

Matplotlib is a Python 2D plotting library for creating 2D plots of arrays in Python. Matplotlib is written in Python and makes use of NumPy arrays. It is well integrated with pandas to read columns and create plots.

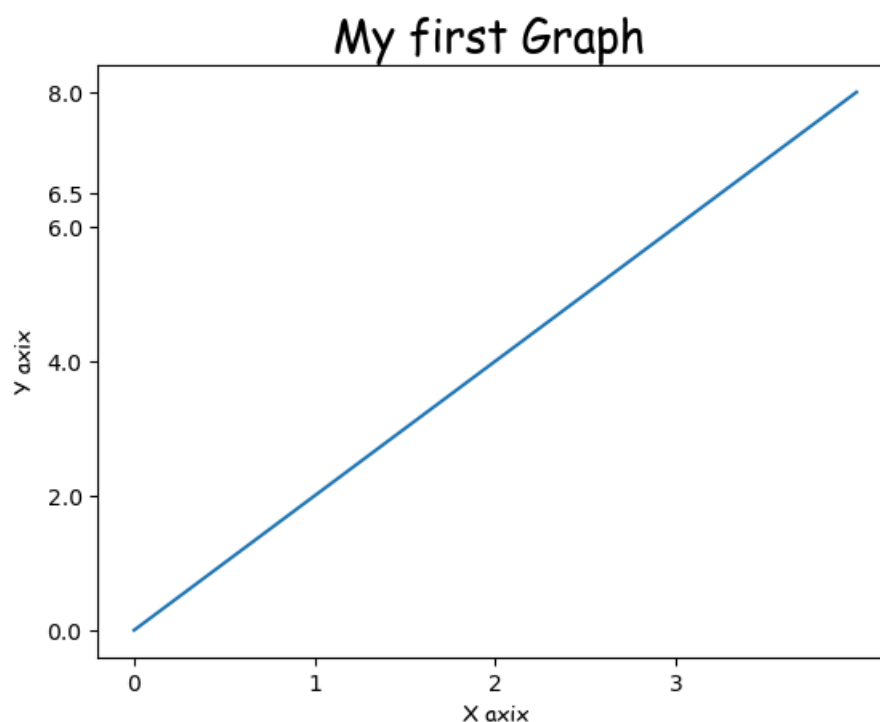
```
In [1]: ▶ import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

Basic Graph

```
In [2]: ▶ plt.title('My first Graph ', fontdict={'fontname':'Comic Sans MS','fontsize':20 })
plt.xlabel('X axis',fontdict={'fontname':'Comic Sans MS'})
plt.ylabel('Y axis' , fontdict={'fontname':'Comic Sans MS'})

x=[0,1,2,3,4]
y=[0,2,4,6,8]

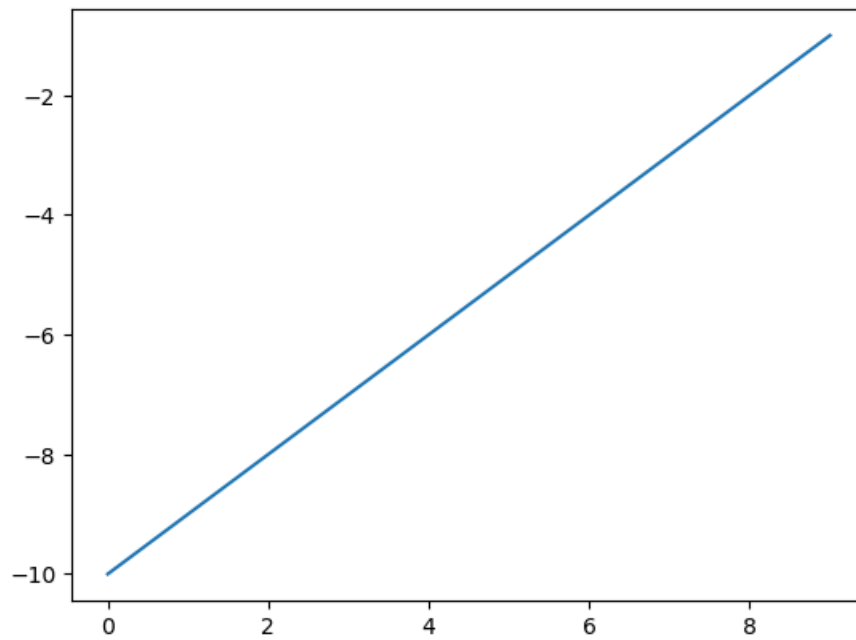
plt.xticks([0,1,2,3])
plt.yticks([0,2,4,6,6.5,8])
plt.plot(x,y)
#plt.plot(x,y,color="red",linewidth =2,marker='.',linestyle='--',markersize=10,markeredgecolor='blue')
plt.show()
```



```
In [3]: ▶ x=list(range(0,10))
y=list(range(-10,0))
```

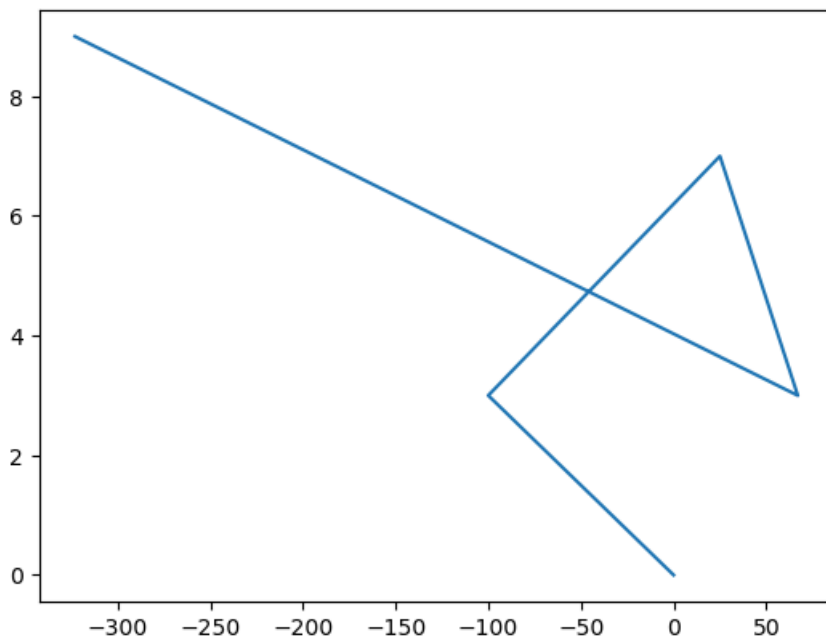
```
In [4]: ▶ plt.plot(x,y)
```

```
Out[4]: [<matplotlib.lines.Line2D at 0x14d422b7cd0>]
```



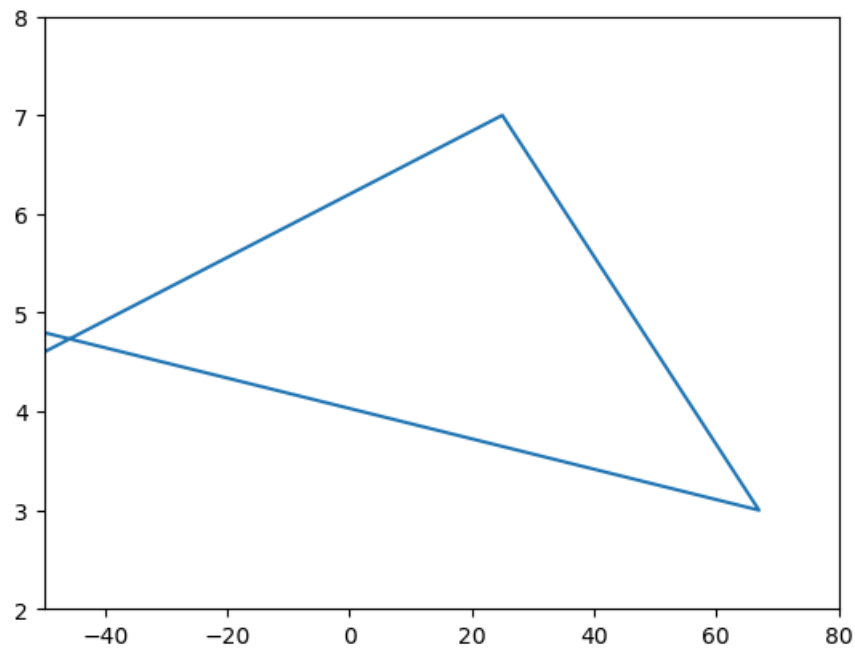
```
In [5]: ▶ a=[0,-100,25,67,-323]  
b=[0,3,7,3,9]  
plt.plot(a,b)
```

```
Out[5]: [<matplotlib.lines.Line2D at 0x14d42d48ee0>]
```



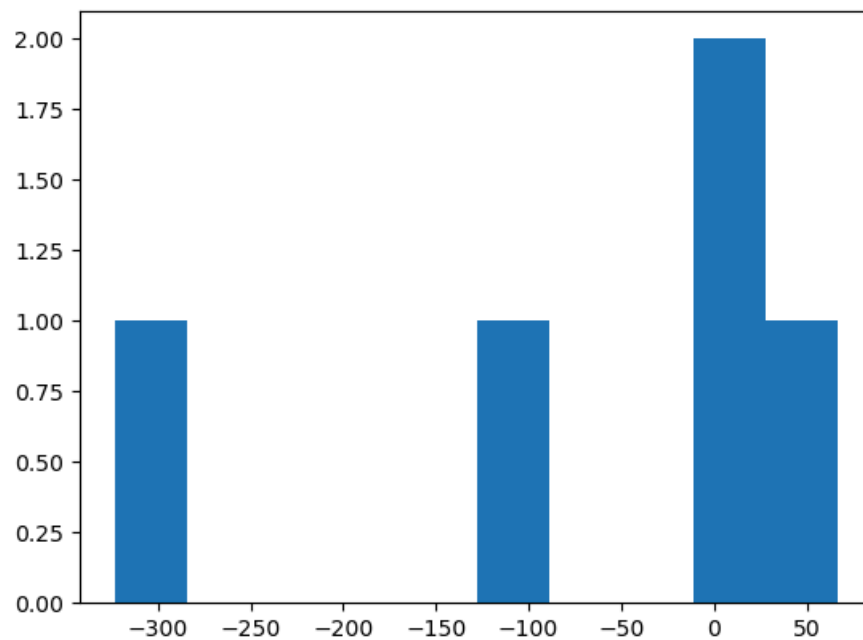
```
In [6]: ▶ plt.axis([-50,80,2,8])  
plt.plot(a,b)
```

```
Out[6]: [matplotlib.lines.Line2D at 0x14d423c25c0]
```



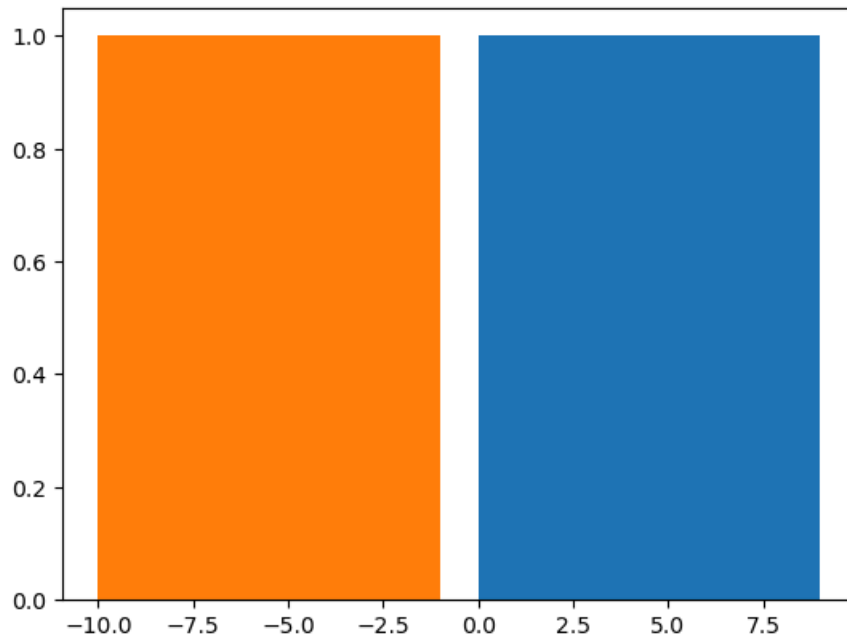
```
In [7]: ▶ plt.hist(a)
```

```
Out[7]: (array([1., 0., 0., 0., 0., 1., 0., 0., 2., 1.]),  
array([-323., -284., -245., -206., -167., -128., -89., -50., -11.,  
        28., 67.]),  
<BarContainer object of 10 artists>)
```



```
In [8]: ▶ plt.hist(x)
plt.hist(y)
```

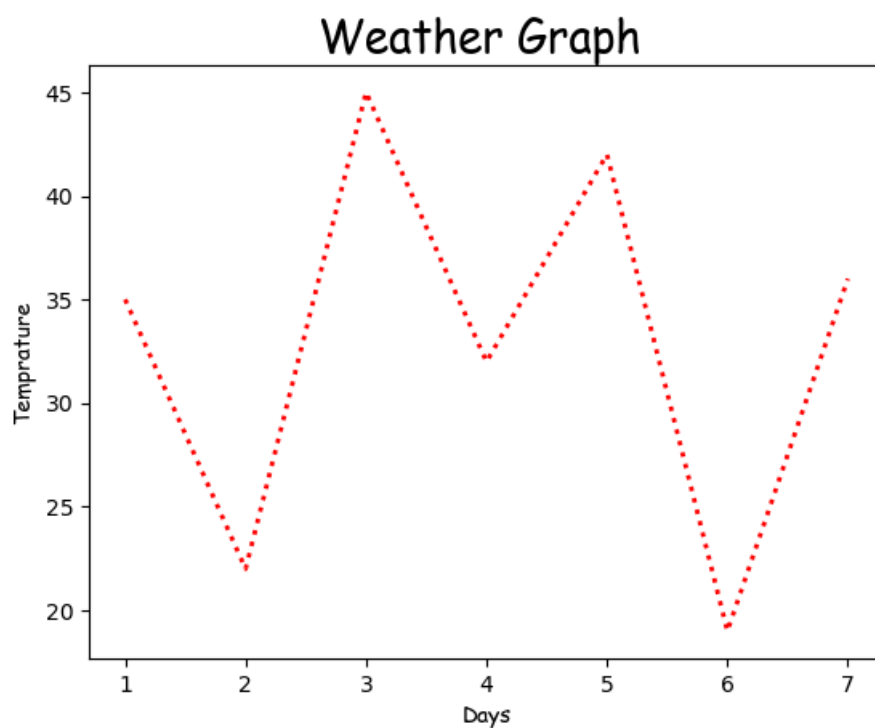
```
Out[8]: (array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]),
array([-10. , -9.1, -8.2, -7.3, -6.4, -5.5, -4.6, -3.7, -2.8,
       -1.9, -1. ]),
<BarContainer object of 10 artists>)
```



```
In [9]: ▶ plt.title('Weather Graph ', fontdict={'fontname':'Comic Sans MS','fontsize':20 })
plt.xlabel('Days',fontdict={'fontname':'Comic Sans MS'})
plt.ylabel('Temprature' , fontdict={'fontname':'Comic Sans MS'})

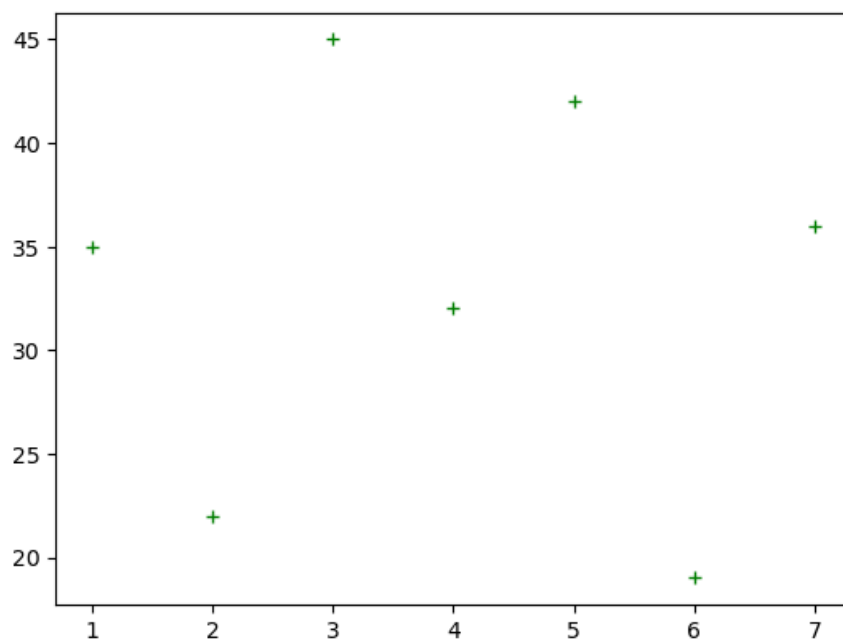
x=[1,2,3,4,5,6,7]
y=[35,22,45,32,42,19,36]

# plt.xticks([0,1,2,3])
# plt.yticks([0,2,4,6,6.5,8])
#plt.plot(x,y)
plt.plot(x,y,color="red",linewidth =2,linestyle='dotted')
plt.show()
```



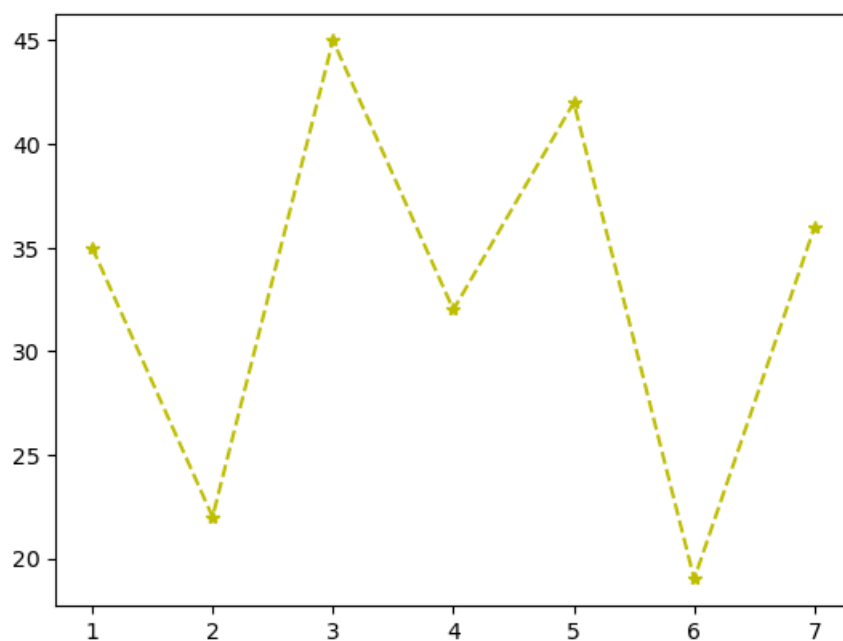
```
In [10]: ▶ plt.plot(x,y,'g+')
```

```
Out[10]: [matplotlib.lines.Line2D at 0x14d43f0ddb0>]
```



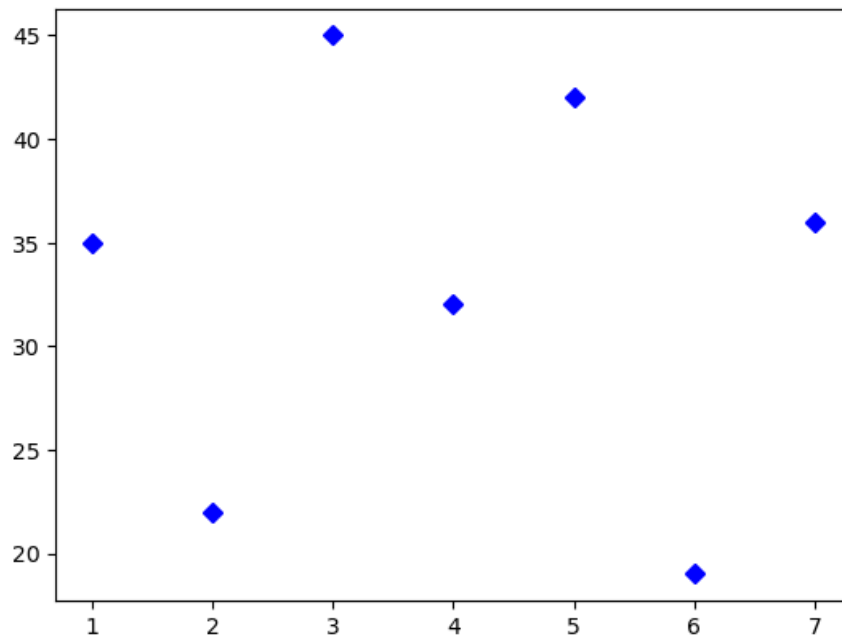
```
In [11]: ▶ plt.plot(x,y,'--y*') #star shaped
```

```
Out[11]: [matplotlib.lines.Line2D at 0x14d4247c820>]
```



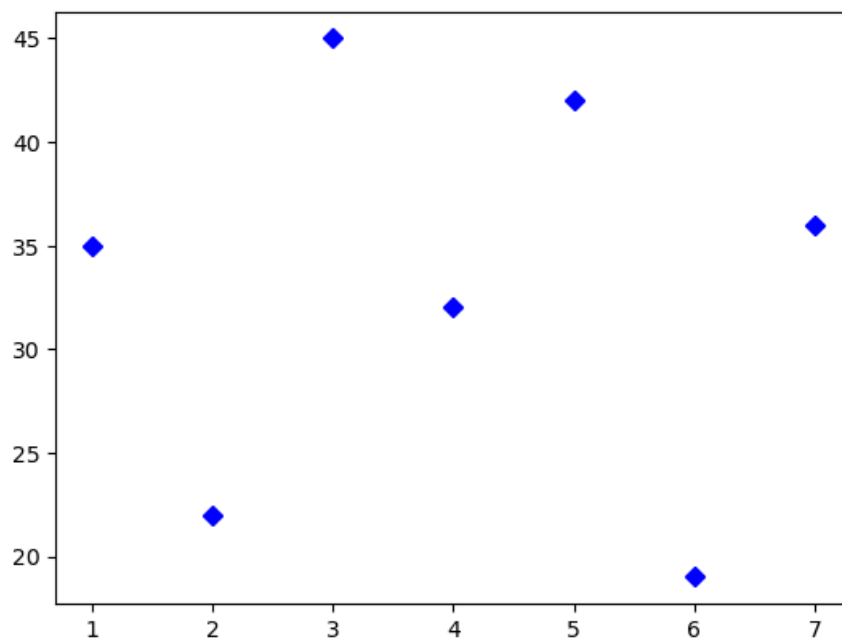
```
In [12]: ▶ plt.plot(x,y,'bD') #diamond shaped
```

```
Out[12]: [matplotlib.lines.Line2D at 0x14d43f873d0>]
```



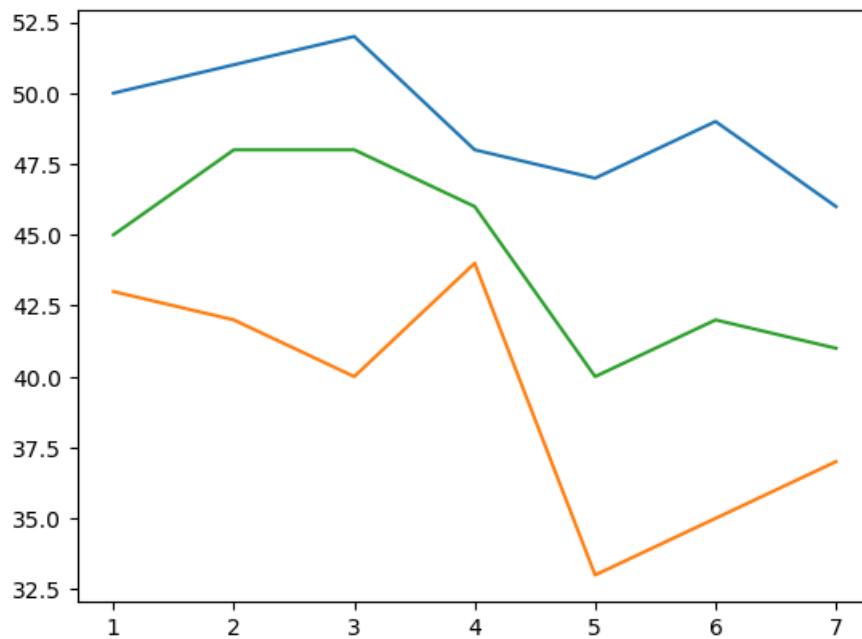
```
In [13]: ▶ plt.plot(x,y,color='blue', marker='D',linestyle='')
```

```
Out[13]: [matplotlib.lines.Line2D at 0x14d44014d30>]
```

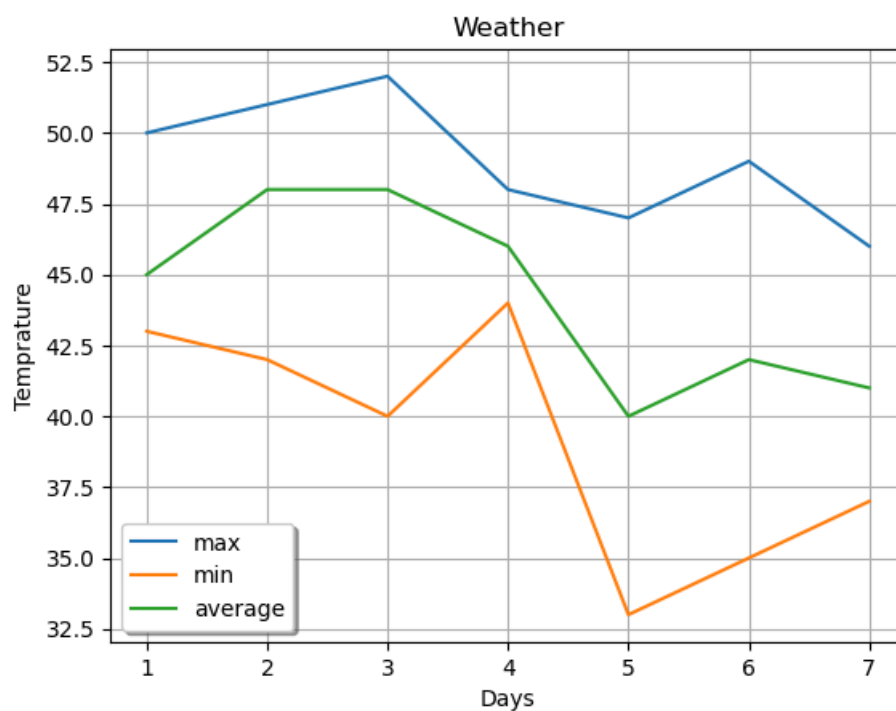


```
In [14]: ▶ days=[1,2,3,4,5,6,7]
max_t=[50,51,52,48,47,49,46]
min_t=[43,42,40,44,33,35,37]
avg_t=[45,48,48,46,40,42,41]
plt.plot(days,max_t)
plt.plot(days,min_t)
plt.plot(days,avg_t)
```

Out[14]: [<matplotlib.lines.Line2D at 0x14d4407b5e0>]



```
In [15]: ▶ plt.title('Weather')
plt.xlabel('Days')
plt.ylabel('Temprature')
plt.plot(days, max_t, label="max")
plt.plot(days, min_t, label="min")
plt.plot(days, avg_t, label="average")
#plt.legend()
#plt.legend(loc="upper right")
plt.legend(loc="best",shadow='true') #put the Legend at best area.
plt.grid()
```



BAR CHART

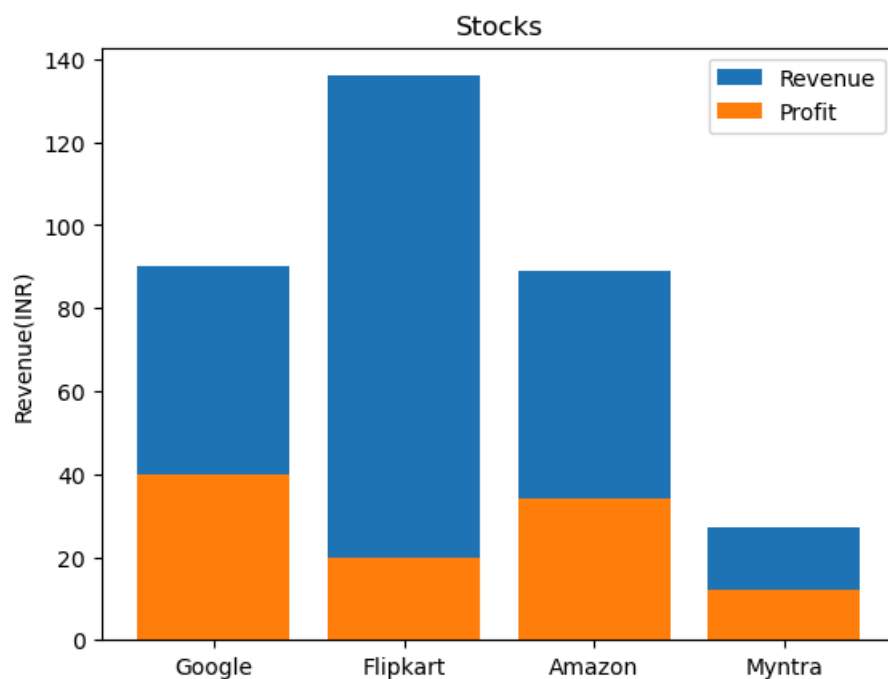
```
In [16]: ▶ company=['Google','Flipkart','Amazon','Myntra']  
revenue=[90,136,89,27]  
profit=[40,20,34,12]
```

```
In [17]: ▶ xpos=np.arange(len(company)) #conver into numbers.  
xpos
```

```
Out[17]: array([0, 1, 2, 3])
```

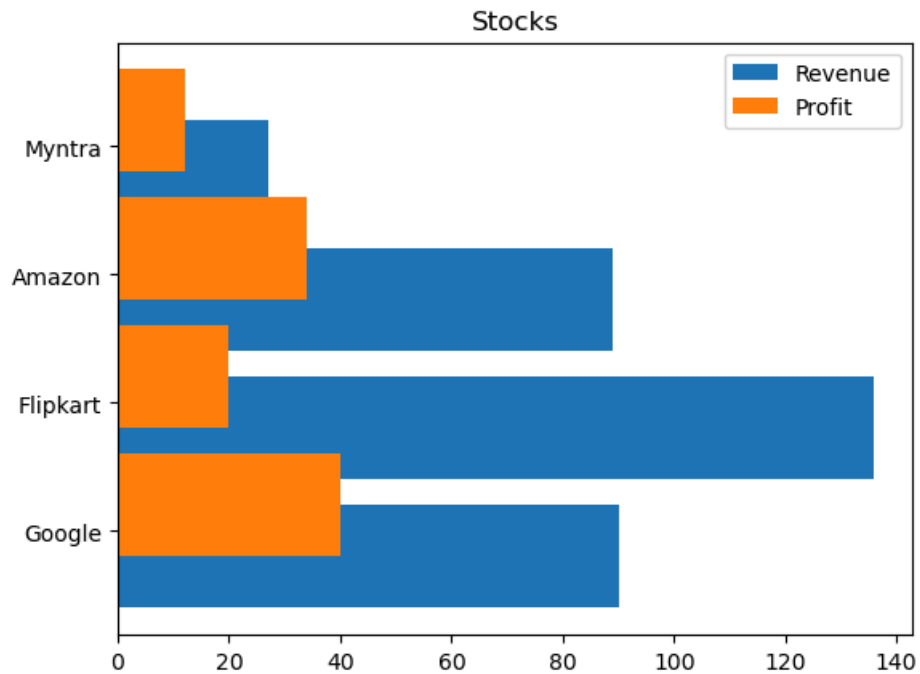
```
In [18]: ▶ plt.title('Stocks')  
plt.xticks(xpos,company)  
plt.ylabel('Revenue(INR)')  
plt.bar(xpos,revenue,label='Revenue')  
plt.bar(xpos,profit,label='Profit')  
plt.legend()
```

```
Out[18]: <matplotlib.legend.Legend at 0x14d42c0dc00>
```




```
In [19]: ► #Horizontal Bar Chart
plt.title('Stocks')
plt.yticks(xpos,company)
#plt.ylabel('Revenue(INR)')
# plt.bar(xpos-.2,revenue,width=.4,label='Revenue')
# plt.bar(xpos+.2,profit,width=.4,label='Profit')
plt.barh(xpos-.2,revenue,label='Revenue')
plt.barh(xpos+.2,profit,label='Profit')
plt.legend()
```

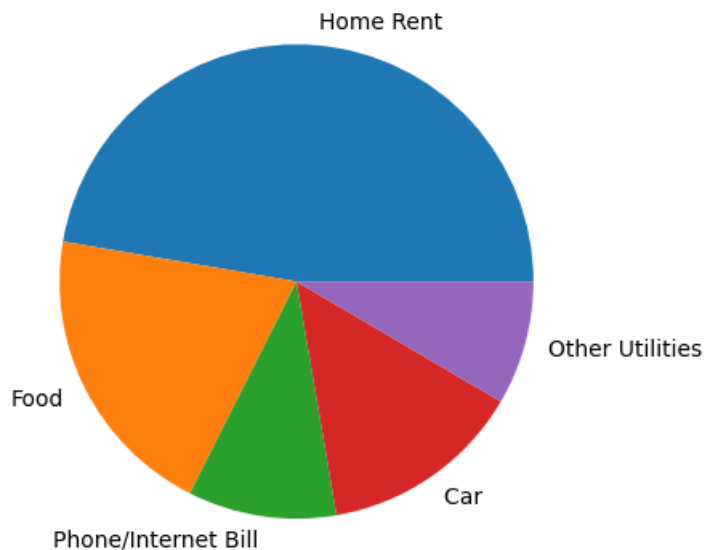
Out[19]: <matplotlib.legend.Legend at 0x14d440ad480>



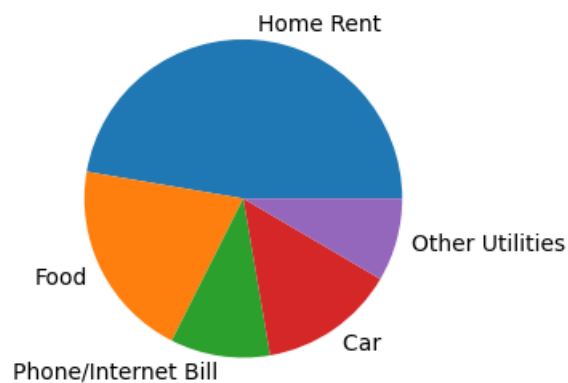
PI CHART

```
In [20]: exp_vals = [1400,600,300,410,250]
exp_labels = ["Home Rent","Food","Phone/Internet Bill","Car ","Other Utilities"]
plt.axis('equal')
plt.pie(exp_vals,labels=exp_labels)
```

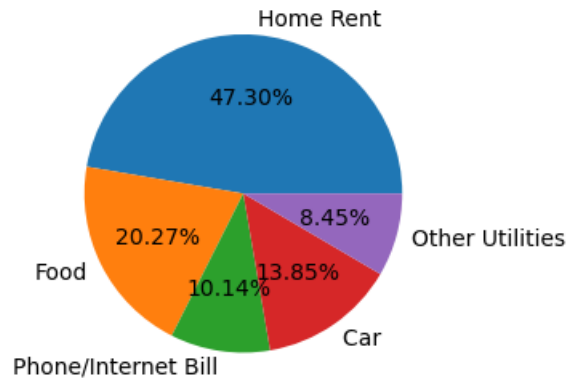
```
Out[20]: ([<matplotlib.patches.Wedge at 0x14d43e71e10>,
<matplotlib.patches.Wedge at 0x14d43e71d20>,
<matplotlib.patches.Wedge at 0x14d43e72560>,
<matplotlib.patches.Wedge at 0x14d43e729e0>,
<matplotlib.patches.Wedge at 0x14d43e72e60>],
[Text(0.09328656407206024, 1.0960372333838069, 'Home Rent'),
Text(-0.9822184890776084, -0.4952240298229684, 'Food'),
Text(-0.16284704617934698, -1.0878790555712807, 'Phone/Internet Bill'),
Text(0.6256100334857941, -0.9047718419590123, 'Car '),
Text(1.0615045230766318, -0.28845822485734873, 'Other Utilities')])
```



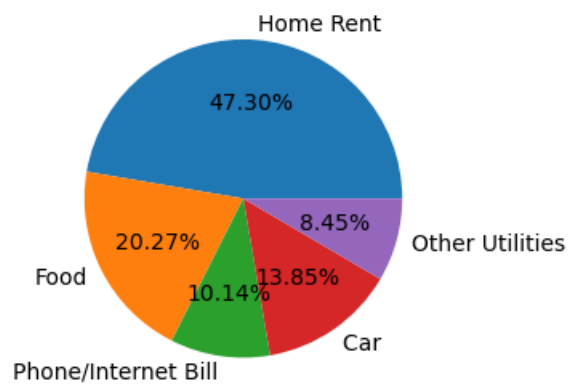
```
In [21]: exp_vals = [1400,600,300,410,250]
exp_labels = ["Home Rent","Food","Phone/Internet Bill","Car ","Other Utilities"]
plt.axis('equal')
plt.pie(exp_vals,labels=exp_labels,radius=.5)
plt.show()
```



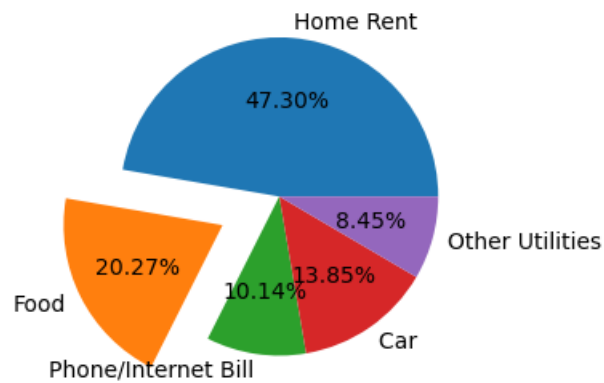
```
In [22]: ▶ exp_vals = [1400,600,300,410,250]
exp_labels = ["Home Rent","Food","Phone/Internet Bill","Car ","Other Utilities"]
plt.axis('equal')
plt.pie(exp_vals,labels=exp_labels,radius=.5,autopct='%1.2f%%') # percentage with after decimal two digit
#plt.pie(exp_vals,labels=exp_labels,radius=.5,autopct='%1.1f%%',shadow=True)
plt.show()
```



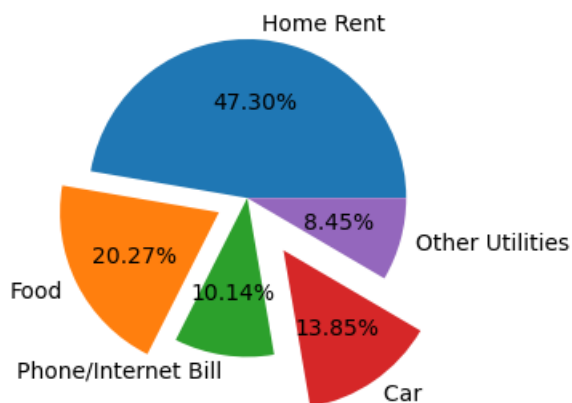
```
In [23]: ▶ exp_vals = [1400,600,300,410,250]
exp_labels = ["Home Rent","Food","Phone/Internet Bill","Car ","Other Utilities"]
plt.axis('equal')
plt.pie(exp_vals,labels=exp_labels,radius=.5,autopct='%1.2f%%',explode=[0,0,0,0,0])
# percentage with after decimal two digits.
plt.show()
```



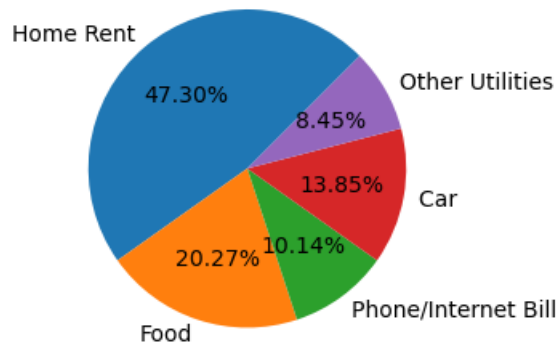
```
In [24]: ► exp_vals = [1400,600,300,410,250]
exp_labels = ["Home Rent","Food","Phone/Internet Bill","Car ","Other Utilities"]
plt.axis('equal')
plt.pie(exp_vals,labels=exp_labels,radius=.5,autopct='%1.2f%%',explode=[0,.2,0,0,0])
# percentage with after decimal two digits.
plt.show()
```



```
In [25]: ► exp_vals = [1400,600,300,410,250]
exp_labels = ["Home Rent","Food","Phone/Internet Bill","Car ","Other Utilities"]
plt.axis('equal')
plt.pie(exp_vals,labels=exp_labels,radius=.5,autopct='%1.2f%%',explode=[0,.1,0,.2,0])
# percentage with after decimal two digits.
plt.show()
```



```
In [26]: ► exp_vals = [1400,600,300,410,250]
exp_labels = ["Home Rent","Food","Phone/Internet Bill","Car ","Other Utilities"]
plt.axis('equal')
plt.pie(exp_vals,labels=exp_labels,radius=.5,autopct='%1.2f%%',explode=[0,0,0,0,0],startangle=45)
# percentage with after decimal two digits.
#explode --- to take out one value out of circle.
#plt.pie(exp_vals,labels=exp_labels,radius=.5,autopct='%1.2f%%',explode=[0,0,0,0,0],startangle=180)
plt.show()
#startangle --- to rotate piechart.
plt.savefig('piechart.jpeg', bbox_inches='tight',pad_inches=2) # savefig---to save this chart as an image
#pad_inches-----to make the border broad.
plt.savefig('piechart.pdf')
```



<Figure size 640x480 with 0 Axes>