data-visualization

September 17, 2024

M.MOHIT

CSD-A

A22126551042

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.

- **Main Features of Matplotlib:**
  - High-quality 2D and 3D plots.
  - Extensive customization options (color, labels, grids, etc.).
  - Integration with other libraries like NumPy, pandas, and more.

Basic Syntax: To use Matplotlib, first import the pyplot module, which provides a MATLAB-like interface for creating plots:

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.

Main Features of Seaborn: * Automatic handling of data frames and statistical plotting. * Built-in themes for aesthetics and improved readability. * Complex plots like violin plots, pair plots, heatmaps, and more, with minimal code.

Basic Syntax: You can start using Seaborn by importing it as:

import seaborn as sns

[2]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

#**Loading iris dataset**:

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)

[4]:
```
iris = sns.load_dataset('iris')
iris
```

[4]:

	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

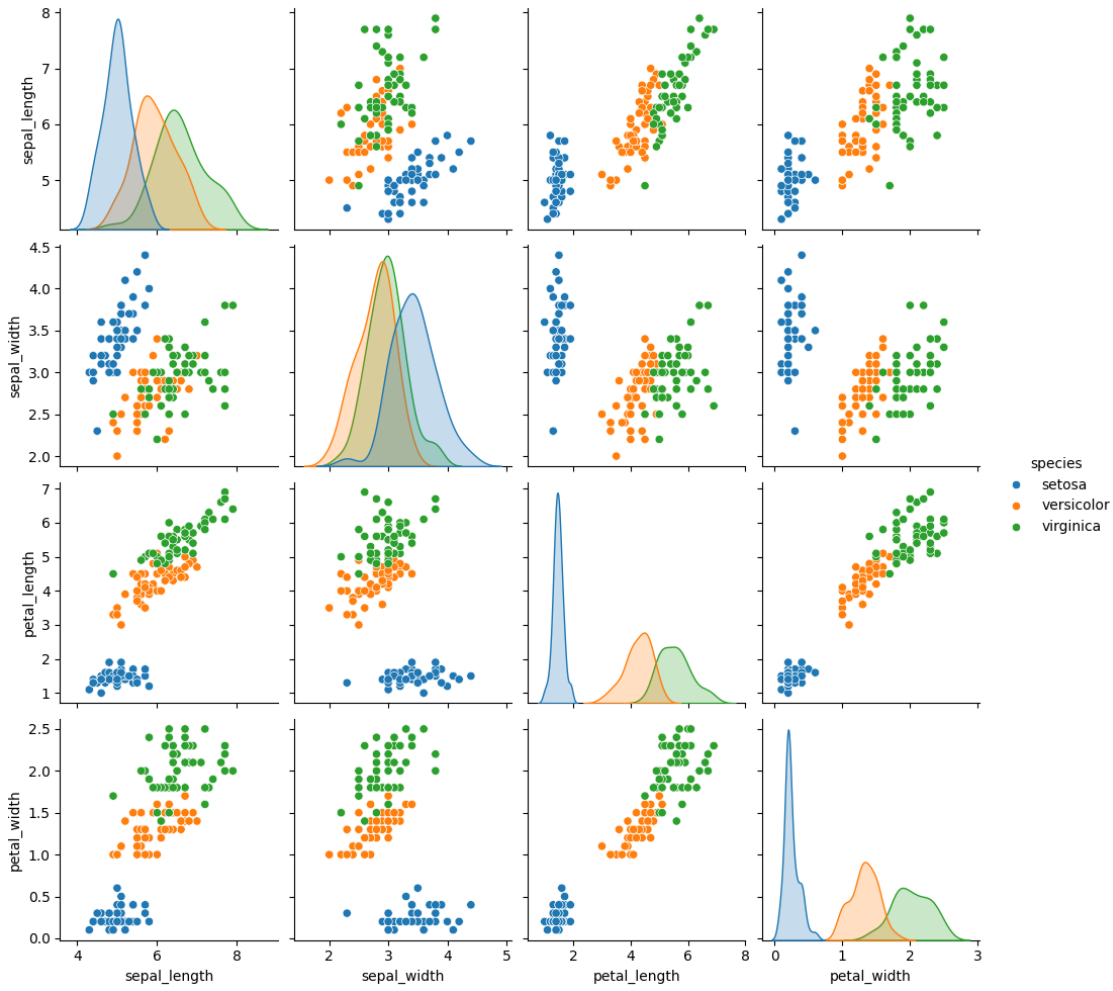
[150 rows x 5 columns]

##**EXERCISE PROBLEMS:**

#**1. 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.

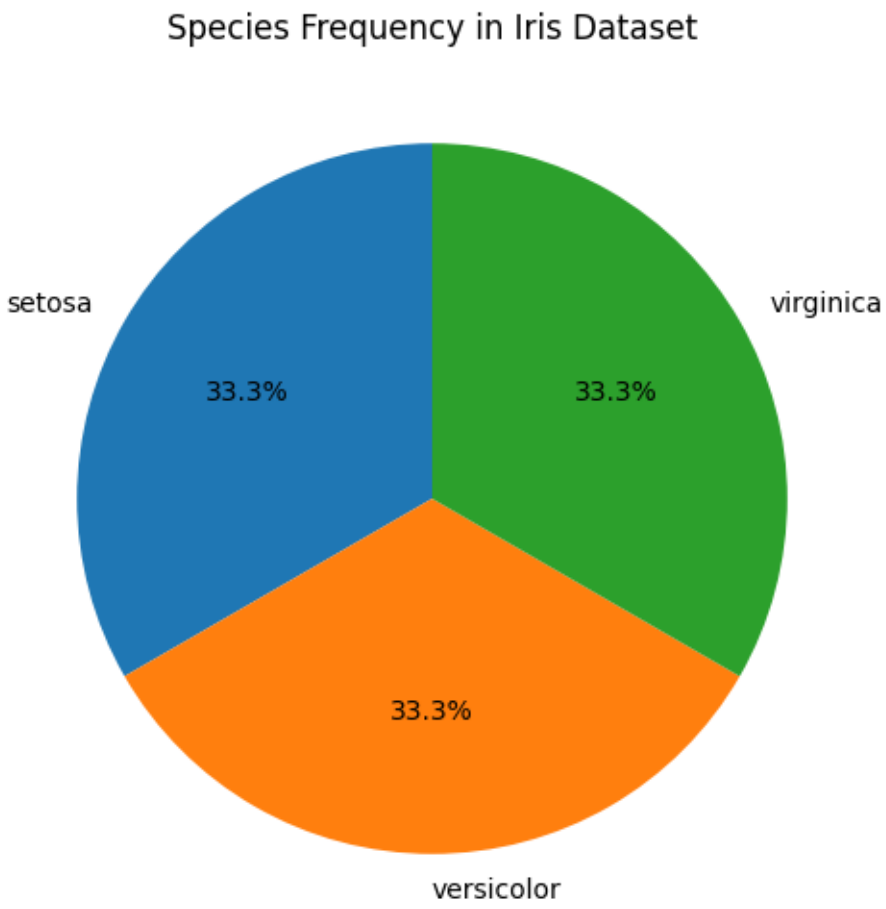
[5]:
```
sns.pairplot(iris, hue='species', height=2.5)
plt.show()
```



#2. 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.

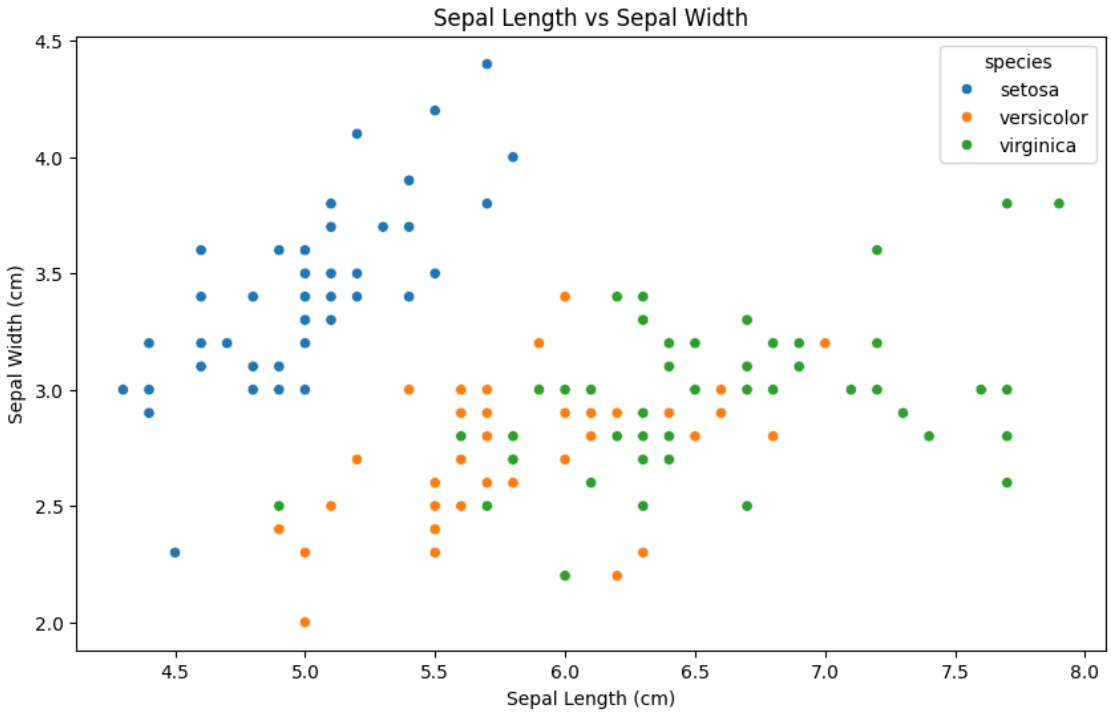
[ ]:
```
species_counts = iris['species'].value_counts()
plt.figure(figsize=(9,9))
plt.pie(species_counts,labels = species_counts.index, autopct="%1.1f%%",␣
 ↪startangle=90)
plt.title('Species Frequency in Iris Dataset')
plt.show()
```



#3. Relationship Between Sepal Length and Sepal width:

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

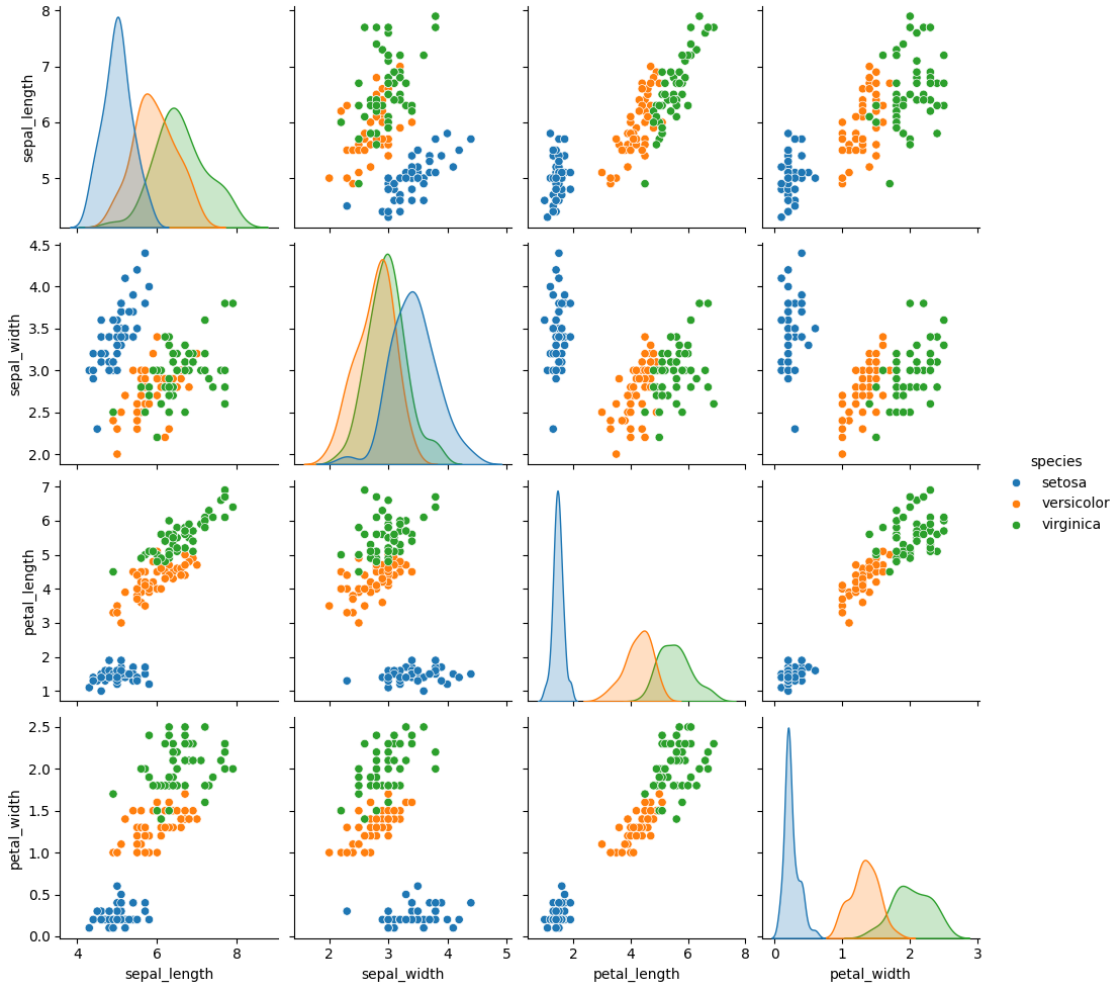
[ ]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', data = iris)
plt.title('Sepal Length vs Sepal Width')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
plt.show()
```



#4. 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.

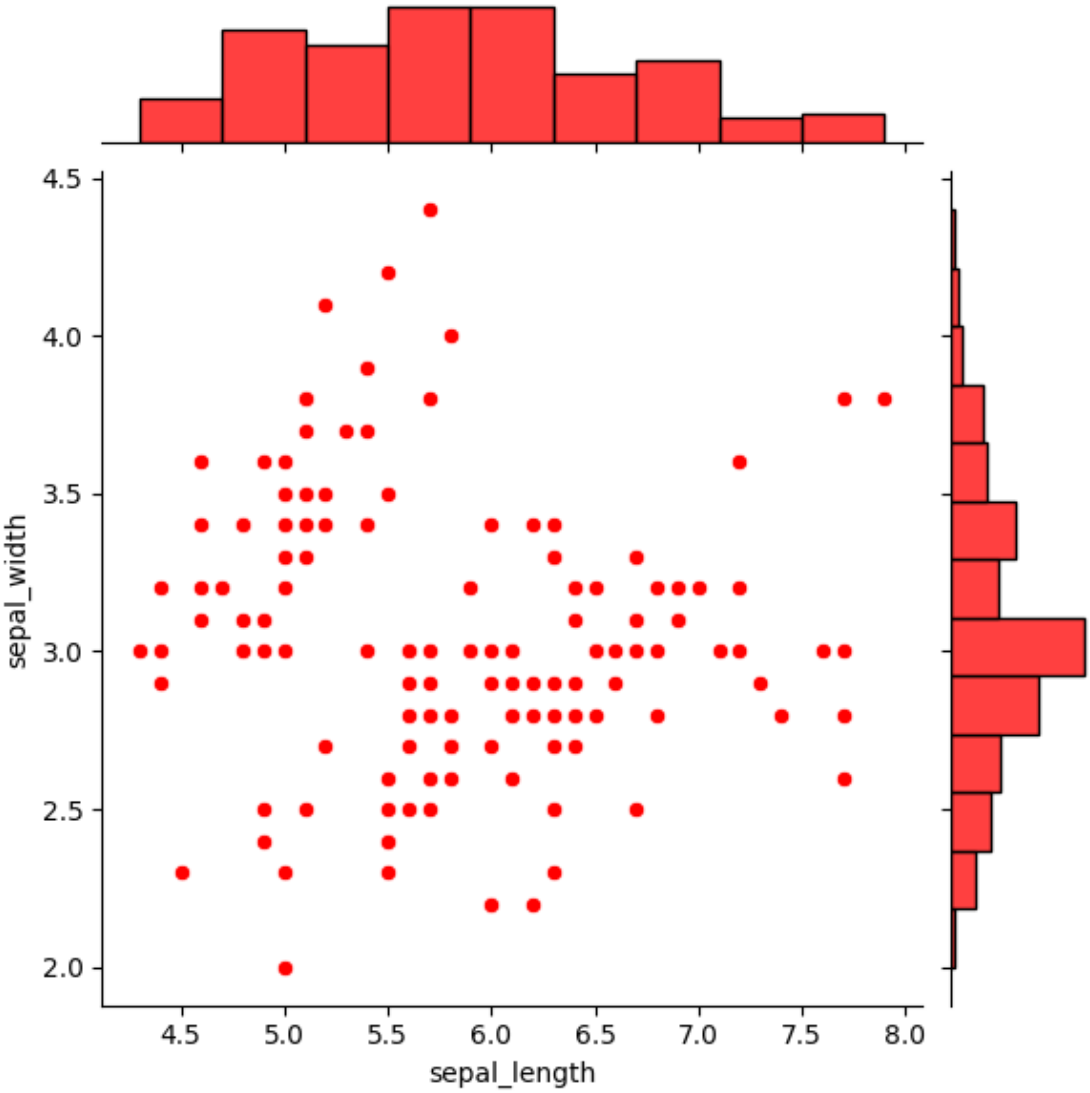
[ ]:
```
sns.pairplot(iris, hue='species', height = 2.5)
plt.show()
```



#5. 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.

[10]:
```
sns.jointplot(x='sepal_length', y='sepal_width', data = iris, kind='scatter',␣
↪color='r')
plt.show()
```



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

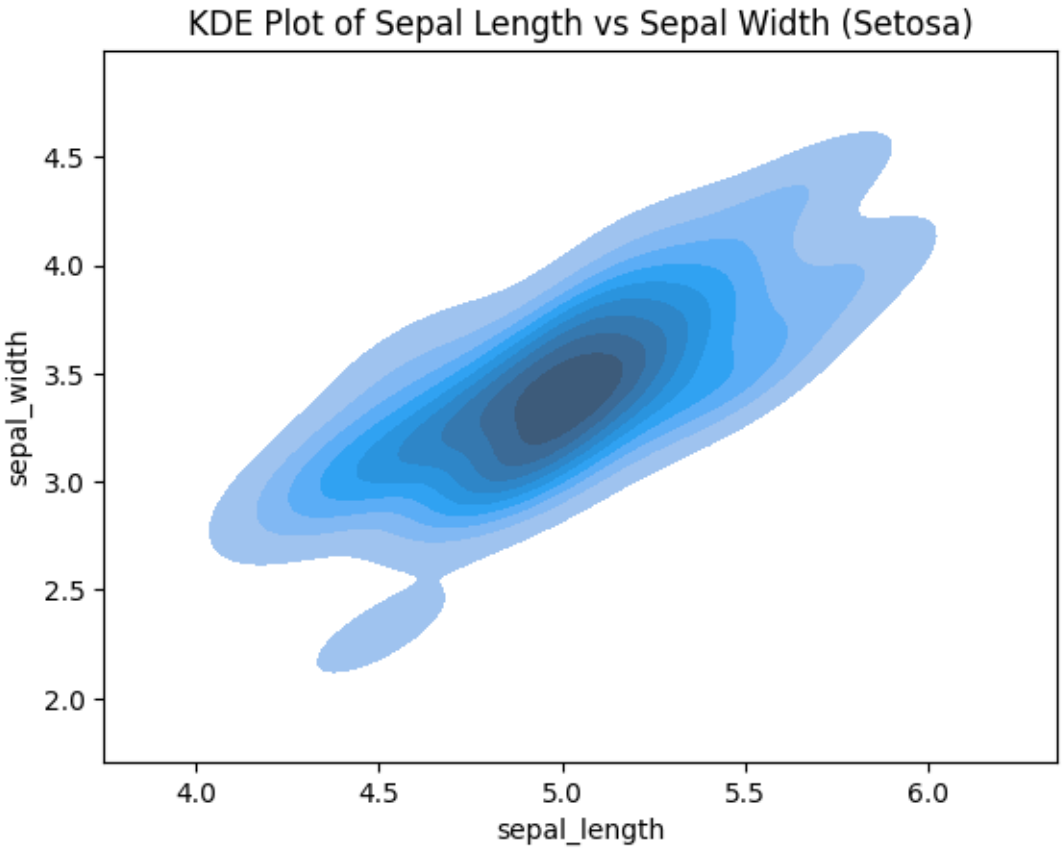
- Write a Python program using seaborn to create a KDE (Kernel Density Estimate) plot of sepal length versus sepal width for the setosa species of the Iris dataset.

[11]:
```
setosa = iris[iris['species'] == 'setosa']
sns.kdeplot(x='sepal_length', y='sepal_width', data=setosa, shade=True)
plt.title('KDE Plot of Sepal Length vs Sepal Width (Setosa)')
plt.show()
```

```
<ipython-input-11-5d03745e2a63>:2: 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)
```



#7. KD plot for Setosa Species (Petal Length vs Petal Width):

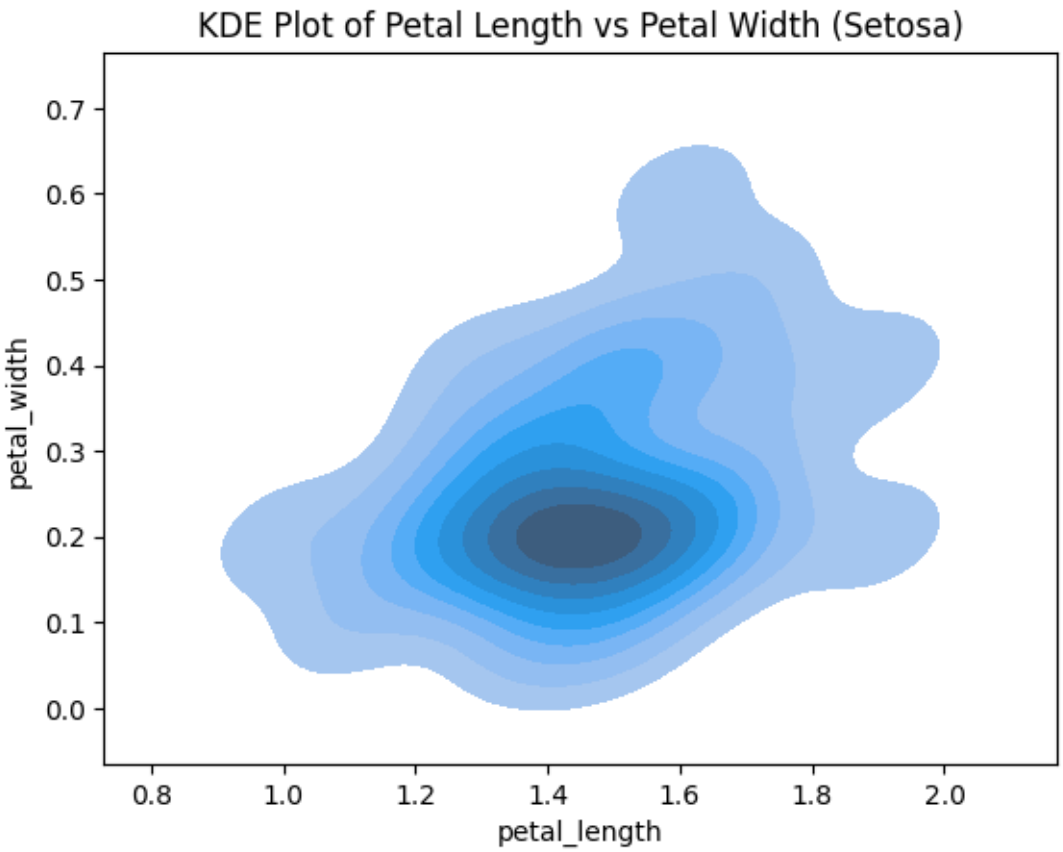
- Write a Python program using seaborn to create a KDE plot of petal length versus petal width for the setosa species.

[12]:
```
sns.kdeplot(x='petal_length', y='petal_width', data=setosa, shade=True)
plt.title('KDE Plot of Petal Length vs Petal Width (Setosa)')
plt.show()
```

```
<ipython-input-12-3e195f3ed5bb>:1: 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)
```


KDE Plot of Petal Length vs Petal Width (Setosa)
petal_width
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
petal_length