Program 5: perform exploratory data analysis using Seaborn and Matplotlib libraries. Create differrent types of plots and visualizations to understand data distribution and relationships. use a publicly available dataset

Load the IRIS dataset and print the head of the dataset

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

# Load the Iris dataset
iris = sns.load_dataset("iris")

# Display the first few rows of the dataset
print(iris.head(100))
```

```
     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
..            ...          ...           ...          ...         ...
95            5.7          3.0           4.2          1.2  versicolor
96            5.7          2.9           4.2          1.3  versicolor
97            6.2          2.9           4.3          1.3  versicolor
98            5.1          2.5           3.0          1.1  versicolor
99            5.7          2.8           4.1          1.3  versicolor

[100 rows x 5 columns]
```
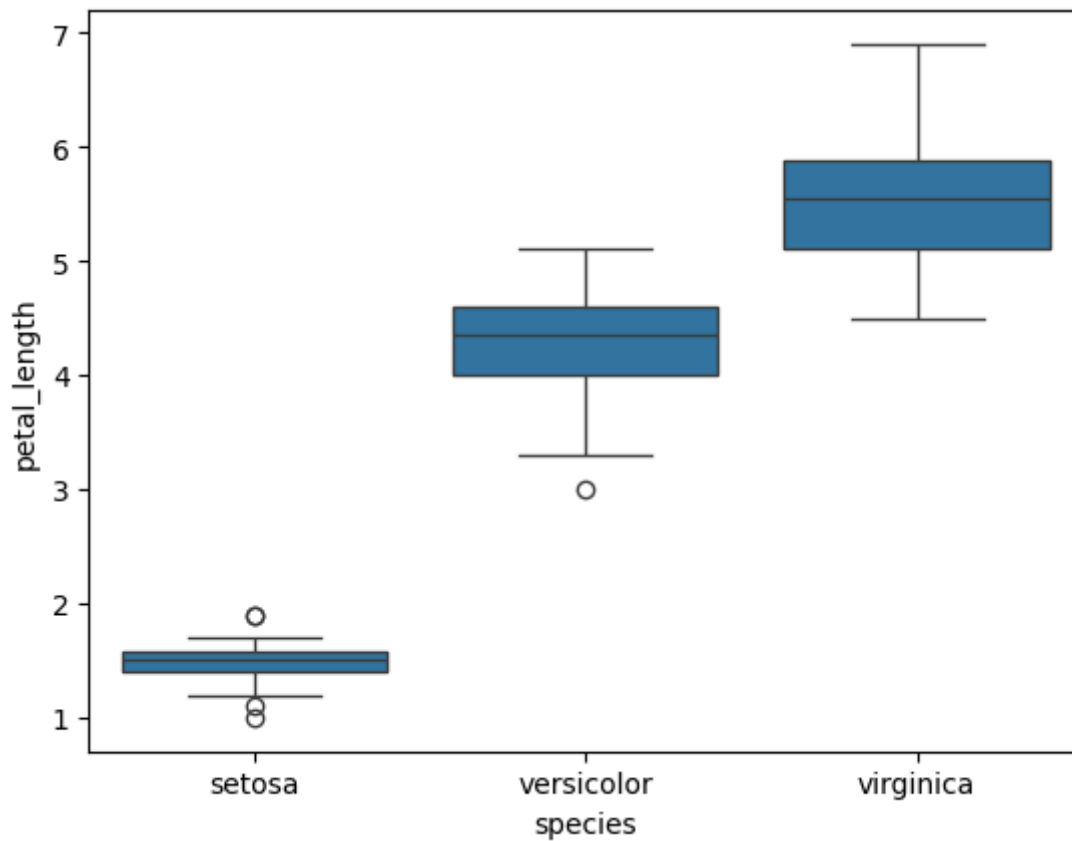
Pairplot: A pairplot allows us to visualize pairwise relationships in the dataset

```
sns.pairplot(iris, hue="species")
plt.show()
```
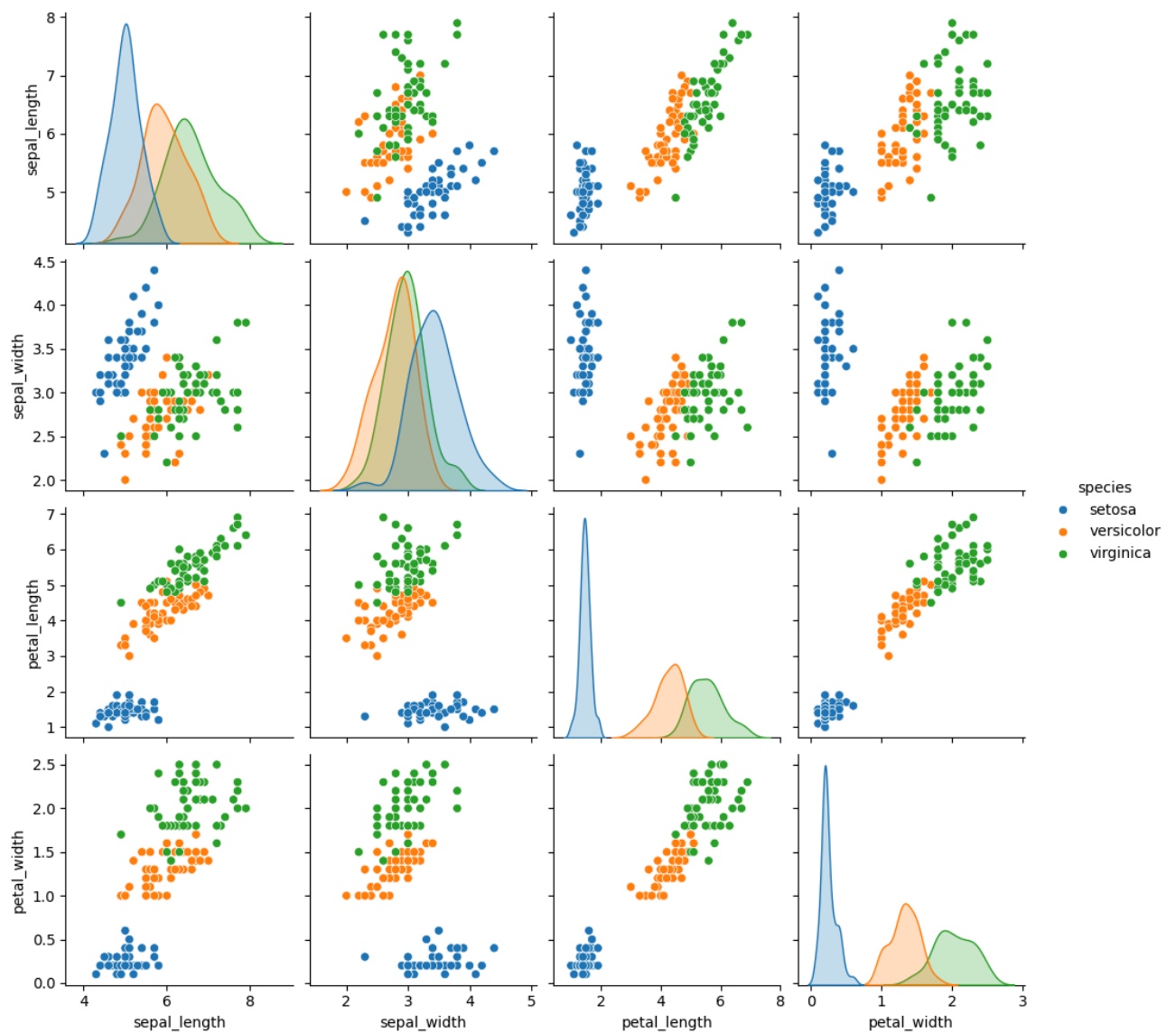
Boxplot: Boxplots can be used to visualize the distribution of a single continuous variable or the distribution of a continuous variable within different categories.

```
sns.boxplot(x="species", y="petal_length", data=iris)
plt.show()
```


Histogram: Histograms are useful for visualizing the distribution of a single continuous variable.

```
sns.histplot(iris["petal_length"], bins=20, kde=True)
plt.xlabel("Petal Length")
plt.ylabel("Frequency")
plt.title("Distribution of Petal Length")
plt.show()
```

Distribution of Petal Length
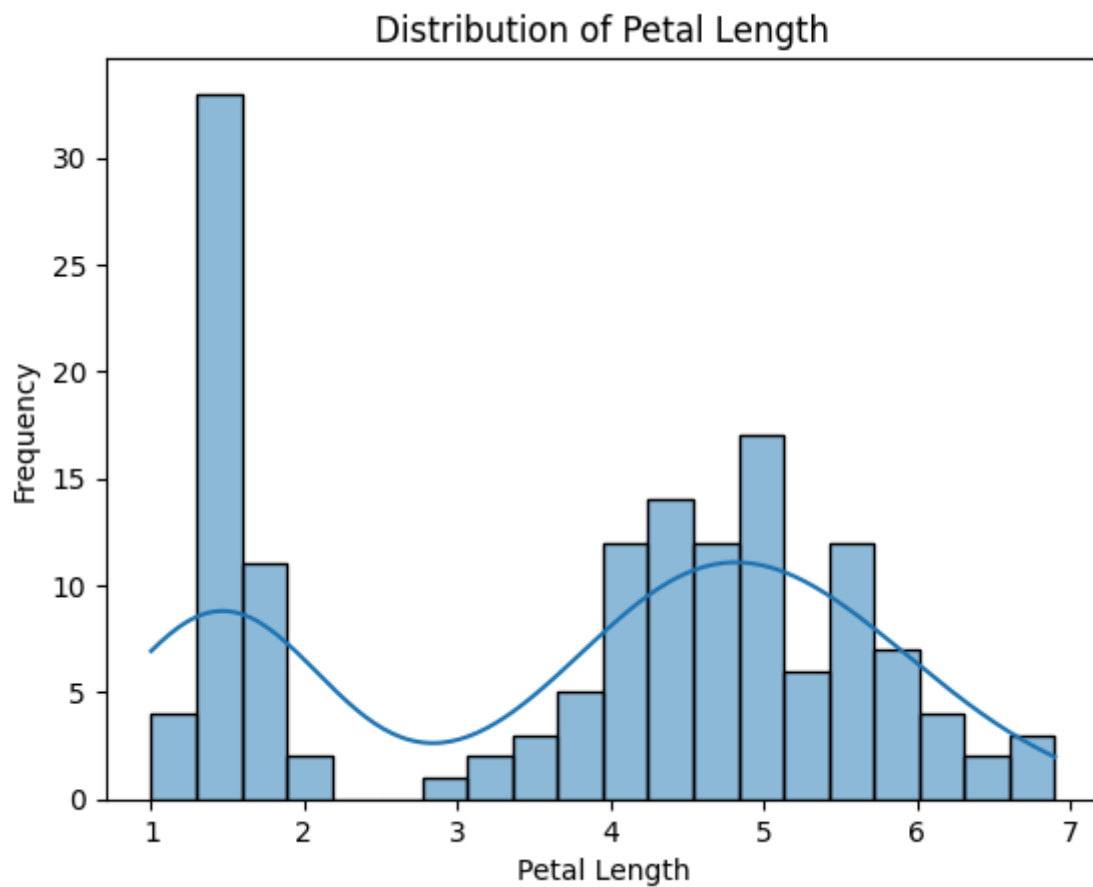
Violin plot: Violin plots are similar to boxplots but also display the probability density of the data at different values.
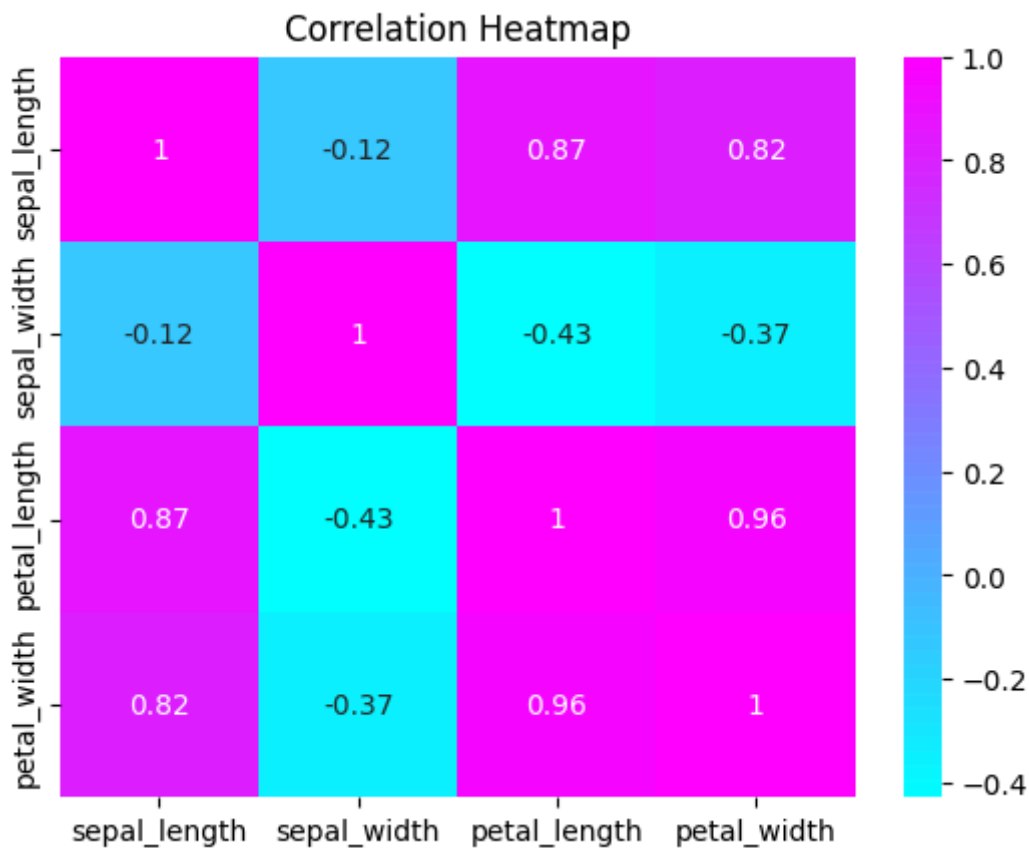
```
sns.violinplot(x="species", y="petal_width", data=iris)
plt.show()
```
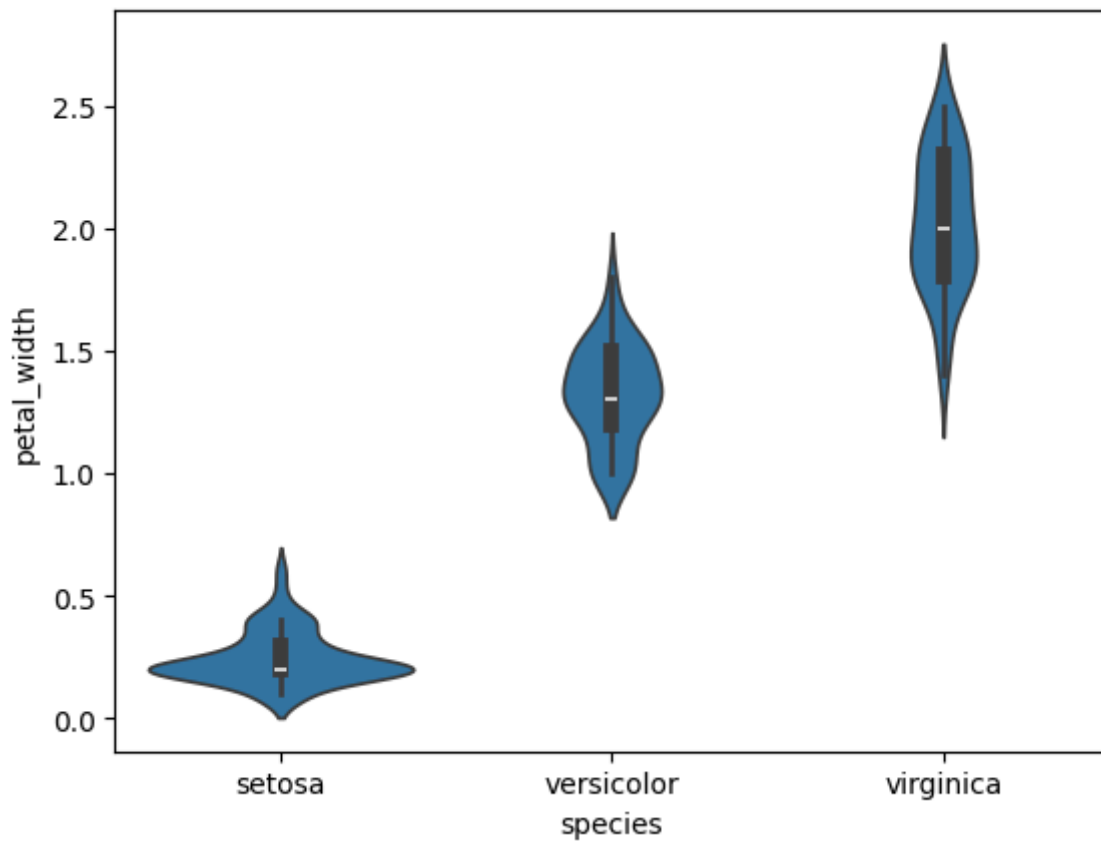
Correlation Heatmap: A correlation heatmap can be useful to visualize the correlation between different variables in the dataset.

```
correlation_matrix = iris.corr()
sns.heatmap(correlation_matrix, annot=True, cmap="cool")
plt.title("Correlation Heatmap")
plt.show()
```

Correlation Heatmap

Barplot: Barplots are helpful for visualizing the distribution of a categorical variable or the relationship between a categorical variable and a continuous variable.

```
sns.barplot(x="species", y="sepal_width", data=iris)
plt.xlabel("Species")
plt.ylabel("Sepal Width")
plt.title("Mean Sepal Width by Species")
plt.show()
```
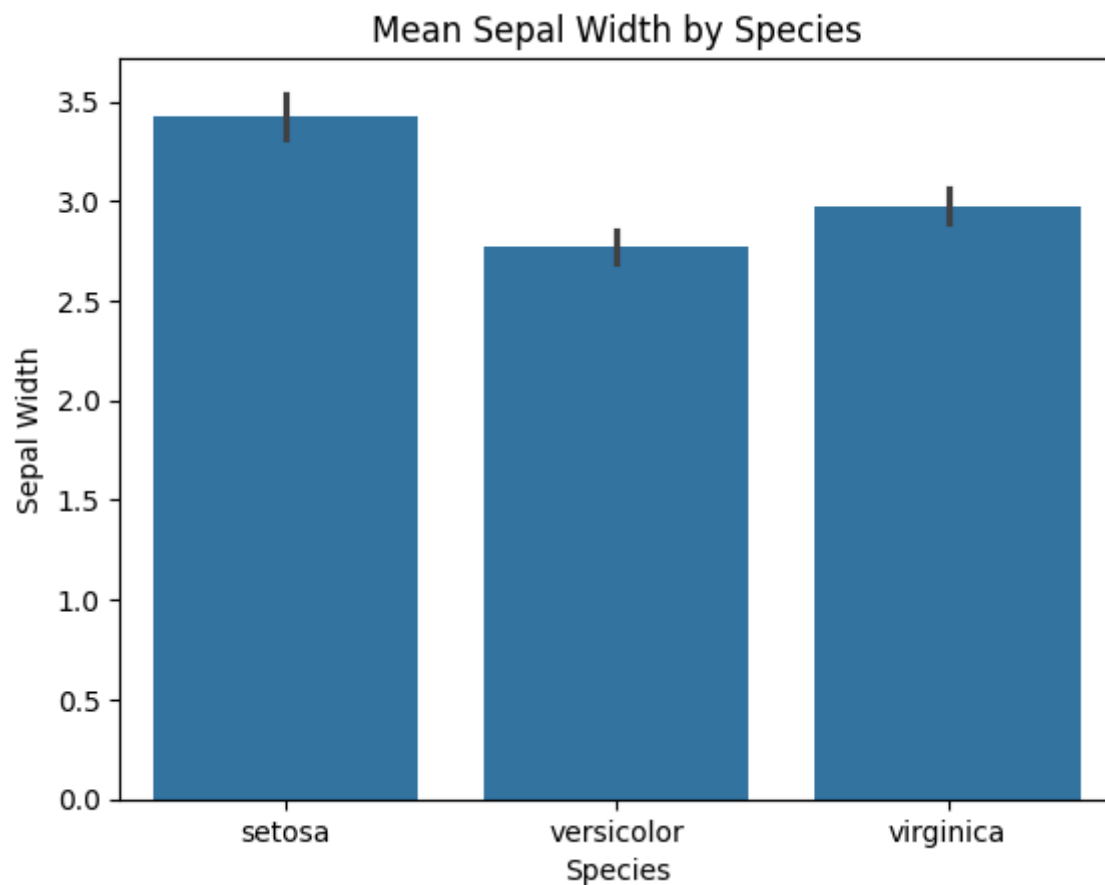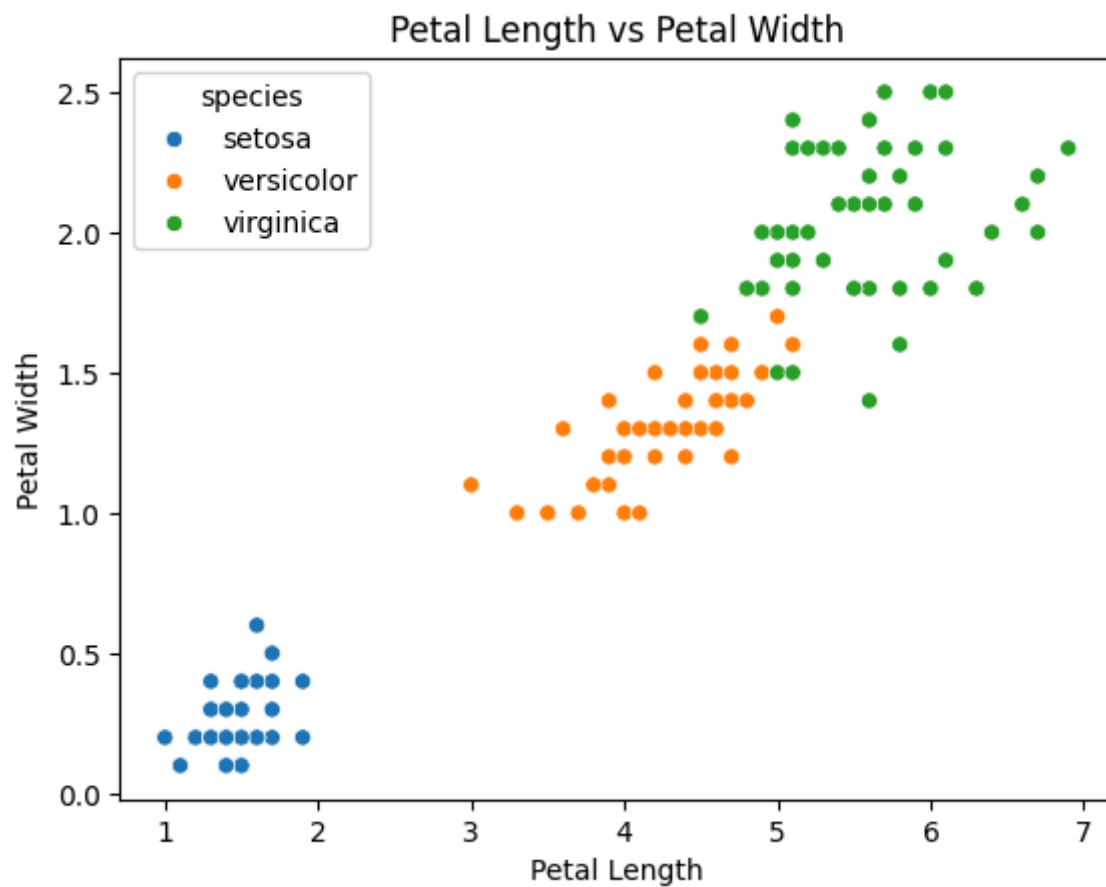
Scatter plot: Scatter plots are useful for visualizing the relationship between two continuous variables.

```
sns.color_palette("Spectral", as_cmap=True)
```



```
sns.scatterplot(x="petal_length", y="petal_width", data=iris, hue="species")
plt.xlabel("Petal Length")
plt.ylabel("Petal Width")
plt.title("Petal Length vs Petal Width")
plt.show()
```

Petal Length vs Petal Width

**Jointplot:** Jointplots can be used to visualize the relationship between two variables along with their individual distributions.

```
sns.jointplot(x="sepal_length", y="sepal_width",palette=None, data=iris, kind="scatter")
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.title(" Jointplot of Sepal Length and Sepal Width")
plt.show()
```
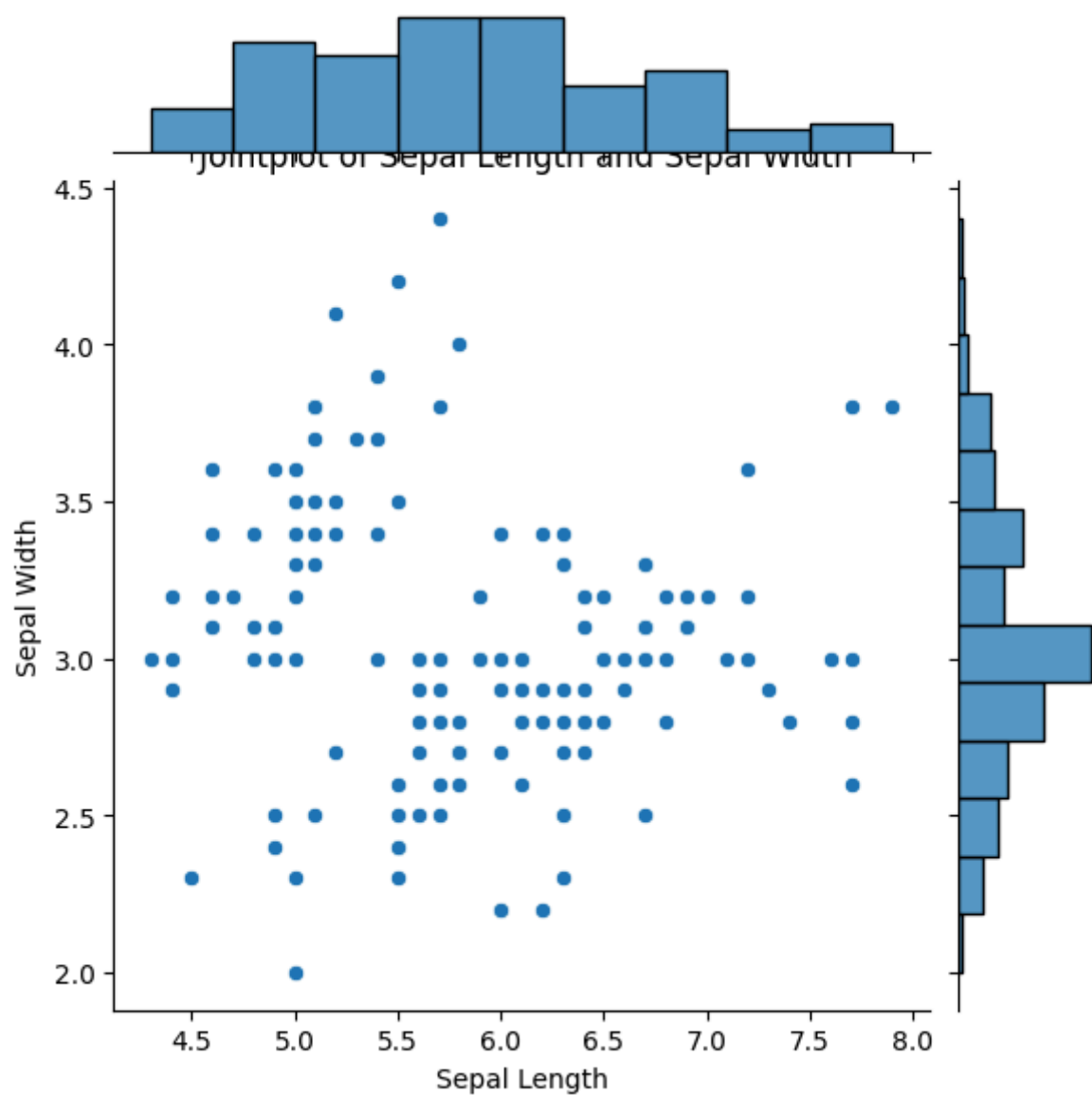
**PairGrid:** PairGrid allows you to create a grid of subplots for pairwise relationships, and it offers flexibility in customizing each subplot.

```
g = sns.PairGrid(iris, hue="species")
g.map_diag(sns.histplot)
g.map_offdiag(sns.scatterplot)
g.add_legend()
plt.show()
```
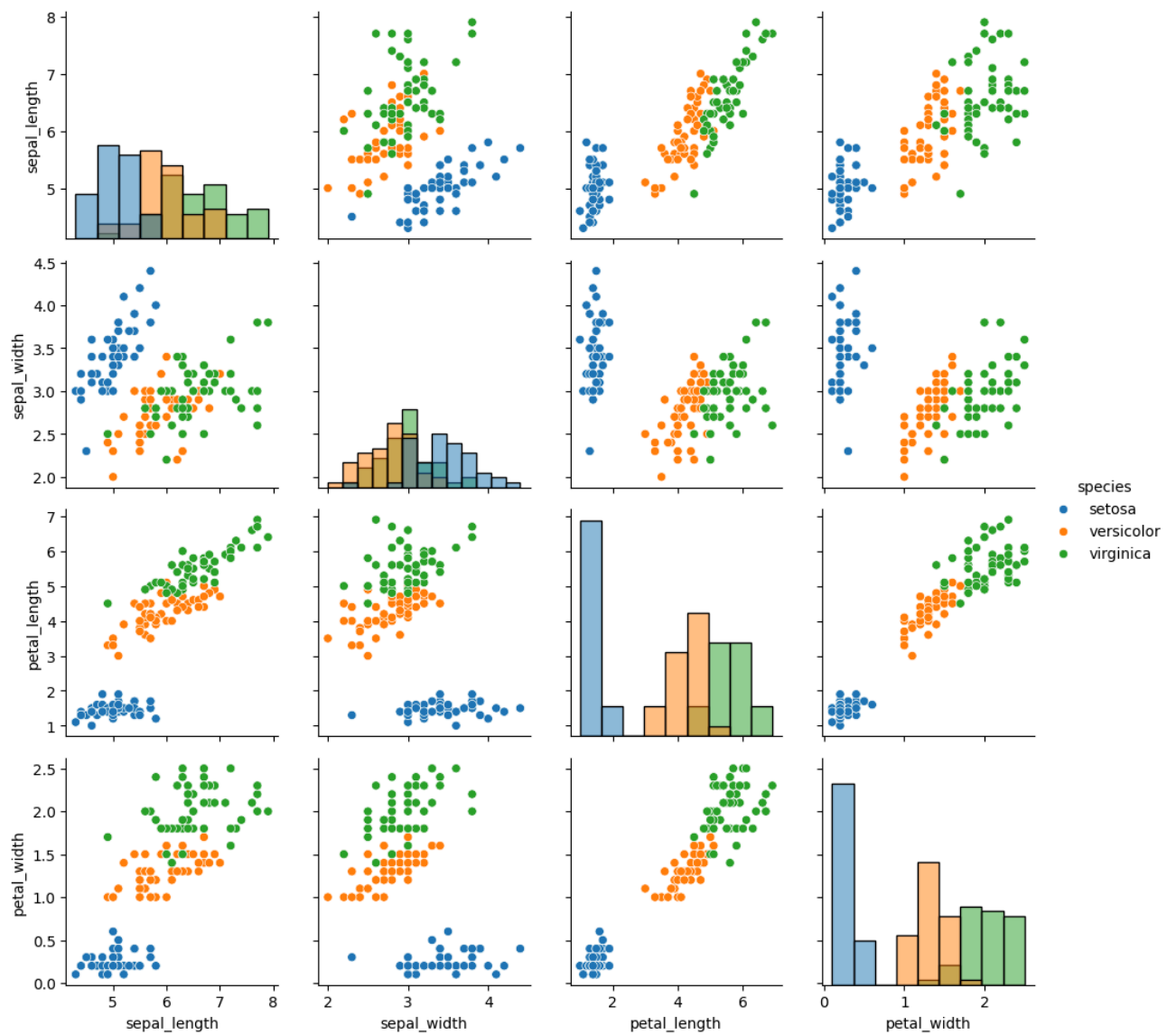
Strip plot: Strip plots are similar to scatter plots but are especially useful for visualizing the distribution of a continuous variable across different categories.

```
sns.stripplot(x="species", y="petal_length", data=iris, jitter=True)
plt.xlabel("Species")
plt.ylabel("Petal Length")
plt.title("Strip Plot of Petal Length by Species")
plt.show()
```
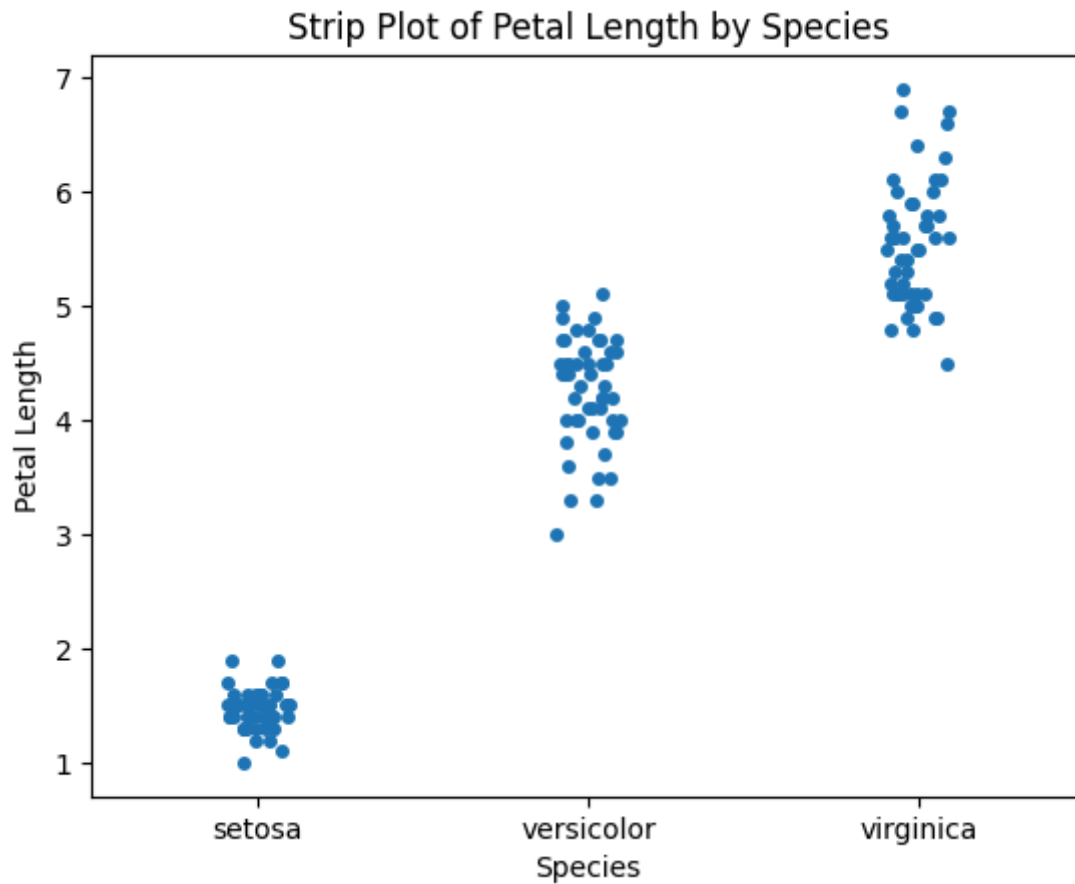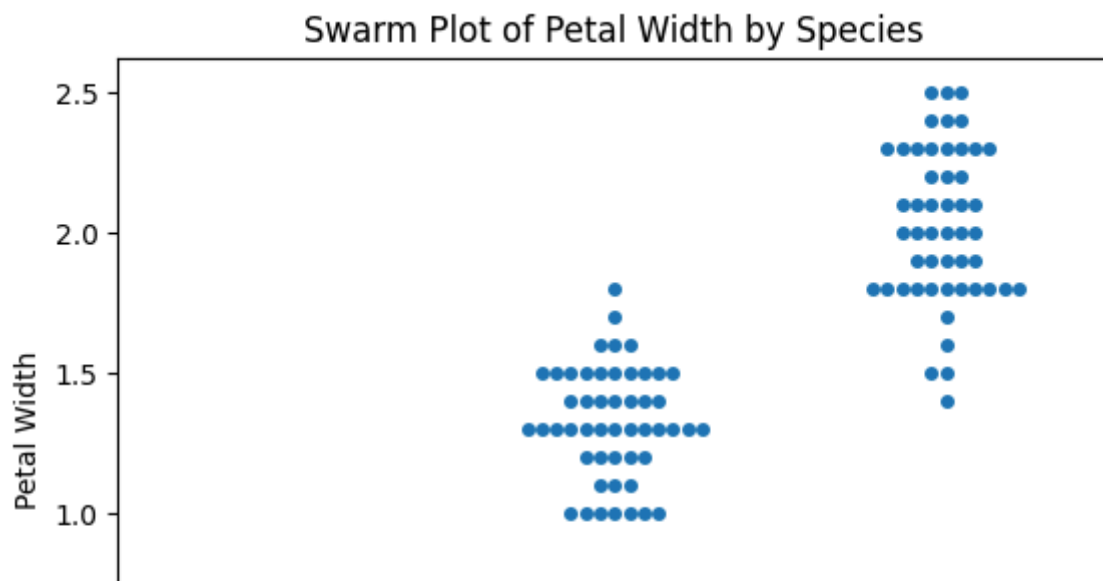

Swarm plot: Swarm plots combine elements of strip plots and violin plots to show each data point while avoiding overlap.

```
sns.swarmplot(x="species", y="petal_width", data=iris)
plt.xlabel("Species")
plt.ylabel("Petal Width")
plt.title("Swarm Plot of Petal Width by Species")
plt.show()
```
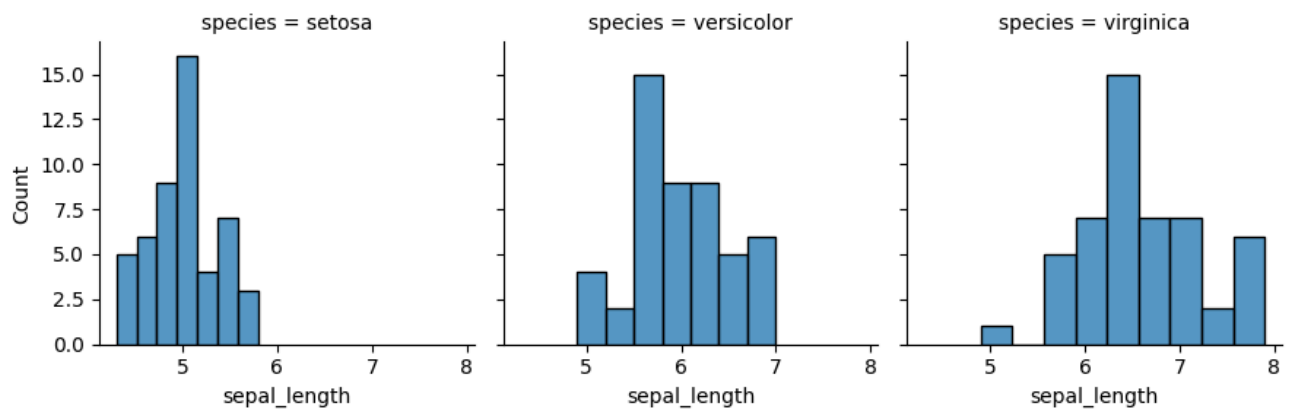
Swarm Plot of Petal Width by Species

FacetGrid: FacetGrid allows you to create multiple plots (facets) based on one or more categorical variables.

```
g = sns.FacetGrid(iris, col="species")
g.map(sns.histplot, "sepal_length")
plt.show()
```

Boxenplot: Boxen plots (also known as letter-value plots) are similar to boxplots but provide more information about the tails of the distribution.