- ## Introduction to Matplotlib and Seaborn:

Matplotlib:

Introduction: Matplotlib is a powerful plotting library for Python, widely used for creating static, animated, and interactive visualizations in a variety of formats.

Installation: Install Matplotlib using pip:

```
!pip install matplotlib
```

Unique Features: Highly customizable, integrates well with other libraries.

Use Cases: Suitable for simple, static charts and visualizations in reports and publications.

Seaborn:

Introduction: Built on top of Matplotlib, with a focus on statistical data visualization. Seaborn is a statistical data visualization library for Python, built on top of Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Installation: Install Seaborn using pip:

```
!pip install seaborn
```

Unique Features: Provides higher-level interfaces for drawing attractive, informative statistical graphics.

Use Cases: Useful for exploring relationships between data, especially in data analysis and machine learning projects.

- ## Graph Types

For each library, document common graph types like line plots, scatter plots, bar charts, histograms, and pie charts (if supported).

Example for Matplotlib and Seaborn:

1. **Matplotlib:**

    - **Scatter plots:**

      Description: A scatter plot displays points on a two-dimensional graph, representing the relationship between two numerical variables.
      Use Case: Ideal for visualizing correlations or relationships between variables, identifying trends, and spotting outliers.

Code Example:

In [1]:
```
import matplotlib.pyplot as plt
import seaborn as sas
Iris = sas.load_dataset('iris')
Iris
```

Out[1]:

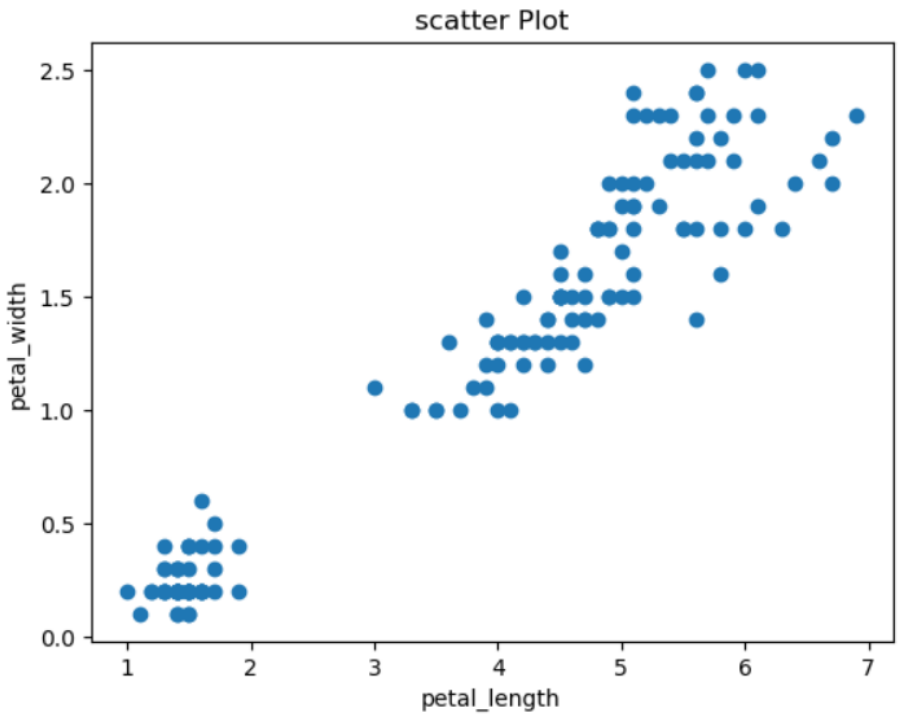
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [2]:
```
x = Iris['petal_length']
y = Iris['petal_width']
```

In [3]:
```
plt.title('scatter Plot')
plt.scatter(x,y)
plt.xlabel('petal_length')
plt.ylabel('petal_width')
```

Out[3]:
```
Text(0, 0.5, 'petal_width')
```



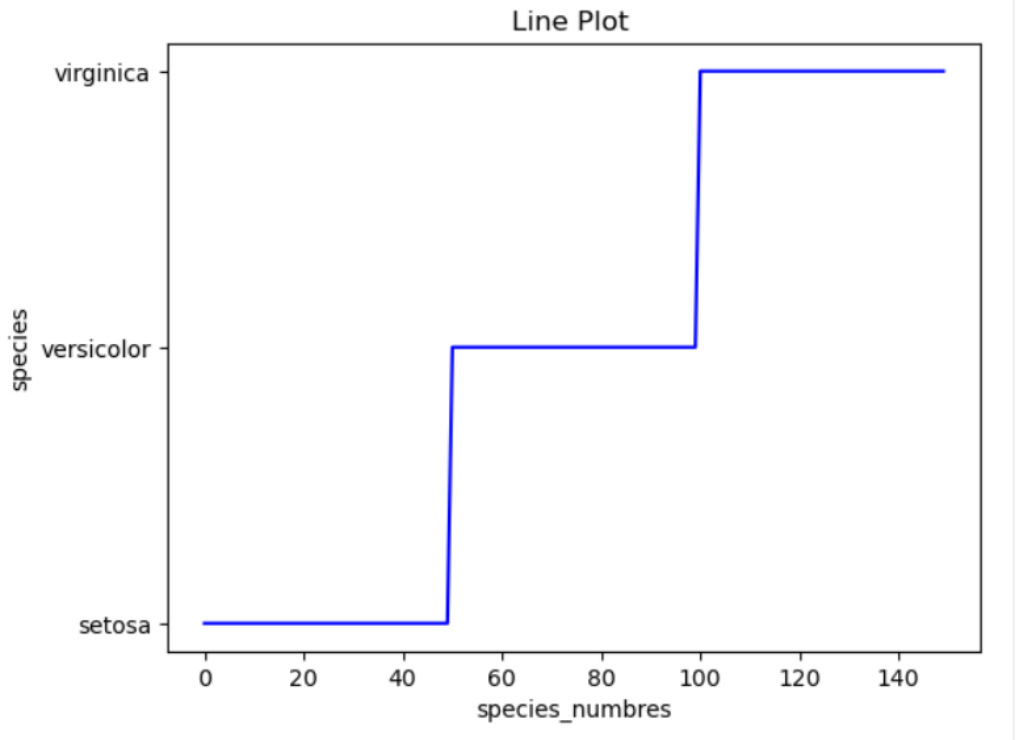
- **Line Plot:**

  Description: A plot showing data points connected by a line.

  Use Case: Useful for tracking changes over time.

  Code Example:

```
In [4]: species = Iris['species']
        plt.plot( species, color='b')
        plt.title('Line Plot ')
        plt.xlabel('species_numbres')
        plt.ylabel('species')
        plt.show()
```



- **Bar Chart:**

  Description: A chart with rectangular bars showing quantities.

  Use Case: Ideal for comparing categories.

  Code Example:

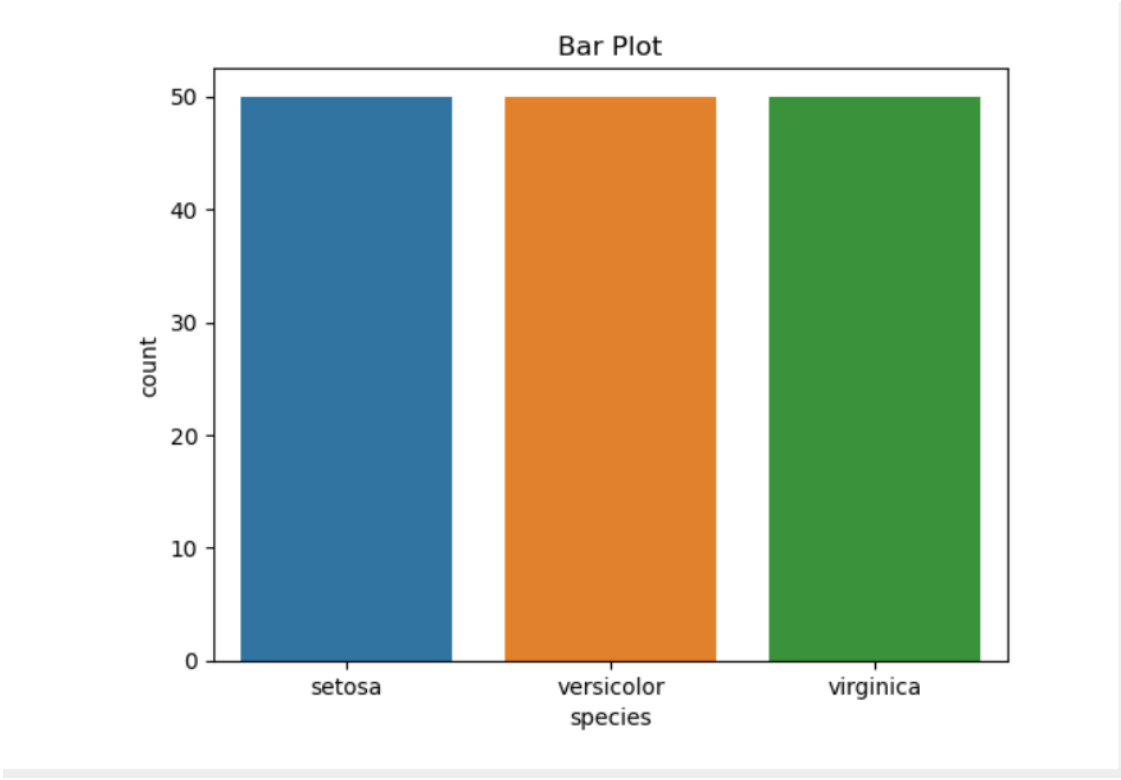
```
In [5]: print(Iris.shape)
        Iris['species'].value_counts()

        (150, 5)

Out[5]: species
        setosa        50
        versicolor    50
        virginica     50
        Name: count, dtype: int64
```

```
In [6]: sas.countplot(x='species', data=Iris)
        plt.title('Bar Plot')

Out[6]: Text(0.5, 1.0, 'Bar Plot')
```



- **Histogram:**

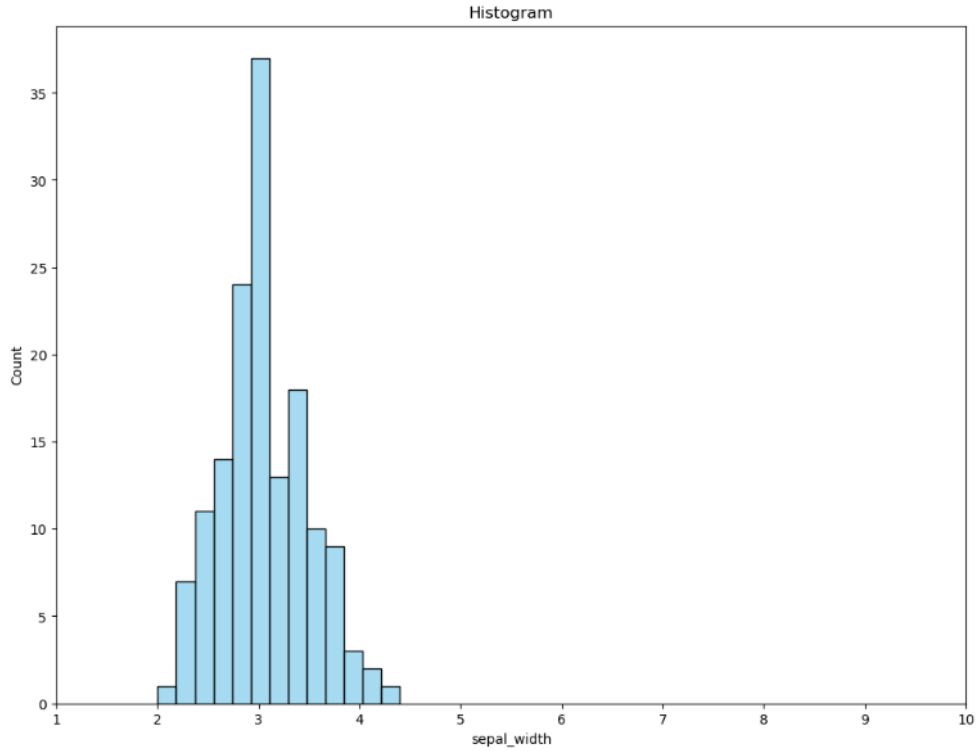
  Description: It is a graphical representation that organizes a group of data points into user-specified ranges. Each bar represents the frequency of data points within each range.

  Use Case: Best for visualizing the distribution of a single numerical variable, understanding data distribution, and identifying patterns such as skewness, modality, and outliers.

  Code Example:

```
In [3]: xdata = [1,2,3,4,5,6,7,8,9,10]
        plt.figure(figsize = (12,9))
        sas.histplot(Iris['sepal_width'], color='skyblue')
        plt.xticks(xdata)
        plt.title('Histogram')
```

```
Out[3]: Text(0.5, 1.0, 'Histogram')
```



2. Seaborn:

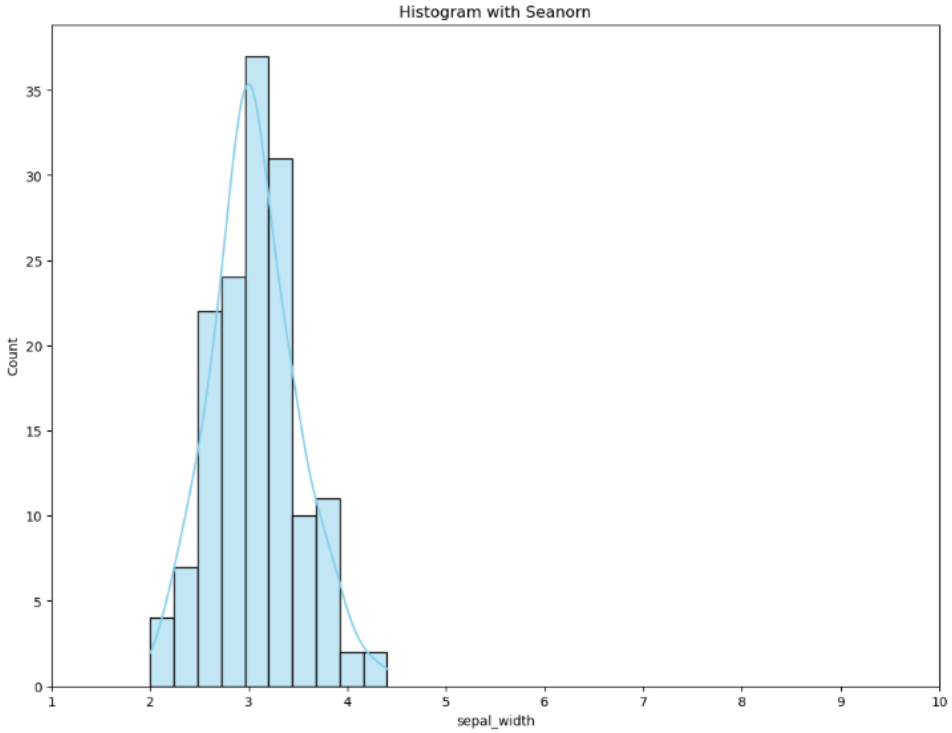
Description: A plot showing individual data points.

Use Case: Useful for examining relationships between variables.

Code Example:

```
In [11]: xdata = [1,2,3,4,5,6,7,8,9,10]
         plt.figure(figsize = (12,9))
         sas.histplot(Iris['sepal_width'], bins=10, kde=True, color='skyblue')
         plt.xticks(xdata)
         plt.title('Histogram with Seanorn')
```

```
Out[11]: Text(0.5, 1.0, 'Histogram with Seanorn')
```



- **Comparison**

Matplotlib:

Strengths:	Highly customizable, widely used, lots of support and documentation.
Weaknesses:	Static plots by default, limited interactivity without additional tools.

Seaborn:

Strengths:	Beautiful statistical plots, easy to use for data exploration.
Weaknesses:	Limited interactivity, depends on Matplotlib.

- **Resources**

Matplotlib Documentation: https://www.geeksforgeeks.org/python-introduction-matplotlib/

Seaborn Documentation: https://www.geeksforgeeks.org/introduction-to-seaborn-python/