1.Figure: It is a top-level container for all plot elements. The Figure object is instantiated by calling the figure() function from the pyplot module −

fig = plt.figure()

Additional parameters −

|  |  |
| --- | --- |
| Figsize | (width,height) tuple in inches |
| Dpi | Dots per inches |
| Facecolor | Figure patch facecolor |
| Edgecolor | Figure patch edge color |
| Linewidth | Edge line width |

2.Axes: Axes object is the region of the image with the data space. A given figure can contain many Axes, but a given Axes object can only be in one Figure. The Axes contains two (or three in the case of 3D) Axis objects. The Axes class and its member functions are the primary entry point to working with the OO interface.

Axes object is added to figure by calling the add\_axes() method. It returns the axes object and adds an axes at position rect [left, bottom, width, height] where all quantities are in fractions of figure width and height. Each number must be between 0 and 1 −

ax=fig.add\_axes([0,0,1,1])

Axis: This function is used to set some axis properties to the graph.xmin, xmax, ymin, ymax:These parameters can be used to  
set the axis limits on the graph

Example:

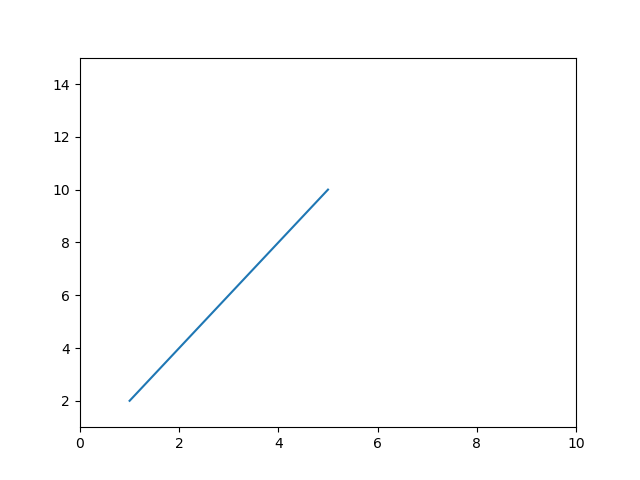
import matplotlib.pyplot as plt

x =[1, 2, 3, 4, 5]

y =[2, 4, 6, 8, 10]

plt.plot(x, y)

plt.axis([0, 10, 1, 15])

plt.show()

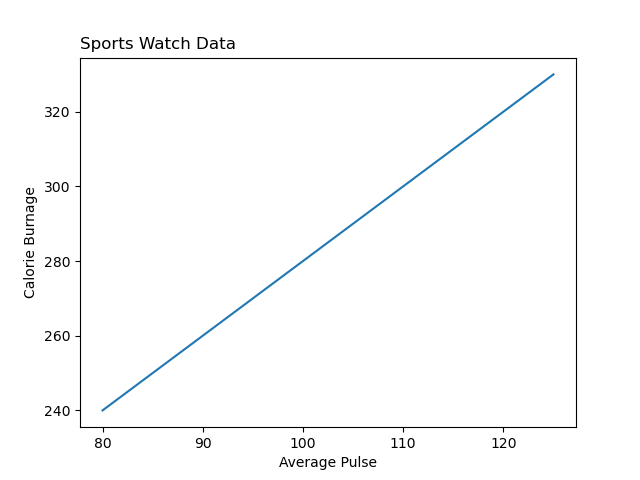
Artist: . The Artist class contains Abstract base class for objects that render into a FigureCanvas. All visible elements in a figure are subclasses of Artist.

Labels: With Pyplot, you can use the xlabel() and ylabel() functions to set a label for the x- and y-axis.

Title: With Pyplot, you can use the title() function to set a title for the plot.

Example:

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", loc = 'left')  
plt.xlabel("Average Pulse")  
plt.ylabel("Calorie Burnage")  
  
plt.plot(x, y)  
plt.show()



Legends: The legend() method of axes class adds a legend to the plot figure. It takes three parameters −

ax.legend(handles, labels, loc)

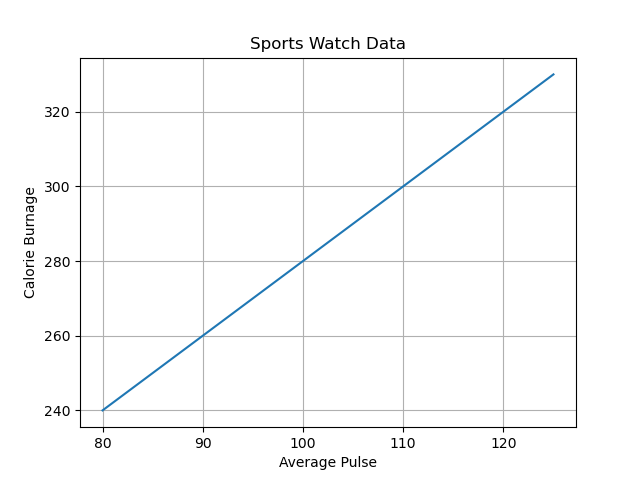
Ticks and Tick Labels: Ticks are the markers denoting data points on axes.The xticks() and yticks() function takes a list object as argument. The elements in the list denote the positions on corresponding action where ticks will be displayed.Tick Labels are the the labels of x-ticks and y-ticks.

Grid: With Pyplot, you can use the grid() function to add grid lines to the plot.

Example:

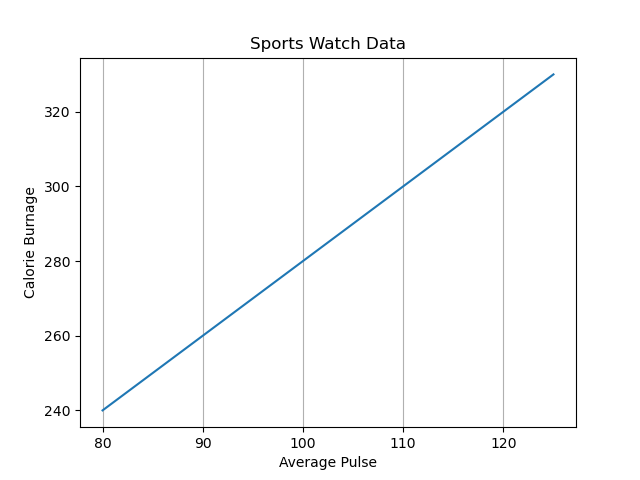
A)

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()



B)

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(axis = 'x')  
  
plt.show()

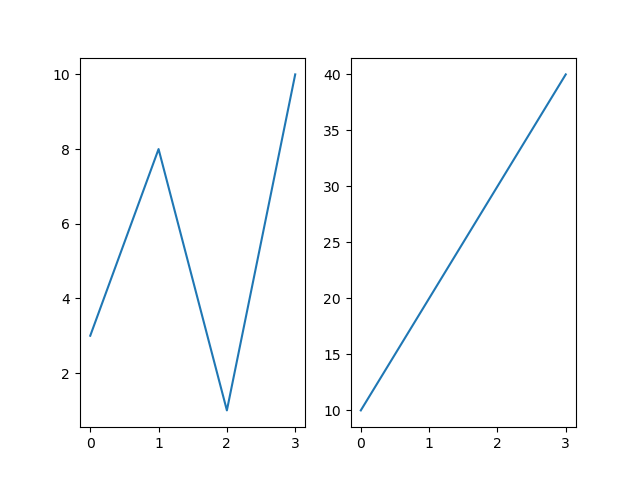


Subplot: With the subplots() function you can draw multiple plots in one figure.

Example:

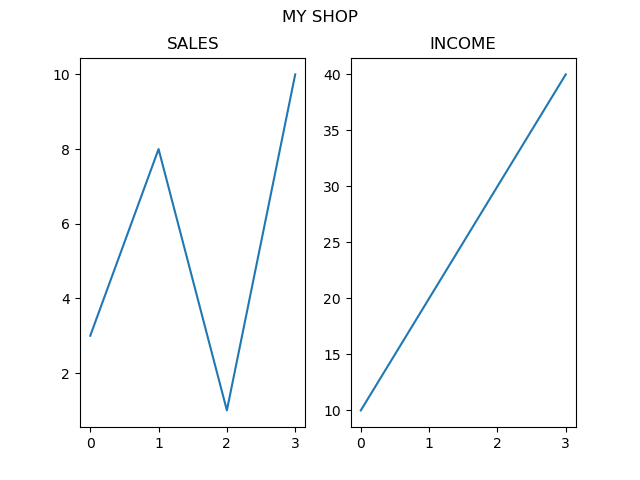
A)

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()



B)

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()

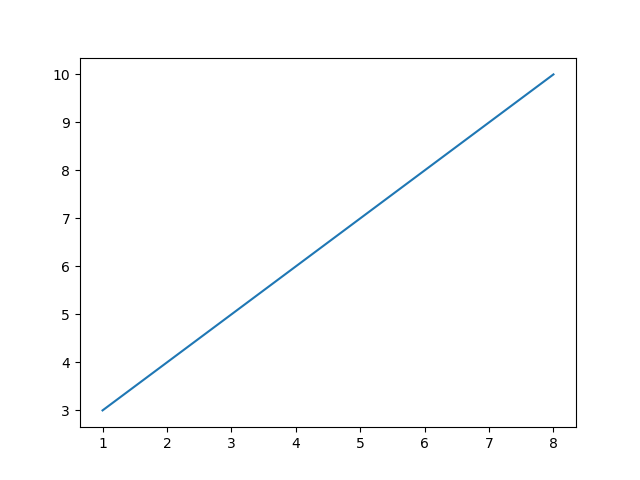


Line Plot: The plot() function is used to draw points (markers) in a diagram.By default, the plot() function draws a line from point to point.

Example:

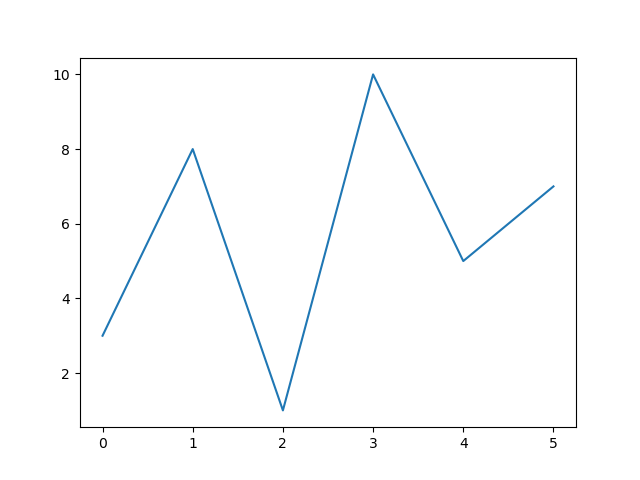
A)

import matplotlib.pyplot as plt  
import numpy as np  
  
xpoints = np.array([1, 8])  
ypoints = np.array([3, 10])  
  
plt.plot(xpoints, ypoints)  
plt.show()



B)

import matplotlib.pyplot as plt  
import numpy as np  
  
ypoints = np.array([3, 8, 1, 10, 5, 7])  
  
plt.plot(ypoints)  
plt.show()



Scatter plot: The scatter() function plots one dot for each observation. It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis.

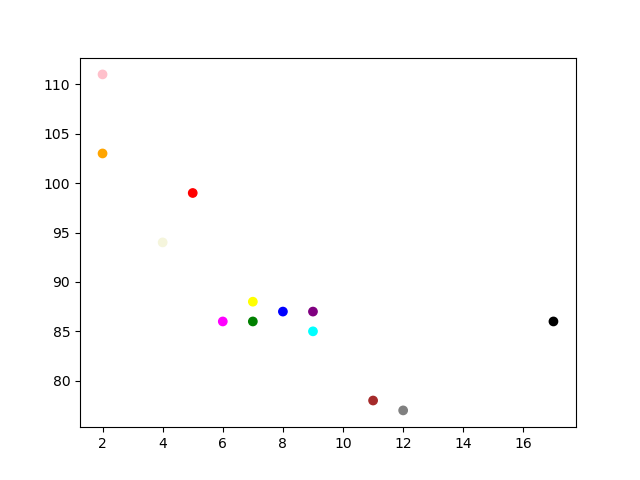
Example:

A)

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
  
plt.scatter(x, y)  
plt.show()

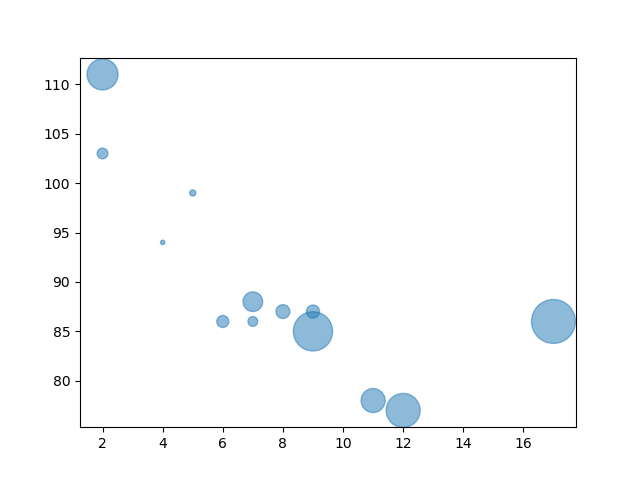
B)

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
colors = np.array(["red","green","blue","yellow","pink","black","orange","purple","beige","brown","gray","cyan","magenta"])  
  
plt.scatter(x, y, c=colors)  
  
plt.show()



C)

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])  
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])  
sizes = np.array([20,50,100,200,500,1000,60,90,10,300,600,800,75])  
  
plt.scatter(x, y, s=sizes, alpha=0.5)  
  
plt.show()

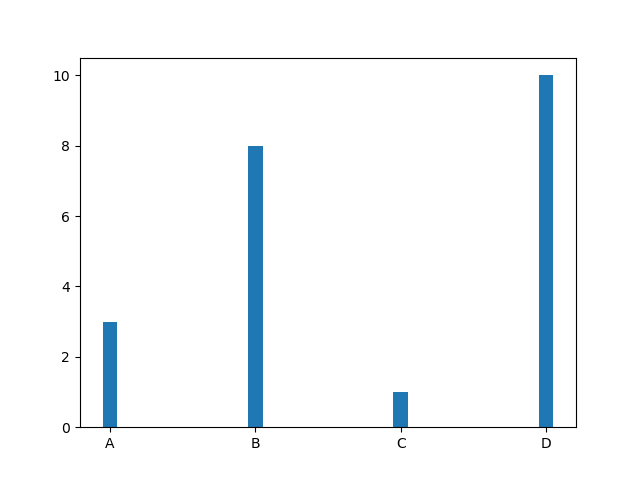


Bar plot: With Pyplot, you can use the bar() function to draw bar graphs.

Example:

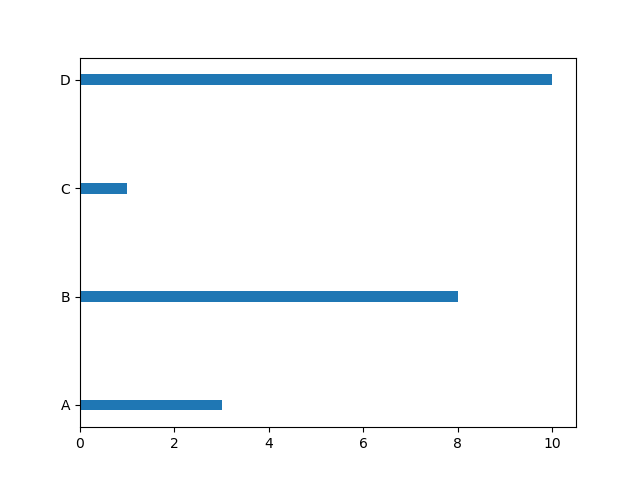
A)

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array(["A", "B", "C", "D"])  
y = np.array([3, 8, 1, 10])  
  
plt.bar(x,y, width = 0.1)  
plt.show()



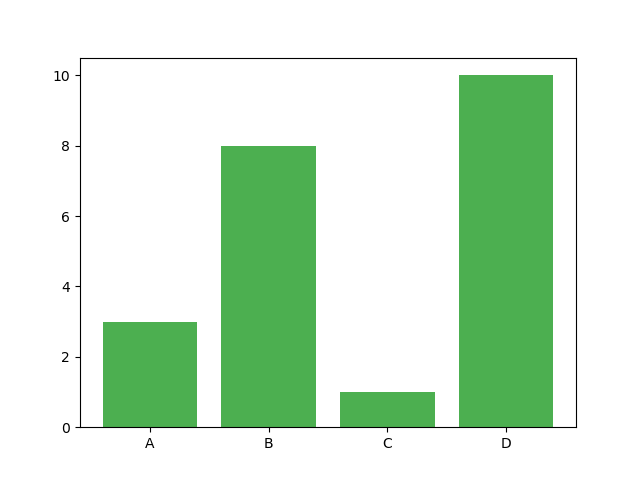
B)

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array(["A", "B", "C", "D"])  
y = np.array([3, 8, 1, 10])  
  
plt.barh(x, y, height = 0.1)  
plt.show()



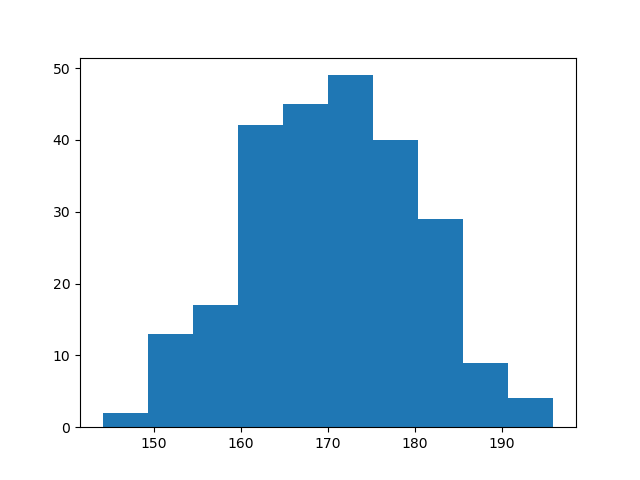
C)

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array(["A", "B", "C", "D"])  
y = np.array([3, 8, 1, 10])  
  
plt.bar(x, y, color = "#4CAF50" )  
plt.show()



Histogram: A histogram is a graph showing frequency distributions. It is a graph showing the number of observations within each given interval.

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.random.normal(170, 10, 250)  
  
plt.hist(x)  
plt.show()



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

Example:

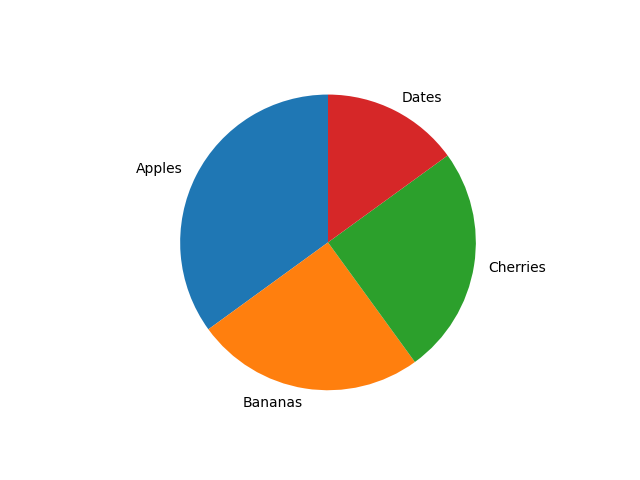
A)

import matplotlib.pyplot as plt  
import numpy as np  
  
y = np.array([35, 25, 25, 15])  
  
plt.pie(y)  
plt.show()



B)

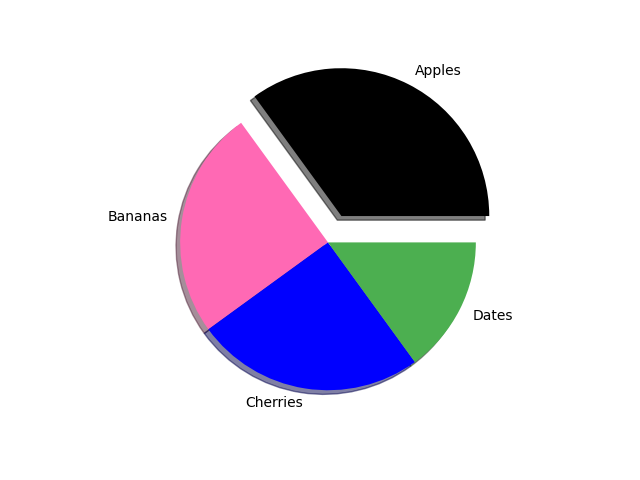
import matplotlib.pyplot as plt  
import numpy as np  
  
y = np.array([35, 25, 25, 15])  
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]  
  
plt.pie(y, labels = mylabels, startangle = 90)  
plt.show()



C)

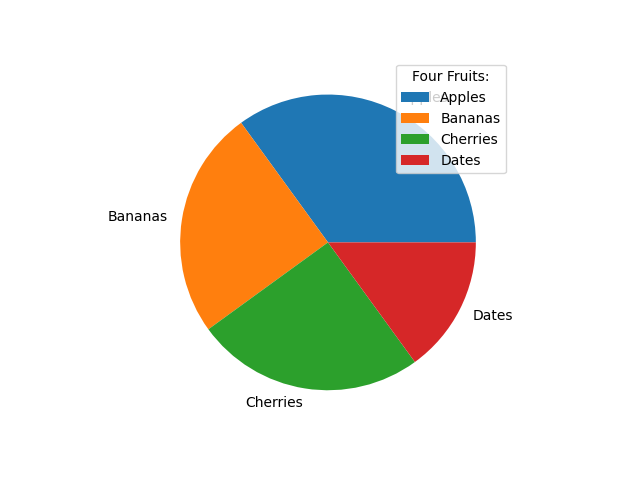
import matplotlib.pyplot as plt  
import numpy as np  
  
y = np.array([35, 25, 25, 15])  
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]

mycolors = ["black", "hotpink", "b", "#4CAF50"]  
myexplode = [0.2, 0, 0, 0]  
  
plt.pie(y, labels = mylabels, colors = mycolors, explode = myexplode, shadow = True)  
plt.show()



D)

import matplotlib.pyplot as plt  
import numpy as np  
  
y = np.array([35, 25, 25, 15])  
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]  
  
plt.pie(y, labels = mylabels)  
plt.legend(title = "Four Fruits:")  
plt.show()



Box Plot:A box plot which is also known as a whisker plot displays a summary of a set of data containing the minimum, first quartile, median, third quartile, and maximum. In a box plot, we draw a box from the first quartile to the third quartile. A vertical line goes through the box at the median. The whiskers go from each quartile to the minimum or maximum.

Example:

**import** matplotlib.pyplot as plt

**import** numpy as np

# Creating dataset

np.random.seed(10)

data\_1 **=** np.random.normal(100, 10, 200)

data\_2 **=** np.random.normal(90, 20, 200)

data\_3 **=** np.random.normal(80, 30, 200)

data\_4 **=** np.random.normal(70, 40, 200)

data **=** [data\_1, data\_2, data\_3, data\_4]

fig **=** plt.figure(figsize **=**(10, 7))

# Creating axes instance

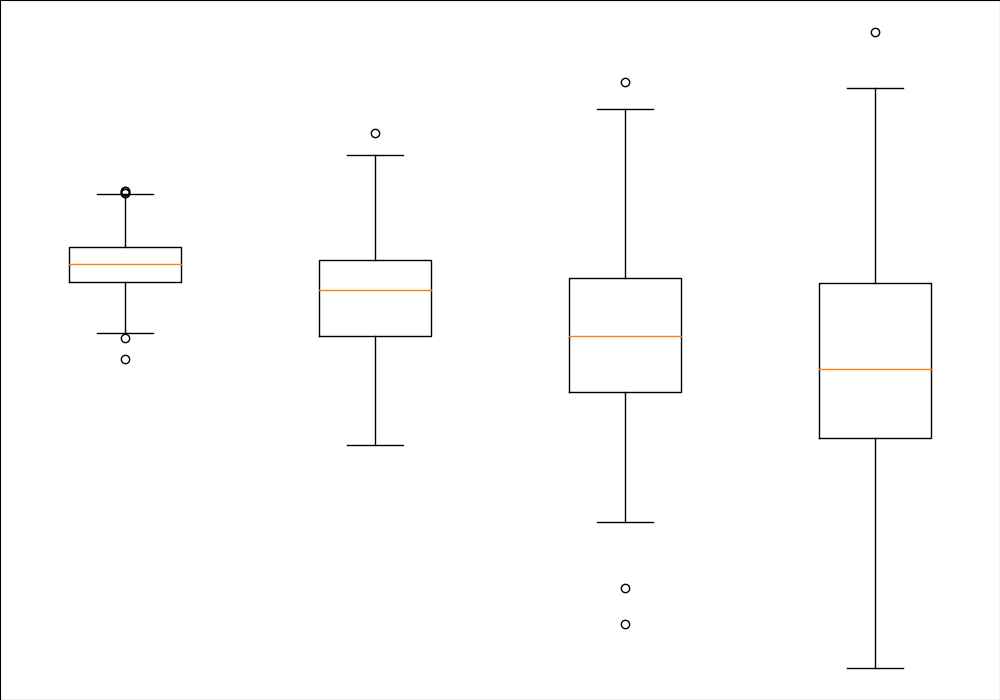
ax **=** fig.add\_axes([0, 0, 1, 1])

# Creating plot

bp **=** ax.boxplot(data)

# show plot

plt.show()



Contour plot: Contour plots (sometimes called Level Plots) are a way to show a three-dimensional surface on a two-dimensional plane. It graphs two predictor variables X Y on the y-axis and a response variable Z as contours.

Example:

from mpl\_toolkits import mplot3dimport numpy as npimport matplotlib.pyplot as pltdef f(x, y):

return np.sin(np.sqrt(x \*\* 2 + y \*\* 2))

x = np.linspace(-6, 6, 30)

y = np.linspace(-6, 6, 30)

X, Y = np.meshgrid(x, y)

Z = f(X, Y)

fig = plt.figure()

ax = plt.axes(projection='3d')

ax.contour3D(X, Y, Z, 50, cmap='binary')

ax.set\_xlabel('x')

ax.set\_ylabel('y')

ax.set\_zlabel('z')

ax.set\_title('3D contour')

plt.show()

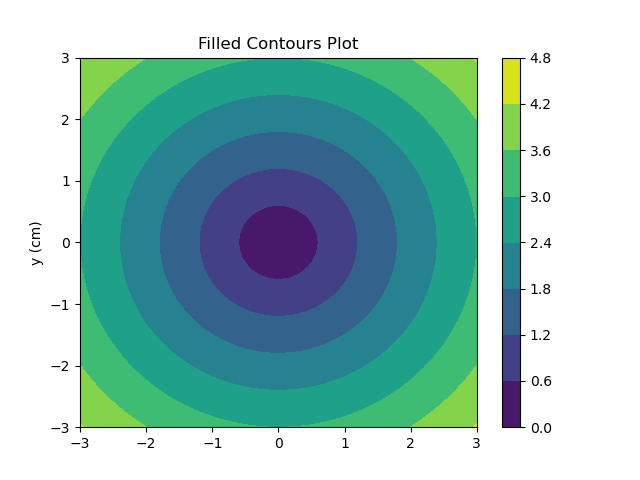


Table plot: Matplotlib.pyplot.table() is a subpart of matplotlib library in which a table is generated using the plotted graph for analysis.

Example:

**import** numpy as np

**import** matplotlib.pyplot as plt

data **=** [[ 66, 174,  71, 58],

        [ 58, 139,  45, 164],

        [ 89,  52, 18, 81],

        [ 78,  58, 123,  68],

        [13, 159, 164, 80]]

val1 **=** ('Geek1', 'Geek2', 'Geek3', 'Geek4')

val2 **=** ['Month % d' **%** x **for** x **in** (5, 4, 3, 2, 1)]

val3 **=** np.arange(0, 2500, 500)

val4 **=** 1000

val5 **=** plt.cm.plasma(np.linspace(0, 0.5, len(val2)))

val6 **=** len(data)

val7 **=** np.arange(len(val1)) **+** 0.3

val8 **=** 0.4

val9 **=** np.zeros(len(val1))

lista **=** []

fig, ax **=** plt.subplots()

**for** row **in** range(val6):

    ax.bar(val7, data[row], val8, bottom **=** val9,

           color **=** val5[row])

    val9 **=** val9 **+** data[row]

    lista.append([(x **//** 50) **for** x **in** val9])

the\_table **=** ax.table(cellText **=** lista,

                      rowLabels **=** val2,

                      rowColours **=** val5,

                      colLabels **=** val1,

                      loc **=**'bottom')

plt.subplots\_adjust(left **=** 0.2, bottom **=** 0.2)

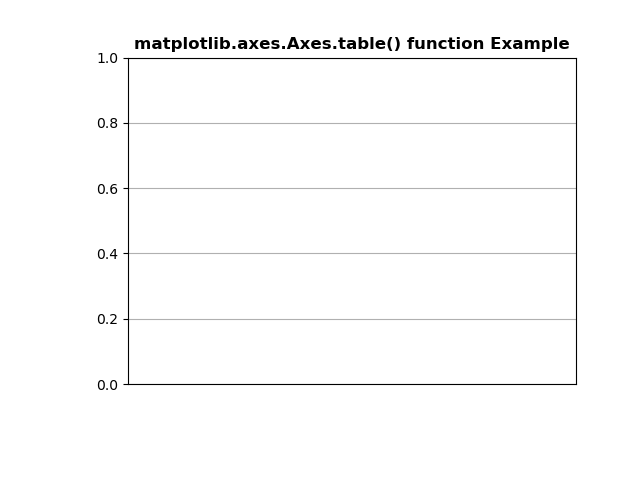
ax.set\_xticks([])

ax.set\_title('matplotlib.axes.Axes.table() function Example',

              fontweight **=**"bold")

plt.grid()

plt.show()



Stack plot: Stack plots are mainly used to see various trends in variables over a specific period of time.

Example:

import matplotlib.pyplot as plt

days = [1,2,3,4,5]

sleep = [6,7,5,8,6]

eat =   [2,2,1,2,1]

work =  [5,7,10,8,6]

exercise=  [3,3,0,1,3]

plt.plot([],[],color='green', label='sleep', linewidth=3)

plt.plot([],[],color='blue', label='eat', linewidth=3)

plt.plot([],[],color='red', label='work', linewidth=3)

plt.plot([],[],color='black', label='play', linewidth=3)

plt.stackplot(days, sleep, eat, work, exercise, colors=['green','blue','red','black'])

plt.xlabel('days')

plt.ylabel('activities')

plt.title('5 DAY ROUTINE')

plt.legend()

plt.show()

