



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

---



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

---

BARI ANKIT (56)

Experiment No.8
Data Visualization: Use different Python libraries for visualization.
Date of Performance:
Date of Submission:

**Experiment No - 8**

**Aim** - Data Visualization: Use Matplotlib and seabom Python libraries for visualization.



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

---

**Objective:-** To understand and apply the Matplotlib and seaborn python libraries for visualization using python.

### *Description-*

Data visualization is an easier way of presenting the data, however complex it is, to analyze trends and relationships amongst variables with the help of pictorial representation.

The following are the advantages of Data Visualization

- Easier representation of complex data
- Highlights good and bad performing areas
- Explores relationship between data points
- Identifies data patterns even for larger data points

While building visualization, it is always a good practice to keep some below mentioned points in mind

- Ensure appropriate usage of shapes, colors, and size while building visualization
- Plots/graphs using a co-ordinate system are more pronounced
- Knowledge of suitable plot with respect to the data types brings more clarity to the information
- Usage of labels, titles, legends and pointers passes seamless information to the wider audience

### Python Libraries

There are a lot of python libraries which could be used to build visualization like matplotlib, vispy, bokeh, seaborn, pygal, folium, plotly, cufflinks, and networkx. Of the many, matplotlib and seaborn seem to be very widely used for basic to intermediate level of visualizations.

### Matplotlib



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

---

It is an amazing visualization library in Python for 2D plots of arrays, It is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. Let's try to understand some of the benefits and features of matplotlib

- It's fast, efficient as it is based on numpy and also easier to build
- Has undergone a lot of improvements from the open source community since inception and hence a better library having advanced features as well
- Well maintained visualization output with high quality graphics draws a lot of users to it
- Basic as well as advanced charts could be very easily built
- From the users/developers point of view, since it has a large community support, resolving issues and debugging becomes much easier

### Seaborn

Conceptualized and built originally at the Stanford University, this library sits on top of matplotlib. In a sense, it has some flavors of matplotlib while from the visualization point, it is much better than matplotlib and has added features as well. Below are its advantages

- Built-in themes aid better visualization
- Statistical functions aiding better data insights
- Better aesthetics and built-in plots
- Helpful documentation with effective examples



### Nature of Visualization

Depending on the number of variables used for plotting the visualization and the type of variables, there could be different types of charts which we could use to understand the relationship. Based on the count of variables, we could have

- Univariate plot(involved only one variable)
- Bivariate plot(more than one variable is required)

A Univariate plot could be for a continuous variable to understand the spread and distribution of the variable while for a discrete variable it could tell us the count

Similarly, a Bivariate plot for continuous variable could display essential statistics like correlation, for a continuous versus discrete variable could lead us to very important conclusions like understanding data distribution across different levels of a categorical variable. A bivariate plot between two discrete variables could also be developed.

### Scatter Plot

Scatter plots or scatter graphs is a bivariate plot having greater resemblance to line graphs in the way they are built. A line graph uses a line on an X-Y axis to plot a continuous function, while a scatter plot relies on dots to represent individual pieces of data. These plots are very useful to see if two variables are correlated. Scatter plot could be 2 dimensional or 3 dimensional.

Syntax: `seaborn.scatterplot(x=None, y=None, hue=None, style=None, size=None, data=None, palette=None, hue_order=None, hue_norm=None, sizes=None, size_order=None, size_norm=None, markers=True, style_order=None, x_bins=None, y_bins=None, units=None, estimator=None, ci=95, n_boot=1000, alpha='auto', jitter=None, yjitter=None, legend='brief', ax=None, **kwargs)`  
Parameters:

`x, y`: Input data variables that should be numeric. `data`: Dataframe where each column is a variable and each row is an observation. `size`: Grouping variable that will produce points with different sizes. `style`: Grouping variable that will produce points with different markers. `palette`: Grouping variable that will produce points with different markers. `markers`: Object determining how to draw the markers for different levels. `alpha`: Proportional opacity of the points.

Returns: This method returns the Axes object with the plot drawn onto it.



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

---

Histograms display counts of data and are hence similar to a bar chart. A histogram plot can also tell us how close a data distribution is to a normal curve. While working out statistical method, it is very important that we have a data which is normally or close to a normal distribution. However, histograms are univariate in nature and bar charts bivariate.

A bar graph charts actual counts against categories e.g. height of the bar indicates the number of items in that category whereas a histogram displays the same categorical variables in bins.

Bins are integral part while building a histogram they control the data points which are within a range. As a widely accepted choice we usually limit bin to a size of 5-20, however this is totally governed by the data points which is present.

### Countplot

A countplot is a plot between a categorical and a continuous variable. The continuous variable in this case being the number of times the categorical is present or simply the frequency. In a sense, count plot can be said to be closely linked to a histogram or a bar graph.

Syntax : `seaborn.countplot(x=None , y=None , hue=None , data=None, order=None, hue_order=None , orient=None, color=None, palette=None , saturation=0.75 , dodge=True , ax=None, **kwargs)`

Parameters : This method is accepting the following parameters that are described below:

- **x, y:** This parameter take names of variables in data or vector data, optional, Inputs for plotting long-form data.
- **hue :** (optional) This parameter take column name for colour encoding.
- **data :** (optional) This parameter take DataFrame , array, or list of arrays, Dataset for plotting.

If x and y are absent, this is interpreted as wide-form. Otherwise it is expected to be long-form.



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

---

- `order, hue_order` : (optional) This parameter take lists of strings. Order to plot the categorical levels in, otherwise the levels are inferred from the data objects.
- `orient` : (optional) This parameter take "v" | "h", Orientation of the plot (vertical or horizontal). This is usually inferred from the dtype of the input variables but can be used to specify when the "categorical" variable is a numeric or when plotting wide-form data.
- `color` : (optional) This parameter take matplotlib color, Color for all of the elements, or seed for a gradient palette.
- `palette` : (optional) This parameter take palette name, list, or dict, Colors to use for the different levels of the hue variable. Should be something that can be interpreted by `color_palette()`, or a dictionary mapping hue levels to matplotlib colors. Program 1- Scatter plot (create or upload any suitable data set)

Code :

```
# Import required libraries
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
import pandas as pd
```

```
# Create a sample dataset
```

```
data = {  
    'Height': [5.1, 5.5, 6.0, 5.8, 5.7, 5.4, 6.1],  
    'Weight': [50, 70, 80, 75, 60, 65, 85],  
    'Species': ['A', 'B', 'A', 'B', 'A', 'B', 'A']  
}
```

```
df = pd.DataFrame(data)
```

```
# Create a scatter plot using Seaborn
```

```
plt.figure(figsize=(10, 6))
```

```
sns.scatterplot(x='Height', y='Weight', hue='Species', data=df, palette='viridis', s=100)
```



```
plt.title('Scatter Plot of Height vs Weight')
```

```
plt.xlabel('Height (in feet)')
```

```
plt.ylabel('Weight (in kg)')
```

```
plt.legend(title='Species')
```

```
plt.grid()
```

```
plt.show()
```

```
# Import required libraries
```

```
import matplotlib.pyplot as plt
```

```
# Sample data
```

```
labels = ['Sedan', 'SUV', 'Hatchback', 'Coupe']
```

```
sizes = [30, 40, 20, 10] # percentage values
```

```
colors = ['gold', 'lightcoral', 'lightskyblue', 'yellowgreen']
```

```
explode = (0.1, 0, 0, 0) # explode 1st slice
```

```
# Create a pie chart
```

```
plt.figure(figsize=(8, 8))
```

```
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
```

```
    autopct='%1.1f%%', shadow=True, startangle=140)
```

```
plt.title('Car Distribution by Types')
```

```
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
```

```
plt.show()
```



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

---



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

---

**Output:**

Differences Between Matplotlib and Seaborn: Matplotlib is more general-purpose with more control over the visualizations, whereas Seaborn is specifically designed for statistical graphics and offers a more aesthetically pleasing default style.

Statistical Graphics Library: Seaborn is primarily used for creating statistical graphics in Python.

Creating a Histogram in Seaborn: The function used to create a histogram in Seaborn is `sns.histplot()` or `sns.histogram()`.