SESSION 7

MATPLOTLIB

1. What is matplotlib?

- Matplotlib is a Python library that helps you create different types of graphs and charts.
- Matplotlib is an amazing visualization library in Python for 2D plots of arrays.
- Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

Usage of matplotlib:

- Matplotlib is often used in machine learning applications to visualize data and model predictions, helping researchers and engineers to evaluate the performance of their models.
- Matplotlib is widely used in education to teach students about data visualization.
- Scientists use Matplotlib to visualize data from experiments and simulations, allowing them to present their findings in a clear and concise manner.
- Matplotlib is commonly used to visualize and explore data, helping researchers and analysts to understand trends, patterns, and relationships between variables.

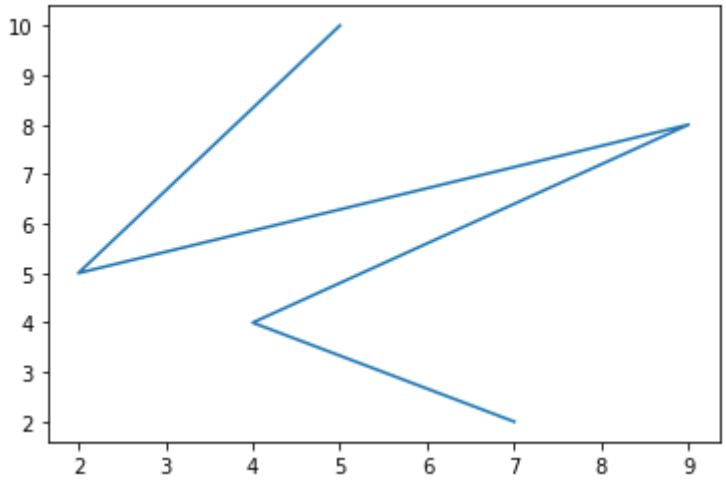
Importing matplotlib

In [ ]:
```
# Importing matplotlib :

from matplotlib import pyplot as plt
# or
import matplotlib.pyplot as plt
```

Line plot

- It also known as a line graph, is a type of graph that displays data points as a series of connected line segments.

In [ ]:
```
# Line plot :

# importing matplotlib module
from matplotlib import pyplot as plt

# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot
plt.plot(x,y)

# function to show the plot
plt.show()
```

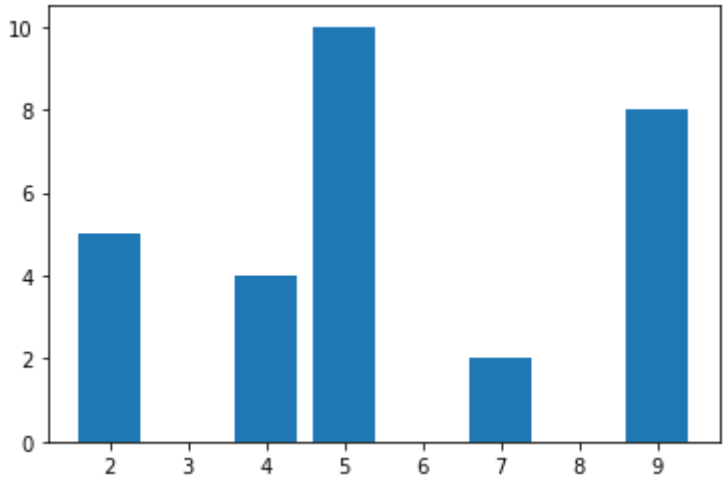


Bar plot

- It is also known as a bar chart, is a type of graph that uses rectangular bars to represent the data values.

In [ ]:

```
#Bar plot :

# importing matplotlib module
from matplotlib import pyplot as plt

# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot the bar
plt.bar(x,y)

# function to show the plot
plt.show()
```

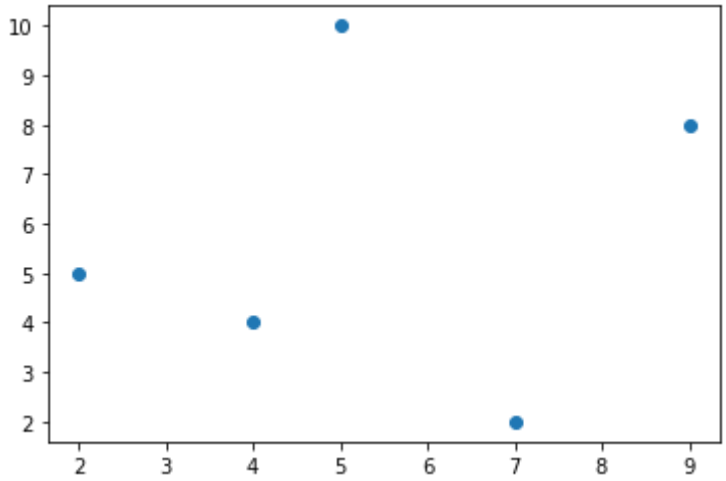


Scatter Plot

- It is also known as a scatter chart, is a type of graph that uses dots to represent individual data points.

In [ ]:

```
# Scatter
# importing matplotlib module
from matplotlib import pyplot as plt

# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot scatter
plt.scatter(x, y)

# function to show the plot
plt.show()
```



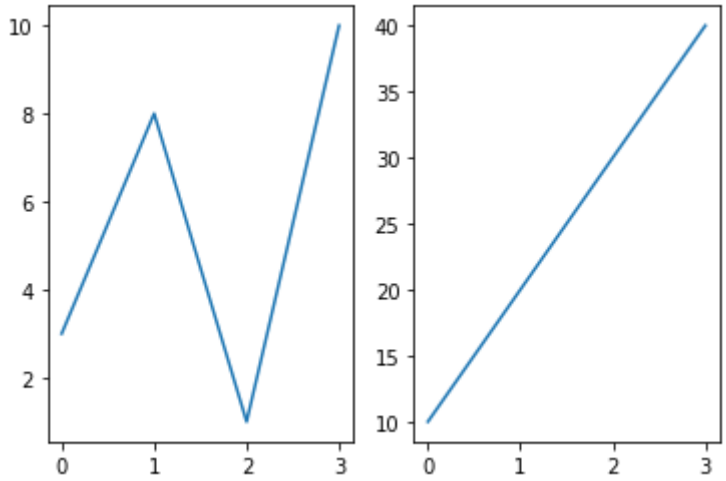
---

2. Display Multiple Plots

With the `subplot()` function you can draw multiple plots in one figure:

In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)

plt.show()
```

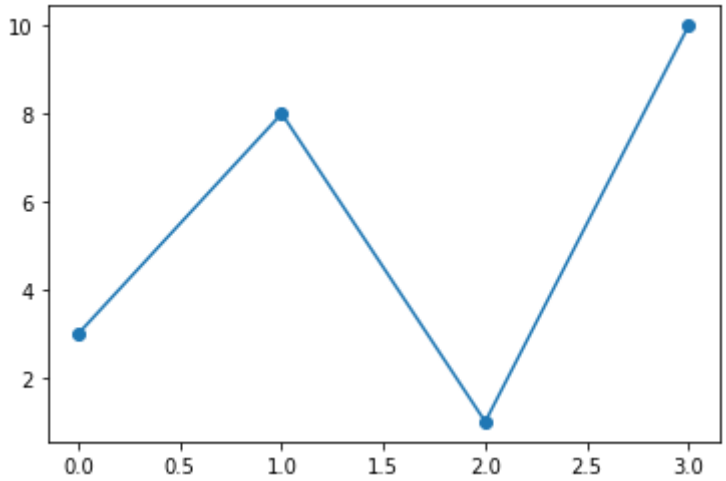


Matplotlib Markers

- Markers are symbols used to represent individual data points in a plot.
- Markers are often used in scatter plots to distinguish between different data points.

In [ ]:

```
# Matplotlib Markers
import matplotlib.pyplot as plt
import numpy as np

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = 'o')
plt.show()
```



3. Matplotlib Pie Charts

With Pyplot, you can use the pie() function to draw pie charts:

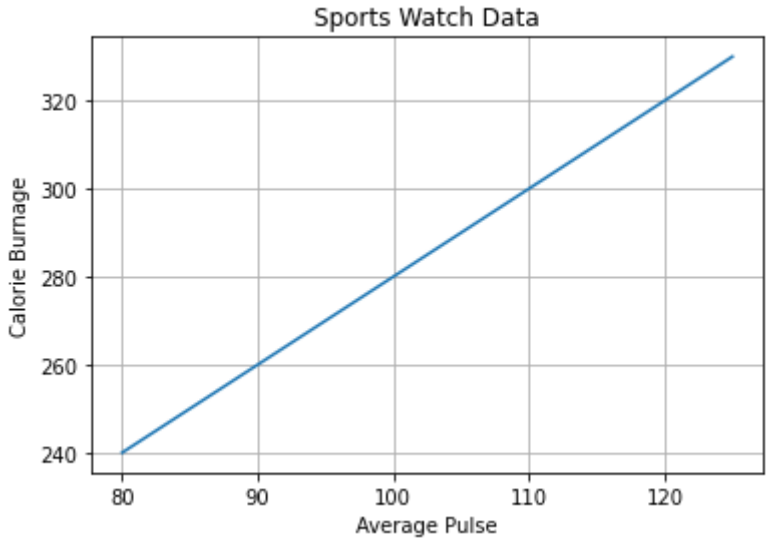
In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np

y = np.array([35, 25, 25, 15])

plt.pie(y)
plt.show()
```

4. Matplotlib Adding Grid Lines

In [ ]:

```
# Matplotlib Adding Grid Lines
import numpy as np
import matplotlib.pyplot as plt

x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.title("Sports Watch Data")
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")

plt.plot(x, y)

plt.grid()

plt.show()
```



5. Matplotlib Subplot

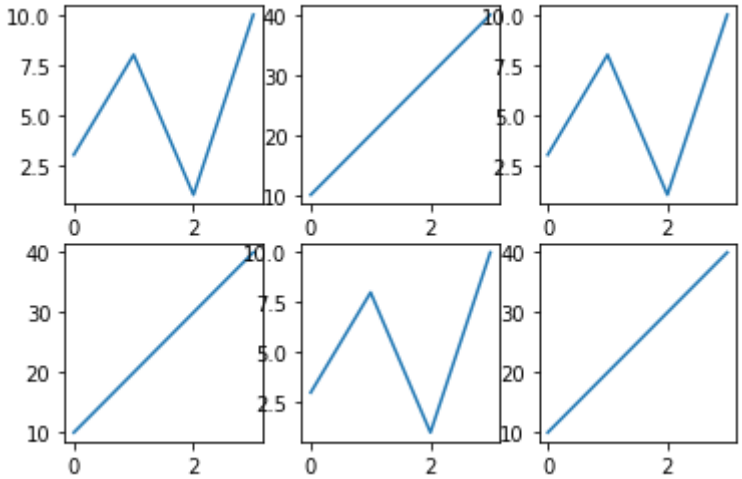
- *Subplot is a way to create multiple plots within the same figure. A figure can contain one or more subplots arranged in a grid of rows and columns.*
- *Matplotlib provides the feature to create a figure with multiple plots in a single call, with proper control over each plot in the figure*

5.1 Draw Multiple plots:

- You can draw as many plots you like on one figure, just descibe the number of rows, columns, and the index of the plot.

In [ ]:
```
# Drawing 6 plots in one figure
import matplotlib.pyplot as plt
import numpy as np

x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 3, 1)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])
```

```
plt.subplot(2, 3, 2)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 3, 3)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 3, 4)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 3, 5)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 3, 6)
plt.plot(x,y)

plt.show()
```



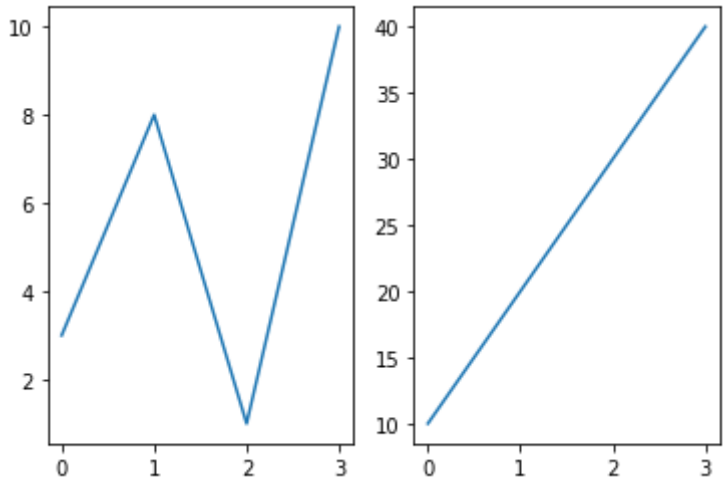
5.2 Subplot function Arguments:

- The `subplot()` function takes three arguments that describes the layout of the figure.
- The layout is organized in rows and columns, which are represented by the first and second argument.
- The third argument represents the index of the current plot.
  Plot 1 - subplot(1, 2, 1) In the case of `plt.subplot(1, 2, 1)` argument, the figure has 1 row, 2 columns, and this plot is the first plot.
  Plot 2 - subplot(1, 2, 2) The `plt.subplots(1, 2, 2)` the figure has 1 row, 2 columns, and this plot is the second plot.

In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

# In the case of the (1, 2, 1) argument, the figure has 1 row, 2 columns, and this plot is the first plot.
plt.subplot(1, 2, 1)
plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])
```

```
# In the case of the (1, 2, 2) argument, the figure has 1 row, 2 columns, and this plot is the second plot.
plt.subplot(1, 2, 2)
plt.plot(x,y)

plt.show()
```



5.3 matplotlib `add_subplot()` function:

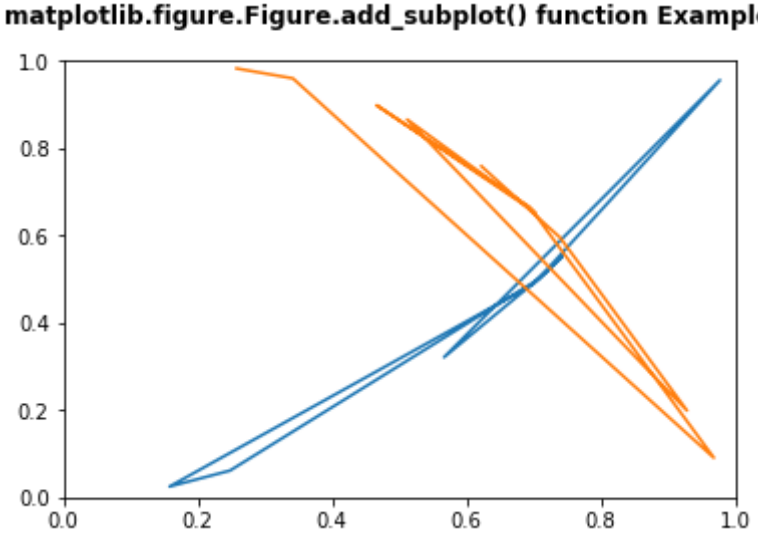
The figure module provides the top-level Artist, the Figure, which contains all the plot elements. This module is used to control the default spacing of the subplots and top level container for all plot elements.

matplotlib.figure.Figure. `add_subplot()` function:
The `add_subplot()` method figure module of matplotlib library is used to add an Axes to the figure as part of a subplot arrangement.

In [ ]:

```
# Implementation of matplotlib function
import matplotlib.pyplot as plt
import numpy as np


np.random.seed(19680801)

xdata = np.random.random([2, 10])
```

```
xdata1 = xdata[0, :]
xdata2 = xdata[1, :]

ydata1 = xdata1 ** 2
ydata2 = 1 - xdata2 ** 3

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.plot(xdata1, ydata1, color ='tab:blue')
ax.plot(xdata2, ydata2, color ='tab:orange')

ax.set_xlim([0, 1])
ax.set_ylim([0, 1])
fig.suptitle('matplotlib.figure.Figure.add_subplot() function Example\n\n', fontweight ="bold")

plt.show()
```

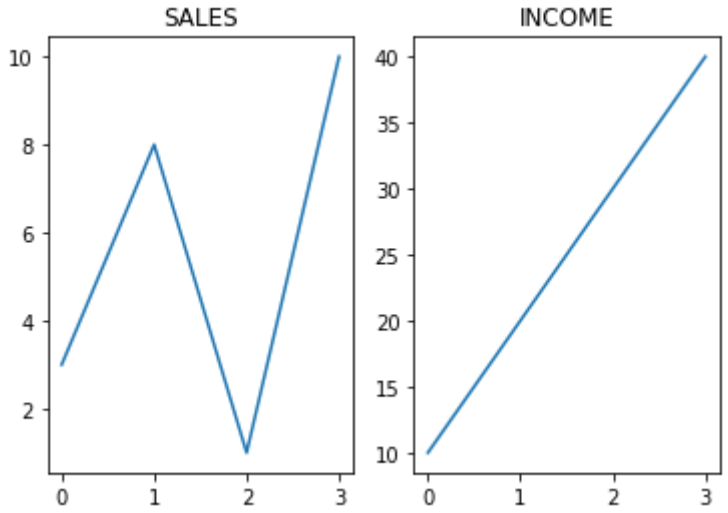


6. Title

- Title is a text label that is displayed at the top of a plot to describe the content of the plot or convey important information about the data being displayed.
- You can add a title to each plot with the title() function.

In [ ]:

```
# Adding Title 'INCOME' in Matplotlib
import matplotlib.pyplot as plt
import numpy as np

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
plt.plot(x,y)
plt.title("SALES")

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title("INCOME")

plt.show()
```

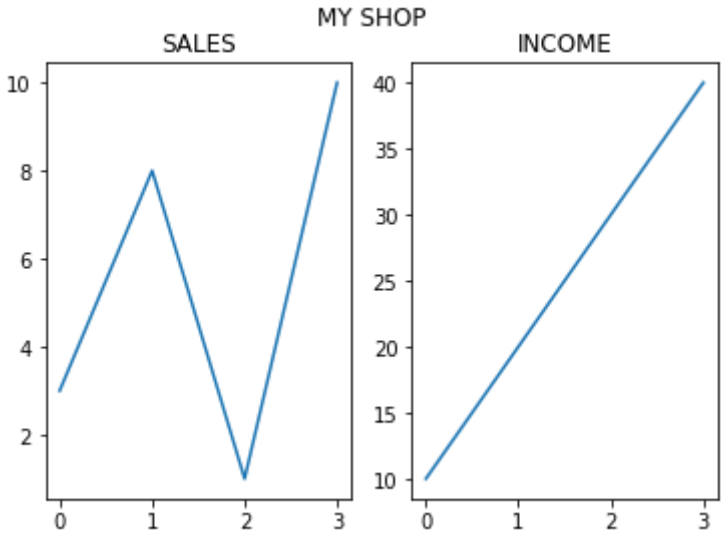


6.1. Super Title

- You can add a title to the entire figure with the suptitle() function

In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
plt.plot(x,y)
plt.title("SALES")

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title("INCOME")

plt.suptitle("MY SHOP")
plt.show()
```

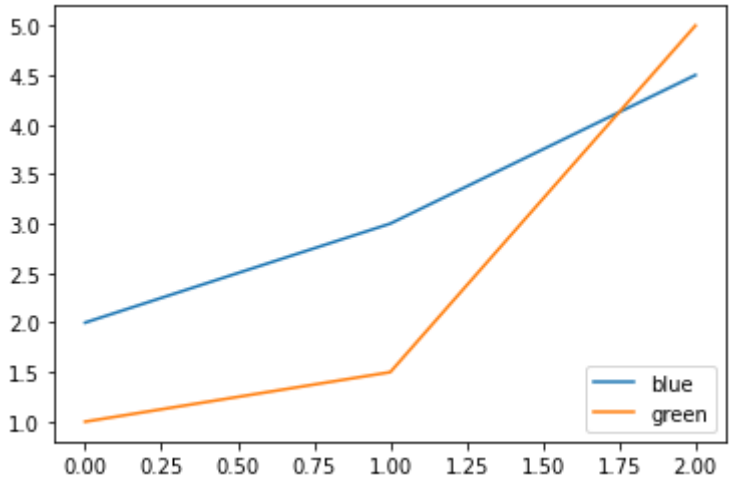


7. Legend function in Matplotlib

- A legend is an area describing the elements of the graph. In the matplotlib library, there's a function called `legend()` which is used to Place a legend on the axes.
- The attribute Loc in `legend()` is used to specify the location of the legend.Default value of loc is loc="best" (upper left). The strings 'upper left', 'upper right', 'lower left', 'lower right' place the legend at the corresponding corner of the axes/figure.
- The attribute `bbox_to_anchor=(x, y)` of `legend()` function is used to specify the coordinates of the legend, and the attribute ncol represents the number of columns that the legend has.It's default value is 1.

In [ ]:

```
# 2 lines
# importing modules
import numpy as np
import matplotlib.pyplot as plt

# Y-axis values
y1 = [2, 3, 4.5]

# Y-axis values
y2 = [1, 1.5, 5]

# Function to plot
plt.plot(y1)
plt.plot(y2)

# Function add a legend
plt.legend(["blue", "green"], loc ="lower right")

# function to show the plot
plt.show()
```

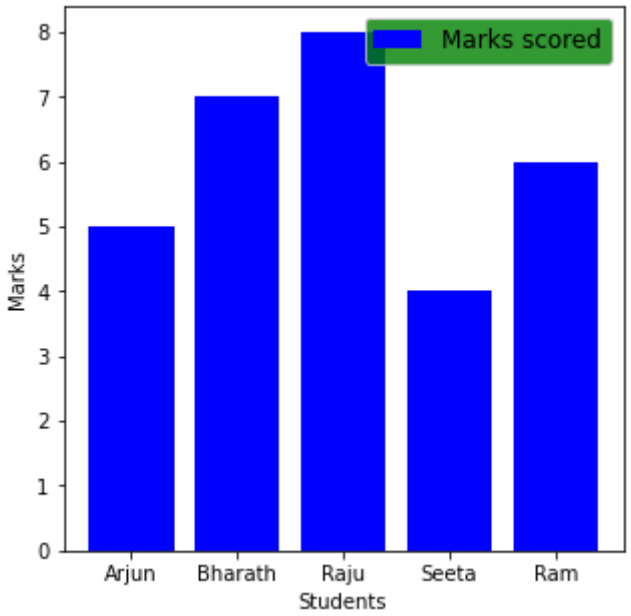


Change Legend Font Size

- Here, we are trying to change the font size of the x and y labels.

In [ ]:

```
# Change Font Size
import matplotlib.pyplot as plt
import numpy as np

plt.figure(figsize = (5, 5))

x = ['Arjun', 'Bharath', 'Raju', 'Seeta', 'Ram']
y = [5, 7, 8, 4, 6]

plt.bar(x, y, color = 'b')

plt.xlabel('Students', fontsize = 10)
plt.ylabel('Marks', fontsize = 10)

#Default fontsize of text using legend
plt.legend(['Marks scored'],fontsize=12,facecolor='green')

plt.show()
```



Homework Questions

1) Make a Matplotlib for below array with `Review and TRP's` of Broadcasting shows, by Adding Grid Lines

```
x : ([87, 81, 19, 95, 10, 105, 1100, 15, 12, 125])
y : ([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])
```

2) Create pie charts using Pyplot library, using the `pie()` function

3) Create Matplotlib Markers with star using Pyplot library

4) Display Multiple Plots for

plot 1: x = np.array([0, 1, 2, 3]) y = np.array([23, 48, 19, 70])

plot 2: x = np.array([0, 1, 2, 3]) y = np.array([10, 20, 30, 40])

5) Make a scatter plot for the given array

```
# x-axis values
x = [15, 32, 79, 94, 77]

# Y-axis values
y = [10, 95, 48, 49, 27]
```

6) Make a Bar plot for the following array

```
# x-axis values
x = [95, 62, 91, 45, 77]

# Y-axis values
y = [10, 25, 84, 49, 27]
```

7) Display Line plot for the below arrays

```
x = [12,23,45,67]
y = [74,12,56,36]
```

8) Display Multiple Plots for the below arrays

```
### plot 1:
x = np.array([10, 12, 25, 73])
y = np.array([43, 48, 29, 40])


### plot 2:
x = np.array([50, 71, 82, 73])
y = np.array([10, 20, 40, 60])
```

9) Display Bar plot for the below arrays

```
x = [25, 32, 49, 74, 87]
y = [10, 25, 68, 54, 72]
```

10) Create a subplot (1, 2, 1) & (1, 2, 2) for the below arrays

```
plot 1:
x = np.array([80, 19, 72, 93])
y = np.array([37, 89, 19, 10])

plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])
```

11) Create a subplot and add subtitle for

```
plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])
plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])
```

12) Create a subplot,subtitle,title and font size

```
plot 1:
a = np.array([29,23,45,76])
b = np.array([21,82,44,65])
plot 2:
x = np.array([102,230,453,564])
y = np.array([10, 20, 30, 40])
```

13) Add legend for

```
#X-axis values
x = [19, 22, 34, 49, 57]

#Y-axis values
y = [14, 84, 29, 16, 25]
```

For solutions of Homework questions, please refer to the `HomeworkSolution.ipynb` file