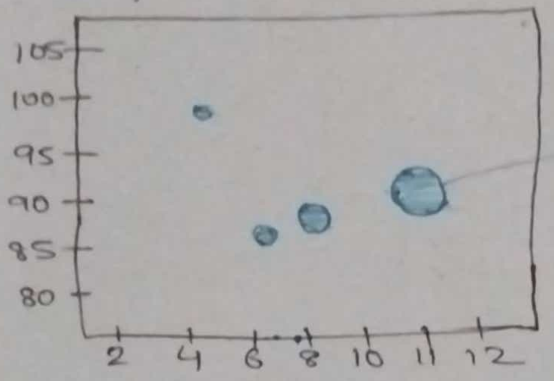
Alpha

You can adjust the transparency of the dots with the alpha argument.

Just like colors, make sure the array for sizes has the ~~samy~~ same length as the arrays for the x- and y-axis.

Set your own size for the markers:

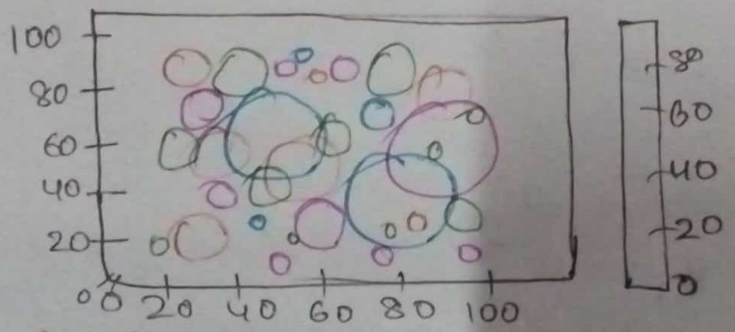
```
x =np.array([5,7,8,11])
y=np.array([99,86,87,90])
sizes = np.array([20,50,100,200])
plt.scatter(x,y, s= sizes, alpha =0.5)
plt.show()
```



It is transparent

Combine Color Size and Alpha

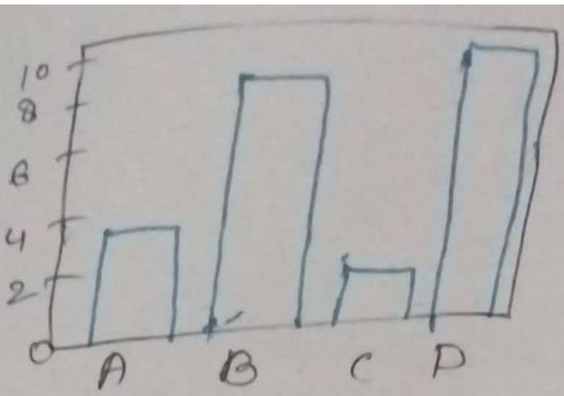
```
x= np.random.randint(100, size =(100))
y = np.random.randint (100, size =(100))
colors = np.random.randint(100, size =(100))
sizes = 10* np.random.randint(100, size=(100))
plt.scatter(x,y, c=colors, s= sizes, alpha=0.5,
                               cmap='nipy_spectral')
plt.colorbar()
plt.show()
```



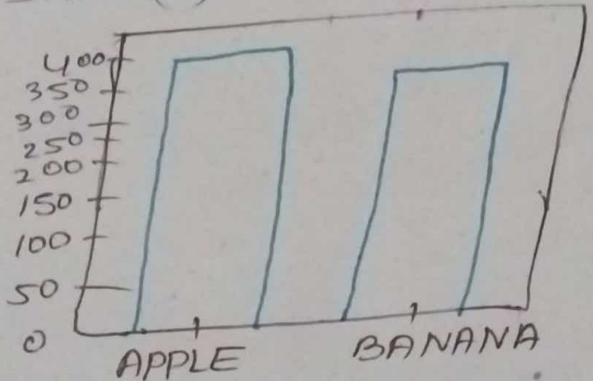
Matplotlib Bars

~~It~~ ~~[illegible]~~ ~~py~~ With pyplot() we can create bar graphs, we can use the bar() function to draw bar graphs.

```
x =np.array(['A','B','C','D'])
y= np.array([3,8,1,10])
plt.bar(x,y)
plt.show()
```



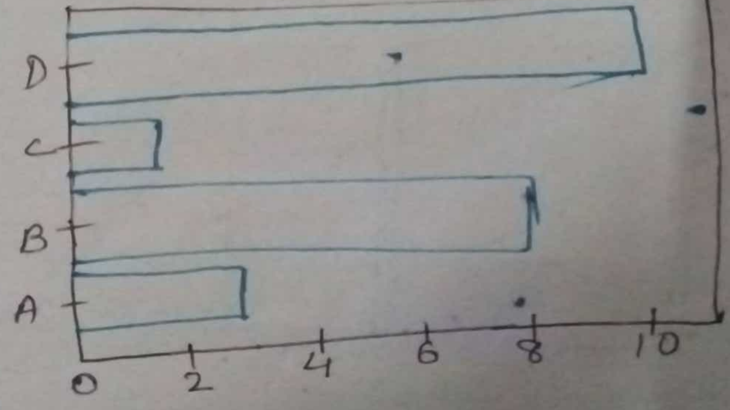
```
x = ['APPLE', 'BANANA']
y = [400, 350]
plt.bar(x,y)
plt.show()
```



Horizontal Bars

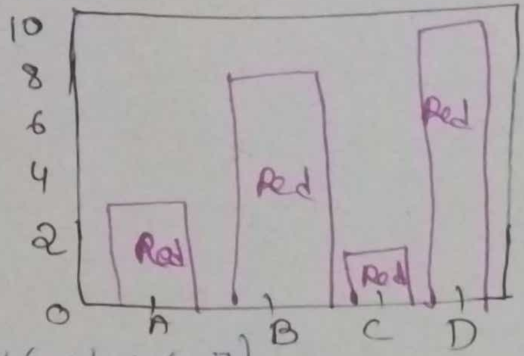
If you want the bars to be displayed horizontally instead of vertically, use the barh() function.

```
x = np.array(['A', 'B', 'C', 'D'])
y = np.array([3, 8, 1, 10])
plt.barh (x,y)
plt.show( )
```

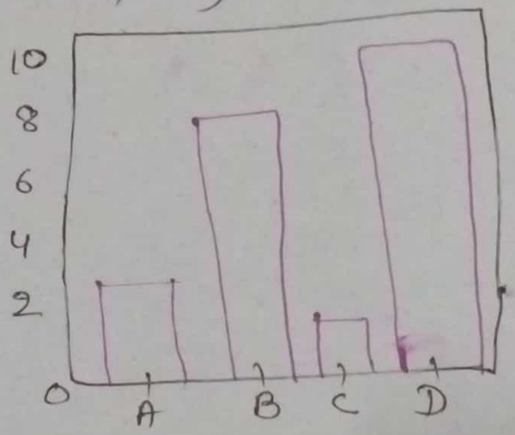


Bar Color

```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])
plt.bar(x,y, color= "red")
plt.show()
```

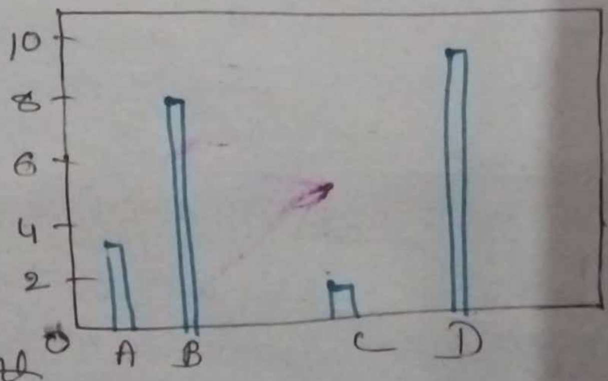


```
x=np.array(['A','B','C','D'])
y=np.array([3,8,1,10])
plt.bar(x,y, color='hotpink')
plt.show()
```



Bar Width

The bar() takes the keyword argument width to set the width of the bars.

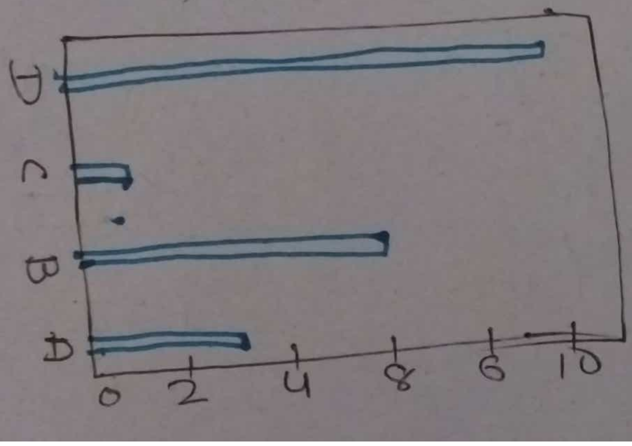


~~Barrheight~~

The default width value is 0.8

Barr Height

```
x = np.array(['A','B','C','D'])
Y = np.array([3, 8, 1, 10])
plt.barh(x,y, height = 0.1)
plt.show()
```



D
C
B
A
0
2
4
8
6
10