Exploratory Data Analysis on Setosa,Virgini and Versicolor:

Actually it is a flower and Setosa,Virginica,Versicolor are the name of the Species.Lets find out how many flowers are Setosa,Virginica,Versicolor.

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

In [2]:
```
iris= pd.read_csv("C:/Users/anirb/Downloads/Iris.csv")
iris
```

Out[2]:

	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

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
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

How Many data Points in each class are present?

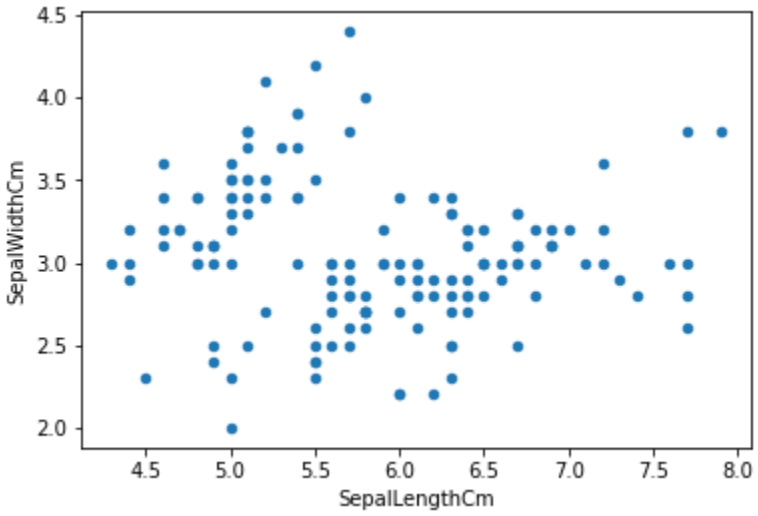
In [3]:
```
iris['Species'].value_counts()
```

Out[3]:
```
Iris-setosa        50
Iris-virginica     50
Iris-versicolor    50
Name: Species, dtype: int64
```

Here the catagories are approximately same. SO iris is a balanced dataset

Scater plot EDA

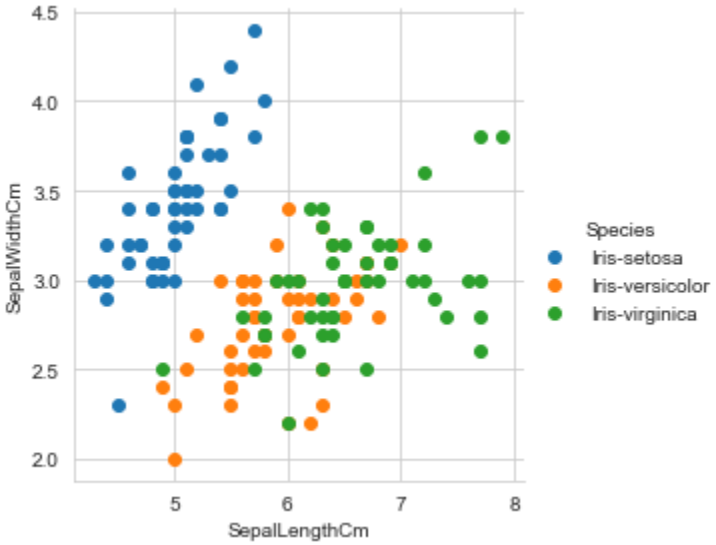
In [4]:
```
iris.plot(kind= 'scatter',x= 'SepalLengthCm',y= 'SepalWidthCm')
plt.show()
```



create a mapping to differentiate according to their species :

In [5]:
```
sns.set_style("whitegrid")
sns.FacetGrid(iris,hue= 'Species',size =4) \
.map(plt.scatter, 'SepalLengthCm','SepalWidthCm') \
.add_legend();
plt.show();
```

```
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\axisgrid.py:316: UserWarning: The `size
` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

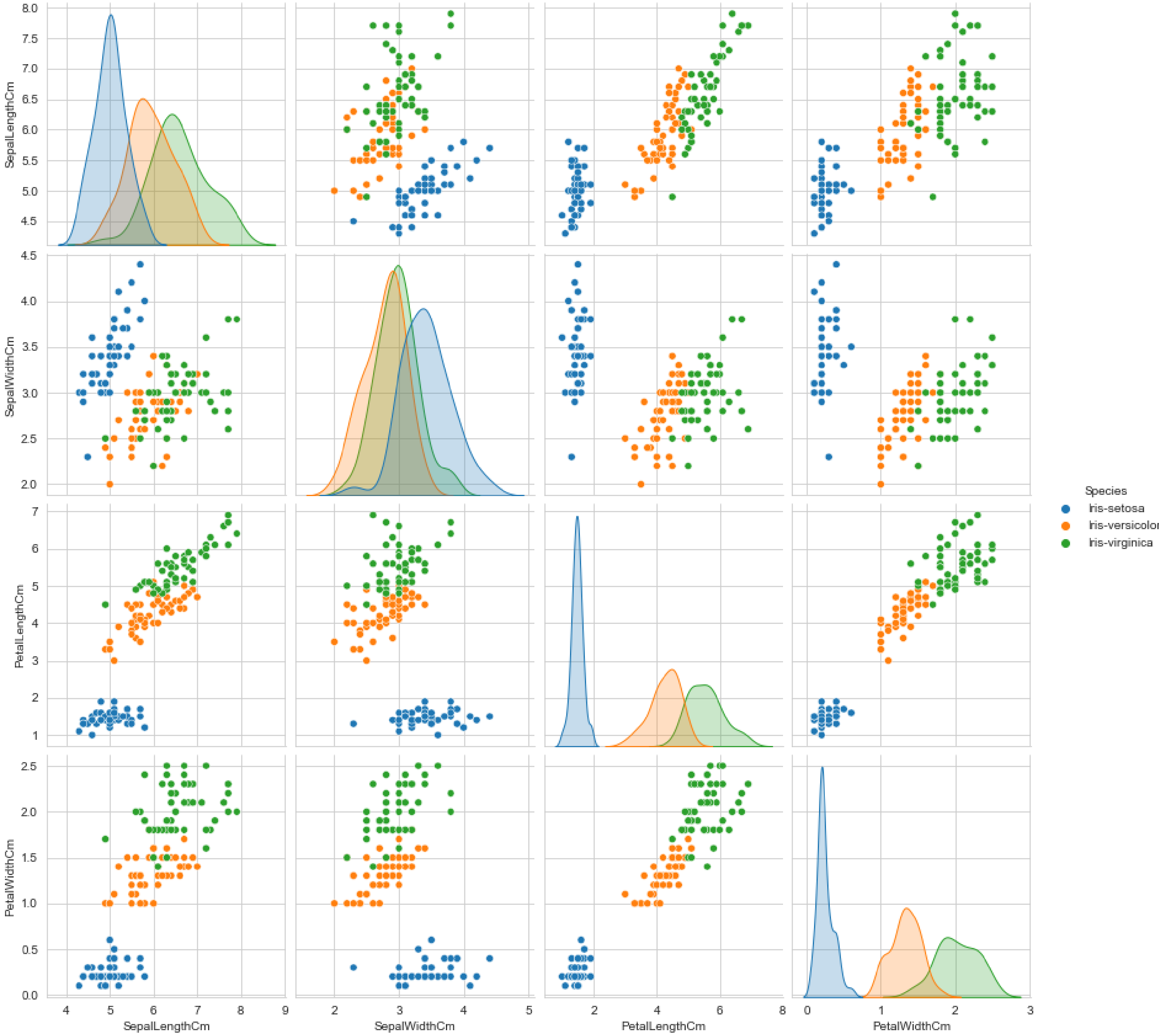


Here with the help of the mapping we can clearly unterstand how many points belongs to the three catagory

Pairplot (EDA)

In [6]:

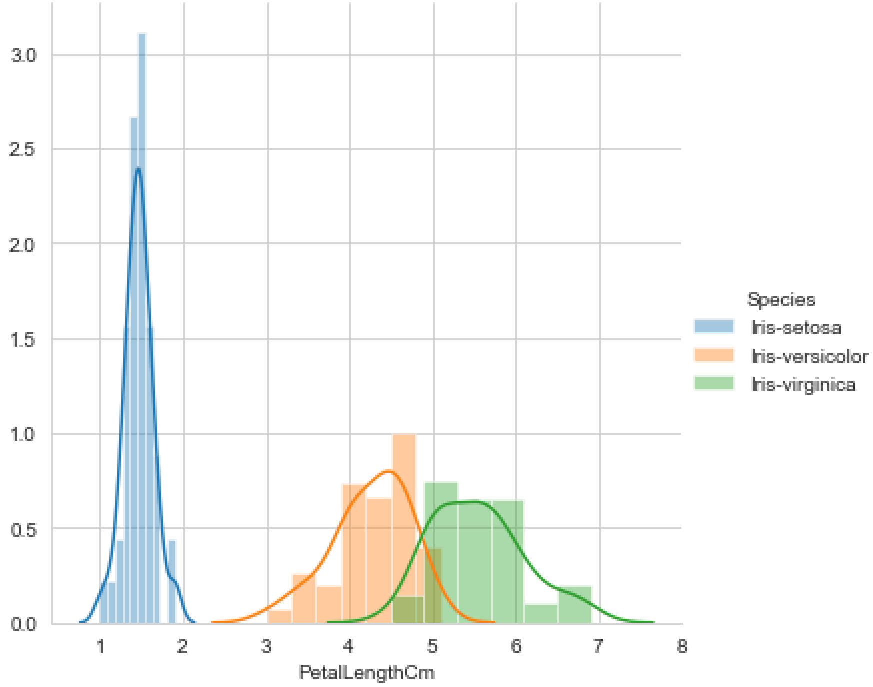
```
sns.set_style("whitegrid")
sns.pairplot(iris,hue= 'Species',height= 3)
plt.show()
```



In [7]:

```
sns.FacetGrid(iris,hue= 'Species',height = 5)\
 .map(sns.distplot,"PetalLengthCm") \
 .add_legend();
plt.show();
```

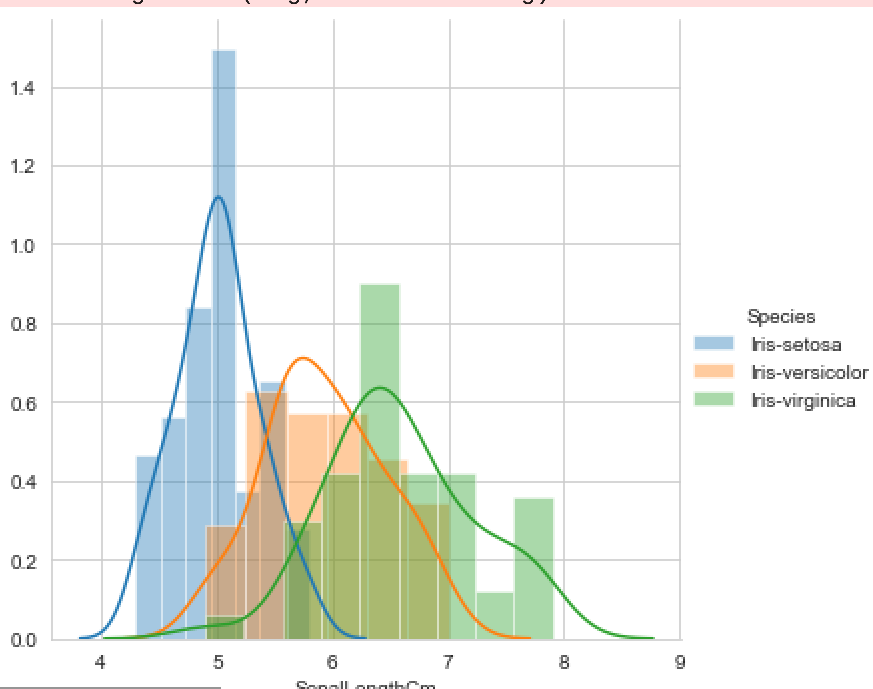
```
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [17]:

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

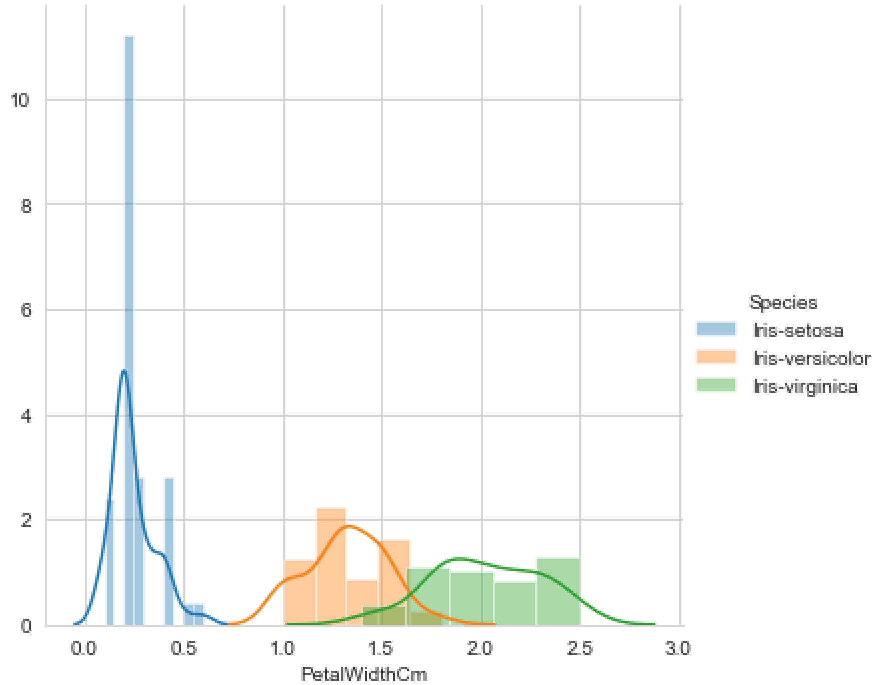
```
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [18]:

```
sns.FacetGrid(iris,hue= 'Species',height = 5)\
 .map(sns.distplot,"PetalWidthCm") \
 .add_legend();
plt.show();
```

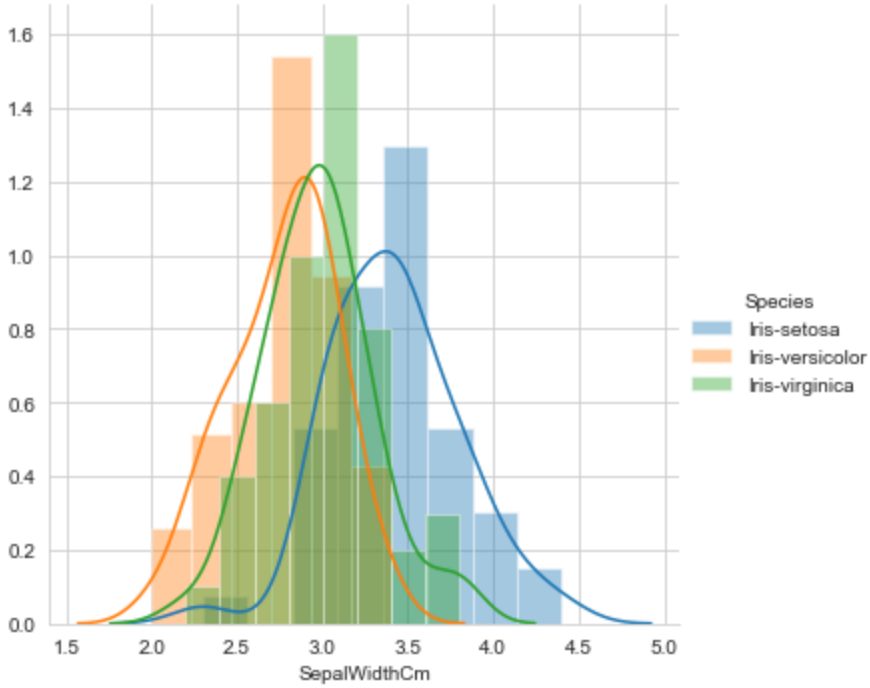
```
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [19]:

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

```
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

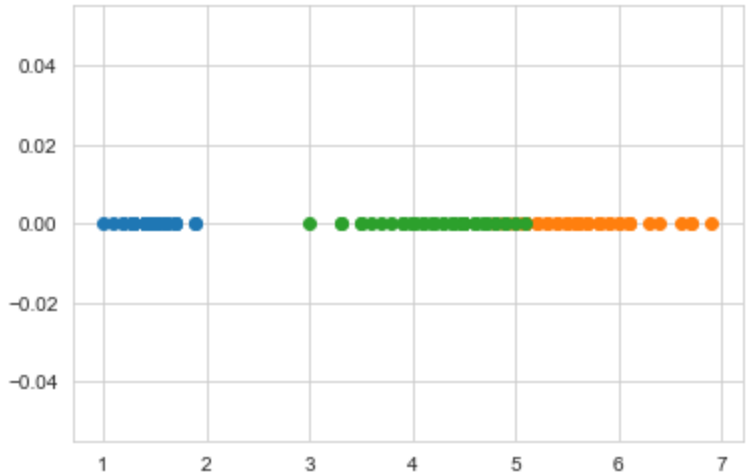


Histogram (EDA)

The bell curve is called PDF

In [14]:
```
iris_setosa= iris.loc[iris['Species']== 'Iris-setosa'];
iris_virginica= iris.loc[iris['Species']== 'Iris-virginica'];
iris_versicolor= iris.loc[iris['Species']== 'Iris-versicolor'];
plt.plot(iris_setosa['PetalLengthCm'],np.zeros_like(iris_setosa['PetalLengthCm']),'o')
plt.plot(iris_virginica['PetalLengthCm'],np.zeros_like(iris_virginica['PetalLengthCm']),'o
plt.plot(iris_versicolor['PetalLengthCm'],np.zeros_like(iris_versicolor['PetalLengthCm']),

plt.show()
```



CDF

In [30]:
```
counts,bin_edges=np.histogram(iris_setosa['PetalLengthCm'],bins= 10)
counts,bin_edges
```

Out[30]:
```
(array([ 1,  1,  2,  7, 12, 14,  7,  4,  0,  2], dtype=int64),
 array([1.  , 1.09, 1.18, 1.27, 1.36, 1.45, 1.54, 1.63, 1.72, 1.81, 1.9 ]))
```

In [25]:
```
pdf= counts/sum(counts)
print(pdf)
```

```
[0.02 0.02 0.04 0.14 0.24 0.28 0.14 0.08 0.   0.04]
```

In [31]:
```
counts1,bin_edges1=np.histogram(iris_virginica['PetalLengthCm'],bins= 10)
counts1,bin_edges1
```

Out[31]:
```
(array([ 1,  5, 12,  4,  9,  8,  5,  2,  1,  3], dtype=int64),
 array([4.5 , 4.74, 4.98, 5.22, 5.46, 5.7 , 5.94, 6.18, 6.42, 6.66, 6.9 ]))
```

In [33]:
```
pdf1= counts1/sum(counts1)
print(pdf1)
```

```
[0.02 0.1  0.24 0.08 0.18 0.16 0.1  0.04 0.02 0.06]
```

In [35]:
```
counts2,bin_edges2=np.histogram(iris_versicolor['PetalLengthCm'],bins= 10)
counts2,bin_edges2
```

Out[35]:
```
(array([ 1,  2,  3,  2,  8,  7,  6, 10,  7,  4], dtype=int64),
 array([3.  , 3.21, 3.42, 3.63, 3.84, 4.05, 4.26, 4.47, 4.68, 4.89, 5.1 ]))
```

In [36]:
```
pdf2= counts2/sum(counts2)
print(pdf2)
```

```
[0.02 0.04 0.06 0.04 0.16 0.14 0.12 0.2  0.14 0.08]
```

In [38]:
```
cdf= np.cumsum(pdf)
cdf
```

Out[38]:
```
array([0.02, 0.12, 0.36, 0.44, 0.62, 0.78, 0.88, 0.92, 0.94, 1.  ])
```

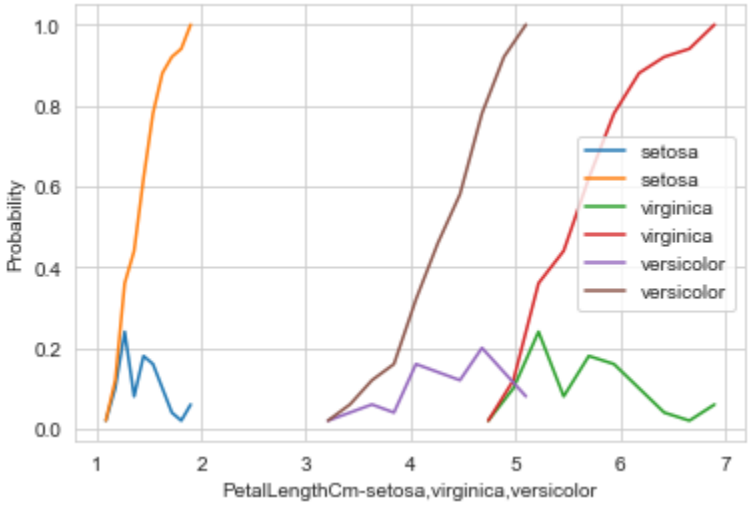
In [39]:
```
cdf1 = np.cumsum(pdf1)
cdf1
```

Out[39]:
```
array([0.02, 0.12, 0.36, 0.44, 0.62, 0.78, 0.88, 0.92, 0.94, 1.  ])
```

In [40]:
```
cdf2= np.cumsum(pdf2)
cdf2
```

Out[40]:
```
array([0.02, 0.06, 0.12, 0.16, 0.32, 0.46, 0.58, 0.78, 0.92, 1.  ])
```

In [46]:
```
plt.plot(bin_edges[1:],pdf,label='setosa')
plt.plot(bin_edges[1:],cdf,label='setosa')
plt.plot(bin_edges1[1:],pdf1,label='virginica')
plt.plot(bin_edges1[1:],cdf1,label='virginica')
plt.plot(bin_edges2[1:],pdf2,label='versicolor')
plt.plot(bin_edges2[1:],cdf2,label='versicolor')
plt.xlabel("PetalLengthCm-setosa,virginica,versicolor")
plt.ylabel("Probability")
plt.legend()
plt.show()
```



Mean and Standard Deviation

In [47]:

```
print("Mean:")
print(np.mean(iris_setosa['PetalLengthCm']))
print(np.mean(iris_virginica['PetalLengthCm']))
print(np.mean(iris_versicolor['PetalLengthCm']))

print("\nstd-dev:")
print(np.std(iris_setosa['PetalLengthCm']))
print(np.std(iris_virginica['PetalLengthCm']))
print(np.std(iris_versicolor['PetalLengthCm']))
```

```
Mean:
1.464
5.552
4.26

std-dev:
0.17176728442867115
0.5463478745268441
0.4651881339845204
```

Median,Percentile,Quantile :

In [48]:

```
print("Median:")
print(np.median(iris_setosa['PetalLengthCm']))
print(np.median(iris_virginica['PetalLengthCm']))
print(np.median(iris_versicolor['PetalLengthCm']))

print("\nQuantile:")
print(np.percentile(iris_setosa['PetalLengthCm'],np.arange(0,100,25)))
print(np.percentile(iris_virginica['PetalLengthCm'],np.arange(0,100,25)))
print(np.percentile(iris_versicolor['PetalLengthCm'],np.arange(0,100,25)))

print("\n90th percentile:")
print(np.percentile(iris_setosa['PetalLengthCm'],90))
print(np.percentile(iris_virginica['PetalLengthCm'],90))
print(np.percentile(iris_versicolor['PetalLengthCm'],90))
```

```
Median:
1.5
5.55
4.35
```

```
Quantile:
[1.    1.4   1.5   1.575]
[4.5   5.1   5.55  5.875]
[3.   4.   4.35 4.6 ]

90th percentile:
1.7
6.31
4.8
```

Median Absolute Deviation(MAD):

In [51]:

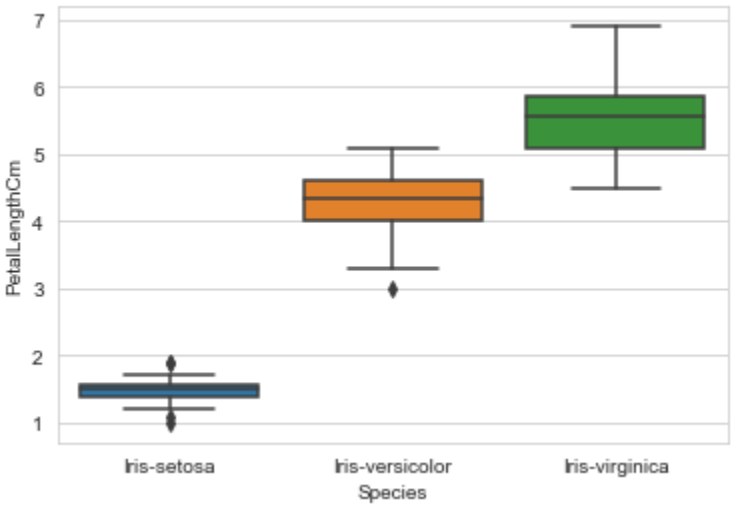
```
from statsmodels import robust
print("MAD:")
print(robust.mad(iris_setosa['PetalLengthCm']))
print(robust.mad(iris_virginica['PetalLengthCm']))
print(robust.mad(iris_versicolor['PetalLengthCm']))
```

```
MAD:
0.14826022185056031
0.6671709983275211
0.5189107764769602
```

Box plot :

In [52]:

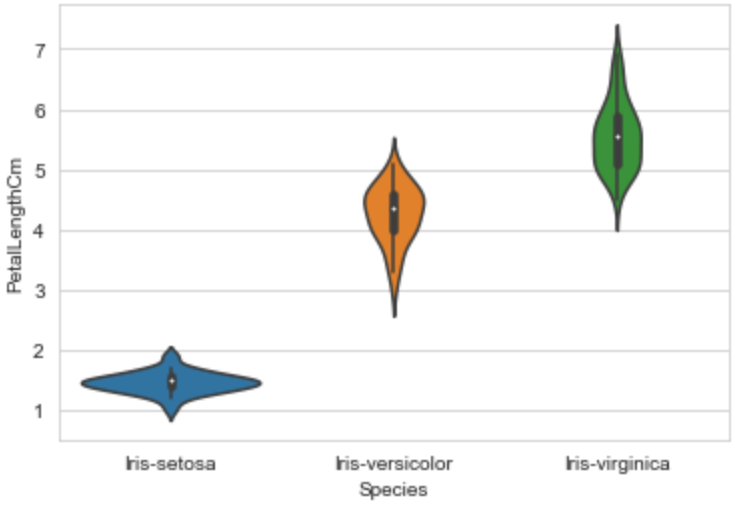
```
sns.boxplot(x='Species',y= 'PetalLengthCm',data= iris)
plt.show()
```



Violin Plot

In [54]:

```
sns.violinplot(x='Species',y= 'PetalLengthCm',data= iris,size=4)
plt.show()
```



In [ ]:

```
Thank You
```