IRIS DATASET VISUALIZATION

In [2]:
```
import numpy as np
import pandas as pd
```

In [3]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
iris = pd.read_csv('/Users/chandnisingh/Downloads/5th, 6th - Sql workshop
```

In [5]:
```
iris
```

Out[5]:

	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 [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

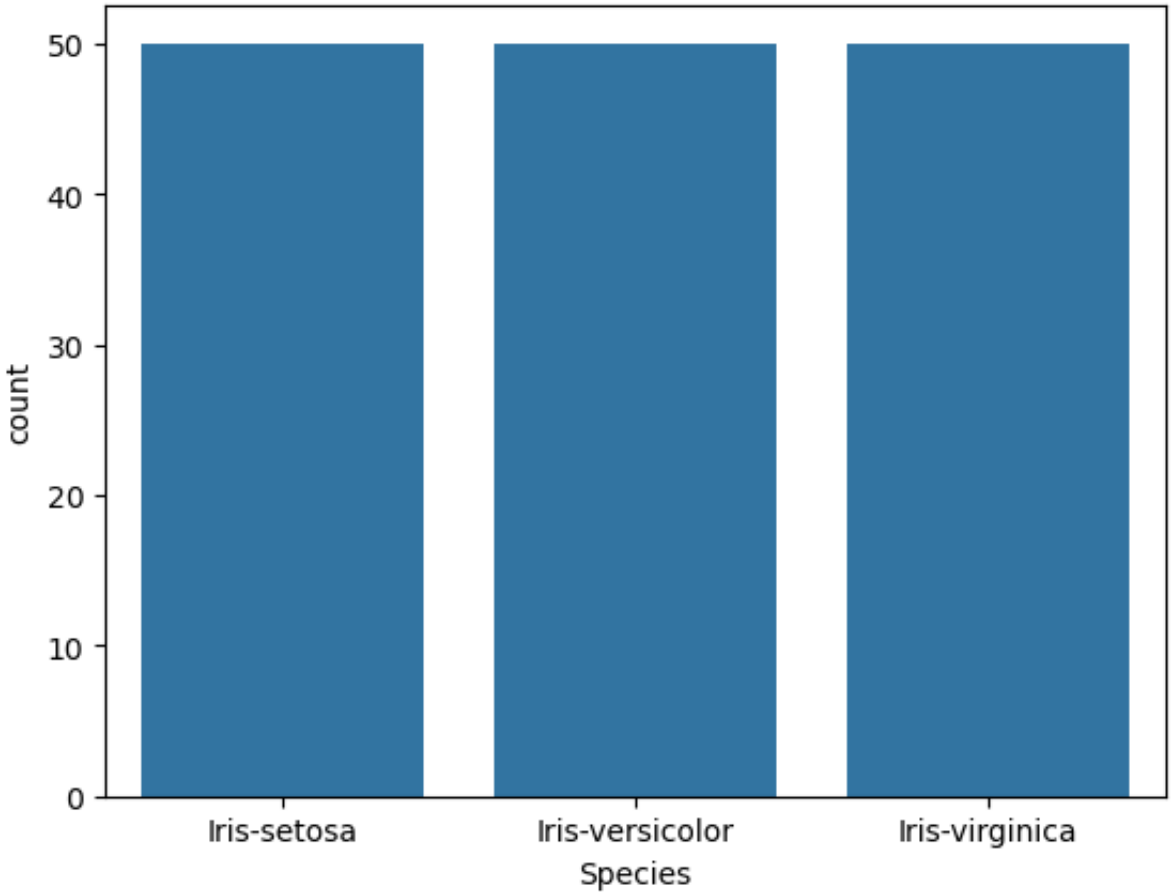
In [9]:
```
iris.info()
```

```
<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 [10]:
```
iris['Species'].value_counts()
```

Out[10]:
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

In [11]:
```
sns.countplot(iris , x='Species' )
plt.show()
```

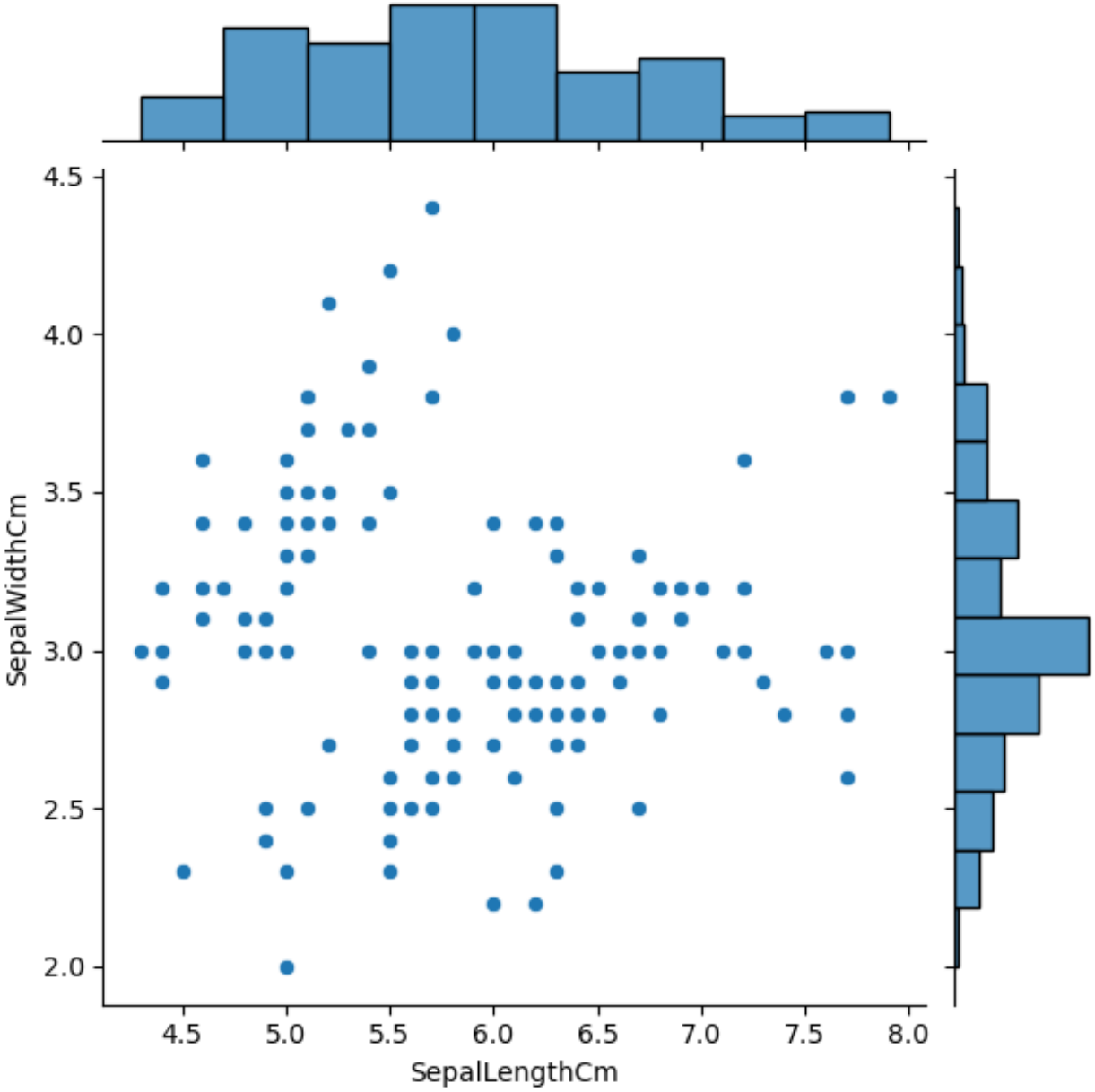


In [12]:
```
iris.head()
```

Out[12]:

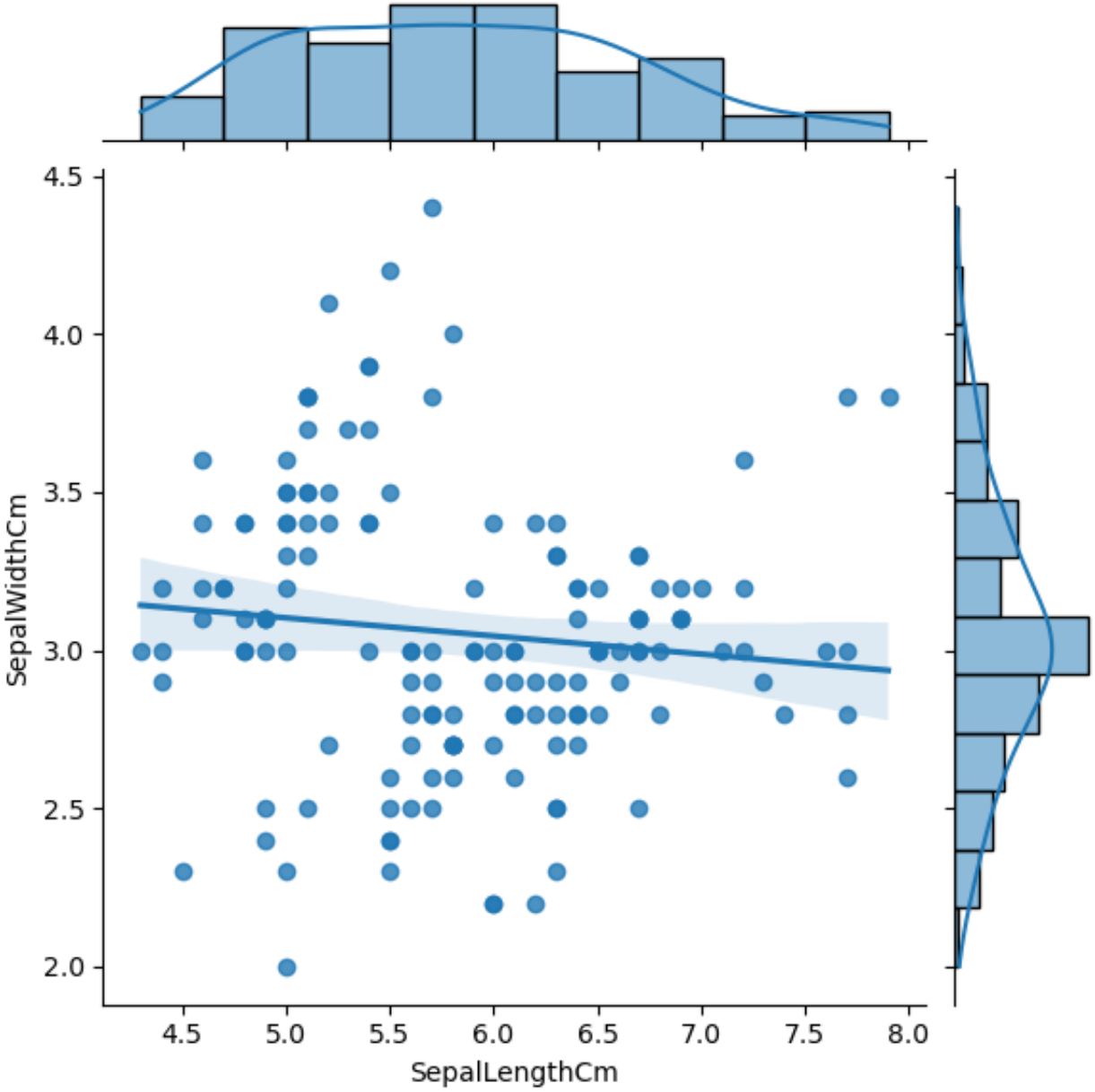
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [13]:
```
fig = sns.jointplot(x = 'SepalLengthCm' , y = 'SepalWidthCm' , data = iri
```



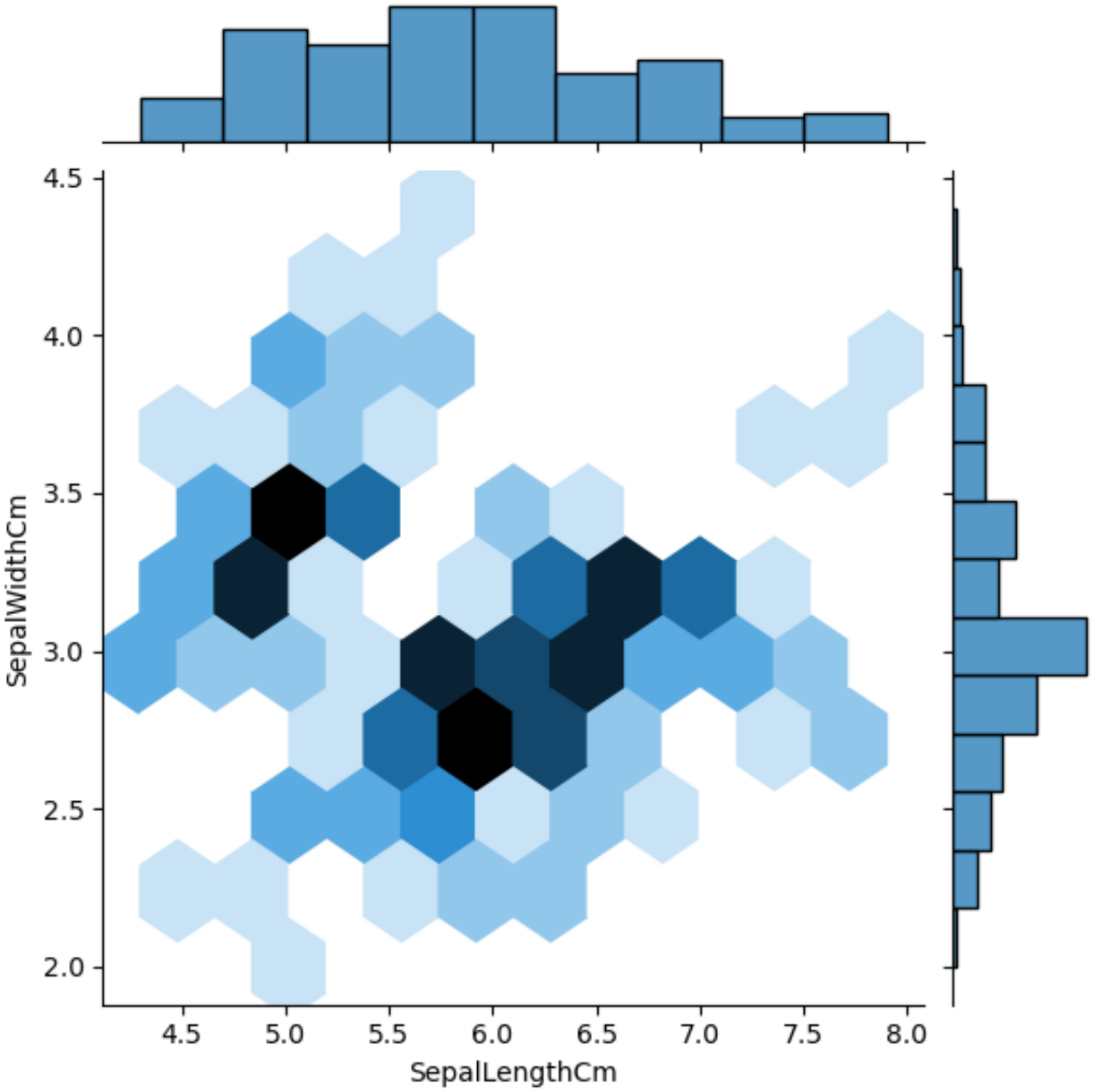
In [14]:
```
sns.jointplot(x = "SepalLengthCm",y = "SepalWidthCm", data=iris, kind="re
```

Out[14]: <seaborn.axisgrid.JointGrid at 0x12fe94440>



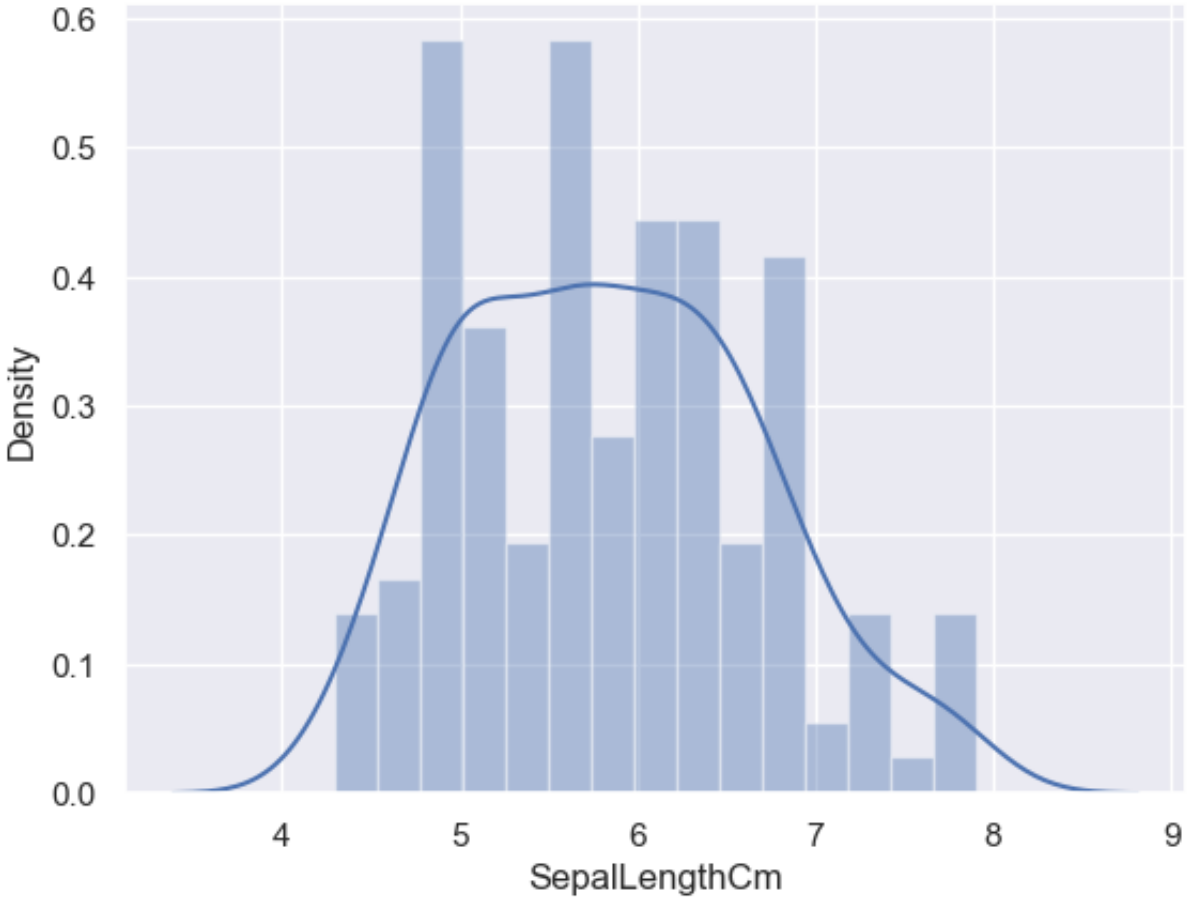
In [15]:
```
sns.jointplot(x = "SepalLengthCm",y = "SepalWidthCm", data=iris, kind="he
```

Out[15]: <seaborn.axisgrid.JointGrid at 0x12ffcad80>



In [105…
```
#Distplot

sns.distplot(iris['SepalLengthCm'],kde = True,bins = 15);
```

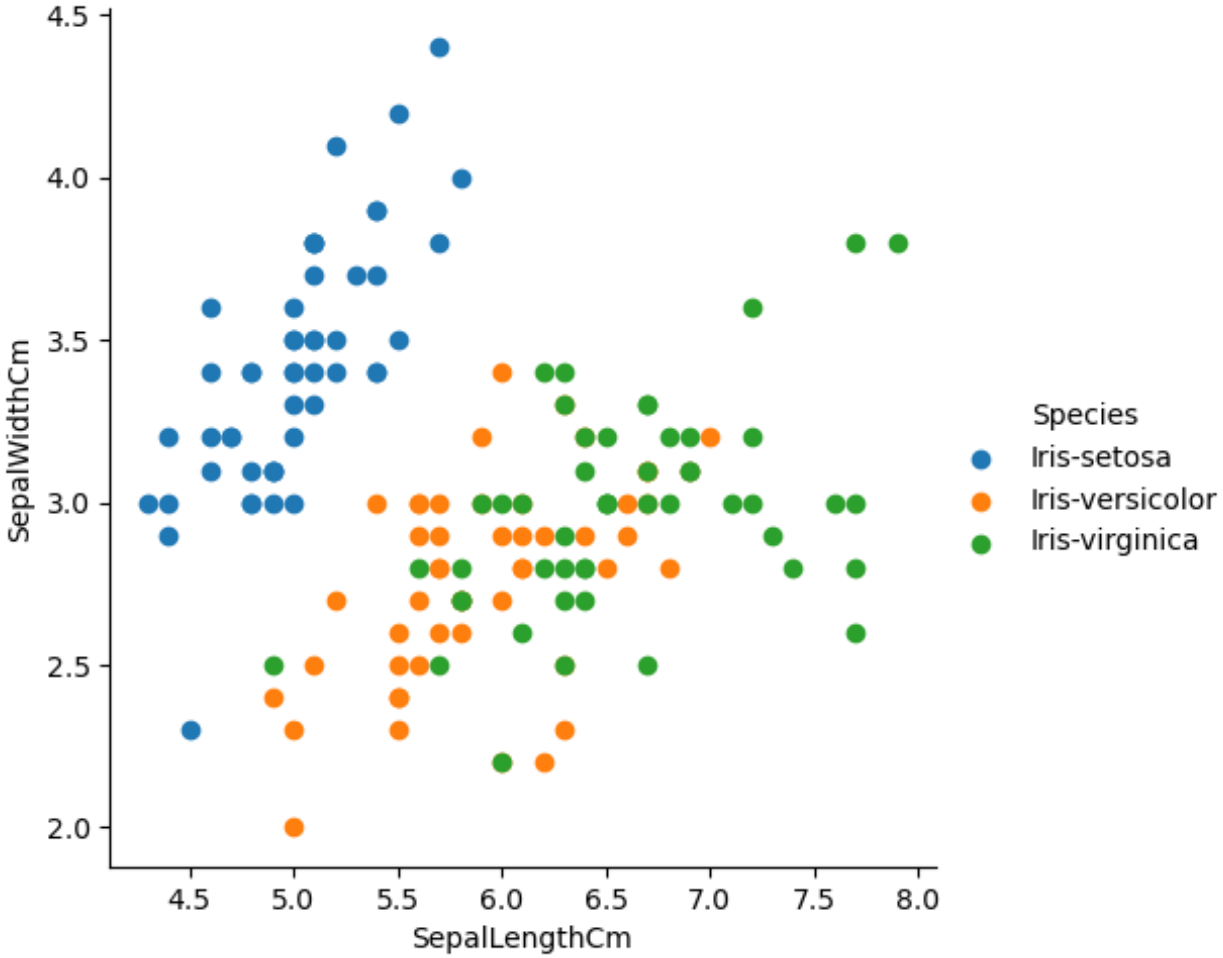
In [107…
```
plt.show()
```



In [16]:
```
%matplotlib inline
```

In [17]:
```
sns.FacetGrid(iris , hue='Species' ,height=5)\
.map(plt.scatter,'SepalLengthCm','SepalWidthCm')\
.add_legend()
```

Out[17]:
```
<seaborn.axisgrid.FacetGrid at 0x12ff70d10>
```

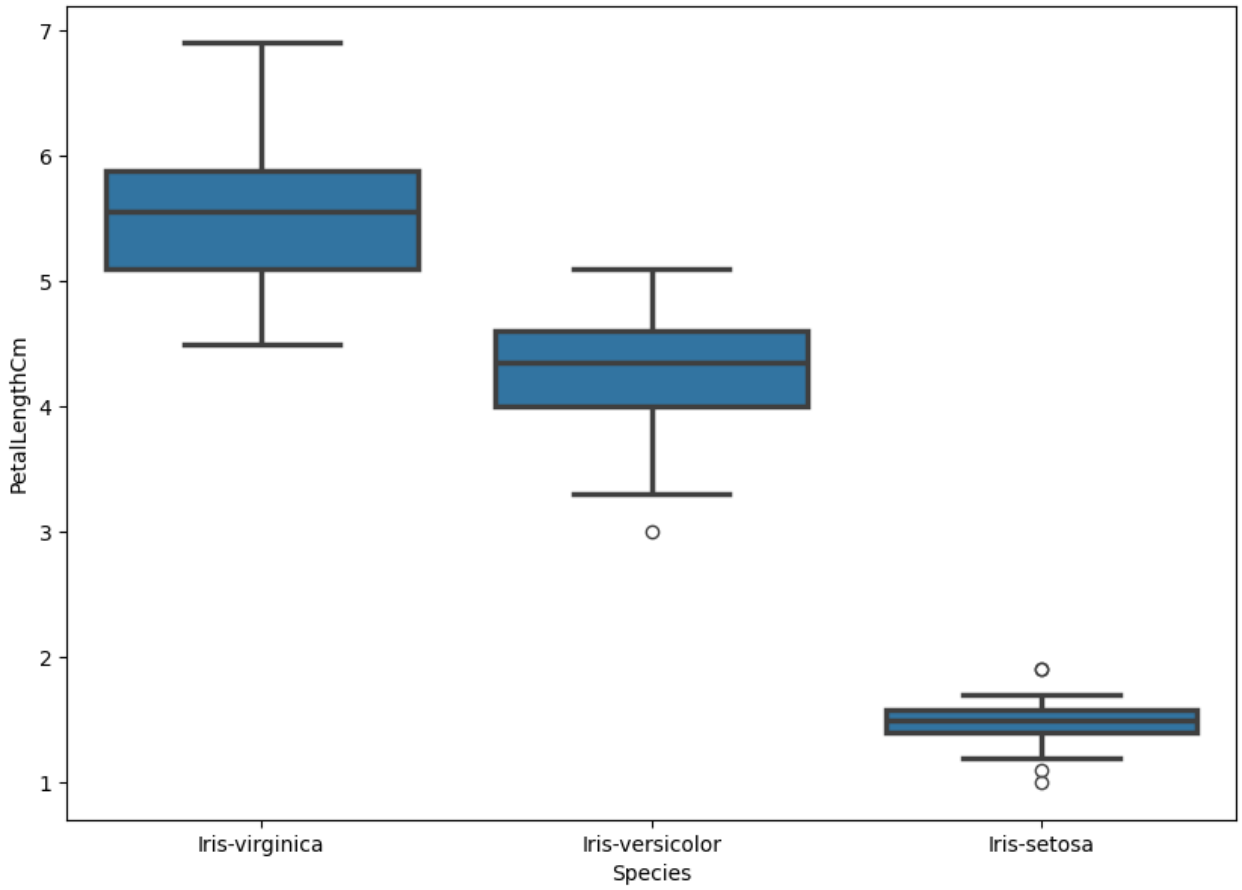


In [18]:
```
iris.head()
```

Out[18]:

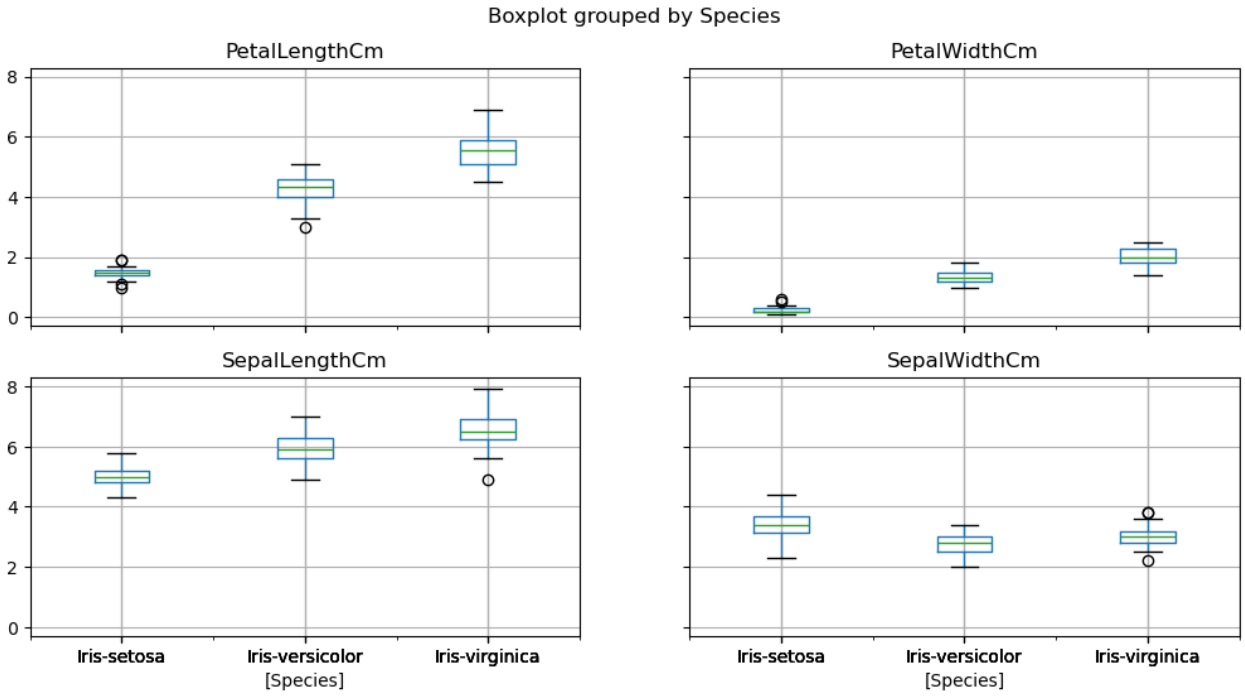
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [19]:
```
fig = plt.gcf()
fig.set_size_inches(10,7)
fig = sns.boxplot(x ='Species' , y = 'PetalLengthCm' , data= iris, order=
linewidth = 2.5,orient = 'v',dodge = False)
```

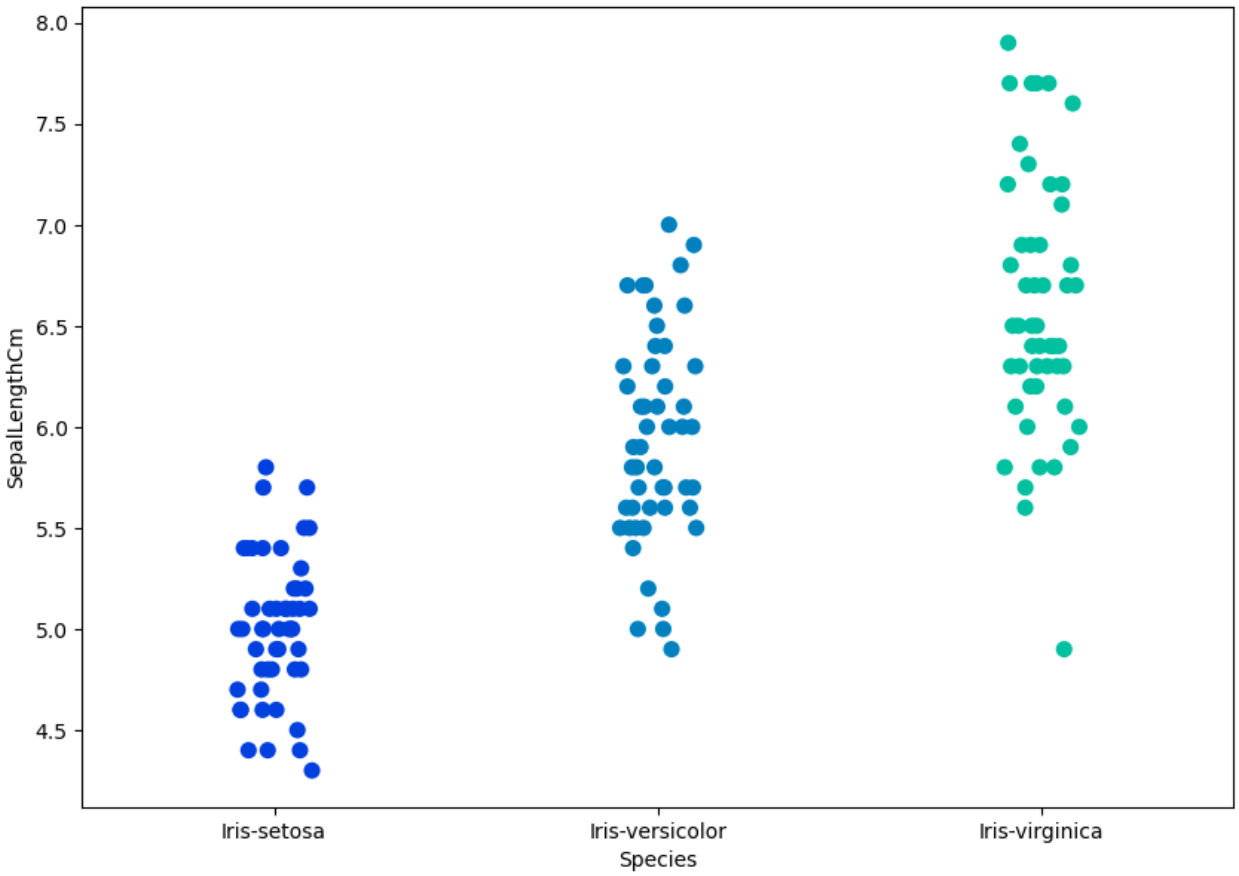


In [20]:
```
iris.boxplot(by="Species", figsize=(12, 6))
```

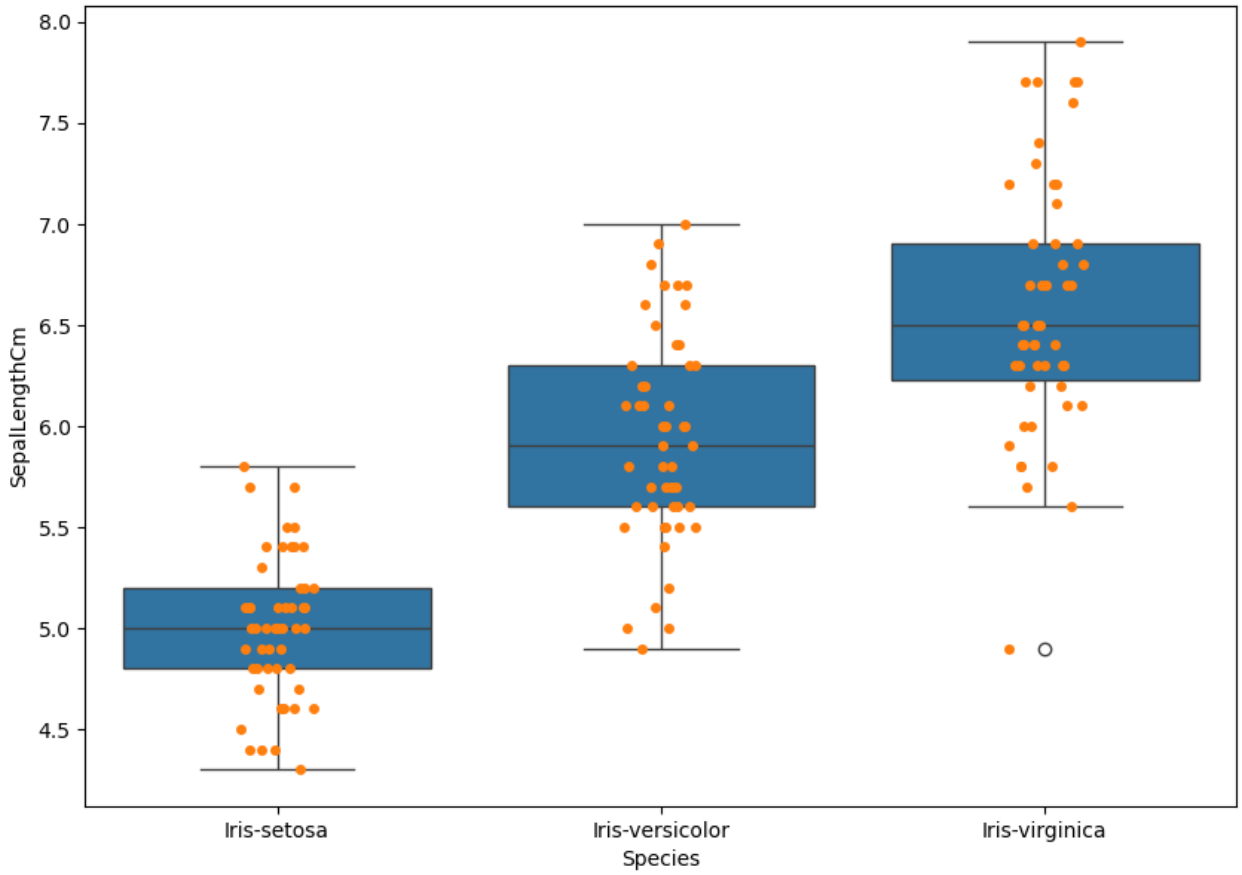
Out[20]:
```
array([[<Axes: title={'center': 'PetalLengthCm'}, xlabel='[Species]'>,
        <Axes: title={'center': 'PetalWidthCm'}, xlabel='[Species]'>],
       [<Axes: title={'center': 'SepalLengthCm'}, xlabel='[Species]'>,
        <Axes: title={'center': 'SepalWidthCm'}, xlabel='[Species]'>]],
      dtype=object)
```



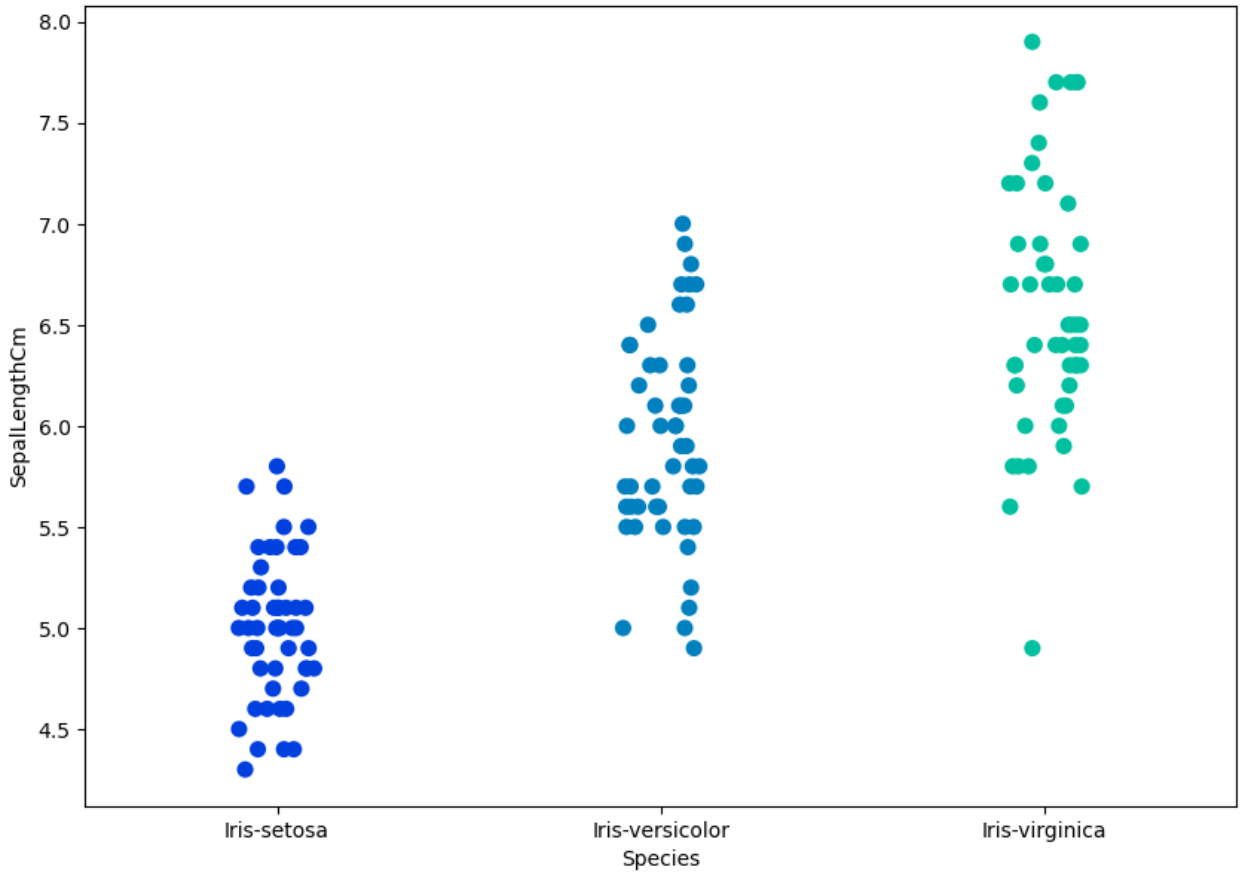
In [21]:
```
fig = plt.gcf()
fig.set_size_inches(10,7)
fig=sns.stripplot(x='Species' ,y = 'SepalLengthCm', data = iris,jitter =
```



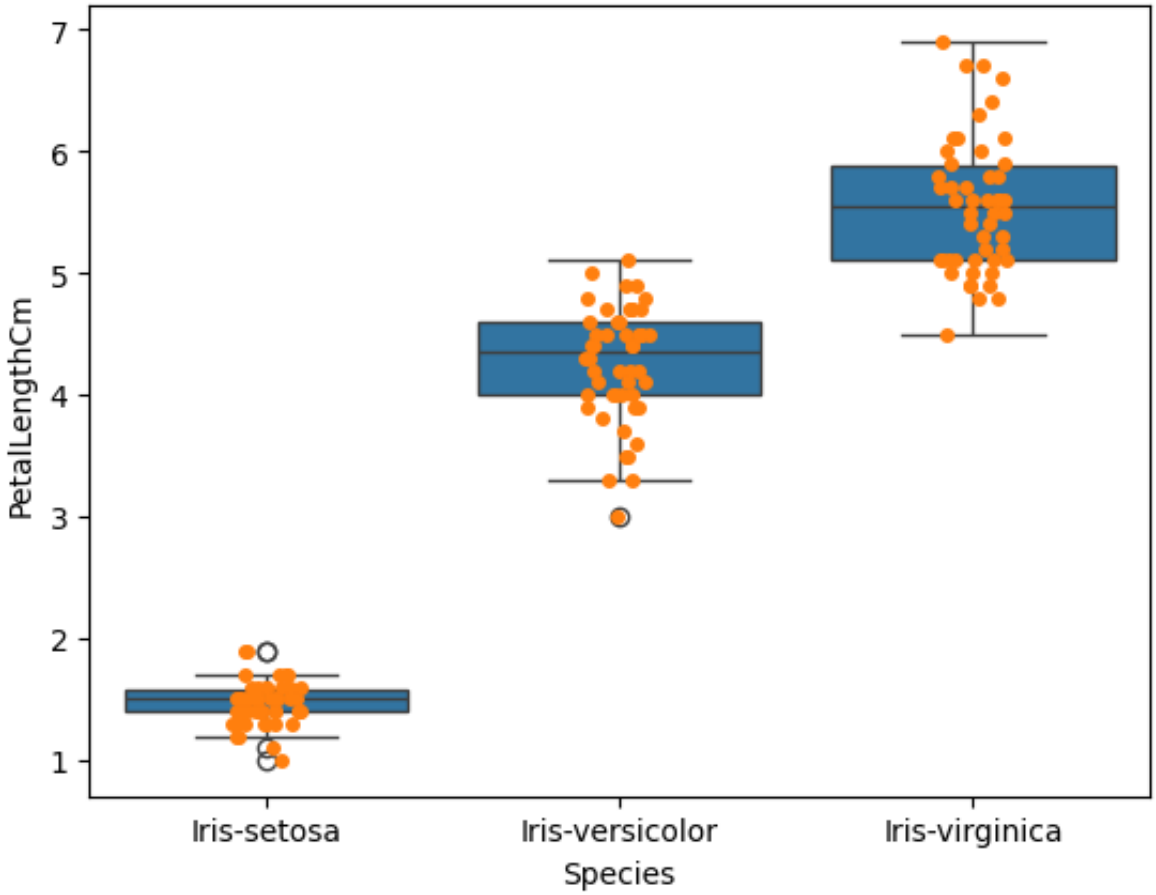
In [22]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='SepalLengthCm',data=iris)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edg
```



In [23]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edg
```

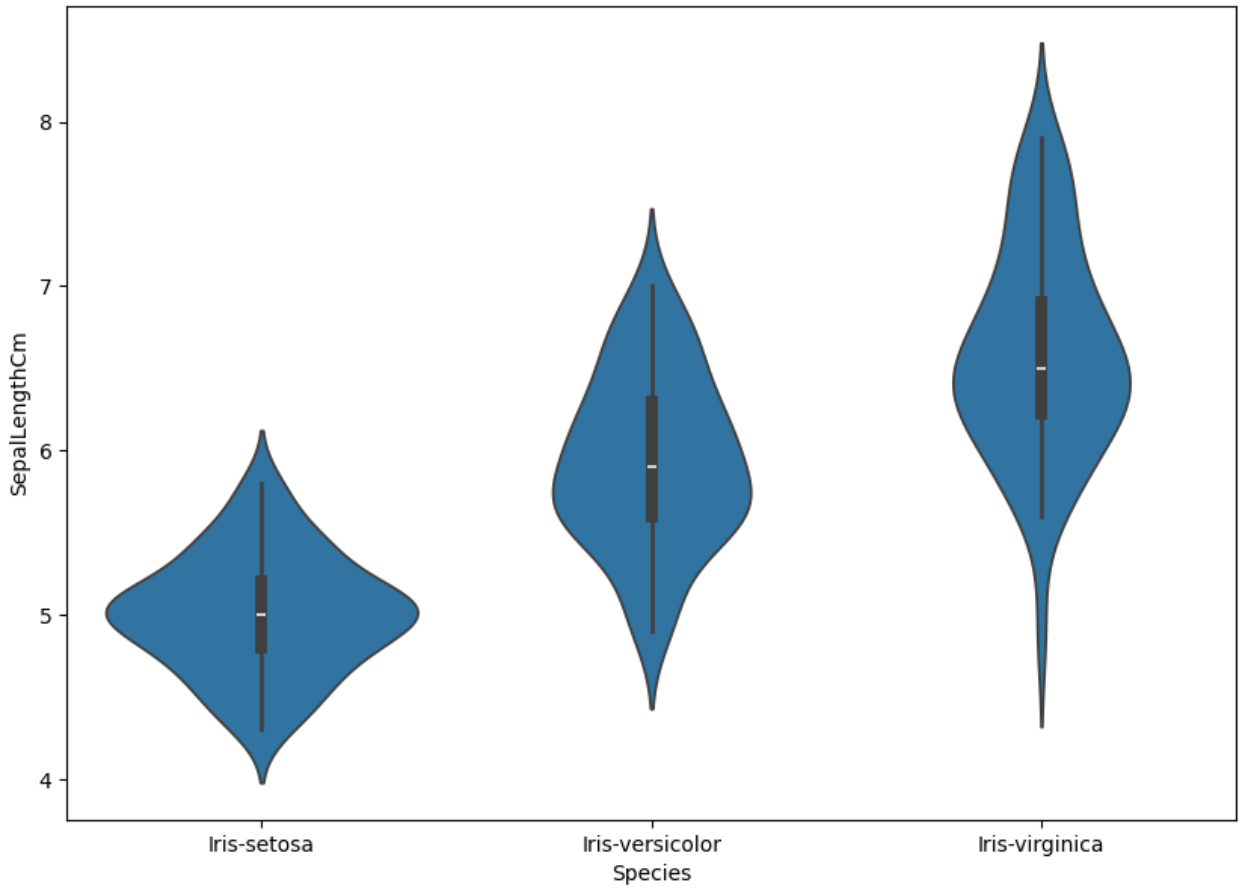


In [24]:
```
ax= sns.boxplot(x="Species", y="PetalLengthCm", data=iris)
ax= sns.stripplot(x="Species", y="PetalLengthCm", data=iris, jitter=True,


plt.show()
```



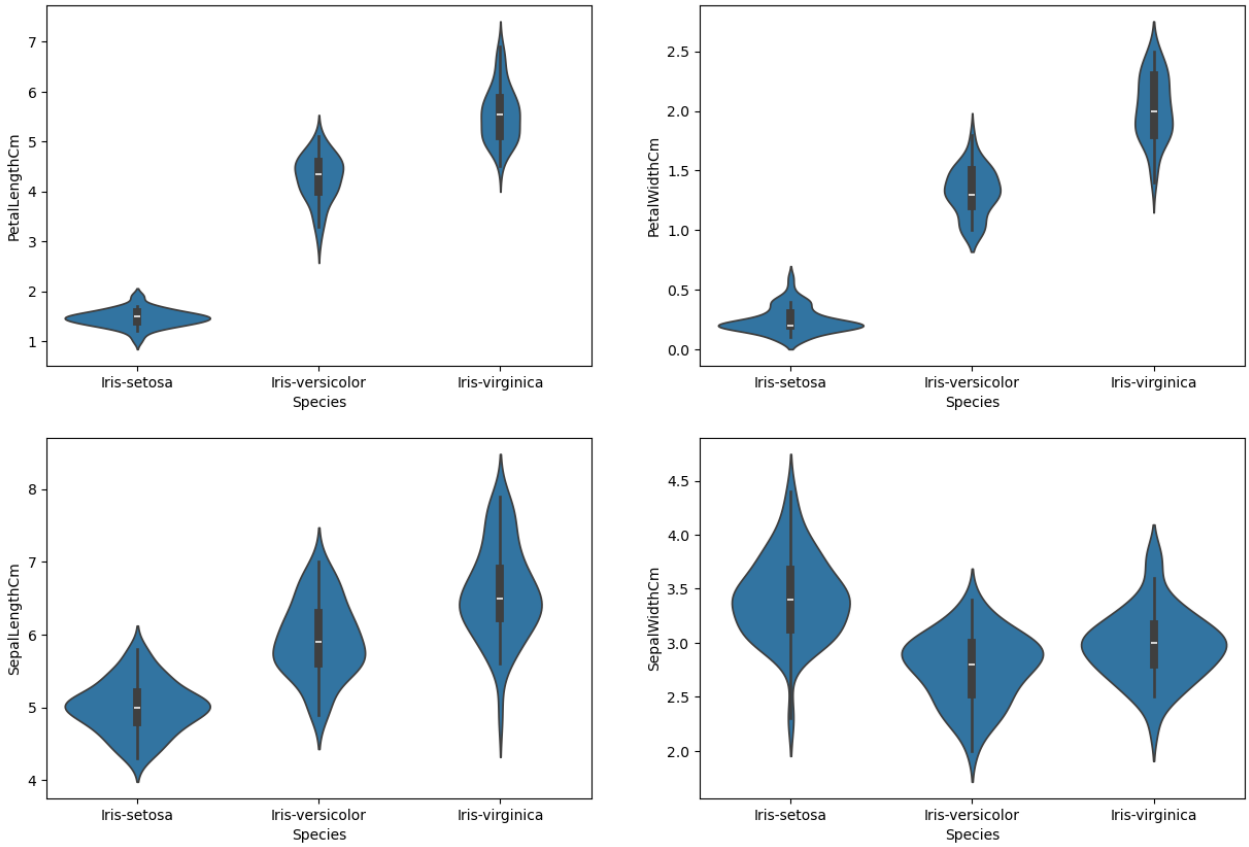
In [25]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [26]:
```
fig = plt.gcf()
fig.set_size_inches(10,7)
fig = sns.violinplot(x = 'Species' , y = 'SepalLengthCm' , data = iris)
```

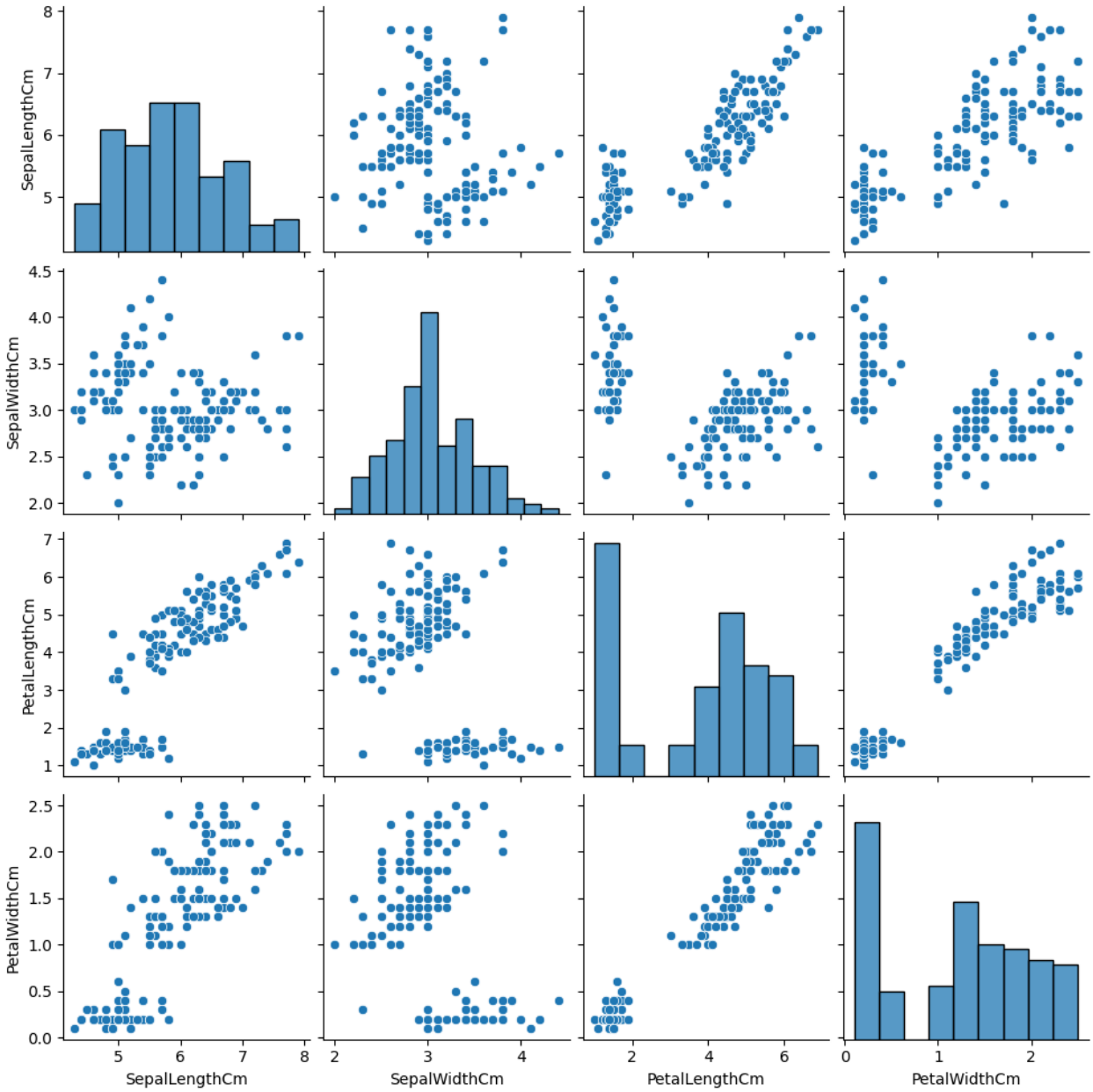


In [27]:
```
plt.figure(figsize = (15,10))
plt.subplot(2,2,1)
sns.violinplot(x = 'Species' , y = 'PetalLengthCm', data = iris)
plt.subplot(2,2,2)
sns.violinplot(x = 'Species' , y = 'PetalWidthCm', data = iris)
plt.subplot(2,2,3)
sns.violinplot(x = 'Species' , y = 'SepalLengthCm', data = iris)
plt.subplot(2,2,4)
sns.violinplot(x = 'Species' , y = 'SepalWidthCm', data = iris)
```

Out[27]: <Axes: xlabel='Species', ylabel='SepalWidthCm'>

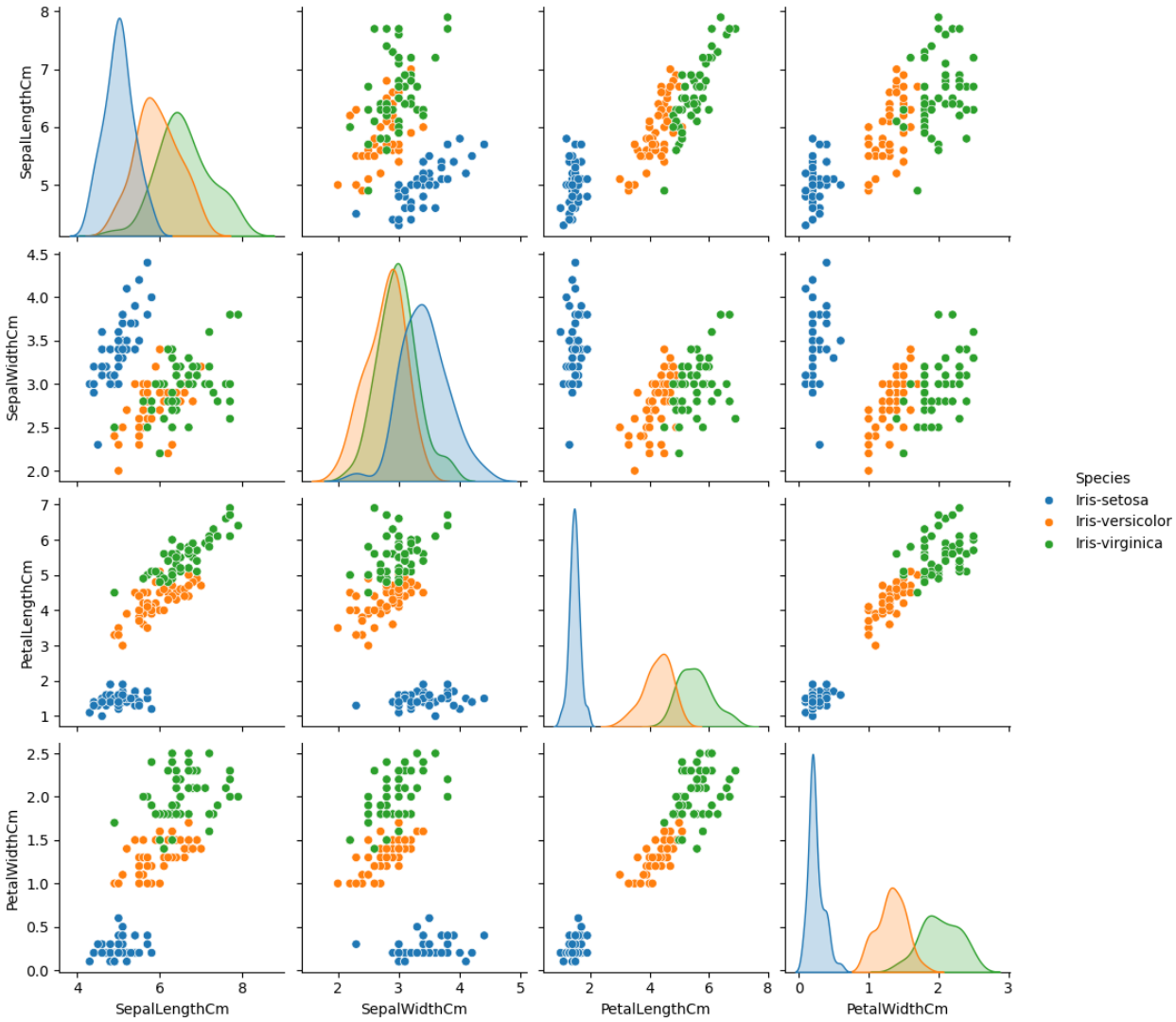


In [28]:
```
# pairplot

l1 = sns.pairplot(data = iris, kind = 'scatter')
```



In [29]:
```
sns.pairplot(data = iris , hue = 'Species')
```

Out[29]:
```
<seaborn.axisgrid.PairGrid at 0x1308a0a10>
```



In [30]:
```
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 [31]:
```
iris1 = iris.copy()
```

In [32]:
```
iris1
```

Out[32]:

	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 [33]:
```
iris1.drop('Species',axis=1, inplace =True)
```

In [34]:
```
iris
```

Out[34]:

	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

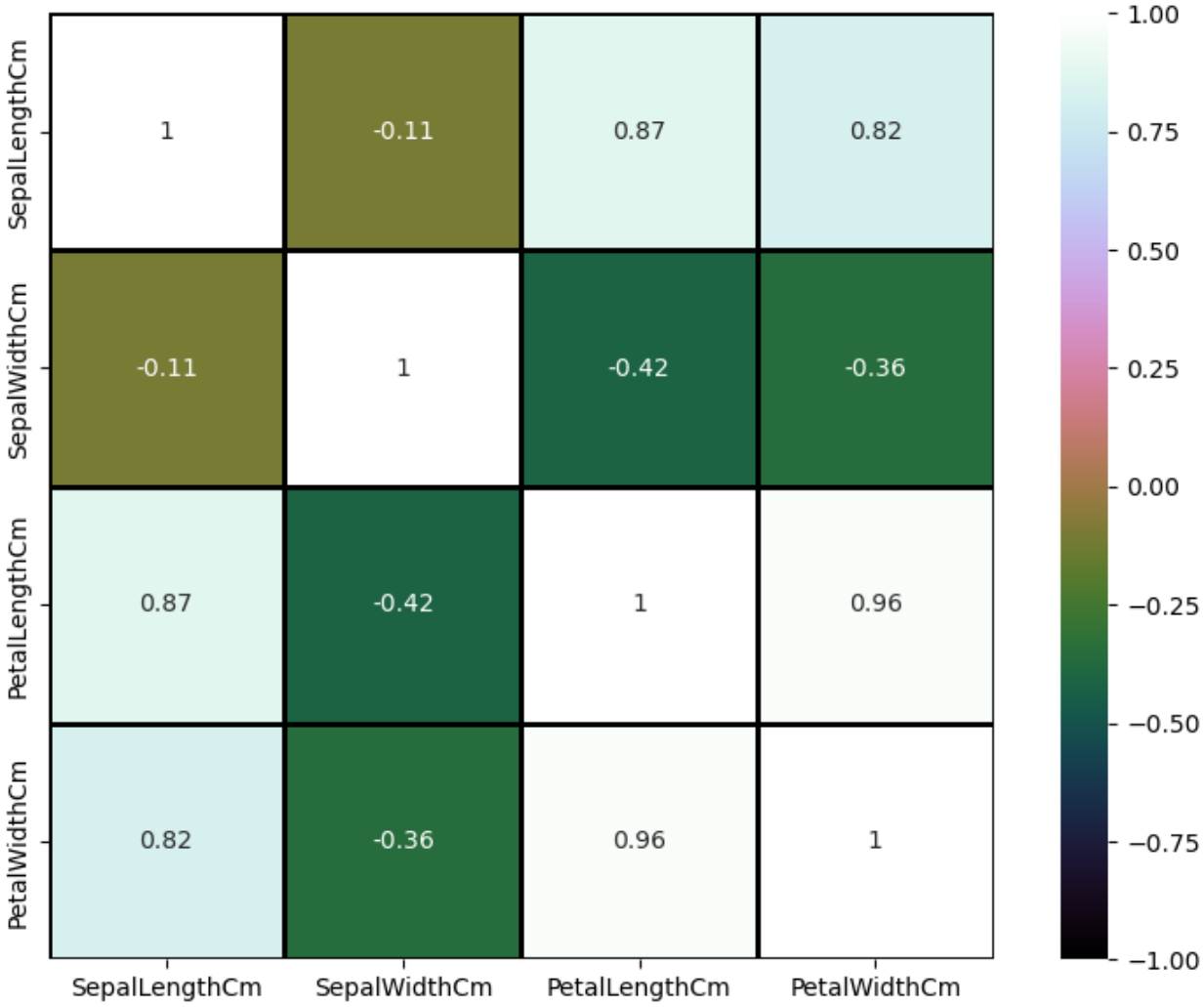
In [35]:
```
iris1
```

Out[35]:

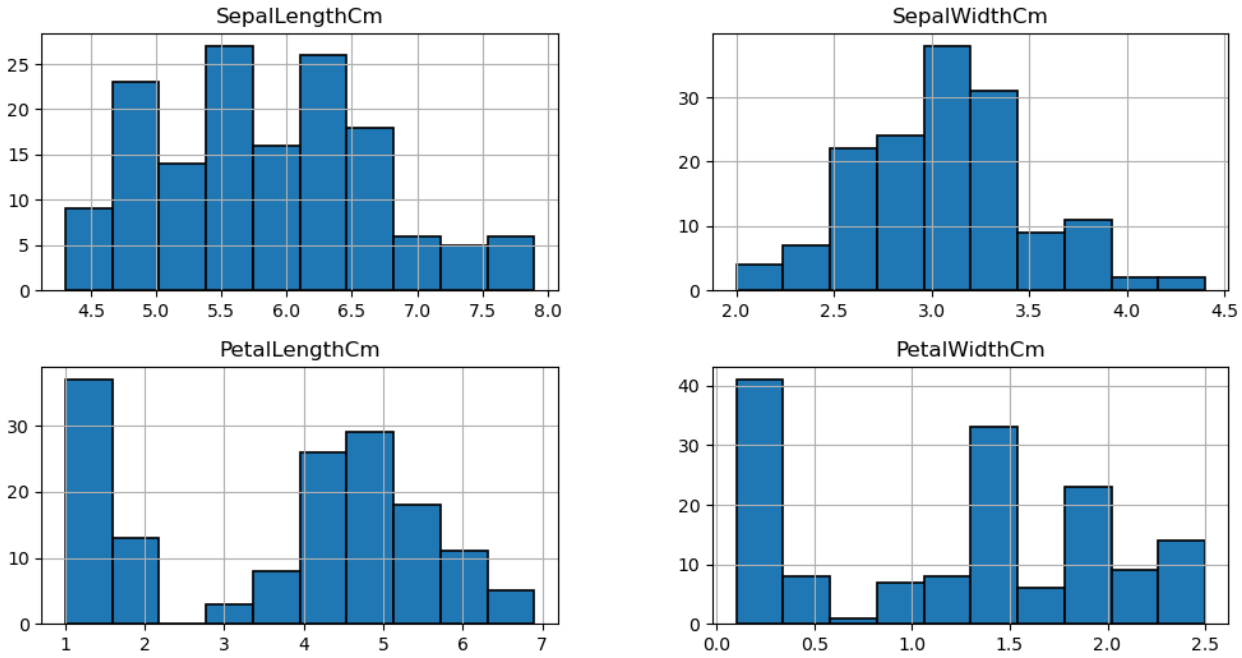
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

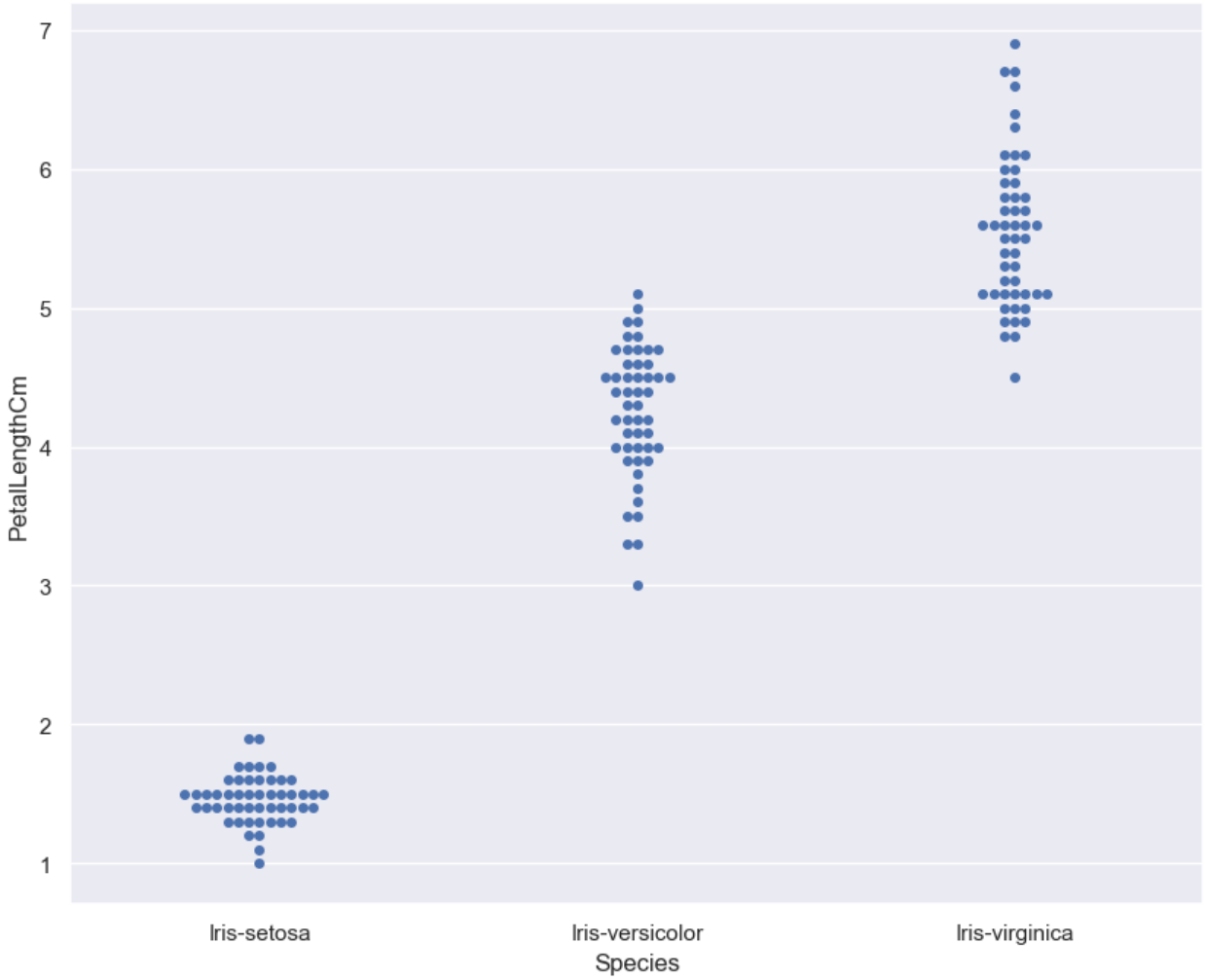
In [36]:
```
fig = plt.gcf()
fig.set_size_inches(10,7)
fig = sns.heatmap(iris1.corr(), annot= True, cmap = 'cubehelix' , linewid
```



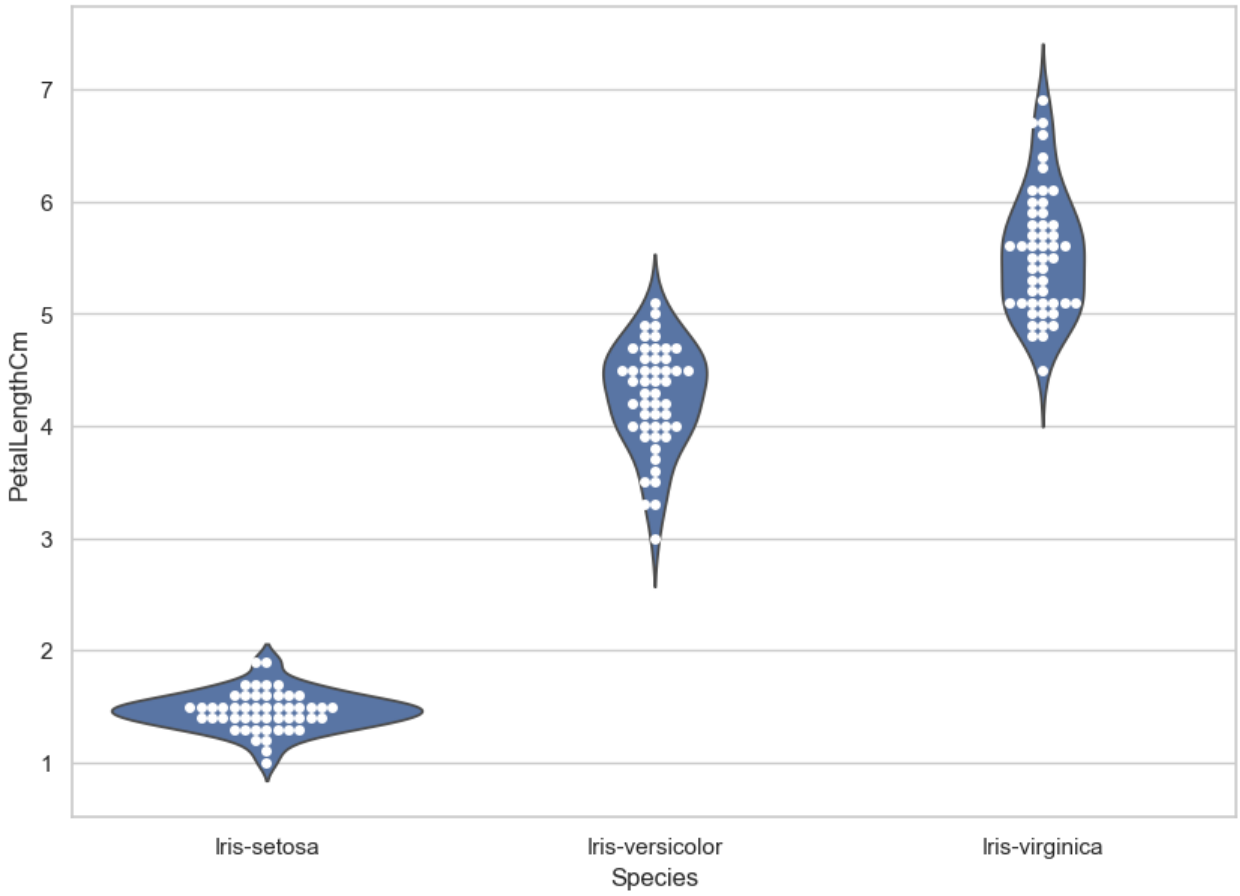
In [37]:
```
# Distribution PLot

iris.hist(edgecolor = 'k', linewidth = 1.2)
fig = plt.gcf()
fig.set_size_inches(12,6)
```



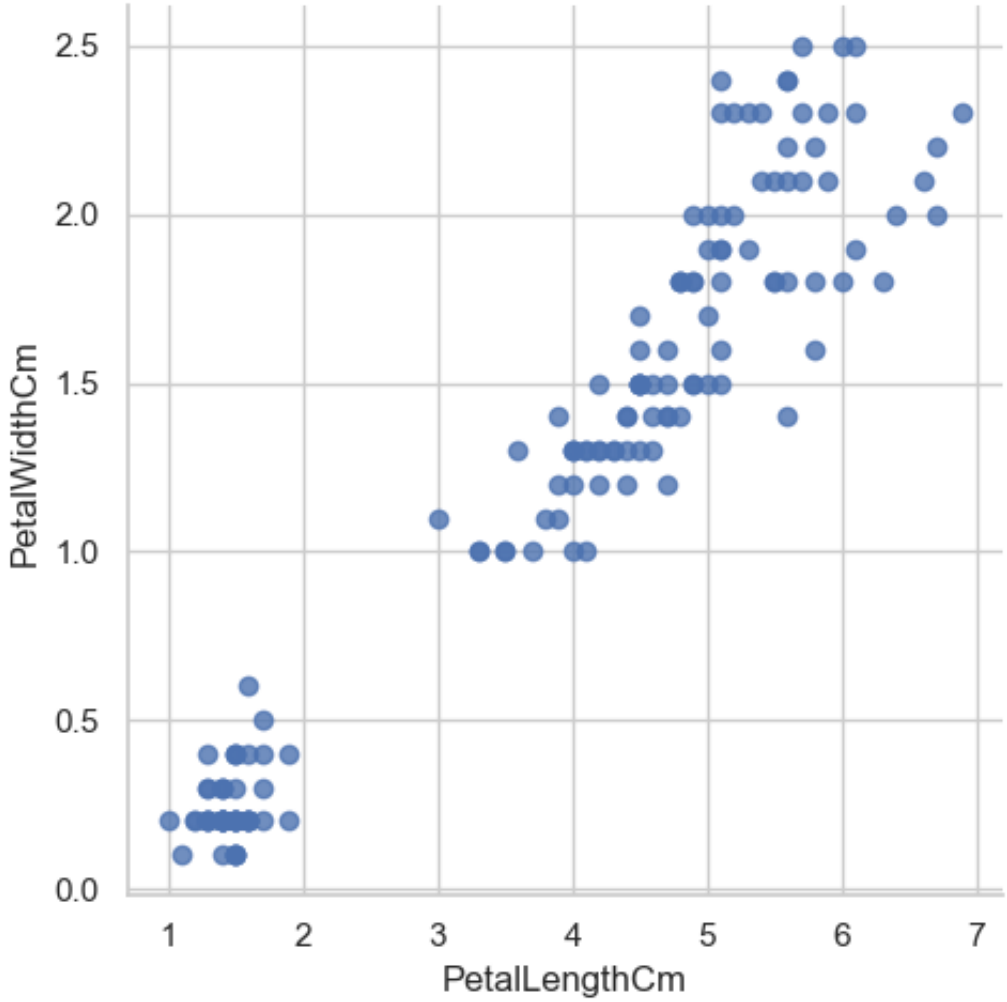
In [38]:
```
# Swarm plot

sns.set(style = 'darkgrid')
fig = plt.gcf()
fig.set_size_inches(10,8)
fig = sns.swarmplot(x= 'Species' , y = 'PetalLengthCm' , data = iris)
```



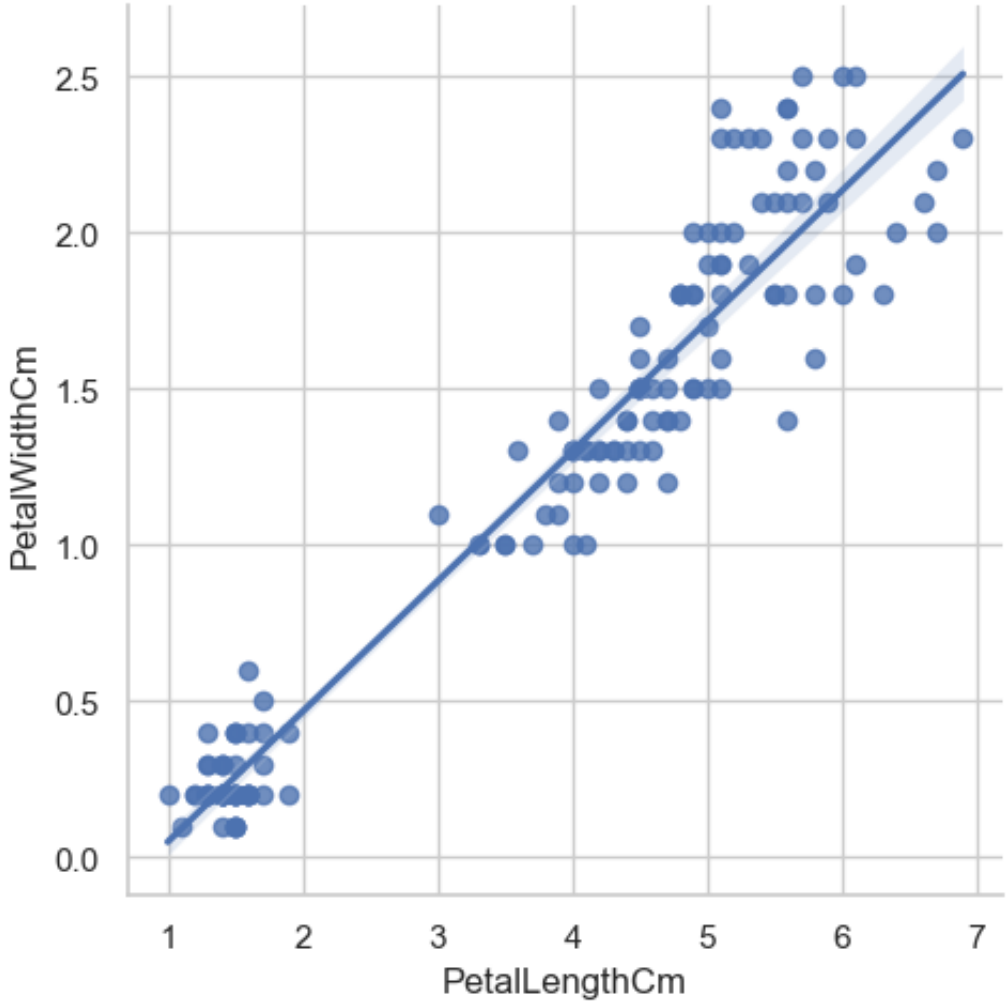
In [39]:
```
sns.set(style= 'whitegrid')
fig = plt.gcf()
fig.set_size_inches(10,7)
ax = sns.violinplot(x = 'Species' , y= 'PetalLengthCm' ,  data = iris, in
ax = sns.swarmplot(x = 'Species' , y= 'PetalLengthCm' ,  data = iris, col
```



In [40]:
```
fig=sns.lmplot(x="PetalLengthCm", y="PetalWidthCm",data=iris, fit_reg = F
```

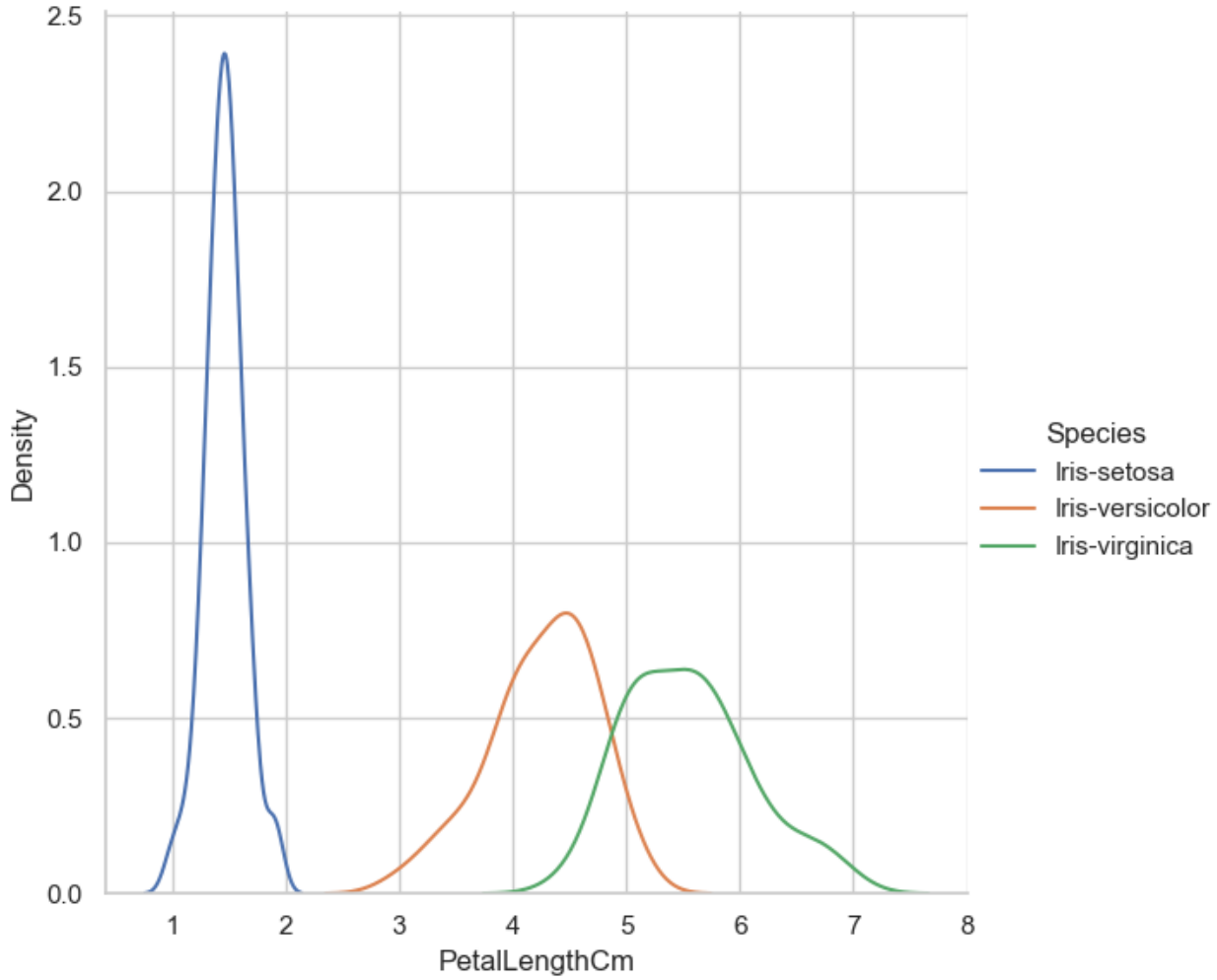


In [41]:
```
fig=sns.lmplot(x="PetalLengthCm", y="PetalWidthCm",data=iris)
```

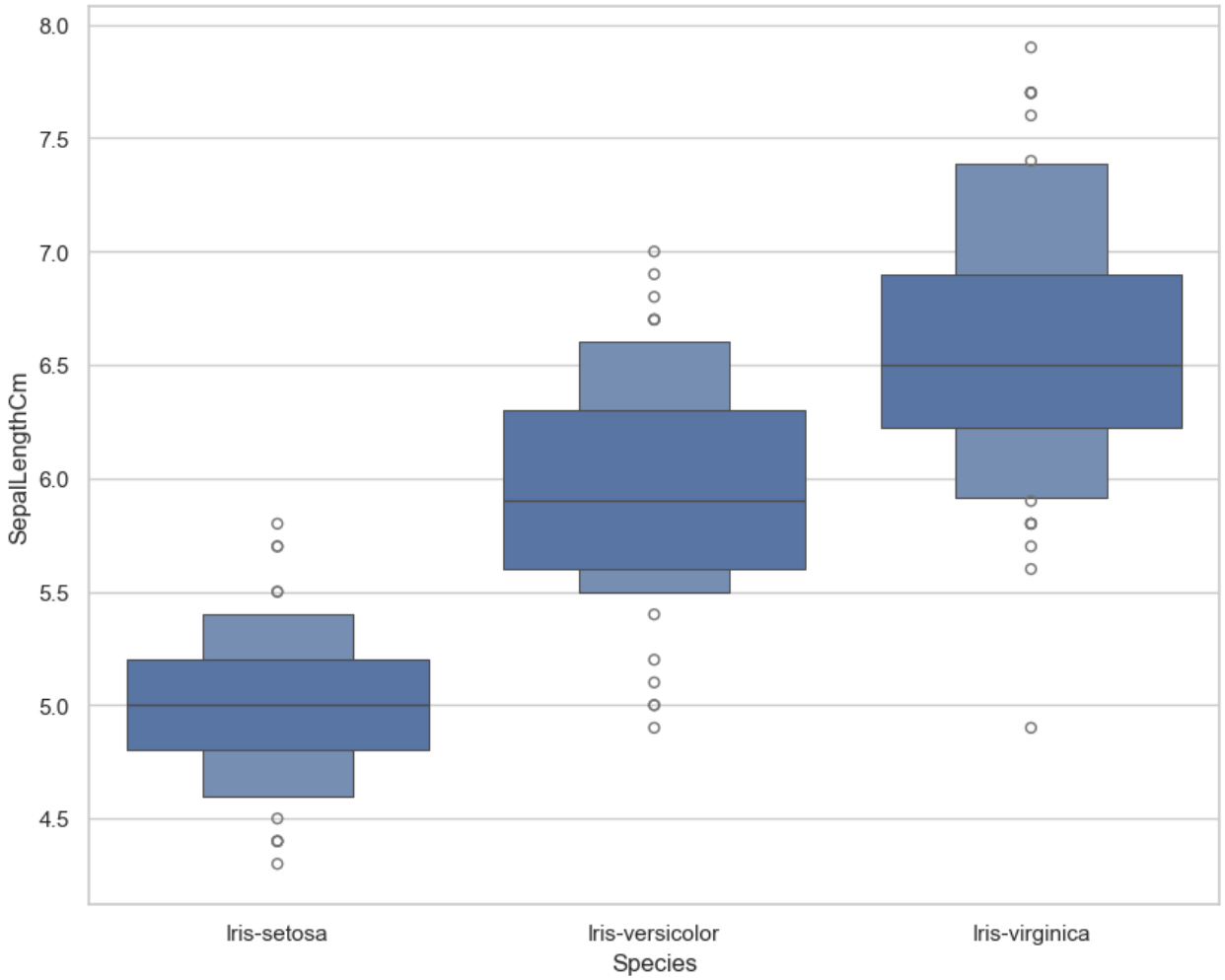


In [42]:
```
# FacetGrid

sns.FacetGrid(iris, hue = 'Species' , height = 6)\
    .map(sns.kdeplot, 'PetalLengthCm')\
    .add_legend()
plt.ioff()
```

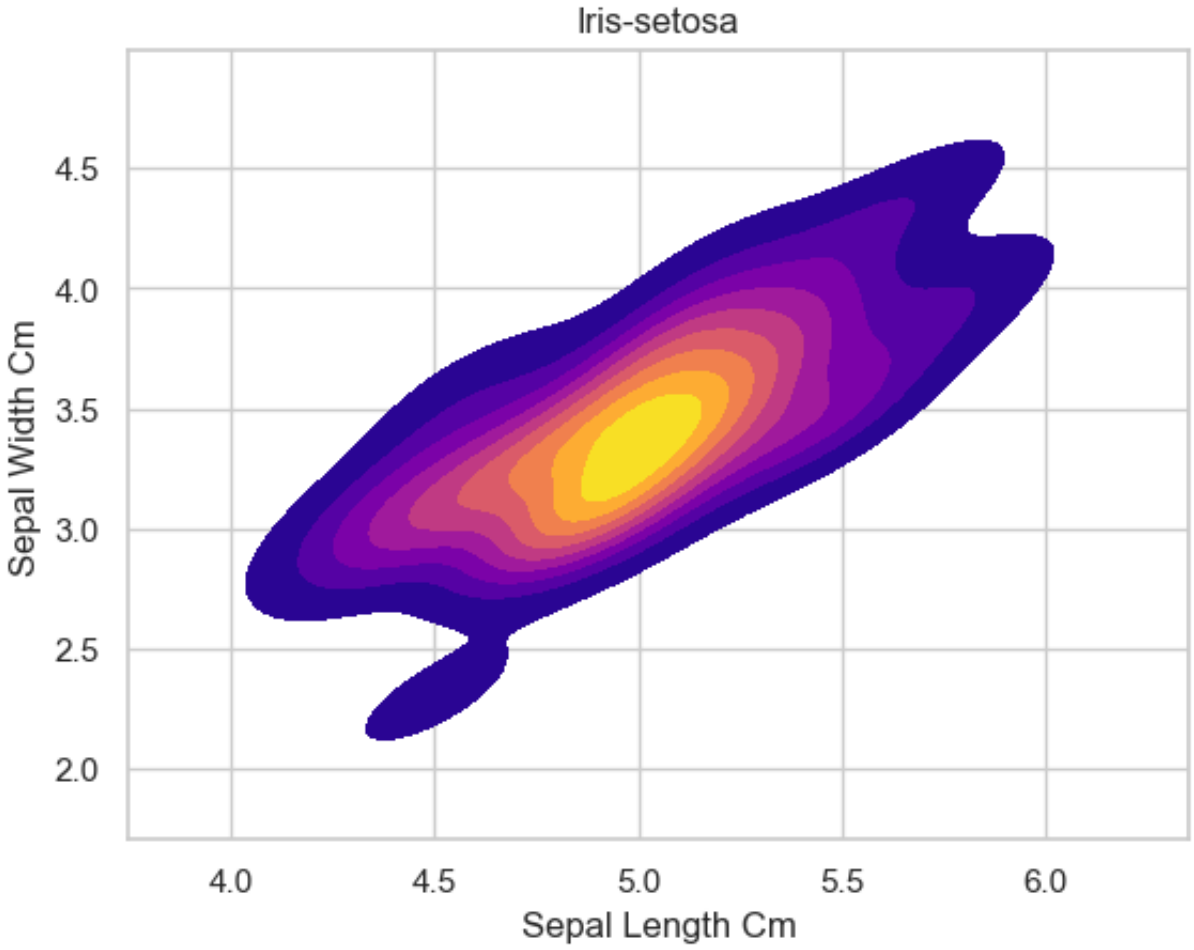
Out[42]: <contextlib.ExitStack at 0x1317567b0>



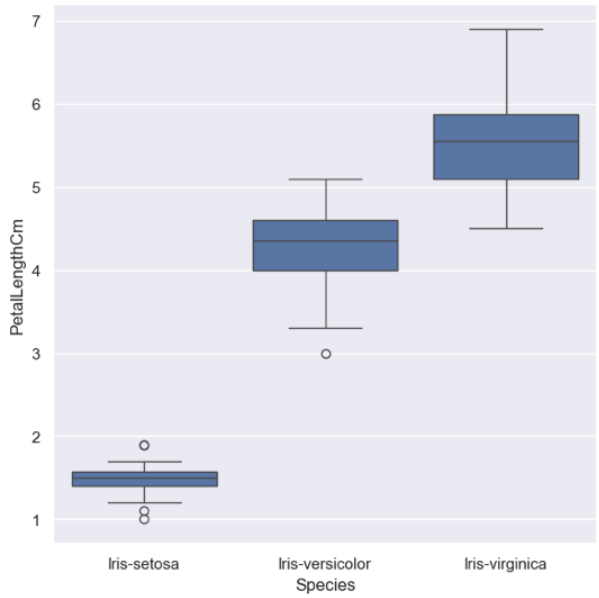
In [43]:
```
fig = plt.gcf()
fig.set_size_inches(10,8)
fig = sns.boxenplot(x = 'Species' , y = 'SepalLengthCm' , data = iris,)
plt.show()
```

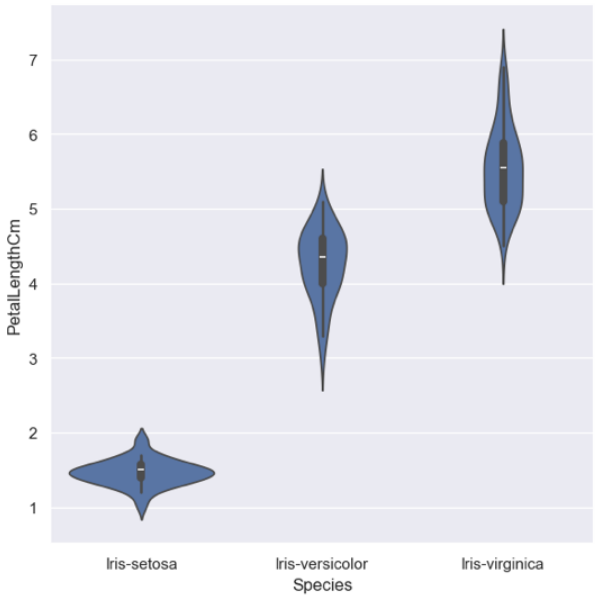


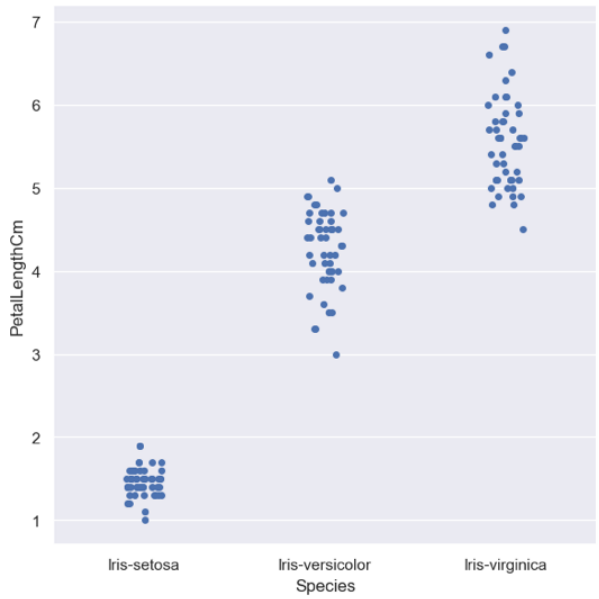
In [44]:
```
sub = iris[iris['Species'] == 'Iris-setosa']
sns.kdeplot(data=sub , x = 'SepalLengthCm', y=  'SepalWidthCm', cmap = 'p
plt.title('Iris-setosa')
plt.xlabel('Sepal Length Cm')
plt.ylabel('Sepal Width Cm')
plt.show()
```

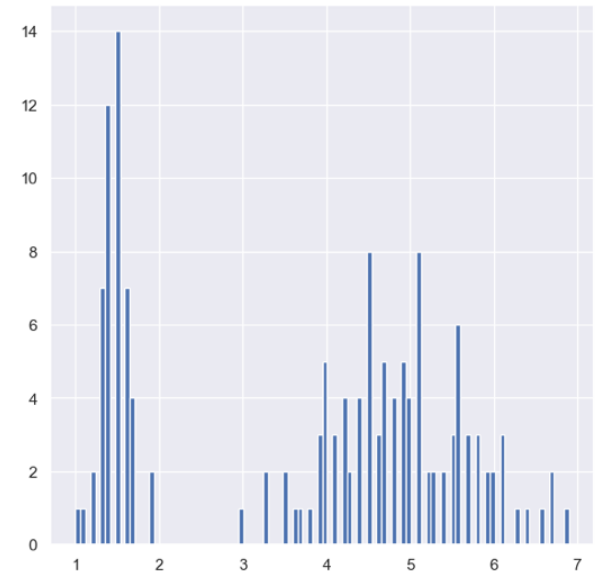


In [45]:
```
sns.set_style('darkgrid')
f,axes = plt.subplots(2,2 , figsize = (15,15))
k1 = sns.boxplot(x = 'Species', y = 'PetalLengthCm', data = iris, ax=axes
k2 = sns.violinplot(x = 'Species', y = 'PetalLengthCm', data = iris, ax=a
k3 = sns.stripplot(x = 'Species', y = 'PetalLengthCm', data = iris, jitte
axes[1,1].hist(iris.PetalLengthCm, bins=100)
plt.show()
```









In [46]:
```
# stacked Histogram


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

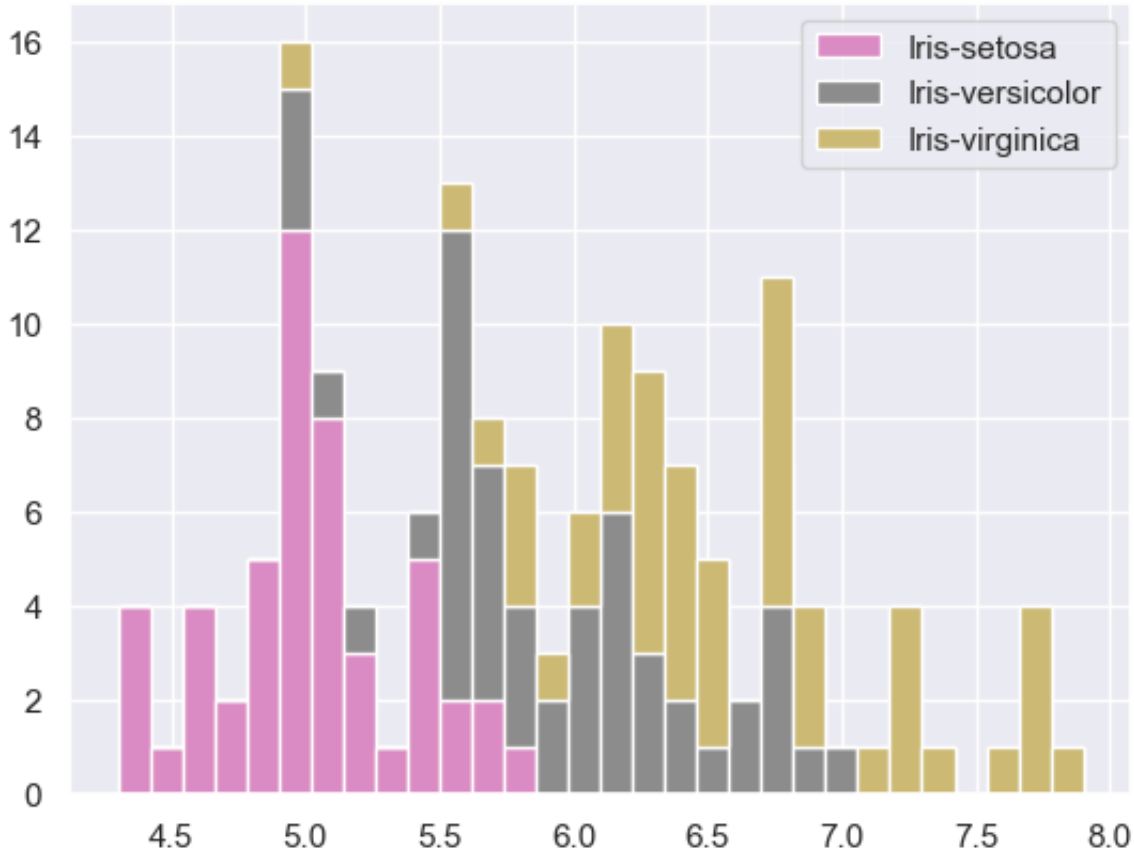
Out[46]:

	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 [93]:
```
iris.info()
```

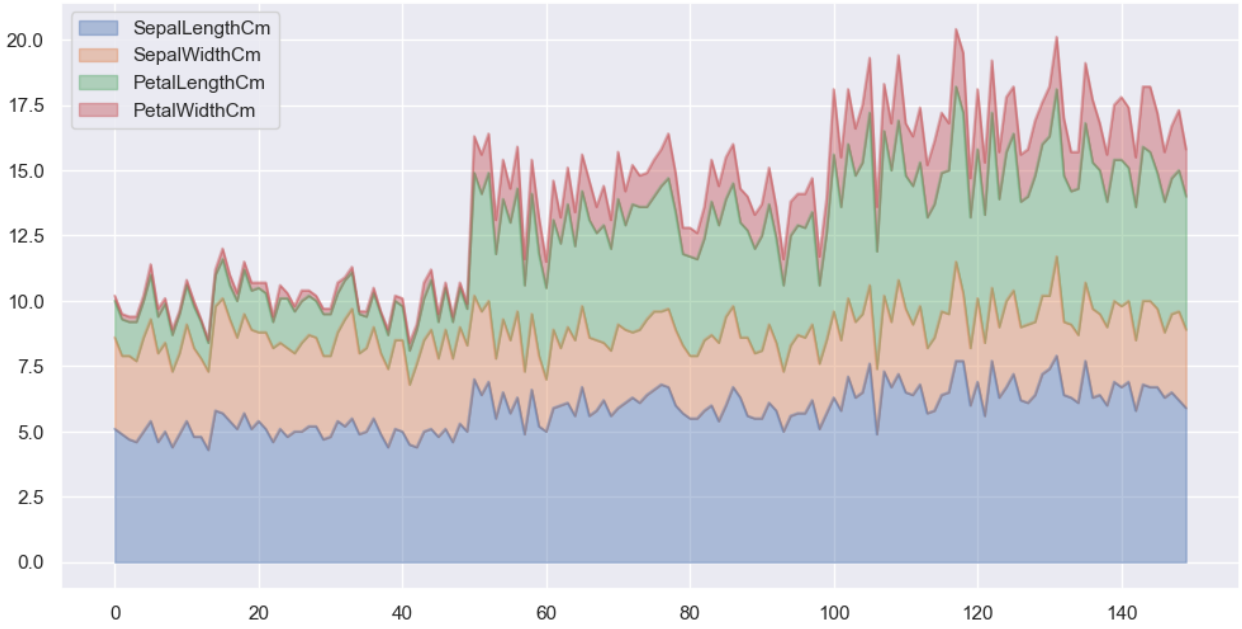
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype   
---  ------         --------------  -----   
 0   SepalLengthCm  150 non-null    float64 
 1   SepalWidthCm   150 non-null    float64 
 2   PetalLengthCm  150 non-null    float64 
 3   PetalWidthCm   150 non-null    float64 
 4   Species        150 non-null    category
dtypes: category(1), float64(4)
memory usage: 5.1 KB
```

In [99]:
```
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 [101…
```
# Area Plot

iris.plot.area(y = ['SepalLengthCm','SepalWidthCm','PetalLengthCm','Petal
```

In [103…
```
plt.show()
```



In [ ]: