In [1]:
```
import numpy as np
```

In [2]:
```
import pandas as pd
```

In [4]:
```
import matplotlib.pyplot as plt
```

In [5]:
```
x1=np.linspace(0,10,150)
```

In [6]:
```
x1
```

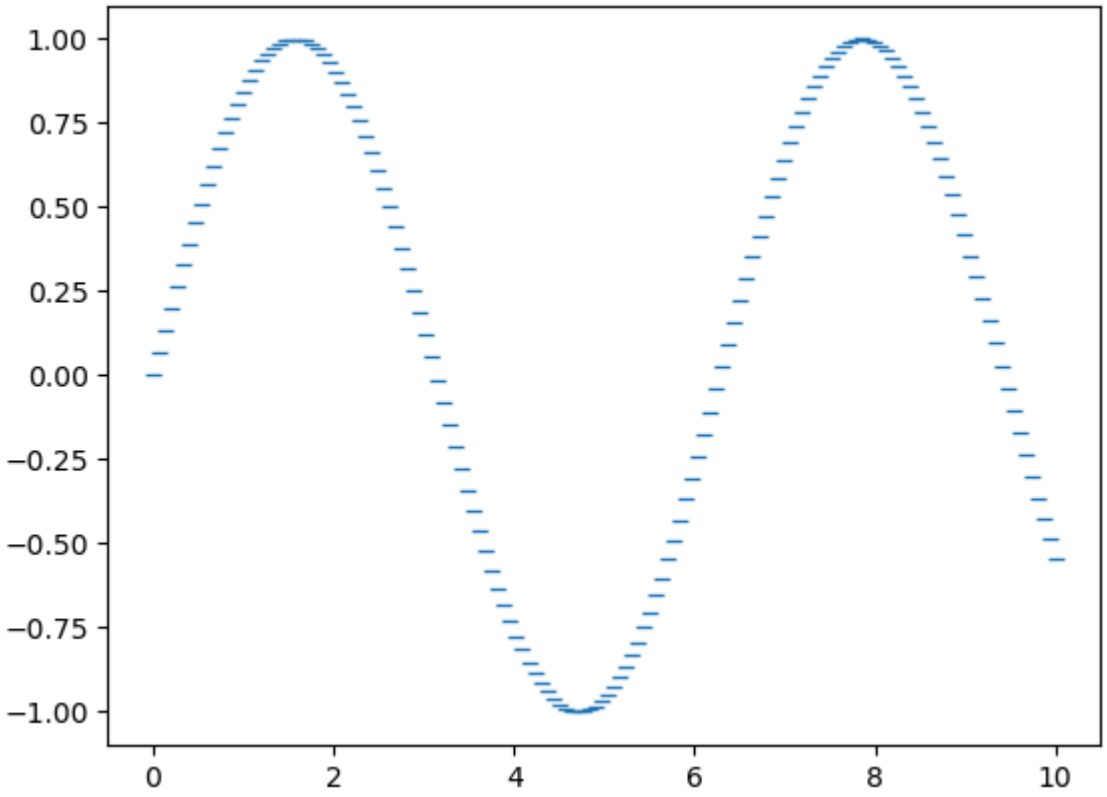
Out[6]:
```
array([ 0.        ,  0.06711409,  0.13422819,  0.20134228,  0.26845638,
        0.33557047,  0.40268456,  0.46979866,  0.53691275,  0.60402685,
        0.67114094,  0.73825503,  0.80536913,  0.87248322,  0.93959732,
        1.00671141,  1.0738255 ,  1.1409396 ,  1.20805369,  1.27516779,
        1.34228188,  1.40939597,  1.47651007,  1.54362416,  1.61073826,
        1.67785235,  1.74496644,  1.81208054,  1.87919463,  1.94630872,
        2.01342282,  2.08053691,  2.14765101,  2.2147651 ,  2.28187919,
        2.34899329,  2.41610738,  2.48322148,  2.55033557,  2.61744966,
        2.68456376,  2.75167785,  2.81879195,  2.88590604,  2.95302013,
        3.02013423,  3.08724832,  3.15436242,  3.22147651,  3.2885906 ,
        3.3557047 ,  3.42281879,  3.48993289,  3.55704698,  3.62416107,
        3.69127517,  3.75838926,  3.82550336,  3.89261745,  3.95973154,
        4.02684564,  4.09395973,  4.16107383,  4.22818792,  4.29530201,
        4.36241611,  4.4295302 ,  4.4966443 ,  4.56375839,  4.63087248,
        4.69798658,  4.76510067,  4.83221477,  4.89932886,  4.96644295,
        5.03355705,  5.10067114,  5.16778523,  5.23489933,  5.30201342,
        5.36912752,  5.43624161,  5.5033557 ,  5.5704698 ,  5.63758389,
        5.70469799,  5.77181208,  5.83892617,  5.90604027,  5.97315436,
        6.04026846,  6.10738255,  6.17449664,  6.24161074,  6.30872483,
        6.37583893,  6.44295302,  6.51006711,  6.57718121,  6.6442953 ,
        6.7114094 ,  6.77852349,  6.84563758,  6.91275168,  6.97986577,
        7.04697987,  7.11409396,  7.18120805,  7.24832215,  7.31543624,
        7.38255034,  7.44966443,  7.51677852,  7.58389262,  7.65100671,
        7.71812081,  7.7852349 ,  7.85234899,  7.91946309,  7.98657718,
        8.05369128,  8.12080537,  8.18791946,  8.25503356,  8.32214765,
        8.38926174,  8.45637584,  8.52348993,  8.59060403,  8.65771812,
        8.72483221,  8.79194631,  8.8590604 ,  8.9261745 ,  8.99328859,
        9.06040268,  9.12751678,  9.19463087,  9.26174497,  9.32885906,
        9.39597315,  9.46308725,  9.53020134,  9.59731544,  9.66442953,
        9.73154362,  9.79865772,  9.86577181,  9.93288591, 10.        ])
```

In [7]:
```
fig=plt.figure()  # size
```

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

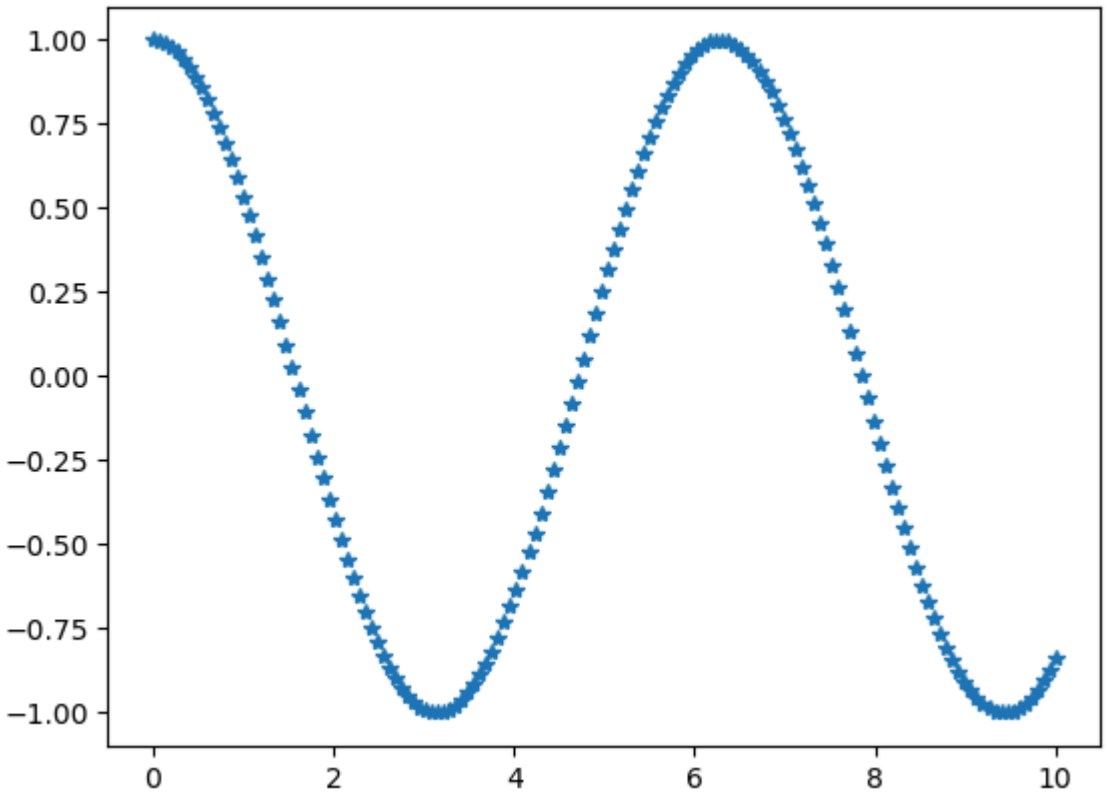
In [8]:
```
plt.plot(x1,np.sin(x1),  '_')
```

Out[8]:
```
[<matplotlib.lines.Line2D at 0x1b61fa11850>]
```



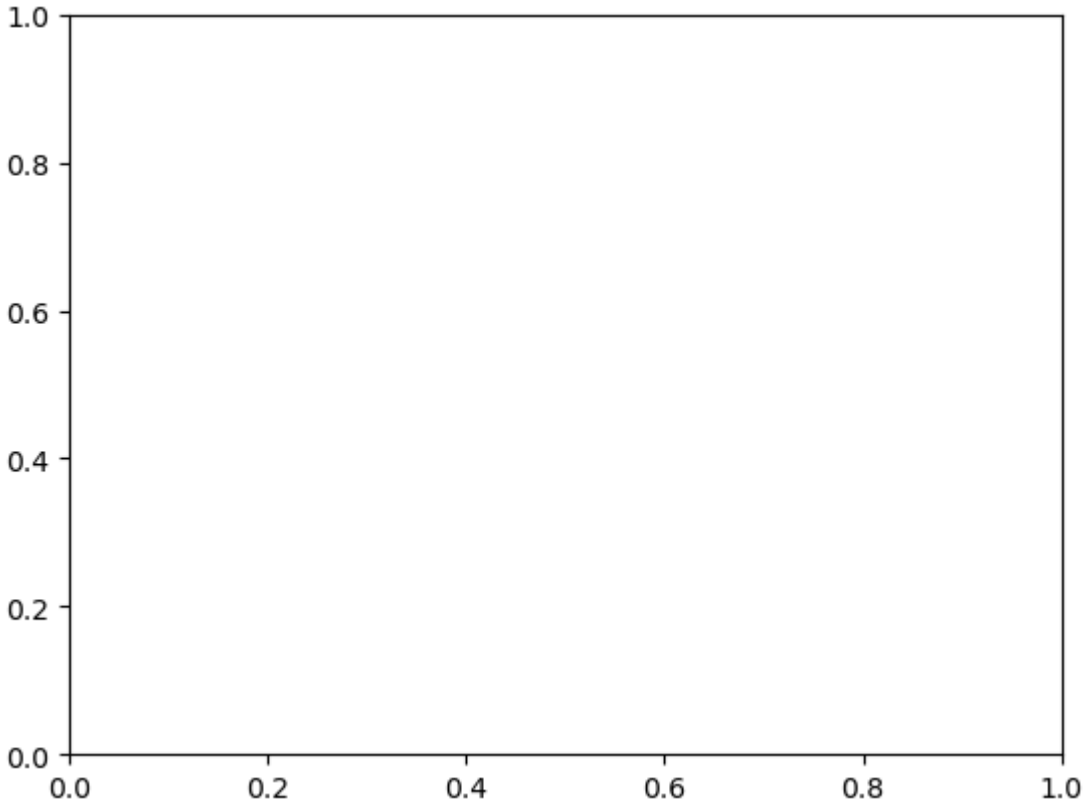

In [10]:
```
plt.plot(x1,np.cos(x1),'*')
```

Out[10]:
```
[<matplotlib.lines.Line2D at 0x1b620724350>]
```




In [11]:
```
plt.gca()
```

Out[11]: <Axes: >




In [12]:
```
plt.gcf()
```

Out[12]: <Figure size 640x480 with 0 Axes>

<Figure size 640x480 with 0 Axes>

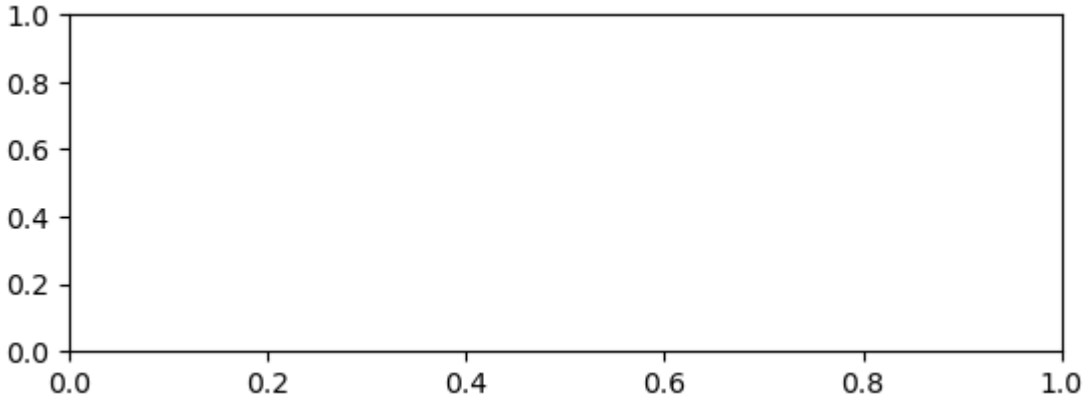
In [13]:
```
plt.figure()
```

Out[13]: <Figure size 640x480 with 0 Axes>

<Figure size 640x480 with 0 Axes>

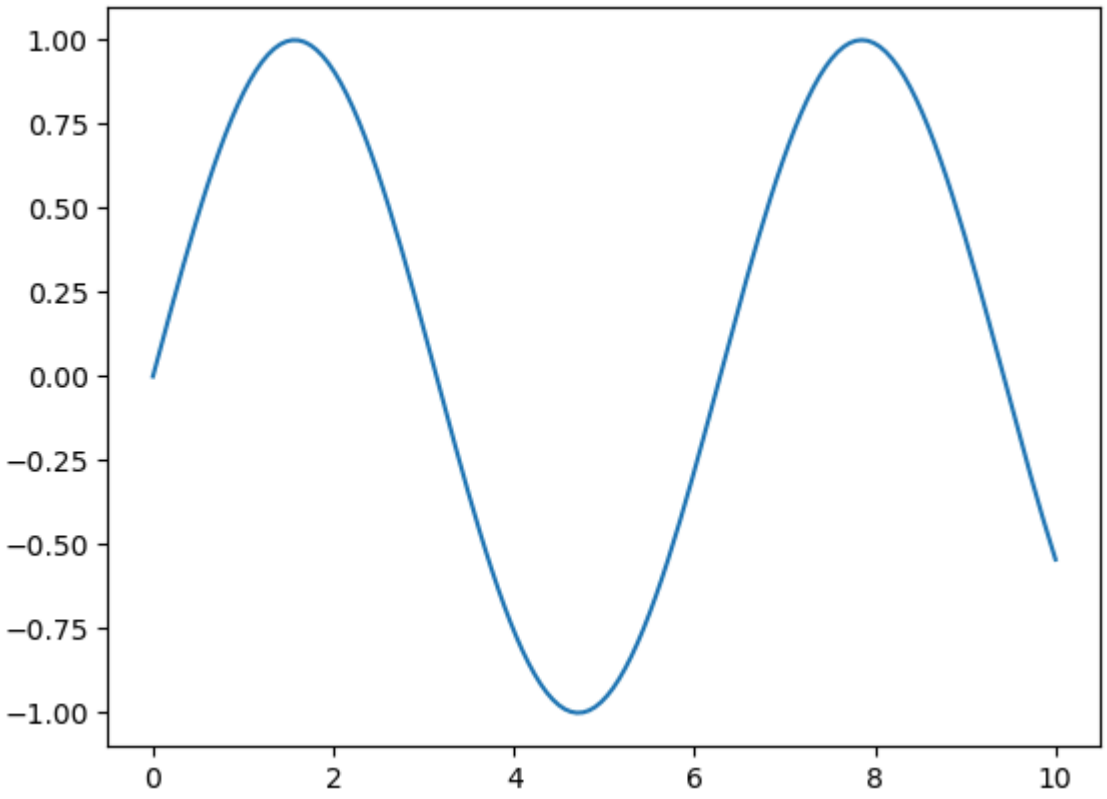
In [15]:
```
plt.subplot(2,1,1)
```

Out[15]: <Axes: >



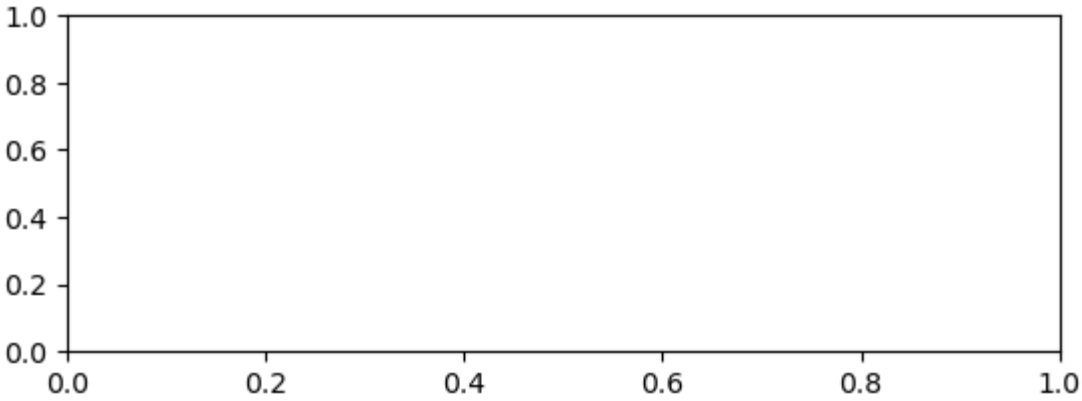
In [16]:
```
plt.plot(x1,np.sin(x1))
```

Out[16]:
```
[<matplotlib.lines.Line2D at 0x1b6232a3d90>]
```



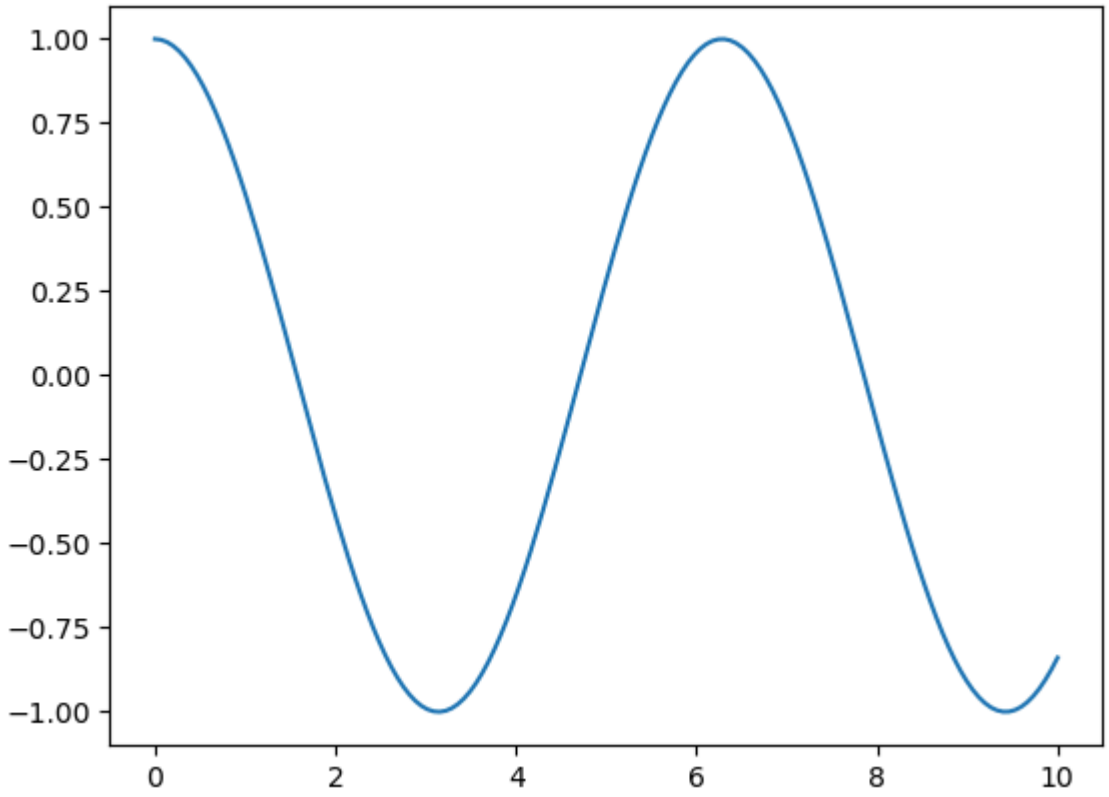

In [17]:
```
plt.subplot(2,1,2)
```

Out[17]:
```
<Axes: >
```

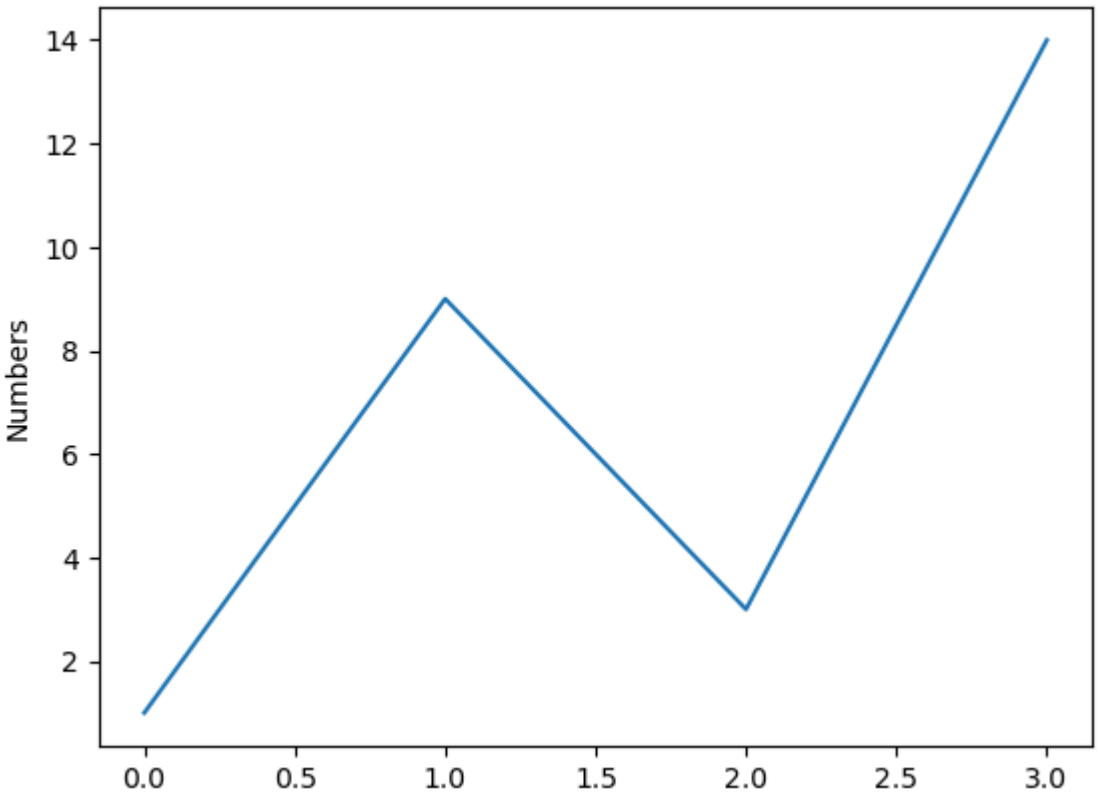



In [18]:
```
plt.plot(x1,np.cos(x1))
```

Out[18]:
```
[<matplotlib.lines.Line2D at 0x1b6233e7490>]
```

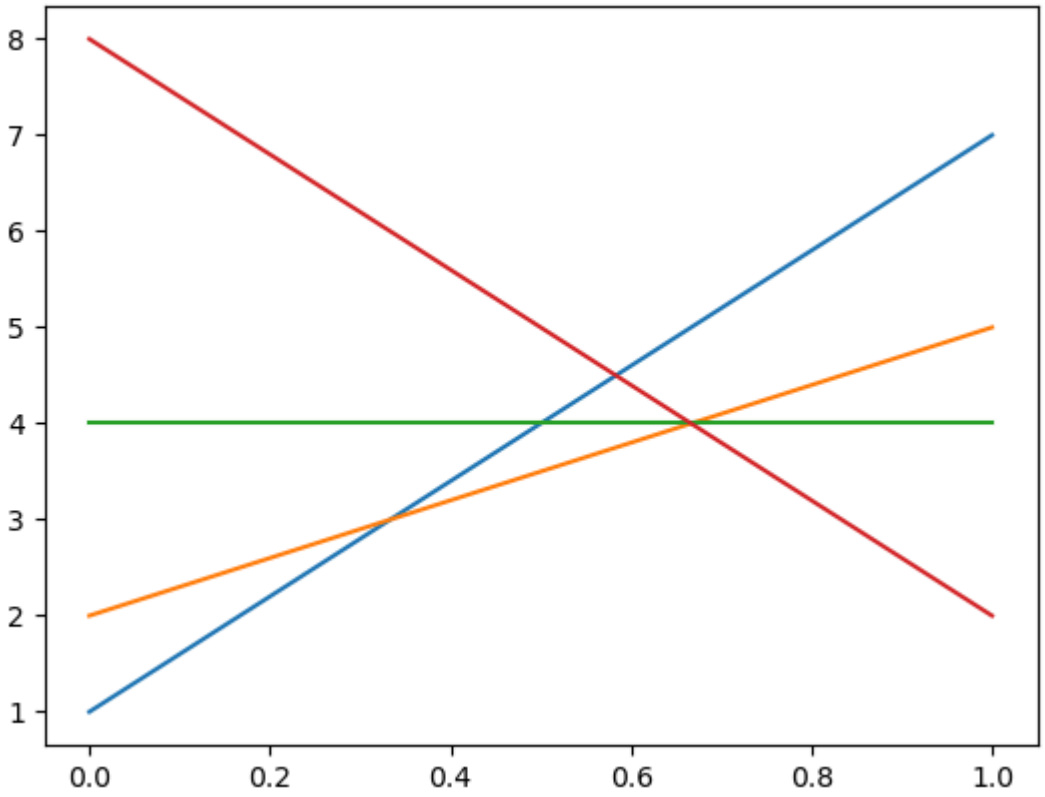



In [19]:
```
plt.plot([1, 9, 3, 14])
plt.ylabel('Numbers')
plt.show()
```

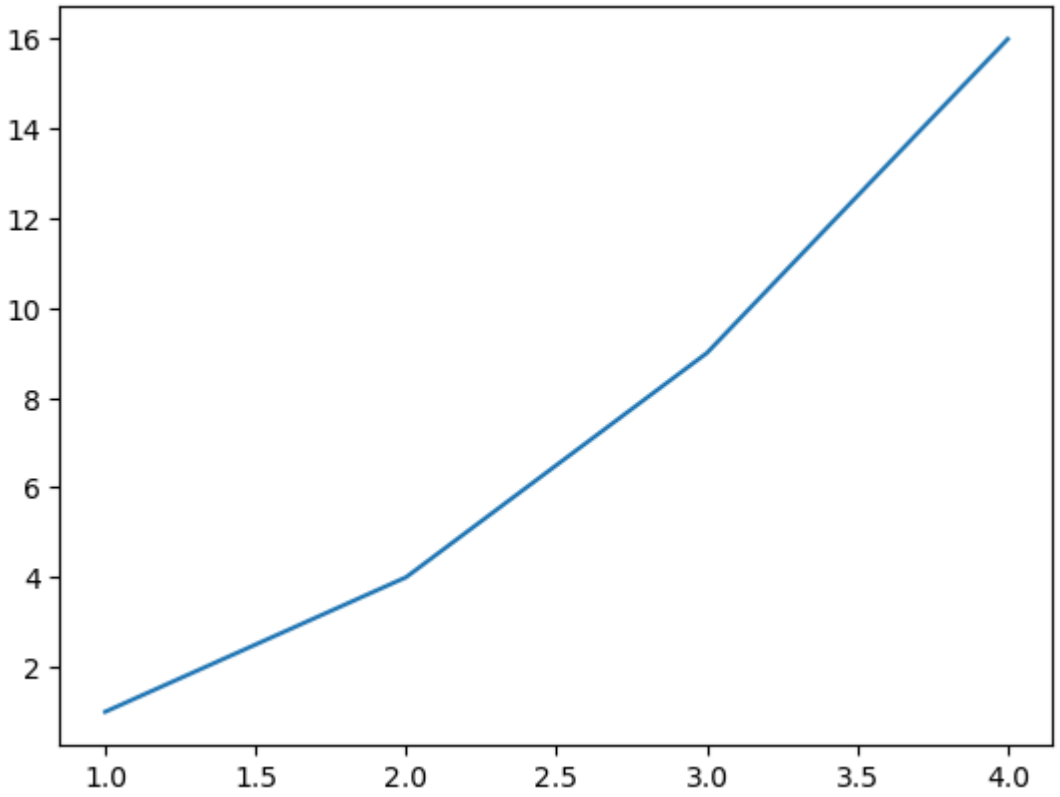



In [20]:
```
plt.plot([[1,2,4,8],[7,5,4,2]])
```

Out[20]:
```
[<matplotlib.lines.Line2D at 0x1b6237cb950>,
 <matplotlib.lines.Line2D at 0x1b6235e3d50>,
 <matplotlib.lines.Line2D at 0x1b6237cbfd0>,
 <matplotlib.lines.Line2D at 0x1b6237d8250>]
```




In [21]:
```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
plt.show()
```

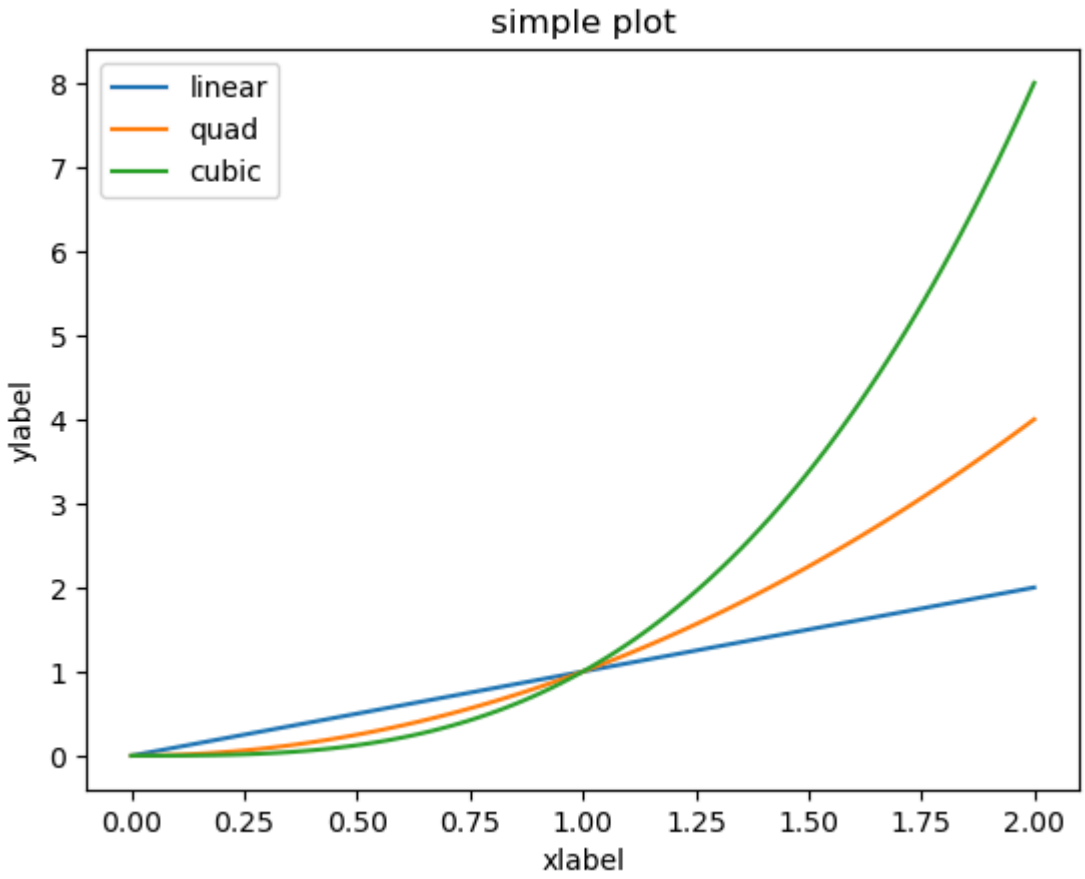



In [27]:
```
x=np.linspace(0,2,100)
x
```

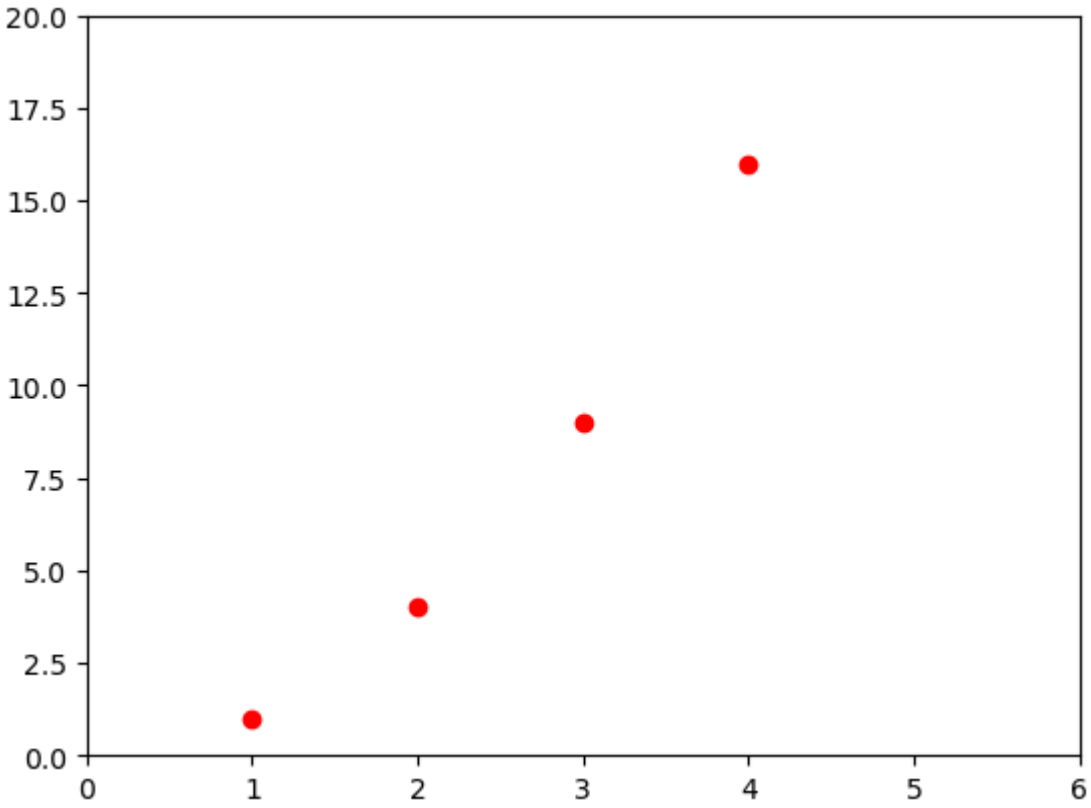
Out[27]:
```
array([0.        , 0.02020202, 0.04040404, 0.06060606, 0.08080808,
       0.1010101 , 0.12121212, 0.14141414, 0.16161616, 0.18181818,
       0.2020202 , 0.22222222, 0.24242424, 0.26262626, 0.28282828,
       0.3030303 , 0.32323232, 0.34343434, 0.36363636, 0.38383838,
       0.4040404 , 0.42424242, 0.44444444, 0.46464646, 0.48484848,
       0.50505051, 0.52525253, 0.54545455, 0.56565657, 0.58585859,
       0.60606061, 0.62626263, 0.64646465, 0.66666667, 0.68686869,
       0.70707071, 0.72727273, 0.74747475, 0.76767677, 0.78787879,
       0.80808081, 0.82828283, 0.84848485, 0.86868687, 0.88888889,
       0.90909091, 0.92929293, 0.94949495, 0.96969697, 0.98989899,
       1.01010101, 1.03030303, 1.05050505, 1.07070707, 1.09090909,
       1.11111111, 1.13131313, 1.15151515, 1.17171717, 1.19191919,
       1.21212121, 1.23232323, 1.25252525, 1.27272727, 1.29292929,
       1.31313131, 1.33333333, 1.35353535, 1.37373737, 1.39393939,
       1.41414141, 1.43434343, 1.45454545, 1.47474747, 1.49494949,
       1.51515152, 1.53535354, 1.55555556, 1.57575758, 1.5959596 ,
       1.61616162, 1.63636364, 1.65656566, 1.67676768, 1.6969697 ,
       1.71717172, 1.73737374, 1.75757576, 1.77777778, 1.7979798 ,
       1.81818182, 1.83838384, 1.85858586, 1.87878788, 1.8989899 ,
       1.91919192, 1.93939394, 1.95959596, 1.97979798, 2.        ])
```

In [26]:
```
plt.plot(x,x,label='linear')
plt.plot(x,x**2,label='quad')
plt.plot(x,x**3,label='cubic')
plt.xlabel('xlabel')
plt.ylabel('ylabel')
plt.title('simple plot')
```

```
plt.legend()
plt.show()
```




In [28]:
```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```

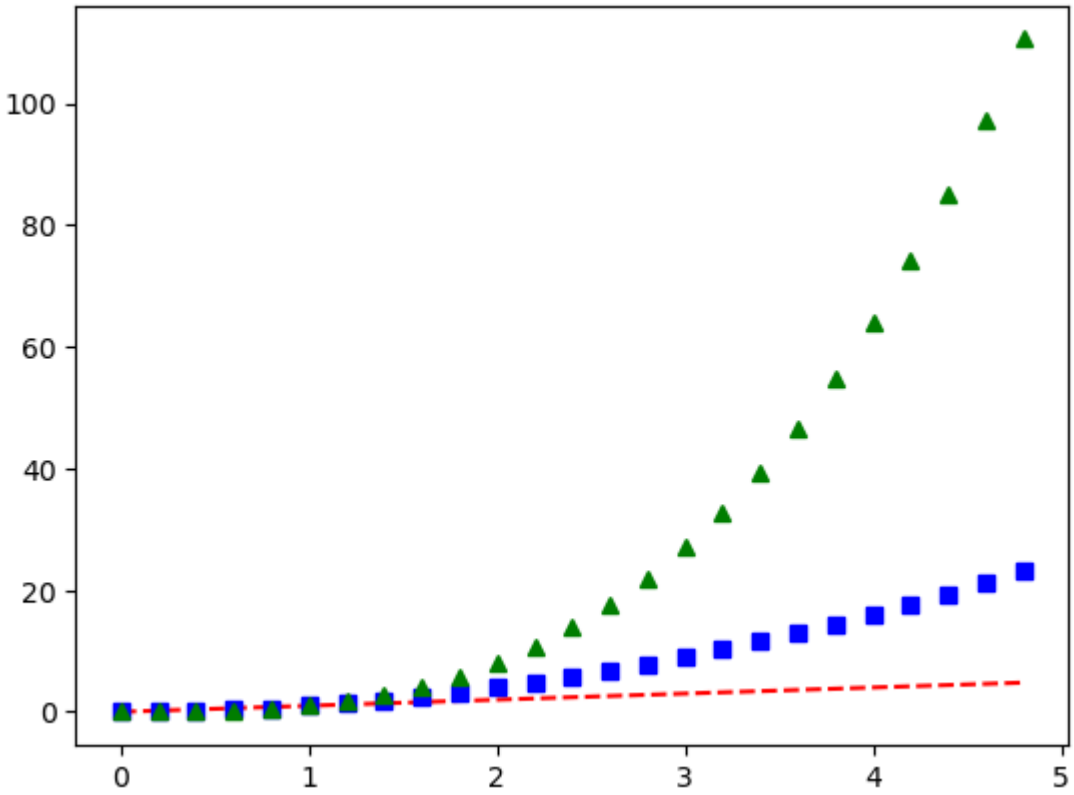



In [29]:
```
t = np.arange(0., 5., 0.2)   # working
```

In [30]:
```
t
```

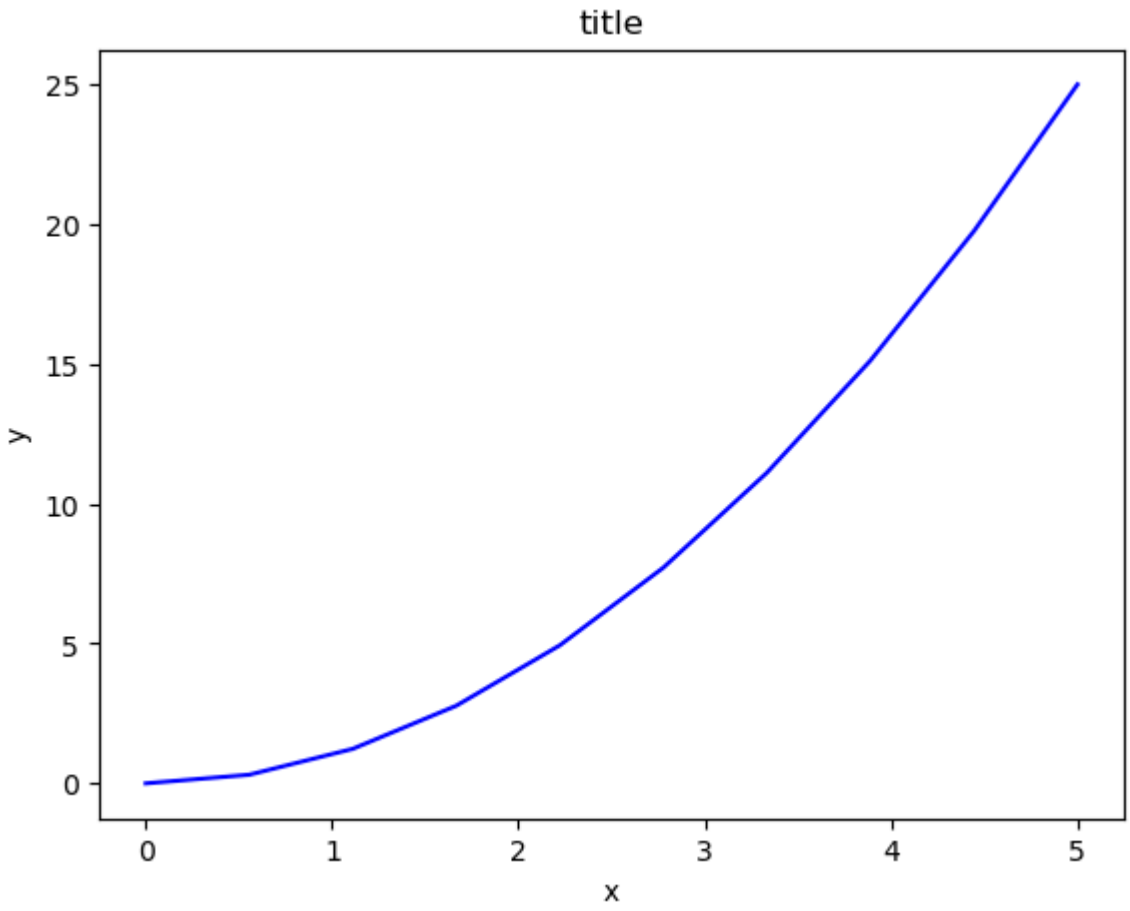
Out[30]:
```
array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4,
       2.6, 2.8, 3. , 3.2, 3.4, 3.6, 3.8, 4. , 4.2, 4.4, 4.6, 4.8])
```

In [31]:
```
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```




In [11]:
```
# object by reference methos
import matplotlib.pyplot as plt
import numpy as np
fig=plt.figure()  # store refer in fig
x=np.linspace(0,5,10)    # internally create array generate   10 numbers from 0 to
y=x**2
axes=fig.add_axes([0.1,0.1,0.8,0.8])
axes.plot(x,y,'b')
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.set_title('title')
```

Out[11]:
```
Text(0.5, 1.0, 'title')
```



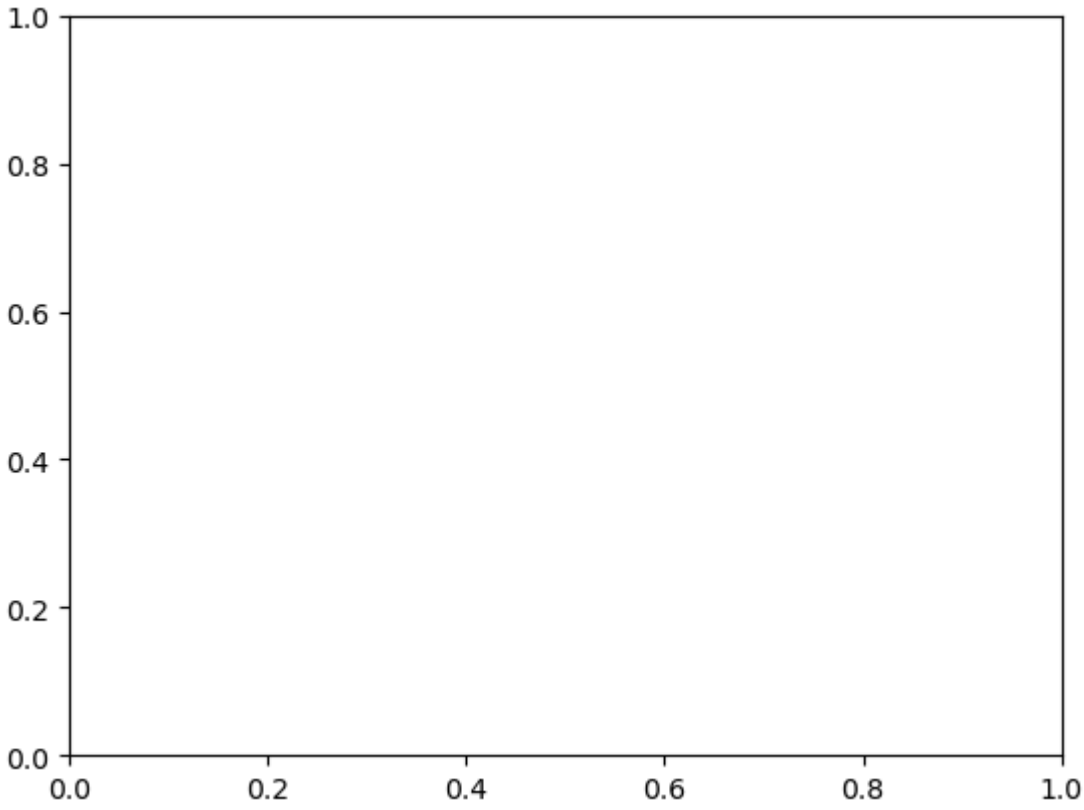

In [7]:
```
x
```

Out[7]:
```
array([0.        , 0.55555556, 1.11111111, 1.66666667, 2.22222222,
       2.77777778, 3.33333333, 3.88888889, 4.44444444, 5.        ])
```

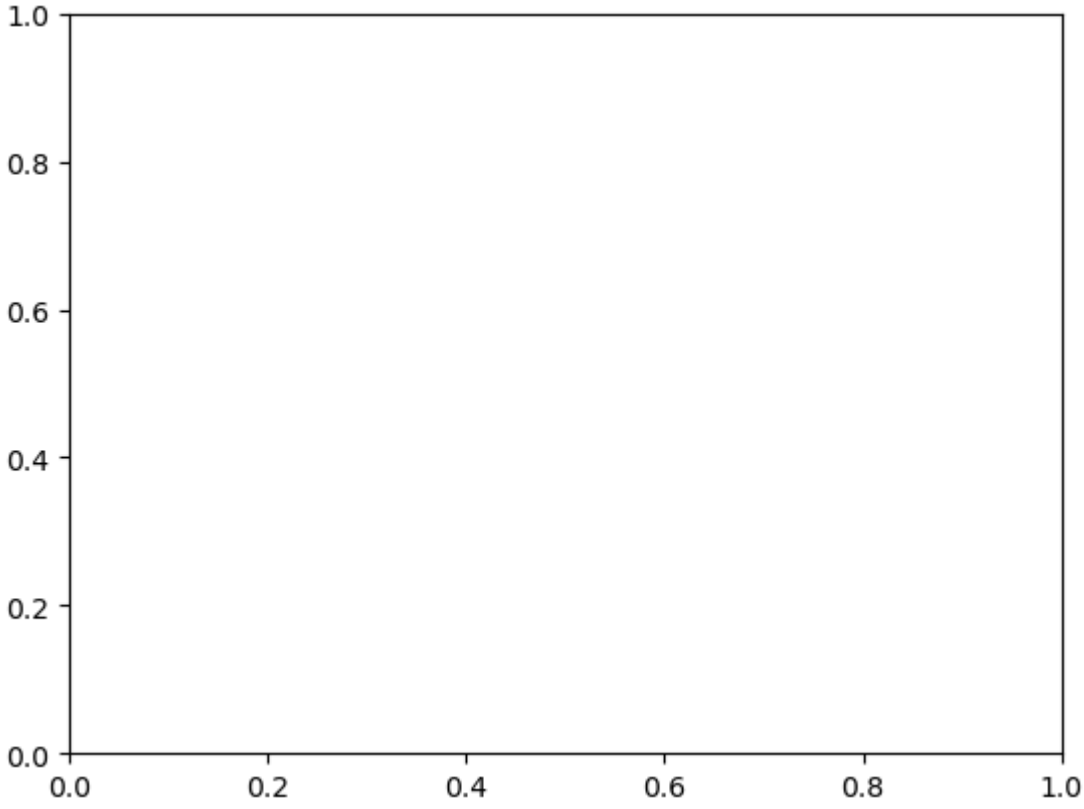
In [2]:
```
import matplotlib.pyplot as plt
```

In [3]:
```
import numpy as np
```

In [4]:
```
fig = plt.figure()

ax = plt.axes()
```

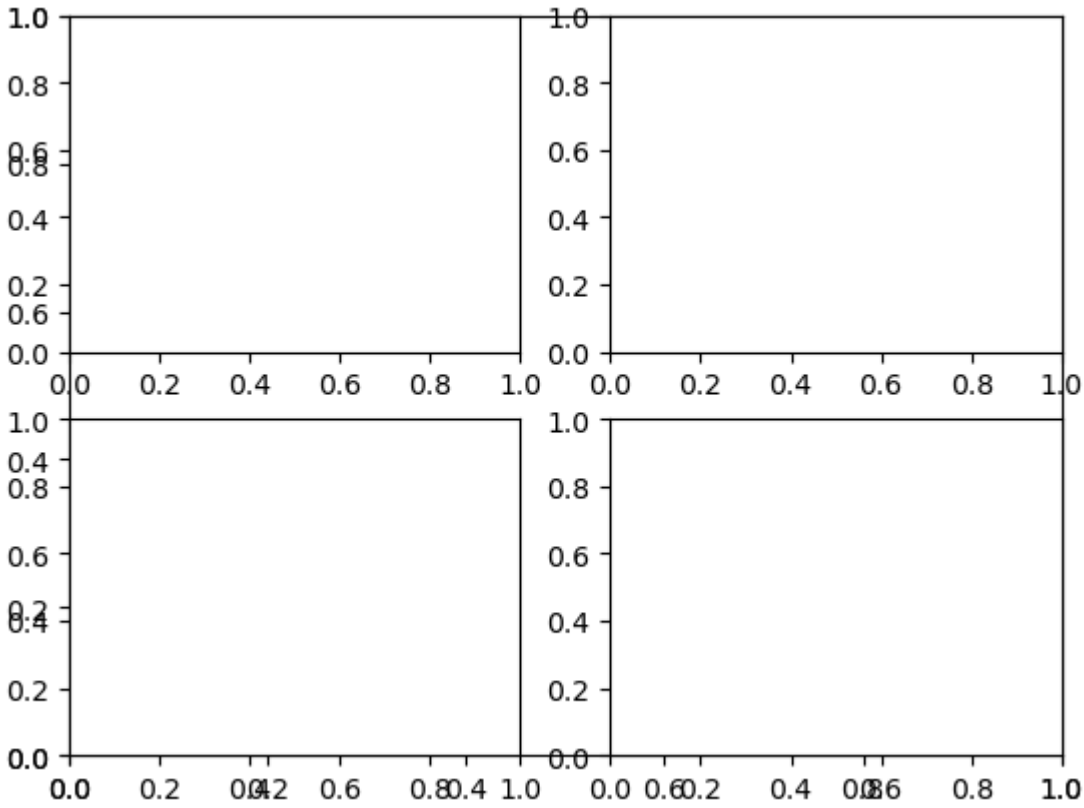



In [5]:
```
fig=plt.figure()
ax=plt.axes()
```

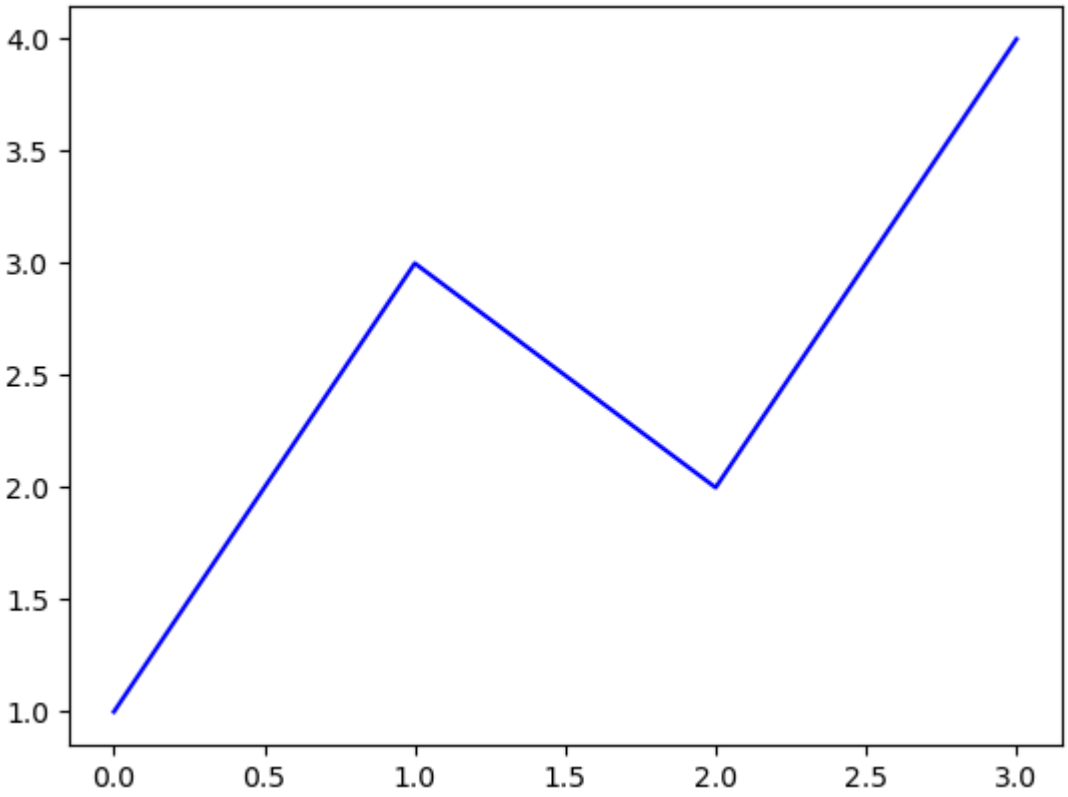



In [7]:
```
fig=plt.figure()
ax=plt.axes()
```

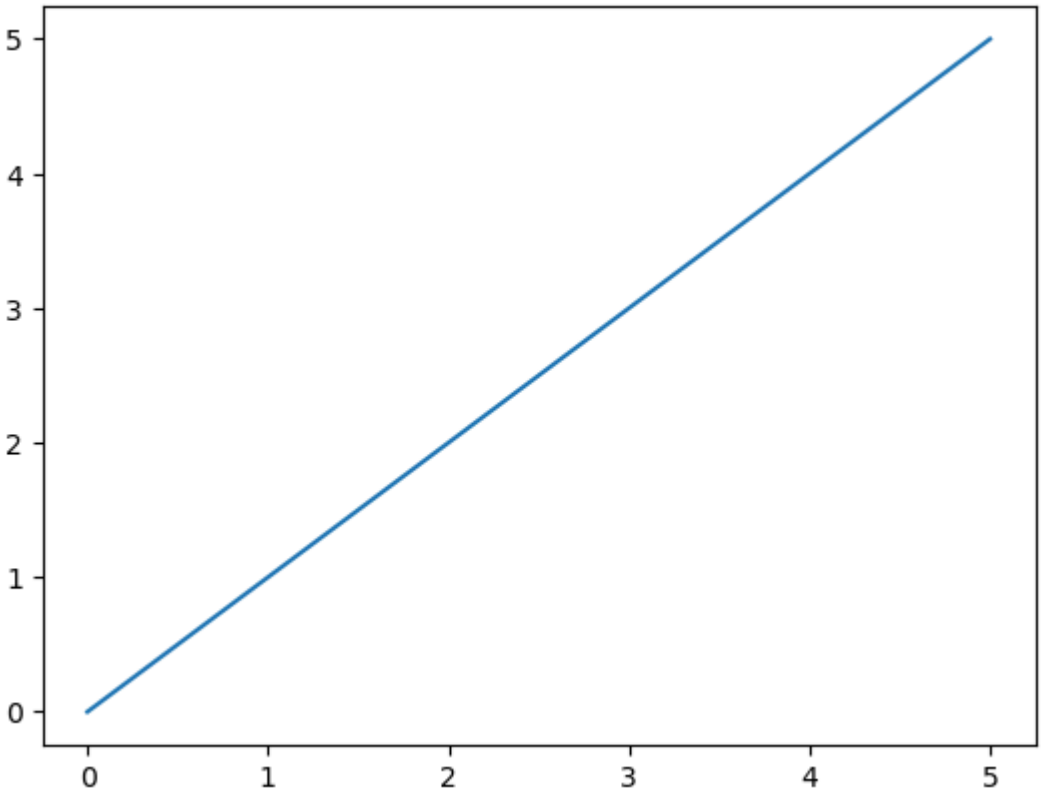
```
ax1=fig.add_subplot(2,2,1)
ax2 = fig.add_subplot(2, 2, 2)

ax3 = fig.add_subplot(2, 2, 3)

ax4 = fig.add_subplot(2, 2, 4)
```



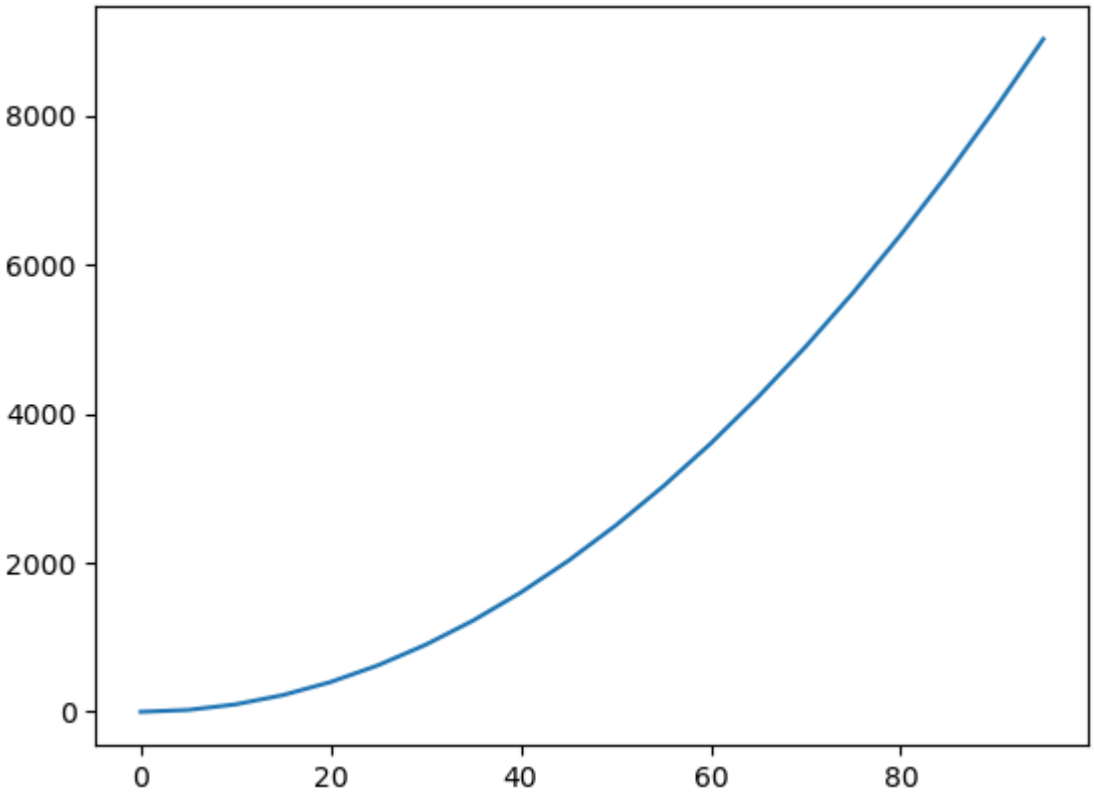
In [10]:
```
#First plot with Matplotlib
plt.plot([1, 3, 2, 4], 'b')

plt.show( )
```




In [24]:
```
x3 = range(6)

plt.plot(x3, [xi for xi in x3])  # both lists

plt.show()
```



In [25]:
```
x=np.arange(0,100,5)
plt.plot(x,[xi**2 for xi in x])
plt.show()
```

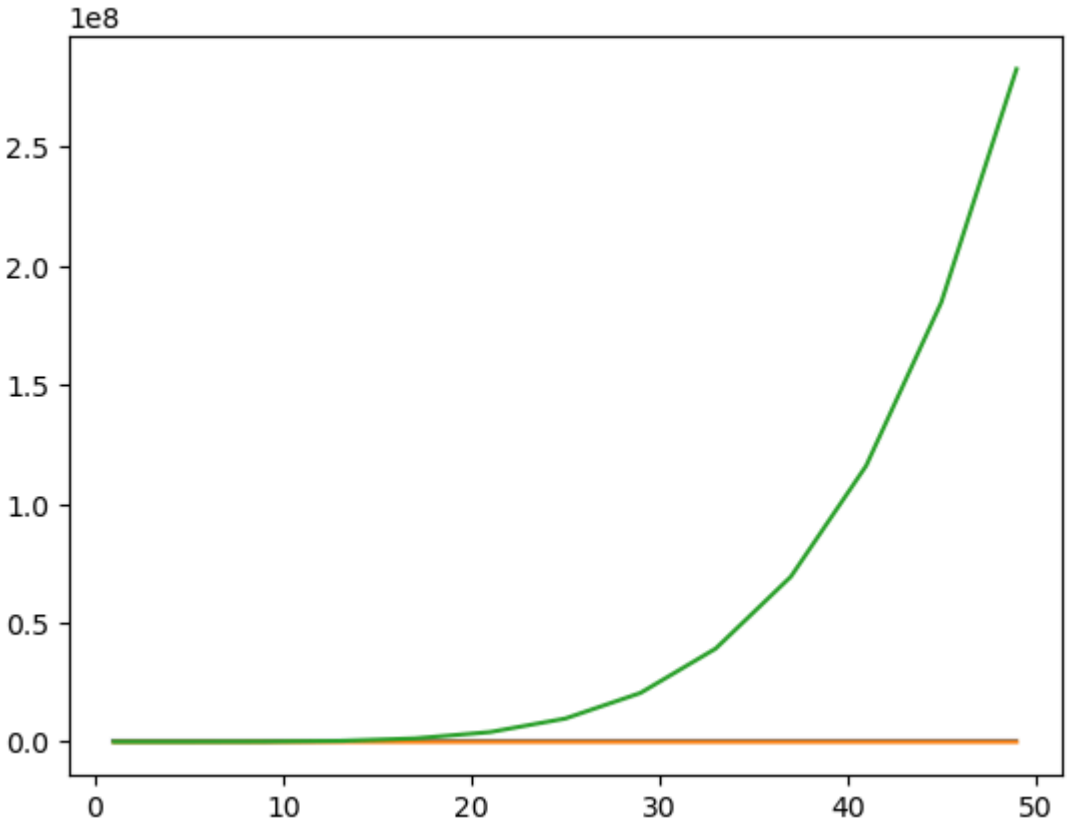



In [26]:
```
# multi line plot
x=np.arange(1,50,4)
```

In [27]:
```
x
```

Out[27]:
```
array([ 1,  5,  9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49])
```

In [30]:
```
plt.plot(x,[xi/2 for xi in x])
plt.plot(x,[xi**2 for  xi in x])
plt.plot(x,[xi**5 for xi in x])
plt.show()
```



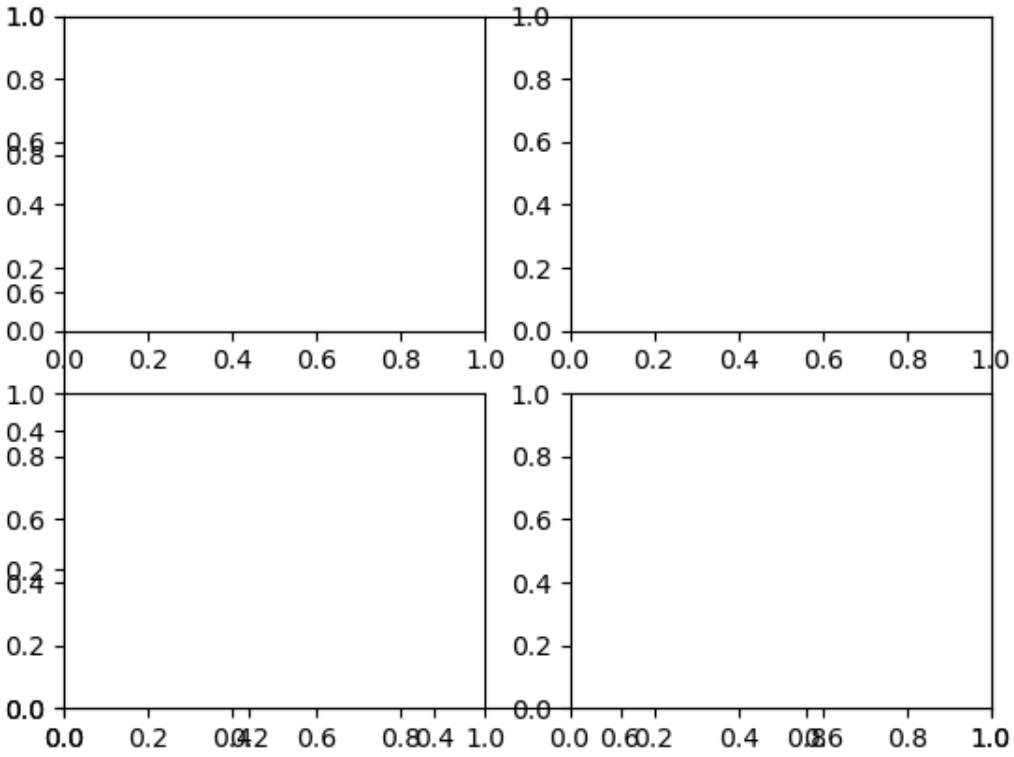

saving the plots

In [34]:
```
fig.savefig('flower.png')
```

In [35]:
```
from IPython.display import Image
```

In [36]:
```
Image('flower.png')
```

Out[36]:

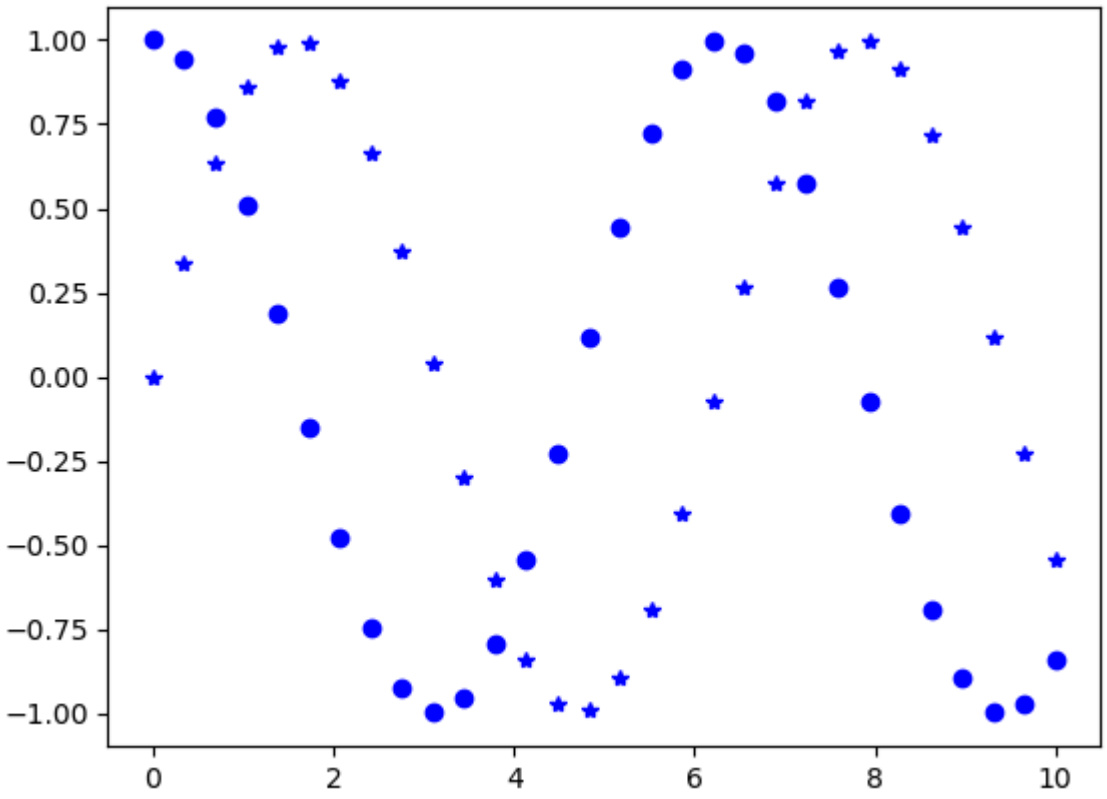


In [37]:
```
# Explore supported file formats


fig.canvas.get_supported_filetypes()
```

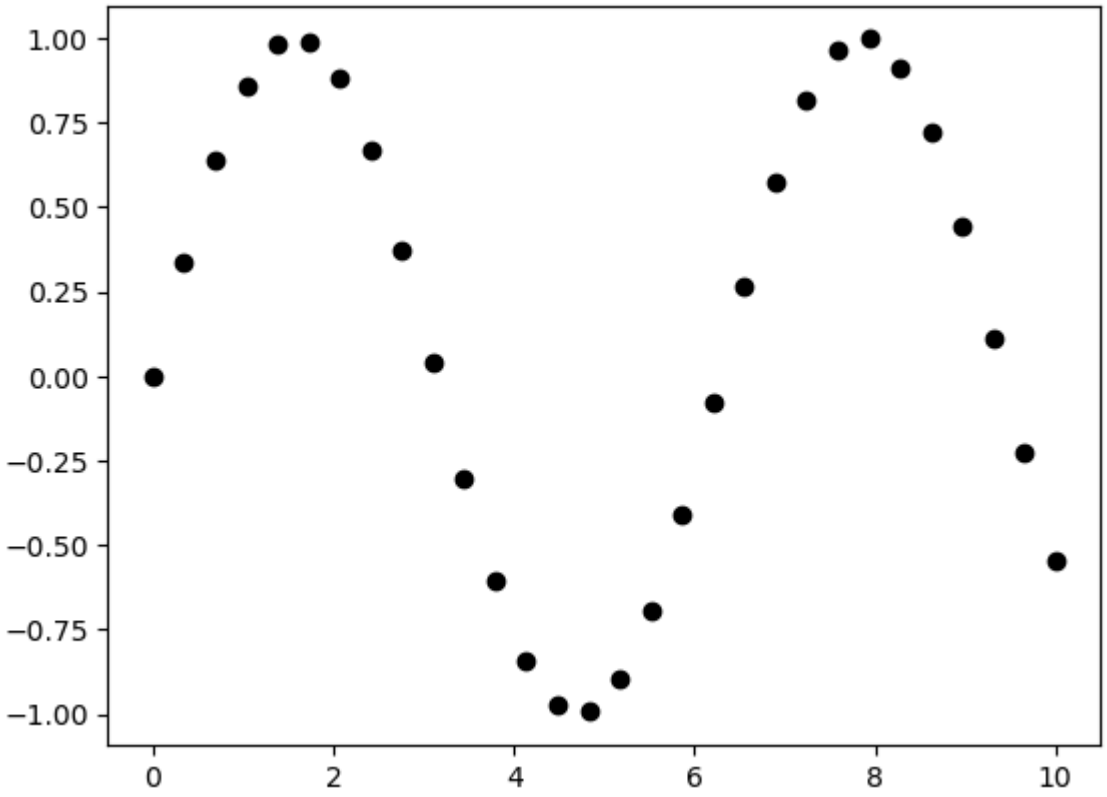
Out[37]:
```
{'eps': 'Encapsulated Postscript',
 'jpg': 'Joint Photographic Experts Group',
 'jpeg': 'Joint Photographic Experts Group',
 'pdf': 'Portable Document Format',
 'pgf': 'PGF code for LaTeX',
 'png': 'Portable Network Graphics',
 'ps': 'Postscript',
 'raw': 'Raw RGBA bitmap',
 'rgba': 'Raw 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 [40]:
```
 #Line Plot
fig=plt.figure()
ax=plt.axes()
x5=np.linspace(0,10,30)
x5
ax.plot(x5,np.sin(x5),'b*')
ax.plot(x5,np.cos(x5),'bo')
```

Out[40]: [<matplotlib.lines.Line2D at 0x29711fe57d0>]




In [41]:
```
x7 = np.linspace(0, 10, 30)

y7 = np.sin(x7)

plt.plot(x7, y7, 'o', color = 'black');
```



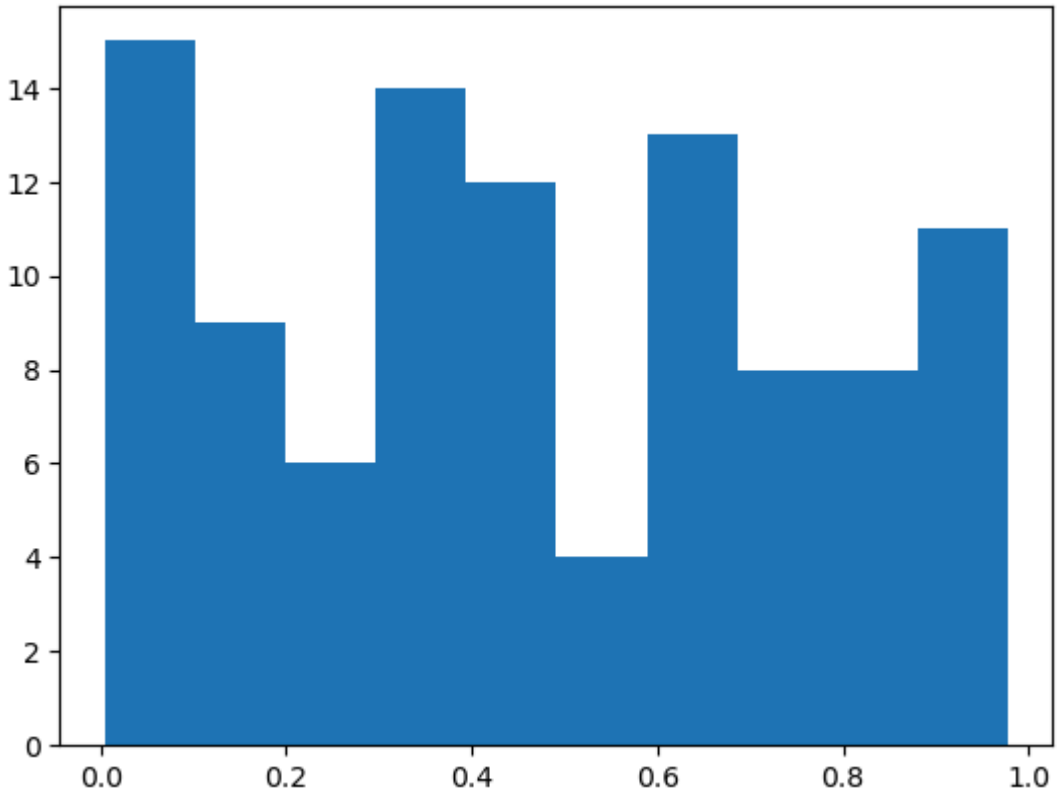

In [50]:
```
x=np.random.rand(100)
```

In [51]:
```
x
```

Out[51]:
```
array([0.56257818, 0.00483951, 0.04385442, 0.43584643, 0.51701122,
       0.35492312, 0.42710522, 0.3100203 , 0.67851979, 0.31773024,
       0.78910553, 0.81310436, 0.16327953, 0.03034015, 0.59496682,
       0.6336203 , 0.88648972, 0.59111038, 0.58533401, 0.65153557,
       0.00506805, 0.24273204, 0.12364865, 0.3765395 , 0.13081211,
       0.94453579, 0.09389532, 0.16660581, 0.48704965, 0.06946834,
       0.26569379, 0.40753336, 0.04276543, 0.09551539, 0.79192364,
       0.63152064, 0.69730969, 0.12501725, 0.30062034, 0.03485365,
       0.97741815, 0.25034883, 0.21584822, 0.11986076, 0.71112784,
       0.07352696, 0.35635239, 0.91648942, 0.07818289, 0.44489129,
       0.78682686, 0.9529792 , 0.69309252, 0.88854834, 0.40337226,
       0.58980929, 0.48444768, 0.54621791, 0.38060525, 0.63497901,
       0.3417195 , 0.63780105, 0.95660729, 0.7310371 , 0.7955125 ,
       0.69245665, 0.3465953 , 0.79312045, 0.79120965, 0.19552354,
       0.02929914, 0.38472619, 0.26704511, 0.74113267, 0.8365103 ,
       0.34469358, 0.95167612, 0.36054837, 0.40393563, 0.40307829,
       0.66571338, 0.93703061, 0.7036194 , 0.62798611, 0.40296591,
       0.65982897, 0.02212695, 0.62047436, 0.35118317, 0.97389008,
       0.01169165, 0.46608786, 0.9739859 , 0.4304517 , 0.1578409 ,
       0.36014814, 0.27794254, 0.70782535, 0.05075391, 0.19092567])
```

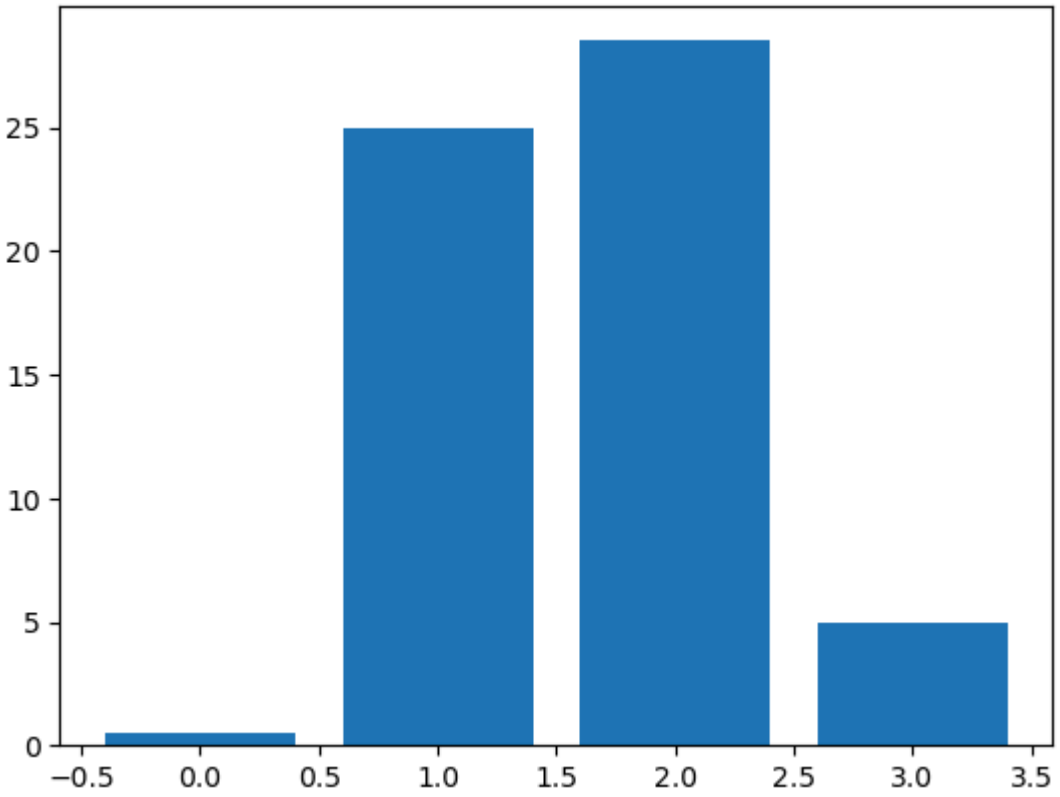
In [52]:
```
plt.hist(x)
```

Out[52]:
```
(array([15.,  9.,  6., 14., 12.,  4., 13.,  8.,  8., 11.]),
 array([0.00483951, 0.10209737, 0.19935524, 0.2966131 , 0.39387096,
        0.49112883, 0.58838669, 0.68564456, 0.78290242, 0.88016028,
        0.97741815]),
 <BarContainer object of 10 artists>)
```

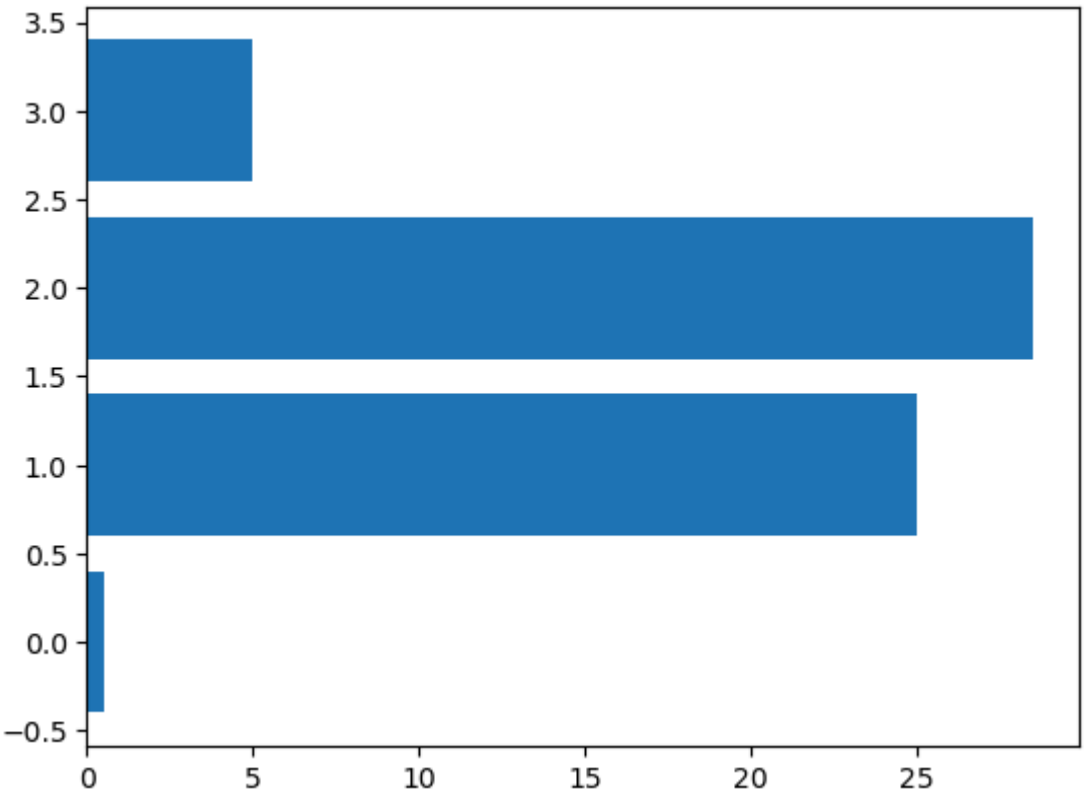



In [53]:
```
data=[.5,25.,28.5,5.]
```

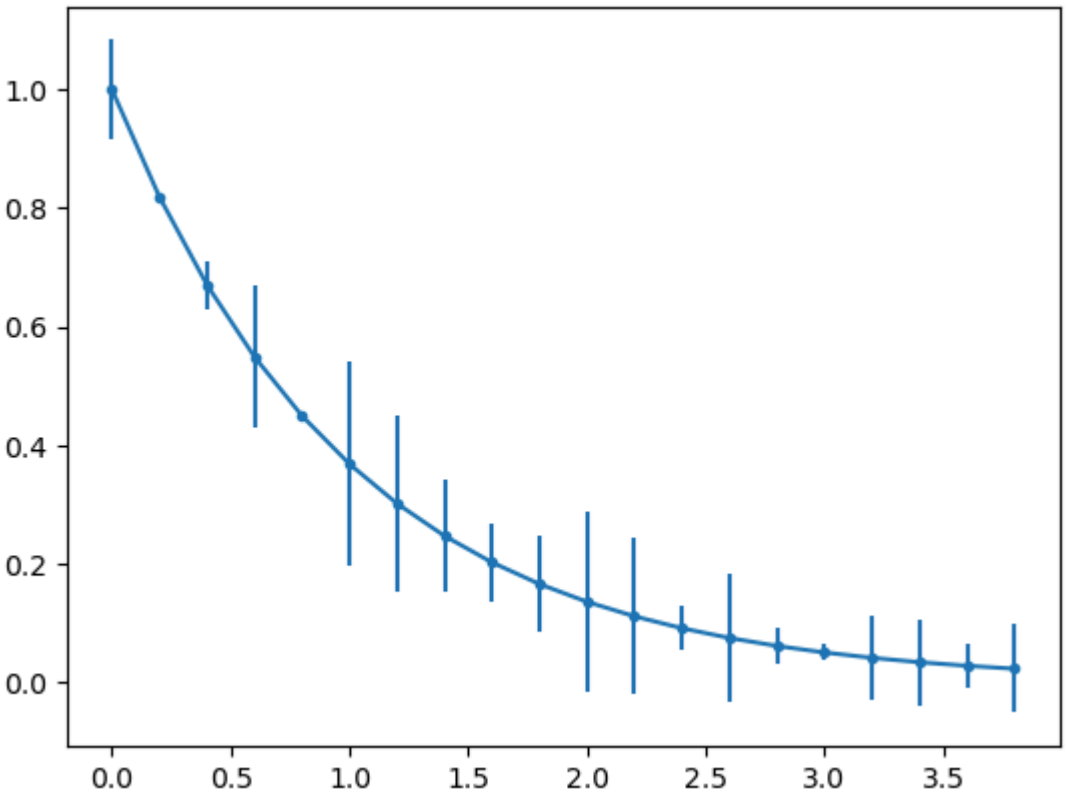
In [54]:
```
plt.bar(range(len(data)),data)
plt.show()
```



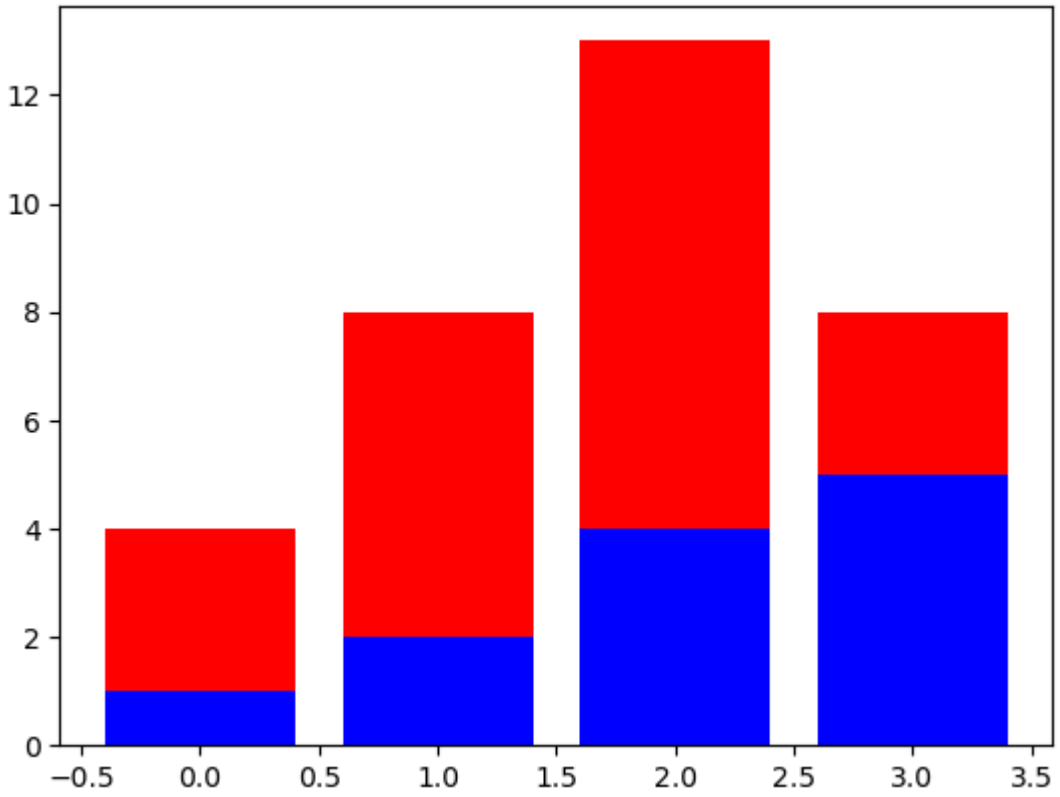

In [56]:
```
# horizontal bar chat
plt.barh(range(len(data)),data)
plt.show()
```



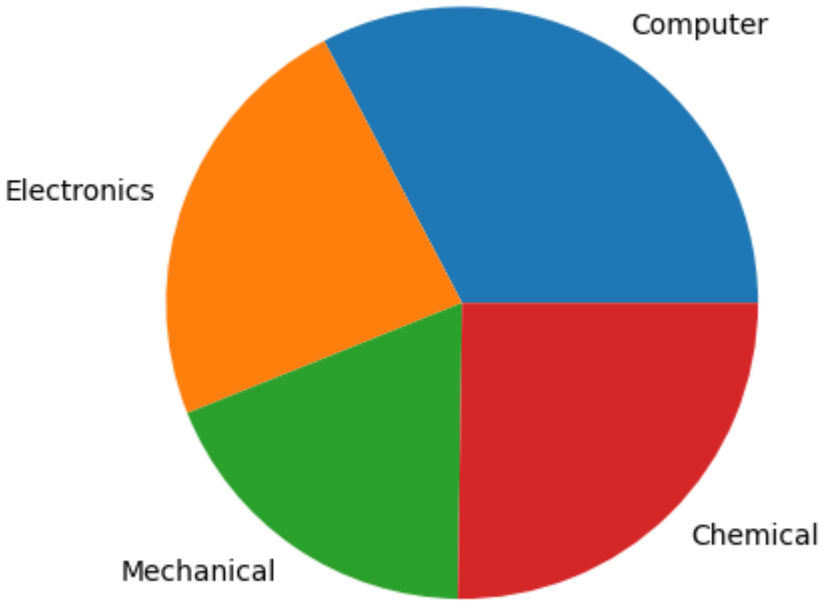
In [57]:
```
x9 = np.arange(0, 4, 0.2)   # error bar b

y9 = np.exp(-x9)

e1 = 0.1 * np.abs(np.random.randn(len(y9)))

plt.errorbar(x9, y9, yerr = e1, fmt = '.-')

plt.show();
```



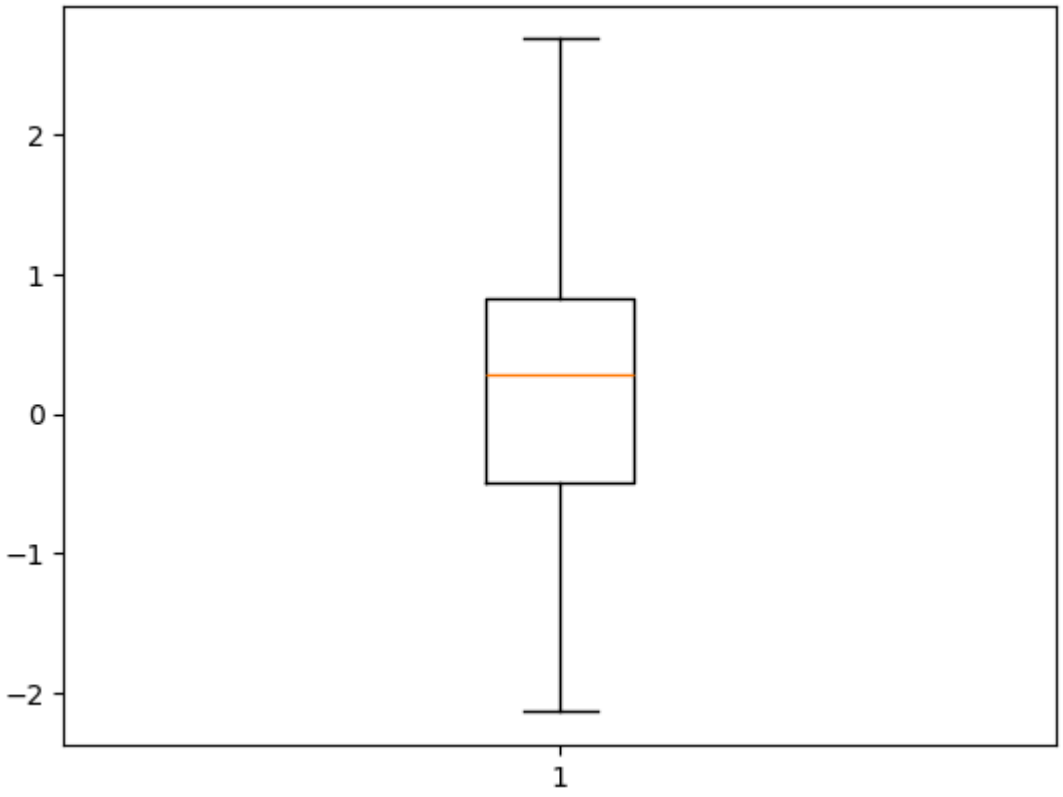

In [58]:
```
x=[1,2,4,5]
y=[3,6,9,3]
z=range(4)
plt.bar(z,x,color='b')
plt.bar(z,y,color='r',bottom=x)
plt.show()
```



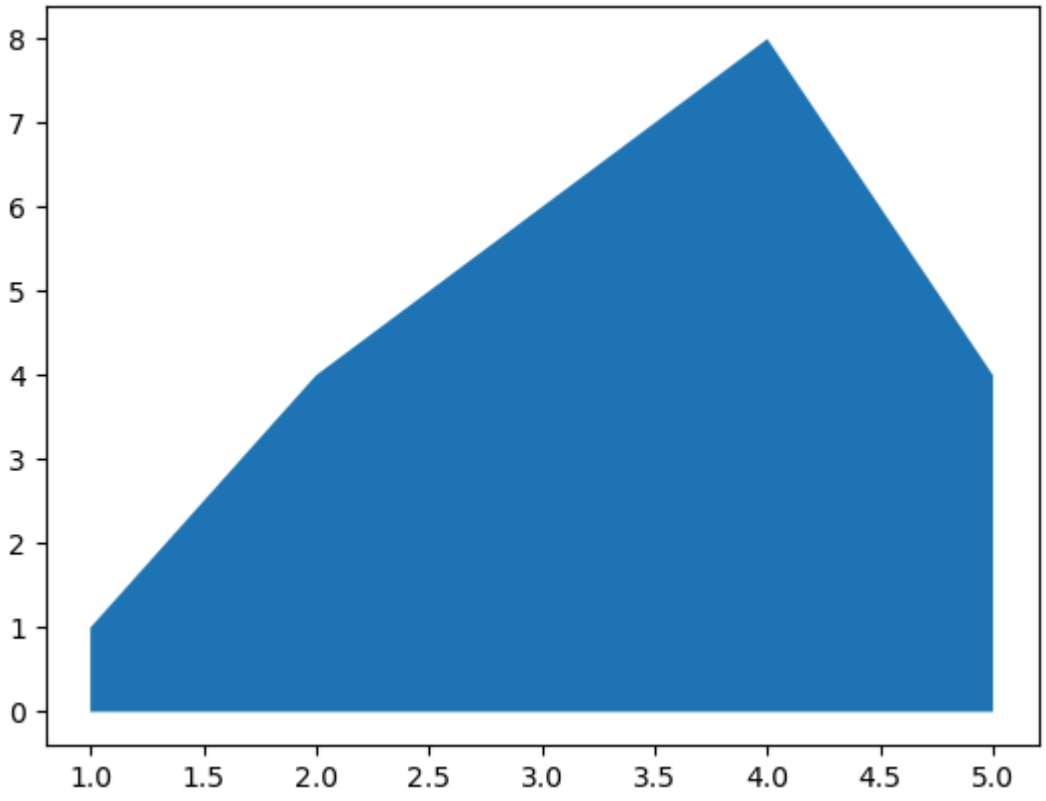

In [61]:
```
x10 = [35, 25, 20, 27]

labels = ['Computer', 'Electronics', 'Mechanical', 'Chemical']

plt.pie(x10, labels=labels);

plt.show()
```




In [62]:
```
data3 = np.random.randn(100)

plt.boxplot(data3)

plt.show();
```

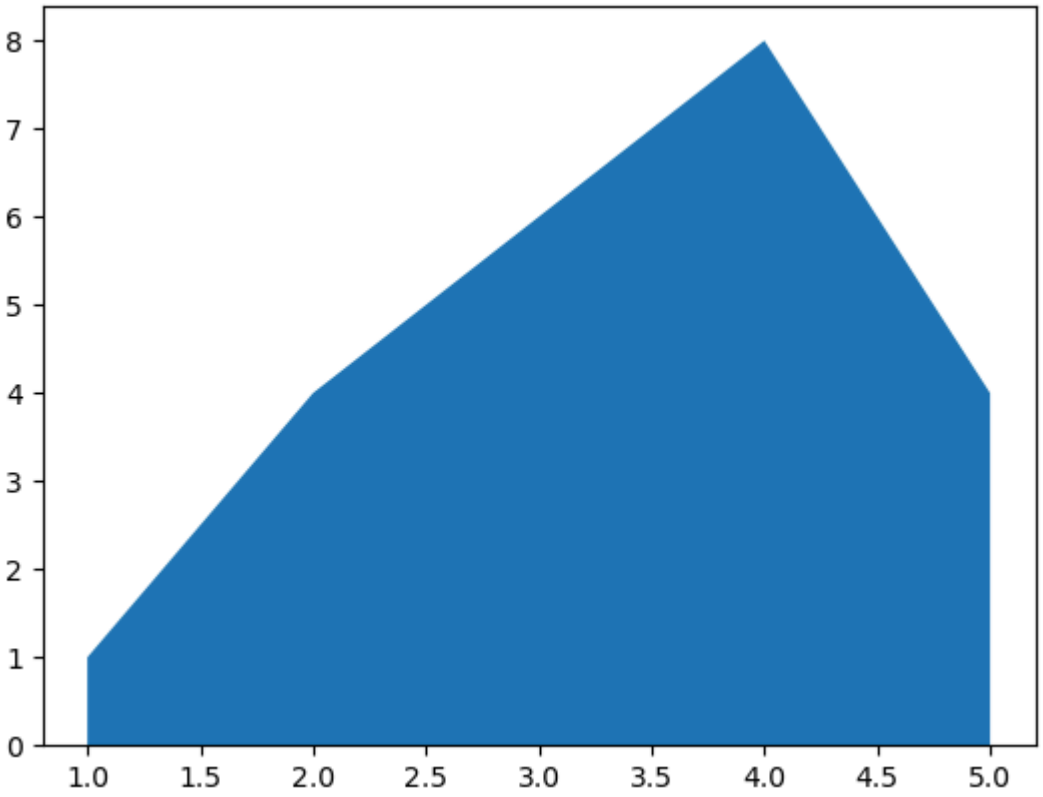


In [63]:
```
# area plot
x12 = range(1, 6)
y12 = [1, 4, 6, 8, 4]
plt.fill_between(x12, y12)
plt.show()
```




In [64]:
```
plt.stackplot(x12, y12)
```

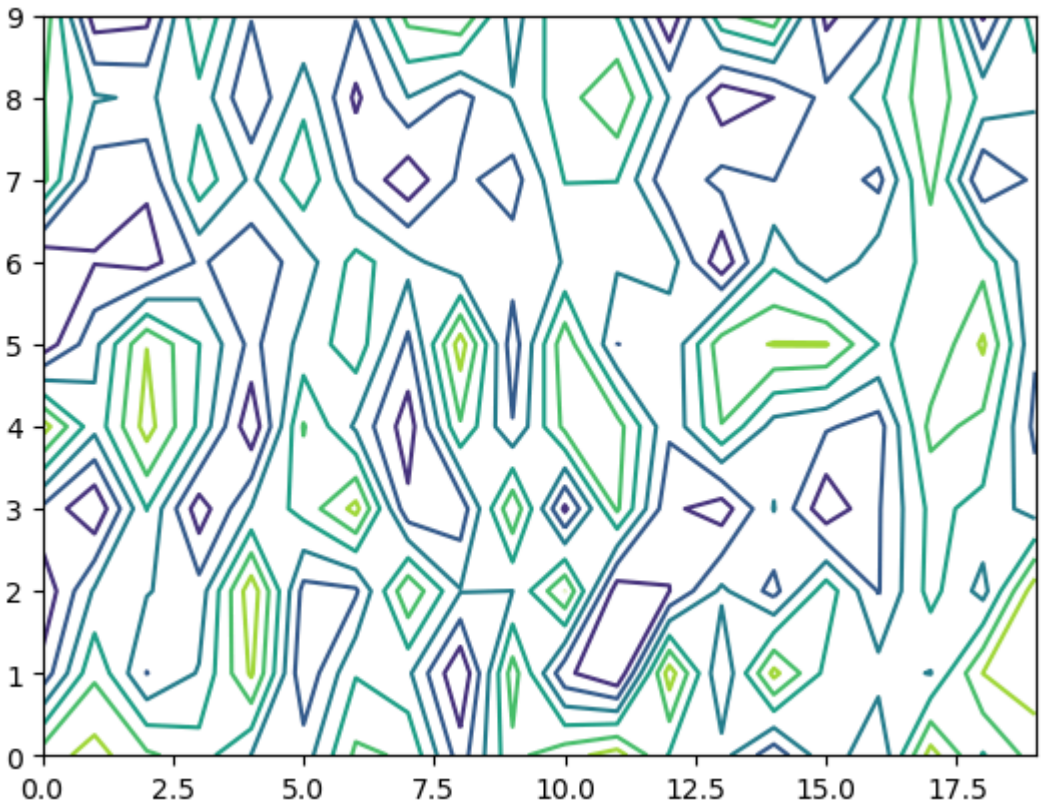
Out[64]:
```
[<matplotlib.collections.PolyCollection at 0x297121bdf50>]
```



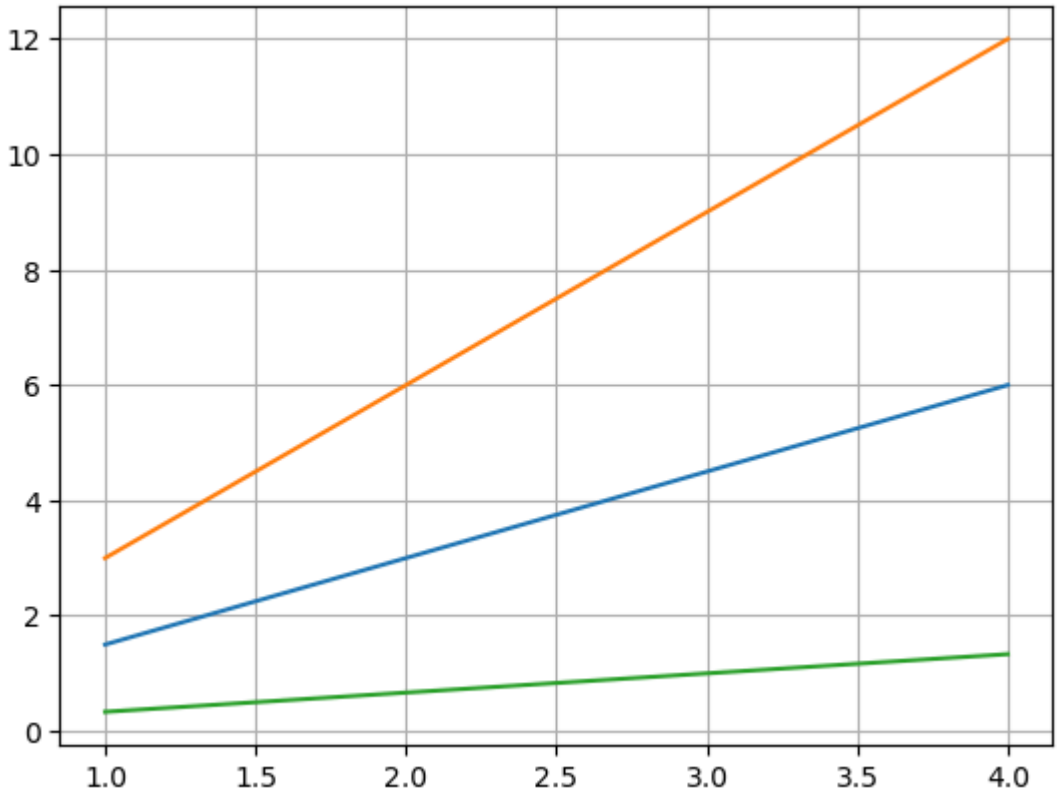

In [66]:
```
# contour plot
matrix1 = np.random.rand(10, 20)
matrix1
```

Out[66]:
```
array([[8.25756115e-01, 9.58677197e-01, 7.77689434e-01, 6.71112653e-01,
        4.43710751e-01, 3.73340566e-01, 7.81724877e-01, 7.29385945e-01,
        2.25247297e-01, 7.10803928e-01, 8.40604788e-01, 9.72242068e-01,
        5.85488124e-01, 5.02853970e-01, 1.98231762e-01, 5.53596645e-01,
        3.13183976e-01, 9.63429220e-01, 5.88301184e-01, 8.13556033e-01],
       [2.00144943e-01, 7.21874263e-01, 2.96449926e-01, 4.62055574e-01,
        9.23502932e-01, 1.77553262e-01, 5.88996325e-01, 4.80944399e-01,
        3.21763942e-03, 8.22534706e-01, 1.86396860e-01, 8.38382218e-04,
        9.79120537e-01, 3.10180248e-01, 9.42519716e-01, 5.57099984e-01,
        5.05640970e-01, 4.43318836e-01, 9.01426526e-01, 9.86964824e-01],
       [2.89950303e-02, 4.71099353e-01, 4.61049418e-01, 3.60340885e-01,
        9.98274738e-01, 2.50534889e-01, 2.75575968e-01, 8.76483109e-01,
        4.57547619e-01, 4.49216374e-01, 9.00052933e-01, 6.30924496e-02,
        1.47451868e-01, 4.81204976e-01, 2.49459526e-01, 6.86528223e-01,
        2.90840017e-01, 6.53207319e-01, 3.94250794e-01, 9.93358623e-01],
       [2.72242104e-01, 1.04035187e-02, 6.02921026e-01, 6.20019024e-02,
        4.59038298e-01, 6.56724430e-01, 9.48864214e-01, 1.88256785e-01,
        2.02701119e-01, 8.75870992e-01, 1.18842950e-01, 7.73799748e-01,
        1.85353555e-01, 6.66672510e-02, 4.56259126e-01, 3.23538288e-02,
        2.85859950e-01, 6.36264966e-01, 5.59170839e-01, 3.76247303e-01],
       [9.76351977e-01, 4.97996393e-01, 9.64217040e-01, 5.68075873e-01,
        3.17378242e-02, 7.61381890e-01, 4.22182837e-01, 6.56743187e-02,
        7.32435314e-01, 3.10280550e-01, 7.90874256e-01, 8.12958605e-01,
        3.27910036e-01, 7.46045705e-01, 3.89199906e-01, 3.18094485e-01,
        2.34379890e-01, 7.98774800e-01, 7.04203573e-01, 2.61697584e-01],
       [4.35931979e-02, 4.08616099e-01, 8.95794646e-01, 6.03461906e-01,
        2.54035227e-01, 5.01131684e-01, 6.92390631e-01, 2.64095153e-01,
        9.76751623e-01, 2.16288088e-01, 8.52379300e-01, 2.99332109e-01,
        3.15165120e-01, 8.41191126e-01, 9.08038653e-01, 9.11238036e-01,
        5.97581744e-01, 6.06965485e-01, 9.26971528e-01, 3.19057916e-01],
       [1.17497695e-03, 1.42958879e-01, 8.01870041e-02, 3.22784619e-01,
        1.94566403e-01, 3.75094540e-01, 6.53420390e-01, 5.03351777e-01,
        3.38955730e-01, 3.75613691e-01, 4.56445585e-01, 4.73285822e-01,
        5.32748890e-01, 3.19358010e-02, 5.70299841e-01, 3.08935605e-01,
        5.47389392e-01, 6.22206910e-01, 6.94690702e-01, 3.10160869e-01],
       [8.04301102e-01, 1.94790595e-01, 1.79299133e-01, 6.93419596e-01,
        4.21813681e-01, 7.29137303e-01, 2.94977136e-01, 3.35896756e-02,
        3.34775575e-01, 2.30639432e-01, 6.06250948e-01, 6.02884578e-01,
        1.77424663e-01, 3.50318741e-01, 2.98064012e-01, 4.06599053e-01,
        2.55018591e-01, 8.03694437e-01, 1.53013249e-01, 3.31094226e-01],
       [7.64317210e-01, 4.65451888e-01, 4.27882388e-01, 5.51646180e-01,
        1.53083501e-01, 5.50121365e-01, 1.16383243e-01, 4.48709290e-01,
        2.46800015e-01, 4.62315929e-01, 6.88870114e-01, 8.85480219e-01,
        4.42268661e-01, 4.97599491e-02, 1.50951563e-01, 3.44629502e-01,
        5.68956691e-01, 8.59210971e-01, 5.57700100e-01, 4.74861546e-01],
       [8.82945633e-01, 6.66450123e-02, 1.03827310e-01, 7.53909975e-01,
        3.12019955e-01, 3.05296302e-01, 3.13100337e-01, 7.97329787e-01,
        9.03160411e-01, 3.75369483e-01, 7.43596319e-01, 5.90558545e-01,
        1.46726253e-02, 7.17727054e-01, 8.48133035e-01, 1.07065689e-01,
        2.95313779e-01, 8.74864549e-01, 1.26380682e-01, 7.05828797e-01]])
```

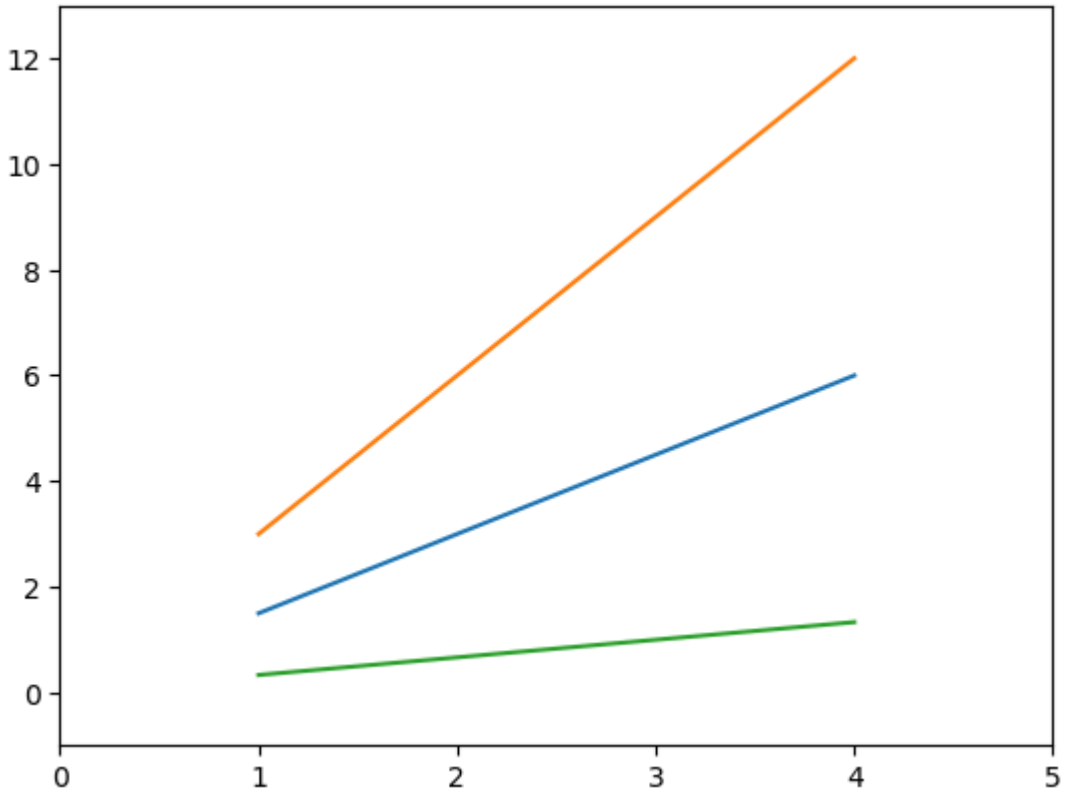
In [67]:
```
cp = plt.contour(matrix1)

plt.show()
```



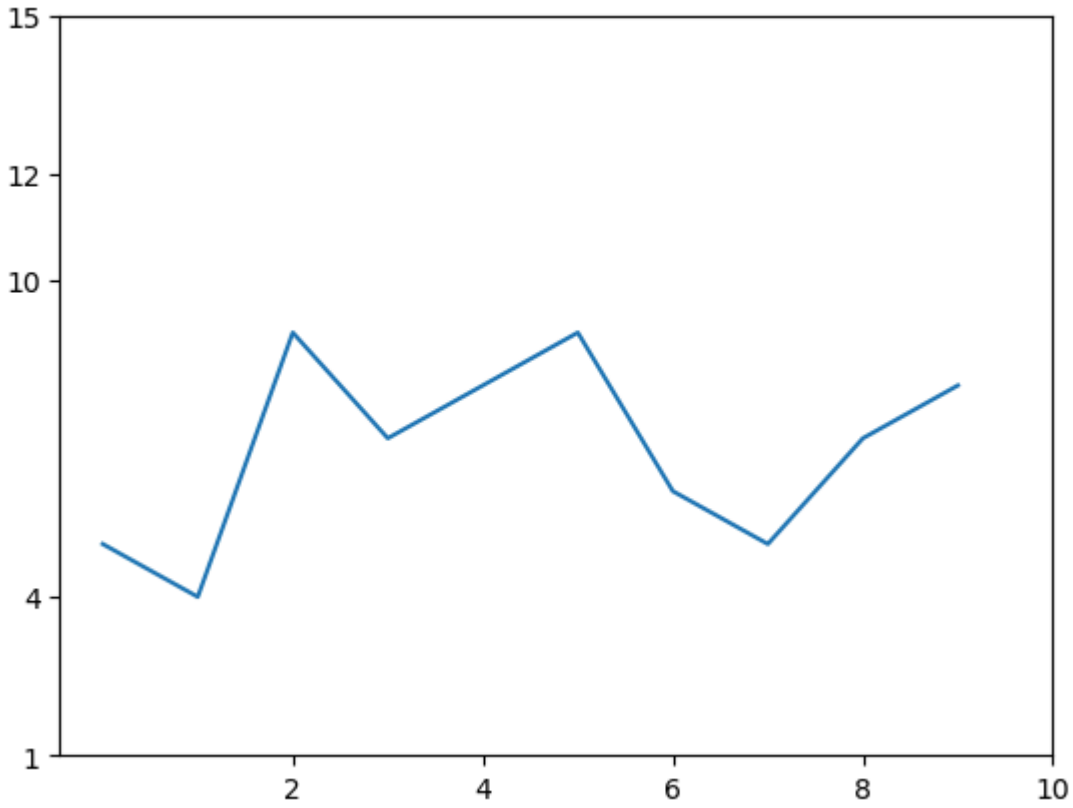

In [70]:
```
x15 = np.arange(1, 5)  # adding grids

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

plt.grid(True)

plt.show()
```



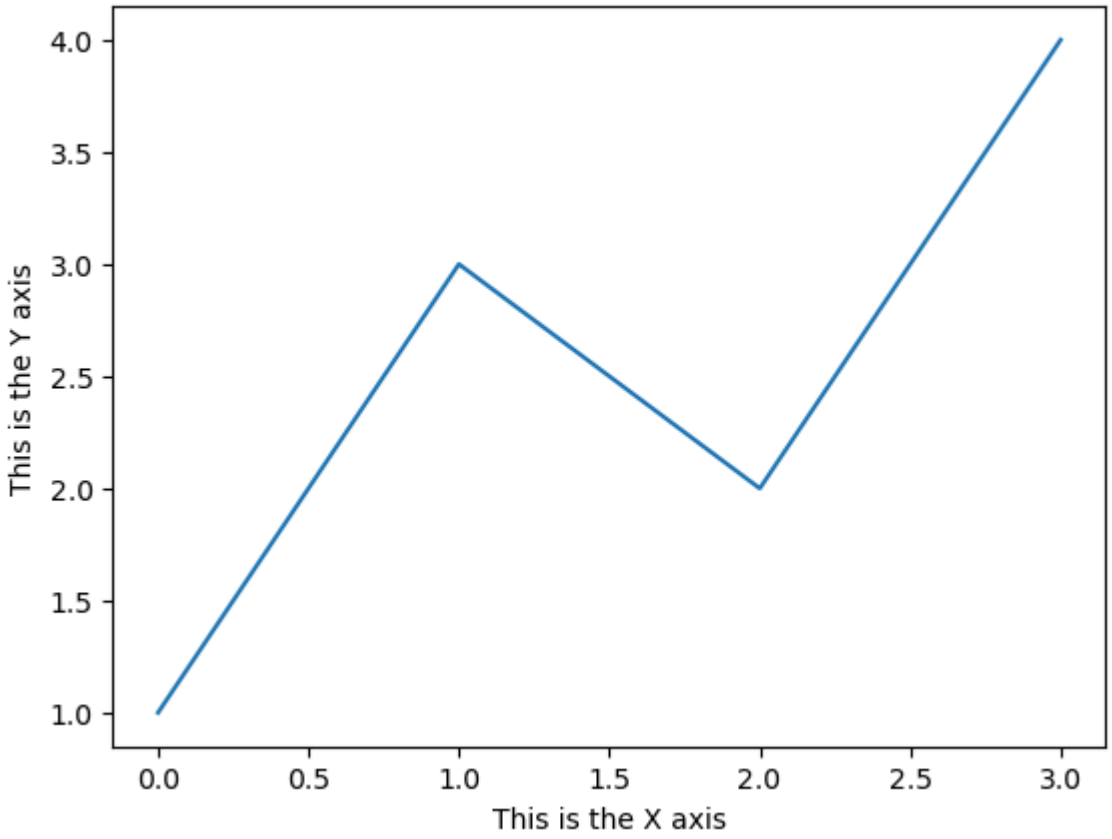

In [71]:
```
x15 = np.arange(1, 5)

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

plt.axis()   # shows the current axis limits values

plt.axis([0, 5, -1, 13])

plt.show()
```



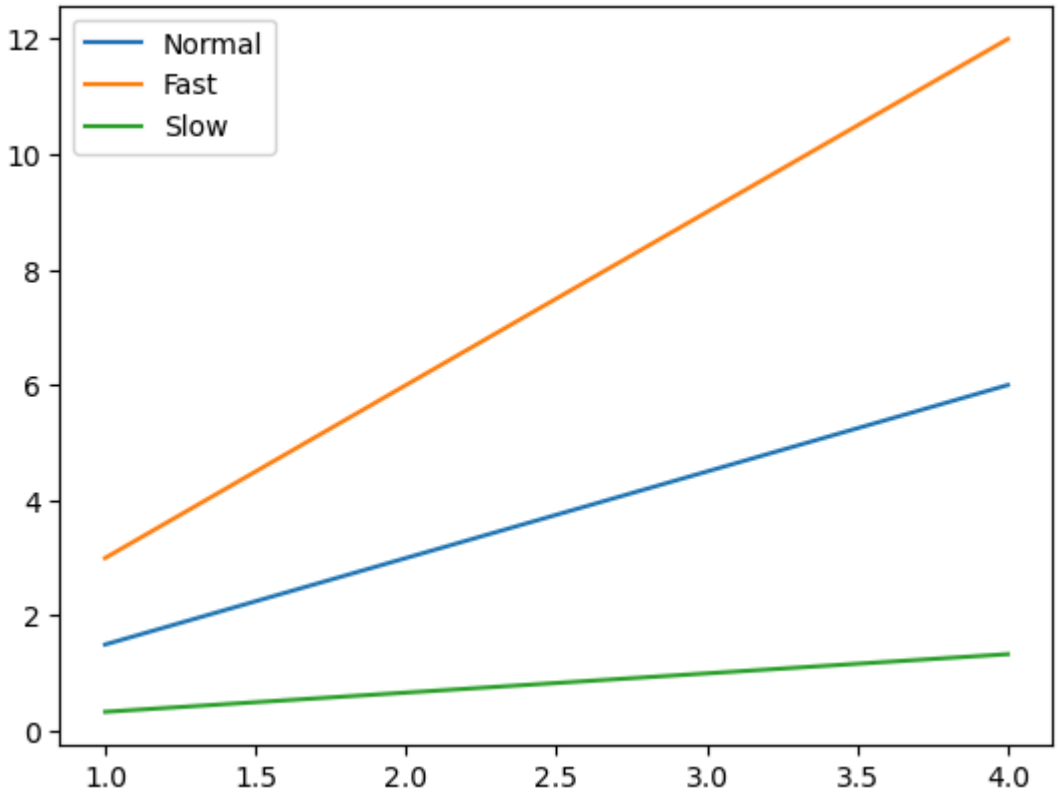

In [73]:
```
r = [5, 4, 9, 7, 8, 9, 6, 5, 7, 8]

plt.plot(r)

plt.xticks([2, 4, 6, 8, 10])
plt.yticks([1, 4, 12, 15, 10])

plt.show()
```



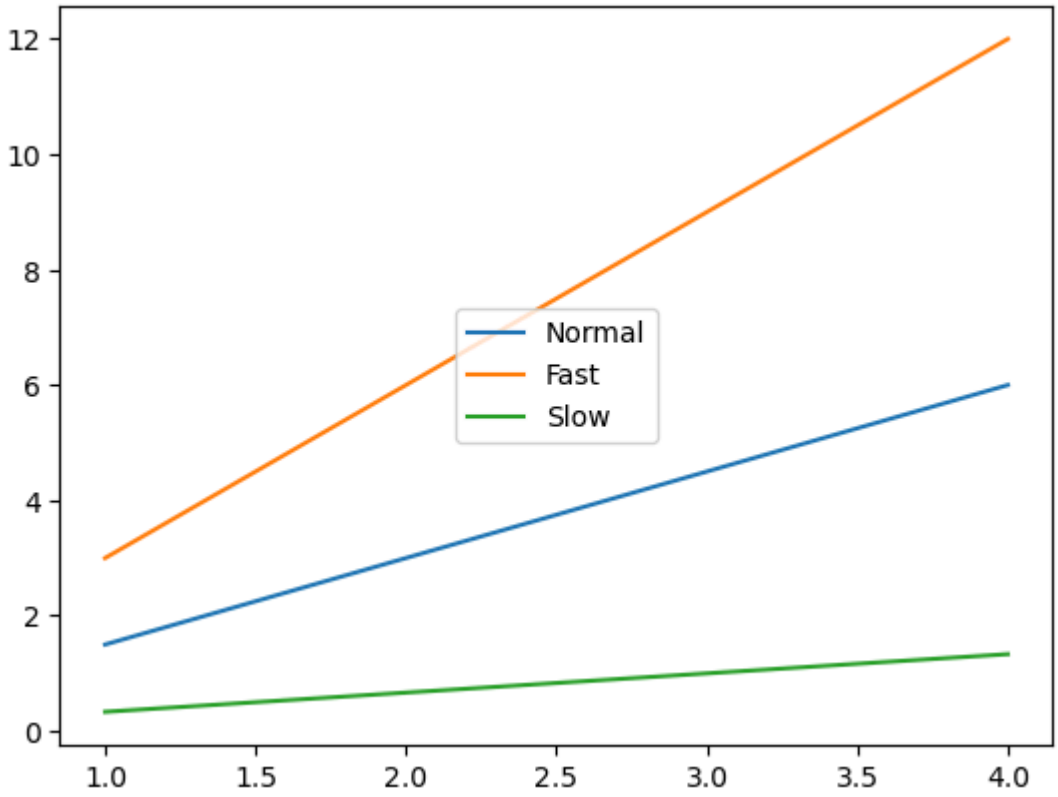

In [74]:
```
plt.plot([1, 3, 2, 4])

plt.xlabel('This is the X axis')

plt.ylabel('This is the Y axis')

plt.show()
```




In [77]:
```
x15 = np.arange(1, 5)

plt.plot(x15, x15*1.5, label='Normal')
plt.plot(x15, x15*3.0, label='Fast')
plt.plot(x15, x15/3.0, label='Slow')

plt.legend();
```



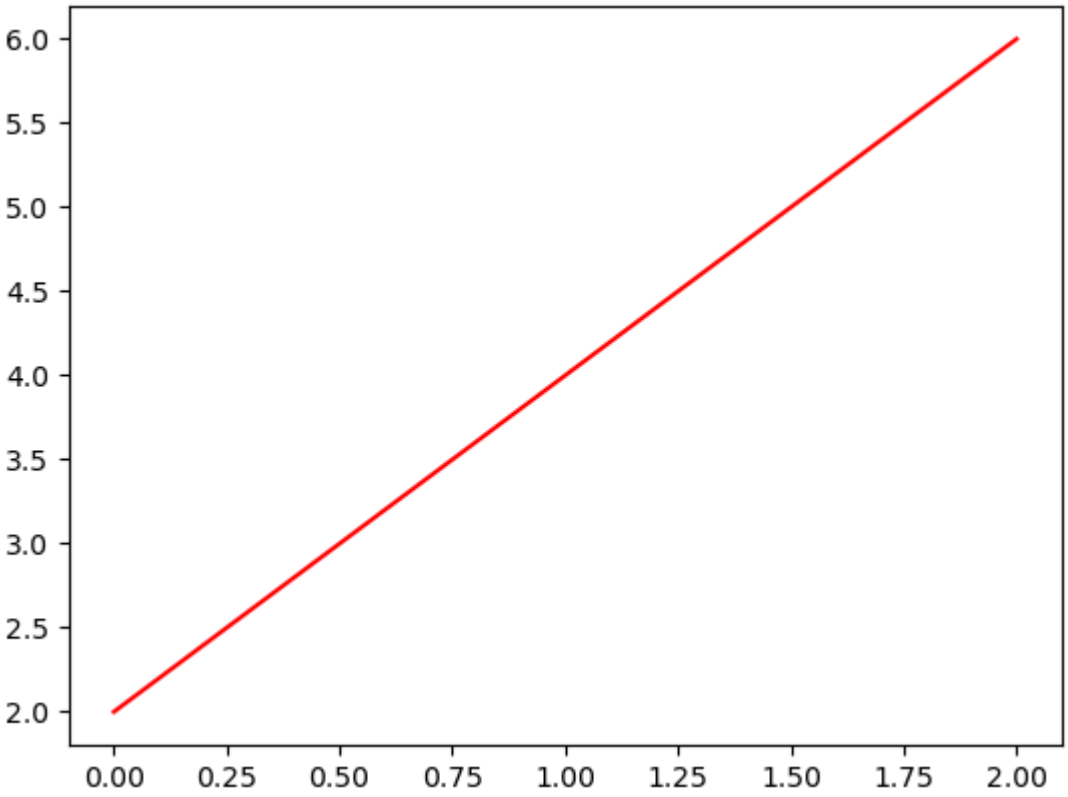

In [81]:
```
x15 = np.arange(1, 5)

fig, ax = plt.subplots()

ax.plot(x15, x15*1.5, label='Normal')
ax.plot(x15, x15*3.0, label='Fast')
ax.plot(x15, x15/3.0, label='Slow')

ax.legend(loc=10);  # loc values various positions
```




In [94]:
```
x12=[2,5,6]
x12
```

Out[94]:
```
[2, 5, 6]
```

In [100…
```
plt.plot([2,4,6],'r')
plt.show()
```




In [ ]: