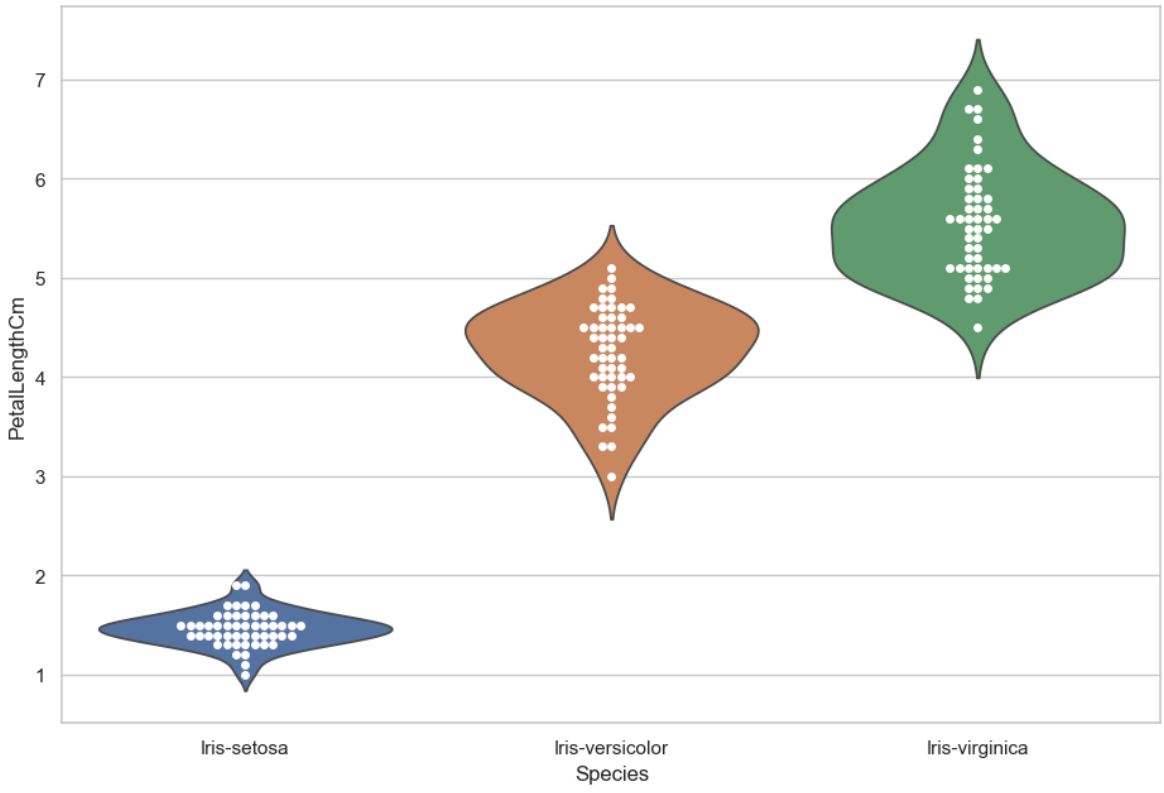


In [77]:
```
sns.set(style='darkgrid')
fig=plt.gcf()
fig.set_size_inches(10,7)
fig = sns.swarmplot(x='Species',y='PetalLengthCm',data=iris,palette='dark')
```

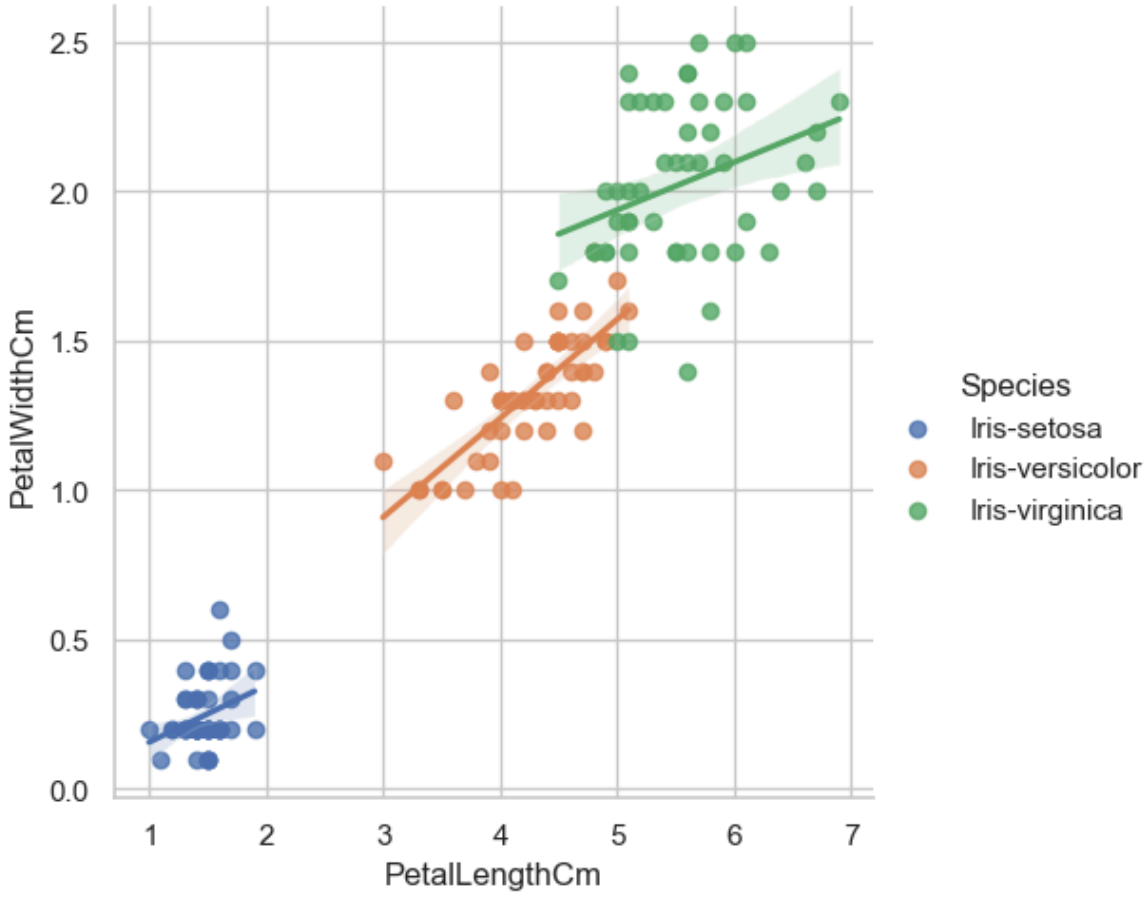
```
plt.show()
```



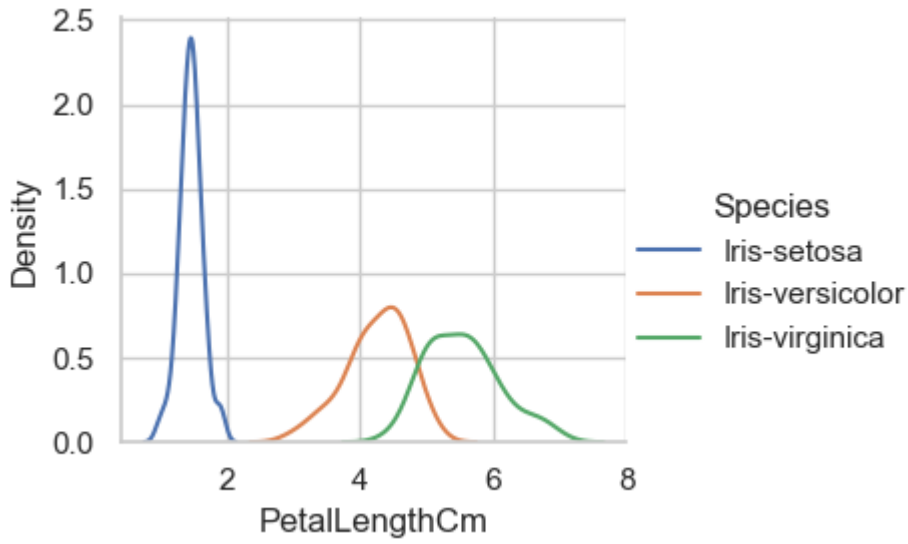
In [90]:
```
sns.set(style='whitegrid')
fig=plt.gcf()
fig.set_size_inches(10,7)
ax = sns.swarmplot(x='Species',y='PetalLengthCm',data=iris,color='white',edgecol
ax = sns.violinplot(x='Species',y='PetalLengthCm', data=iris,inner=None,hue='Spe

plt.show()
```



In [89]:
```
fig=sns.lmplot(x='PetalLengthCm',y='PetalWidthCm', hue='Species',data=iris)

plt.show()
```

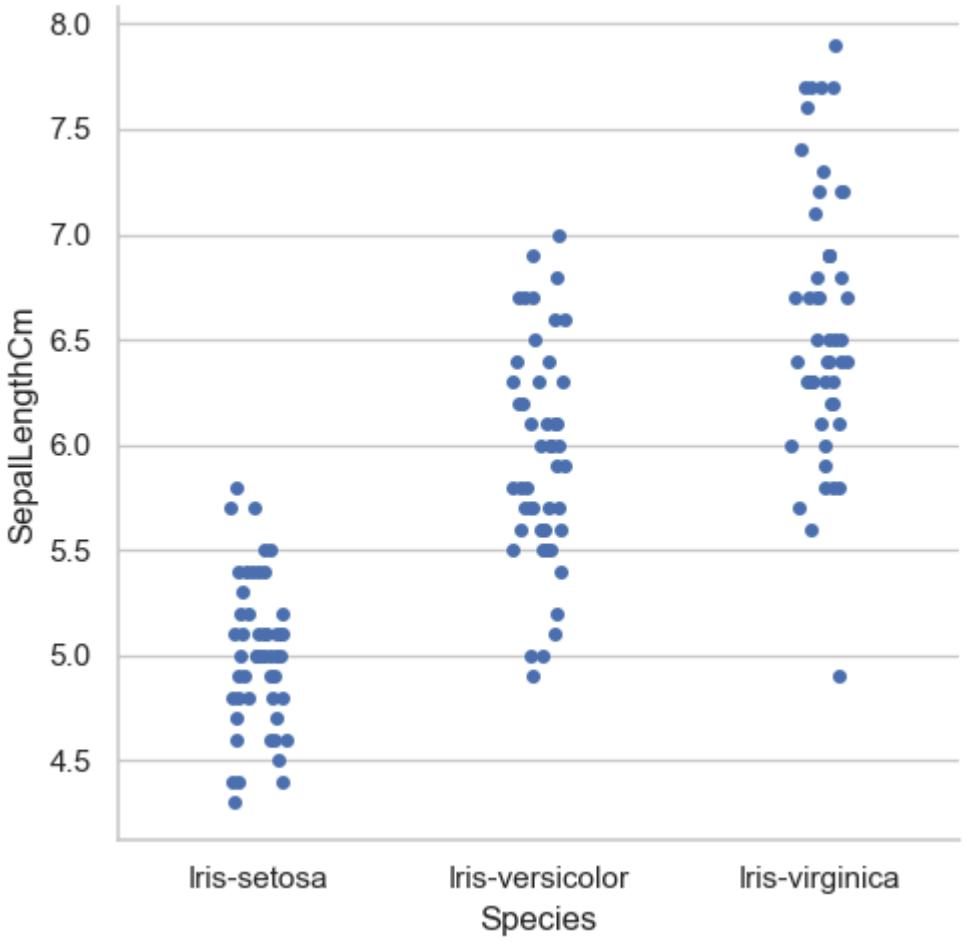


In [95]:
```
sns.FacetGrid(iris, hue="Species",aspect=1.2) \
   .map(sns.kdeplot, "PetalLengthCm") \
   .add_legend()
plt.ioff()

plt.show()
```



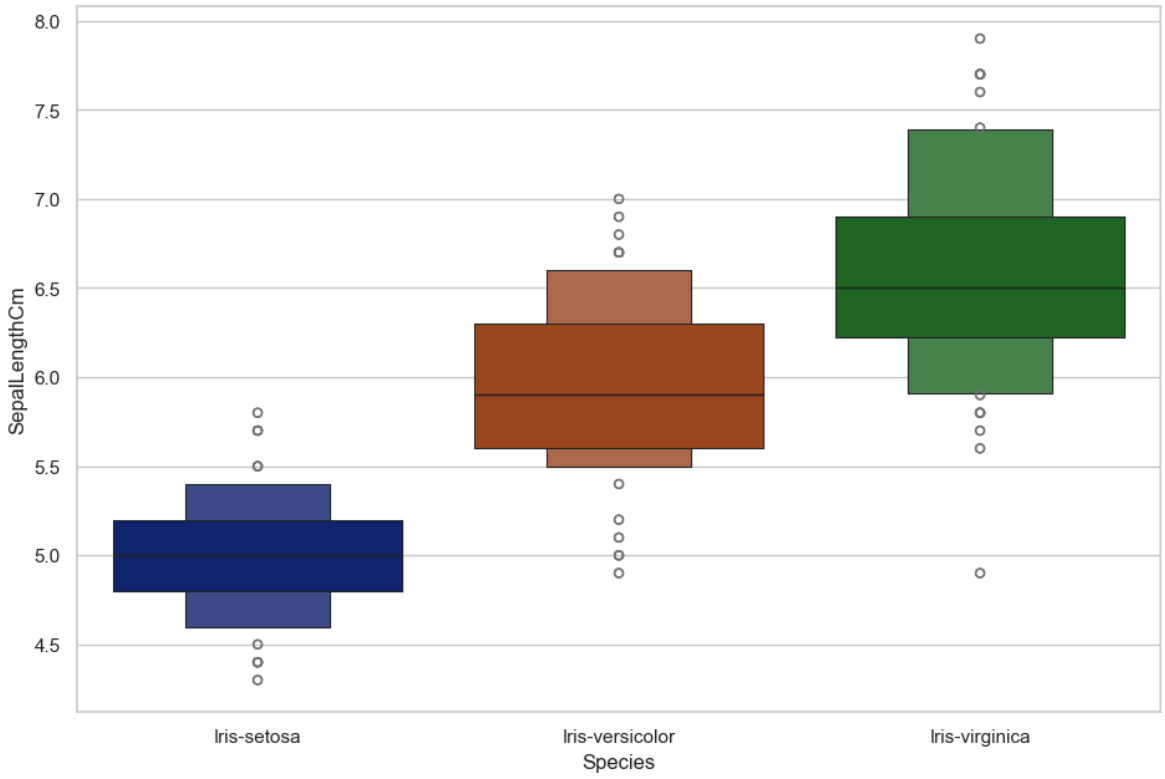
In [110…
```
sns.catplot(x='Species',y='SepalLengthCm',data=iris)
plt.ioff()
```

```
plt.show()
```



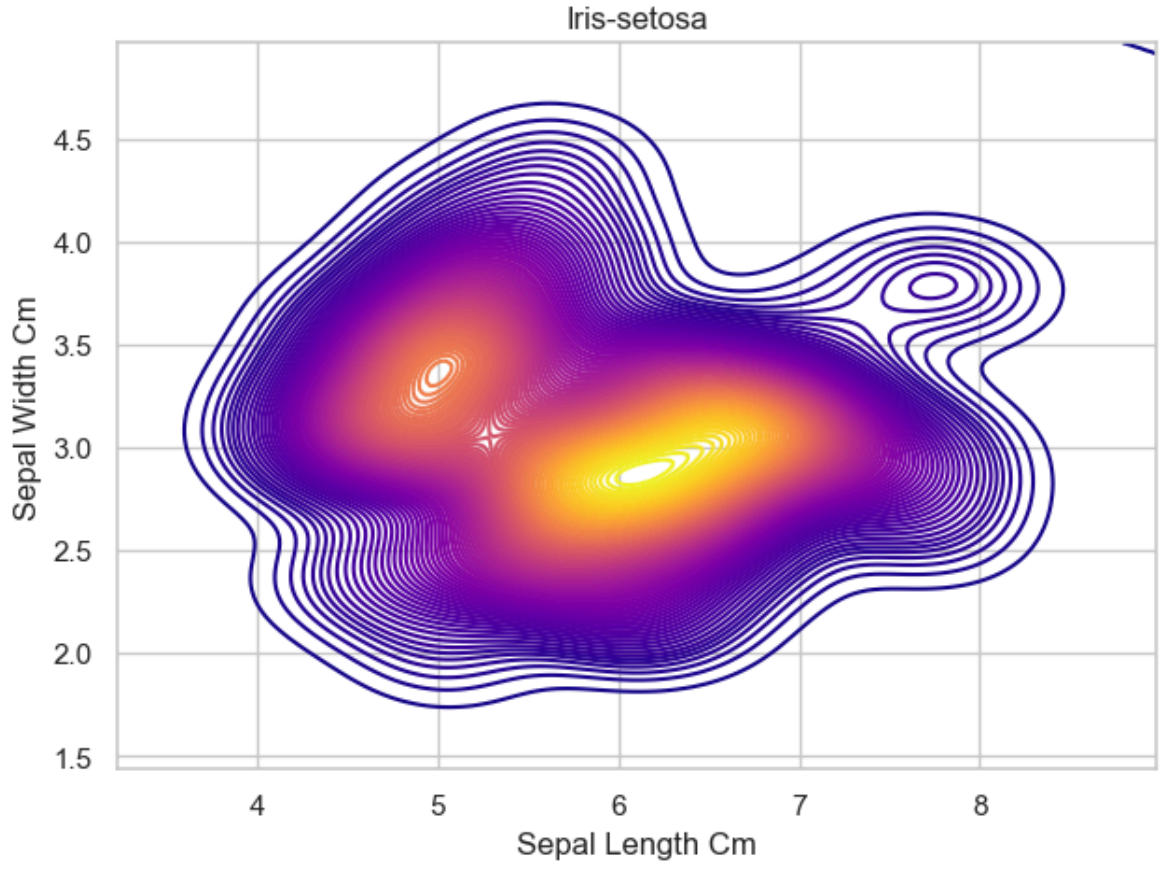
In [112…

```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig = sns.boxenplot(x='Species',y='SepalLengthCm',data=iris,palette='dark')

plt.show()
```



In [121…

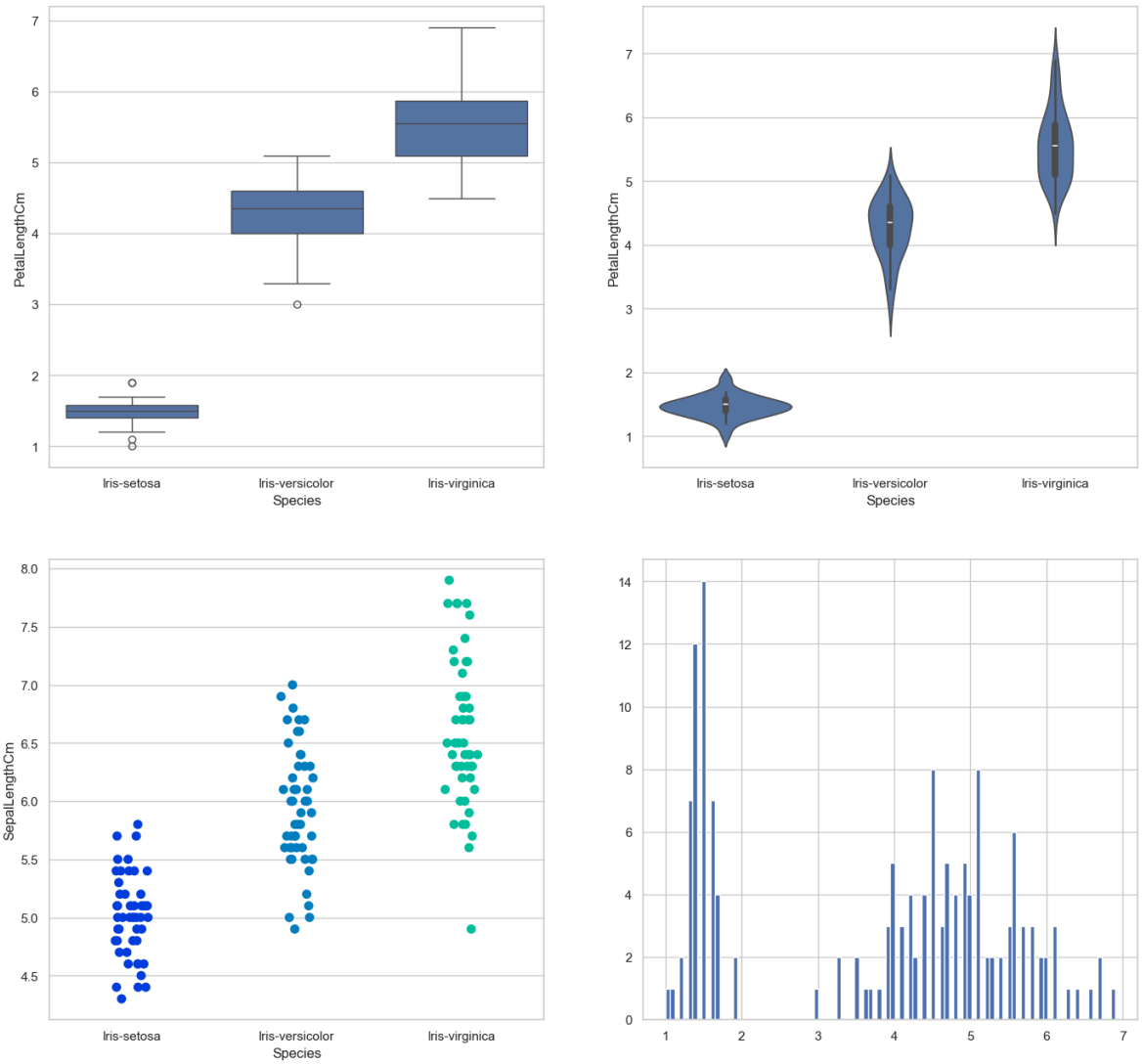
```
# Create a kde plot of sepal_length versus sepal width for setosa species of flo
sub=iris[iris['Species']=='Iris-setosa']
sns.kdeplot(data=iris,x='SepalLengthCm',y='SepalWidthCm',cmap="plasma", thresh=0
plt.title('Iris-setosa')
plt.xlabel('Sepal Length Cm')
plt.ylabel('Sepal Width Cm')

plt.show()
```



DASHBOARD

In [123…

```
sns.set_style('whitegrid')
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='
#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 the dashboard we have shown how to create multiple plots to foam a dashboard using Python.In this plot we have demonstrated how to plot Seaborn and Matplotlib plots on the same Dashboard.

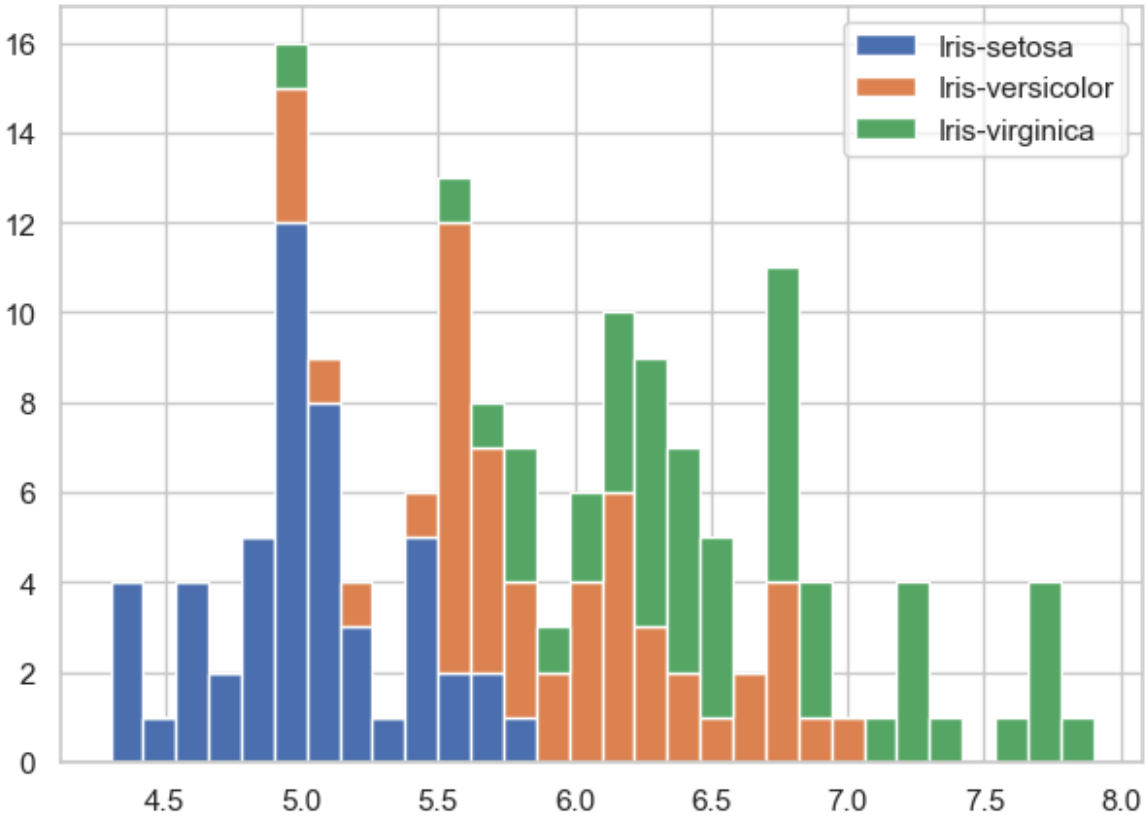
Stacked Hist

In [126…

```
iris['Species'] = iris['Species'].astype('category')
```

In [128…

```
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()
```

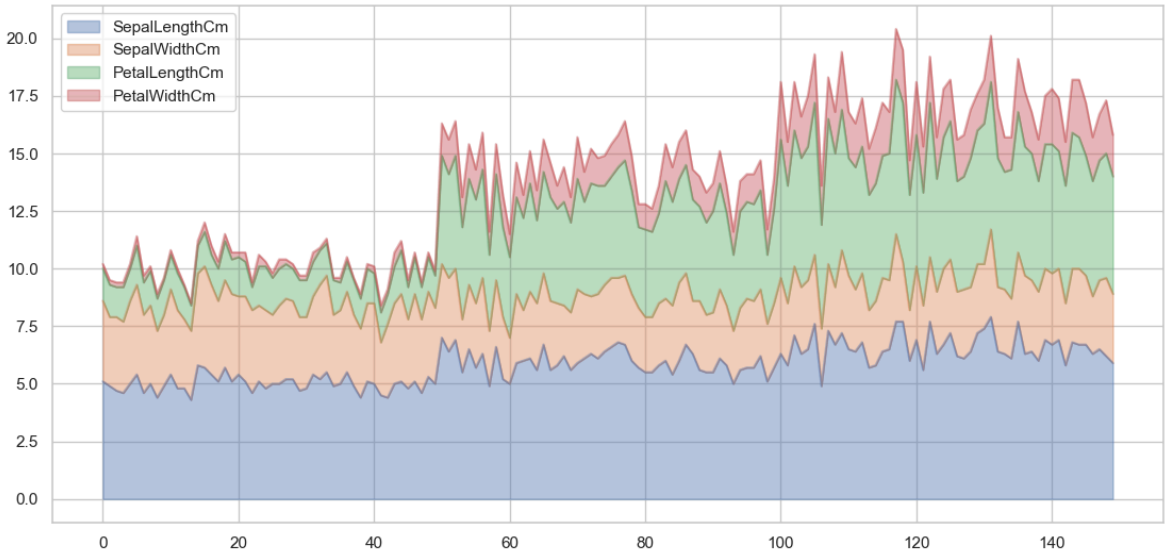


With Stacked Histogram we can see the distribution of Sepal Length of Different Species together.This shows us the range of Sepan Length for the three different Species of Iris Flower.

Area Plot: Area Plot gives us a visual representation of Various dimensions of Iris flower and their range in dataset.

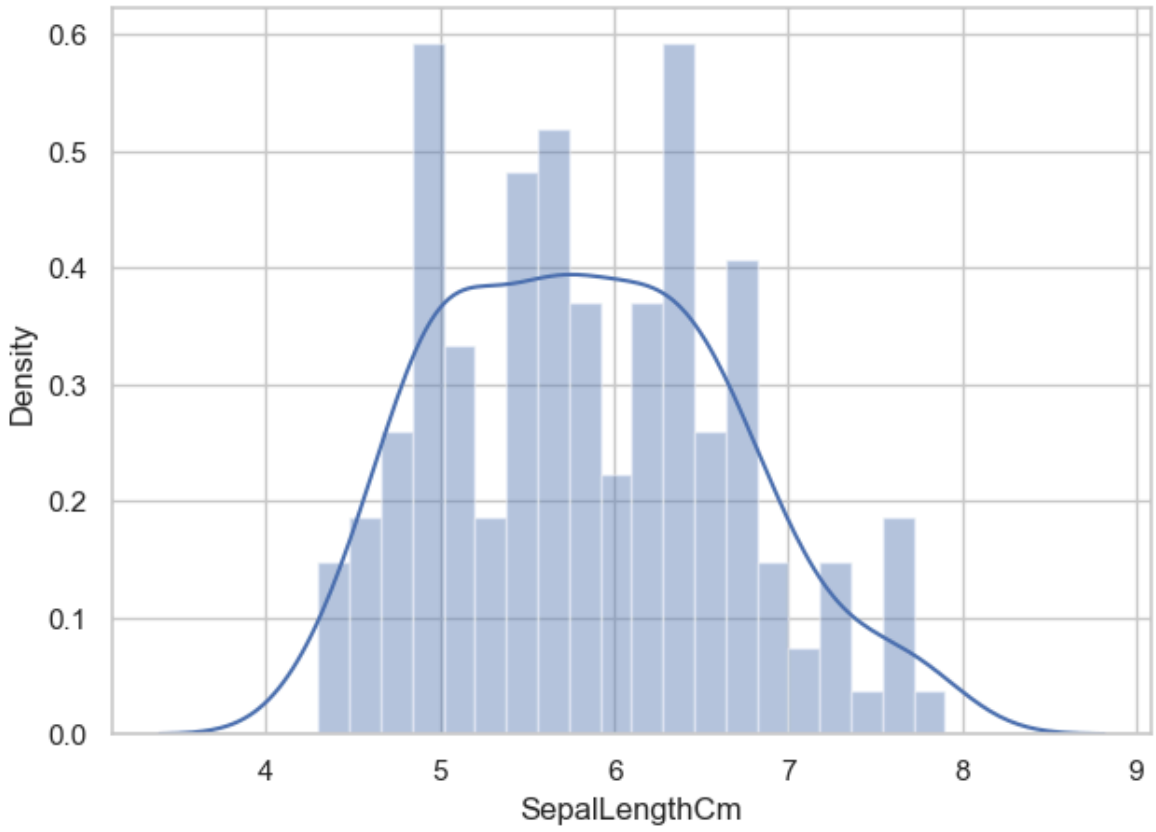
In [131…

```
iris.plot.area(y=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'])

plt.show()
```



In [133…

```
sns.distplot(iris['SepalLengthCm'],kde=True,bins=20)

plt.show()
```



we have finally completed working on IRIS EDA Project

In [ ]: