Python Libraries : Matplot Lib

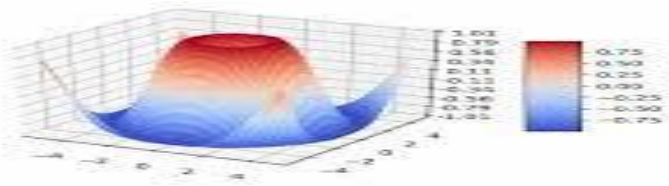
Agenda

- Introduction to MatplotLib
- Graphical representation of data
- PyPlot-API

What is Matplotlib?

- Matplotlib is one of the most popular Python packages used for data visualization.
- **Matplotlib** tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code.
- It is a cross-platform library for making 2D plots from data in arrays.
- Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python.

Key Features

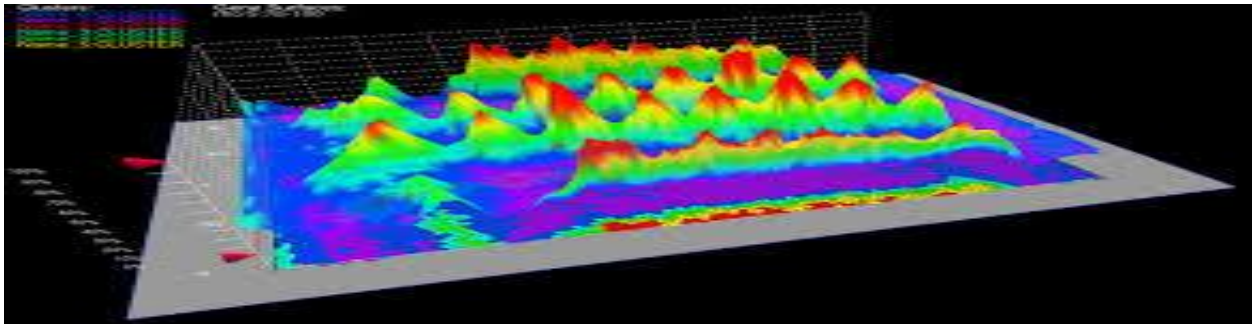


Matplotlib is an excellent 2D and 3D graphics library for generating scientific figures. Some of the many advantages of this library include:

- Easy to get started & Support for LATEX formatted labels and texts
- Great control of every element in a figure, including figure size and DPI.
- High-quality output in many formats, including PNG, PDF, SVG, EPS, and PGF.
- GUI for interactively exploring figures *and* support for headless generation of figure files (useful for batch jobs).

- *Programmatically* controlled features- important for reproducibility and convenient when one needs to regenerate the figure with updated data & appearance.

What is Data Visualization?



- Data visualization refers to techniques used to communicate insights from data through visual representation.

- Its main goal is to distill large datasets into visual graphics to allow for easy understanding of complex relationships within the data.

- Data visualization is both an art and a science.

Environment Setup

matplotlib

INTERNSHIPSTUDIO

To get matplotlib up and lets need to import it.

import **matplotlib.pyplot** as plt

- It is common practice to import matplotlib under the alias **plt —** a short cut.

- Whenever you plot with matplotlib, the two main code lines should be,

  - Type of graph — this is where you define a bar chart, line chart, etc.

  - Show the graph — this is to display the graph.

Graphical Representation

matplotlib

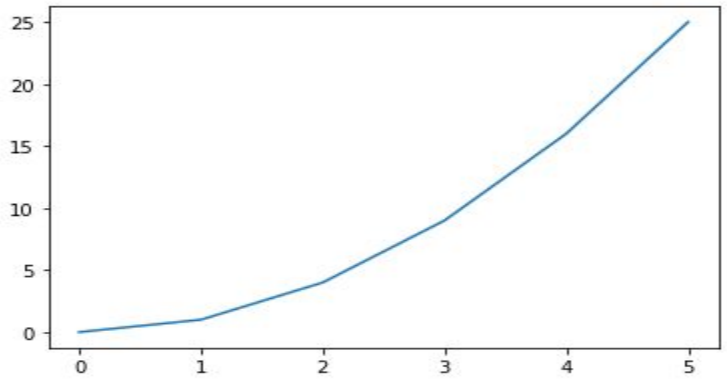
INTERNSHIPSTUDIO

Line Graphs

```
#create data for plotting
x_values = [0,1,2,3,4,5]
squares = [0,1,4,9,16,25]

#the default graph style for plot is a line
plt.plot(x_values, squares)

#display the graph
plt.show
```



Bar graphs

matplotlib

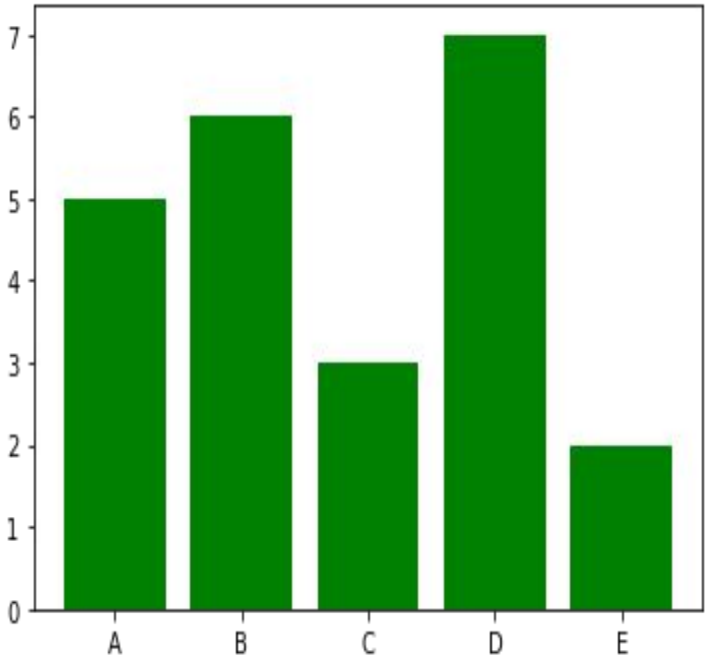
INTERNSHIPSTUDIO

```
#create data for plotting
x_values = [5,6,3,7,2]
y_values = ["A", "B", "C", "D", "E"]


plt.bar(y_val,x_values, color = "green")
plt.show()
```

- When using a bar graph, the change in code will be from *plt.plot()* to *plot.bar()* changes it into a bar chart.

- If you look inside the body of the code, I also added an argument **color** — this helps us quickly customize the color of the graph.



Graphical Representation

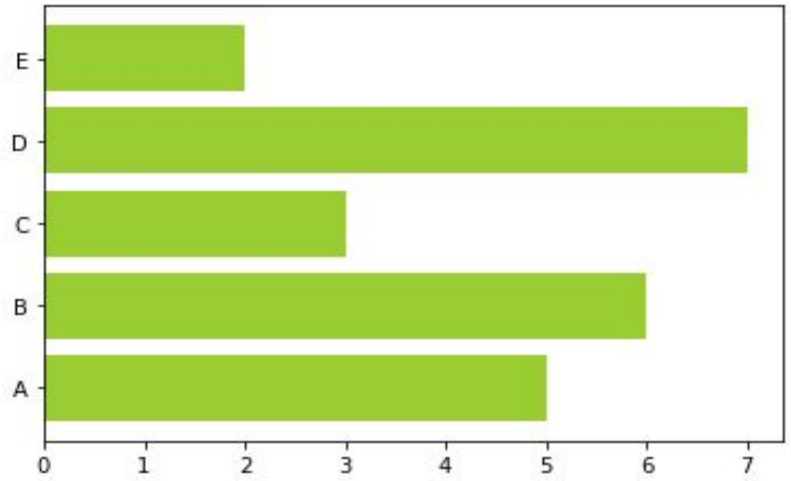
matplotlib

INTERNSHIPSTUDIO

We can also flip the bar graph horizontally with the following,

```
#create data for plotting
x_values = [5,6,3,7,2]
y_val = ["A", "B", "C", "D", "E"]


# Adding an "h" after bar will flip the graph
plt.barh(y_val,x_values, color ="yellowgreen")
plt.show()
```



Notice the change in the color argument

Scatter Plots

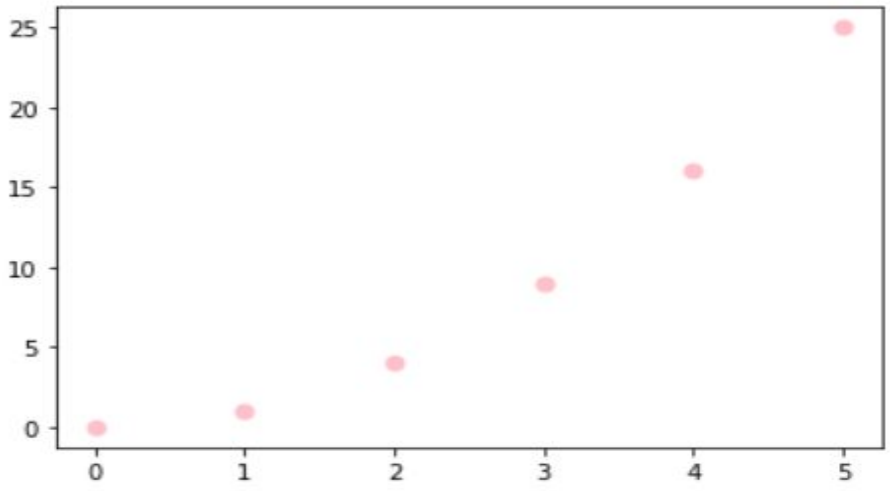
matplotlib

INTERNSHIPSTUDIO

```
#create data for plotting
x_values = [0,1,2,3,4,5]
squares = [0,1,4,9,16,25]


plt.scatter(x_values,squares, s=10, color = "pink")
plt.show()
```

- Can you see the pattern? Now the code changed from *plt.bar()* to *plt.scatter().*
- *A*lso added the **s** argument. The s stands for size, and it allows us to control size of the points on the graph.



matplotlib

INTERNSHIPSTUDIO

Q.1 What is Data Visualization?

Q.2 What is Matplotlib?

Q.3 How can we import Matplotlib?

Q.4 How can we draw Line Graphs using Matplotlib?

Q.5 How can we draw Bar Graphs using Matplotlib?