Matplotlib

Introduction

- Matplotlib is a plotting library for the Python programming language and its extension NumPy.

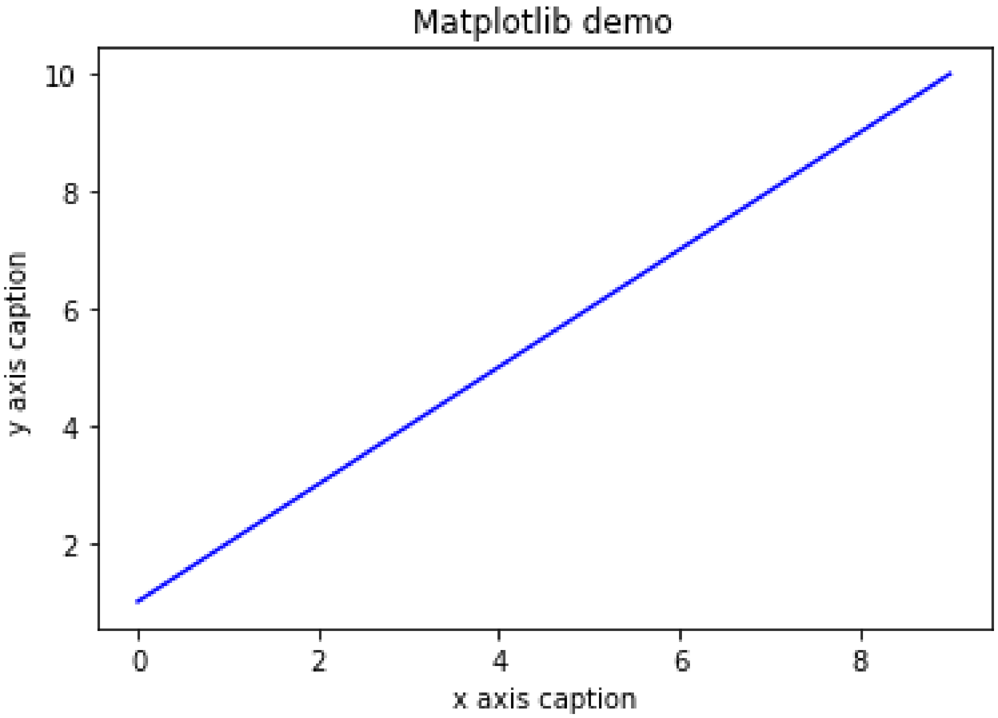
- import matplotlib.pyplot as plt

OR

- from matplotlib import pyplot as plt

Line Chart

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

x = np.arange(0, 10)
y = x + 1

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")

plt.plot(x, y, linestyle='-', color='b')

plt.show()
```



Color

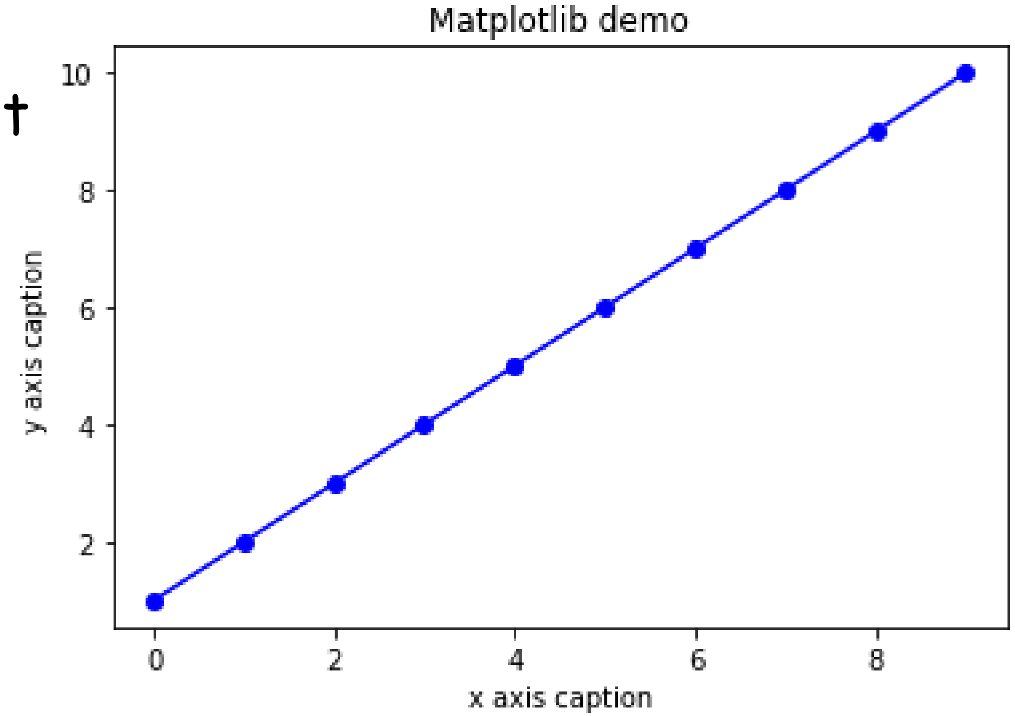
Character	Color
'b'	Blue
'g'	Green
'r'	Red
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'w'	White

Plot’s linestyle

linestyle	description
'-' or 'solid'	solid line
'--' or 'dashed'	dashed line
'-.' or 'dashdot'	dash-dotted line
':' or 'dotted'	dotted line
'None'	draw nothing
' '	draw nothing
''	draw nothing

Line Chart with marker

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

x = np.arange(0, 10)
y = x + 1

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")

plt.plot(x, y, linestyle='-‘, color=‘b’, marker=‘o’)

plt.show()
```



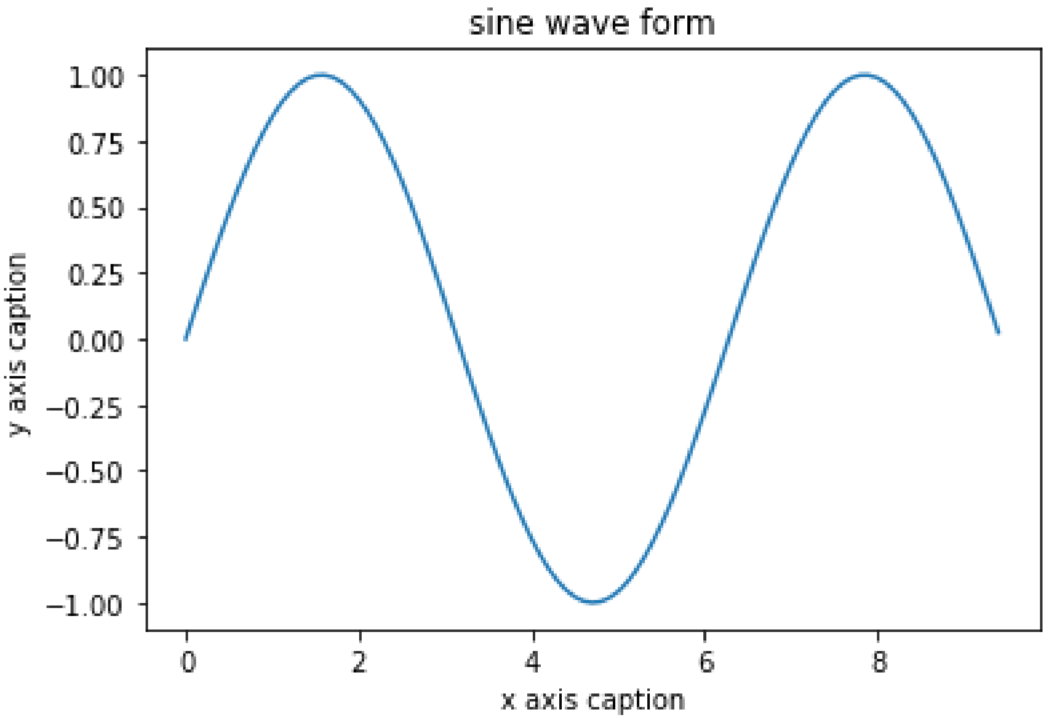
Note: In the above code, making linestyle=‘ ’, we will have scatter plot.

Plot's Marker

Sr. No.	Character & Description
1	'.' Point marker
2	',' Pixel marker
3	'o' Circle marker
4	'v' Triangle_down marker
5	'^' Triangle_up marker
6	'<' Triangle_left marker
7	'>' Triangle_right marker
8	'1' Tri_down marker
9	'2' Tri_up marker
10	'3' Tri_left marker
11	'4' Tri_right marker
12	's' Square marker
13	'p' Pentagon marker
14	'*' Star marker
15	'h' Hexagon1 marker
16	'H' Hexagon2 marker
17	'+' Plus marker
18	'x' X marker
19	'D' Diamond marker
20	'd' Thin_diamond marker
21	'|' Vline marker
22	'_' Hline marker

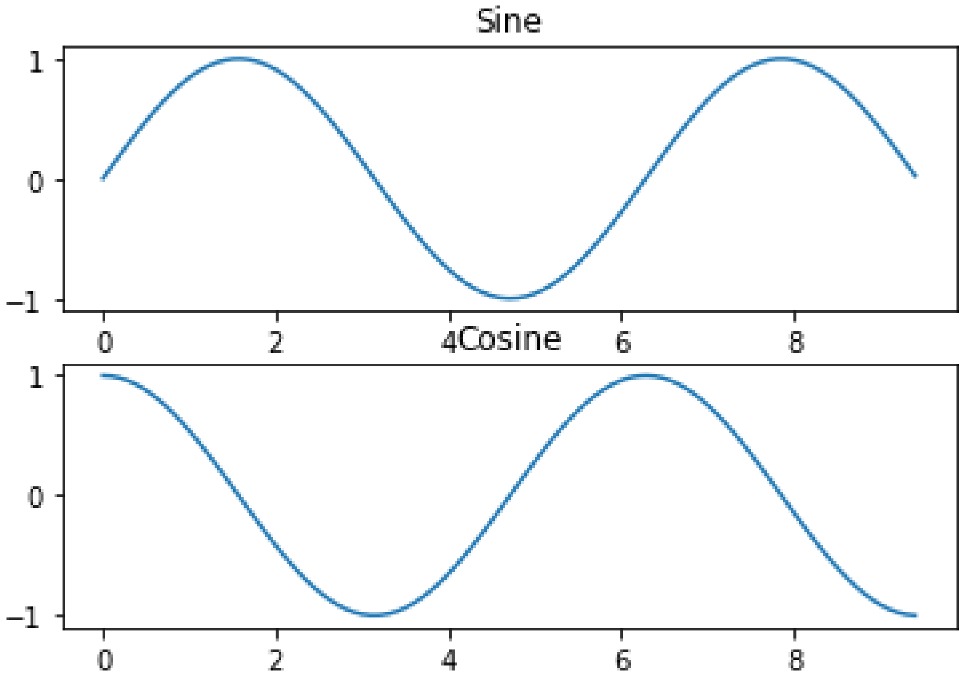
Sine Wave

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

# Compute the x and y coordinates
# for points on a sine curve
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)

plt.title("sine wave form")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")

plt.plot(x,y)
plt.show()
```



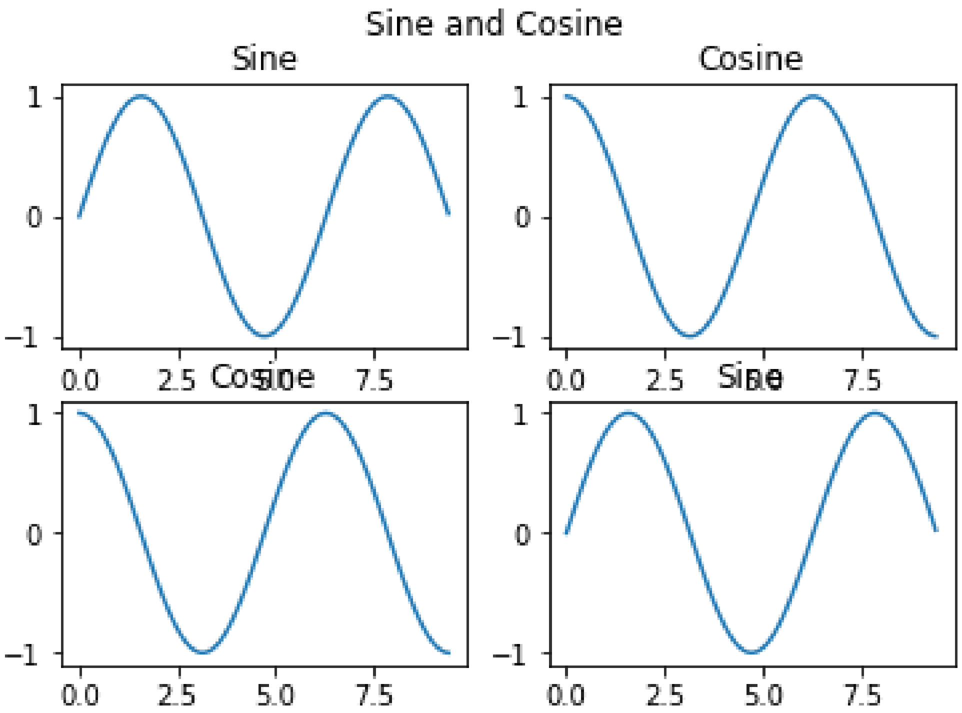
Subplot

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

x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)

plt.subplot(2, 1, 1)

# Make the first plot
plt.plot(x, y_sin)
plt.title('Sine')

# Set the second subplot as active, and # make the second plot.
plt.subplot(2, 1, 2)
plt.plot(x, y_cos)
plt.title('Cosine')

# Show the figure.
plt.show()
```



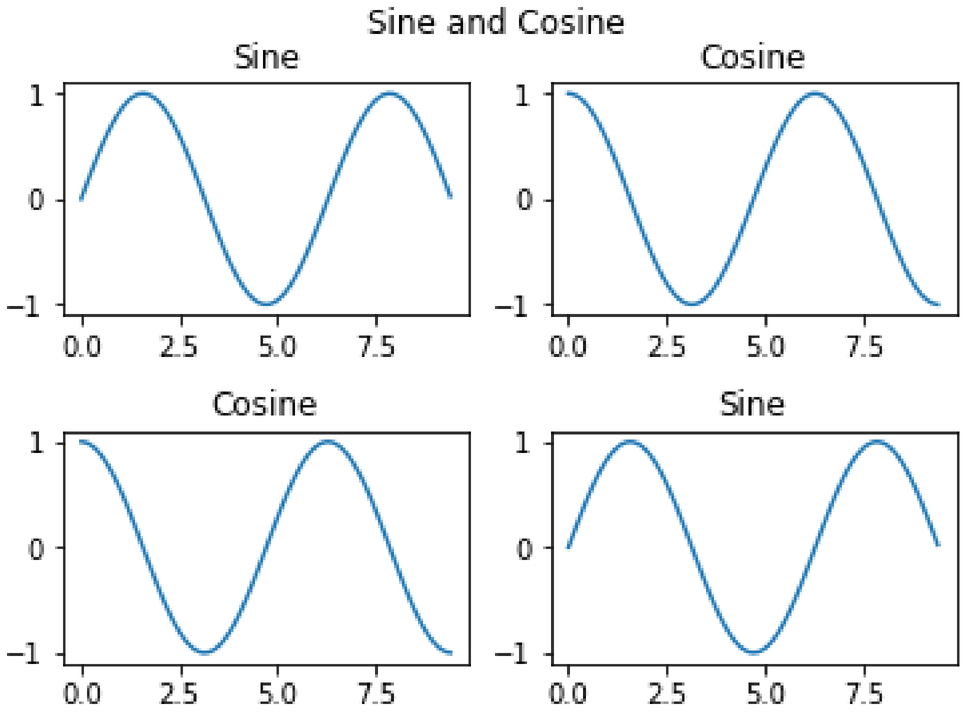
Subplot with suptitle

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

x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)

plt.subplot(2, 2, 1)
plt.plot(x, y_sin)
plt.title('Sine')

plt.subplot(2, 2, 2)
plt.plot(x, y_cos)
plt.title('Cosine')

plt.subplot(2, 2, 3)
plt.plot(x, y_cos)
plt.title('Cosine')

plt.subplot(2, 2, 4)
plt.plot(x, y_sin)
plt.title('Sine')

plt.suptitle('Sine and Cosine')

plt.show()
```



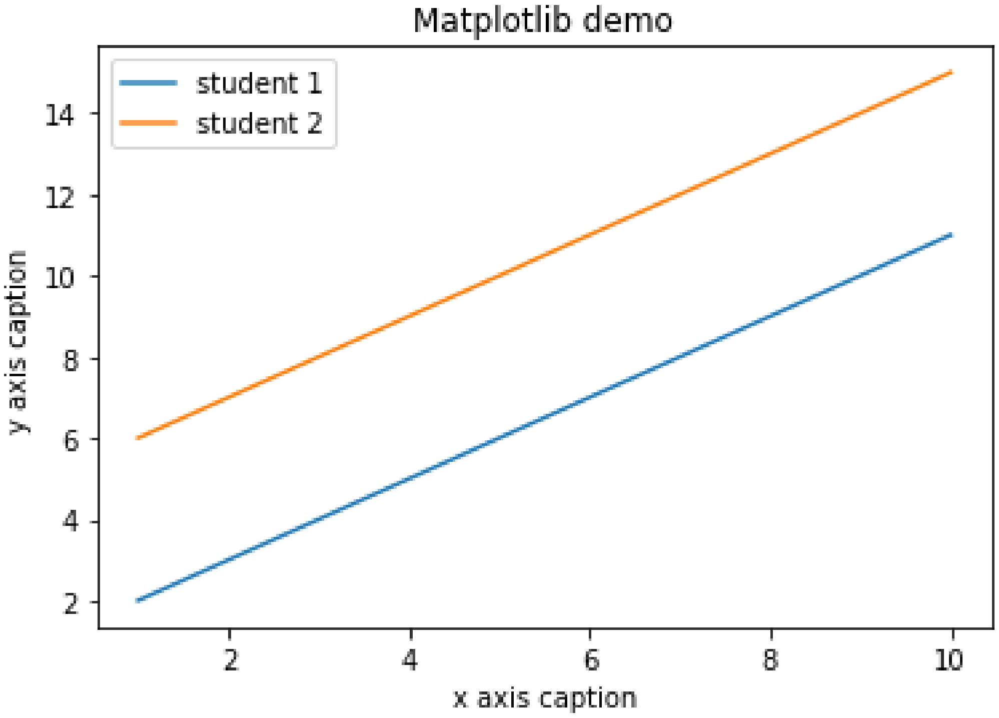
Subplot with hspace

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)
plt.subplot(2, 2, 1)
plt.plot(x, y_sin)
plt.title('Sine')
plt.subplot(2, 2, 2)
plt.plot(x, y_cos)
plt.title('Cosine')
plt.subplot(2, 2, 3)
plt.plot(x, y_cos)
plt.title('Cosine')
plt.subplot(2, 2, 4)
plt.plot(x, y_sin)
plt.title('Sine')
plt.subplots_adjust(hspace=0.5)     #wspace
plt.suptitle('Sine and Cosine')
plt.show()
```



Legend

```
x = np.arange(1,11)
y1 = x + 1
y2 = x + 5

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")

plt.plot(x,y1,label="student 1")
plt.plot(x,y2,label="student 2")

plt.legend(loc=2)

plt.show()
```

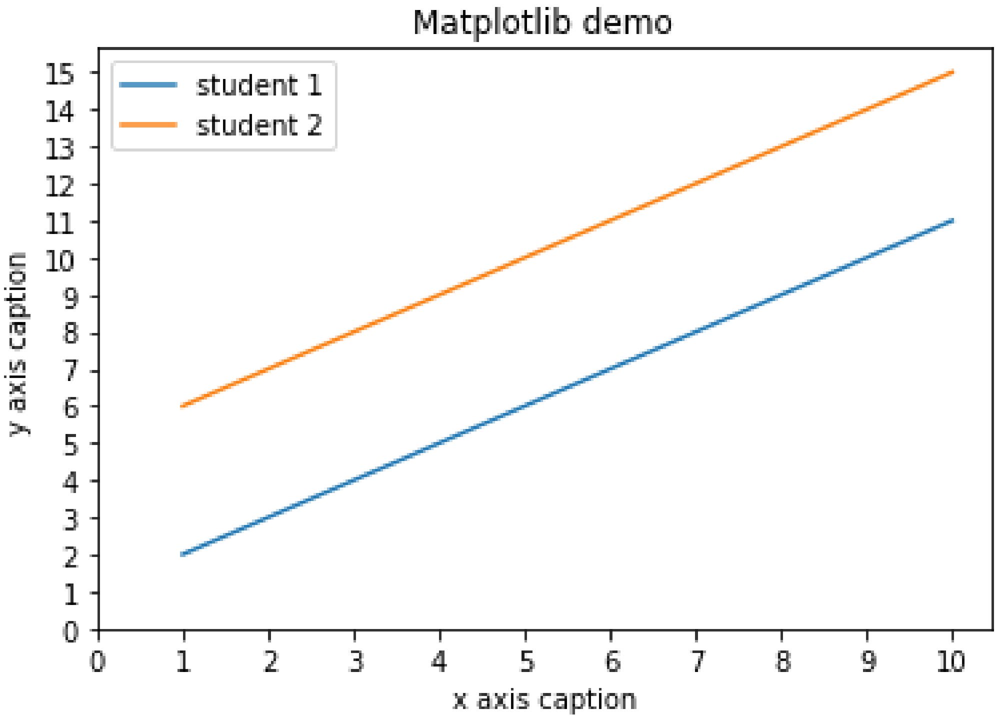


Legend-loc

Location String	Location Code
‘best’	0
‘upper right’	1
‘upper left’	2
‘lower left’	3
‘lower right’	4
‘right’	5
‘center left’	6
‘center right’	7
‘lower center’	8
‘upper center’	9
‘center’	10

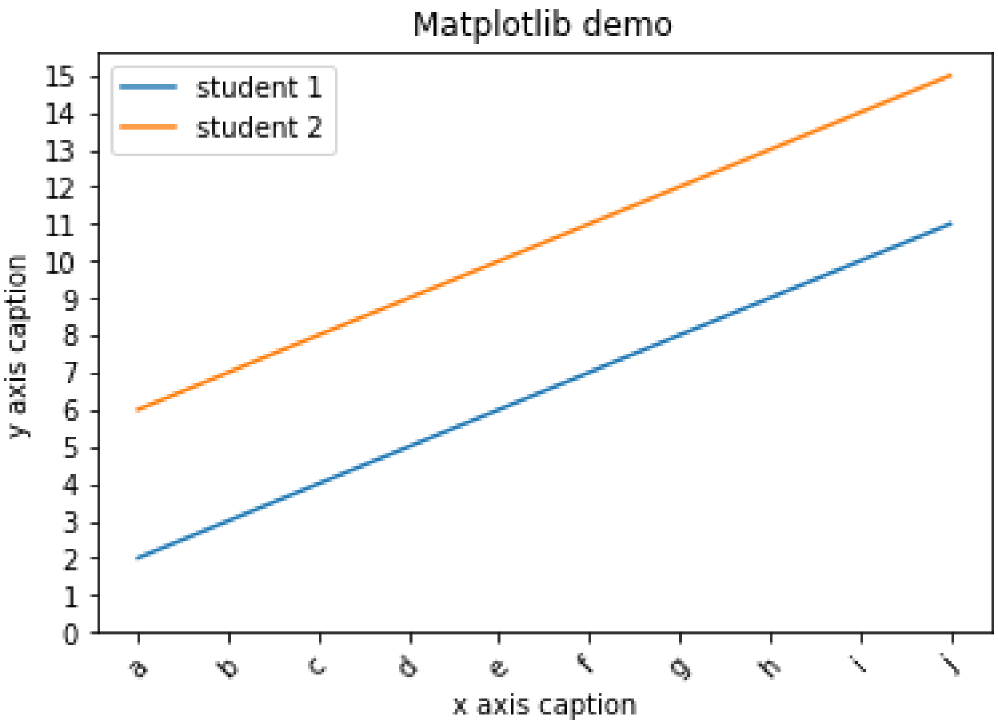
Ticks

```
x = np.arange(1,11)
y1 = x + 1
y2 = x + 5

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")

plt.plot(x,y1,label="student 1")
plt.plot(x,y2,label="student 2")

plt.legend(loc=2)

plt.xticks(np.arange(11))
plt.yticks(np.arange(16))

plt.show()
```



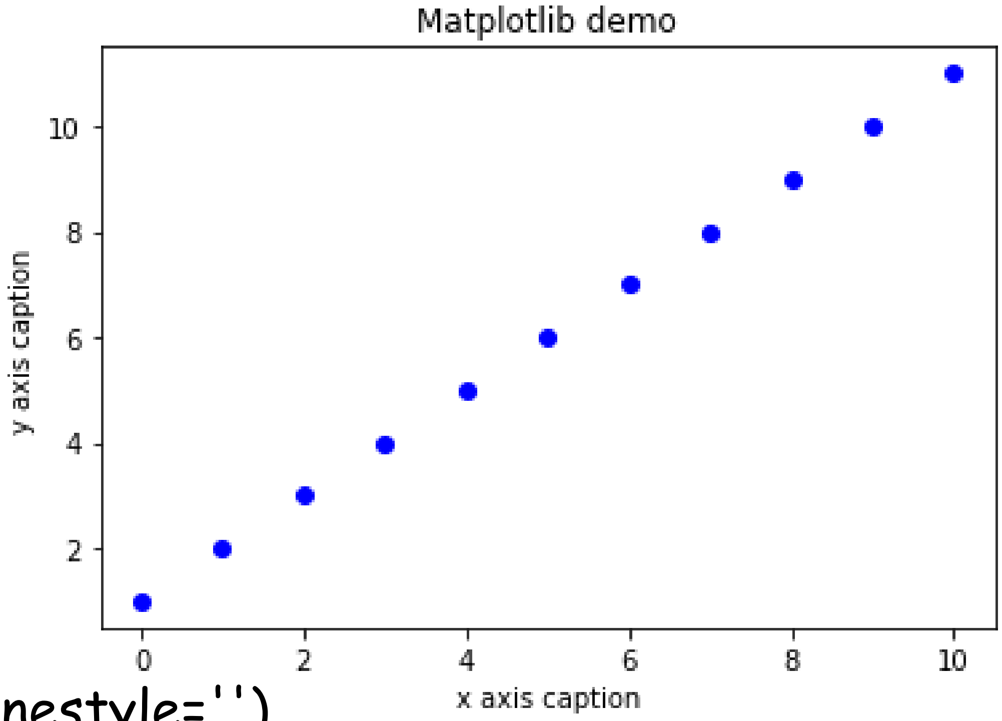
Ticks - User Defined

```
x = np.arange(1,11)
y1 = x + 1
y2 = x + 5

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")

plt.plot(x,y1,label="student 1")
plt.plot(x,y2,label="student 2")

plt.legend(loc=2)

plt.xticks(np.arange(1,11),['a','b','c','d'
,'e','f','g','h','i','j'], rotation=45)

plt.yticks(np.arange(16))

plt.show()
```



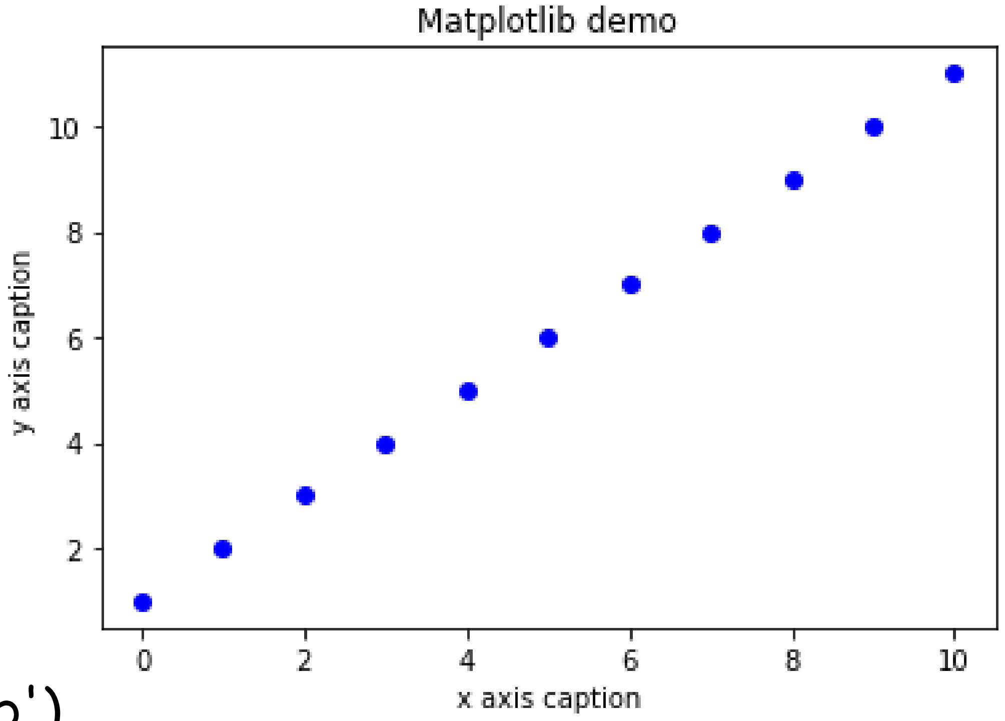
Scatter Plot with Plot

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

x = np.arange(0,11)
y = x + 1

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")

plt.plot(x, y, marker='o', color='b' , linestyle='')

plt.show()
```



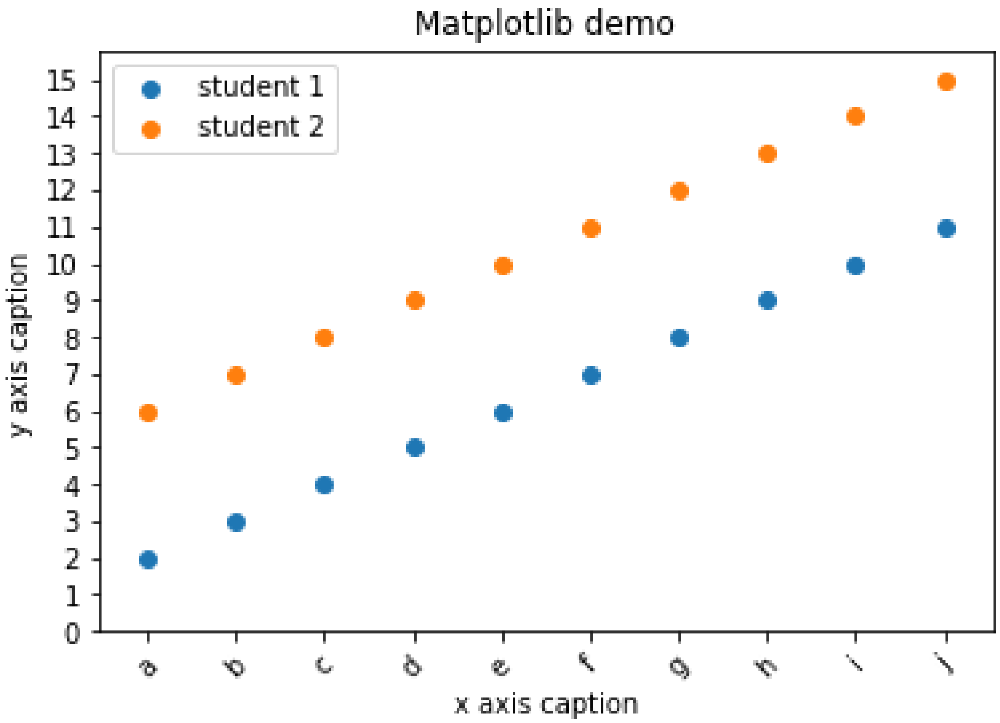
Scatter Plot with plt.scatter

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

x = np.arange(0,11)
y = x + 1

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")

plt.scatter(x, y, marker='o', color='b')

plt.show()
```



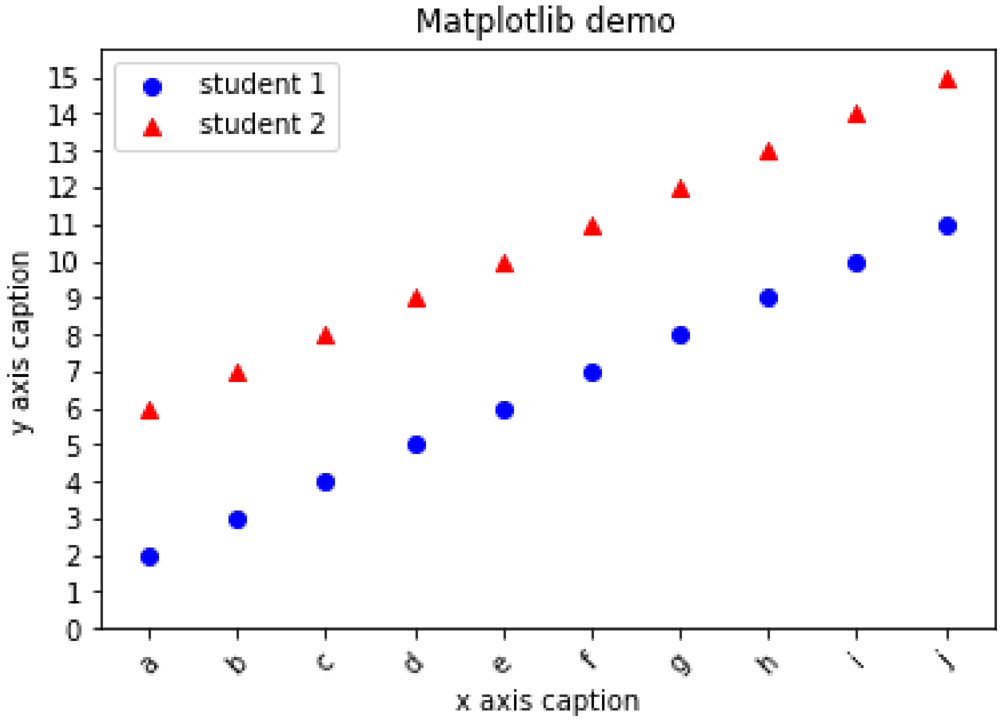
Scatter Plot with plt.scatter

```
x = np.arange(1,11)
y1 = x + 1
y2 = x + 5

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")

plt.scatter(x, y1, label="student 1")
plt.scatter(x, y2, label="student 2")

plt.legend(loc=2)

plt.xticks(np.arange(1, 11), ['a','b','c','d','e','f','g','h','i','j'], rotation=45)

plt.yticks(np.arange(16))

plt.show()
```



Scatter Plot with plt.scatter

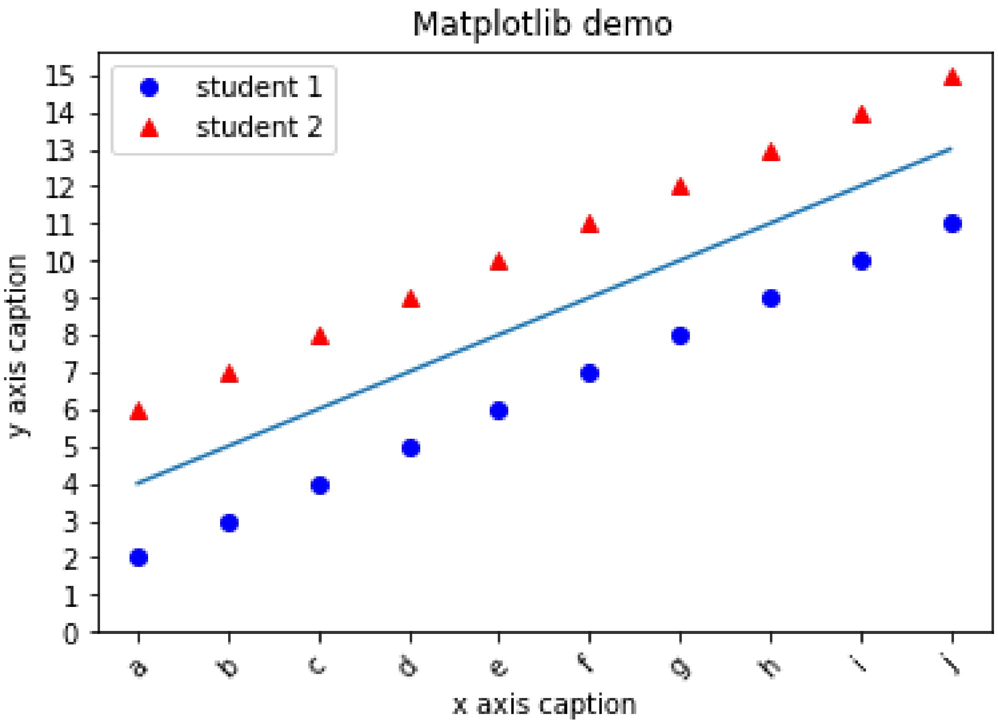
```
x = np.arange(1,11)
y1 = x + 1
y2 = x + 5

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
```



```
plt.scatter(x, y1, label="student 1", marker='o', color='b')
plt.scatter(x, y2, label="student 2", marker='^', color='r')

plt.legend(loc=2)

plt.xticks(np.arange(1, 11), ['a','b','c','d','e','f','g','h','i','j'], rotation=45)

plt.yticks(np.arange(16))

plt.show()
```

Scatter Plot and Line Plot with Plot

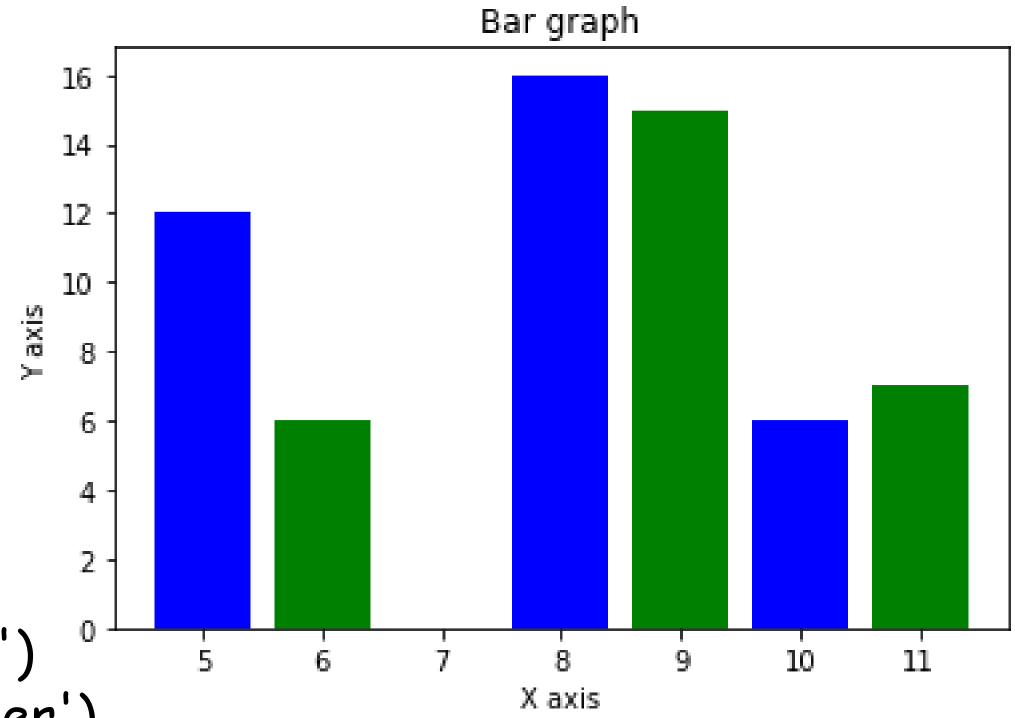
```
x = np.arange(1,11)
y1 = x + 1
y2 = x + 5

plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
```



```
plt.plot(x, y1, label="student 1", marker='o', color='b', linestyle='')
plt.plot(x, y2, label="student 2", marker='^', color='r', linestyle='')
plt.plot(x, y1+2)
plt.legend(loc=2)

plt.xticks(np.arange(1,11),['a','b','c','d','e','f','g','h','i','j'], rotation=45)

plt.yticks(np.arange(16))

plt.show()
```

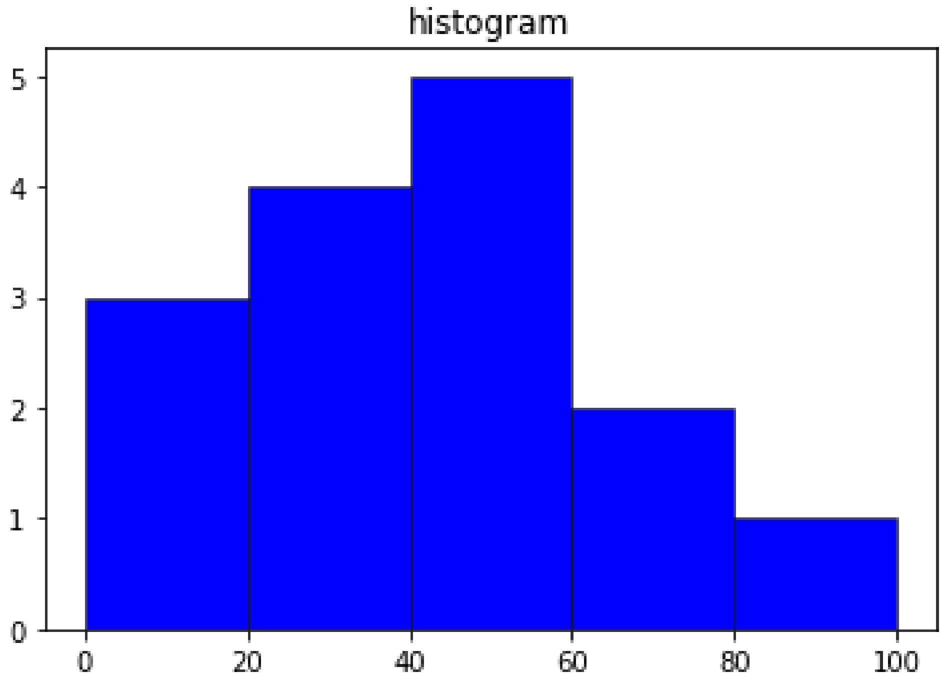
Bar Graph

```
from matplotlib import pyplot as plt

x = [5, 8, 10]
y = [12, 16, 6]

x2 = [6, 9, 11]
y2 = [6, 15, 7]


plt.bar(x, y, color = 'b', align = 'center')
plt.bar(x2, y2, color = 'g', align = 'center')

plt.title('Bar graph')
plt.ylabel('Y axis')
plt.xlabel('X axis')

plt.show()
```



Histogram

```
from matplotlib import pyplot as plt

import numpy as np

a = np.array([22,87,5,43,56,73,55,54,11,20,51,5,79,31,27])

plt.hist(a, bins = [0,20,40,60,80,100], color = 'b',  edgecolor='k', linewidth = 0.5)

plt.title("histogram")

plt.show()
```



OpenCV - Installing and Importing

```
pip install opencv-python
```

```
import cv2
```

OpenCV - Reading an Image

```
img = cv2.imread(“Desert.jpg”)
```

OpenCV - Showing an Image

```
cv2.imshow(“First Figure”,img)
```

OpenCV – Writing an Image

```
cv2.imwrite(“Desert1.jpg”, img)
```

OpenCV – Resizing an Image

```
img1=cv2.resize(img, (360, 512))
```

or

```
img1=cv2.resize(img, None, fx=0.5, fy=0.5)
#fx -> columns
```

OpenCV

- Around 2500 efficient algorithms
- Face detection applications
- Object identification applications
- Anomaly detection from a video
- Content-based image retrieval

Reading data from a CSV File

```
import pandas
import numpy as np

data=pandas.read_csv('temp.csv',header=None)
#default header argument is infer

data=data.as_matrix() # or data=data.values

print(data)
```

Writing data to a CSV File

```
import numpy
import pandas as pd

a = numpy.array([ [1,2,3], [4,5,6], [7,8,9] ])

df = pd.DataFrame(a)
df.to_csv("file.csv", header=None, index=False)
```

Disclaimer

- Content of this presentation is not original and it has been prepared from various sources for teaching purpose.