data-visualsation-1

September 17, 2024

DATA VISUALTIZATION USING MATPLOTLIB AND SEABOEN PACKAGES IN PYTHON # MATPLOTLIB and SEABORN

# Matplotlib

- Purpose: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
- Features: It provides a wide range of plotting functions, including line plots, scatter plots, bar charts, histograms, and more. It's highly customizable, allowing you to tweak almost every aspect of your plots.
- Usage: Ideal for creating basic plots and for users who need extensive customization options. # Seaborn
- Purpose: Seaborn is built on top of Matplotlib and is designed for making statistical graphics.
- Features: It offers a high-level interface for drawing attractive and informative statistical graphics. Seaborn integrates closely with Pandas data structures and provides functions to visualize univariate and bivariate distributions, categorical data, and more.
- Usage: Great for creating more complex visualizations with less code. It also comes with built-in themes and color palettes to make your plots look more aesthetically pleasing.

1 1. General Statistics Plot (Matplotlib AND 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.

[ ]:

```python
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
```

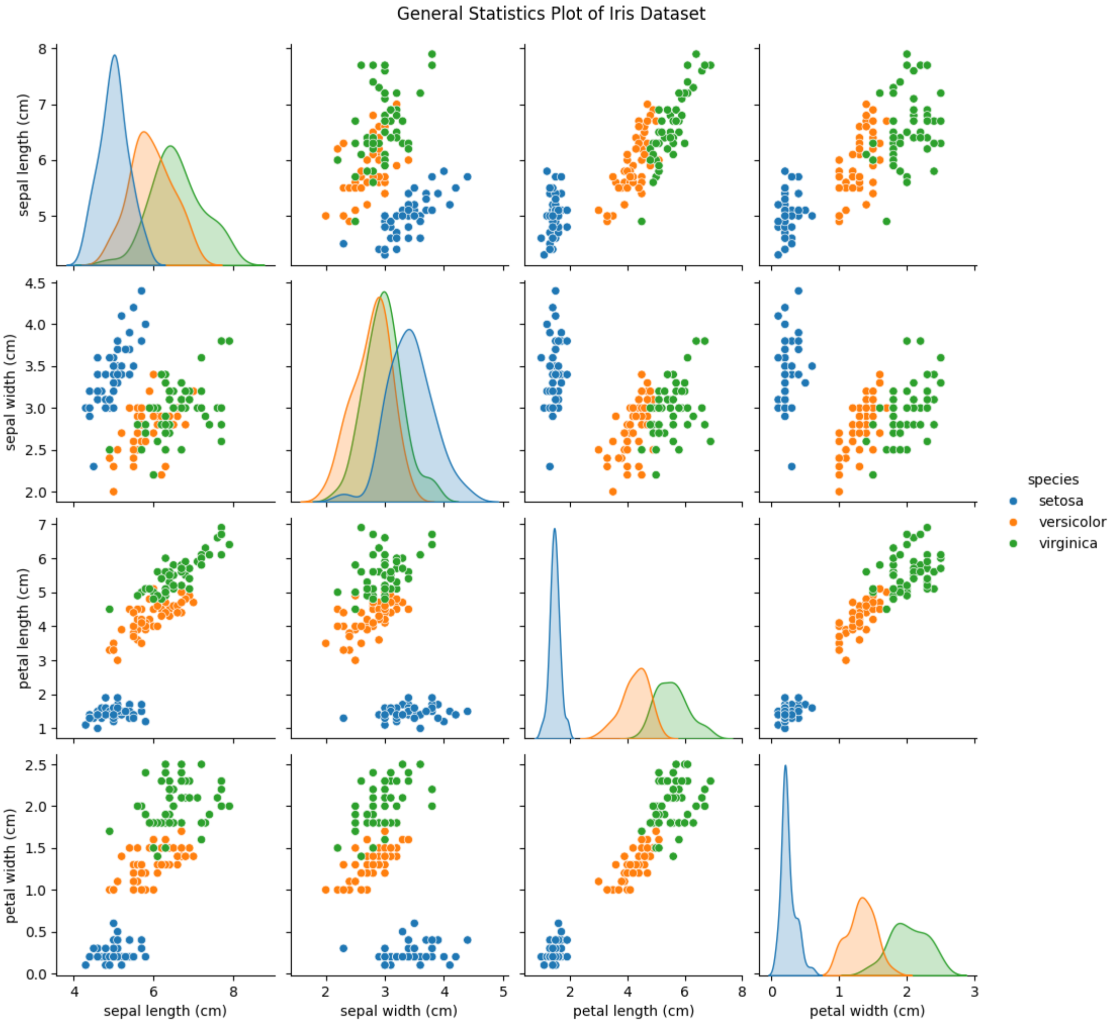
[ ]:

```python
# Load the Iris dataset
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['species'] = pd.Categorical.from_codes(iris.target, iris.target_names)
```

[ ]:

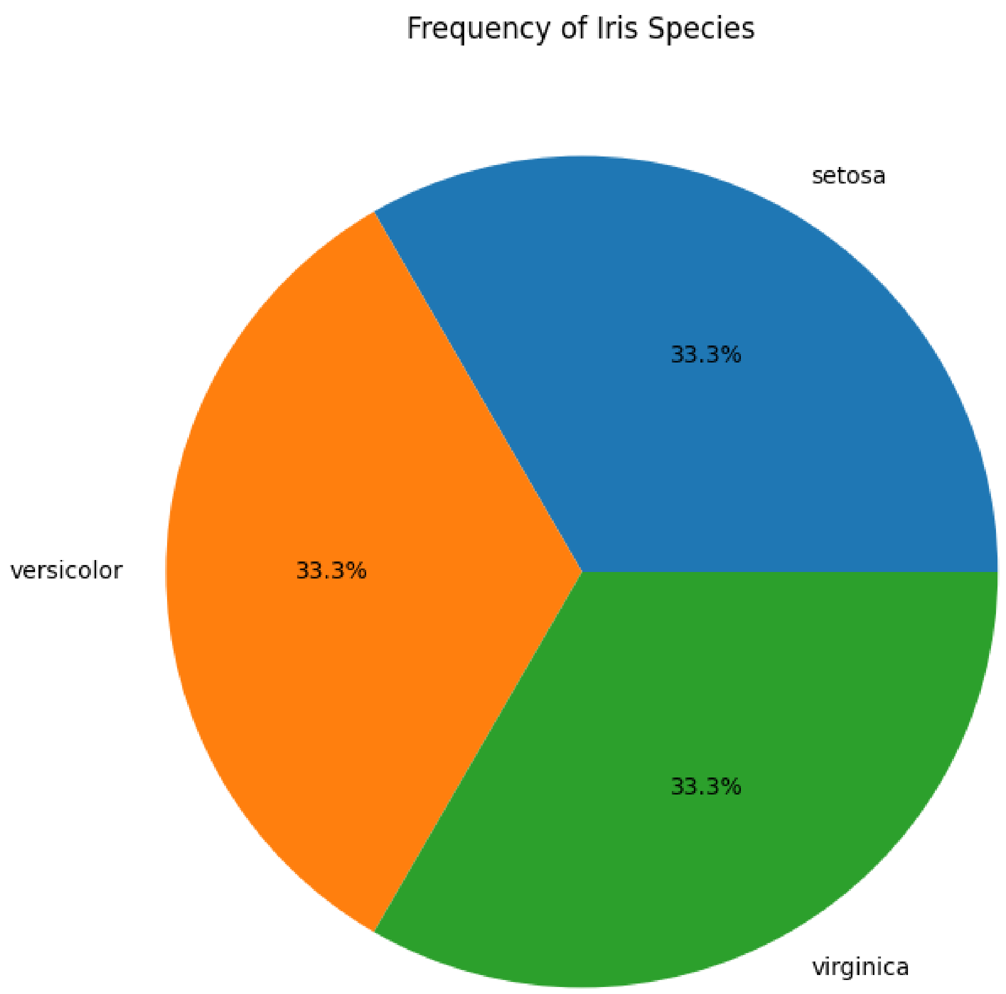
```python
# 1. General Statistics Plot
print("Exercise 1: General Statistics Plot")
sns.pairplot(df, hue='species')
plt.suptitle("General Statistics Plot of Iris Dataset", y=1.02)
plt.show()
```

```
Exercise 1: General Statistics Plot
```



```
[ ]: # 2. Pie Plot for Species Frequency
     print("\nExercise 2: Pie Plot for Species Frequency")
     species_counts = df['species'].value_counts()
     plt.figure(figsize=(8, 8))
     plt.pie(species_counts.values, labels=species_counts.index, autopct='%1.1f%%')
     plt.title("Frequency of Iris Species")
     plt.show()
```

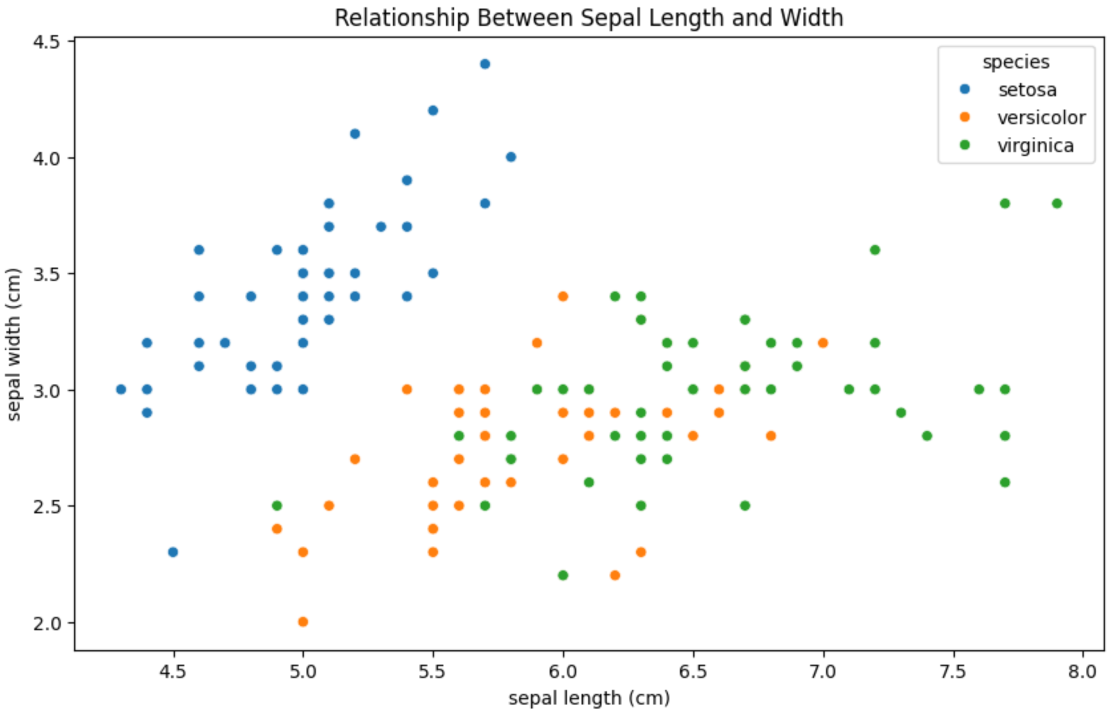
```
Exercise 2: Pie Plot for Species Frequency
```



[ ]:

```
# 3. Relationship Between Sepal Length and Width
print("\nExercise 3: Relationship Between Sepal Length and Width")
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='sepal length (cm)', y='sepal width (cm)',␣
 ↪hue='species')
plt.title("Relationship Between Sepal Length and Width")
plt.show()
```

```
Exercise 3: Relationship Between Sepal Length and Width
```

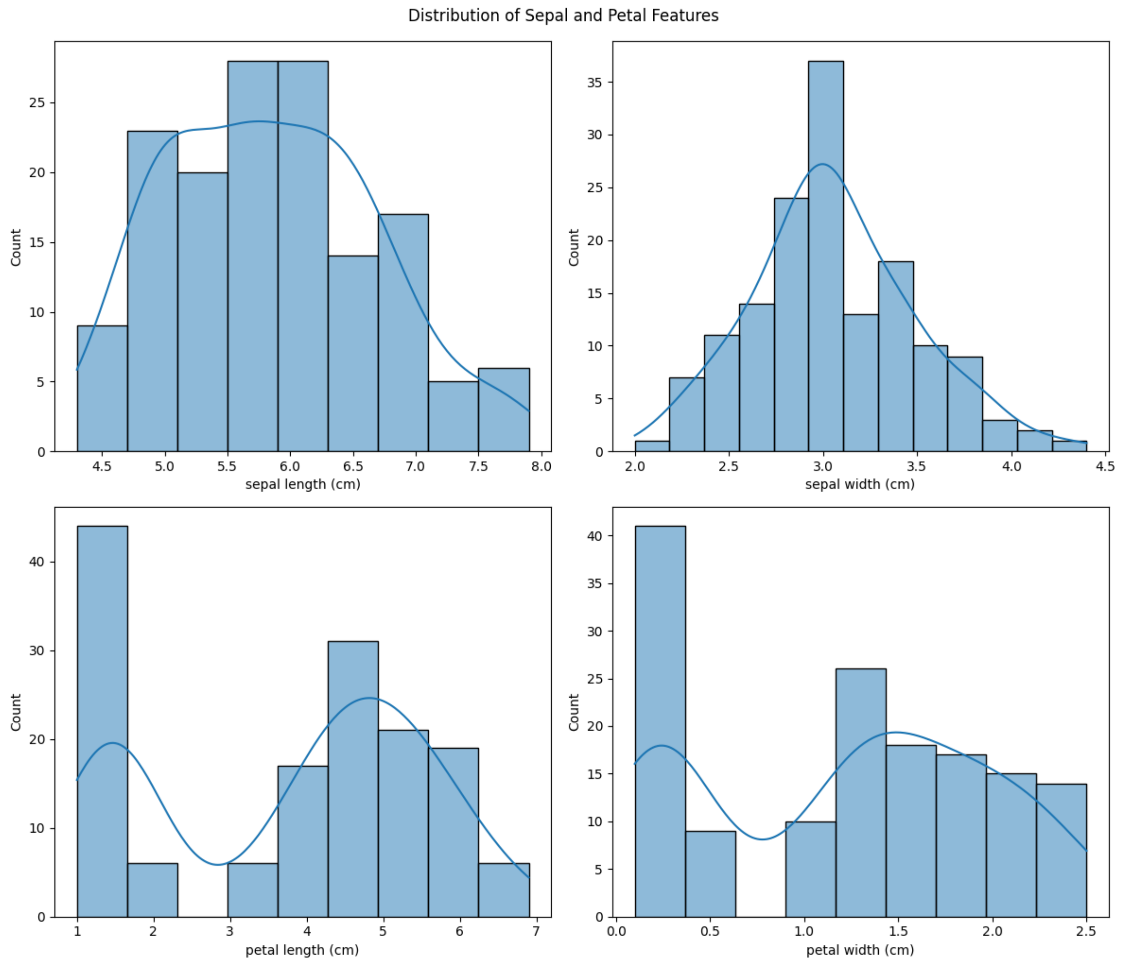


8TH WEEK

[ ]:

```python
# 1. Distribution of Sepal and Petal Features
print("\nExercise 1: Distribution of Sepal and Petal Features")
fig, axes = plt.subplots(2, 2, figsize=(12, 10))
sns.histplot(data=df, x='sepal length (cm)', kde=True, ax=axes[0, 0])
sns.histplot(data=df, x='sepal width (cm)', kde=True, ax=axes[0, 1])
sns.histplot(data=df, x='petal length (cm)', kde=True, ax=axes[1, 0])
sns.histplot(data=df, x='petal width (cm)', kde=True, ax=axes[1, 1])
plt.tight_layout()
plt.suptitle("Distribution of Sepal and Petal Features", y=1.02)
plt.show()
```

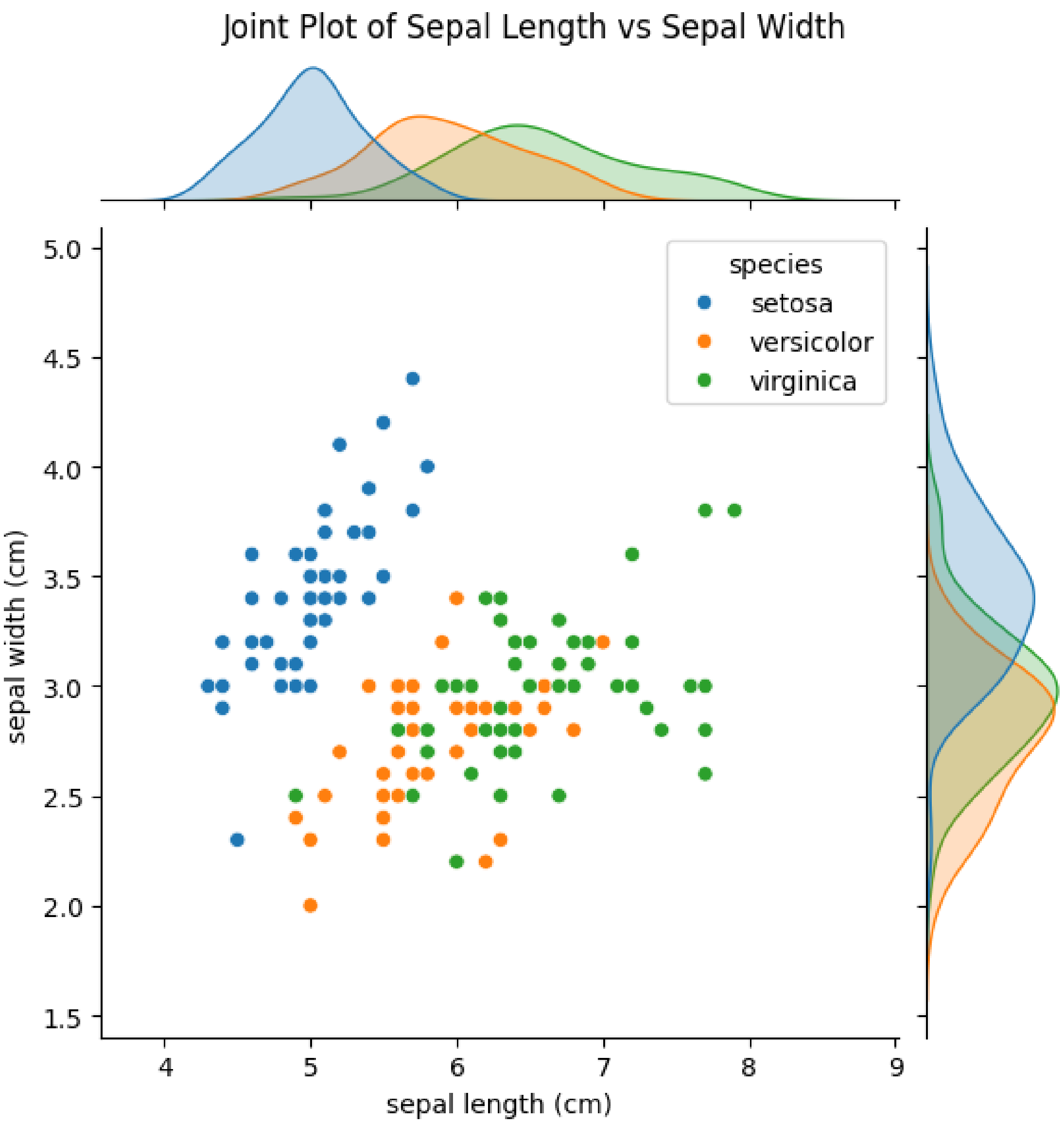
```
Exercise 1: Distribution of Sepal and Petal Features
```



[ ]:

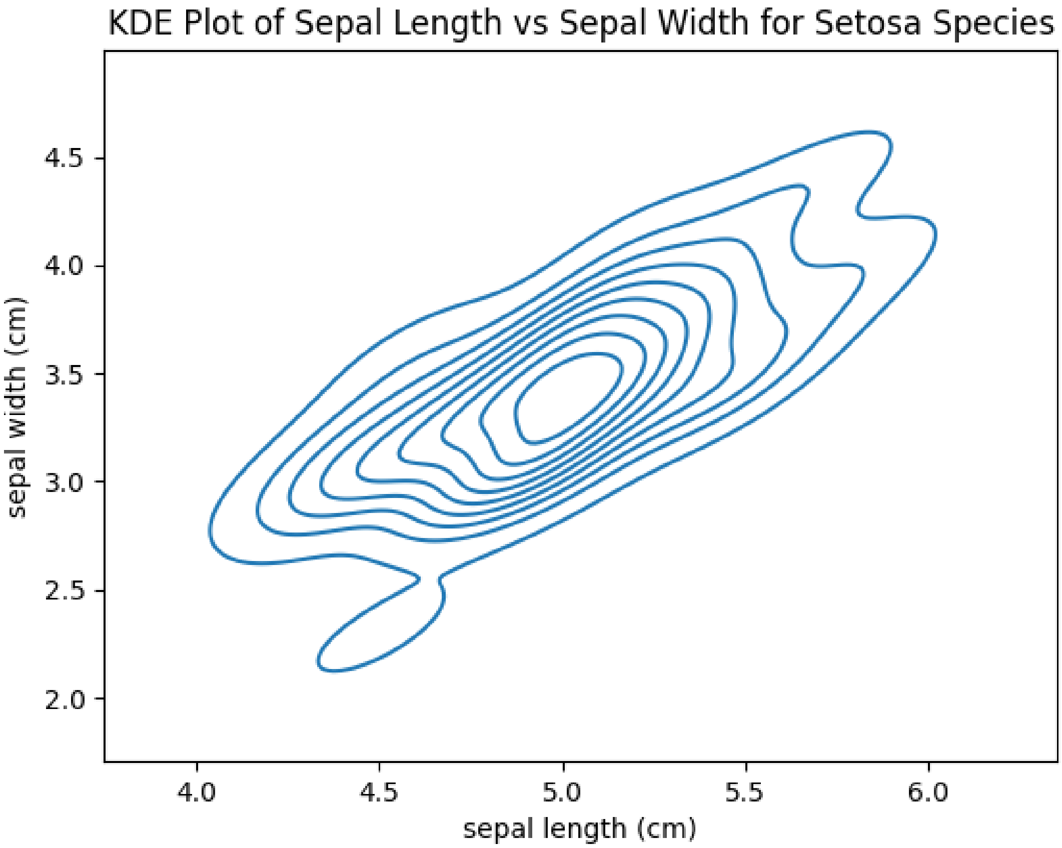
```
# 2. Jointplot of Sepal Length vs Sepal Width
print("\nExercise 2: Jointplot of Sepal Length vs Sepal Width")
sns.jointplot(data=df, x='sepal length (cm)', y='sepal width (cm)',␣
  ↪hue='species')
plt.suptitle("Joint Plot of Sepal Length vs Sepal Width", y=1.02)
plt.show()
```

```
Exercise 2: Jointplot of Sepal Length vs Sepal Width
```



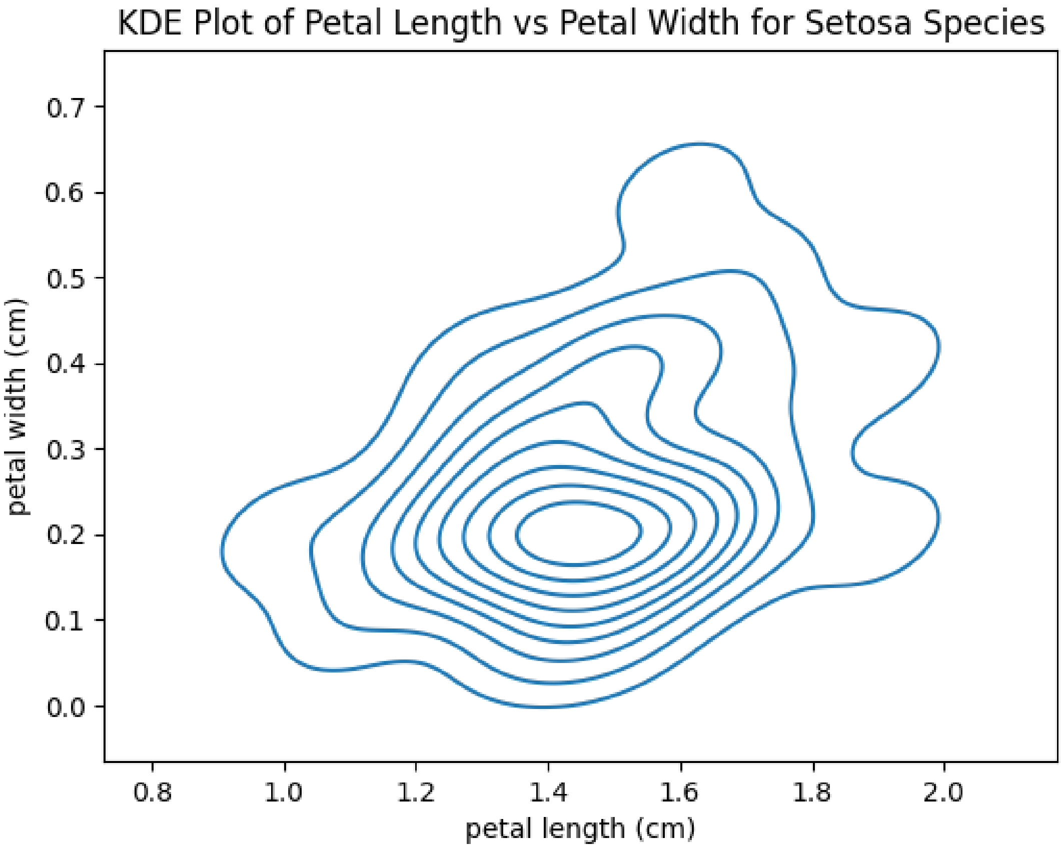
```
[ ]: # 3. KDE Plot for Setosa Species (Sepal Length vs Sepal Width)
     print("\nExercise 3: KDE Plot for Setosa Species (Sepal Length vs Sepal Width)")
     setosa_df = df[df['species'] == 'setosa']
     sns.kdeplot(data=setosa_df, x='sepal length (cm)', y='sepal width (cm)')
     plt.title("KDE Plot of Sepal Length vs Sepal Width for Setosa Species")
     plt.show()
```

```
Exercise 3: KDE Plot for Setosa Species (Sepal Length vs Sepal Width)
```



```
[ ]: # 4. KDE Plot for Setosa Species (Petal Length vs Petal Width)
     print("\nExercise 4: KDE Plot for Setosa Species (Petal Length vs Petal Width)")
     sns.kdeplot(data=setosa_df, x='petal length (cm)', y='petal width (cm)')
     plt.title("KDE Plot of Petal Length vs Petal Width for Setosa Species")
     plt.show()
```

```
Exercise 4: KDE Plot for Setosa Species (Petal Length vs Petal Width)
```


KDE Plot of Petal Length vs Petal Width for Setosa Species
petal width (cm)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.8
1.0
1.2
1.4
1.6
1.8
2.0
petal length (cm)

[ ]: