IRIS DATASET VISUALIZATION(SEABORN,MATPLOTLIB)

In [4]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

In [6]:
```
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 displa
```

Importing Iris data set

In [8]:
```
iris=pd.read_csv(r"D:\Sid 17-03-2025\SIDDHARTH BOSE\FSDS & GEN AI\March\28th - I
```

In [9]:
```
iris
```

Out[9]:

	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

Displaying Data

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

Out[11]:

	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 [12]:
```
iris.drop('Id',axis=1,inplace=True)
```

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]:
```
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 [15]:
```
iris.isnull().any()
```

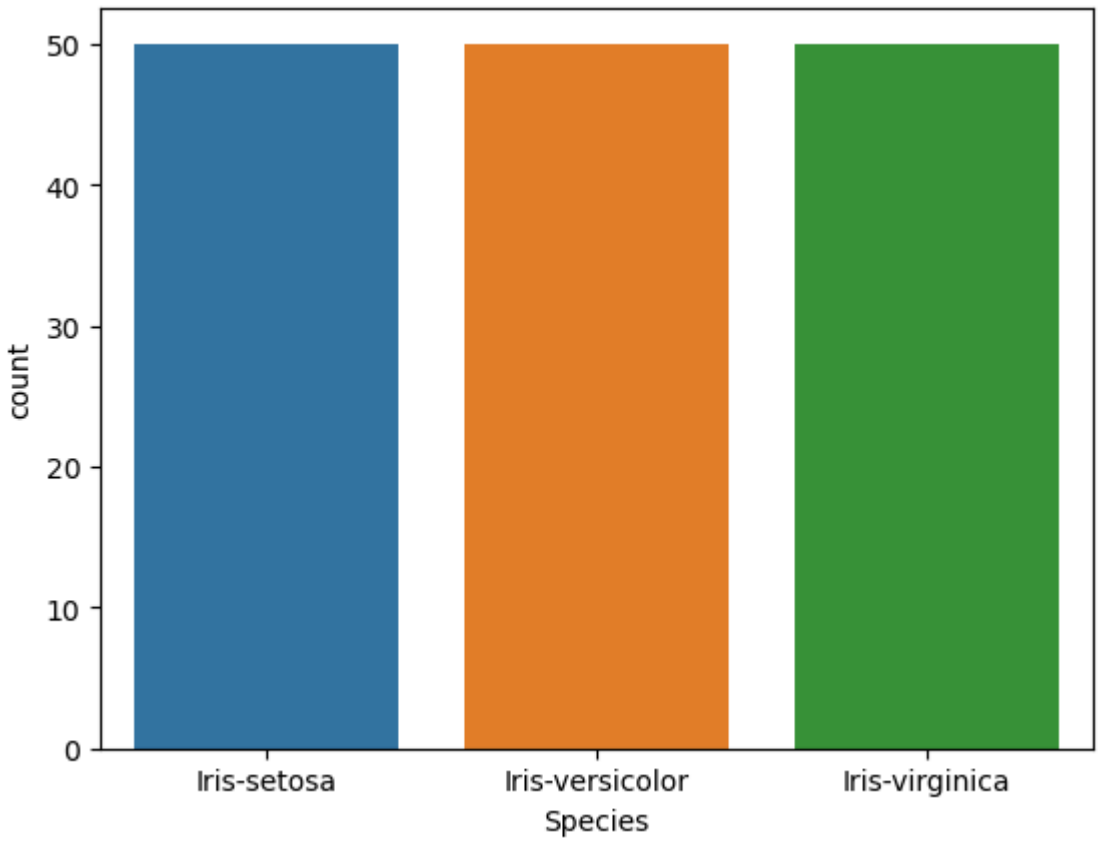
Out[15]:
```
SepalLengthCm    False
SepalWidthCm     False
PetalLengthCm    False
PetalWidthCm     False
Species          False
dtype: bool
```

In [16]:
```
iris['Species'].value_counts()
```

Out[16]:
```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

Bar Plot: Here the frequency of the observation is plotted.In this case we are plotting the frequency of the three species in the Iris Dataset

In [33]:
```
sns.countplot(x='Species', data=iris, hue='Species')
plt.show()
```

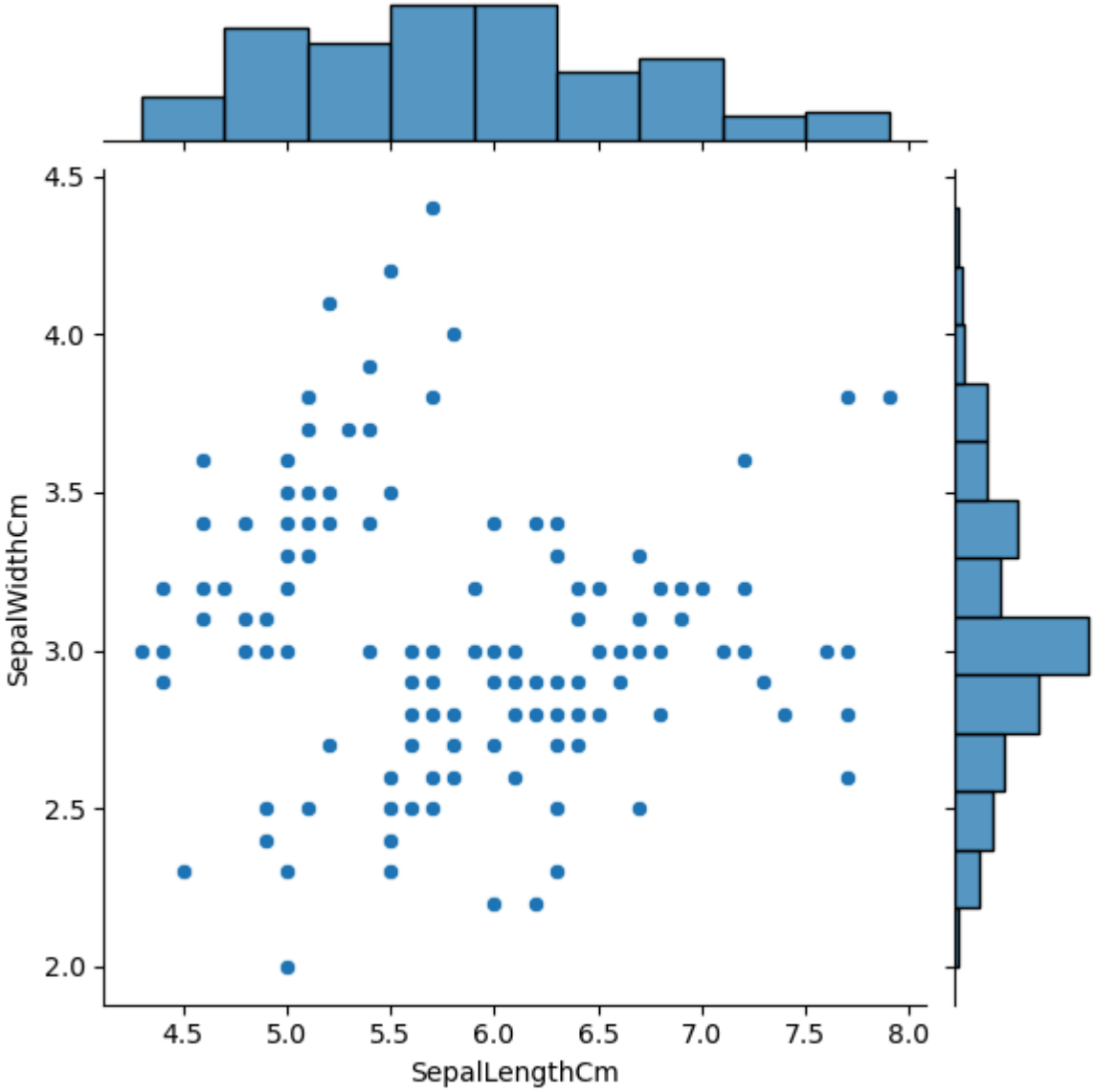


In [39]:
```
iris.head()
```

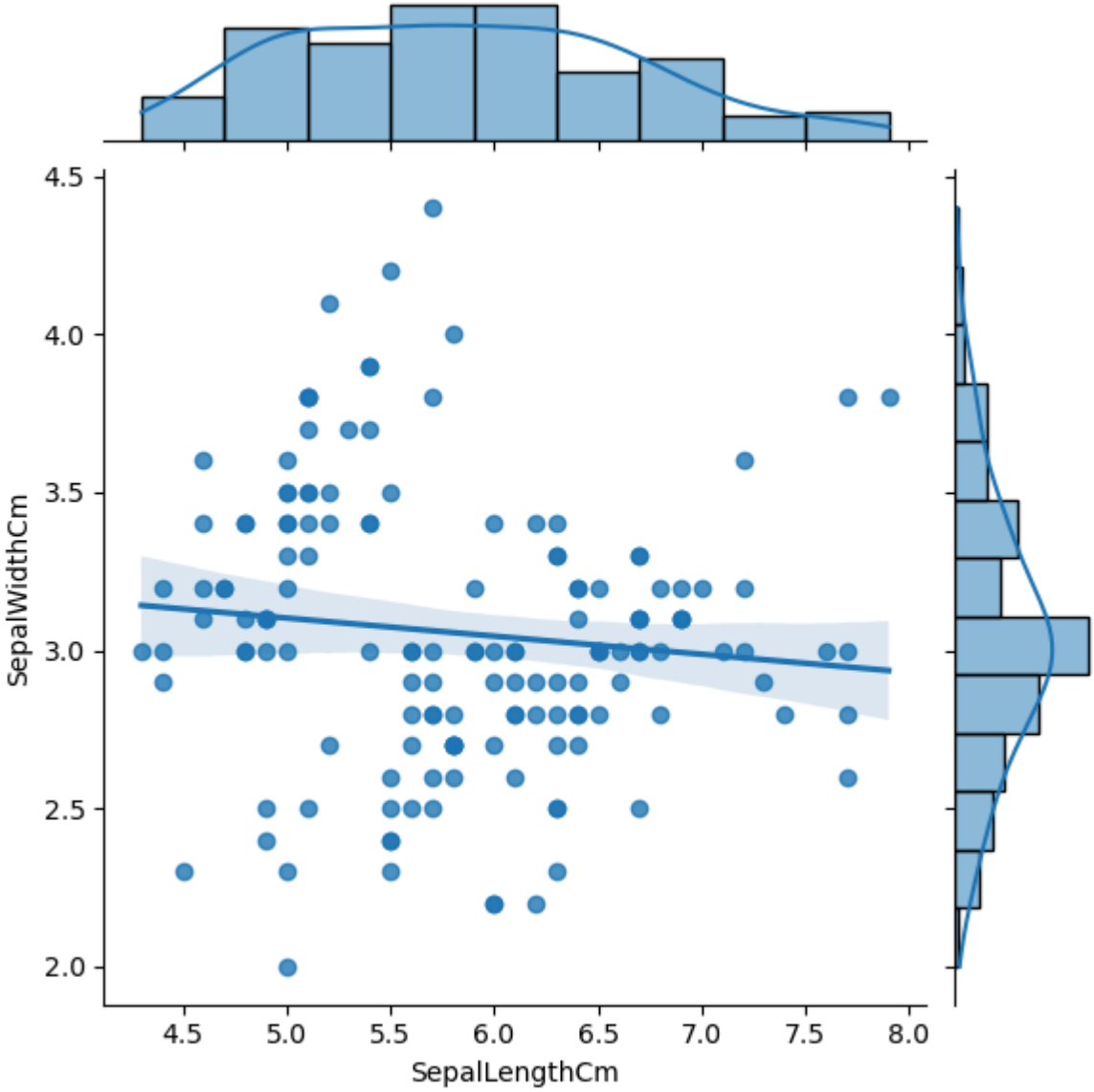
Out[39]:

	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

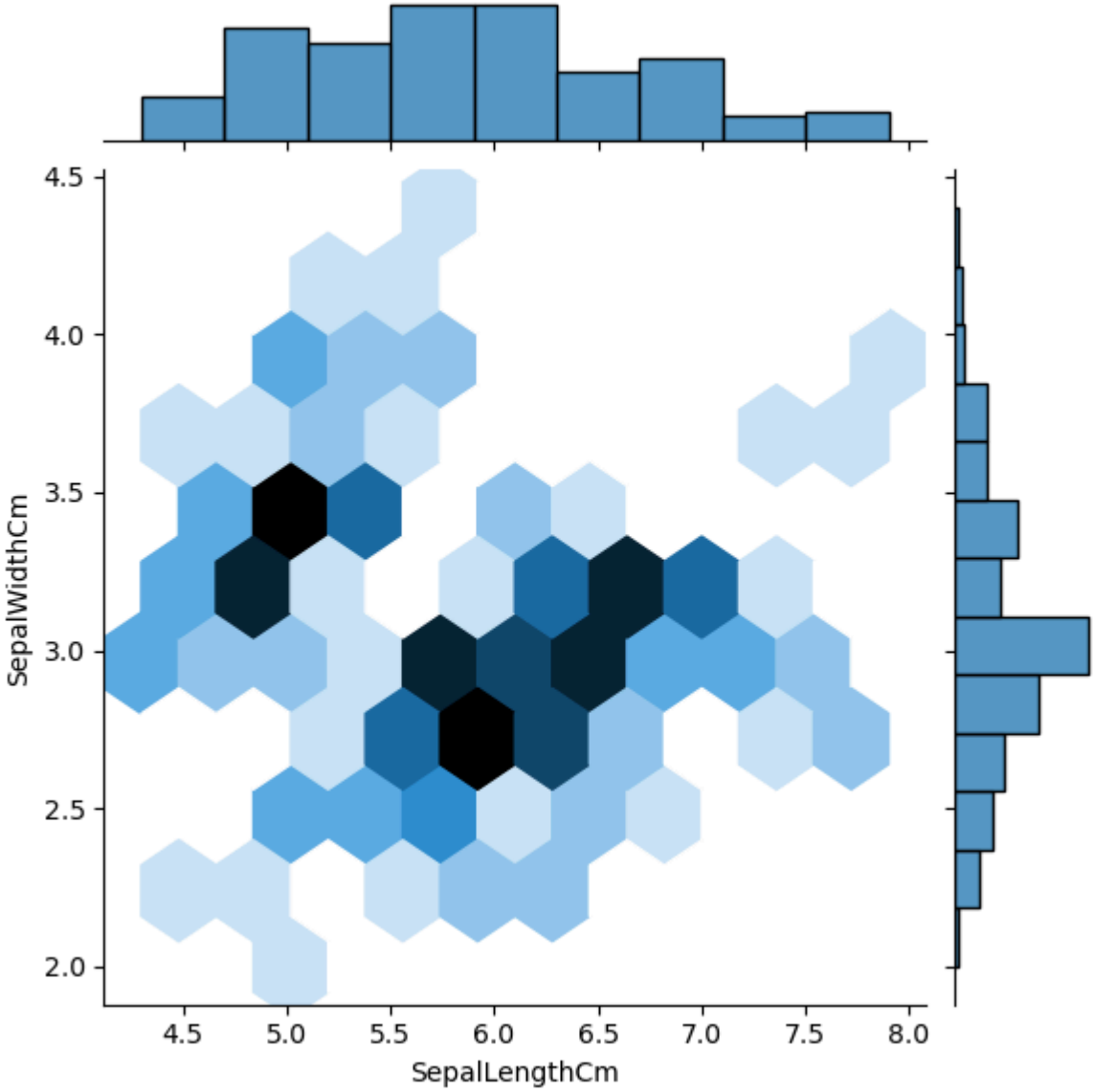
In [35]:
```
fig=sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',data=iris)
plt.show()
```



In [39]:
```
sns.jointplot(x="SepalLengthCm", y="SepalWidthCm", data=iris, kind="reg")
plt.show()
```

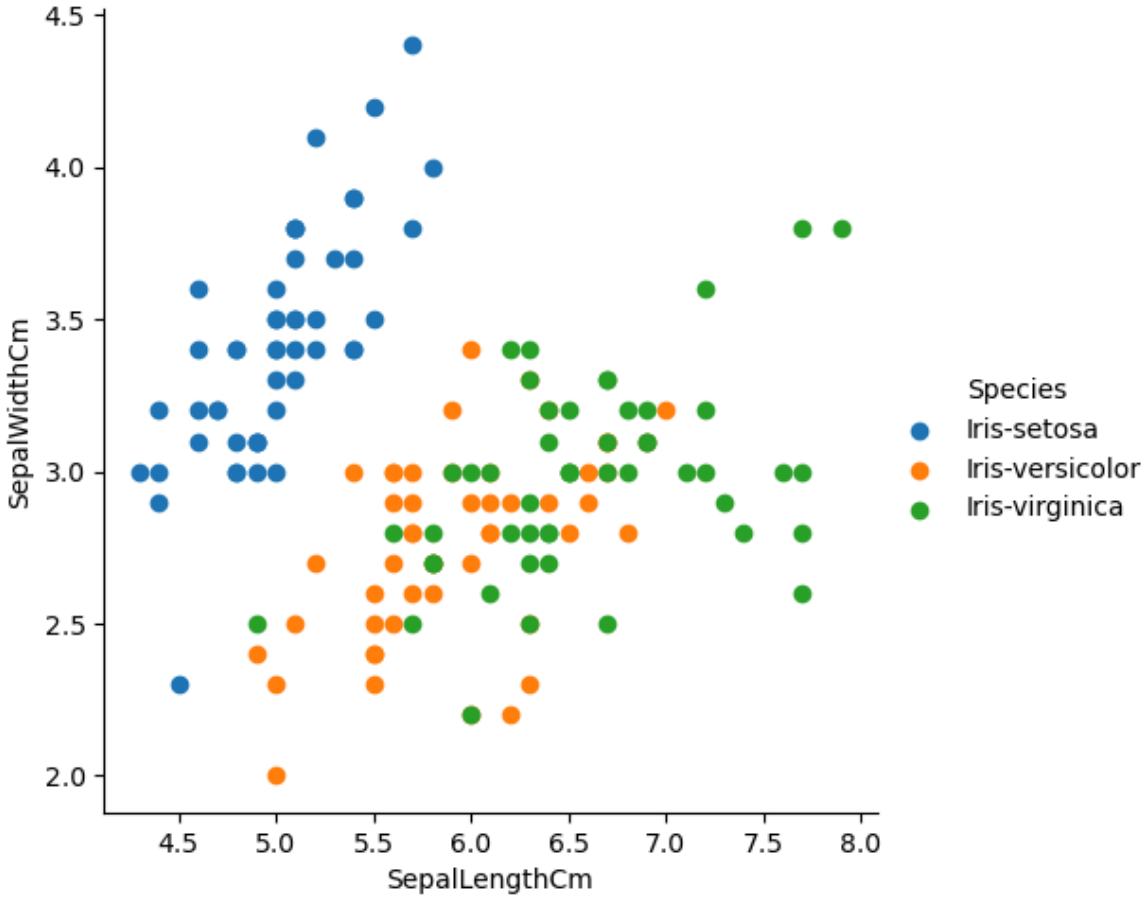


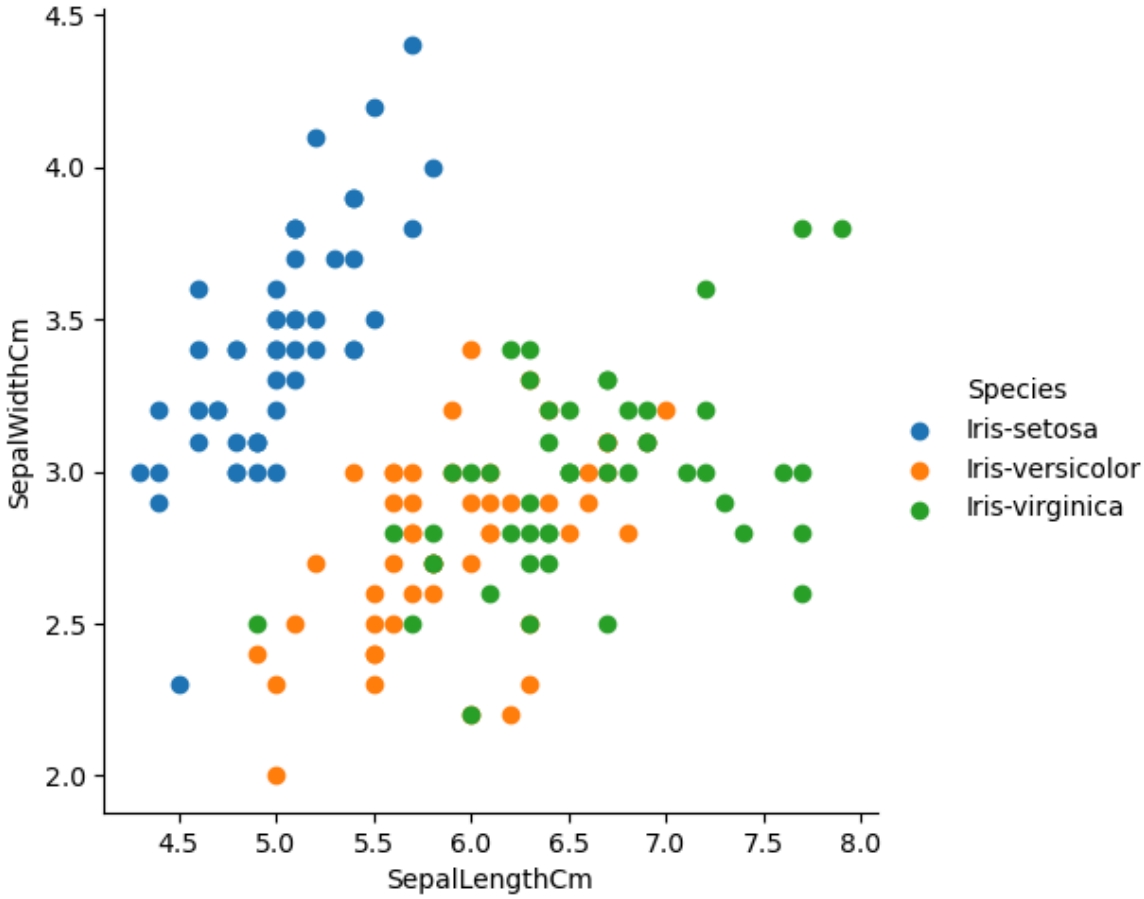
In [41]:
```
fig=sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',kind='hex',data=iris)
plt.show()
```



FacetGrid Plot

In [48]:
```
import matplotlib.pyplot as plt
%matplotlib inline

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





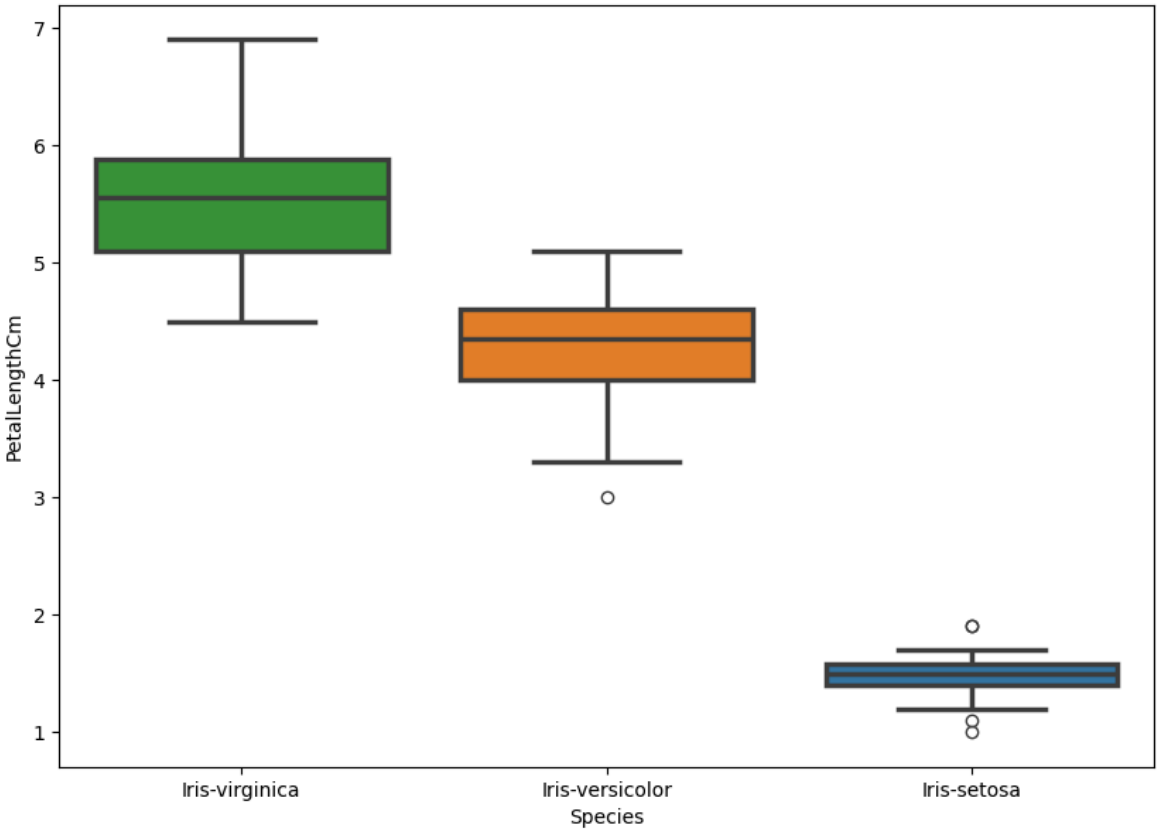
Boxplot or Whisker plot

In [51]:
```
iris.head()
```

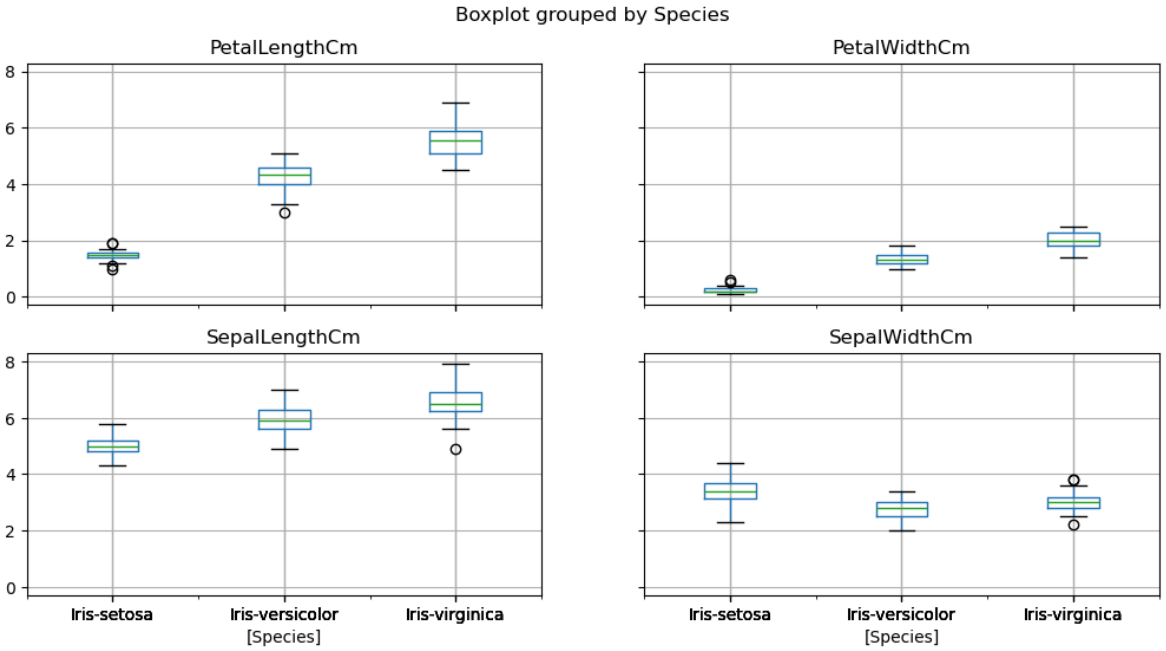
Out[51]:

	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 [57]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='PetalLengthCm',data=iris,order=['Iris-virginica',
plt.show()
```

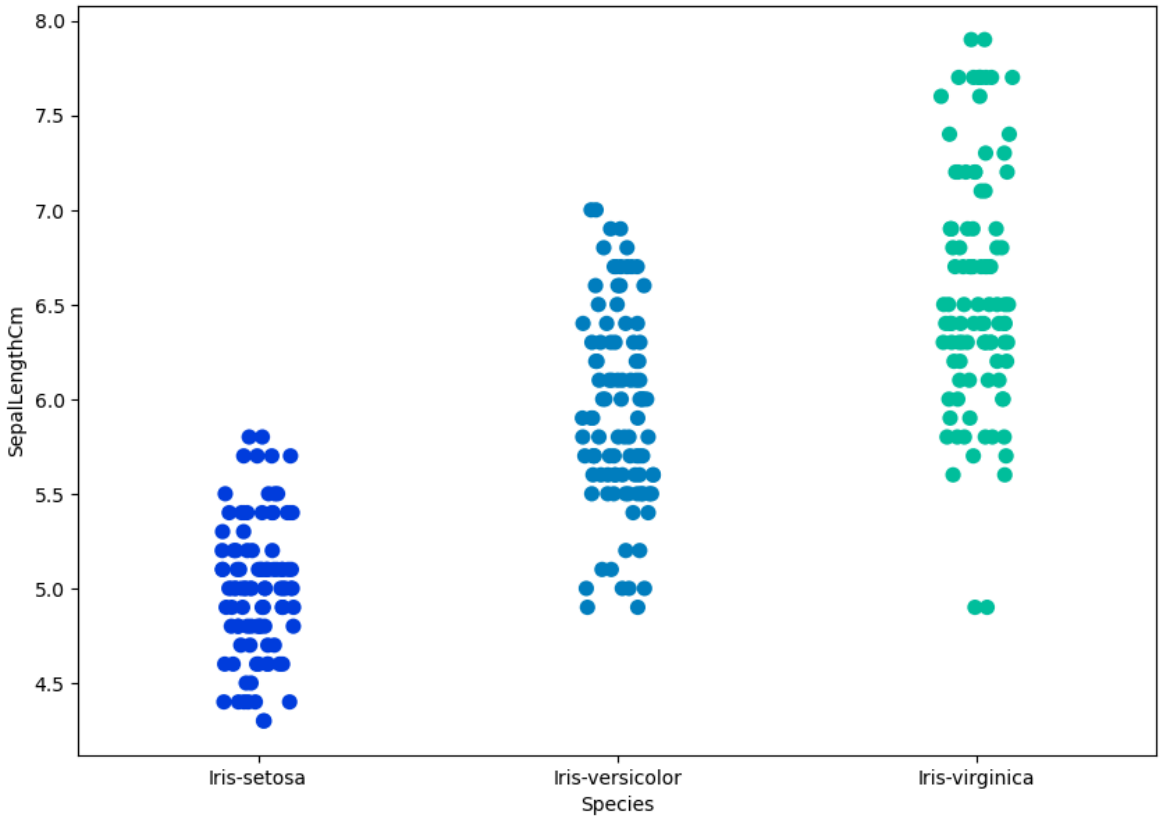


In [59]:
```
#iris.drop("Id", axis=1).boxplot(by="Species", figsize=(12, 6))
iris.boxplot(by="Species", figsize=(12, 6))
plt.show()
```



Strip Plot

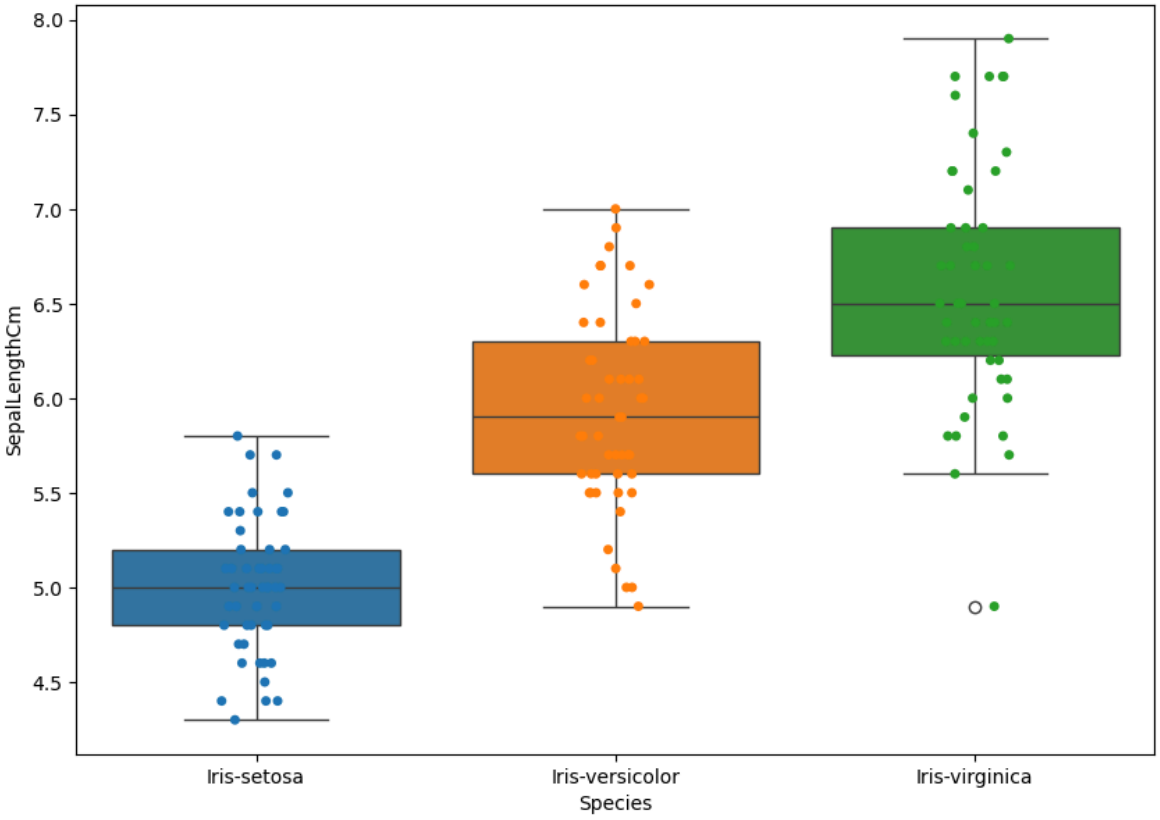
In [64]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True,edgecolor=
plt.show()
```



Combining Box and Strip Plots

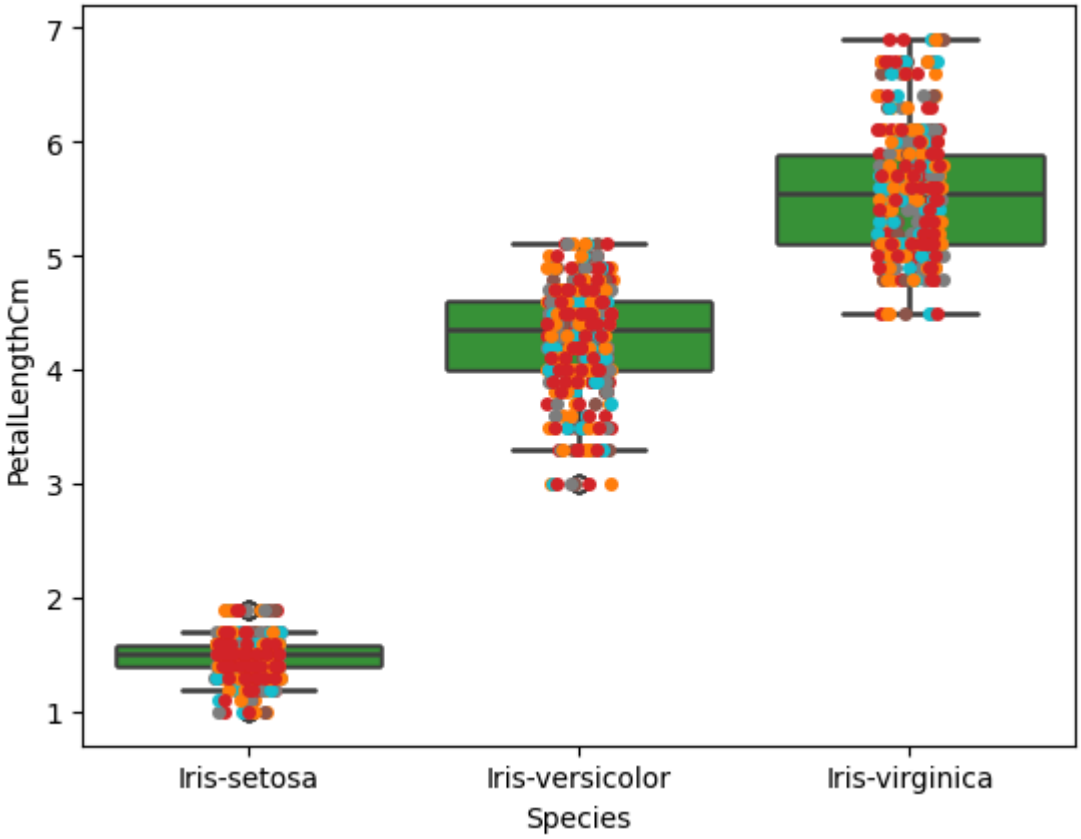
In [71]:
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxplot(x='Species',y='SepalLengthCm',data=iris, hue='Species')
```

```
fig=sns.stripplot(x='Species',y='SepalLengthCm',data=iris,jitter=True, hue='Spec
plt.show()
```



In [106…

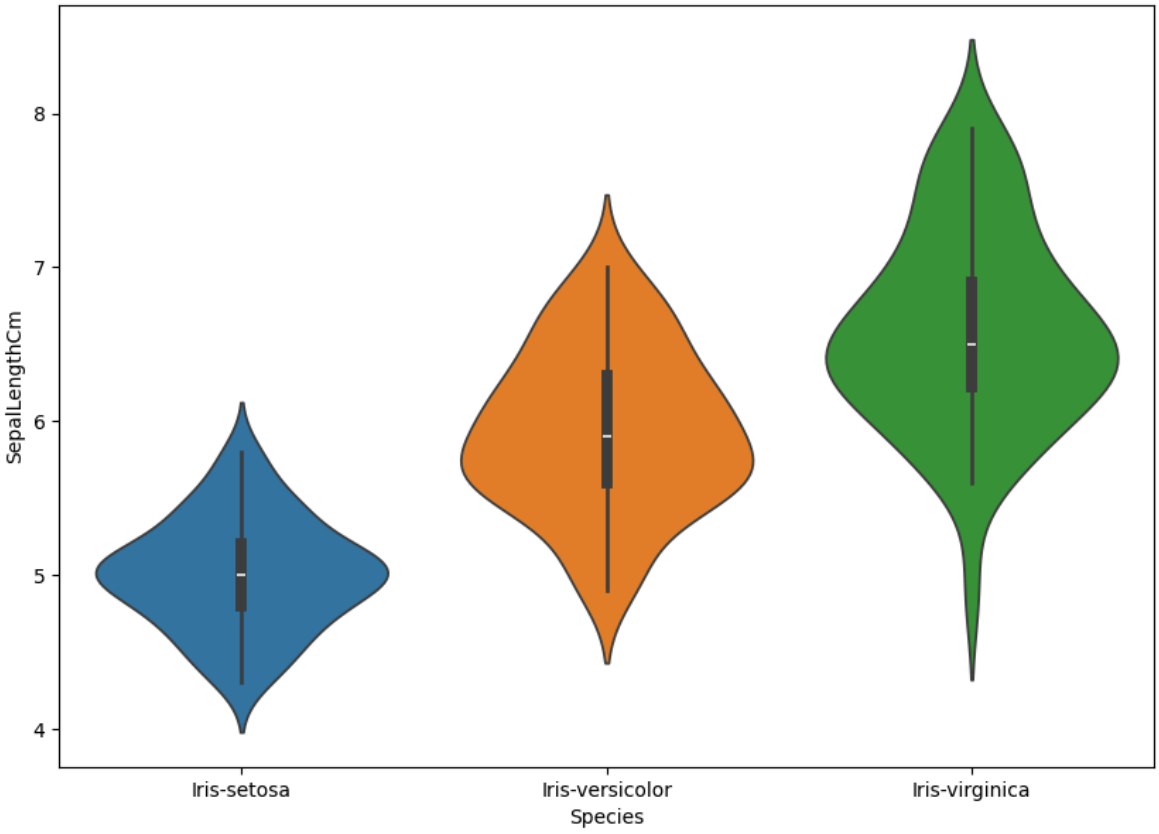
```
ax= sns.boxplot(x="Species", y="PetalLengthCm", data=iris)
ax= sns.stripplot(x="Species", y="PetalLengthCm", data=iris, jitter=True, edgeco

#boxtwo = ax.artists
#boxtwo.set_facecolor('yellow')
#boxtwo.set_edgecolor('black')
#boxthree=ax.artists
#boxthree.set_facecolor('red')
#boxthree.set_edgecolor('black')
#boxthree=ax.artists
#boxthree.set_facecolor('green')
#boxthree.set_edgecolor('black')
plt.show()
```



Violin Plot

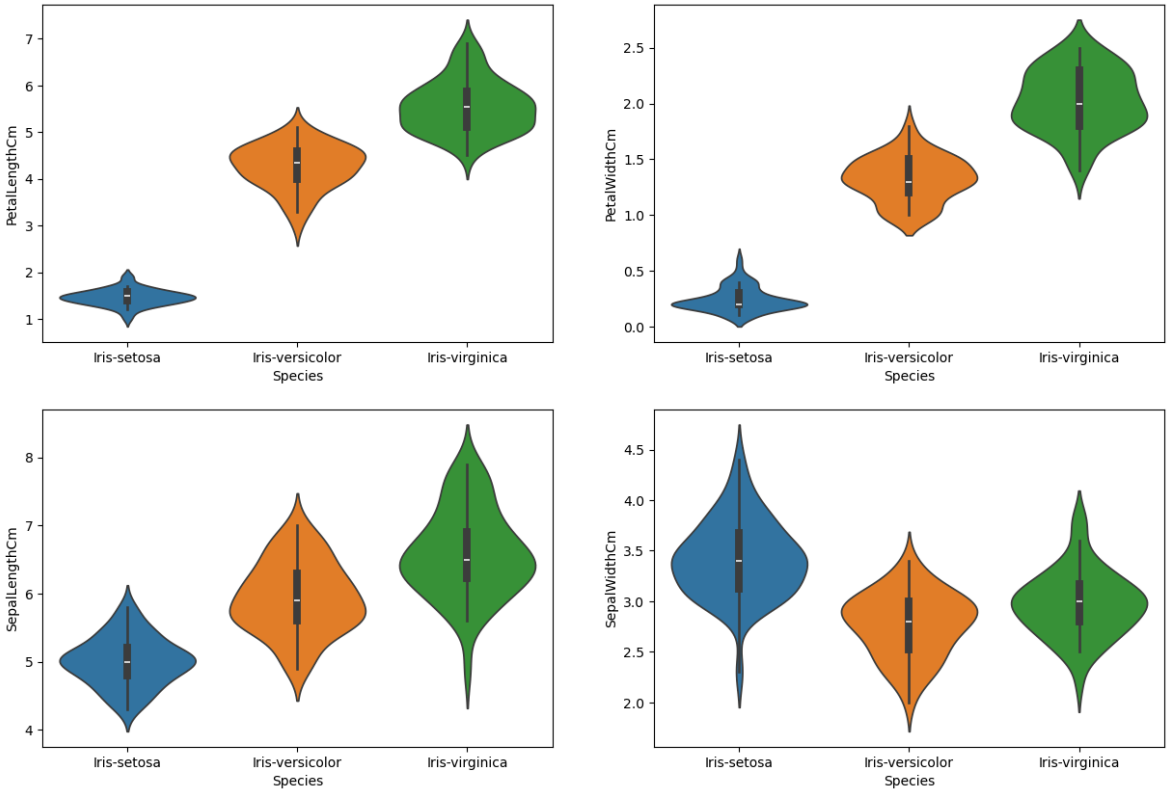
In [108…

```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.violinplot(x='Species',y='SepalLengthCm',data=iris, hue='Species')
plt.show()
```



In [112…

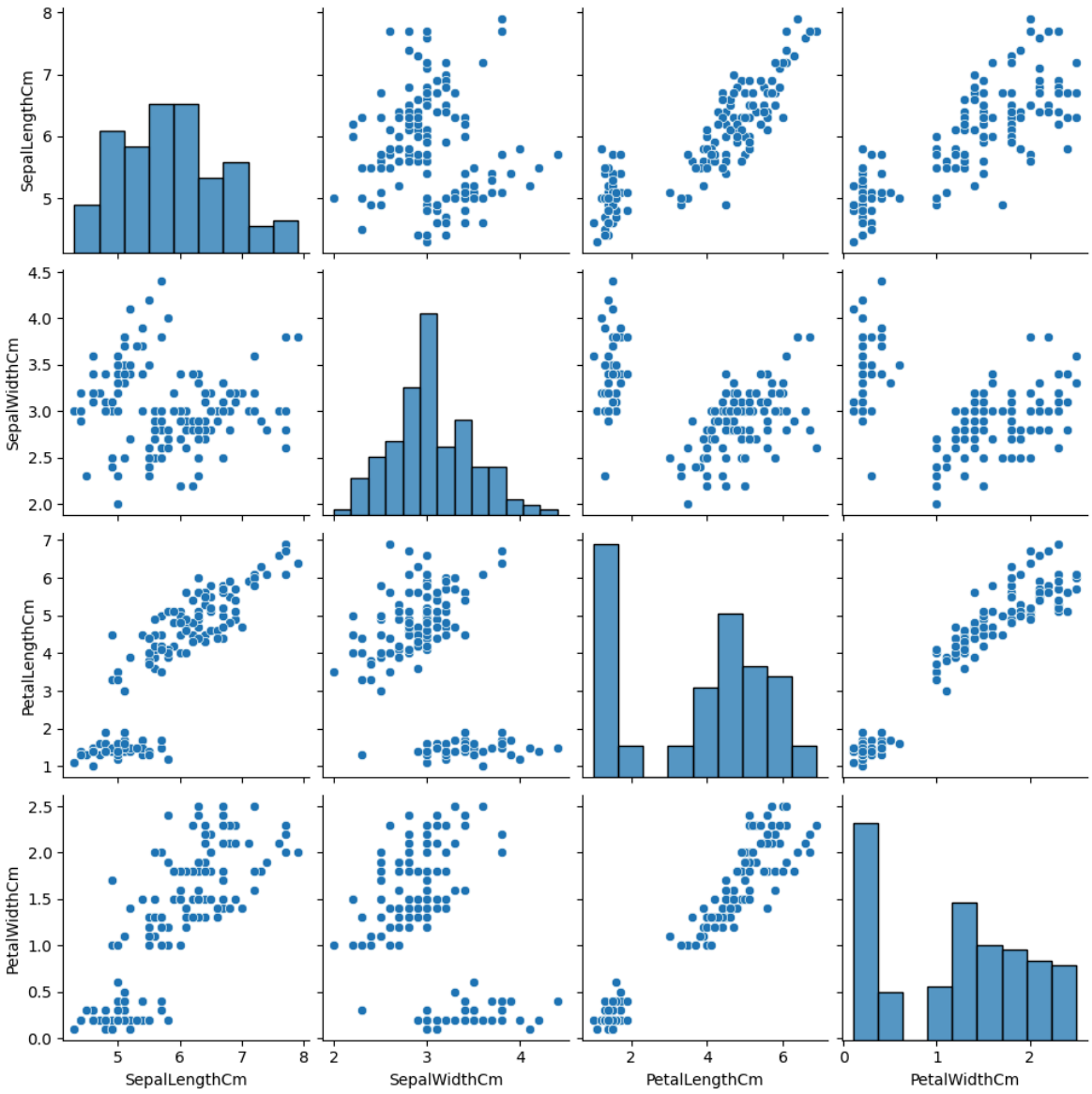
```
plt.figure(figsize=(15,10))
plt.subplot(2,2,1)
sns.violinplot(x='Species',y='PetalLengthCm',data=iris, hue='Species')
plt.subplot(2,2,2)
sns.violinplot(x='Species',y='PetalWidthCm',data=iris, hue='Species')
plt.subplot(2,2,3)
sns.violinplot(x='Species',y='SepalLengthCm',data=iris, hue='Species')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

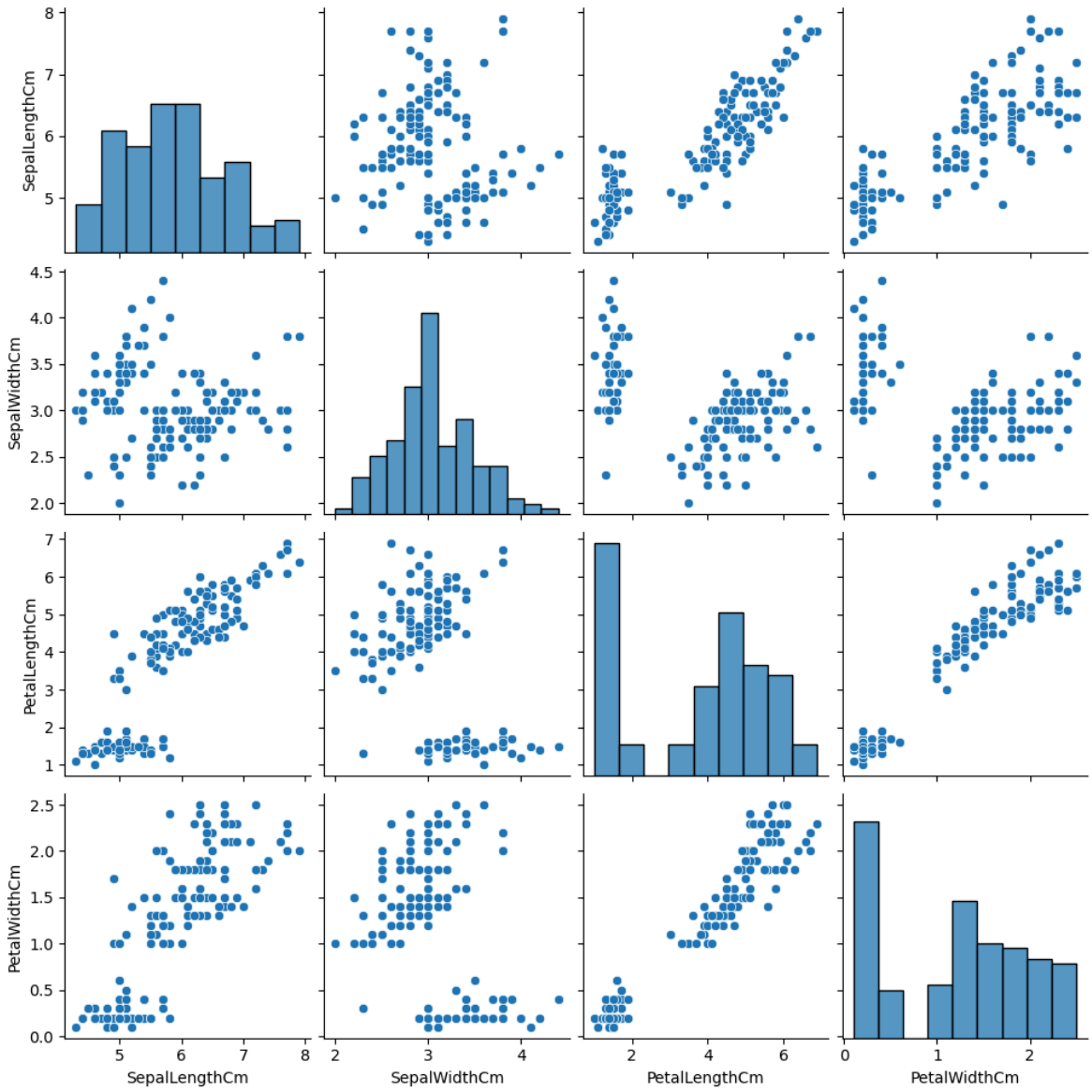


Pair Plot:

In [117…

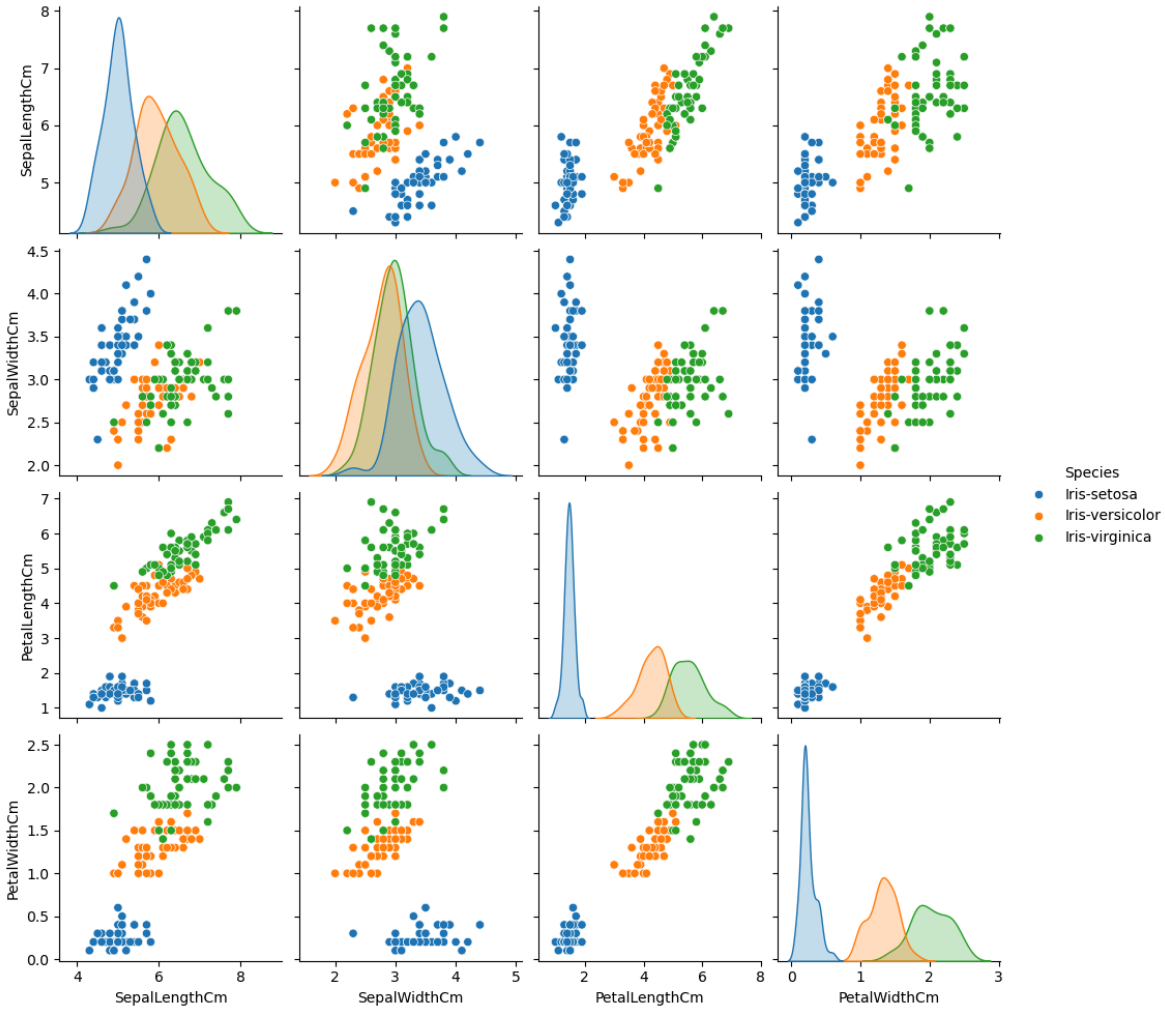
```
sns.pairplot(data=iris,kind='scatter')
plt.show()
```

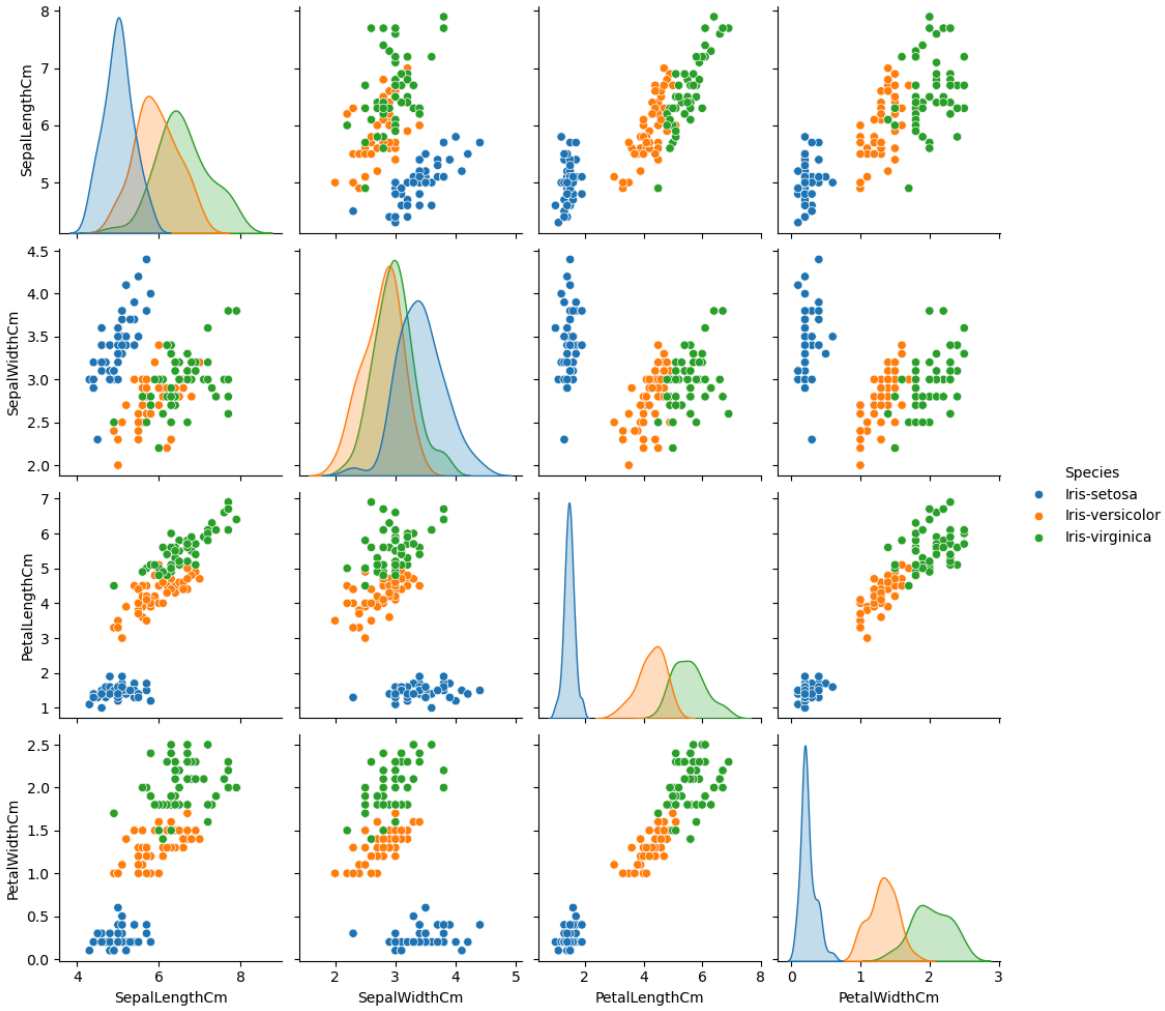

SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm



In [121…

```
sns.pairplot(iris,hue='Species');
plt.show()
```

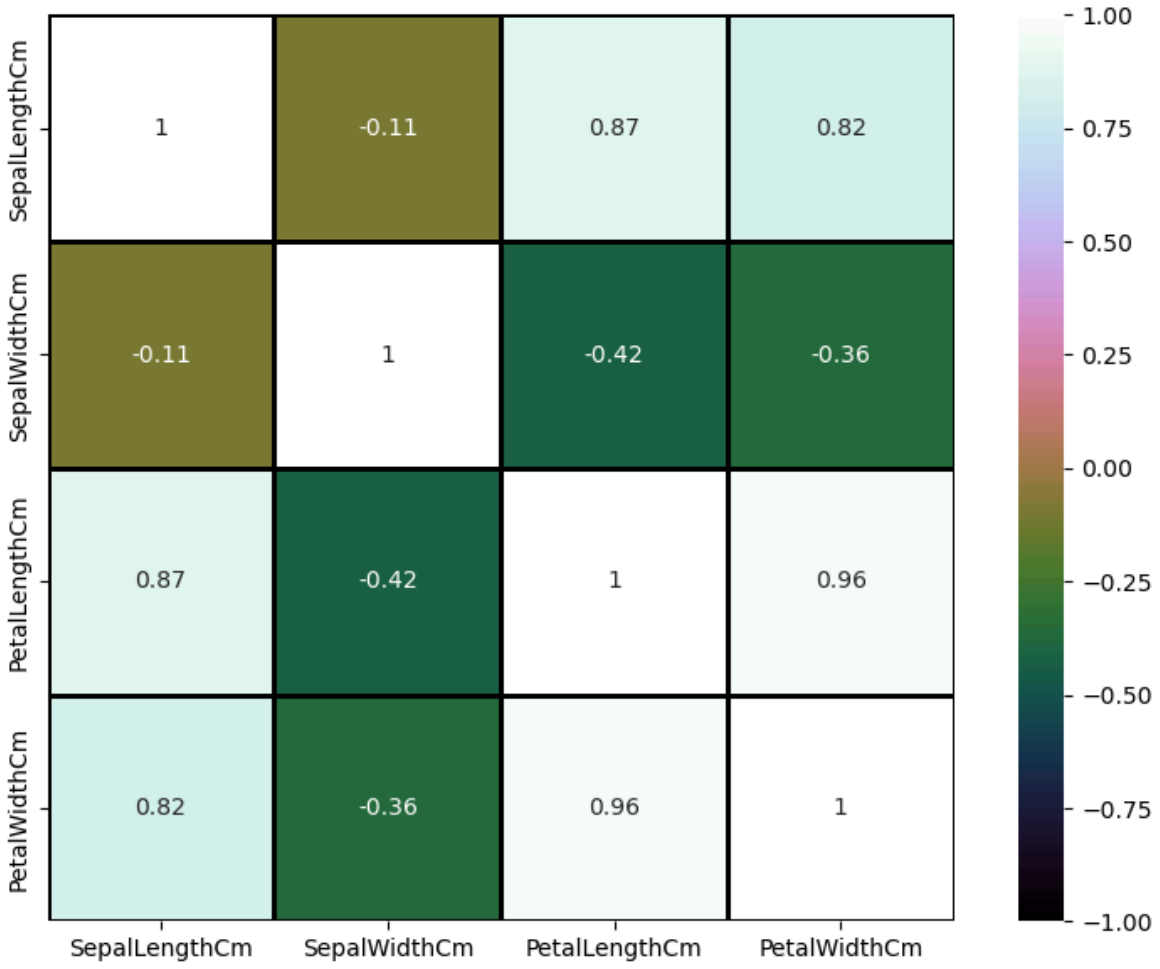

SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
Species
Iris-setosa
Iris-versicolor
Iris-virginica



Heat map

In [126…

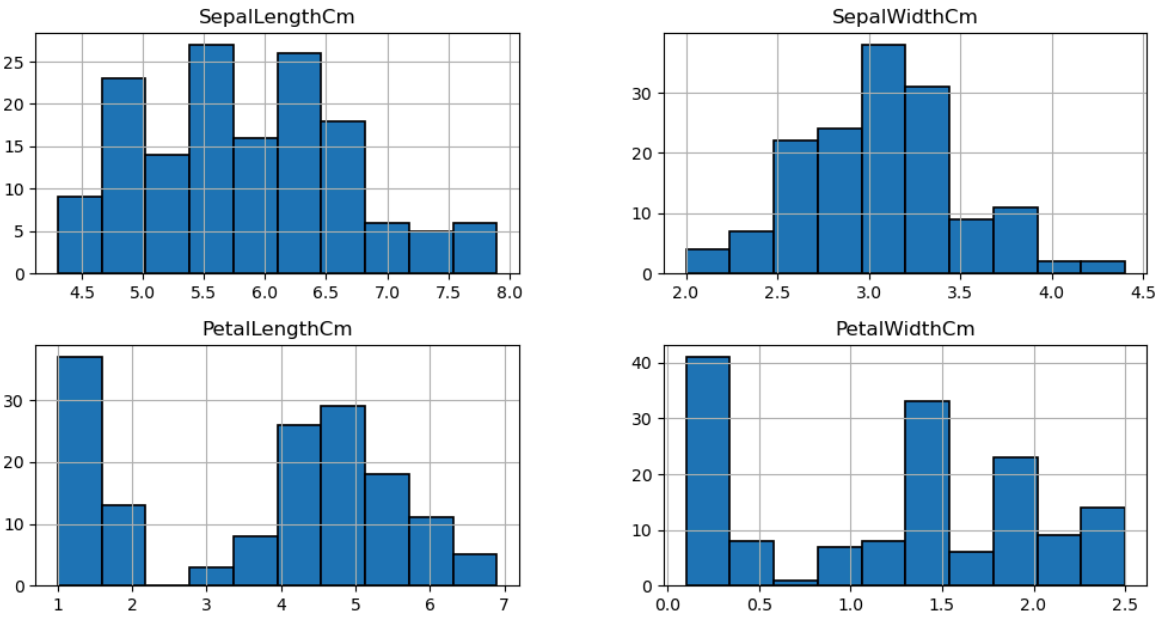
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.heatmap(iris.drop(columns=['Species']).corr(),annot=True,cmap='cubehelix
plt.show()
```



Distribution plot:

In [129…

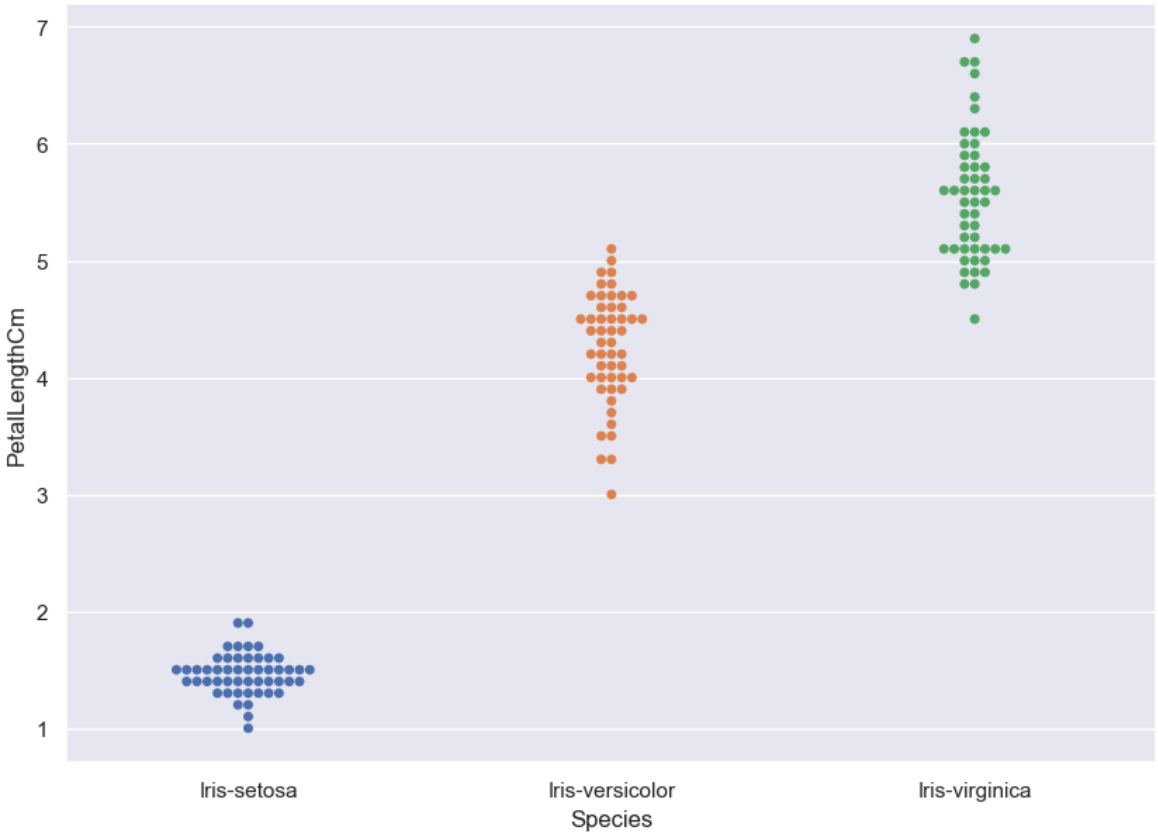
```
iris.hist(edgecolor='black', linewidth=1.2)
fig=plt.gcf()
fig.set_size_inches(12,6)
plt.show()
```



Swarm plot

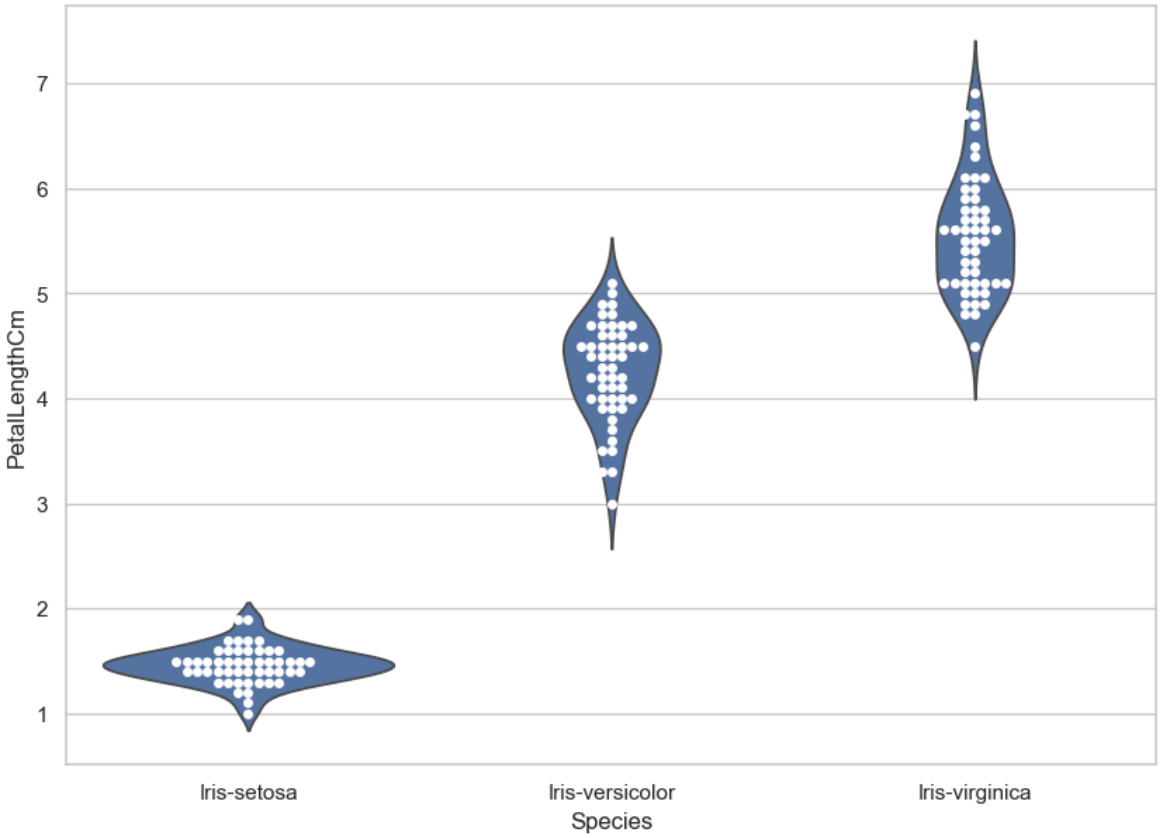
In [132…

```
sns.set(style="darkgrid")
fig=plt.gcf()
fig.set_size_inches(10,7)
fig = sns.swarmplot(x="Species", y="PetalLengthCm", data=iris,hue='Species')
plt.show()
```



In [136…

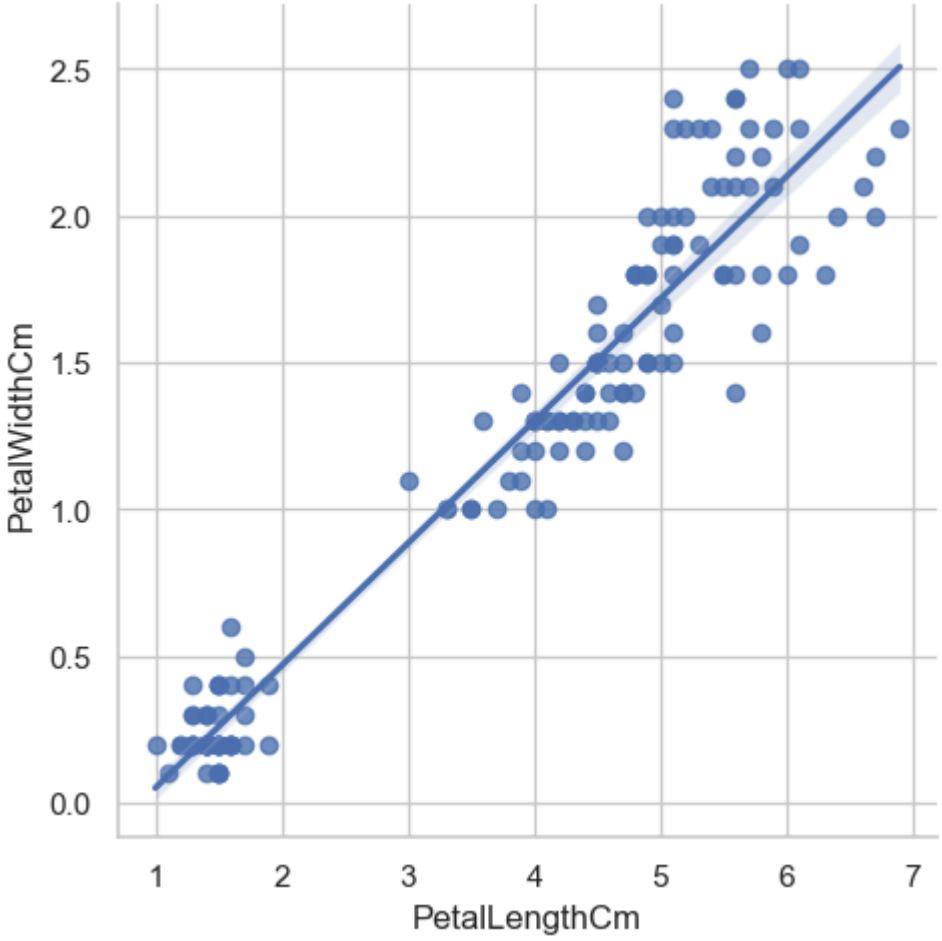
```
sns.set(style="whitegrid")
fig=plt.gcf()
fig.set_size_inches(10,7)
ax = sns.violinplot(x="Species", y="PetalLengthCm", data=iris, inner=None)
ax = sns.swarmplot(x="Species", y="PetalLengthCm", data=iris,color="white", edge
plt.show()
```



LM PLot

In [139…

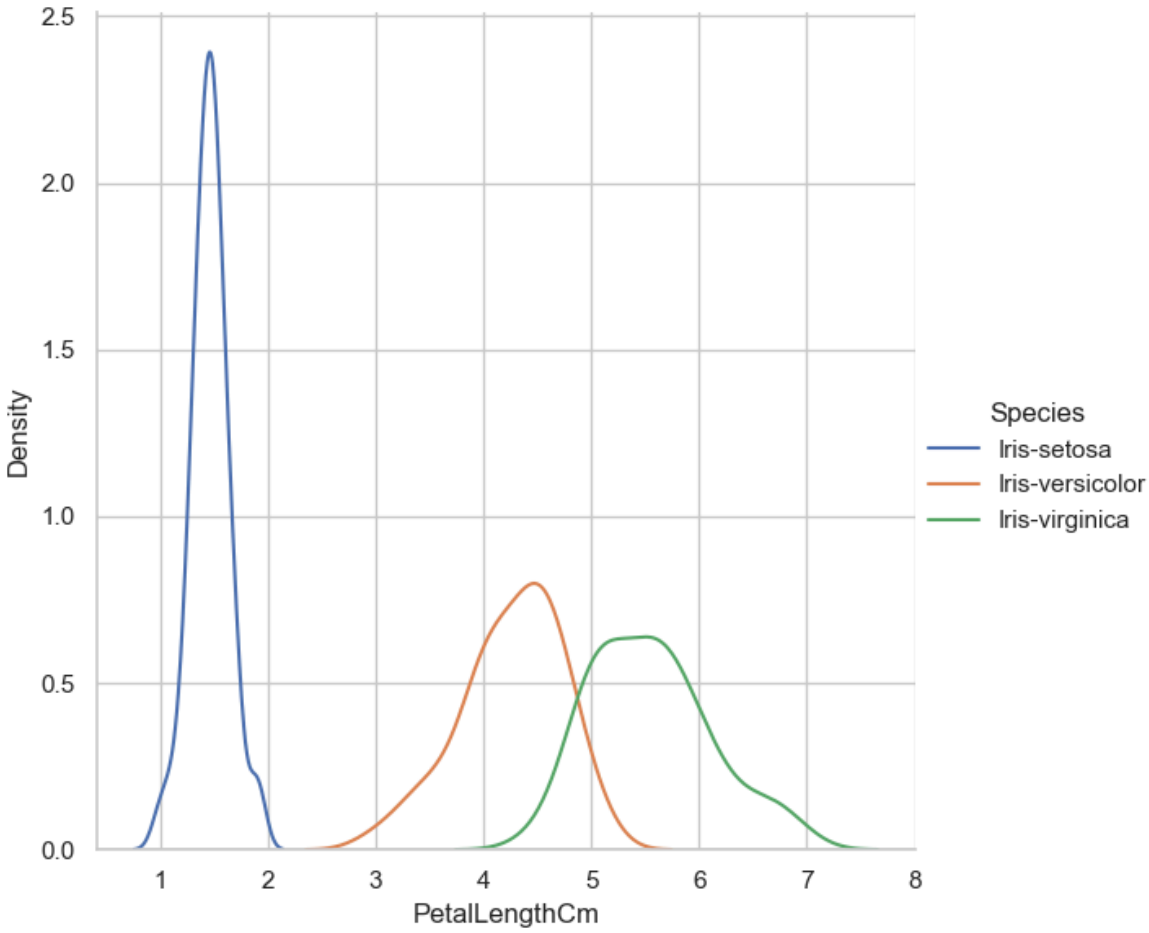
```
fig=sns.lmplot(x="PetalLengthCm", y="PetalWidthCm",data=iris)
plt.show()
```



FacetGrid

In [144…

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



Factor Plot

In [153…

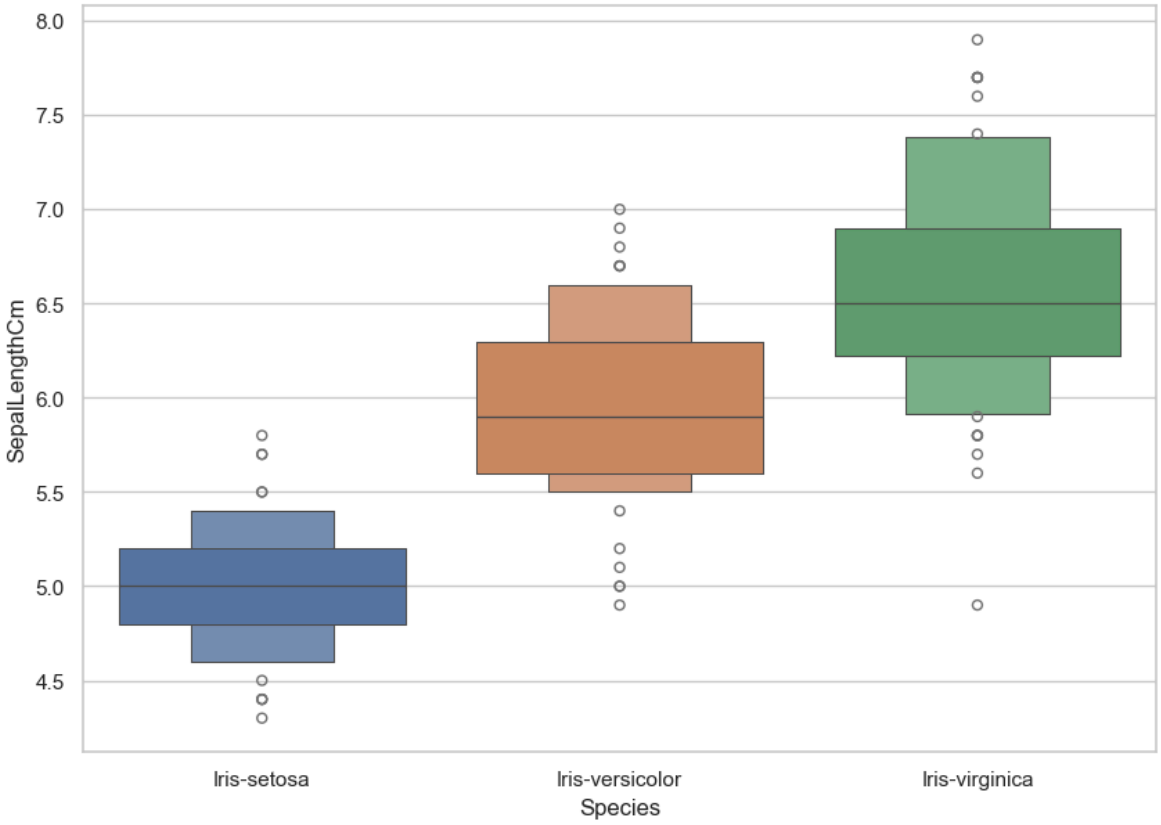
```
#f,ax=plt.subplots(1,2,figsize=(18,8))
sns.factorplot(x='Species',y='SepalLengthCm',data=iris)
plt.ioff()
plt.show()
#sns.factorplot('Species','SepalLengthCm',data=iris,ax=ax[0][0])
#sns.factorplot('Species','SepalWidthCm',data=iris,ax=ax[0][1])
#sns.factorplot('Species','PetalLengthCm',data=iris,ax=ax[1][0])
#sns.factorplot('Species','PetalWidthCm',data=iris,ax=ax[1][1])
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[153], line 2
      1 #f,ax=plt.subplots(1,2,figsize=(18,8))
----> 2 sns.factorplot(x='Species',y='SepalLengthCm',data=iris)
      3 plt.ioff()
      4 plt.show()

AttributeError: module 'seaborn' has no attribute 'factorplot'
```

Boxen Plot

In [162…

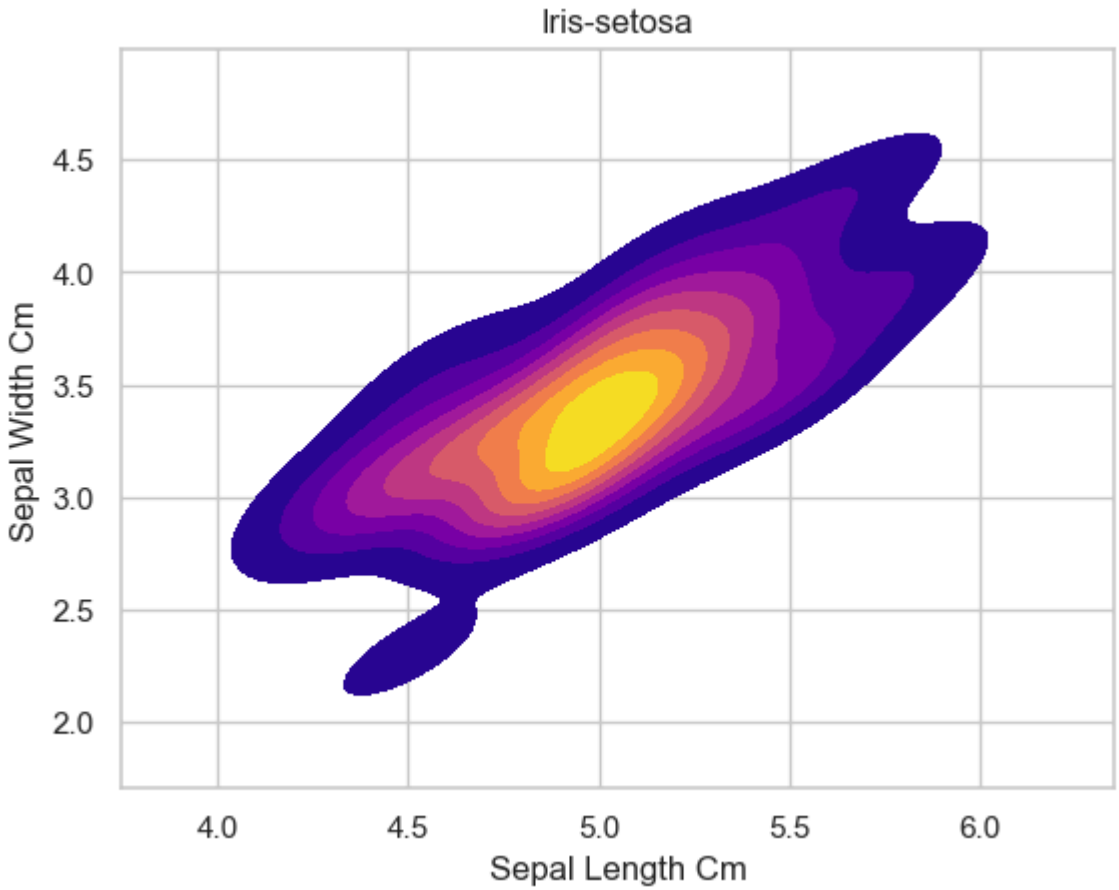
```
fig=plt.gcf()
fig.set_size_inches(10,7)
fig=sns.boxenplot(x='Species',y='SepalLengthCm',data=iris, hue='Species')
plt.show()
```



KDE Plot

In [173…

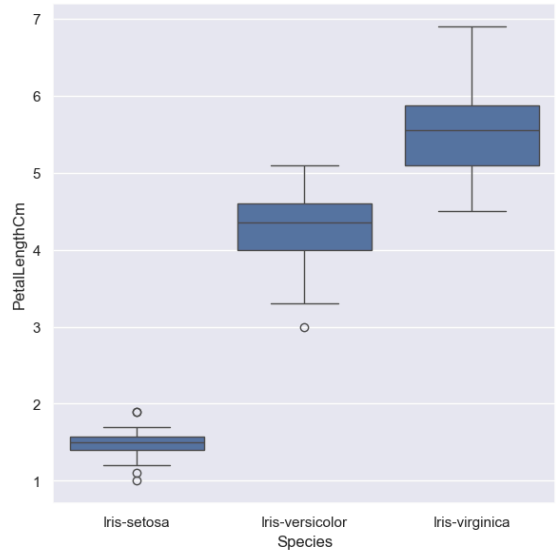
```
# Create a kde plot of sepal_length versus sepal width for setosa species of flo
sub=iris[iris['Species']=='Iris-setosa']
sns.kdeplot(x=sub['SepalLengthCm'],y=sub['SepalWidthCm'], cmap="plasma", shade=T
plt.title('Iris-setosa')
plt.xlabel('Sepal Length Cm')
plt.ylabel('Sepal Width Cm')
plt.show()
```

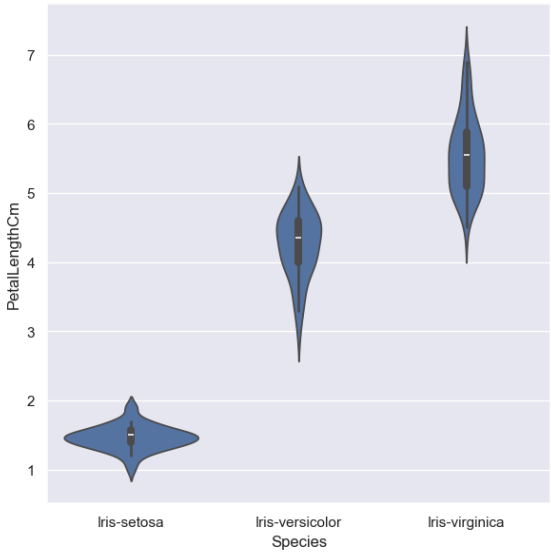


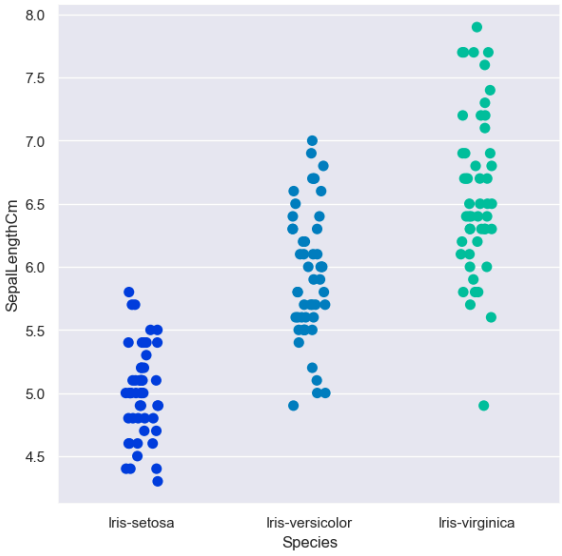
Dashboard

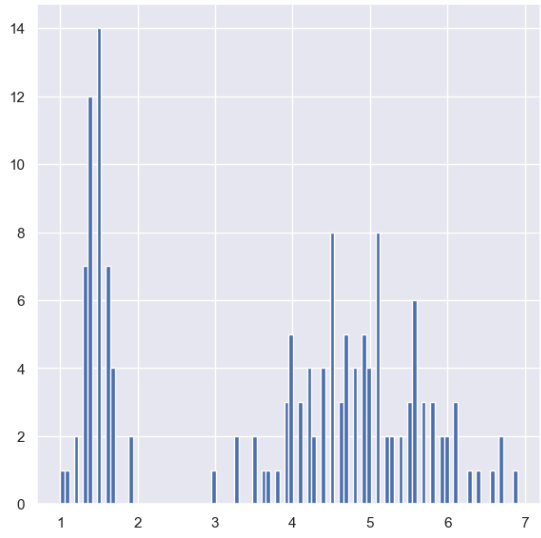
In [176…

```
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='
#axes[1,1].hist(iris.hist,bin=10)
axes[1,1].hist(iris.PetalLengthCm,bins=100)
#k2.set(xlim=(-1,0.8))
plt.show()
```





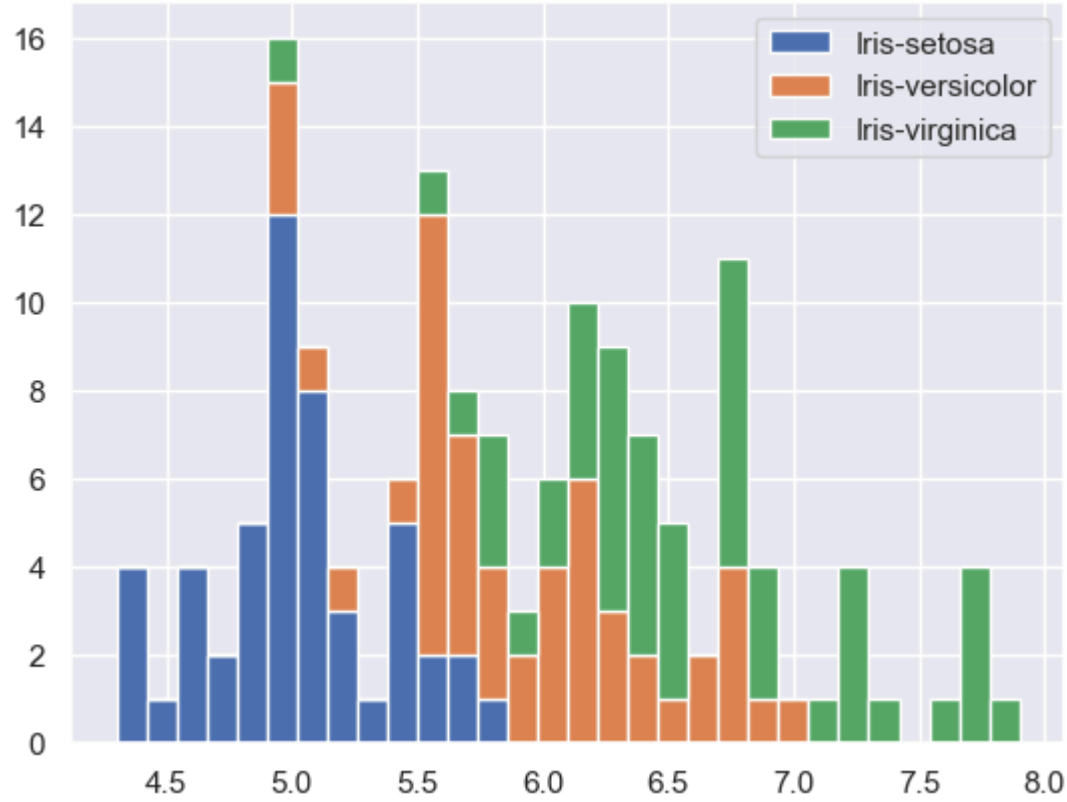




Stacked Histogram

In [179…
```
iris['Species'] = iris['Species'].astype('category')
#iris.head()
```

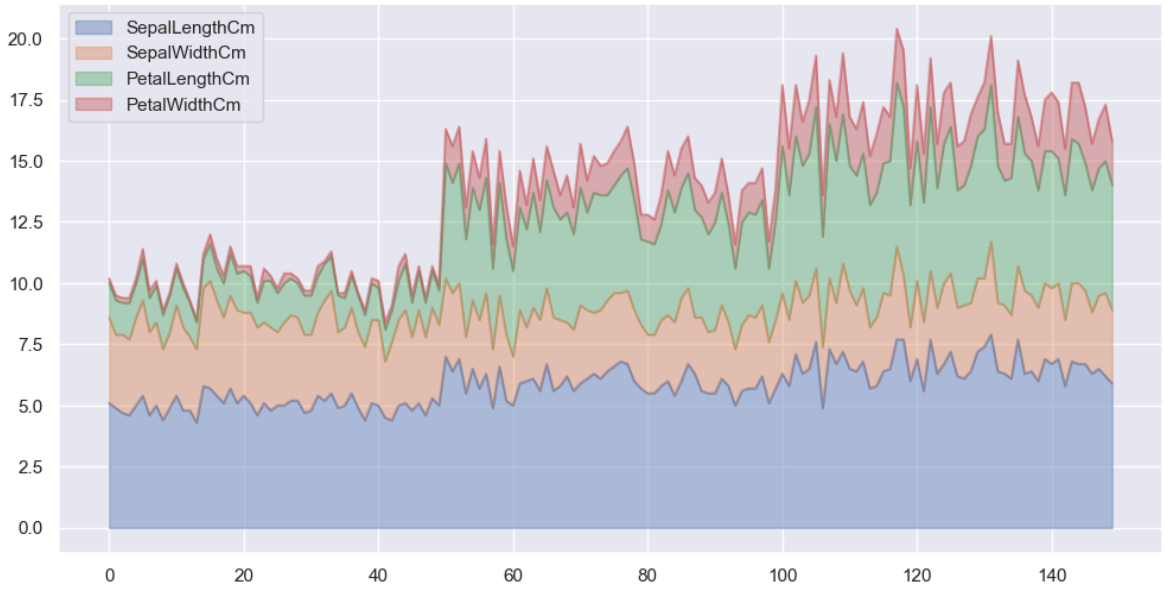
In [181…
```
list1=list()
mylabels=list()
for gen in iris.Species.cat.categories:
    list1.append(iris[iris.Species==gen].SepalLengthCm)
    mylabels.append(gen)

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



Area Plot:

In [184…

```
#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'])
plt.show()
```



Distplot:

In [187…

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

```
plt.show()
```