irisdata (2)

September 17, 2024

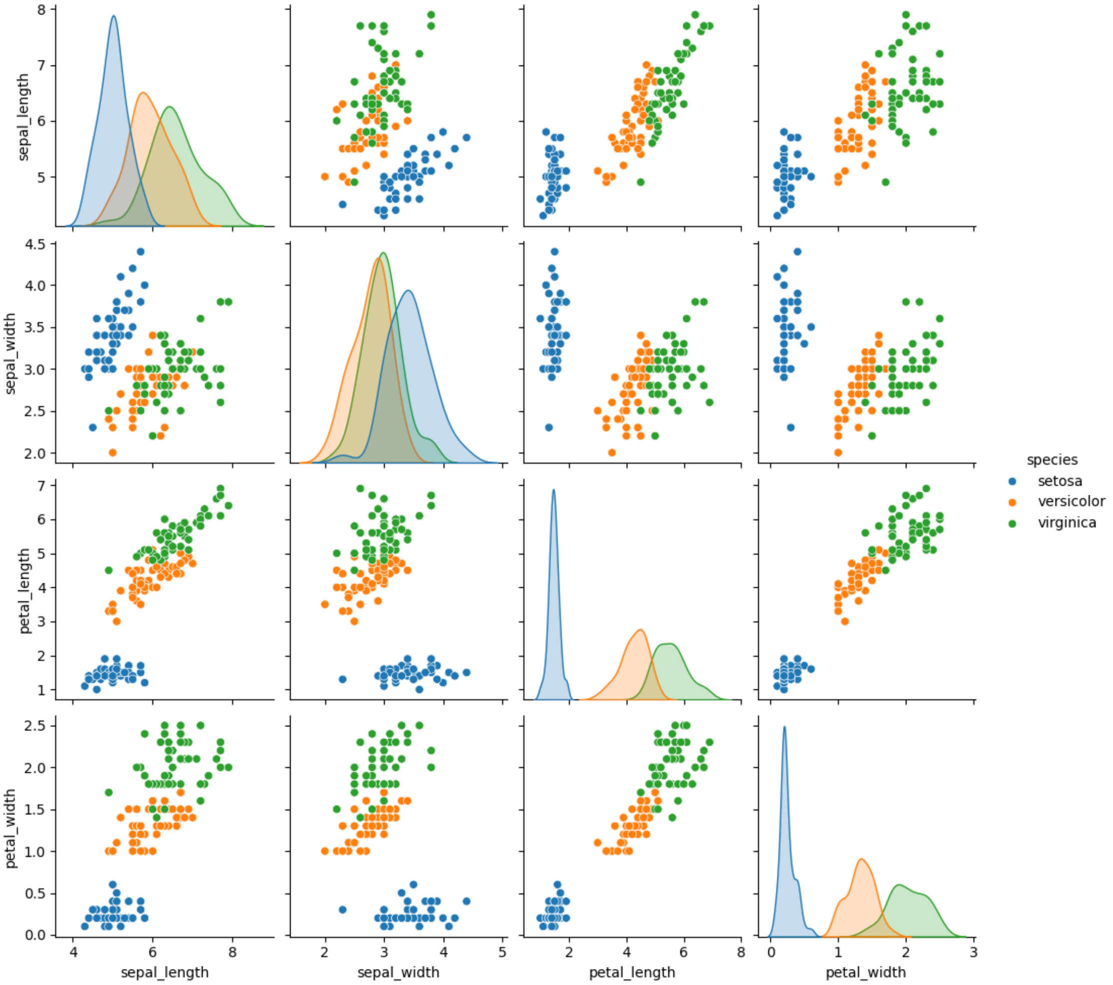
[ ]:
```python
import seaborn as sns
```

[ ]:
```python
import matplotlib.pyplot as plt
#load dataset
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):

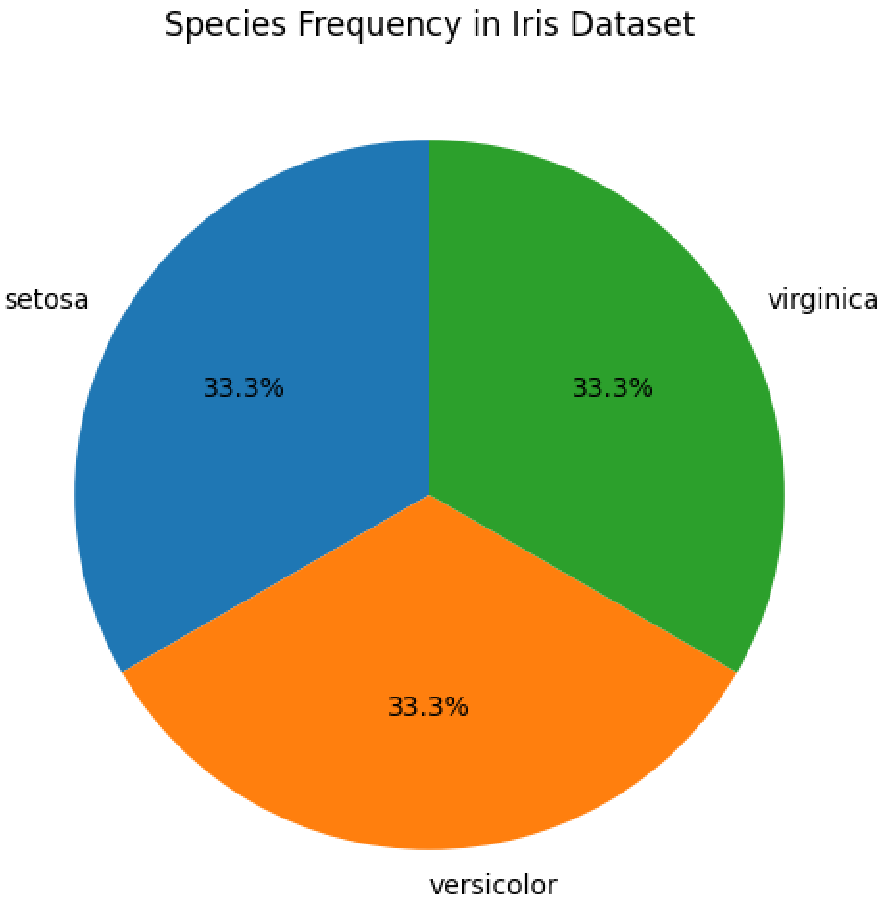
[ ]:
```python
#hue='species' colors the points by species.
#height=2.5 sets the size of each subplot in the grid.
sns.pairplot(iris, hue='species', height=2.5)
plt.show()
```



Pie Plot for Species Frequency:

[ ]:

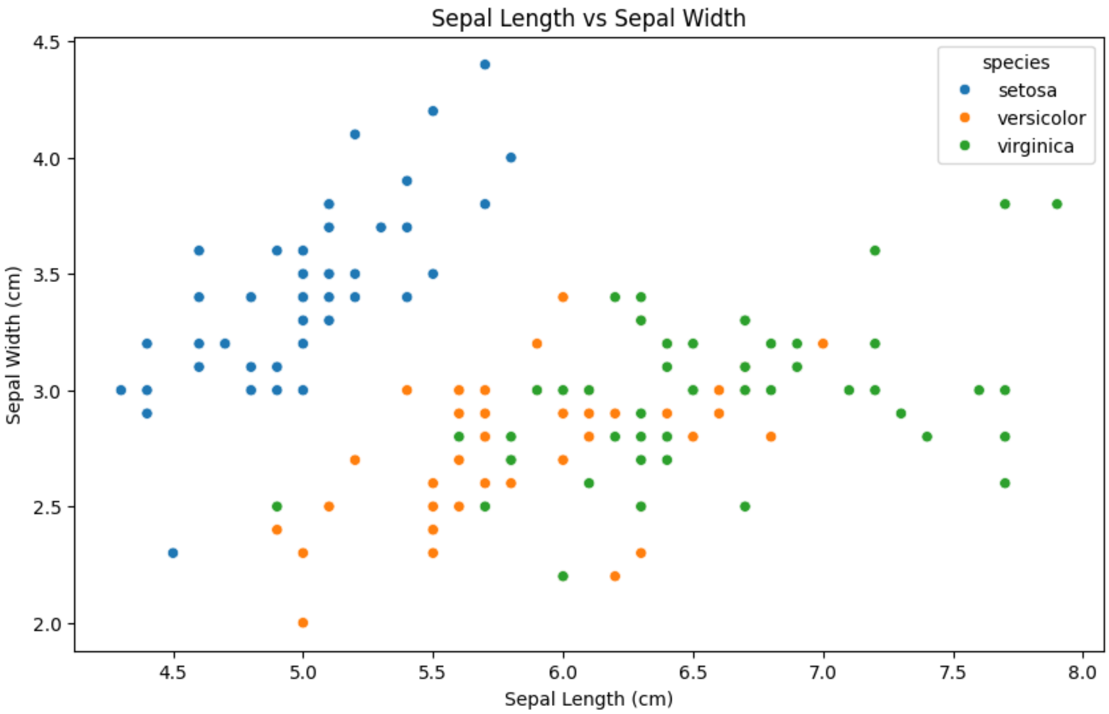
```
# Count the occurrences of each species
# Calculate the frequency of each species in the 'species' column of the iris DataFrame.
species_counts = iris['species'].value_counts()

# Create a new figure for the pie chart with a specified size (6 inches by 6 inches).
plt.figure(figsize=(6, 6))

# Create the pie chart using the species counts.
# labels: Set the labels for each slice to the species names (index of species_counts).
# autopct: Format the percentage values displayed on each slice with one decimal place.
# startangle: Set the starting angle of the first slice to 90 degrees (top␣
 ↪center).
plt.pie(species_counts, labels=species_counts.index, autopct='%1.1f%%',␣
 ↪startangle=90)

# Set the title of the pie chart.
plt.title('Species Frequency in Iris Dataset')

# Display the pie chart.
plt.show()
```



Relationship Between Sepal Length and Width:

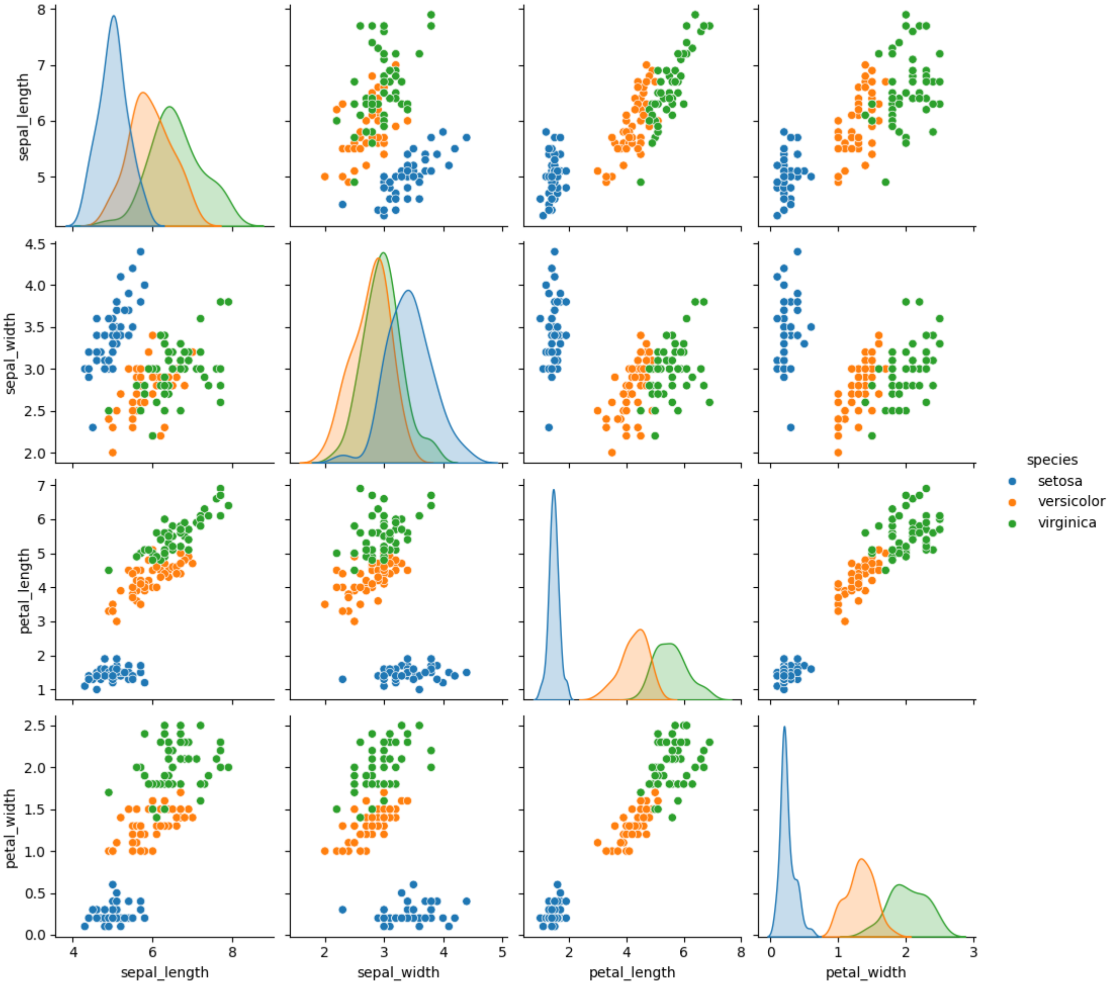
[ ]:
```
# Create a new figure for the scatter plot with a specified size (10 inches by␣
 ↪6 inches).
plt.figure(figsize=(10, 6))
# Create a scatter plot using Seaborn.
```

```python
# x: Set the x-axis to 'sepal_length'.
# y: Set the y-axis to 'sepal_width'.
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', data=iris)
# Set the title of the scatter plot.
plt.title('Sepal Length vs Sepal Width')
# Set the label for the x-axis.
plt.xlabel('Sepal Length (cm)')
# Set the label for the y-axis.
plt.ylabel('Sepal Width (cm)')
# Display the scatter plot.
plt.show()
```



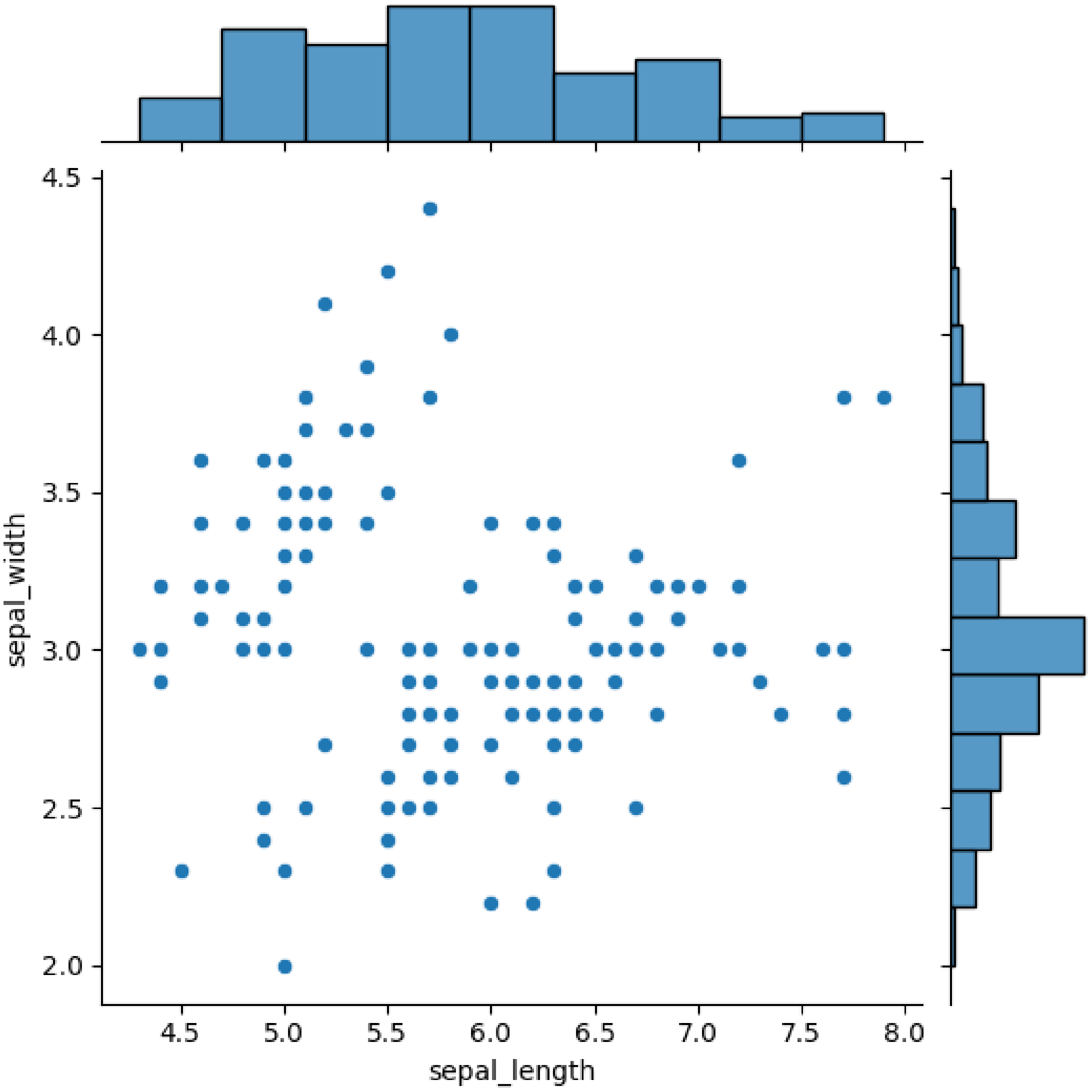
Distribution of Sepal and Petal Features:

```python
[ ]: # Create a pair plot using Seaborn to visualize relationships between all pairs of features in the iris dataset.
# hue: Color the points in each plot based on the 'species' column.
# height: Set the height of each individual plot to 2.5 inches.
sns.pairplot(iris, hue='species', height=2.5)
# Display the pair plot.
plt.show()
```



Jointplot of Sepal Length vs Sepal Width

[ ]:
```
# Create a joint plot using Seaborn to visualize the relationship between sepal length and sepal width.
# x: Set the x-axis to 'sepal_length'.
# y: Set the y-axis to 'sepal_width'.
# data: Use the iris DataFrame for the data.
sns.jointplot(x='sepal_length', y='sepal_width', data=iris, kind='scatter')
# Display the joint plot.
plt.show()
```

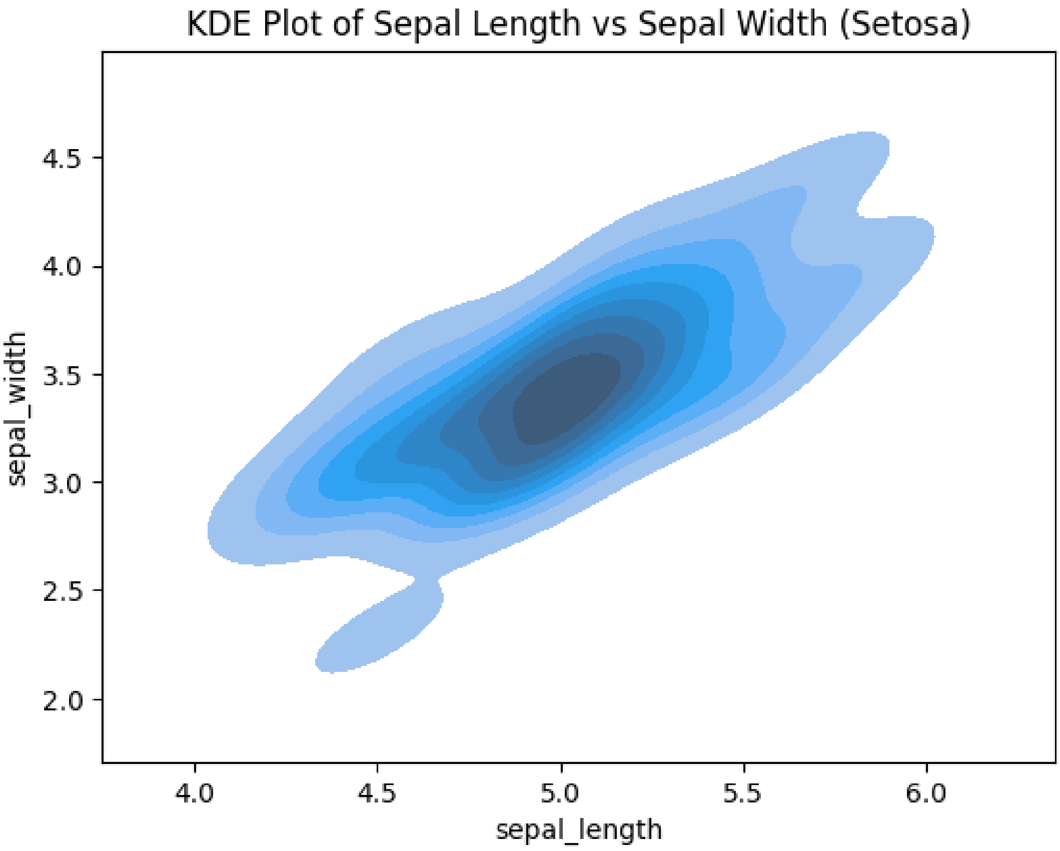


KDE Plot for Setosa Species(Sepal Length vs Sepal Width)

```
[ ]: # Create a new DataFrame called 'setosa' that contains only the rows from the␣
      ↪iris DataFrame where the 'species' is 'setosa'.
     setosa = iris[iris['species'] == 'setosa']

     # Create a KDE plot using Seaborn to visualize the distribution of sepal length␣
      ↪and sepal width for the setosa species.
     # x: Set the x-axis to 'sepal_length'.
     # y: Set the y-axis to 'sepal_width'.
     # data: Use the setosa DataFrame for the data.
     sns.kdeplot(x='sepal_length', y='sepal_width', data=setosa, shade=True)
```

```python
# Set the title of the KDE plot.
plt.title('KDE Plot of Sepal Length vs Sepal Width (Setosa)')

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

```
<ipython-input-25-61f82c7c836f>: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)
```



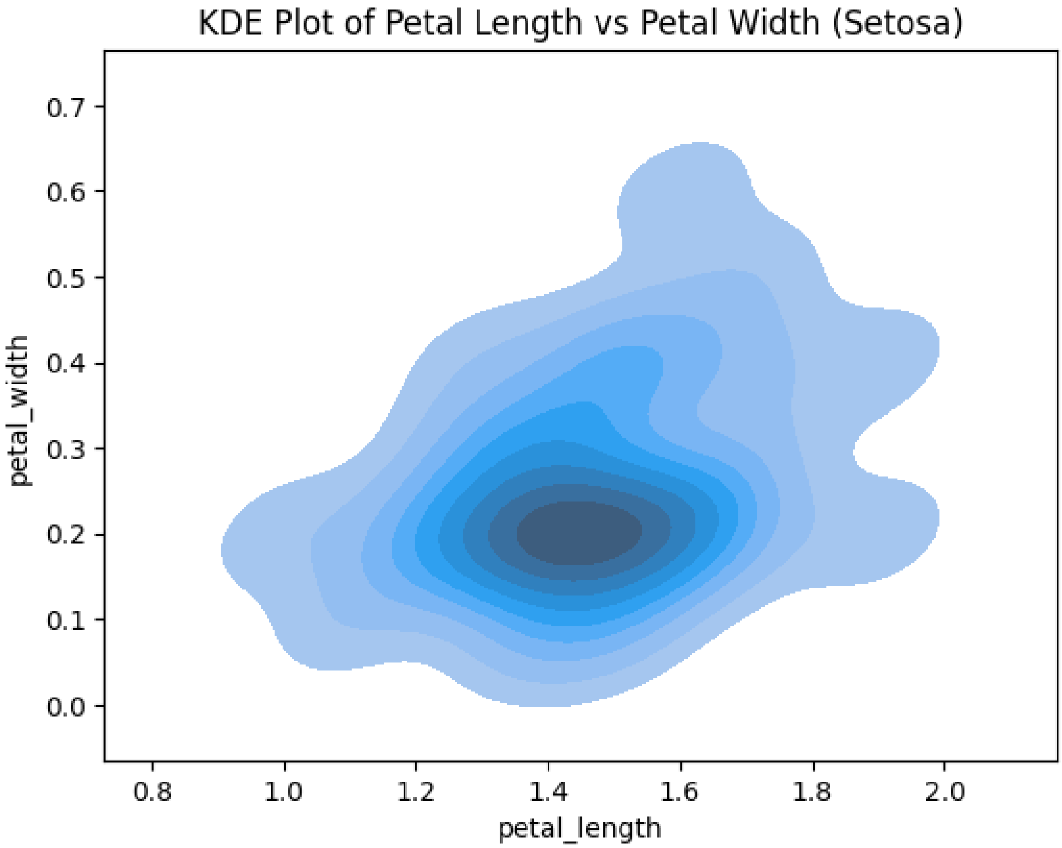
KDE Plot for Setosa Species (Petal Length vs Petal Width)

[ ]:

```python
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-18-5a61f912262a>: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
petal_length