Matplotlib

This project is all about Matplotlib, the basic data visualization tool of Python programming language. I have discussed Matplotlib object hierarchy, various plot types with Matplotlib and customization techniques associated with Matplotlib.

This project is divided into various sections based on contents which are listed below:-

Before, we need to actually start using Matplotlib, we need to import it. We can import Matplotlib as follows:-

```
import matplotlib
```

Most of the time, we have to work with **pyplot** interface of Matplotlib. So, I will import **pyplot** interface of Matplotlib as follows:-

```
import matplotlib.pyplot
```

To make things even simpler, we will use standard shorthand for Matplotlib imports as follows:-

```
import matplotlib.pyplot as plt
```

In [6]:
```
import matplotlib as pl
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
print(pl.__version__)
```

```
3.8.2
```

In [3]:
```
np.__version__
```

Out[3]:
```
'1.26.4'
```

In [13]:
```
import matplotlib.pyplot as plt
```

In [14]:
```
x=np.array([0,6])
```

In [15]:
```
x
```

Out[15]:
```
array([0, 6])
```

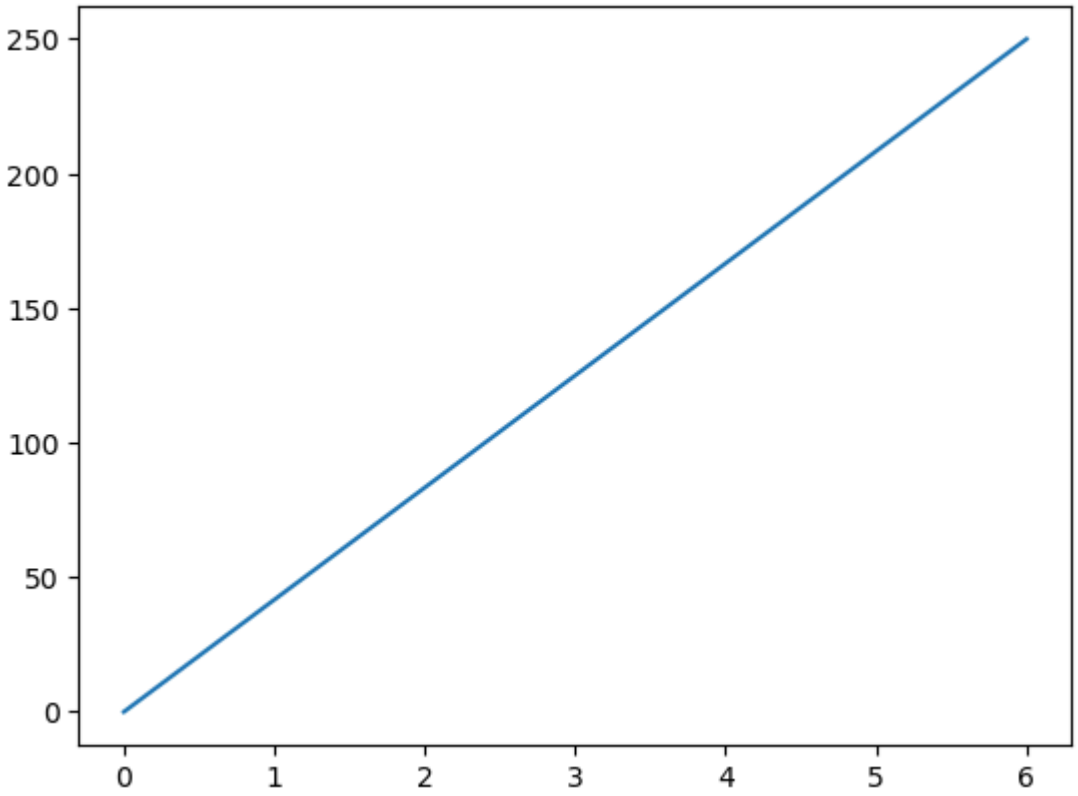
In [16]:
```
y=np.array([0,250] )
```

In [17]:
```
y
```

Out[17]:
```
array([  0, 250])
```

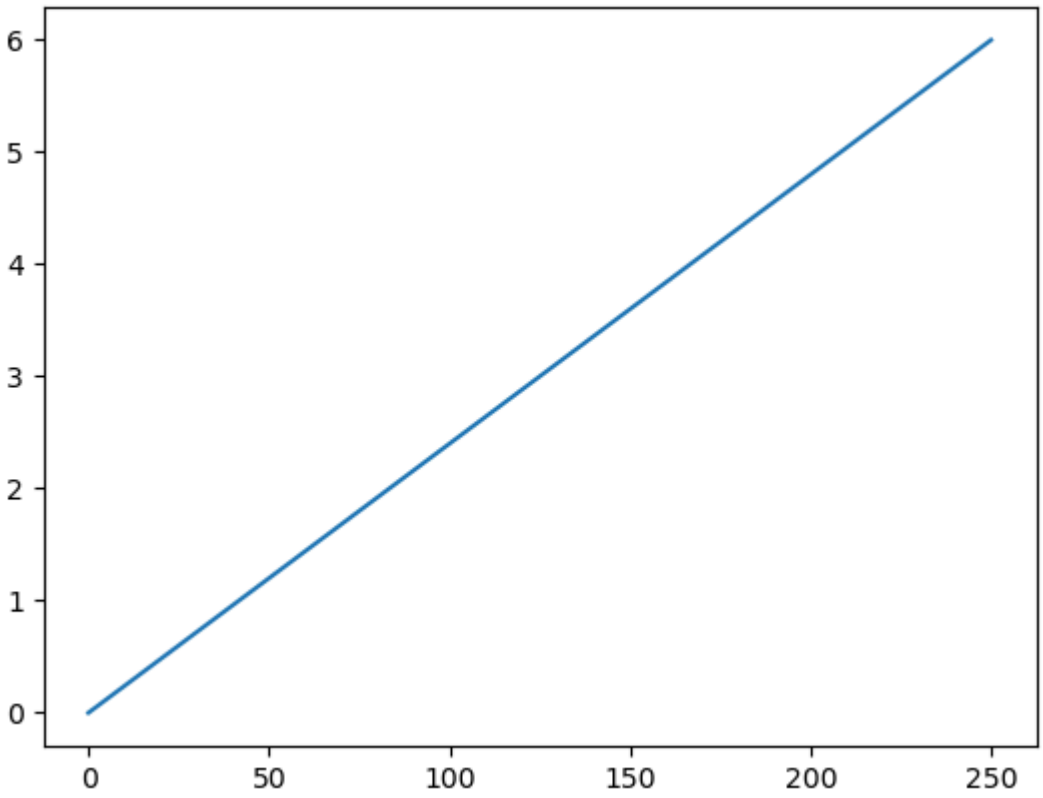
In [18]:
```
plt.plot(x,y)
```

Out[18]: [<matplotlib.lines.Line2D at 0x20836371210>]



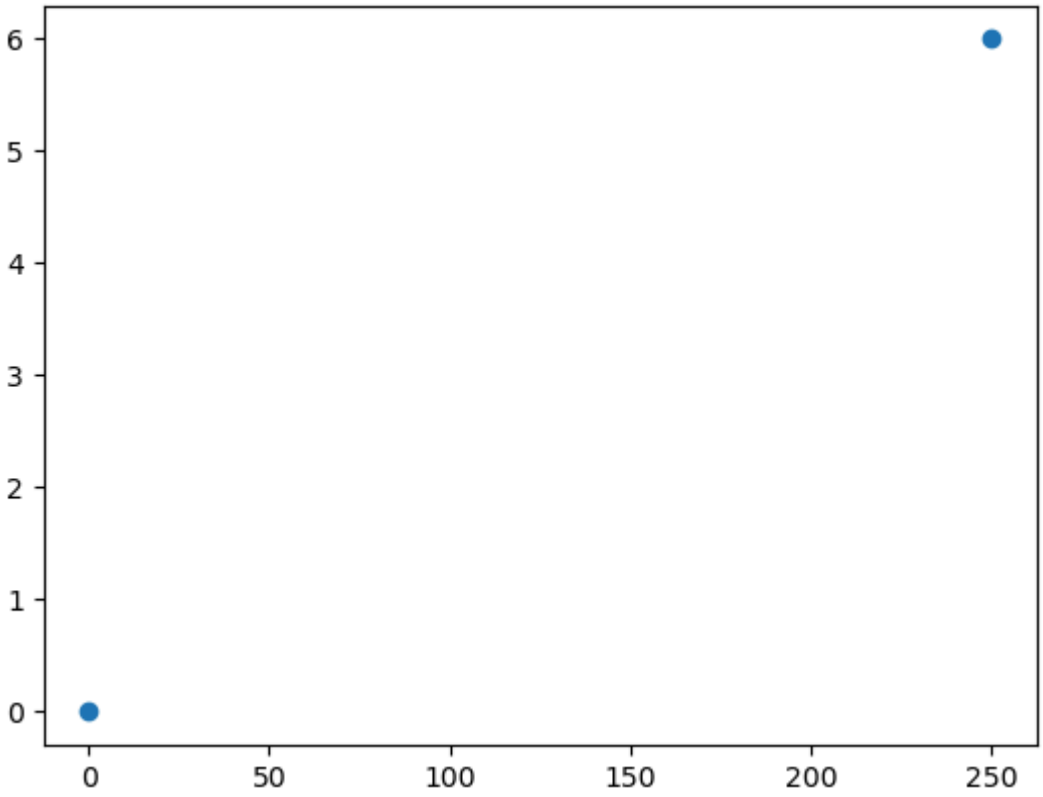
In [19]:
```
plt.plot(y,x)
```

Out[19]: [<matplotlib.lines.Line2D at 0x20836c1ba50>]



In [20]:
```
plt.plot(y,x,"o")
```

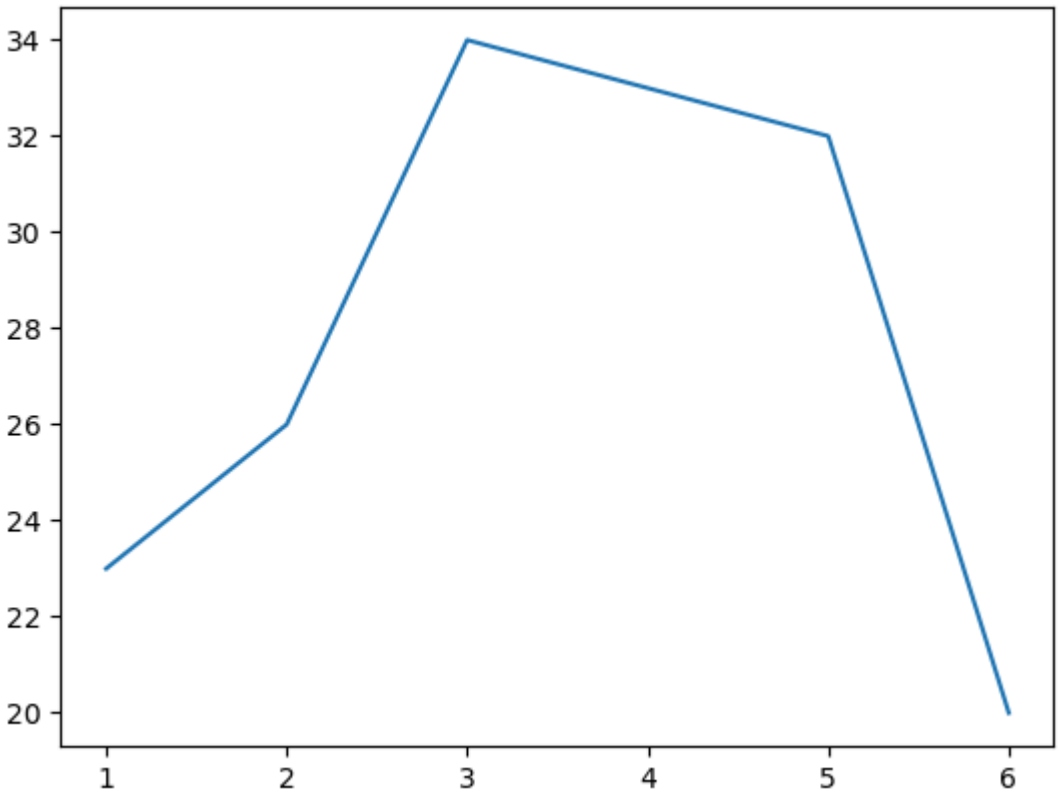
Out[20]: [<matplotlib.lines.Line2D at 0x20836c6c8d0>]



In [21]:
```
x1=np.array([1,2,3,5,6])
y1=np.array([23,26,34,32,20])
```

In [22]:
```
plt.plot(x1,y1)
```

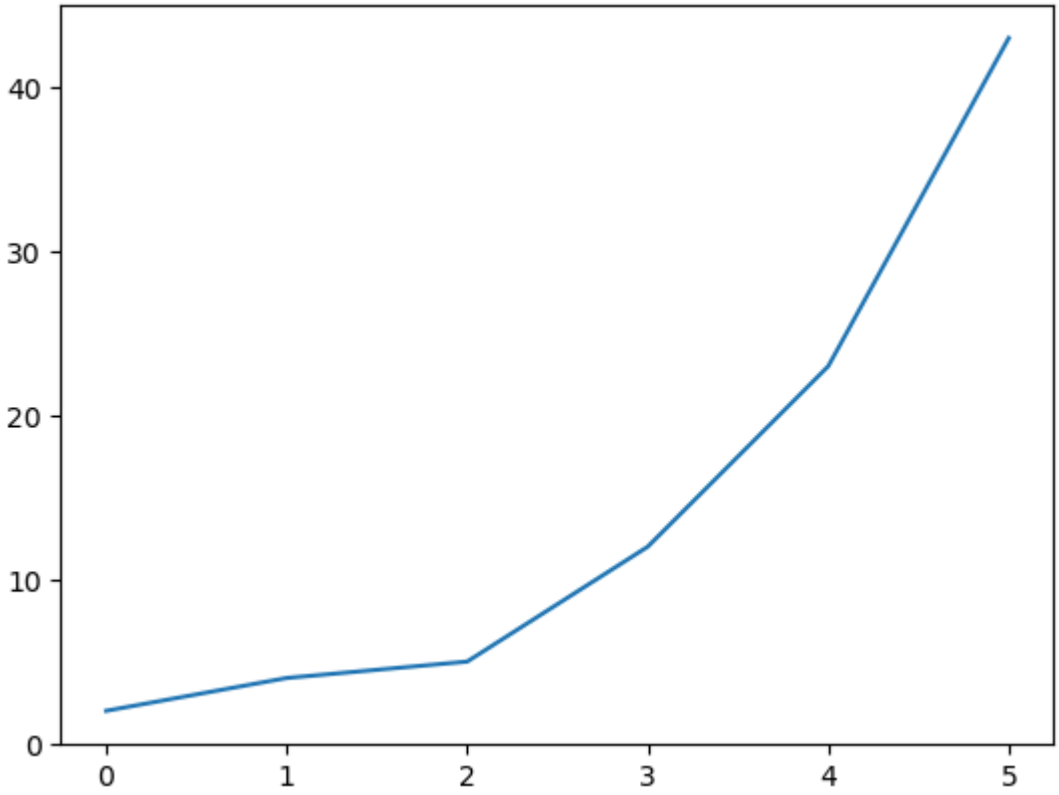
Out[22]:
```
[<matplotlib.lines.Line2D at 0x2083643c450>]
```



In [63]:
```
p=np.array([2,4,5,12,23,43])
```

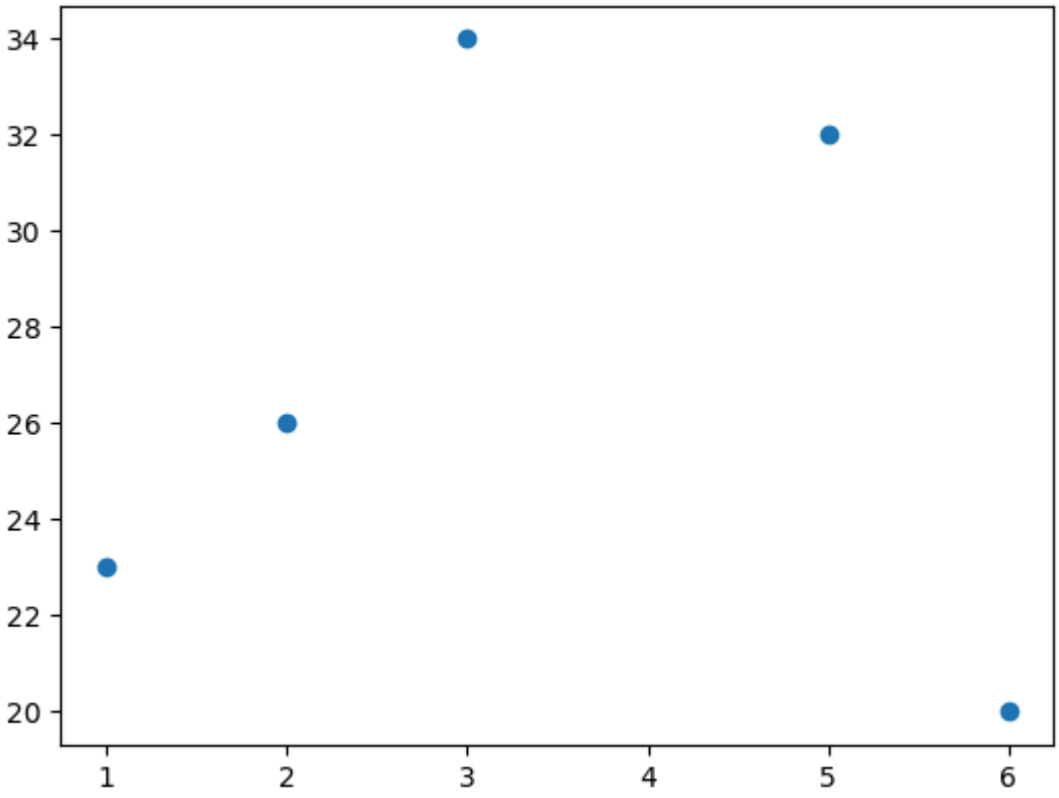
In [69]:
```
plt.plot(p)
```

Out[69]:
```
[<matplotlib.lines.Line2D at 0x20840c131d0>]
```



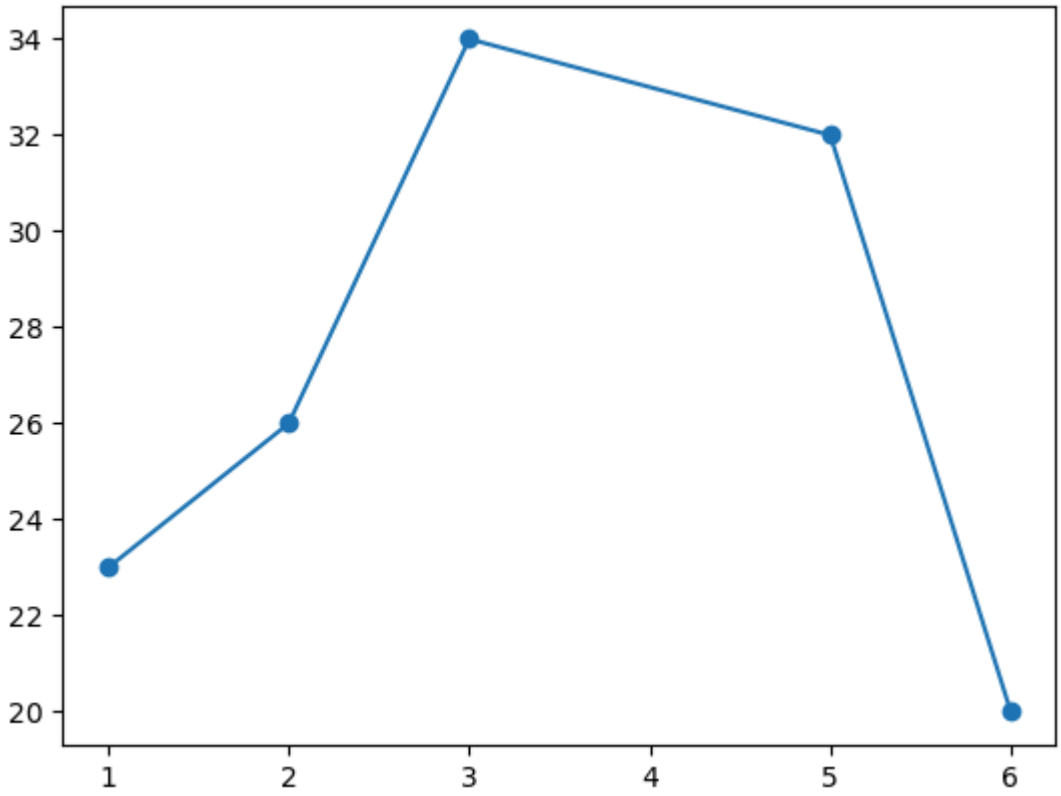
In [23]:
```
plt.plot(x1,y1,"o")
```

Out[23]:
```
[<matplotlib.lines.Line2D at 0x20836c51dd0>]
```



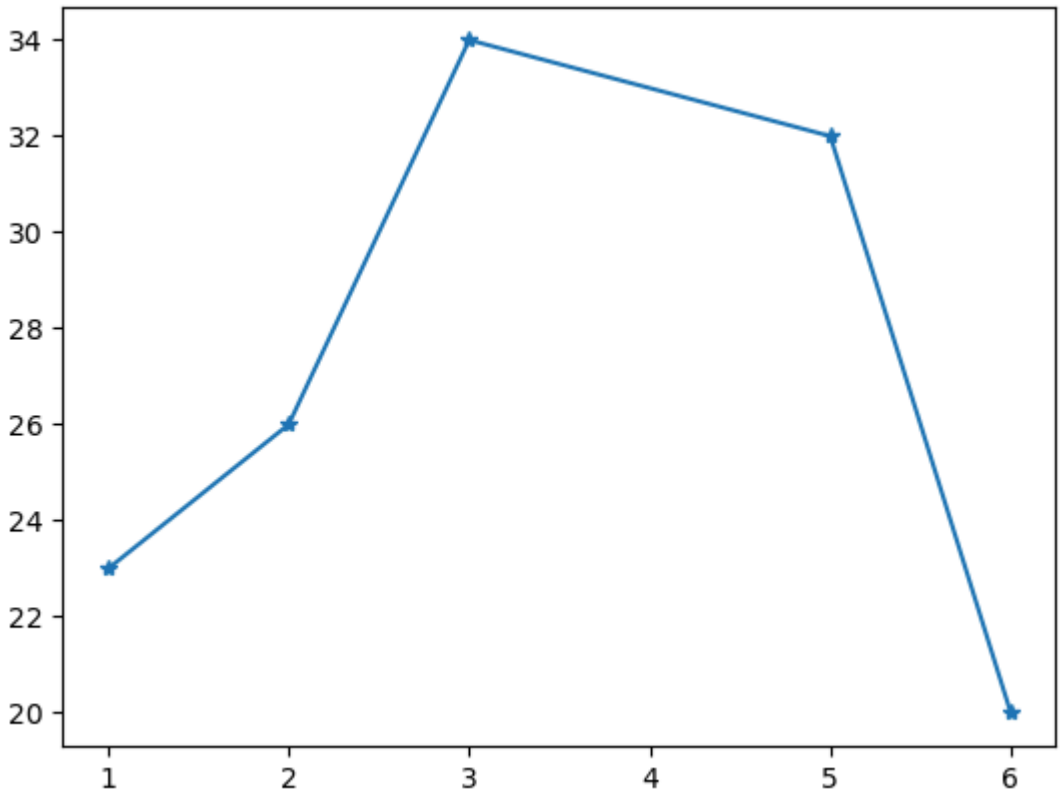
In [24]:
```
plt.plot(x1,y1,marker='o')
```

Out[24]: [<matplotlib.lines.Line2D at 0x20837df8450>]



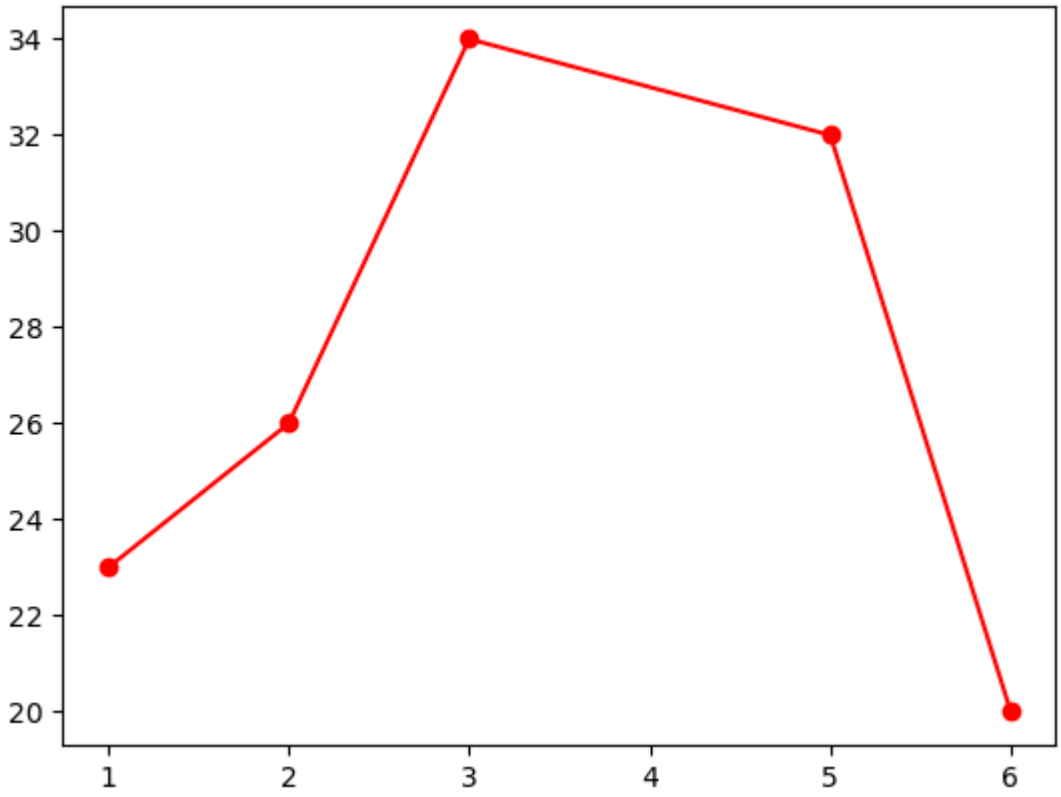
In [25]:
```
plt.plot(x1,y1,marker='*')
```

Out[25]: [<matplotlib.lines.Line2D at 0x20837e68450>]



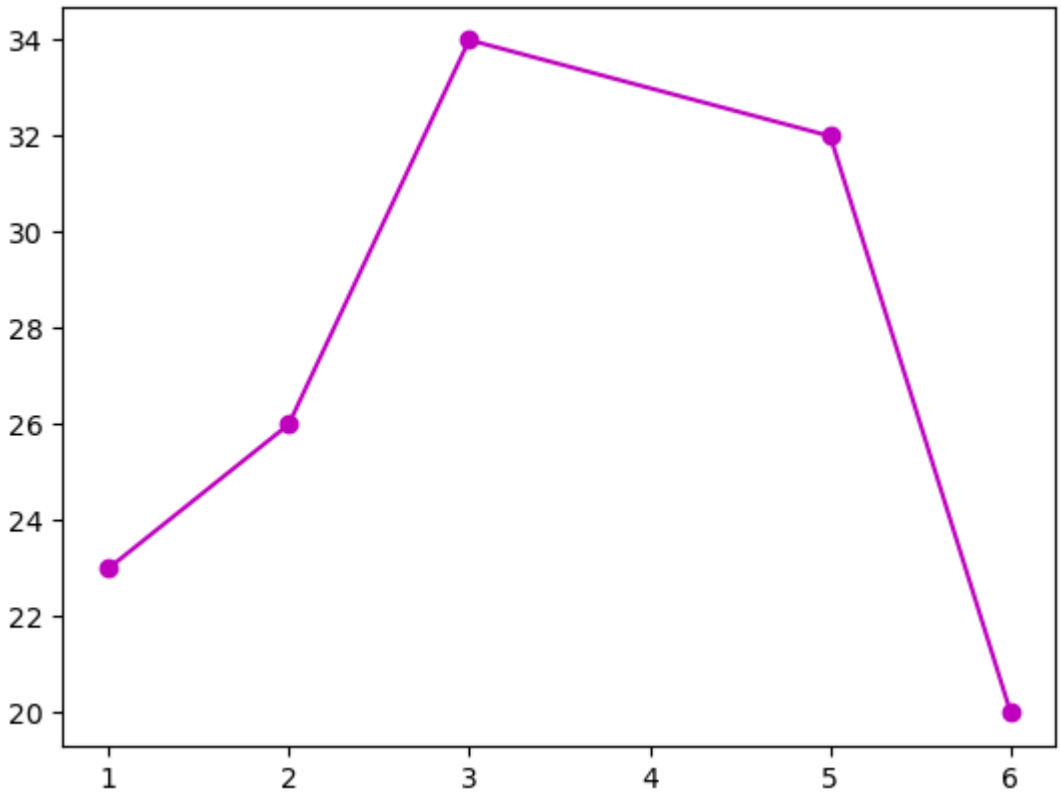
In [26]:
```
plt.plot(x1,y1,marker='o',color='red')
```

Out[26]: [<matplotlib.lines.Line2D at 0x20837eef610>]



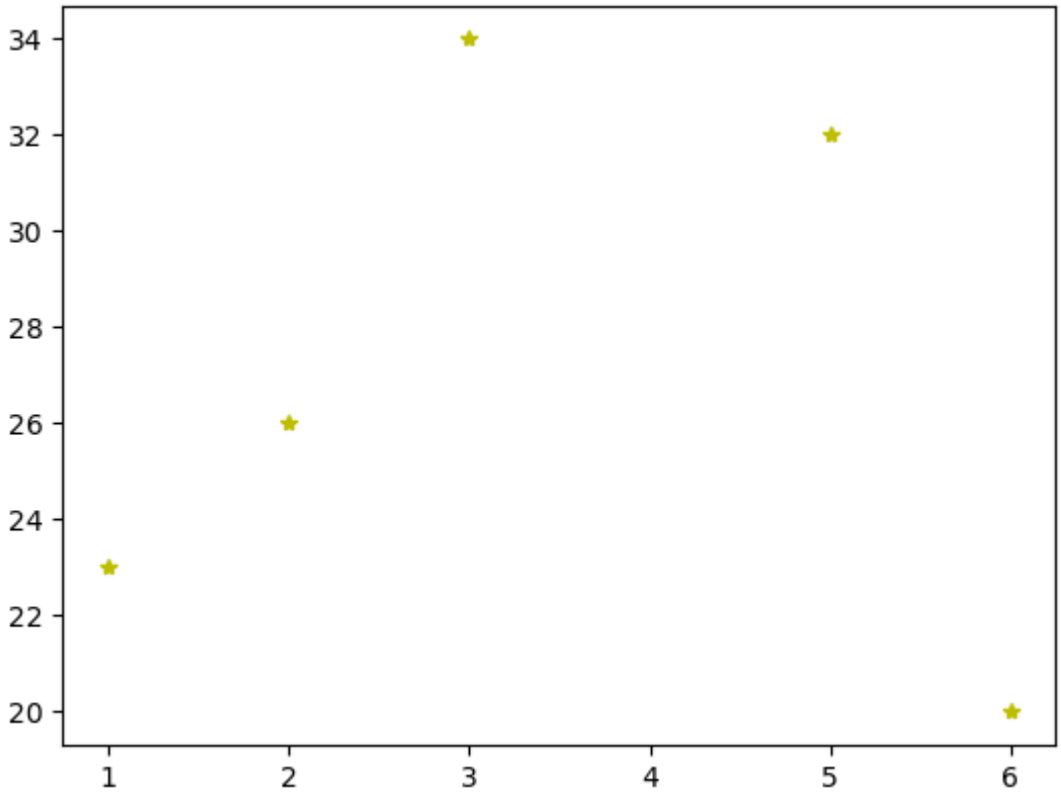
In [27]:
```
plt.plot(x1,y1,marker='o',color='m')
```

Out[27]: [<matplotlib.lines.Line2D at 0x20837f70110>]



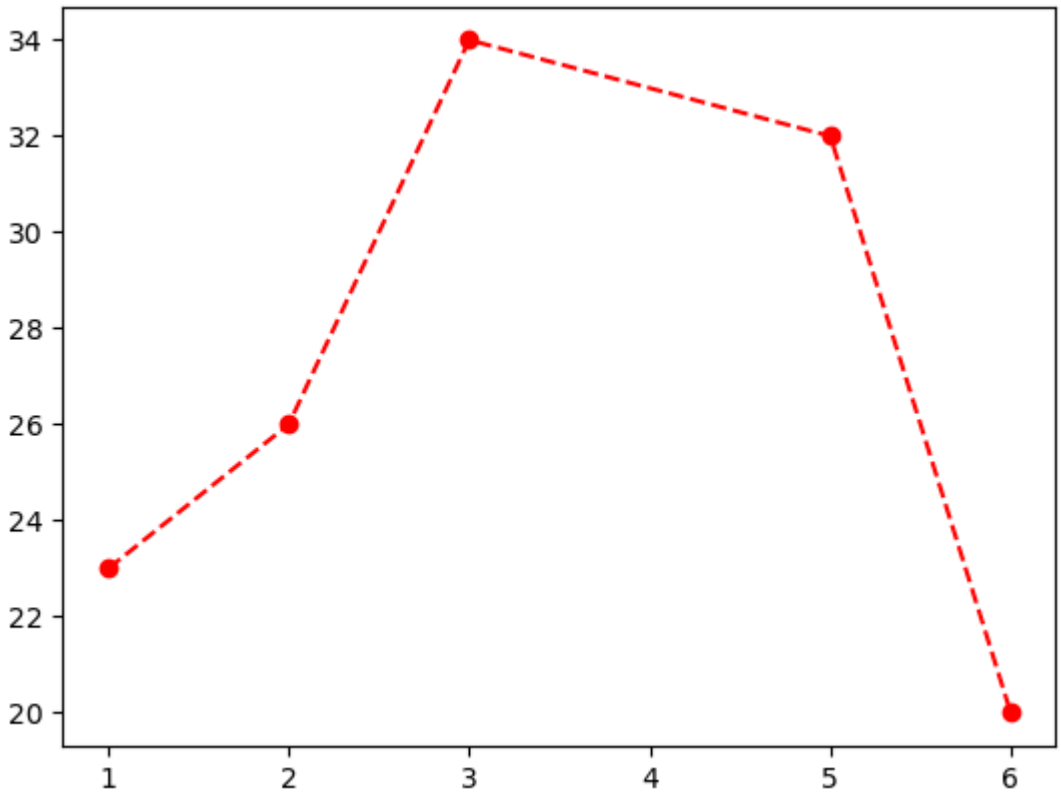
In [28]:
```
plt.plot(x1,y1,'*''y')
```

Out[28]: [<matplotlib.lines.Line2D at 0x20837e22810>]



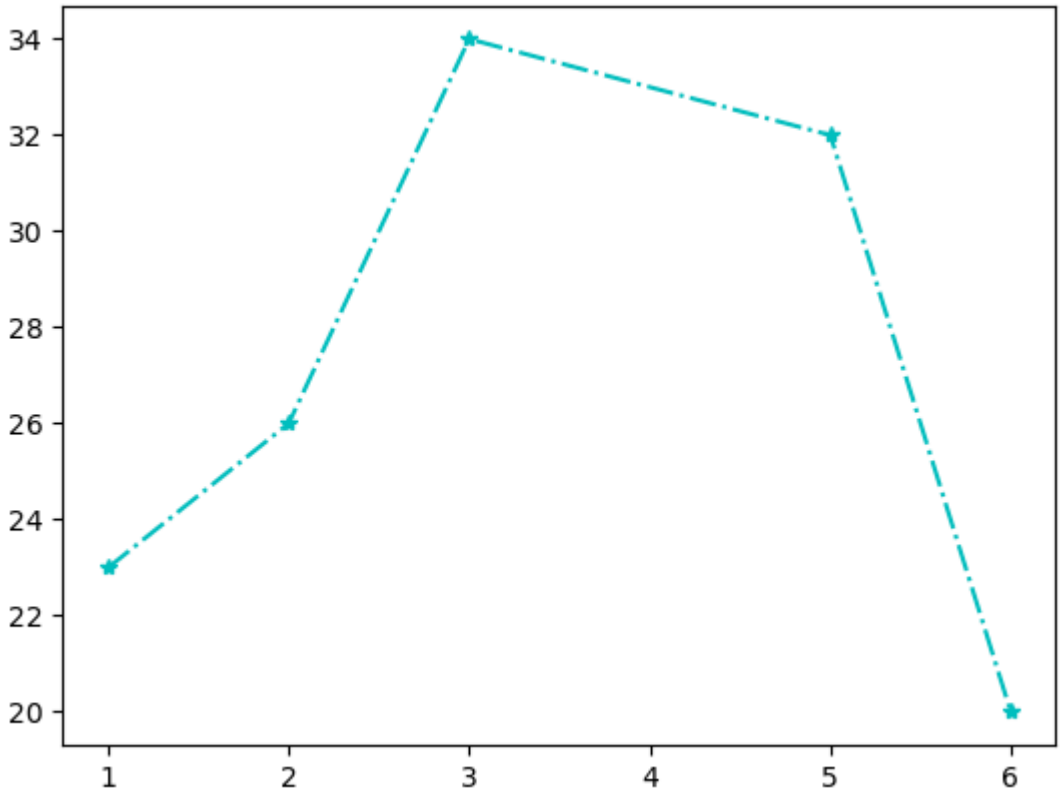
In [29]:
```
plt.plot(x1,y1,marker='o',color='red',linestyle='--')
```

Out[29]: [<matplotlib.lines.Line2D at 0x2083802a810>]



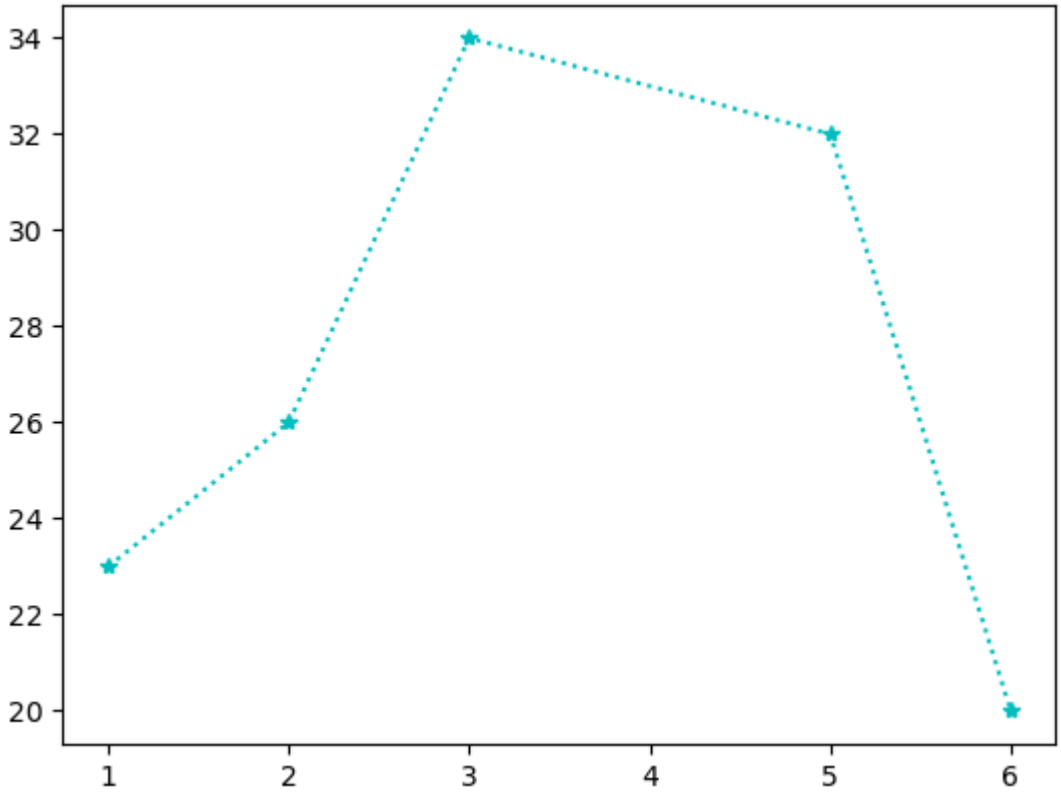
In [30]:
```
plt.plot(x1,y1,marker="*",color="c",linestyle='-.')
```

Out[30]: [<matplotlib.lines.Line2D at 0x208380c62d0>]



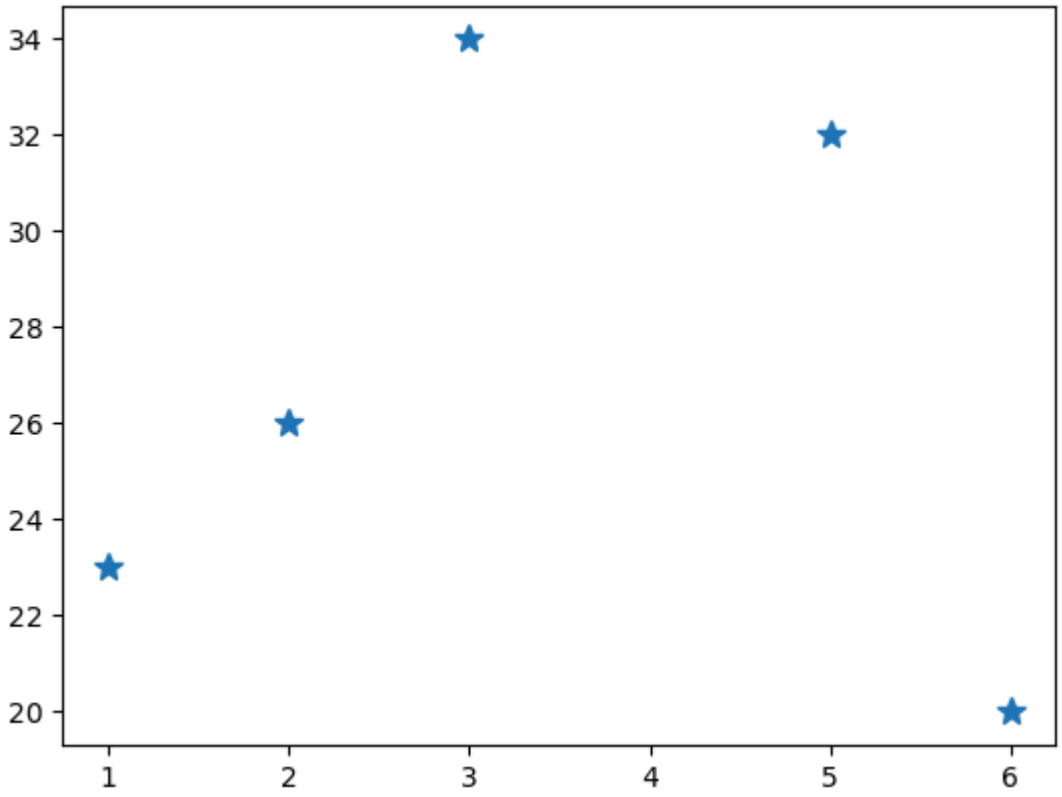
In [31]:
```
plt.plot(x1,y1,'*:c')
```

Out[31]: [<matplotlib.lines.Line2D at 0x208380e2750>]



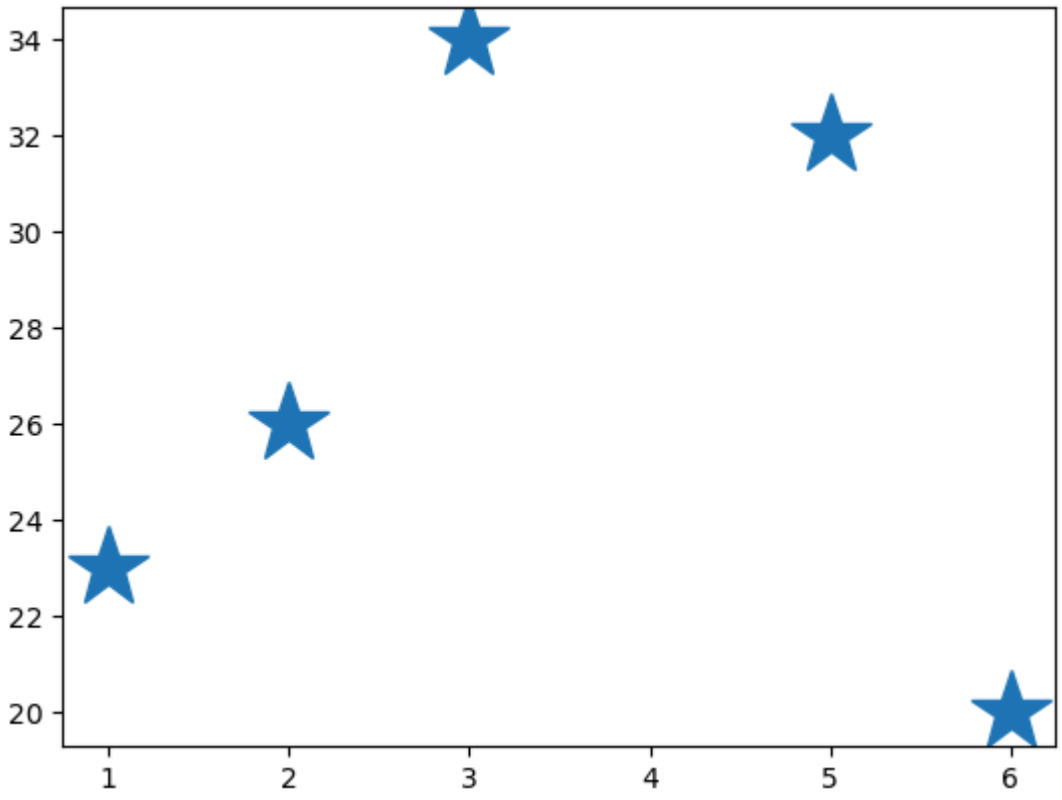
In [33]:
```
plt.plot(x1,y1,'*',ms=10) #marker size(ms)
```

Out[33]: [<matplotlib.lines.Line2D at 0x20836ca5710>]



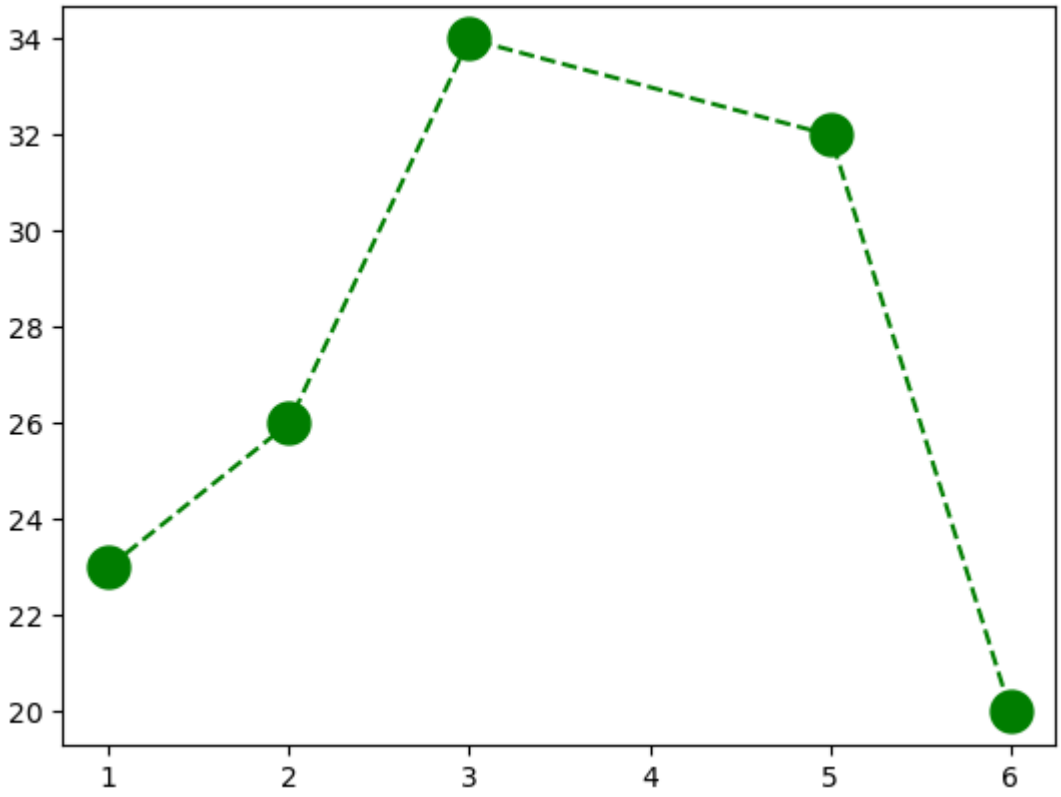
In [34]:
```
plt.plot(x1,y1,'*',ms=30)
```

Out[34]: [<matplotlib.lines.Line2D at 0x20836cc88d0>]



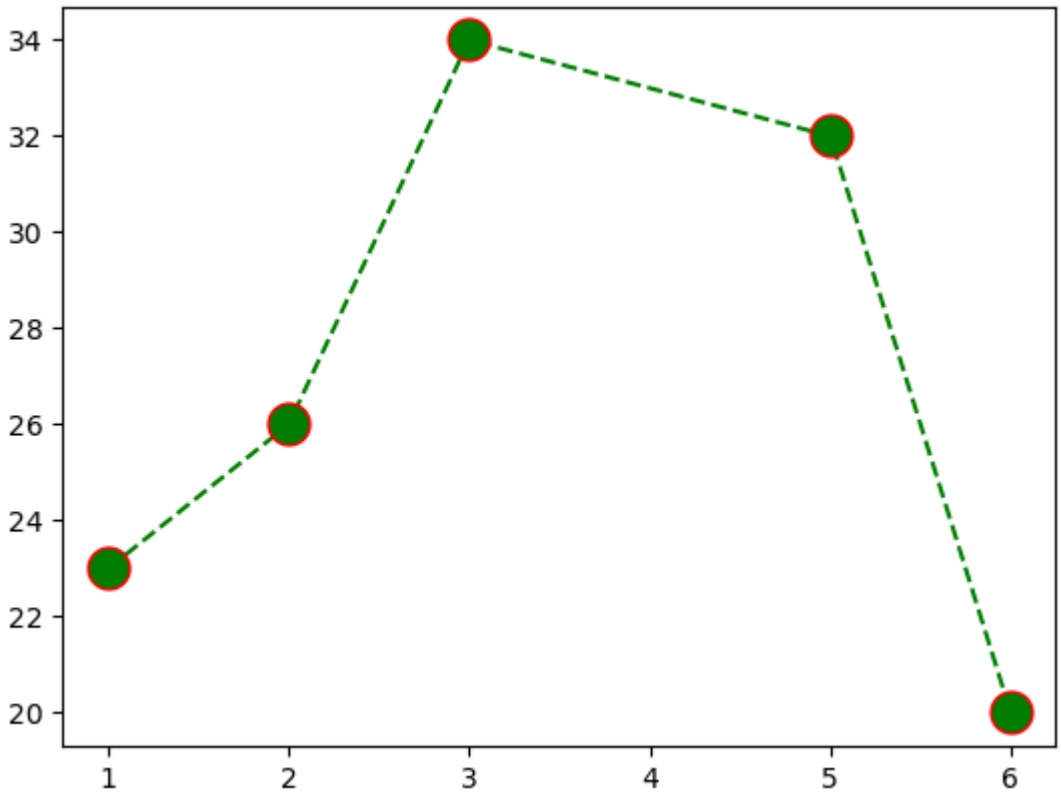
In [35]:
```
plt.plot(x1,y1,'o--g',ms=15)
```

Out[35]: [<matplotlib.lines.Line2D at 0x2083818be10>]



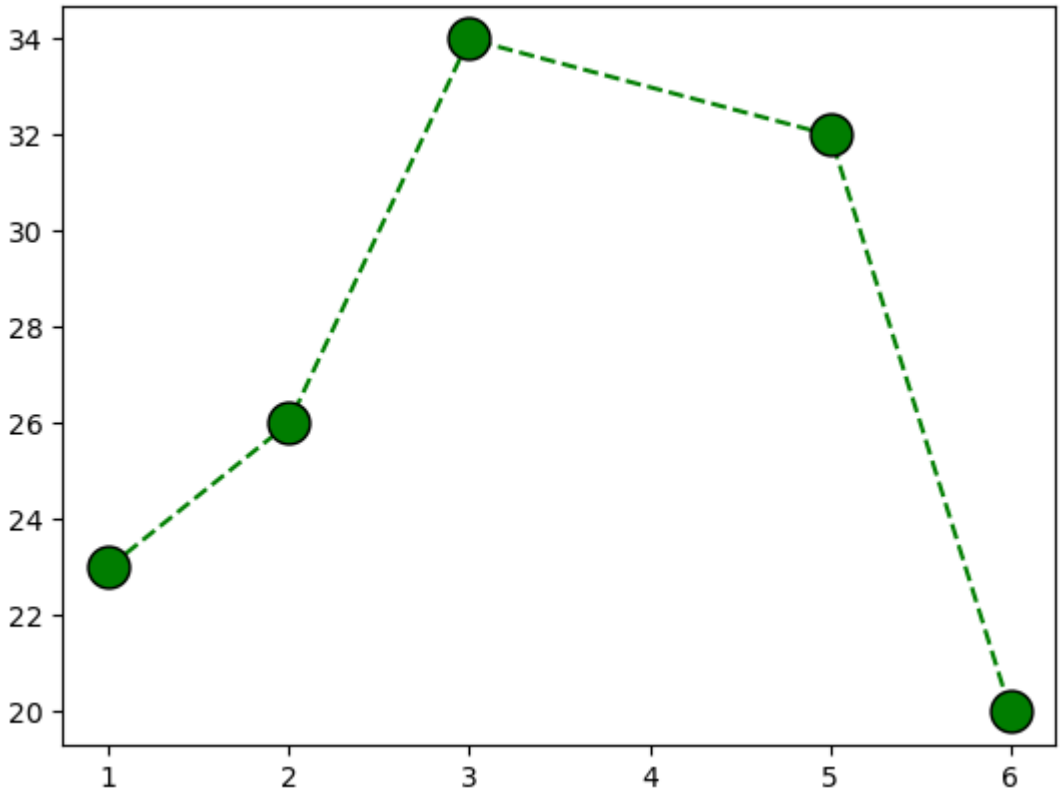
In [36]:
```
plt.plot(x1,y1,'o--g',ms=15,mec='r')# marker edge color-mec
```

Out[36]: [<matplotlib.lines.Line2D at 0x20836d2d5d0>]



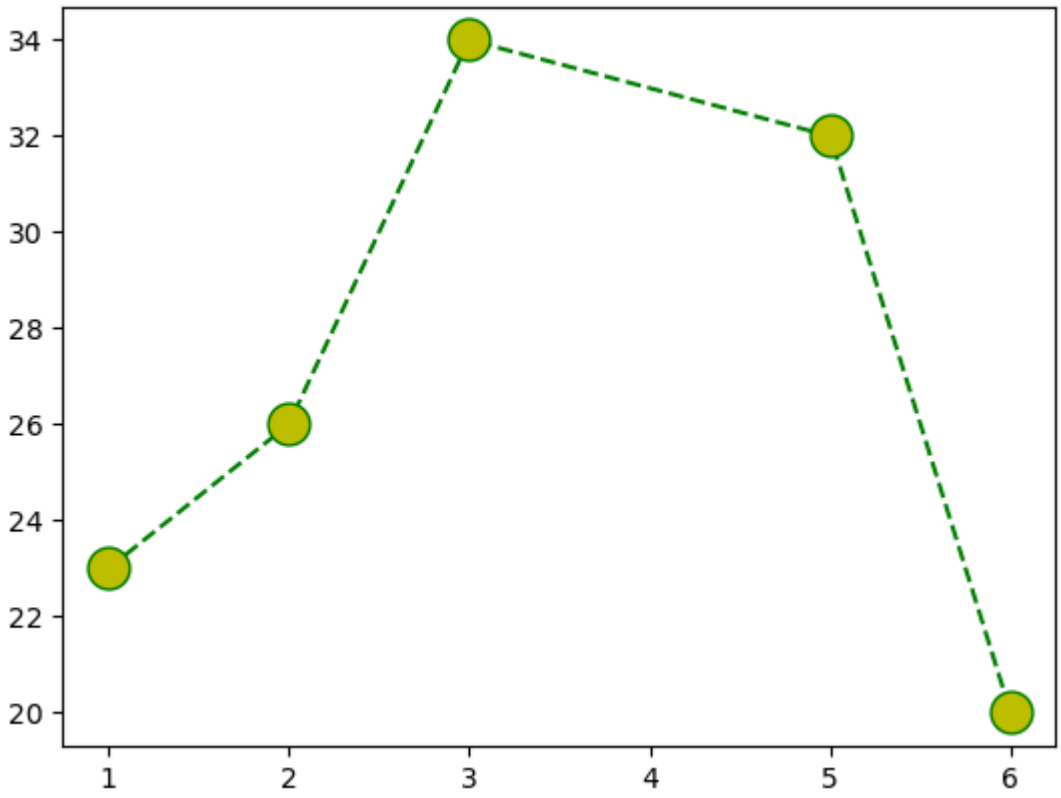
In [37]:
```
plt.plot(x1,y1,'o--g',ms=15,mec='k')
```

Out[37]: [<matplotlib.lines.Line2D at 0x2083827f1d0>]



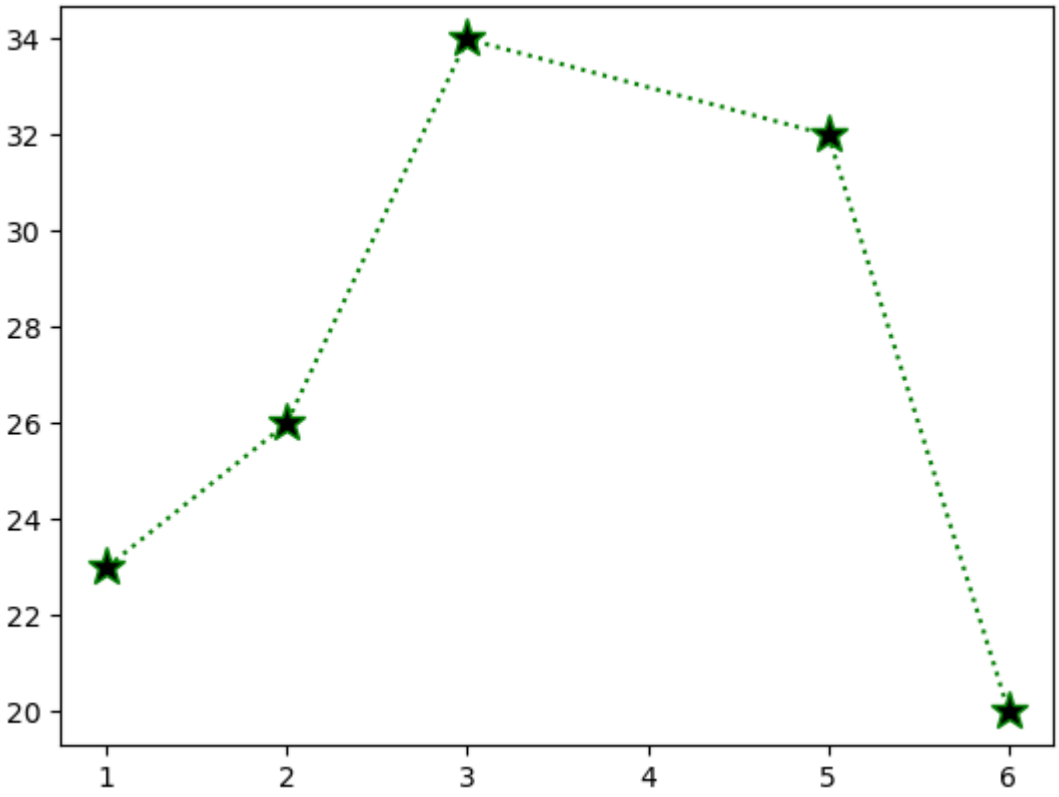
In [38]:
```
plt.plot(x1,y1,'o--g',ms=15,mfc='y') #marker face color (mfc)
```

Out[38]:
```
[<matplotlib.lines.Line2D at 0x208394430d0>]
```



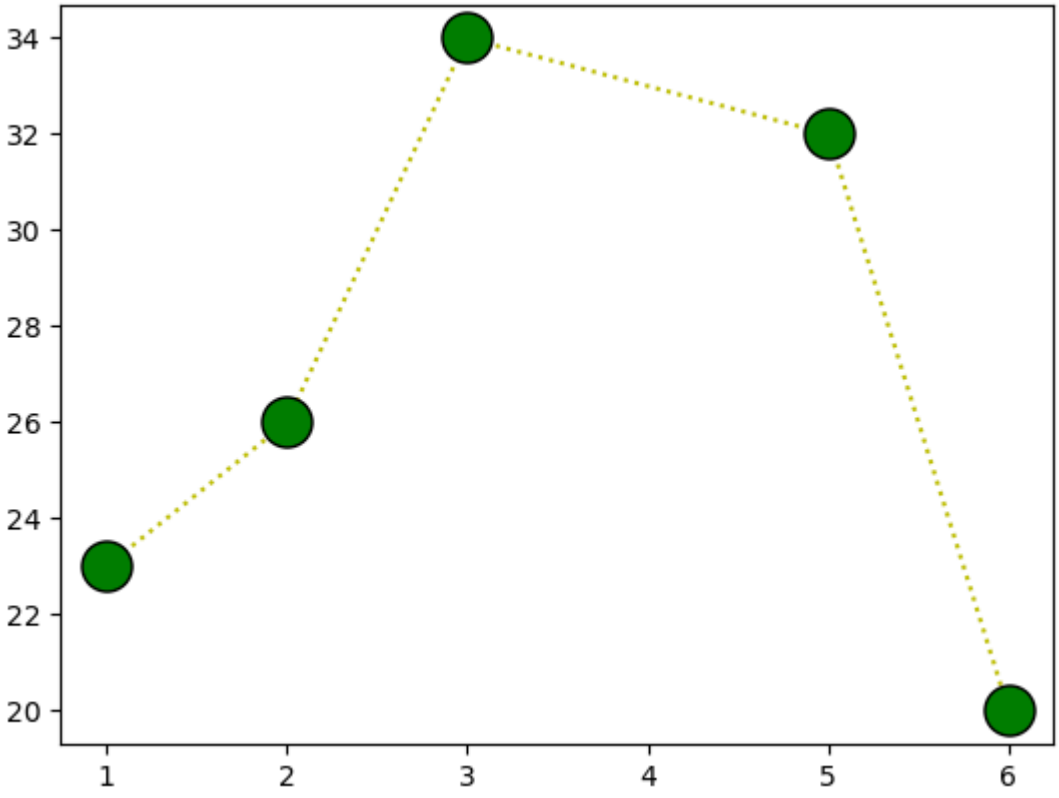
In [39]:
```
plt.plot(x1,y1,'*:g',ms=14,mfc='k')
```

Out[39]:
```
[<matplotlib.lines.Line2D at 0x20839483cd0>]
```



In [40]:
```
plt.plot(x1,y1,'o:y',ms=18,mec='k',mfc='g')
```

Out[40]:
```
[<matplotlib.lines.Line2D at 0x20839530150>]
```

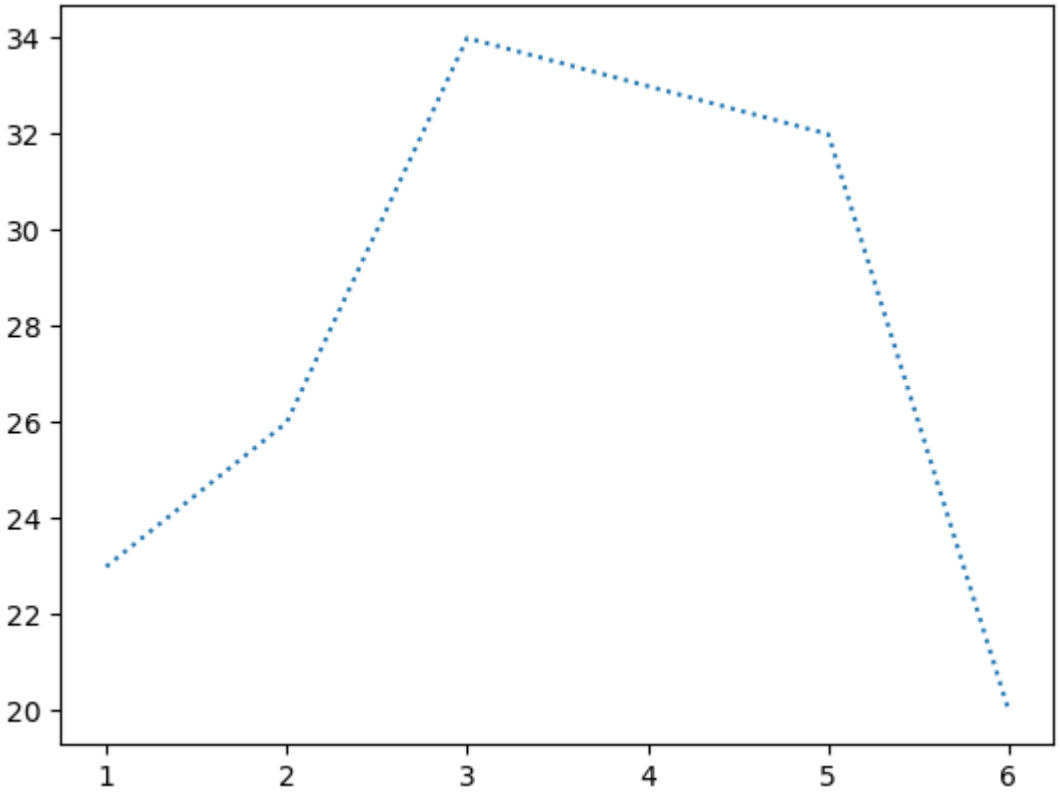


linestyle or ls

- is used to change the style of the plotted line

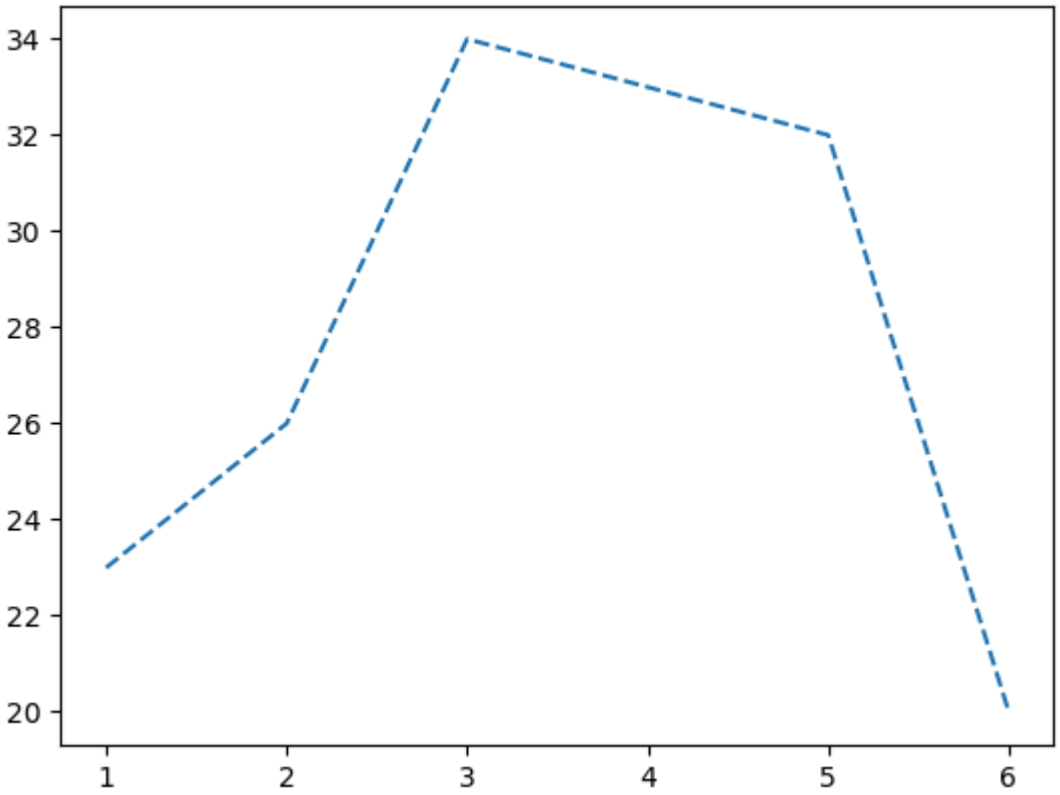
In [41]:
```
plt.plot(x1,y1,ls=':') #linestyle(ls)
```

Out[41]:
```
[<matplotlib.lines.Line2D at 0x208395a4110>]
```



In [42]:
```
plt.plot(x1,y1,ls='--')
```

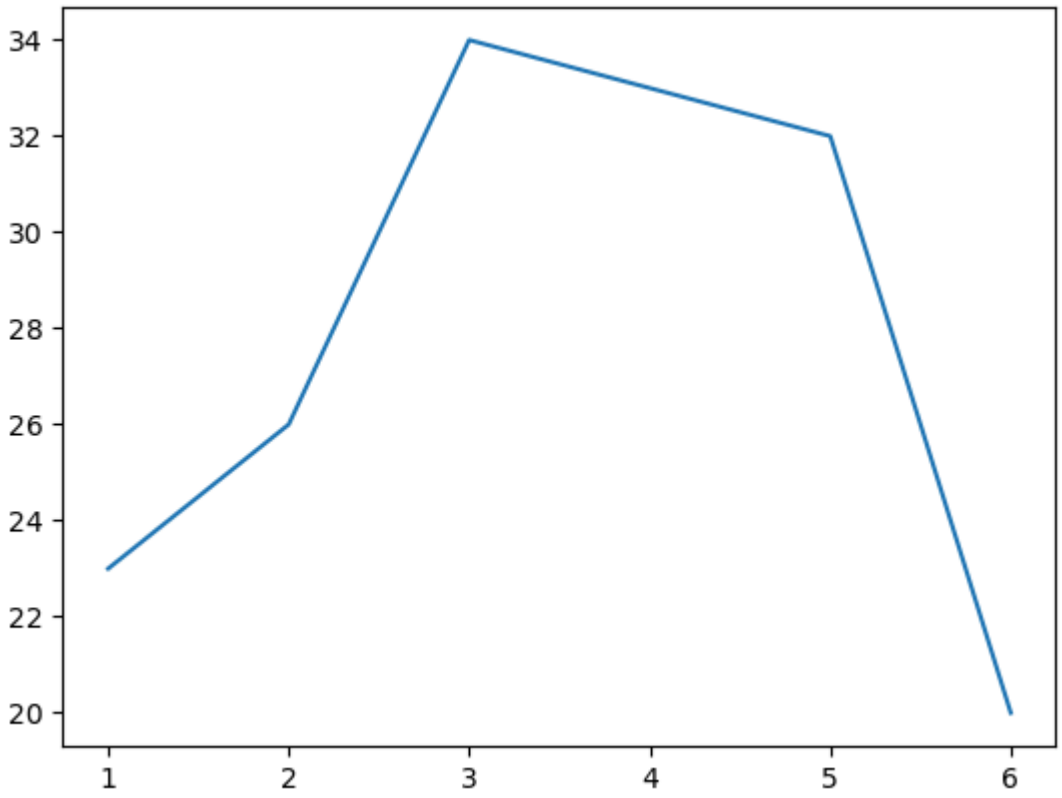
Out[42]:
```
[<matplotlib.lines.Line2D at 0x208392d8650>]
```



line width

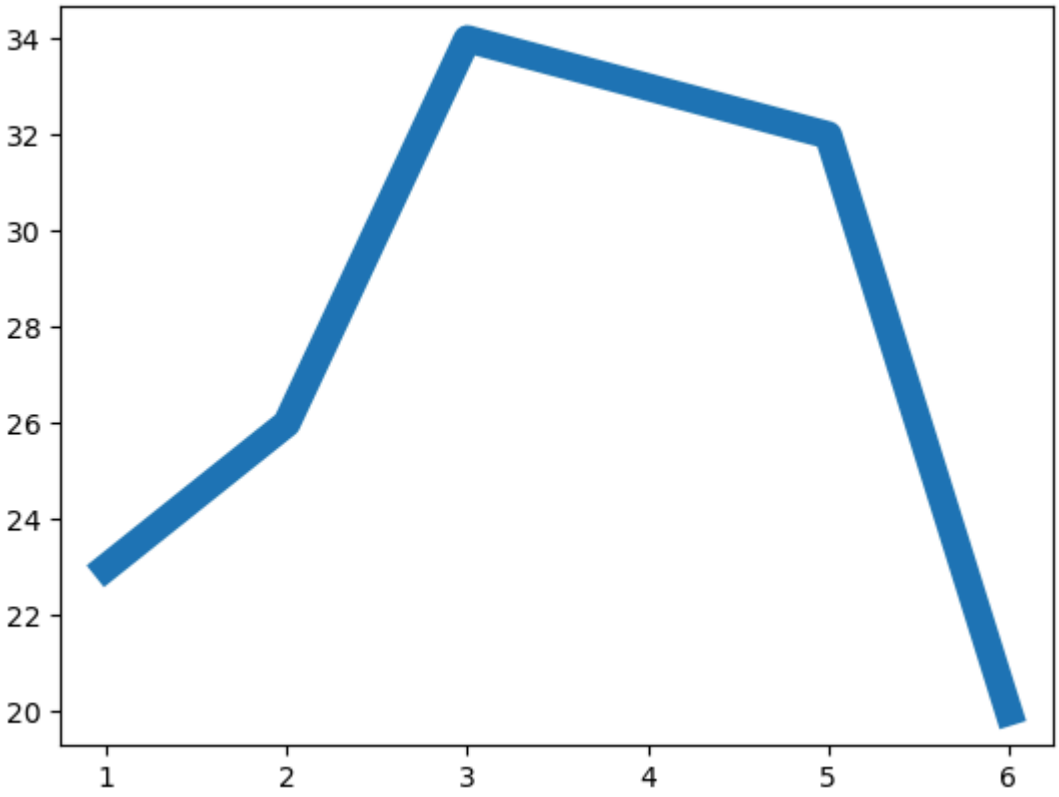
In [43]:
```
plt.plot(x1,y1)
```

Out[43]:
```
[<matplotlib.lines.Line2D at 0x20839350fd0>]
```



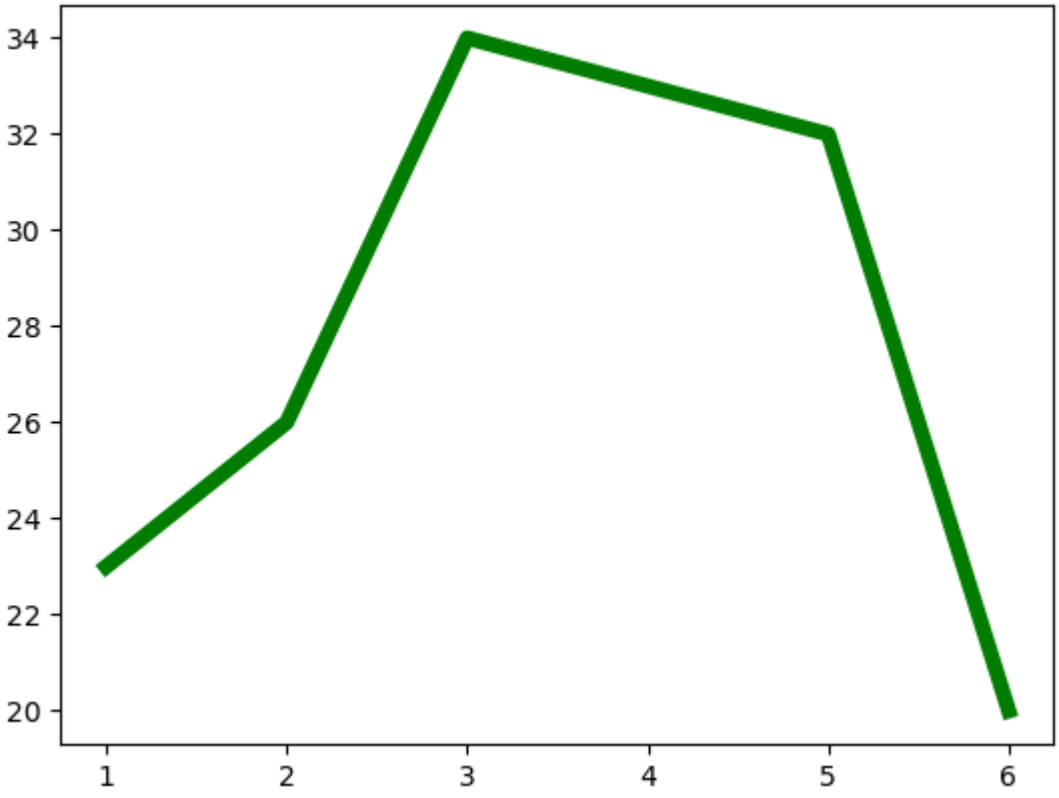
In [44]:
```
plt.plot(x1,y1,linewidth='10')
```

Out[44]:
```
[<matplotlib.lines.Line2D at 0x20839605350>]
```



In [45]:
```
plt.plot(x1,y1,'g',linewidth="5.5")
```

Out[45]:
```
[<matplotlib.lines.Line2D at 0x20839648d90>]
```



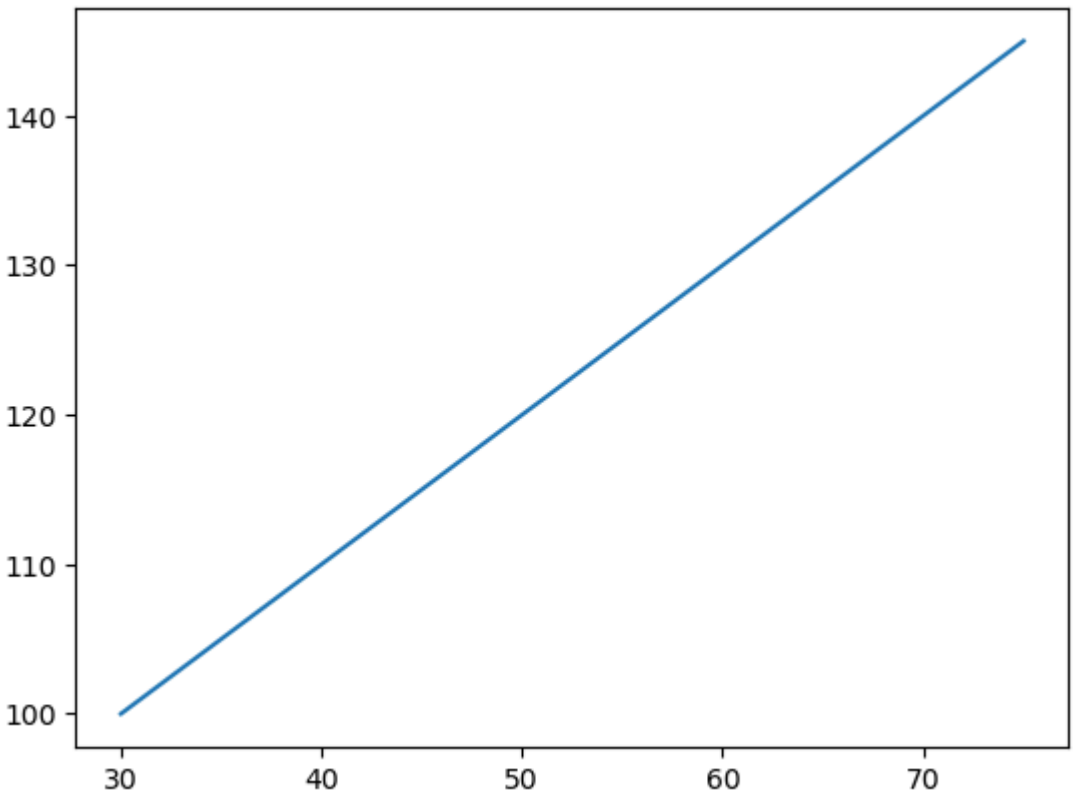
Create labels for a plot

In [46]:
```
x2=np.array([30,35,40,45,50,55,60,65,70,75])
```

```
y2=np.array([100,105,110,115,120,125,130,135,140,145])
```

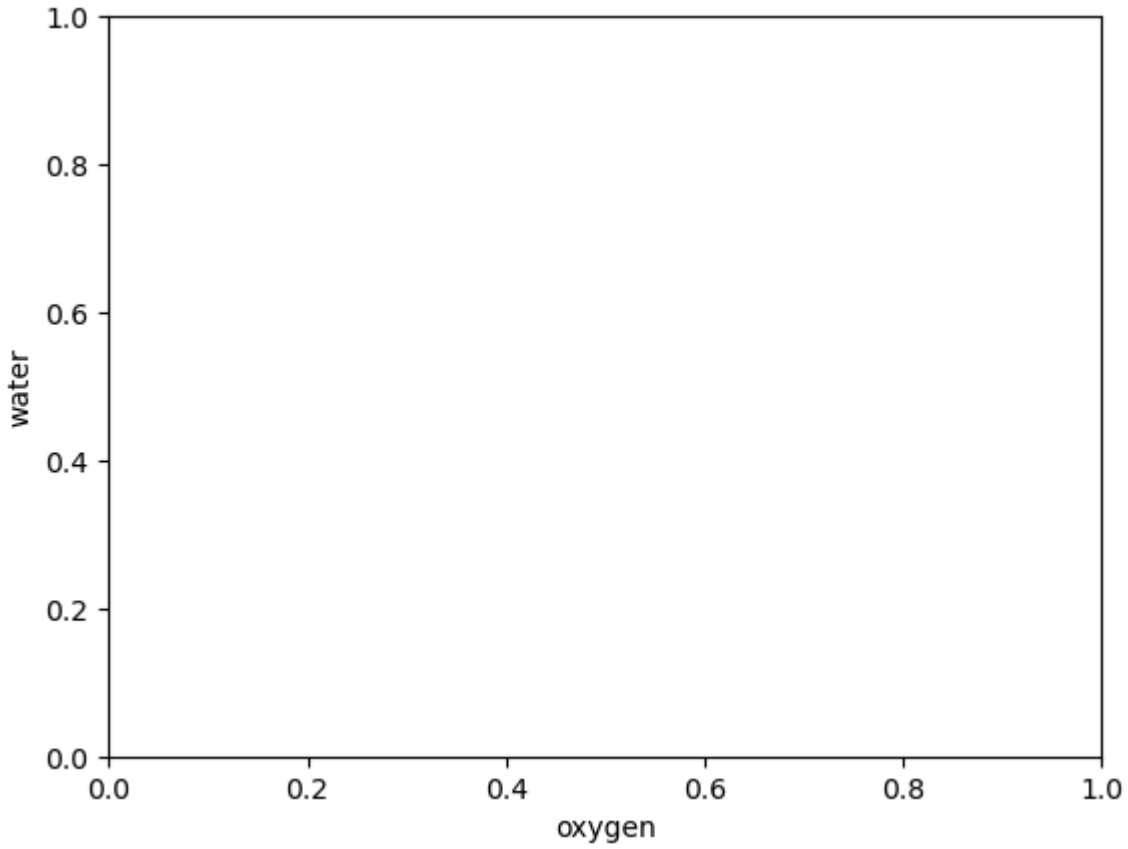
In [47]:
```
plt.plot(x2,y2)
```

Out[47]:
```
[<matplotlib.lines.Line2D at 0x208396e5f90>]
```



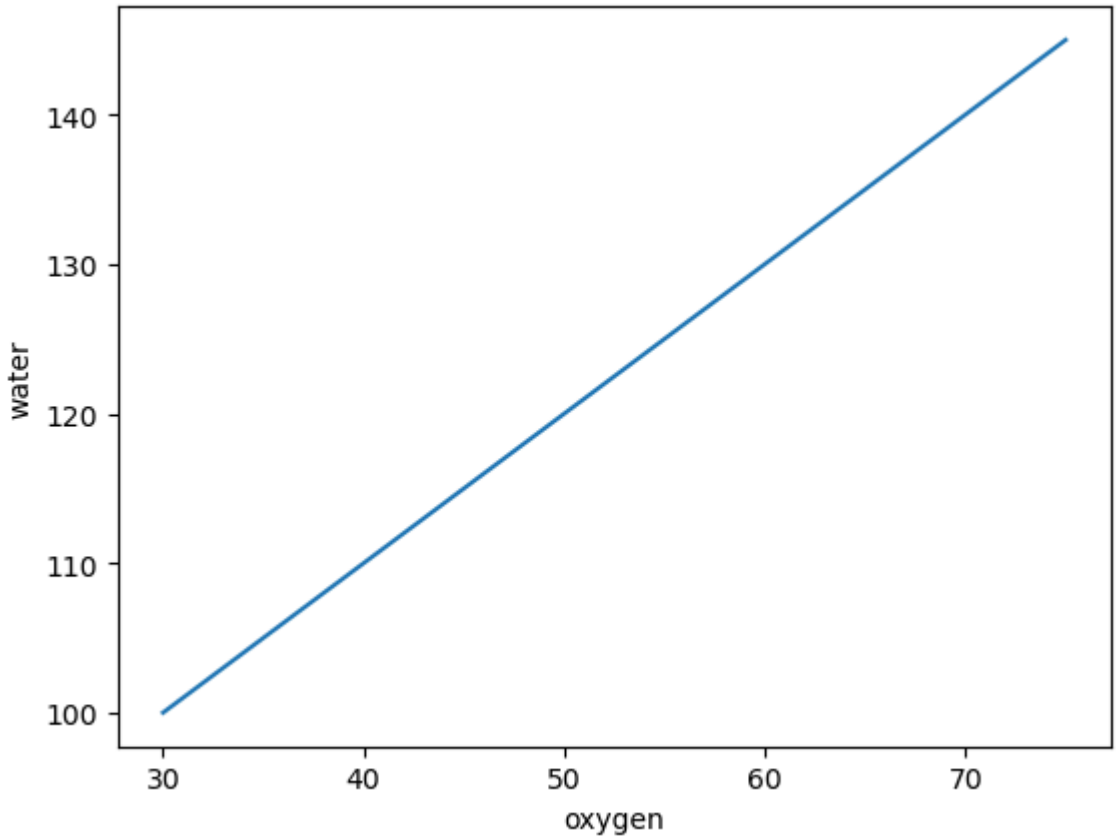
In [48]:
```
plt.xlabel('oxygen')
plt.ylabel('water')
```

Out[48]:
```
Text(0, 0.5, 'water')
```



In [49]:
```
plt.plot(x2,y2)
plt.xlabel('oxygen')
plt.ylabel('water')
```

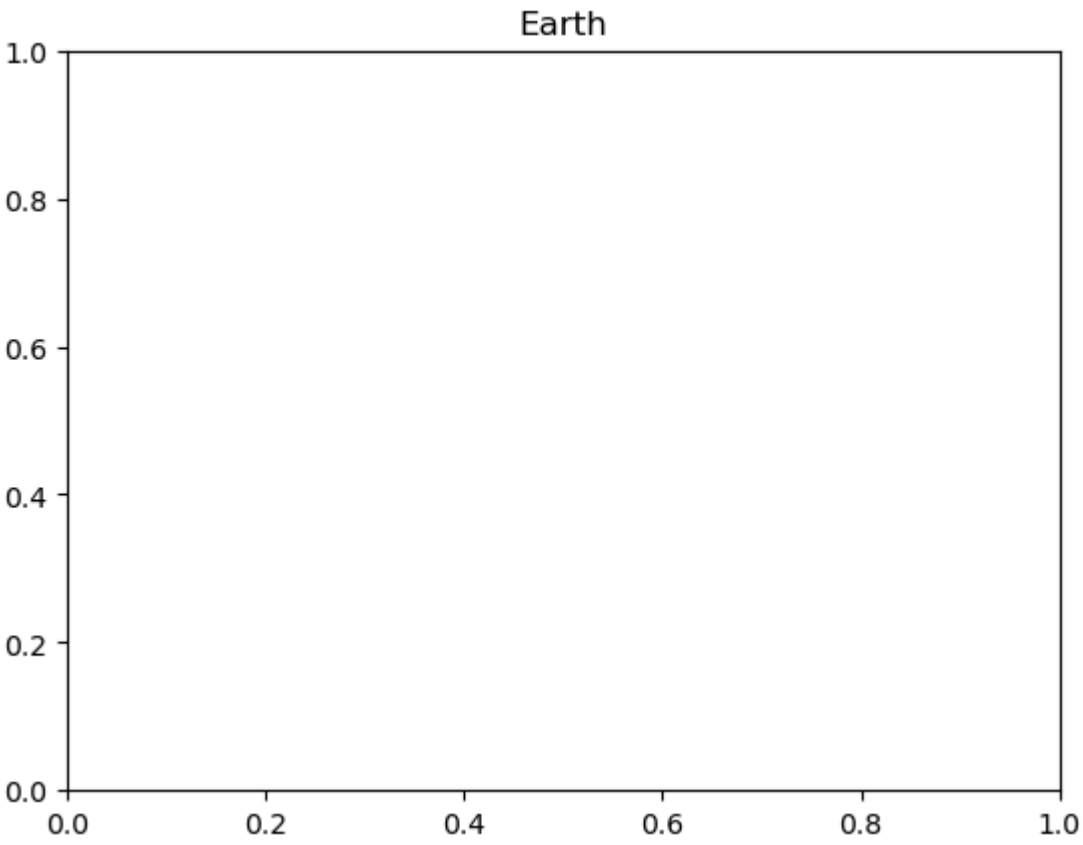
Out[49]: Text(0, 0.5, 'water')



Title for the plot

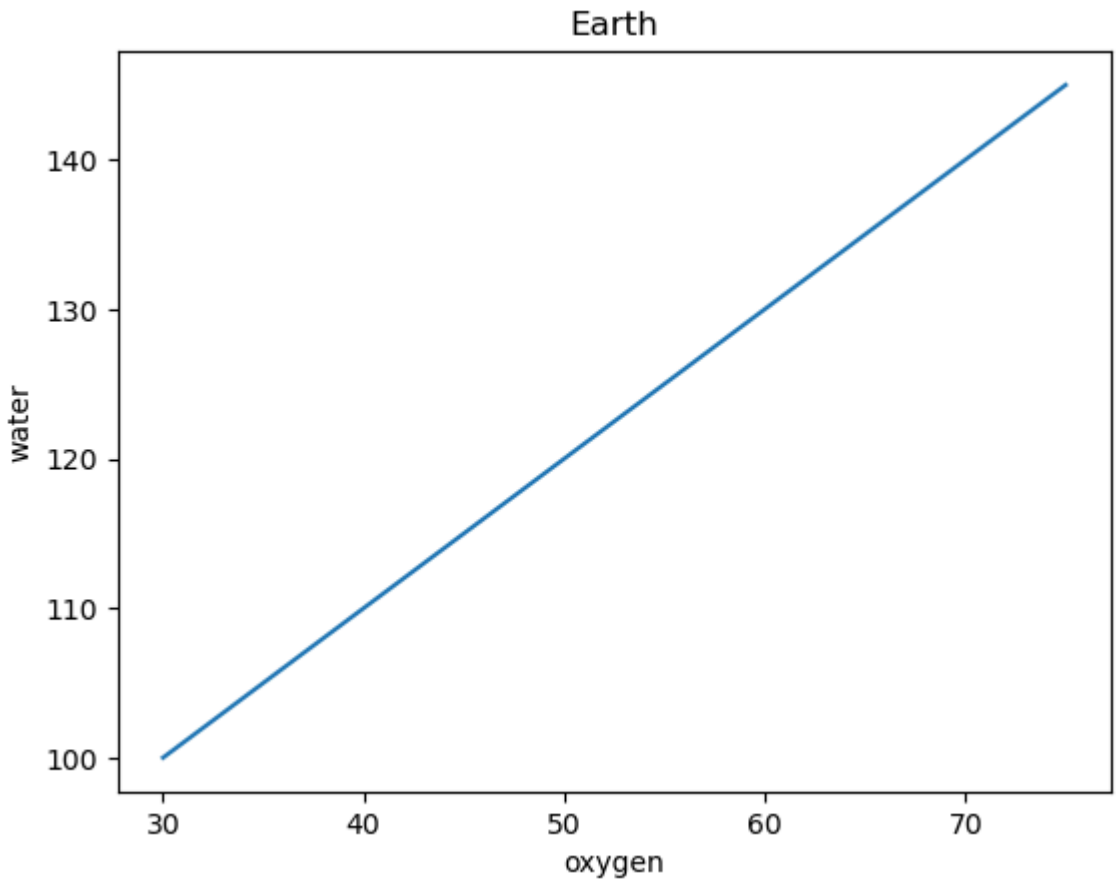
In [50]:
```
plt.title('Earth')
```

Out[50]:
```
Text(0.5, 1.0, 'Earth')
```



In [51]:
```
plt.plot(x2,y2)
plt.title('Earth')
plt.xlabel('oxygen')
plt.ylabel('water')
```

Out[51]:
```
Text(0, 0.5, 'water')
```

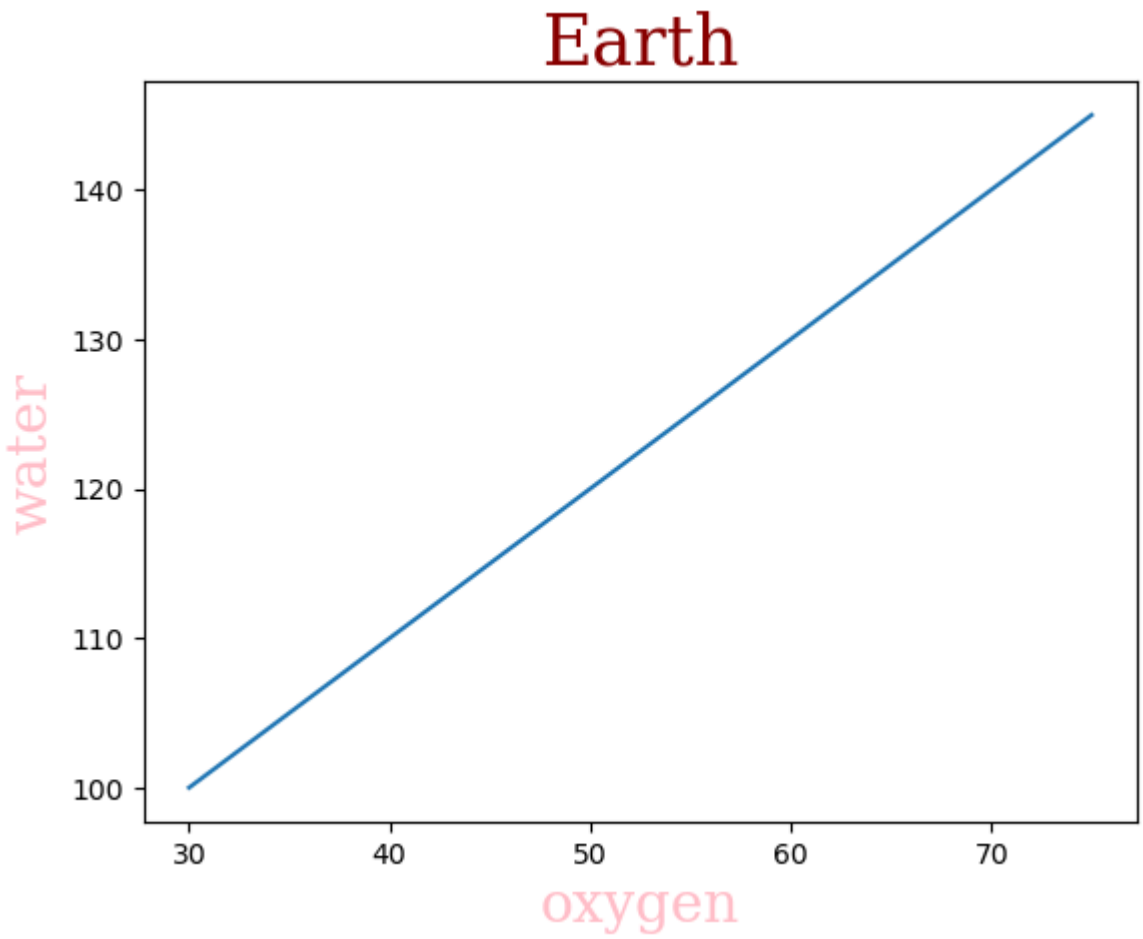


Font property for title and labels

In [52]:
```
f1={'family':'serif','color':'pink','size':20}
f2={'family':'serif','color':'darkred','size':25}
```

In [53]:
```
plt.plot(x2,y2)
plt.title('Earth',f2)
plt.xlabel('oxygen',f1)
plt.ylabel('water',f1)
```

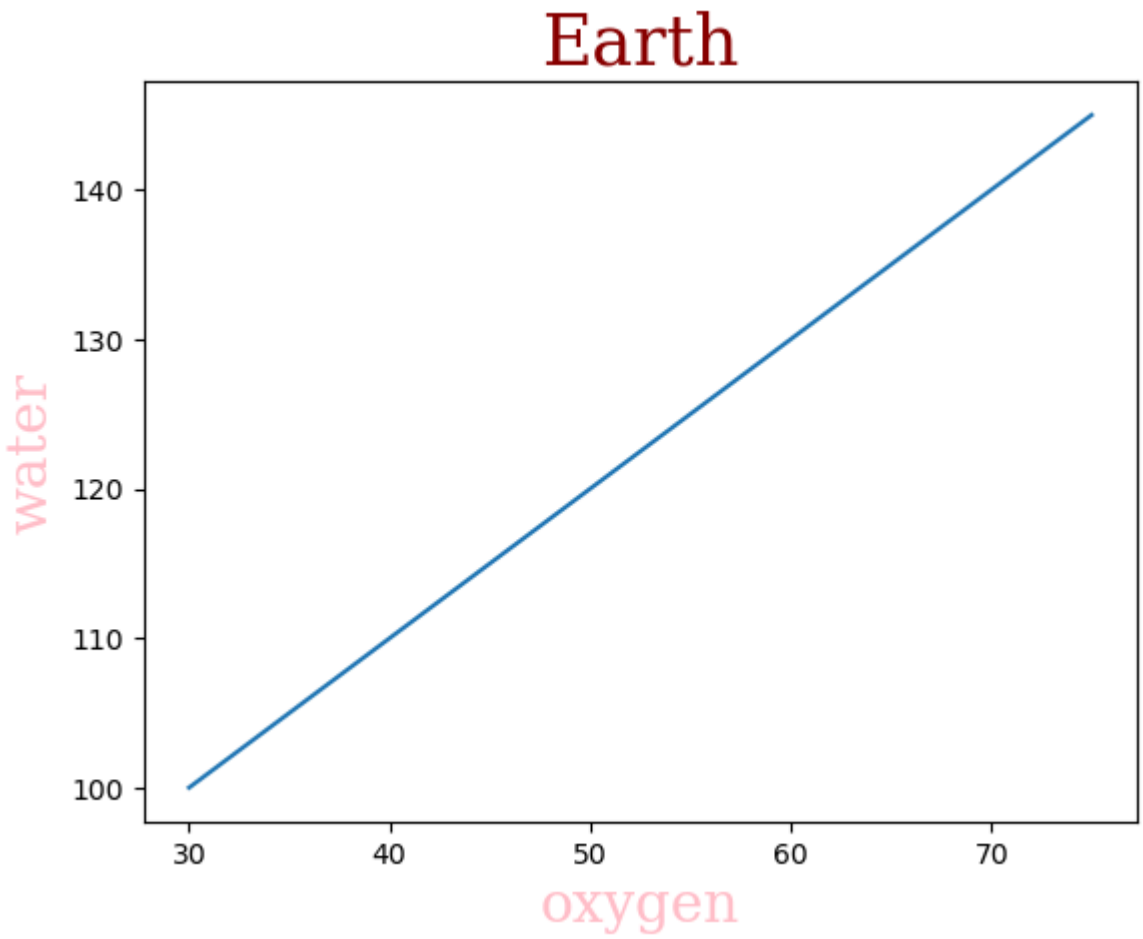
Out[53]:
```
Text(0, 0.5, 'water')
```



Change the location of title

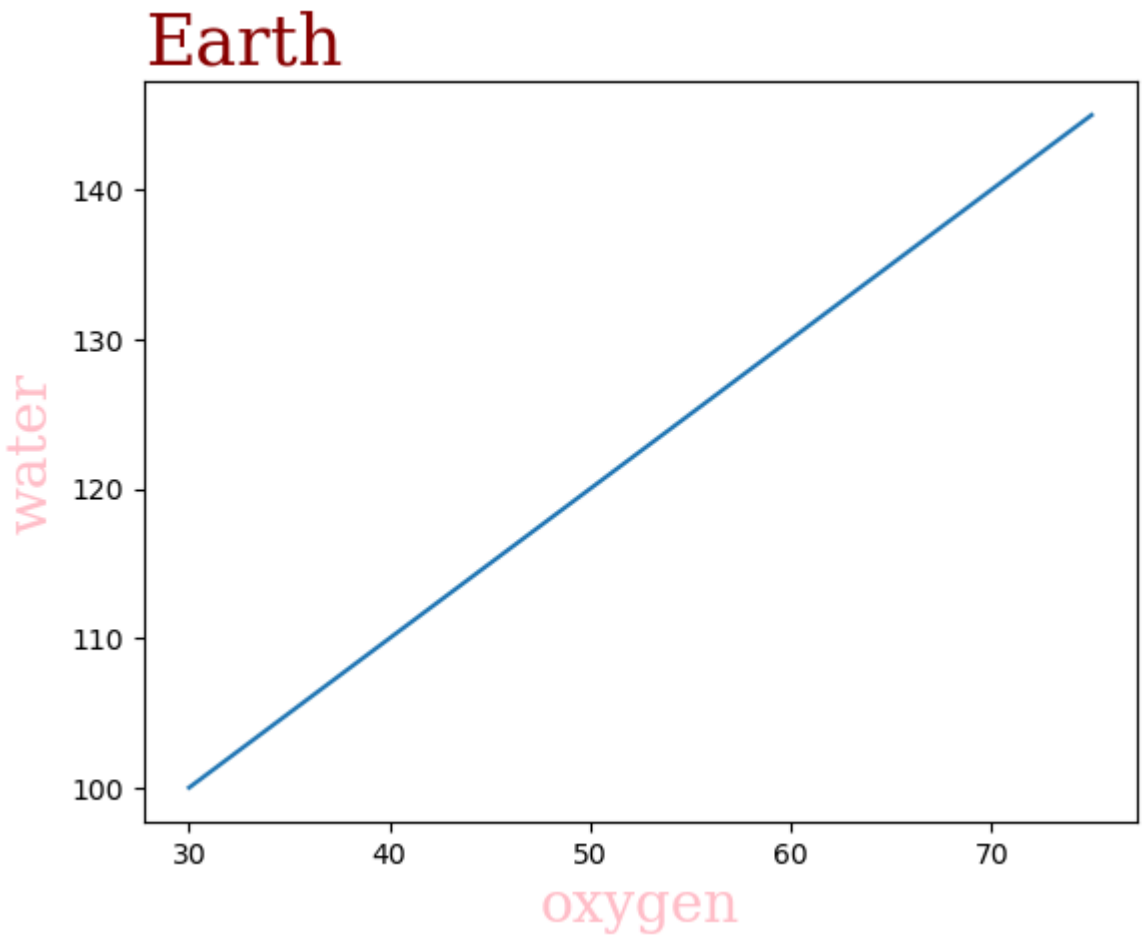
In [54]:
```
plt.plot(x2,y2)
plt.title('Earth',f2)
plt.xlabel('oxygen',f1)
plt.ylabel('water',f1)
```

Out[54]:
```
Text(0, 0.5, 'water')
```



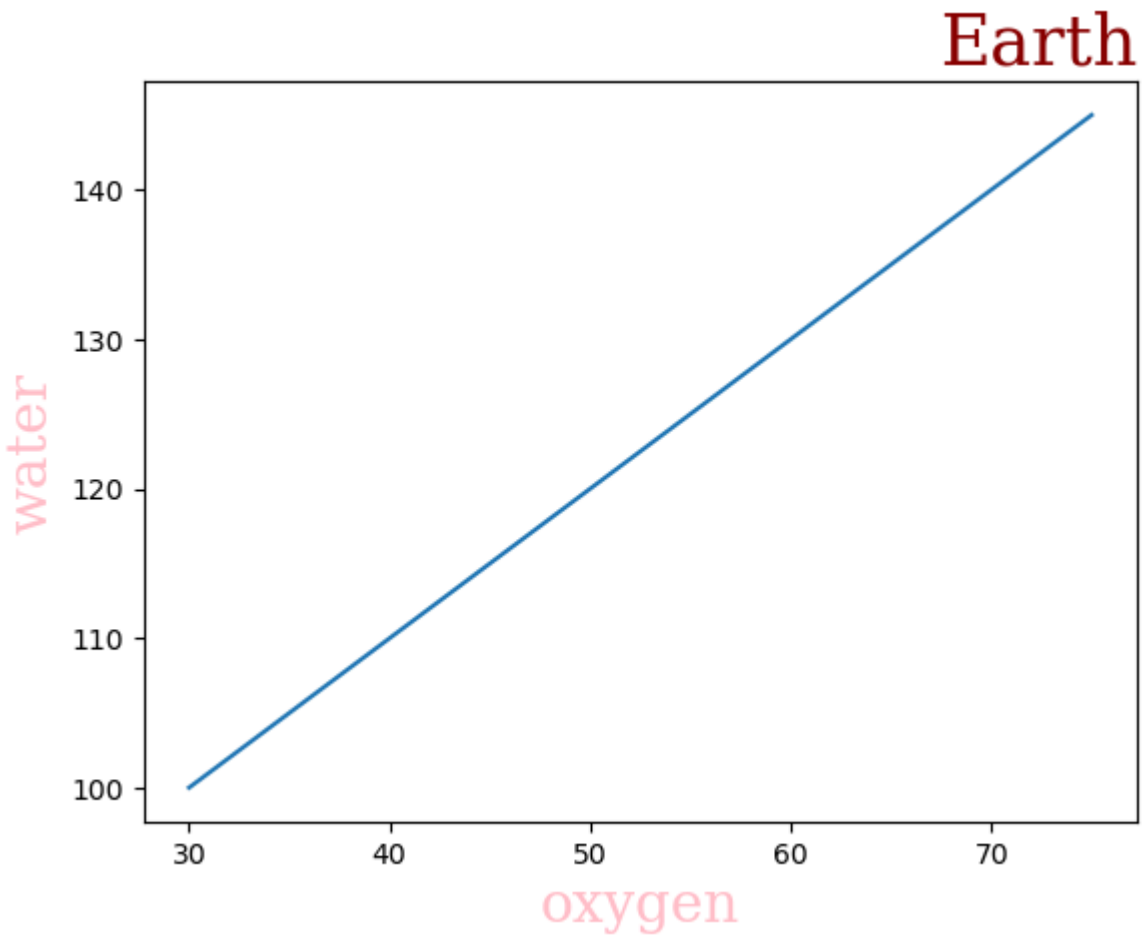
In [55]:
```
plt.plot(x2,y2)
plt.title('Earth',f2,loc='left')# location (loc)
plt.xlabel('oxygen',f1)
plt.ylabel('water',f1)
```

Out[55]:
```
Text(0, 0.5, 'water')
```



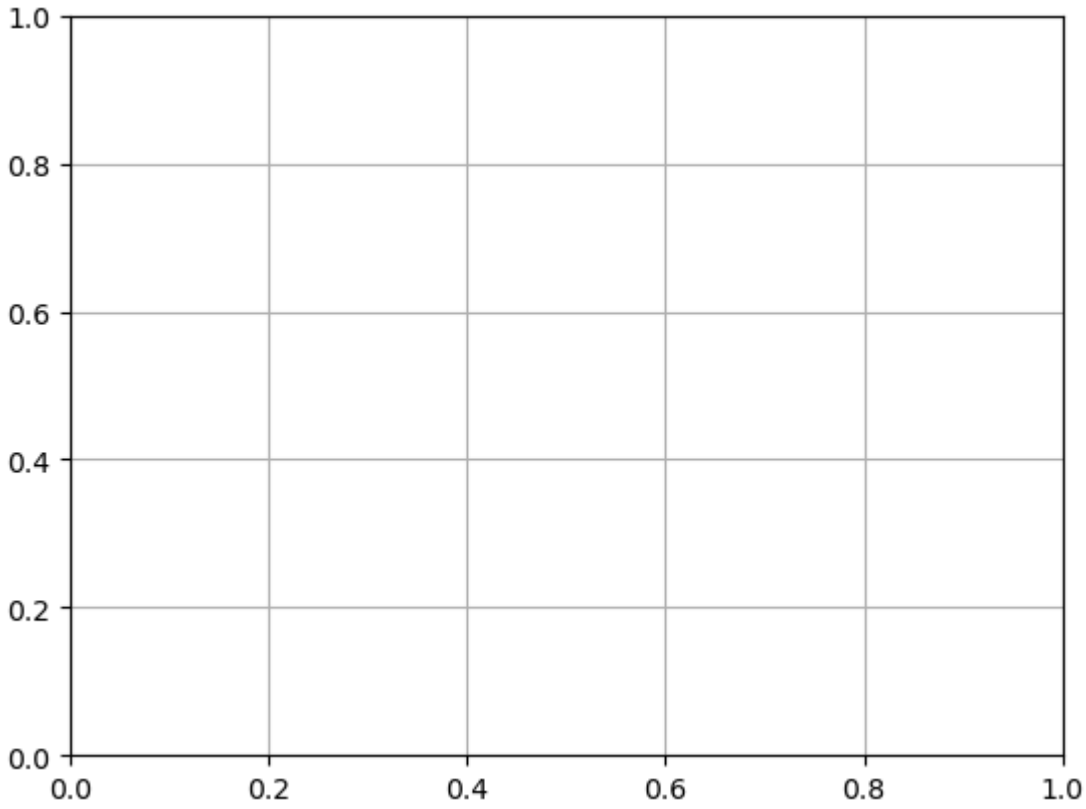
In [56]:
```
plt.plot(x2,y2)
plt.title('Earth',f2,loc='right')
plt.xlabel('oxygen',f1)
plt.ylabel('water',f1)
```

Out[56]:
```
Text(0, 0.5, 'water')
```

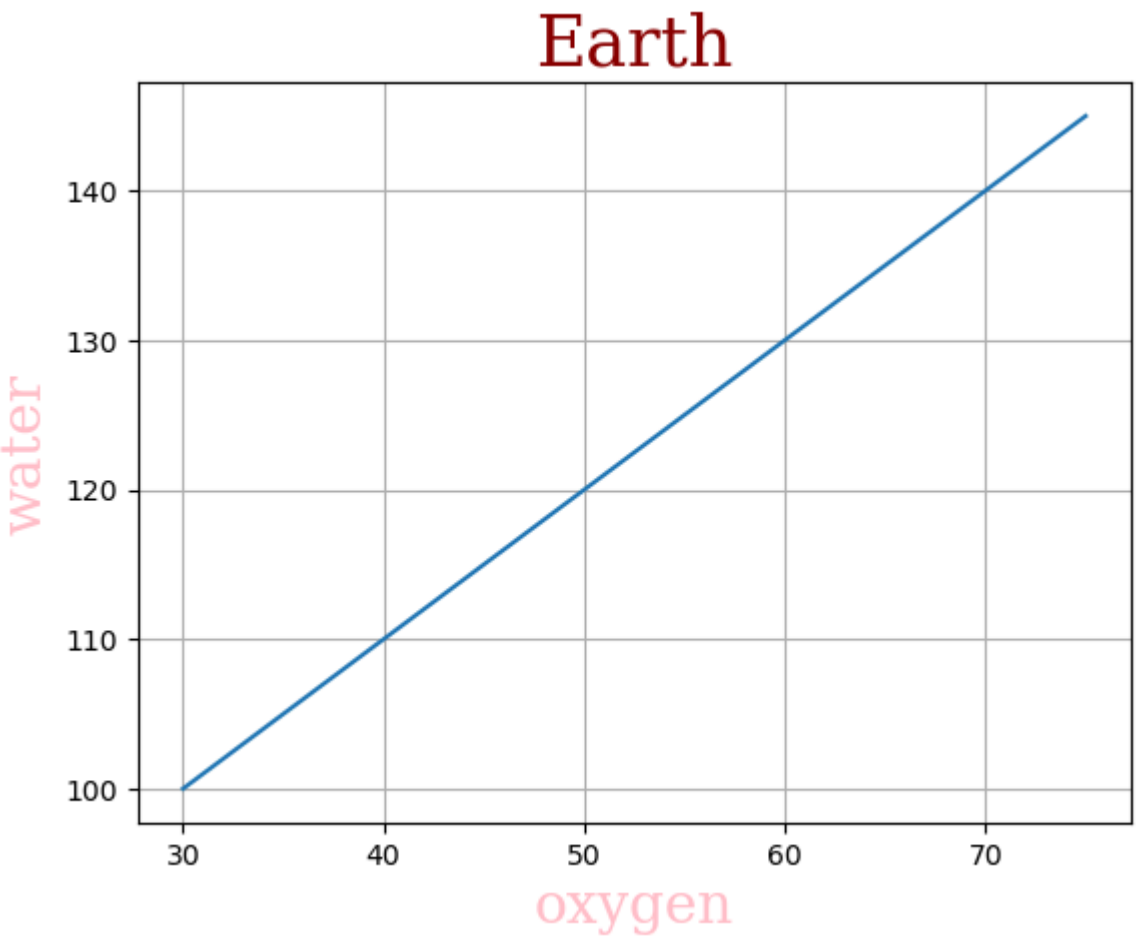


Adding the grid lines to a plot

In [57]:
```
plt.grid()
```

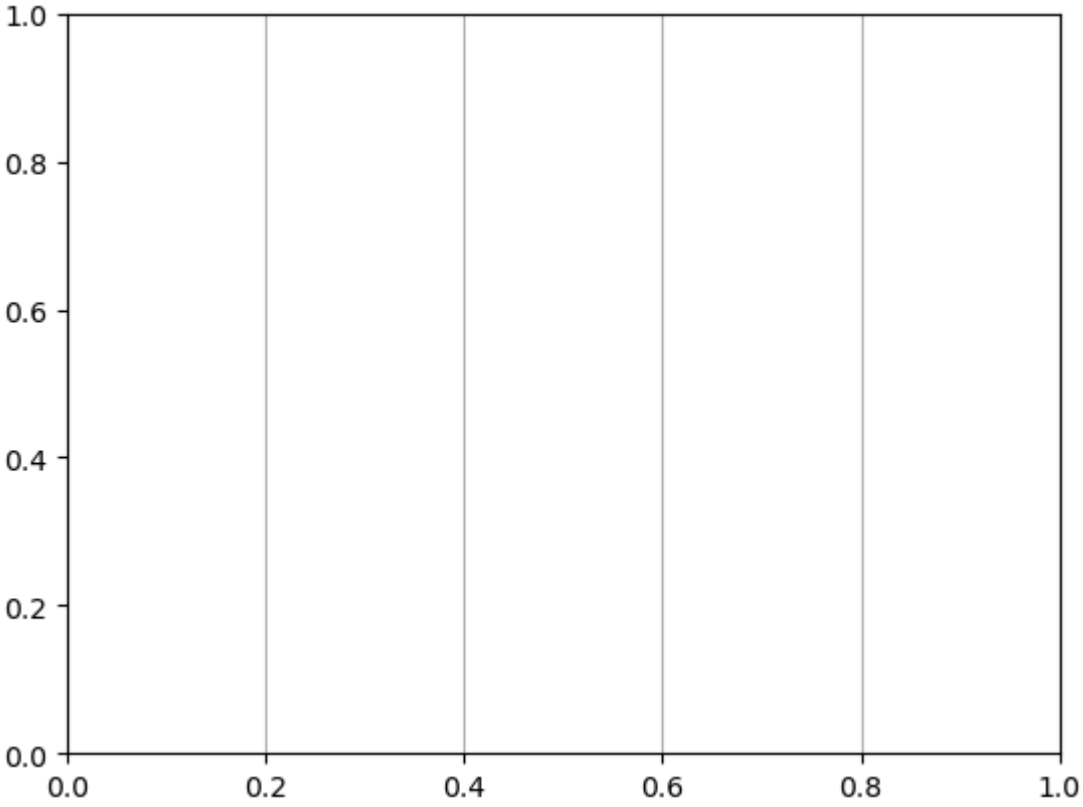


In [58]:
```
plt.plot(x2,y2)
plt.title('Earth',f2)
plt.xlabel('oxygen',f1)
plt.ylabel('water',f1)
plt.grid()
```

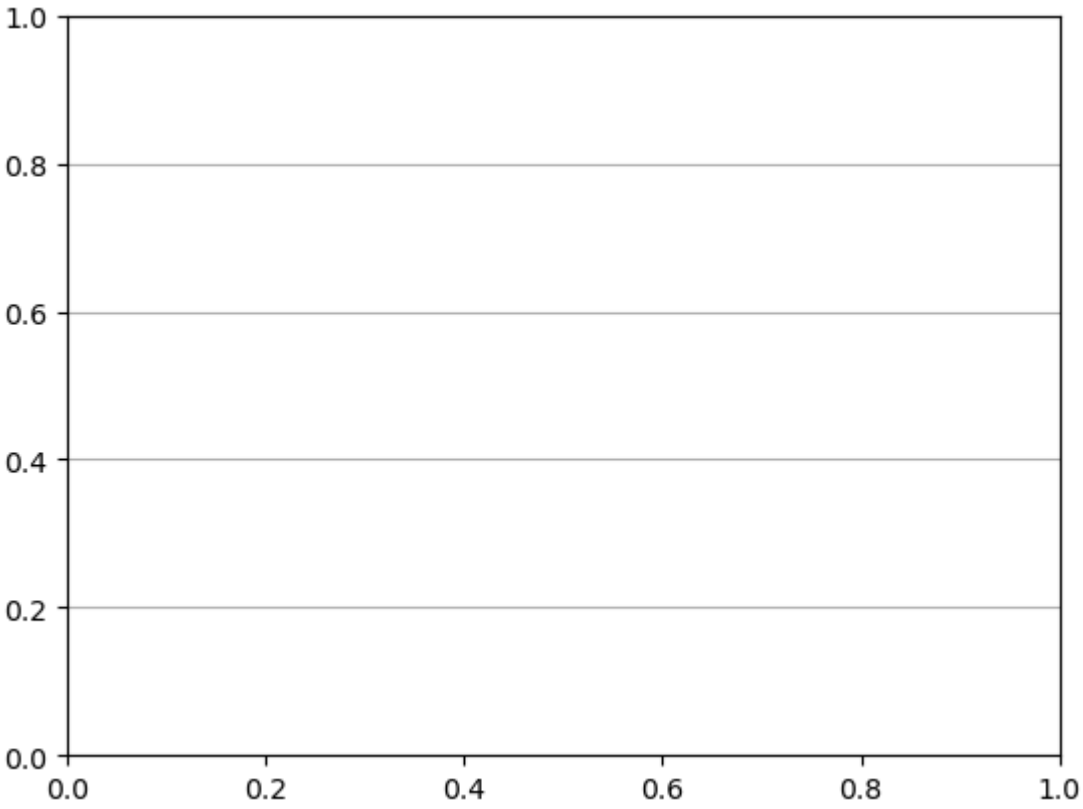


Now we will specify which grid line to display
`x axis or y axis`

In [59]:
```
plt.grid(axis='x')
```



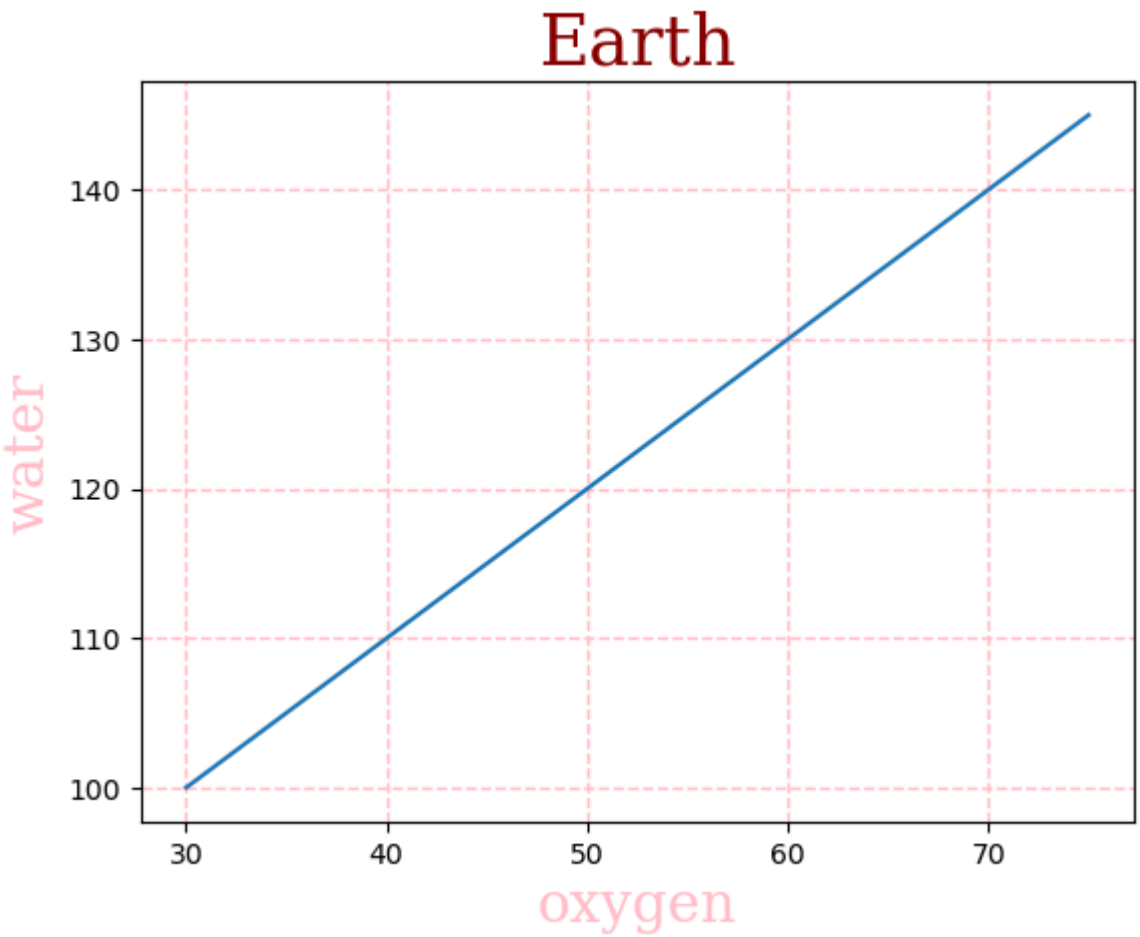
In [60]:
```
plt.grid(axis='y')
```



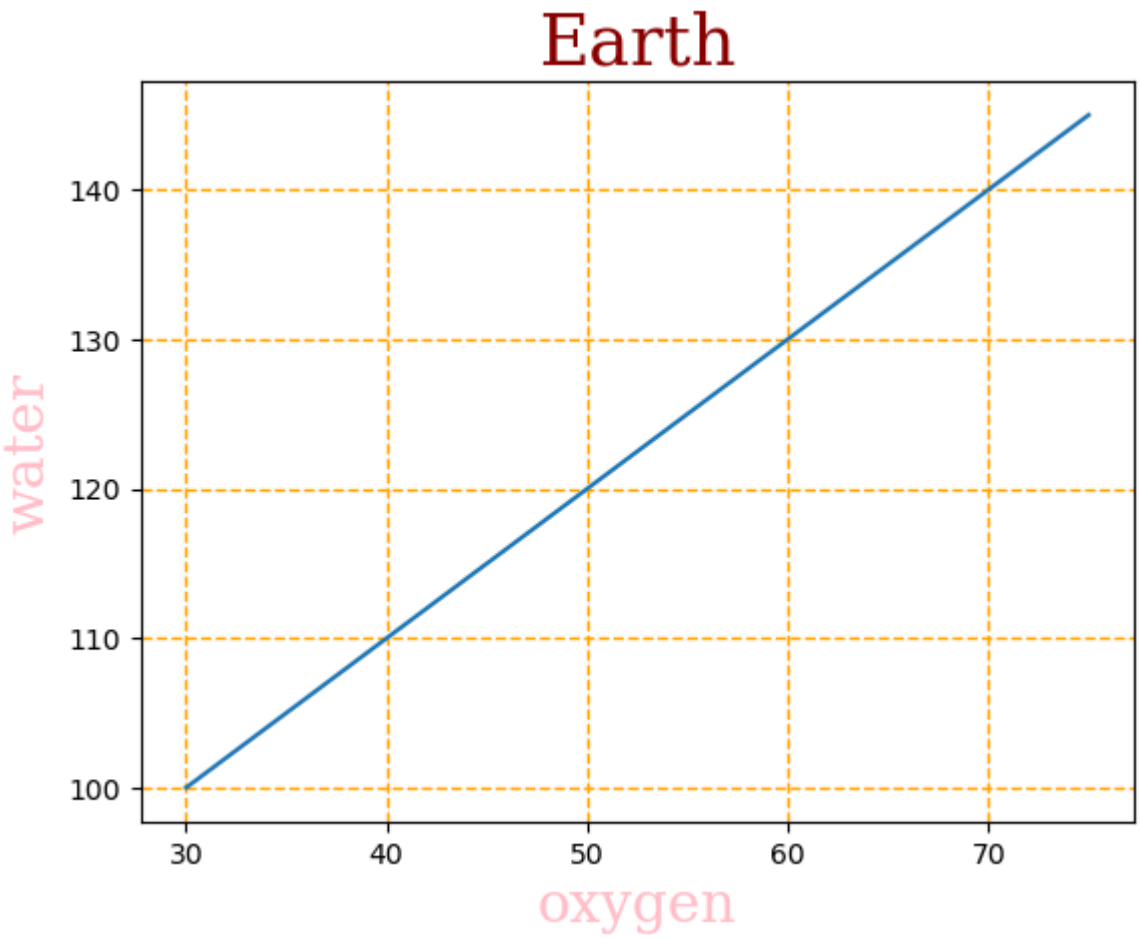
line properties for the grid

In [61]:
```
plt.plot(x2,y2)
plt.title('Earth',f2)
plt.xlabel('oxygen',f1)
```

```
plt.ylabel('water',f1)
plt.grid(color='pink',ls='--',linewidth=1)
```



In [62]:
```
plt.plot(x2,y2)
plt.title('Earth',f2)
plt.xlabel('oxygen',f1)
plt.ylabel('water',f1)
plt.grid(color='orange',ls='--',linewidth=1)
```



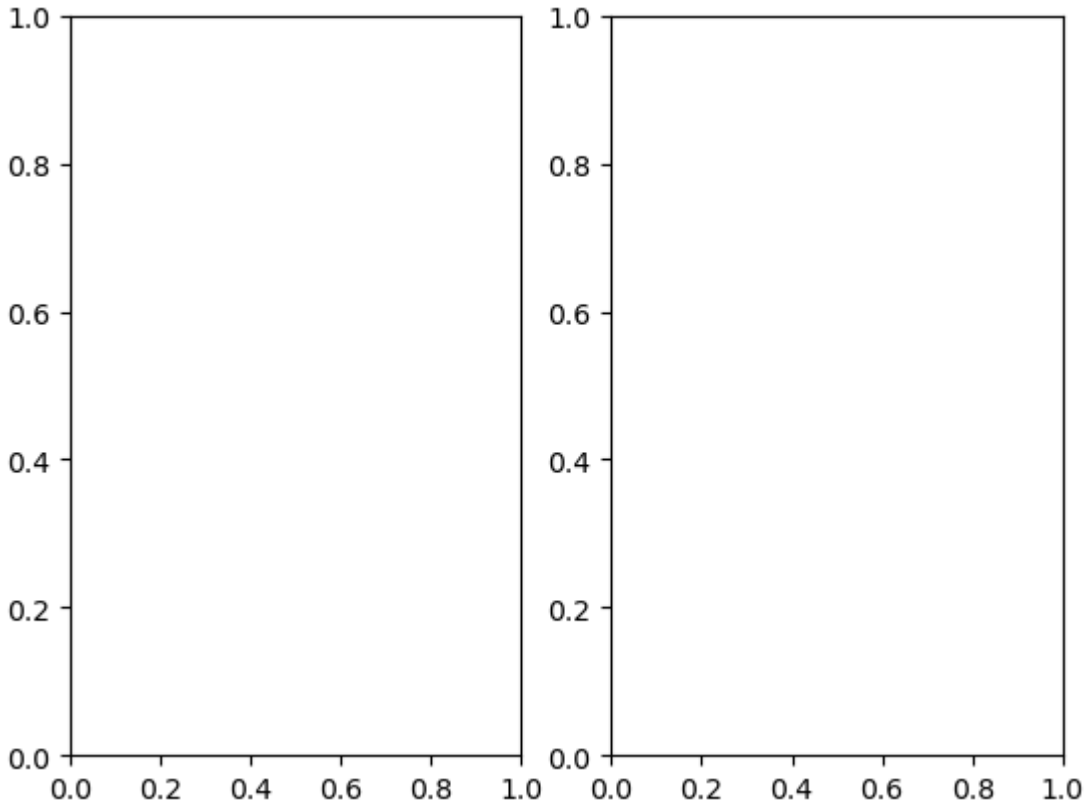
Display the multiple plots

In [113…

```
x3=np.array([0,2,4,6,8])
y3=np.array([5,3,6,3,12])
plt.subplot(1,2,1) # (rows, columns, panel number)
x4=np.array([1,2,3,4,5,6])
y4=np.array([0,30,24,35,43,29])
plt.subplot(1,2,2) # (rows, columns, panel number)
```

Out[113…

```
<Axes: >
```

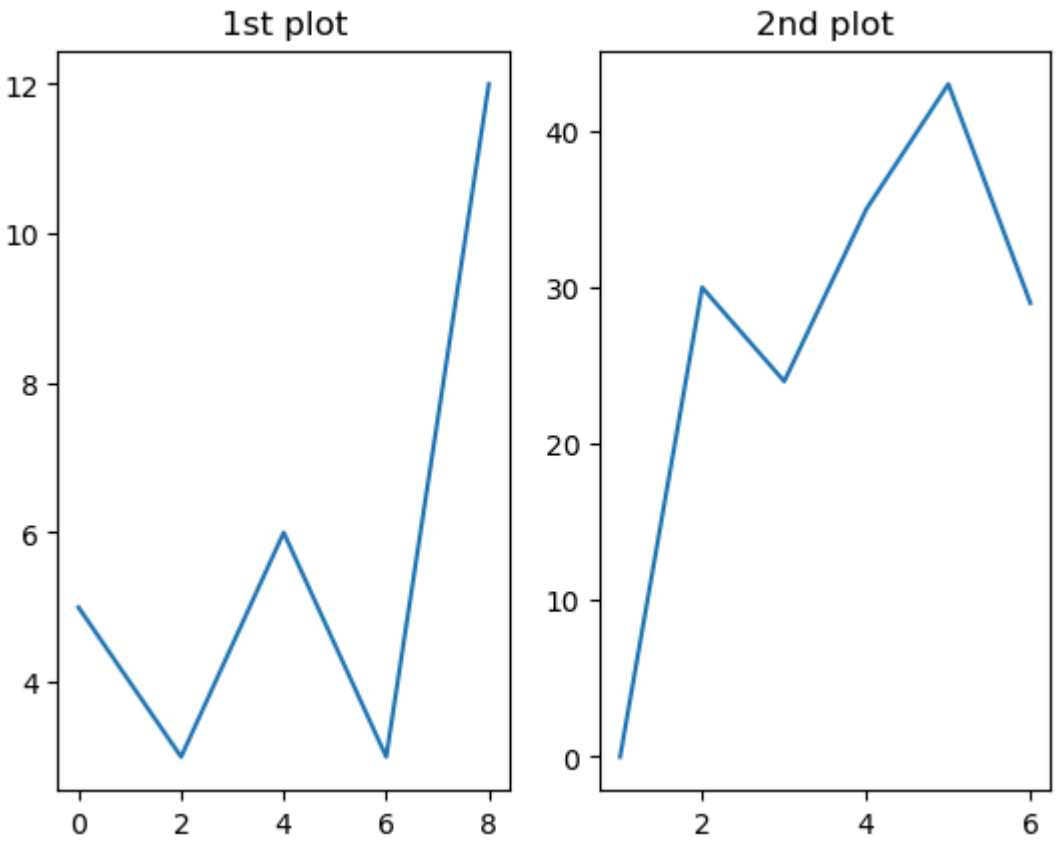


In [125…

```
x3=np.array([0,2,4,6,8])
y3=np.array([5,3,6,3,12])
plt.subplot(1,2,1) # (rows, columns, panel number)
plt.plot(x3,y3)
plt.title('1st plot')


x4=np.array([1,2,3,4,5,6])
y4=np.array([0,30,24,35,43,29])
plt.subplot(1,2,2) # (rows, columns, panel number)
plt.plot(x4,y4)
plt.title('2nd plot')
```

Out[125…

```
Text(0.5, 1.0, '2nd plot')
```

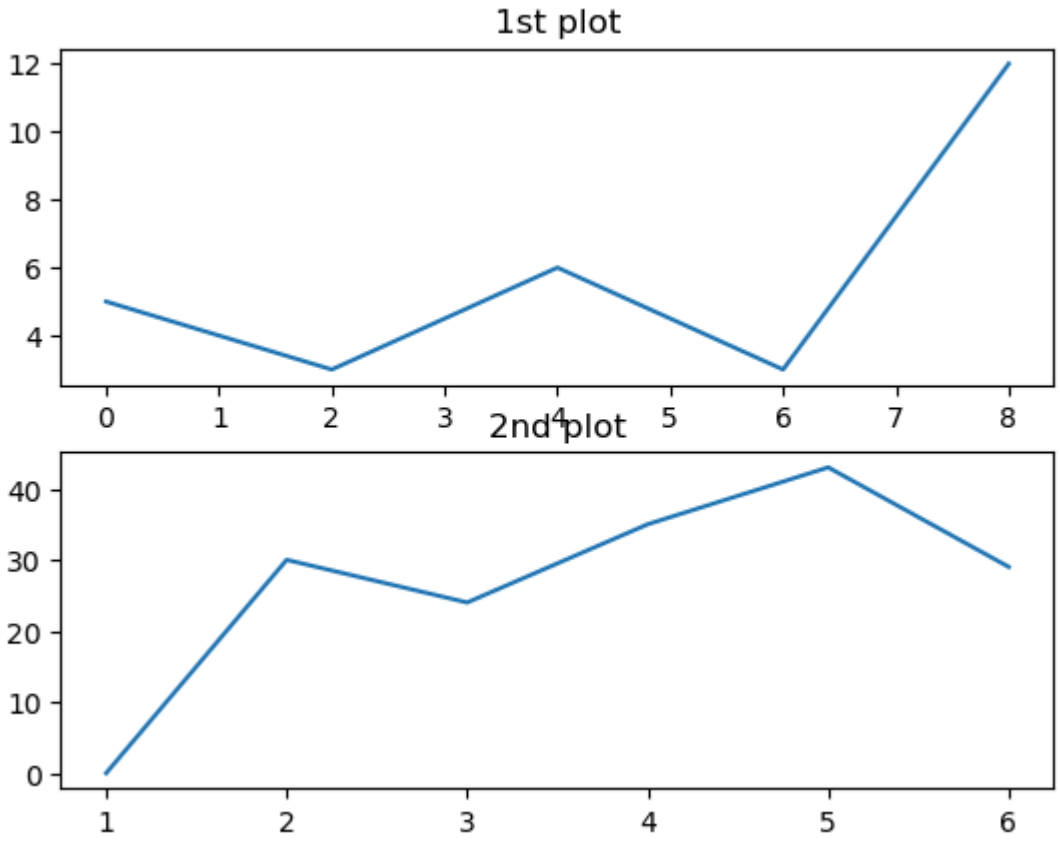


In [129…

```
x3=np.array([0,2,4,6,8])
y3=np.array([5,3,6,3,12])
plt.subplot(2,1,1) # (rows, columns, panel number)
plt.plot(x3,y3)
plt.title('1st plot')



x4=np.array([1,2,3,4,5,6])
y4=np.array([0,30,24,35,43,29])
plt.subplot(2,1,2) # (rows, columns, panel number)
plt.plot(x4,y4)
plt.title('2nd plot')
```

Out[129…

```
Text(0.5, 1.0, '2nd plot')
```

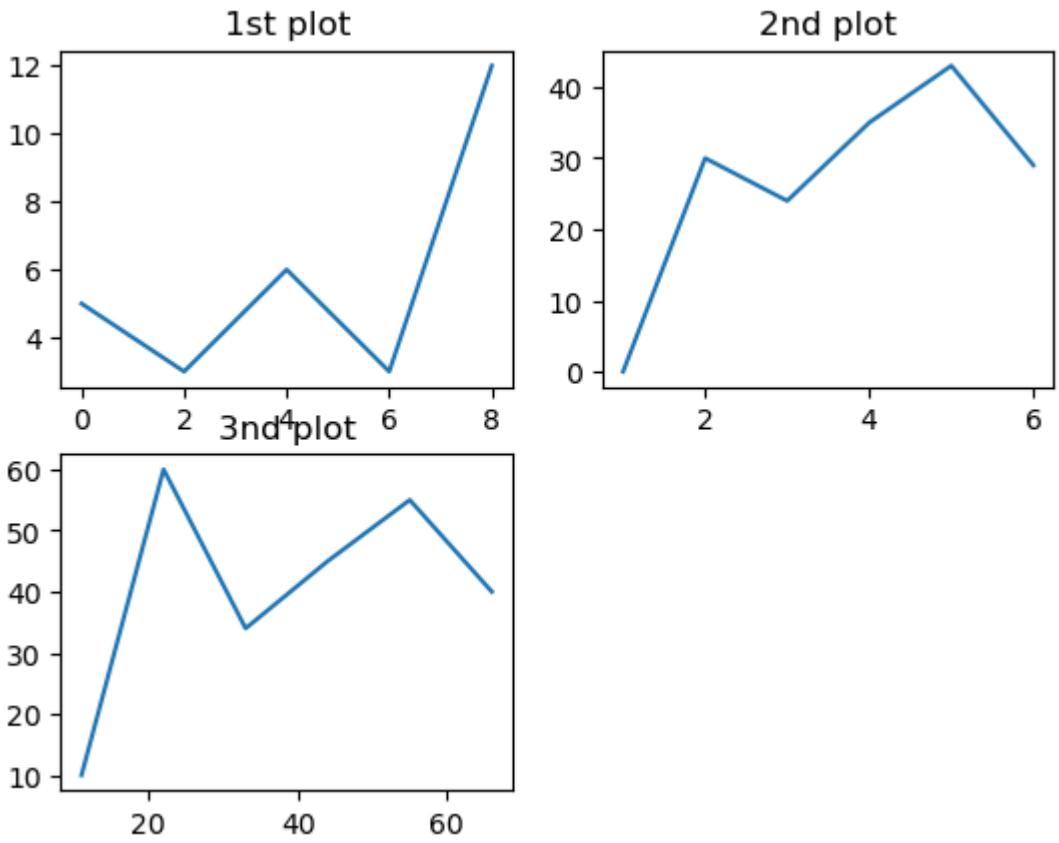


In [146…

```
x3=np.array([0,2,4,6,8])
y3=np.array([5,3,6,3,12])
plt.subplot(2,2,1) # (rows, columns, panel number)
plt.plot(x3,y3)
plt.title('1st plot')


x4=np.array([1,2,3,4,5,6])
y4=np.array([0,30,24,35,43,29])
plt.subplot(2,2,2) # (rows, columns, panel number)
plt.plot(x4,y4)
plt.title('2nd plot')



x5=np.array([11,22,33,44,55,66])
y5=np.array([10,60,34,45,55,40])
plt.subplot(2,2,3) # (rows, columns, panel number)
plt.plot(x5,y5)
plt.title('3nd plot')
```

Out[146…

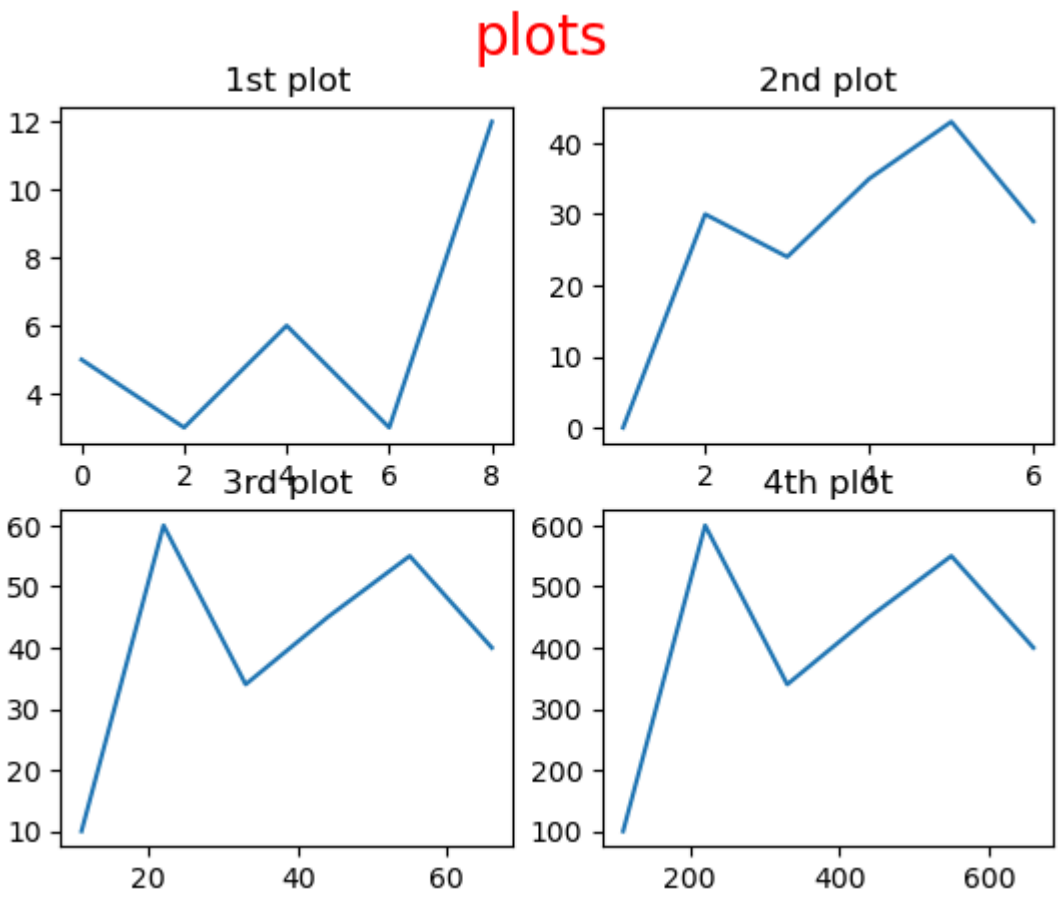
```
Text(0.5, 1.0, '3nd plot')
```



In [161…

```
x3=np.array([0,2,4,6,8])
y3=np.array([5,3,6,3,12])
plt.subplot(2,2,1) # (rows, columns, panel number)
plt.plot(x3,y3)
plt.title('1st plot')


x4=np.array([1,2,3,4,5,6])
y4=np.array([0,30,24,35,43,29])
plt.subplot(2,2,2) # (rows, columns, panel number)
plt.plot(x4,y4)
plt.title('2nd plot')



x5=np.array([11,22,33,44,55,66])
y5=np.array([10,60,34,45,55,40])
plt.subplot(2,2,3) # (rows, columns, panel number)
plt.plot(x5,y5)
plt.title('3rd plot')


x6=np.array([110,220,330,440,550,660])
y6=np.array([100,600,340,450,550,400])
plt.subplot(2,2,4) # (rows, columns, panel number)
plt.plot(x6,y6)
plt.title('4th plot')


plt.suptitle('plots',color='red',size=20)
```

Out[161… `Text(0.5, 0.98, 'plots')`

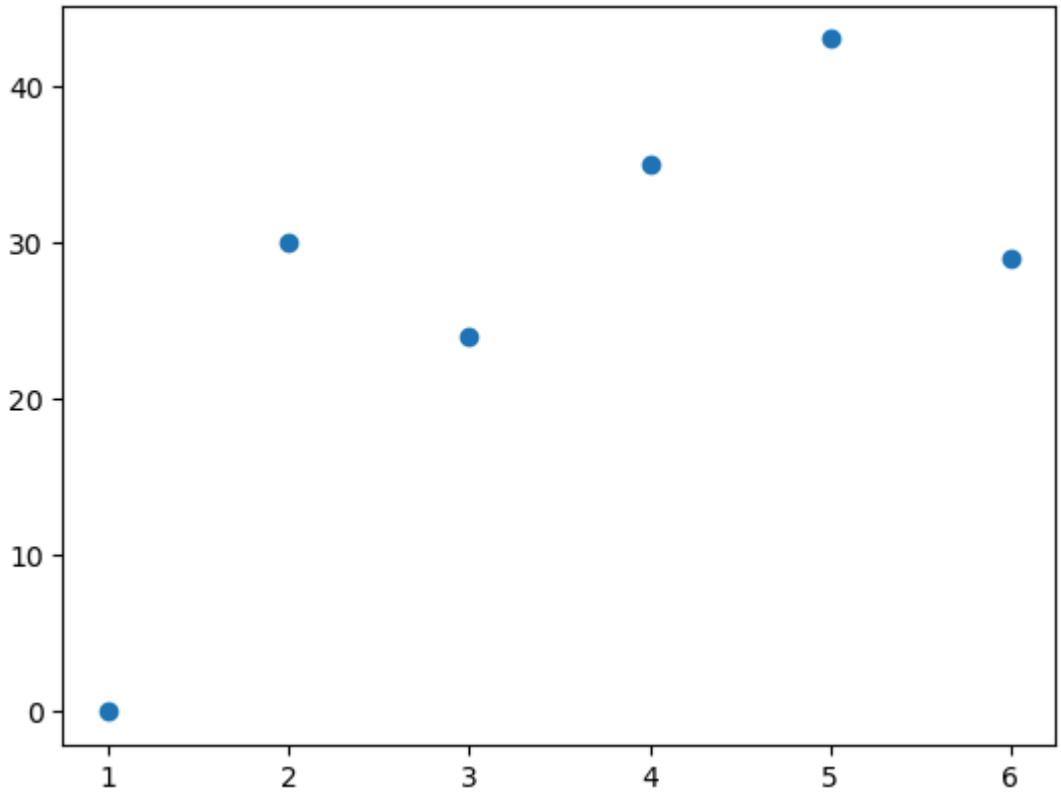


Scatter

the scatter() function plots oe for each observation...

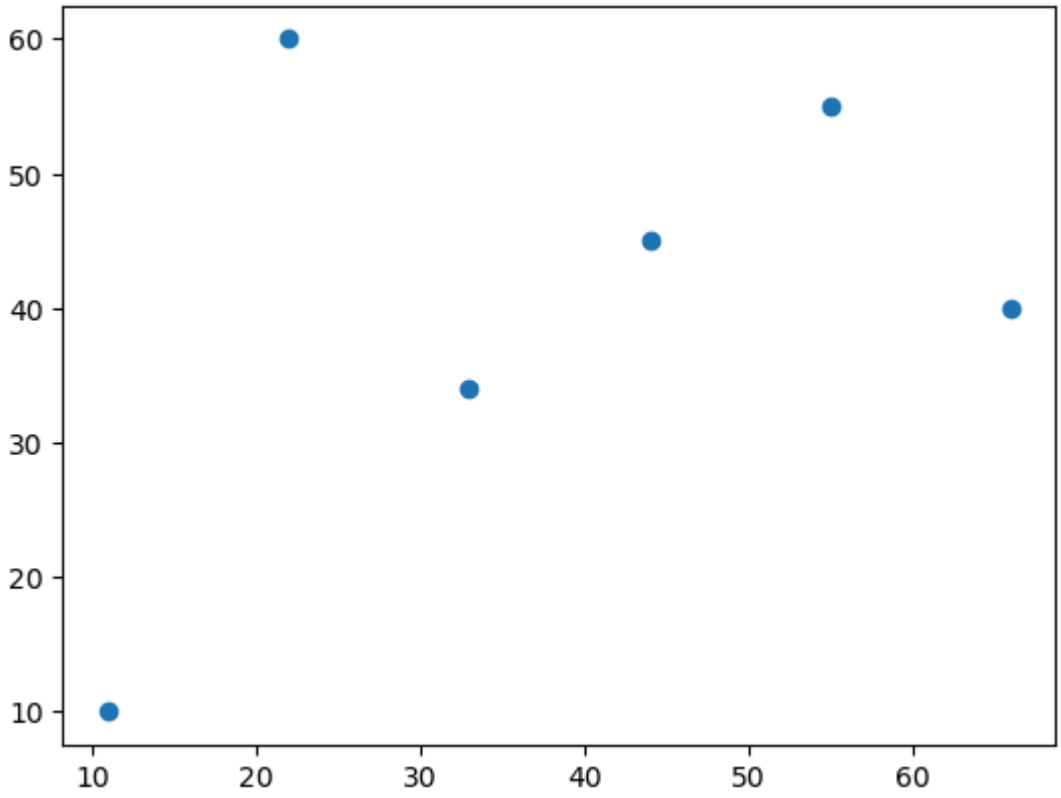
In [166…
```
print(plt.scatter(x4,y4))
```

```
<matplotlib.collections.PathCollection object at 0x0000020848AFCB90>
```



In [165…
```
print(plt.scatter(x5,y5))
```

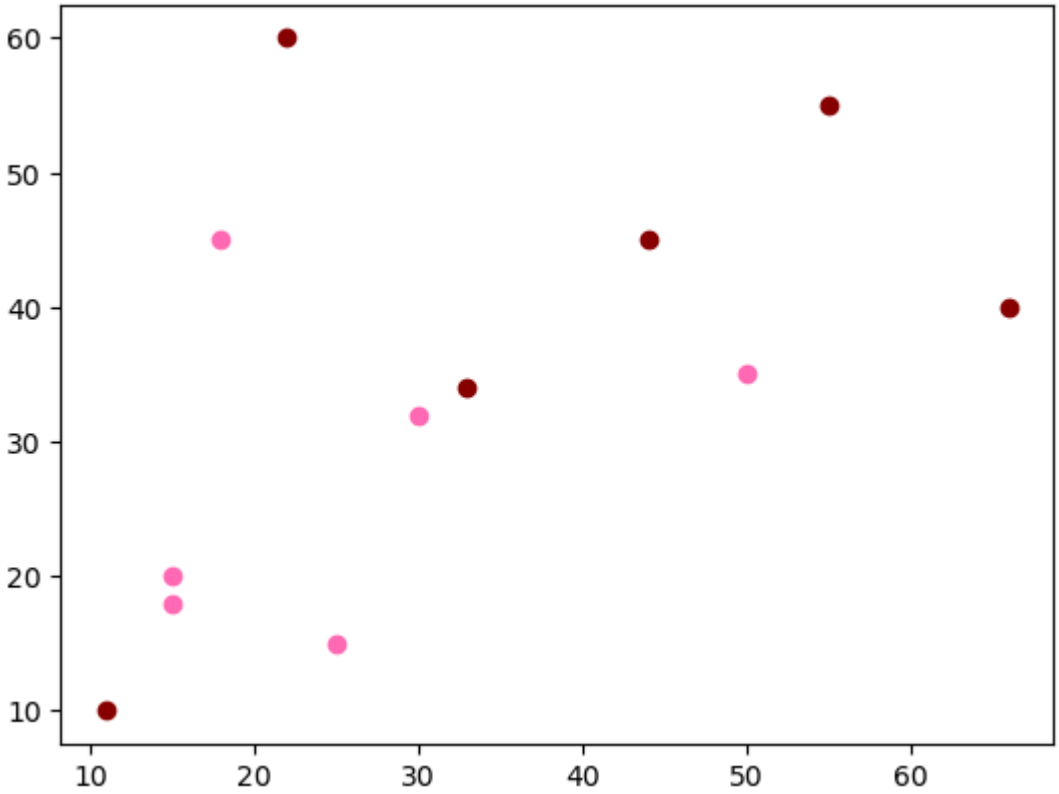
```
<matplotlib.collections.PathCollection object at 0x0000020848A06AD0>
```



In [167…
```
x7=np.array([15,18,25,15,30,50])
y7=np.array([18,45,15,20,32,35])
```

In [170…
```
print(plt.scatter(x5,y5,color='darkred'))
print(plt.scatter(x7,y7,color='hotpink'))
```

```
<matplotlib.collections.PathCollection object at 0x0000020848CEE490>
<matplotlib.collections.PathCollection object at 0x0000020848D00B90>
```

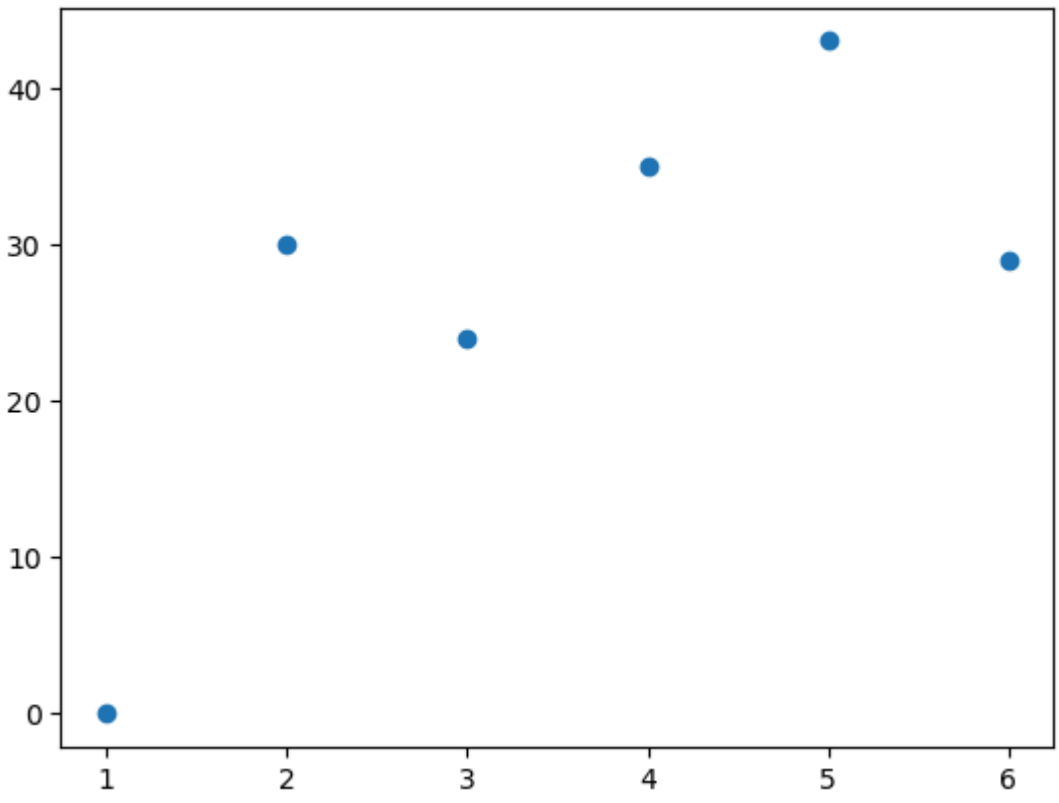


Change the color for dots

In [171…

```
print(plt.scatter(x4,y4))
```

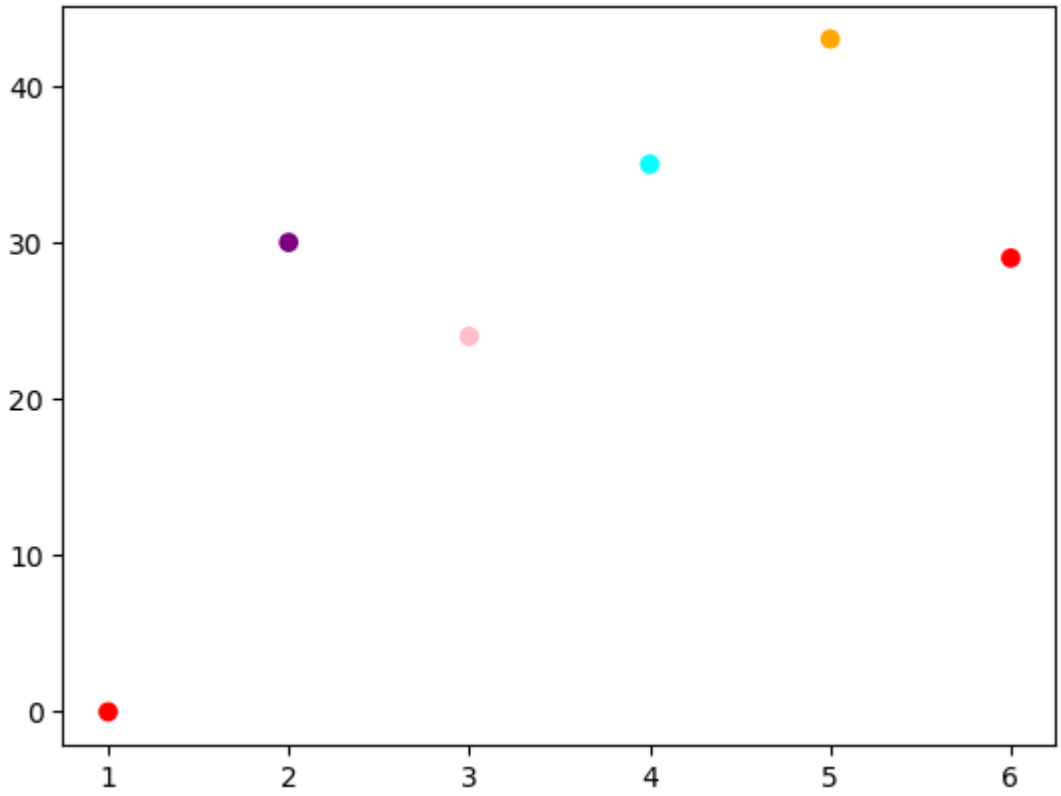
```
<matplotlib.collections.PathCollection object at 0x0000020848C97490>
```



In [177…
```
c1=(['red','purple','pink','cyan','orange','red'])
```

In [178…
```
print(plt.scatter(x4,y4,c=c1))
```

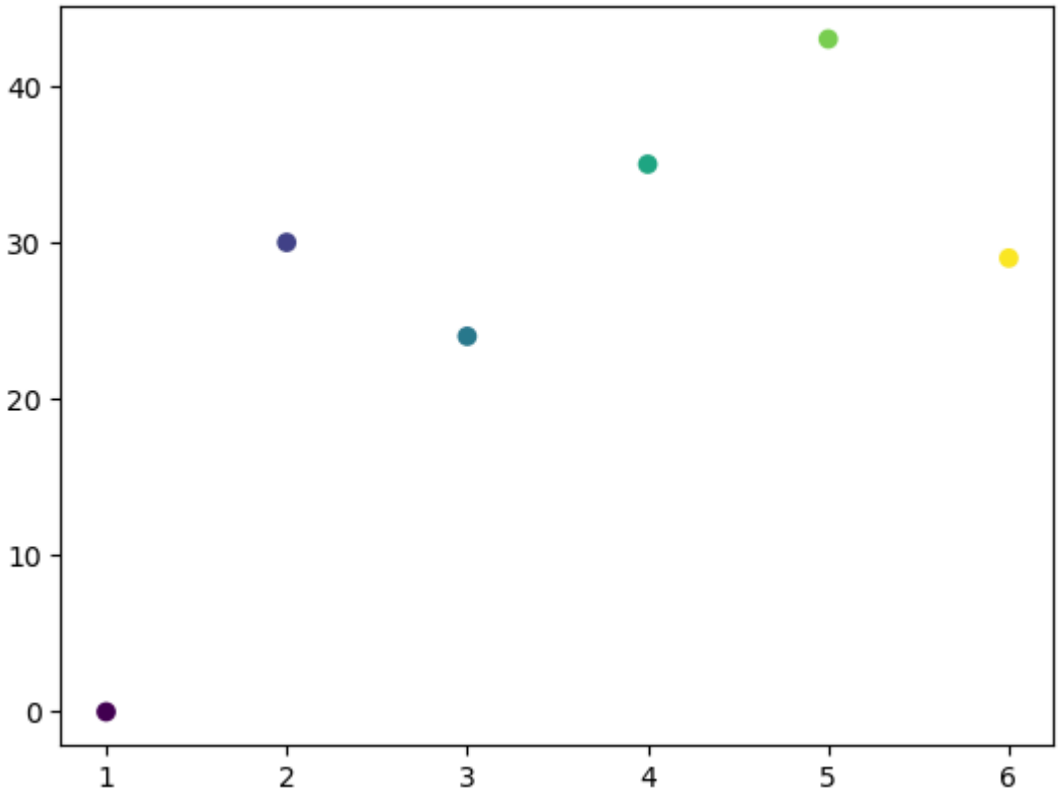
```
<matplotlib.collections.PathCollection object at 0x000002084887F750>
```



Specify a colormap in scatter plot

In [191…
```
c1=np.array([0,10,20,30,40,50])
plt.scatter(x4,y4,c=c1,cmap='viridis')
```

Out[191…
```
<matplotlib.collections.PathCollection at 0x2084ac2ad90>
```



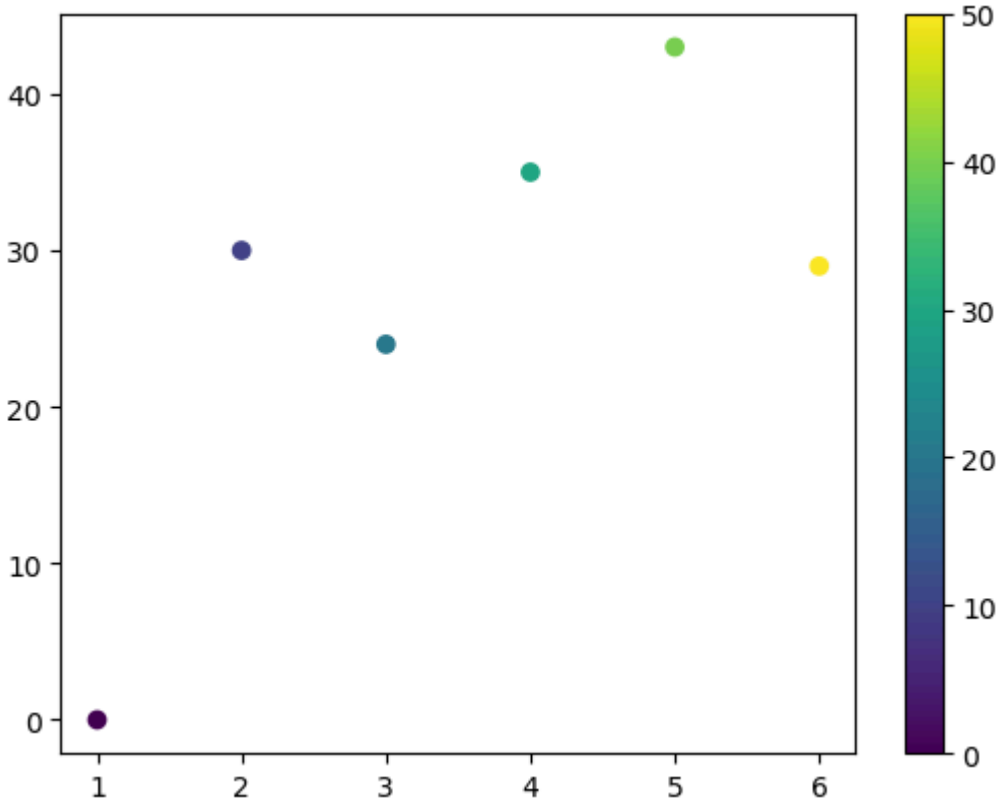
Specify a colorbar in scatter plot

In [196…

```
plt.scatter(x4,y4,c=c1,cmap='viridis')
plt.colorbar()
```

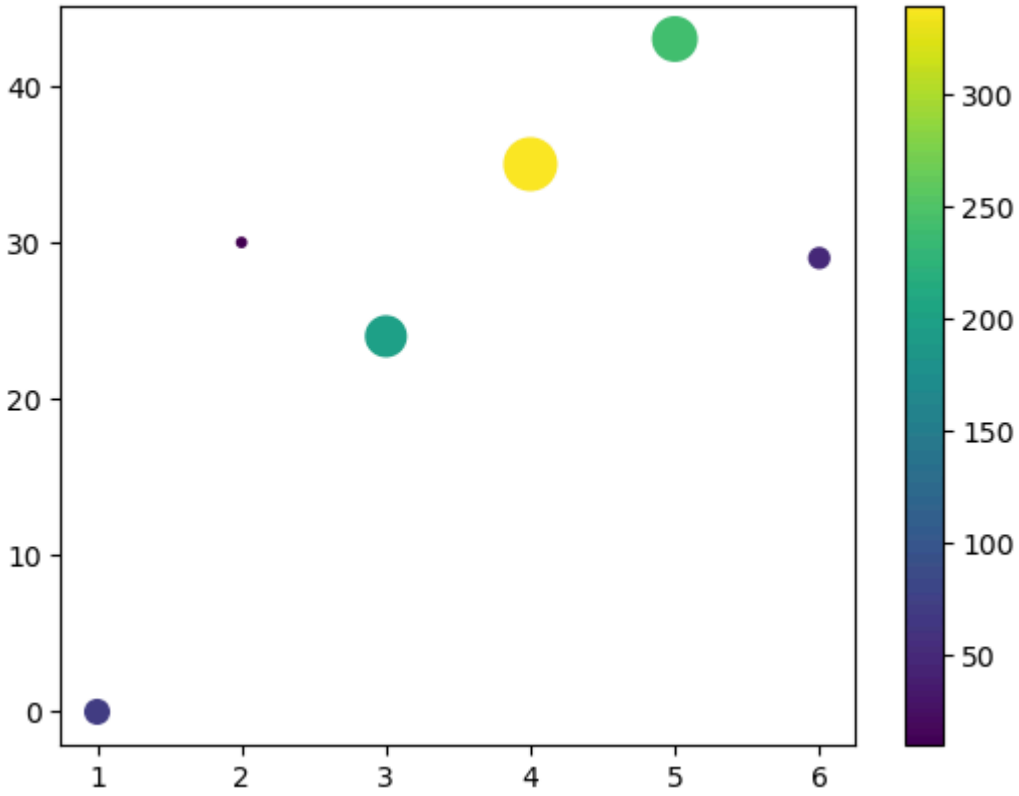
Out[196…

```
<matplotlib.colorbar.Colorbar at 0x2084abfba10>
```



In [202…
```
c1=np.array([70,10,200,340,240,50])
plt.scatter(x4,y4,c=c1,cmap='viridis',s=c1)
plt.colorbar()
```

Out[202…
```
<matplotlib.colorbar.Colorbar at 0x2084af20250>
```

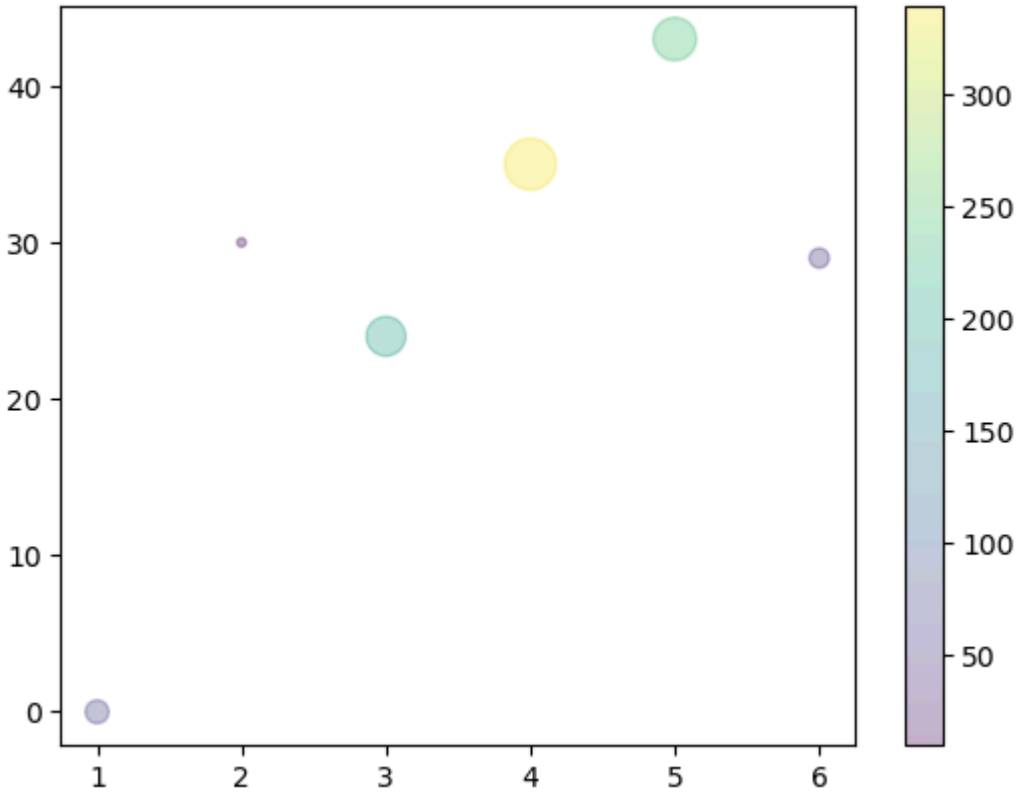


Alpha

Adjust the transprency of the dots

In [206…
```
plt.scatter(x4,y4,c=c1,cmap='viridis',s=c1,alpha=0.3)
plt.colorbar()
```

Out[206…
```
<matplotlib.colorbar.Colorbar at 0x2084aee3a10>
```

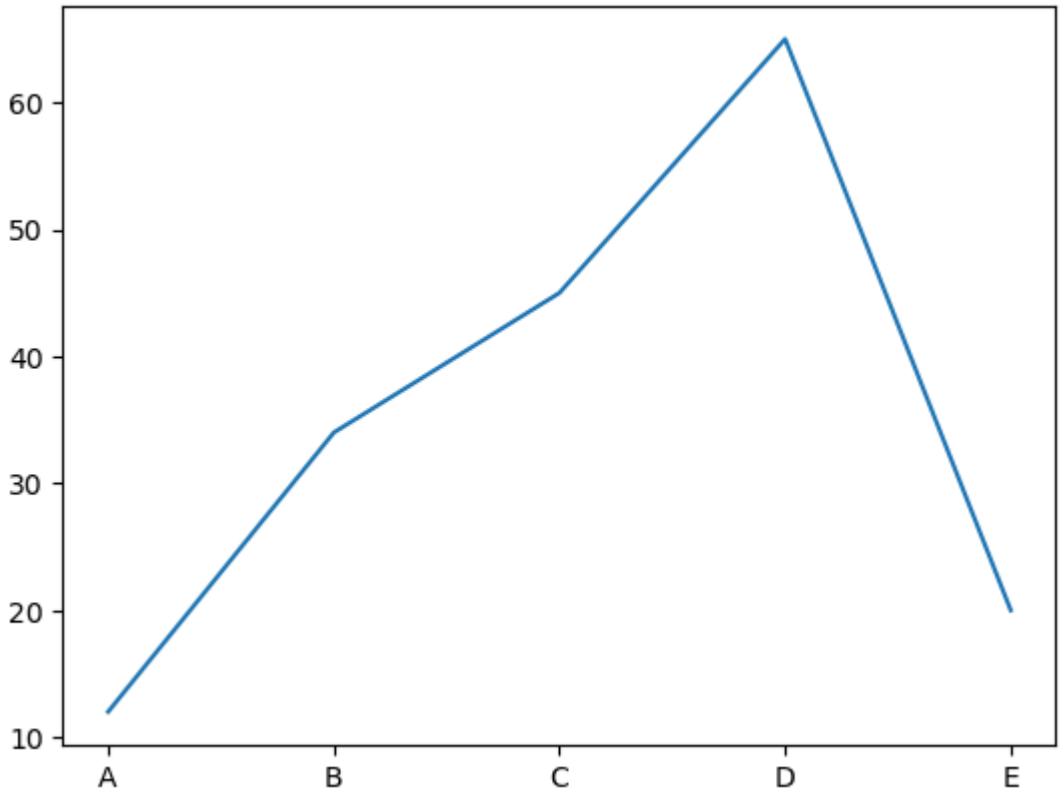


Create bars--(bar())

In [207…
```
x8=np.array(["A","B","C","D","E"])
y8=np.array([12,34,45,65,20])
```

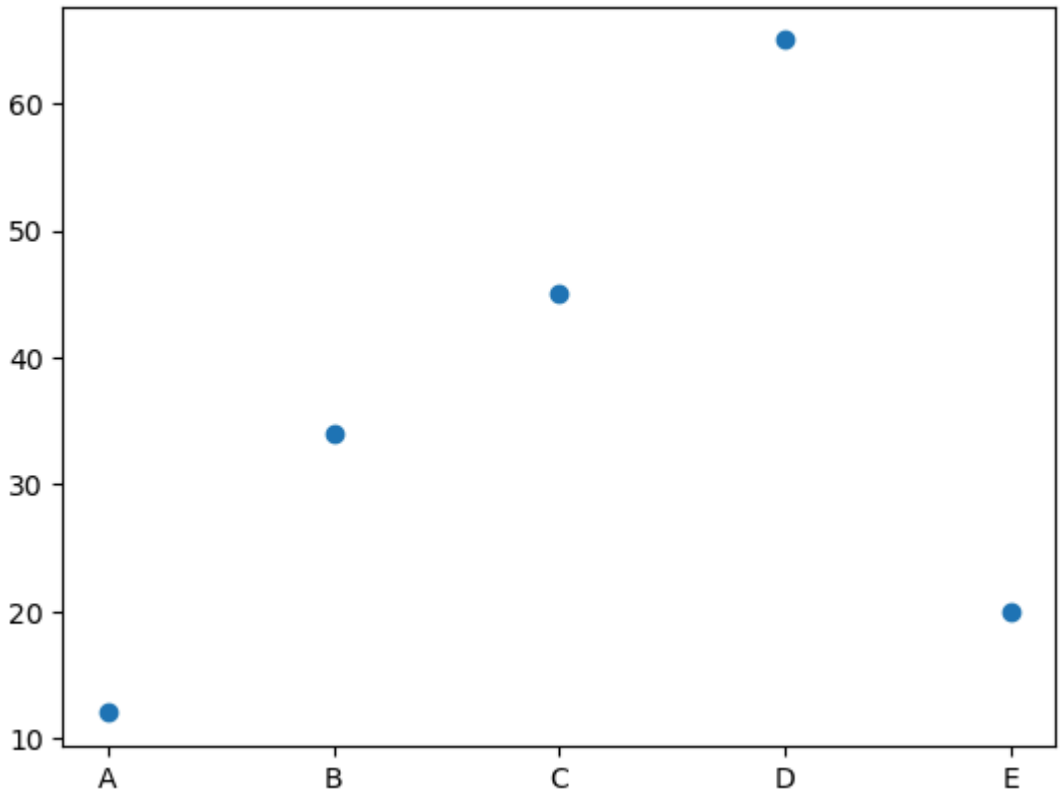
In [208…
```
plt.plot(x8,y8)
```

Out[208…
```
[<matplotlib.lines.Line2D at 0x2084c3b6a50>]
```



In [209…
```
plt.scatter(x8,y8)
```

Out[209…
```
<matplotlib.collections.PathCollection at 0x2084c3b4350>
```

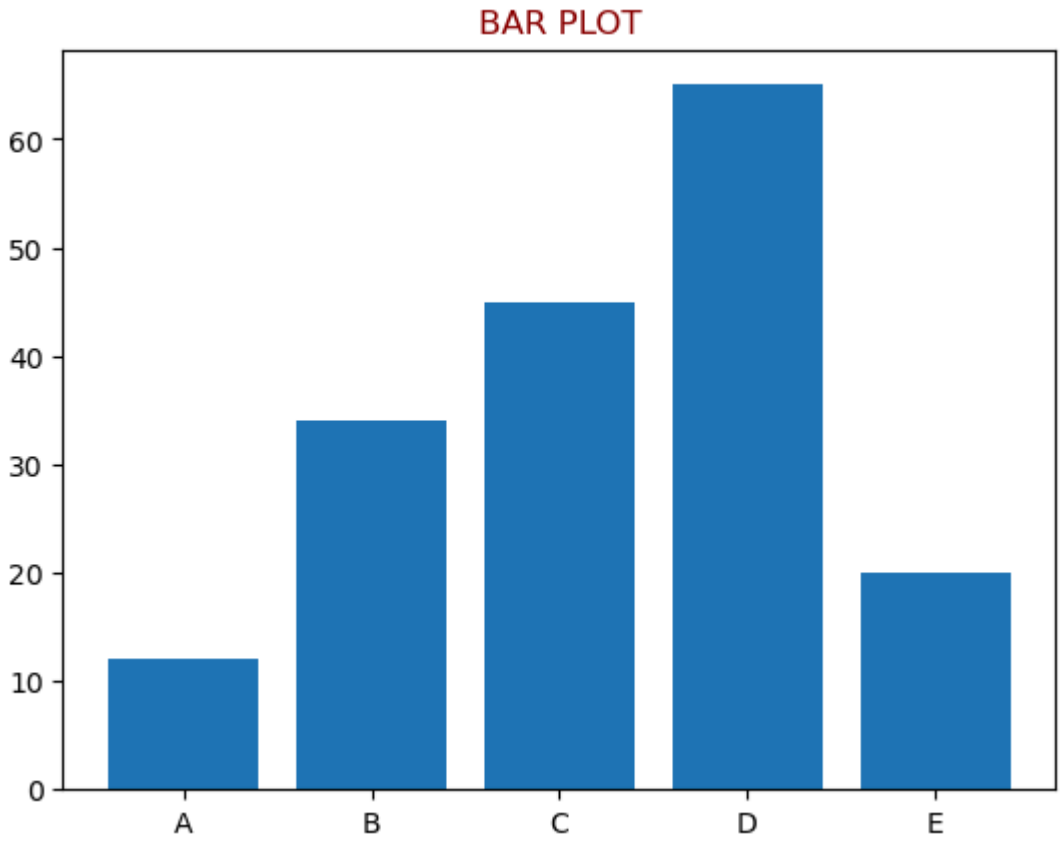


Vertical bar

In [216…
```
plt.bar(x8,y8) #(bar())
```

```
plt.title('BAR PLOT',color='darkred')
```

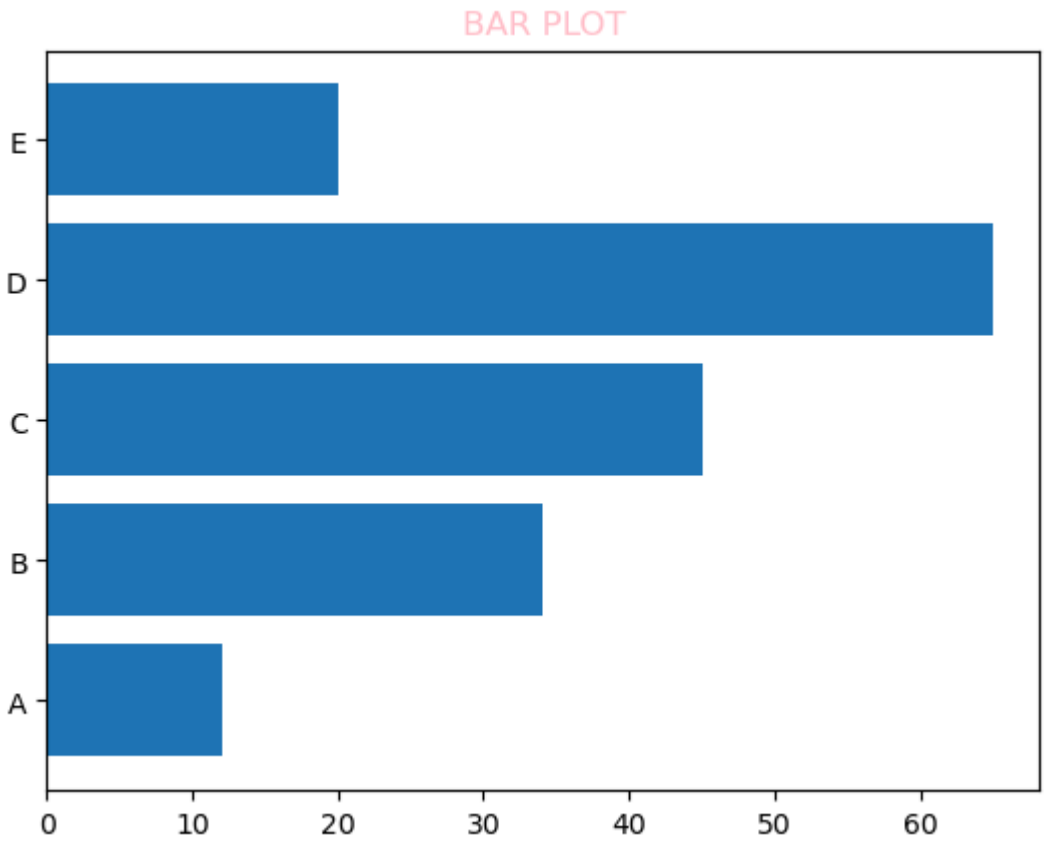
Out[216… Text(0.5, 1.0, 'BAR PLOT')



Horizontal bar

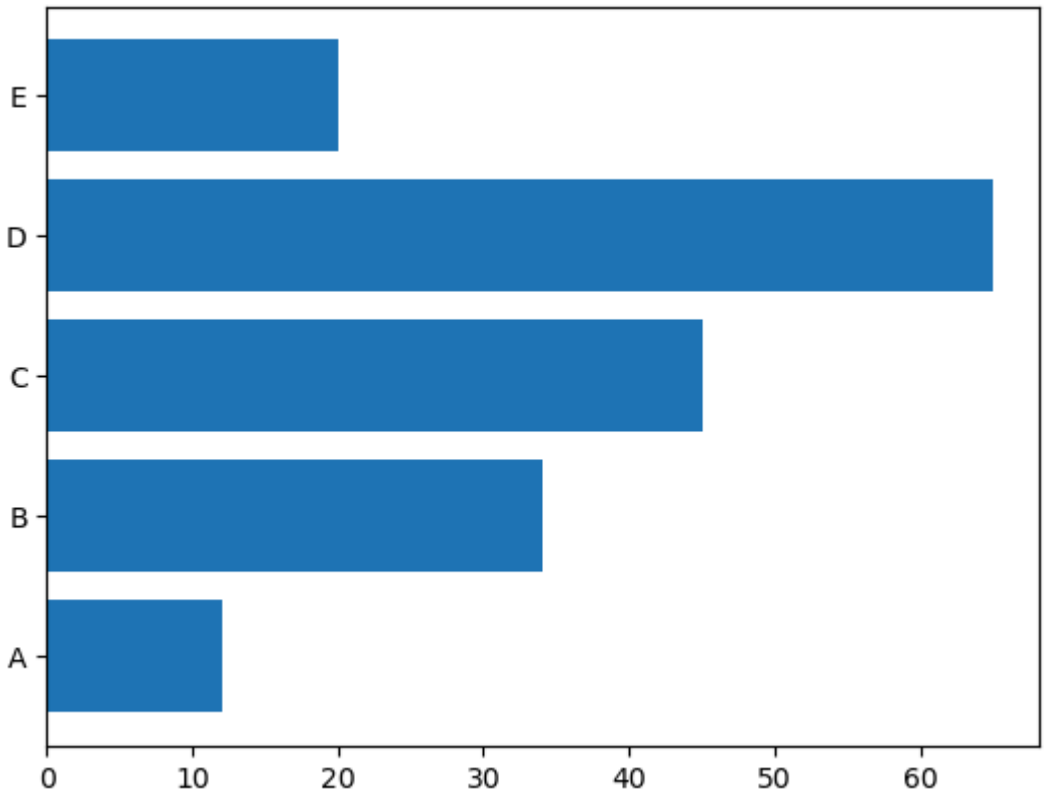
In [223…
```
plt.barh(x8,y8) #(bar())
plt.title('BAR PLOT',color='pink')
```

Out[223… Text(0.5, 1.0, 'BAR PLOT')



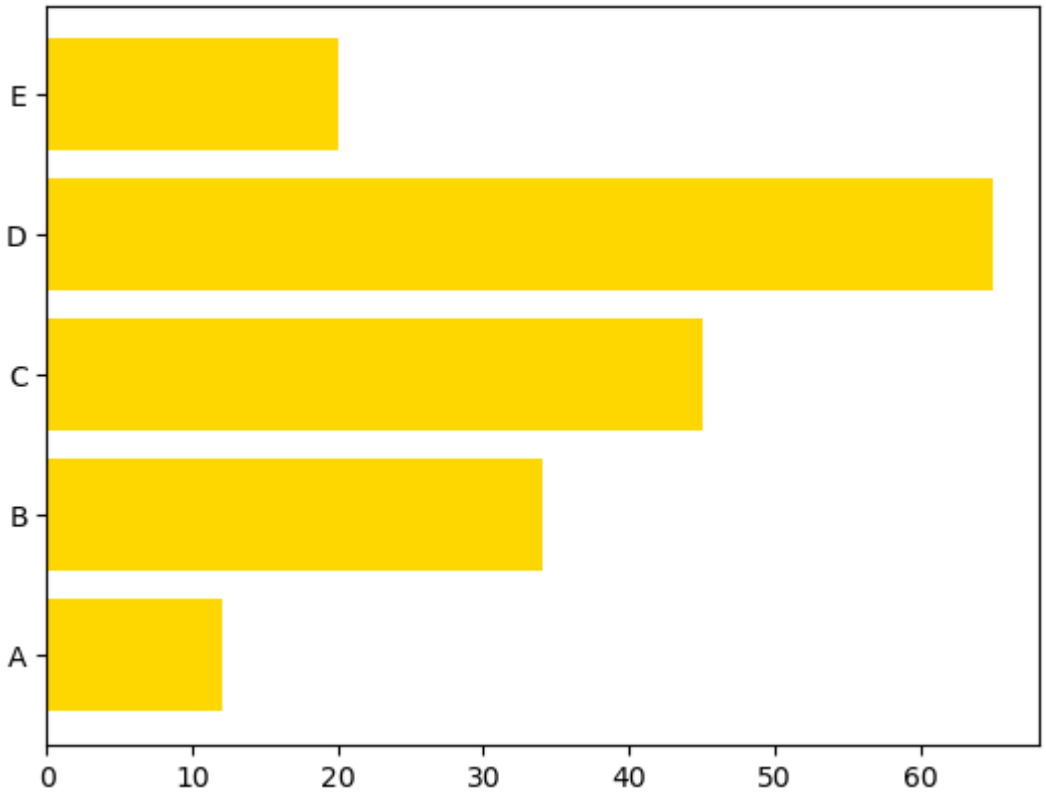
In [226…
```
plt.barh(x8,y8) #(bar())
```

Out[226…
```
<BarContainer object of 5 artists>
```



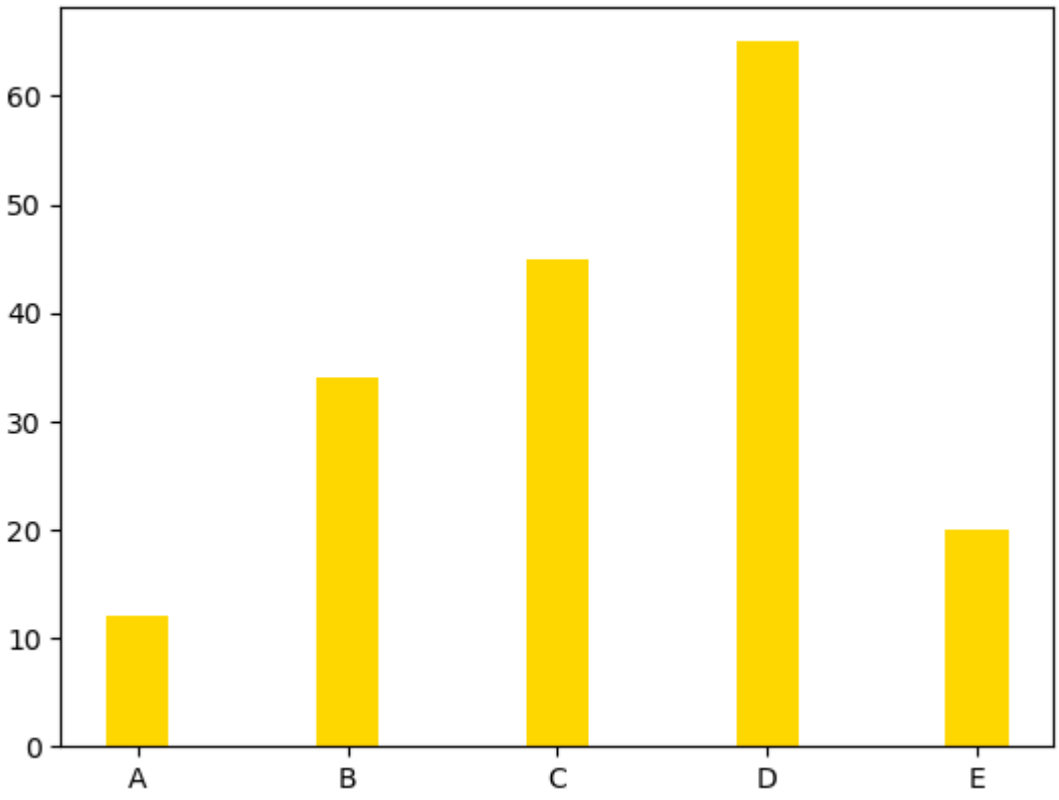
In [231…
```
plt.barh(x8,y8,color='gold') #(bar())
```

Out[231…
```
<BarContainer object of 5 artists>
```



In [235…
```
plt.bar(x8,y8,color='gold',width=0.3)
```

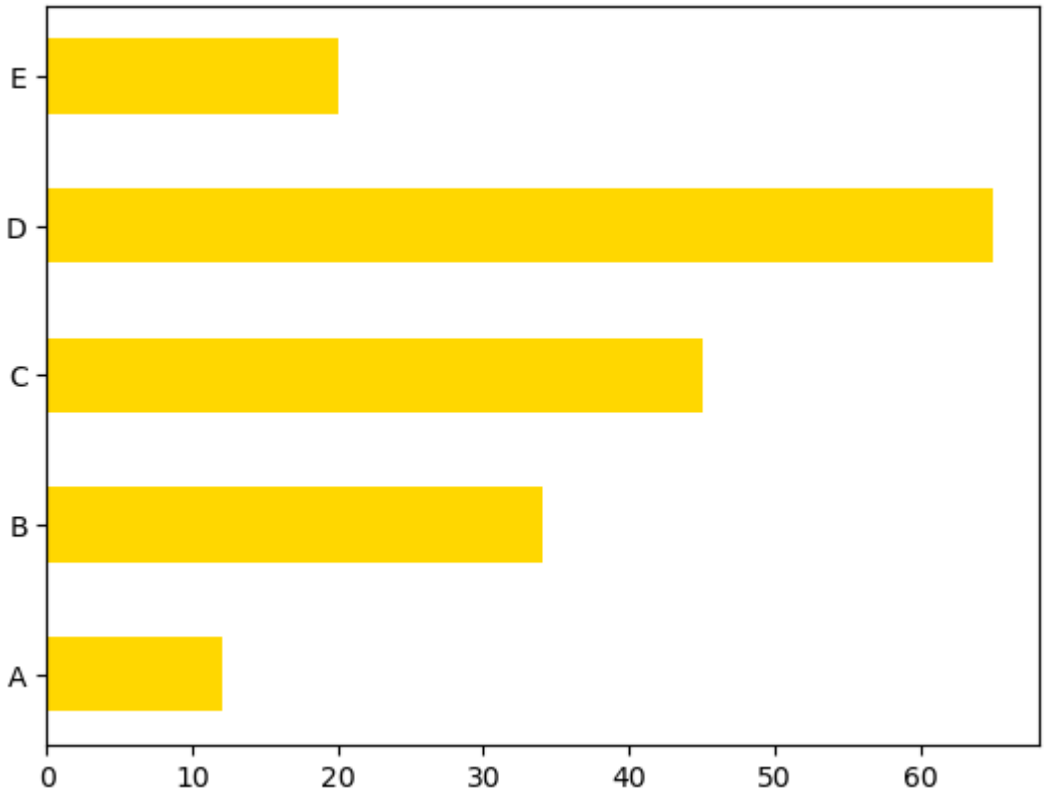
Out[235…
```
<BarContainer object of 5 artists>
```



****** For horizontal to use height insted of width*****

In [238…
```
plt.barh(x8,y8,color='gold',height=0.5)
```

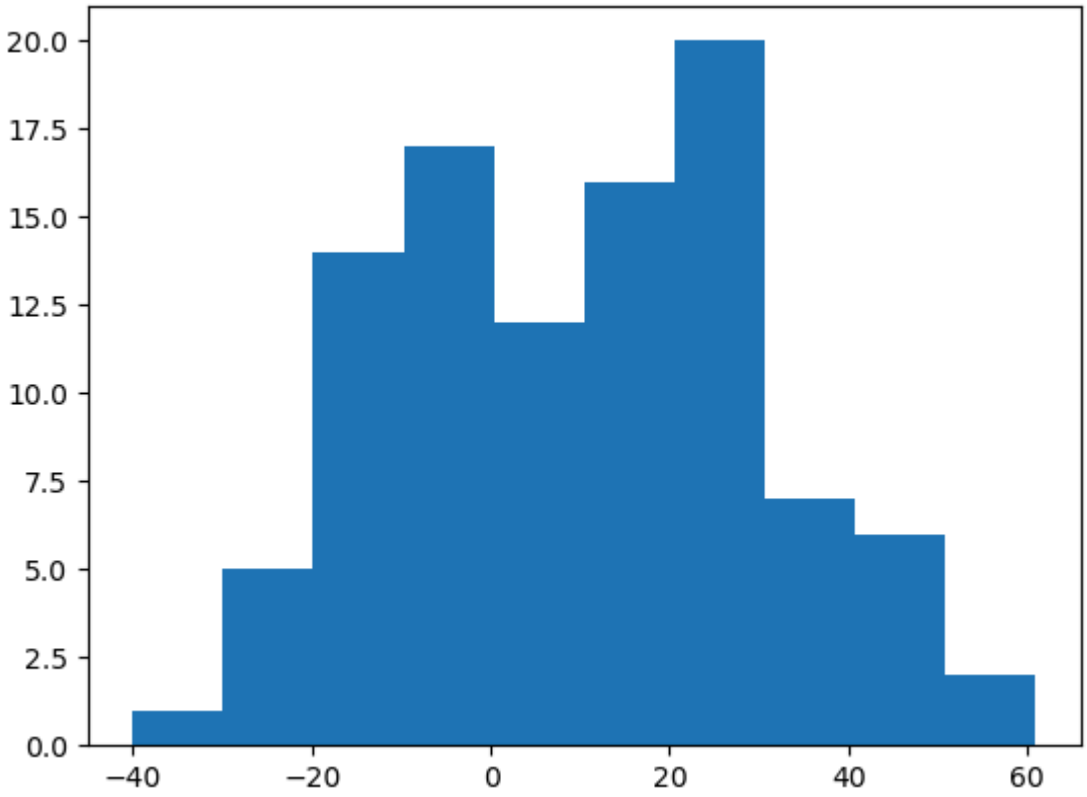
Out[238… <BarContainer object of 5 artists>



Histogram

In [253…
```
d1 = np.random.normal(10,20,100)
plt.hist(d1)
```

Out[253…
```
(array([ 1.,  5., 14., 17., 12., 16., 20.,  7.,  6.,  2.]),
 array([-40.01284227, -29.92193117, -19.83102007,  -9.74010897,
          0.35080212,  10.44171322,  20.53262432,  30.62353542,
         40.71444651,  50.80535761,  60.89626871]),
 <BarContainer object of 10 artists>)
```

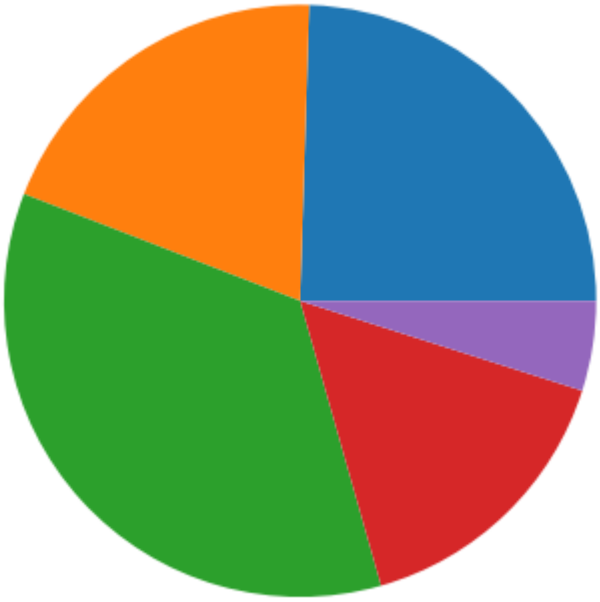


Piechart

In [264…
```
x9=np.array([25,20,36,16,5])
```

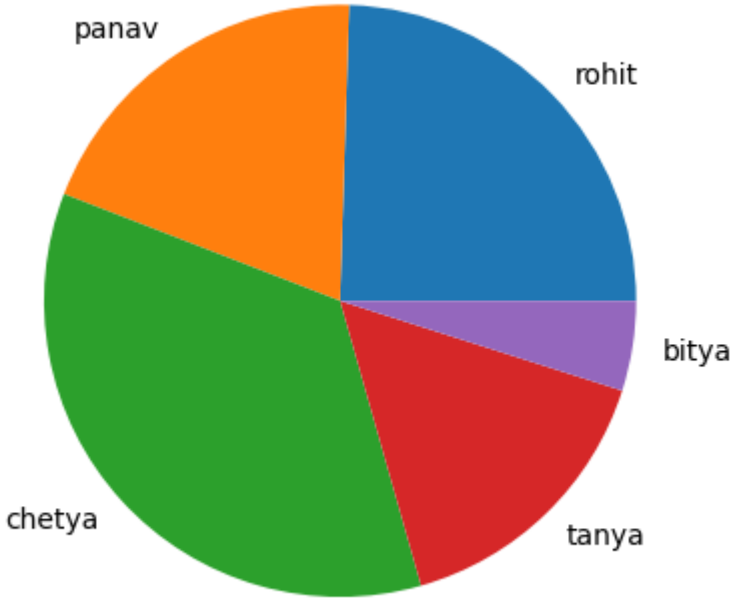
In [265…
```
plt.pie(x9)
```

Out[265…
```
([<matplotlib.patches.Wedge at 0x2084fc06710>,
  <matplotlib.patches.Wedge at 0x2084fc07050>,
  <matplotlib.patches.Wedge at 0x2084fc07e50>,
  <matplotlib.patches.Wedge at 0x20800004ed0>,
  <matplotlib.patches.Wedge at 0x20800005ed0>],
 [Text(0.7897031146679996, 0.7657473412840952, ''),
  Text(-0.607601481198508, 0.9169626164928313, ''),
  Text(-0.8129097722102662, -0.7410652483047991, ''),
  Text(0.7657474152214336, -0.7897030429735556, ''),
  Text(1.0869820386106088, -0.16873069589711656, '')])
```



In [268…
```
mlabels=["rohit","panav","chetya","tanya","bitya"]
plt.pie(x9,labels=mlabels)
```

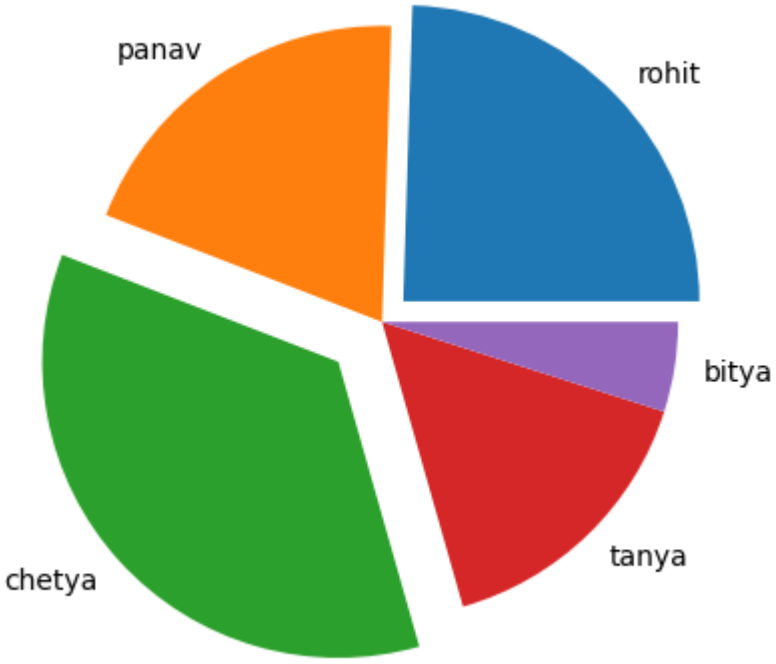
Out[268…
```
([<matplotlib.patches.Wedge at 0x2080013d910>,
  <matplotlib.patches.Wedge at 0x2080013e110>,
  <matplotlib.patches.Wedge at 0x2080013f1d0>,
  <matplotlib.patches.Wedge at 0x208001542d0>,
  <matplotlib.patches.Wedge at 0x20800155210>],
 [Text(0.7897031146679996, 0.7657473412840952, 'rohit'),
  Text(-0.607601481198508, 0.9169626164928313, 'panav'),
  Text(-0.8129097722102662, -0.7410652483047991, 'chetya'),
  Text(0.7657474152214336, -0.7897030429735556, 'tanya'),
  Text(1.0869820386106088, -0.16873069589711656, 'bitya')])
```



**** Explode****

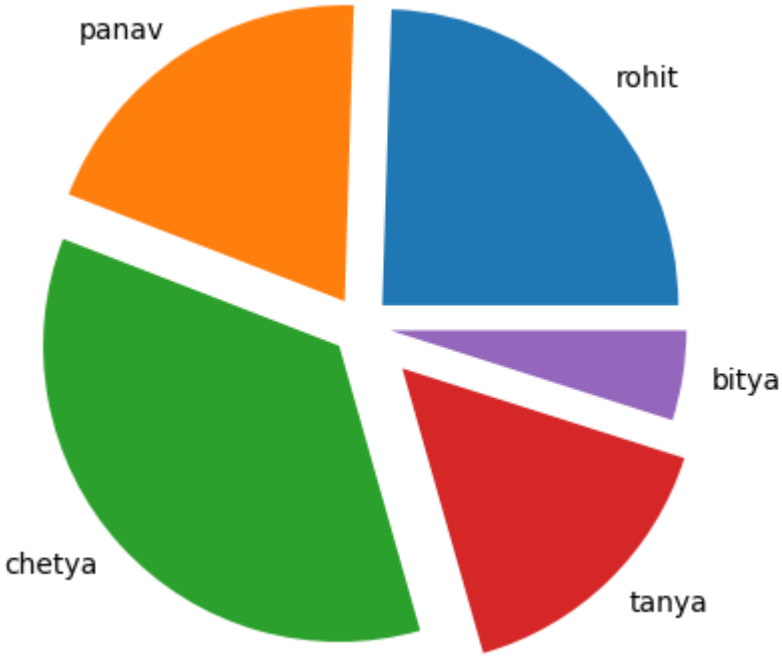
In [283…
```
e2=[0.1,0,0.2,0,0 ]
plt.pie(x9,labels=mlabels,explode=e2)
```

Out[283…
```
([<matplotlib.patches.Wedge at 0x2080151bc90>,
  <matplotlib.patches.Wedge at 0x20801524b10>,
  <matplotlib.patches.Wedge at 0x20801525f10>,
  <matplotlib.patches.Wedge at 0x20801527390>,
  <matplotlib.patches.Wedge at 0x20801534610>],
 [Text(0.861494306910545, 0.8353607359462856, 'rohit'),
  Text(-0.607601481198508, 0.9169626164928313, 'panav'),
  Text(-0.9607115489757692, -0.8758043843602171, 'chetya'),
  Text(0.7657474152214336, -0.7897030429735556, 'tanya'),
  Text(1.0869820386106088, -0.16873069589711656, 'bitya')])
```



In [284…
```
e1=[0.1,0.1 ,0.1 ,0.2,0.1 ]
plt.pie(x9,labels=mlabels,explode=e1)
```

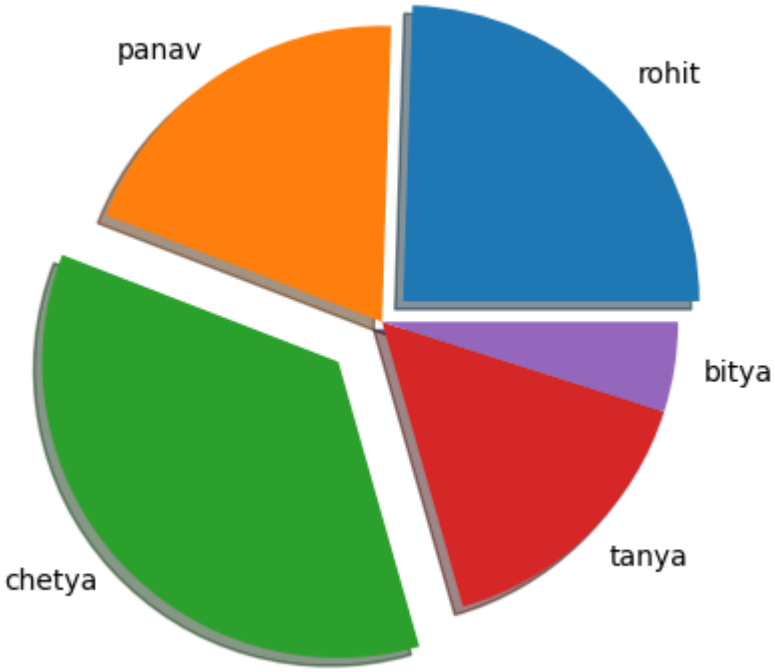
Out[284…
```
([<matplotlib.patches.Wedge at 0x20801578910>,
  <matplotlib.patches.Wedge at 0x208015793d0>,
  <matplotlib.patches.Wedge at 0x2080157a450>,
  <matplotlib.patches.Wedge at 0x2080157b710>,
  <matplotlib.patches.Wedge at 0x20801584a90>],
 [Text(0.861494306910545, 0.8353607359462856, 'rohit'),
  Text(-0.6628379794892815, 1.000322854355816, 'panav'),
  Text(-0.8868106605930176, -0.8084348163325081, 'chetya'),
  Text(0.904974217988967, -0.933285414423293, 'tanya'),
  Text(1.1857985875752095, -0.1840698500695817, 'bitya')])
```



****Shadow ****

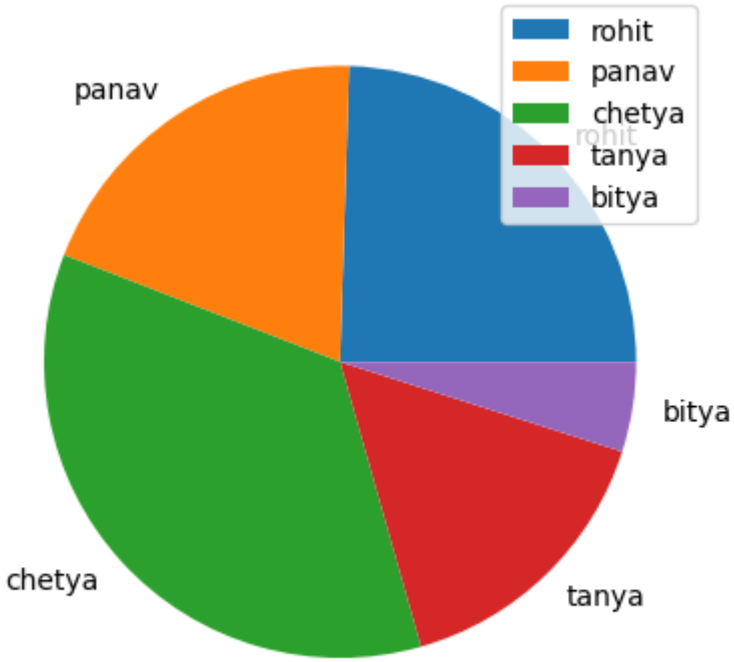
In [285…
```
plt.pie(x9,labels=mlabels,explode=e2,shadow=True)
```

Out[285…
```
([<matplotlib.patches.Wedge at 0x208015c4990>,
  <matplotlib.patches.Wedge at 0x208015c5bd0>,
  <matplotlib.patches.Wedge at 0x208015c7890>,
  <matplotlib.patches.Wedge at 0x208015d1710>,
  <matplotlib.patches.Wedge at 0x208015d3350>],
 [Text(0.861494306910545, 0.8353607359462856, 'rohit'),
  Text(-0.607601481198508, 0.9169626164928313, 'panav'),
  Text(-0.9607115489757692, -0.8758043843602171, 'chetya'),
  Text(0.7657474152214336, -0.7897030429735556, 'tanya'),
  Text(1.0869820386106088, -0.16873069589711656, 'bitya')])
```

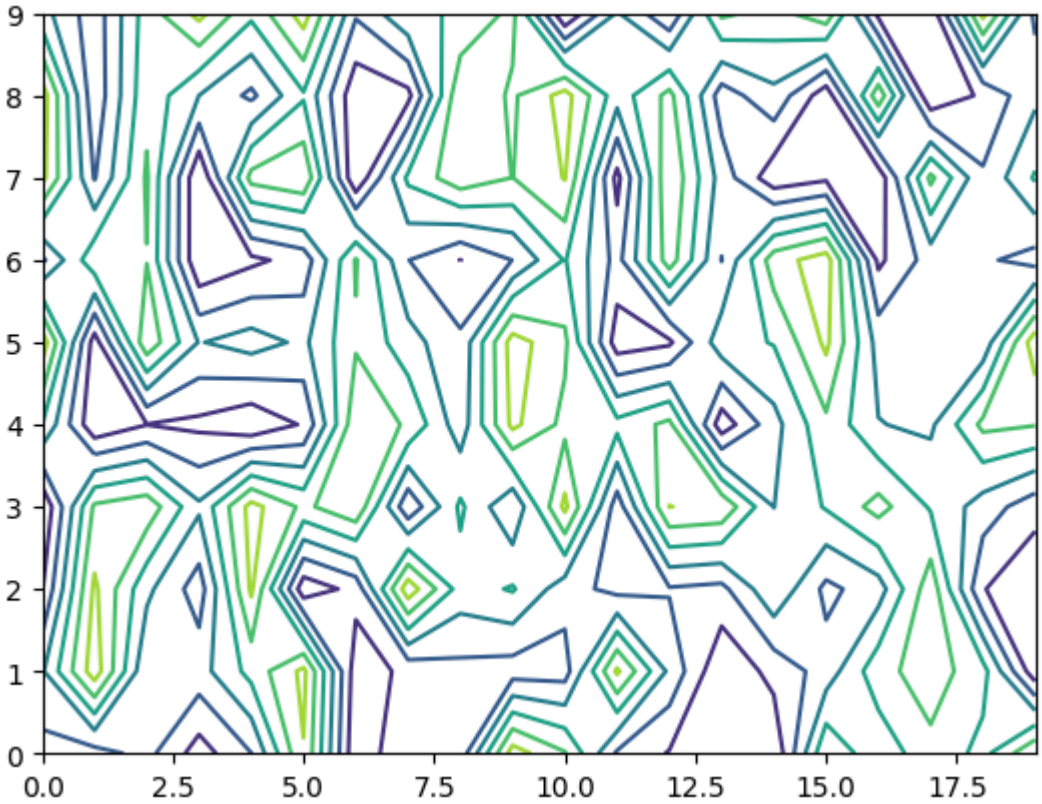


In [315…
```
plt.pie(x9,labels=mlabels)
plt.legend(loc=1, fontsize='medium',)
```

Out[315…
```
<matplotlib.legend.Legend at 0x20802d1fe10>
```



In [14]:
```
data3 = np.random.rand(10,20)
cp = plt.contour(data3)

plt.show()
```



In [ ]: