IRIS DATASET ANALYSIS

Dataset Information

The dataset contains 3 classes of 50 instances each,where each class refers to a type of iris plant.One class is linerly seperable from the other two.

Attribute Information: 1.sepal length in cm 2.sepal width in cm 3.petal length in cm 4.petal width in cm 5.classes are:- Iris setosa ;Iris versicolor ;Iris virginica

Import Modules

In [1]:

```
import pandas as pd
import numpy as np
import os
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
data = pd.read_csv("C:/Users/priya/OneDrive/Desktop/Iris.csv")
```

In [3]:

```
#shape of dataset.
data.shape
```

Out[3]:

```
(150, 6)
```

In [4]:

```
#description about stats of dataset.
data.describe()
```

Out[4]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [5]:

```
#heads of dataset.
data.head(150)
```

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]:

```
#tail of dataset.
data.tail()
```

Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
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

In [7]:

```
#columns of dataset.
data.columns
```

Out[7]:

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

In [8]:

```
#basic information of dataset.
data.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 [9]:

```
Iris_species = data.groupby('Species')
```

In [10]:

```
#description of stats among the each species class.
Iris_species.describe().T
```

Out[10]:

	Species	Iris-setosa	Iris-versicolor	Iris-virginica
Id	count	50.000000	50.000000	50.000000
	mean	25.500000	75.500000	125.500000
	std	14.577380	14.577380	14.577380
	min	1.000000	51.000000	101.000000
	25%	13.250000	63.250000	113.250000
	50%	25.500000	75.500000	125.500000
	75%	37.750000	87.750000	137.750000
	max	50.000000	100.000000	150.000000
SepalLengthCm	count	50.000000	50.000000	50.000000
	mean	5.006000	5.936000	6.588000
	std	0.352490	0.516171	0.635880
	min	4.300000	4.900000	4.900000
	25%	4.800000	5.600000	6.225000
	50%	5.000000	5.900000	6.500000
	75%	5.200000	6.300000	6.900000
	max	5.800000	7.000000	7.900000
SepalWidthCm	count	50.000000	50.000000	50.000000
	mean	3.418000	2.770000	2.974000
	std	0.381024	0.313798	0.322497
	min	2.300000	2.000000	2.200000
	25%	3.125000	2.525000	2.800000
	50%	3.400000	2.800000	3.000000
	75%	3.675000	3.000000	3.175000
	max	4.400000	3.400000	3.800000
PetalLengthCm	count	50.000000	50.000000	50.000000
	mean	1.464000	4.260000	5.552000
	std	0.173511	0.469911	0.551895
	min	1.000000	3.000000	4.500000
	25%	1.400000	4.000000	5.100000
	50%	1.500000	4.350000	5.550000
	75%	1.575000	4.600000	5.875000
	max	1.900000	5.100000	6.900000
PetalWidthCm	count	50.000000	50.000000	50.000000
	mean	0.244000	1.326000	2.026000
	std	0.107210	0.197753	0.274650
	min	0.100000	1.000000	1.400000
	25%	0.200000	1.200000	1.800000
	50%	0.200000	1.300000	2.000000
	75%	0.300000	1.500000	2.300000
	max	0.600000	1.800000	2.500000

In [11]:

```
#check for null values.
data.isnull().sum()
```

Out[11]:

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

Exploratory Data Analysis

Analysis of Dataset using Graph plotting

List of Graphs plot :-

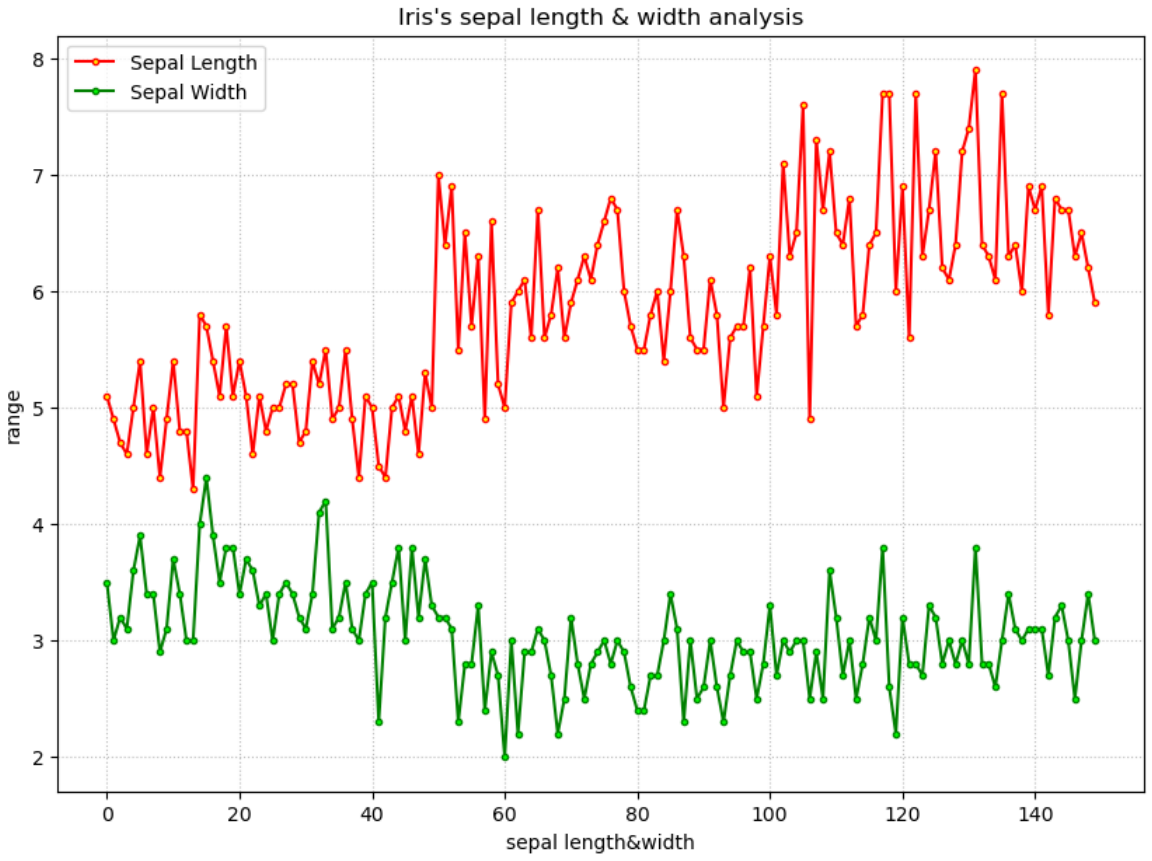
1. Line Graph
2. Histogram
3. Bar Graph
4. Scatter Plot
5. Pie Chart
6. Box Graph

LINE Graph

Line Graph of Sepal Length & Width

In [12]:

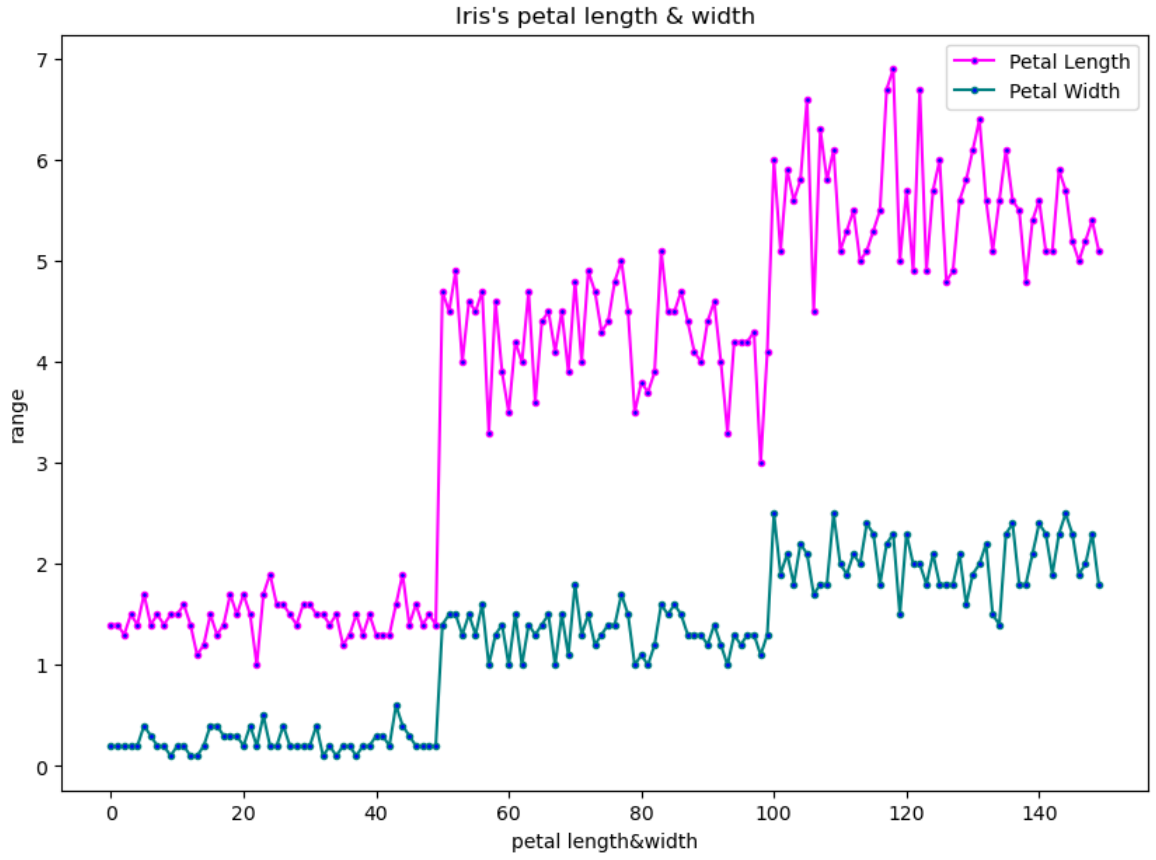
```
from matplotlib import rcParams
rcParams['figure.figsize'] = 10,7
plt.plot(data.SepalLengthCm, color="red",marker='.',markerfacecolor='yellow',label='Sepal Length')
plt.plot(data.SepalWidthCm, color="green",marker='.',markerfacecolor='lime',label='Sepal Width')
plt.title("Iris's sepal length & width analysis")
plt.xlabel("sepal length&width")
plt.ylabel("range")
plt.grid(True, color='silver',linestyle=":")
plt.legend()
plt.show()
```



Line Graph of Petal Length & Width

In [13]:

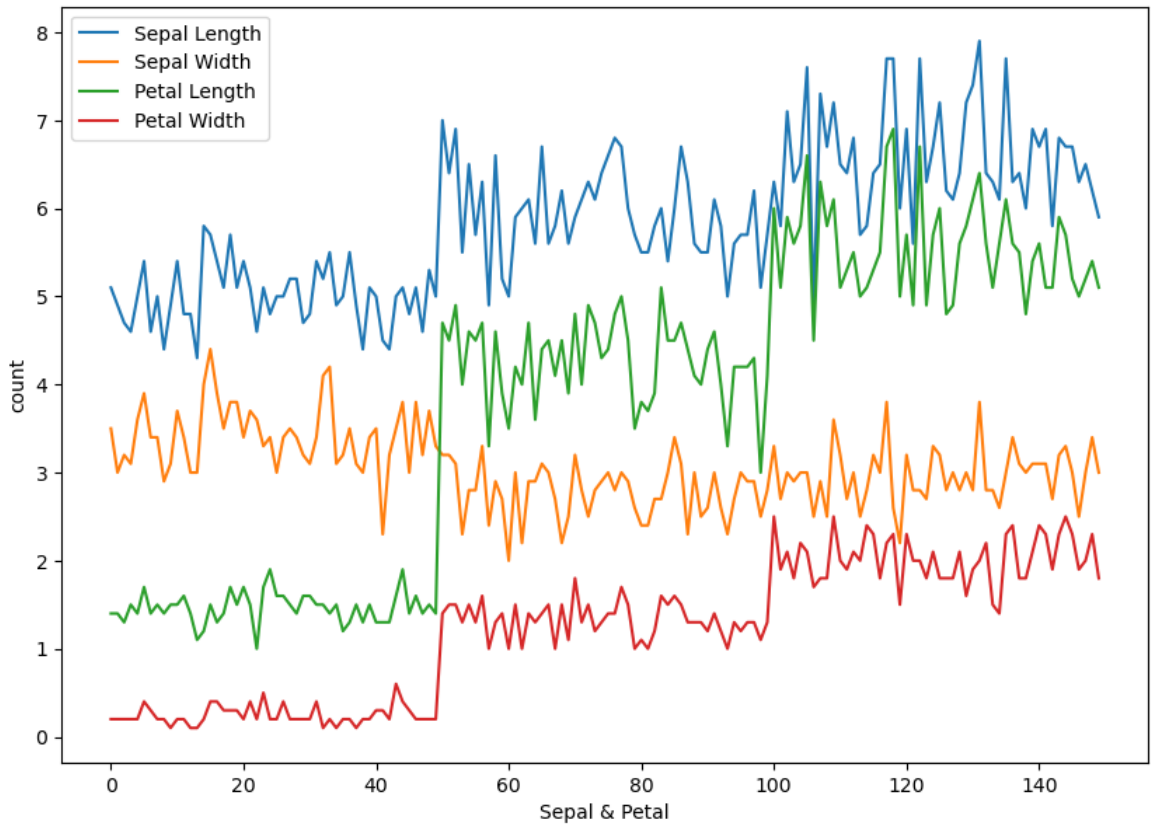
```
from matplotlib import rcParams
rcParams['figure.figsize']=10,7
plt.plot(data.PetalLengthCm,color="magenta",marker=".",markerfacecolor="blue",label='Petal Length')
plt.plot(data.PetalWidthCm,color="teal",marker=".",markerfacecolor="blue",label='Petal Width')
plt.title("Iris's petal length & width")
plt.xlabel('petal length&width')
plt.ylabel('range')
plt.legend()
plt.show()
```



Line Graph of Sepal & Petal

In [14]:

```
plt.plot(data.SepalLengthCm,label='Sepal Length')
plt.plot(data.SepalWidthCm,label='Sepal Width')
plt.plot(data.PetalLengthCm,label='Petal Length')
plt.plot(data.PetalWidthCm,label='Petal Width')
plt.xlabel('Sepal & Petal')
plt.ylabel('count')
plt.legend();
```

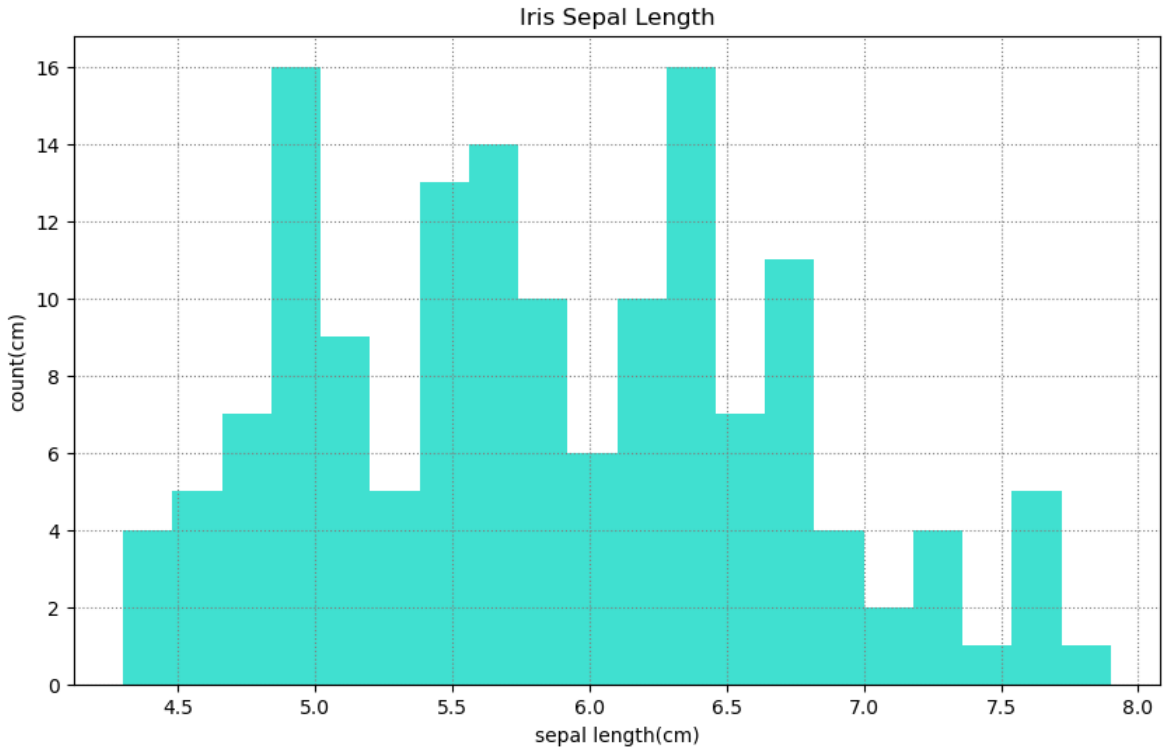


HISTOGRAM

Histogram graph for Sepal Length

In [15]:

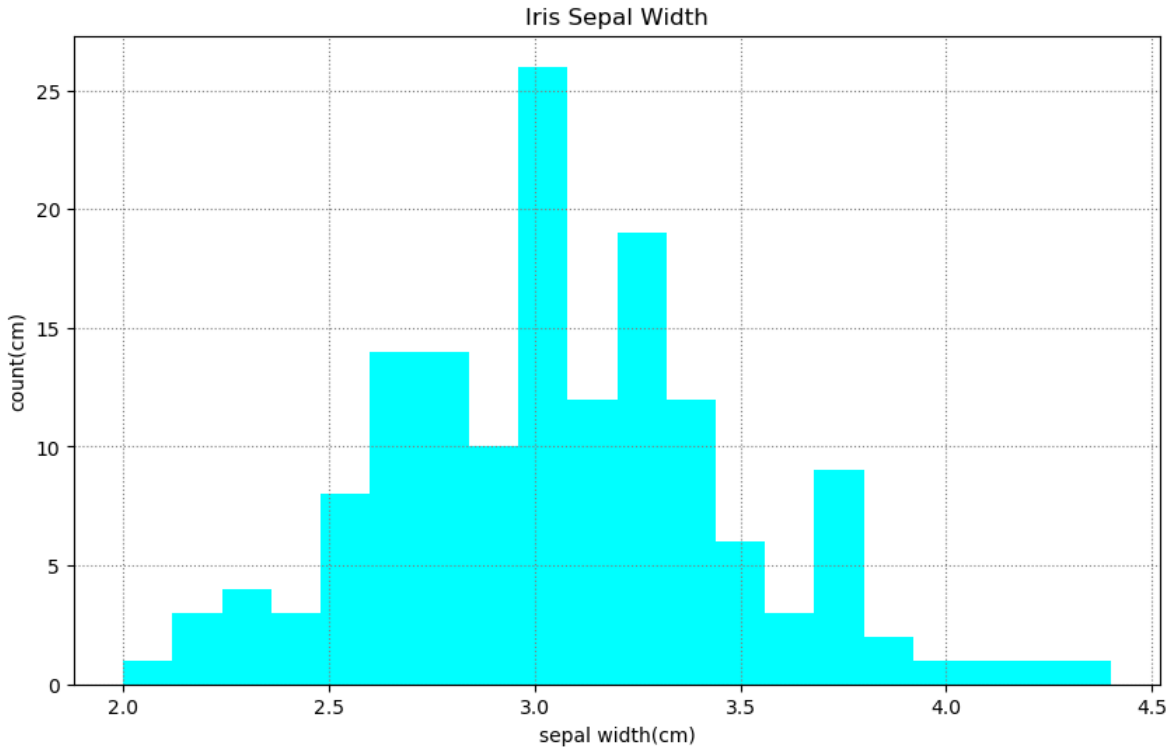
```
plt.figure(figsize=(10,6))
plt.hist(data.SepalLengthCm,color="turquoise",bins=20)
plt.title('Iris Sepal Length')
plt.xlabel('sepal length(cm)')
plt.ylabel('count(cm)')
plt.grid(True,color='grey',linestyle=':');
```



Histogram Graph for Sepal Width

In [16]:

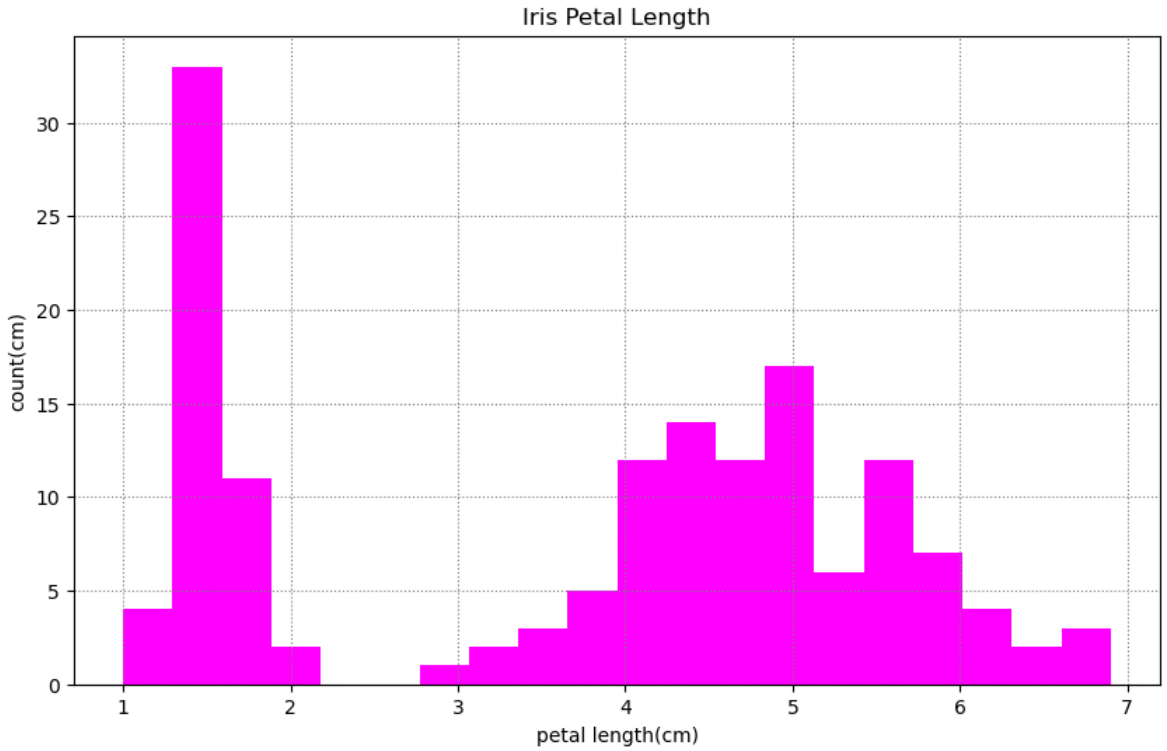
```
plt.figure(figsize=(10,6))
plt.hist(data.SepalWidthCm,color="aqua",bins=20)
plt.title('Iris Sepal Width')
plt.xlabel('sepal width(cm)')
plt.ylabel('count(cm)')
plt.grid(True,color='grey',linestyle=':');
```



Histogram Graph for Petal Length

In [17]:

```
plt.figure(figsize=(10,6))
plt.hist(data.PetalLengthCm,color="magenta",bins=20)
plt.title('Iris Petal Length')
plt.xlabel('petal length(cm)')
plt.ylabel('count(cm)')
plt.grid(True,color='grey',linestyle=':');
```



Histogram Graph for Petal Width

In [18]:

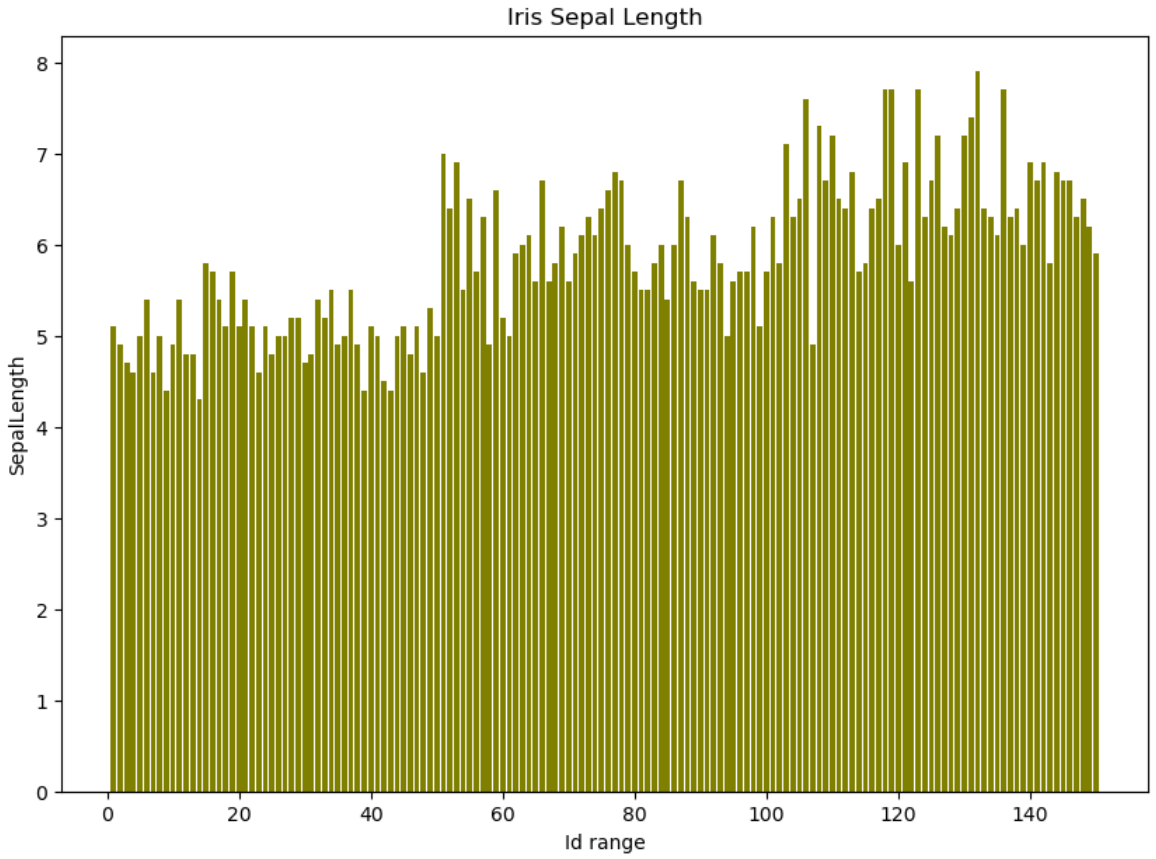
```
plt.figure(figsize=(10,6))
plt.hist(data.PetalWidthCm,color="red",bins=20)
plt.title('Iris Petal Width')
plt.xlabel('petal width(cm)')
plt.ylabel('count(cm)')
plt.grid(True,color='grey',linestyle=':');
```

BAR GRAPH

Bar Graph for Sepal Length

In [19]:

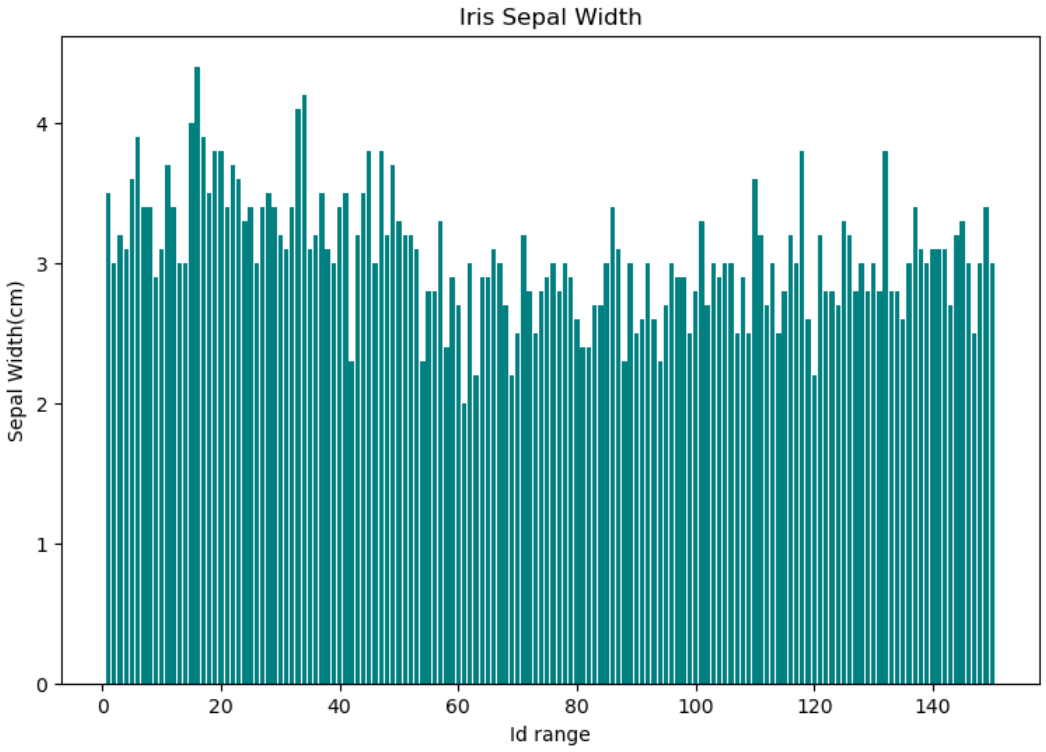
```
x=data['Id']
y=data['SepalLengthCm']
plt.bar(x,y,color='olive')
plt.xlabel("Id range")
plt.ylabel("SepalLength")
plt.title("Iris Sepal Length");
```



Bar Graph for Sepal width

In [20]:

```
plt.figure(figsize=(9,6))
plt.bar(data.Id,data.SepalWidthCm,color='teal')
plt.title('Iris Sepal Width')
plt.ylabel('Sepal Width(cm)')
plt.xlabel('Id range');
```

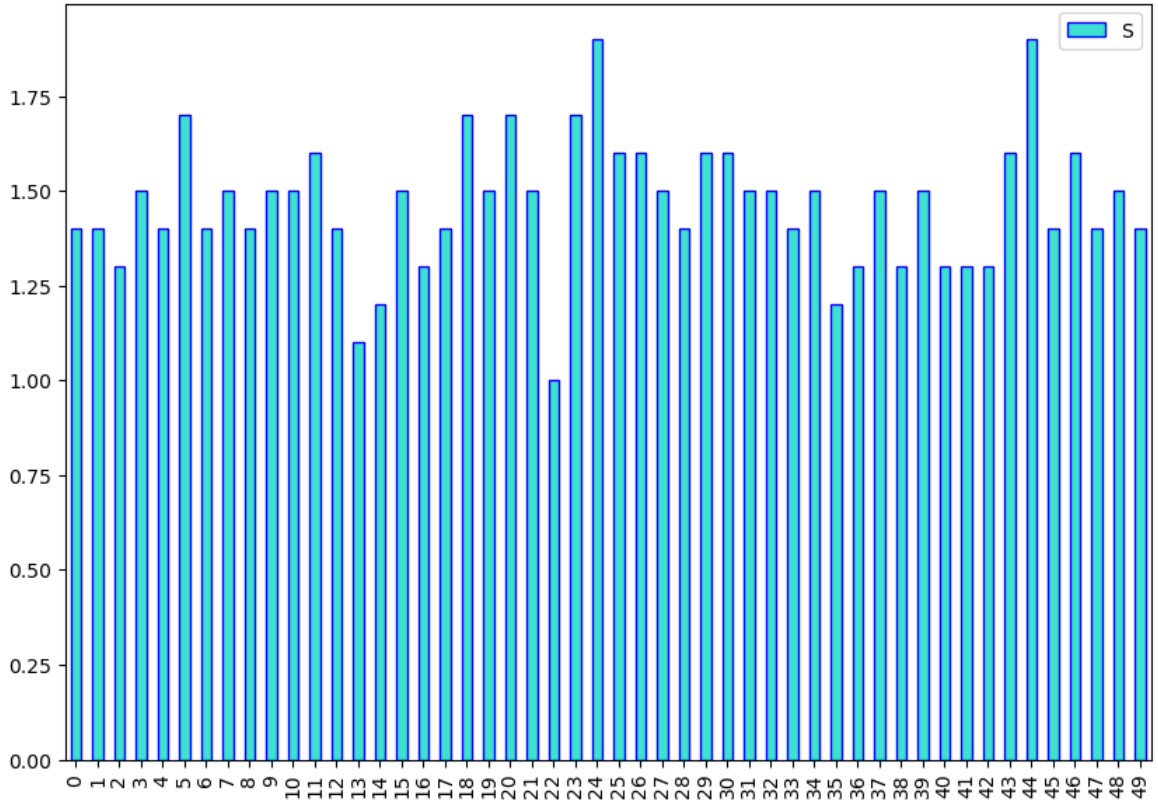


Bar Graph for Petal Length

In [21]:

```
m=list(data['PetalLengthCm'])
n=m[0:50]
n=pd.DataFrame(n)
print(type(n))
n.plot(kind='bar',color='turquoise',edgecolor='blue')
plt.legend("Species")
plt.show();
```

```
<class 'pandas.core.frame.DataFrame'>
```

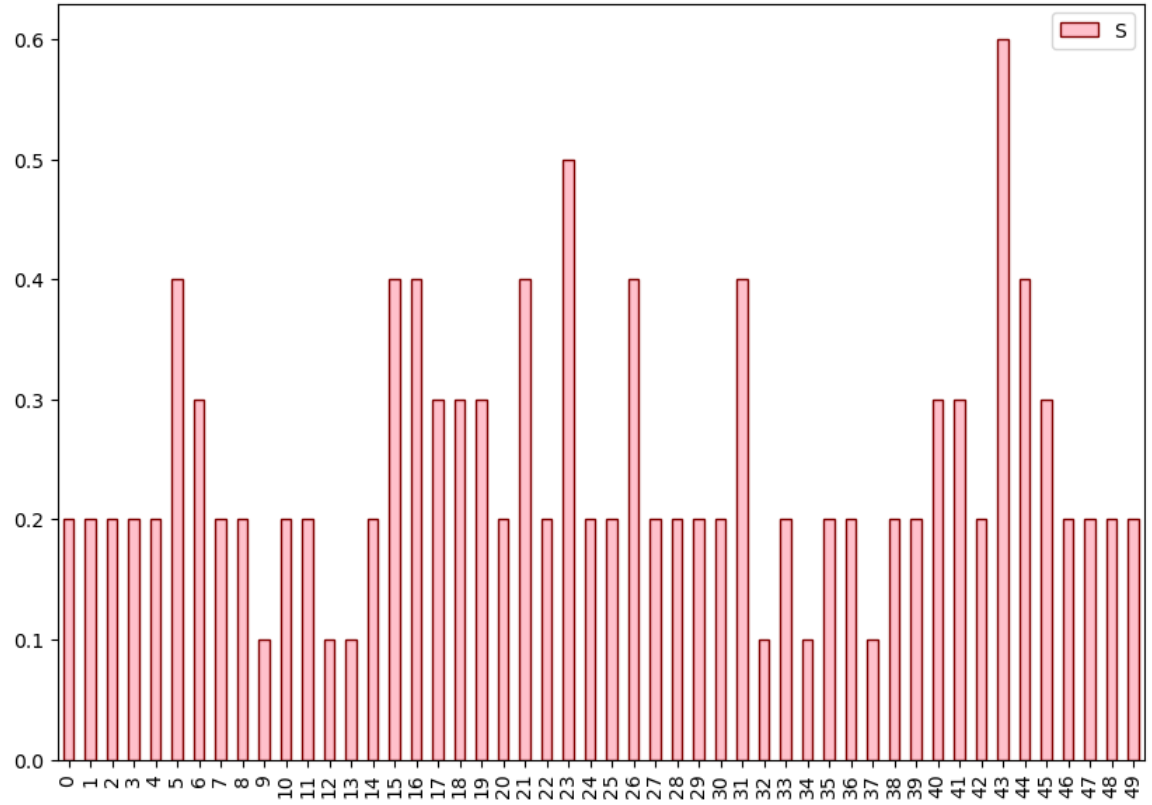


Bar Graph for Petal Width

In [22]:

```
m=list(data['PetalWidthCm'])
n=m[0:50]
n=pd.DataFrame(n)
print(type(n))
n.plot(kind='bar',color='pink',edgecolor='maroon')
plt.legend("Species")
plt.show();
```

```
<class 'pandas.core.frame.DataFrame'>
```

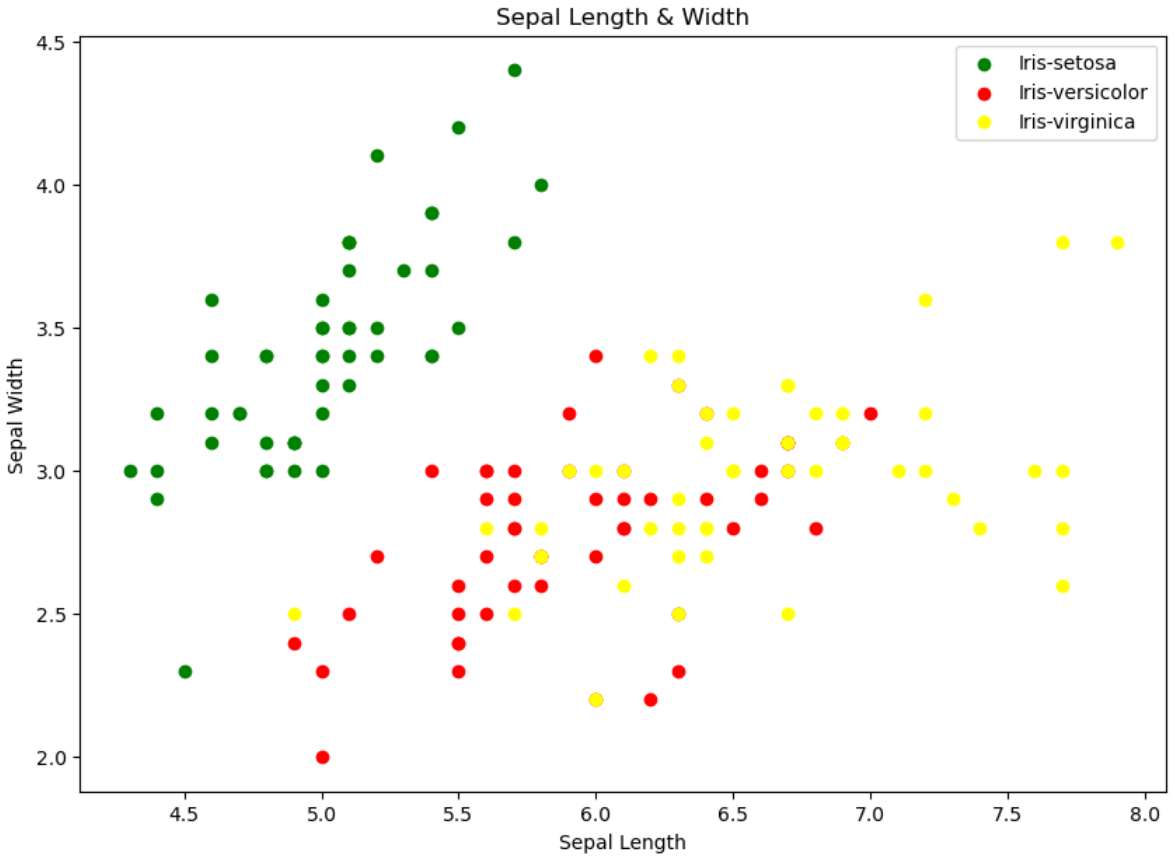


SCATTER PLOT

ScatterPlot for Sepal Length & Width

In [23]:

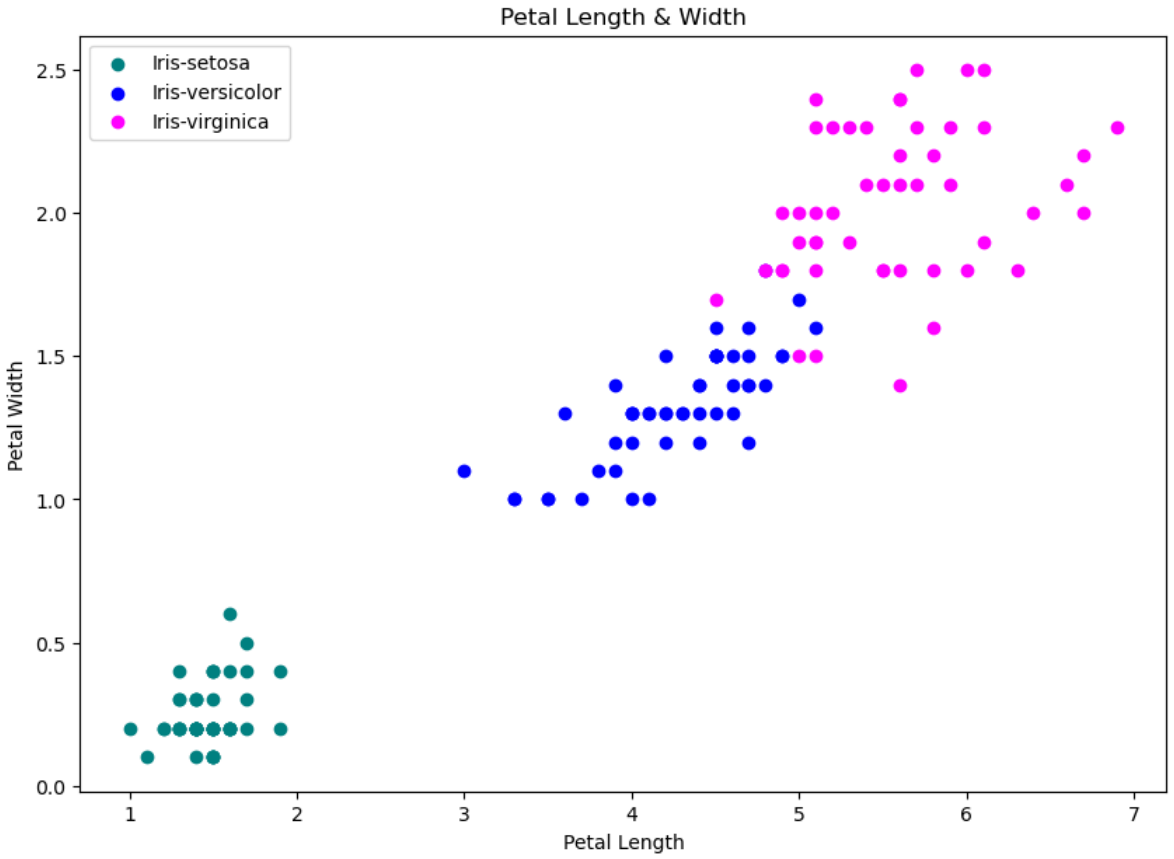
```
colors = ['green', 'red', 'yellow']
species = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
for i in range(3):
    x = data[data['Species'] == species[i]]
    plt.scatter(x['SepalLengthCm'], x['SepalWidthCm'], c= colors[i], label = species[i])
    plt.title('Sepal Length & Width')
    plt.xlabel("Sepal Length")
    plt.ylabel("Sepal Width")
    plt.legend();
```



Scatter Plot for Petal Length & Width

In [24]:

```
colors = ['teal', 'blue', 'magenta']
species = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
for i in range(3):
    x = data[data['Species'] == species[i]]
    plt.scatter(x['PetalLengthCm'], x['PetalWidthCm'], c= colors[i], label = species[i])
    plt.title('Petal Length & Width')
    plt.xlabel("Petal Length")
    plt.ylabel("Petal Width")
    plt.legend();
```



PIE CHART

Pie Chart for Iris Species

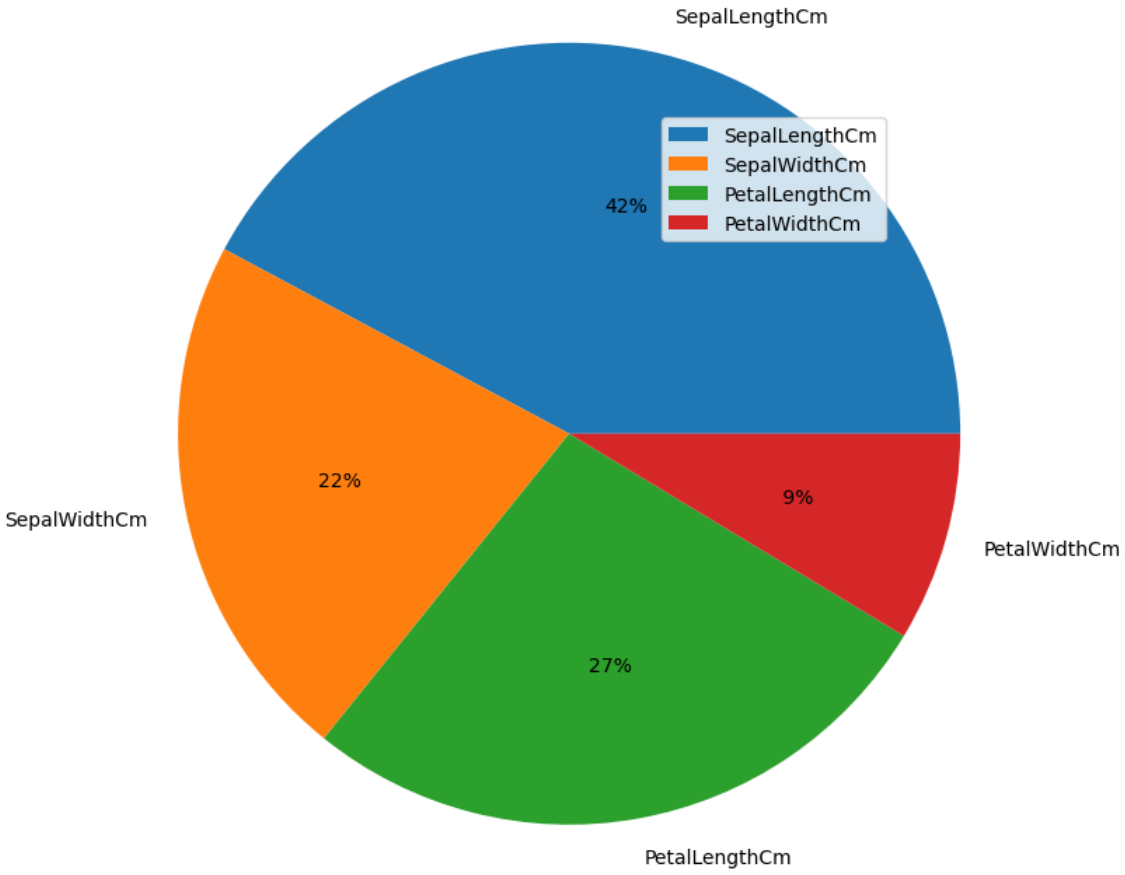
In [25]:

```
#deleting a column from dataset
data = data.drop(columns=['Id'])
```

In [26]:

```
a = list(data.columns)
a = a[0:4]
plt.figure(figsize= (6,8))
plt.pie(data.mean(0),labels=a,radius=1.5,autopct ="%2.0f%%")
plt.legend();
```

```
C:\Users\priya\AppData\Local\Temp\ipykernel_4468\2211255378.py:4: FutureWarning: Dropping of nuisance columns in Data
Frame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select onl
y valid columns before calling the reduction.
  plt.pie(data.mean(0),labels=a,radius=1.5,autopct ="%2.0f%%")
```

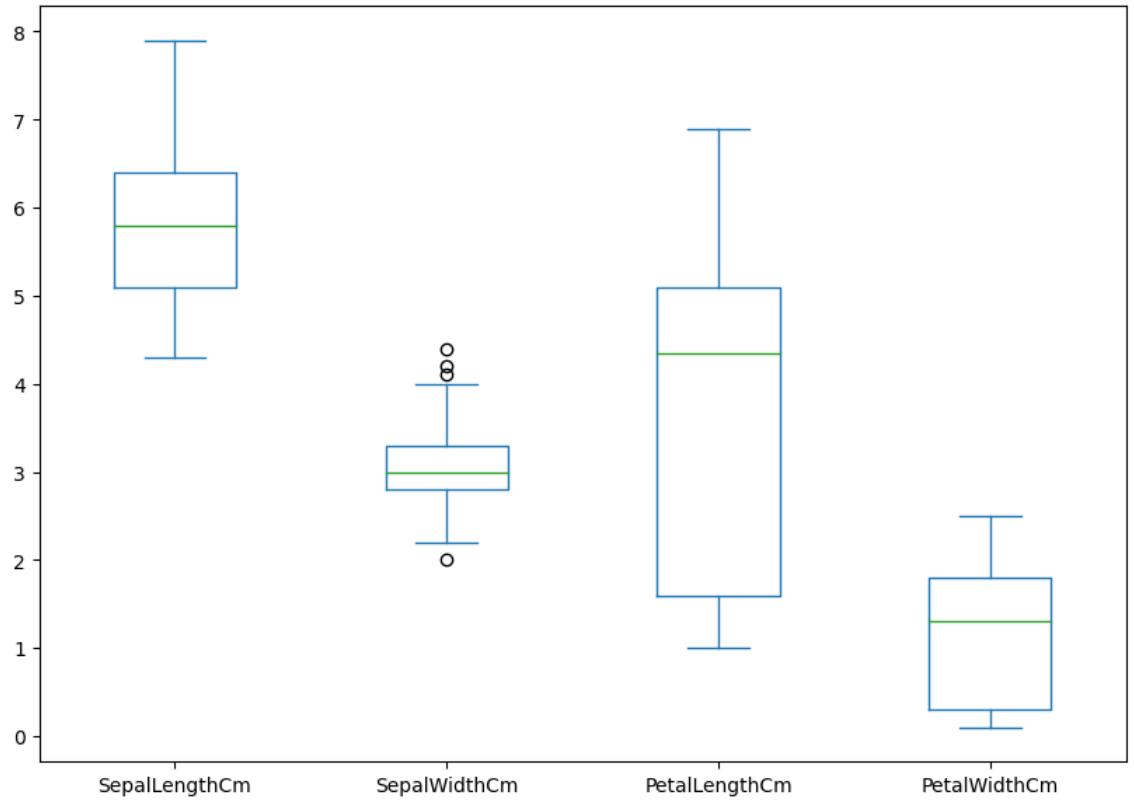


BOX PLOT

Box Plot for Iris Species

In [27]:

```
data = pd.DataFrame(data)
data.plot(kind='box', vert=True);
```



Correlation Matrix

The correlation matrix display correlation coefficients for different variables.The value is in range of -1 to 1.

In [28]:

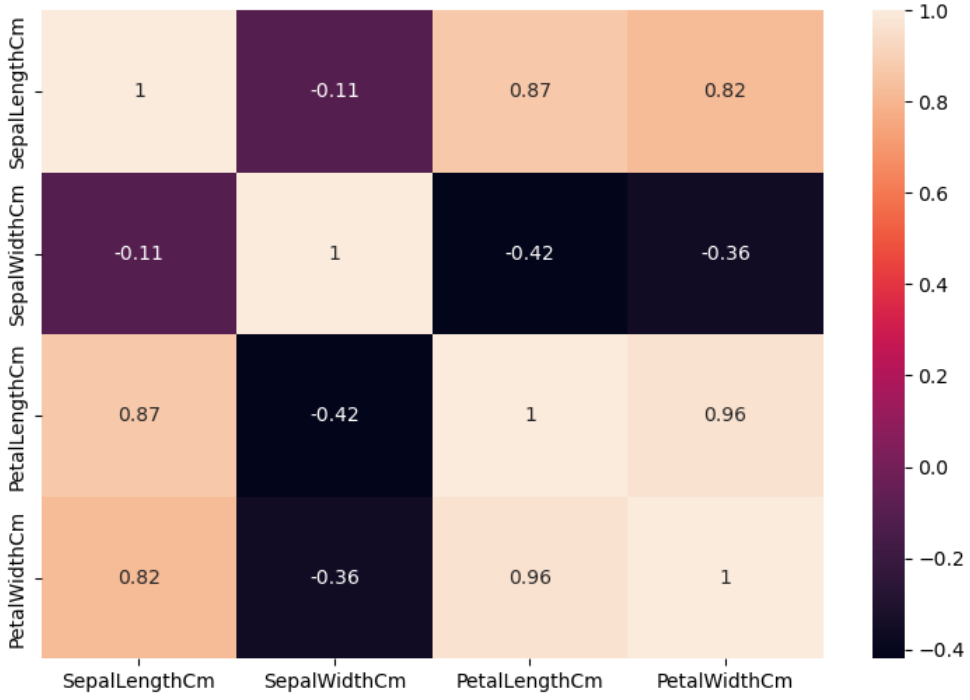
```
data.corr()
```

Out[28]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000

In [29]:

```
corr=data.corr()
plt.subplots(figsize=(9,6))
sns.heatmap(corr,annot=True);
```



THANKS:)