

# What is Matplotlib

Matplotlib is a low level graph plotting library in python that serves as a visualization utility. Most of the Matplotlib utilities lie under the pyplot submodule and are usually imported under the `plt` alias: `import matplotlib.pyplot as plt` now the pyplot package can be referred to as `plt`. Example: Draw a line in a diagram from position (0,0) to position (6,250):

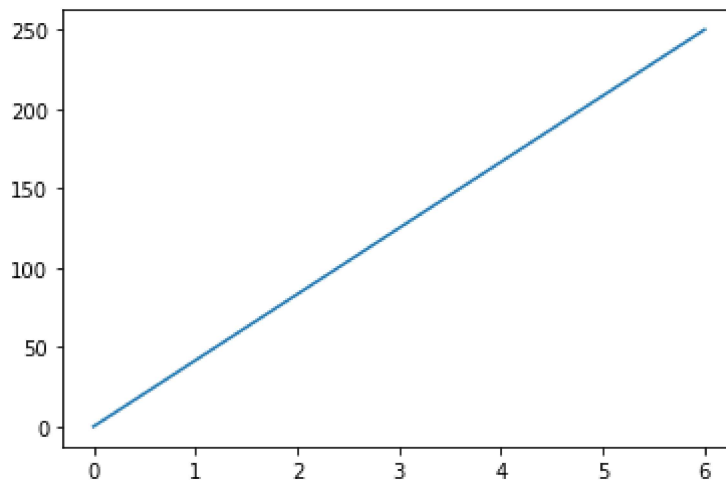
```
In [3]: import matplotlib
```

## Matplotlib pyplot

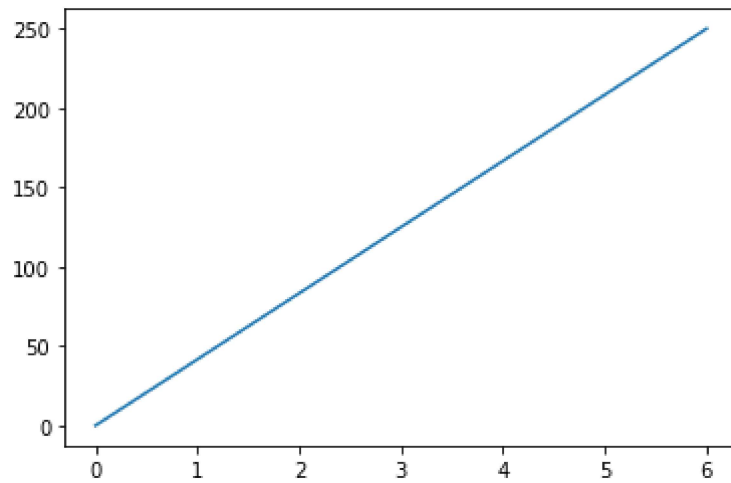
pyplot: Most of the matplotlib utilities lie under the pyplot submodule and are usually imported under the `plt` alias: `import matplotlib.pyplot as plt`. Now the pyplot package can be referred to as `plt`.

```
In [4]: #example  
#Draw a line in a diagram from position (0,0) to position (6,250)
```

```
In [5]: import matplotlib.pyplot as plt  
import numpy as np  
xpoints=np.array([0,6])  
ypoints=np.array([0,250])  
plt.plot(xpoints,ypoints)  
plt.show()
```



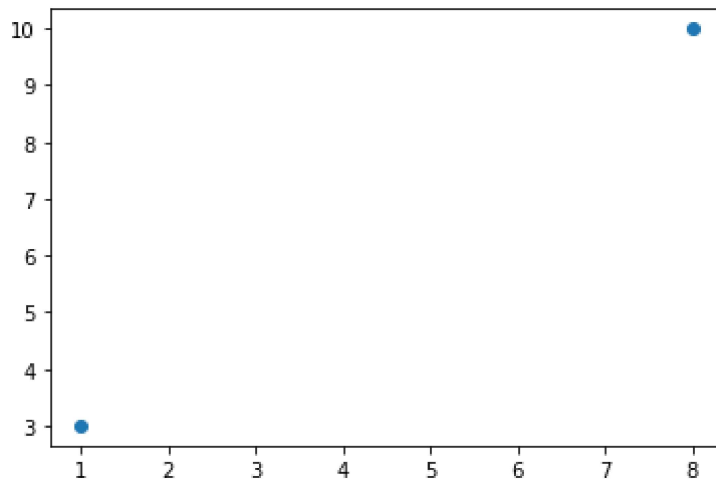
```
In [6]: a = np.array([0,6])  
b = np.array([0,250])  
plt.plot(a,b)  
plt.show()
```



## Matpoltlib plotting

```
In [7]: ## Draw the line a diagram from position (1.3) to possition (8,10)
```

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

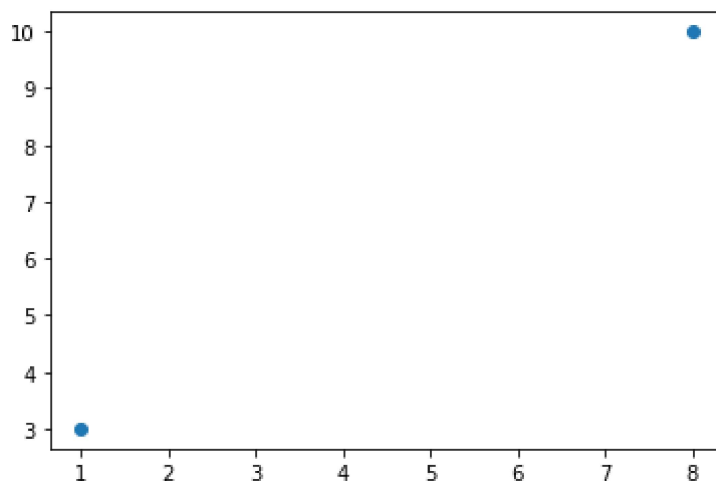


## plotting without line

**Draw two points in the diagram one at position (1,3) and one in position (8,10)**

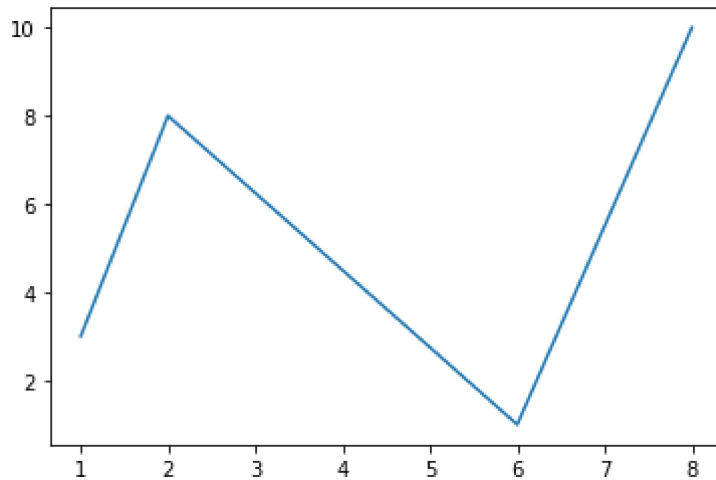
```
In [9]: import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([1,8])
ypoints = np.array([3,10])
plt.plot(xpoints,ypoints,'o')
plt.show
```

Out[9]: <function matplotlib.pyplot.show(close=None, block=None)>



## multiple points

```
In [10]: xpoints = np.array([1,2,6,8])
ypoints = np.array([3,8,1,10])
plt.plot(xpoints,ypoints)
plt.show()
```

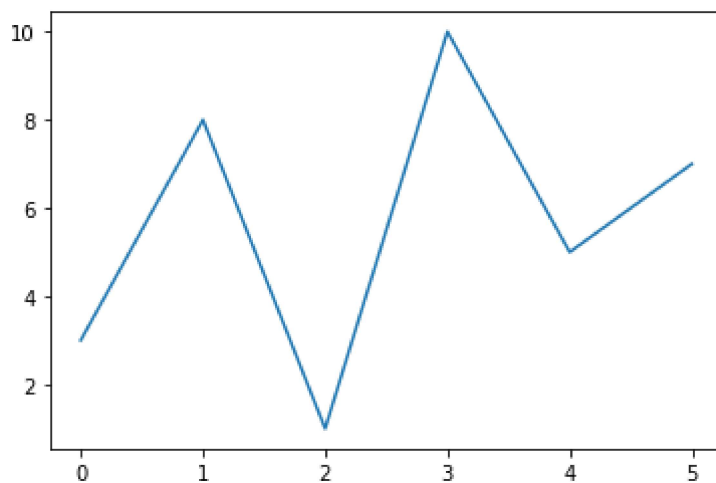


## Default x- points

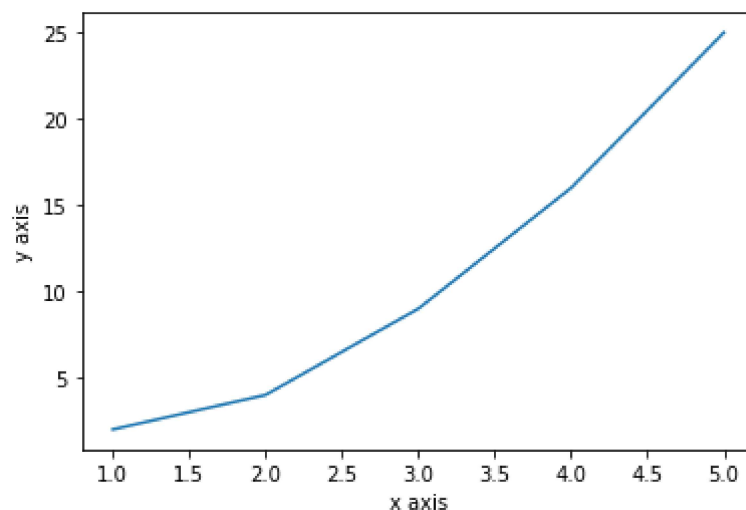
```
In [11]: #Example : plotting without x-points

import matplotlib.pyplot as plt
import numpy as np

ypoints=np.array([3,8,1,10,5,7])
plt.plot(ypoints)
plt.show()
```

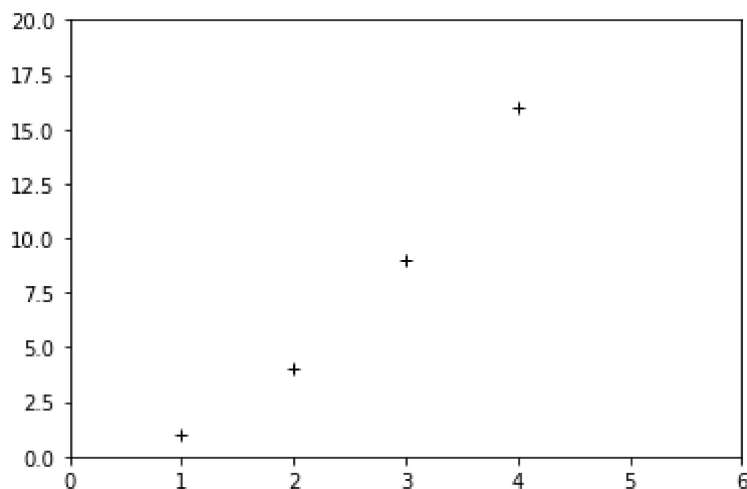


```
In [12]: from matplotlib import pyplot as plt
plt.plot([1,2,3,4,5],[2,4,9,16,25])
plt.ylabel("y axis")
plt.xlabel("x axis")
plt.show()
```



## Formating the style of the plot

```
In [13]: from matplotlib import pyplot as plt
plt.plot([1,2,3,4,5],[1,4,9,16,25], 'k+')
plt.axis([0,6,0,20])
plt.show()
```



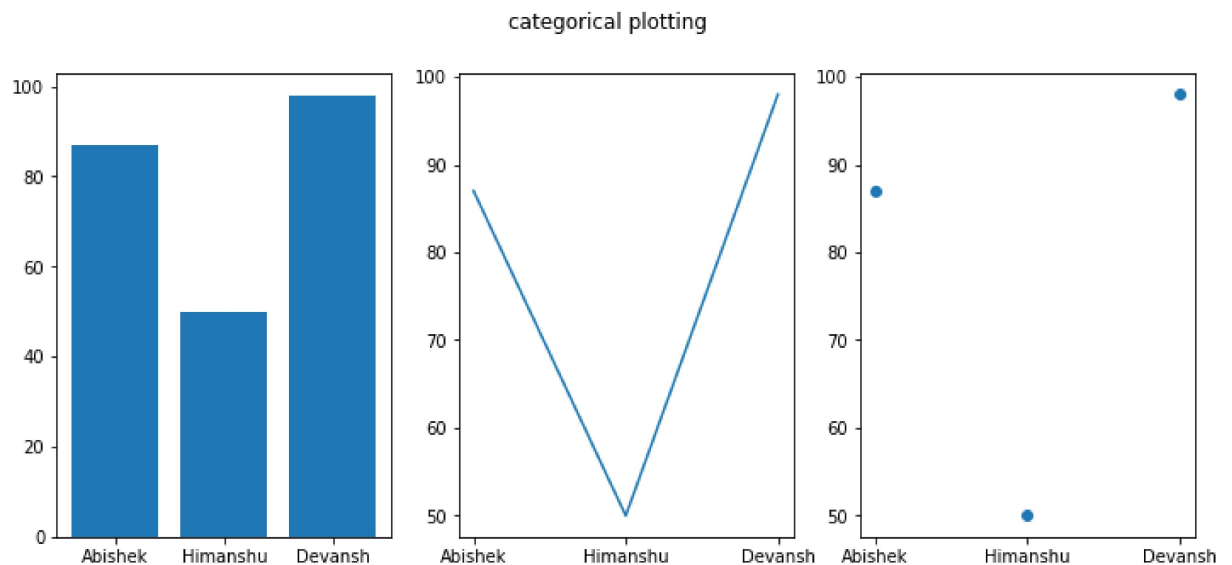
**The matplotlib supports the following color abbreviation:**

'b'-Blue 'm'-Magenta 'g'-Green 'y'-yellow 'r'-Red 'k'-Black 'c'-cyan 'w'-White Plotting with categorical variables Matplotlib allows us to pass categorical variables directly to many plotting functions: consider the following example.

```
In [14]: from matplotlib import pyplot
names=['Abishek', 'Himanshu', 'Devansh']
marks=[87,50,98]

plt.figure(figsize=(12,5))    # Fix sized of graph

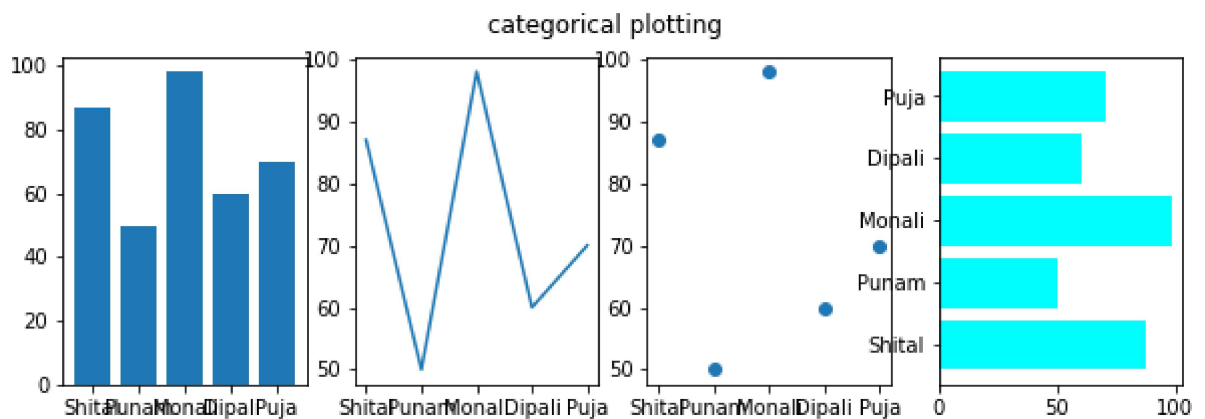
plt.subplot(131)              # 1st is row, 2nd is column, 3rd is index.
plt.bar(names,marks)
plt.subplot(133)
plt.scatter(names,marks)
plt.subplot(132)
plt.plot(names,marks)
plt.suptitle('categorical plotting')
plt.show()
```



```
In [15]: from matplotlib import pyplot
names=['Shital','Punam','Monali','Dipali','Puja']
marks=[87,50,98,60,70]

plt.figure(figsize=(10,3))

plt.subplot(141)
plt.bar(names,marks)
plt.subplot(143)
plt.scatter(names,marks)
plt.subplot(142)
plt.plot(names,marks)
plt.subplot(144)
plt.barh(names,marks,color='cyan')
plt.suptitle('categorical plotting')
plt.show()
```

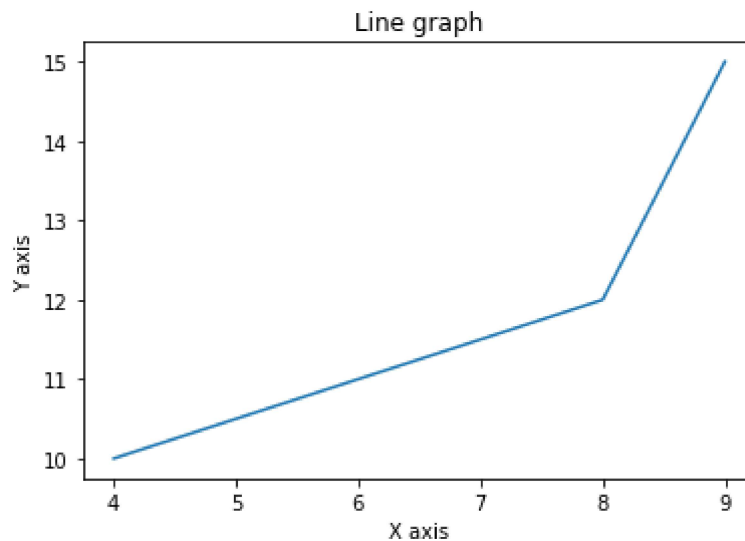


## Creating different types of graphs

### 1. Line Graph¶

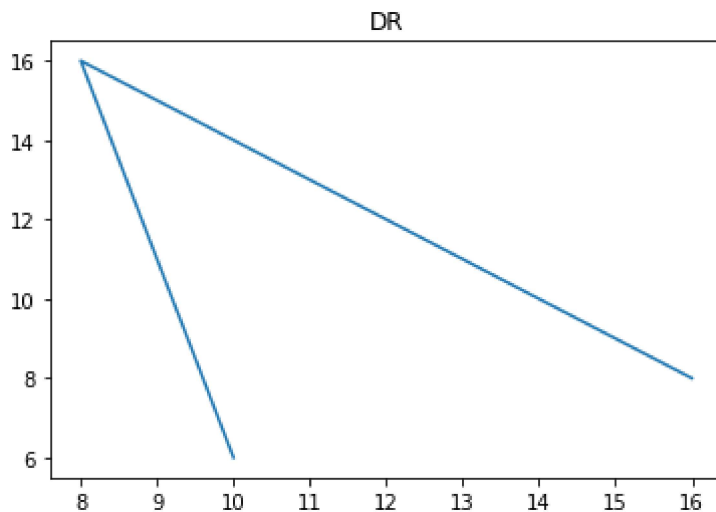
```
In [1]: from matplotlib import pyplot as plt
```

```
x=[4,8,9]  
y=[10,12,15]  
  
plt.plot(x,y)  
  
plt.title("Line graph")  
plt.ylabel('Y axis')  
plt.xlabel('X axis')  
plt.show()
```



```
In [3]: from matplotlib import pyplot as plt
```

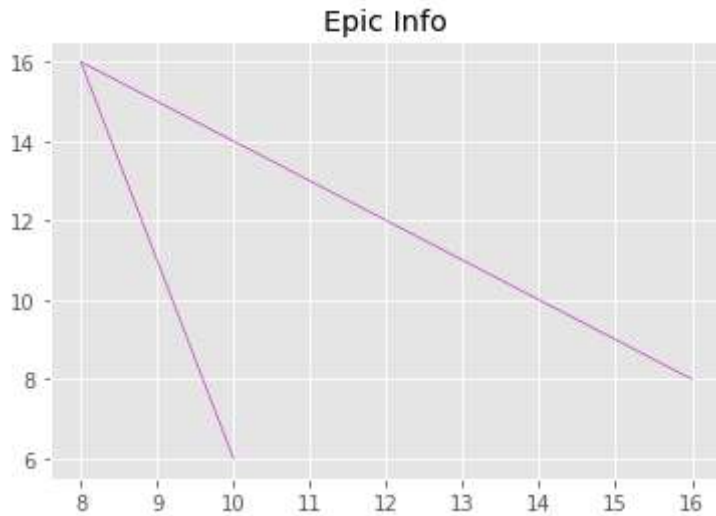
```
x=[16,8,10]  
y=[8,16,6]  
plt.plot(x,y)  
plt.title('DR')  
plt.show()
```





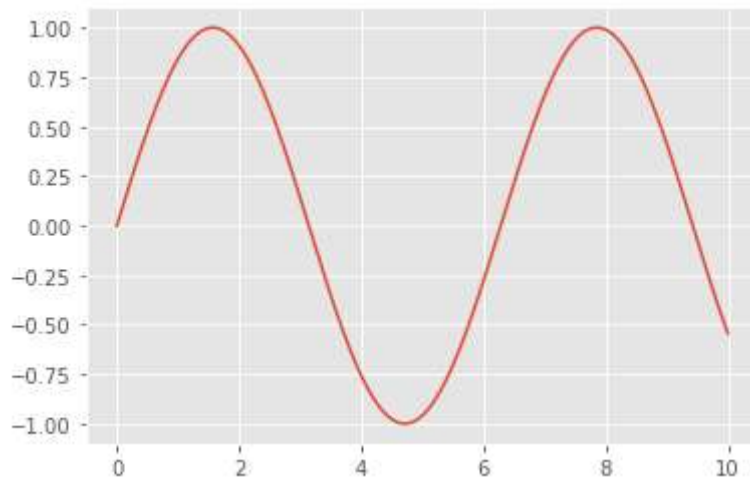
```
In [4]: from matplotlib import pyplot as plt
from matplotlib import style

style.use('ggplot')
x=[16,8,10]
y=[8,16,6]
plt.plot(x,y,'m',linewidth=0.5)
plt.title('Epic Info')
plt.show()
```



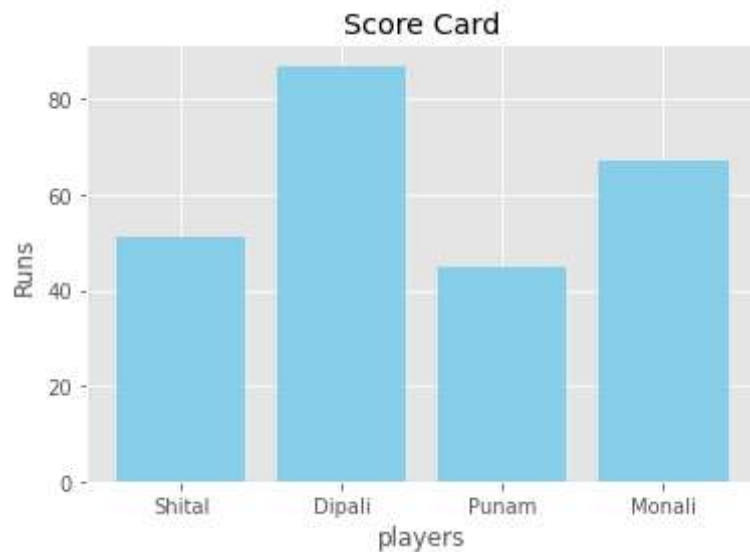
```
In [5]: import numpy as np
import matplotlib.pyplot as plt
fig=plt.figure()
ax=plt.axes()
x=np.linspace(0,10,1000)
ax.plot(x,np.sin(x))
```

Out[5]: [



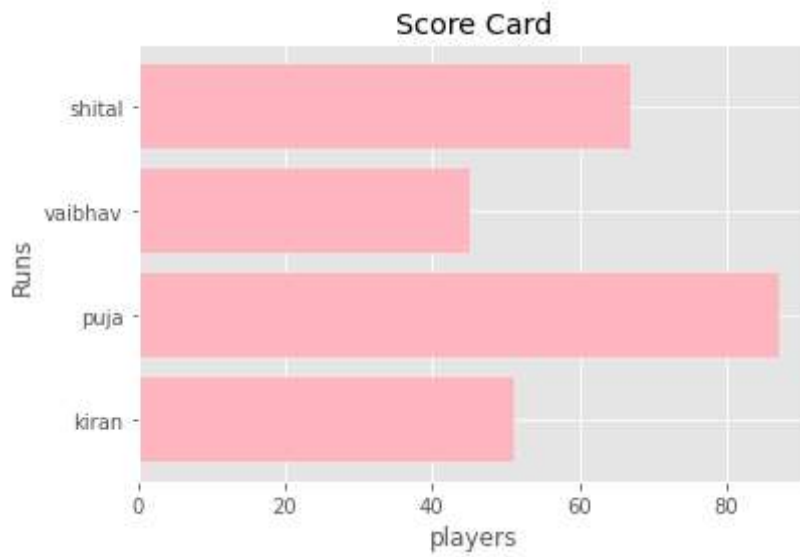
## Bar Graph

```
In [6]: from matplotlib import pyplot as plt
players=['Shital','Dipali','Punam','Monali']
runs=[51,87,45,67]
plt.bar(players,runs,color='skyblue')
plt.title('Score Card')
plt.xlabel('players')
plt.ylabel('Runs')
plt.show()
```



```
In [7]: from matplotlib import pyplot as plt
players=['kiran','puja','vaibhav','shital']
runs=[51,87,45,67]
plt.barh(players,runs,color='lightpink')
plt.title('Score Card')
plt.xlabel('players')
plt.ylabel('Runs')
plt
```

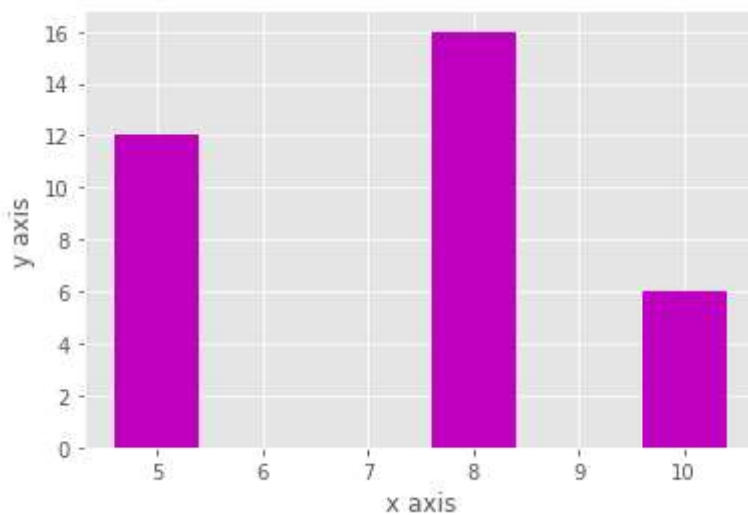
Out[7]: <module 'matplotlib.pyplot' from 'D:\\Users\\Admin\\anaconda3\\lib\\site-packages\\matplotlib\\pyplot.py'>



```
In [8]: #Lets have look on the other example using the style() function
```

```
In [17]: from matplotlib import pyplot as plt
from matplotlib import style
style.use('ggplot')
x = [5,8,10]
y = [12,16,6]
x2 = [6,9,11]
y2 = [7,15,7]
plt.bar(x,y,color='b',align='center')
plt.bar(x,y,color='m',align='center')
plt.ylabel('y axis')
plt.xlabel('x axis')
```

Out[17]: Text(0.5, 0, 'x axis')



```
In [2]: from matplotlib import pyplot as plt
import numpy as np

countries = ['USA', 'India', 'china', 'Russia', 'germany']
bronzes = np.array([38,17,26,19,15])
silvers = np.array([37,23,18,18,10])
golds = np. array([46,27,26,19,17])
ind=[x for x,_in enumerate(countries)]

plt.bar(ind,golds,width = 0.5,label = 'golds', color='golds',bottom = silvers+bronzes)
plt.bar(ind,silvers, width=0.5,label = 'slivers',color='silvers',bottom=bronze)
plt.bar(ind,bronzes,width=0.5, label='bronzes',color='#CD853F')
plt.xticks(ind,countries)
plt.ylabel("Medals")
plt.xlabel("countries")
plt.legend(loc = "upper right")
plt.title("2019 Olympic Top Score")
```

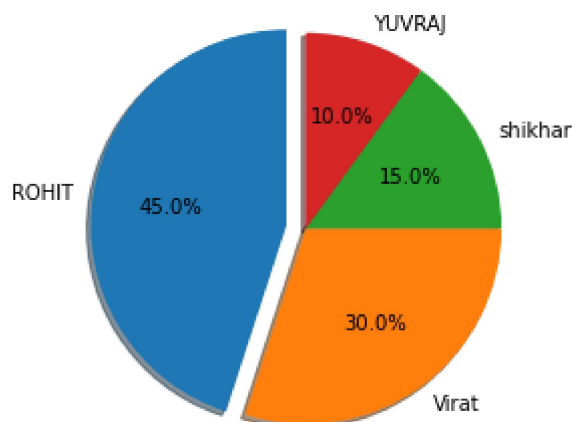
Input In [2]

```
ind=[x for x,_in enumerate(countries)]
```

**SyntaxError:** invalid syntax

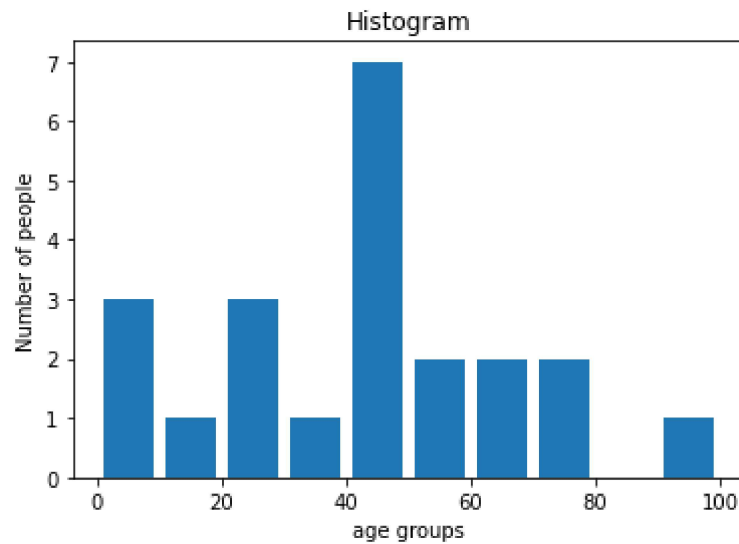
## pie charts

```
In [5]: from matplotlib import pyplot as plt
#pie chart wherw the slice will be oereed and plotted clowisw
Players = 'ROHIT','Virat','shikhar','YUVRAJ'
Runs = [45,30,15,10]
explode = (0.1,0,0,0)
fig1,ax1=plt.subplots()
ax1.pie(Runs,explode=explode,labels=Players,autopct='%1.1f%%',shadow=True,startangle=90)
ax1.axis('equal')# equal aspect ratio ensures that pie is drawn as a circle
plt.show()
```

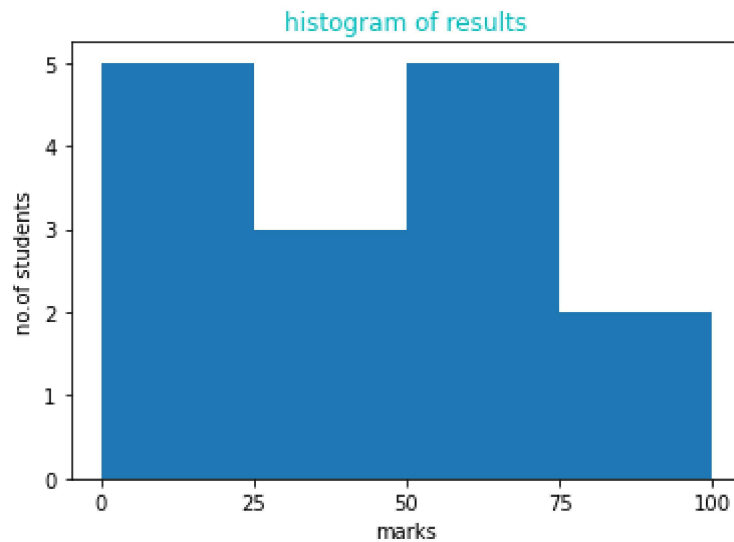


## 4.Histogram

```
In [6]: from matplotlib import pyplot as plt
population_age=[21,53,60,49,25,27,30,42,40,1,2,102,95,8,15,105,70,75,60,52,44,43,
bins=[0,10,20,30,40,50,60,70,80,90,100]
plt.hist(population_age,bins,histtype='bar',rwidth=0.8)
plt.xlabel('age groups')
plt.ylabel('Number of people')
plt.title('Histogram')
plt.show()
```



```
In [20]: from matplotlib import pyplot as plt
import numpy as np
fig,ax=plt.subplots(1,1)
a=np.array([22,87,5,43,56,73,55,54,11,20,51,5,79,31,27])
ax.hist(a,bins=[0,25,50,75,100])
ax.set_title("histogram of results", color = 'c')
ax.set_xticks([0,25,50,75,100])
ax.set_xlabel('marks')
ax.set_ylabel('no.of students')
plt.show()
```

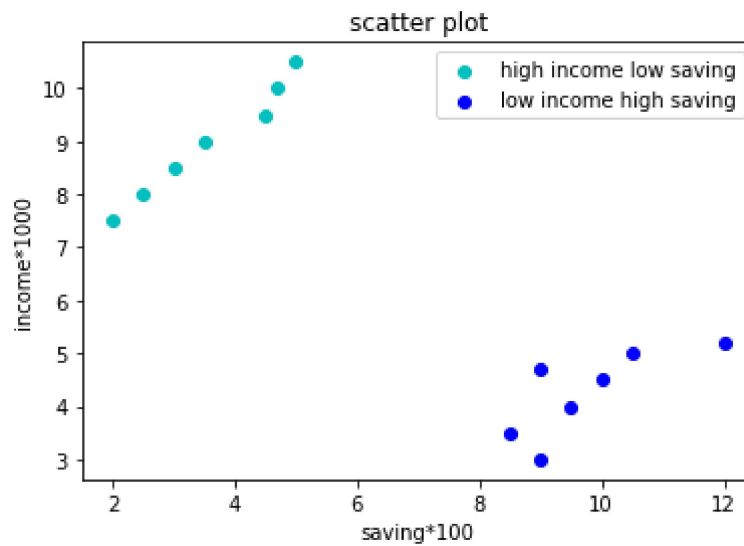


## 5.Scatter Plot

```
In [16]: #Scatter Plot
```

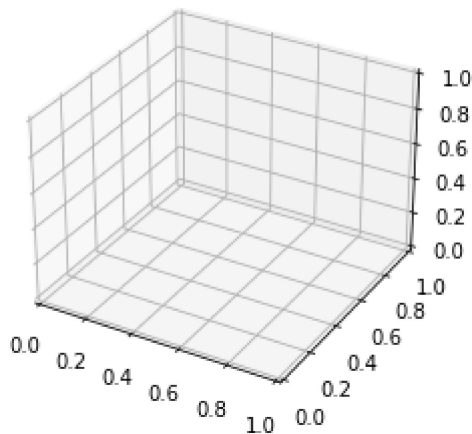
```
In [19]: import matplotlib.pyplot as plt
x=[2,2.5,3,3.5,4.5,4.7,5.0]
y=[7.5,8,8.5,9,9.5,10,10.5]

x1=[9,8.5,9,9.5,10,10.5,12]
y1=[3,3.5,4.7,4,4.5,5,5.2]
plt.scatter(x,y,label='high income low saving',color='c')
plt.scatter(x1,y1,label='low income high saving',color='b')
plt.xlabel('saving*100')
plt.ylabel('income*1000')
plt.title('scatter plot')
plt.legend()
plt.show()
```



## 6. 3D Graph Plot

```
In [13]: import numpy as np
import matplotlib.pyplot as plt
fig=plt.figure()
ax=plt.axes(projection='3d')
```





```
In [14]: import numpy as np
import matplotlib.pyplot as plt

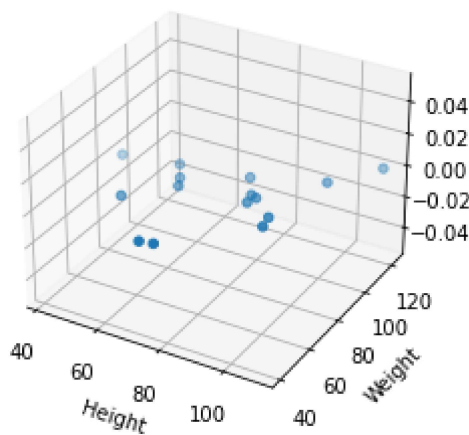
height=np.array([100,110,87,85,65,80,96,75,42,59,54,63,95,71,86])
weight=np.array([105,123,84,85,78,95,69,42,87,91,63,83,75,41,80])

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

# This is used to plot 3d scatter

ax.scatter3D(height,weight)
plt.title("3D Scatter Plot")
plt.xlabel("Height")
plt.ylabel("Weight")
plt.title("3D Scatter Plot")
plt.xlabel("Height")
plt.ylabel("Weight")
plt.show()
```

3D Scatter Plot



In [ ]: