data-visualization-1

September 17, 2024

0.1 DATA VISUALIZATION WITH MATPLOTLIB AND SEABORN USING IRIS DATASET

Data visualization is the graphical representation of data and information. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to observe and understand trends, outliers, patterns, and relationships within datasets. Python's Matplotlib and Seaborn libraries are two widely used tools for creating such visualizations.

0.2 1. Matplotlib:

Matplotlib is a fundamental data visualization library in Python that offers great flexibility and control over plots. It allows users to create static, animated, and interactive visualizations. Matplotlib serves as the backbone for many other data visualization libraries (like Seaborn).

Advantages: High flexibility, supports complex plots. Disadvantages: Somewhat verbose and less aesthetically appealing by default compared to other libraries like Seaborn.

0.3 2. Seaborn:

Seaborn is built on top of Matplotlib and offers a higher-level interface for drawing attractive and informative statistical graphics. It provides built-in themes and color palettes to make it easier to create complex and aesthetically pleasing plots with less code.

Advantages: Easy to use, aesthetically pleasing, statistical support. Disadvantages: Limited flexibility in comparison to Matplotlib for highly customized plots.

0.4 3. The Iris Dataset:

The Iris dataset is one of the most well-known datasets used in machine learning, especially for classification tasks. It contains 150 samples from three different species of iris flowers (Setosa, Versicolor, and Virginica), with 50 samples per species. The features of each flower are:

Sepal length Sepal width Petal length Petal width Each flower species can be identified based on these features, and the dataset is ideal for visualizations and statistical analysis.

To use the matplotlib and seaborn libraries,firstly import pyplot module and seaborn module as shown below:

[1]:
```
import seaborn as sns
```

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

0.4.1 Loading the Iris Dataset

The Iris dataset is a famous dataset in machine learning and statistics, consisting of measurements of different attributes (sepal length, sepal width, petal length, petal width) for three species of iris flowers (Setosa, Versicolor, and Virginica). We'll use the dataset from Seaborn's built-in datase

[72]:
```
# Load the Iris dataset
iris = sns.load_dataset('iris')
print(iris)
```

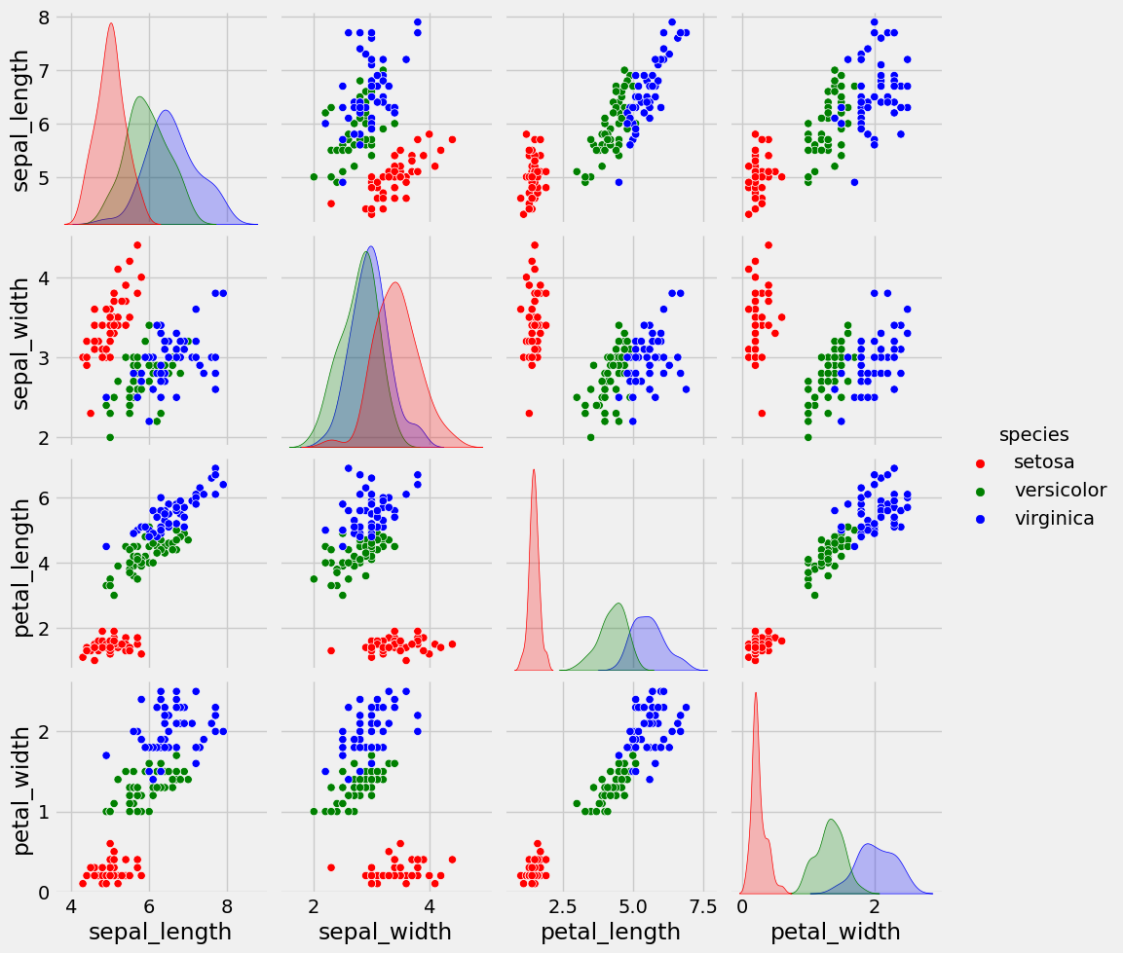
```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          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]
```

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

to create a plot that gives a general statistical summary of the Iris data.

[81]:
```
sns.pairplot(iris,hue='species',height=2.5,palette={'setosa':'red','versicolor':
  ↪'green','virginica':'blue'})
plt.show()
```

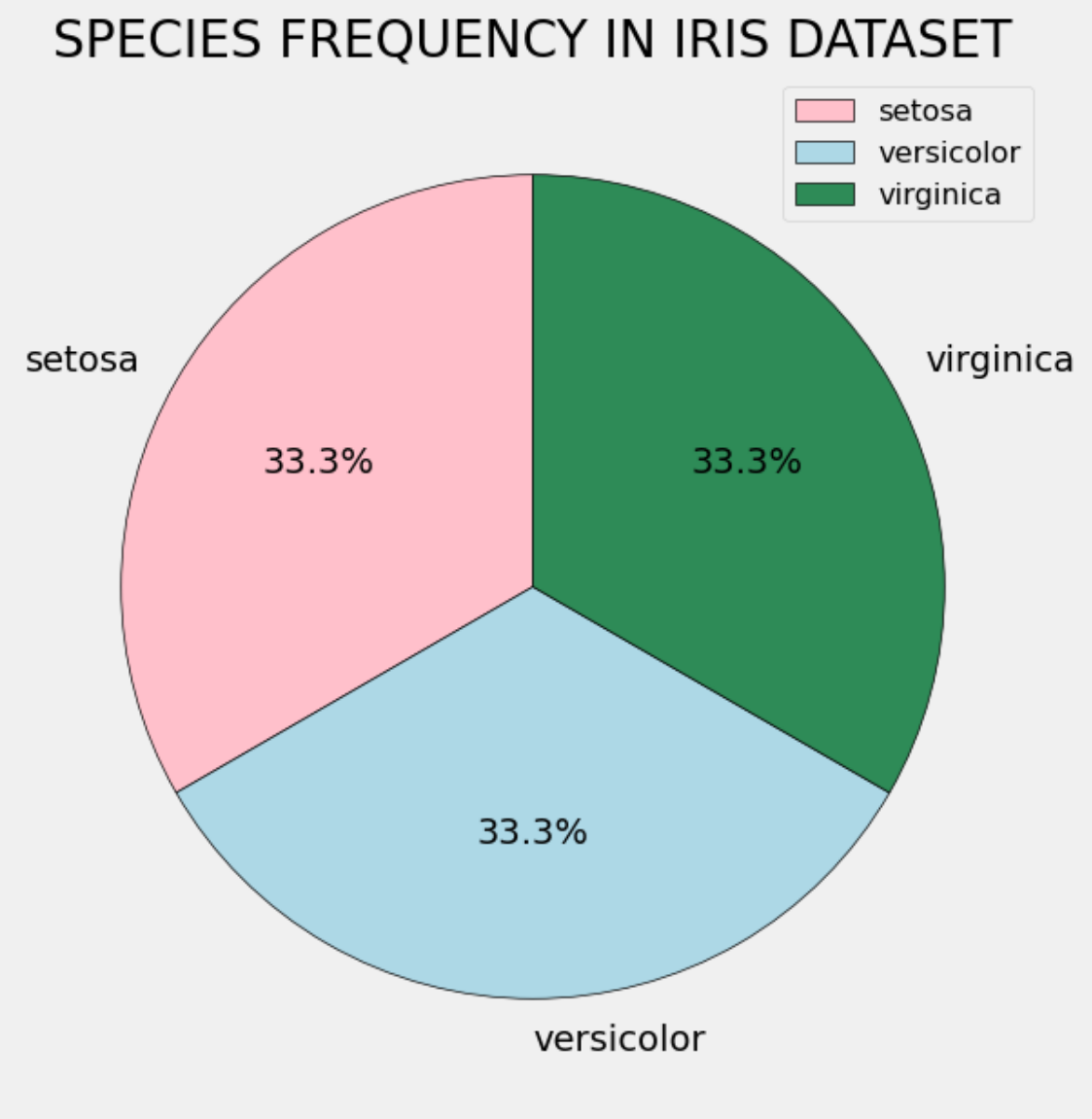
```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



0.4.3 2. Pie Plot for Species Frequency:

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

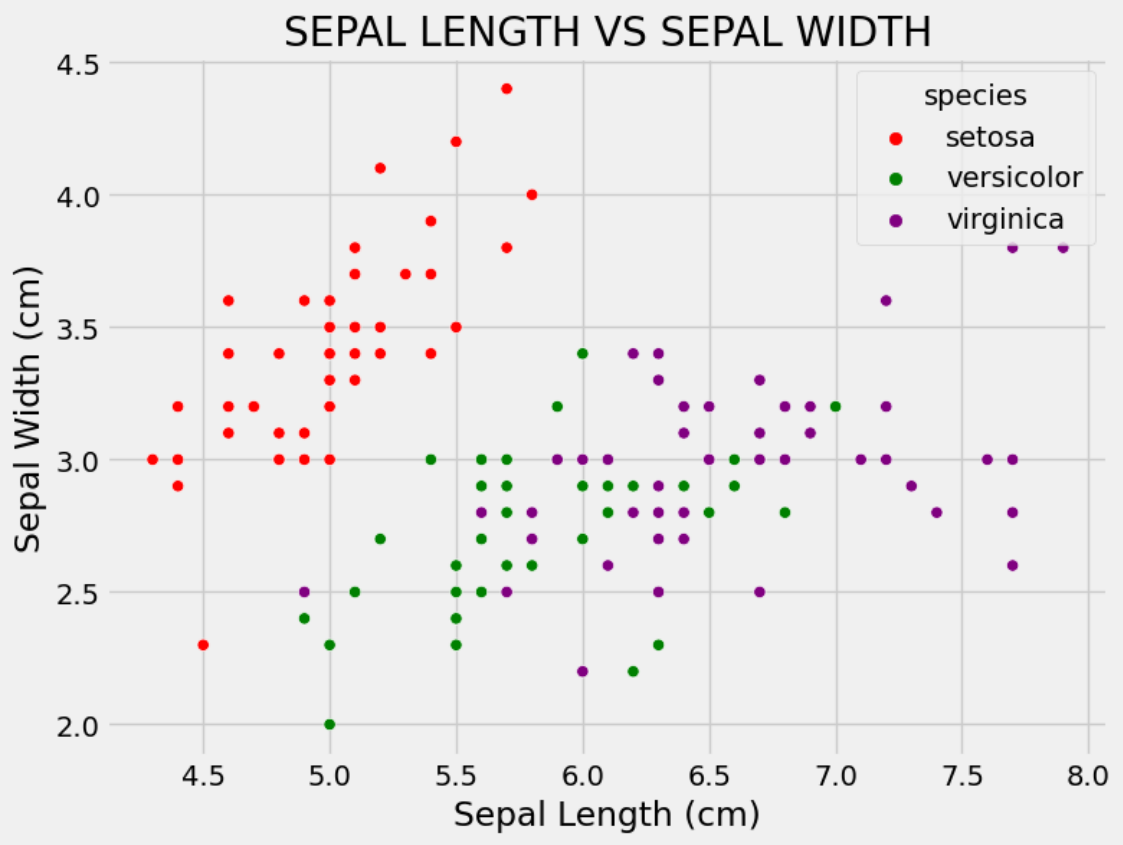
[57]:
```
species_counts=iris['species'].value_counts()
plt.figure(figsize=(7,7))
plt.tight_layout()
plt.style.use('fivethirtyeight')
plt.pie(species_counts,labels=species_counts.index,autopct='%1.
  ↪1f%%',startangle=90,wedgeprops={'edgecolor':
  ↪'black'},colors=['pink','lightblue','seagreen'])
plt.title('SPECIES FREQUENCY IN IRIS DATASET')
plt.legend(fontsize='small')
plt.show()
```



0.4.4 3. Relationship Between Sepal Length and Width:

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

[80]:
```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species',␣
↪data=iris,palette={'setosa':'red','versicolor':'green','virginica':'purple'})
plt.title('SEPAL LENGTH VS SEPAL WIDTH')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
plt.show()
```

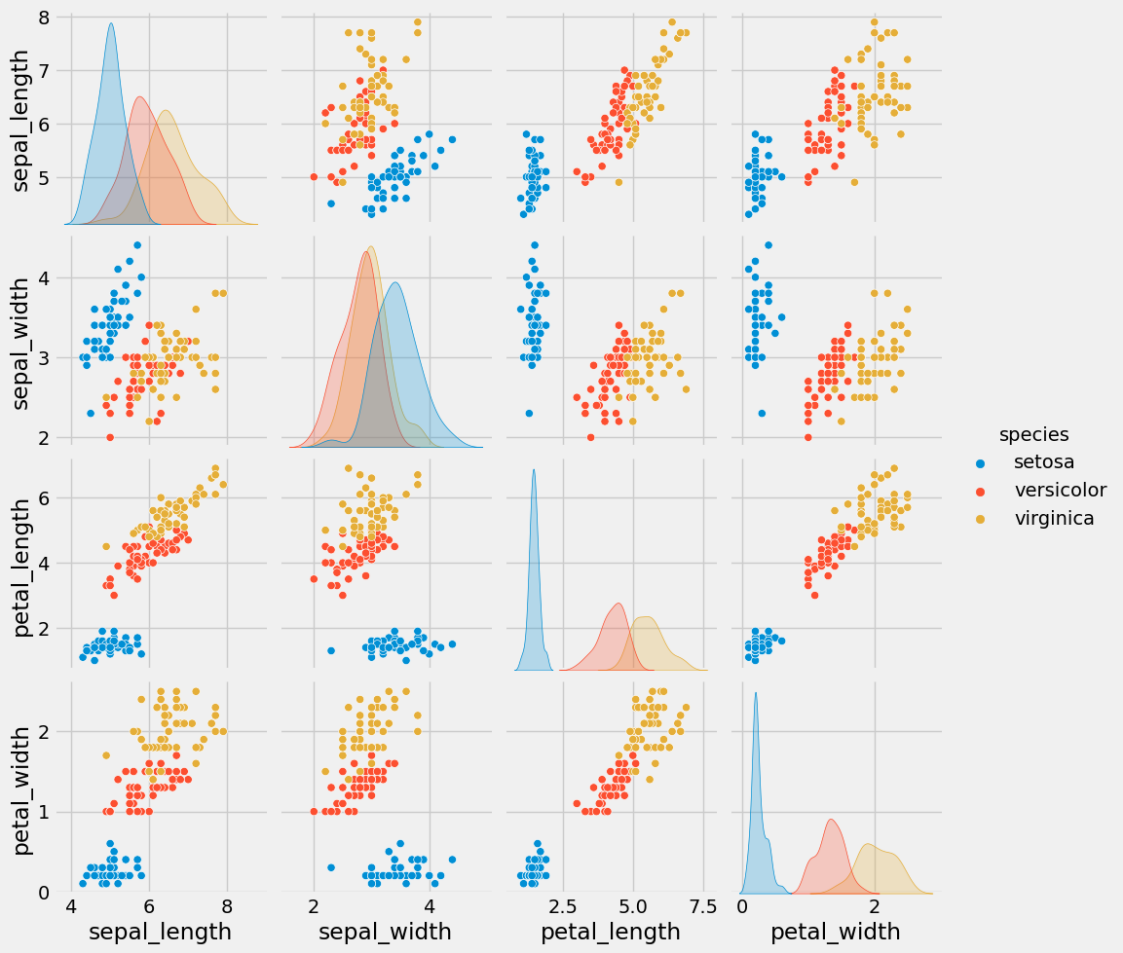


0.4.5 4. Distribution of Sepal and Petal Features:

To create a plot that shows how the length and width of sepal length, sepal width, petal length, and petal width are distributed.

[71]:
```
sns.pairplot(iris, hue='species', height=2.5)

plt.show()
```

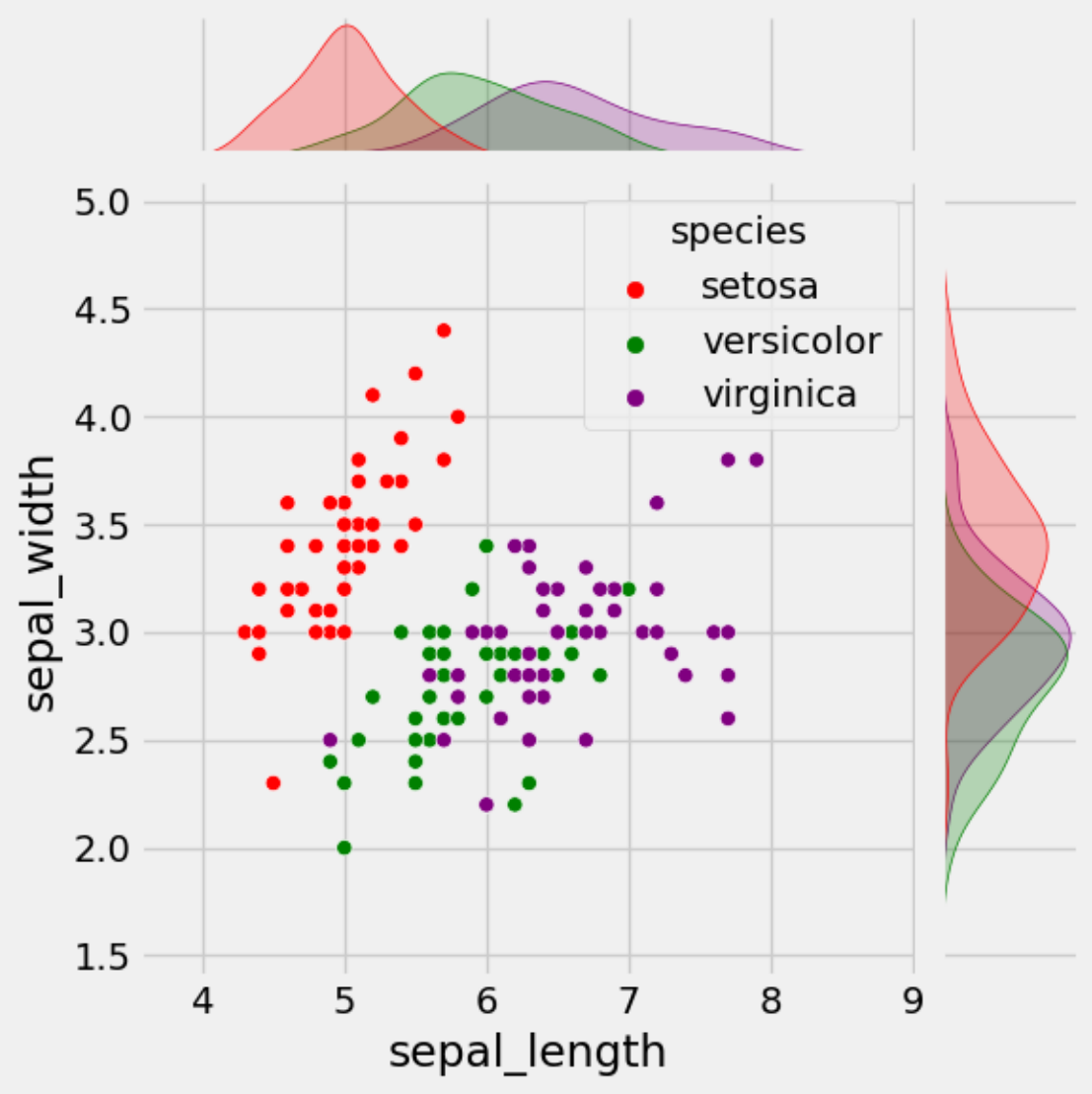
```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



0.4.6 5.Jointplot of Sepal Length vs Sepal Width:

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

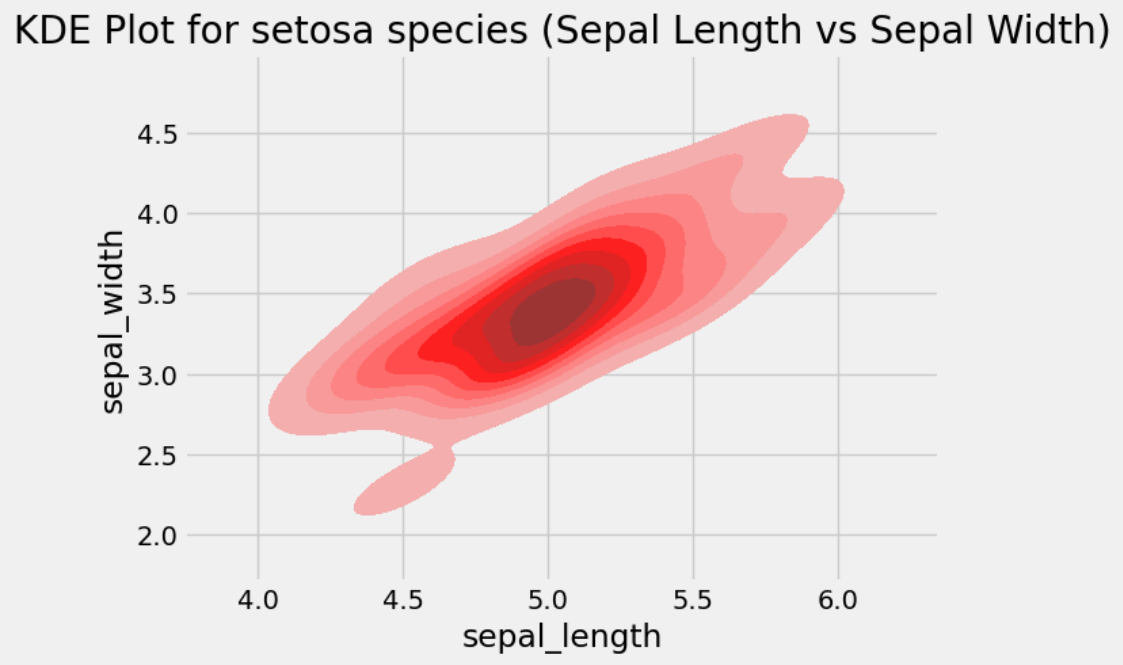
[79]:
```
sns.jointplot(x='sepal_length', y='sepal_width', data=iris,␣
 ↪kind='scatter',hue='species',palette={'setosa':'red','versicolor':
 ↪'green','virginica':'purple'})
plt.show()
```



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

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

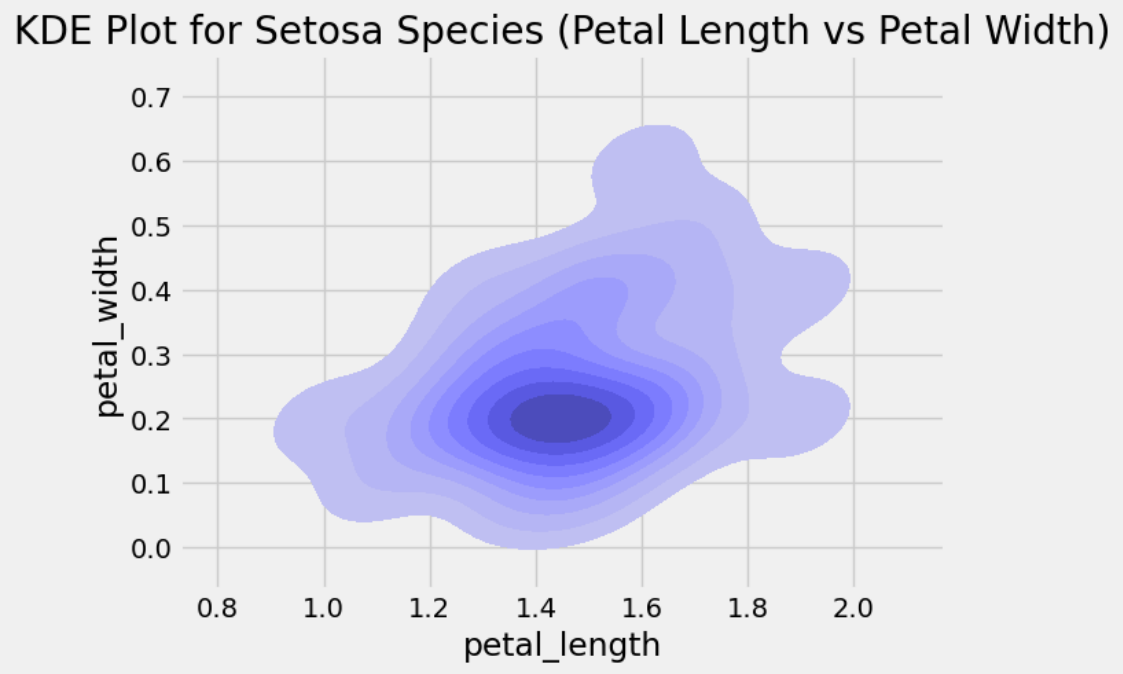
[73]:
```
setosa = iris[iris['species'] == 'setosa']
sns.kdeplot(x='sepal_length', y='sepal_width', data=setosa,␣
 ↪fill=True,color='red')
plt.title('KDE Plot for setosa species (Sepal Length vs Sepal Width)')
plt.show()
```



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

To create a KDE plot of petal length versus petal width for the setosa species.

[77]:
```
sns.kdeplot(x='petal_length', y='petal_width',␣
  ↪data=setosa,fill=True,color='blue')
plt.title('KDE Plot for Setosa Species (Petal Length vs Petal Width)')
plt.show()
```


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

[ ]: