Infomation about the data:

```
-> The dataset contains the following features:

    -> Column 1: Sepal lenght

    -> Column 2: Sepal Width

    -> Column 3: Petal Length

    -> Column 4: Petal Width

    -> Column 5: Type of Species
```

Objective:

```
-> The main objective is to perform exploratory data analysis

-> To find the best features that are useful to determine the specie
s type

-> Examining the observations by using Histograms, PDF, CDF
```

Importing the required libraries:

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as mp
import seaborn as s
```

Loading the dataset:

In [2]:

```
iris = pd.read_csv('iris.csv')
```

Information about data:

```
-> The shape of the data

-> Dimensionality of the data

-> Attributes of the data

-> Sample of the data
```

In [4]:

```
print(iris.shape)
print(iris.ndim)
print(iris.columns)
print(iris.head(5))
```

```
(150, 5)
2
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

Weights of the class labels in the data:

In [5]:

```
iris['species'].value_counts()
```

Out[5]:

```
virginica     50
versicolor    50
setosa        50
Name: species, dtype: int64
```

Observation:

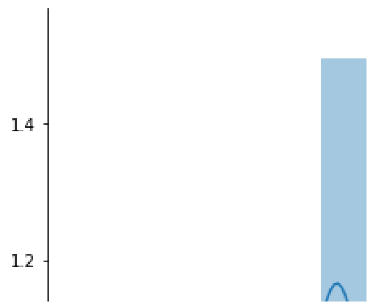
```
-> The data is perfectly balanced

-> Each class label is having the same weights
```

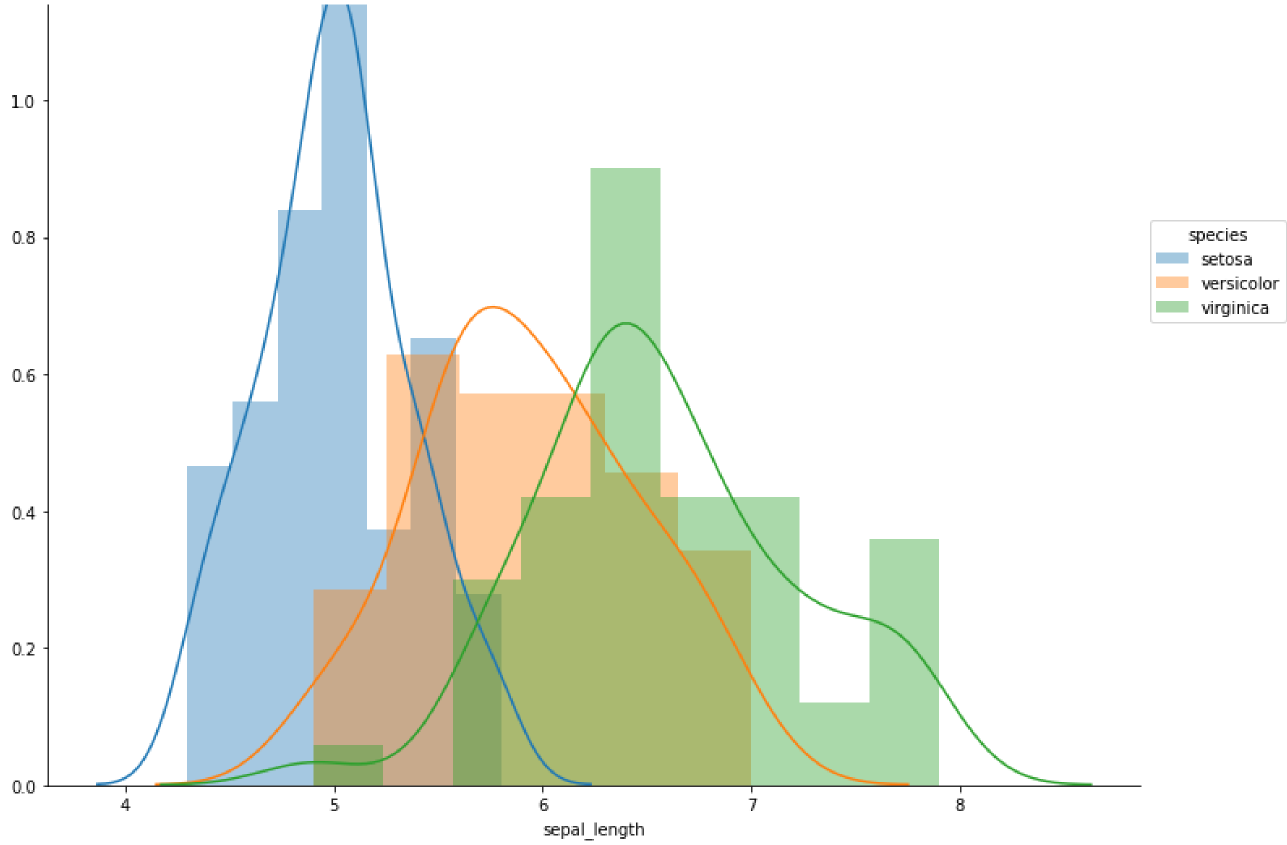
Univariate Analysis:

```
Histograms
```

In [6]:

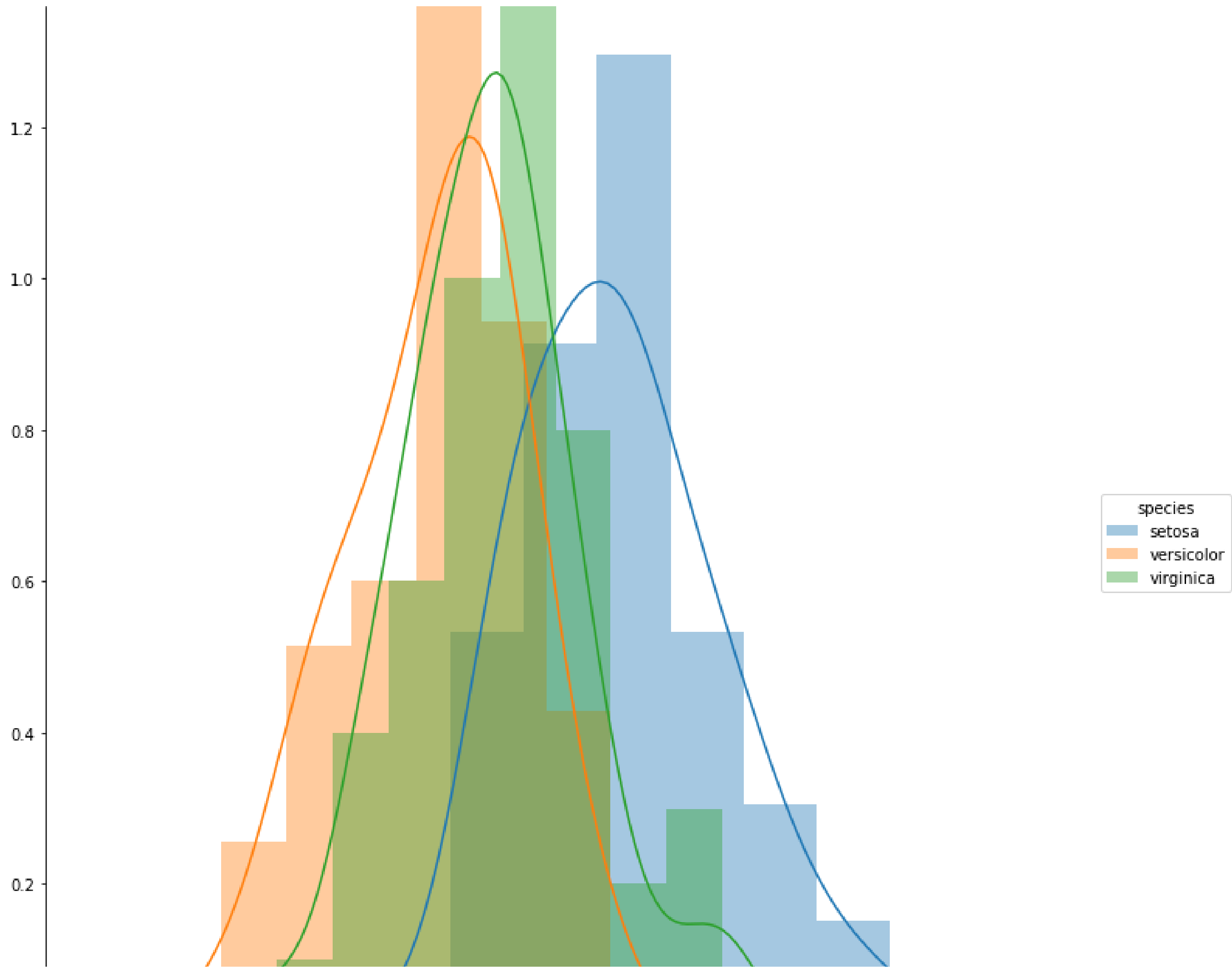
```
s.FacetGrid(data=iris,hue='species',size=10).map(s.distplot,'sepal_length')
.add_legend()
mp.show()
```

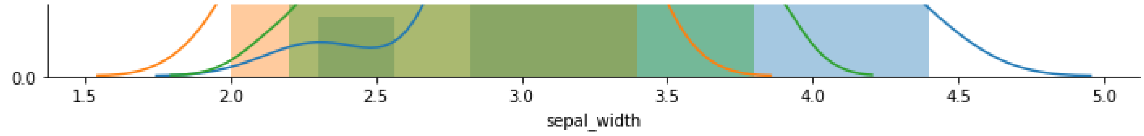




In [7]:

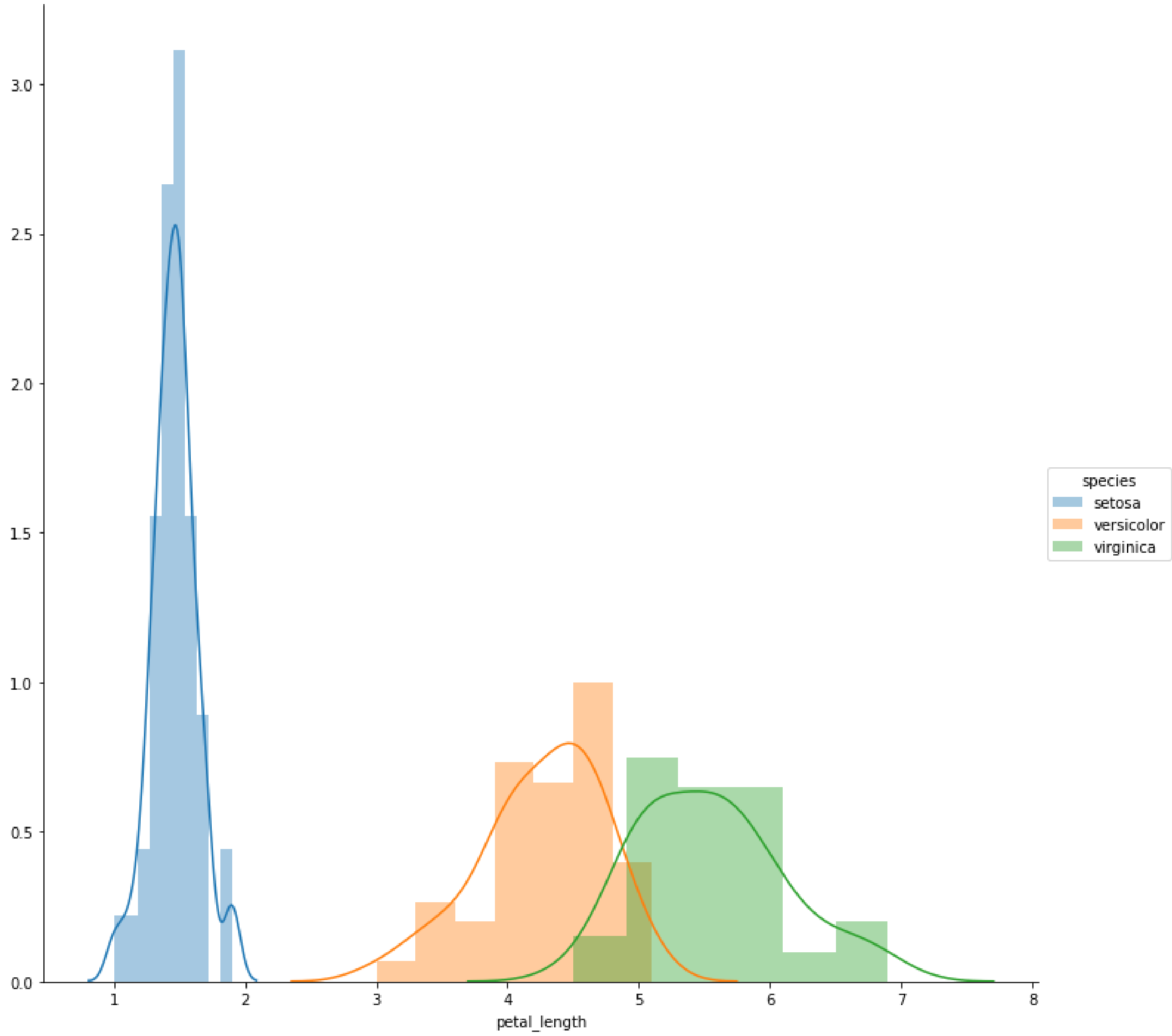
```
s.FacetGrid(data=iris,hue='species',size=10).map(s.distplot,'sepal_width').
add_legend()
mp.show()
```





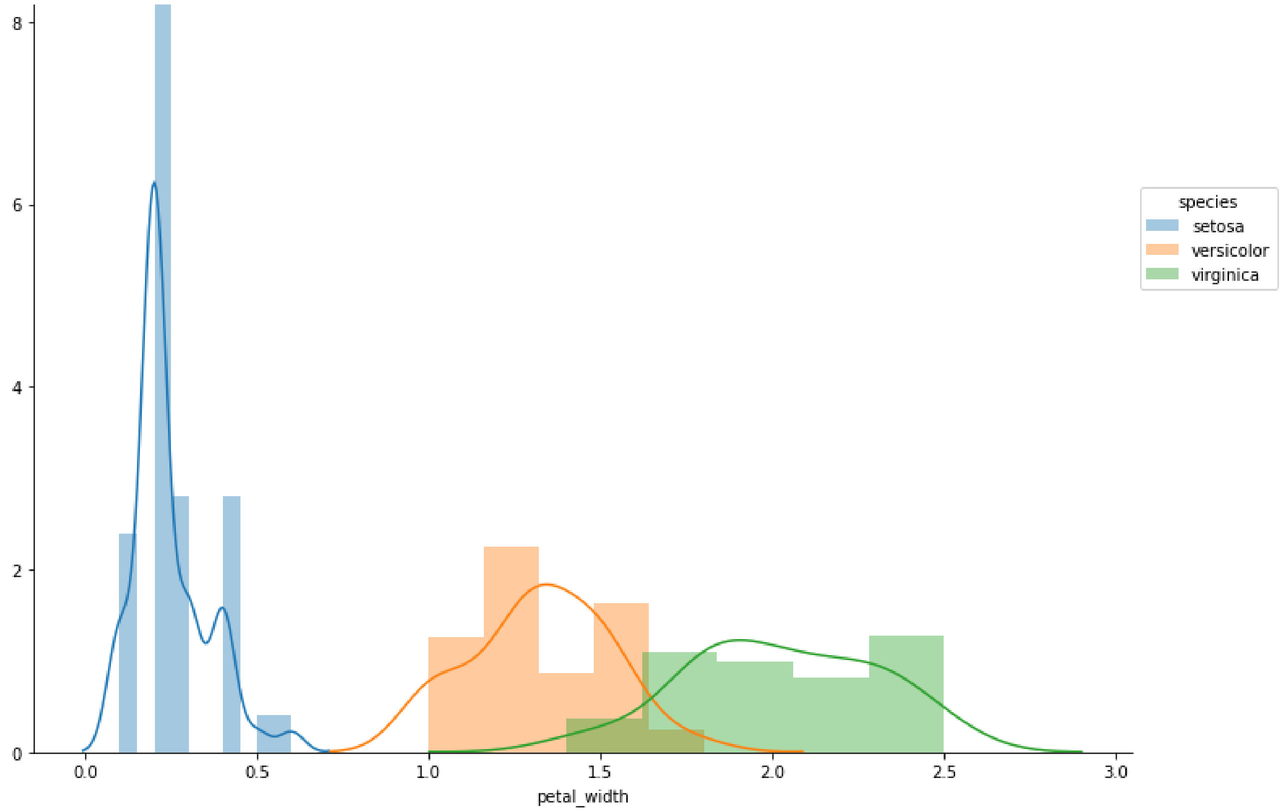
In [8]:

```
s.FacetGrid(data=iris,hue='species',size=10).map(s.distplot,'petal_length')
.add_legend()
mp.show()
```



In [9]:

```
s.FacetGrid(data=iris,hue='species',size=10).map(s.distplot,'petal_width').
add_legend()
mp.show()
```



Observations:

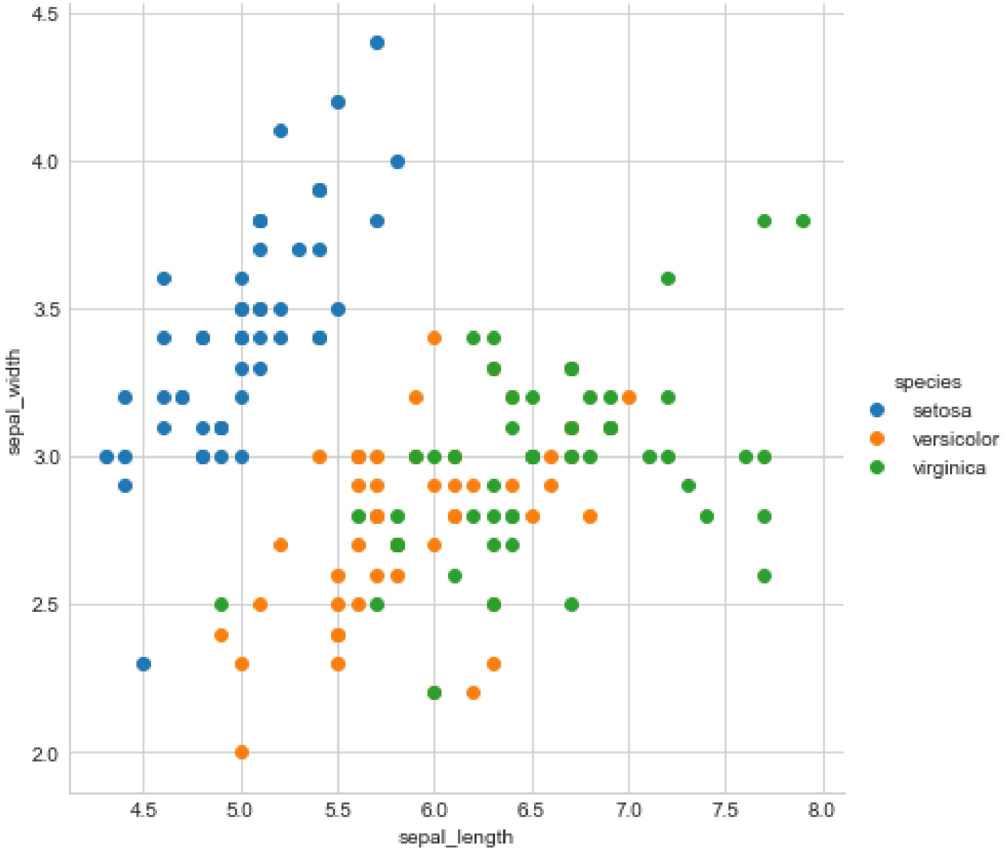
```
Univariate Analysis:

    Histograms:

-> Sepal Lenght: Most of the features are overlapping

-> Sepal Width: Among all the features there is more overlap by usin
g this attribute

-> Petal Length: Setosa is well seperated and there is overlap betwe
en the versicolor and
    virginica

-> Petal Width: There is overlap between the versicolor and
virginica
```

Bivariate Analysis:

```
Scatter Plots:

    -> Plotting of histogrmas is done by only one feature which help
s in the visualization of
       distribution of the feature

    -> Scatter plots helps in visualization of more than one feature
```

In [12]:

```
s.set_style('whitegrid')
s.FacetGrid(iris,hue='species',size=6).map(mp.scatter,'sepal_length','sepal_width').add_legend()
mp.show()
```

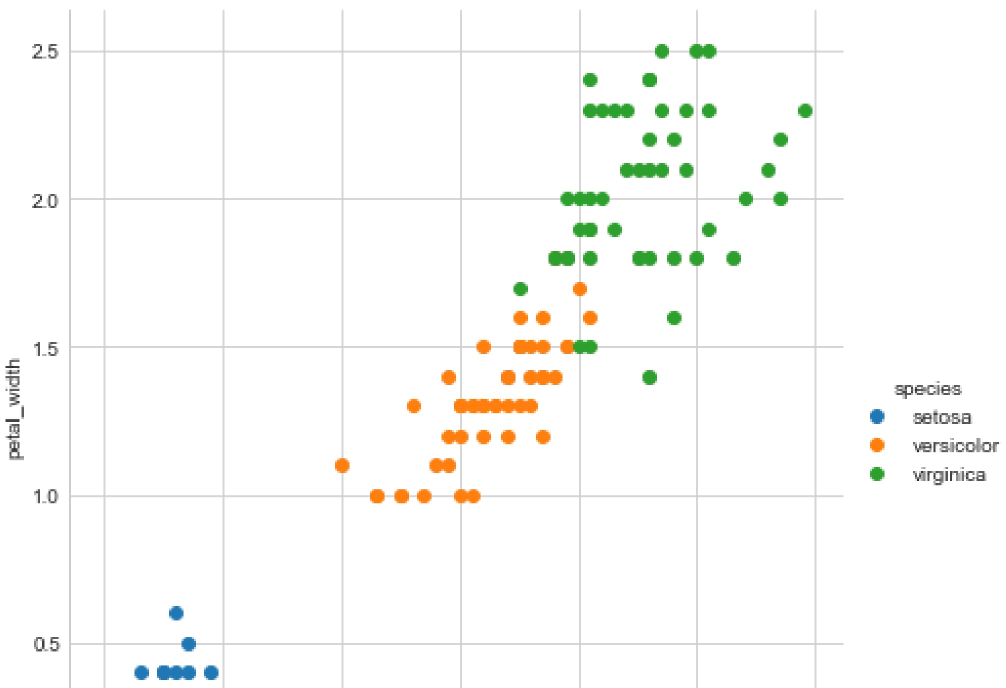


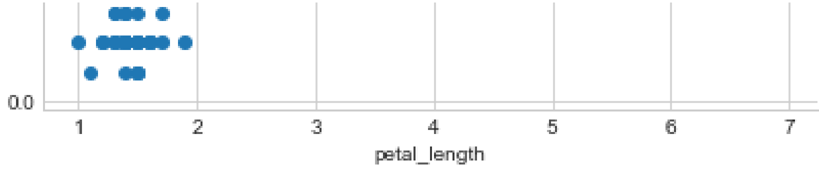
In [13]:

```
s.set_style('whitegrid')
s.FacetGrid(iris,hue='species',size=6).map(mp.scatter,'petal_length','petal_width').add_legend()
```

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x29999eee710>
```





Observation:

-> Sepal lenght and Sepal width has lot of overlapping

-> By using Petal length and Petal width, Setosa are well seperated but there is little
overlap between the Versicolor and Virginica

PAIR PLOTS:

-> We can also do the bivariate analysis using the pairplots

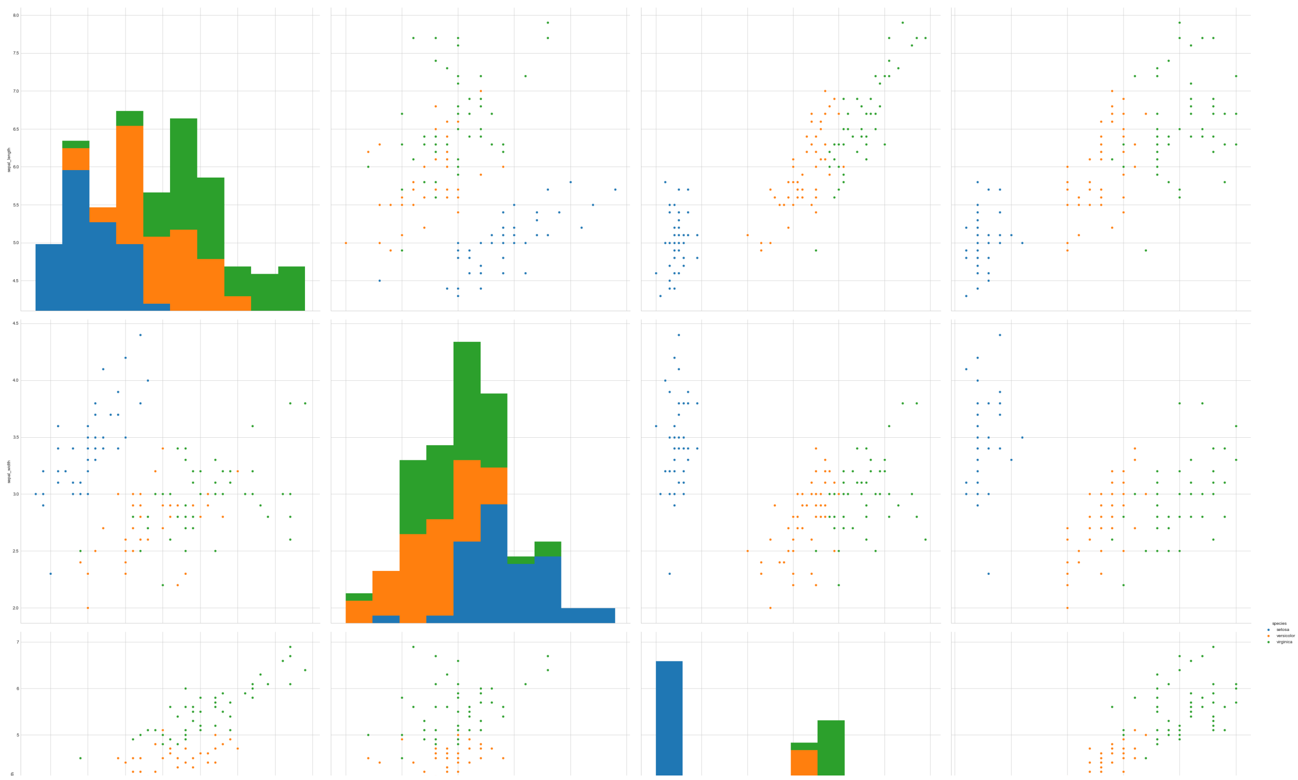
-> Pariplot uses all the features and plots the graph between every combination of features

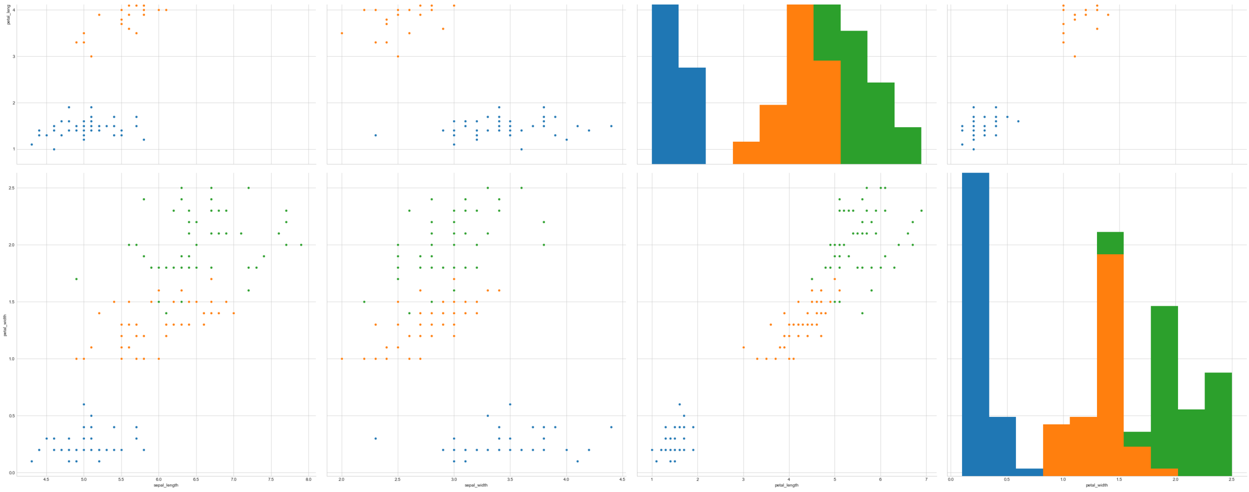
-> When there were high number of features we get very high number o f plots

-> So in high dimensional Pair plots are not the best choices for vi sualization

In [15]:

```
s.set_style('whitegrid')
s.pairplot(iris,hue='species',size=10)
mp.show()
```





Observation:

    -> From all the figures by using petal lenght and petal width is mor
    e useful to distinguish

    -> There is very less overlap by using the petal length and petal wi
    dth

Numerical Analysis:

    -> Mean : Average value

    -> Median : Middle value

    -> Standard Deviation : Variance

    -> Percentiles : Percentile value(1,100)

    -> Quantiles : Percentiles values at each quarter(25,50,75,100)

In [23]:

```
iris_setosa = iris.loc[iris['species']=='setosa']
iris_versicolor = iris.loc[iris['species']=='versicolor']
iris_virginica = iris.loc[iris['species']=='virginica']
```

Mean:

In [36]:

```
print("The mean values of the Setosa is: {}".format(iris_setosa.mean()))
print("The mean values of the Versicolor is: {}".format(iris_versicolor.mea
n()))
print("The mean values of the Virginica is: {}".format(iris_virginica.mean(
)))
```

```
The mean values of the Setosa is: sepal_length    5.006
sepal_width     3.418
petal_length    1.464
petal_width     0.244
```

```
petal_width     0.244
dtype: float64
The mean values of the Versicolor is: sepal_length    5.936
sepal_width     2.770
petal_length    4.260
petal_width     1.326
dtype: float64
The mean values of the Virginica is: sepal_length    6.588
sepal_width     2.974
petal_length    5.552
petal_width     2.026
dtype: float64
```

Median:

In [37]:

```
print("The median values of the Setosa is: {}".format(iris_setosa.median()))
print("The median values of the Versicolor is: {}".format(iris_versicolor.median()))
print("The median values of the Virginica is: {}".format(iris_virginica.median()))
```

```
The median values of the Setosa is: sepal_length    5.0
sepal_width     3.4
petal_length    1.5
petal_width     0.2
dtype: float64
The median values of the Versicolor is: sepal_length    5.90
sepal_width     2.80
petal_length    4.35
petal_width     1.30
dtype: float64
The median values of the Virginica is: sepal_length    6.50
sepal_width     3.00
petal_length    5.55
petal_width     2.00
dtype: float64
```

Standard Deviation:

In [38]:

```
print("The standard deviation of the Setosa is: {}".format(iris_setosa.std()))
print("The standard deviation of the Versicolor is: {}".format(iris_versicolor.std()))
print("The standard deviation of the Virginica is: {}".format(iris_virginica.std()))
```

```
The standard deviation of the Setosa is: sepal_length    0.352490
sepal_width     0.381024
petal_length    0.173511
petal_width     0.107210
dtype: float64
The standard deviation of the Versicolor is: sepal_length    0.516171
sepal_width     0.313798
petal_length    0.469911
petal_width     0.197753
```

```
dtype: float64
The standard deviation of the Virginica is: sepal_length    0.635880
sepal_width     0.322497
petal_length    0.551895
petal_width     0.274650
dtype: float64
```

Percentiles:

```
Gives the n^th percentile value, while the "n" range between 1 to 10
0
```

In [70]:

```
z = iris_setosa.drop(columns='species',axis=0)
z.columns
```

Out[70]:

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width'], dtype
='object')
```

In [75]:

```
l = np.arange(1,100,19)
l
```

Out[75]:

```
array([ 1, 20, 39, 58, 77, 96])
```

In [93]:

```
iris_setosa_columns = iris_setosa.drop(columns='species',axis=0)
l = np.arange(10,100,30)
print("Calculating percentile values for",l)
print("Iris Setosa")
for j in iris_setosa_columns.columns:
    print(j)
    print("----------------------------------------------------------------------
-----------------------")
    for n in l:
        print("The {}th percentile value is : {} ".format(n,np.percentile(i
ris_setosa[j],n)))
```

```
Calculating percentile values for [10 40 70]
Iris Setosa
sepal_length
---------------------------------------------------------------------------
-----------
The 10th percentile value is : 4.59
The 40th percentile value is : 4.96
The 70th percentile value is : 5.1
sepal_width
---------------------------------------------------------------------------
-----------
The 10th percentile value is : 3.0
The 40th percentile value is : 3.3600000000000003
The 70th percentile value is : 3.53
```

```
petal_length
------------------------------------------------------------------------------------
-----------
The 10th percentile value is : 1.3
The 40th percentile value is : 1.4
The 70th percentile value is : 1.5
petal_width
------------------------------------------------------------------------------------
-----------
The 10th percentile value is : 0.1
The 40th percentile value is : 0.2
The 70th percentile value is : 0.3
```

In [94]:

```
iris_versicolor_columns = iris_versicolor.drop(columns='species',axis=0)
l = np.arange(10,100,30)
print("Calculating percentile values for",l)
print("Iris Versicolor")
for j in iris_versicolor_columns.columns:
    print(j)
    print("------------------------------------------------------------------------------------
-----------------------")
    for n in l:
        print("The {}th percentile value is : {} ".format(n,np.percentile(i
ris_versicolor[j],n)))
```

```
Calculating percentile values for [10 40 70]
Iris Versicolor
sepal_length
------------------------------------------------------------------------------------
-----------
The 10th percentile value is : 5.38
The 40th percentile value is : 5.7
The 70th percentile value is : 6.2
sepal_width
------------------------------------------------------------------------------------
-----------
The 10th percentile value is : 2.3
The 40th percentile value is : 2.7
The 70th percentile value is : 3.0
petal_length
------------------------------------------------------------------------------------
-----------
The 10th percentile value is : 3.5900000000000003
The 40th percentile value is : 4.2
The 70th percentile value is : 4.5
petal_width
------------------------------------------------------------------------------------
-----------
The 10th percentile value is : 1.0
The 40th percentile value is : 1.3
The 70th percentile value is : 1.4299999999999997
```

In [95]:

```
iris_virginica_columns = iris_virginica.drop(columns='species',axis=0)
l = np.arange(10,100,30)
```

```
print("Calculating percentile values for",l)
print("Iris virginica")
for j in iris_virginica_columns.columns:
    print(j)
    print("----------------------------------------------------------------
-----------------------")
    for n in l:
        print("The {}th percentile value is : {} ".format(n,np.percentile(i
ris_virginica[j],n)))
```

```
Calculating percentile values for [10 40 70]
Iris virginica
sepal_length
---------------------------------------------------------------------------
-----------
The 10th percentile value is : 5.8
The 40th percentile value is : 6.4
The 70th percentile value is : 6.83
sepal_width
---------------------------------------------------------------------------
-----------
The 10th percentile value is : 2.5900000000000003
The 40th percentile value is : 2.9
The 70th percentile value is : 3.1
petal_length
---------------------------------------------------------------------------
-----------
The 10th percentile value is : 4.9
The 40th percentile value is : 5.36
The 70th percentile value is : 5.8
petal_width
---------------------------------------------------------------------------
-----------
The 10th percentile value is : 1.7900000000000003
The 40th percentile value is : 1.9
The 70th percentile value is : 2.2
```

Quantiles:

```
Calculates the percetile values at each quarter
```

In [96]:

```
iris_setosa_columns = iris_setosa.drop(columns='species',axis=0)
l = np.arange(0,100,25)
print("Calculating quantile values for",l)
print("Iris Setosa")
for j in iris_setosa_columns.columns:
    print(j)
    print("----------------------------------------------------------------
-----------------------")
    for n in l:
        print("The {}th percentile value is : {} ".format(n,np.percentile(i
ris_setosa['petal_length'],n)))
```

```
Calculating quantile values for [ 0 25 50 75]
Iris Setosa
```

```
Iris Setosa
sepal_length
----------------------------------------------------------------------------
-----------
The 0th percentile value is : 1.0
The 25th percentile value is : 1.4
The 50th percentile value is : 1.5
The 75th percentile value is : 1.5750000000000002
sepal_width
----------------------------------------------------------------------------
-----------
The 0th percentile value is : 1.0
The 25th percentile value is : 1.4
The 50th percentile value is : 1.5
The 75th percentile value is : 1.5750000000000002
petal_length
----------------------------------------------------------------------------
-----------
The 0th percentile value is : 1.0
The 25th percentile value is : 1.4
The 50th percentile value is : 1.5
The 75th percentile value is : 1.5750000000000002
petal_width
----------------------------------------------------------------------------
-----------
The 0th percentile value is : 1.0
The 25th percentile value is : 1.4
The 50th percentile value is : 1.5
The 75th percentile value is : 1.5750000000000002
```

In [98]:

```
iris_versicolor_columns = iris_versicolor.drop(columns='species',axis=0)
l = np.arange(0,100,25)
print("Calculating quantile values for",l)
print("Iris Versicolor")
for j in iris_versicolor_columns.columns:
    print(j)
    print("----------------------------------------------------------------
-----------------------")
    for n in l:
        print("The {}th percentile value is : {} ".format(n,np.percentile(i
ris_versicolor[j],n)))
```

```
Calculating quantile values for [ 0 25 50 75]
Iris Versicolor
sepal_length
----------------------------------------------------------------------------
-----------
The 0th percentile value is : 4.9
The 25th percentile value is : 5.6
The 50th percentile value is : 5.9
The 75th percentile value is : 6.3
sepal_width
----------------------------------------------------------------------------
-----------
The 0th percentile value is : 2.0
The 25th percentile value is : 2.525
The 50th percentile value is : 2.8
```

```
The 75th percentile value is : 3.0
petal_length
--------------------------------------------------------------------------------
-----------
The 0th percentile value is : 3.0
The 25th percentile value is : 4.0
The 50th percentile value is : 4.35
The 75th percentile value is : 4.6
petal_width
--------------------------------------------------------------------------------
-----------
The 0th percentile value is : 1.0
The 25th percentile value is : 1.2
The 50th percentile value is : 1.3
The 75th percentile value is : 1.5
```

In [99]:

```
iris_virginica_columns = iris_virginica.drop(columns='species',axis=0)
l = np.arange(0,100,25)
print("Calculating quantile values for",l)
print("Iris virginica")
for j in iris_virginica_columns.columns:
    print(j)
    print("------------------------------------------------------------------
-----------------------")
    for n in l:
        print("The {}th percentile value is : {} ".format(n,np.percentile(i
ris_virginica[j],n)))
```

```
Calculating quantile values for [ 0 25 50 75]
Iris virginica
sepal_length
--------------------------------------------------------------------------------
-----------
The 0th percentile value is : 4.9
The 25th percentile value is : 6.2250000000000005
The 50th percentile value is : 6.5
The 75th percentile value is : 6.9
sepal_width
--------------------------------------------------------------------------------
-----------
The 0th percentile value is : 2.2
The 25th percentile value is : 2.8
The 50th percentile value is : 3.0
The 75th percentile value is : 3.1750000000000003
petal_length
--------------------------------------------------------------------------------
-----------
The 0th percentile value is : 4.5
The 25th percentile value is : 5.1
The 50th percentile value is : 5.55
The 75th percentile value is : 5.875000000000001
petal_width
--------------------------------------------------------------------------------
-----------
The 0th percentile value is : 1.4
The 25th percentile value is : 1.8
The 50th percentile value is : 2.0
```

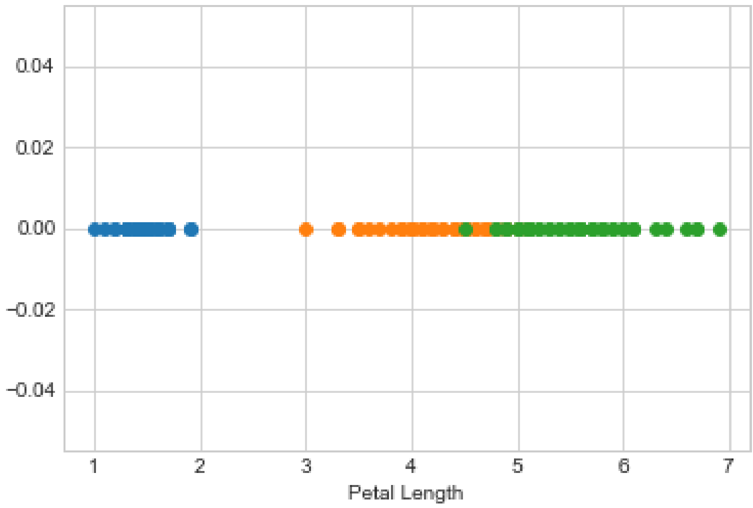
```
The 75th percentile value is : 2.3
```

Spread of the attributes:

Petal Lenght Spread:

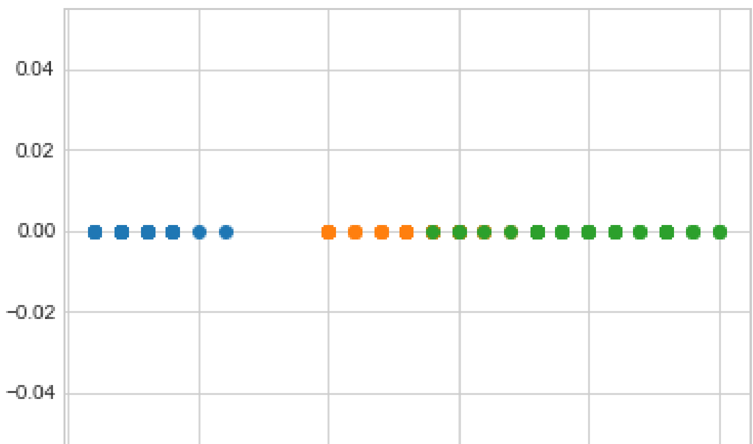
In [101]:

```
mp.plot(iris_setosa['petal_length'],np.zeros_like(iris_setosa['petal_length
']),'o',label='irissetosa')
mp.plot(iris_versicolor['petal_length'],np.zeros_like(iris_versicolor['peta
l_length']),'o',label='irisversicolor')
mp.plot(iris_virginica['petal_length'],np.zeros_like(iris_virginica['petal_
length']),'o',label='irisvirginica')
mp.xlabel("Petal Length")
mp.show()
```



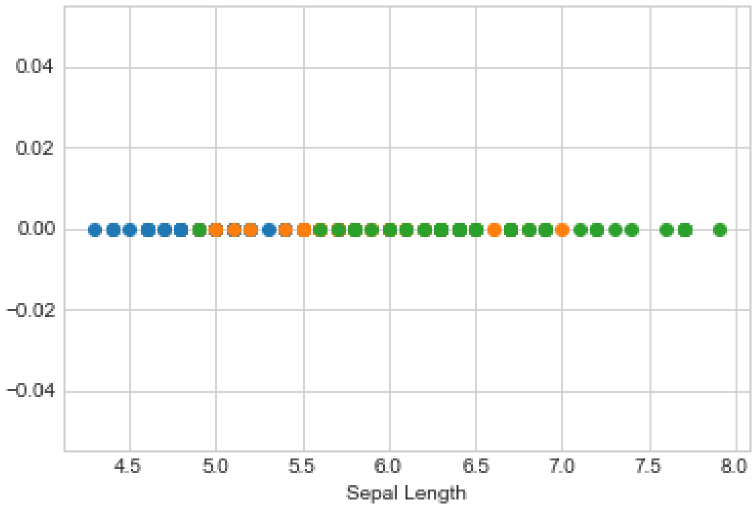
In [102]:

```
mp.plot(iris_setosa['petal_width'],np.zeros_like(iris_setosa['petal_width']
),'o',label='irissetosa')
mp.plot(iris_versicolor['petal_width'],np.zeros_like(iris_versicolor['petal
_width']),'o',label='irisversicolor')
mp.plot(iris_virginica['petal_width'],np.zeros_like(iris_virginica['petal_w
idth']),'o',label='irisvirginica')
mp.xlabel("Petal Width")
mp.show()
```



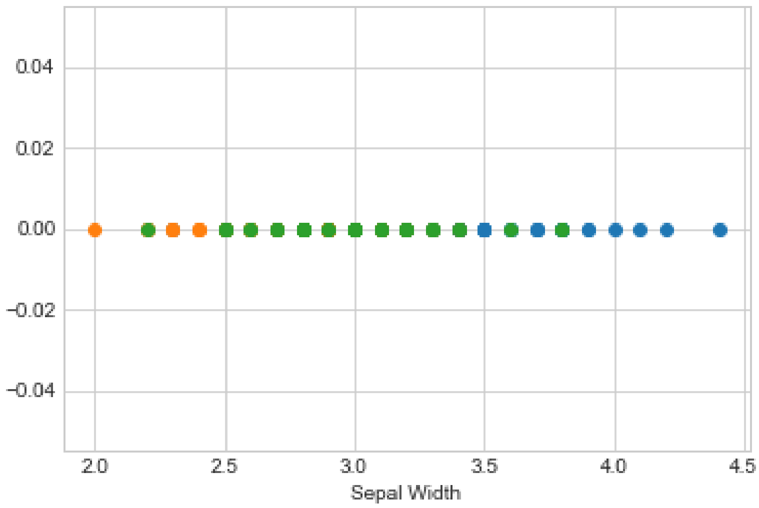
In [103]:

```
mp.plot(iris_setosa['sepal_length'],np.zeros_like(iris_setosa['sepal_length']),'o',label='irissetosa')
mp.plot(iris_versicolor['sepal_length'],np.zeros_like(iris_versicolor['sepal_length']),'o',label='irisversicolor')
mp.plot(iris_virginica['sepal_length'],np.zeros_like(iris_virginica['sepal_length']),'o',label='irisvirginica')
mp.xlabel("Sepal Length")
mp.show()
```



In [104]:

```
mp.plot(iris_setosa['sepal_width'],np.zeros_like(iris_setosa['sepal_width']),'o',label='irissetosa')
mp.plot(iris_versicolor['sepal_width'],np.zeros_like(iris_versicolor['sepal_width']),'o',label='irisversicolor')
mp.plot(iris_virginica['sepal_width'],np.zeros_like(iris_virginica['sepal_width']),'o',label='irisvirginica')
mp.xlabel("Sepal Width")
mp.show()
```



Observation:

-> The spread of the petal length and petal width are more useful than other aattributes

PDF,CDF:

PDF: We can get the frequency at any point

CDF: At any point, we can get the percentage of data that is under the point

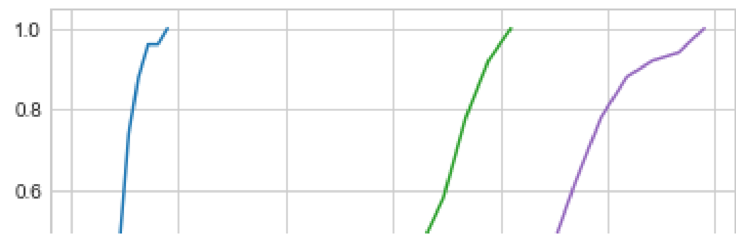
In [117]:

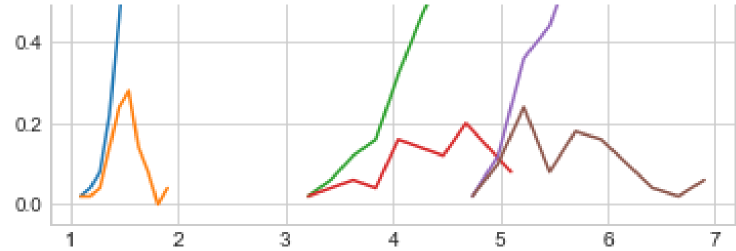
```
counts,binedges = np.histogram(iris_setosa['petal_length'],bins=10,density=True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
print(pdf)
print(binedges)
print(cdf)
mp.plot(binedges[1:],cdf)
mp.plot(binedges[1:],pdf)
counts,binedges = np.histogram(iris_versicolor['petal_length'],bins=10,density=True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
print(pdf)
print(binedges)
print(cdf)
mp.plot(binedges[1:],cdf)
mp.plot(binedges[1:],pdf)
counts,binedges = np.histogram(iris_virginica['petal_length'],bins=10,density=True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
print(pdf)
print(binedges)
print(cdf)
mp.plot(binedges[1:],cdf)
mp.plot(binedges[1:],pdf)
```

```
[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 0.04 0.08 0.22 0.46 0.74 0.88 0.96 0.96 1.  ]
[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 ]
[0.02 0.06 0.12 0.16 0.32 0.46 0.58 0.78 0.92 1.  ]
[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 0.12 0.36 0.44 0.62 0.78 0.88 0.92 0.94 1.  ]
```

Out[117]:

```
[<matplotlib.lines.Line2D at 0x18c97c2b9e8>]
```





Difference between Mean and Median:

In [109]:

```
print("Mean calculations")
print("The mean of the IRIS Setosa before adding noise:")
print(iris_setosa['petal_length'].mean())
print("The mean of the IRIS Setosa after adding noise:")
print(np.mean(np.append(iris_setosa['petal_length'],100)))
print("-------------------------------------------------")
print("Median calculations")
print("The median of the IRIS Setosa before adding noise:")
print(iris_setosa['petal_length'].median())
print("The median of the IRIS Setosa after adding noise:")
print(np.median(np.append(iris_setosa['petal_length'],100)))
```

```
Mean calculations
The mean of the IRIS Setosa before adding noise:
1.464
The mean of the IRIS Setosa after adding noise:
3.396078431372549
-------------------------------------------------
Median calculations
The median of the IRIS Setosa before adding noise:
1.5
The median of the IRIS Setosa after adding noise:
1.5
```

Observation:

    -> The mean is deviated by the outliers or noise

    -> The median is robust to outliers or noise

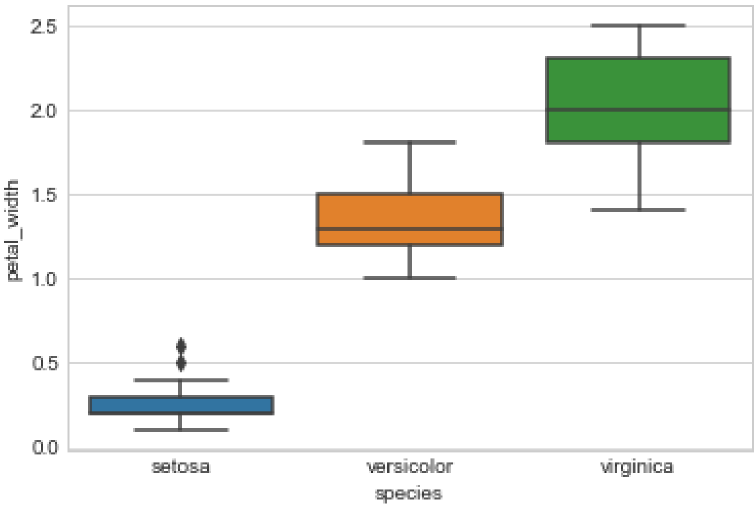
BOX PLOTS:

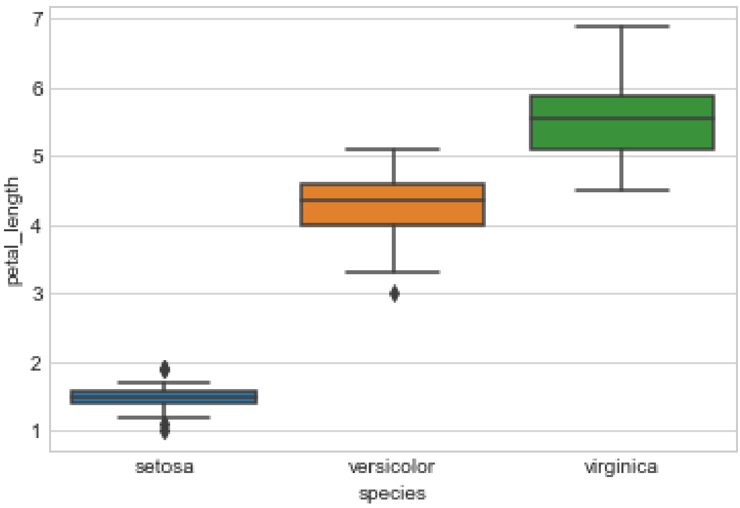
    From box plots we can get the quantile values

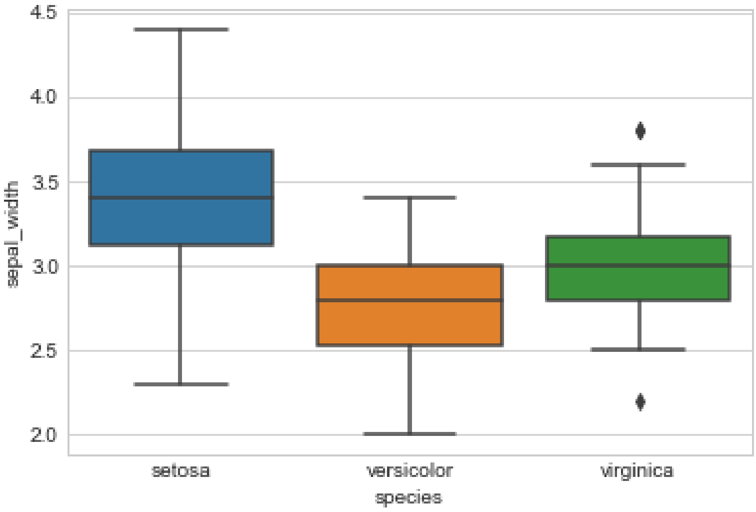
In [112]:

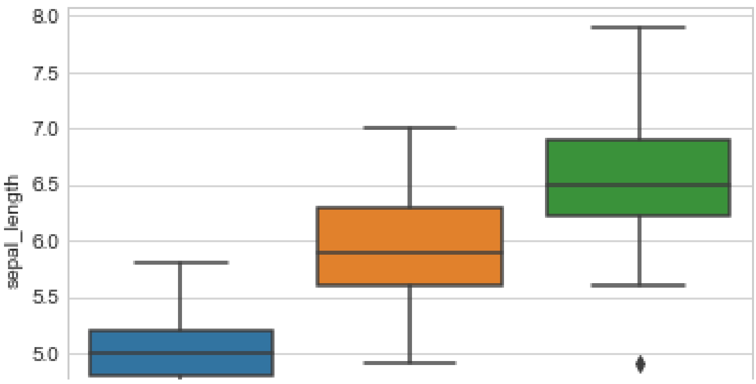
```
mp.figure(211)
s.boxplot(x='species',y='petal_width',data=iris)
mp.figure(212)
s.boxplot(x='species',y='petal_length',data=iris)
mp.figure(213)
s.boxplot(x='species',y='sepal_width',data=iris)
mp.figure(214)
s.boxplot(x='species',y='sepal_length',data=iris)
```

```
mp.show()
```


2.5
2.0
1.5
1.0
0.5
0.0
petal_width
setosa
versicolor
virginica
species


7
6
5
4
3
2
1
petal_length
setosa
versicolor
virginica
species


4.5
4.0
3.5
3.0
2.5
2.0
sepal_width
setosa
versicolor
virginica
species

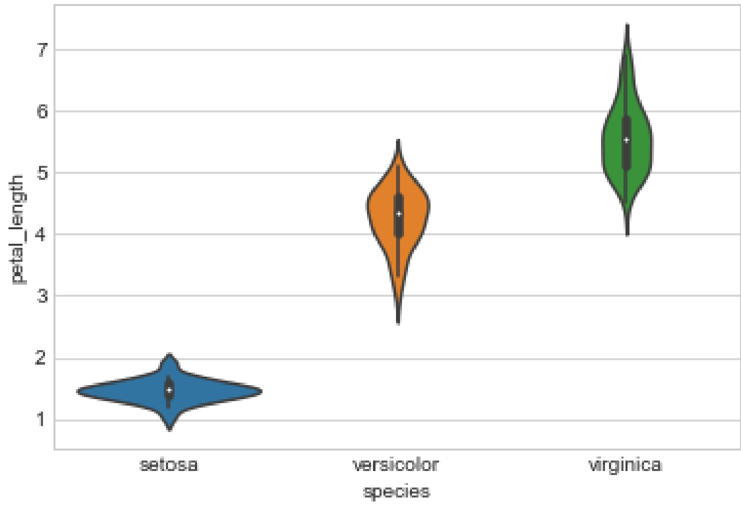

8.0
7.5
7.0
6.5
6.0
5.5
5.0
sepal_length

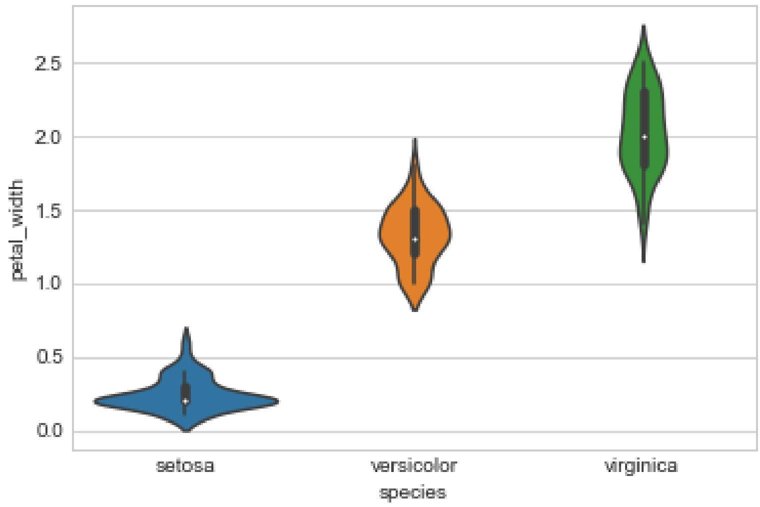
VIOLIN PLOTS:

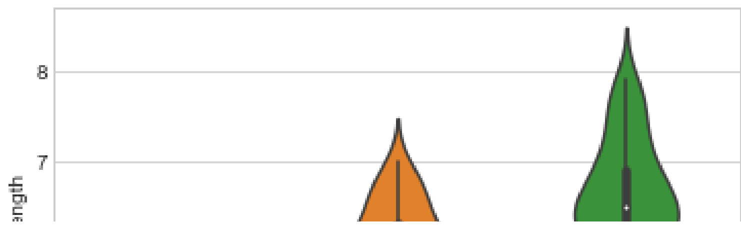
    These plots are the combination of Whisker plots and their PDF

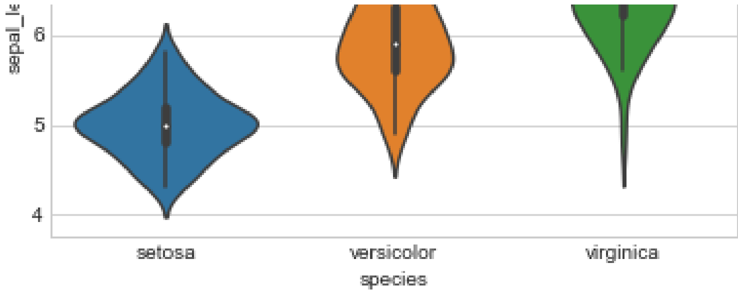
In [114]:

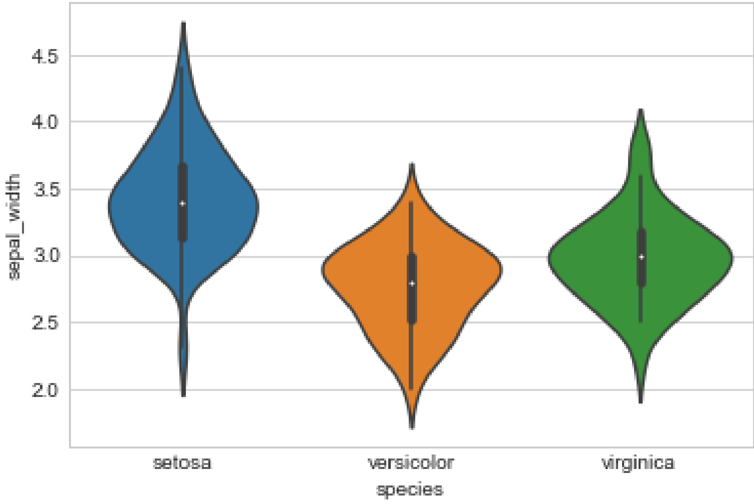
```
mp.figure(211)
s.violinplot(x='species',y='petal_length',data=iris)
mp.figure(212)
s.violinplot(x='species',y='petal_width',data=iris)
mp.figure(213)
s.violinplot(x='species',y='sepal_length',data=iris)
mp.figure(214)
s.violinplot(x='species',y='sepal_width',data=iris)
mp.show()
```











CONCLUSION:

```
-> Performed the numerical analysis and graphical analysis on each a
ttribute

-> The analysis between the petal length and petal width are more us
eful than the other attributes

-> Different analysis results of petal length and petal width are mo
re useful in determining the type of the species
```