

MatPlotLib

March 26, 2021

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

```
In [2]: %matplotlib inline #for jupyter
```

1 Simple way: functional

```
In [4]: x= np.linspace(0,5,11)  
y=x**2
```

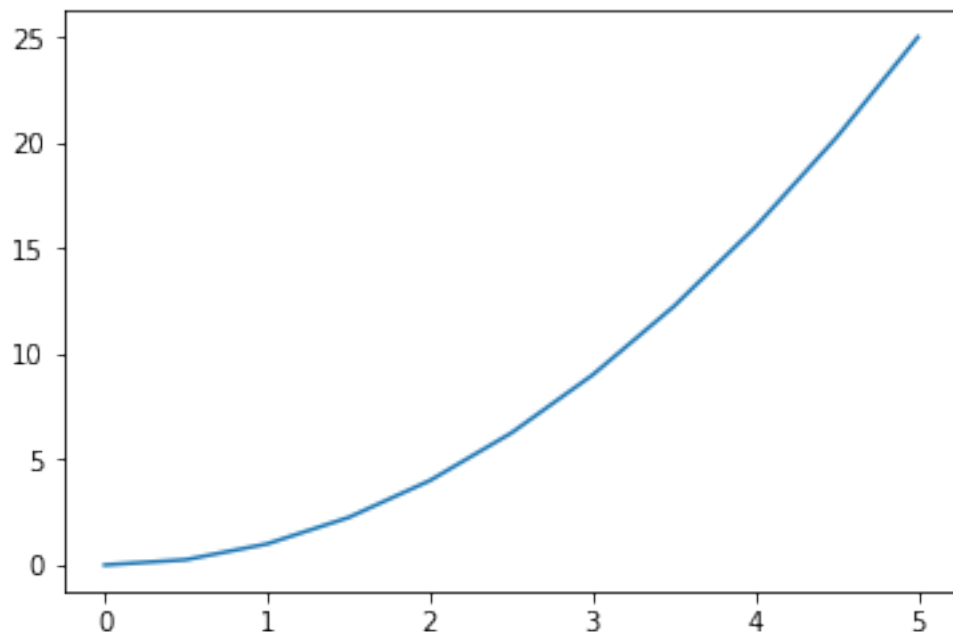
```
In [5]: x
```

```
Out[5]: array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

```
In [6]: y
```

```
Out[6]: array([ 0. ,  0.25,  1. ,  2.25,  4. ,  6.25,  9. , 12.25, 16. ,  
                20.25, 25. ])
```

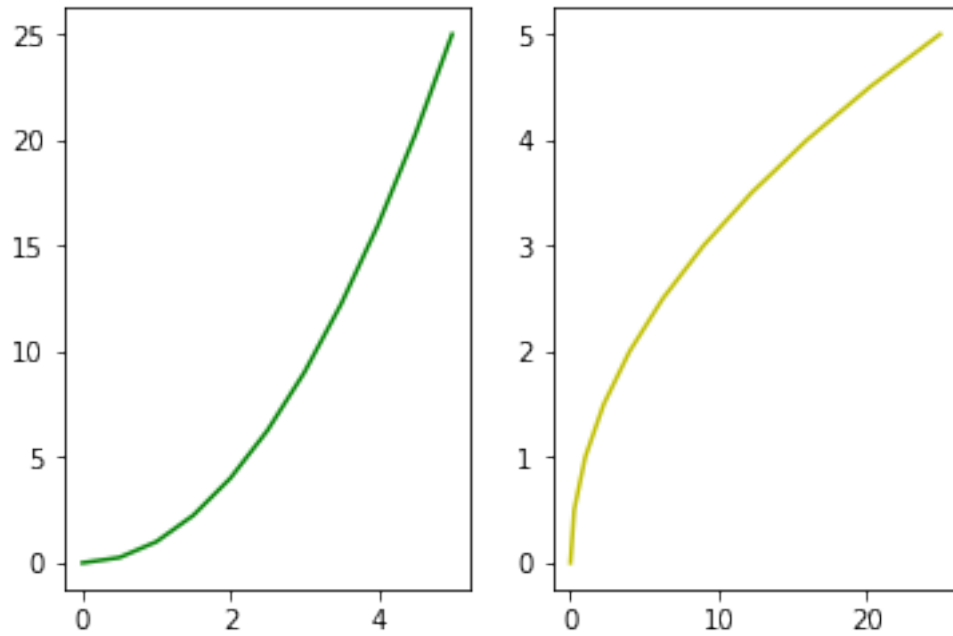
```
In [8]: plt.plot(x,y)  
plt.show()
```



```
In [12]: plt.plot(x,y, 'r')
plt.xlabel('My X')
plt.ylabel('My Y')
plt.title('My first plt')
plt.show()
```



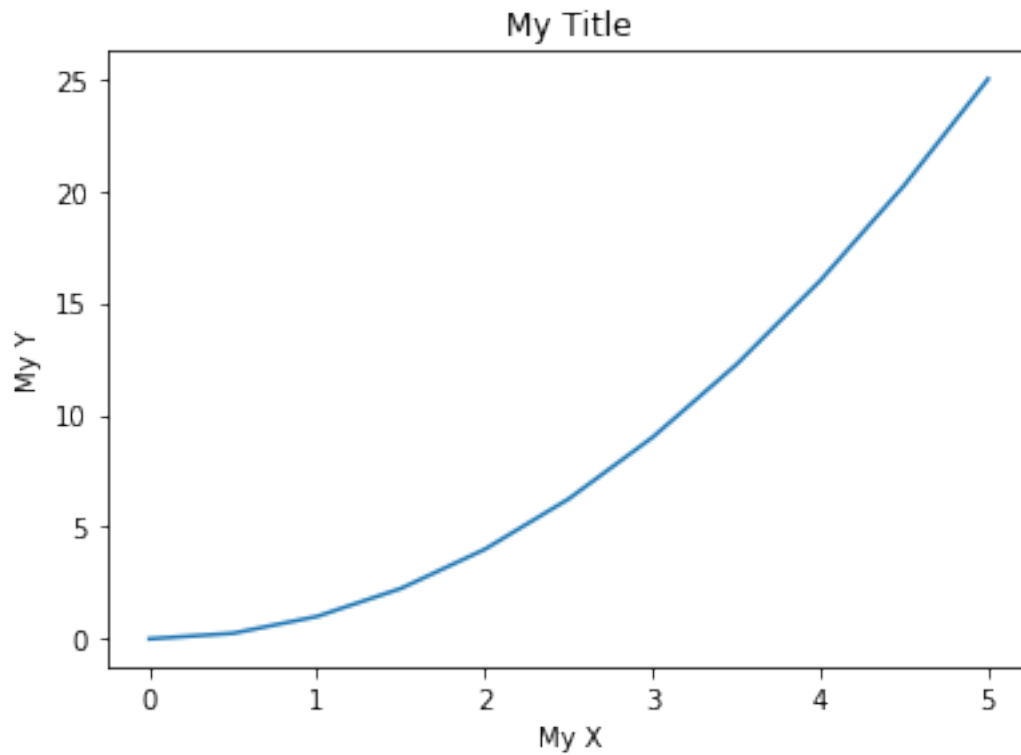
```
In [19]: plt.subplot(1,2,1) #nb of row,nb of columns, plot index/place(left=1)
plt.plot(x,y, 'g')
plt.subplot(1,2,2) #nb of row,nb of columns, plot index/place(left+1=2)
plt.plot(y,x, 'y')
plt.show()
```



2 Better way: object oriented plotting

```
In [34]: fig = plt.figure()
axes=fig.add_axes([0.1,0.1,0.8,0.8])
#for add_axes: x max, y max, xwidth, height on [0,1] each,don't need to be understood
axes.plot(x,y)
axes.set_xlabel('My X')
axes.set_ylabel('My Y')
axes.set_title('My Title')
```

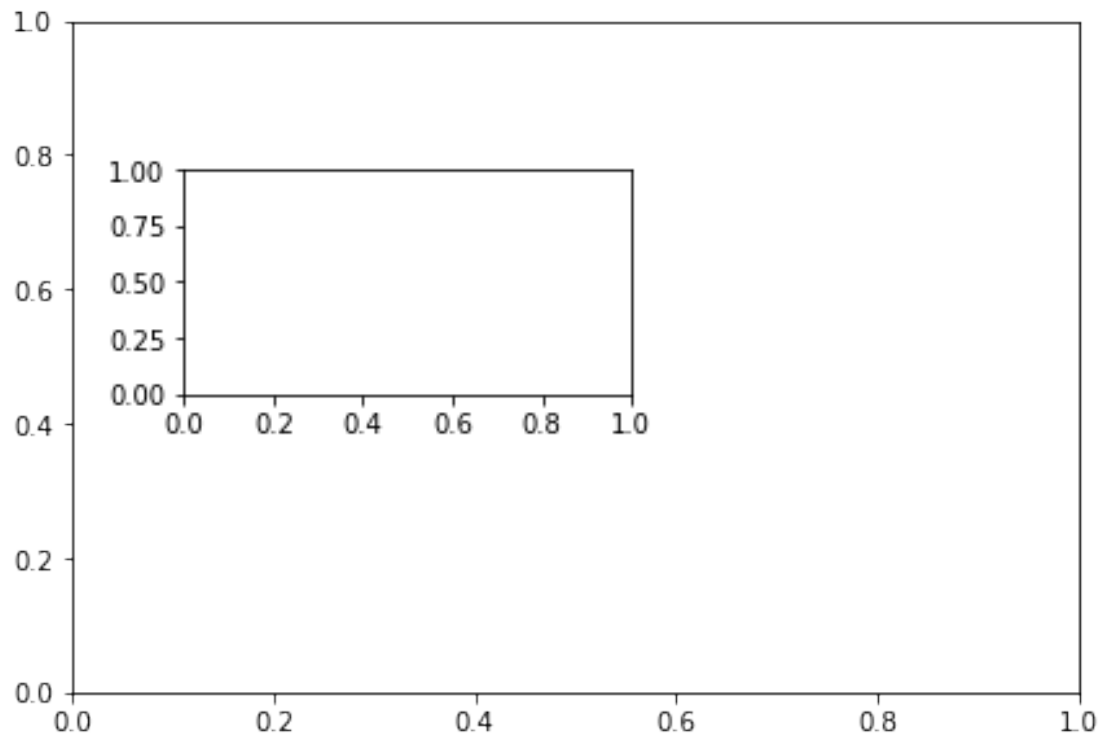
```
Out[34]: Text(0.5,1,'My Title')
```



In [32]: `axes.plot(x,y)` *#won't work if you put this line in a seperate cell, so do as in [34]*

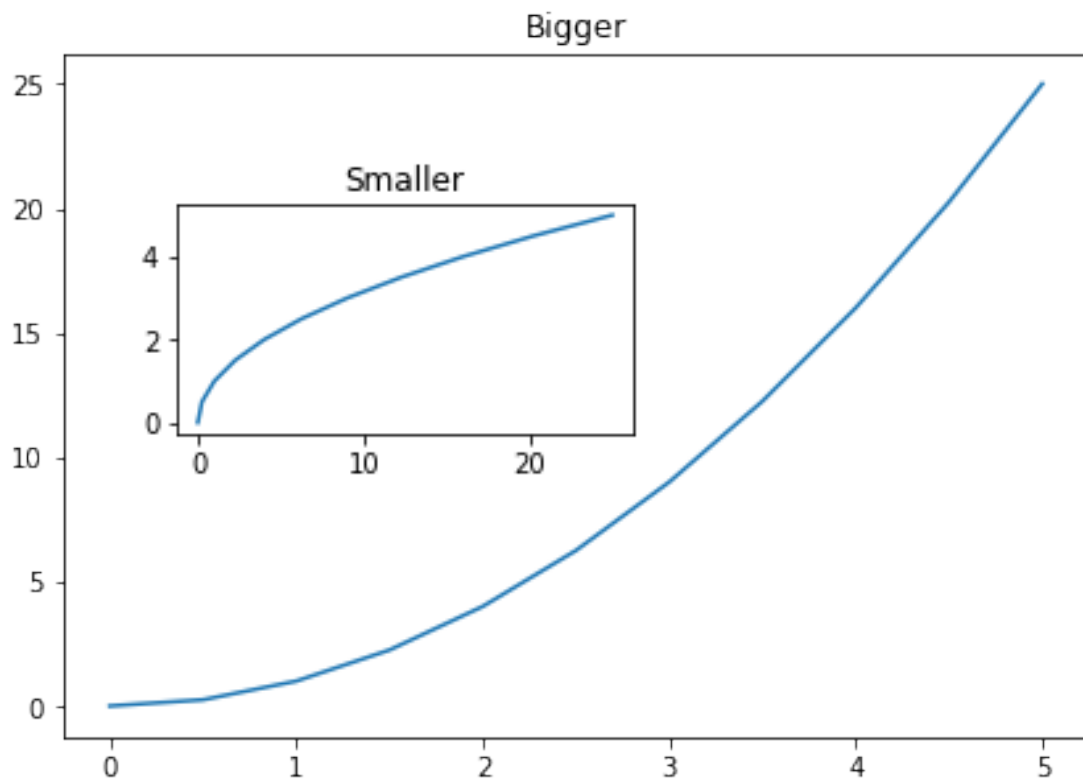
Out[32]: [`<matplotlib.lines.Line2D at 0x7f2e8b697908>`]

In [48]: `fig = plt.figure()`
`axes1=fig.add_axes([0.1,0.1,0.9,0.9])`
`axes2=fig.add_axes([0.2,0.5,0.4,0.3])`
#second plot starting point(left bottom corner) starts
#20% from the left, 50% from the bottom,40% of total width,30% of total height



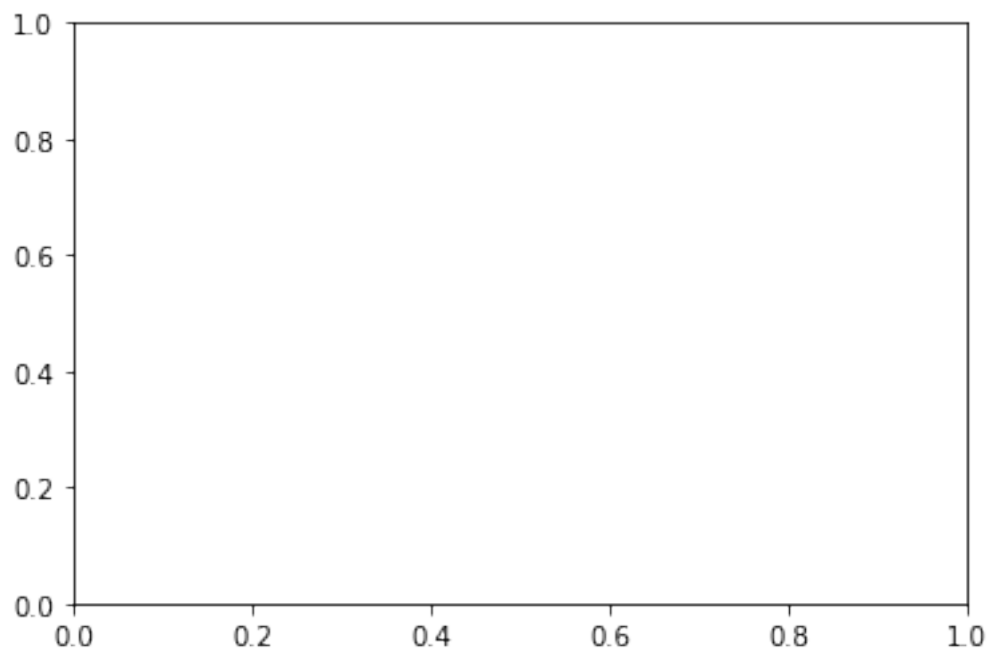
```
In [53]: fig = plt.figure()
axes1=fig.add_axes([0.1,0.1,0.9,0.9])
axes1.plot(x,y)
axes1.set_title('Bigger')
axes2=fig.add_axes([0.2,0.5,0.4,0.3])
axes2.plot(y,x)
axes2.set_title('Smaller')
```

```
Out[53]: Text(0.5,1,'Smaller')
```

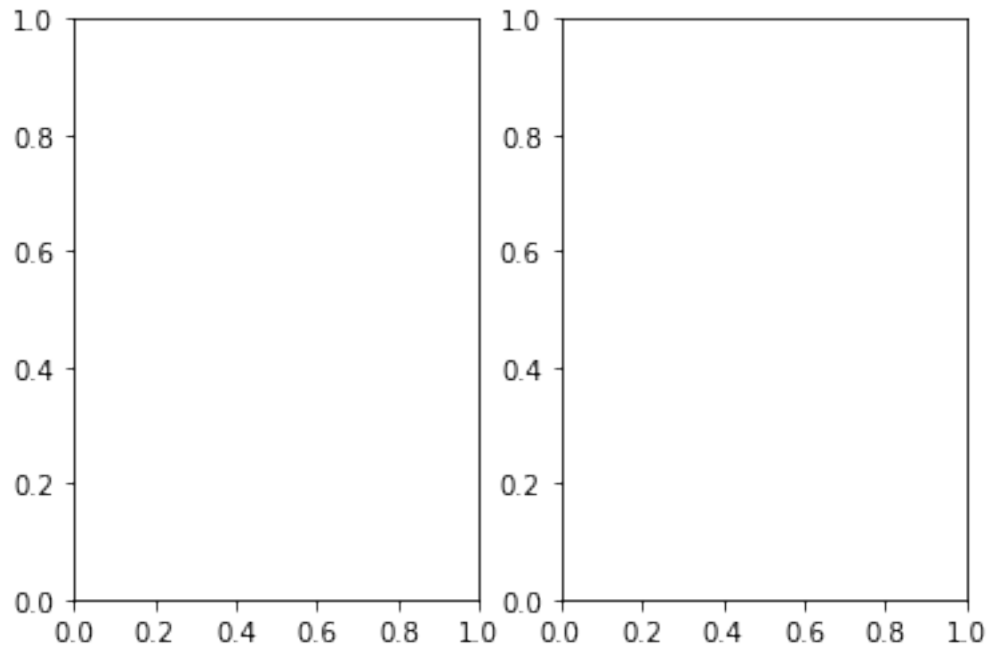


2.1 Subplots in object oriented plotting (càd 'figure')

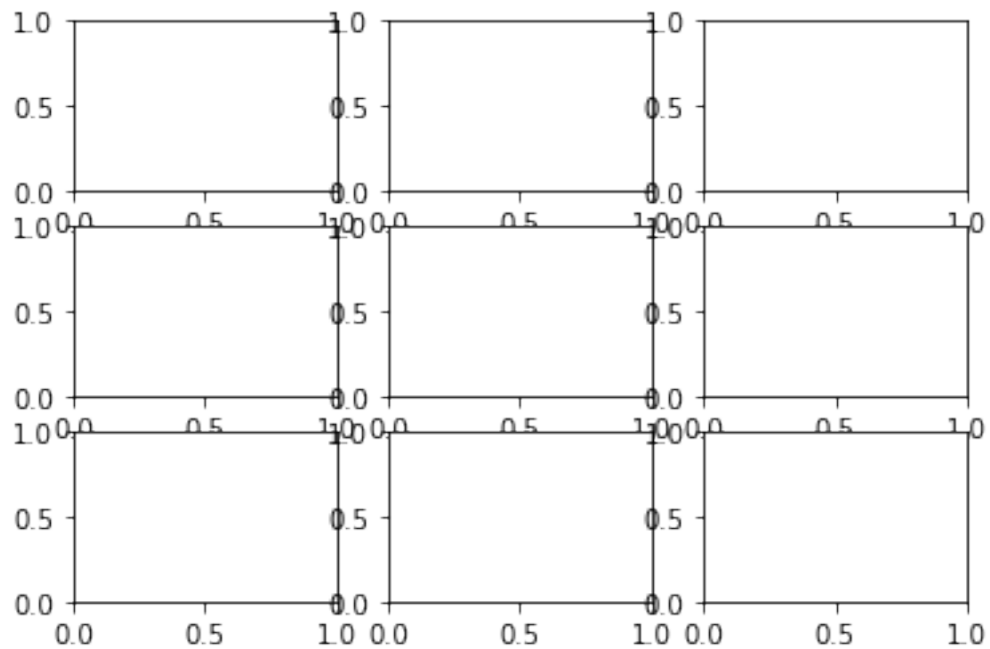
In [56]: `fig1,axes=plt.subplots()`



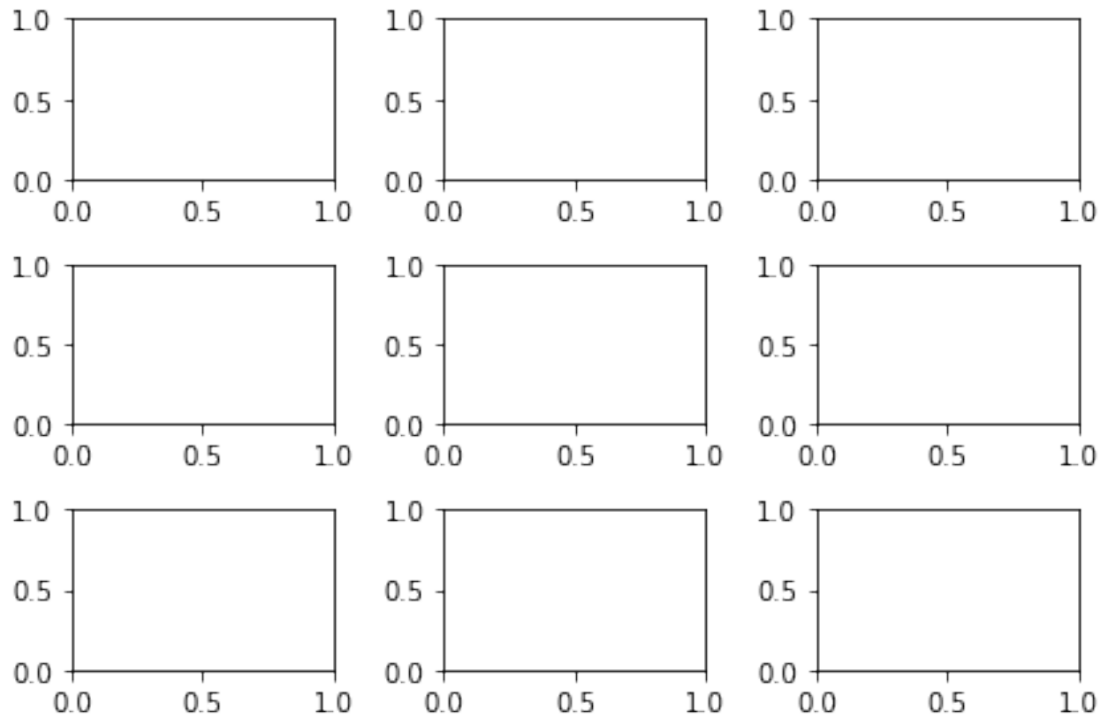
```
In [57]: fig1, axes=plt.subplots(nrows=1, ncols=2) #when you add nrows/ncols then add_axes is m
```



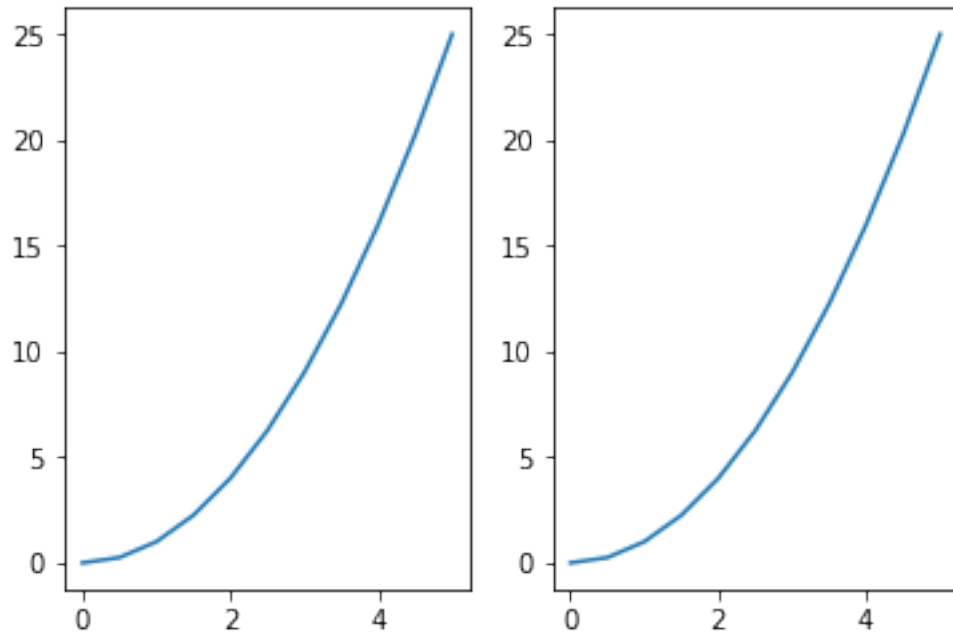
```
In [58]: fig1, axes=plt.subplots(nrows=3, ncols=3)
```



```
In [59]: fig1, axes=plt.subplots(nrows=3, ncols=3)
plt.tight_layout() #stop overlapping
```

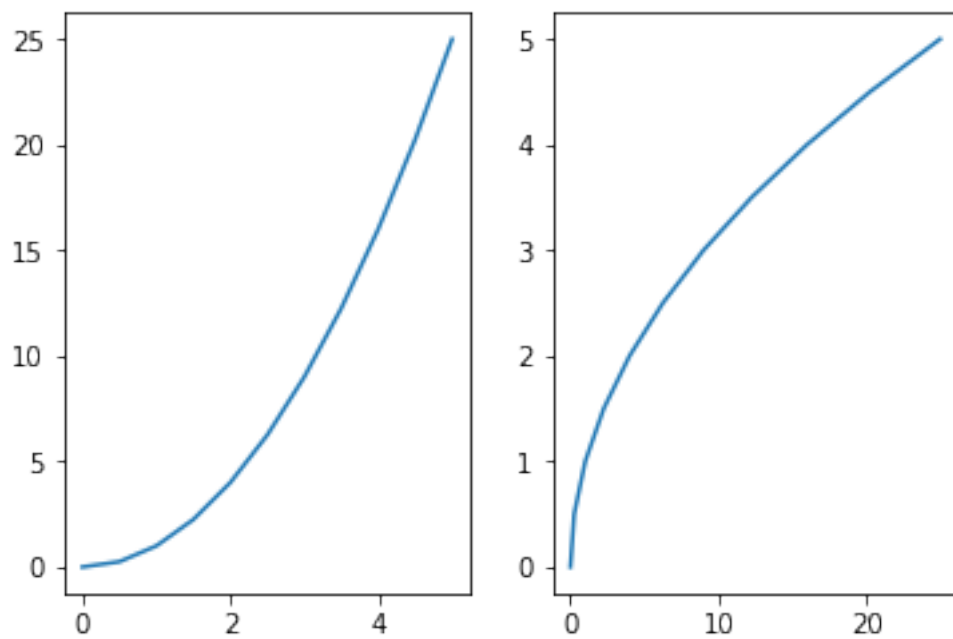


```
In [60]: fig1, axes=plt.subplots(nrows=1, ncols=2)
#axes is a actually a list, so you can do 'for', access by index, apply methods, ect
for my_curr_axe in axes:
    my_curr_axe.plot(x,y)
```

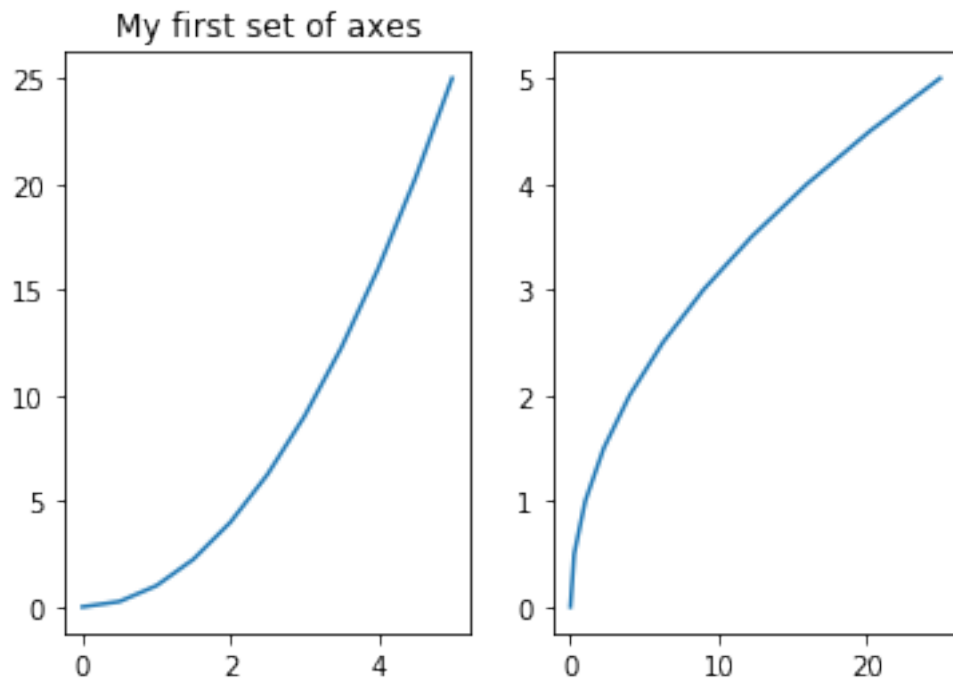
```
In [66]: fig1, axes=plt.subplots(nrows=1, ncols=2)
         #axes is a actually a list, so you can do 'for', access by index, apply methods, ect
         axes[0].plot(x,y)
         axes[1].plot(y,x)
```

```
Out [66]: [<matplotlib.lines.Line2D at 0x7f2e8a83f4e0>]
```



```
In [68]: fig1, axes=plt.subplots(nrows=1, ncols=2)
         #axes is a actually a list, so you can do 'for', access by index, apply methods, ect
         axes[0].plot(x,y)
         axes[0].set_title('My first set of axes')
         axes[1].plot(y,x)
```

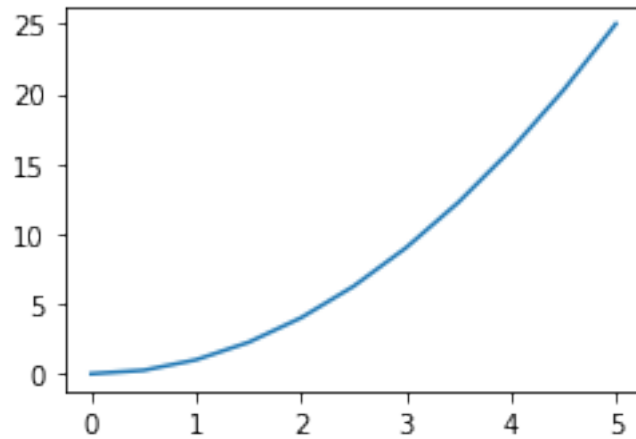
```
Out[68]: [<matplotlib.lines.Line2D at 0x7f2e8b44c4e0>]
```



2.2 Figure size

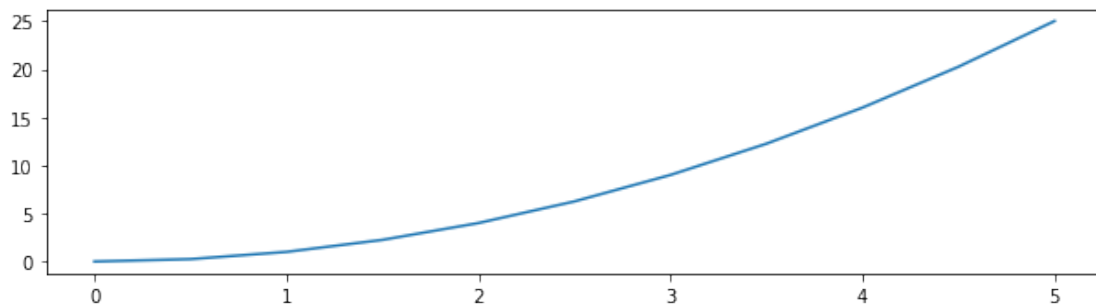
```
In [70]: fig = plt.figure(figsize=(3,2))
         ax=fig.add_axes([0,0,1,1])
         ax.plot(x,y)
```

```
Out[70]: [<matplotlib.lines.Line2D at 0x7f2e8ac73f60>]
```



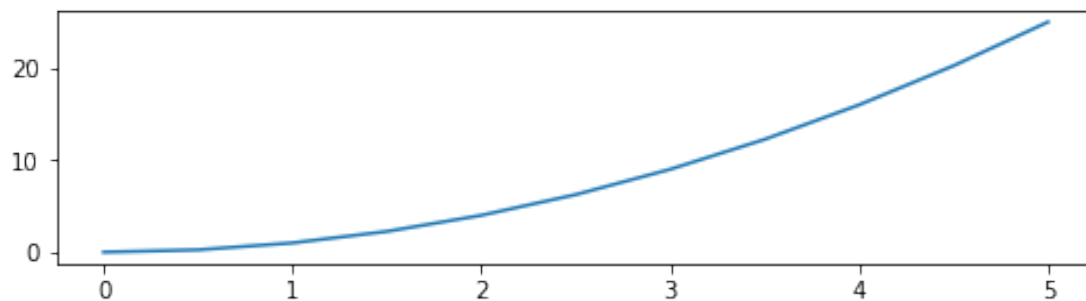
```
In [71]: fig = plt.figure(figsize=(8,2))  
         ax=fig.add_axes([0,0,1,1])  
         ax.plot(x,y)
```

```
Out [71]: [<matplotlib.lines.Line2D at 0x7f2e8ad904e0>]
```



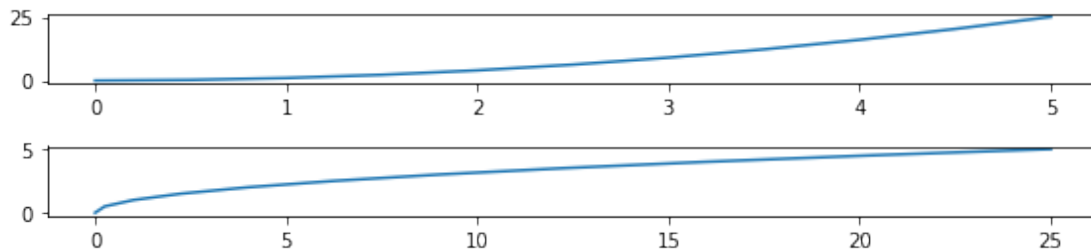
```
In [73]: fig,axes = plt.subplots(figsize=(8,2))  
         axes.plot(x,y)
```

```
Out [73]: [<matplotlib.lines.Line2D at 0x7f2e8b5c4860>]
```



```
In [77]: fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(8, 2))
```

```
axes[0].plot(x, y)
axes[1].plot(y, x)
plt.tight_layout()
```



2.3 Save a figure

```
In [78]: fig.savefig('my_pic.png') #jpg, jpeg
```

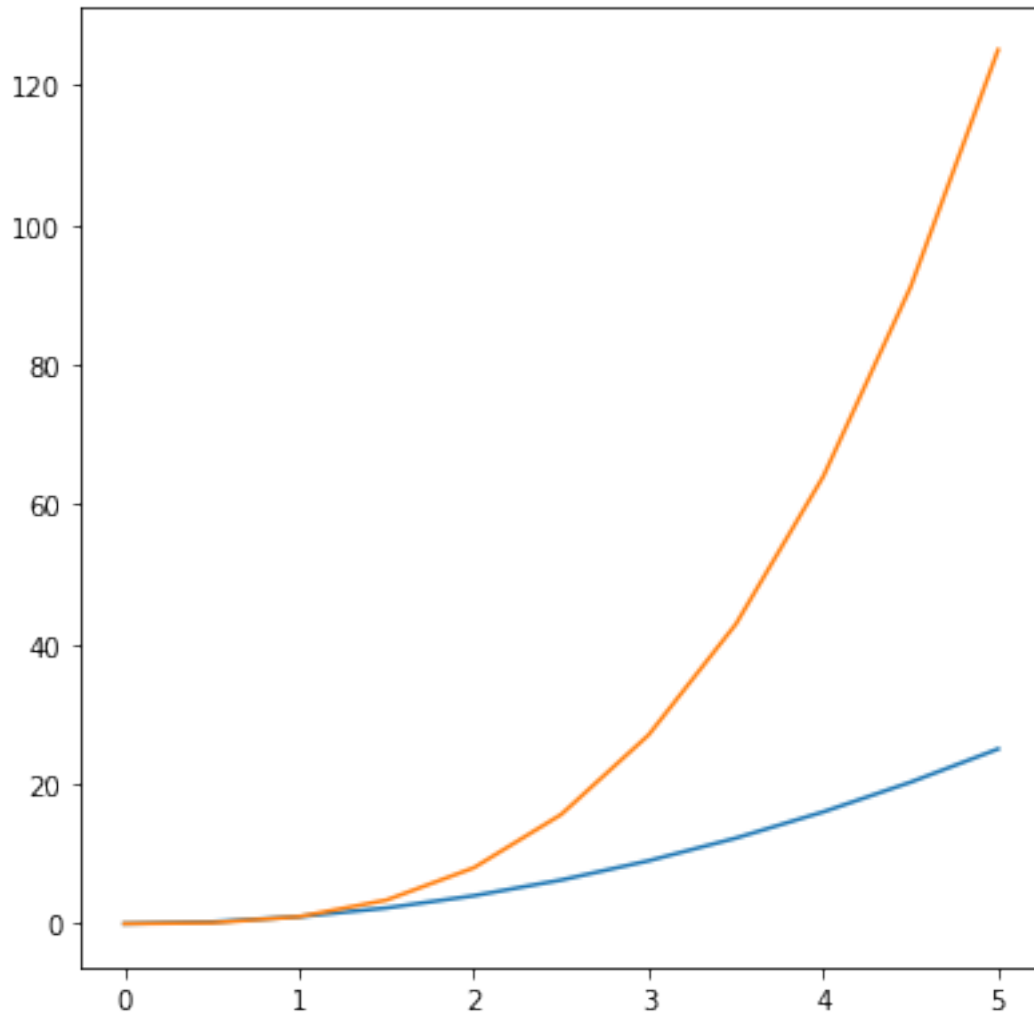
if you want better resolution, specify DPI (dots per inch) as we have `dpi=100` by default:

```
In [80]: fig.savefig('my_pic.png', dpi=200) #jpg, jpeg
```

2.4 Several graphs on the same plot + legends

```
In [84]: fig = plt.figure(figsize=(5, 5))
ax=fig.add_axes([0, 0, 1, 1])
ax.plot(x, x**2)
ax.plot(x, x**3)
```

```
Out[84]: [<matplotlib.lines.Line2D at 0x7f2e8ade50b8>]
```

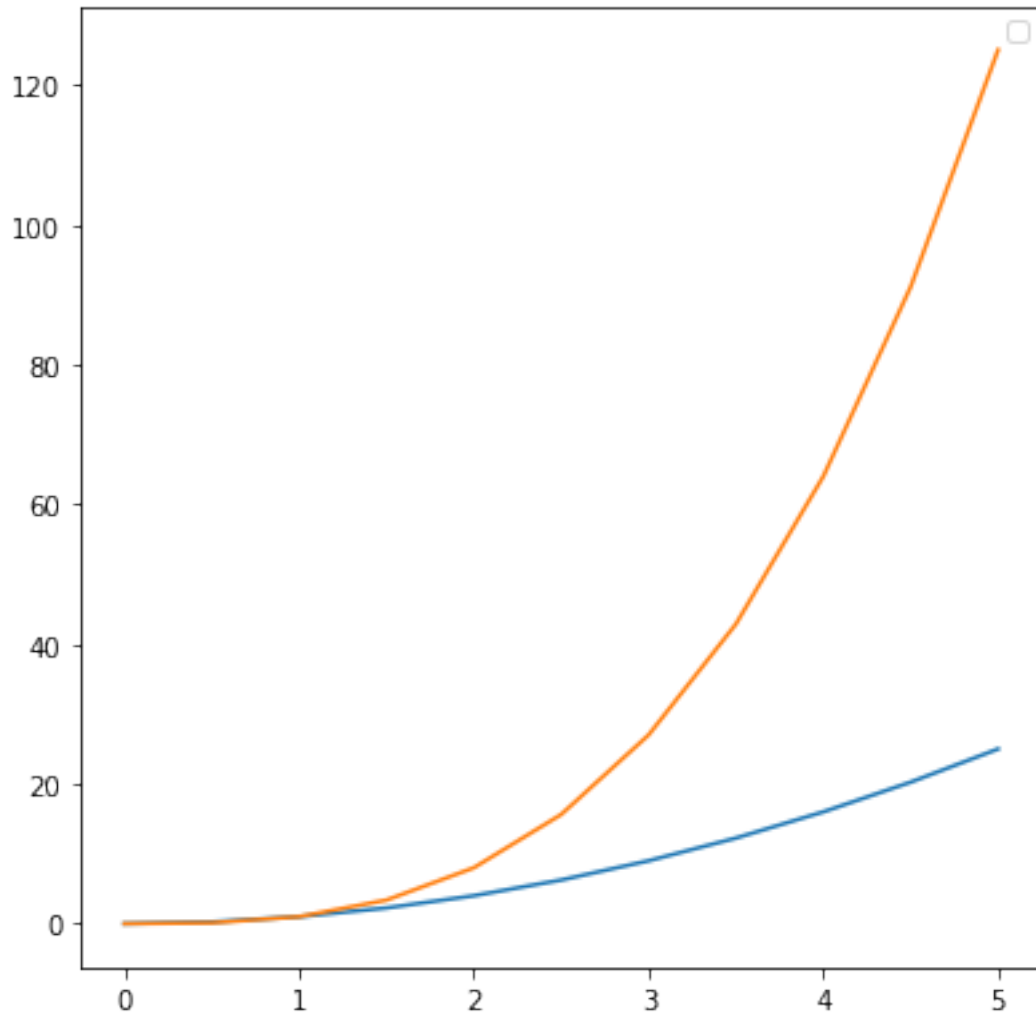


```
In [85]: fig = plt.figure(figsize=(5,5))
         ax=fig.add_axes([0,0,1,1])
         ax.plot(x,x**2)
         ax.plot(x,x**3)

         ax.legend()
```

No handles with labels found to put in legend.

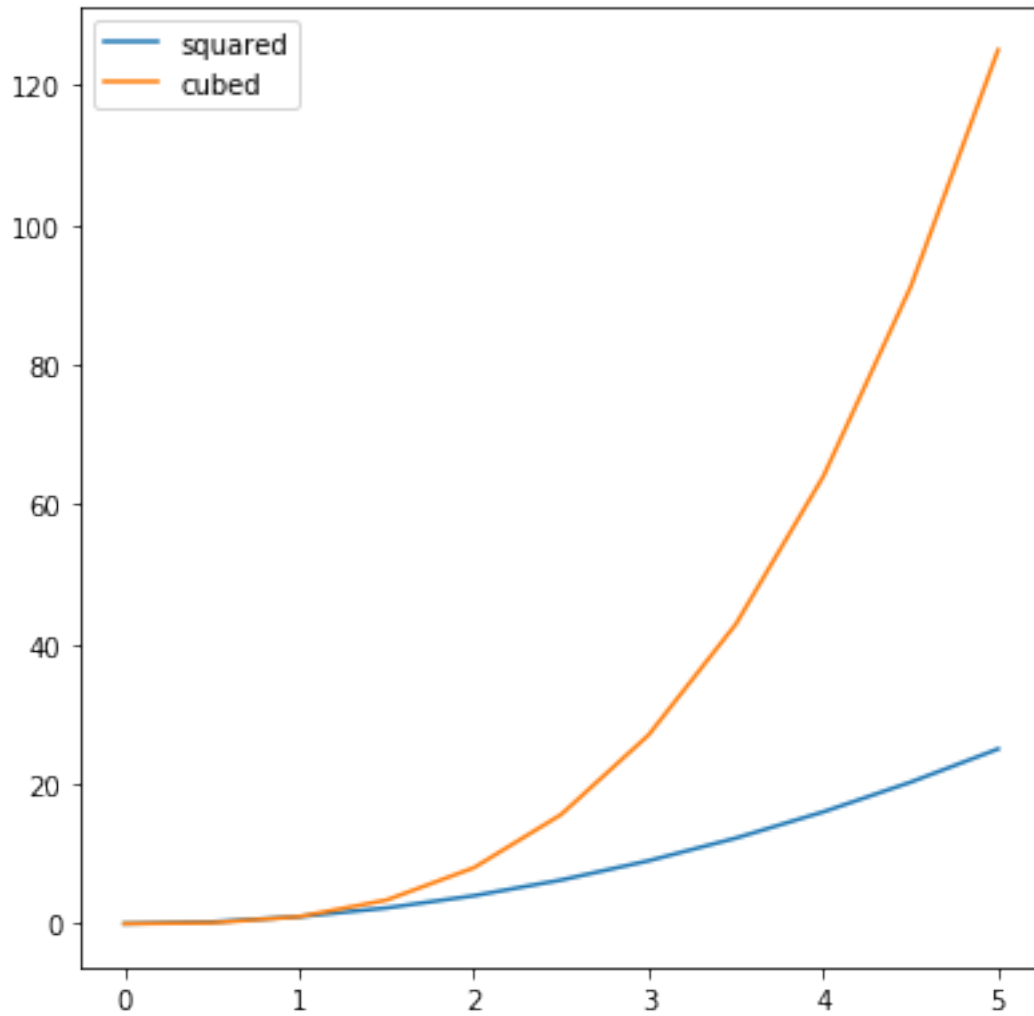
```
Out [85]: <matplotlib.legend.Legend at 0x7f2e8a51a160>
```



```
In [87]: fig = plt.figure(figsize=(5,5))
ax=fig.add_axes([0,0,1,1])
ax.plot(x,x**2, label='squared')
ax.plot(x,x**3, label='cubed')

ax.legend()
```

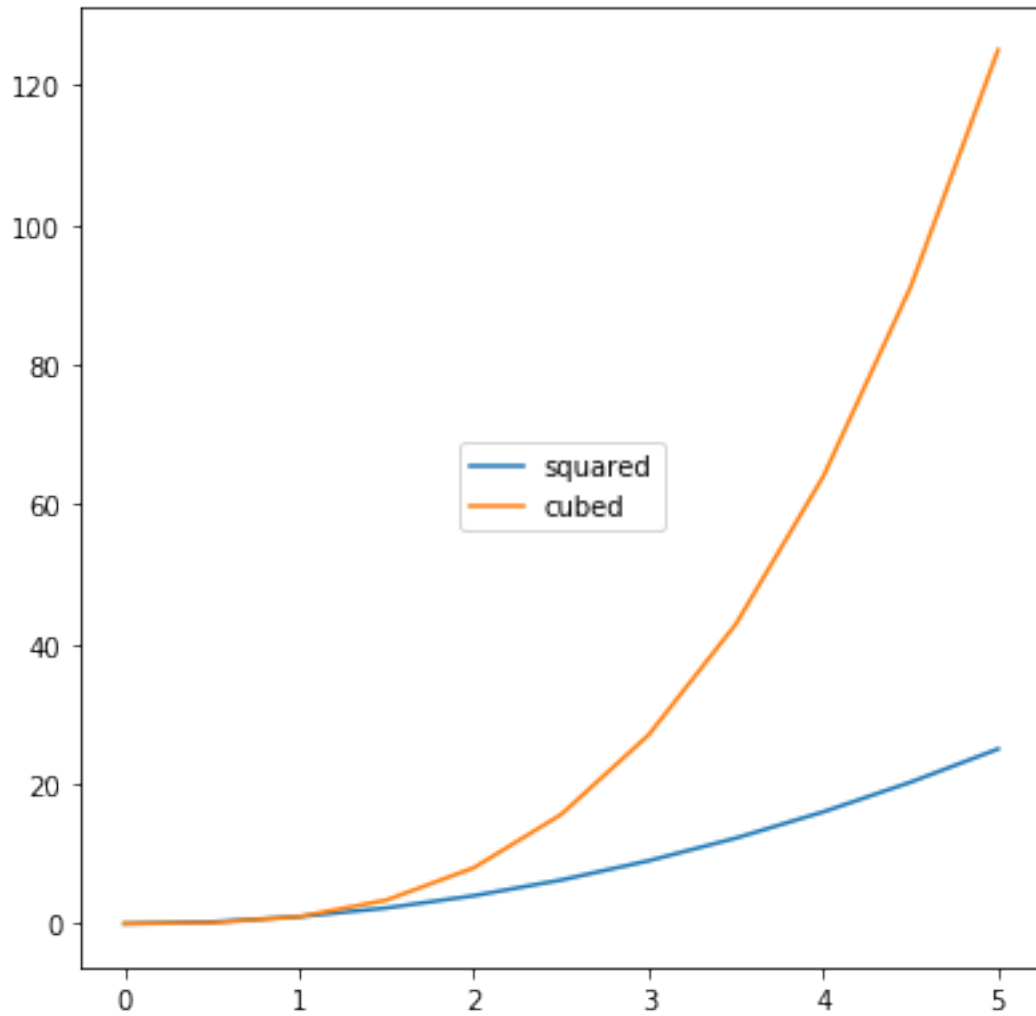
```
Out[87]: <matplotlib.legend.Legend at 0x7f2e8a05af60>
```



```
In [90]: fig = plt.figure(figsize=(5,5))
ax=fig.add_axes([0,0,1,1])
ax.plot(x,x**2, label='squared')
ax.plot(x,x**3, label='cubed')

ax.legend(loc=10) #check google 0=best,1=upper right, 2=upper left, .., 10=center

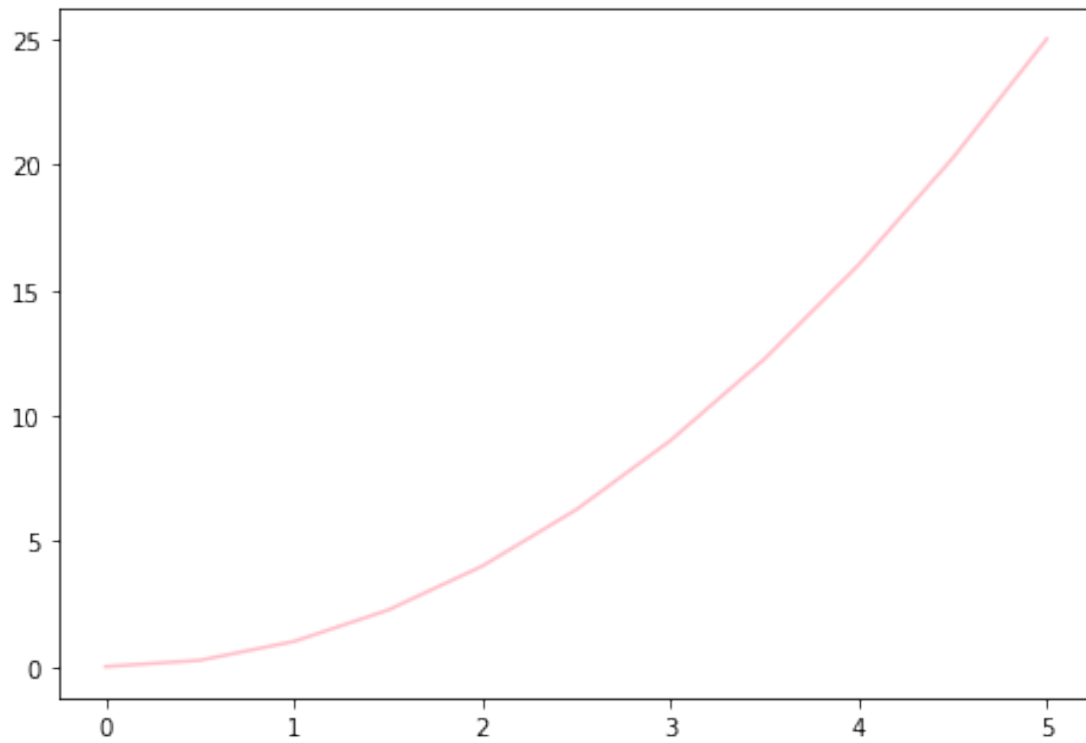
Out[90]: <matplotlib.legend.Legend at 0x7f2e89f2f400>
```



2.5 Style :)

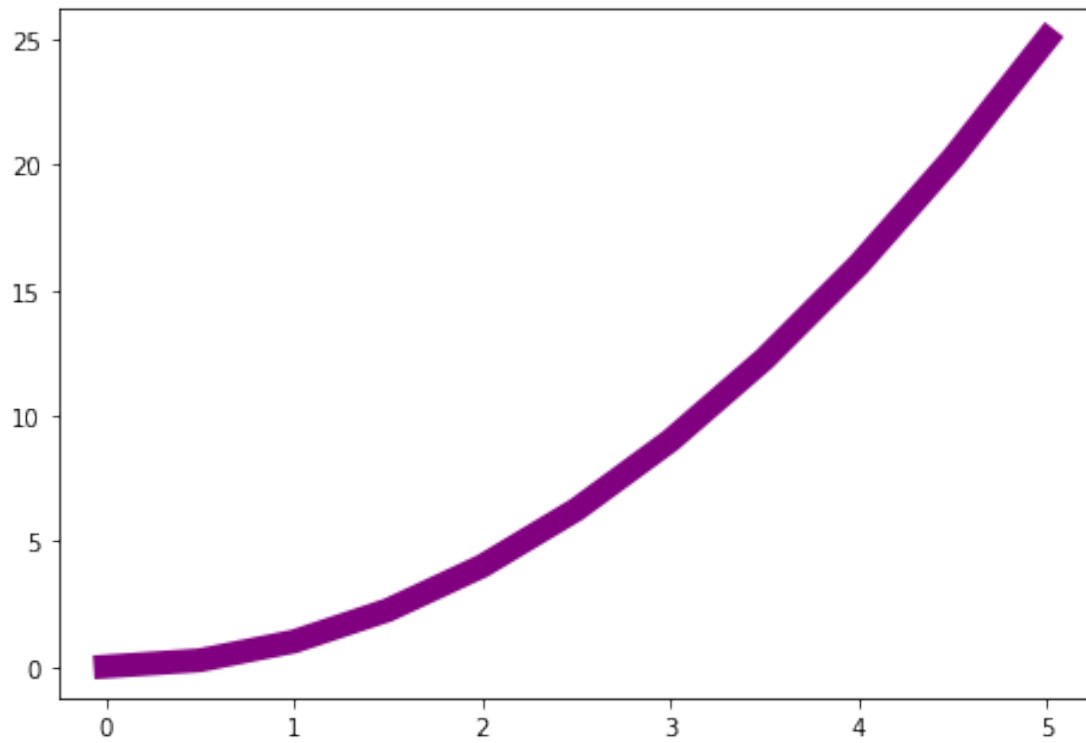
```
In [93]: fig = plt.figure()  
         ax=fig.add_axes([0,0,1,1])  
         ax.plot(x,y,color='purple')
```

```
Out[93]: [<matplotlib.lines.Line2D at 0x7f2e89f6e048>]
```

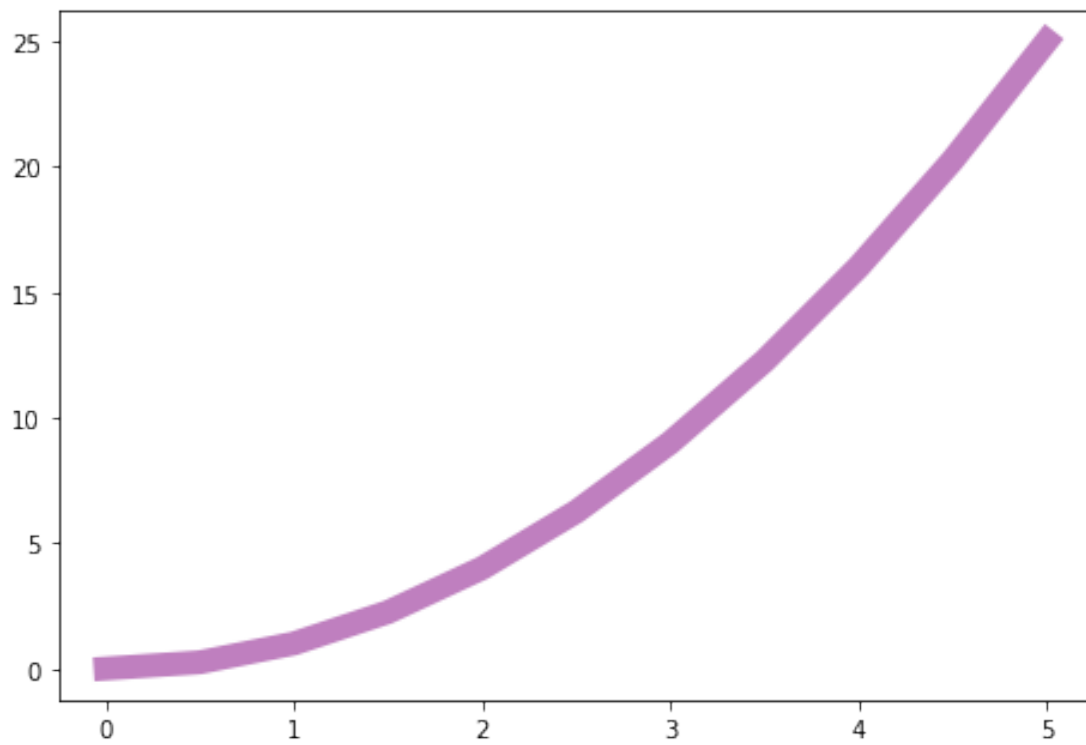
```
In [94]: fig = plt.figure()
         ax=fig.add_axes([0,0,1,1])
         ax.plot(x,y,color='purple',linewidth=10) #or just lw
```

```
Out[94]: [<matplotlib.lines.Line2D at 0x7f2e89df8f28>]
```



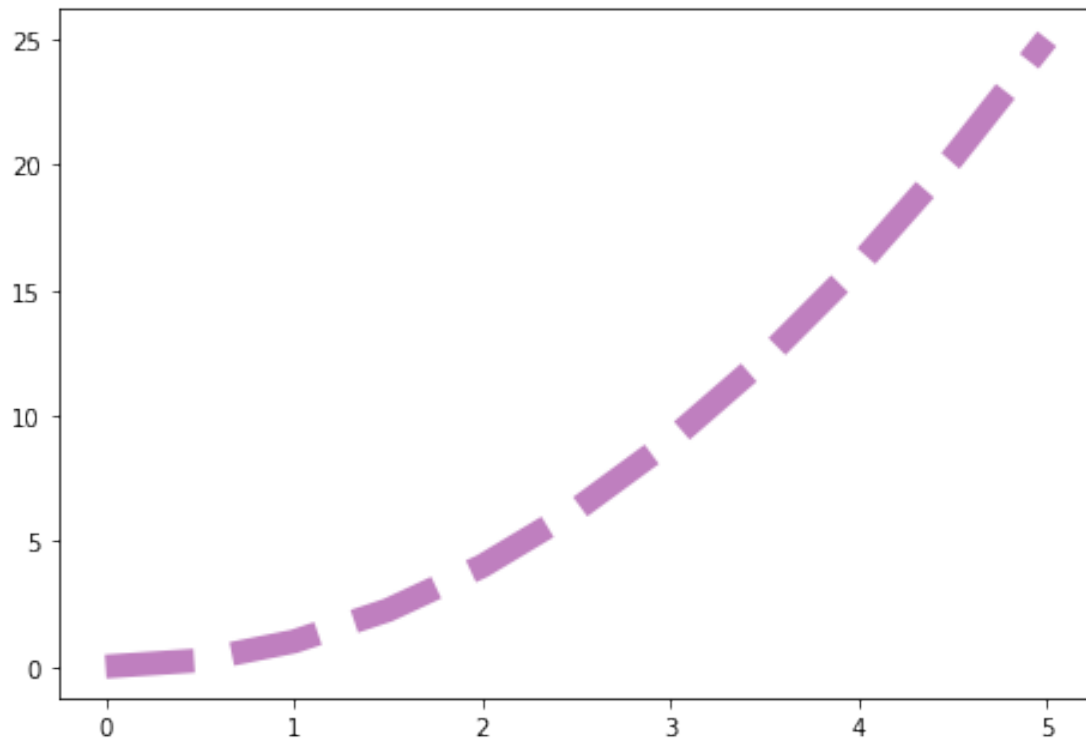
```
In [96]: fig = plt.figure()
         ax=fig.add_axes([0,0,1,1])
         ax.plot(x,y,color='purple',lw=10, alpha=0.5) #transparency
```

```
Out[96]: [<matplotlib.lines.Line2D at 0x7f2e89d4b908>]
```



```
In [98]: fig = plt.figure()
         ax=fig.add_axes([0,0,1,1])
         ax.plot(x,y,color='purple',lw=10, alpha=0.5, linestyle= '--') # or ls= '-.', ':', 'step'
```

```
Out[98]: [<matplotlib.lines.Line2D at 0x7f2e89c835f8>]
```

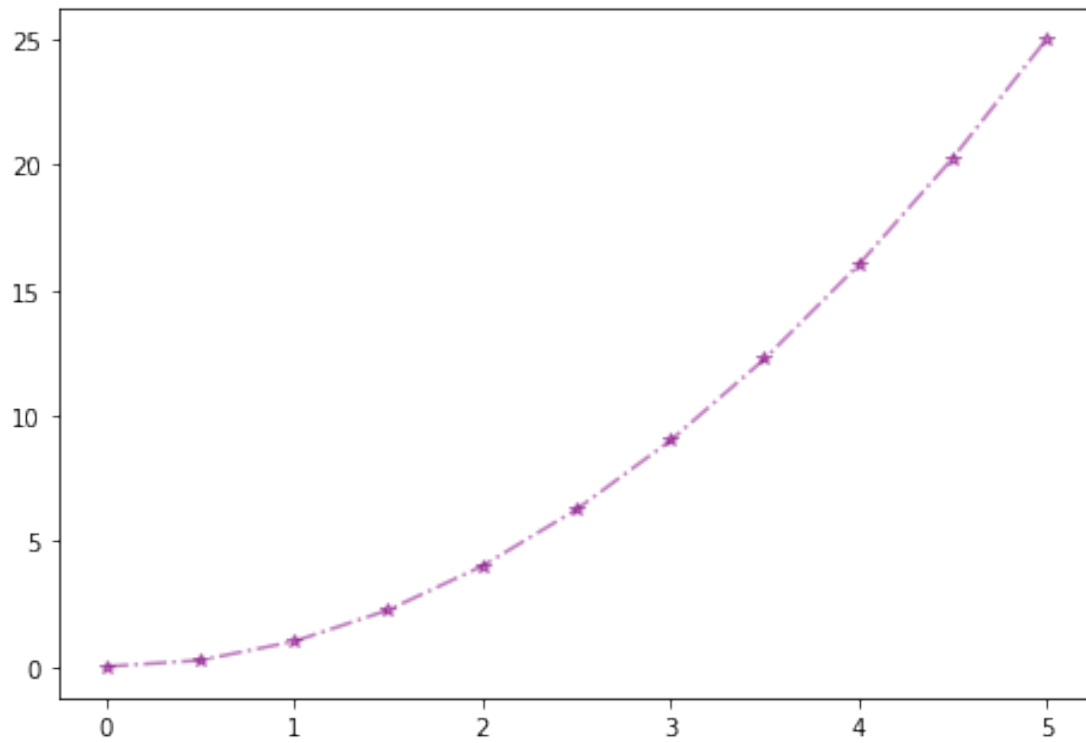


```
In [99]: x
```

```
Out[99]: array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

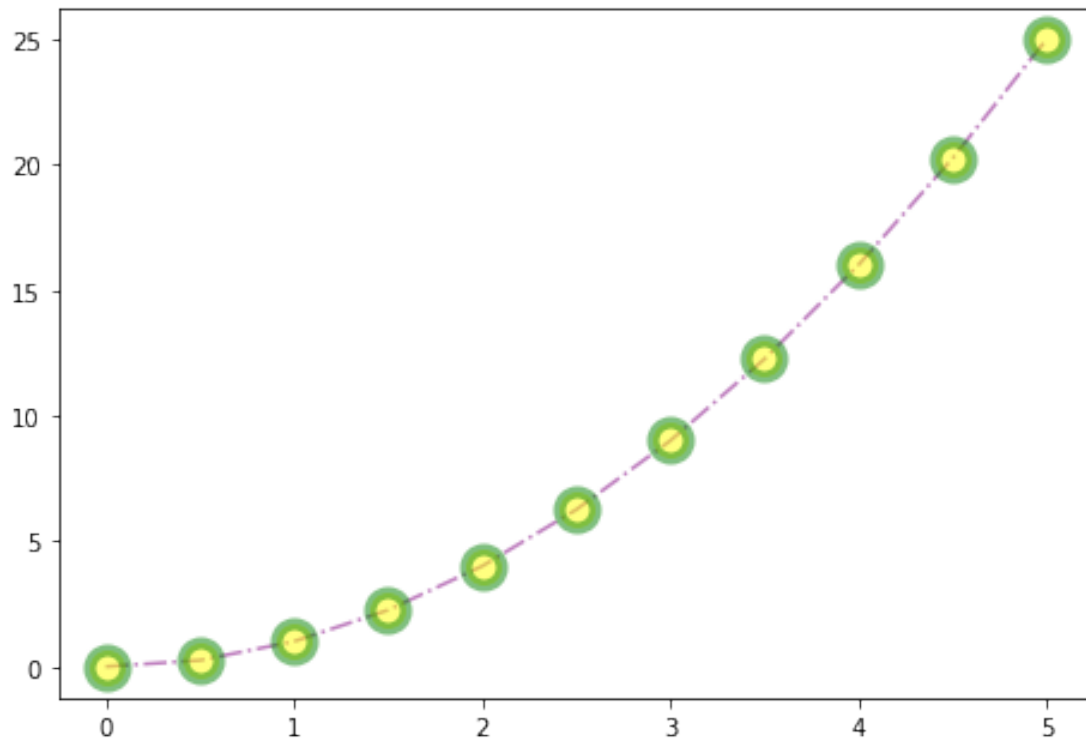
```
In [100]: fig = plt.figure()
          ax=fig.add_axes([0,0,1,1])
          ax.plot(x,y,color='purple',lw=1.5, alpha=0.5, ls= '-.', marker = '*') #or 'o', '+', 'x'
```

```
Out[100]: [<matplotlib.lines.Line2D at 0x7f2e89c6c940>]
```



```
In [103]: fig = plt.figure()
          ax=fig.add_axes([0,0,1,1])
          ax.plot(x,y,color='purple',lw=1.5, alpha=0.5, ls= '-.', marker = 'o', markersize=15)
```

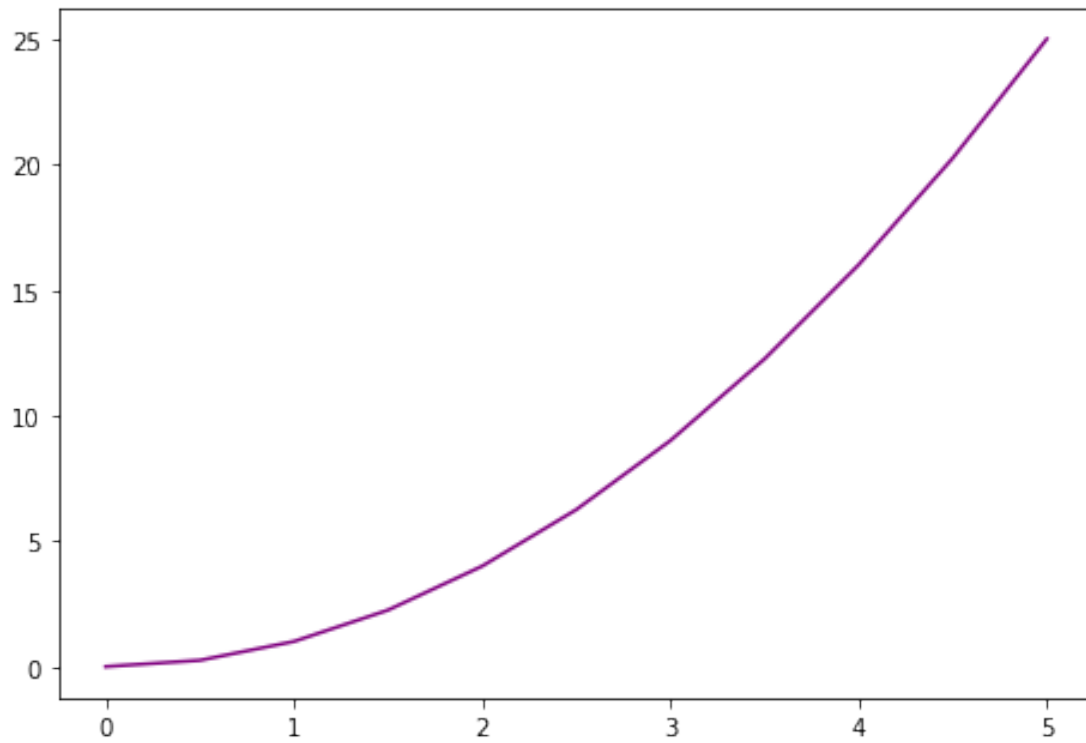
```
Out[103]: [<matplotlib.lines.Line2D at 0x7f2e89b776a0>]
```

2.6 Axes limits

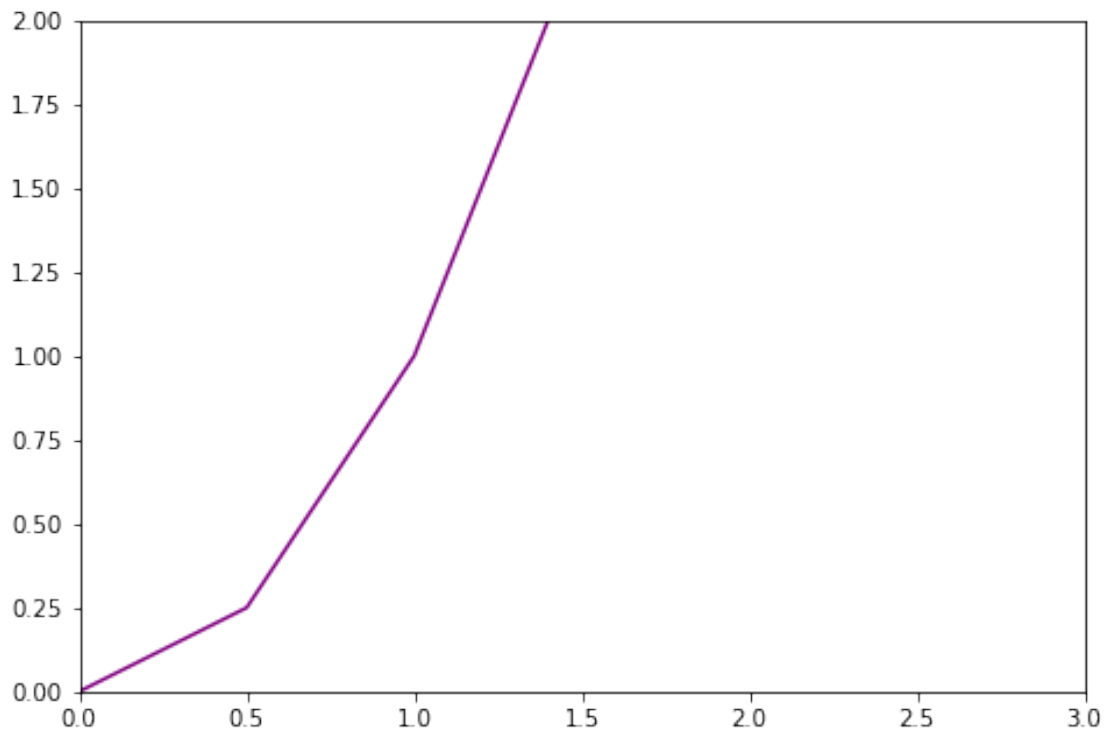
```
In [107]: fig = plt.figure()  
          ax=fig.add_axes([0,0,1,1])  
          ax.plot(x,y,color='purple')
```

```
Out[107]: [<matplotlib.lines.Line2D at 0x7f2e89a52ac8>]
```



```
In [109]: fig = plt.figure()
          ax=fig.add_axes([0,0,1,1])
          ax.plot(x,y,color='purple')
          ax.set_xlim([0,3])
          ax.set_ylim([0,2])
```

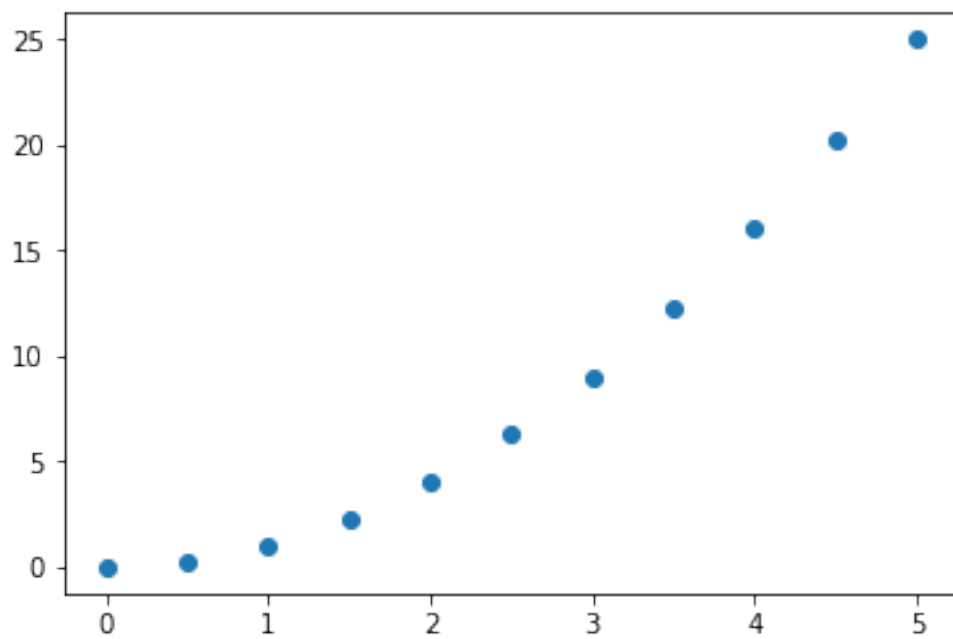
```
Out[109]: (0, 2)
```

2.7 Special plot types

In [110]: `plt.scatter(x,y)`

Out[110]: `<matplotlib.collections.PathCollection at 0x7f2e898ef278>`



```
In [112]: from random import sample
```

```
In [117]: data=sample(range(1,1000),100)  
plt.hist(data, edgecolor = 'red' )
```

```
Out[117]: (array([13.,  8.,  7.,  6.,  8., 15., 12.,  7., 12., 12.]),  
array([  8. , 105.4, 202.8, 300.2, 397.6, 495. , 592.4, 689.8, 787.2,  
      884.6, 982. ]),  
<a list of 10 Patch objects>)
```

