Data Visualization Using Matplotlib and Seaborn using Iris Dataset

Introduction to Matplotlib :

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is widely used for plotting data and offers great flexibility and control over the appearance of plots. Matplotlib can produce a variety of plots such as line plots, scatter plots, histograms, bar charts, and more.

Basic Syntax:

import matplotlib.pyplot as plt

Introduction to Seaborn:

Seaborn is a Python data visualization library built on top of Matplotlib. It is designed to provide a high-level interface for drawing attractive and informative statistical graphics. Seaborn works seamlessly with pandas data structures and simplifies the process of creating complex visualizations with just a few lines of code.

Basic Syntax:

import seaborn as sns

The Iris dataset consists of 150 observations from 3 species of iris flowers (setosa, versicolor, and virginica). Each observation contains the following features: sepal length (cm), sepal width (cm), petal length (cm), petal width (cm).

load the dataset using the following code:

```
import seaborn as sns
import matplotlib.pyplot as plt
iris=sns.load_dataset('iris')
print(iris)
```

```
     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
```

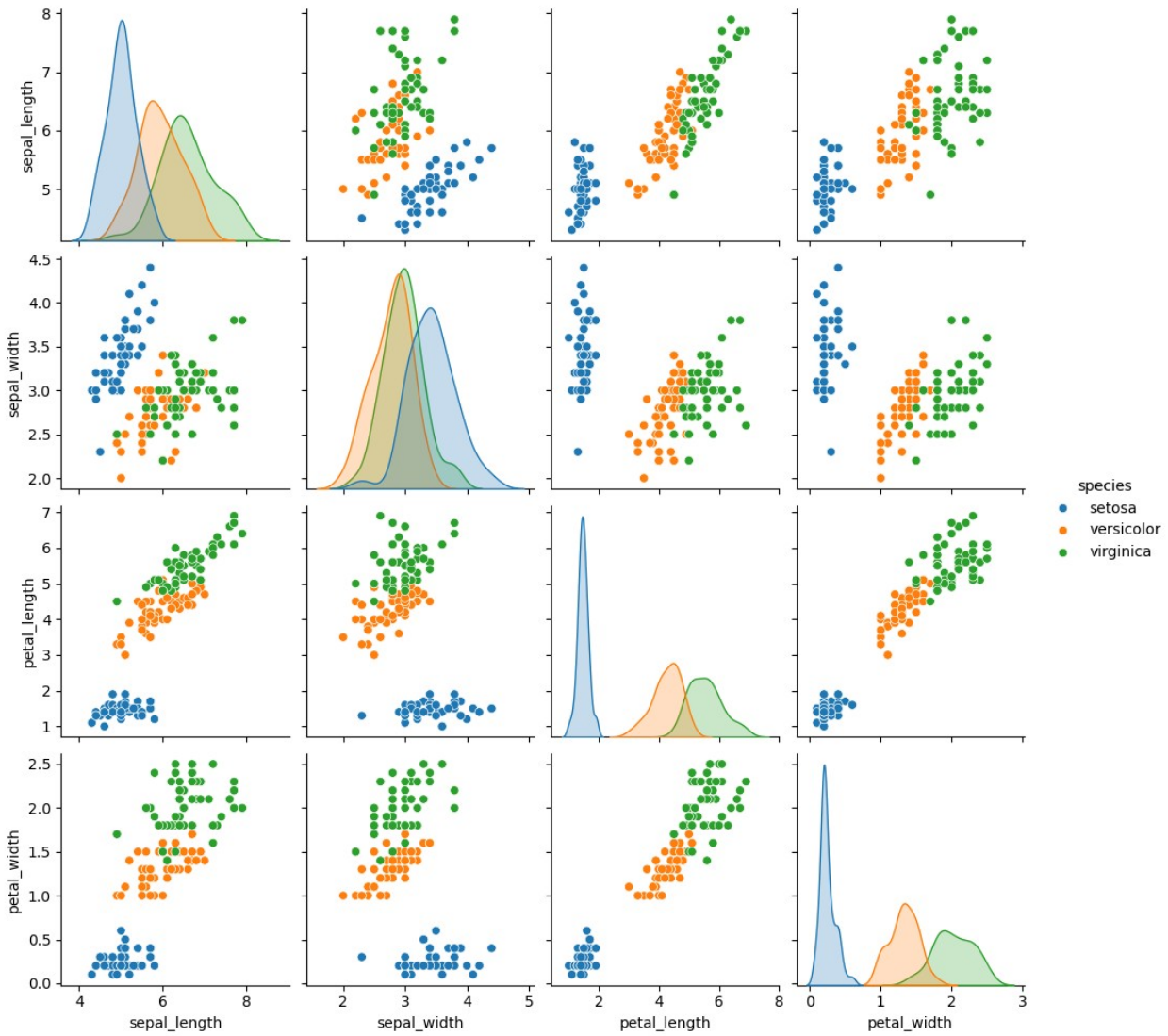
```
..            ...          ...           ...          ...          ...
145           6.7          3.0           5.2          2.3   virginica
146           6.3          2.5           5.0          1.9   virginica
147           6.5          3.0           5.2          2.0   virginica
148           6.2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

[150 rows x 5 columns]
```

General Statistics Plot (Matplotlib or Seaborn):

Write a Python program to create a plot that gives a general statistical summary of the Iris data. You can use seaborn's pairplot or pandas' describe() for guidance.

```
#General Statistics Plot (Matplotlib or Seaborn):
sns.pairplot(iris, hue='species', height=2.5) # pair plot using seaborn
plt.show() #display the pairplot
```

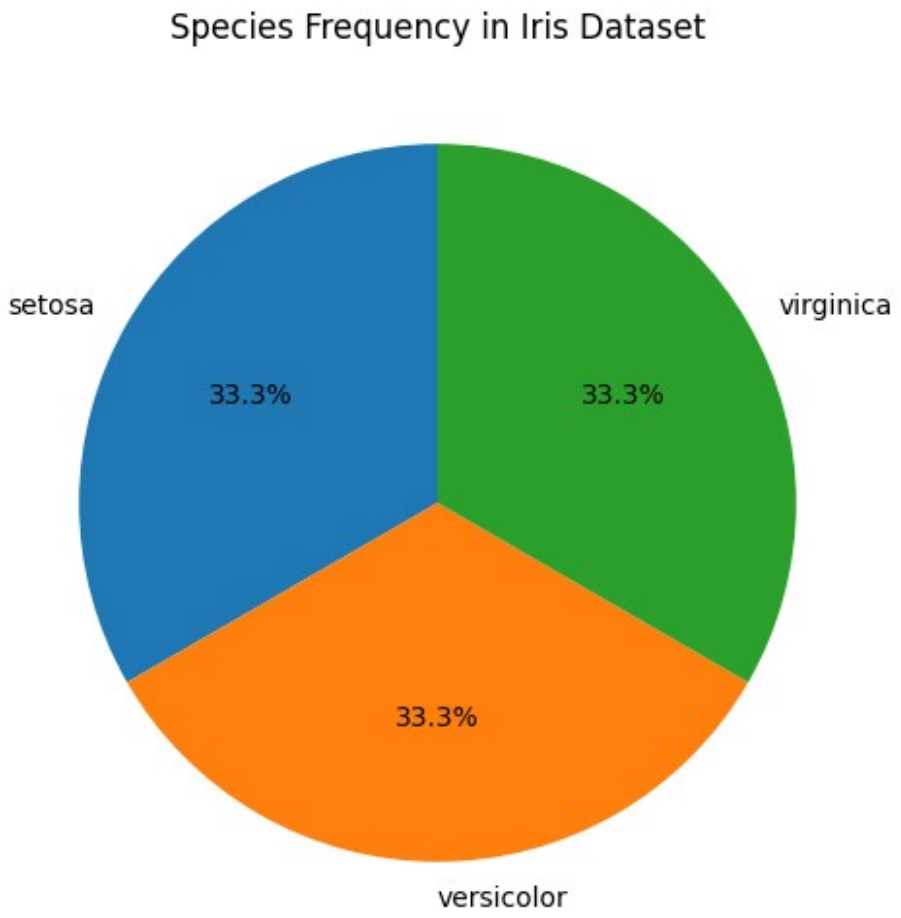


Pie Plot for Species Frequency:

Write a Python program to create a pie chart to display the frequency of the three species(setosa, versicolor, virginica) in the Iris dataset.

```
#Pie Plot for Species Frequency:
species_counts = iris['species'].value_counts()
#figure size
plt.figure(figsize=(8,6))
 # Data to plot
 # Labels for each slice
 # Format for the percentage labels
 # Start angle for the pie chart
plt.pie(species_counts, labels=species_counts.index,autopct='%1.1f%%',
startangle=90)
```

```
plt.title('Species Frequency in Iris Dataset') # adding title to the
pie chart
plt.show() #Display the  plot
```

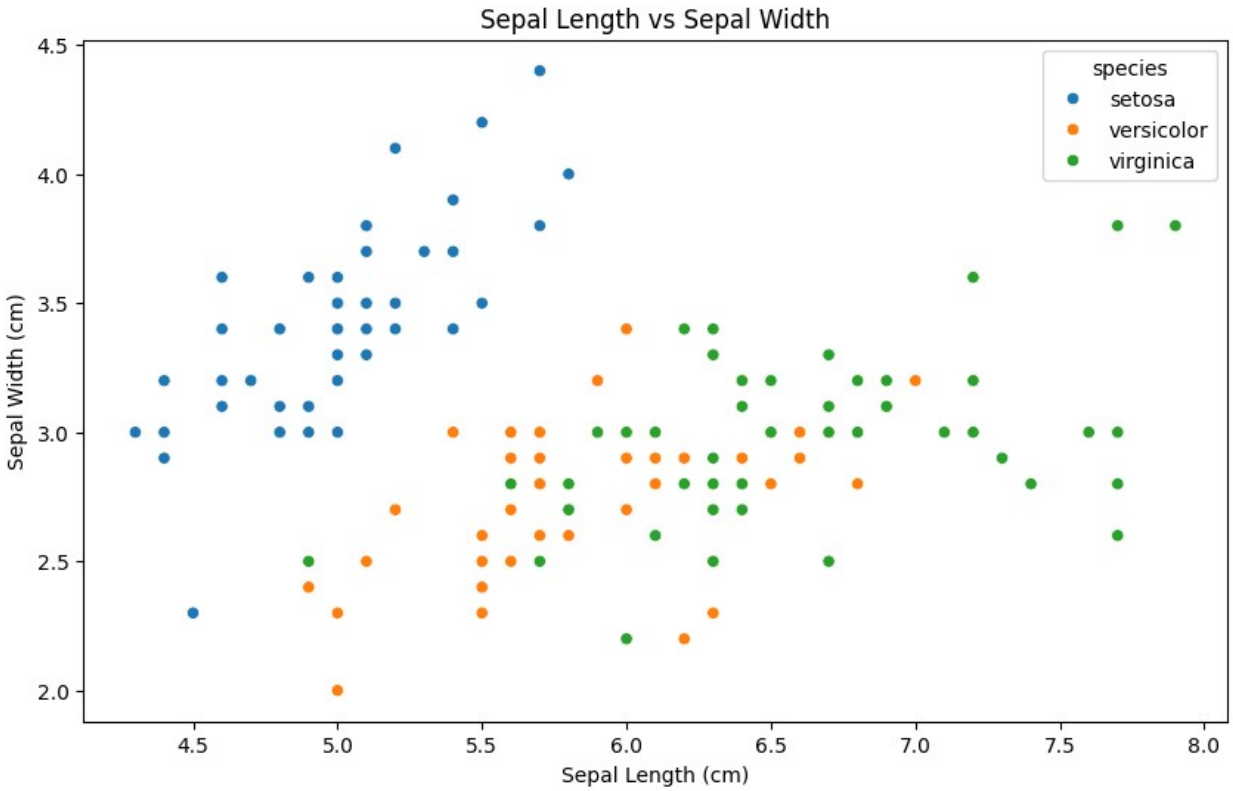


Relationship Between Sepal Length and Width:

Write a Python program to create a scatter plot to find the relationship between sepal length and sepal width for the Iris dataset.

```
#Relationship Between Sepal Length and Width:
plt.figure(figsize=(10, 6)) # figure size
 #Data to plot
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species',
data=iris)
#main title to the scatterplot
plt.title('Sepal Length vs Sepal Width')
#title for xlabel
plt.xlabel('Sepal Length (cm)')
```

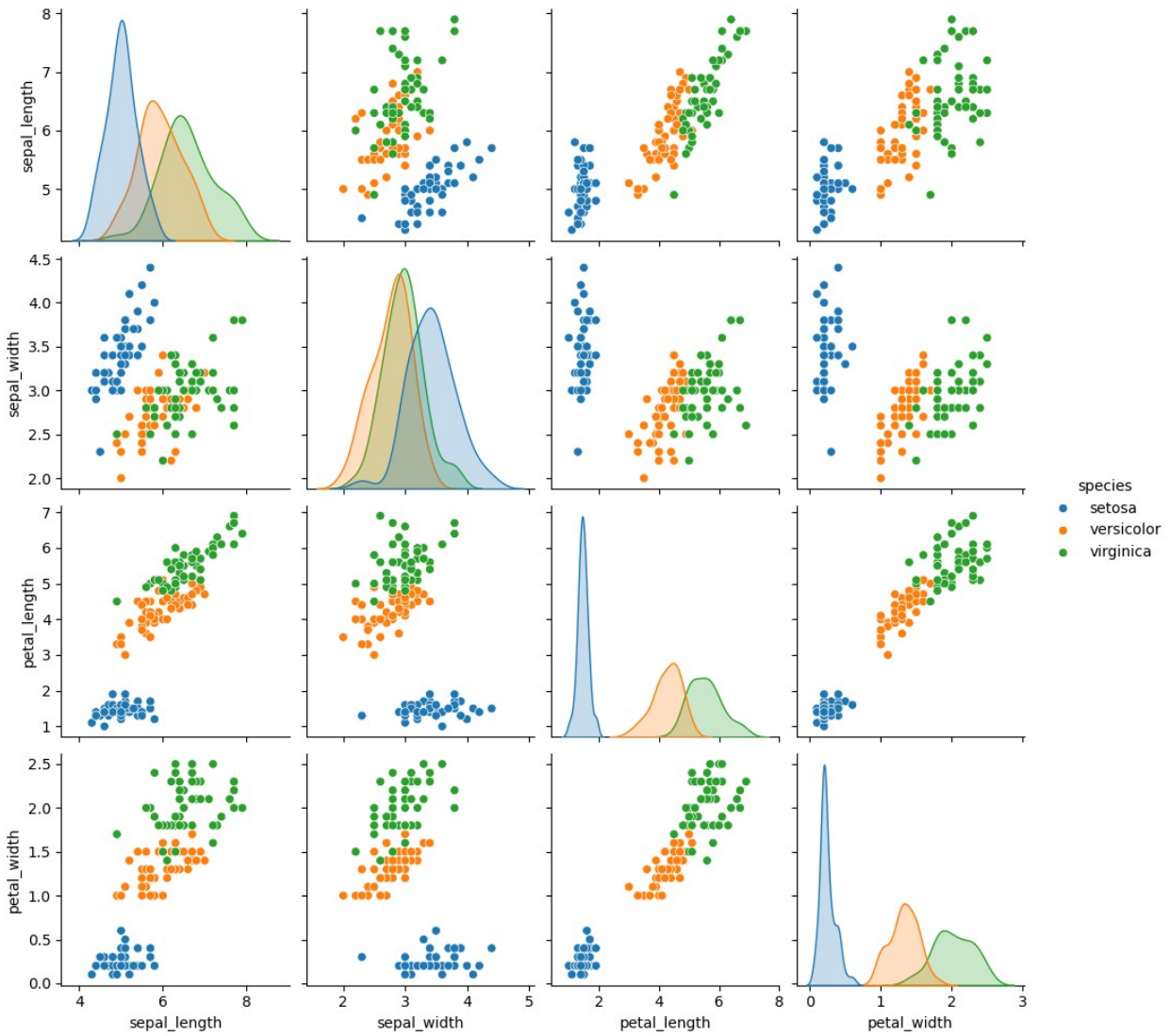
```
#title to the ylabel
plt.ylabel('Sepal Width (cm)')
#Display the plot
plt.show()
```



Distribution of Sepal and Petal Features:

Write a Python program to create a plot that shows how the length and width of sepal length, sepal width, petal length, and petal width are distributed.

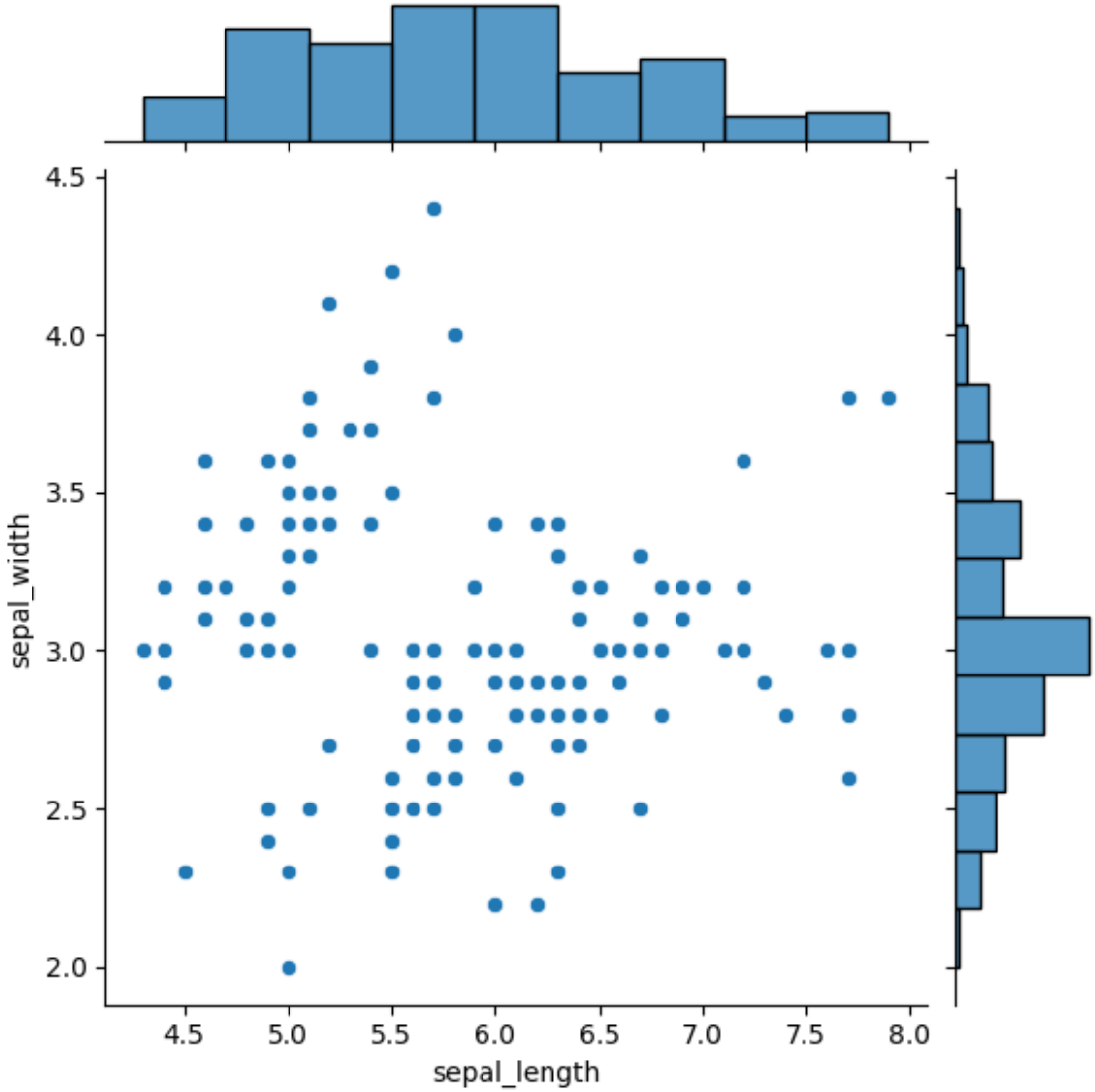
```
# Distribution of Sepal and Petal Features:
sns.pairplot(iris, hue='species', height=2.5)# Creating pairplot using seaborn
plt.show() #Display the plot
```



Jointplot of Sepal Length vs Sepal Width:

Write a Python program to create a joint plot to describe the individual distributions on the same plot between sepal length and sepal width.

```
# Jointplot of Sepal Length vs Sepal Width:
# x and y represents the labels of scatter plot axis
# kind='scatter' specifies that the plot type is a scatter plot
sns.jointplot(x='sepal_length', y='sepal_width', data=iris,
kind='scatter')
plt.show() #Display the plot
```

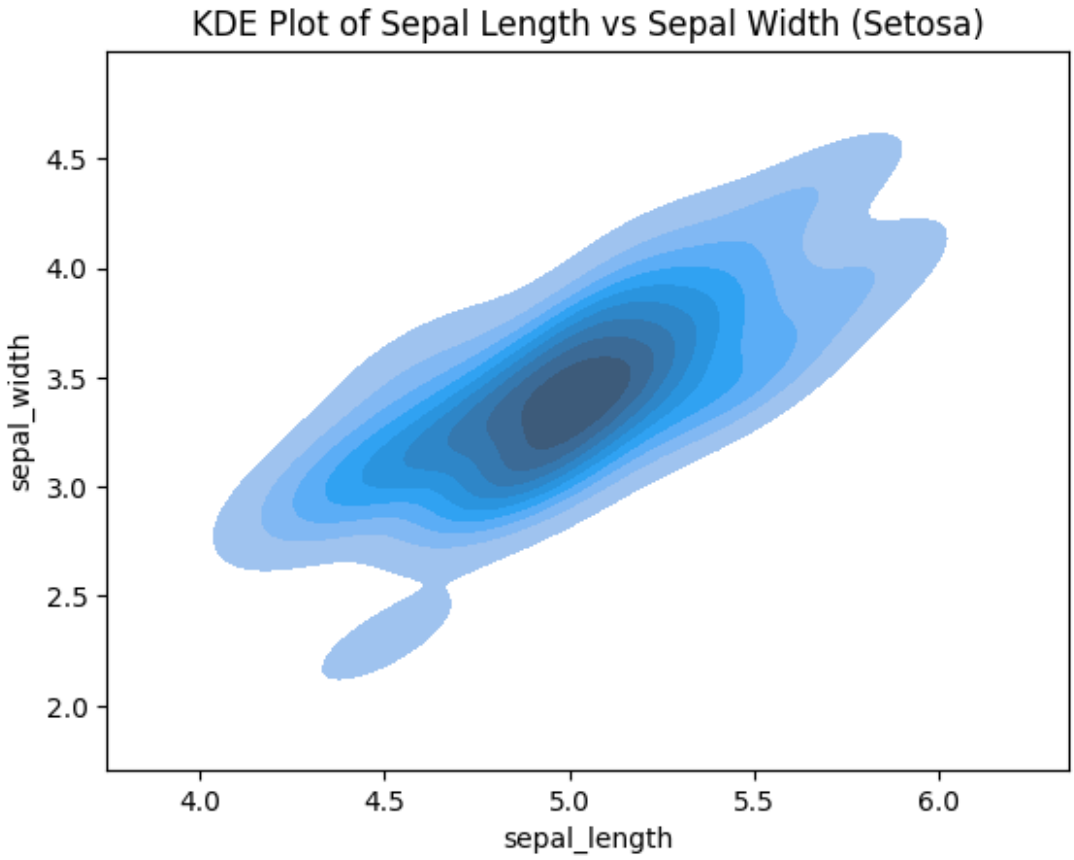


KDE Plot for Setosa Species (Sepal Length vs Sepal Width):

Write a Python program using seaborn to create a KDE (Kernel Density Estimate) plot of sepal length versus sepal width for the setosa species of the Iris dataset.

```
#KDE Plot for Setosa Species (Sepal Length vs Sepal Width):
setosa = iris[iris['species'] == 'setosa']
# sns.kdeplot() creates a smooth estimate of the data's density
# 'x' and 'y' represents the column names for the x and y axes
# 'data' specifies the DataFrame to use for plotting
# 'shade=True' fills the area under the KDE curve with color
sns.kdeplot(x='sepal_length', y='sepal_width', data=setosa, fill=True)
```

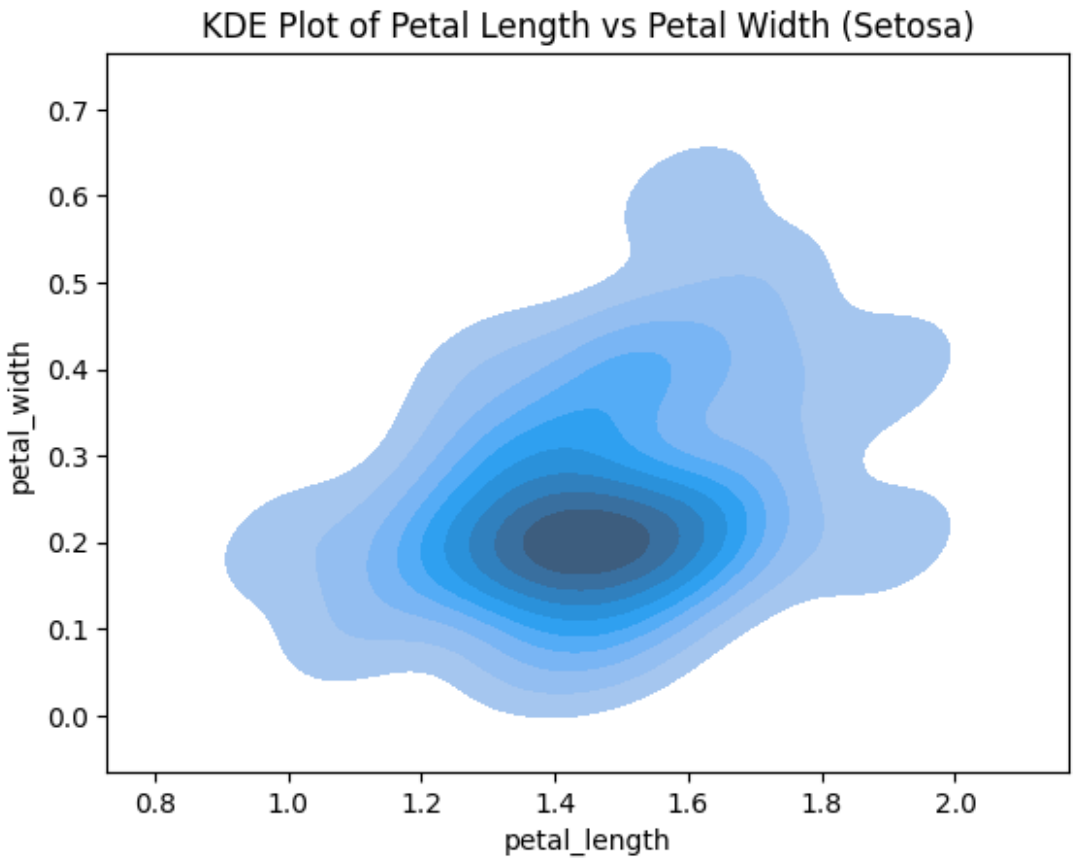
```
plt.title('KDE Plot of Sepal Length vs Sepal Width (Setosa)')#Add
title to the plot
plt.show() # Display the plot
```



KDE Plot for Setosa Species (Petal Length vs Petal Width):

Write a Python program using seaborn to create a KDE plot of petal length versus petal width for the setosa species.

```
#KDE Plot for Setosa Species (Petal Length vs Petal Width):
# sns.kdeplot() creates a smooth estimate of the density of the data
# 'x' and 'y' represents the column names for the x and y axes
# 'data' specifies the DataFrame to use for plotting
# 'shade=True' fills the area under the KDE curve with color
sns.kdeplot(x='petal_length', y='petal_width', data=setosa, fill=True)
plt.title('KDE Plot of Petal Length vs Petal Width (Setosa)') #Add
title
plt.show() #Display the plot
```


KDE Plot of Petal Length vs Petal Width (Setosa)
petal_width
petal_length
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1.0
1.2
1.4
1.6
1.8
2.0