Iris Flower Classsification

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import svm, datasets
```

In [64]:
```
pwd
```

Out[64]:
```
'C:\\Users\\Shivv'
```

In [2]:
```
df = pd.read_csv('C:\\Users\\Shivv\\iris.data')
```

In [4]:
```
df.set_axis(['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm','Species'],
            axis = 'columns', inplace = True)

```

In [67]:
```
df.head()
```

Out[67]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [68]:
```
1 df.tail()
```

Out[68]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

In [69]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149 entries, 0 to 148
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   SepalLengthCm  149 non-null    float64
 1   SepalWidthCm   149 non-null    float64
 2   PetalLengthCm  149 non-null    float64
 3   PetalWidthCm   149 non-null    float64
 4   Species        149 non-null    object 
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
```

In [70]:
```
1 df.nunique()
```

Out[70]:

```
SepalLengthCm    35
SepalWidthCm     23
PetalLengthCm    43
PetalWidthCm     22
Species           3
dtype: int64
```

In [71]:
```
1 df.describe()
```

Out[71]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	149.000000	149.000000	149.000000	149.000000
mean	5.848322	3.051007	3.774497	1.205369
std	0.828594	0.433499	1.759651	0.761292
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.400000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [72]:
```
1 df.keys
```

Out[72]:

```
<bound method NDFrame.keys of      SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm         Species
0              4.9           3.0            1.4           0.2     Iris-setosa
1              4.7           3.2            1.3           0.2     Iris-setosa
2              4.6           3.1            1.5           0.2     Iris-setosa
3              5.0           3.6            1.4           0.2     Iris-setosa
4              5.4           3.9            1.7           0.4     Iris-setosa
..             ...           ...            ...           ...             ...
144            6.7           3.0            5.2           2.3  Iris-virginica
145            6.3           2.5            5.0           1.9  Iris-virginica
146            6.5           3.0            5.2           2.0  Iris-virginica
147            6.2           3.4            5.4           2.3  Iris-virginica
148            5.9           3.0            5.1           1.8  Iris-virginica

[149 rows x 5 columns]>
```

In [73]:
```
1 df.values
```

Out[73]:
```
array([[4.9, 3.0, 1.4, 0.2, 'Iris-setosa'],
       [4.7, 3.2, 1.3, 0.2, 'Iris-setosa'],
       [4.6, 3.1, 1.5, 0.2, 'Iris-setosa'],
       [5.0, 3.6, 1.4, 0.2, 'Iris-setosa'],
       [5.4, 3.9, 1.7, 0.4, 'Iris-setosa'],
       [4.6, 3.4, 1.4, 0.3, 'Iris-setosa'],
       [5.0, 3.4, 1.5, 0.2, 'Iris-setosa'],
       [4.4, 2.9, 1.4, 0.2, 'Iris-setosa'],
       [4.9, 3.1, 1.5, 0.1, 'Iris-setosa'],
       [5.4, 3.7, 1.5, 0.2, 'Iris-setosa'],
       [4.8, 3.4, 1.6, 0.2, 'Iris-setosa'],
       [4.8, 3.0, 1.4, 0.1, 'Iris-setosa'],
       [4.3, 3.0, 1.1, 0.1, 'Iris-setosa'],
       [5.8, 4.0, 1.2, 0.2, 'Iris-setosa'],
       [5.7, 4.4, 1.5, 0.4, 'Iris-setosa'],
       [5.4, 3.9, 1.3, 0.4, 'Iris-setosa'],
       [5.1, 3.5, 1.4, 0.3, 'Iris-setosa'],
       [5.7, 3.8, 1.7, 0.3, 'Iris-setosa'],
       [5.1, 3.8, 1.5, 0.3, 'Iris-setosa'],
```

In [74]:
```
1 df.shape
```

Out[74]:
```
(149, 5)
```

In [75]:
```
1 df.isna()
```

Out[75]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
144	False	False	False	False	False
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False

149 rows × 5 columns

In [76]:
```
1 df.isna().sum()
```

Out[76]:
```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [77]:
```
1 df['Species'].value_counts()
```

Out[77]:
```
Iris-versicolor    50
Iris-virginica     50
Iris-setosa        49
Name: Species, dtype: int64
```

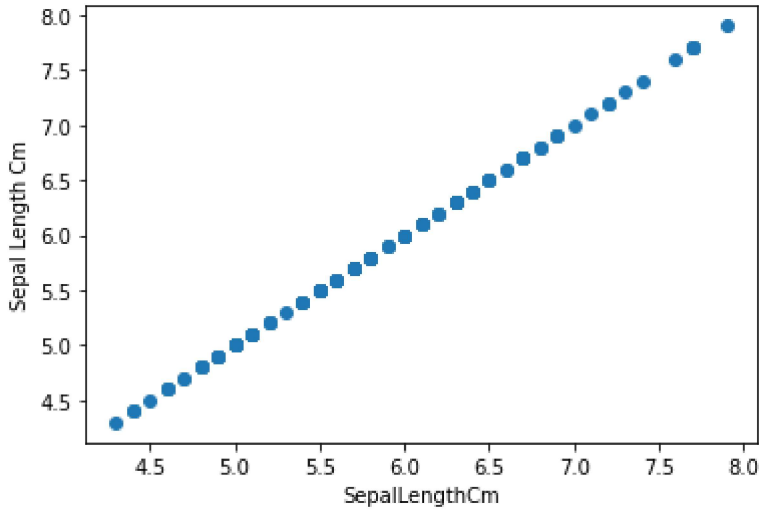
In [78]:
```
1 df.columns
```

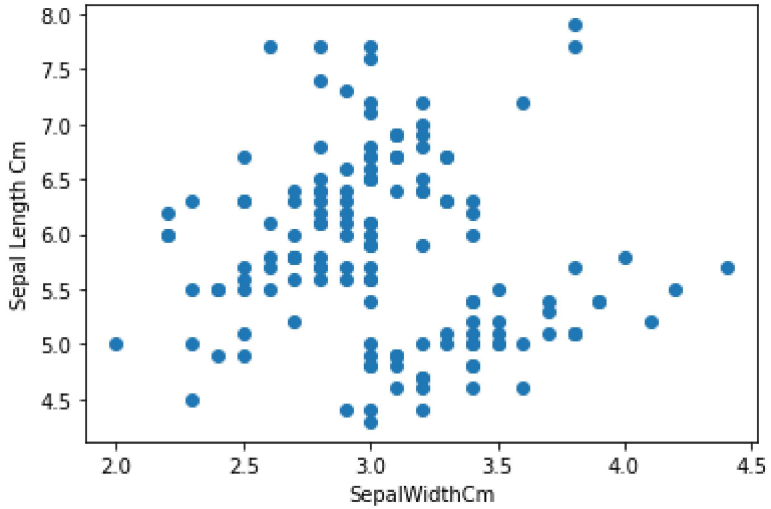
Out[78]:
```
Index(['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

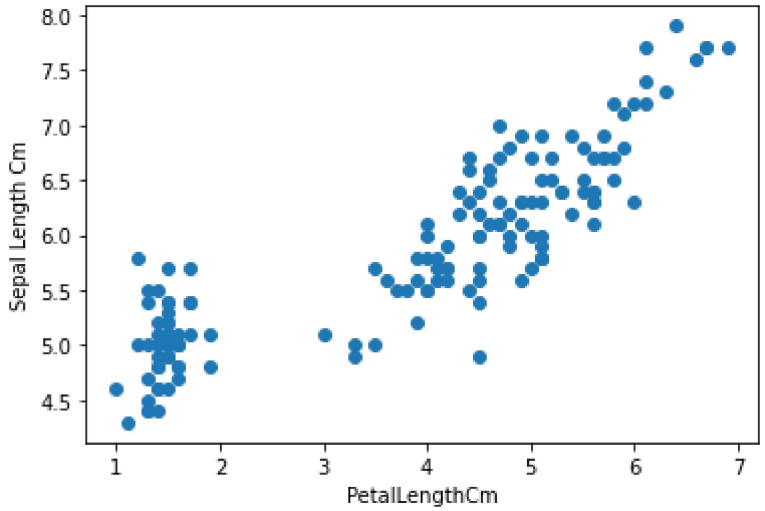
Scatter Plot

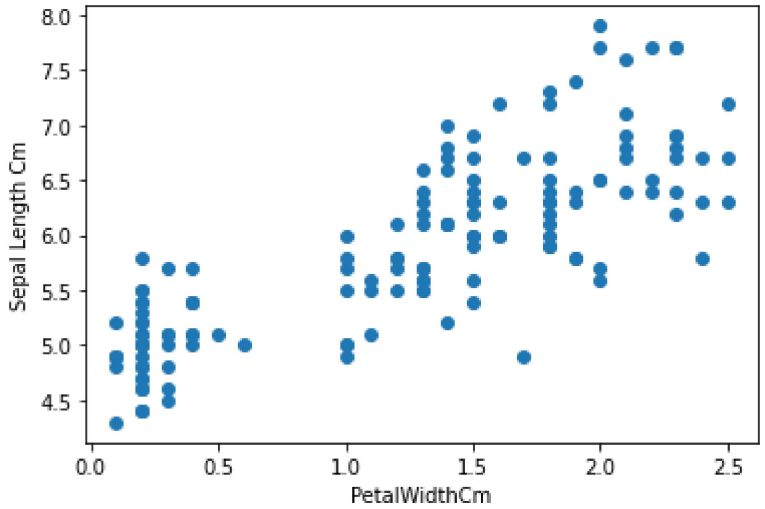
In [60]:

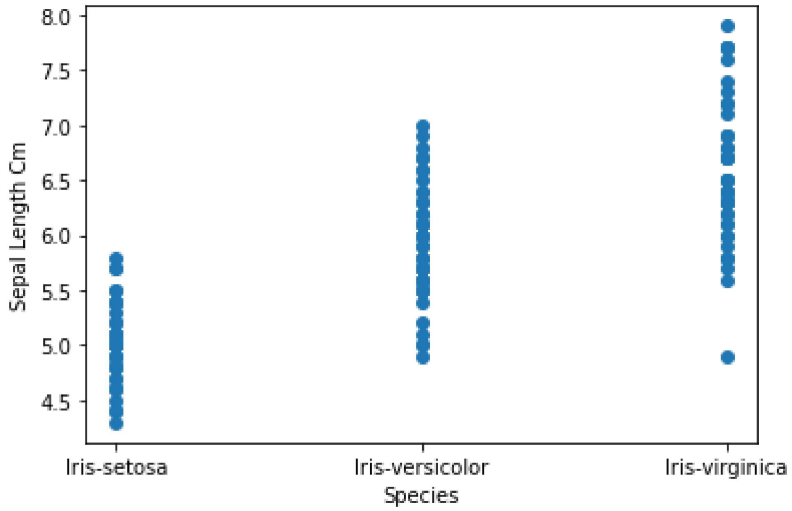
```
for i in df.columns:
    plt.scatter(df[str(i)],df['SepalLengthCm'])
    plt.xlabel(i)
    plt.ylabel("Sepal Length Cm")
    plt.show()
```






Sepal Length Cm
PetalLengthCm
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
1
2
3
4
5
6
7

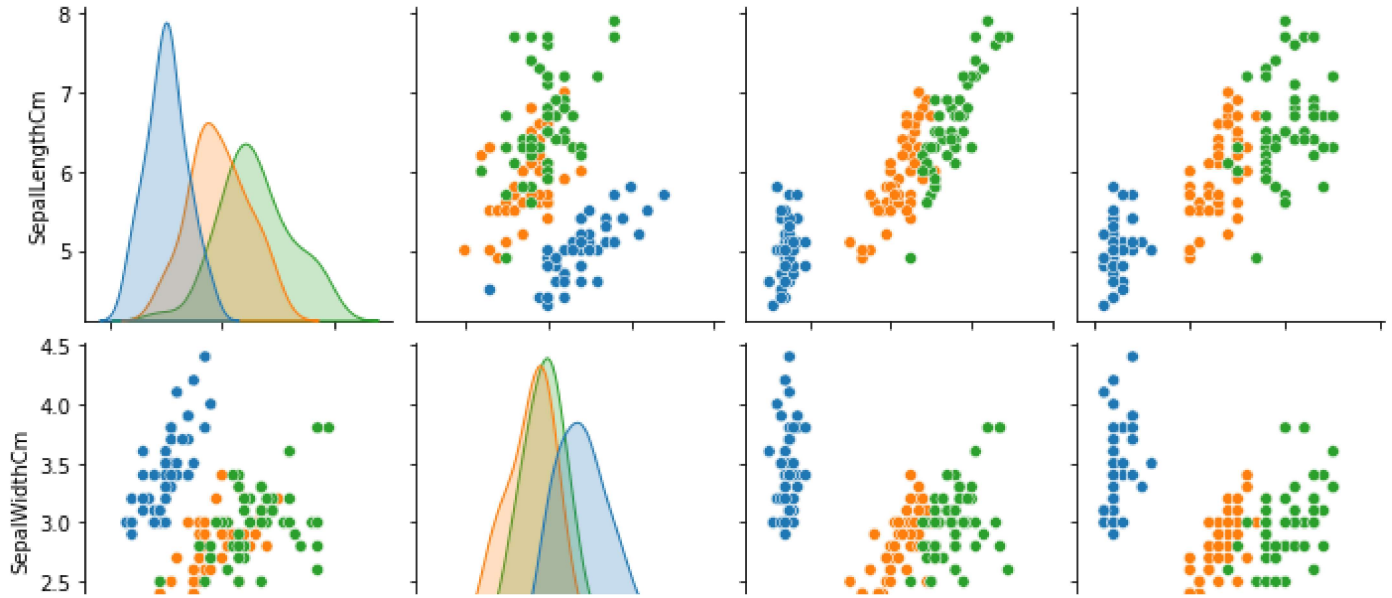

Sepal Length Cm
PetalWidthCm
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
0.0
0.5
1.0
1.5
2.0
2.5



Pair Plot

In [5]:
```
1 sns.pairplot(df, hue = 'Species')
```

Out[5]: <seaborn.axisgrid.PairGrid at 0x12988d38070>



Type *Markdown* and LaTeX: α^2

In [25]:
```
1 df.corr()
```

Out[25]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.103784	0.871283	0.816971
SepalWidthCm	-0.103784	1.000000	-0.415218	-0.350733
PetalLengthCm	0.871283	-0.415218	1.000000	0.962314
PetalWidthCm	0.816971	-0.350733	0.962314	1.000000

In [26]:
```
1 df.columns
```

Out[26]:
```
Index(['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

In [59]:
```
1  # col = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']
2  # plt.figure(figsize= (35,5))
3  # i = 1
4  # for j in col:
5  #     plt.subplot(1,10,i)
6  #     sns.distplot(df[j])
7  #     i = i+1
8  #     plt.tight_layout()
9  #     plt.show()
10
```

In [44]:
```
1 # plt.figure(figsize=(10,10))
2 # sns.heatmap(df.corr(), cmap = 'Blue', annot = True)
```

In [32]:
```
1 df.describe()
```

Out[32]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	149.000000	149.000000	149.000000	149.000000
mean	5.848322	3.051007	3.774497	1.205369
std	0.828594	0.433499	1.759651	0.761292
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.400000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

FacetGrit

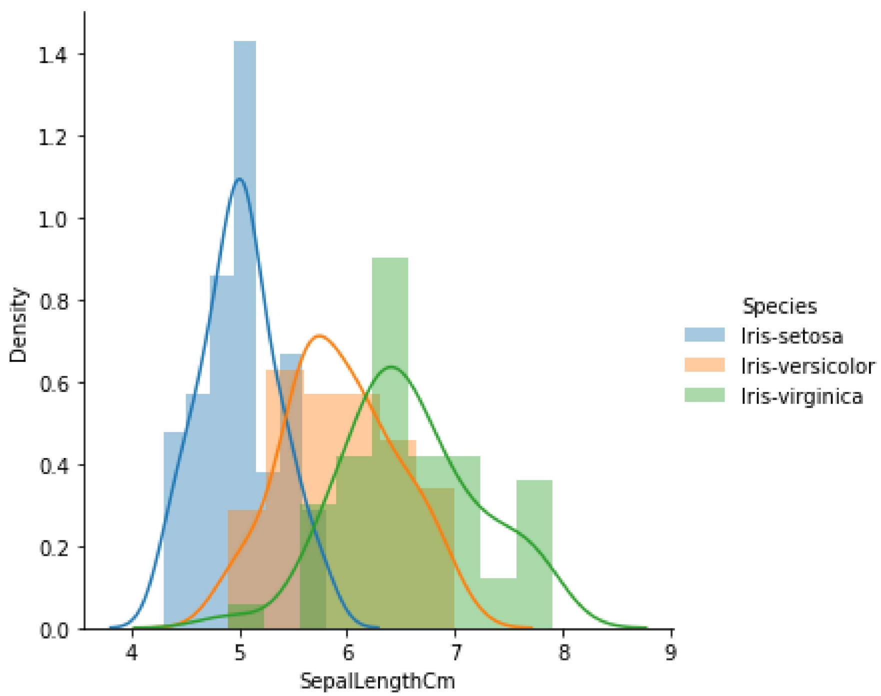
In [35]:

```
sns.FacetGrid(df, hue = "Species", height = 5).map(sns.distplot,"SepalLengthCm").add_legend()
```

```
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[35]:

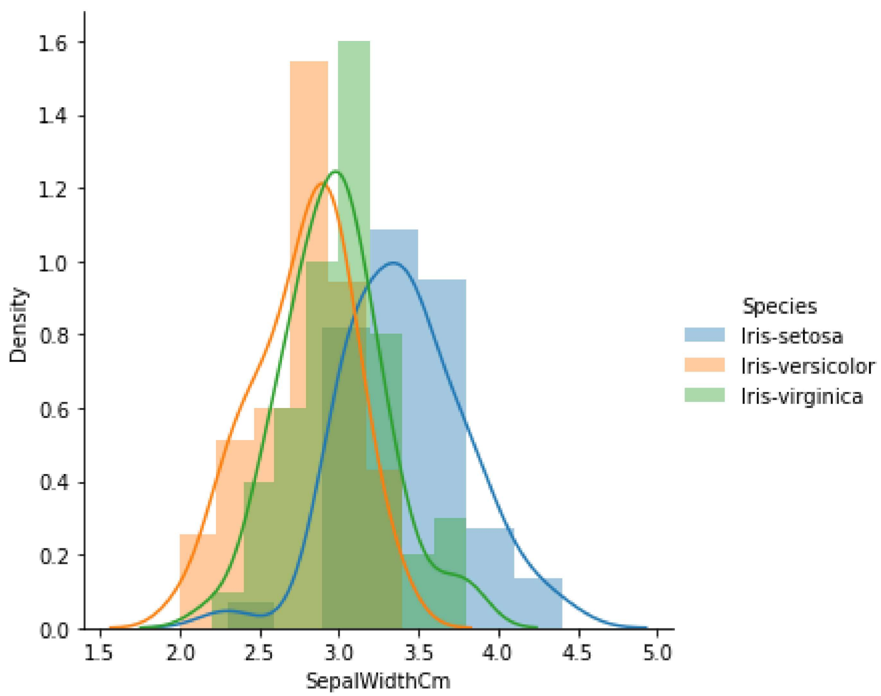
```
<seaborn.axisgrid.FacetGrid at 0x25035bf8d30>
```



In [36]:
```
sns.FacetGrid(df, hue = "Species", height = 5).map(sns.distplot,"SepalWidthCm").add_legend()
```

```
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

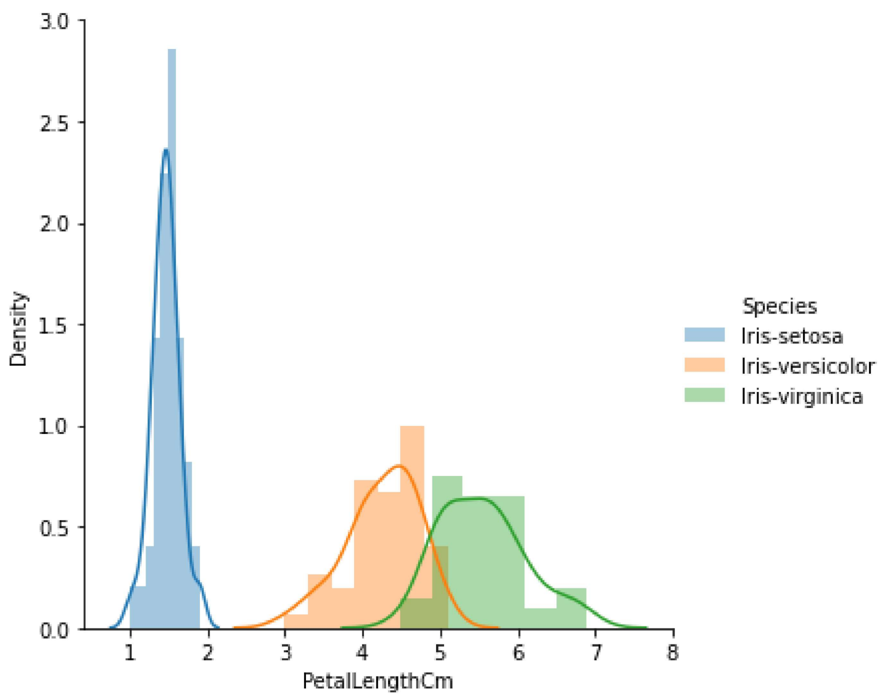
Out[36]: <seaborn.axisgrid.FacetGrid at 0x25038f04d00>



In [37]:
```
1 sns.FacetGrid(df, hue = "Species", height = 5).map(sns.distplot,"PetalLengthCm").add_legend()
```

```
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

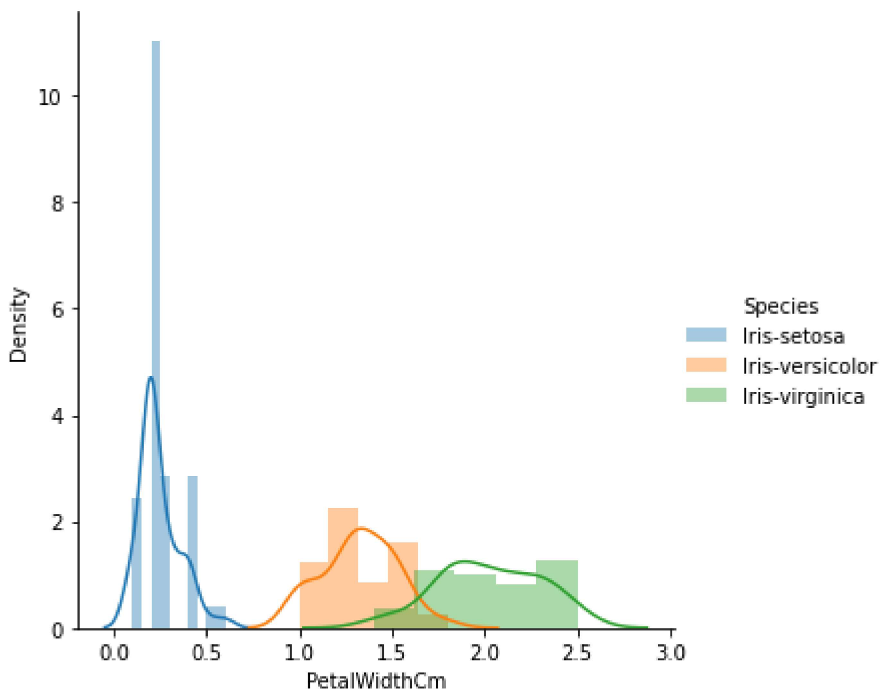
Out[37]: <seaborn.axisgrid.FacetGrid at 0x25038fa7730>



In [38]:
```
sns.FacetGrid(df, hue = "Species", height = 5).map(sns.distplot,"PetalWidthCm").add_legend()
```

```
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shivv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

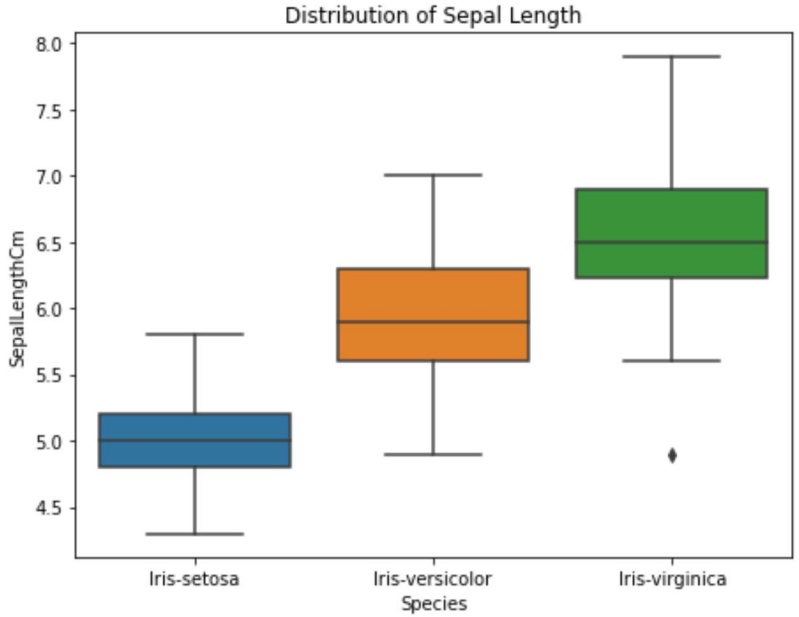
Out[38]: <seaborn.axisgrid.FacetGrid at 0x250390404f0>

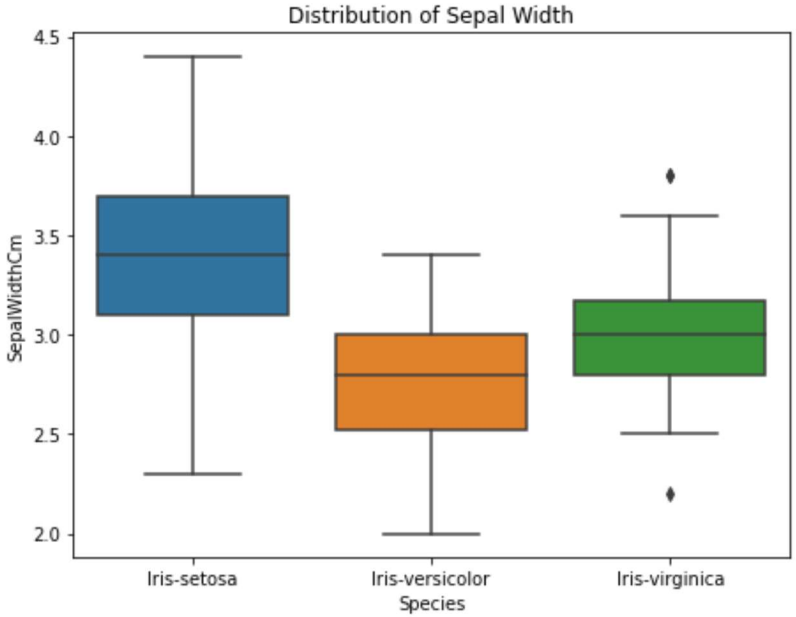


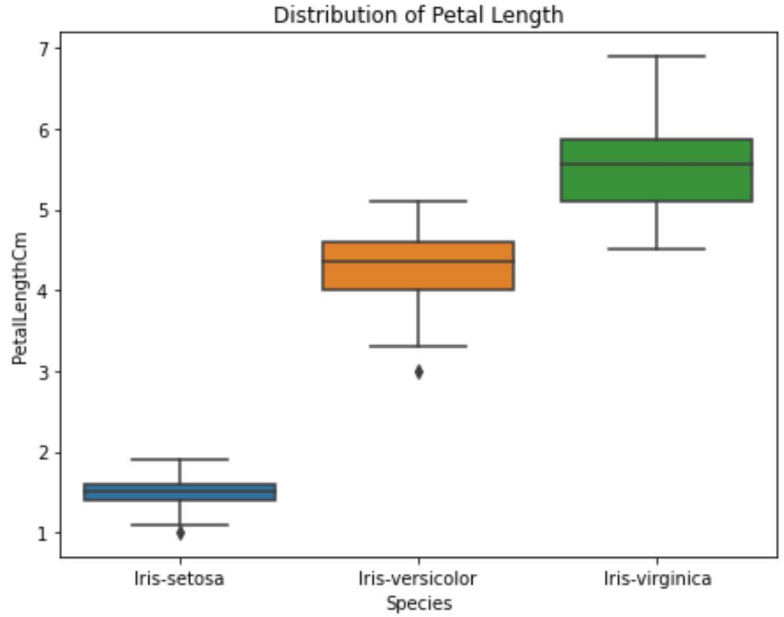
SubPlots

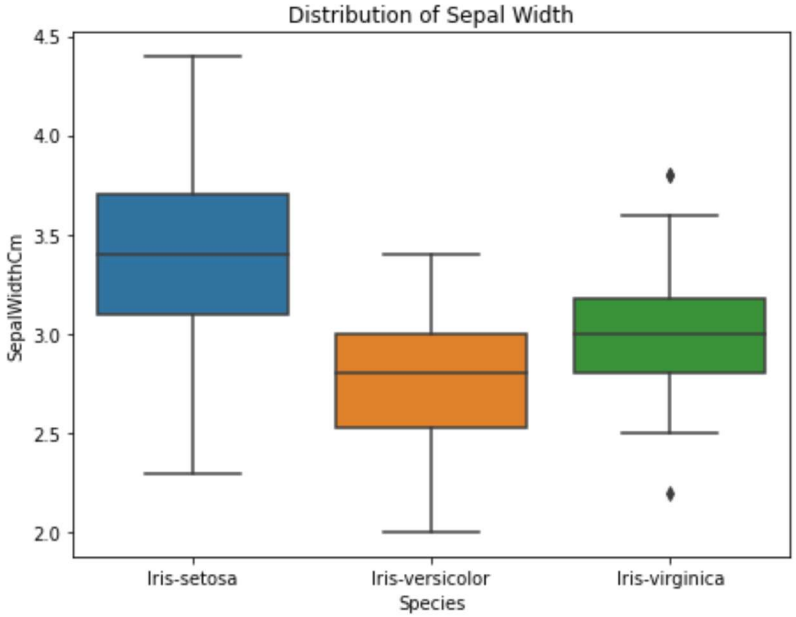
In [42]:

```
fig, axes = plt.subplots(2,2, figsize=(16,12))
axes[0,0].set_title("Distribution of Sepal Length")
sns.boxplot(y = "SepalLengthCm", x ="Species", data= df , orient = 'v', ax = axes[0,0])
axes[0,1].set_title("Distribution of Sepal Width")
sns.boxplot(y = "SepalWidthCm", x ="Species", data= df , orient = 'v', ax = axes[0,1])
axes[1,0].set_title("Distribution of Petal Length")
sns.boxplot(y = "PetalLengthCm", x ="Species", data= df , orient = 'v', ax = axes[1,0])
axes[1,1].set_title("Distribution of Sepal Width")
sns.boxplot(y = "SepalWidthCm", x ="Species", data= df , orient = 'v', ax = axes[1,1])
plt.show()
```


Distribution of Sepal Length
SepalLengthCm
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
Iris-setosa
Iris-versicolor
Iris-virginica
Species


Distribution of Sepal Width
SepalWidthCm
4.5
4.0
3.5
3.0
2.5
2.0
Iris-setosa
Iris-versicolor
Iris-virginica
Species

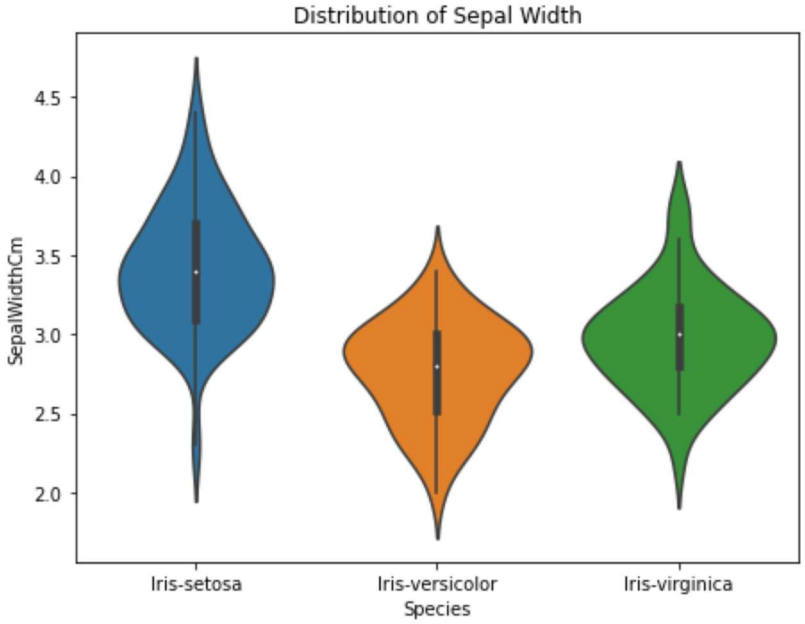

Distribution of Petal Length
PetalLengthCm
7
6
5
4
3
2
1
Iris-setosa
Iris-versicolor
Iris-virginica
Species

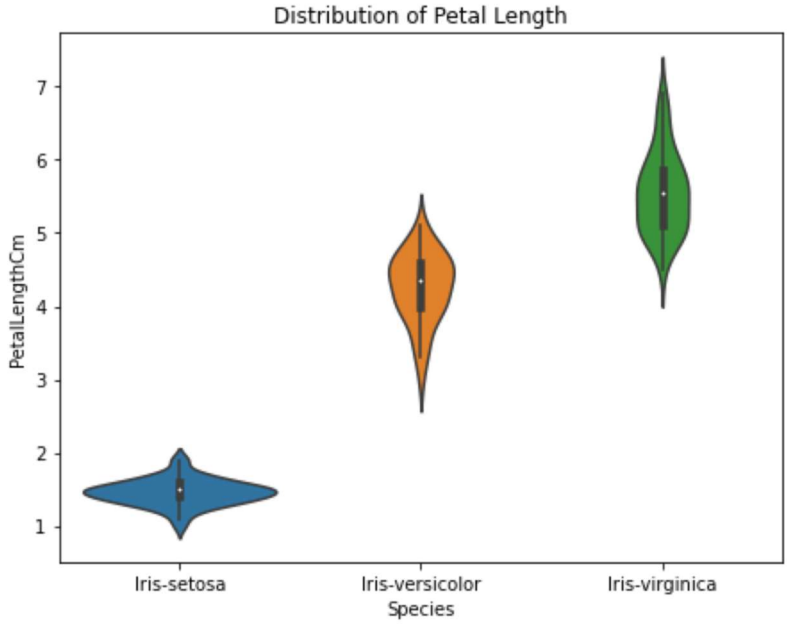

Distribution of Sepal Width
SepalWidthCm
4.5
4.0
3.5
3.0
2.5
2.0
Iris-setosa
Iris-versicolor
Iris-virginica
Species

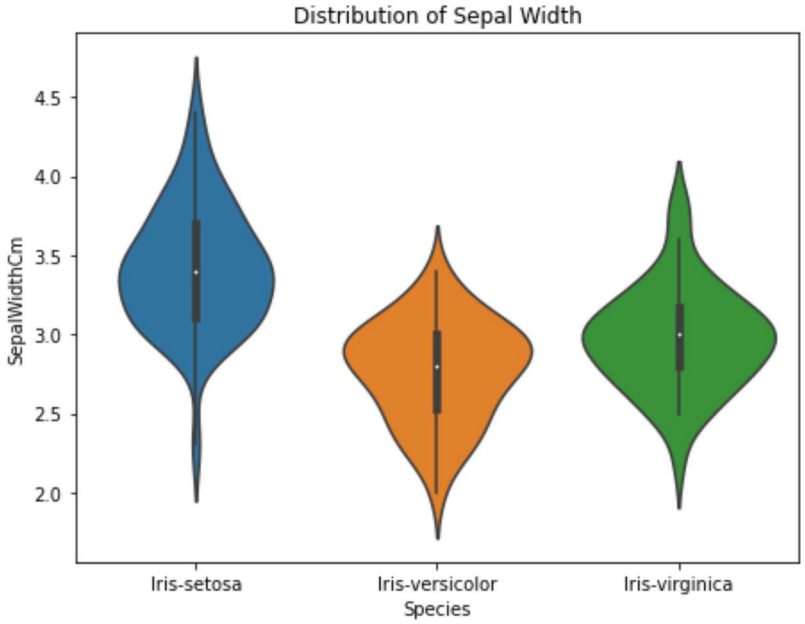
ViolinPlot

In [44]:

```
fig, axes = plt.subplots(2,2, figsize=(16,12))
axes[0,0].set_title("Distribution of Sepal Length")
sns.violinplot(y = "SepalLengthCm", x ="Species", data= df , orient = 'v', ax = axes[0,0])
axes[0,1].set_title("Distribution of Sepal Width")
sns.violinplot(y = "SepalWidthCm", x ="Species", data= df , orient = 'v', ax = axes[0,1])
axes[1,0].set_title("Distribution of Petal Length")
sns.violinplot(y = "PetalLengthCm", x ="Species", data= df , orient = 'v', ax = axes[1,0])
axes[1,1].set_title("Distribution of Sepal Width")
sns.violinplot(y = "SepalWidthCm", x ="Species", data= df , orient = 'v', ax = axes[1,1])
plt.show()
```



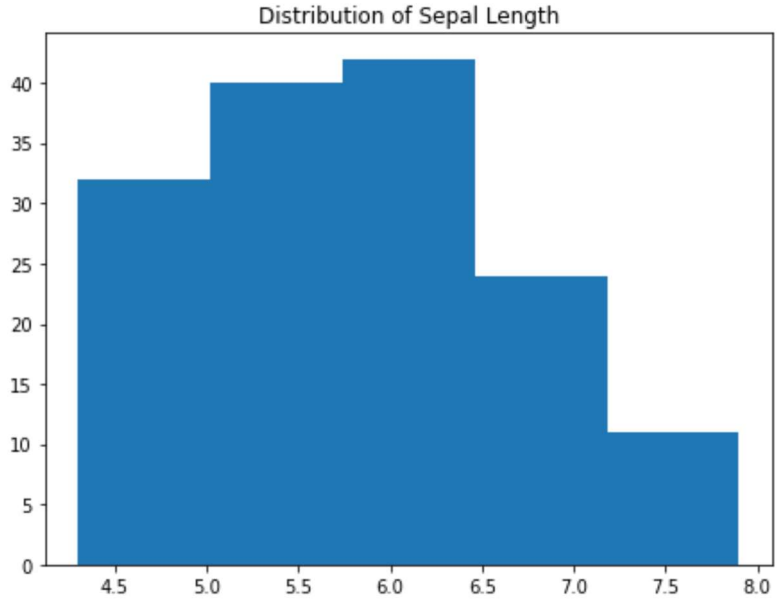


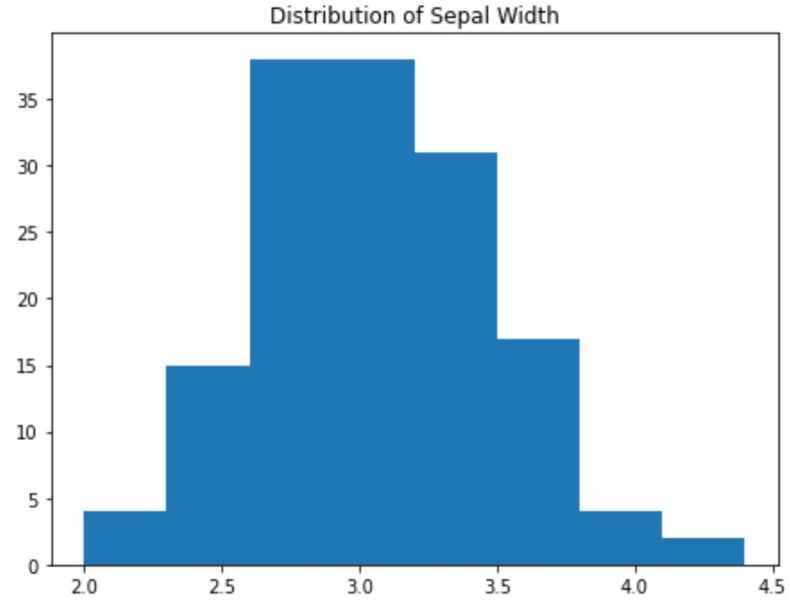


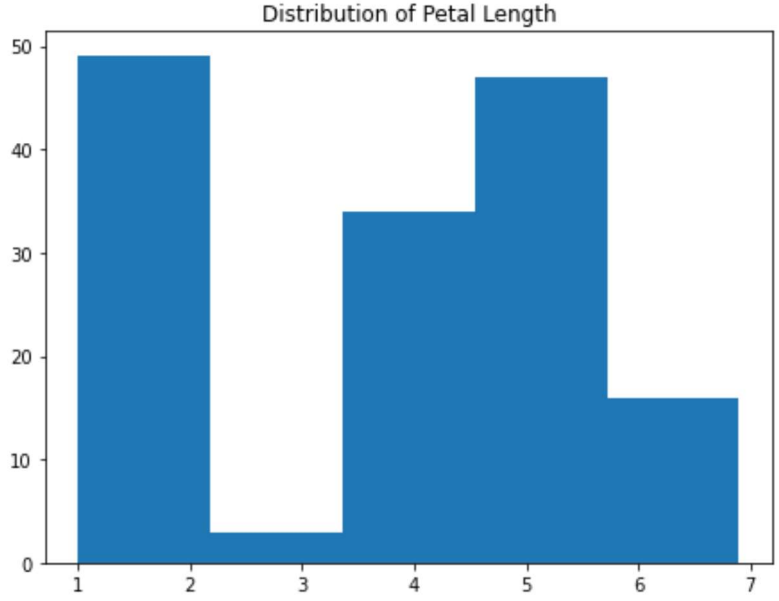
Histogram Plot

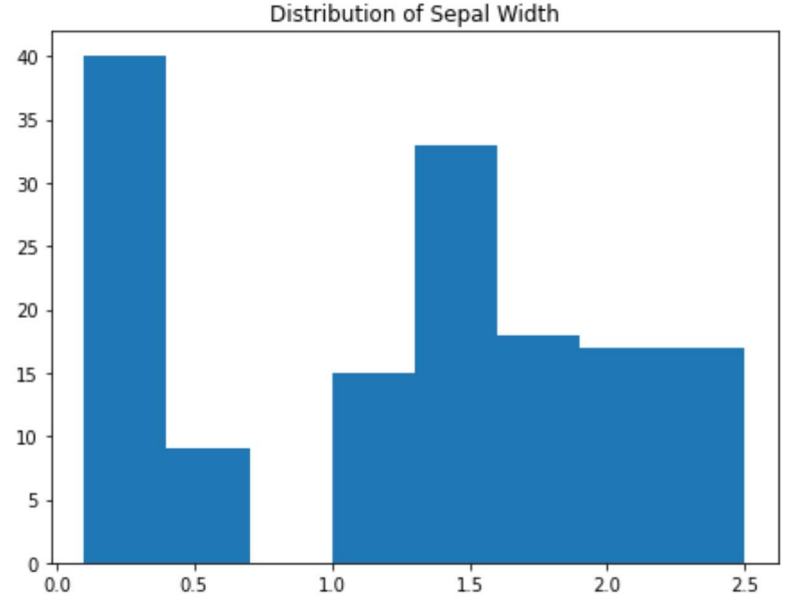
In [13]:

```
fig, axes = plt.subplots(2,2, figsize=(16,12))
axes[0,0].set_title("Distribution of Sepal Length")
axes[0,0].hist(df["SepalLengthCm"], bins = 5);
axes[0,1].set_title("Distribution of Sepal Width")
axes[0,1].hist(df["SepalWidthCm"], bins = 8);
axes[1,0].set_title("Distribution of Petal Length")
axes[1,0].hist(df["PetalLengthCm"], bins = 5);
axes[1,1].set_title("Distribution of Sepal Width")
axes[1,1].hist(df["PetalWidthCm"], bins = 8);
plt.show()
```


Distribution of Sepal Length
0
5
10
15
20
25
30
35
40
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

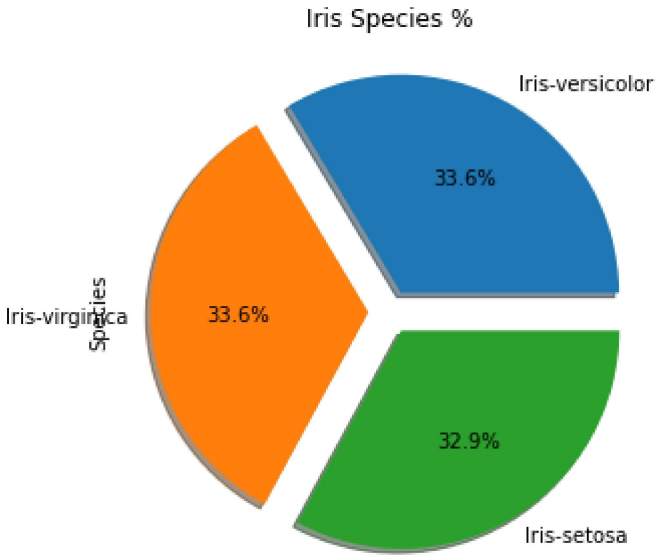

Distribution of Sepal Width
0
5
10
15
20
25
30
35
2.0
2.5
3.0
3.5
4.0
4.5


Distribution of Petal Length
0
10
20
30
40
50
1
2
3
4
5
6
7


Distribution of Sepal Width
0
5
10
15
20
25
30
35
40
0.0
0.5
1.0
1.5
2.0
2.5

Pie Plot

In [38]:
```
import pandas as pd
import matplotlib.pyplot as plt
plt.figure(figsize= (50,15))
axes = plt.subplot(111)
df['Species'].value_counts().plot.pie(explode=[0.1,0.1,0.1],autopct = '%1.1f%%', shadow = True,figsize= (
plt.title("Iris Species %")
plt.show()
```



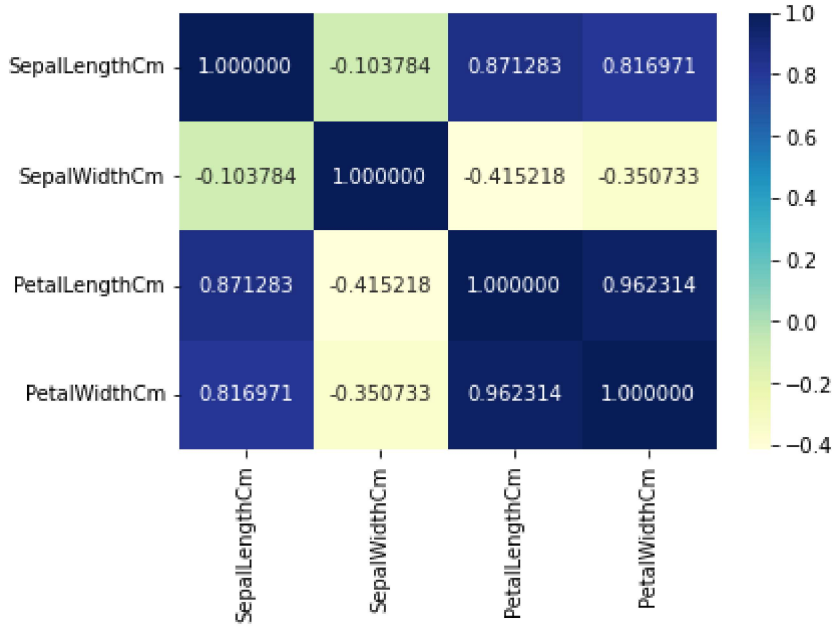
In [62]:
```
df.corr()
```

Out[62]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.103784	0.871283	0.816971
SepalWidthCm	-0.103784	1.000000	-0.415218	-0.350733
PetalLengthCm	0.871283	-0.415218	1.000000	0.962314
PetalWidthCm	0.816971	-0.350733	0.962314	1.000000

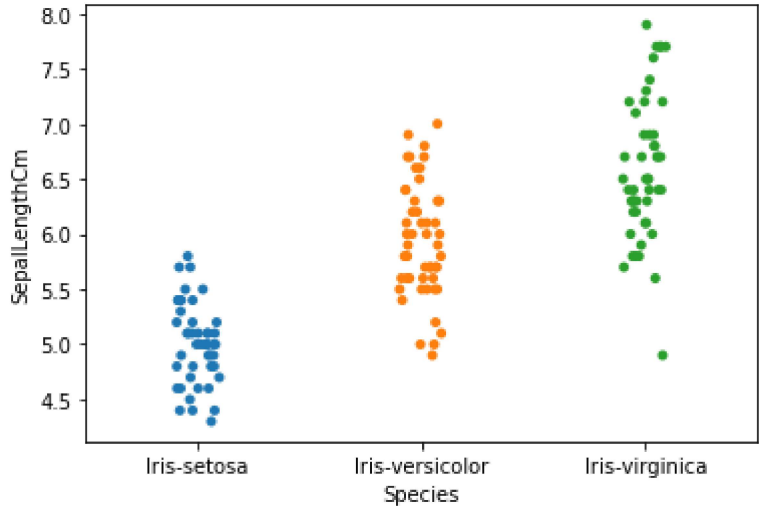
Heat Map

In [40]:
```
sns.heatmap(df.corr(), cmap = "YlGnBu", annot = True, fmt = "f")
plt.show()
```



Strip Plot

In [41]:
```
sns.stripplot( y = 'SepalLengthCm', x = 'Species', data = df)
plt.show()
```



In [45]:
```
#seperat features and target
data = df.values
X = data[:,0:4]
Y = data[:,4]
```

Train and Test Data

In [47]:
```
#split the data to train and test
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2)
```

Support Vector Machine

In [51]:
```
#support vector machine algorithm
from sklearn.svm import SVC
svn = SVC()
svn.fit(X_train, Y_train)
```

Out[51]:
```
SVC()
```

In [54]:
```
#predict from the dataset
predictions = svn.predict(X_test)

```

accuracy of data

In [55]:
```
from sklearn.metrics import accuracy_score
accuracy_score(Y_test, predictions)
```

Out[55]:
```
0.8666666666666667
```

Classification Report

In [56]:
```
#distibuted the classification report
from sklearn.metrics import classification_report
classification_report(Y_test, predictions)
```

Out[56]:
```
'                 precision    recall  f1-score   support\n\n    Iris-setosa       1.00      1.00      1.00
6\nIris-versicolor       0.77      0.91      0.83        11\n Iris-virginica       0.91      0.77      0.83
13\n\n       accuracy                           0.87        30\n      macro avg       0.89      0.89      0.
89        30\n   weighted avg       0.88      0.87      0.87        30\n'
```

Test Model

In [58]:
```
#Test the model
x_new = np.array([[2,3,5,5],[1,2,3.8,0],[5.3,2.5,4.6,1.9]])
#prediction of the species from the input vector
predictions = svn.predict(x_new)
print("Predictions of Species: {}".format(predictions))
```

```
Predictions of Species: ['Iris-virginica' 'Iris-versicolor' 'Iris-versicolor']
```

Thank You!

In [ ]:
```

```