In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [4]:

```
%matplotlib inline
```

In [6]:

```
x1=np.linspace(0,10,100)
```

In [7]:

```
x1
```

Out[7]:

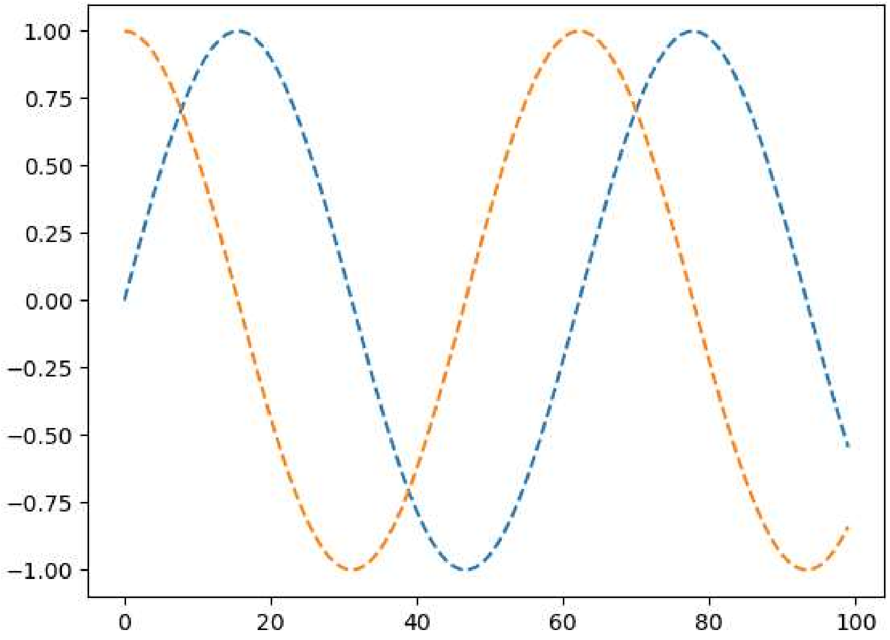
```
array([ 0.        ,  0.1010101 ,  0.2020202 ,  0.3030303 ,  0.4040404 ,
        0.50505051,  0.60606061,  0.70707071,  0.80808081,  0.90909091,
        1.01010101,  1.11111111,  1.21212121,  1.31313131,  1.41414141,
        1.51515152,  1.61616162,  1.71717172,  1.81818182,  1.91919192,
        2.02020202,  2.12121212,  2.22222222,  2.32323232,  2.42424242,
        2.52525253,  2.62626263,  2.72727273,  2.82828283,  2.92929293,
        3.03030303,  3.13131313,  3.23232323,  3.33333333,  3.43434343,
        3.53535354,  3.63636364,  3.73737374,  3.83838384,  3.93939394,
        4.04040404,  4.14141414,  4.24242424,  4.34343434,  4.44444444,
        4.54545455,  4.64646465,  4.74747475,  4.84848485,  4.94949495,
        5.05050505,  5.15151515,  5.25252525,  5.35353535,  5.45454545,
        5.55555556,  5.65656566,  5.75757576,  5.85858586,  5.95959596,
        6.06060606,  6.16161616,  6.26262626,  6.36363636,  6.46464646,
        6.56565657,  6.66666667,  6.76767677,  6.86868687,  6.96969697,
        7.07070707,  7.17171717,  7.27272727,  7.37373737,  7.47474747,
        7.57575758,  7.67676768,  7.77777778,  7.87878788,  7.97979798,
        8.08080808,  8.18181818,  8.28282828,  8.38383838,  8.48484848,
        8.58585859,  8.68686869,  8.78787879,  8.88888889,  8.98989899,
        9.09090909,  9.19191919,  9.29292929,  9.39393939,  9.49494949,
        9.5959596 ,  9.6969697 ,  9.7979798 ,  9.8989899 , 10.        ])
```

In [10]:

```
fig=plt.figure()

plt.plot(np.sin(x1),'--')
plt.plot(np.cos(x1),'--')

```

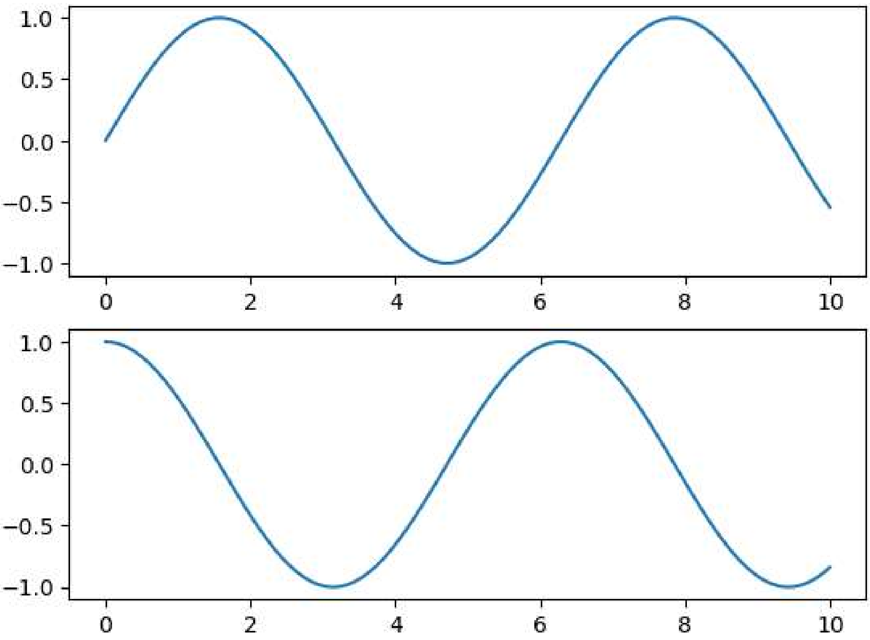
Out[10]:

```
[<matplotlib.lines.Line2D at 0x14895cc5850>]
```



In [13]:

```python
plt.subplot(2,1,1)
plt.plot(x1,np.sin(x1))
plt.subplot(2, 1, 2)   # (rows, columns, panel number)
plt.plot(x1, np.cos(x1));
```



In [15]:

```python
plt.gcf() # get current figure informaton
```

Out[15]:

```
<Figure size 640x480 with 0 Axes>
```

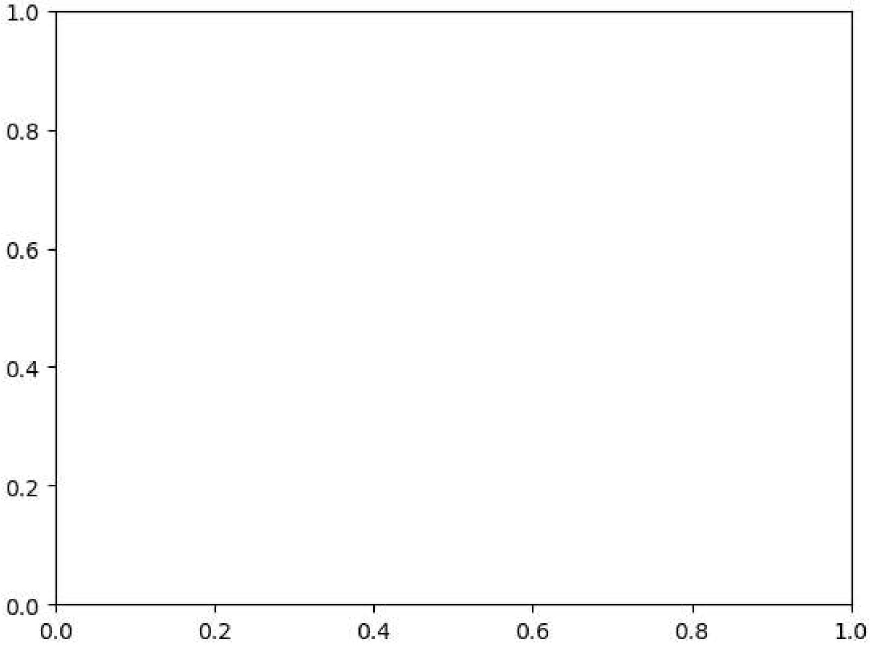
```
<Figure size 640x480 with 0 Axes>
```

In [16]:

```python
plt.gca() # get current axis information
```

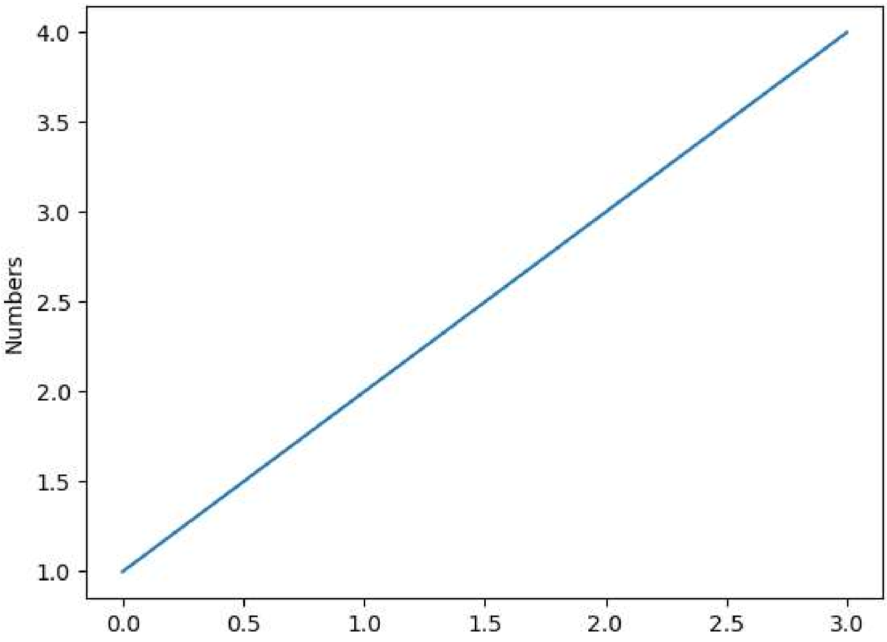
Out[16]:

```
<Axes: >
```



In [17]:

```python
plt.plot([1,2,3,4])
plt.ylabel('Numbers')
plt.show()
```

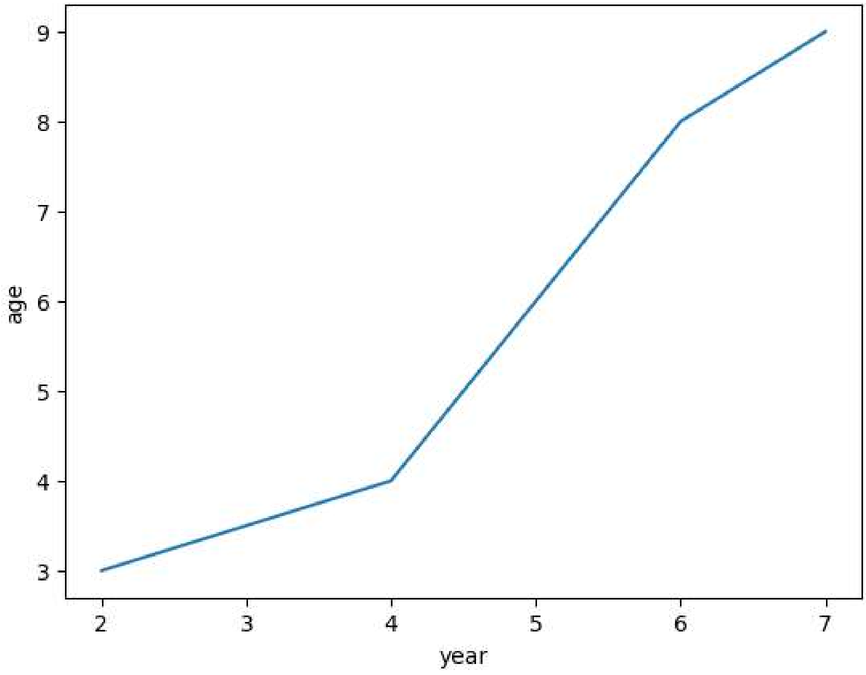


In [20]:

```python
plt.plot([2,4,6,7],[3,4,8,9])
plt.ylabel('age')
plt.xlabel('year')
```

Out[20]:

```
Text(0.5, 0, 'year')
```

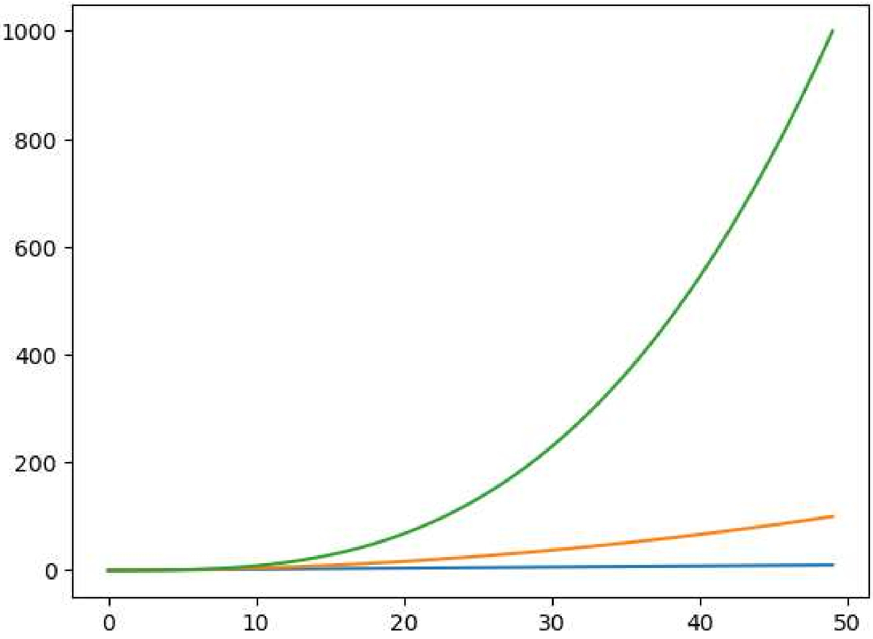


In [26]:

```
x = np.linspace(0,10,50)
plt.plot(x, label='linear')
plt.plot(x**2, label='quadratic')
plt.plot(x**3, label='cubic')
```

Out[26]:

```
[<matplotlib.lines.Line2D at 0x14898bf7e10>]
```

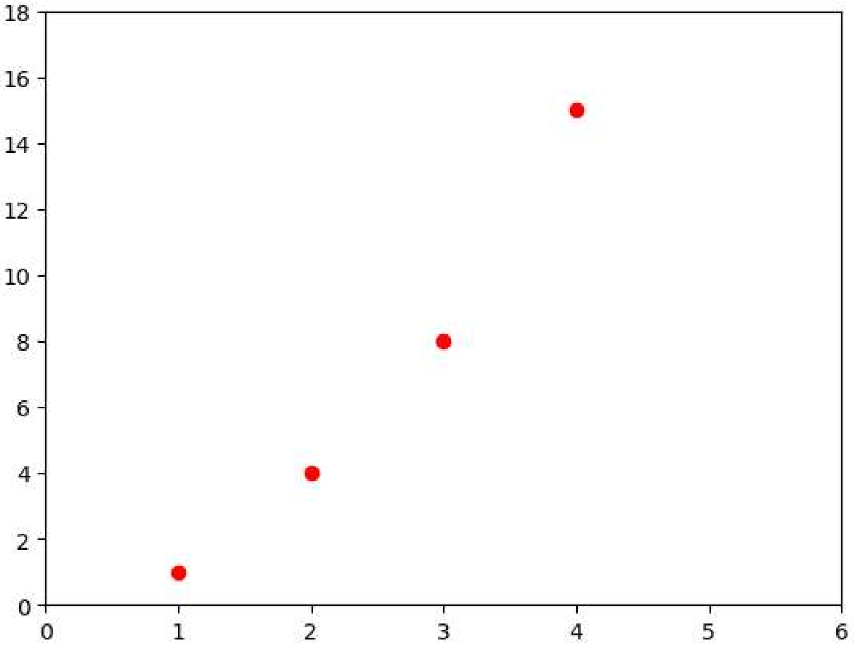


In [30]:

```
plt.plot([1,2,3,4],[1,4,8,15],'ro')
plt.axis([0,6,0,18])
```

Out[30]:

```
(0.0, 6.0, 0.0, 18.0)
```

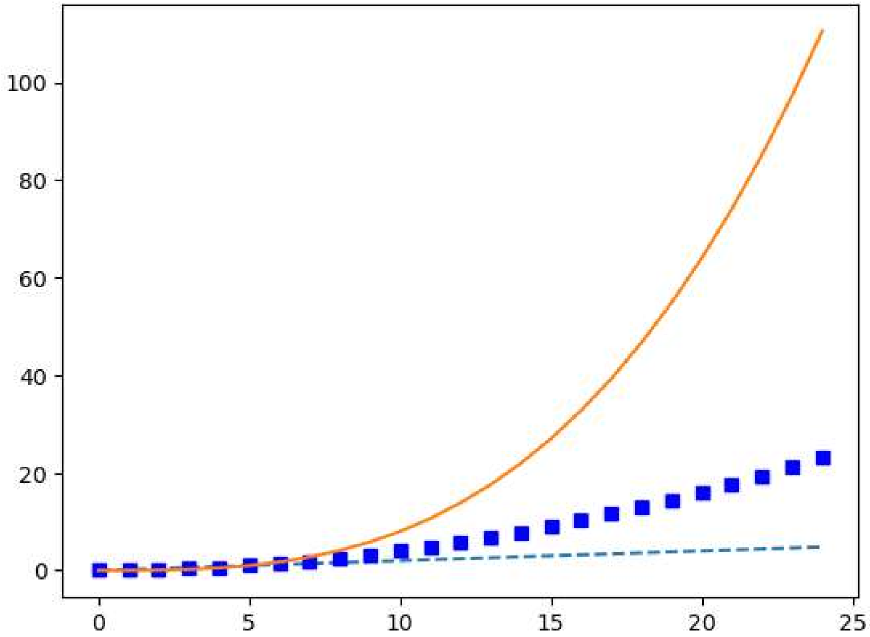


In [32]:

```
t=np.arange(0.,5.,0.2)
```

In [34]:
```
plt.plot(t,'--',t**2,'bs',t**3)
```

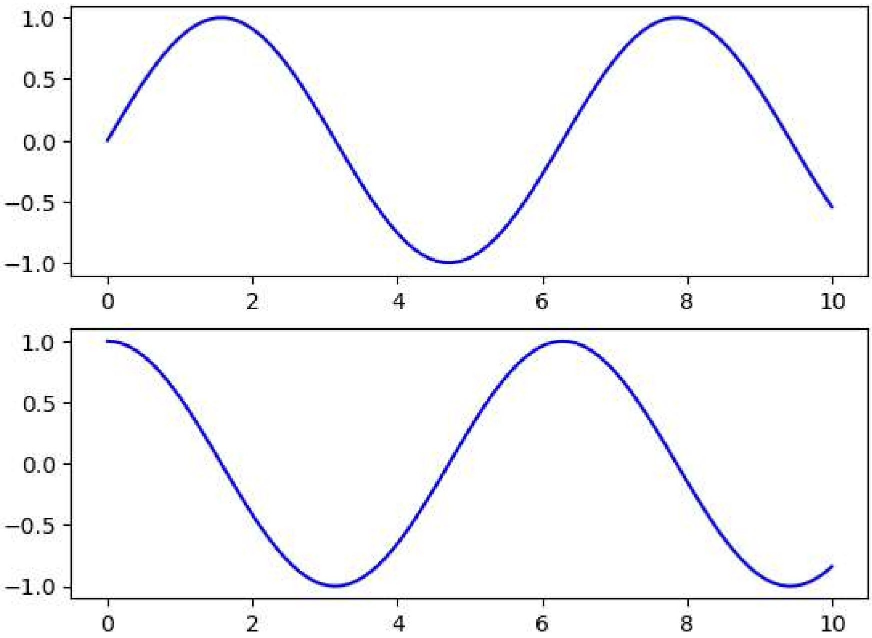
Out[34]:
```
[<matplotlib.lines.Line2D at 0x14899edfc90>,
 <matplotlib.lines.Line2D at 0x14897e29290>,
 <matplotlib.lines.Line2D at 0x14899edc9d0>]
```



In [35]:
```
# Object-Oriented API
```

In [42]:
```
# First create a grid of plots
# ax will be an array of two Axes objects
fig, ax=plt.subplots(2)

# Call plot() method on the appropriate object
ax[0].plot(x1,np.sin(x1),'b-')
ax[1].plot(x1,np.cos(x1),'b-')
```

Out[42]:
```
[<matplotlib.lines.Line2D at 0x14897a1d1d0>]
```

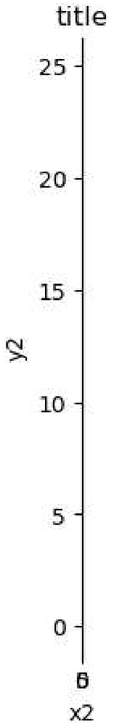


In [48]:
```

```

```
<Figure size 640x480 with 0 Axes>
```

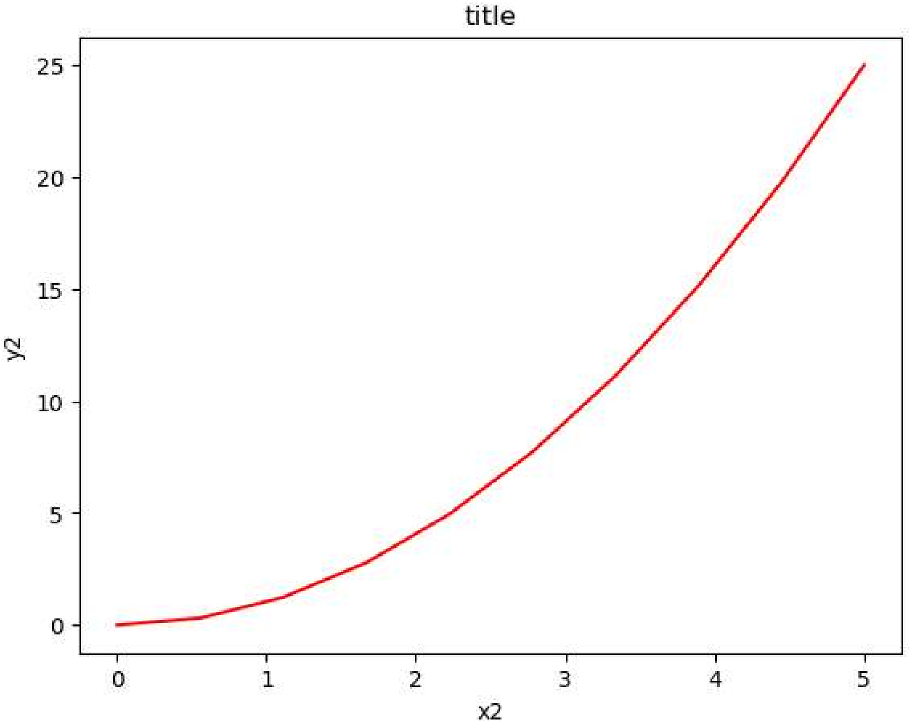
In [54]:

```
fig=plt.figure()
x2=np.linspace(0, 5, 10)
y2=x2**2

axes=fig.add_axes([0.1, 0.1, 0.0, 0.8 ])
axes.plot(x2,y2,'r')

axes.set_xlabel('x2')
axes.set_ylabel('y2')
axes.set_title('title');
```



In [51]:

```python
fig = plt.figure()

x2 = np.linspace(0, 5, 10)
y2 = x2 ** 2

axes = fig.add_axes([0.1, 0.1, 0.8, 0.8])

axes.plot(x2, y2, 'r')

axes.set_xlabel('x2')
axes.set_ylabel('y2')
axes.set_title('title');
```



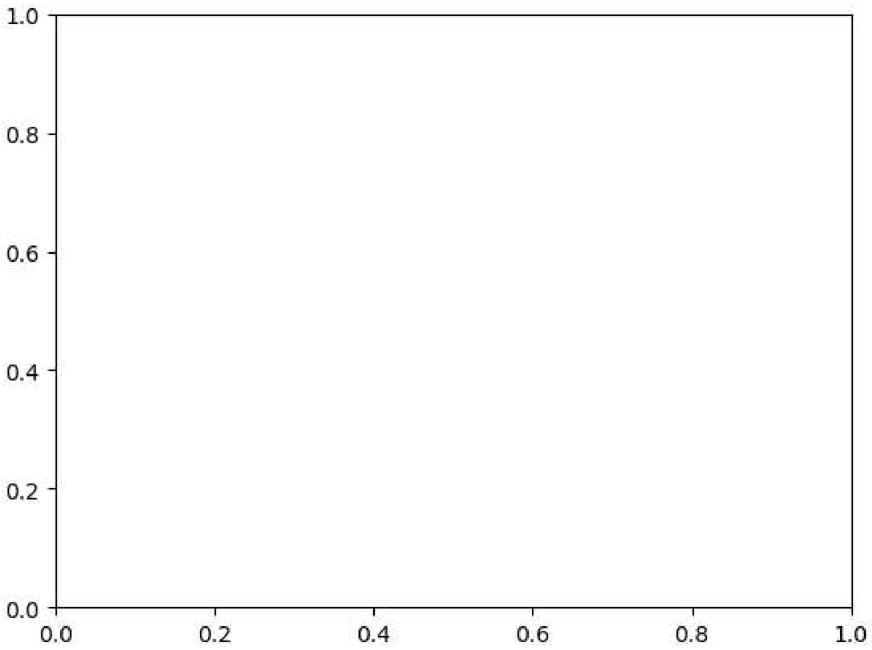
In [55]:

```python
fig=plt.figure()
```

```
<Figure size 640x480 with 0 Axes>
```

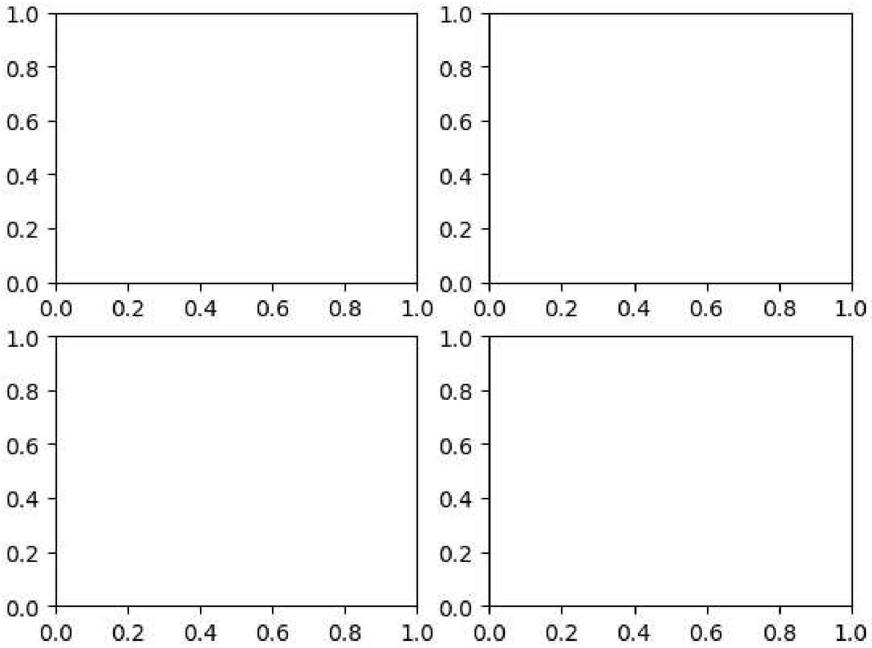
In [56]:

```python
ax=plt.axes()
```



In [58]:

```
fig=plt.figure()
ax1=fig.add_subplot(2,2,1)
ax2=fig.add_subplot(2,2,2)
ax3=fig.add_subplot(2,2,3)
ax4=fig.add_subplot(2,2,4)

```

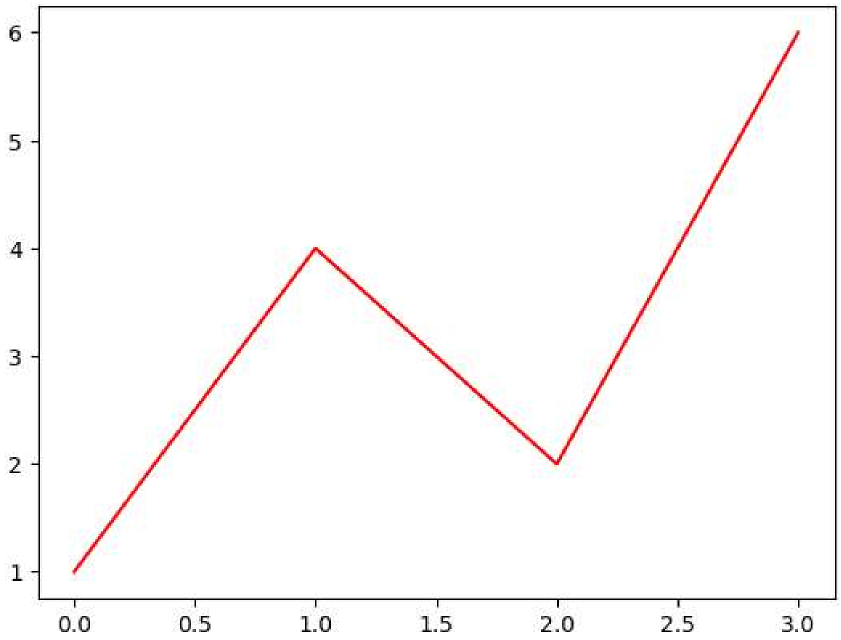


In [62]:

```
plt.plot([1,4,2,6], 'r-')
```

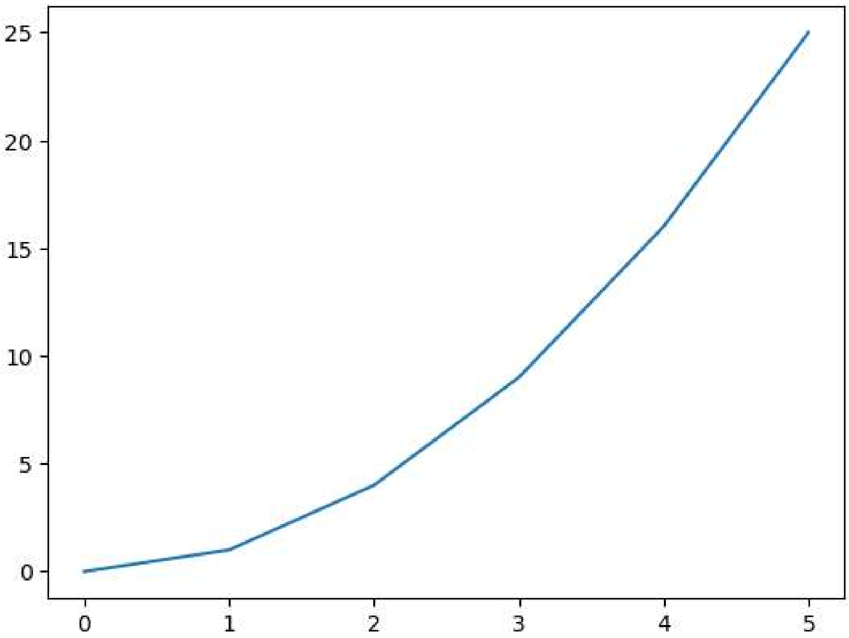
Out[62]:

```
[<matplotlib.lines.Line2D at 0x1489b020750>]
```



```
In [63]:
x3 = range(6)

plt.plot(x3, [xi**2 for xi in x3])
```

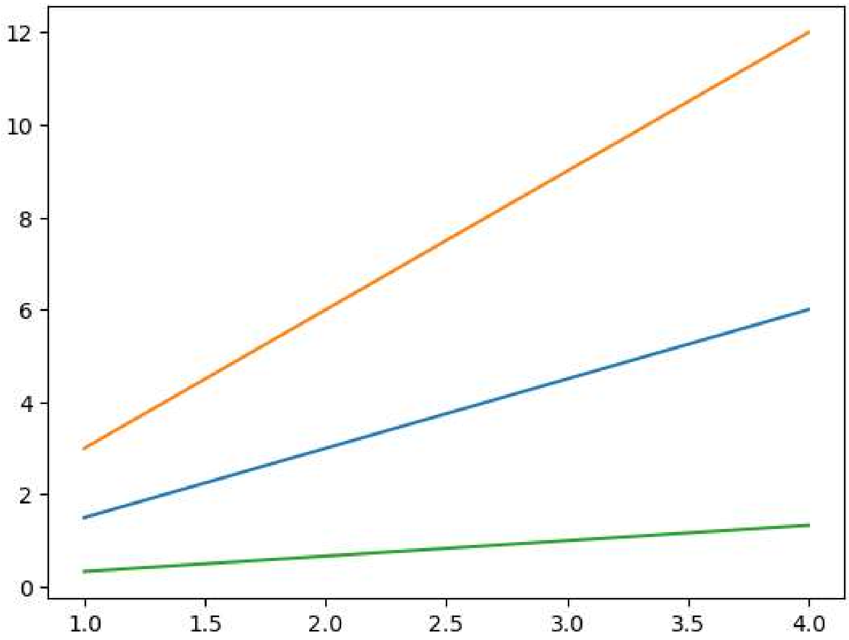
```
Out[63]: [<matplotlib.lines.Line2D at 0x1489ad5ead0>]
```



```
In [64]:
# multiline plot
```

```
In [65]:
x4 = range(1, 5)

plt.plot(x4, [xi*1.5 for xi in x4])

plt.plot(x4, [xi*3 for xi in x4])

plt.plot(x4, [xi/3.0 for xi in x4])

```

```
Out[65]: [<matplotlib.lines.Line2D at 0x1489ae8fbd0>]
```



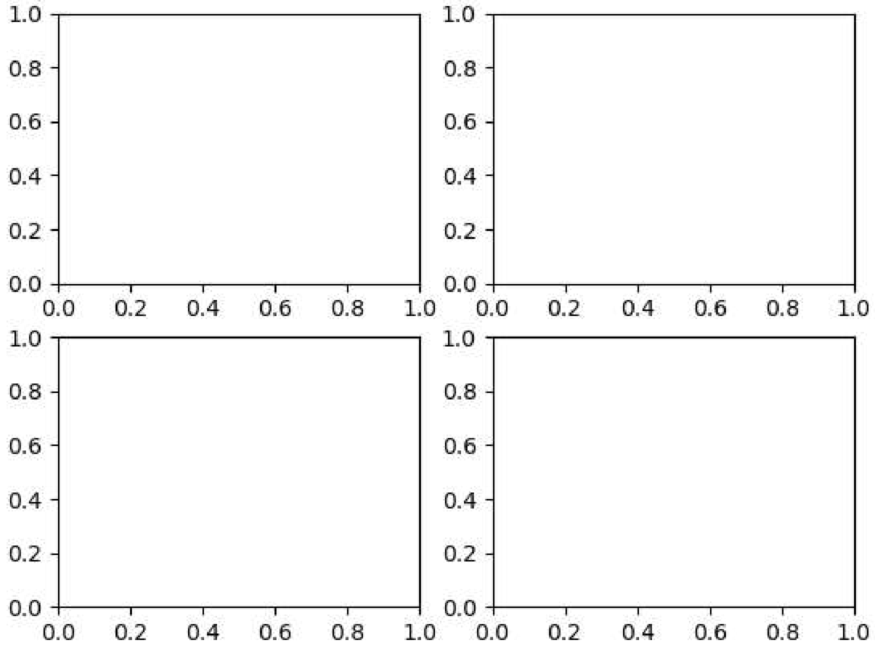
```
In [66]:
# saving plot
```

In [68]:
```
fig.savefig('plot1.png')
```

In [70]:
```
from IPython.display import Image
```

In [75]:
```
Image('plot1.png')
```

Out[75]:



In [76]:
```
fig.canvas.get_supported_filetypes()
```

Out[76]:
```
{'eps': 'Encapsulated Postscript',
 'jpg': 'Joint Photographic Experts Group',
 'jpeg': 'Joint Photographic Experts Group',
 'pdf': 'Portable Document Format',
 'pgf': 'PGF code for LaTeX',
 'png': 'Portable Network Graphics',
 'ps': 'Postscript',
 'raw': 'Raw RGBA bitmap',
 'rgba': 'Raw RGBA bitmap',
 'svg': 'Scalable Vector Graphics',
 'svgz': 'Scalable Vector Graphics',
 'tif': 'Tagged Image File Format',
 'tiff': 'Tagged Image File Format',
 'webp': 'WebP Image Format'}
```

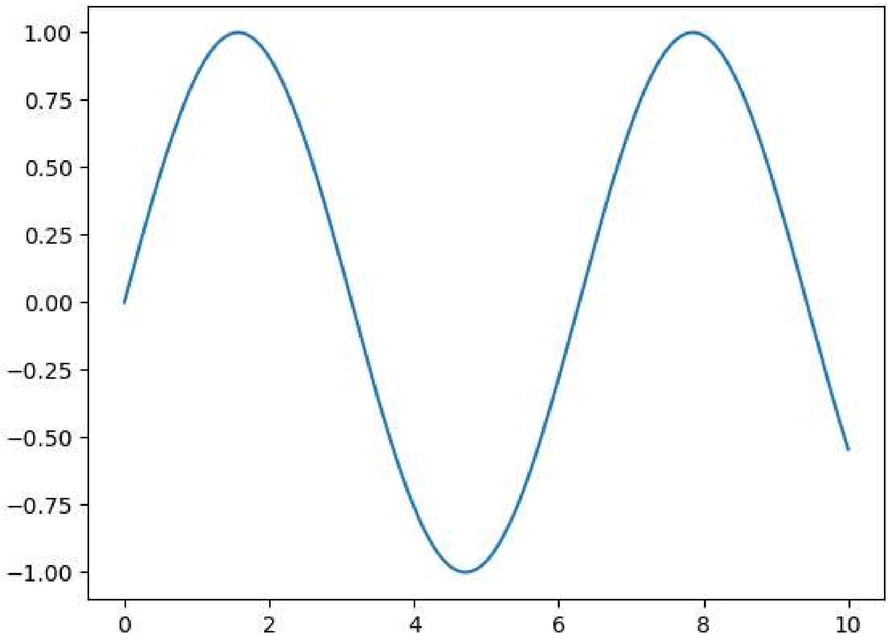
In [77]:
```
#line plot
```

In [79]:

```
fig=plt.figure()
ax=plt.axes()
x5=np.linspace(0,10,1000)
ax.plot(x5, np.sin(x5))
```

Out[79]:

```
[<matplotlib.lines.Line2D at 0x1489ca45150>]
```



In [80]:

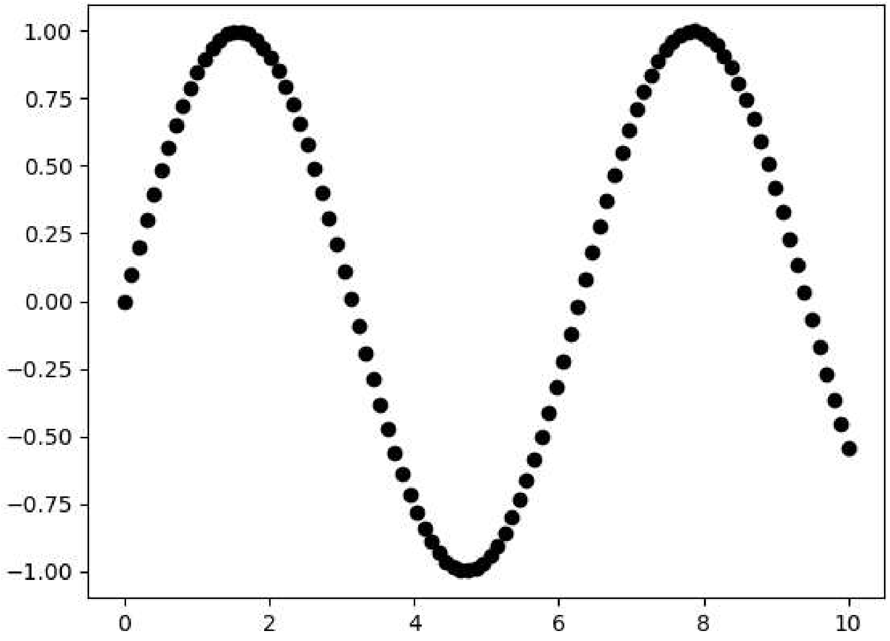
```
# scatterplot
```

In [83]:

```
x7=np.linspace(0,10,100)
y7=np.sin(x7)
plt.plot(x7,y7,'o' ,color='black' ,)
```

Out[83]:

```
[<matplotlib.lines.Line2D at 0x1489b8c1010>]
```



In [84]:

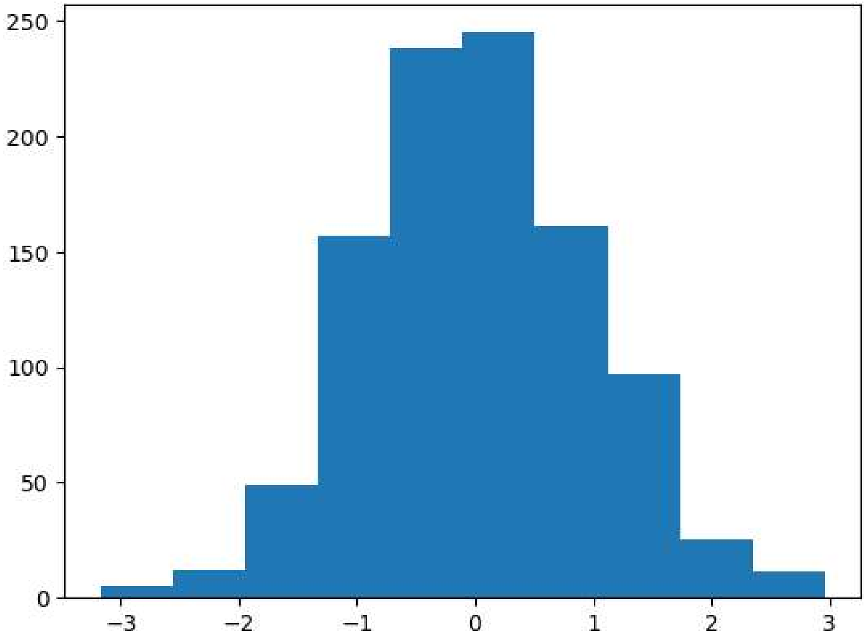
```
# histogram
```

In [87]:

```
r=np.random.randn(1000)
plt.hist(r)
```

Out[87]:

```
(array([  5.,  12.,  49., 157., 238., 245., 161.,  97.,  25.,  11.]),
 array([-3.17104034, -2.55764453, -1.94424873, -1.33085293, -0.71745712,
        -0.10406132,  0.50933449,  1.12273029,  1.7361261 ,  2.3495219 ,
         2.96291771]),
 <BarContainer object of 10 artists>)
```



In [88]:

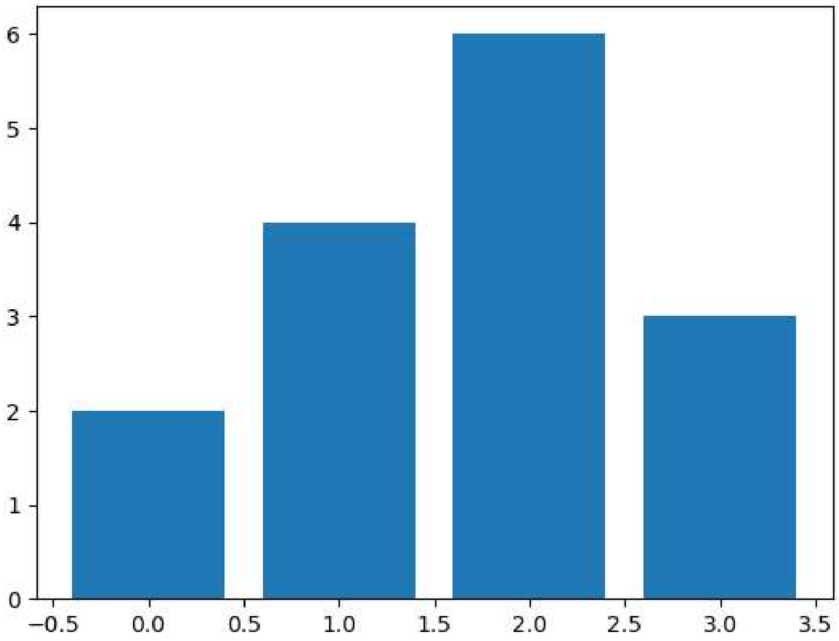
```
# barchart
```

In [89]:

```
data2=[2,4,6,3]
plt.bar(range(len(data2)),data2)
```

Out[89]:

```
<BarContainer object of 4 artists>
```

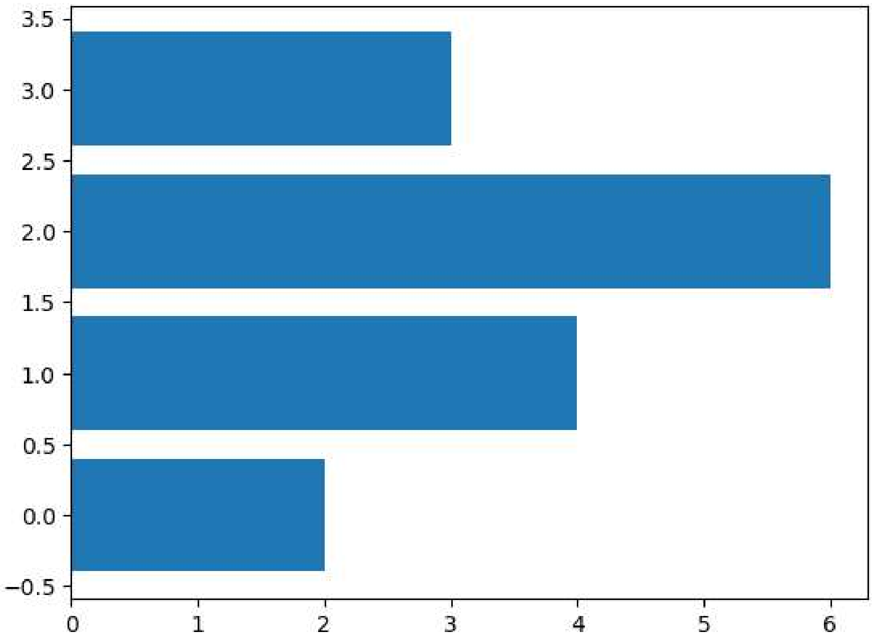


In [90]:

```
# horizontal bar chart
```

In [91]:
```
data2=[2,4,6,3]
plt.barh(range(len(data2)),data2)
```

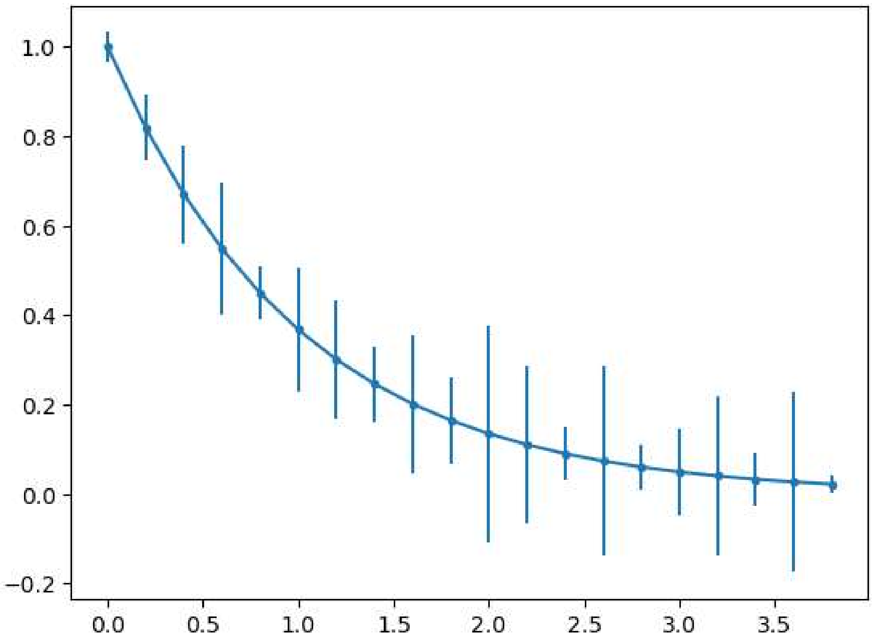
Out[91]:
```
<BarContainer object of 4 artists>
```



In [92]:
```
# Error bar chart
```

In [95]:
```
x9=np.arange(0,4,0.2)
y9=np.exp(-x9)
e1=0.1*np.abs(np.random.randn(len(y9)))
plt.errorbar(x9,y9, yerr=e1,fmt='.-')
```

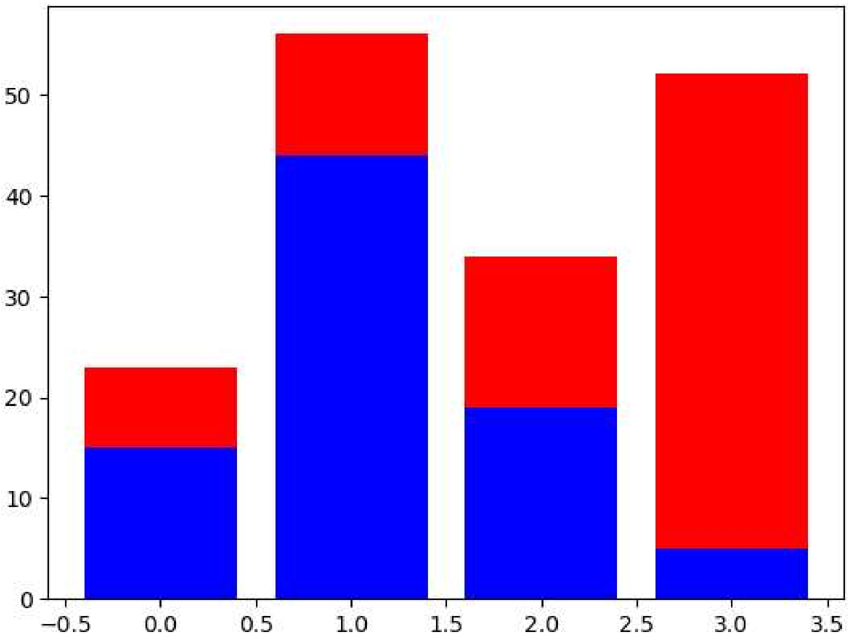
Out[95]:
```
<ErrorbarContainer object of 3 artists>
```



In [96]:
```
# stacked bar chart
```

In [104]:
```
a=[23,56,34,52]
b=[15,44,19,5]
z2=range(4)

plt.bar(z2, a, color='r')
plt.bar(z2, b, color='b' )
```

Out[104]:
```
<BarContainer object of 4 artists>
```



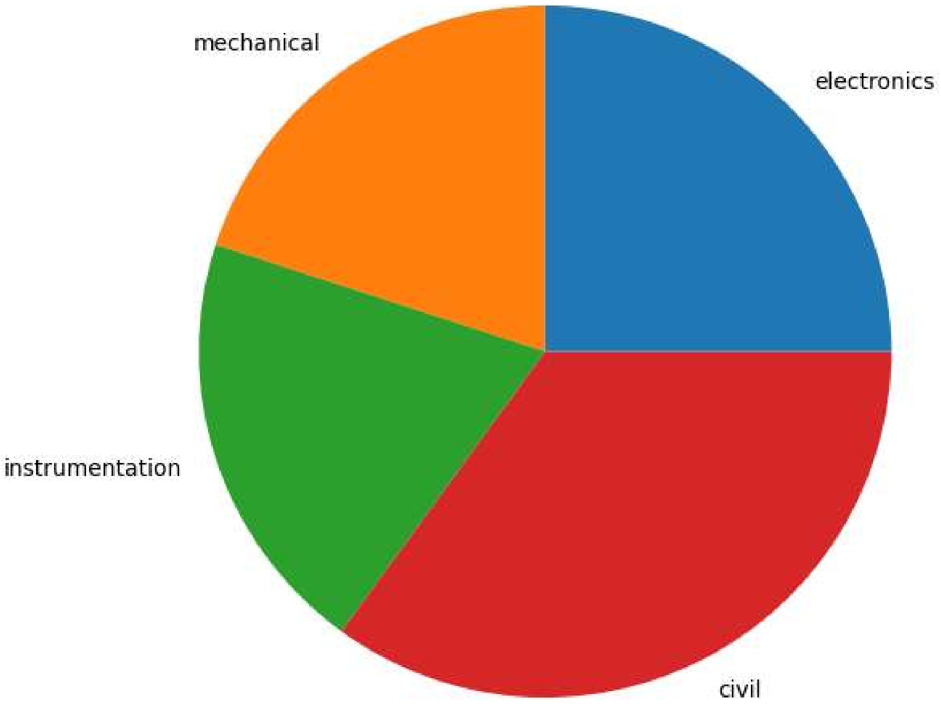
In [ ]:
```
# pie chart
```

In [107]:

```
plt.figure(figsize=(7,7))

x10=[25,20,20,35]
label=['electronics','mechanical','instrumentation','civil']

plt.pie(x10,labels=label)
```

Out[107]:

```
([<matplotlib.patches.Wedge at 0x1489f048510>,
  <matplotlib.patches.Wedge at 0x1489f02ead0>,
  <matplotlib.patches.Wedge at 0x1489e13cfd0>,
  <matplotlib.patches.Wedge at 0x1489efd5590>],
 [Text(0.7778174593052024, 0.7778174593052024, 'electronics'),
  Text(-0.6465637858537404, 0.8899186877588754, 'mechanical'),
  Text(-1.046162158377023, -0.33991872319707317, 'instrumentation'),
  Text(0.4993895680663522, -0.98010716725596, 'civil')])
```



In [108]:

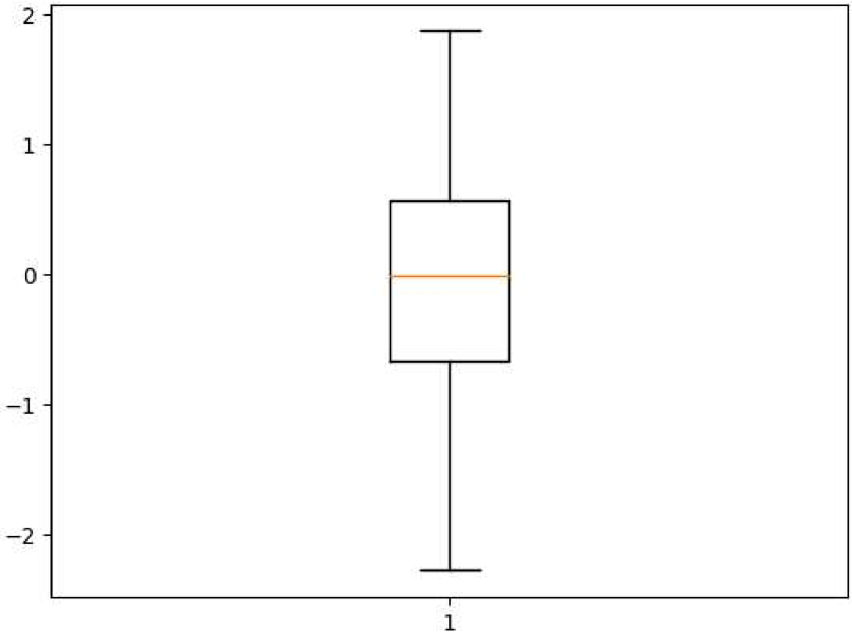
```
# box plot
```

In [116]:

```
data3=np.random.randn(100)
plt.boxplot(data3)
```

Out[116]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x148a1764290>,
  <matplotlib.lines.Line2D at 0x148a1764ed0>],
 'caps': [<matplotlib.lines.Line2D at 0x148a1765890>,
  <matplotlib.lines.Line2D at 0x148a1766110>],
 'boxes': [<matplotlib.lines.Line2D at 0x148a1753c50>],
 'medians': [<matplotlib.lines.Line2D at 0x148a17669d0>],
 'fliers': [<matplotlib.lines.Line2D at 0x148a1766d10>],
 'means': []}
```



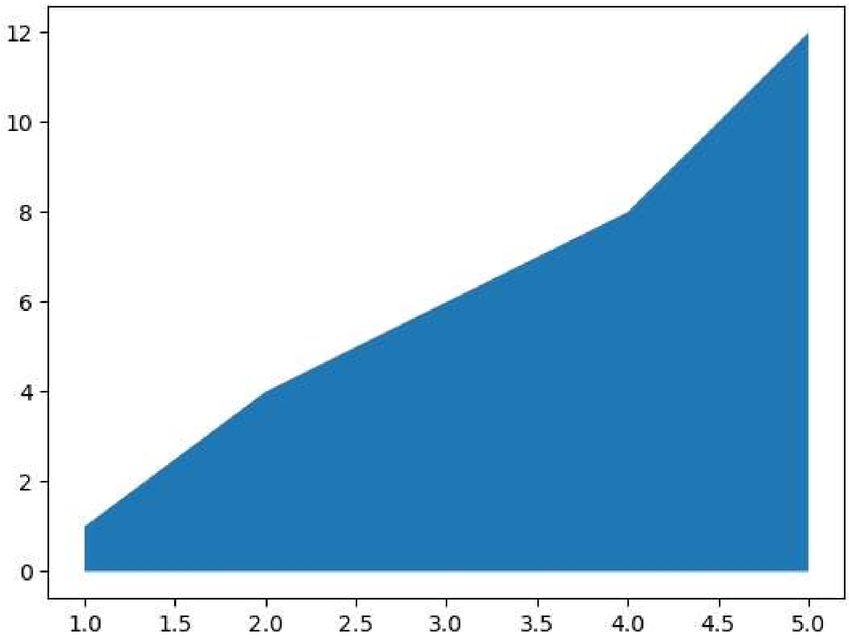
In [117]:

```
# area chart
```

In [118]:

```
x12=range(1,6)
y12=[1,4,6,8,12]

plt.fill_between(x12,y12)
```

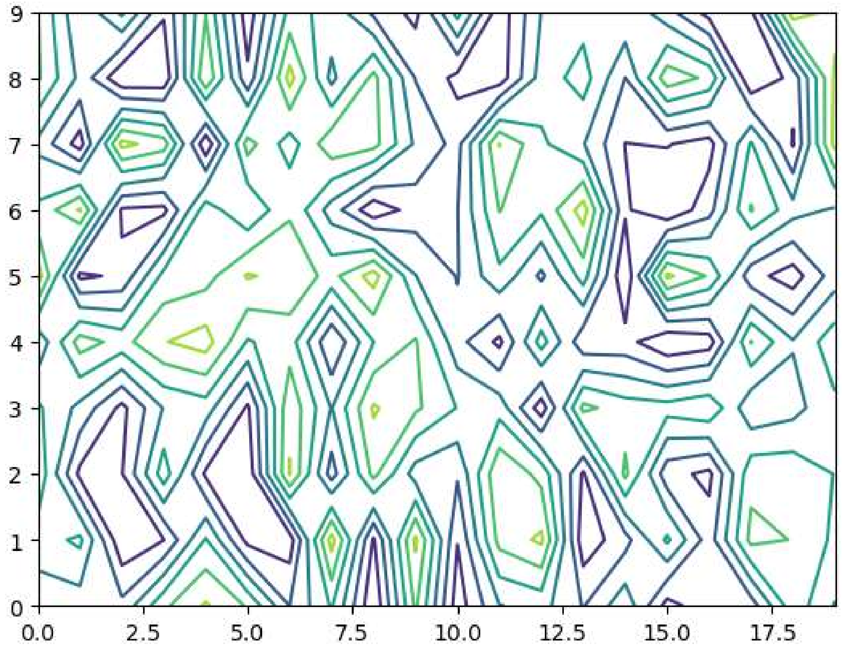
Out[118]: `<matplotlib.collections.PolyCollection at 0x148a17aef50>`



In [119]:
```
# contour plot
```

In [120]:
```
matrix1=np.random.rand(10,20)

plt.contour(matrix1)
```

Out[120]:
```
<matplotlib.contour.QuadContourSet at 0x148a143fb50>
```



In [121]:
```
# style in matplotlib
```

In [122]:
```
plt.style.available
```

Out[122]:
```
['Solarize_Light2',
 '_classic_test_patch',
 '_mpl-gallery',
 '_mpl-gallery-nogrid',
 'bmh',
 'classic',
 'dark_background',
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'seaborn-v0_8',
 'seaborn-v0_8-bright',
 'seaborn-v0_8-colorblind',
 'seaborn-v0_8-dark',
 'seaborn-v0_8-dark-palette',
 'seaborn-v0_8-darkgrid',
 'seaborn-v0_8-deep',
 'seaborn-v0_8-muted',
 'seaborn-v0_8-notebook',
 'seaborn-v0_8-paper',
 'seaborn-v0_8-pastel',
 'seaborn-v0_8-poster',
 'seaborn-v0_8-talk',
 'seaborn-v0_8-ticks',
 'seaborn-v0_8-white',
 'seaborn-v0_8-whitegrid',
 'tableau-colorblind10']
```

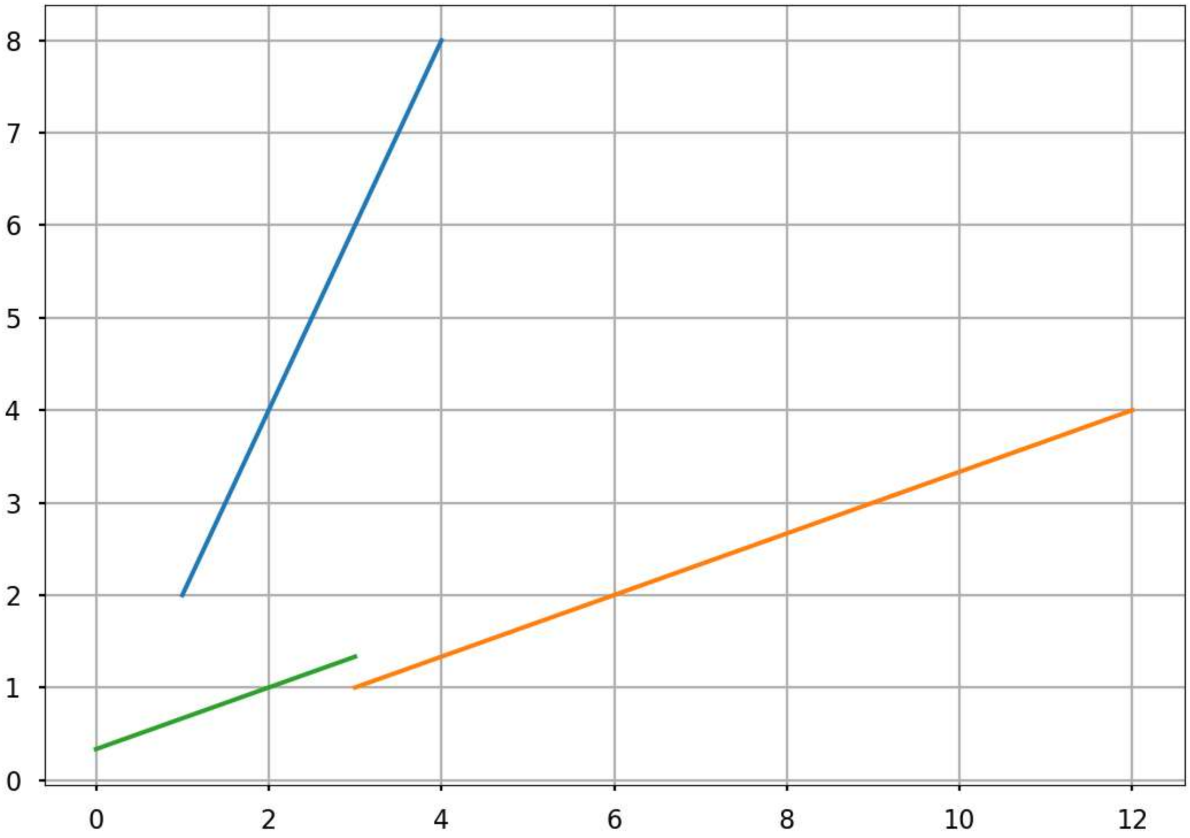
In [124]:
```
plt.style.use('seaborn-v0_8-poster')
```

In [ ]:

```
# grid
```

In [137]:

```
x15 = np.arange(1,5)

plt.plot(x15,x15*2,x15*3,x15,x15/3.0 )
plt.grid(True)

```



In [138]:

```
# handling axes
```

In [ ]:

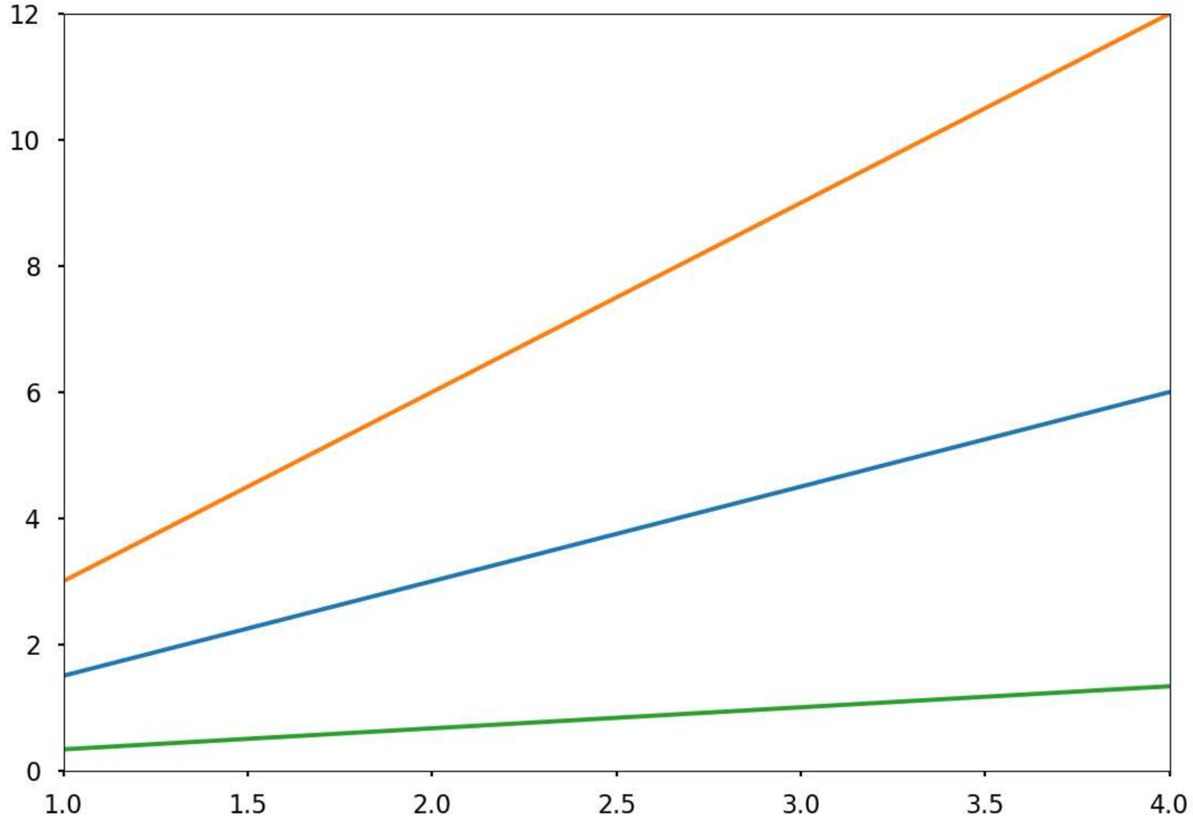
```
x16=np.arange(1,5)
plt.plot(x16, x16*1.5, x16, x16*3.0, x16, x16/3.0)
plt.axes()
plt.axis([0, 5, -1, 13])
```

In [144]:

```
x17=np.arange(1,5)

plt.plot(x15, x15*1.5, x15, x15*3.0, x15, x15/3.0)

plt.xlim([1.0, 4.0])
plt.ylim([0.0, 12.0])
```

Out[144]:

```
(0.0, 12.0)
```



In [145]:

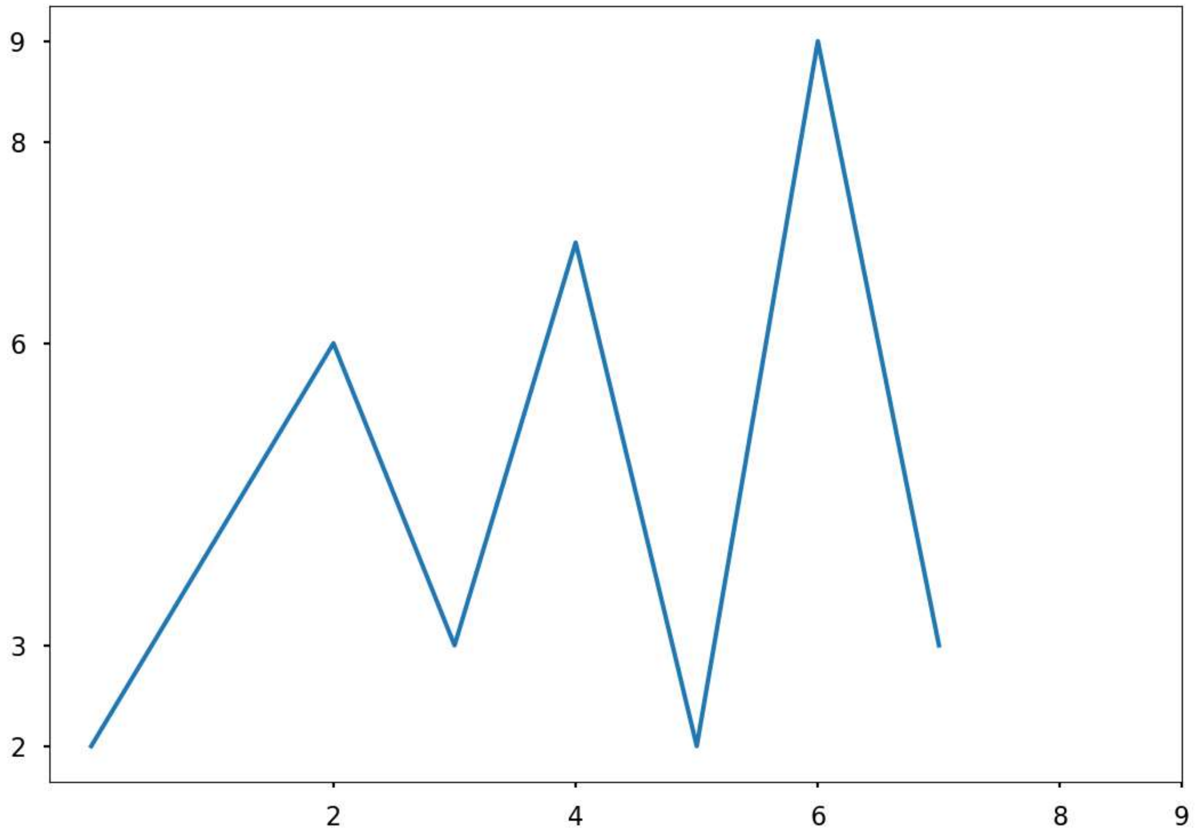
```
# Handling x and y ticks
```

In [150]:

```
u=[2,4,6,3,7,2,9,3]
plt.plot(u)

plt.xticks([2,4,6,8,9])
plt.yticks([2,3,6,8,9])
```

Out[150]:

```
([<matplotlib.axis.YTick at 0x148a51dfd10>,
  <matplotlib.axis.YTick at 0x148a51ce290>,
  <matplotlib.axis.YTick at 0x148a51bf290>,
  <matplotlib.axis.YTick at 0x148a41edcd0>,
  <matplotlib.axis.YTick at 0x148a5211a90>],
 [Text(0, 2, '2'),
  Text(0, 3, '3'),
  Text(0, 6, '6'),
  Text(0, 8, '8'),
  Text(0, 9, '9')])
```

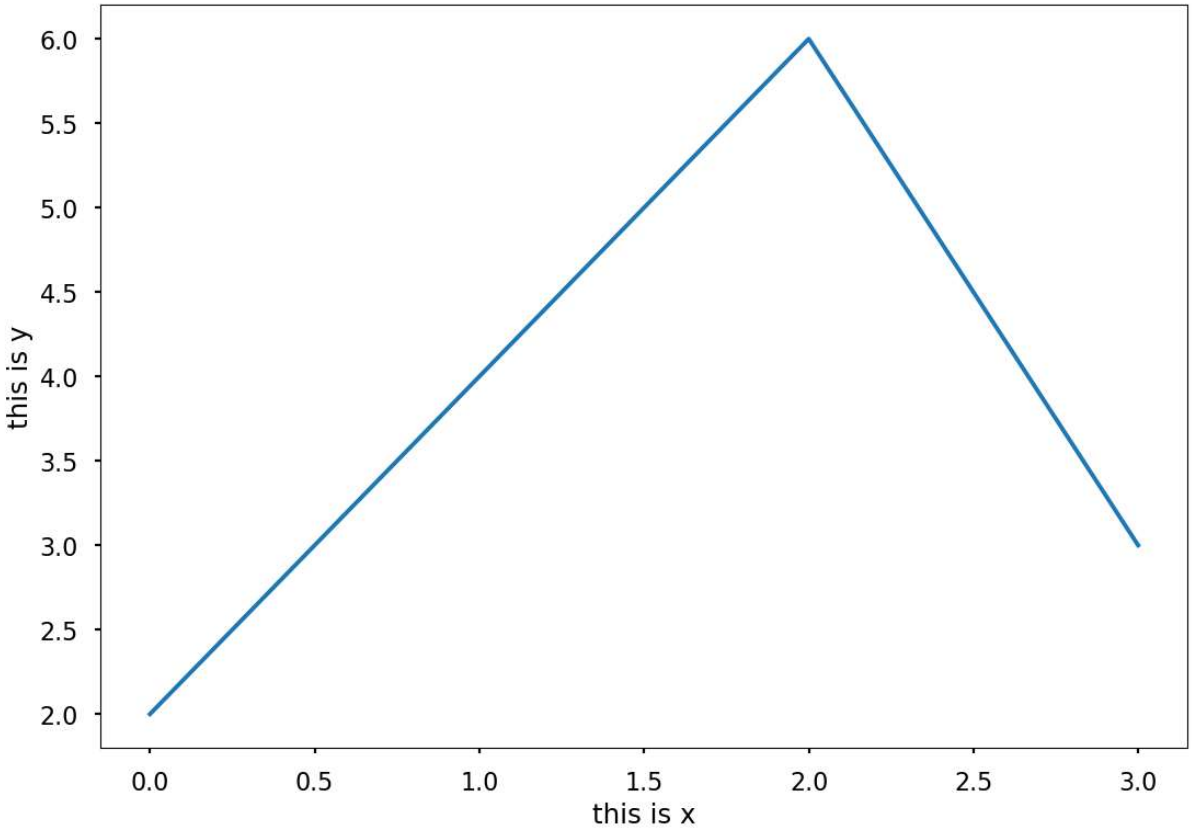


In [151]:

```
# addin labels
```

In [154]:
```
plt.plot([2,4,6,3])
plt.xlabel('this is x')
plt.ylabel('this is y')
```

Out[154]:
```
Text(0, 0.5, 'this is y')
```



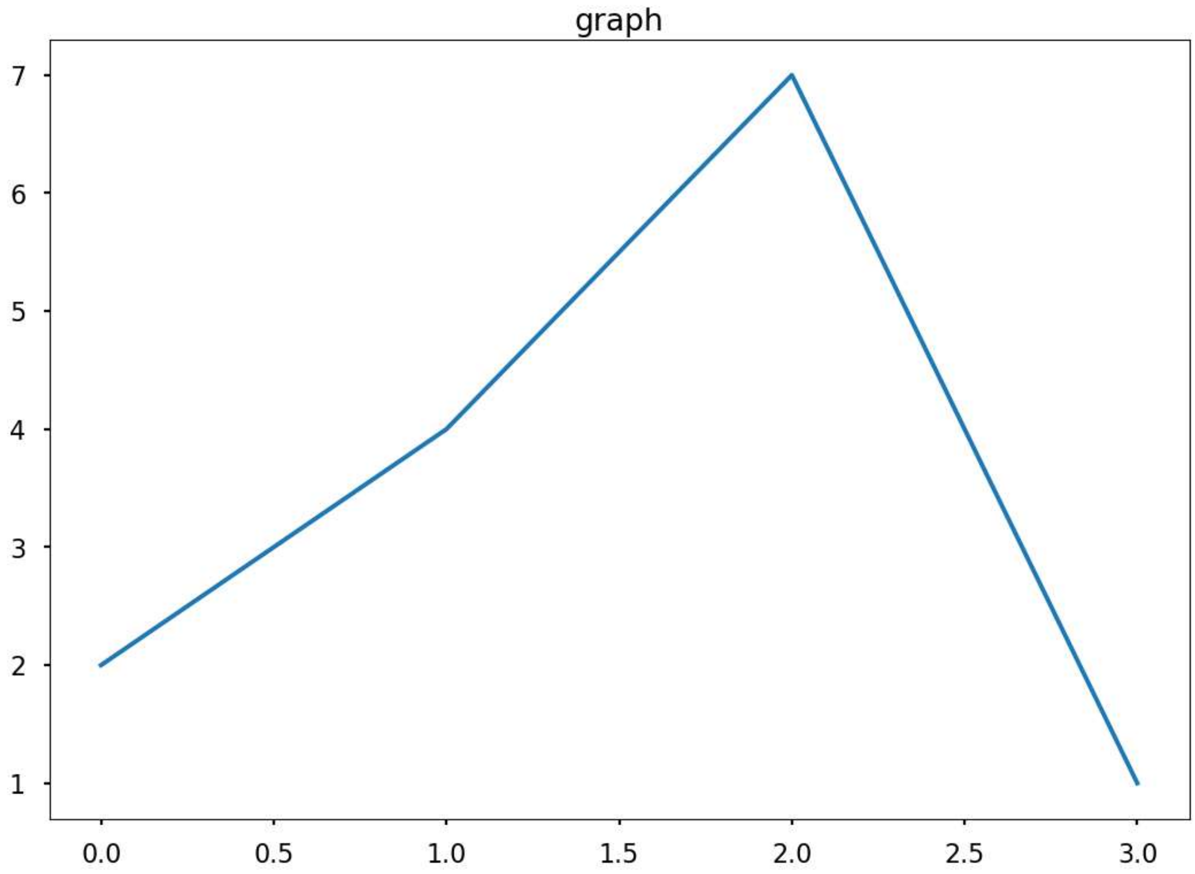
In [155]:
```
# adding a title
```

In [158]:

```
plt.plot([2,4,7,1])
plt.title('graph')
```

Out[158]:

```
Text(0.5, 1.0, 'graph')
```



In [159]:

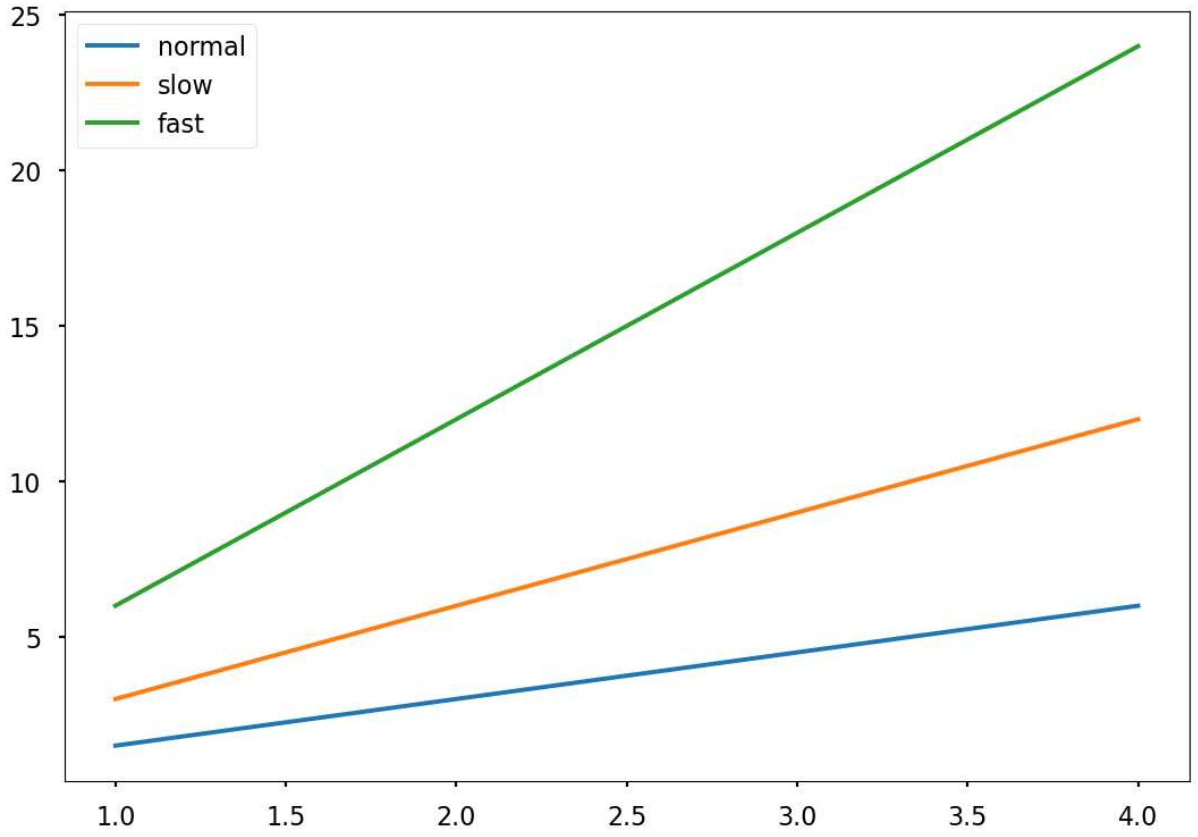
```
# adding a legend
```

In [170]:

```
x20=np.arange(1,5)
fig,ax=plt.subplots()

ax.plot(x20,x20*1.5)
ax.plot(x20,x20*3)
ax.plot(x20,x20*6)
ax.legend(['normal','slow','fast'])
```

Out[170]:

```
<matplotlib.legend.Legend at 0x1489cec9610>
```



In [ ]:

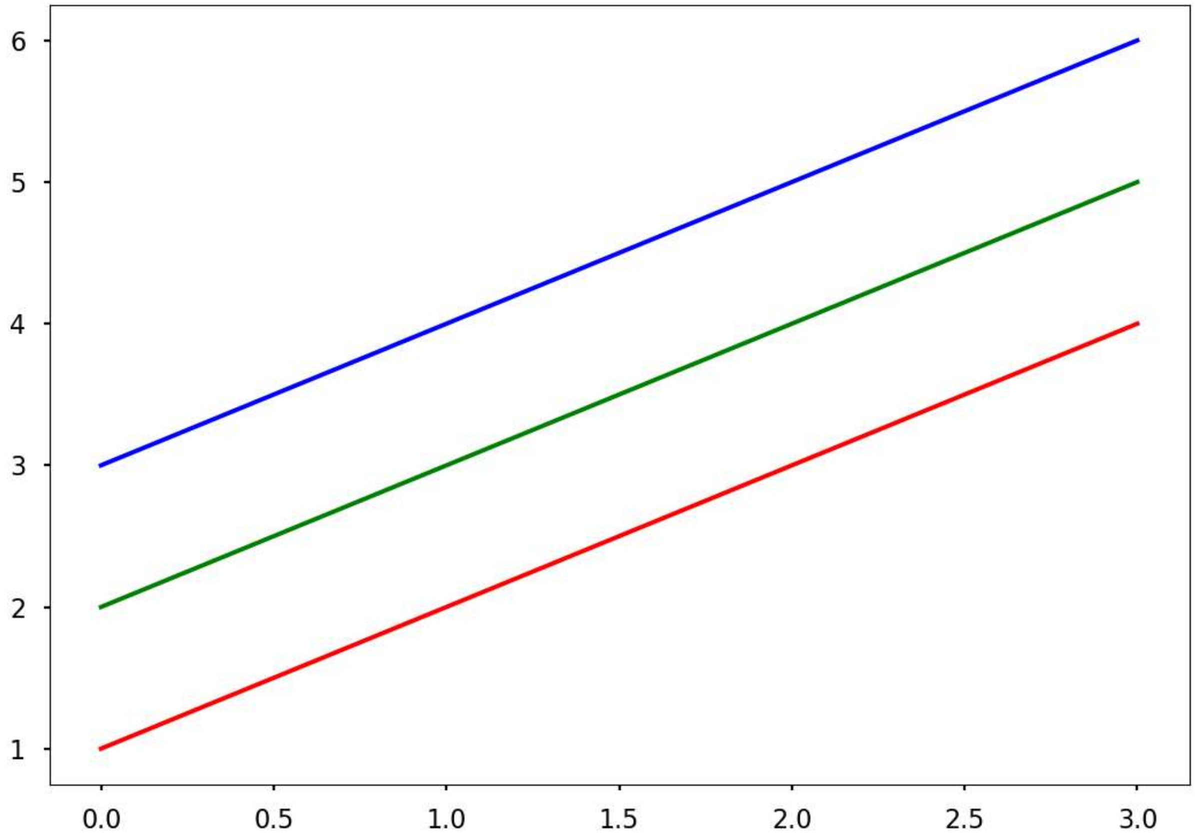
```
# color
```

In [171]:

```
x16 = np.arange(1, 5)

plt.plot(x16, 'r')
plt.plot(x16+1, 'g')
plt.plot(x16+2, 'b')
```

Out[171]:

```
[<matplotlib.lines.Line2D at 0x1489e4c1b10>]
```



In [172]:

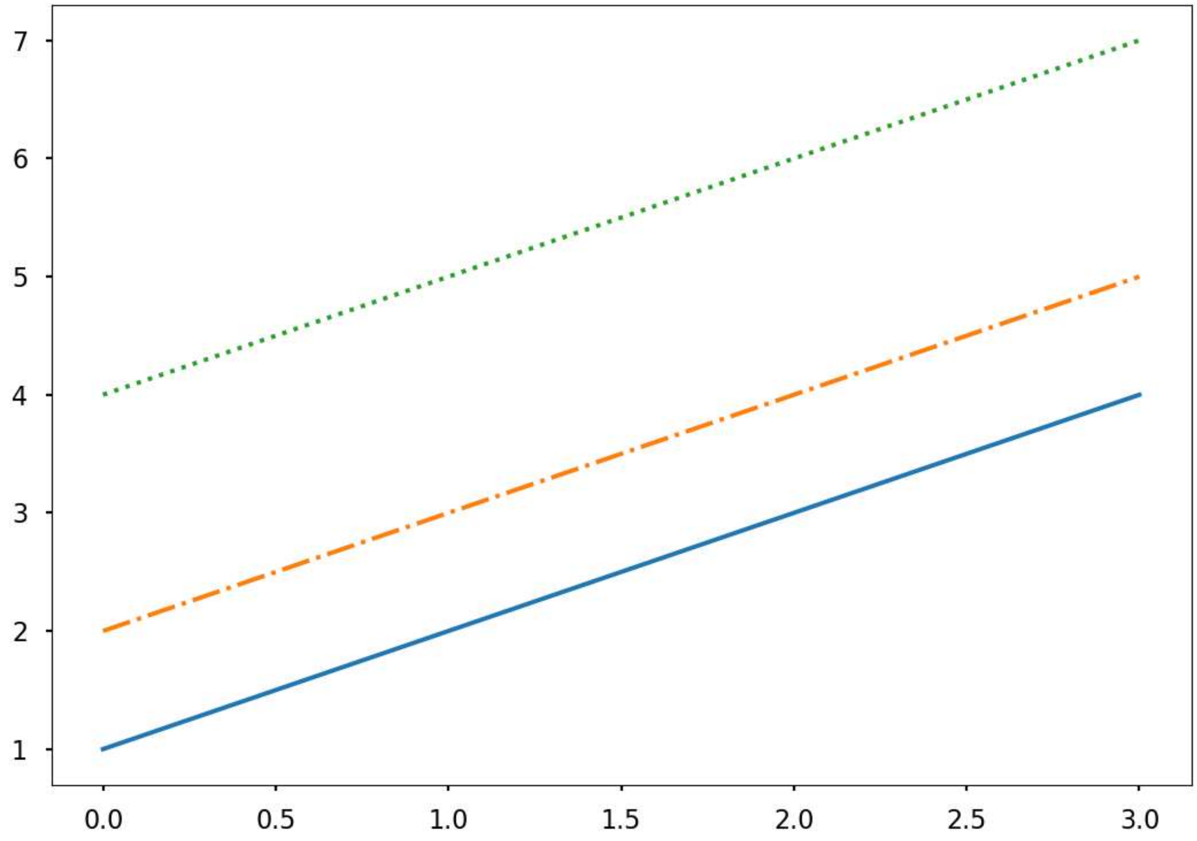
```
# line style
```

In [178]:

```
x21=np.arange(1,5)
plt.plot(x21,'-', x21+1,'-.', x21+3,':')
```

Out[178]:

```
[<matplotlib.lines.Line2D at 0x148a4e15b10>,
 <matplotlib.lines.Line2D at 0x148a4e156d0>,
 <matplotlib.lines.Line2D at 0x148a4e17990>]
```



In [ ]:

```

```