visualization-with-iris-dataset-1

September 17, 2024

#Data Visualization Using Matplotlib and Seaborn using Iris Dataset

#1. Iris Dataset The Iris dataset is a classic dataset in machine learning and statistics. It was introduced by the botanist Edgar Anderson and is often used as a beginner's dataset for testing and demonstrating algorithms. The dataset consists of measurements of iris flowers from three different species: setosa, versicolor, and virginica. Each sample includes four features:

Sepal length

Sepal width

Petal length

Petal width

The dataset is commonly used for classification tasks, where the goal is to predict the species of iris based on these features.

#2. Matplotlib Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is highly customizable and provides a range of plotting options. Key features include:

Basic Plotting: Create line plots, scatter plots, bar charts, histograms, and more.

Customization: Customize plot appearance, such as colors, markers, and line styles.

Subplots: Combine multiple plots into a single figure for comparative analysis.

Interactive Plots: Use interactive backends for zooming and panning.

matplotlib can be imported as shown below:

[10]:
```
import matplotlib.pyplot as plt
```

#3. Seaborn Seaborn is a statistical data visualization library built on top of Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Key features include:

Statistical Plots: Create plots such as violin plots, box plots, and pair plots.

Built-in Datasets: Easily access and visualize a variety of built-in datasets, including the Iris dataset.

Aesthetics: Enhanced color palettes and styles to make plots more visually appealing.

Seaborn can be imported as shown below:

[11]:
```
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)

we can load the dataset using the following code:

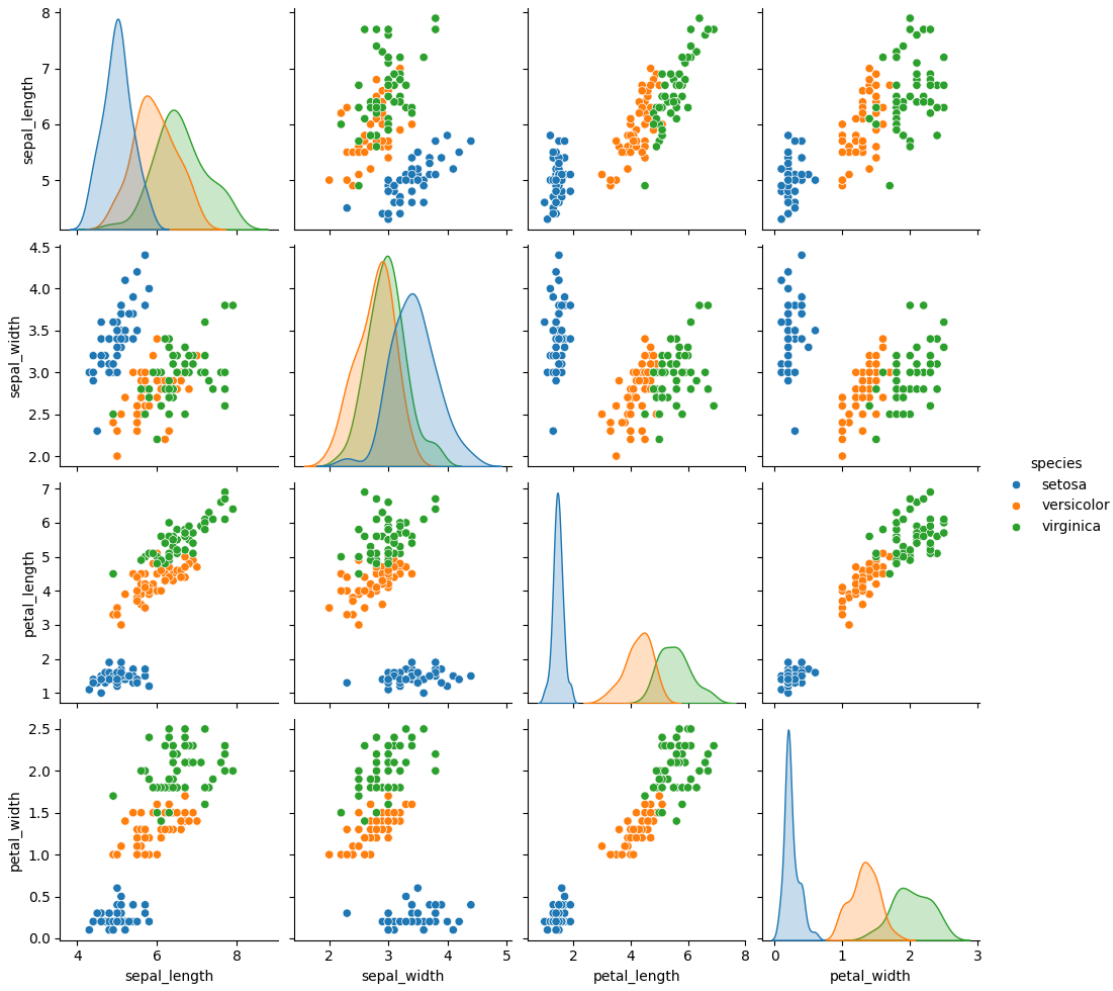
[12]:
```
import seaborn as sns
import matplotlib.pyplot as plt
# Load the Iris dataset
iris= sns.load_dataset('iris')
```

#Exercise Problems

#1. General Statistics Plot (Matplotlib or Seaborn):

o 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

[13]:
```
sns.pairplot(iris, hue='species', height=2.5) # Create a pair plot using Seaborn
plt.show() #Display the plot
```

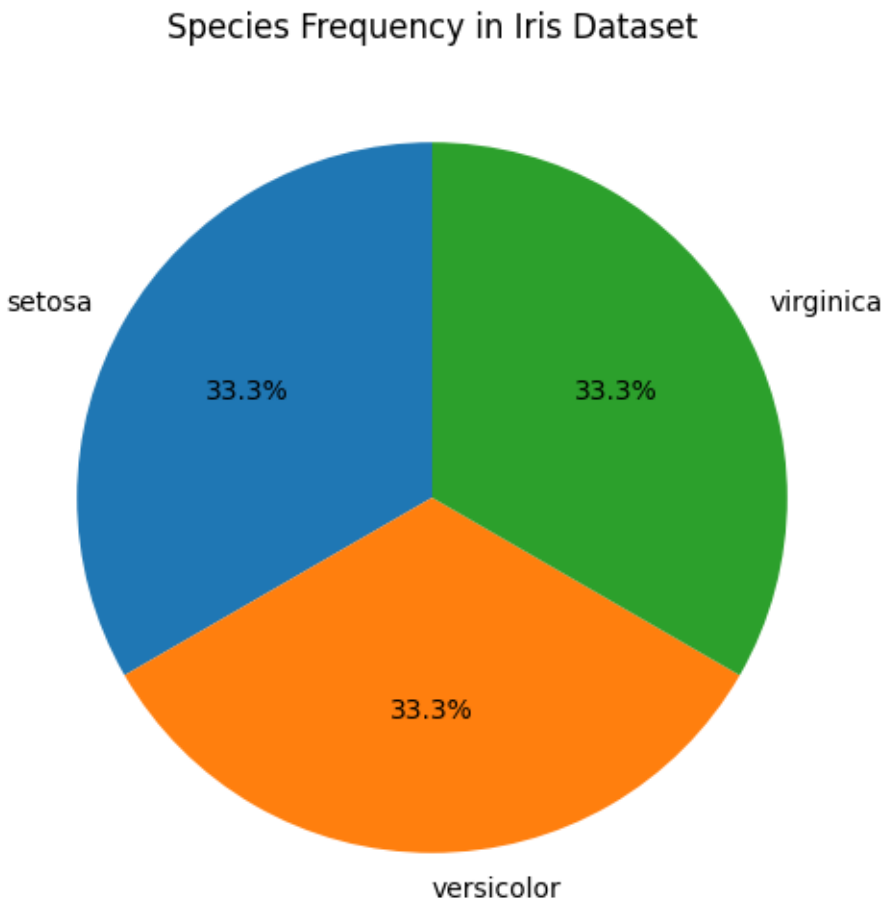


#2. Pie Plot for Species Frequency:

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

[14]:
```
# Calculate the count of each species

species_counts = iris['species'].value_counts()

#fixing the size of the

plt.figure(figsize=(6,6))

 # 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)

# Add title to the pie chart
plt.title('Species Frequency in Iris Dataset')

#Display the  plot
plt.show()
```

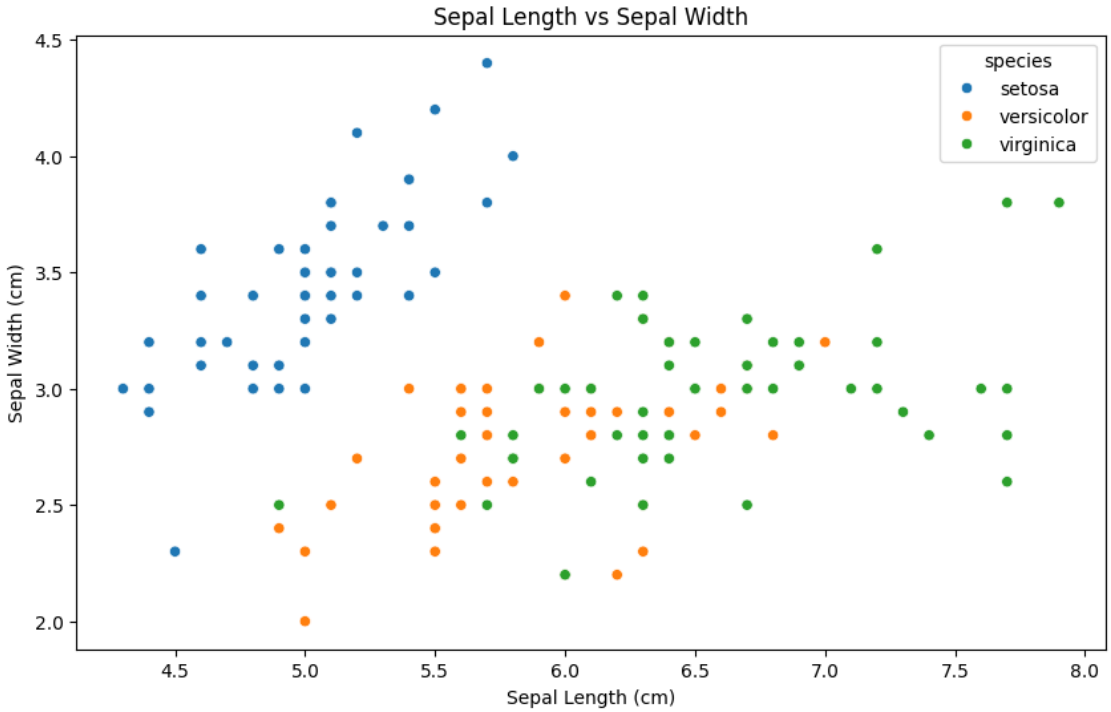


#3. Relationship Between Sepal Length and Width:

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

[15]:
```
#fixing the size
plt.figure(figsize=(10, 6))
```

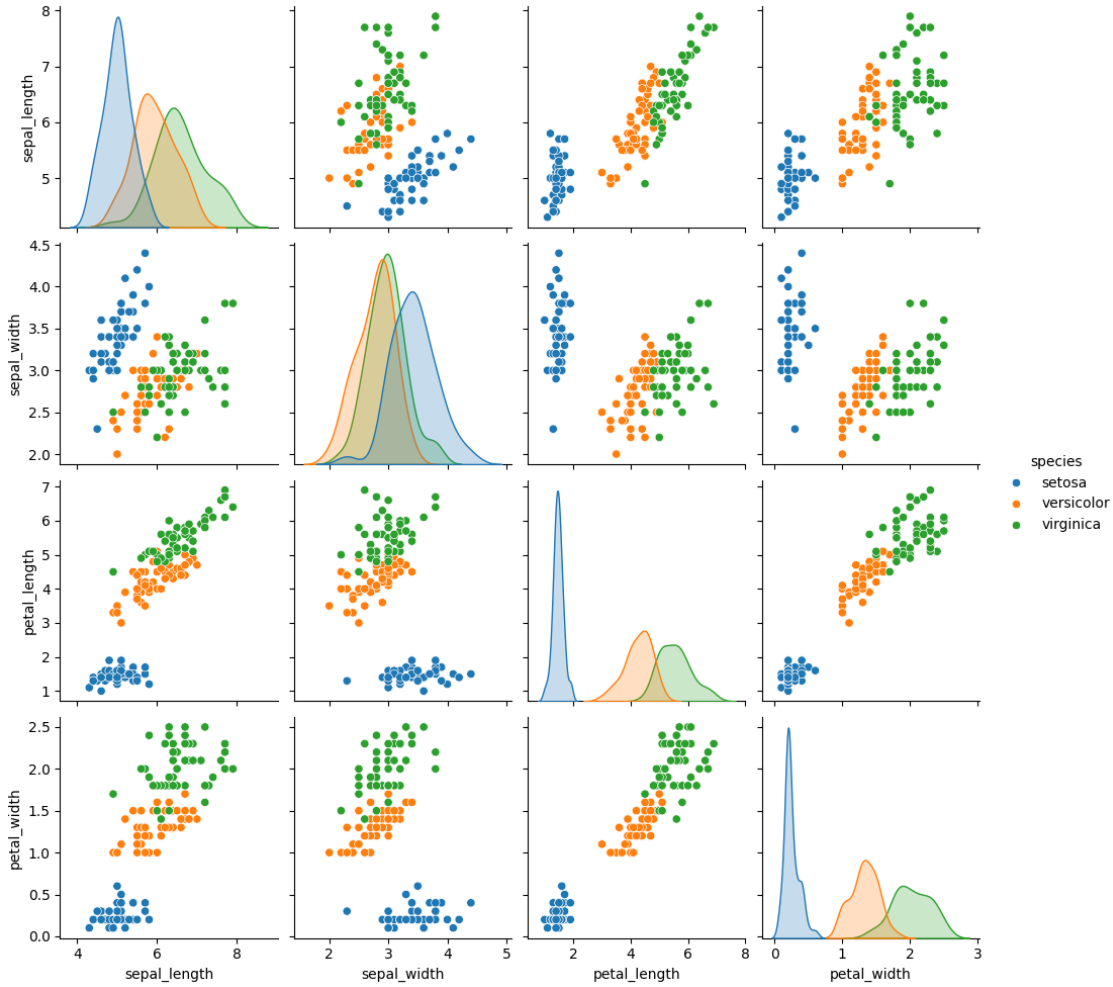
```
 #Data to plot
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', data=iris)

#Add main title to the scatterplot
plt.title('Sepal Length vs Sepal Width')
#Add title for xlabel
plt.xlabel('Sepal Length (cm)')
#Add title to the ylabel
plt.ylabel('Sepal Width (cm)')
#Display the plot
plt.show()
```



#Exercise 2

##1. Distribution of Sepal and Petal Features: o 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.

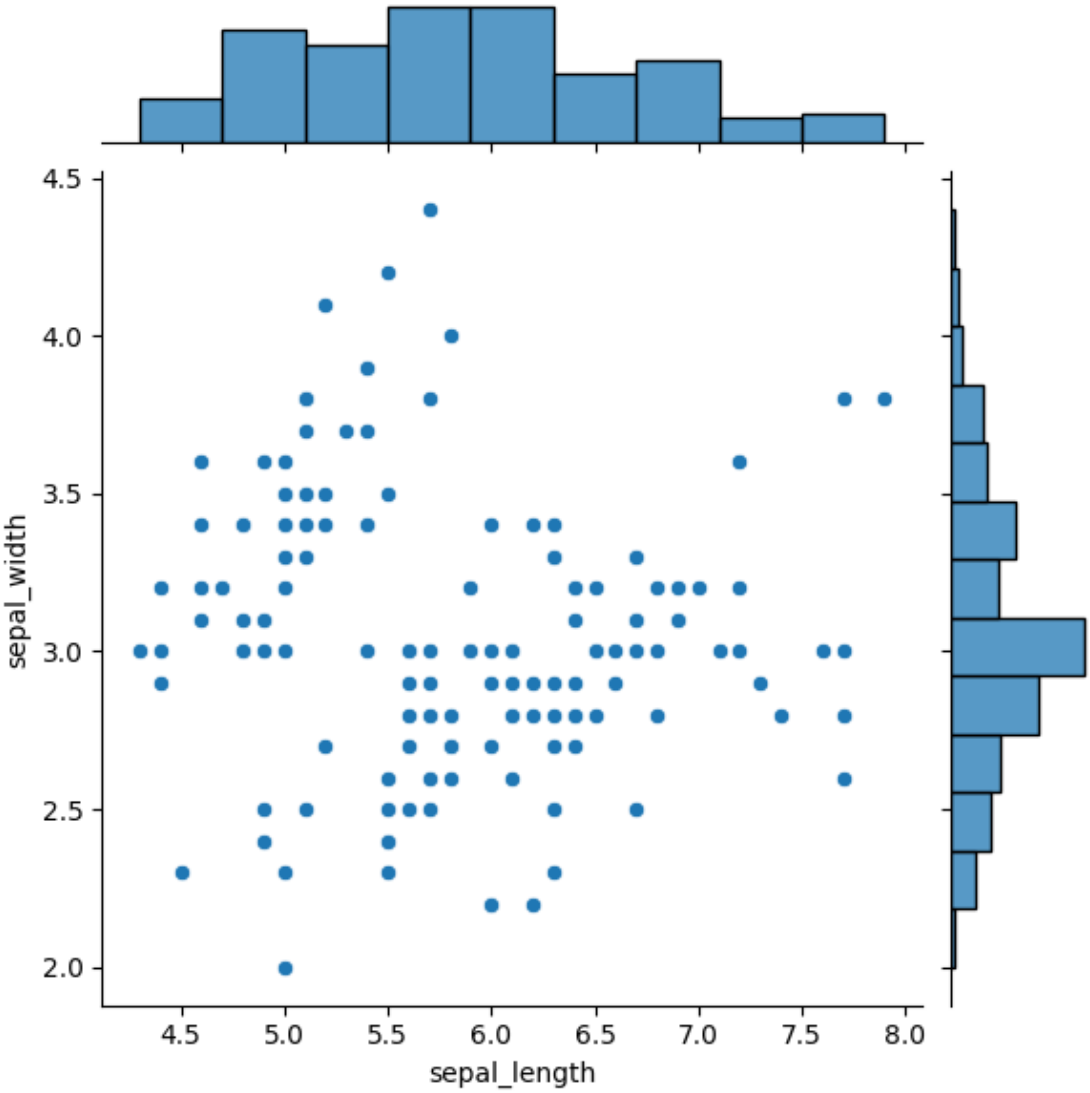
[16]:
```
sns.pairplot(iris, hue='species', height=2.5)# Creating pairplot using seaborn
plt.show() #Display the plot
```



#2. Jointplot of Sepal Length vs Sepal Width:

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

[17]:
```
# Generate a joint plot using seaborn
# sns.jointplot() creates a multi-panel plot with scatter plots and histograms
# 'x' and 'y' specify the column names for the scatter plot axes
# 'data' specifies the DataFrame to use
# '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
```



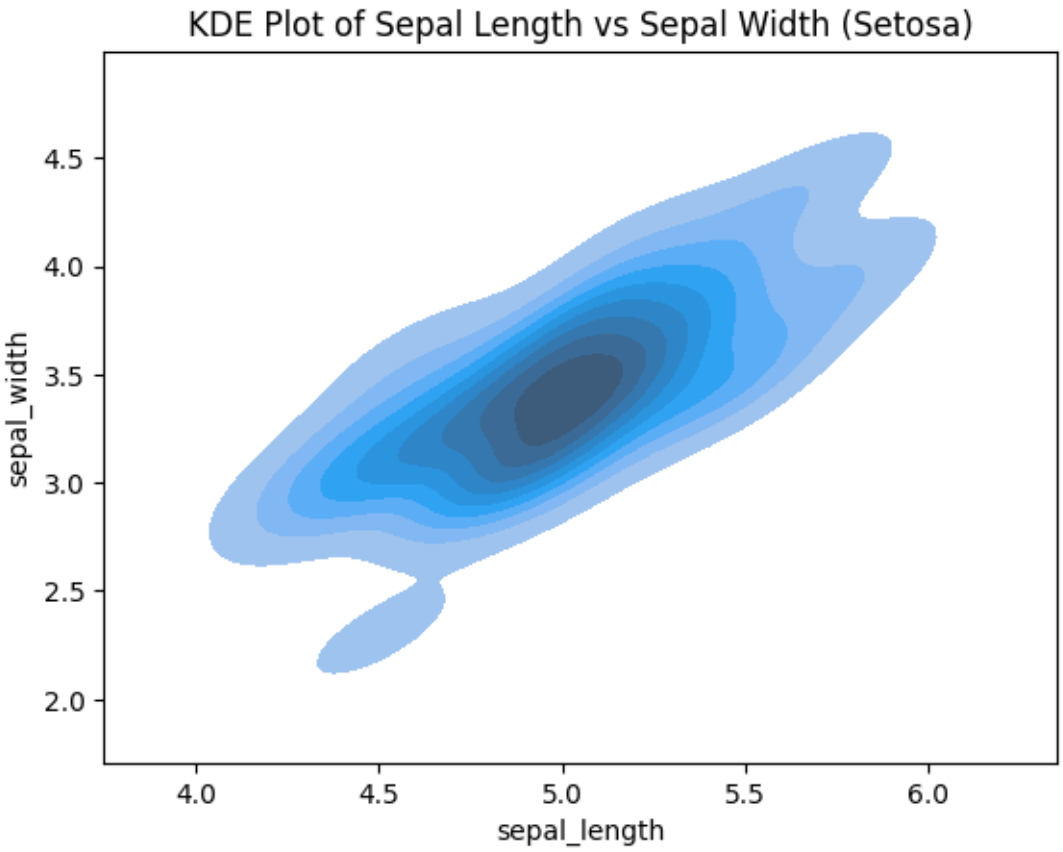
#3. KDE Plot for Setosa Species (Sepal Length vs Sepal Width):

o 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.

[18]:
```
# This creates a new DataFrame called setosa that only contains data for the␣
 ↪Setosa species
setosa = iris[iris['species'] == 'setosa']

# Create a Kernel Density Estimate (KDE) plot
# sns.kdeplot() creates a smooth estimate of the data's density
# 'x' and 'y' specify 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, shade=True)
plt.title('KDE Plot of Sepal Length vs Sepal Width (Setosa)')#Add title to the␣
 ↪plot
plt.show() # Display the plot
```

```
C:\Users\sivasai\AppData\Local\Temp\ipykernel_3496\1825985453.py:9:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(x='sepal_length', y='sepal_width', data=setosa, shade=True)
```



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

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

[19]:
```
# Create a Kernel Density Estimate (KDE) plot
# sns.kdeplot() creates a smooth estimate of the density of the data
# 'x' and 'y' specify 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, shade=True)
plt.title('KDE Plot of Petal Length vs Petal Width (Setosa)') #Add title
plt.show() #Display the plot
```

```
C:\Users\sivasai\AppData\Local\Temp\ipykernel_3496\2579371601.py:7:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(x='petal_length', y='petal_width', data=setosa, shade=True)
```