Load the necessary libraries:

1. 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.

Common Plot Types in Matplotlib:

- Line plots for continuous data.
- Scatter plots for visualizing the relationship between two numerical variables.
- Histograms for understanding the distribution of data.

2. 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.
- Plot Types in Seaborn:
- Pair plots to show pairwise relationships in a dataset.

In [1]:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
iris = sns.load_dataset('iris')
```

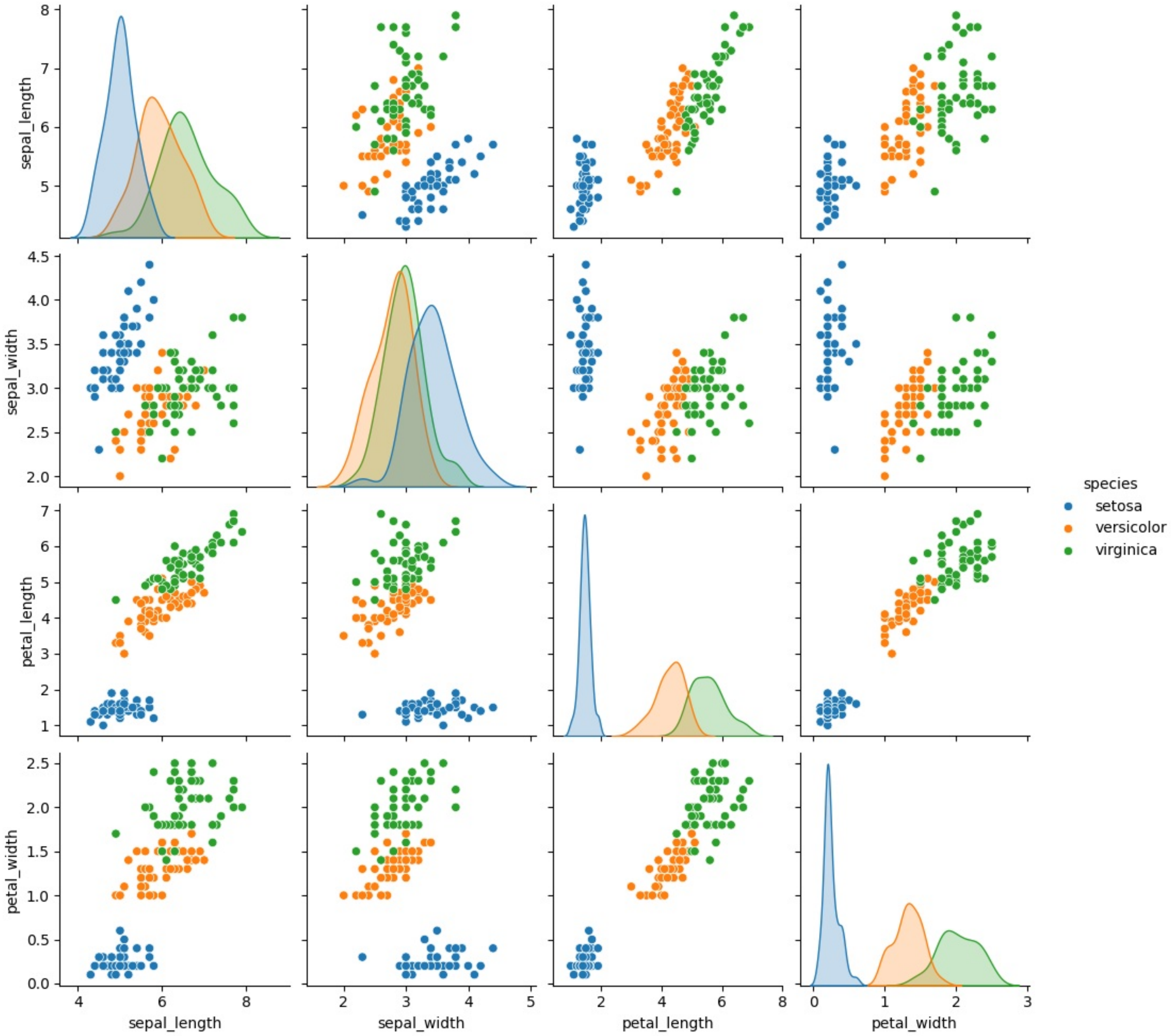
1. General Statistics Plot (Matplotlib or Seaborn)

- sns.pairplot(iris, hue='species', height=2.5): Creates a pair plot of all feature pairs; points are colored by species.
- plt.show(): Displays the plot.

This plot is useful for understanding the pairwise relationships and distributions of features in the dataset. It helps in spotting patterns, correlations, and potential clusters between features.

In [3]:

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



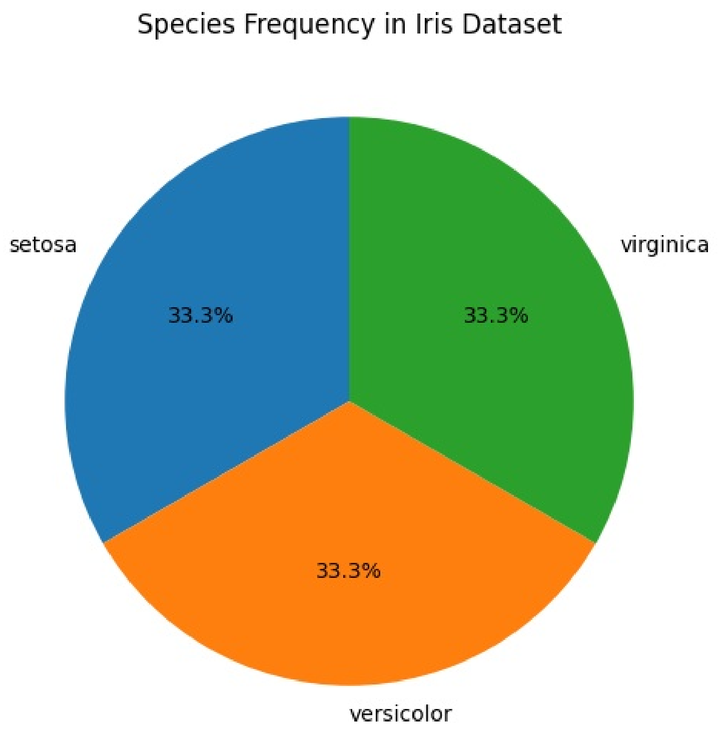
2. Pie Plot for Species Frequency

- species_counts = iris['species'].value_counts(): Counts occurrences of each species.
- plt.figure(figsize=(6,6)): Sets figure size to 6x6 inches.
- plt.pie(...): Creates a pie chart of species frequencies with percentages.
- plt.title(...): Adds a title to the pie chart.
- plt.show(): Displays the pie chart.

A pie chart is ideal for visualizing the proportion of each category (species) in the dataset, providing an immediate sense of the distribution.

In [4]:

```
species_counts=iris['species'].value_counts()
plt.figure(figsize=(6,6))
plt.pie(species_counts,labels=species_counts.index, autopct='%1.1f%%', startangle=90)
plt.title('Species Frequency in Iris Dataset')
plt.show()
```



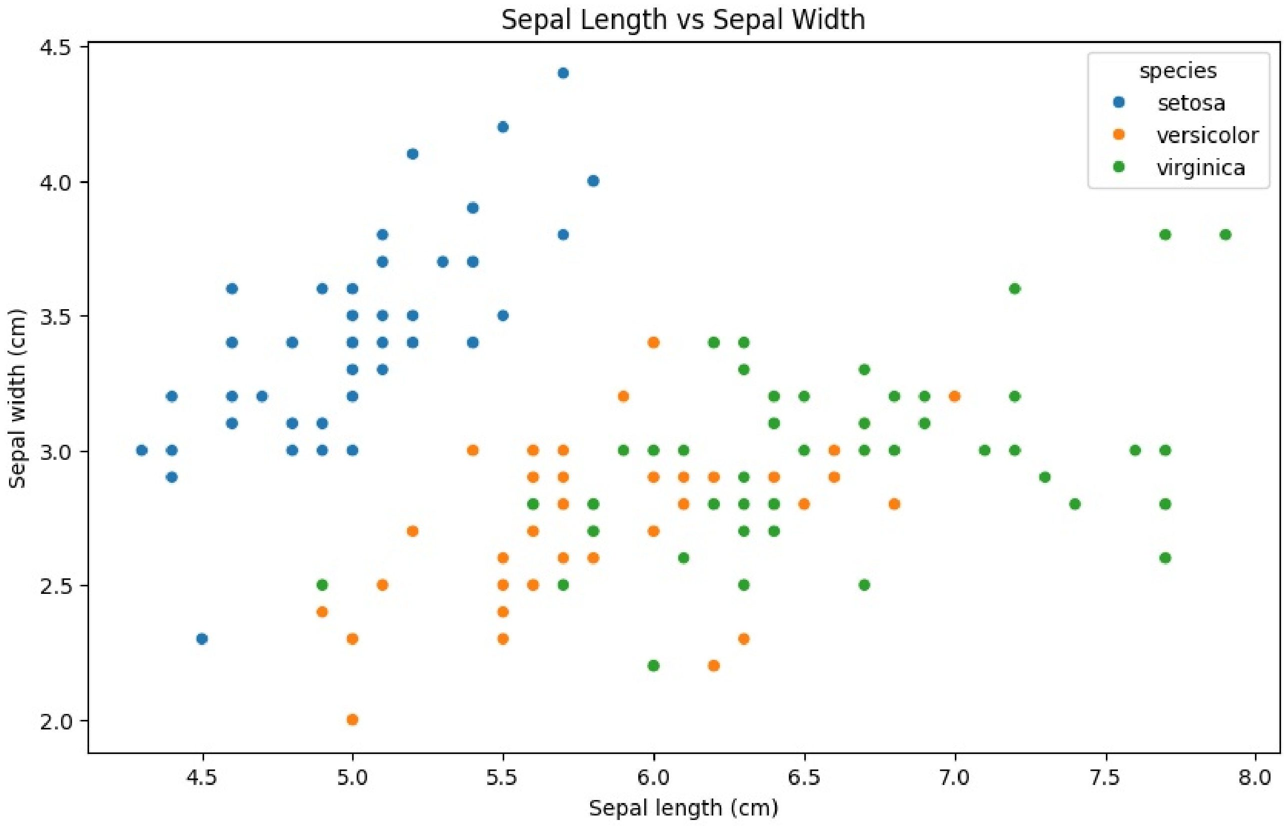
3. Relationship Between Sepal Length and Sepal Width

- sns.scatterplot(...): Creates a scatter plot of sepal length vs. sepal width with points colored by species.

A scatter plot is suitable for examining the relationship between two continuous variables (sepal length and sepal width) and identifying any patterns or clusters based on species.

In [5]:

```
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

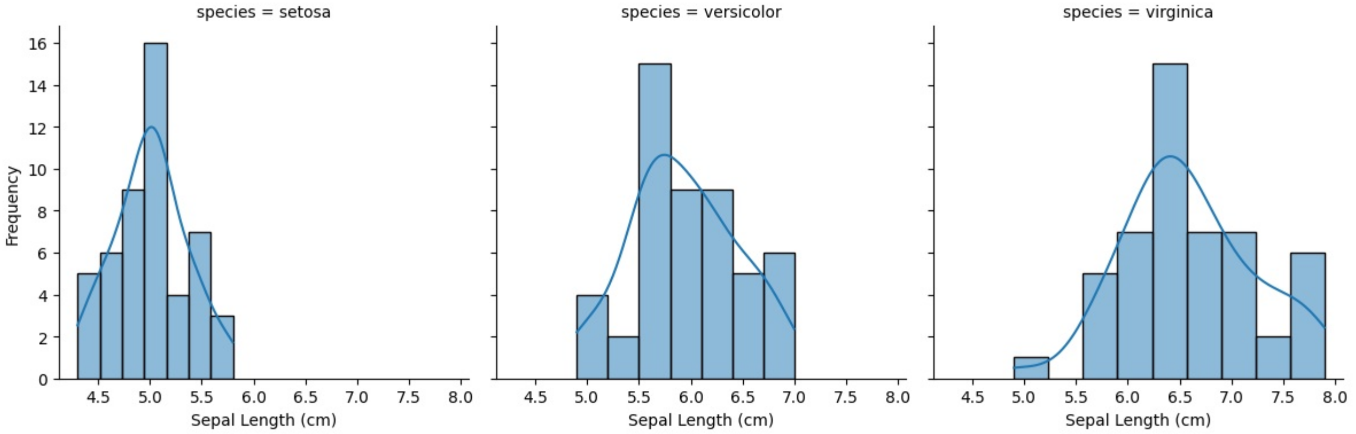
- FacetGrid: Creates a grid of subplots by species.

- map(sns.histplot, 'sepal_length', kde=True): Plots histograms with KDE for sepal_length on each subplot.
- set_axis_labels('Sepal Length (cm)', 'Frequency'): Labels x-axis as "Sepal Length (cm)" and y-axis as "Frequency".
- plt.show(): Displays the plot.

This visualization uses FacetGrid to create separate histograms for sepal_length across different species, allowing for easy comparison. The histograms include KDE curves to show the distribution shape. This approach clearly illustrates how the distribution of sepal length varies among species, revealing both the frequency and density of measurements.

In [6]:

```
sns.FacetGrid(iris, col='species', height=4).map(sns.histplot, 'sepal_length', kde=True).set_axis_labels('Sepal
plt.show()
```



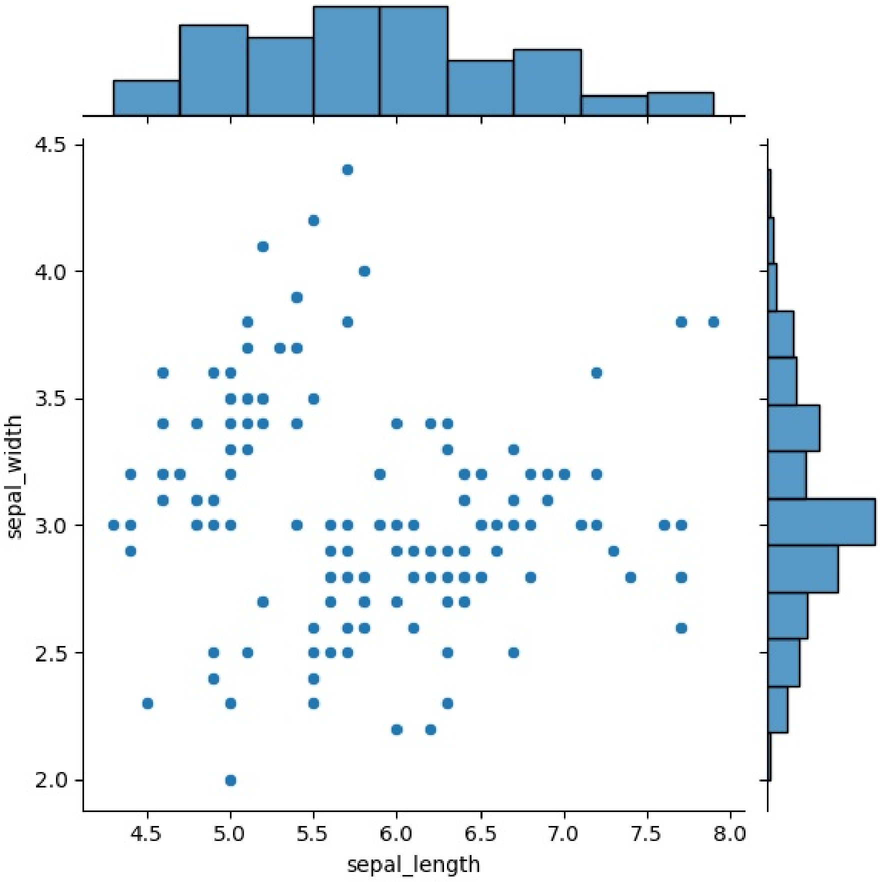
5. Jointplot of Sepal Length vs Sepal Width

- sns.jointplot(...): Plots sepal length vs. sepal width with scatter plot and marginal histograms.
- plt.show(): Displays the joint plot.

Visualization provides a detailed view of the relationship between two variables and their individual distributions, useful for understanding the nature of their relationship and the distribution of values.

In [7]:

```
sns.jointplot(x='sepal_length',y='sepal_width', data=iris, kind='scatter')
plt.show()
```



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

- setosa = iris[iris['species'] == 'setosa']: Filters data for the 'setosa' species.

- sns.kdeplot(...): Plots KDE of sepal length vs. sepal width for 'setosa'.
- plt.title(...): Adds a title to the KDE plot.
- plt.show(): Displays the KDE plot.

KDE plots are useful for understanding the density and distribution of data points within the feature space, highlighting areas with higher concentration of data for a specific species.

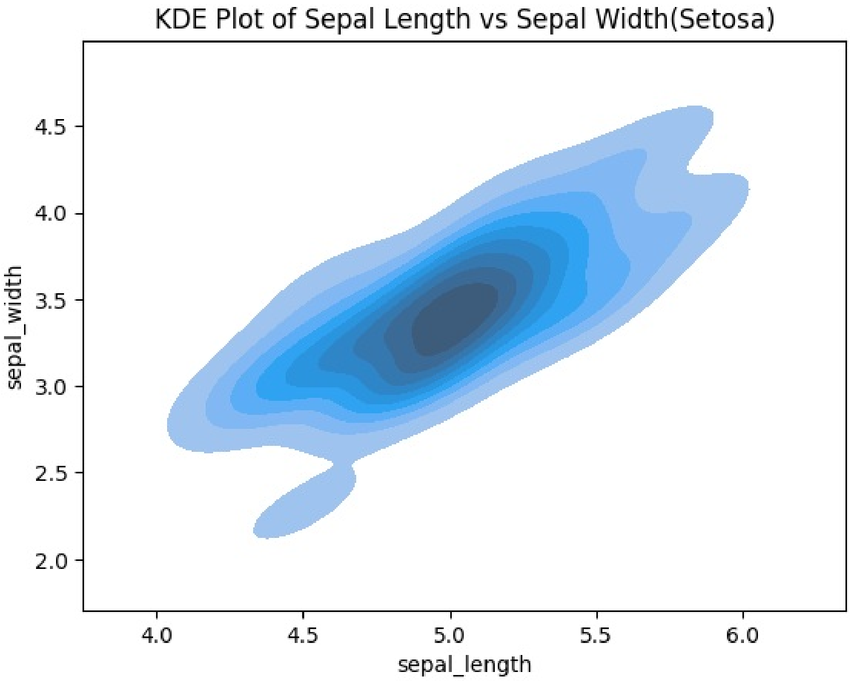
In [8]:

```
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()
```

```
C:\Users\pedum\AppData\Local\Temp\ipykernel_19916\1005398603.py: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. KDE Plot for Setosa Species (Petal Length vs Petal Width)

- sns.kdeplot(...): Plots KDE of petal length vs. petal width for 'setosa'.
- plt.title(...): Adds a title to the KDE plot.
- plt.show(): Displays the KDE plot.

This KDE plot helps visualize the density and distribution of petal features, providing insights into the feature relationships for the 'setosa' species.

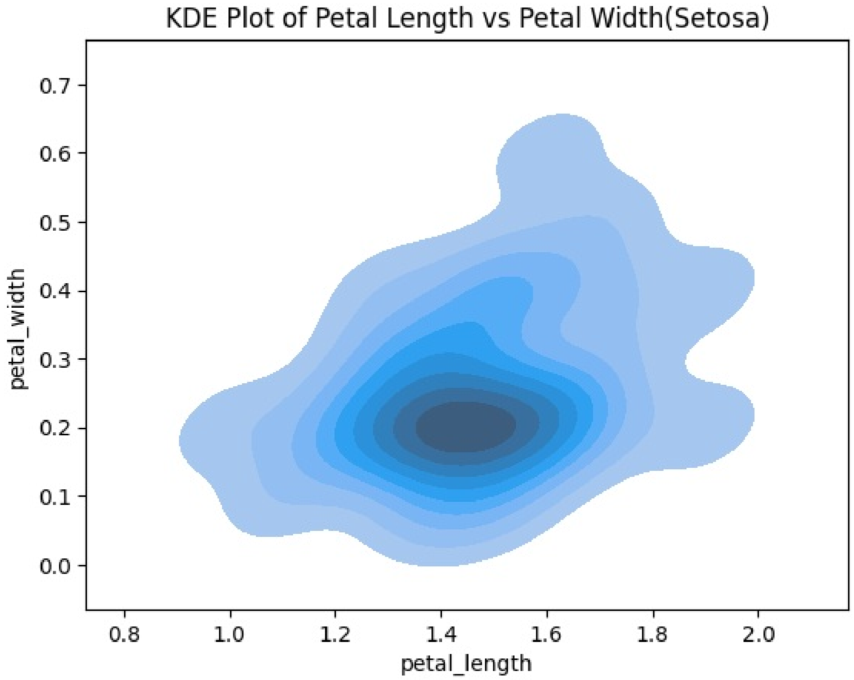
In [9]:

```
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()
```

```
C:\Users\pedum\AppData\Local\Temp\ipykernel_19916\2819947582.py: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)
```



In [ ]:

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js