Exploratory Data Analysis on Iris Dataset

About Data

Iris dataset is taken from https://www.kaggle.com/arshid/iris-flower-dataset (https://www.kaggle.com/arshid/iris-flower-dataset)

1) The dataset contains 4 features and 3 classes

2) Number of Instances: 150

3) Number of Attributes: 4 (including the class atribute)

4) Attribute Information: a) sepal_length (numerical) b) sepal_width (numerical) c) petal_length (numerical) d) petal_width (numerical)

5) Missing Attribute Values: None

Objective:

Classify a new flower as belonging to one of the 3 classes given the 4 features.

Loading the dataset

In [1]:

```
# Importing necessary libraries

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

#Load iris.csv into a pandas dataFrame.
iris_df = pd.read_csv("iris.csv")
iris_df
```

Out[1]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

Analysing the Data (High level statistics of the dataset)

In [2]:

```
# (Q) how many data-points and features?
print (iris_df.shape)
```

```
(150, 5)
```

Observation:

1) Number of data points corresponding to each feature in the data set = 150

2) Numer of features in the data set = 5 inclusing class label

In [3]:

```
#(Q) What are the column names in our dataset?
print (iris_df.columns)
```

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

Observation:

1) We have 3 independent variables ('sepal_length','sepal_width', 'petal_length','petal_width')

2) 1 dependent variable ('species')

In [4]:

```
#(Q) How many data points for each class are present?

iris_df["species"].value_counts()
```

Out[4]:

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

Observation

we have three classes

1) versicolor 2) setosa 3) virginica

No-of data points per class

1) We have 50 entries of class "versicolor"

2) We have 50 entries of class "setosa"

3) We have 50 entries of class "virginica"

we can observe that the data is balanced.

In [5]:

```
# Checking wether data has any missing values

print(iris_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

Observation:

Dataset doen't have any missing values. So no need to handle missing data

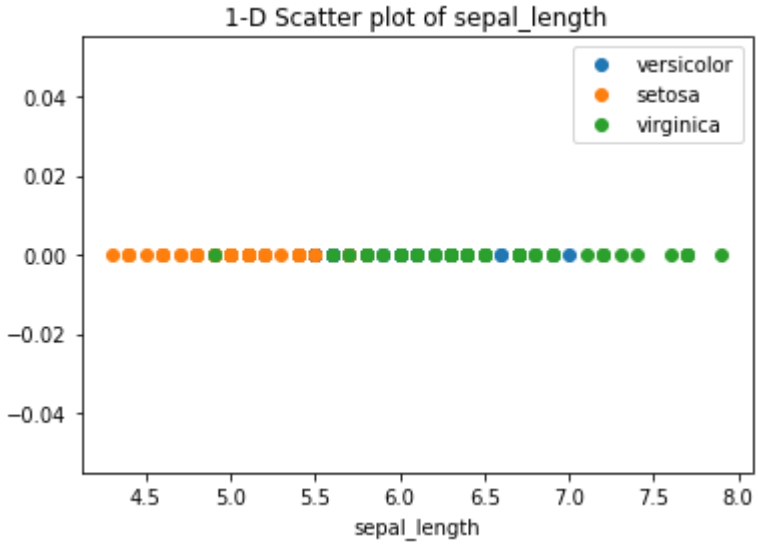
1. Univaraite analysis

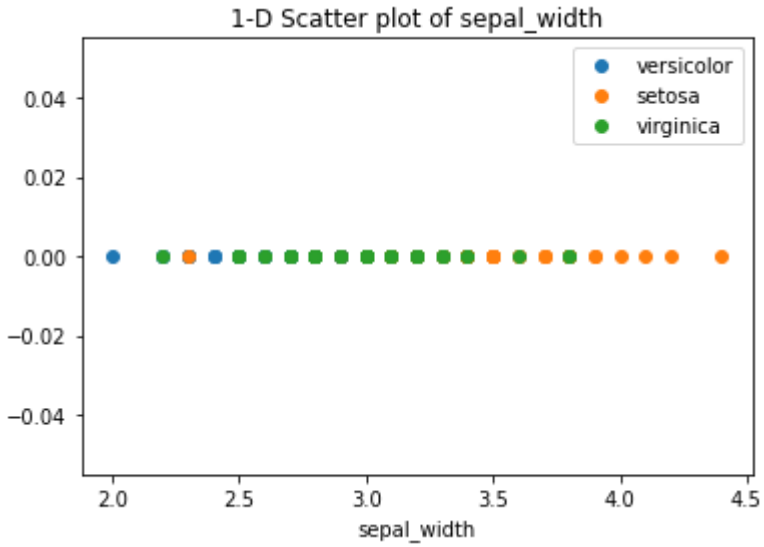
we will do univaraite analysis to understand which features are useful for classification of class label 'species'

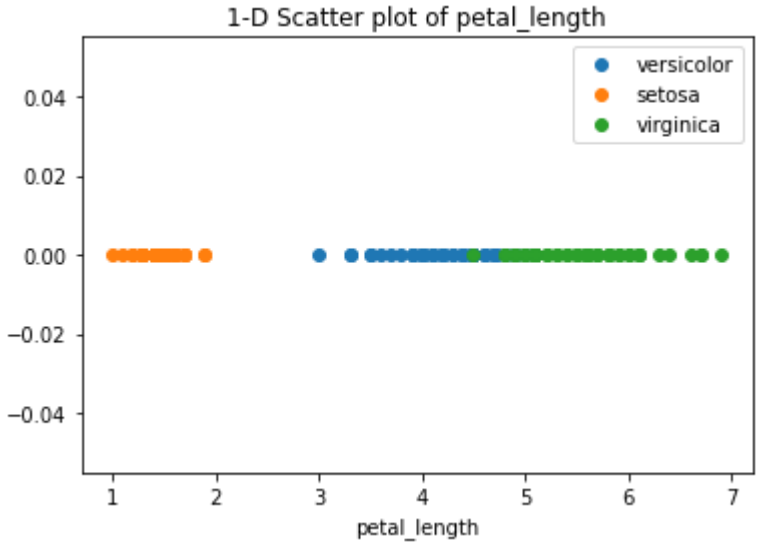
1.1 1-D Scatter plots

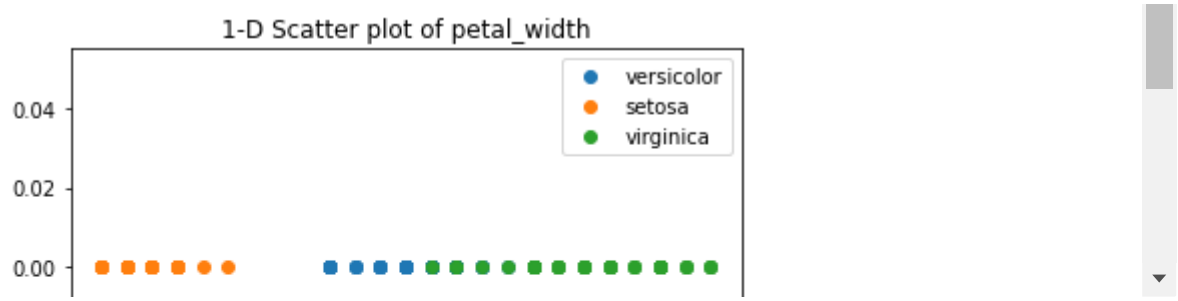
In [6]:

```
import numpy as np

versicolor = iris_df.loc[iris_df["species"] == "versicolor"]
setosa = iris_df.loc[iris_df["species"] == "setosa"]
virginica = iris_df.loc[iris_df["species"] == "virginica"]


"""To Draw 1-D Scatter Plot we are making x-axis = Feature, Y-axis = zeros """

#1-D scatter plot of sepal_length
plt.figure(1)
plt.plot(versicolor["sepal_length"], np.zeros_like(versicolor['sepal_length']), 'o', label
plt.plot(setosa["sepal_length"], np.zeros_like(setosa['sepal_length']), 'o', label = 'setos
plt.plot(virginica["sepal_length"], np.zeros_like(virginica['sepal_length']), 'o', label =
plt.title("1-D Scatter plot of sepal_length")
plt.xlabel("sepal_length")
plt.legend()
plt.show()

#1-D scatter plot of sepal_width
plt.figure(1)
plt.plot(versicolor["sepal_width"], np.zeros_like(versicolor['sepal_width']), 'o', label =
plt.plot(setosa["sepal_width"], np.zeros_like(setosa['sepal_width']), 'o', label = 'setosa'
plt.plot(virginica["sepal_width"], np.zeros_like(virginica['sepal_width']), 'o', label = 'v
plt.title("1-D Scatter plot of sepal_width")
plt.xlabel("sepal_width")
plt.legend()
plt.show()

#1-D scatter plot of petal_length
plt.figure(1)
plt.plot(versicolor["petal_length"], np.zeros_like(versicolor['petal_length']), 'o', label
plt.plot(setosa["petal_length"], np.zeros_like(setosa['petal_length']), 'o', label = 'setos
plt.plot(virginica["petal_length"], np.zeros_like(virginica['petal_length']), 'o', label =
plt.title("1-D Scatter plot of petal_length")
plt.xlabel("petal_length")
plt.legend()
plt.show()

#1-D scatter plot of petal_width
plt.figure(1)
plt.plot(versicolor["petal_width"], np.zeros_like(versicolor['petal_width']), 'o', label =
plt.plot(setosa["petal_width"], np.zeros_like(setosa['petal_width']), 'o', label = 'setosa'
plt.plot(virginica["petal_width"], np.zeros_like(virginica['petal_width']), 'o', label = 'v
plt.title("1-D Scatter plot of petal_width")
plt.xlabel("petal_width")
plt.legend()
plt.show()
```


1-D Scatter plot of sepal_length
versicolor
setosa
virginica
0.04
0.02
0.00
−0.02
−0.04
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
sepal_length


1-D Scatter plot of sepal_width
versicolor
setosa
virginica
0.04
0.02
0.00
−0.02
−0.04
2.0
2.5
3.0
3.5
4.0
4.5
sepal_width


1-D Scatter plot of petal_length
versicolor
setosa
virginica
0.04
0.02
0.00
−0.02
−0.04
1
2
3
4
5
6
7
petal_length



Oberservation:

1)From Figure(1): In 1-D scatter plot of sepal_length, the sepal_length data points belong to versicolor,setosa and virginica are mosly overlaped. We cannot interpret any conclusion from the figure(1).

2)From Figure(2): In 1-D scatter plot of sepal_width, the sepal_width data points belong to versicolor,setosa and virginica are overlaped. We cannot interpret any conclusion from the figure(2).

3)From Figure(3): In 1-D scatter plot of petal_length, we can observe the following

```
                a) setosa petal_length is less when compared to versicolor and virgi
    nica. Mostly petal_length of setosa
                  lies below 2 units


                b) virginica has more petal_length when compared to setosa and versi
    color
```

4)From Figure(4): In 1-D scatter plot of petal_width, we can observe the following

```
                a) setosa petal_width is less when compared to versicolor and virgin
    ica. Mostly petal_length of setosa
                  lies below 0.6 units


                b) virginica has more petal_width when compared to setosa and versic
    olor
```

We will proceed our further analysis using Histogram, PDF, CDF

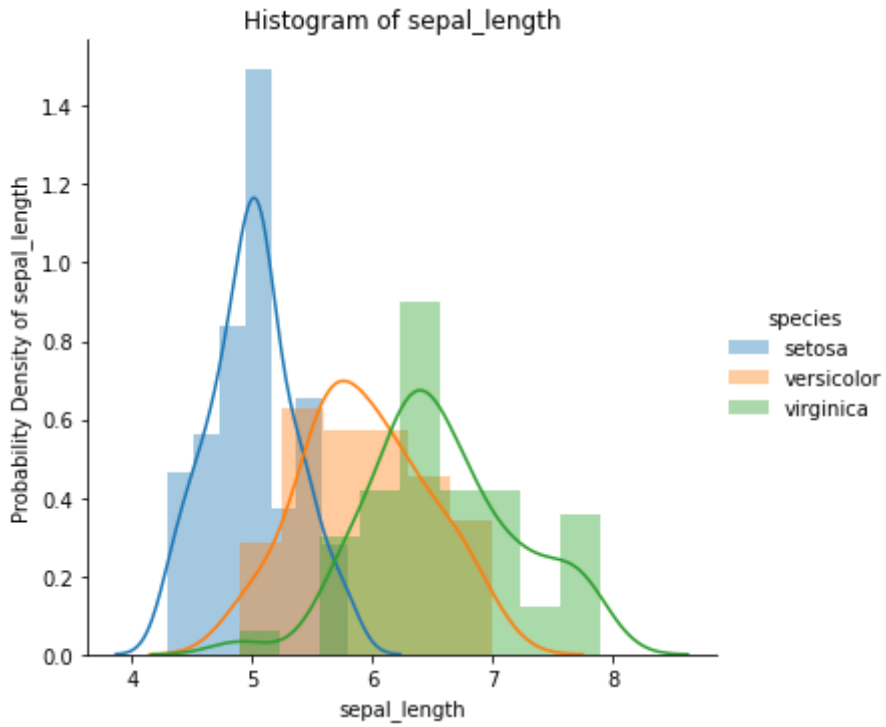
In [ ]:

1.2 Histogram and PDF

In [7]:

```
# Histogram and PDF for the Independent variable 'sepal_length'

sns.FacetGrid(iris_df, hue="species", size=5) \
   .map(sns.distplot, "sepal_length") \
   .add_legend();
plt.title("Histogram of sepal_length")
plt.ylabel("Probability Density of sepal_length")
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\axisgrid.py:243: UserWarn
ing: The `size` parameter has been renamed to `height`; please update your c
ode.
  warnings.warn(msg, UserWarning)
```



Observation:

PDFs and Histograms drawn considering 'sepal_length' feature is overlapped for setosa, versicolor and virginica

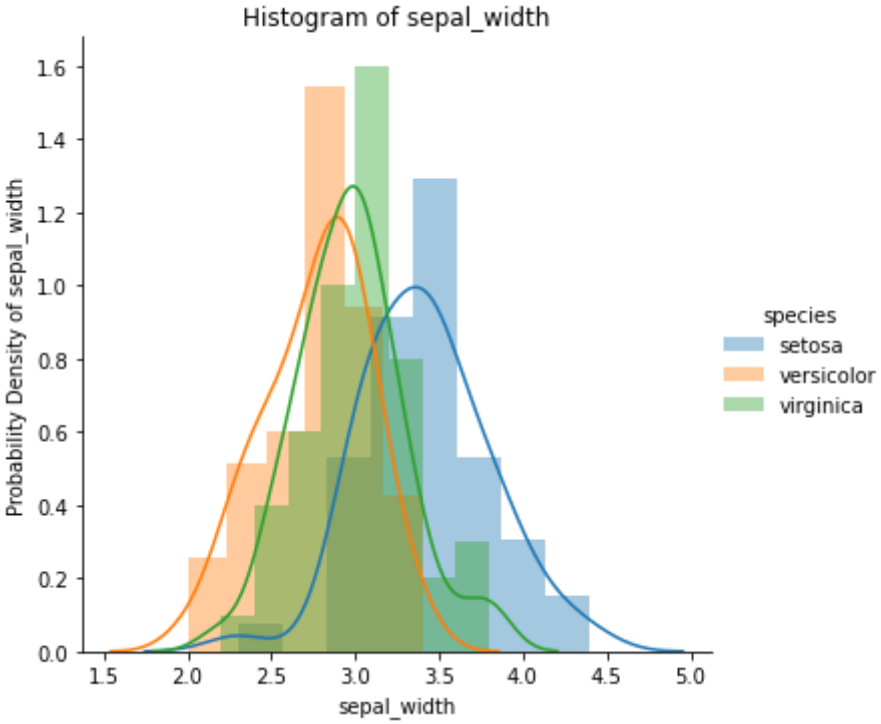
1) However from above PDF we can say that sepal_length for setosa is less when compared to versicolor and virginica

we cannot classify flowers perfectly based on 'sepal_length' alone

In [8]:

```
# Histogram and PDF for the Independent variable 'sepal_width'

sns.FacetGrid(iris_df, hue="species", size=5) \
   .map(sns.distplot, "sepal_width") \
   .add_legend();
plt.title("Histogram of sepal_width")
plt.ylabel("Probability Density of sepal_width")
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\axisgrid.py:243: UserWarn
ing: The `size` parameter has been renamed to `height`; please update your c
ode.
  warnings.warn(msg, UserWarning)
```



Observation:

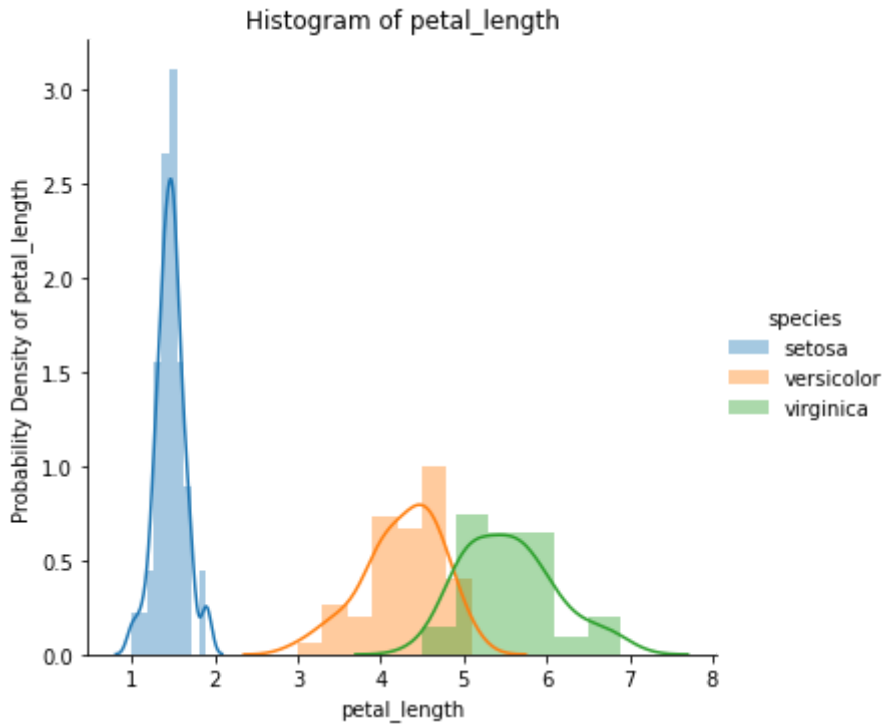
PDFs and Histograms drawn considering 'sepal_width' feature is almost completley overlapped for setosa, versicolor and virginica

we cannot classify flowers perfectly based on 'sepal_width' alone

In [9]:

```
# Histogram and PDF for the Independent variable 'petal_length'

sns.FacetGrid(iris_df, hue="species", size=5) \
   .map(sns.distplot, "petal_length") \
   .add_legend();
plt.title("Histogram of petal_length")
plt.ylabel("Probability Density of petal_length")
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\axisgrid.py:243: UserWarn
ing: The `size` parameter has been renamed to `height`; please update your c
ode.
  warnings.warn(msg, UserWarning)
```



Observation:

1) PDFs and Histograms drawn considering 'petal_length' feature is overlapped for versicolor and virginica. But the PDF of setosa is well seperated from for versicolor and virginica.

2) From the above PDFs we can say petal_length of setosa lies between 1 and 2 units

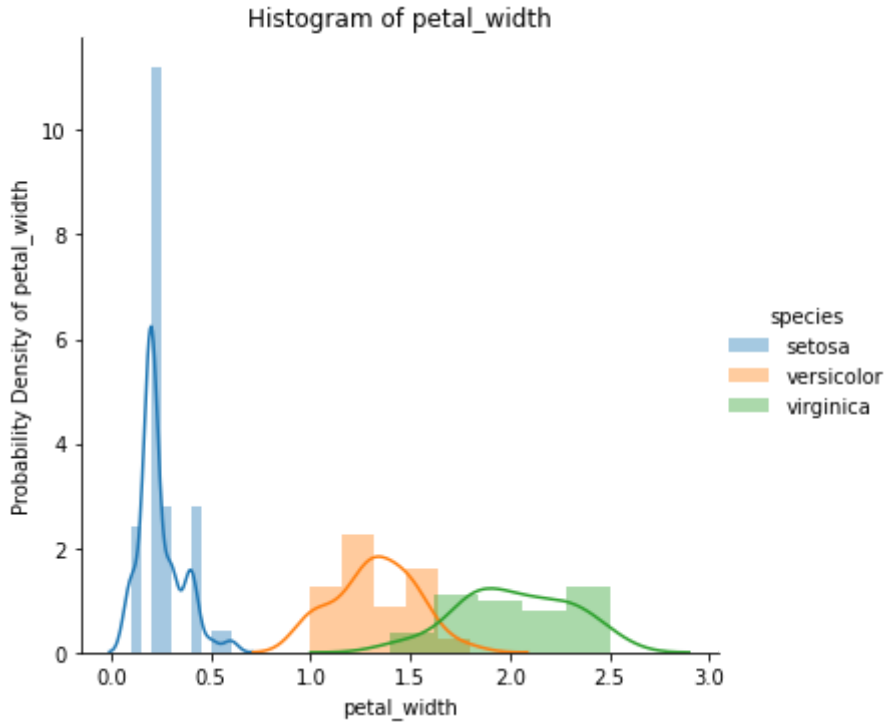
3) We can say petal_length of virginica is more when compared to setosa and versicolor

we can seperate/classify setosa flower perfectly based on 'petal_length' alone

In [10]:

```
# Histogram and PDF for the Independent variable 'petal_width'

sns.FacetGrid(iris_df, hue="species", size=5) \
   .map(sns.distplot, "petal_width") \
   .add_legend();
plt.title("Histogram of petal_width")
plt.ylabel("Probability Density of petal_width")
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\axisgrid.py:243: UserWarn
ing: The `size` parameter has been renamed to `height`; please update your c
ode.
  warnings.warn(msg, UserWarning)
```



Observation:

1) PDFs and Histograms drawn considering 'petal_width' feature is overlapped for versicolor and virginica. But the PDF of setosa is well seperated from for versicolor and virginica.

2) From the above PDFs we can say petal_width of setosa lies below 0.5 units

3) We can say petal_width of virginica is more when compared to setosa and versicolor

we can seperate/classify setosa flower perfectly based on 'petal_width' alone

We can conclude that the features 'petal_length' and 'petal_width' are very useful to classify flowers (setosa, versicolor and virginica)

We will proceed our further analysis using CDF to get more results

Conclusion from Histogram and PDFs of all independent variables:

1) petal_length of setosa lies between 1 and 2 units and petal_length of virginica is more when compared to setosa and versicolor

2) petal_width of setosa lies below 0.5 units and petal_width of virginica is more when compared to setosa and versicolor

In [ ]:

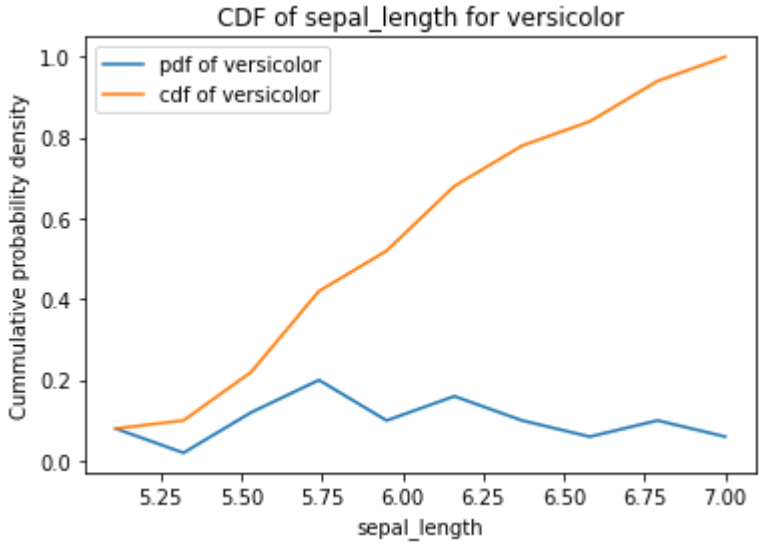
```
```

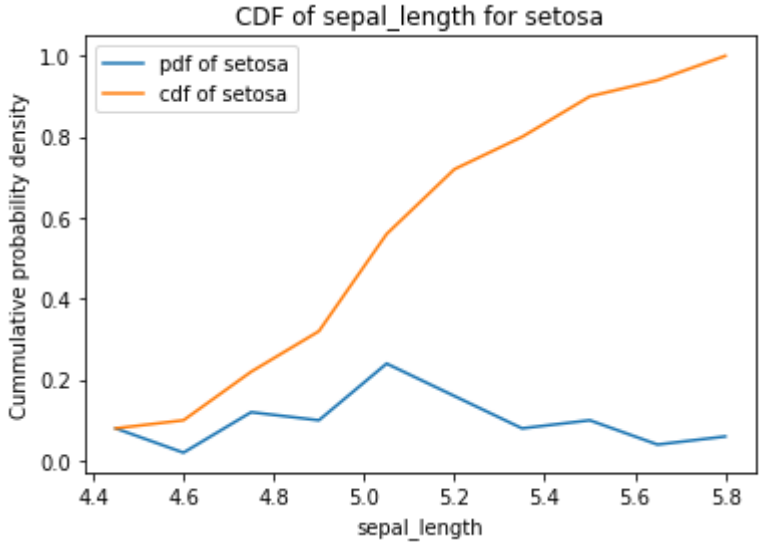
1.3 CDF

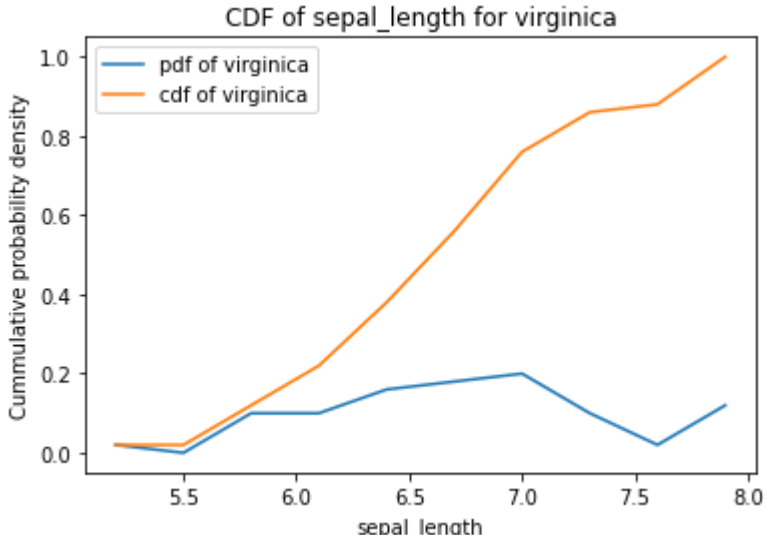
In [11]:

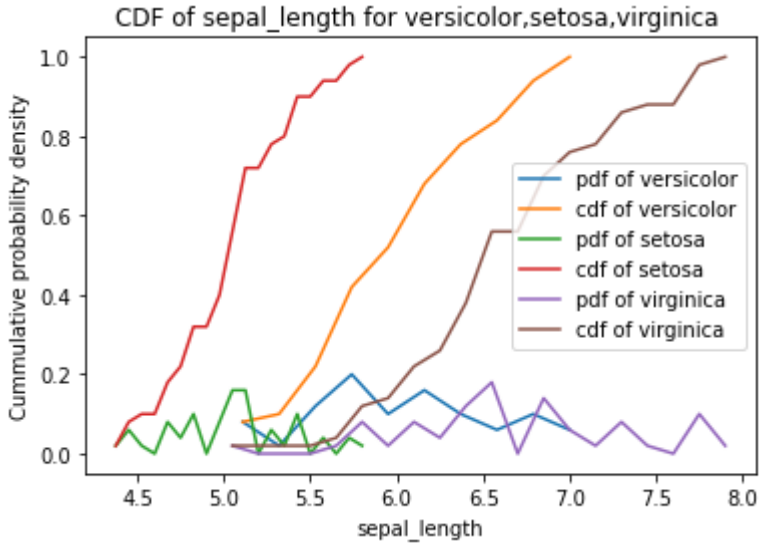
```
# CDF for the Independent variable 'sepal_length'

versicolor = iris_df.loc[iris_df["species"] == "versicolor"]
setosa = iris_df.loc[iris_df["species"] == "setosa"]
virginica = iris_df.loc[iris_df["species"] == "virginica"]


# CDF of sepal_length for versicolor
plt.figure(1)
counts, bin_edges = np.histogram(versicolor['sepal_length'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of versicolor');
plt.plot(bin_edges[1:], cdf, label = 'cdf of versicolor')
plt.xlabel("sepal_length")
plt.ylabel("Cummulative probability density")
plt.title("CDF of sepal_length for versicolor")
plt.legend()


# CDF of sepal_length for setosa
plt.figure(2)
counts, bin_edges = np.histogram(setosa['sepal_length'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of setosa');
plt.plot(bin_edges[1:], cdf, label = 'cdf of setosa')
plt.xlabel("sepal_length")
plt.ylabel("Cummulative probability density")
plt.title("CDF of sepal_length for setosa")
plt.legend()


# CDF of sepal_length for virginica
plt.figure(3)
counts, bin_edges = np.histogram(virginica['sepal_length'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of virginica');
plt.plot(bin_edges[1:], cdf, label = 'cdf of virginica')
plt.xlabel("sepal_length")
plt.ylabel("Cummulative probability density")
plt.title("CDF of sepal_length for virginica")
plt.legend()



# CDFs of versicolor, setosa,virginica for the feature sepal_length in a single plot
plt.figure(4)

counts, bin_edges = np.histogram(versicolor['sepal_length'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of versicolor');
plt.plot(bin_edges[1:], cdf, label = 'cdf of versicolor')
```

```
counts, bin_edges = np.histogram(setosa['sepal_length'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of setosa');
plt.plot(bin_edges[1:],cdf, label = 'cdf of setosa')


counts, bin_edges = np.histogram(virginica['sepal_length'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of virginica');
plt.plot(bin_edges[1:],cdf, label = 'cdf of virginica')



plt.xlabel("sepal_length")
plt.ylabel("Cummulative probability density")
plt.title("CDF of sepal_length for versicolor,setosa,virginica ")
plt.legend()
plt.show()
```









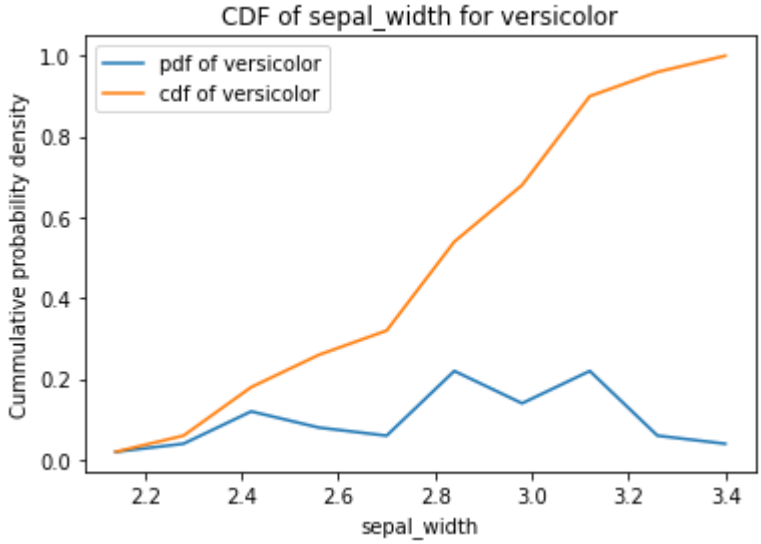
Observation:

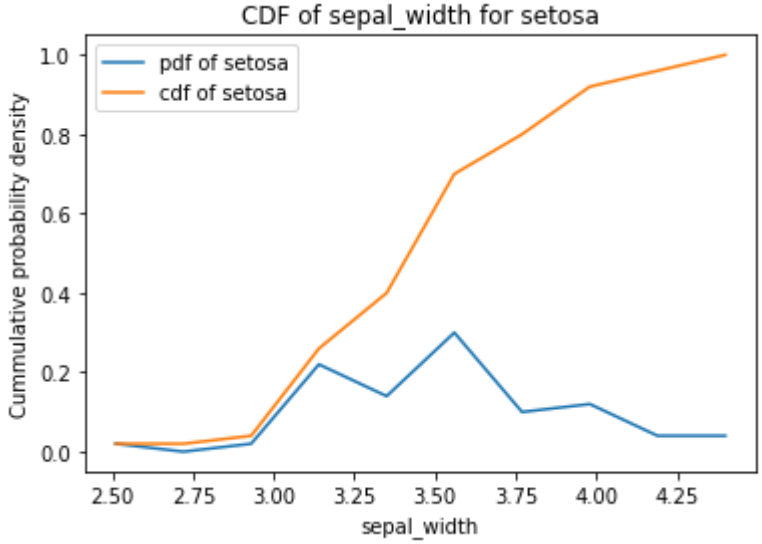
1) From CDF we can observer that 100% of the setosa flowers have sepal_length less than 5.5. We can say if the sepal_length is <= 5.5 units, then the probability of the flower being setosa is high

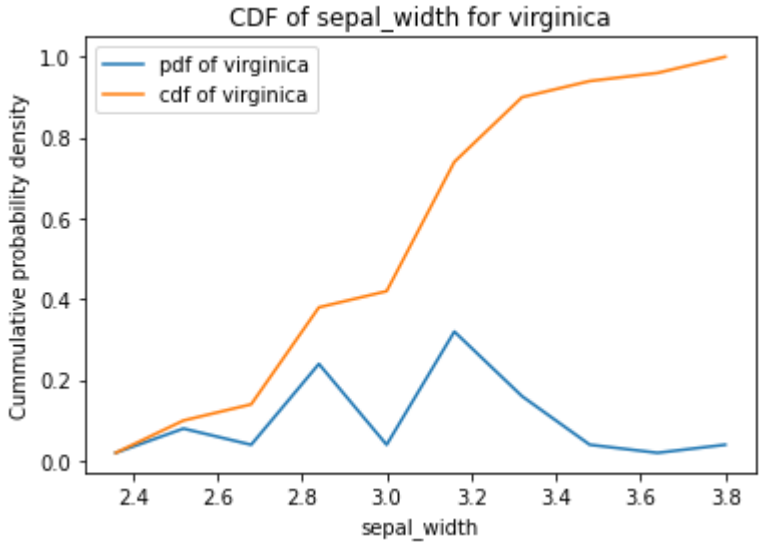
In [12]:

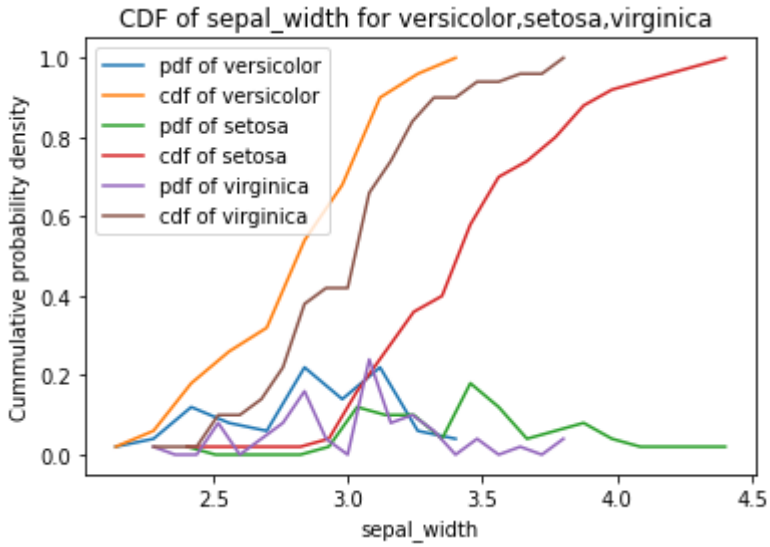
```
# CDF for the Independent variable 'sepal_width'

versicolor = iris_df.loc[iris_df["species"] == "versicolor"]
setosa = iris_df.loc[iris_df["species"] == "setosa"]
virginica = iris_df.loc[iris_df["species"] == "virginica"]


# CDF of sepal_length for versicolor
plt.figure(1)
counts, bin_edges = np.histogram(versicolor['sepal_width'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of versicolor');
plt.plot(bin_edges[1:], cdf, label = 'cdf of versicolor')
plt.xlabel("sepal_width")
plt.ylabel("Cummulative probability density")
plt.title("CDF of sepal_width for versicolor")
plt.legend()


# CDF of sepal_length for setosa
plt.figure(2)
counts, bin_edges = np.histogram(setosa['sepal_width'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of setosa');
plt.plot(bin_edges[1:], cdf, label = 'cdf of setosa')
plt.xlabel("sepal_width")
plt.ylabel("Cummulative probability density")
plt.title("CDF of sepal_width for setosa")
plt.legend()


# CDF of sepal_length for virginica
plt.figure(3)
counts, bin_edges = np.histogram(virginica['sepal_width'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of virginica');
plt.plot(bin_edges[1:], cdf, label = 'cdf of virginica')
plt.xlabel("sepal_width")
plt.ylabel("Cummulative probability density")
plt.title("CDF of sepal_width for virginica")
plt.legend()



# CDFs of versicolor, setosa,virginica for the feature sepal_length in a single plot
plt.figure(4)

counts, bin_edges = np.histogram(versicolor['sepal_width'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of versicolor');
plt.plot(bin_edges[1:], cdf, label = 'cdf of versicolor')
```

```
counts, bin_edges = np.histogram(setosa['sepal_width'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of setosa');
plt.plot(bin_edges[1:],cdf, label = 'cdf of setosa')


counts, bin_edges = np.histogram(virginica['sepal_width'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of virginica');
plt.plot(bin_edges[1:],cdf, label = 'cdf of virginica')



plt.xlabel("sepal_width")
plt.ylabel("Cummulative probability density")
plt.title("CDF of sepal_width for versicolor,setosa,virginica ")
plt.legend()
plt.show()
```









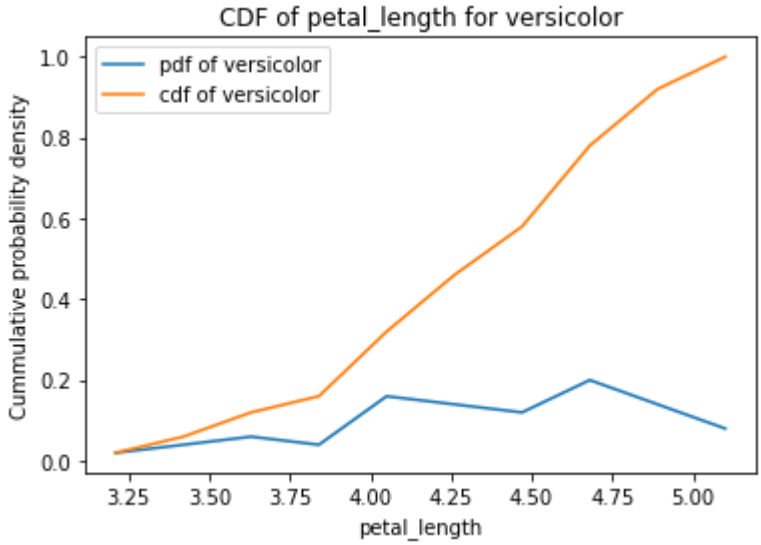
Observation:

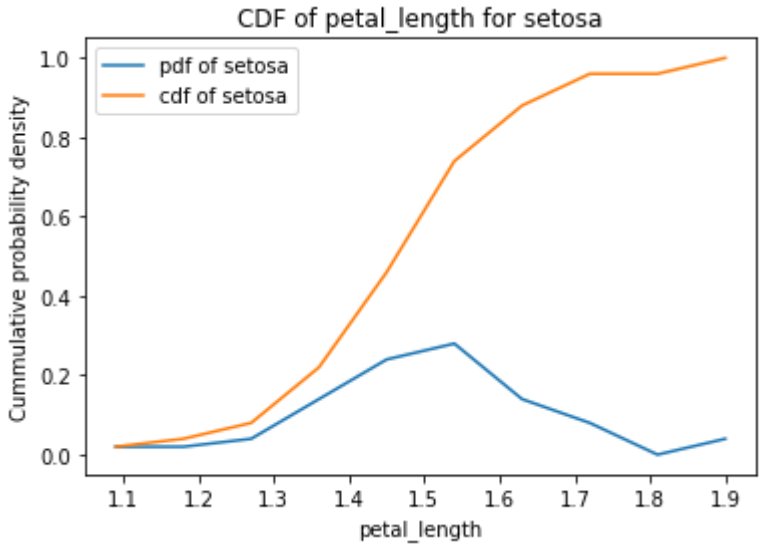
1) From the above plot we can observe that PDFs of sepal_width for versicolor,setosa and virginica are almost all overlaped. So we cannot interpret much from the sepal_width feature

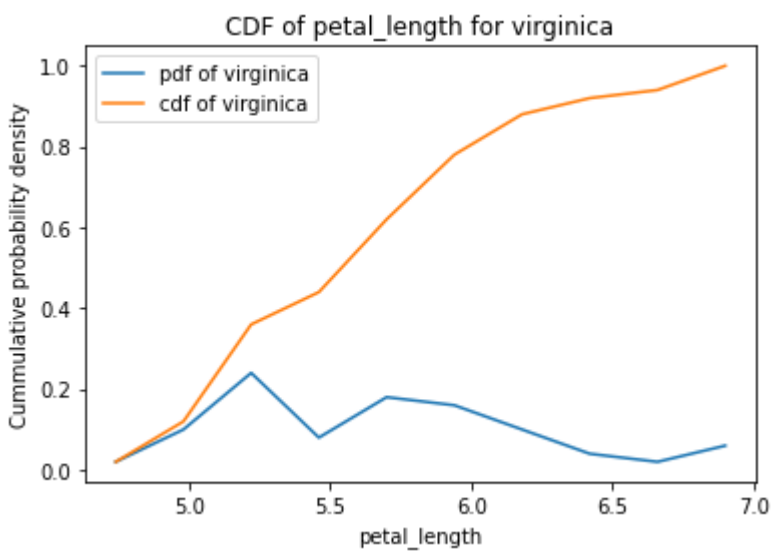
In [13]:

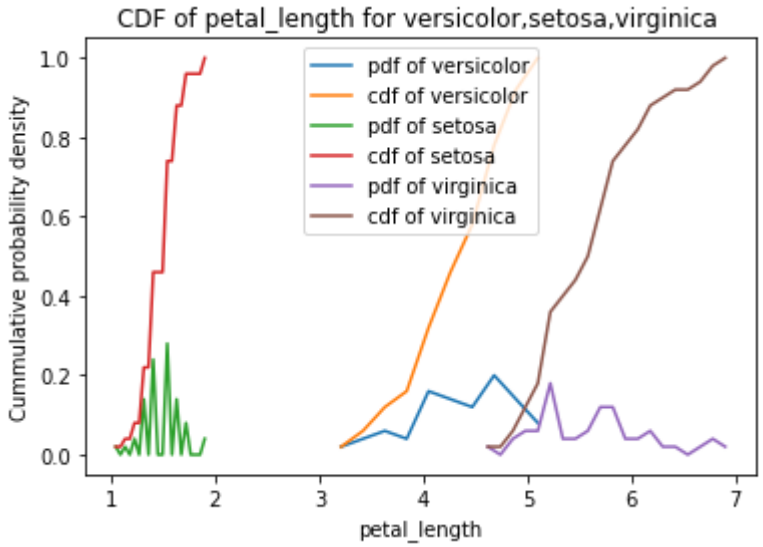
```
# CDF for the Independent variable 'petal_length'

versicolor = iris_df.loc[iris_df["species"] == "versicolor"]
setosa = iris_df.loc[iris_df["species"] == "setosa"]
virginica = iris_df.loc[iris_df["species"] == "virginica"]


# CDF of petal_length for versicolor
plt.figure(1)
counts, bin_edges = np.histogram(versicolor['petal_length'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of versicolor');
plt.plot(bin_edges[1:], cdf, label = 'cdf of versicolor')
plt.xlabel("petal_length")
plt.ylabel("Cummulative probability density")
plt.title("CDF of petal_length for versicolor")
plt.legend()


# CDF of petal_length for setosa
plt.figure(2)
counts, bin_edges = np.histogram(setosa['petal_length'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of setosa');
plt.plot(bin_edges[1:], cdf, label = 'cdf of setosa')
plt.xlabel("petal_length")
plt.ylabel("Cummulative probability density")
plt.title("CDF of petal_length for setosa")
plt.legend()


# CDF of petal_length for virginica
plt.figure(3)
counts, bin_edges = np.histogram(virginica['petal_length'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of virginica');
plt.plot(bin_edges[1:], cdf, label = 'cdf of virginica')
plt.xlabel("petal_length")
plt.ylabel("Cummulative probability density")
plt.title("CDF of petal_length for virginica")
plt.legend()



# CDFs of versicolor, setosa,virginica for the feature petal_length in a single plot
plt.figure(4)

counts, bin_edges = np.histogram(versicolor['petal_length'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of versicolor');
plt.plot(bin_edges[1:], cdf, label = 'cdf of versicolor')
```

```
counts, bin_edges = np.histogram(setosa['petal_length'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of setosa');
plt.plot(bin_edges[1:],cdf, label = 'cdf of setosa')


counts, bin_edges = np.histogram(virginica['petal_length'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of virginica');
plt.plot(bin_edges[1:],cdf, label = 'cdf of virginica')



plt.xlabel("petal_length")
plt.ylabel("Cummulative probability density")
plt.title("CDF of petal_length for versicolor,setosa,virginica ")
plt.legend()
plt.show()
```









Observation:

1) From CDF we can observe that 100% of the setosa flowers have petal_length less than 2. We can say if the petal_length is <= 2 units, then the probability of the flower being setosa is very high

2) From CDF we can observe that 100% of the versicolor flowers have petal_length between 3 and 5 units. We can say if the petal_length is in the range 3 and 5 units, then the probability of the flower being versicolor is very high

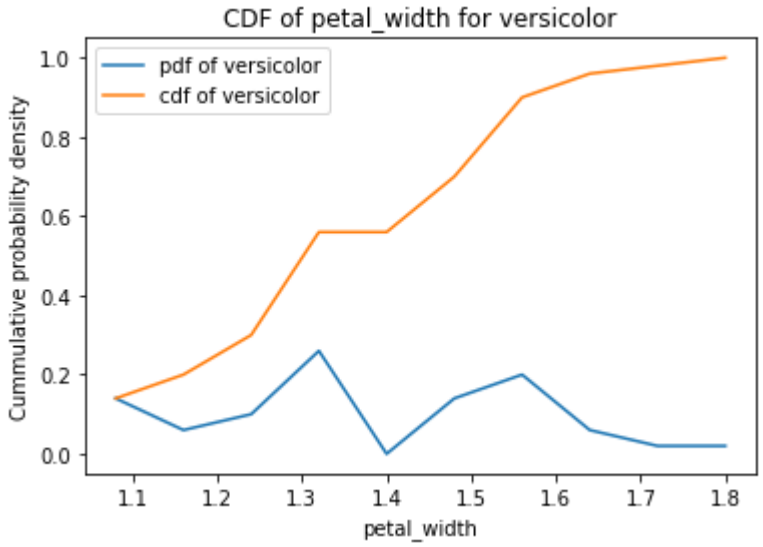
3) we can observe slight overlap of PDFs around 5 units, if we put threshold at 5 units, then the probability of flower being versicolor is more if the petal_lenghth is < 5 Units (but > 3 units)

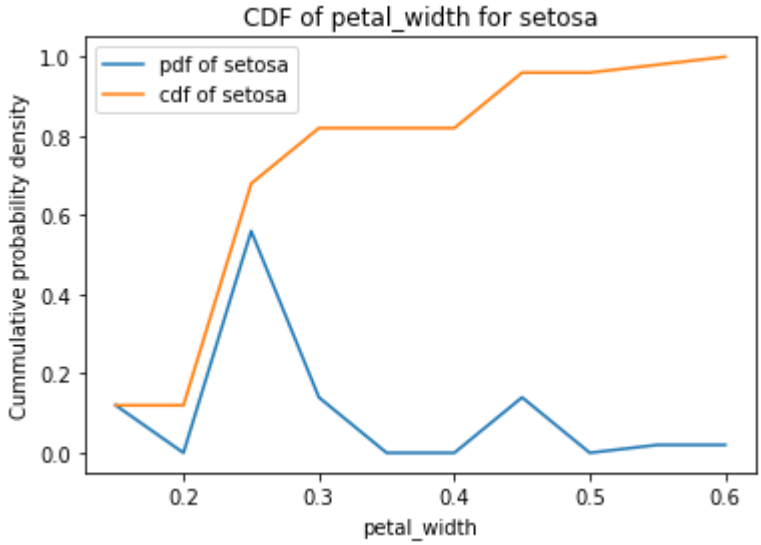
4) We can observe that petal_lenghth for virginica is more when compared to versicolor, setosa

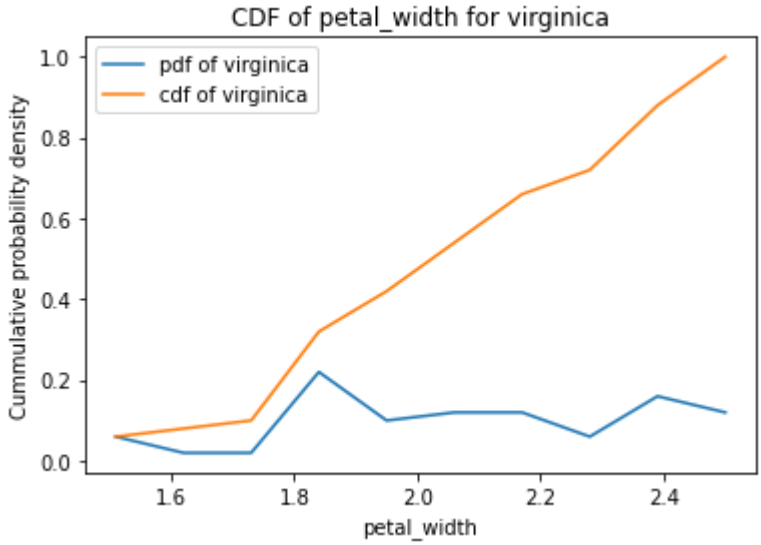
In [14]:

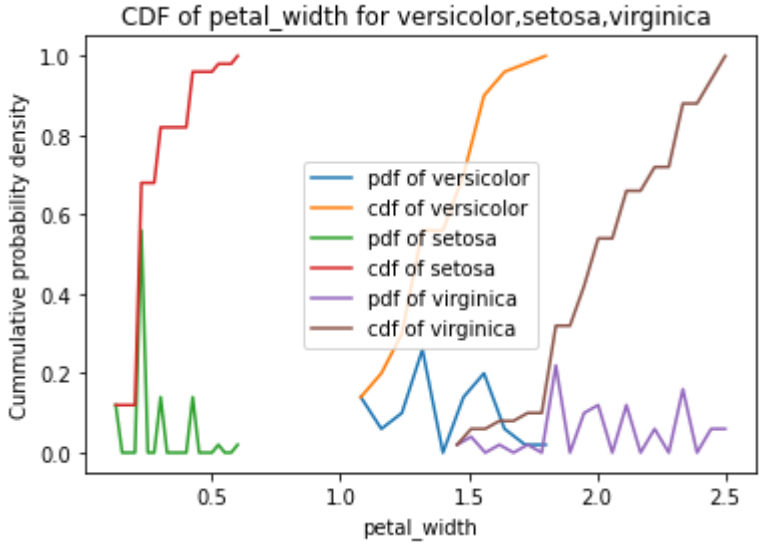
```
# CDF for the Independent variable 'petal_width'

versicolor = iris_df.loc[iris_df["species"] == "versicolor"]
setosa = iris_df.loc[iris_df["species"] == "setosa"]
virginica = iris_df.loc[iris_df["species"] == "virginica"]


# CDF of petal_width for versicolor
plt.figure(1)
counts, bin_edges = np.histogram(versicolor['petal_width'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of versicolor');
plt.plot(bin_edges[1:], cdf, label = 'cdf of versicolor')
plt.xlabel("petal_width")
plt.ylabel("Cummulative probability density")
plt.title("CDF of petal_width for versicolor")
plt.legend()


# CDF of petal_width for setosa
plt.figure(2)
counts, bin_edges = np.histogram(setosa['petal_width'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of setosa');
plt.plot(bin_edges[1:], cdf, label = 'cdf of setosa')
plt.xlabel("petal_width")
plt.ylabel("Cummulative probability density")
plt.title("CDF of petal_width for setosa")
plt.legend()


# CDF of petal_length for virginica
plt.figure(3)
counts, bin_edges = np.histogram(virginica['petal_width'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of virginica');
plt.plot(bin_edges[1:], cdf, label = 'cdf of virginica')
plt.xlabel("petal_width")
plt.ylabel("Cummulative probability density")
plt.title("CDF of petal_width for virginica")
plt.legend()



# CDFs of versicolor, setosa,virginica for the feature petal_width in a single plot
plt.figure(4)

counts, bin_edges = np.histogram(versicolor['petal_width'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of versicolor');
plt.plot(bin_edges[1:], cdf, label = 'cdf of versicolor')
```

```
counts, bin_edges = np.histogram(setosa['petal_width'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of setosa');
plt.plot(bin_edges[1:],cdf, label = 'cdf of setosa')


counts, bin_edges = np.histogram(virginica['petal_width'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf, label = 'pdf of virginica');
plt.plot(bin_edges[1:],cdf, label = 'cdf of virginica')



plt.xlabel("petal_width")
plt.ylabel("Cummulative probability density")
plt.title("CDF of petal_width for versicolor,setosa,virginica ")
plt.legend()
plt.show()
```




CDF of petal_width for setosa
pdf of setosa
cdf of setosa
Cummulative probability density
petal_width


CDF of petal_width for virginica
pdf of virginica
cdf of virginica
Cummulative probability density
petal_width


CDF of petal_width for versicolor,setosa,virginica
pdf of versicolor
cdf of versicolor
pdf of setosa
cdf of setosa
pdf of virginica
cdf of virginica
Cummulative probability density
petal_width

Observation:

1) From CDF we can observe that 100% of the setosa flowers have petal_width less than 0.6 units. We can say if the petal_width is <= 0.6 units, then the probability of the flower being setosa is very high.

2) petal_width for versicolor ranges from 10 to 16 units. In this range the probability of flower being versicolor is more

3) We can say virginica has more petal_width when compared to setosa,versicolor

Conclusion from above CDFs:

1) For setosa flower , petal_length will be lessthan or equal to 2 units and petal_width will be lessthan 0.6 units

2) virginica has more petal_length,petal_width when compared to setosa,versicolor

2. Statistical analysis using Mean, Median, STD, Quantiles and Percentiles

2.1 Mean and Std-dev

In [15]:

```
#Mean, Variance, Std-deviation,

versicolor = iris_df.loc[iris_df["species"] == "versicolor"]
setosa = iris_df.loc[iris_df["species"] == "setosa"]
virginica = iris_df.loc[iris_df["species"] == "virginica"]

print('Means:')
print("Mean of sepal_length for versicolor  = ", np.mean(versicolor["sepal_length"]))
print("Mean of sepal_length for setosa = " , np.mean(setosa["sepal_length"]))
print("Mean of sepal_length for virginica  = ", np.mean(virginica["sepal_length"]))
print("**********************************************************************")
print("Mean of sepal_width for versicolor  = ", np.mean(versicolor["sepal_width"]))
print("Mean of sepal_width for setosa = " , np.mean(setosa["sepal_width"]))
print("Mean of sepal_width for virginica  = ", np.mean(virginica["sepal_width"]))
print("**********************************************************************")
print("Mean of petal_length for versicolor  = ", np.mean(versicolor["petal_length"]))
print("Mean of petal_length for setosa = " , np.mean(setosa["petal_length"]))
print("Mean of petal_length for virginica  = ", np.mean(virginica["petal_length"]))
print("**********************************************************************")
print("Mean of petal_width for versicolor  = ", np.mean(versicolor["petal_width"]))
print("Mean of petal_width for setosa = " , np.mean(setosa["petal_width"]))
print("Mean of petal_width for virginica  = ", np.mean(virginica["petal_width"]))


print("\nStd-dev:");
print("Std-dev of sepal_length for versicolor = ", np.std(versicolor["sepal_length"]))
print("Std-dev of sepal_length for setosa = ", np.std(setosa["sepal_length"]))
print("Std-dev of sepal_length for virginica =", np.std(virginica["sepal_length"]))
print("**********************************************************************")
print("Std-dev of sepal_width for versicolor = ", np.std(versicolor["sepal_width"]))
print("Std-dev of sepal_width for setosa = ", np.std(setosa["sepal_width"]))
print("Std-dev of sepal_width for virginica =", np.std(virginica["sepal_width"]))
print("**********************************************************************")
print("Std-dev of petal_length for versicolor = ", np.std(versicolor["petal_length"]))
print("Std-dev of petal_length for setosa = ", np.std(setosa["petal_length"]))
print("Std-dev of petal_length for virginica =", np.std(virginica["petal_length"]))
print("**********************************************************************")
print("Std-dev of petal_width for versicolor = ", np.std(versicolor["petal_width"]))
print("Std-dev of petal_width for setosa = ", np.std(setosa["petal_width"]))
print("Std-dev of petal_width for virginica =", np.std(virginica["petal_width"]))
print("**********************************************************************")
```

```
Means:
Mean of sepal_length for versicolor  =  5.936
Mean of sepal_length for setosa =  5.006
Mean of sepal_length for virginica  =  6.587999999999998
**********************************************************************
**********
Mean of sepal_width for versicolor  =  2.7700000000000005
Mean of sepal_width for setosa =  3.418
Mean of sepal_width for virginica  =  2.974
**********************************************************************
**********
Mean of petal_length for versicolor  =  4.26
Mean of petal_length for setosa =  1.464
Mean of petal_length for virginica  =  5.552000000000000
**********************************************************************
**********
```

```
Mean of petal_width for versicolor  =  1.3259999999999998
Mean of petal_width for setosa =  0.244
Mean of petal_width for virginica  =  2.0260000000000002

Std-dev:
Std-dev of sepal_length for versicolor =  0.5109833656783751
Std-dev of sepal_length for setosa =  0.3489469873777391
Std-dev of sepal_length for virginica = 0.6294886813914926
**********************************************************************
********************
Std-dev of sepal_width for versicolor =  0.31064449134018135
Std-dev of sepal_width for setosa =  0.37719490982779713
Std-dev of sepal_width for virginica = 0.3192553836664309
**********************************************************************
********************
Std-dev of petal_length for versicolor =  0.4651881339845203
Std-dev of petal_length for setosa =  0.17176728442867112
Std-dev of petal_length for virginica = 0.546347874526844
**********************************************************************
********************
Std-dev of petal_width for versicolor =  0.19576516544063705
Std-dev of petal_width for setosa =  0.10613199329137281
Std-dev of petal_width for virginica = 0.2718896835115301
**********************************************************************
********************
```

Observations from mean and STD:

We know that mean is the average value , it is a central tendency value . Variance is the spread around mean

1) From above statistical data we can observe that setosa flower has small petal_length (central tendency at 1.4 units with variance of 0.17 units) and petal width (central tendency at 0.24 units with variance of 0.10 units)

2) We can observe that sepal_length , sepal_width , petal_length , petal_width are more for virginica

In [ ]:

2.2 Median, Percentile, Quantile, IQR, MAD

In [16]:

```
#Median, Quantiles, Percentiles, IQR.

versicolor = iris_df.loc[iris_df["species"] == "versicolor"]
setosa = iris_df.loc[iris_df["species"] == "setosa"]
virginica = iris_df.loc[iris_df["species"] == "virginica"]

print("\nMedians:")
# Median is also a central tendency value like mean, but mean and std can be easily corrupt
#so better to use Median and MAD
print("median of sepal_length for versicolor = ", np.median(versicolor["sepal_length"]))
print("median of sepal_length for setosa = " , np.median(setosa["sepal_length"]))
print("median of sepal_length for virginica = ", np.median(virginica["sepal_length"]))
print("*************************************************************************")
print("median of sepal_width for versicolor = ", np.median(versicolor["sepal_width"]))
print("median of sepal_width for setosa = " , np.median(setosa["sepal_width"]))
print("median of sepal_width for virginica = ", np.median(virginica["sepal_width"]))
print("*************************************************************************")
print("median of petal_length for versicolor = ", np.median(versicolor["petal_length"]))
print("median of petal_length for setosa = " , np.median(setosa["petal_length"]))
print("median of petal_length for virginica = ", np.median(virginica["petal_length"]))
print("*************************************************************************")
print("median of petal_width for versicolor = ", np.median(versicolor["petal_width"]))
print("median of petal_width for setosa = " , np.median(setosa["petal_width"]))
print("median of petal_width for virginica = ", np.median(virginica["petal_width"]))
print("*************************************************************************")


print("\nQuantiles:")

print("Quantiles of sepal_length for versicolor = ", np.percentile(versicolor["sepal_length
print("Quantiles of sepal_length for setosa = " , np.percentile(setosa["sepal_length"],np.a
print("Quantiles of sepal_length for virginica = ", np.percentile(virginica["sepal_length"]
print("*************************************************************************")
print("Quantiles of sepal_width for versicolor = ", np.percentile(versicolor["sepal_width"]
print("Quantiles of sepal_width for setosa = " , np.percentile(setosa["sepal_width"],np.ara
print("Quantiles of sepal_width for virginica = ", np.percentile(virginica["sepal_width"],n
print("*************************************************************************")
print("Quantiles of petal_length for versicolor = ", np.percentile(versicolor["petal_length
print("Quantiles of petal_length for setosa = " , np.percentile(setosa["petal_length"],np.a
print("Quantiles of petal_length for virginica = ", np.percentile(virginica["petal_length"]
print("*************************************************************************")
print("Quantiles of petal_width for versicolor = ", np.percentile(versicolor["petal_width"]
print("Quantiles of petal_width for setosa = " , np.percentile(setosa["petal_width"],np.ara
print("Quantiles of petal_width for virginica = ", np.percentile(virginica["petal_width"],n
print("*************************************************************************")

print("\n90th Percentiles:")

print("90th Percentiles of sepal_length for versicolor = ", np.percentile(versicolor["sepal
print("90th Percentiles of sepal_length for setosa = " , np.percentile(setosa["sepal_length
print("90th Percentiles of sepal_length for virginica = ", np.percentile(virginica["sepal_l
print("*************************************************************************")
print("90th Percentiles of sepal_width for versicolor = ", np.percentile(versicolor["sepal_
print("90th Percentiles of sepal_width for setosa = " , np.percentile(setosa["sepal_width"]
print("90th Percentiles of sepal_width for virginica = ", np.percentile(virginica["sepal_wi
print("*************************************************************************")
print("90th Percentiles of petal_length for versicolor = ", np.percentile(versicolor["petal
print("90th Percentiles of petal_length for setosa = " , np.percentile(setosa["petal_length
print("90th Percentiles of petal_length for virginica = ", np.percentile(virginica["petal_l
```

```
print("*************************************************************************
print("90th Percentiles of petal_width for versicolor = ", np.percentile(versicolor["petal_
print("90th Percentiles of petal_width for setosa = " , np.percentile(setosa["petal_width"]
print("90th Percentiles of petal_width for virginica = ", np.percentile(virginica["petal_wi
print("*************************************************************************



print ("\nMedian Absolute Deviation:")
import numpy as np
from statsmodels import robust

print("Median Absolute Deviation of sepal_length for versicolor = ", robust.mad(versicolor[
print("Median Absolute Deviation of sepal_length for setosa = " , robust.mad(setosa["sepal_
print("Median Absolute Deviation of sepal_length for virginica = ", robust.mad(virginica["s
print("*************************************************************************
print("Median Absolute Deviation of sepal_width for versicolor = ", robust.mad(versicolor["
print("Median Absolute Deviation of sepal_width for setosa = " , robust.mad(setosa["sepal_w
print("Median Absolute Deviation of sepal_width for virginica = ", robust.mad(virginica["se
print("*************************************************************************
print("Median Absolute Deviation of petal_length for versicolor = ", robust.mad(versicolor[
print("Median Absolute Deviation of petal_length for setosa = " , robust.mad(setosa["petal_
print("Median Absolute Deviation of petal_length for virginica = ", robust.mad(virginica["p
print("*************************************************************************
print("Median Absolute Deviation of petal_width for versicolor = ", robust.mad(versicolor["
print("Median Absolute Deviation of petal_width for setosa = " , robust.mad(setosa["petal_w
print("Median Absolute Deviation of petal_width for virginica = ", robust.mad(virginica["pe
print("*************************************************************************
```

```
Medians:
median of sepal_length for versicolor =  5.9
median of sepal_length for setosa =  5.0
median of sepal_length for virginica =  6.5
**************************************************************************
******************
median of sepal_width for versicolor =  2.8
median of sepal_width for setosa =  3.4
median of sepal_width for virginica =  3.0
**************************************************************************
******************
median of petal_length for versicolor =  4.35
median of petal_length for setosa =  1.5
median of petal_length for virginica =  5.55
**************************************************************************
******************
median of petal_width for versicolor =  1.3
median of petal_width for setosa =  0.2
median of petal_width for virginica =  2.0
**************************************************************************
******************

Quantiles:
Quantiles of sepal_length for versicolor =  [4.9 5.6 5.9 6.3]
Quantiles of sepal_length for setosa =  [4.3 4.8 5.  5.2]
Quantiles of sepal_length for virginica =  [4.9   6.225 6.5   6.9  ]
**************************************************************************
******************
Quantiles of sepal_width for versicolor =  [2.    2.525 2.8   3.   ]
```

```
Quantiles of sepal_width for setosa =  [2.3   3.125 3.4   3.675]
Quantiles of sepal_width for virginica =  [2.2   2.8   3.    3.175]
************************************************************************
******************
Quantiles of petal_length for versicolor =  [3.   4.   4.35 4.6 ]
Quantiles of petal_length for setosa =  [1.    1.4   1.5   1.575]
Quantiles of petal_length for virginica =  [4.5   5.1   5.55  5.875]
************************************************************************
******************
Quantiles of petal_width for versicolor =  [1.  1.2 1.3 1.5]
Quantiles of petal_width for setosa =  [0.1 0.2 0.2 0.3]
Quantiles of petal_width for virginica =  [1.4 1.8 2.  2.3]
************************************************************************
******************

90th Percentiles:
90th Percentiles of sepal_length for versicolor =  6.7
90th Percentiles of sepal_length for setosa =  5.41
90th Percentiles of sepal_length for virginica =  7.61
************************************************************************
******************
90th Percentiles of sepal_width for versicolor =  3.1100000000000003
90th Percentiles of sepal_width for setosa =  3.9
90th Percentiles of sepal_width for virginica =  3.31
************************************************************************
******************
90th Percentiles of petal_length for versicolor =  4.8
90th Percentiles of petal_length for setosa =  1.7
90th Percentiles of petal_length for virginica =  6.3100000000000005
************************************************************************
******************
90th Percentiles of petal_width for versicolor =  1.5100000000000002
90th Percentiles of petal_width for setosa =  0.4
90th Percentiles of petal_width for virginica =  2.4
************************************************************************
******************

Median Absolute Deviation:
Median Absolute Deviation of sepal_length for versicolor =  0.51891077647696
08
Median Absolute Deviation of sepal_length for setosa =  0.29652044370112063
Median Absolute Deviation of sepal_length for virginica =  0.593040887402241
3
************************************************************************
******************
Median Absolute Deviation of sepal_width for versicolor =  0.296520443701120
63
Median Absolute Deviation of sepal_width for setosa =  0.44478066555168033
Median Absolute Deviation of sepal_width for virginica =  0.2965204437011206
3
************************************************************************
******************
Median Absolute Deviation of petal_length for versicolor =  0.51891077647696
02
Median Absolute Deviation of petal_length for setosa =  0.14826022185056031
Median Absolute Deviation of petal_length for virginica =  0.667170998327521
1
************************************************************************
******************
Median Absolute Deviation of petal_width for versicolor =  0.222390332775840
3
```

```
Median Absolute Deviation of petal_width for setosa =  0.0
Median Absolute Deviation of petal_width for virginica =  0.2965204437011203
**************************************************************************
******************
```

Observations from Median, Quantiles, Percentiles, IQR:

Mean can be easily impacted/corrupted by outliers. So We go for Median

1) From median values we can observe that setosa flower has less petal_length and petal_width

2) 90% of the setosa flowers have petal_length < 1.7 units and petal_length < 0.4

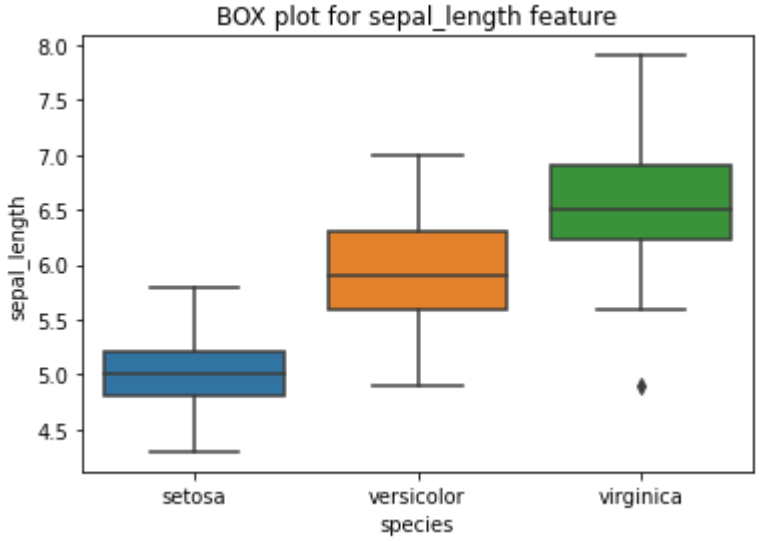
3) We can observe that sepal_length , sepal_width , petal_length , petal_width are more for virginica

3. Univariate analysis with BOX plots and Violin plots

3.1 Box plot and Whiskers

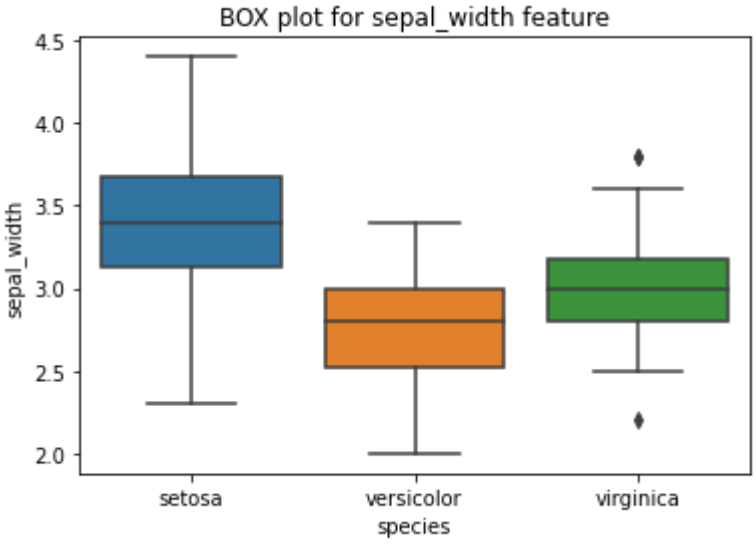
In [17]:

```
# Box plot for the feature 'sepal_length'
# Box plot is drawn for visualizing percentile, quantile.

sns.boxplot(x='species',y='sepal_length', data=iris_df)
plt.title("BOX plot for sepal_length feature")
plt.show()
```



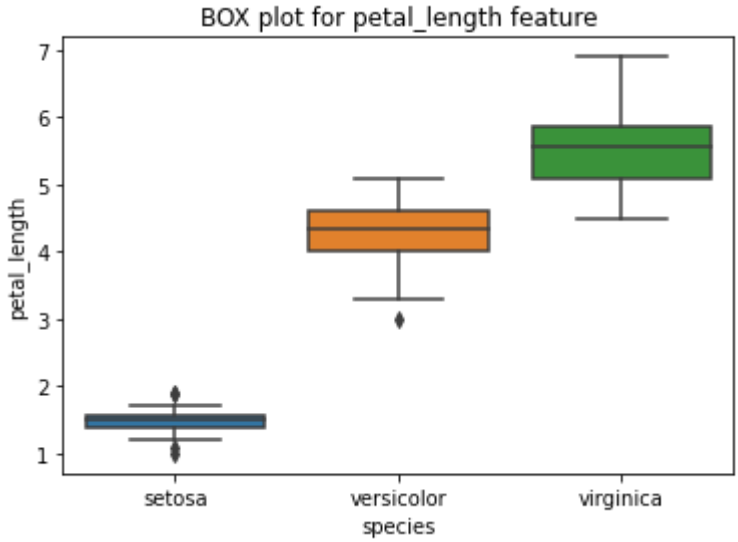
In [18]:

```
# Box plot for the feature 'sepal_length'
# Box plot is drawn for visualizing percentile, quantile.

sns.boxplot(x='species',y='sepal_width', data=iris_df)
plt.title("BOX plot for sepal_width feature")
plt.show()
```



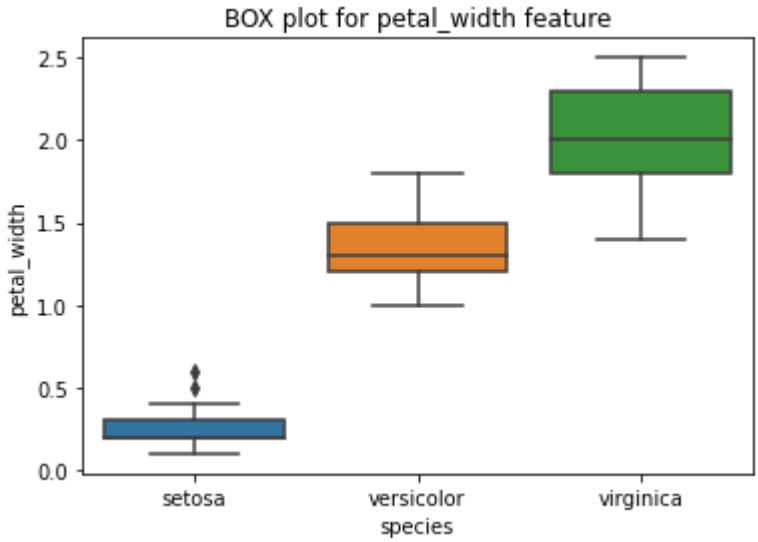
In [19]:

```
# Box plot for the feature 'sepal_length'
# Box plot is drawn for visualizing percentile, quantile.

sns.boxplot(x='species',y='petal_length', data=iris_df)
plt.title("BOX plot for petal_length feature")
plt.show()
```



In [20]:

```
# Box plot for the feature 'sepal_length'
# Box plot is drawn for visualizing percentile, quantile.

sns.boxplot(x='species',y='petal_width', data=iris_df)
plt.title("BOX plot for petal_width feature")
plt.show()
```



Observation from Box plot:

We know that with in the BOX 25th percentile to 75th percentile values lies. Means 50% of the points lies with in the box.

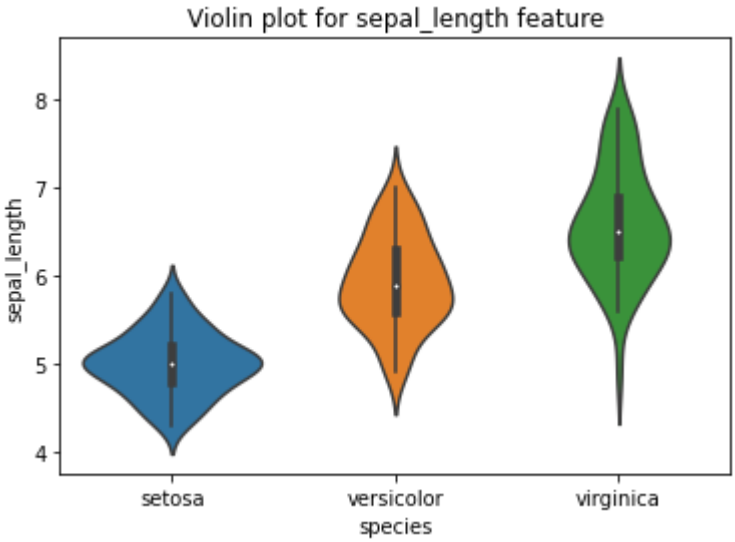
1) From the Box plots we can observe that box plots of features ('sepal_length','sepal_width', 'petal_length','petal_width') for 3 different classes (versicolor, setosa, virginica) are not overlaped completely. It means all four features are important in classification task

2) If we observe Box plots of 'petal_length' and 'petal_width' features we can say using these two features we can sepearte setosa from versicolor and virginica as Box plots of 'petal_length' and 'petal_width' features of setosa doesnot overlap with versicolor and virginica

3.2 Violin plots

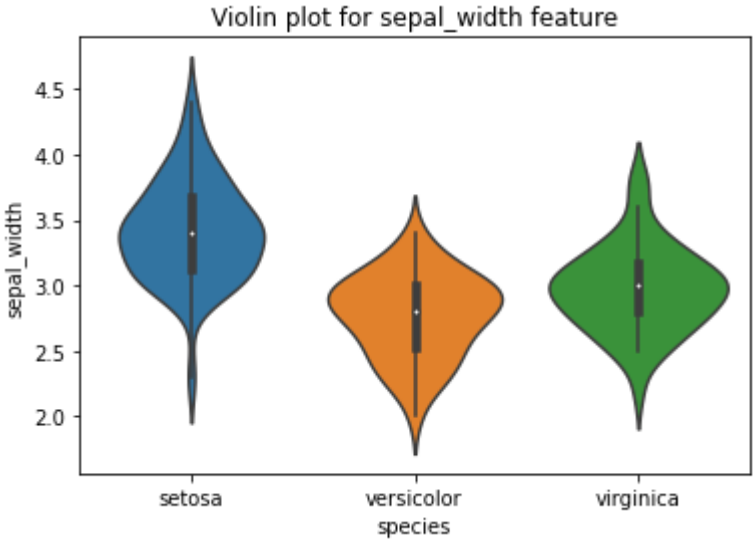
In [21]:

```
# violin plot for the feature 'sepal_length'
# Violin plot combines the advantages of BOX Plot and PDfS

sns.violinplot(x="species", y="sepal_length", data=iris_df, size=8)
plt.title("Violin plot for sepal_length feature")
plt.show()
```



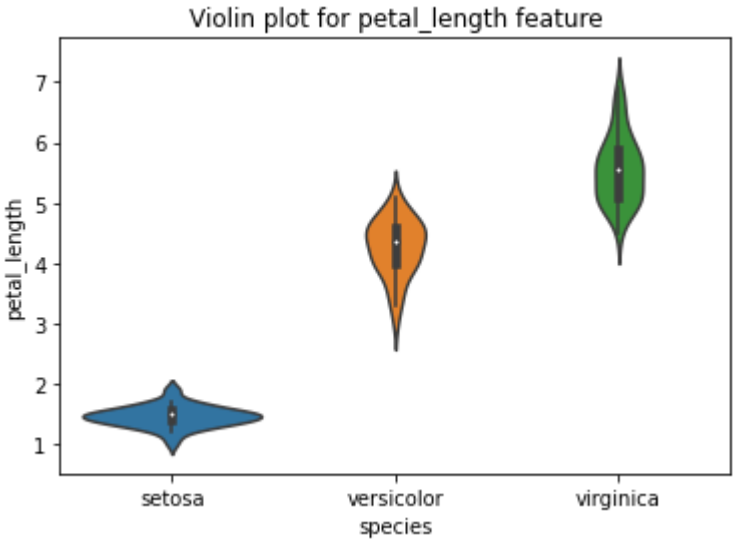
In [22]:

```
# violin plot for the feature 'sepal_length'
# Violin plot combines the advantages of BOX Plot and PDfS

sns.violinplot(x="species", y="sepal_width", data=iris_df, size=8)
plt.title("Violin plot for sepal_width feature")
plt.show()
```



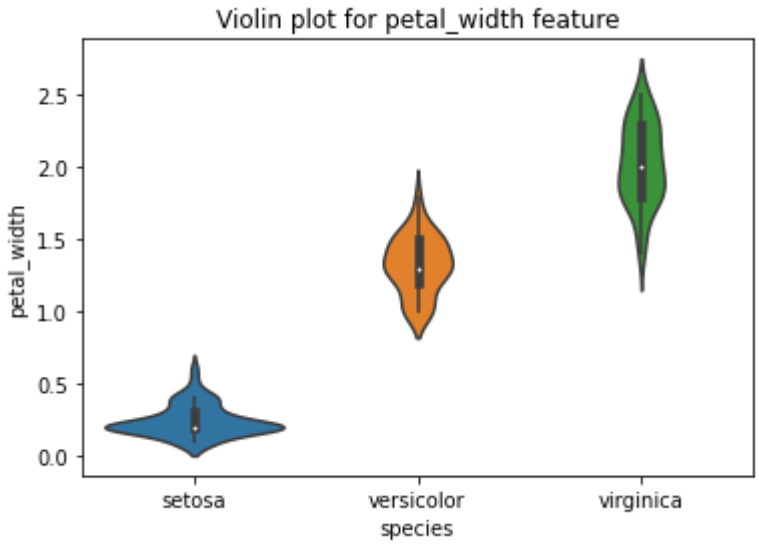
In [23]:

```
# violin plot for the feature 'sepal_length'
# Violin plot combines the advantages of BOX Plot and PDfS

sns.violinplot(x="species", y="petal_length", data=iris_df, size=8)
plt.title("Violin plot for petal_length feature")
plt.show()
```



In [24]:

```
# violin plot for the feature 'sepal_length'
# Violin plot combines the advantages of BOX Plot and PDfS

sns.violinplot(x="species", y="petal_width", data=iris_df, size=8)
plt.title("Violin plot for petal_width feature")
plt.show()
```



Observations from violin plot:

Violin plot combines the advantages of BOX Plot and PDfS

1) From the Violin plots we can observe that box plots of features ('sepal_length','sepal_width', 'petal_length','petal_width') for 3 different classes (versicolor, setosa, virginica) are not overlaped completely. It means all four features are important in classification task

In [ ]:

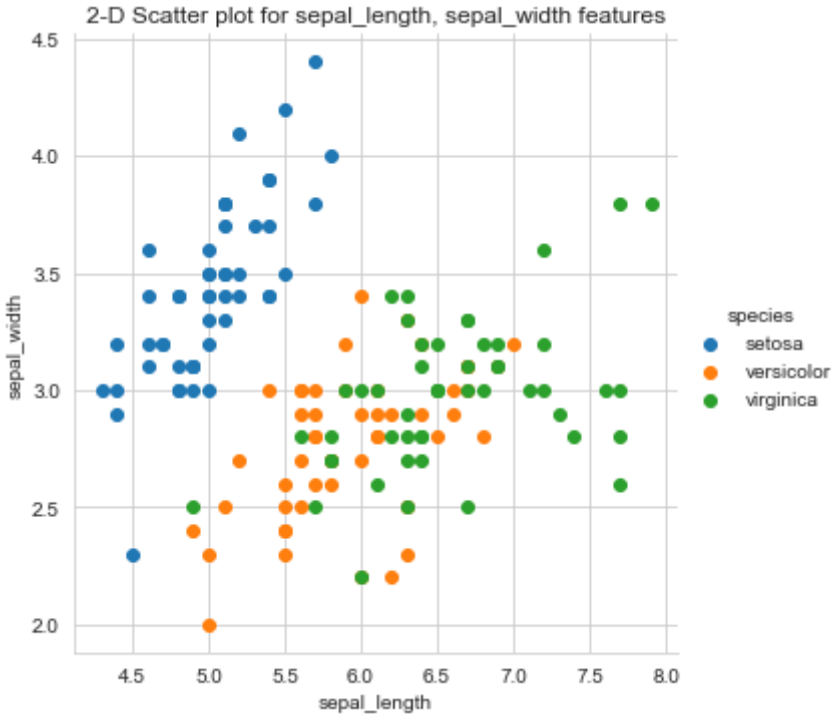
4. Bi-Variate analysis / Multivariate analysis

4.1 2-D Scatter Plots

In [25]:

```
# 2-D Scatter plot

sns.set_style("whitegrid");
sns.FacetGrid(iris_df, hue="species", size=5) \
   .map(plt.scatter, "sepal_length", "sepal_width") \
   .add_legend();
plt.title("2-D Scatter plot for sepal_length, sepal_width features")
plt.show();
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\axisgrid.py:243: UserWarn
ing: The `size` parameter has been renamed to `height`; please update your c
ode.
  warnings.warn(msg, UserWarning)
```



Observation from 2-D Scatter plot:

The 2-D scatter plot drawn above is almost overlapped for versicolor and virginica. How ever we can interpret the following points from the plot.

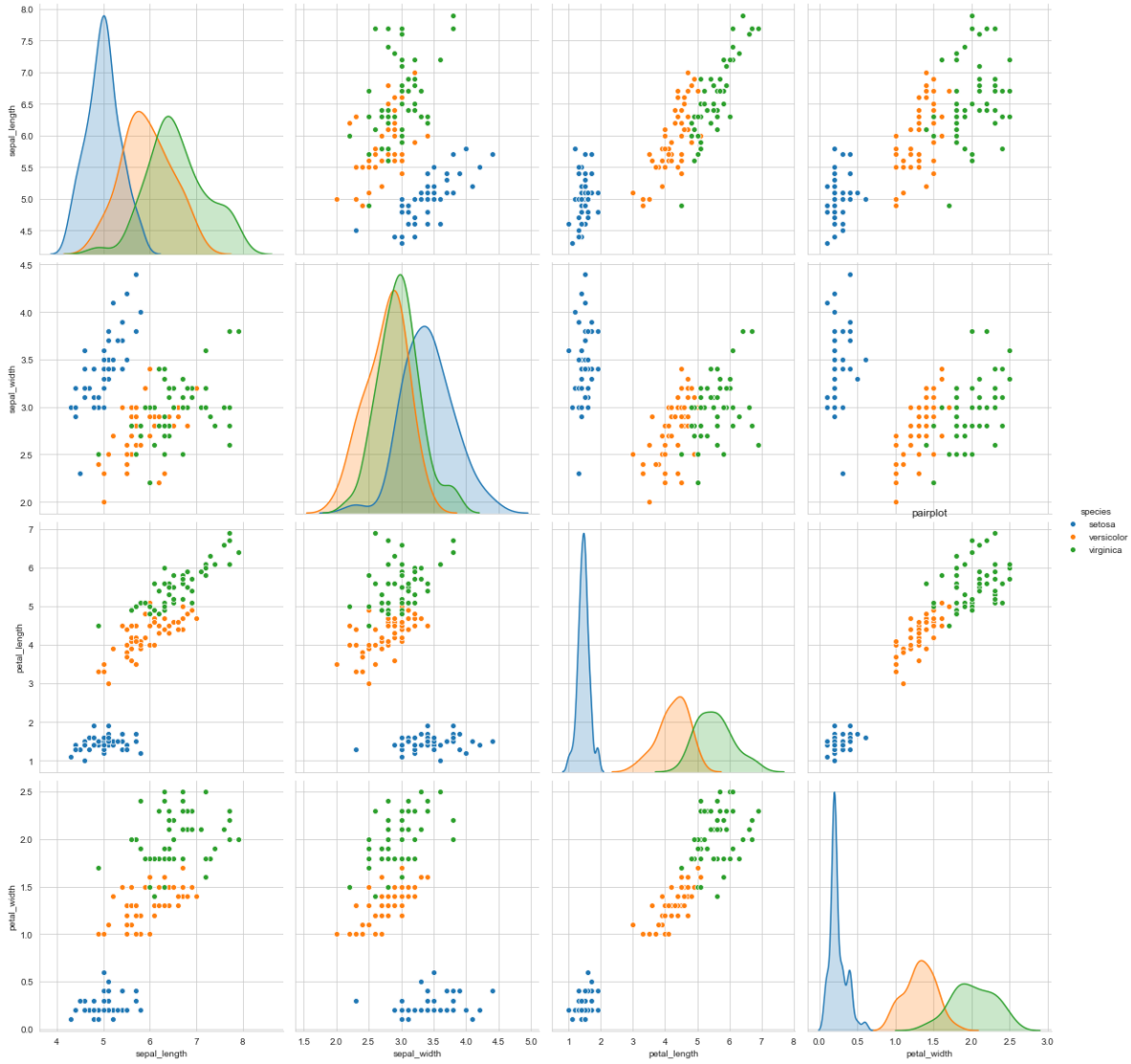
1) We can separate setosa from versicolor and virginica using sepal_length and sepal_width features

4.2 Pair-plot (To observe all 2-D Scatter Plot)

Objective here is to find which two independent variables are better in classifying the flowers

In [26]:

```
sns.set_style("whitegrid");
sns.pairplot(iris_df, hue="species",vars=['sepal_length', 'sepal_width', 'petal_length', 'p
plt.title("pairplot");
plt.show()
```



In [27]:

```
'sepal_length', 'sepal_width', 'petal_length', 'petal_width'
```

Out[27]:

```
('sepal_length', 'sepal_width', 'petal_length', 'petal_width')
```

Reading the pairplot:

Here we will get 4c2 = 6 plots (4 Independent variables selecting 2 at a time). In pair plot we won't consider principle diagonal plots. Plots above the diagonal and below the diagonal are just mirror images of each other

Observations from pair plot:

1) From pair plot , we observe 2-D scatter plots, as described below

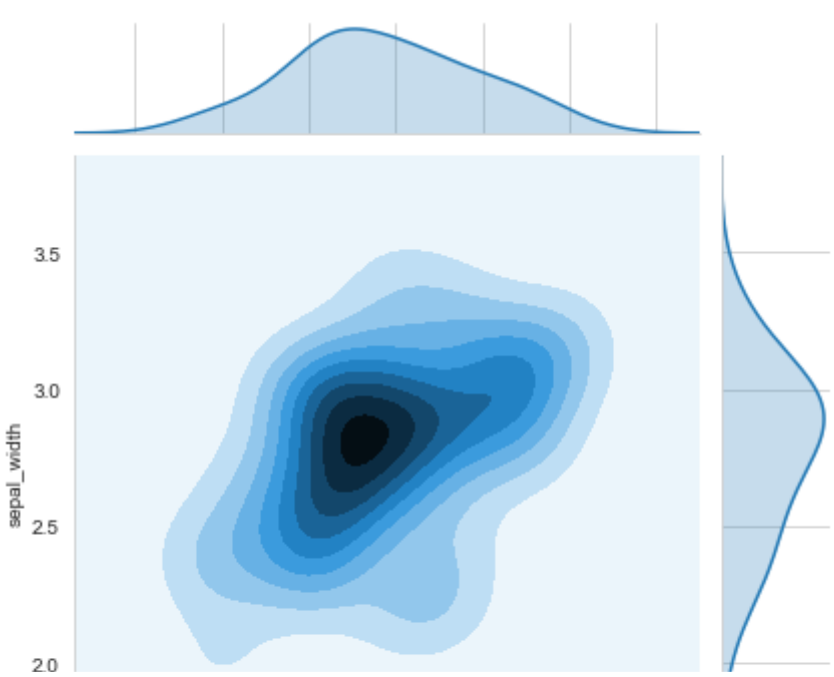
```
 1) From the plots between ('petal_length', 'petal_width') , ('sepal_width', 'peta
l_width'),
  ('sepal_length','petal_width') we can observe that setosa is well seperated from
versicolor and virginica.

 2)  From the plots between ('petal_length', 'petal_width') we can seperate  versi
color and virginica eventhouh
     there slight overlap
```

5. Multivariate probability density plot or Contour Plot

Contour plot is the density plot in 2-D considering two features at a time. The more denser area will be darker in colour and less denser area is lighter in colour. We can imagine Contour plot like hill coming out of the screen

In [28]:

```
#2D Density plot, contors-plot
versicolor = iris_df.loc[iris_df["species"] == "versicolor"]
setosa = iris_df.loc[iris_df["species"] == "setosa"]
virginica = iris_df.loc[iris_df["species"] == "virginica"]

# For versicolor
sns.jointplot(x="sepal_length", y="sepal_width", data=versicolor, kind="kde");
plt.show();

# For setosa
sns.jointplot(x="sepal_length", y="sepal_width", data=setosa, kind="kde");
plt.show();

# For virginica
sns.jointplot(x="sepal_length", y="sepal_width", data=virginica, kind="kde");
plt.show();

# For versicolor
sns.jointplot(x="petal_length", y="petal_width", data=versicolor, kind="kde");
plt.show();

# For setosa
sns.jointplot(x="petal_length", y="petal_width", data=setosa, kind="kde");
plt.show();

# For virginica
sns.jointplot(x="petal_length", y="petal_width", data=virginica, kind="kde");
plt.show();
```



Observations from contour plots:

1) From the contour plot drawn between petal_width and petal_length of setos flower we can observe that most of the setosa flowers will have petal_length between 1.4 to 1.6 units and petal width in the range 0.1 to 0.2 units.

Conclusion from above analysis

1) All four features 'sepal_length', 'sepal_width', 'petal_length', 'petal_width' are useful in classifying the species

2) Using 'petal_length', 'petal_width' features we can perfectly sepearte setosa from versicolor,virginica

3) Using 'petal_length', 'petal_width' features we can seperate versicolor and virginica to some extent

4) Most of the virginica flowers have high values for all the four features 'sepal_length', 'sepal_width', 'petal_length', 'petal_width'

References:

https://www.appliedaicourse.com (https://www.appliedaicourse.com)

https://en.wikipedia.org/wiki/Iris_flower_data_set (https://en.wikipedia.org/wiki/Iris_flower_data_set)

In [ ]: