Seaborn Cheat Sheet

This cheat sheet covers the essential and advanced features of Seaborn, a powerful library for creating informative and attractive statistical graphics in Python.

1. Introduction to Seaborn

1.1 What is Seaborn?

Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

1.2 Installing Seaborn

You can install Seaborn using pip:

```
pip install seaborn
```

Or, if you are using Anaconda:

```
conda install seaborn
```

1.3 Importing Seaborn

To use Seaborn, you need to import it in your script:

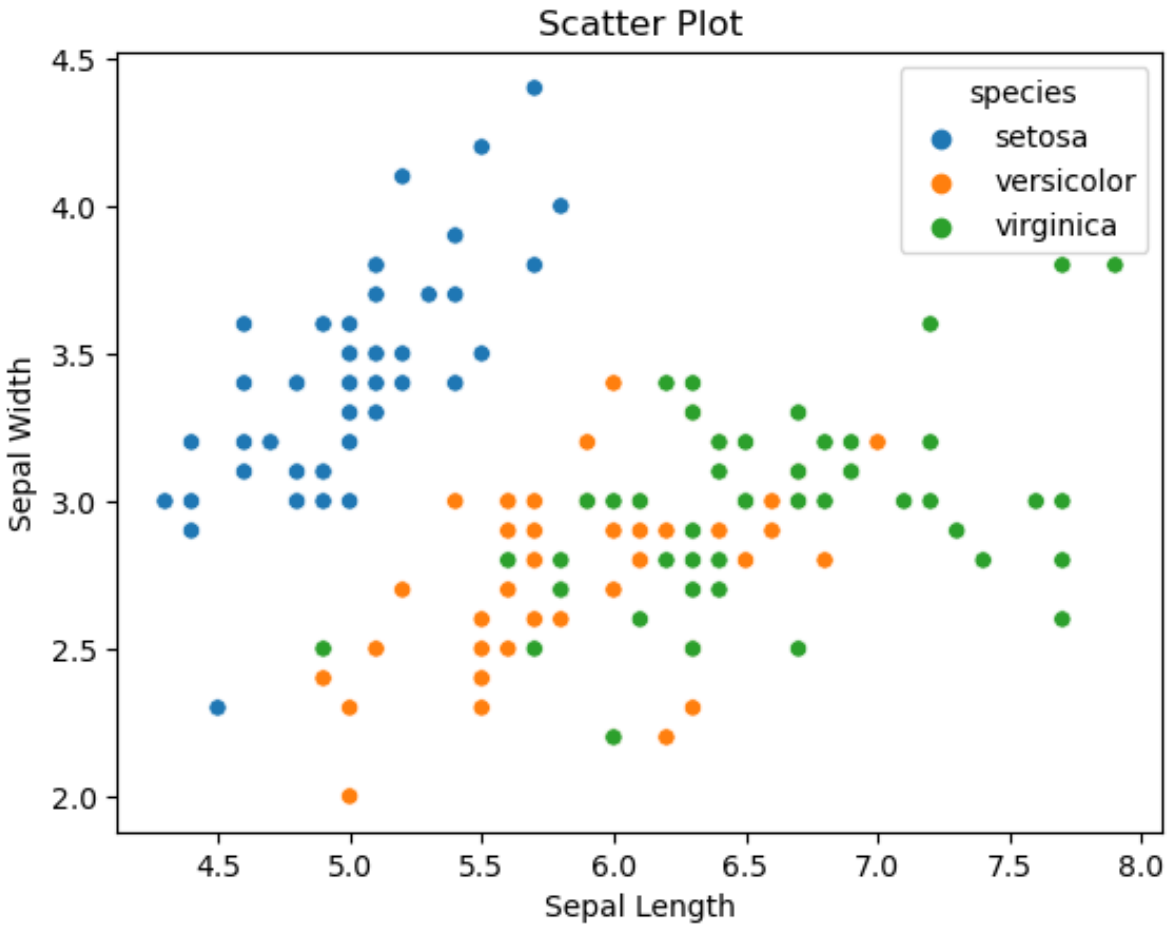
In [1]:
```
import seaborn as sns
import matplotlib.pyplot as plt
# Now you can use Seaborn functions with the sns prefix
```

2. Basic Plots

2.1 Scatter Plot

Scatter plots are used to represent individual data points.

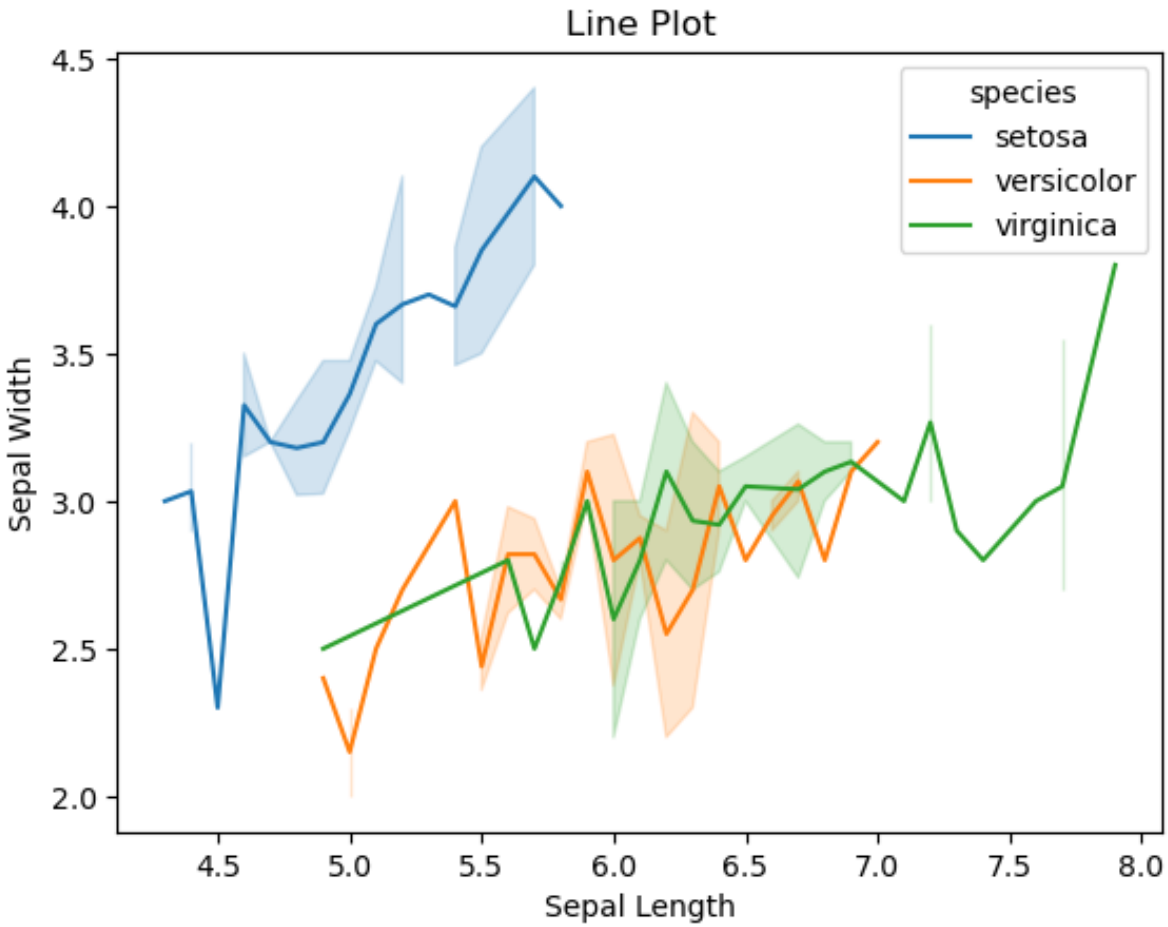
In [2]:
```
data = sns.load_dataset('iris')
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', d
plt.title('Scatter Plot')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.show()
```



2.2 Line Plot

Line plots are used to represent data points connected by straight lines.

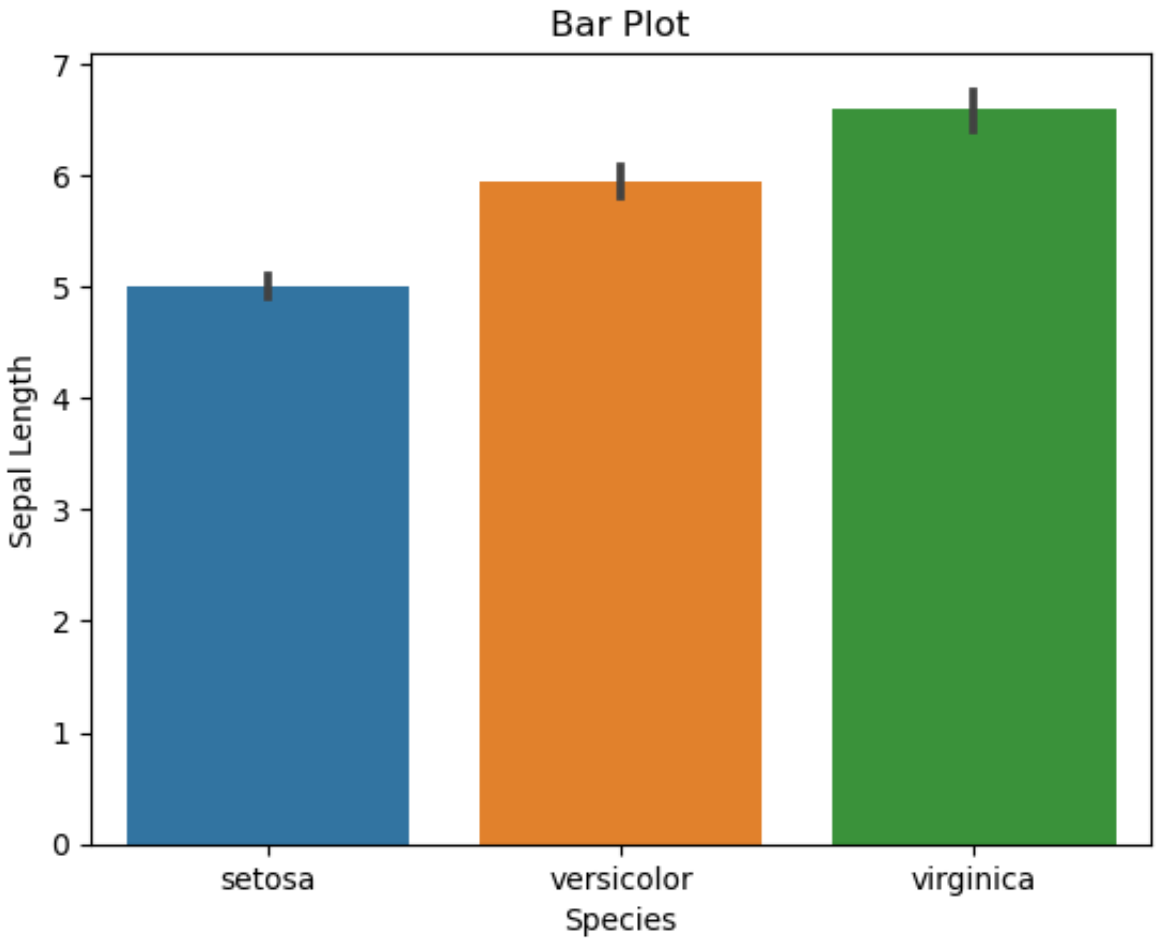
In [3]:
```
sns.lineplot(x='sepal_length', y='sepal_width', hue='species', data
plt.title('Line Plot')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.show()
```



2.3 Bar Plot

Bar plots are used to represent data with rectangular bars.

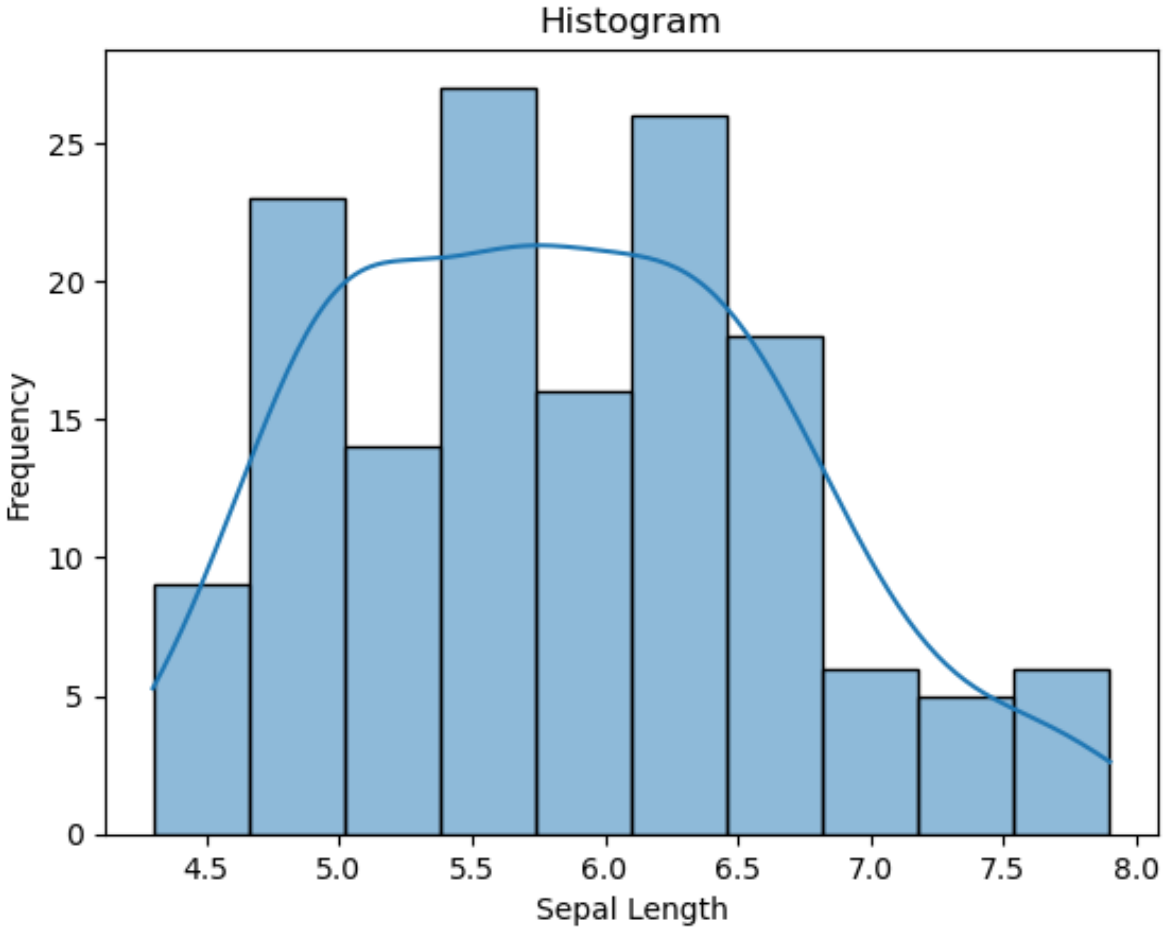
In [4]:
```
sns.barplot(x='species', y='sepal_length', data=data)
plt.title('Bar Plot')
plt.xlabel('Species')
plt.ylabel('Sepal Length')
plt.show()
```



2.4 Histogram

Histograms are used to represent the distribution of a dataset.

In [5]:
```
sns.histplot(data['sepal_length'], bins=10, kde=True)
plt.title('Histogram')
plt.xlabel('Sepal Length')
plt.ylabel('Frequency')
plt.show()
```

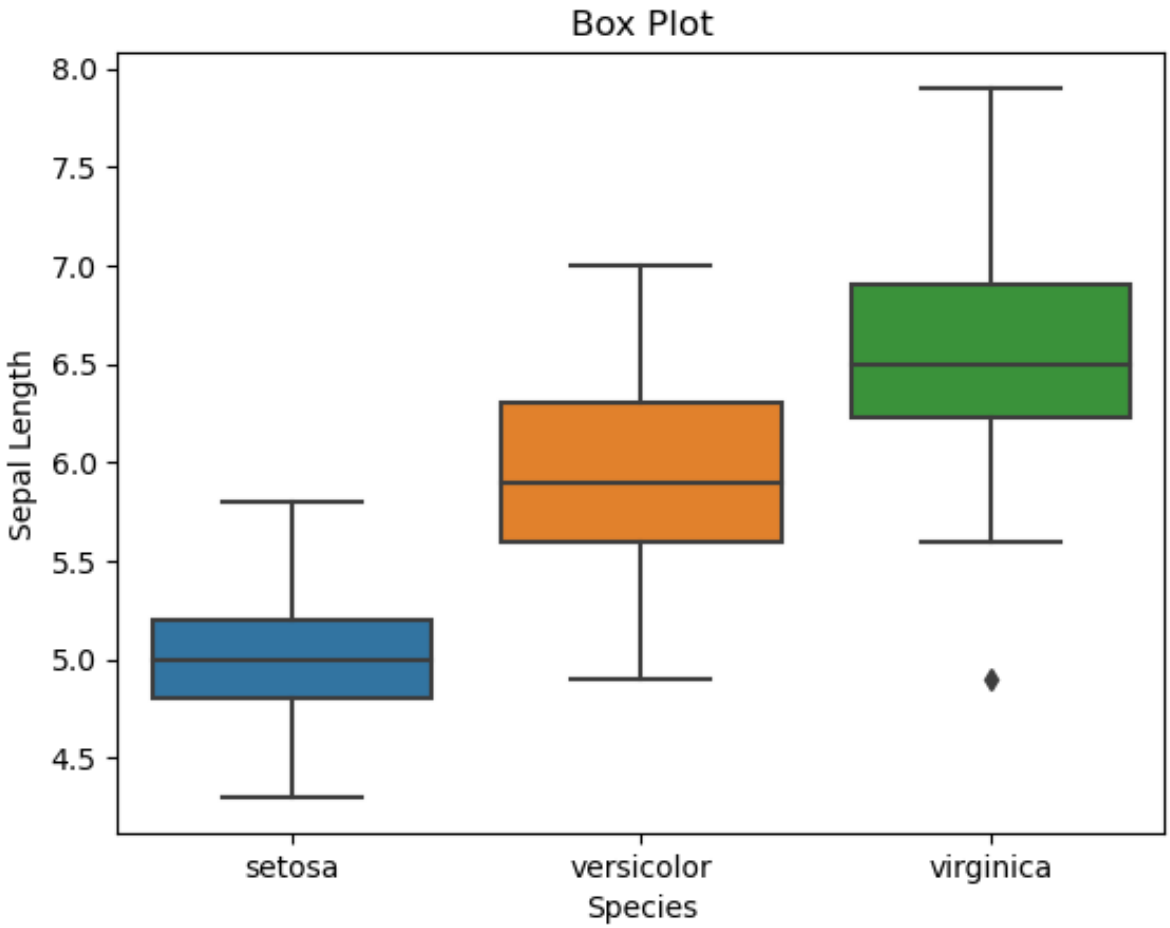


3. Statistical Plots

3.1 Box Plot

Box plots are used to visualize the distribution of data based on a five-number summary.

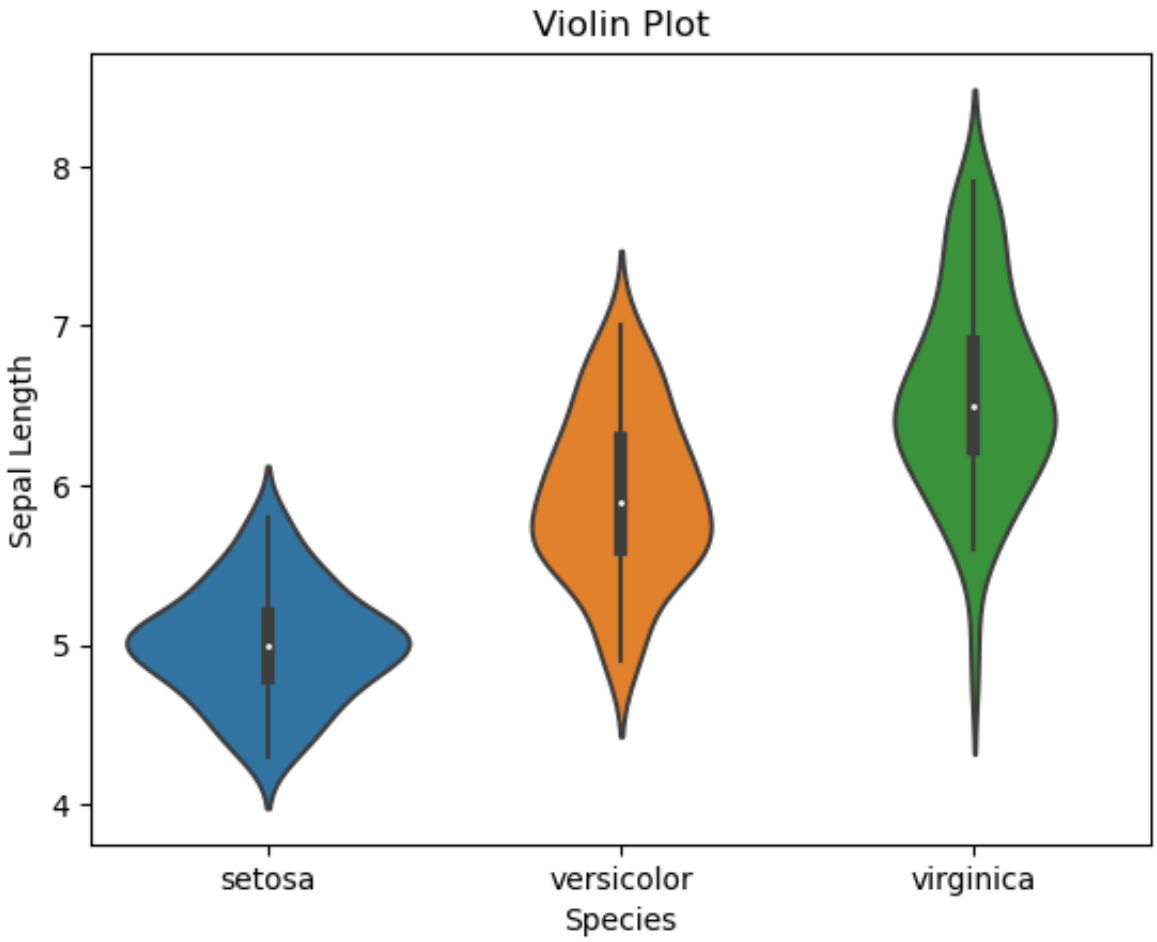
In [6]:
```
sns.boxplot(x='species', y='sepal_length', data=data)
plt.title('Box Plot')
plt.xlabel('Species')
plt.ylabel('Sepal Length')
plt.show()
```



3.2 Violin Plot

Violin plots are similar to box plots but also show the density of the data at different values.

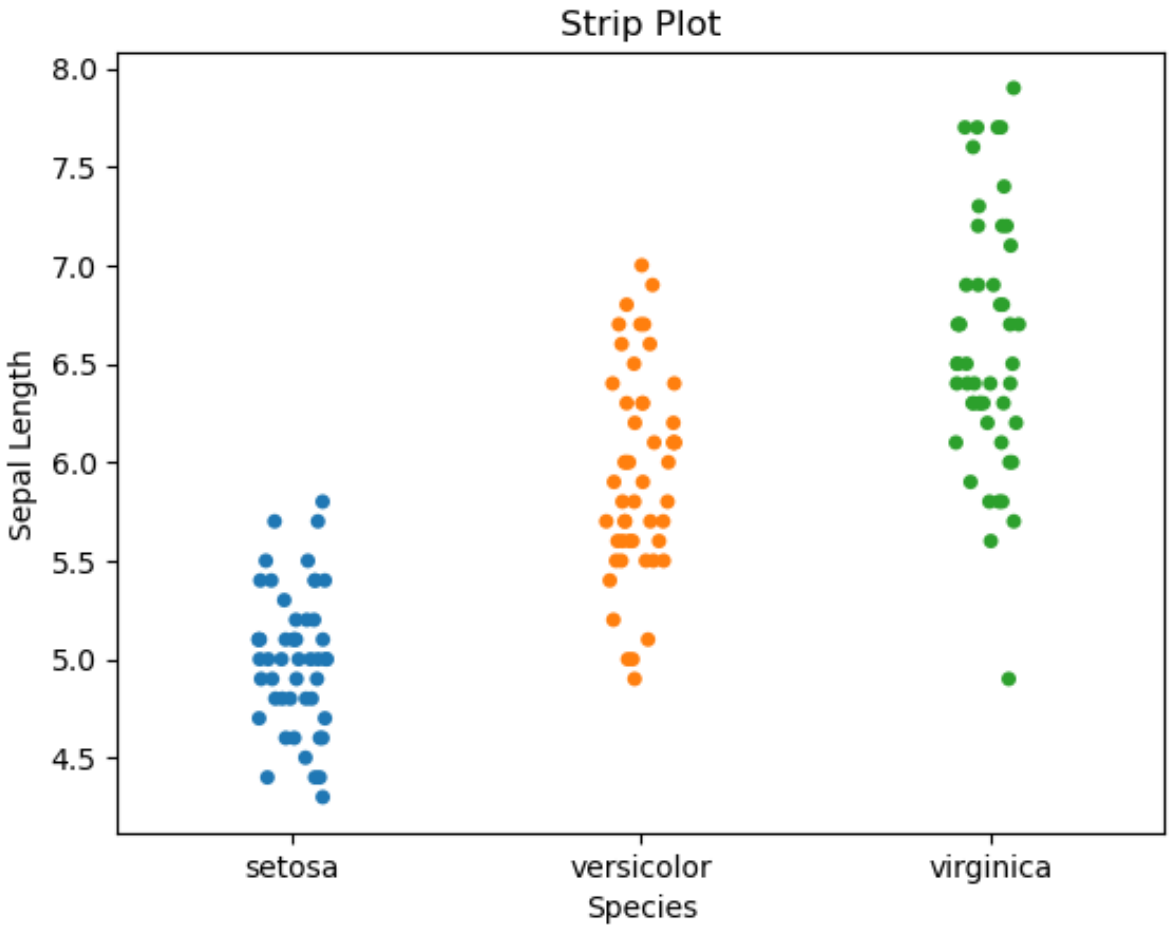
In [7]:
```
sns.violinplot(x='species', y='sepal_length', data=data)
plt.title('Violin Plot')
plt.xlabel('Species')
plt.ylabel('Sepal Length')
plt.show()
```



3.3 Strip Plot

Strip plots are used to represent individual data points with a jitter to avoid overlap.

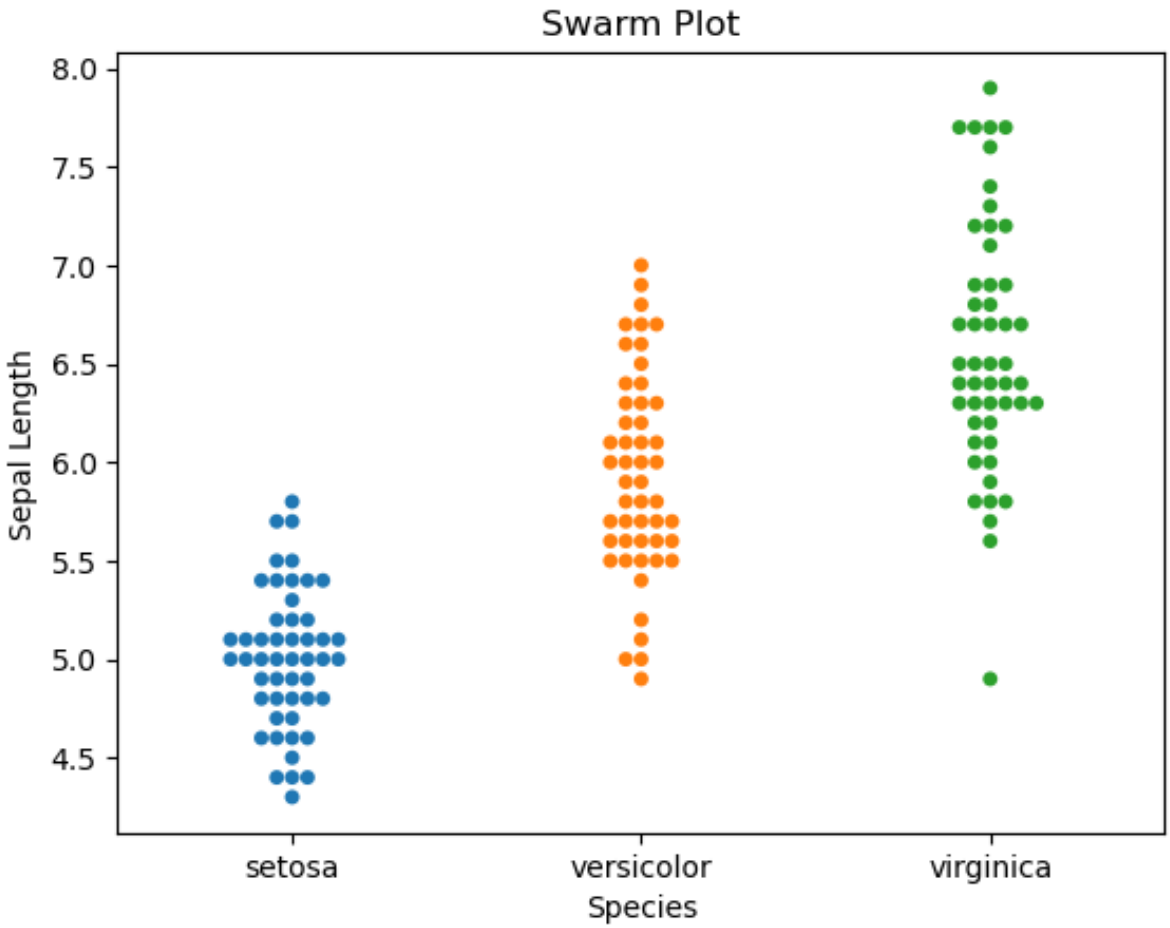
In [8]:
```
sns.stripplot(x='species', y='sepal_length', data=data)
plt.title('Strip Plot')
plt.xlabel('Species')
plt.ylabel('Sepal Length')
plt.show()
```



3.4 Swarm Plot

Swarm plots are similar to strip plots but avoid overlapping data points.

In [9]:
```
sns.swarmplot(x='species', y='sepal_length', data=data)
plt.title('Swarm Plot')
plt.xlabel('Species')
plt.ylabel('Sepal Length')
plt.show()
```

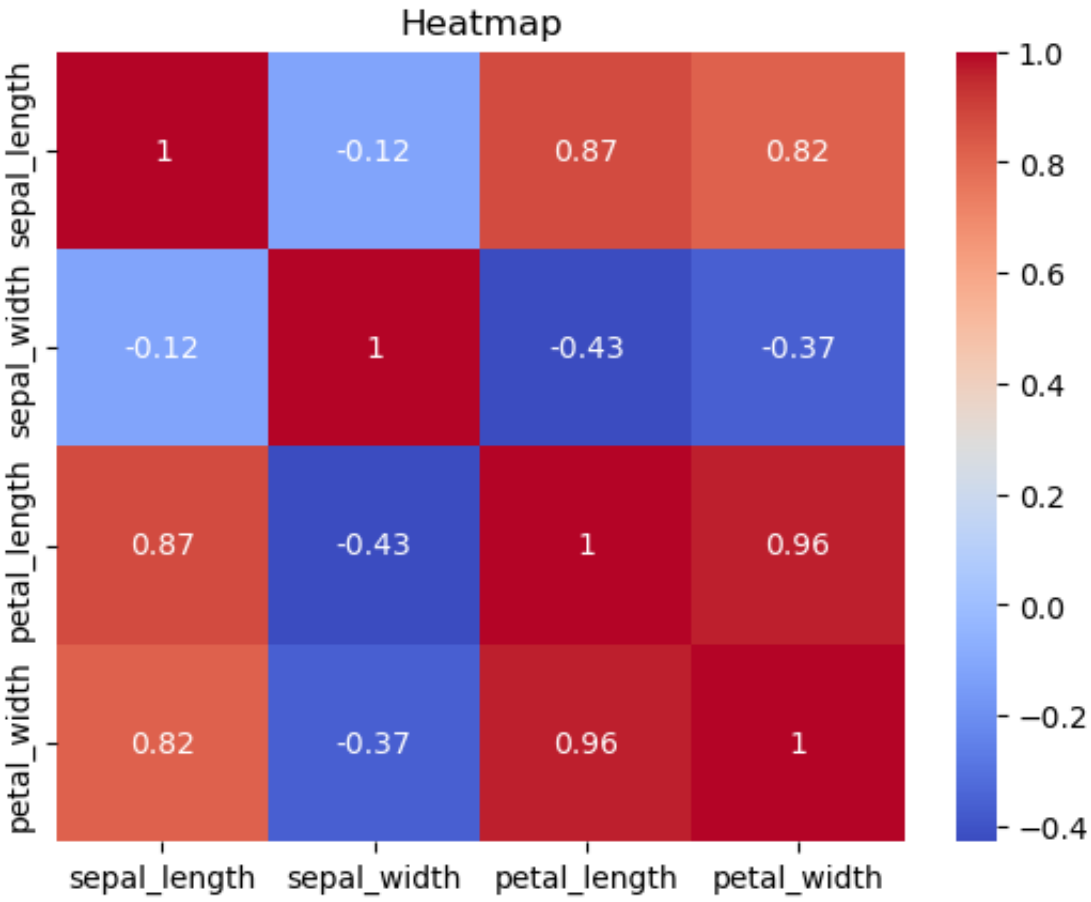


4. Matrix Plots

4.1 Heatmap

Heatmaps are used to represent data values as colors in a matrix.

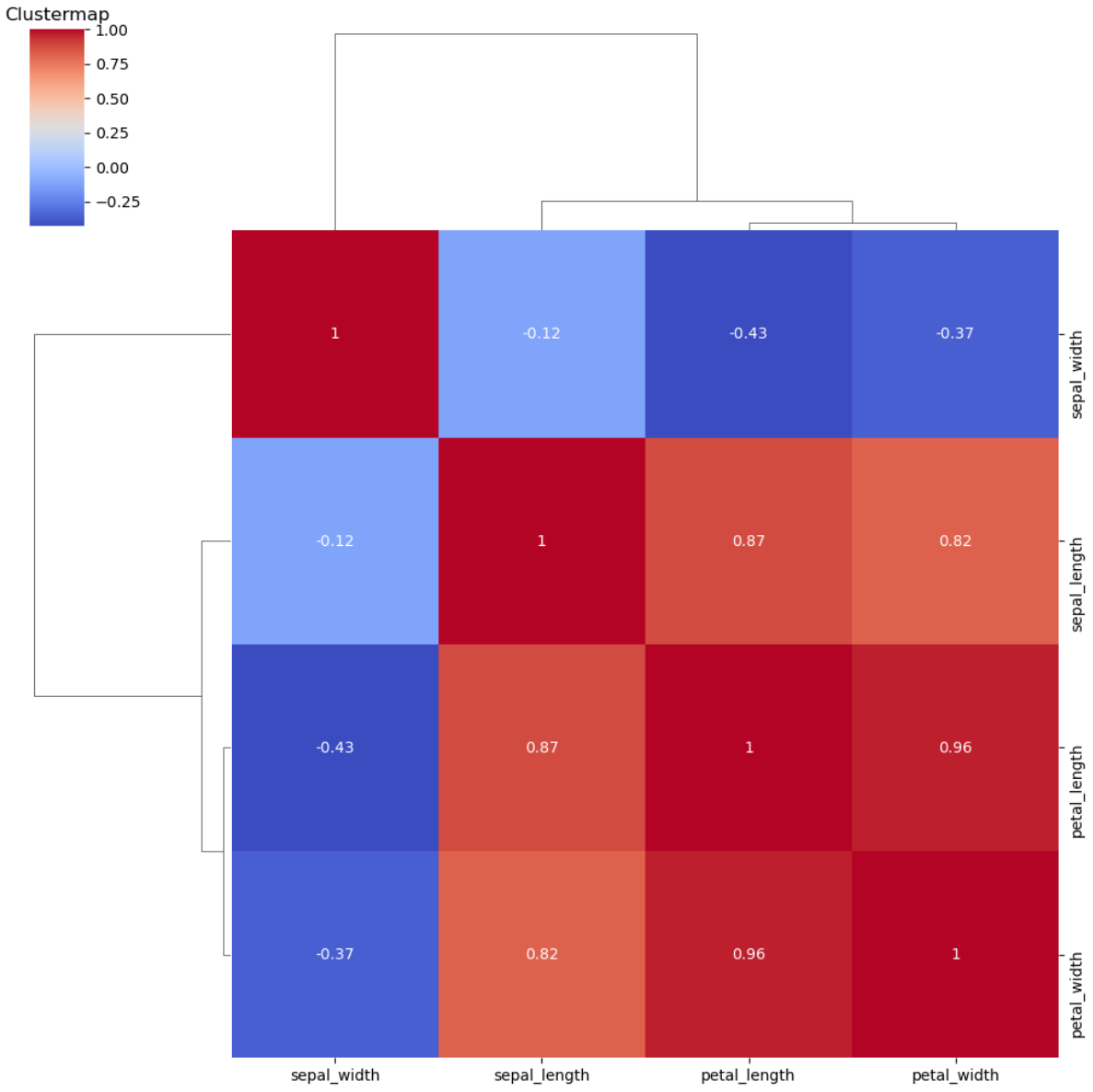
In [10]:
```
corr = data.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Heatmap')
plt.show()
```



4.2 Clustermap

Clustermaps are used to represent data values as colors in a matrix with hierarchical clustering.

In [11]:
```
sns.clustermap(corr, annot=True, cmap='coolwarm')
plt.title('Clustermap')
plt.show()
```

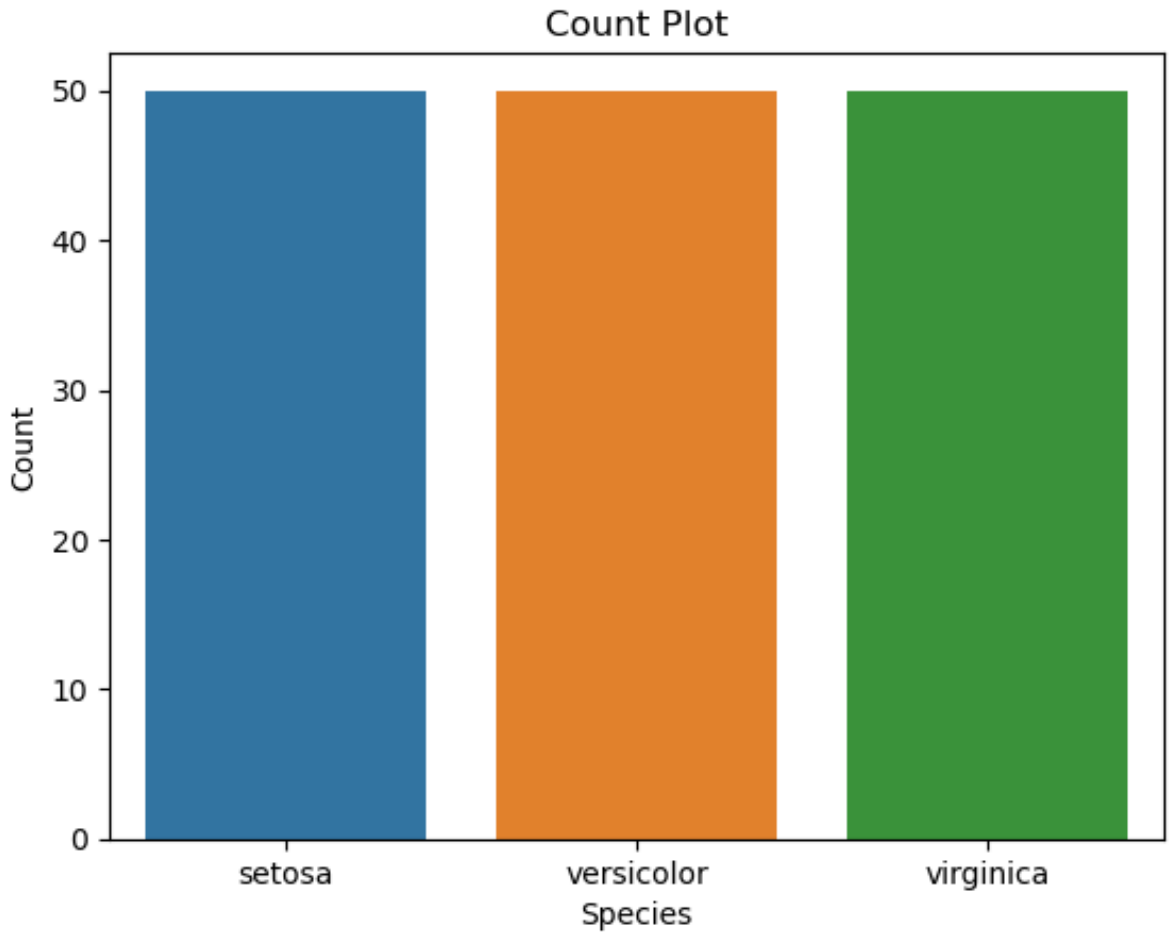


5. Categorical Plots

5.1 Count Plot

Count plots are used to show the counts of observations in each categorical bin using bars.

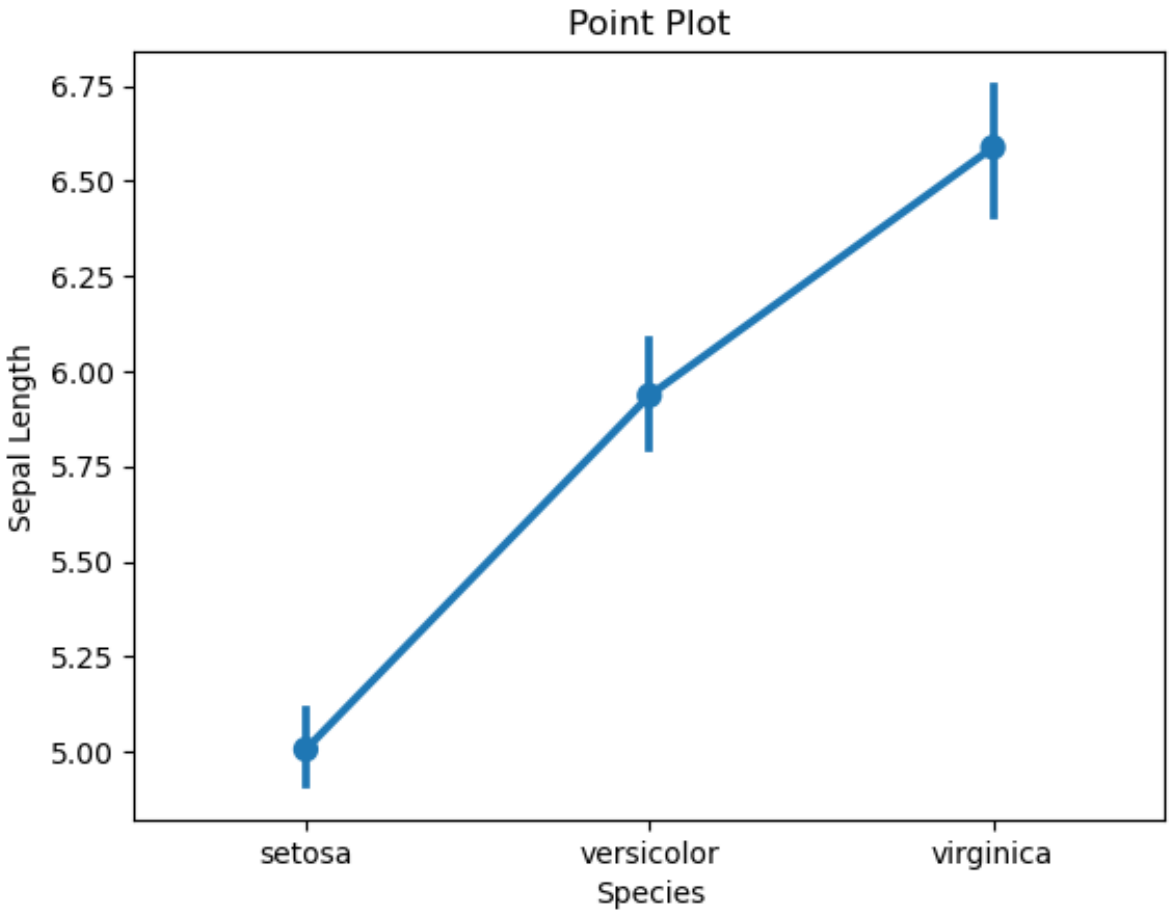
In [12]:
```
sns.countplot(x='species', data=data)
plt.title('Count Plot')
plt.xlabel('Species')
plt.ylabel('Count')
plt.show()
```



5.2 Point Plot

Point plots are used to show point estimates and confidence intervals using scatter points.

In [13]:
```
sns.pointplot(x='species', y='sepal_length', data=data)
plt.title('Point Plot')
plt.xlabel('Species')
plt.ylabel('Sepal Length')
plt.show()
```

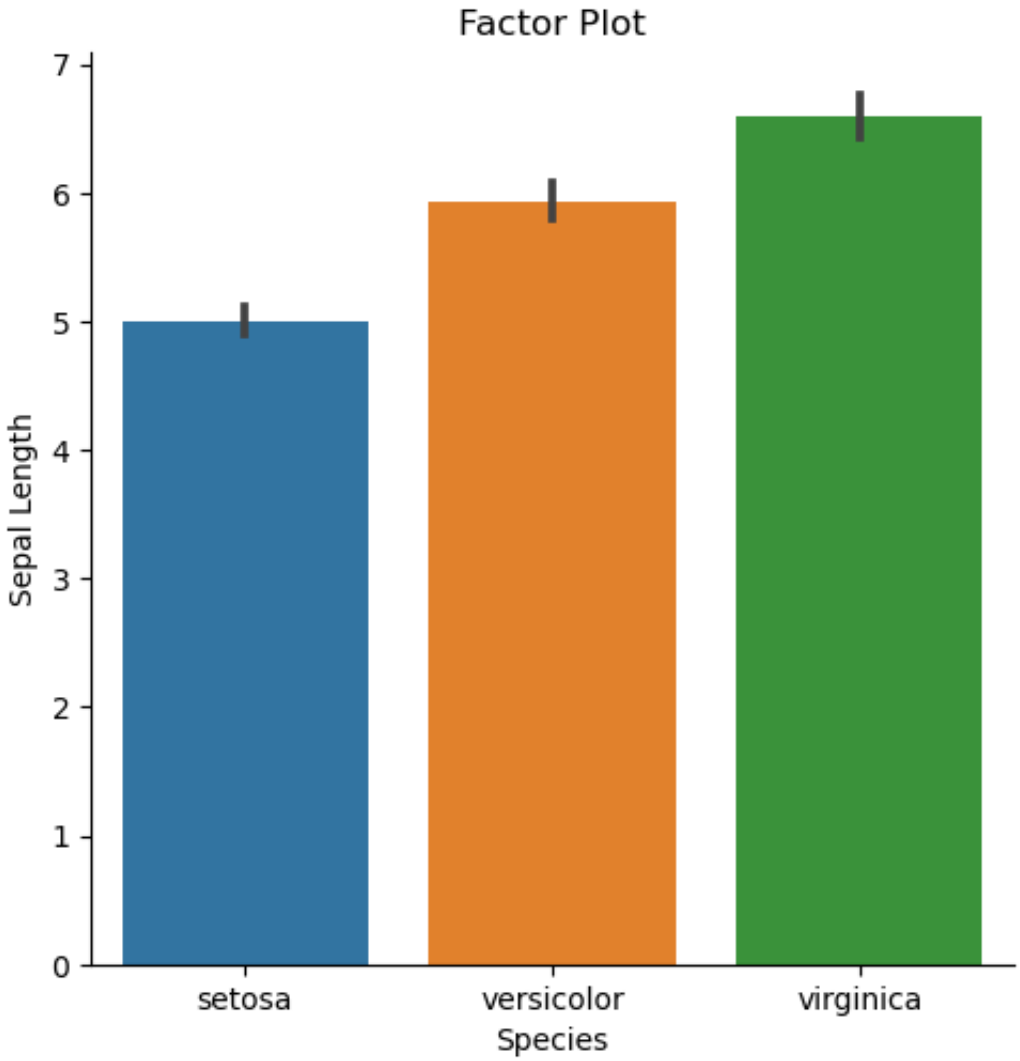


5.3 Factor Plot

Factor plots are used to draw multi-faceted categorical plots.

In [14]:
```
sns.factorplot(x='species', y='sepal_length', data=data, kind='bar'
plt.title('Factor Plot')
plt.xlabel('Species')
plt.ylabel('Sepal Length')
plt.show()
```

```
/Users/ashishzangra/opt/anaconda3/lib/python3.9/site-packages/seab
orn/categorical.py:3717: UserWarning: The `factorplot` function ha
s been renamed to `catplot`. The original name will be removed in
a future release. Please update your code. Note that the default `
kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplo
t`.
  warnings.warn(msg)
```

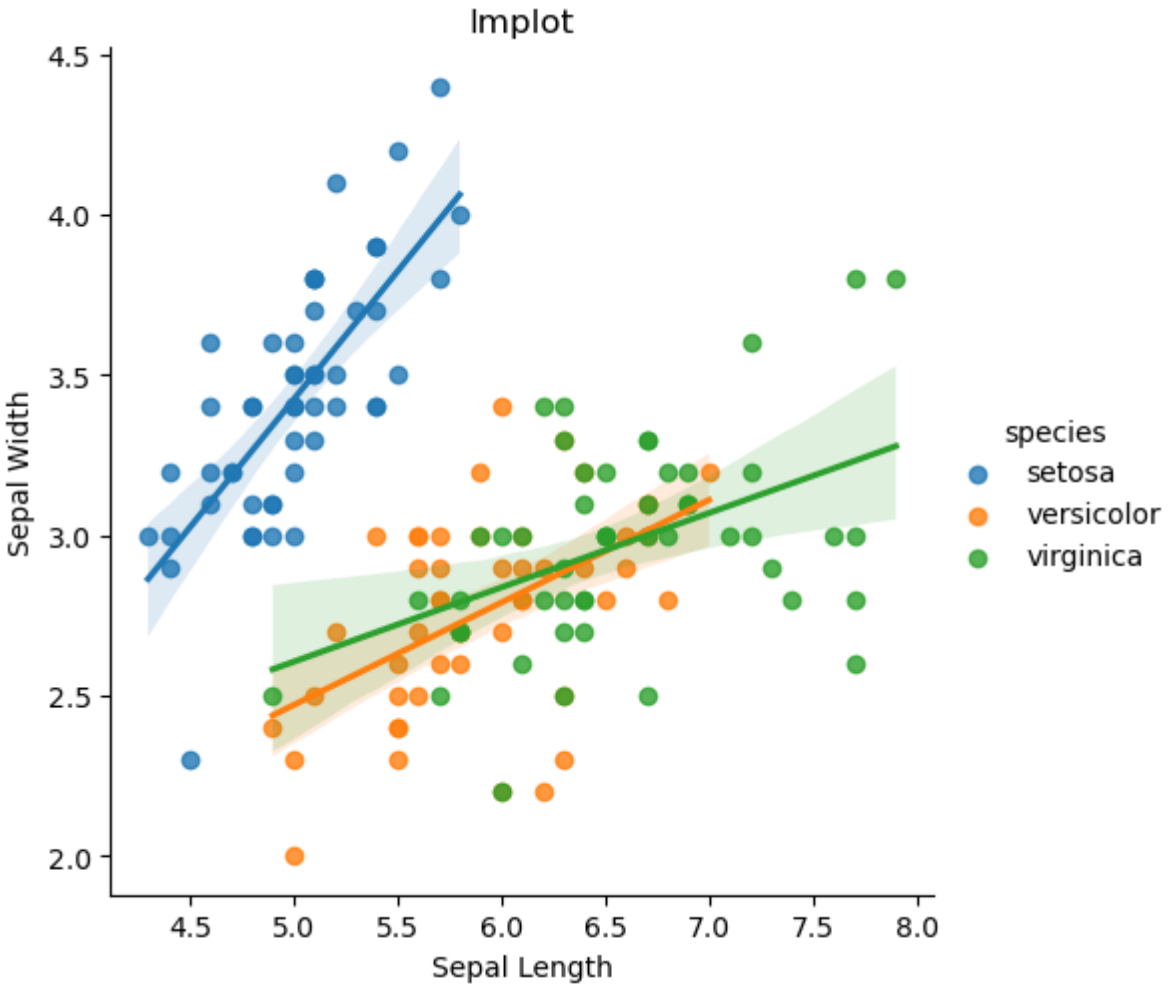


6. Regression Plots

6.1 Implot

Implot is used to draw a scatter plot with a linear regression model fit.

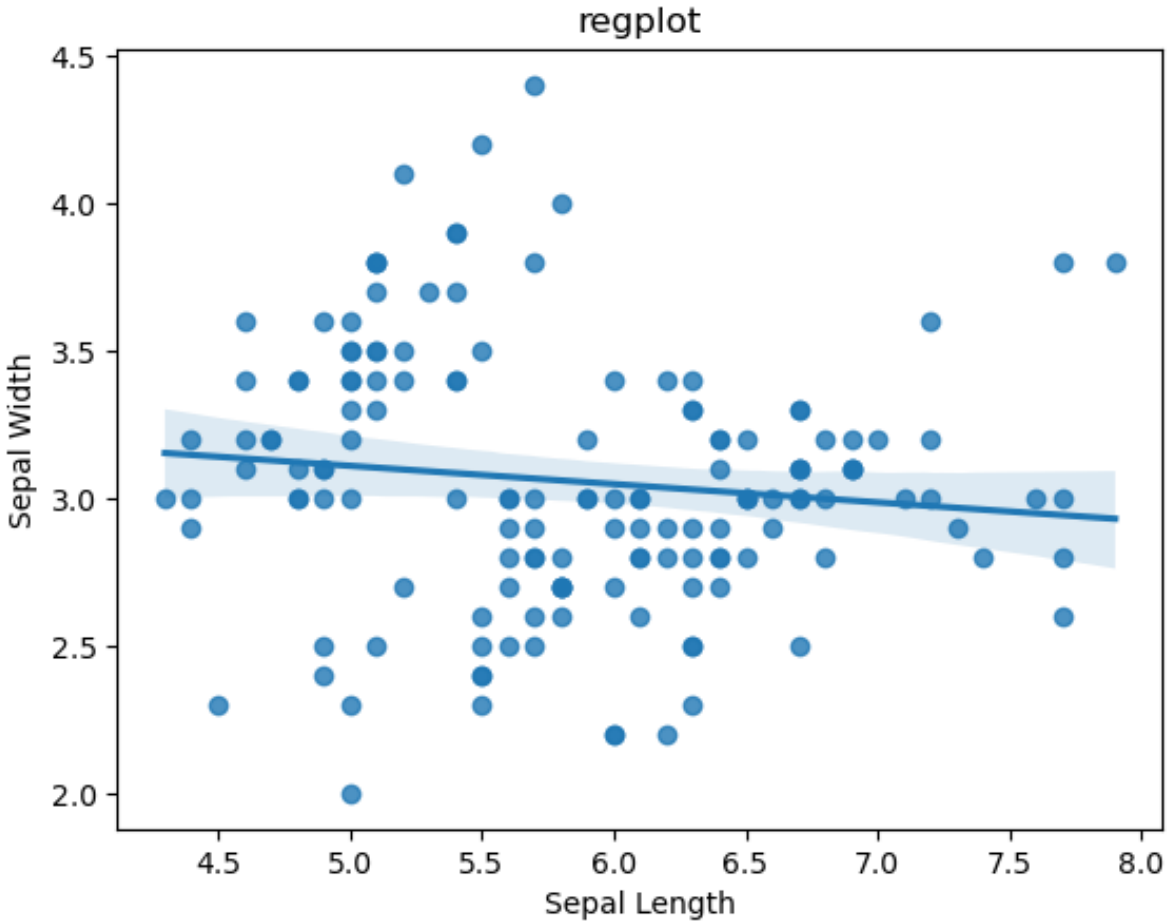
In [15]:
```
sns.lmplot(x='sepal_length', y='sepal_width', hue='species', data=d
plt.title('lmplot')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.show()
```



6.2 regplot

regplot is used to draw a scatter plot with a linear regression model fit.

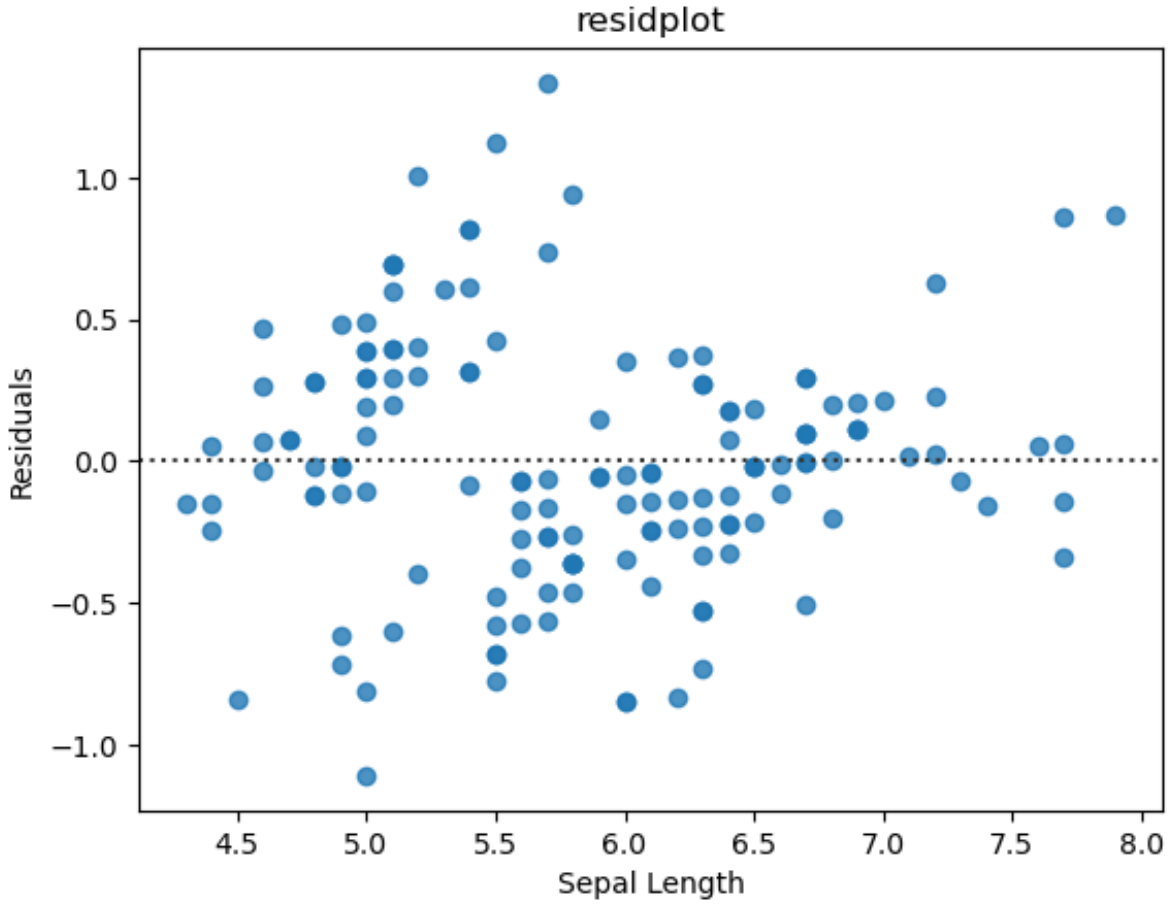
In [16]:
```
sns.regplot(x='sepal_length', y='sepal_width', data=data)
plt.title('regplot')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.show()
```



6.3 residplot

residplot is used to draw a scatter plot of the residuals of a linear regression.

In [17]:
```
sns.residplot(x='sepal_length', y='sepal_width', data=data)
plt.title('residplot')
plt.xlabel('Sepal Length')
plt.ylabel('Residuals')
plt.show()
```



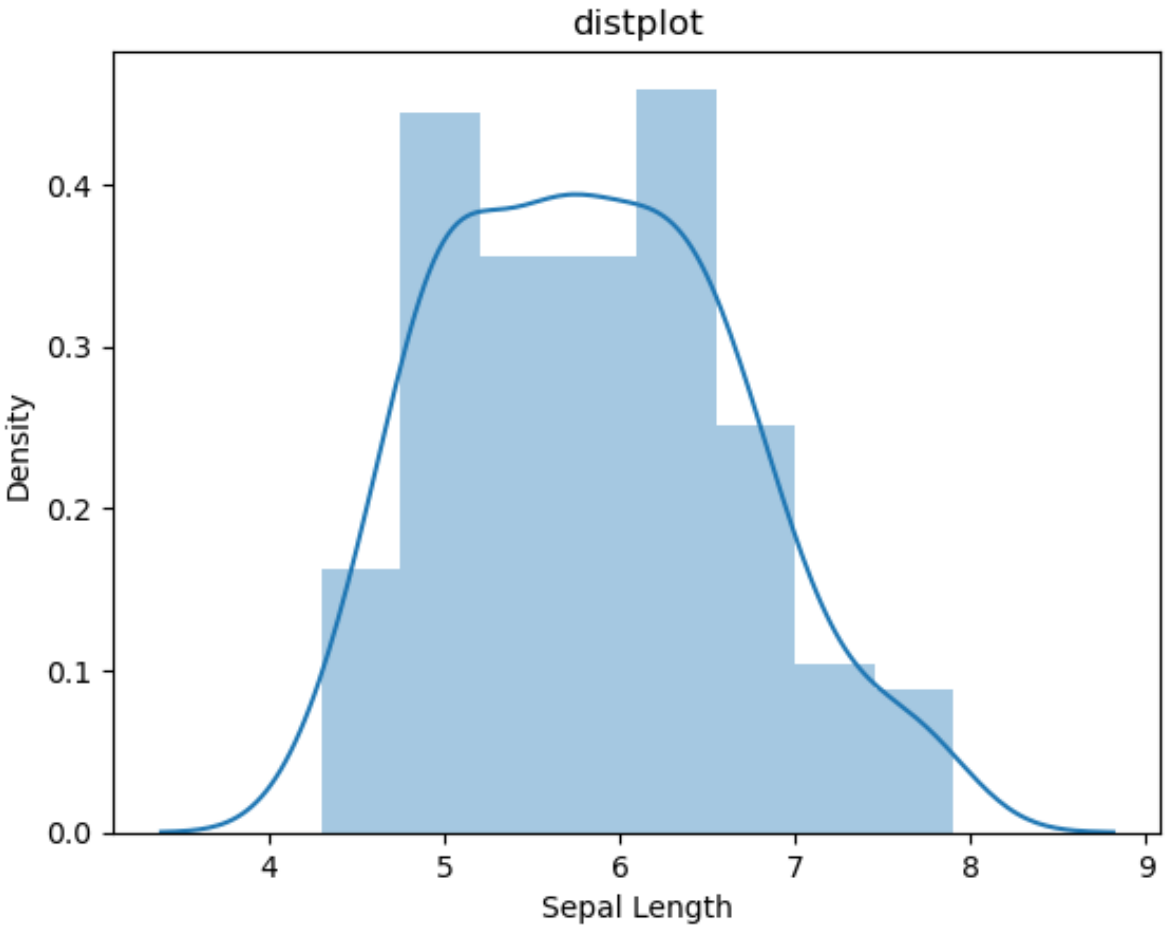
7. Distribution Plots

7.1 distplot

distplot is used to plot a univariate distribution of observations.

In [18]:
```
sns.distplot(data['sepal_length'])
plt.title('distplot')
plt.xlabel('Sepal Length')
plt.ylabel('Density')
plt.show()
```

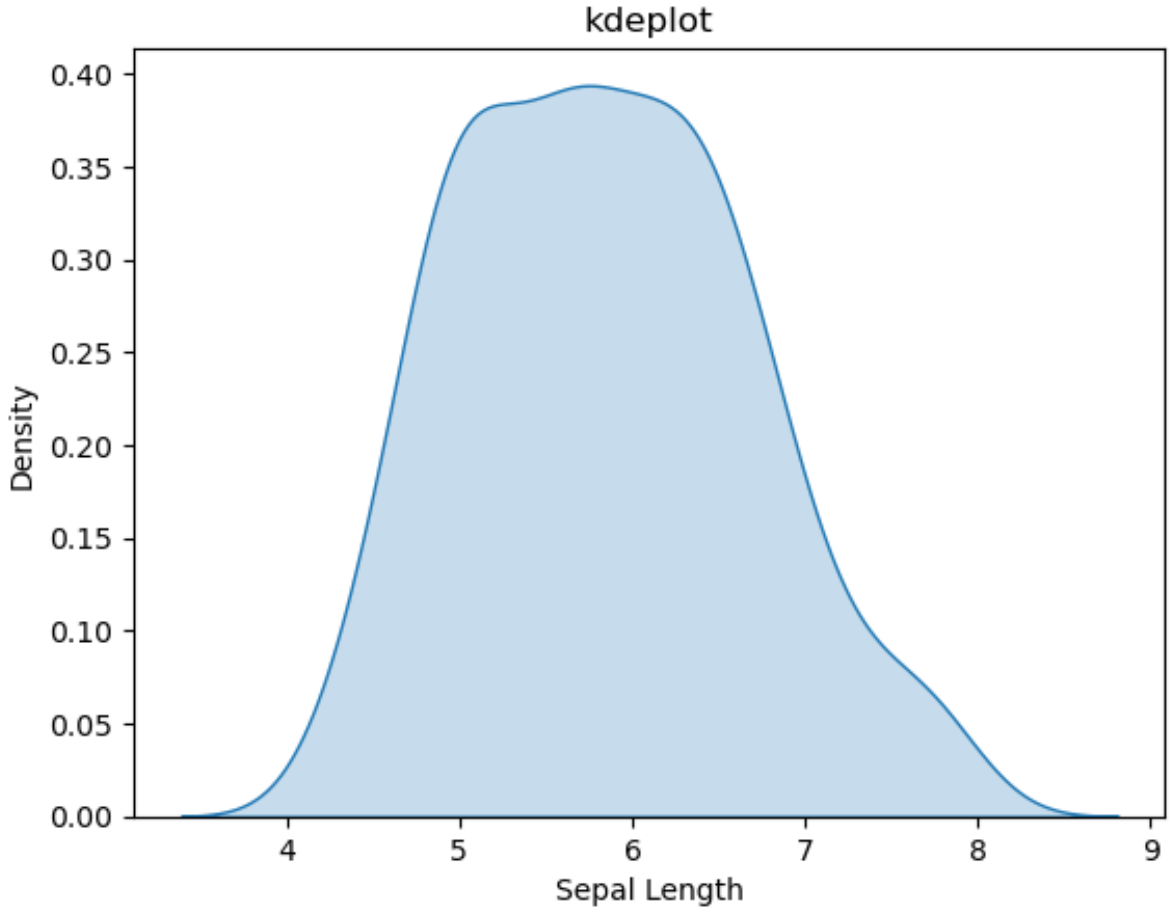
```
/Users/ashishzangra/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



7.2 kdeplot

kdeplot is used to plot a univariate or bivariate kernel density estimate.

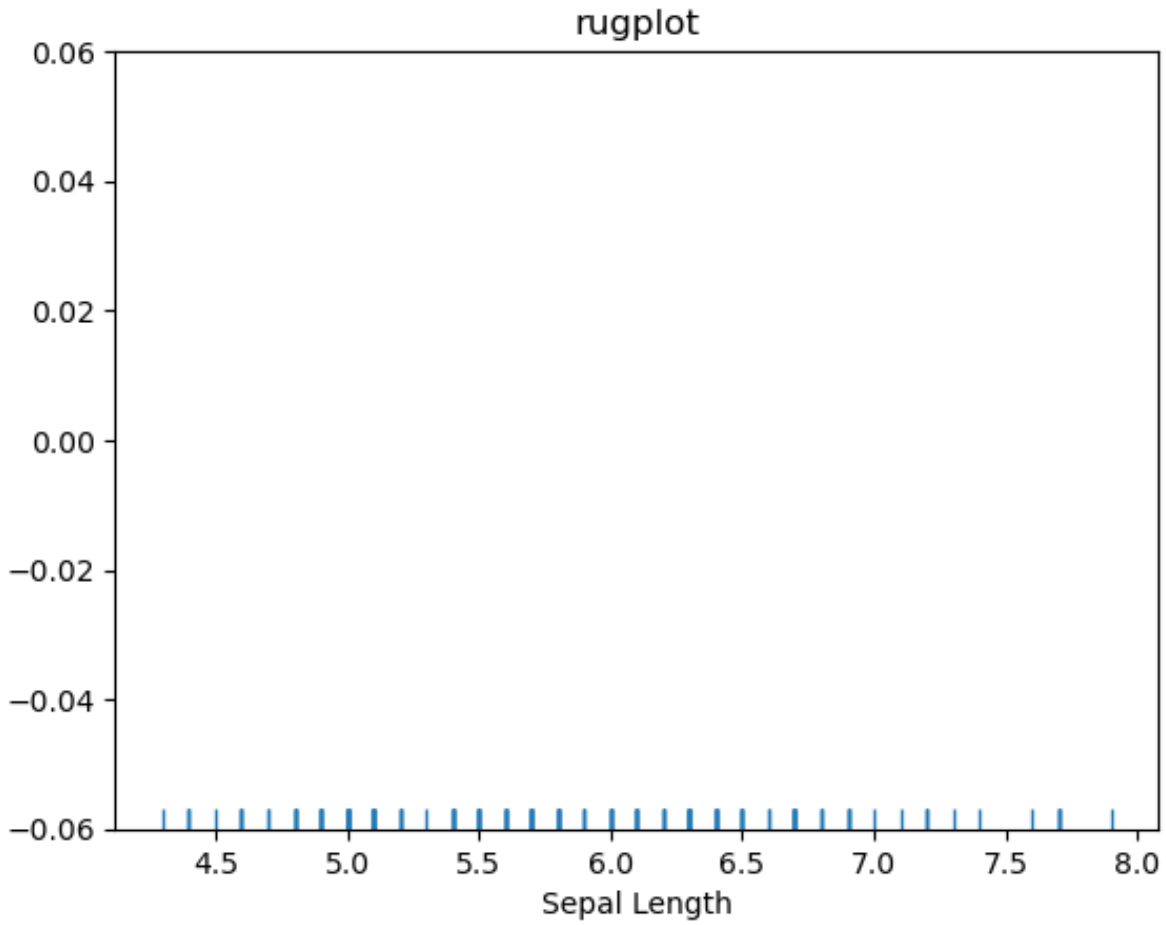
In [19]:
```
sns.kdeplot(data['sepal_length'], shade=True)
plt.title('kdeplot')
plt.xlabel('Sepal Length')
plt.ylabel('Density')
plt.show()
```



7.3 rugplot

rugplot is used to plot datapoints in an array as sticks on an axis.

In [20]:
```
sns.rugplot(data['sepal_length'])
plt.title('rugplot')
plt.xlabel('Sepal Length')
plt.show()
```

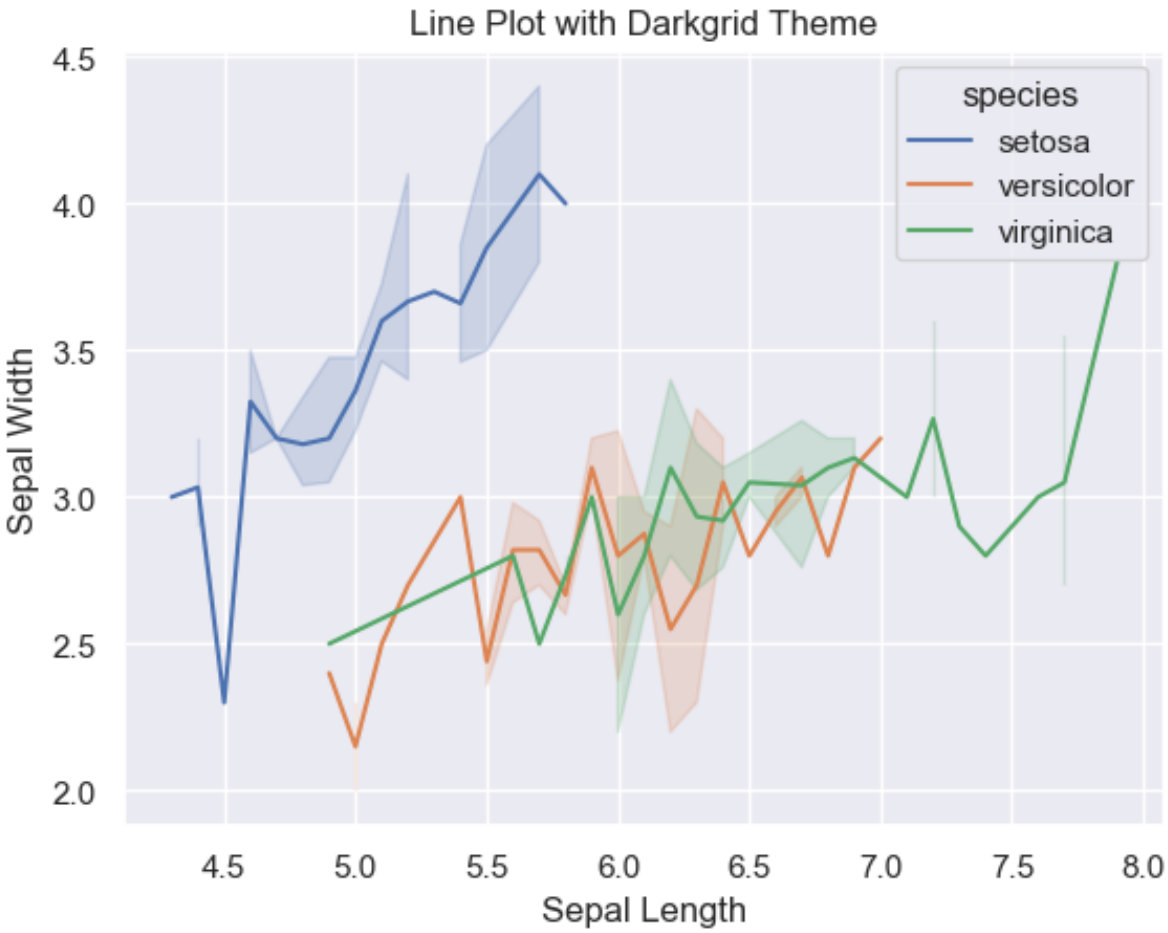


8. Customizing Plots

8.1 Themes and Styles

Seaborn allows you to set different themes and styles for your plots.

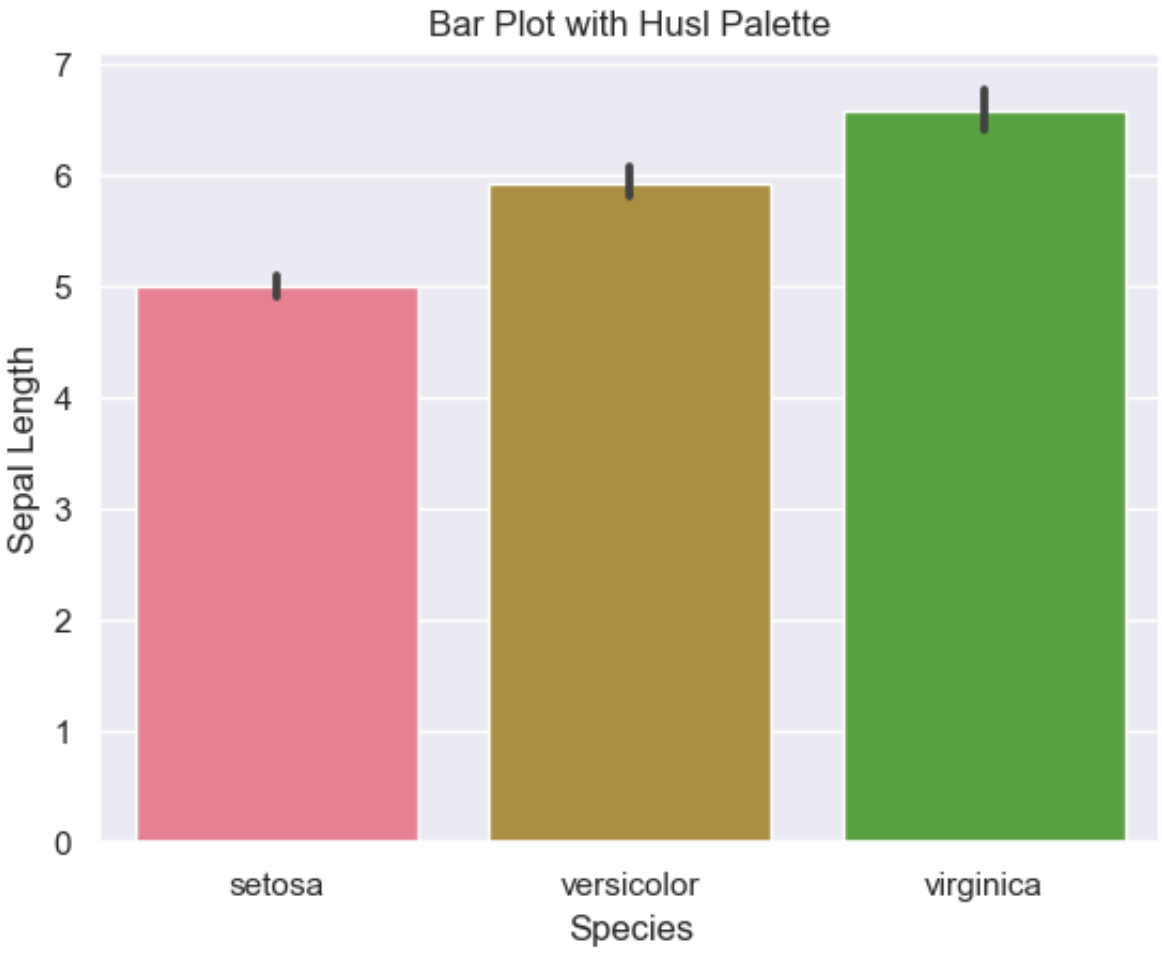
In [21]:
```
sns.set_theme(style='darkgrid')
sns.lineplot(x='sepal_length', y='sepal_width', hue='species', data
plt.title('Line Plot with Darkgrid Theme')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.show()
```



8.2 Color Palettes

Seaborn provides various color palettes to enhance the visual appeal of your plots.

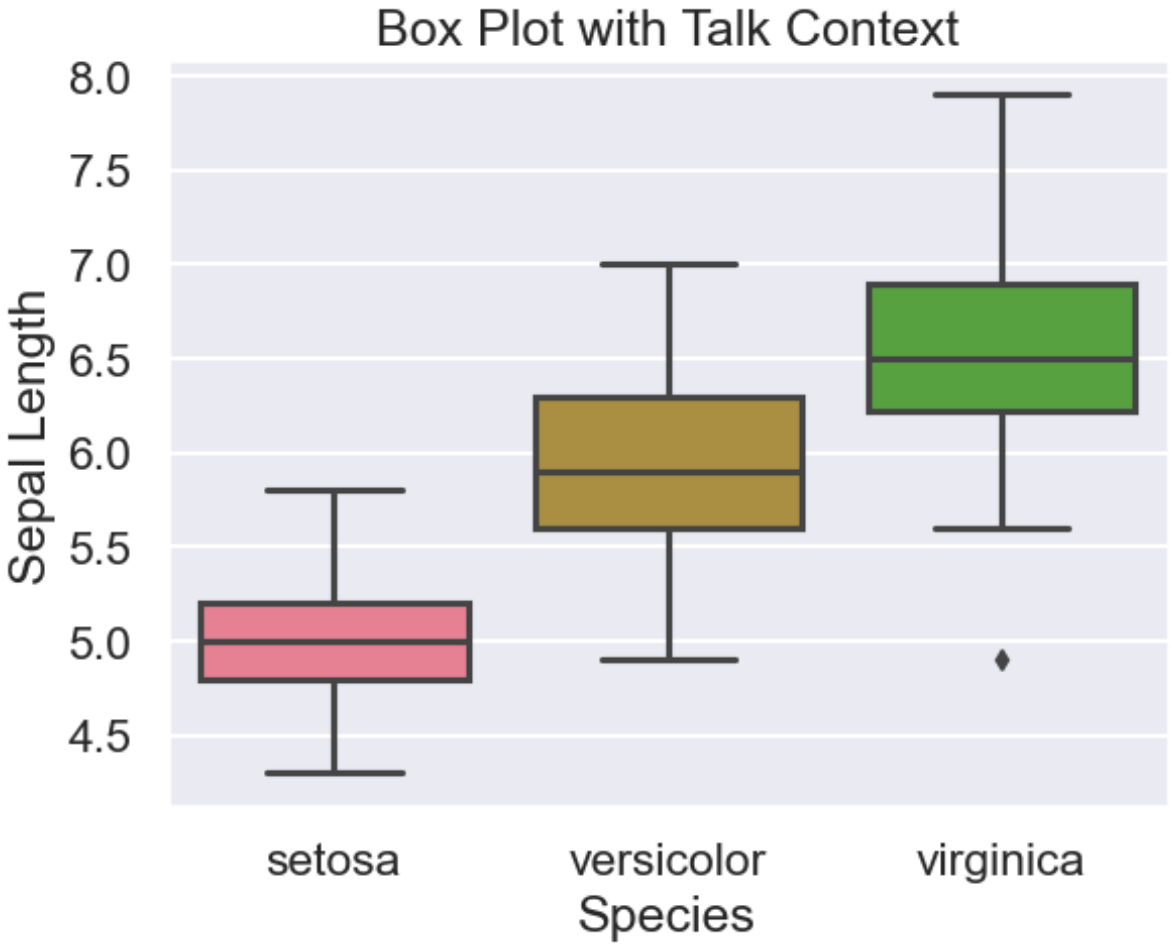
In [22]:
```
sns.set_palette('husl')
sns.barplot(x='species', y='sepal_length', data=data)
plt.title('Bar Plot with Husl Palette')
plt.xlabel('Species')
plt.ylabel('Sepal Length')
plt.show()
```



8.3 Contexts

Seaborn contexts allow you to customize the scaling of plot elements to improve readability.

In [23]:
```
sns.set_context('talk')
sns.boxplot(x='species', y='sepal_length', data=data)
plt.title('Box Plot with Talk Context')
plt.xlabel('Species')
plt.ylabel('Sepal Length')
plt.show()
```



9. Working with Inbuilt Datasets

Seaborn provides several built-in datasets for practice and demonstration purposes.

9.1 Tips Dataset

The Tips dataset contains information about tips received based on various factors.

In [24]:
```
tips = sns.load_dataset('tips')
print(tips.head())
```

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

9.2 Flights Dataset

The Flights dataset contains information about the number of passengers on flights over time.

In [25]:
```
flights = sns.load_dataset('flights')
print(flights.head())
```

```
   year month  passengers
0  1949   Jan         112
1  1949   Feb         118
2  1949   Mar         132
3  1949   Apr         129
4  1949   May         121
```

9.3 Iris Dataset

The Iris dataset contains information about the measurements of iris flowers.

In [26]:
```
iris = sns.load_dataset('iris')
print(iris.head())
```

```
   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
```

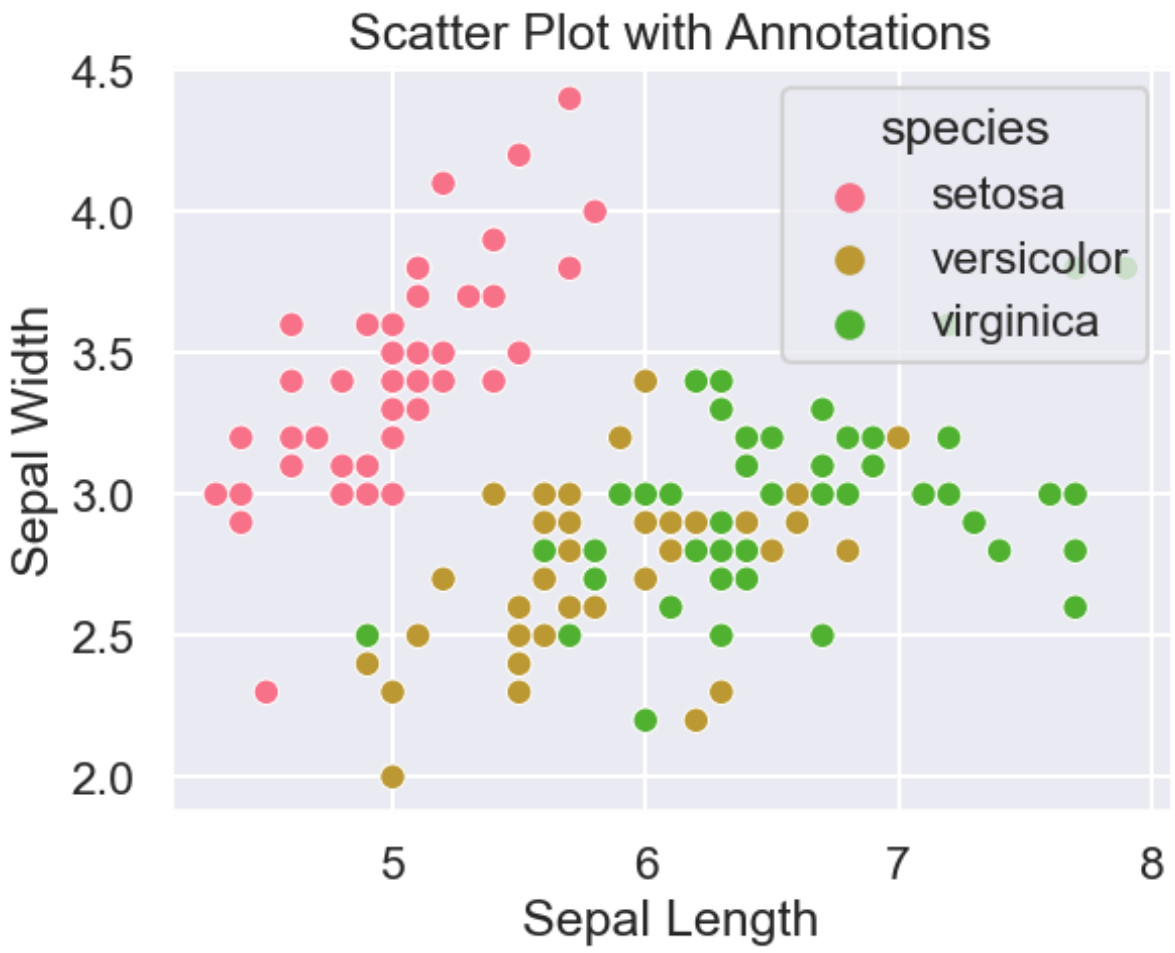
10. Advanced Customizations

Learn how to customize your plots to make them more informative and visually appealing.

10.1 Adding Annotations

You can add annotations to your plots to highlight important data points.

In [29]:
```
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', d
plt.title('Scatter Plot with Annotations')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
# for i in range(len(iris)):
#     plt.text(iris['sepal_length'][i], iris['sepal_width'][i], iri
plt.show()
```



10.2 Custom Legends

Customize the legends of your plots to improve clarity.

In [28]:
```
sns.lineplot(x='sepal_length', y='sepal_width', hue='species', data
plt.title('Line Plot with Custom Legends')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.legend(title='Species', loc='upper left', labels=['Setosa', 'Ve
plt.show()
```