In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
idata=pd.read_csv(r'S:\DOCS\Sep\Sep_7\IRIS DATASET _ ADVANCE VISUALIZATION _ EDA 2\Iris.csv')
```

In [3]:
```
idata
```

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]:
```
idata.shape
```

Out[4]:
```
(150, 6)
```

In [5]:
```
idata.head()
```

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

In [6]:
```
idata.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
```

In [7]:
```
idata['Species'].unique()
```

Out[7]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [8]:
```
1 idata.Species.value_counts()
```

Out[8]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [9]:
```
1 # there are threee species of plant
```

In [10]:
```
1 idata.isnull().any().any()
```

Out[10]:
```
False
```

In [11]:
```
1 # no null value in dataset
```

In [12]:
```
1 idata.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
```

In [13]:
```
1 # Species is categorical variable type,so convert from object to categorical datatype
```

In [14]:
```
1 idata['Species'].astype('category')
```

Out[14]:
```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
            ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: category
Categories (3, object): ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
```

In [15]:
```
1 idata.head()
```

Out[15]:

	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 [16]:
```
1 # there is no need of id column as it is not useful in our analysis
```

In [17]:
```
1 idata.drop('Id', axis=1, inplace=True)
```

In [18]:
```
1 idata.head()
```

Out[18]:

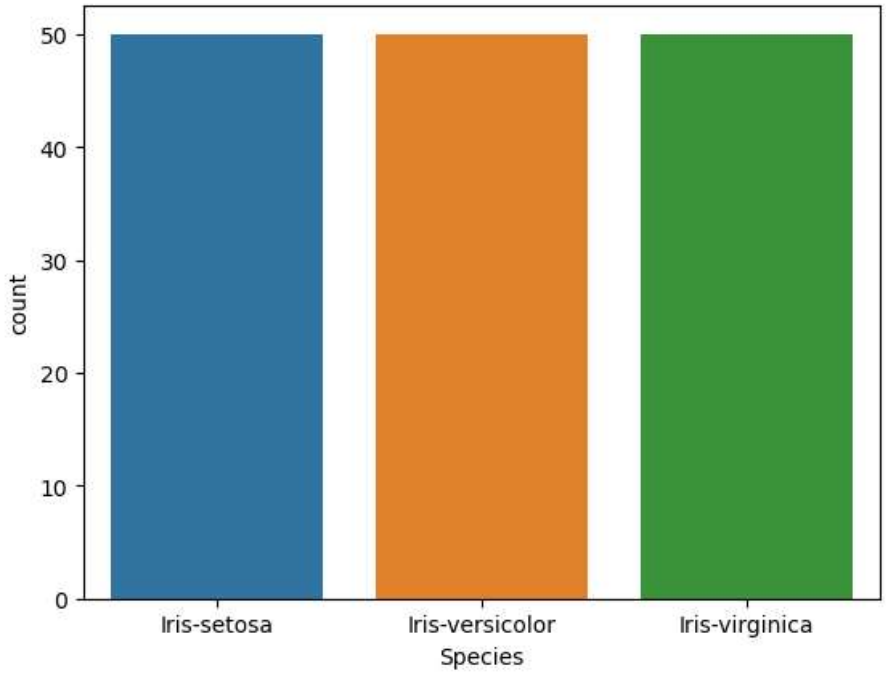
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [19]:
```
1 # Species is dependent variable and others are independent variable.
```

In [20]:
```
1 # Univariate analysis
```

In [21]:
```
1
2 sns.countplot(x=idata['Species'])
```

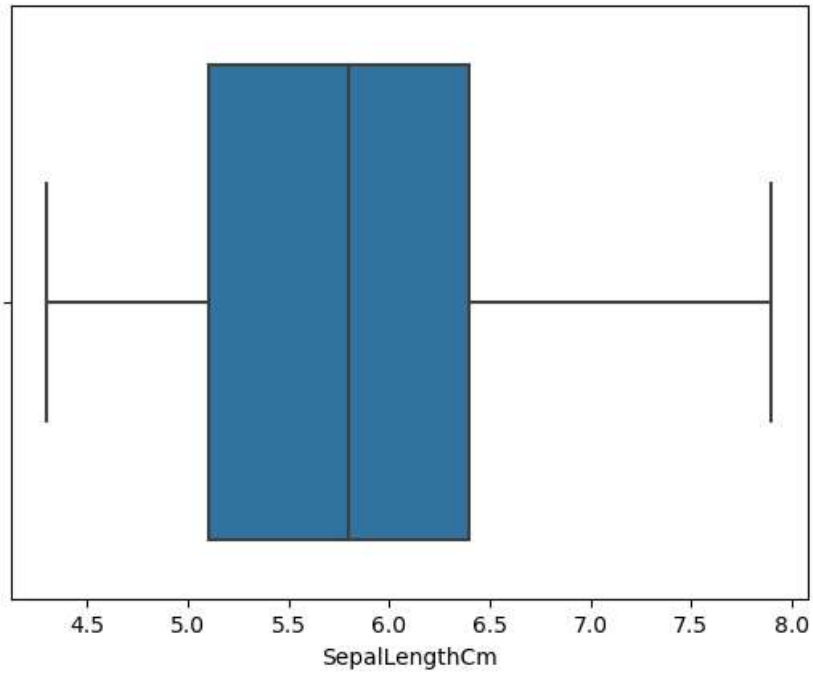
Out[21]: <Axes: xlabel='Species', ylabel='count'>



In [22]:
```
1 # counts of eeach species is same
```

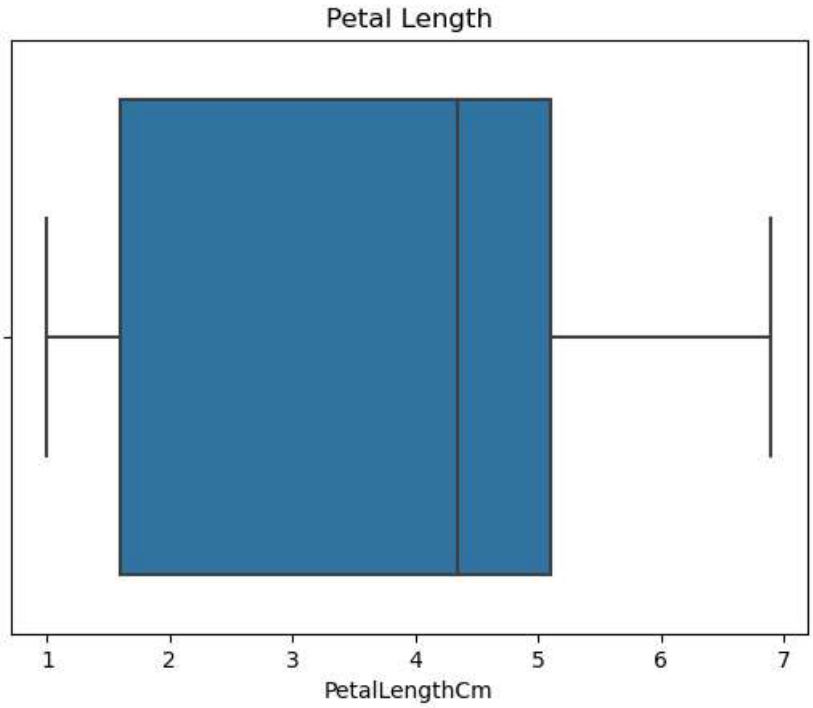
In [23]:
```
1 sns.boxplot(data=idata, x='SepalLengthCm', hue='Species')
```

Out[23]: <Axes: xlabel='SepalLengthCm'>



In [24]:
```
1 sns.boxplot(data=idata, x='PetalLengthCm')
2 plt.title('Petal Length')
```

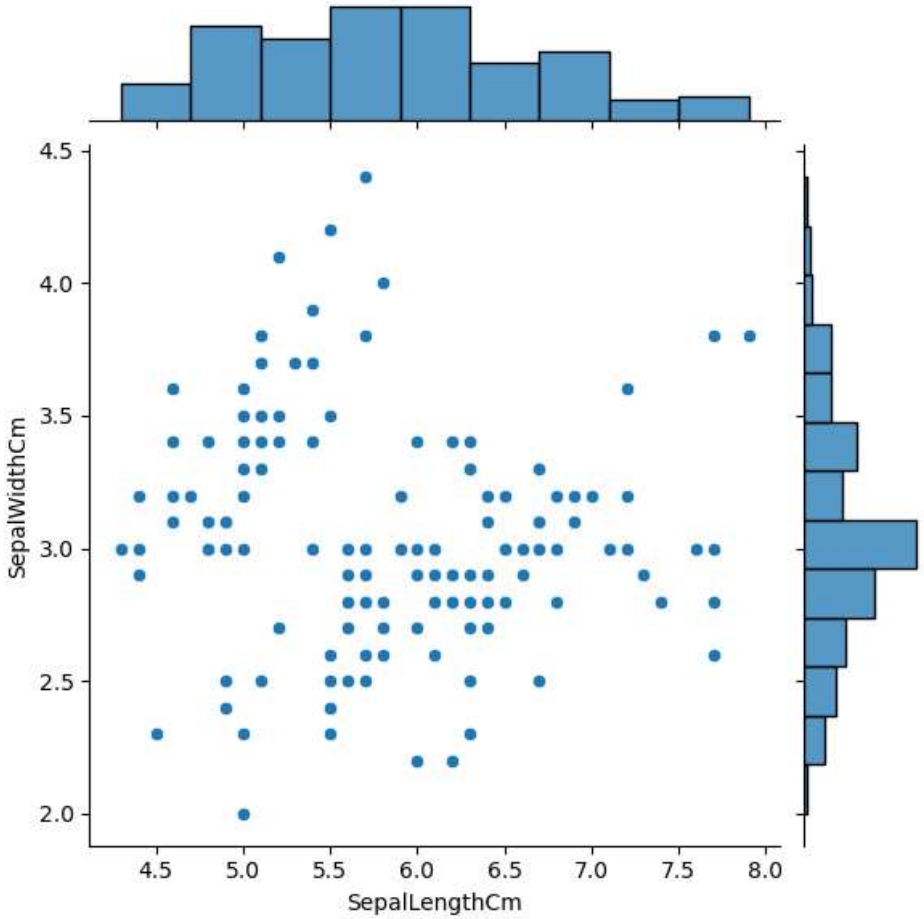
Out[24]: Text(0.5, 1.0, 'Petal Length')



In [25]:
```
1 # bi variate analysis
2 # jointplot
```

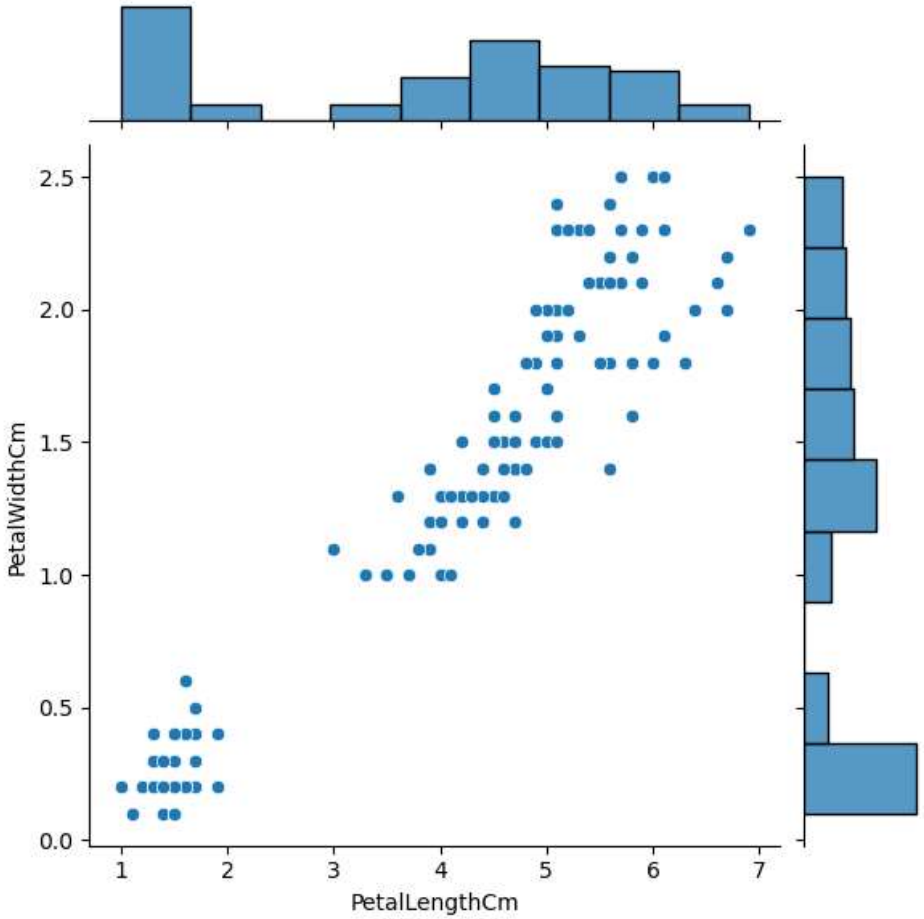
In [26]:
```
1 sns.jointplot(data=idata, x='SepalLengthCm', y='SepalWidthCm')
```

Out[26]: <seaborn.axisgrid.JointGrid at 0x20310635610>



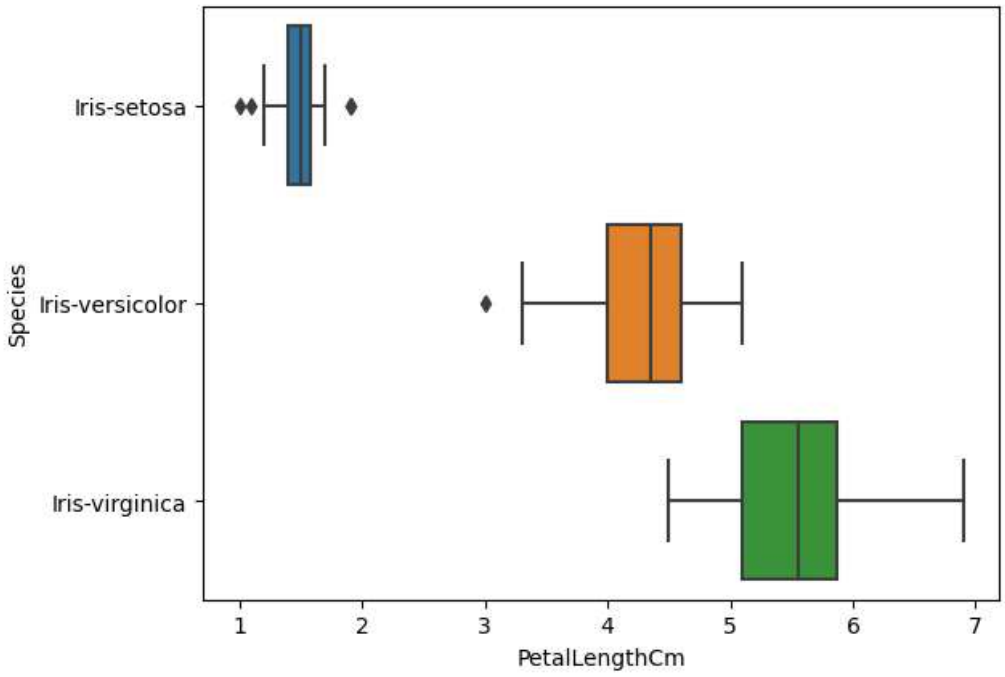
In [27]:
```
1 sns.jointplot(data=idata, x='PetalLengthCm', y='PetalWidthCm')
```

Out[27]: <seaborn.axisgrid.JointGrid at 0x203104413d0>



In [30]:
```
1 sns.boxplot(data=idata, x='PetalLengthCm',y='Species')
```

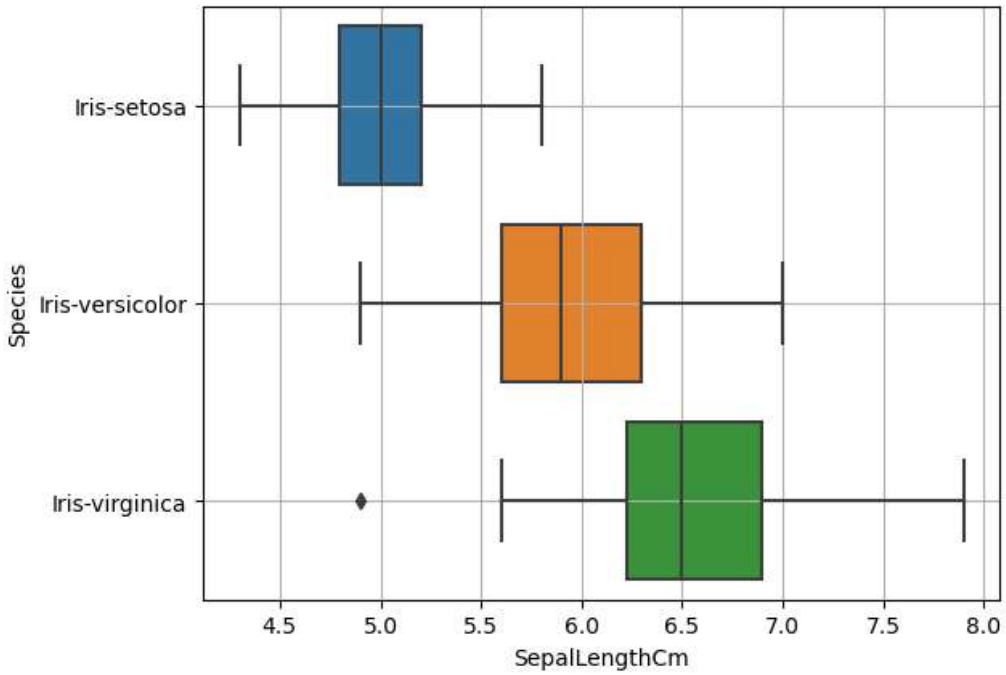
Out[30]: <Axes: xlabel='PetalLengthCm', ylabel='Species'>



In [31]:
```
# Findings
'''
* petal length of serosa plant is comparatively less around 1 to 2 cm
* petwl length of versicolor in between 3 to 5 cm
* virginica have largest petal length around 4.5 to 7 cm
'''
```

Out[31]:
```
'\n* petal length of serosa plant is comparatively less around 1 to 2 cm \n* petwl length of vers
icolor in between 3 to 5 cm\n* virginica have largest petal length around 4.5 to 7 cm \n'
```

In [43]:
```
sns.boxplot(data=idata, x='SepalLengthCm', y='Species')
plt.grid()
```



In [36]:
```
'''
iris-setosa is shortest sepal with median length of 5 cm
iris-versicolor have slightly longer sepal length with median of 5.9 cm
iris-virginica have longer sepal length with median of 6.5 cm


'''
```

Out[36]:
```
'\niris-setosa is shortest sepal with median length of 5 cm\niris-versicolor have slightly longer
sepal length with median of 5.9 cm\niris-virginica have longer sepa length with median of 6.5 cm
\n\n'
```

In [56]:
```
setosa_data=idata['Species']=='Iris-setosa'
idata[setosa_data].describe()
```

Out[56]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.00000
mean	5.00600	3.418000	1.464000	0.24400
std	0.35249	0.381024	0.173511	0.10721
min	4.30000	2.300000	1.000000	0.10000
25%	4.80000	3.125000	1.400000	0.20000
50%	5.00000	3.400000	1.500000	0.20000
75%	5.20000	3.675000	1.575000	0.30000
max	5.80000	4.400000	1.900000	0.60000

In [59]:
```
virginica_data=idata['Species'] == 'Iris-virginica'
idata[virginica_data].describe()
```

Out[59]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.00000
mean	6.58800	2.974000	5.552000	2.02600
std	0.63588	0.322497	0.551895	0.27465
min	4.90000	2.200000	4.500000	1.40000
25%	6.22500	2.800000	5.100000	1.80000
50%	6.50000	3.000000	5.550000	2.00000
75%	6.90000	3.175000	5.875000	2.30000
max	7.90000	3.800000	6.900000	2.50000

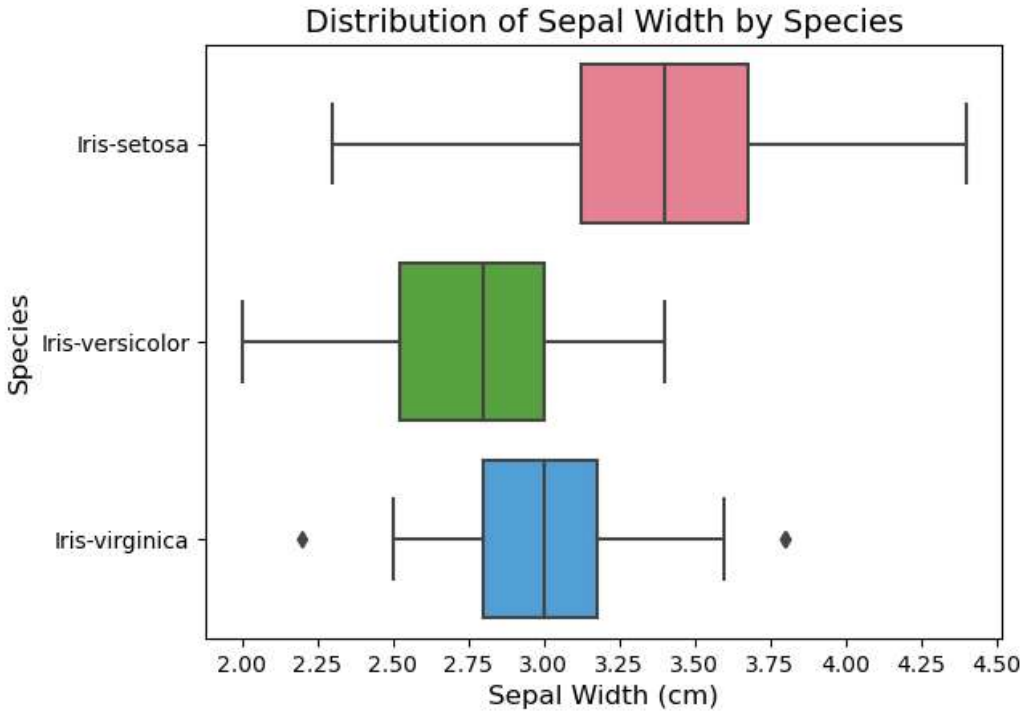
In [60]:
```
versicolor=idata['Species'] == 'Iris-versicolor'
idata[versicolor].describe()
```

Out[60]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.000000	50.000000	50.000000	50.000000
mean	5.936000	2.770000	4.260000	1.326000
std	0.516171	0.313798	0.469911	0.197753
min	4.900000	2.000000	3.000000	1.000000
25%	5.600000	2.525000	4.000000	1.200000
50%	5.900000	2.800000	4.350000	1.300000
75%	6.300000	3.000000	4.600000	1.500000
max	7.000000	3.400000	5.100000	1.800000

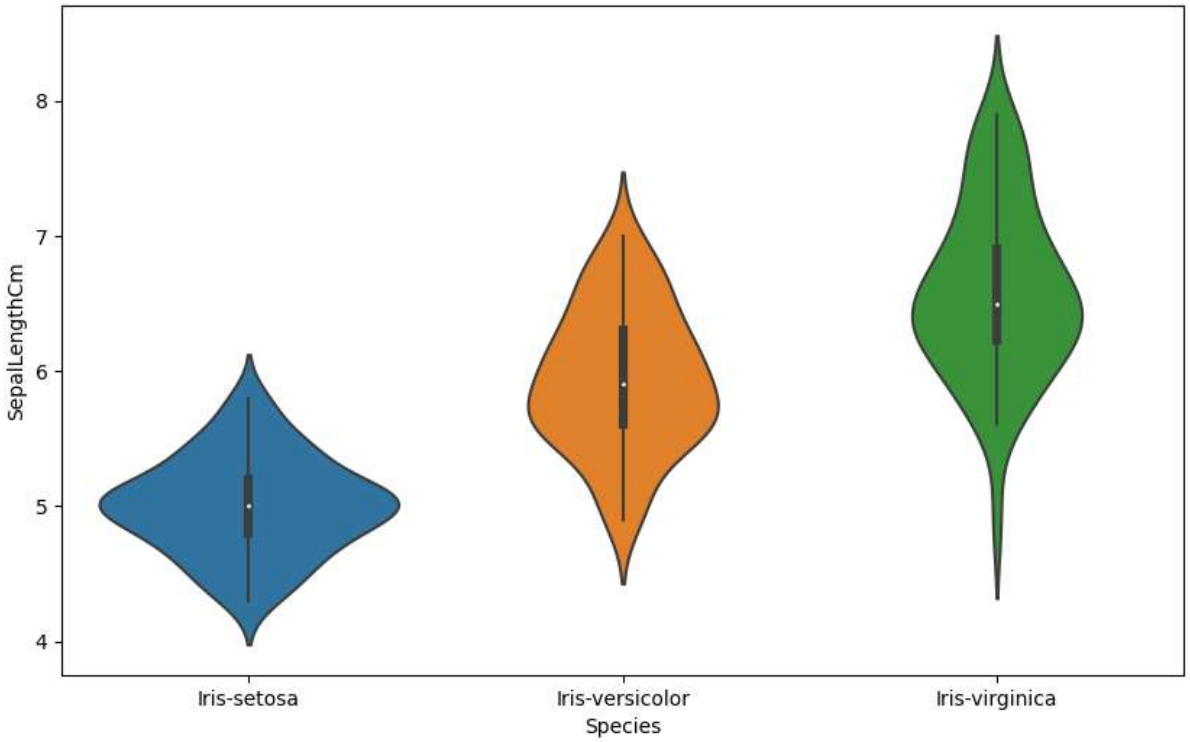
In [83]:
```
sns.boxplot(data=idata, x='SepalWidthCm', y='Species', palette="husl")
plt.xlabel('Sepal Width (cm)', fontsize=12)
plt.ylabel('Species', fontsize=12)
plt.title('Distribution of Sepal Width by Species', fontsize=14)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.locator_params(axis='x', nbins=11)

```



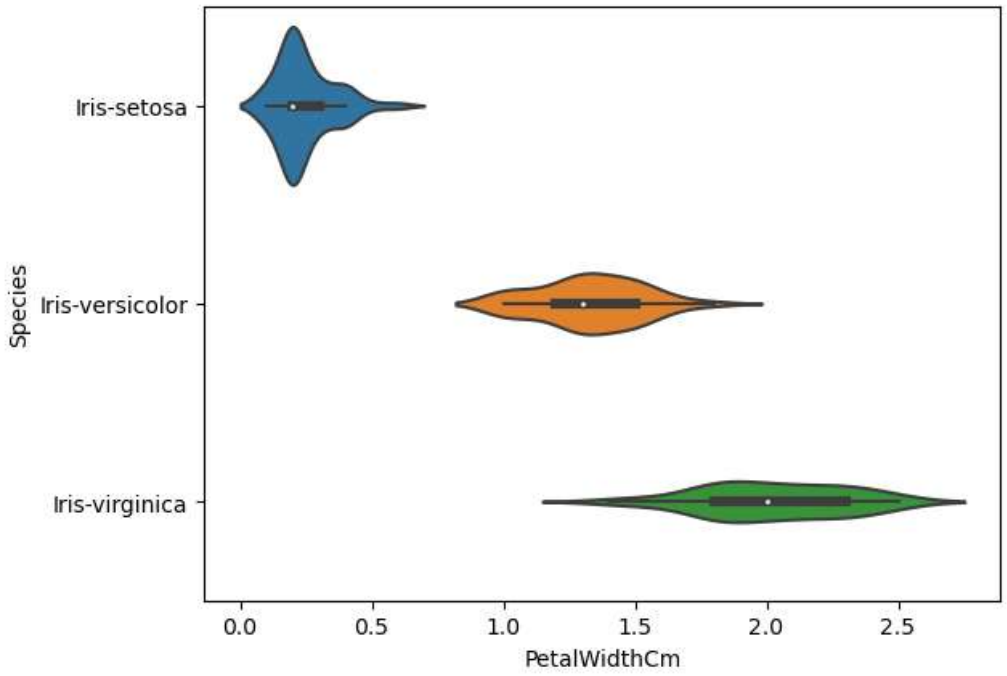
In [ ]:
```
'''
iris-setosa havethe largest sepal width with median of 3.4 cm
iris-versicolor have smallest sepal width with median of 2.8 cm
iris-virginica sepal width is slightly wider than versicolor
setosa have the widest range between 2.3 cm to 4.4 cm
'''
```

In [90]:
```
fig=plt.gcf()
fig.set_size_inches(10,6)
fig=sns.violinplot(data=idata, x='Species' ,y='SepalLengthCm')
```



In [92]:
```
sns.violinplot(data=idata, x='PetalWidthCm', y='Species')
```

Out[92]: <Axes: xlabel='PetalWidthCm', ylabel='Species'>



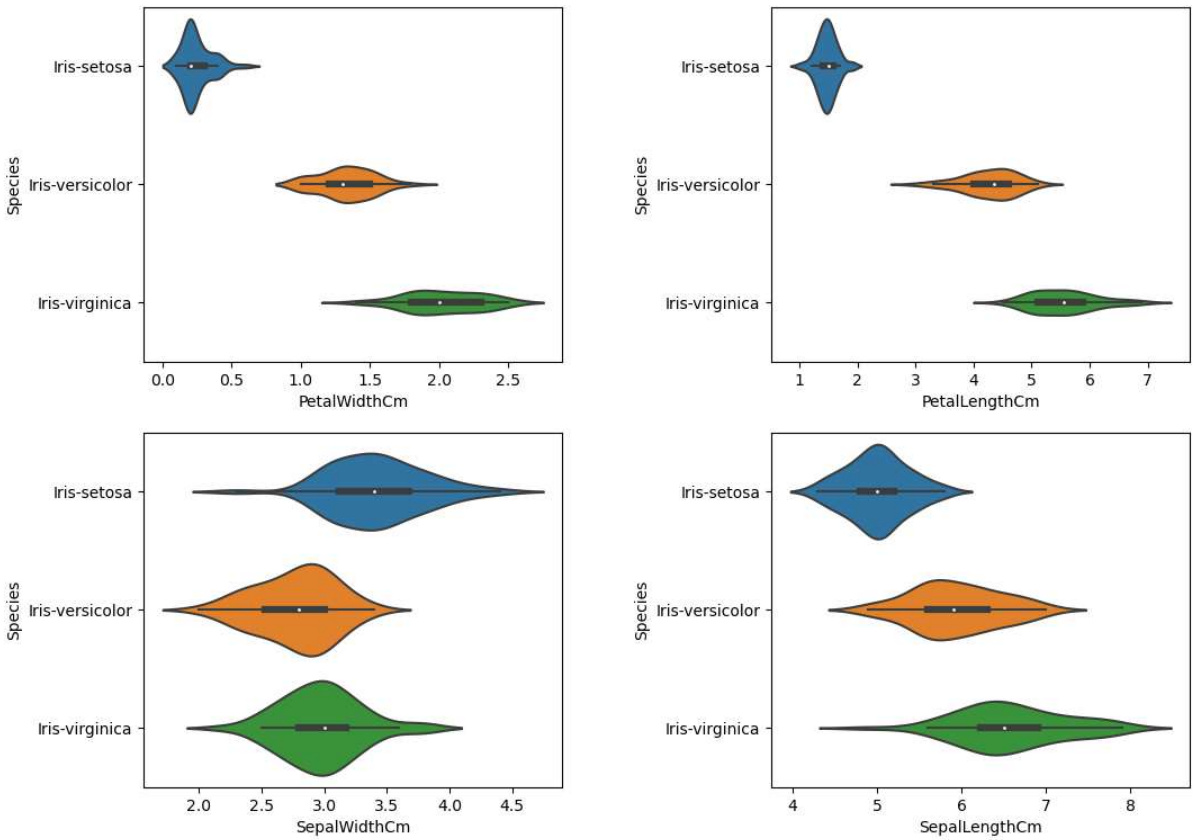
In [93]:
```
'''
setosa have smallest petal width
versicolor have larger petal width
virginica have largest petal width
petals of all species is skewed to right, means there are more data points at lower range
virginica have widest violin shape. means petal width are more spread out
'''
```

Out[93]:
```
'\nsetosa have smallest petal width \nversicolor have larger petal width\nvirginica have largest petal width\npetals of all species is skewed to right, means there are more data points at lower range\nvirginica have widest violin shape. means petal width are more spread out\n'
```

In [94]:
```
# multivariate analysis
```

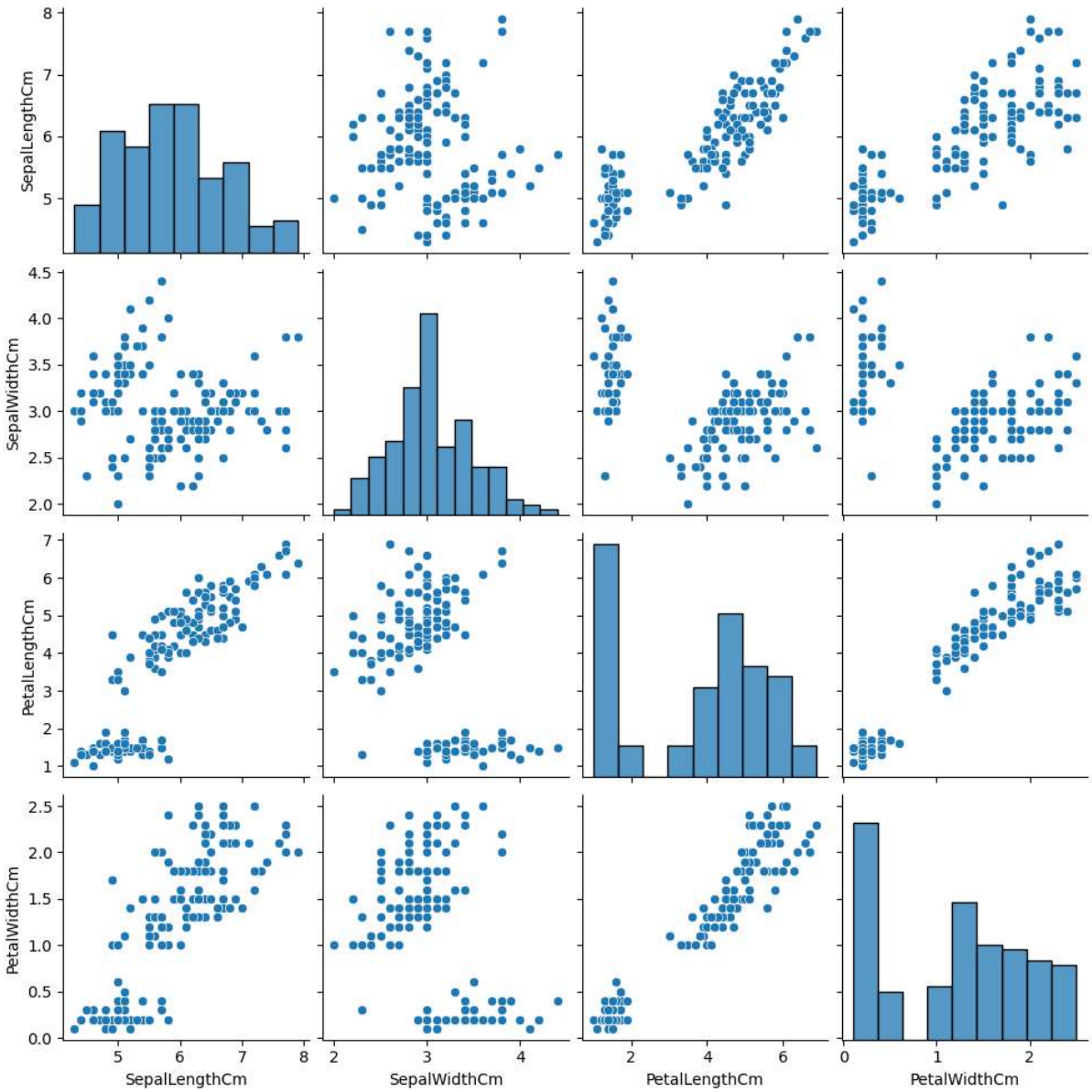
In [112]:
```
plt.figure(figsize=(12,9))
plt.subplots_adjust(wspace=0.5)
plt.subplot(2,2,1)
sns.violinplot(data=idata, x='PetalWidthCm', y='Species')
plt.subplot(2,2,2)
sns.violinplot(data=idata,x='PetalLengthCm',y='Species')
plt.subplot(2,2,3)
sns.violinplot(data=idata,x='SepalWidthCm',y='Species')
plt.subplot(2,2,4)
sns.violinplot(data=idata,x='SepalLengthCm',y='Species')
```

Out[112]:
```
<Axes: xlabel='SepalLengthCm', ylabel='Species'>
```



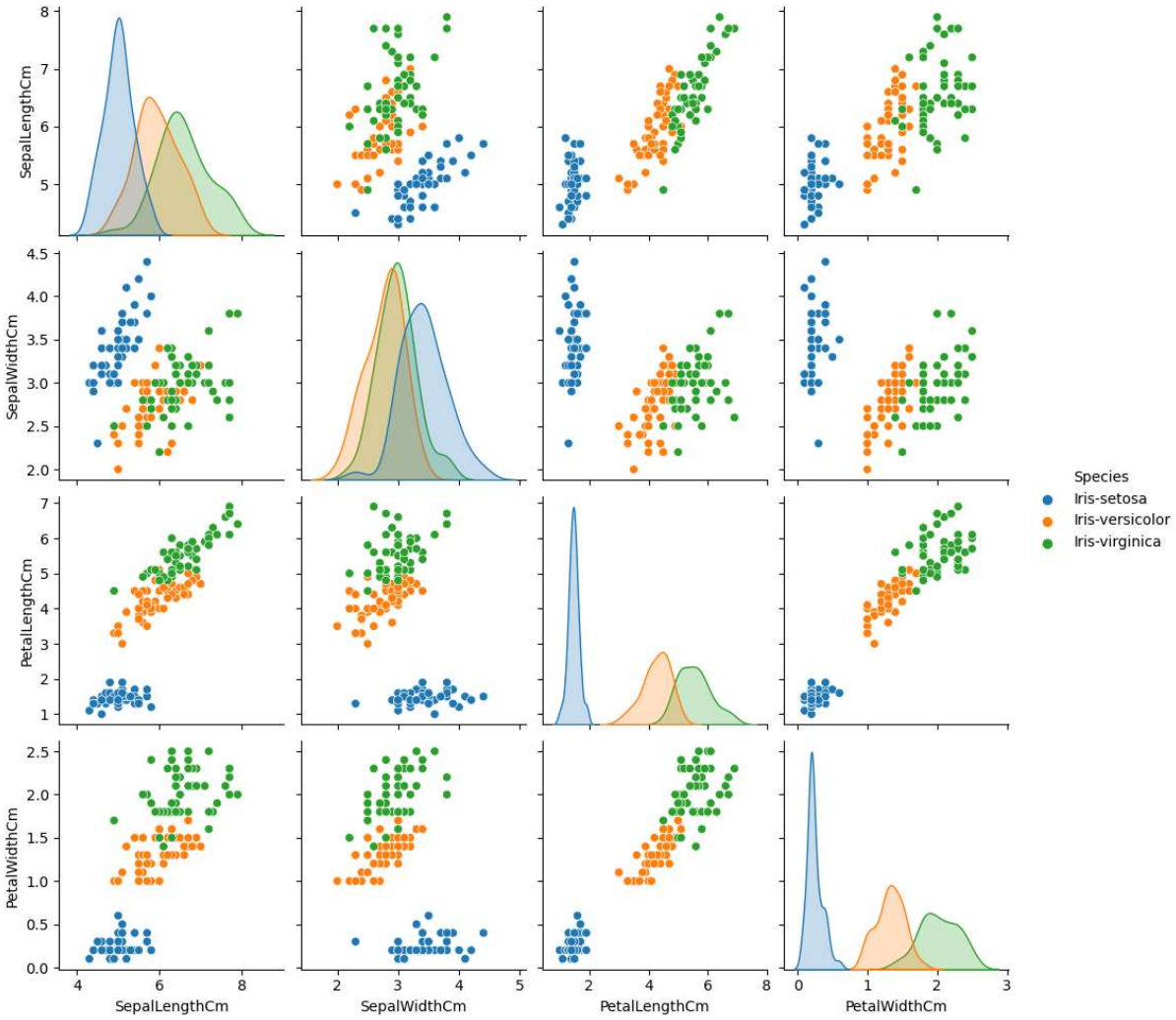
In [115]:
```
1 sns.pairplot(data=idata)
```

Out[115]: <seaborn.axisgrid.PairGrid at 0x2031bb612d0>



In [116]:
```
1 sns.pairplot(data=idata,hue='Species')
```

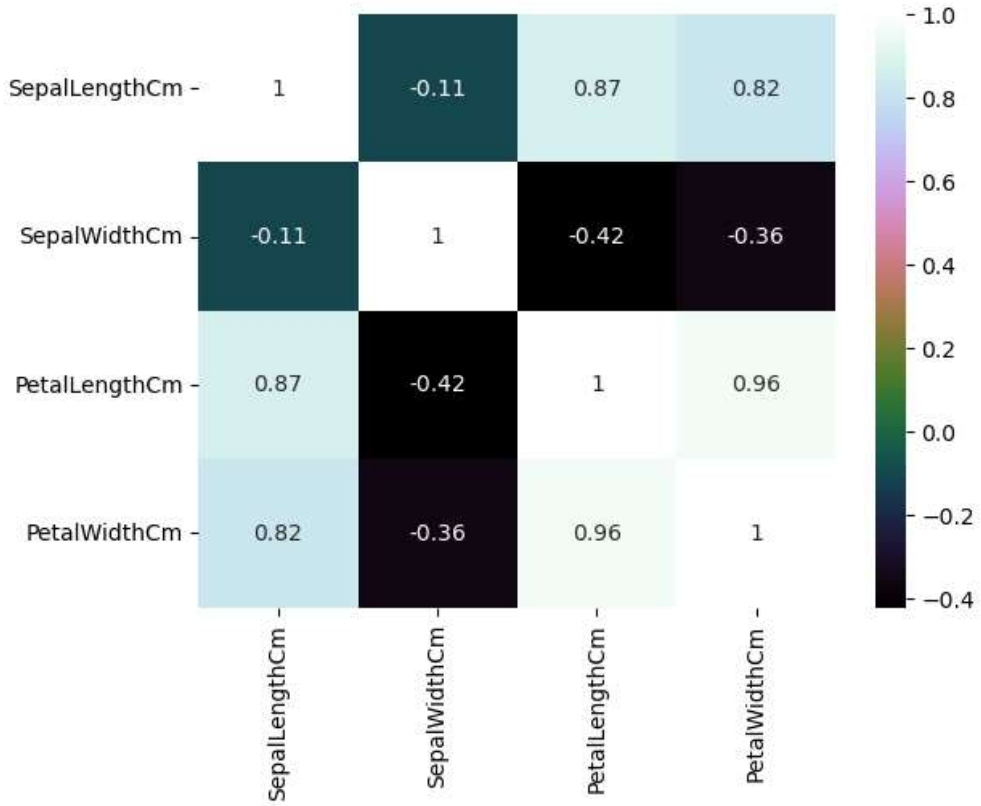
Out[116]: <seaborn.axisgrid.PairGrid at 0x2031def92d0>



In [121]:
```
1 sns.heatmap(idata.corr(),annot=True,cmap='cubehelix')
```

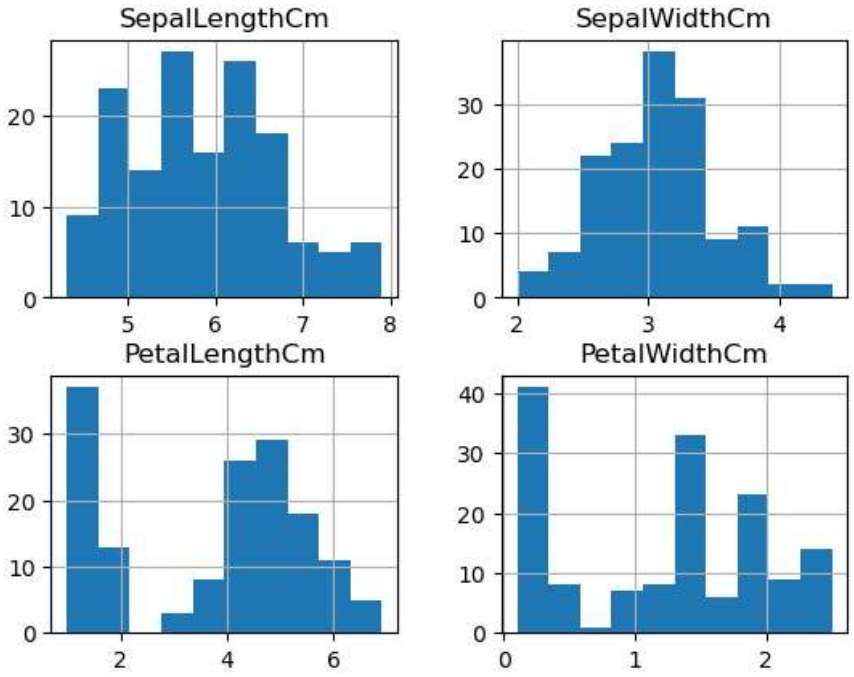
```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_10876\2409565093.py:1: FutureWarning: The default valu
e of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False.
Select only valid columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(idata.corr(),annot=True,cmap='cubehelix')
```

Out[121]: <Axes: >



In [122]:
```
1 idata.hist() # distribution plot
```

Out[122]:
```
array([[<Axes: title={'center': 'SepalLengthCm'}>,
        <Axes: title={'center': 'SepalWidthCm'}>],
       [<Axes: title={'center': 'PetalLengthCm'}>,
        <Axes: title={'center': 'PetalWidthCm'}>]], dtype=object)
```

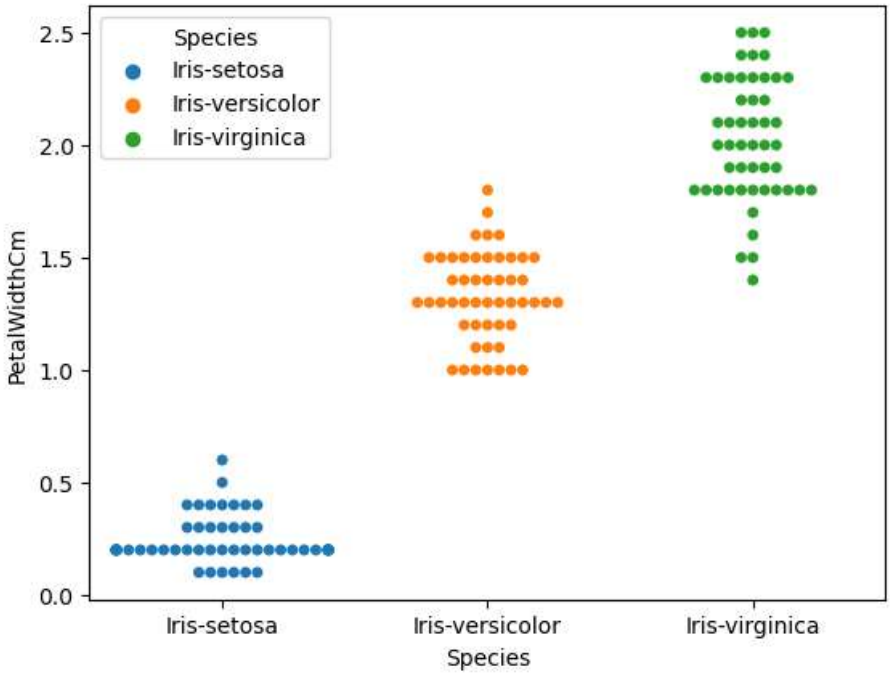


In [128]:
```
1 sns.swarmplot(data=idata,x='Species',y='PetalWidthCm', hue="Species")
```

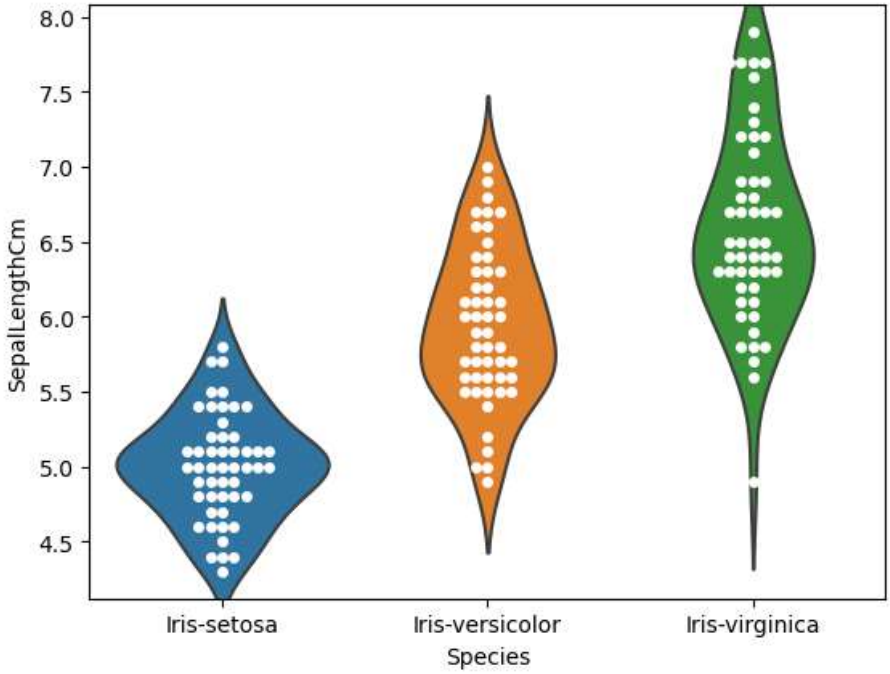
```
C:\Users\ASUS\anaconda3\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 6.0% of the p
oints cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[128]:
```
<Axes: xlabel='Species', ylabel='PetalWidthCm'>
```

```
C:\Users\ASUS\anaconda3\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 18.0% of the
points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

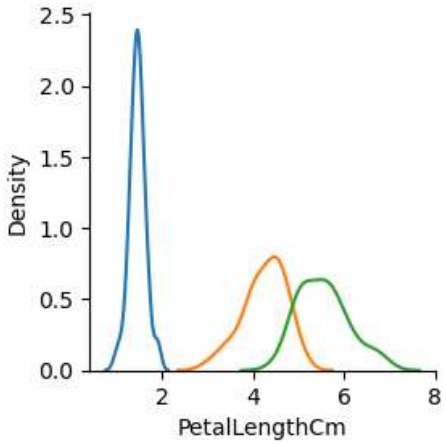


In [163]:
```
1 sns.violinplot(data=idata,x='Species',y='SepalLengthCm' ,inner=None)
2 sns.swarmplot(data=idata,x='Species', y='SepalLengthCm', color='white')
3 plt.show()
```

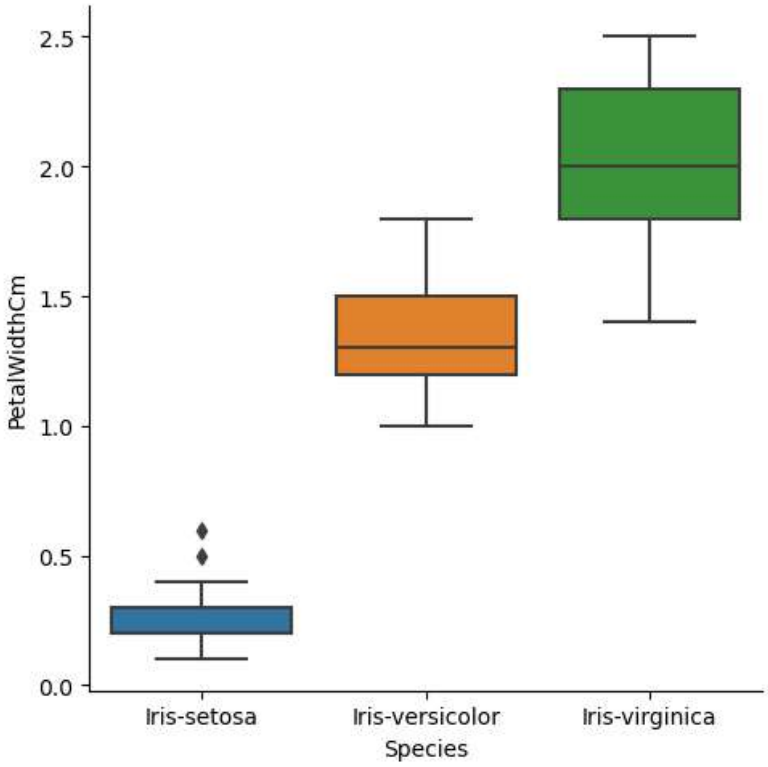


In [164]:
```
plt.gcf()
g=sns.FacetGrid(idata, hue='Species')
g.map(sns.kdeplot,'PetalLengthCm')
plt.ioff()
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```

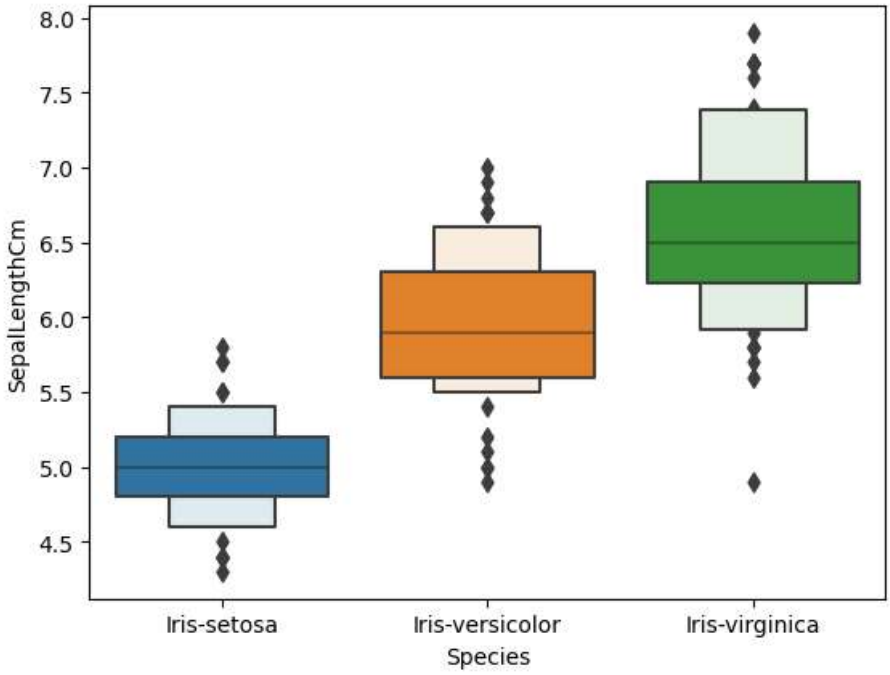


In [174]:
```
sns.catplot(data=idata,x='Species',y='PetalWidthCm' ,kind='box')
plt.show()
```



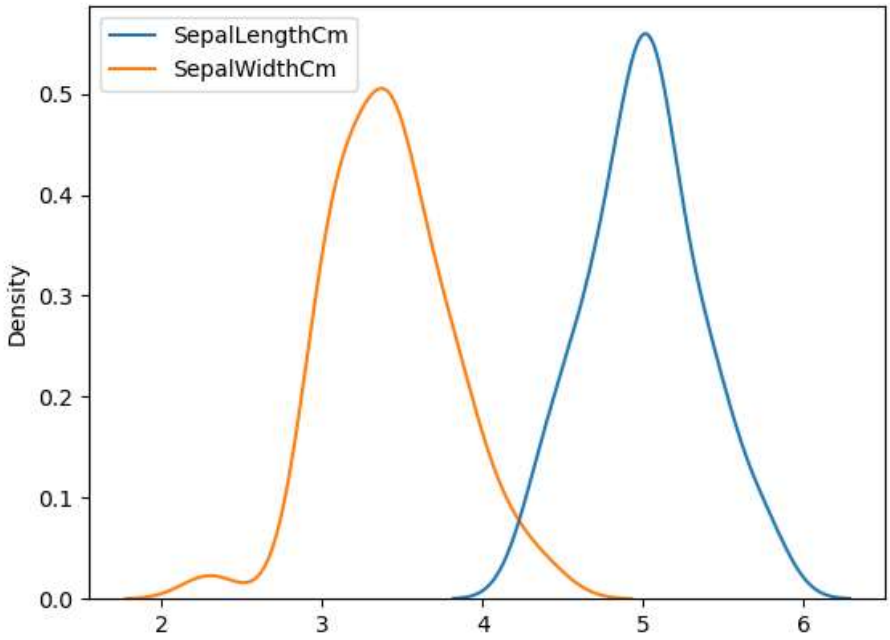
In [175]:
```
1 sns.boxenplot(data=idata,x='Species',y='SepalLengthCm')
```

Out[175]: <Axes: xlabel='Species', ylabel='SepalLengthCm'>



In [191]:
```
1 sub=idata[idata['Species'] =='Iris-setosa']
2 sns.kdeplot(data=sub[['SepalLengthCm', 'SepalWidthCm']] )
```

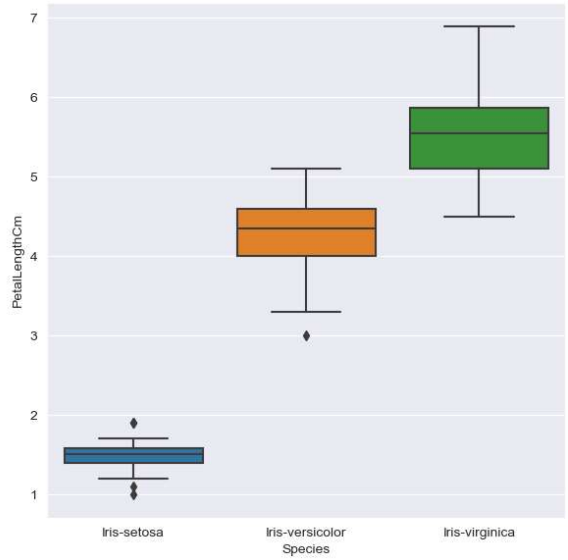
Out[191]: <Axes: ylabel='Density'>

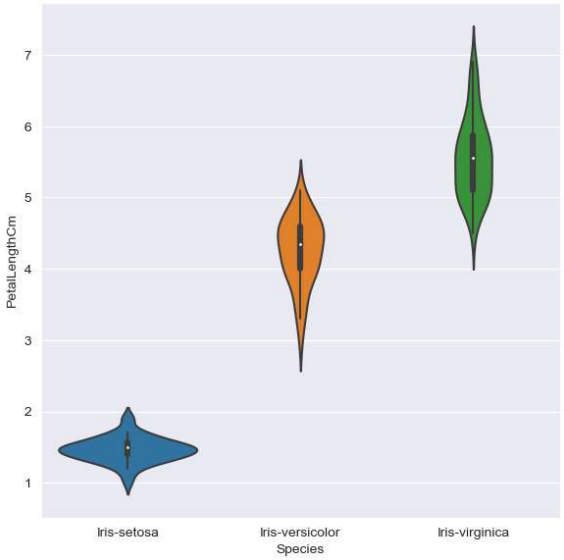


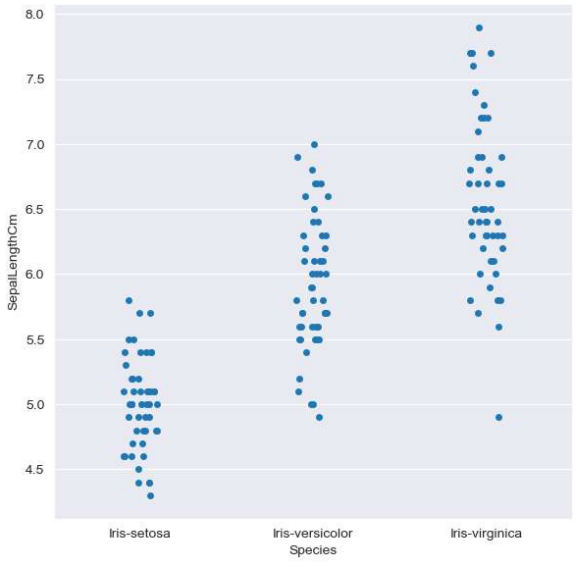
In [195]:

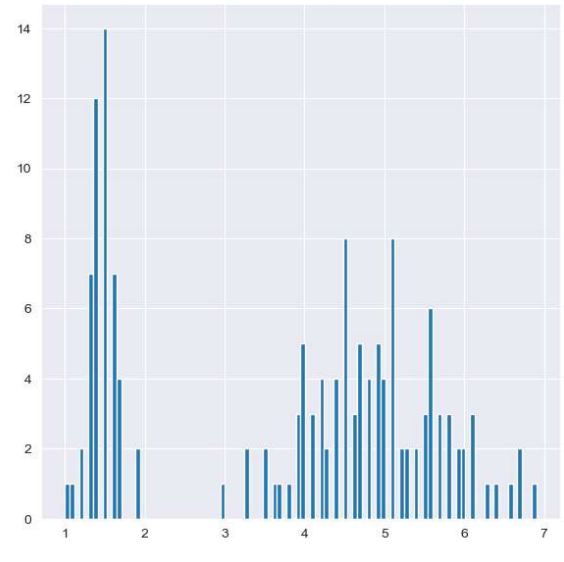
```
sns.set_style('darkgrid')
f,axes=plt.subplots(2,2,figsize=(15,15))

k1=sns.boxplot(x="Species", y="PetalLengthCm", data=idata,ax=axes[0,0])
k2=sns.violinplot(x='Species',y='PetalLengthCm',data=idata,ax=axes[0,1])
k3=sns.stripplot(x='Species',y='SepalLengthCm',data=idata,jitter=True,ax=axes[1,0])
axes[1,1].hist(idata.PetalLengthCm,bins=100)
plt.show()
```

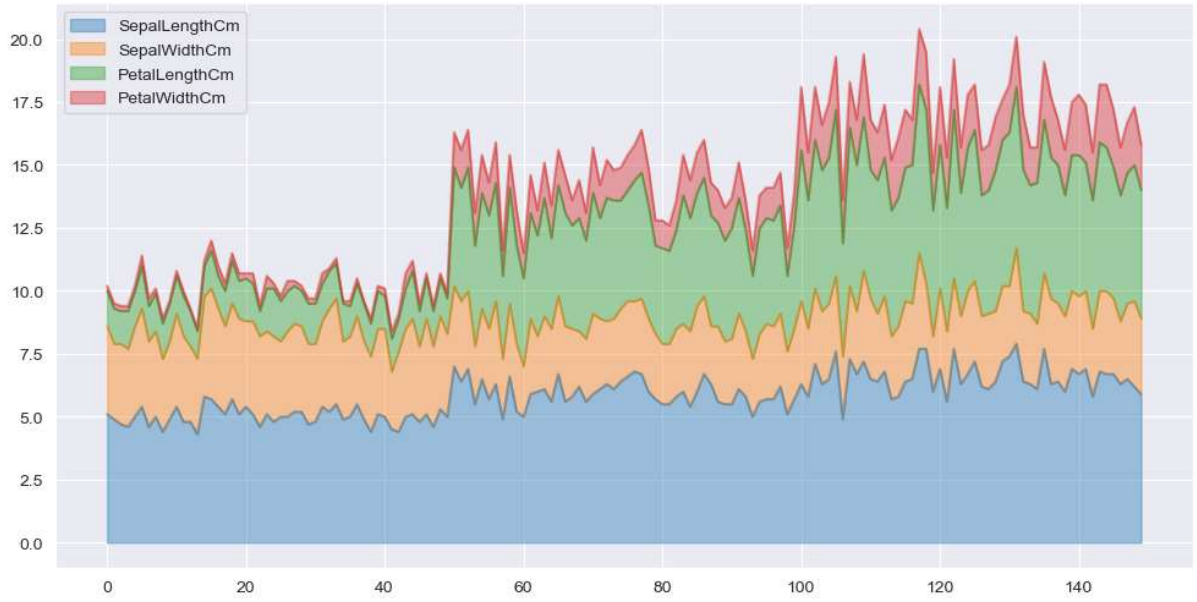








In [197]:
```
1 idata.plot.area(y=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'],alpha=0.4,fi
```



In [ ]:
```
1
```