In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
iris=pd.read_csv(r'E:\AI\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.drop('Id',axis=1,inplace=True)
```

In [5]:

```
iris
```

Out[5]:

	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 [6]:

```
iris.isnull().sum()
```

Out[6]:

```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [7]:

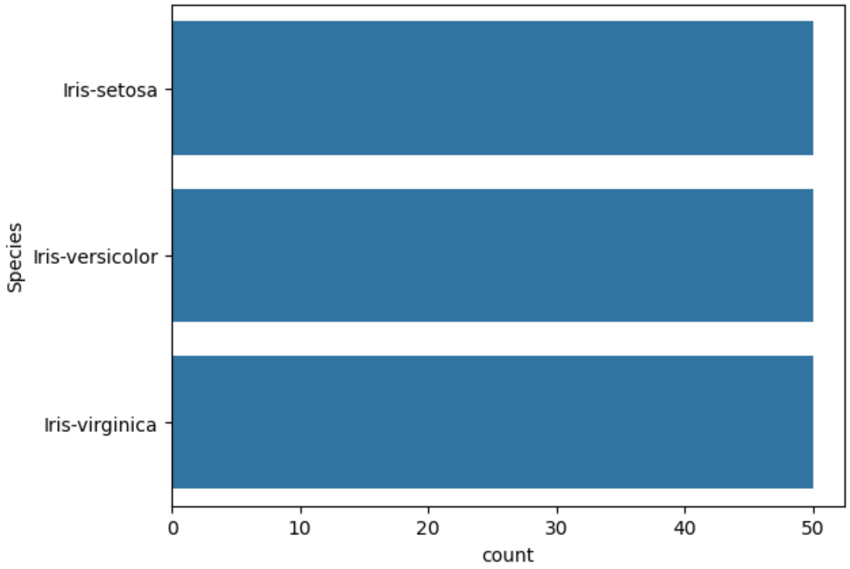
```
iris['Species'].value_counts()
```

Out[7]:

```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

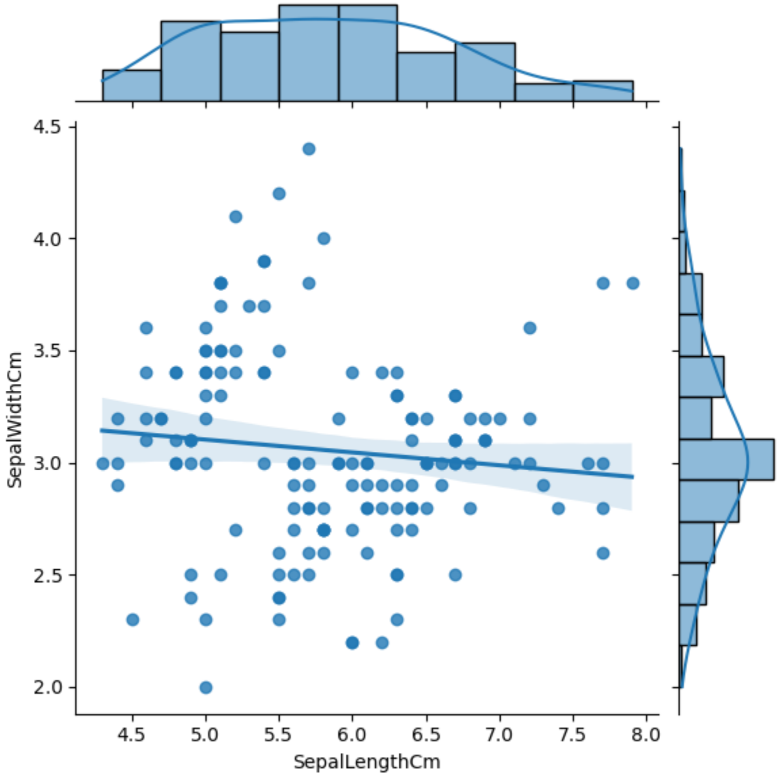
```
In [8]: sns.countplot(iris['Species'])
```

```
Out[8]: <Axes: xlabel='count', ylabel='Species'>
```



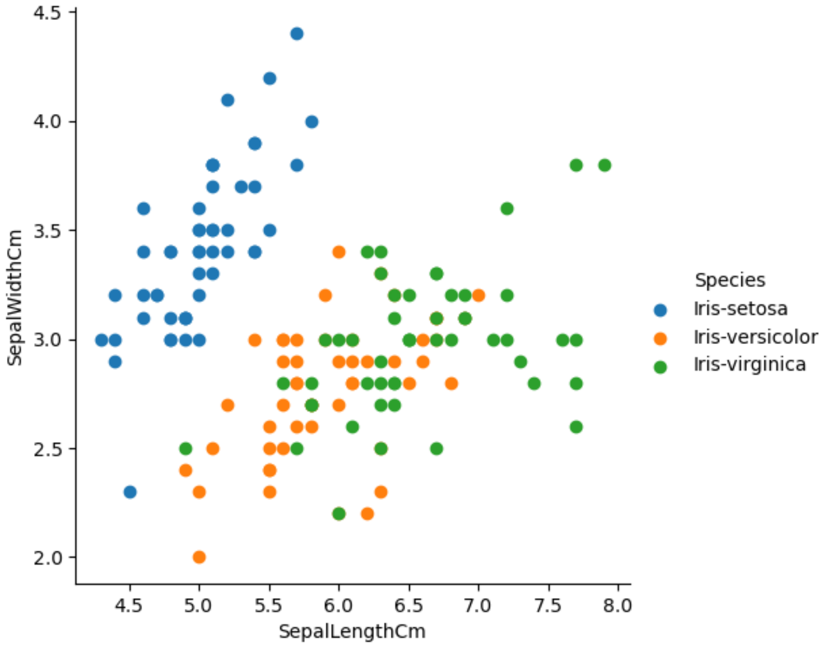
```
In [9]: sns.jointplot(data=iris,x='SepalLengthCm',y='SepalWidthCm',kind='reg')
```

```
Out[9]: <seaborn.axisgrid.JointGrid at 0x1b36b67f4d0>
```



In [10]:
```
sns.FacetGrid(iris,hue='Species',height=5,aspect=1).map(plt.scatter,'SepalLengthCm','SepalWidthCm').add_legend()
```

Out[10]:
```
<seaborn.axisgrid.FacetGrid at 0x1b36b72d450>
```

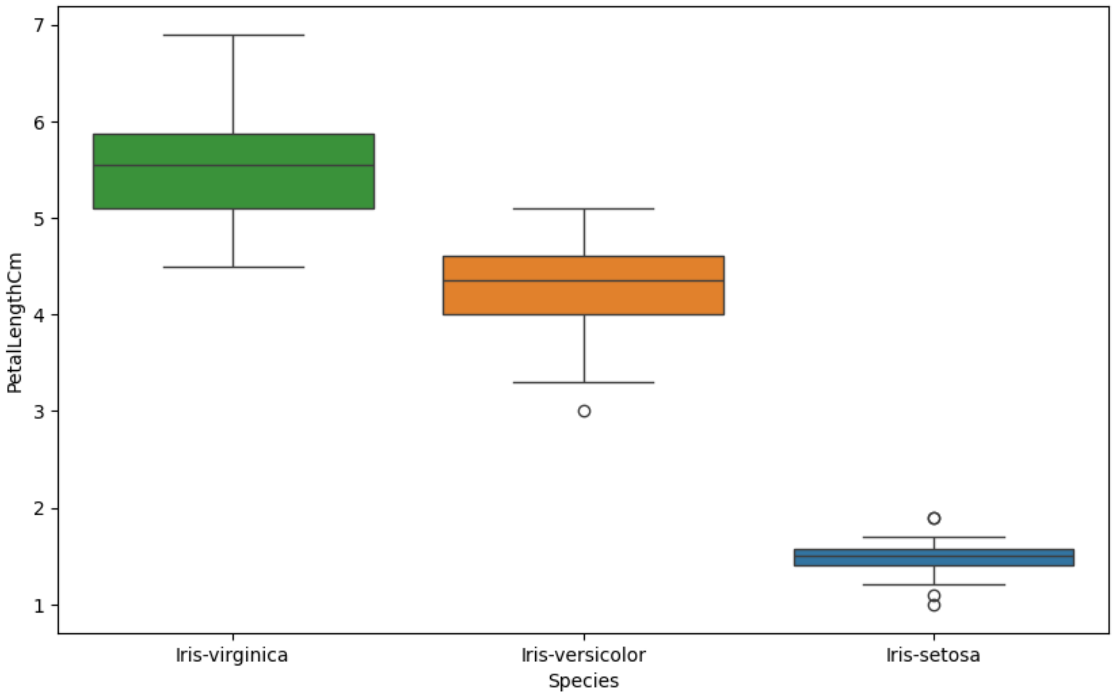


In [11]:
```
iris.head()
```

Out[11]:

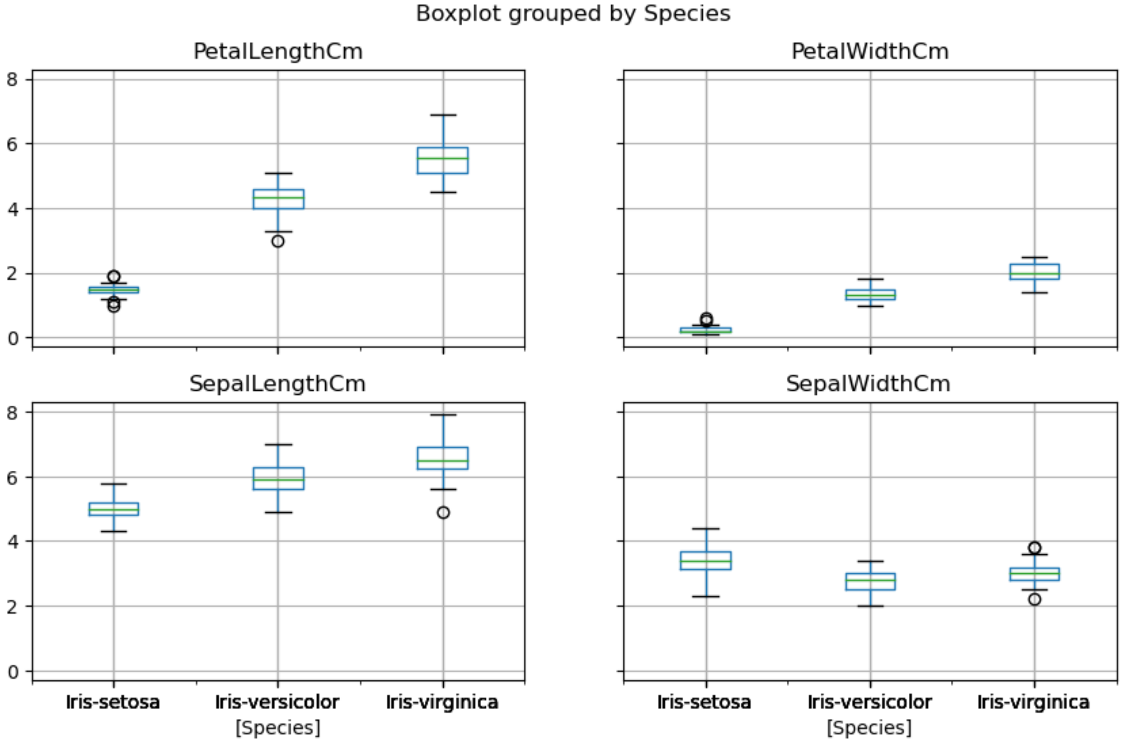
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [12]:
```
fig=plt.gcf()
fig.set_size_inches(10,6)
fig=sns.boxplot(data=iris,x='Species',y='PetalLengthCm',order=['Iris-virginica','Iris-versicolor','Iris-setosa'],hue='Species'
```

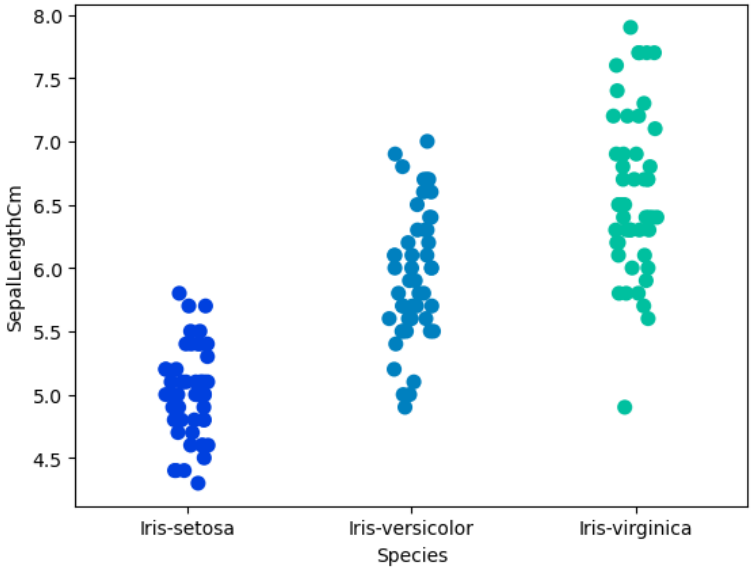


```
In [14]: iris.boxplot(by='Species',figsize=(10,6))
```

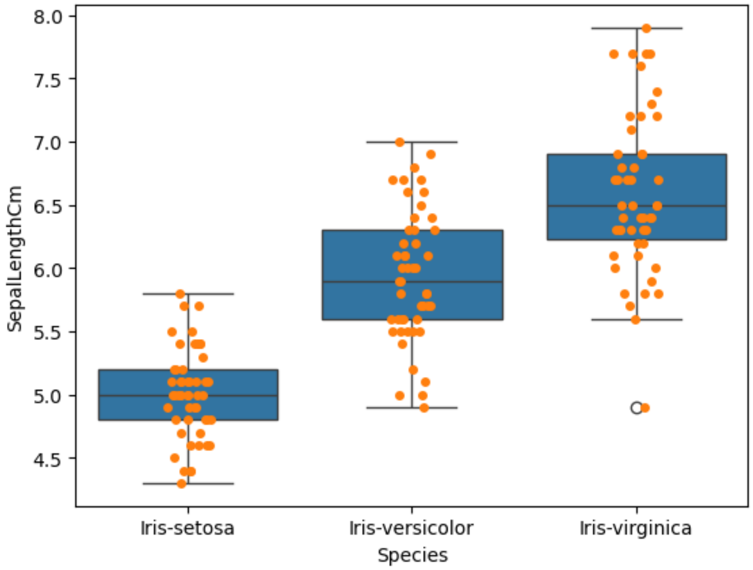
```
Out[14]: array([[<Axes: title={'center': 'PetalLengthCm'}, xlabel='[Species]'>,
                 <Axes: title={'center': 'PetalWidthCm'}, xlabel='[Species]'>],
                [<Axes: title={'center': 'SepalLengthCm'}, xlabel='[Species]'>,
                 <Axes: title={'center': 'SepalWidthCm'}, xlabel='[Species]'>]],
               dtype=object)
```



```
In [16]: fig=sns.stripplot(data=iris,x='Species',y='SepalLengthCm',jitter=True,edgecolor='grey',size=8,palette='winter',orient='v')
```



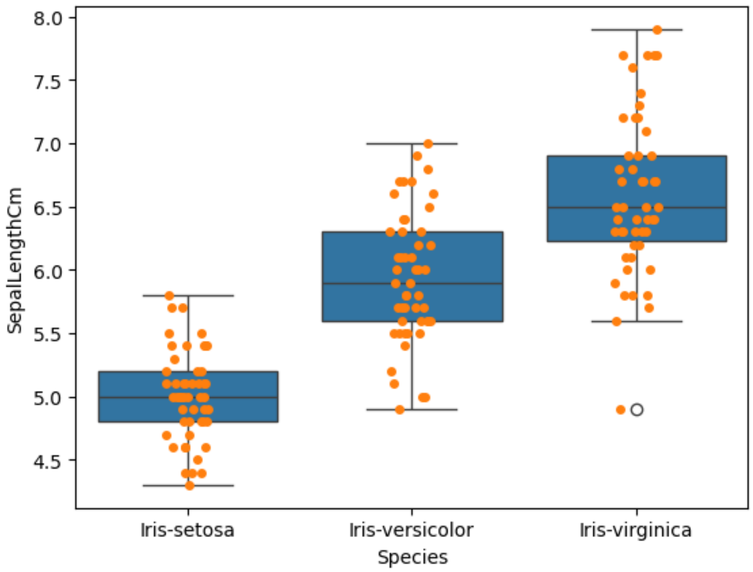
```
In [17]: fig=sns.boxplot(data=iris,x='Species',y='SepalLengthCm')
         fig=sns.stripplot(data=iris,x='Species',y='SepalLengthCm',jitter=True,edgecolor='grey')
```



In [31]:

```
ax=sns.boxplot(data=iris,x='Species',y='SepalLengthCm')
ax=sns.stripplot(data=iris,x='Species',y='SepalLengthCm',jitter=True,edgecolor='grey')
artists = ax.artists
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')
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[31], line 4
      2 ax=sns.stripplot(data=iris,x='Species',y='SepalLengthCm',jitter=True,edgecolor='grey')
      3 artists = ax.artists
----> 4 boxtwo = ax.artists[2]
      5 boxtwo.set_facecolor('yellow')
      6 boxtwo.set_edgecolor('black')

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:1457, in _AxesBase.ArtistList.__getitem__(self, key)
   1456 def __getitem__(self, key):
-> 1457     return [artist
   1458             for artist in self._axes._children
   1459             if self._type_check(artist)][key]

IndexError: list index out of range
```



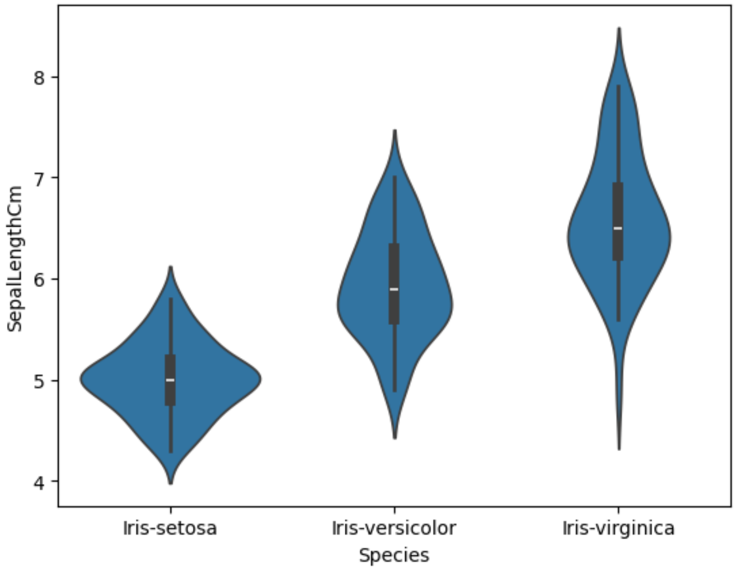
In [29]:

```
len(artists)
```

Out[29]:

```
0
```

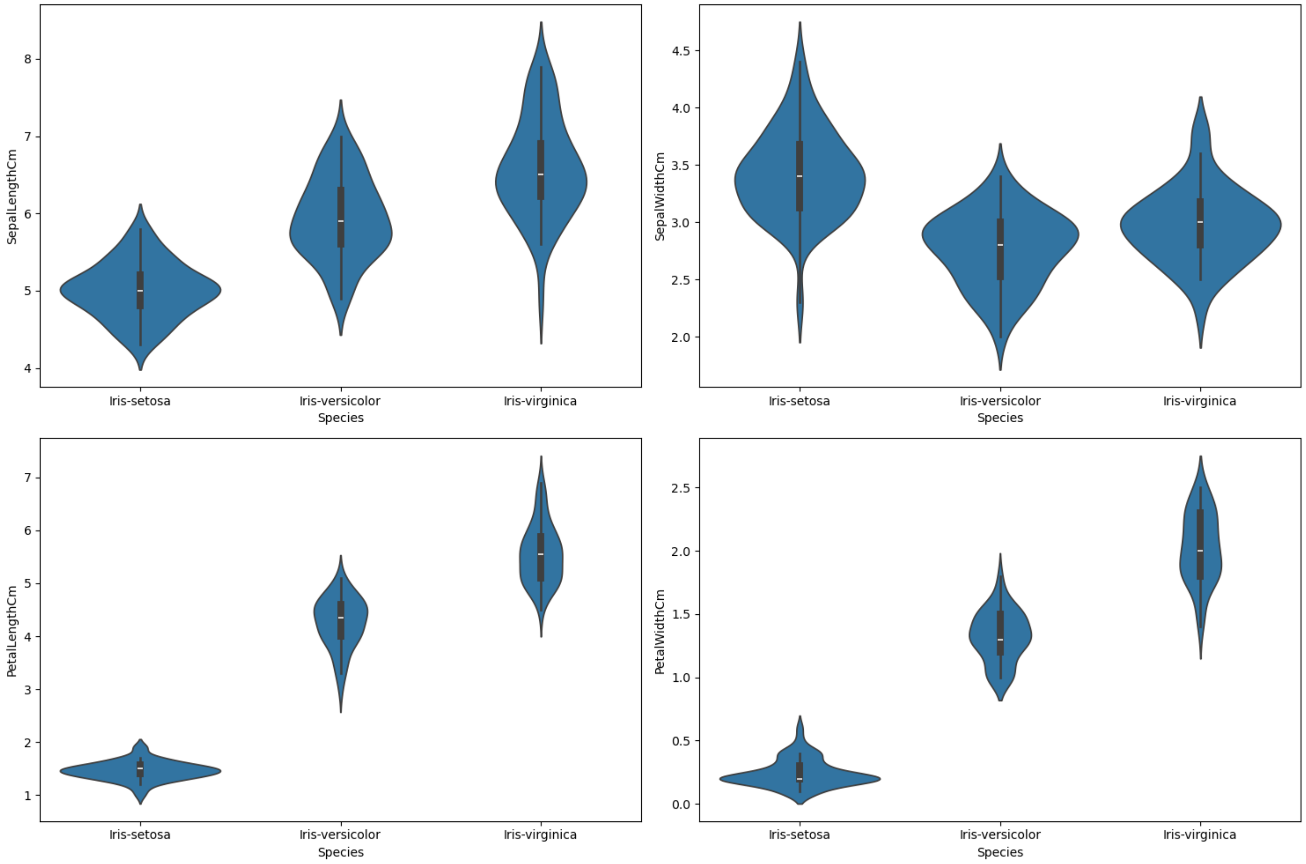
In [32]:
```
fig=sns.violinplot(data=iris,x='Species',y='SepalLengthCm')
```



In [34]:
```
iris.columns
```

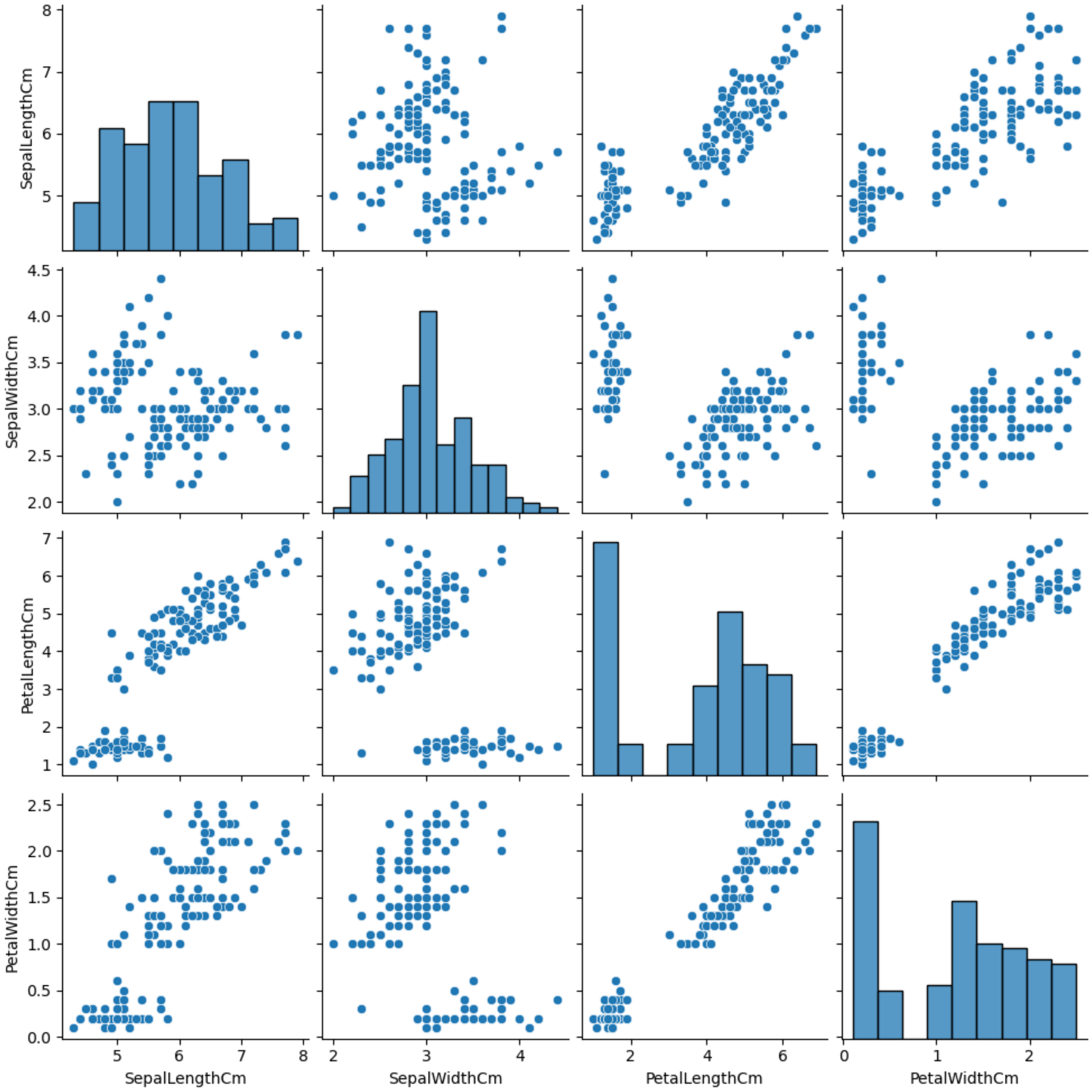
Out[34]:
```
Index(['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

In [36]:
```
plt.figure(figsize=(15,10))
plt.subplot(2,2,1)
sns.violinplot(data=iris,x='Species',y='SepalLengthCm')
plt.subplot(2,2,2)
sns.violinplot(data=iris,x='Species',y='SepalWidthCm')
plt.subplot(2,2,3)
sns.violinplot(data=iris,x='Species',y='PetalLengthCm')
plt.subplot(2,2,4)
sns.violinplot(data=iris,x='Species',y='PetalWidthCm')
plt.tight_layout() # Adjust layout to prevent overlap
```



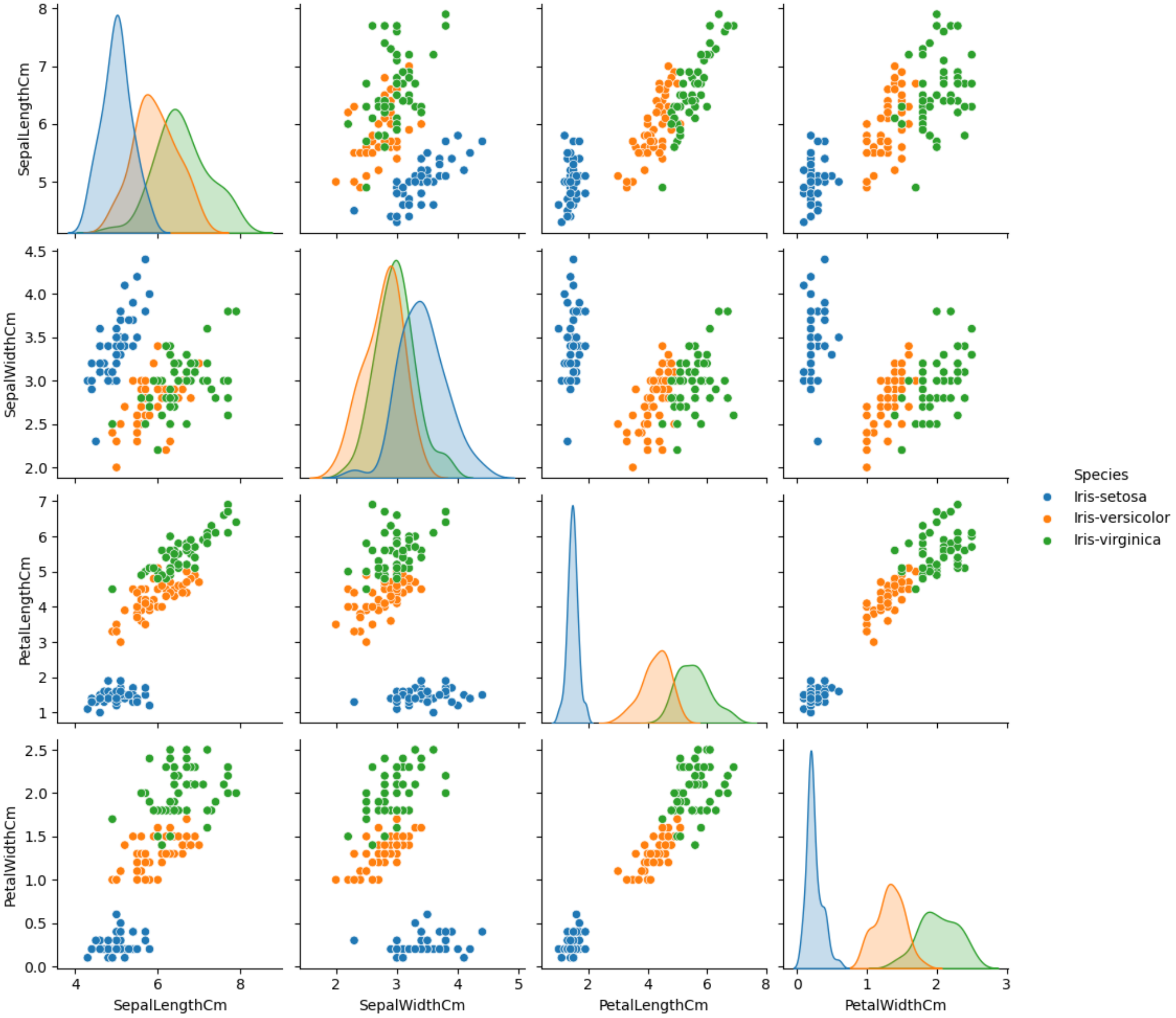
```
In [37]: sns.pairplot(data=iris)
```

```
Out[37]: <seaborn.axisgrid.PairGrid at 0x1b377cf6a10>
```



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

```
Out[38]: <seaborn.axisgrid.PairGrid at 0x1b3785c1450>
```



In [42]:
```
fig=sns.heatmap(iris.corr(),annot=True,cmap='cubehelix',linewidths=1,linecolor='k', square=True,mask=False,vmax=1,vmin=-1,cbar_
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[42], line 1
----> 1 fig=sns.heatmap(iris.corr(),annot=True,cmap='cubehelix',linewidths=1,linecolor='k', square=True,mask=False,vmax=1,vmi
n=-1,cbar_kws={"orientation":"vertical"},cbar=True)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:10054, in DataFrame.corr(self, method, min_periods, numeric_only)
  10052 cols = data.columns
  10053 idx = cols.copy()
> 10054 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  10056 if method == "pearson":
  10057     correl = libalgos.nancorr(mat, minp=min_periods)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:1838, in DataFrame.to_numpy(self, dtype, copy, na_value)
   1836 if dtype is not None:
   1837     dtype = np.dtype(dtype)
-> 1838 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
   1839 if result.dtype is not dtype:
   1840     result = np.array(result, dtype=dtype, copy=False)

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1732, in BlockManager.as_array(self, dtype, copy, na_val
ue)
   1730         arr.flags.writeable = False
   1731 else:
-> 1732     arr = self._interleave(dtype=dtype, na_value=na_value)
   1733     # The underlying data was copied within _interleave, so no need
   1734     # to further copy if copy=True or setting na_value
   1736 if na_value is not lib.no_default:

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1794, in BlockManager._interleave(self, dtype, na_value)
   1792     else:
   1793         arr = blk.get_values(dtype)
-> 1794     result[rl.indexer] = arr
   1795     itemmask[rl.indexer] = 1
   1797 if not itemmask.all():

ValueError: could not convert string to float: 'Iris-setosa'
```

In [46]:
```
fig2=iris.drop('Species',axis=1)
```

In [47]:
```
fig2
```

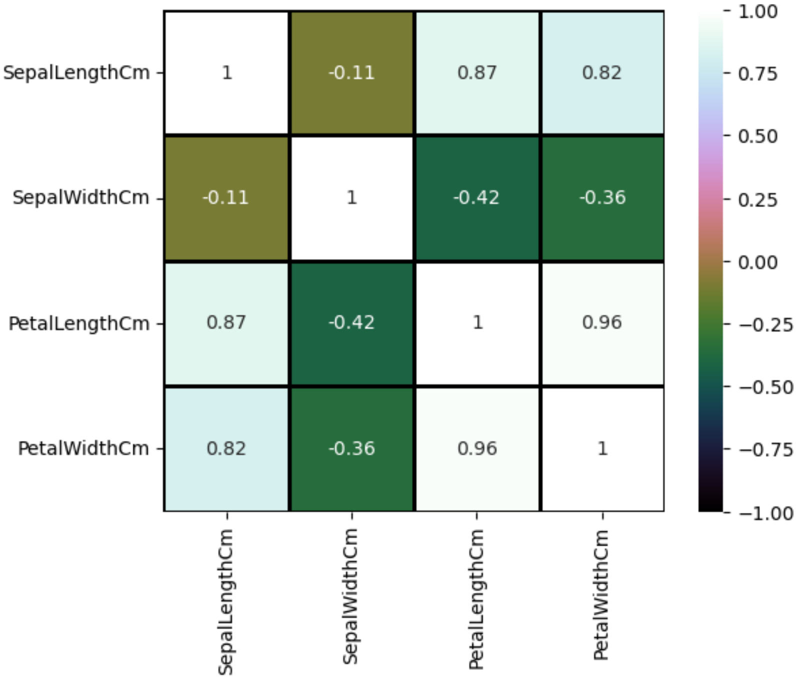
Out[47]:

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

150 rows × 4 columns

In [48]:

```
cmap='cubehelix',linewidths=1,linecolor='k', square=True,mask=False,vmax=1,vmin=-1,cbar_kws={"orientation":"vertical"},cbar=Tru
```

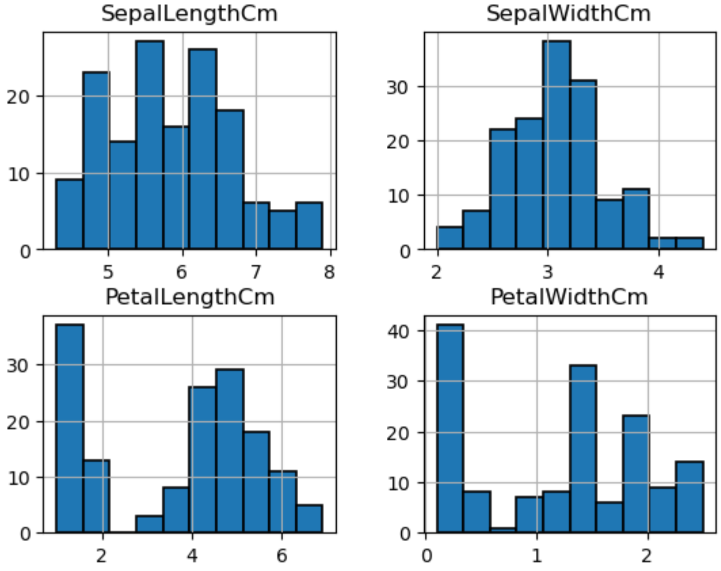


In [50]:

```
iris.hist(edgecolor='black', linewidth=1.2)
```

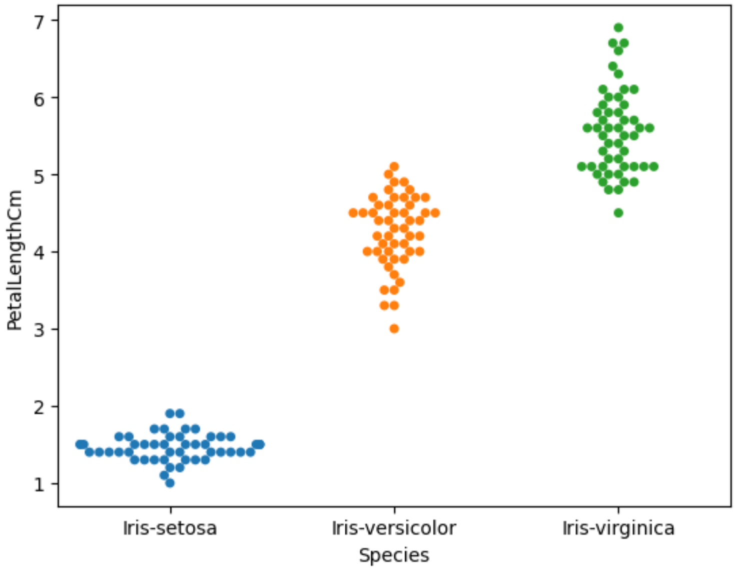
Out[50]:

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

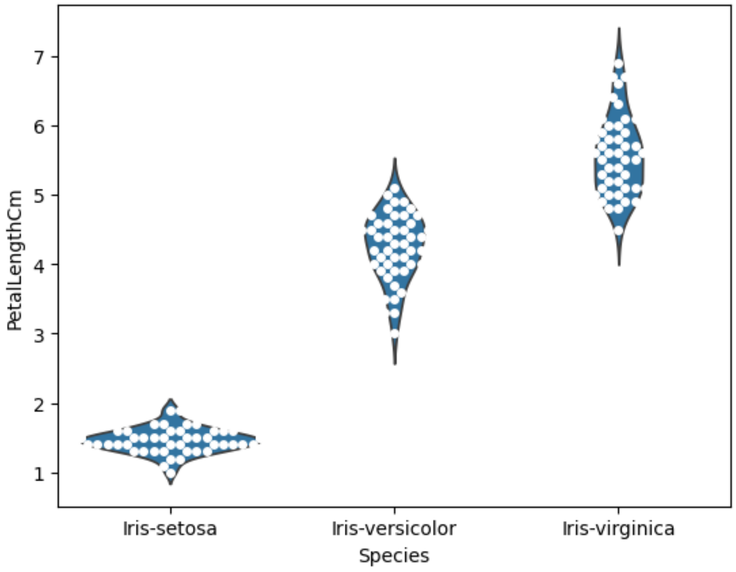


```
In [52]: sns.swarmplot(data=iris,x='Species',y='PetalLengthCm',hue='Species')
```

```
Out[52]: <Axes: xlabel='Species', ylabel='PetalLengthCm'>
```



```
In [56]: zx=sns.violinplot(data=iris,x='Species',y='PetalLengthCm',inner=None)
         zx=sns.swarmplot(data=iris,x='Species',y='PetalLengthCm',edgecolor='black',color='white')
```

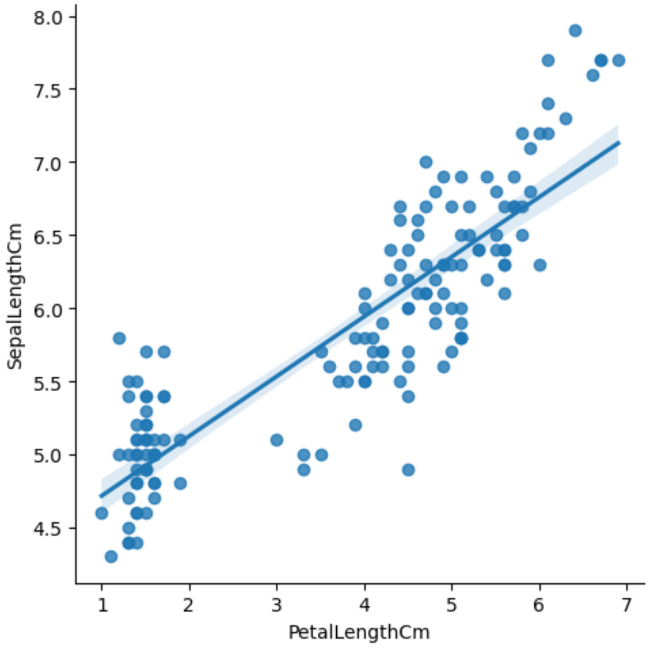


In [57]:

```
sns.lmplot(data=iris,x='PetalLengthCm',y='SepalLengthCm')
```

Out[57]:

```
<seaborn.axisgrid.FacetGrid at 0x1b379546890>
```

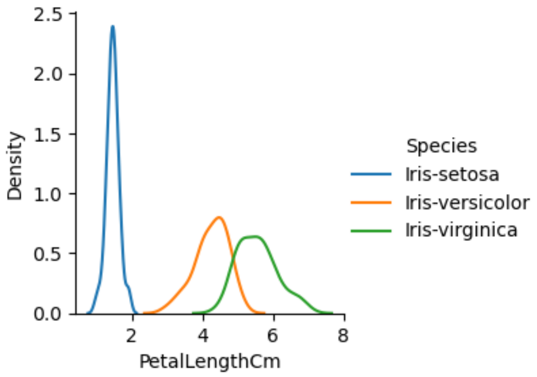


In [58]:

```
sns.FacetGrid(data=iris,hue='Species').map(sns.kdeplot,'PetalLengthCm').add_legend()
```

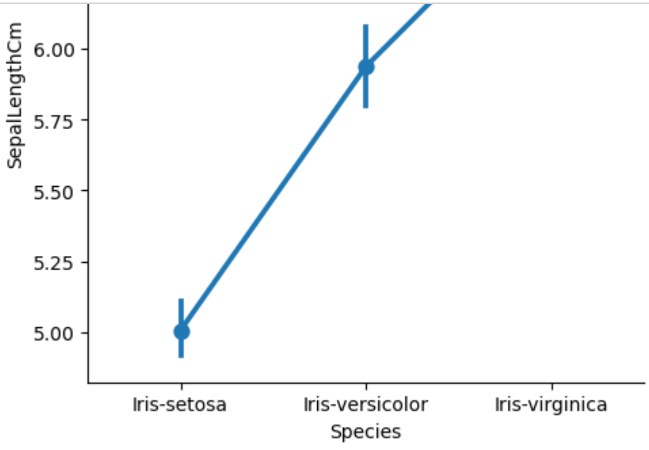
Out[58]:

```
<seaborn.axisgrid.FacetGrid at 0x1b37d89e5d0>
```

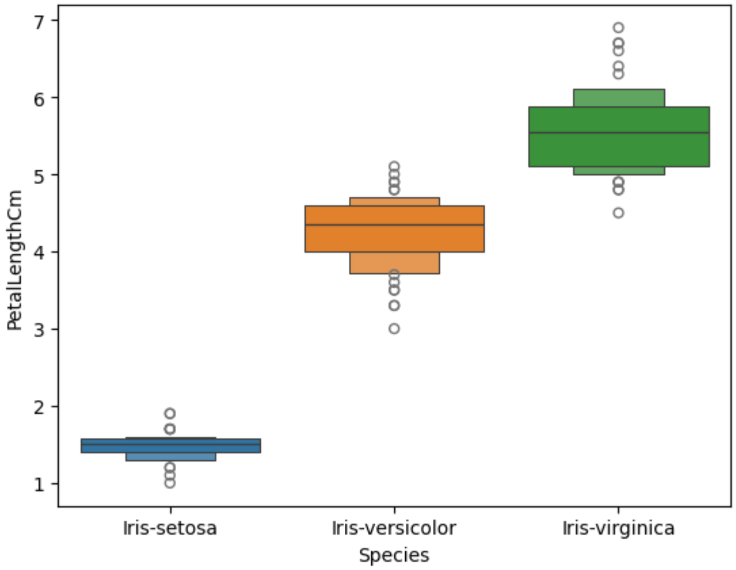


In [70]:

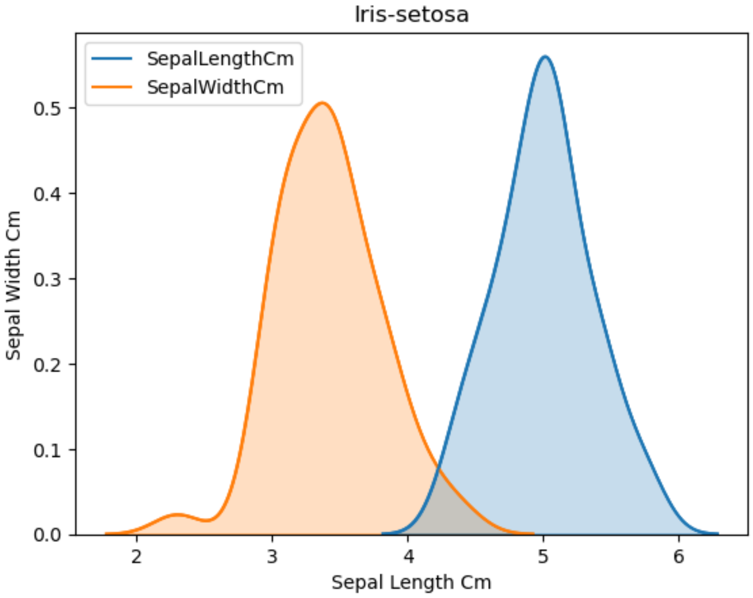
```
sns.catplot(x='Species',y='SepalLengthCm',data=iris,kind='point')
plt.ioff()
plt.show()
```



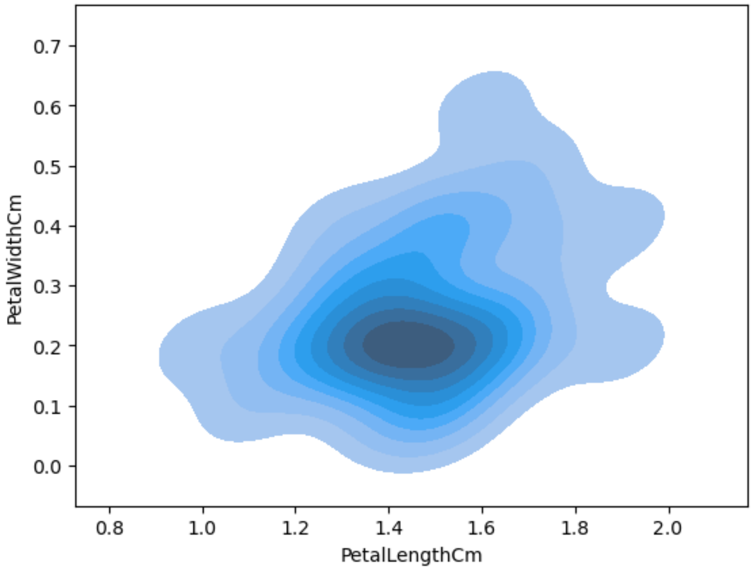
In [73]:
```
sns.boxenplot(data=iris,x='Species',y='PetalLengthCm',hue='Species')
plt.show()
```



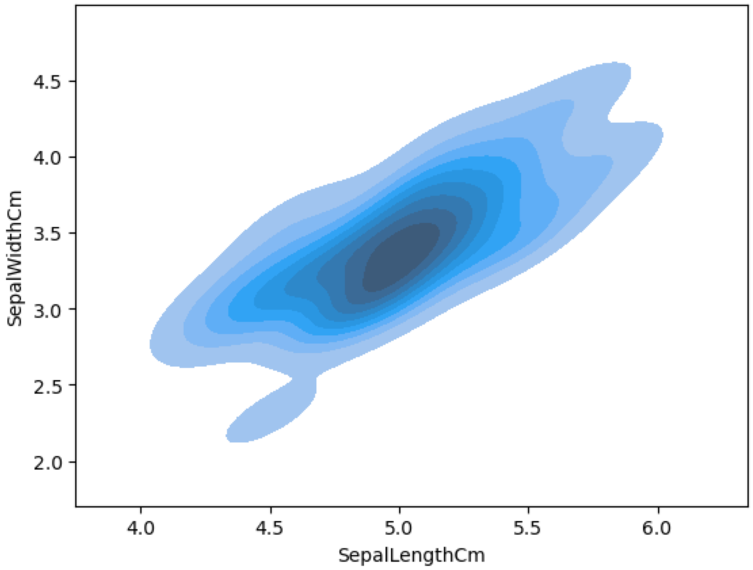
In [77]:
```
sub=iris[iris['Species']=='Iris-setosa']
sns.kdeplot(data=sub[['SepalLengthCm','SepalWidthCm']])
plt.title('Iris-setosa')
plt.xlabel('Sepal Length Cm')
plt.ylabel('Sepal Width Cm')
plt.show()
```



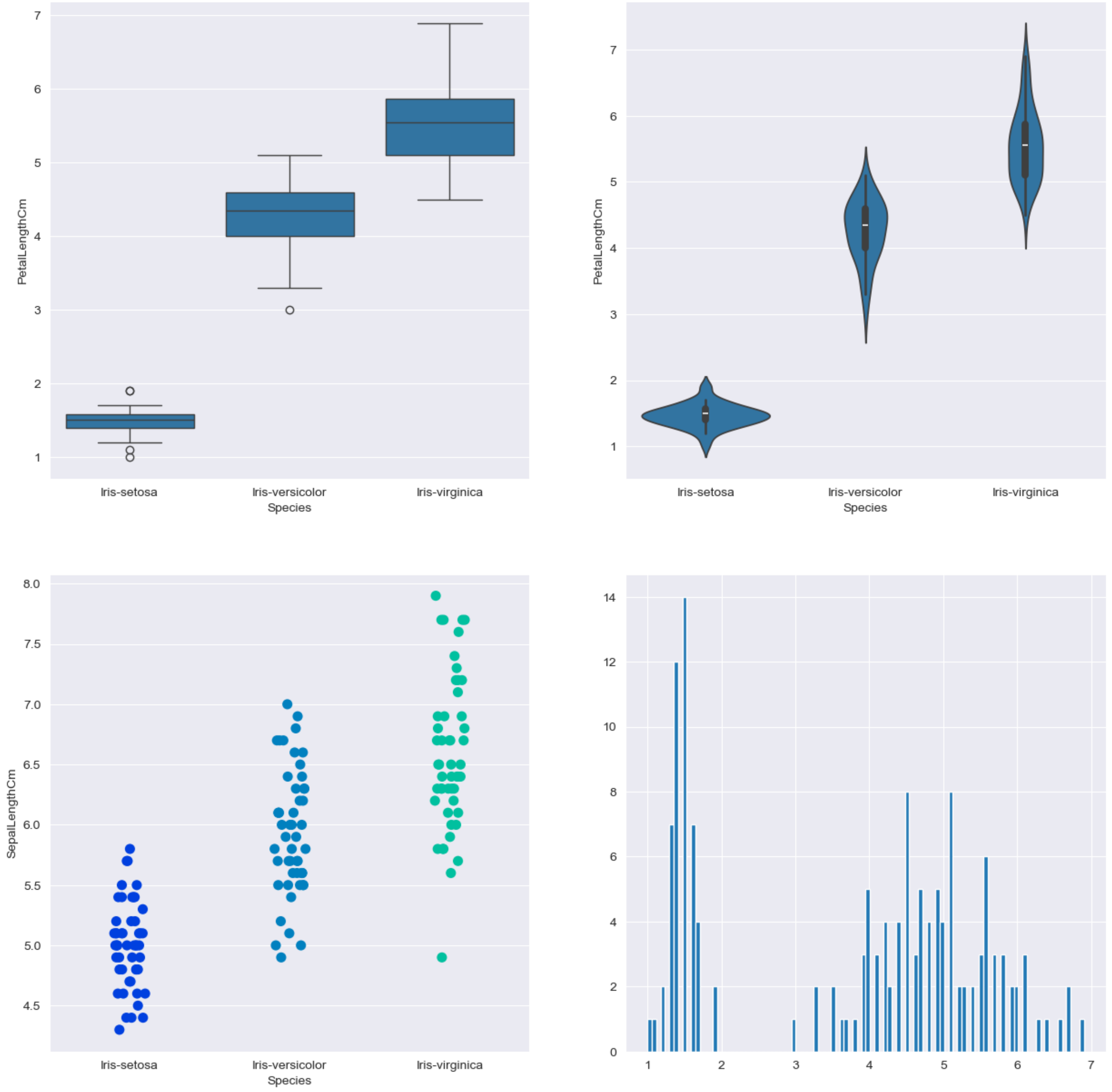
In [84]:
```
sub=iris[iris['Species']=='Iris-setosa']
sns.kdeplot(data=sub,x='PetalLengthCm',y='PetalWidthCm', shade=True, shade_lowest=False)
plt.show()
```



In [85]:
```
sub=iris[iris['Species']=='Iris-setosa']
sns.kdeplot(data=sub,x='SepalLengthCm',y='SepalWidthCm', shade=True, shade_lowest=False)
plt.show()
```



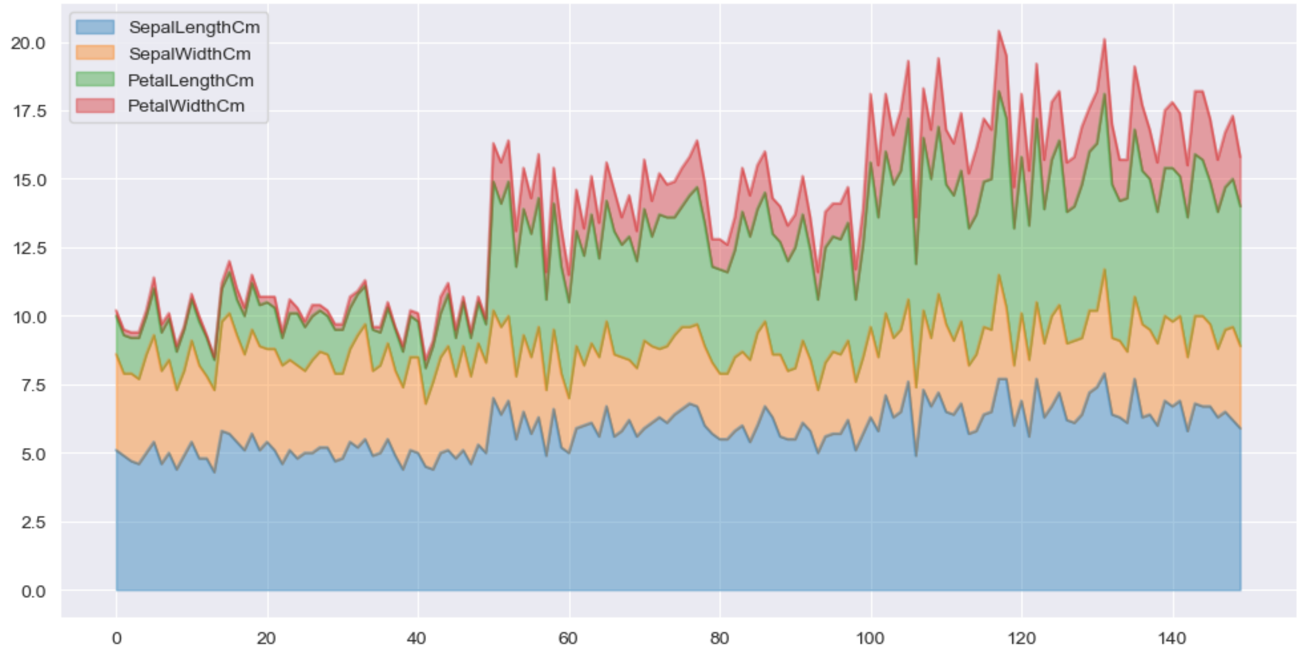
In [92]:
```
sns.set_style('darkgrid')
f,axes=plt.subplots(2,2,figsize=(15,15))

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='gray',size=8,palette='winter',orient='v',ax=axe
axes[1,1].hist(iris.PetalLengthCm,bins=100)
plt.show()
```



In [93]:
```
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 [96]:
```
iris.plot.area(y=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'],alpha=0.4,figsize=(12, 6))
plt.show()
plt.gcf()
```



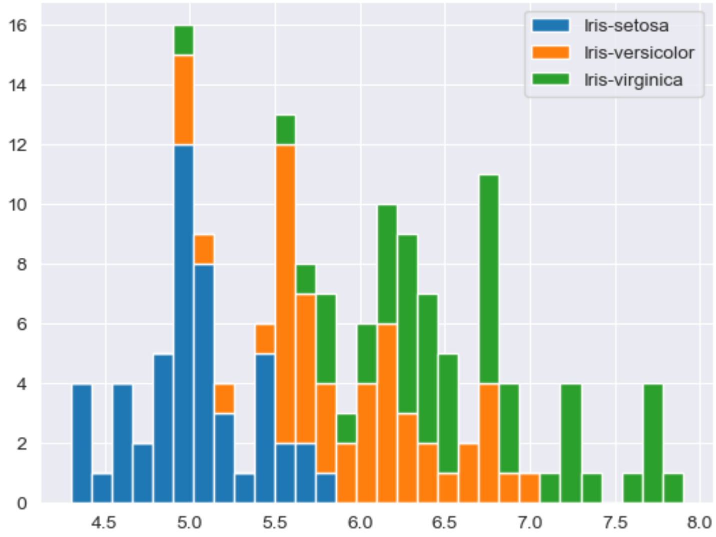
Out[96]:
```
<Figure size 640x480 with 0 Axes>
```

In [99]:
```
iris['Species']=iris['Species'].astype('category')
```

In [100]:
```
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    category
dtypes: category(1), float64(4)
memory usage: 5.1 KB
```

```
In [101]: 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 [ ]:
```