Iris Dataset

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

In [3]:
```
iris=pd.read_csv("Iris.csv")
```

In [4]:
```
print(iris.shape)
```

```
(150, 6)
```

In [5]:
```
print(iris.columns)
```

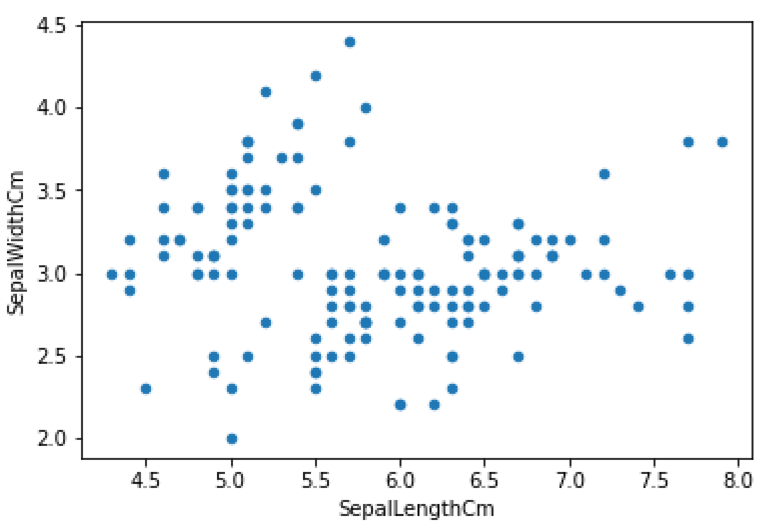
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthC
m',
       'Species'],
      dtype='object')
```

In [6]:
```
iris["Species"].value_counts()
```

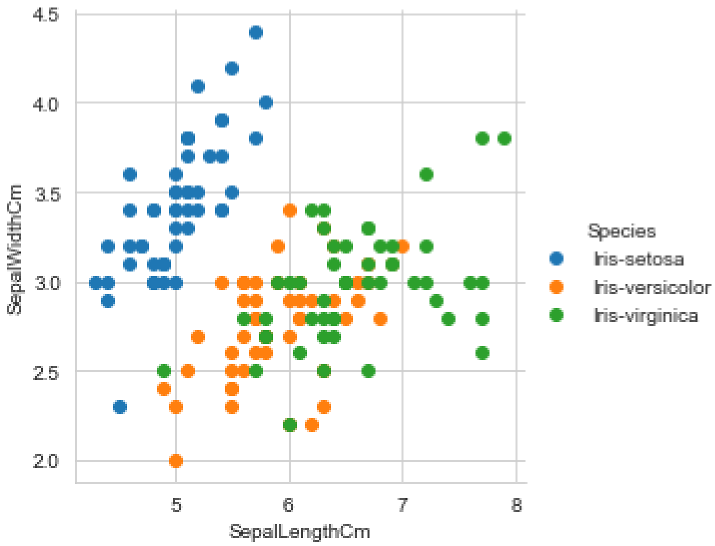
Out[6]:
```
Iris-versicolor    50
Iris-setosa        50
Iris-virginica     50
Name: Species, dtype: int64
```

2D Scatter Plot

In [7]:
```
iris.plot(kind='scatter',x='SepalLengthCm',y='SepalWidthCm');
plt.show()
```



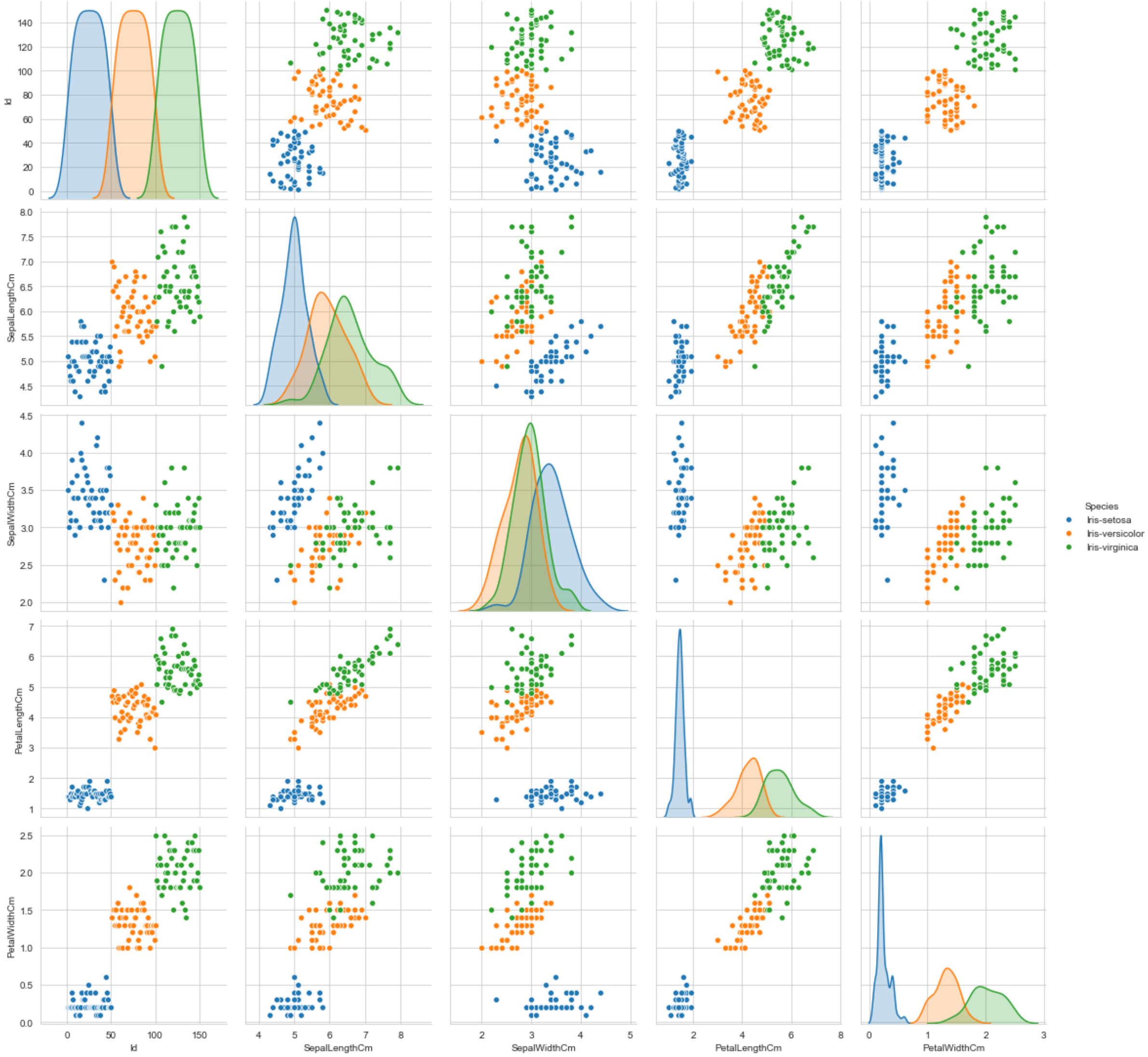
In [8]:
```
sb.set_style("whitegrid");
sb.FacetGrid(iris,hue="Species",height=4).map(plt.scatter,"SepalLengthCm","SepalWidthCm").add_legend();
#hue means which column in the data frame will be used for color encoding.
```



3D Scatter Plots

Pair Plots can be used to visualize when number of features are more.

In [9]:
```
plt.close();
sb.set_style("whitegrid");
sb.pairplot(iris,hue="Species",height=3);
plt.show()
```



In [10]:
```
# in PL vs PW graph,blue cluster much more far away from other color dots.
#PL and PW is important features
```

Histogram, PDF, CDF

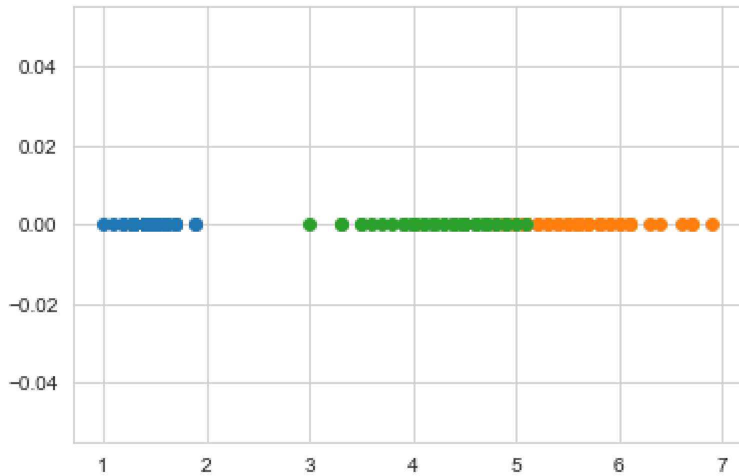
we are creating 1-D scatter plot for each class using petallength as feature(x-axis) it can be easily seen from the graph that it is hard to make sense from graph as there is alot of overlapping so we are trying to use histograms better way of visualizing 1-D scatter plot

In [11]:
```
iris_setosa=iris.loc[iris["Species"]=="Iris-setosa"];
```

In [12]:
```
iris_virginica=iris.loc[iris["Species"]=="Iris-virginica"];
```

In [13]:
```
iris_versicolor=iris.loc[iris["Species"]=="Iris-versicolor"];
```

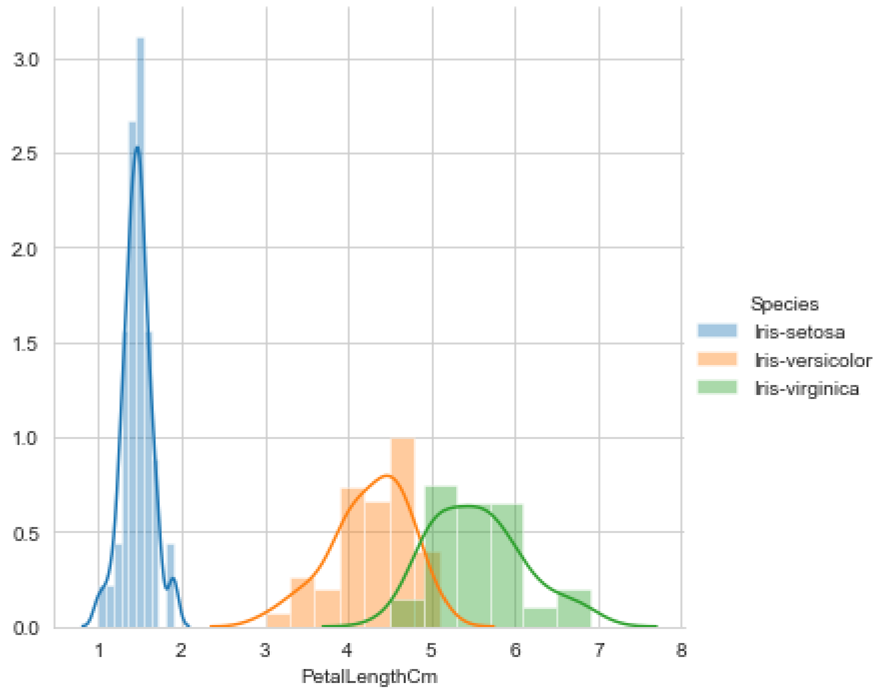
In [14]:
```
plt.plot(iris_setosa["PetalLengthCm"],np.zeros_like(iris_setosa['PetalLengthC
m']),'o')
plt.plot(iris_virginica["PetalLengthCm"],np.zeros_like(iris_virginica['PetalLe
ngthCm']),'o')
plt.plot(iris_versicolor["PetalLengthCm"],np.zeros_like(iris_versicolor['Petal
LengthCm']),'o')
plt.show()
```



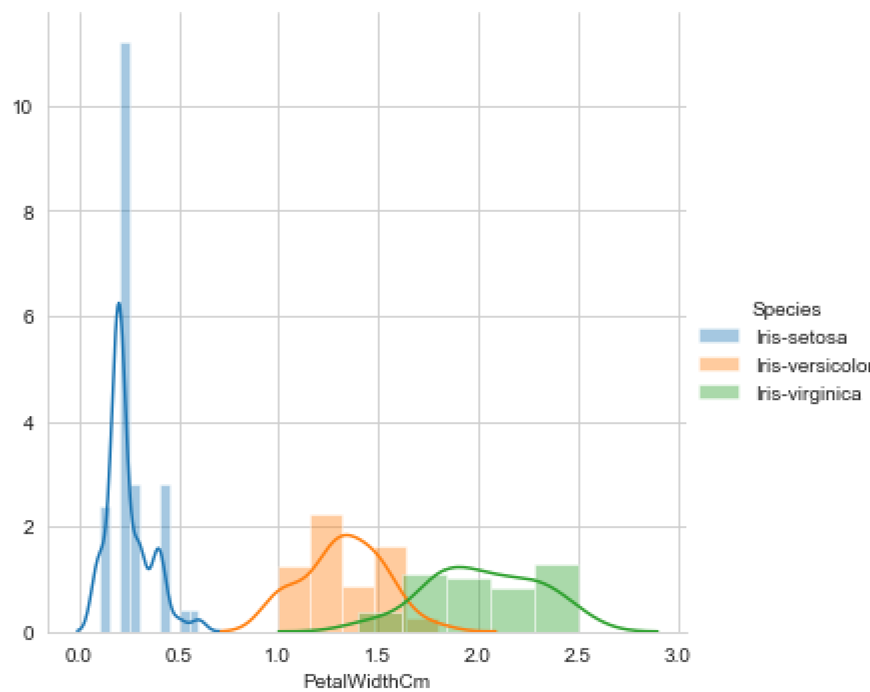
univariate analysis : analysis by using one variable.

we are using distribution plots for every feature to know the distribution of data better. plotting the plots for each feature.

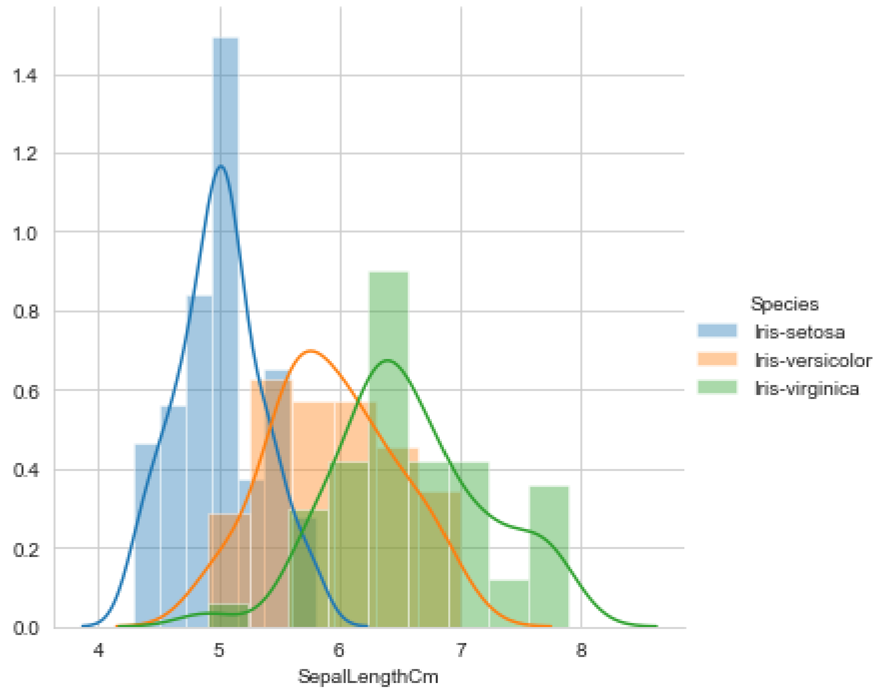
In [15]:
```
sb.FacetGrid(iris,hue="Species",height=5).map(sb.distplot,"PetalLengthCm").add
_legend();
plt.show();
```



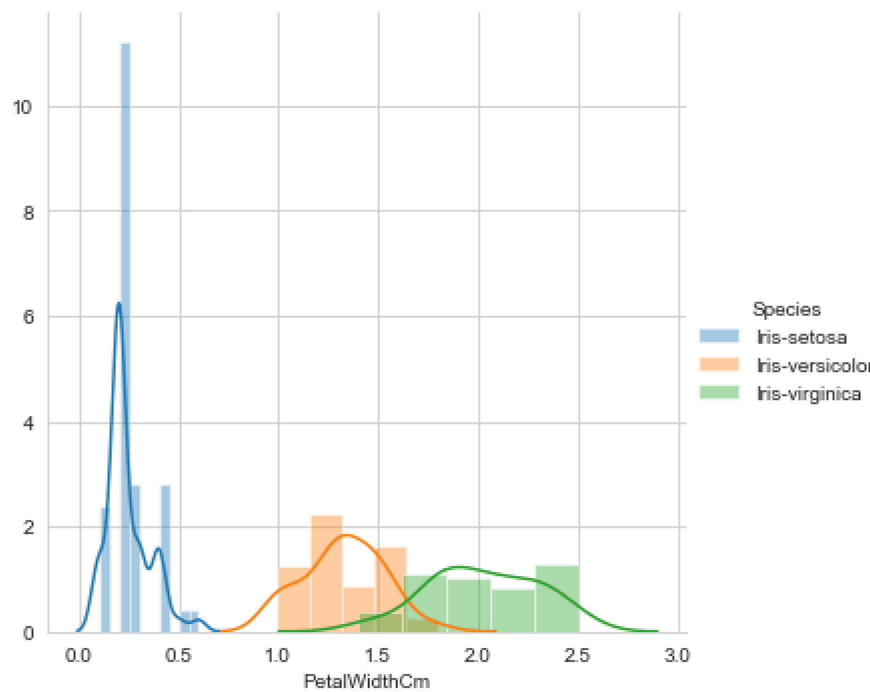
In [16]:
```
sb.FacetGrid(iris,hue="Species",height=5).map(sb.distplot,"PetalWidthCm").add_
legend();
plt.show();
```



In [17]:
```
sb.FacetGrid(iris,hue="Species",height=5).map(sb.distplot,"SepalLengthCm").add
_legend();
plt.show();
```



In [18]:
```
sb.FacetGrid(iris,hue="Species",height=5).map(sb.distplot,"PetalWidthCm").add_
legend();
plt.show();
```



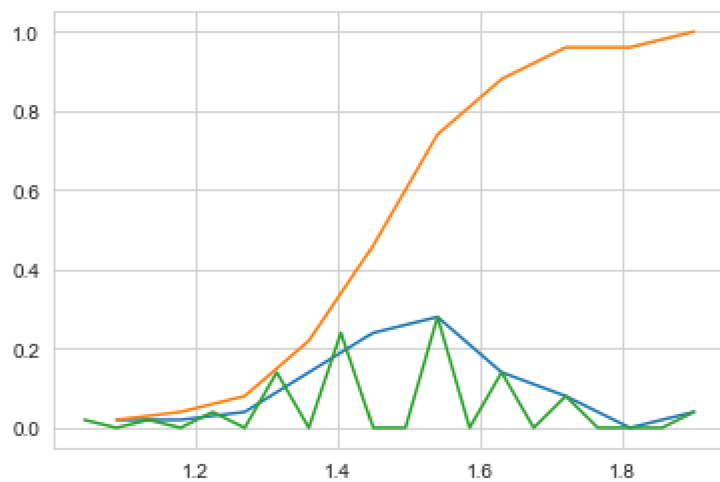
these vertical lines are the histogram and the smooth distribution of these is the pdf of respective class(virginica,setosa,versicolor)

- y -axis is the counts of each class.
- it shows density of points that's why it is also called as density plots
- As you can see the best feature is petal length for this dataset.

CDF cumulative density function

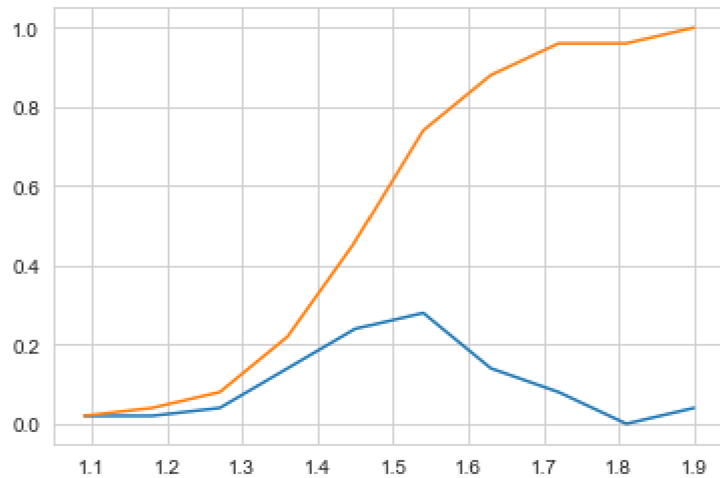
In [19]:
```
#plot pdf and cdf of petal_length
counts,bin_edges=np.histogram(iris_setosa['PetalLengthCm'],bins=10,density=True)
pdf=counts/sum(counts)
print(pdf)
print(bin_edges)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)

counts,bin_edges=np.histogram(iris_setosa['PetalLengthCm'],bins=20,density=True)
pdf=counts/sum(counts)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.show()
```

```
[0.02 0.02 0.04 0.14 0.24 0.28 0.14 0.08 0.   0.04]
[1.   1.09 1.18 1.27 1.36 1.45 1.54 1.63 1.72 1.81 1.9 ]
```



In [20]:
```
counts,bin_edges=np.histogram(iris_setosa['PetalLengthCm'],bins=10,density=True)
pdf=counts/sum(counts)
print(pdf)
print(bin_edges)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.show()
```

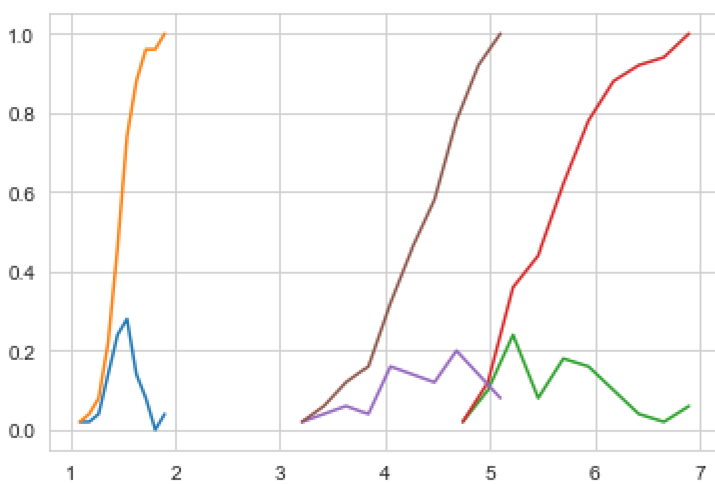
```
[0.02 0.02 0.04 0.14 0.24 0.28 0.14 0.08 0.   0.04]
[1.   1.09 1.18 1.27 1.36 1.45 1.54 1.63 1.72 1.81 1.9 ]
```



- yellow line is cdf of iris_setosa.
- blue line is pdf of iris_setosa.
- y-axis is Petal length
- x-axis is probability

In [21]:
```
counts, bin_edges = np.histogram(iris_setosa['PetalLengthCm'], bins=10, densit
y = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:], cdf)

counts, bin_edges = np.histogram(iris_virginica['PetalLengthCm'], bins=10, den
sity = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:], cdf)

counts, bin_edges = np.histogram(iris_versicolor['PetalLengthCm'], bins=10, de
nsity = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:], cdf)

plt.show()
```

```
[0.02 0.02 0.04 0.14 0.24 0.28 0.14 0.08 0.   0.04]
[1.   1.09 1.18 1.27 1.36 1.45 1.54 1.63 1.72 1.81 1.9 ]
[0.02 0.1  0.24 0.08 0.18 0.16 0.1  0.04 0.02 0.06]
[4.5  4.74 4.98 5.22 5.46 5.7  5.94 6.18 6.42 6.66 6.9 ]
[0.02 0.04 0.06 0.04 0.16 0.14 0.12 0.2  0.14 0.08]
[3.   3.21 3.42 3.63 3.84 4.05 4.26 4.47 4.68 4.89 5.1 ]
```



- the bigger curves are the cdf of respective classes,
- the smaller curves are the pdf
- We can classify these with some if else conditions.
- we can classify virginica &versicolor with some error because there is some overlap.
- virginica can be classified with 90% accuracy while versicolor can be classified with 95% accuracy

Mean, Variance, Std-deviation,

In [23]:
```
print("Means:")
print(np.mean(iris_setosa["PetalLengthCm"]))
#Mean with an outlier.
print(np.mean(np.append(iris_setosa["PetalLengthCm"],50)));
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"]))
```

```
Means:
1.464
2.4156862745098038
5.552
4.26

Std-dev:
0.17176728442867115
0.5463478745268441
0.4651881339845204
```

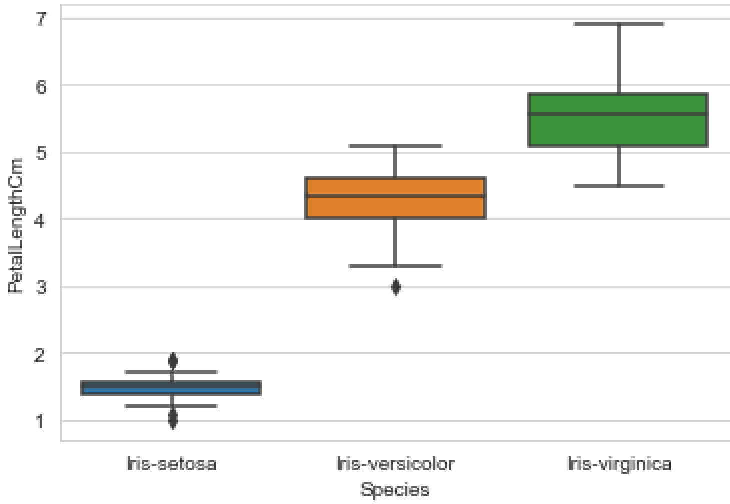
Median, Quantiles, Percentiles, IQR.

In [24]:
```
print("\nMedians:")
print(np.median(iris_setosa["PetalLengthCm"]))
#Median with an outlier
print(np.median(np.append(iris_setosa["PetalLengthCm"],50)));
print(np.median(iris_virginica["PetalLengthCm"]))
print(np.median(iris_versicolor["PetalLengthCm"]))
print("\nQuantiles:")
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 Percentiles:")
print(np.percentile(iris_setosa["PetalLengthCm"],90))
print(np.percentile(iris_virginica["PetalLengthCm"],90))
print(np.percentile(iris_versicolor["PetalLengthCm"], 90))
from statsmodels import robust
print ("\nMedian Absolute Deviation")
print(robust.mad(iris_setosa["PetalLengthCm"]))
```

```
Medians:
1.5
1.5
5.55
4.35

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

90th Percentiles:
1.7
6.3100000000000005
4.8

Median Absolute Deviation
0.14826022185056031
```

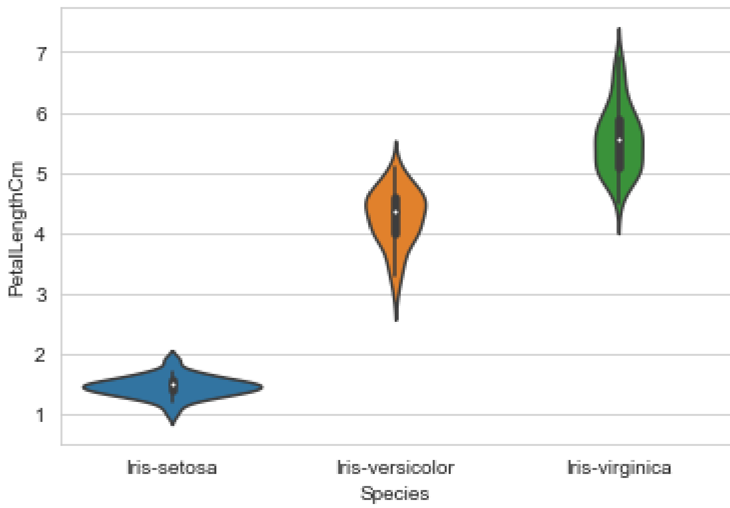
Box Plots

In [27]:
```
sb.boxplot(x='Species',y='PetalLengthCm', data=iris)
plt.show()
```



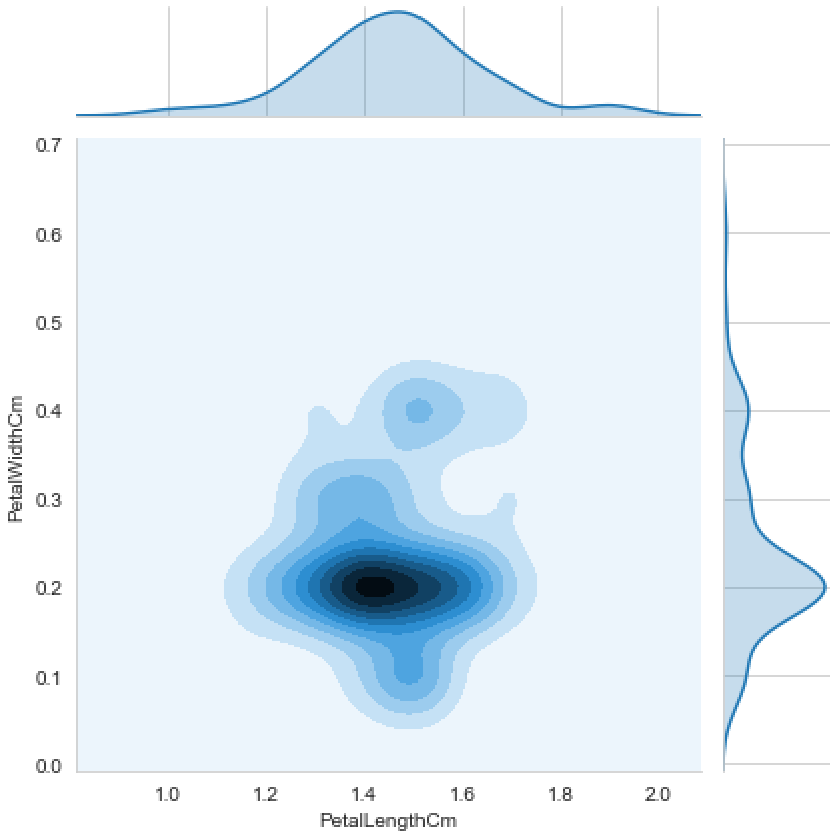
Violin Plots

In [28]:
```
sb.violinplot(x="Species", y="PetalLengthCm", data=iris, size=8)
plt.show()
```



2-D Density Plots ,Contours Plot

In [30]:
```
sb.jointplot(x="PetalLengthCm", y="PetalWidthCm", data=iris_setosa, kind="kde"
);
plt.show();
```



In [ ]: