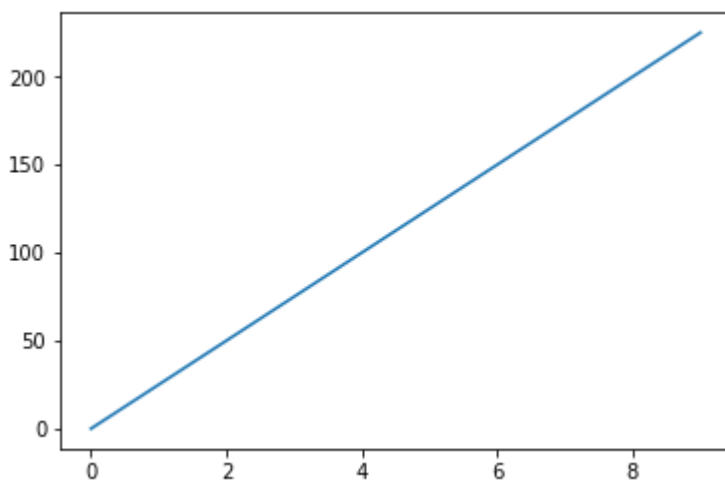
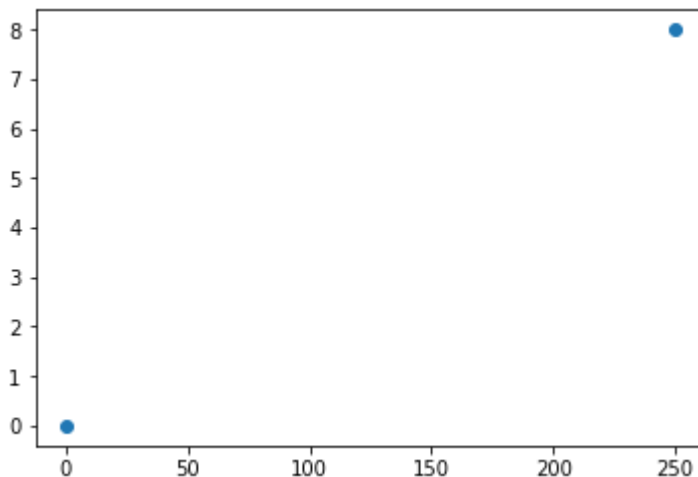


```
In [1]: #import package for matplotlib and numpy
import matplotlib.pyplot as plt
import numpy as np
```

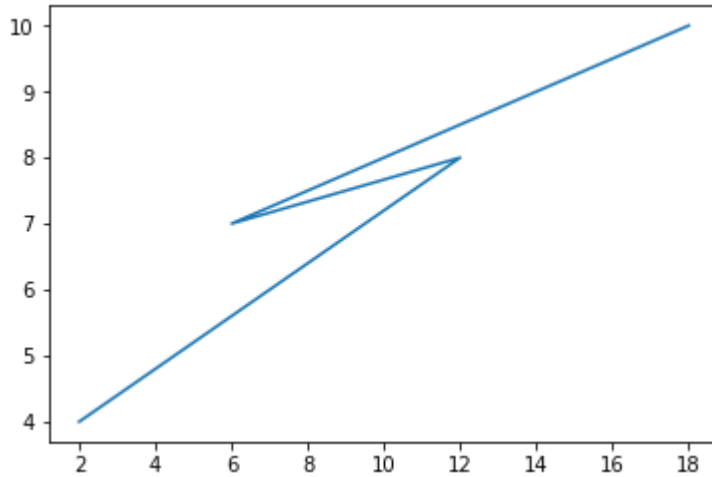
```
In [2]: #Draw a Line in a diagram with x and y points
xpoints = np.array([0,9])
ypoints = np.array([0,225])
plt.plot(xpoints,ypoints)
plt.show()
```



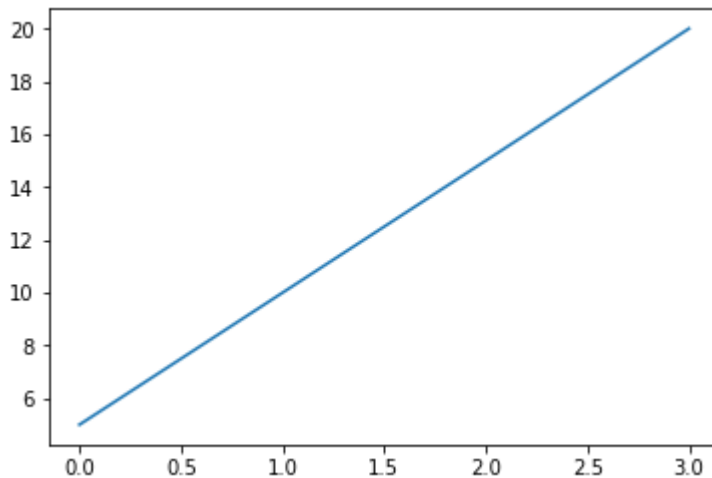
```
In [3]: #Draw two points in the diagram without a line
x=np.array([0,250])
y=np.array([0,8])
plt.plot(x,y, 'o')
plt.show()
```



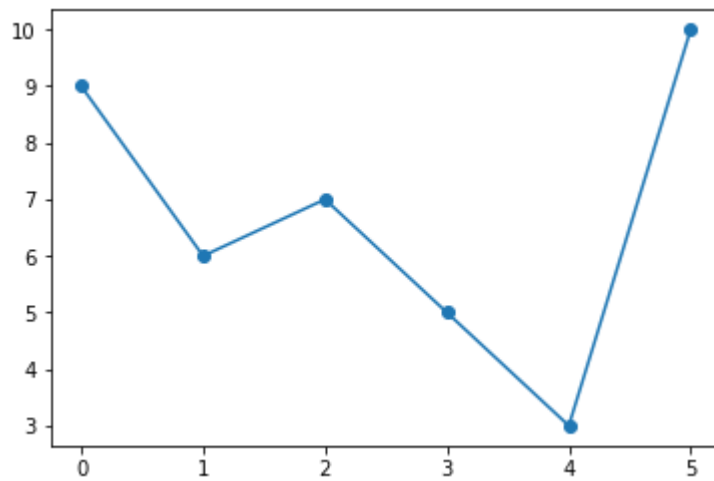
```
In [4]: 1 #Plot many points in the diagram with a line
2 xpoints = np.array([2, 12, 6, 18])
3 ypoints = np.array([4, 8, 7, 10])
4
5 plt.plot(xpoints, ypoints)
6 plt.show()
7
```



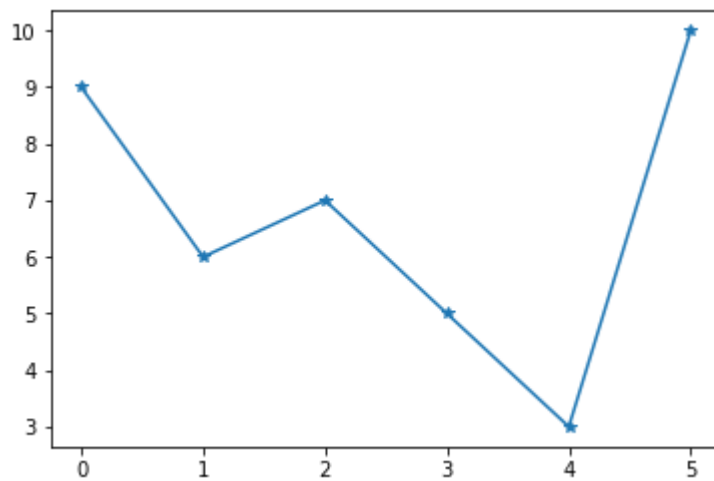
```
In [5]: #plotting without x points
y=np.array([5,10,15,20])
plt.plot(y)
plt.show()
```



```
In [6]: # To use the keyword argument marker to emphasize each point with a specified marker  
x=np.array([9,6,7,5,3,10])  
plt.plot(x,marker='o')  
plt.show()
```

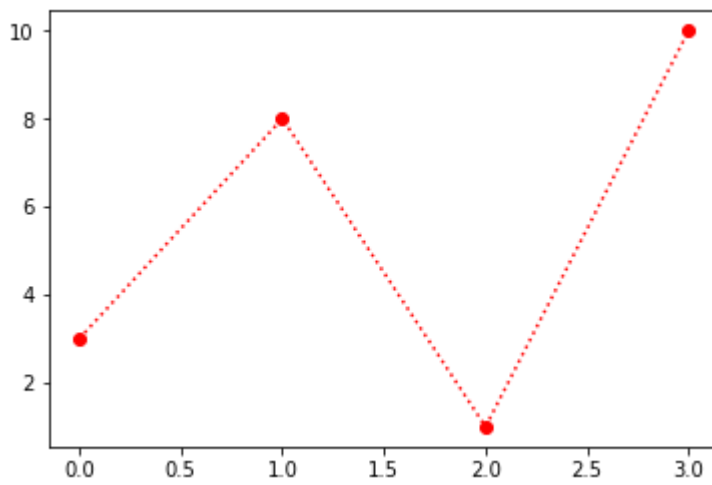


```
In [8]: plt.plot(x,marker='*')  
plt.show()
```

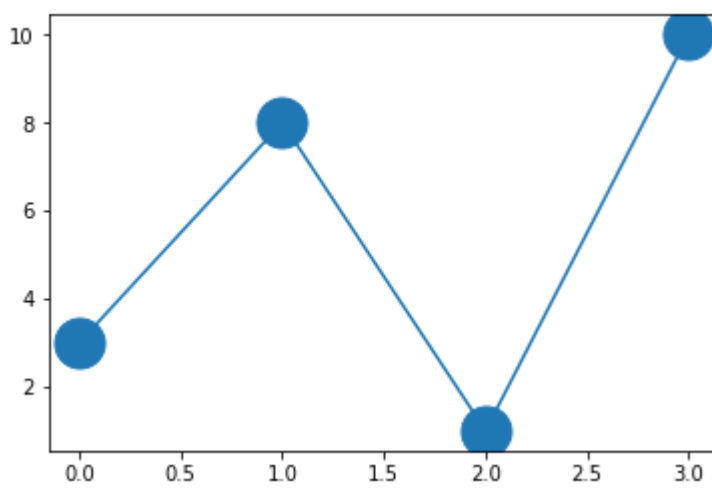


```
In [10]: #Specify the marker using color and line syntax
ypoints = np.array([3, 8, 1, 10])

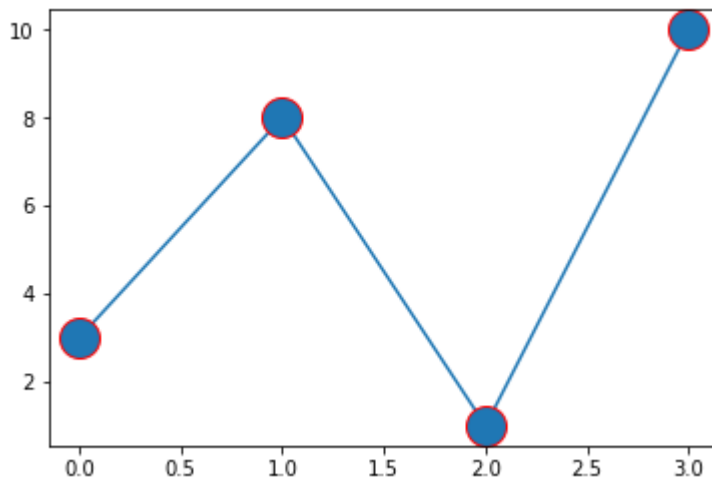
plt.plot(ypoints, 'o:r')
plt.show()
```



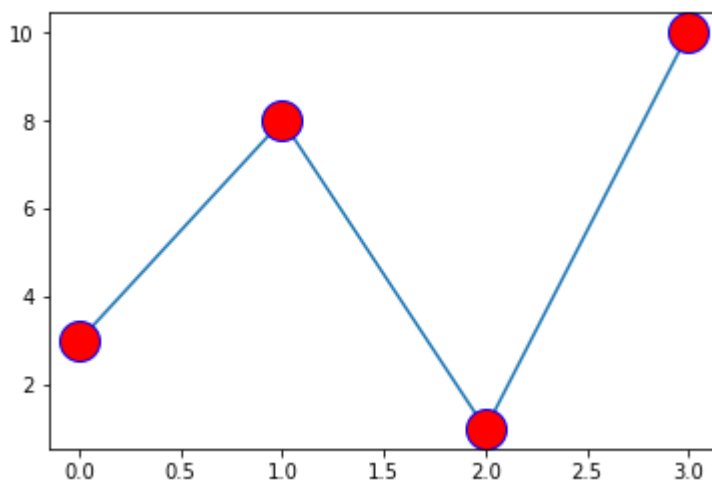
```
In [12]: #To specify the marker size
plt.plot(ypoints, marker = 'o', ms = 25)
plt.show()
```



```
In [14]: plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r')  
plt.show()
```

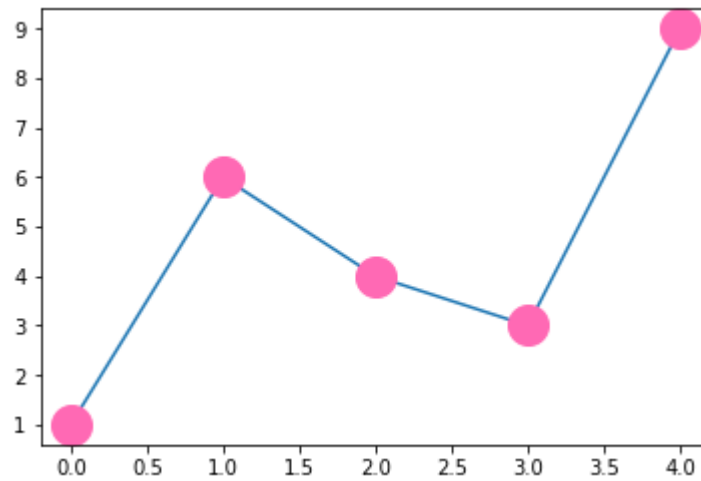


```
In [15]: #To use both the mec and mfc arguments to color of the entire marker:  
ypoints = np.array([3, 8, 1, 10])  
  
plt.plot(ypoints, marker = 'o', ms = 20, mec = 'b', mfc = 'r')  
plt.show()
```

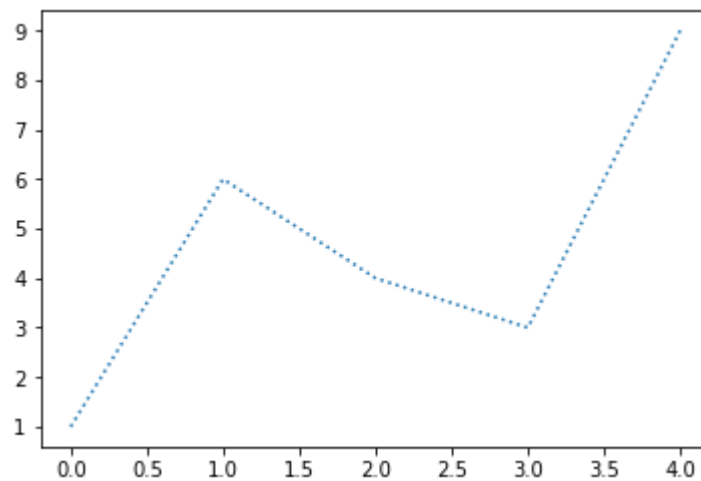


```
In [16]: ypoints=np.array([1,6,4,3,9])  
plt.plot(ypoints, marker = 'o', ms = 20, mec = 'hotpink', mfc = 'hotpink')
```

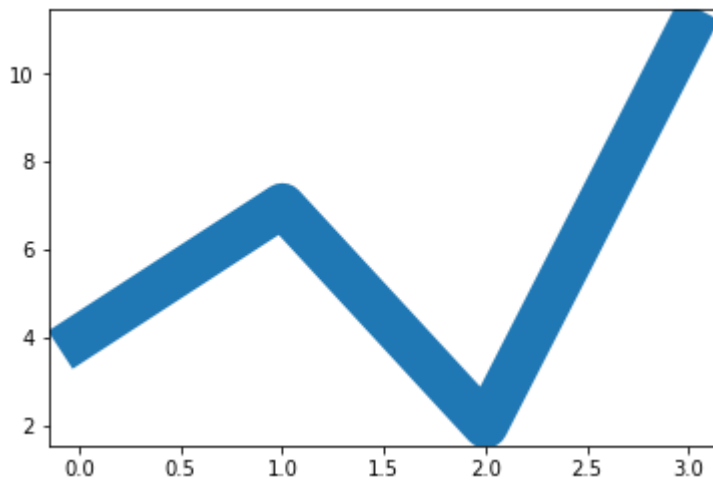
```
Out[16]: [<matplotlib.lines.Line2D at 0x76afb50>]
```



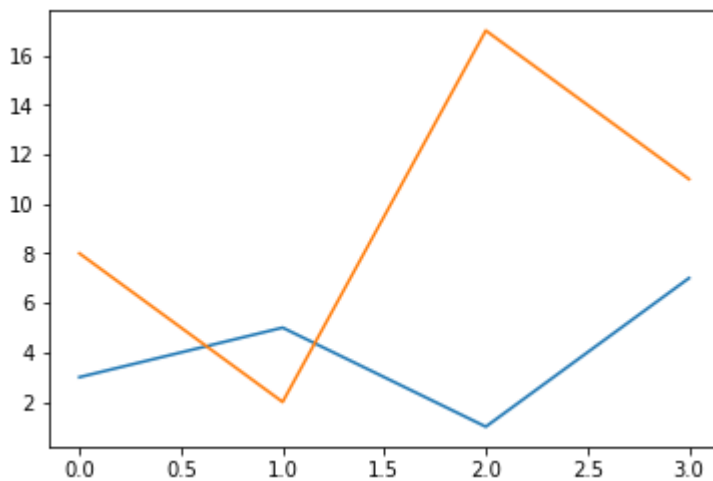
```
In [17]: #To use the keyword argument linestyle, or shorter ls, to change the style of the  
plt.plot(ypoints, linestyle = 'dotted')  
plt.show()
```



```
In [18]: #To use the keyword argument linewidth or the shorter lw to change the width of the line  
ypoints = np.array([4, 7, 2, 11])  
  
plt.plot(ypoints, Lw = '22.5')  
plt.show()
```



```
In [19]: #To plot multiple lines  
y1 = np.array([3, 5, 1, 7])  
y2 = np.array([8, 2, 17, 11])  
  
plt.plot(y1)  
plt.plot(y2)  
  
plt.show()
```

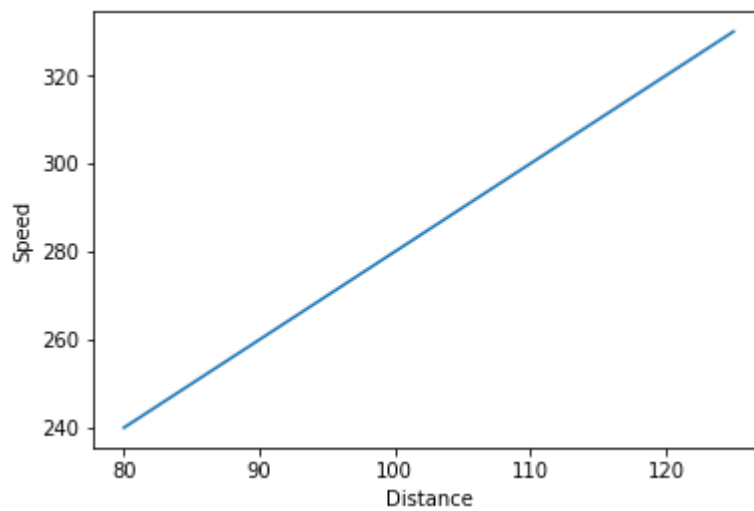


```
In [22]: #Add label to x and y axis
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.plot(x, y)

plt.xlabel("Distance")
plt.ylabel("Speed")
```

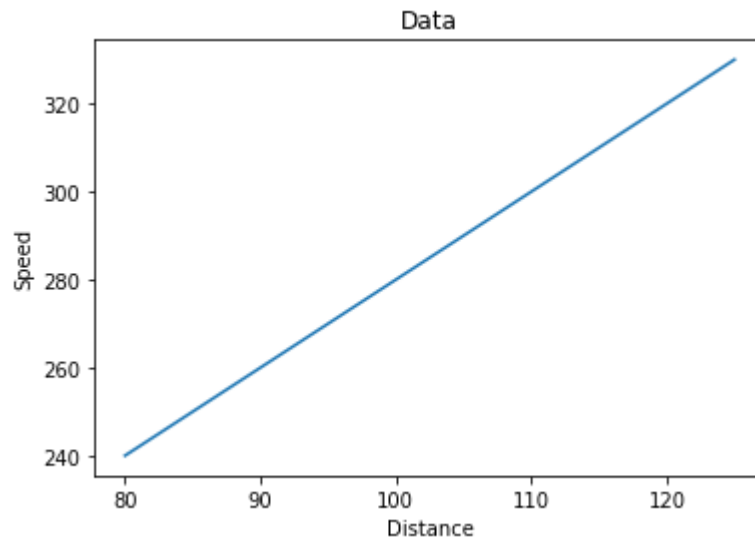
Out[22]: Text(0,0.5,'Speed')




```
In [24]: #To create title for a plot
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.plot(x, y)

plt.xlabel("Distance")
plt.ylabel("Speed")
plt.title("Data")
plt.show()
```

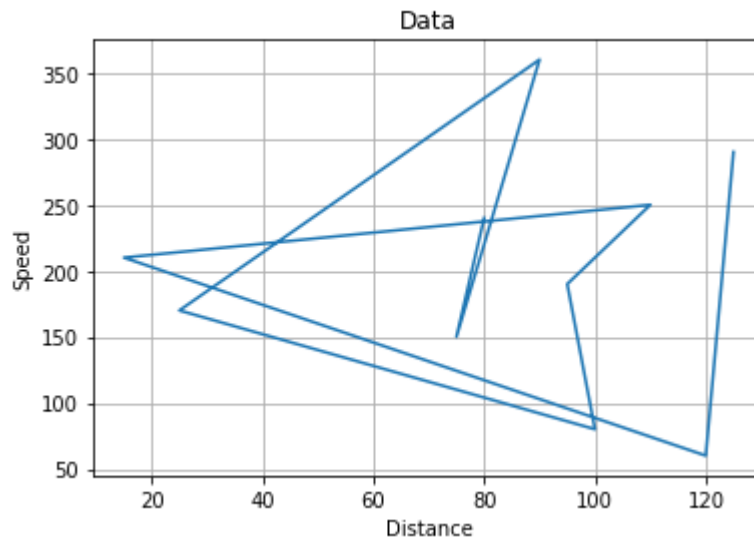


```
In [25]: #To add grid lines to the plot
x = np.array([80, 75, 90, 25, 100, 95, 110, 15, 120, 125])
y = np.array([240, 150, 360, 170, 80, 190, 250, 210, 60, 290])

plt.xlabel("Distance")
plt.ylabel("Speed")
plt.title("Data")
plt.plot(x, y)

plt.grid()

plt.show()
```

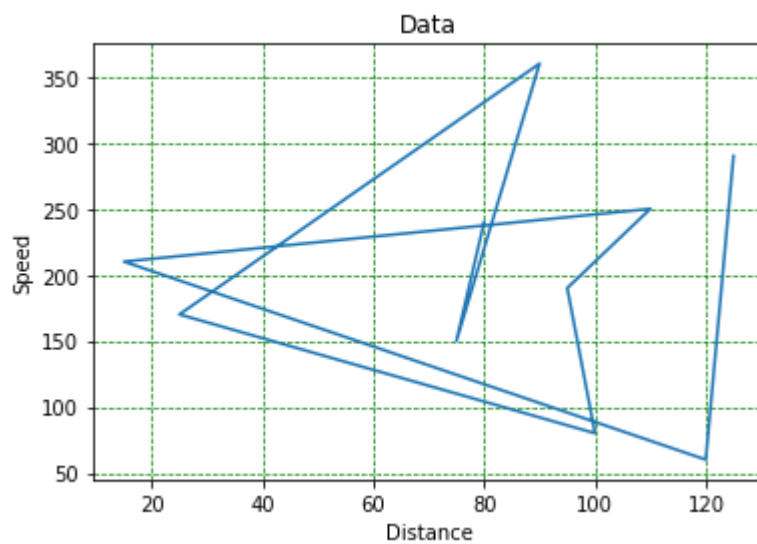


```
In [26]: #Set Line Properties for the Grid
x = np.array([80, 75, 90, 25, 100, 95, 110, 15, 120, 125])
y = np.array([240, 150, 360, 170, 80, 190, 250, 210, 60, 290])

plt.xlabel("Distance")
plt.ylabel("Speed")
plt.title("Data")
plt.plot(x, y)

plt.grid(color = 'green', linestyle = '--', linewidth = 0.7)

plt.show()
```



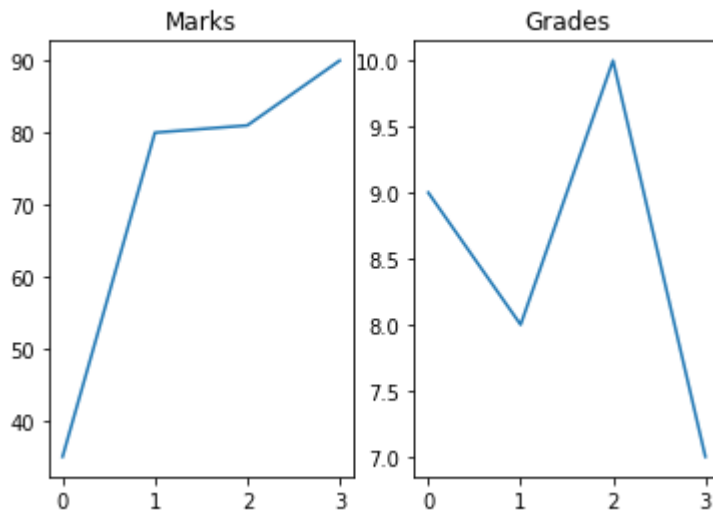
```
In [27]: #To display multiple plots
#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([35, 80, 81, 90])

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

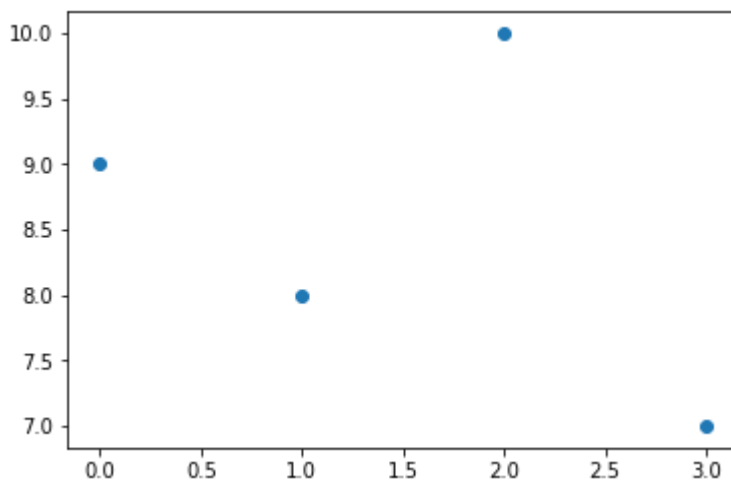
#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([9, 8, 10, 7])

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

plt.show()
```



```
In [28]: #To create scatter plots
plt.scatter(x,y)
plt.show()
```

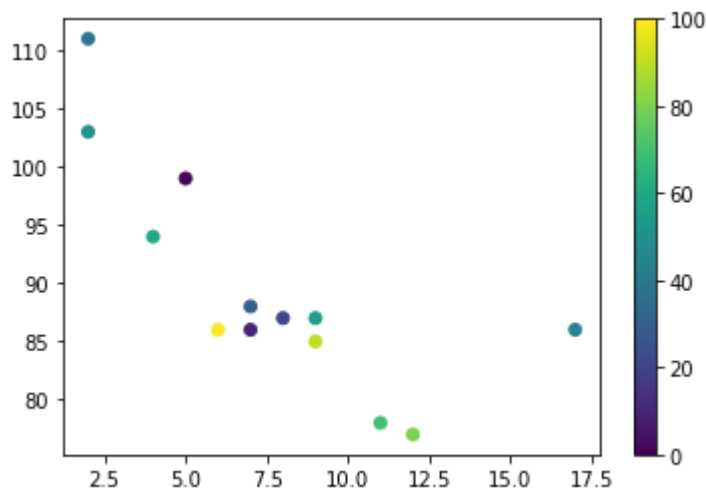


```
In [29]: #include color map in the diagram
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
colors = np.array([0, 10, 20, 30, 40, 45, 50, 55, 60, 70, 80, 90, 100])

plt.scatter(x, y, c=colors, cmap='viridis')

plt.colorbar()

plt.show()
```

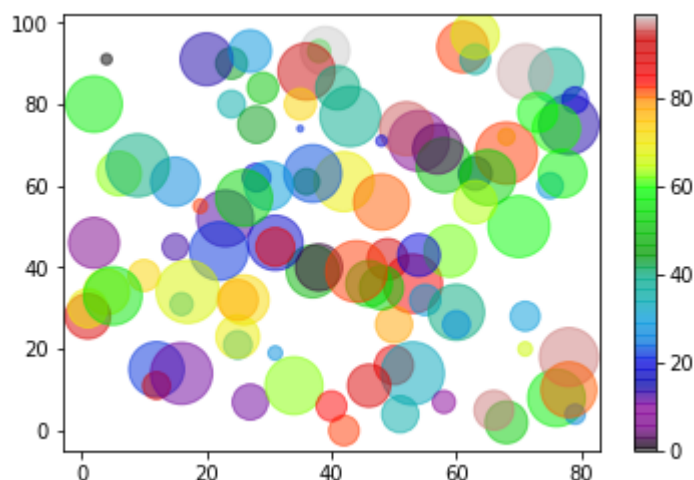


```
In [30]: #To combine colorsize and alpha
x = np.random.randint(80, size=(100))
y = np.random.randint(100, size=(100))
colors = np.random.randint(100, size=(100))
sizes = 10 * np.random.randint(100, size=(100))

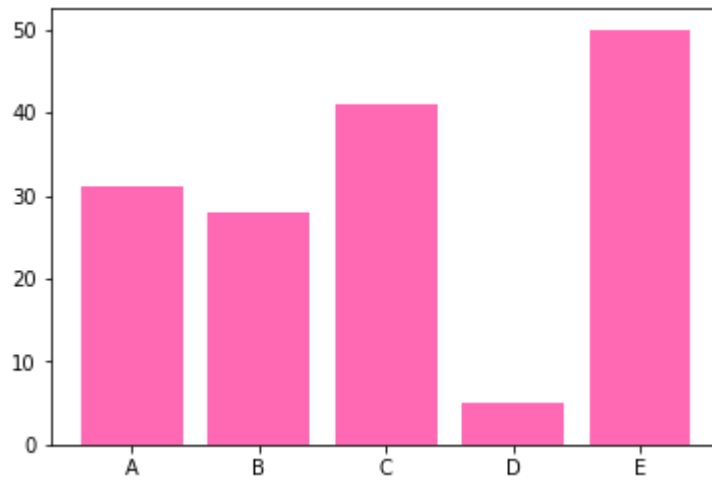
plt.scatter(x, y, c=colors, s=sizes, alpha=0.5, cmap='nipy_spectral')

plt.colorbar()

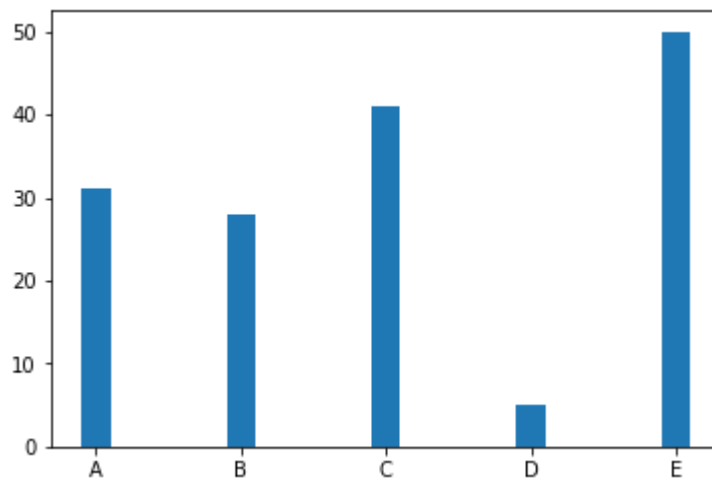
plt.show()
```



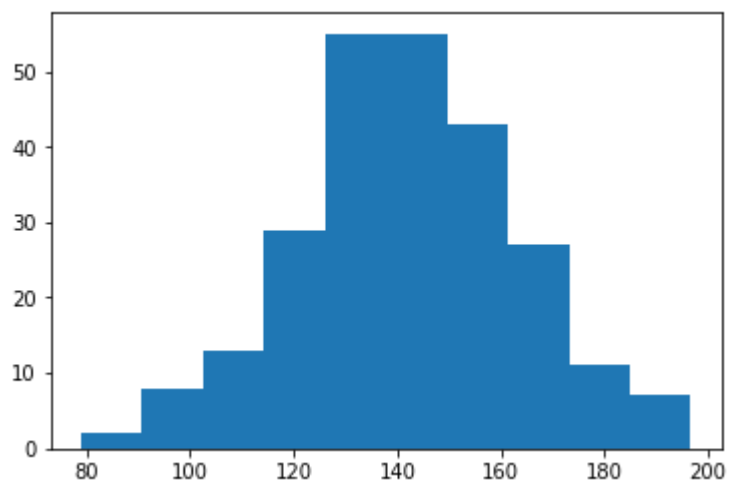
```
In [31]: #To create bar in the diagram using bar() function  
x = np.array(["A", "B", "C", "D", "E"])  
y = np.array([31, 28, 41, 5, 50])  
  
plt.bar(x, y, color = "hotpink")  
plt.show()
```



```
In [34]: plt.bar(x, y, width = 0.2)  
plt.show()
```

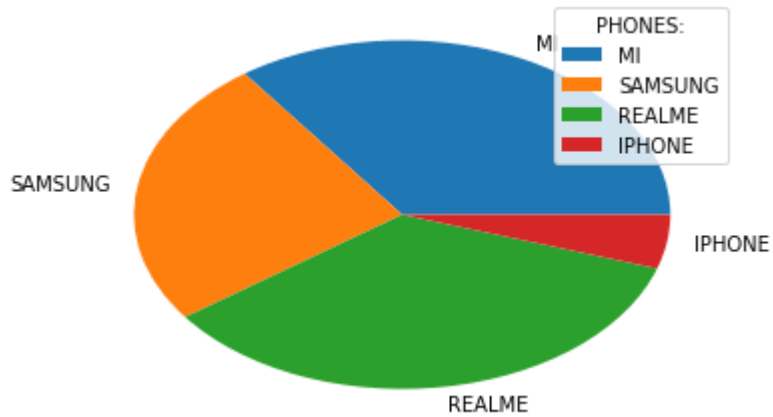


```
In [35]: #To create histogram using hist() function  
x = np.random.normal(140, 20, 250)  
  
plt.hist(x)  
plt.show()
```



```
In [36]: #To create a piechart
y = np.array([35, 25, 35, 5])
mylabels = ["MI", "SAMSUNG", "REALME", "IPHONE"]

plt.pie(y, labels = mylabels)
plt.legend(title = "PHONES:")
plt.show()
```



In []: