Matplotlib

In [2]:

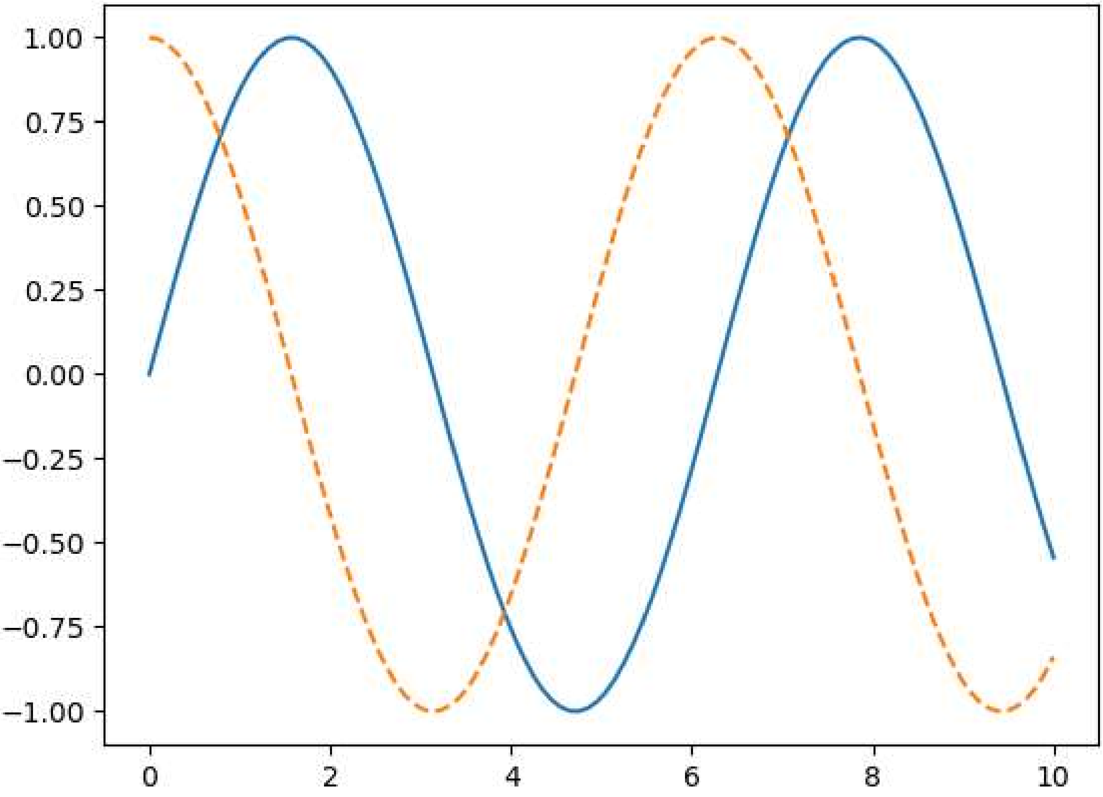
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:

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

fig = plt.figure()

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

Out[4]:

```
[<matplotlib.lines.Line2D at 0x175ae5074d0>]
```

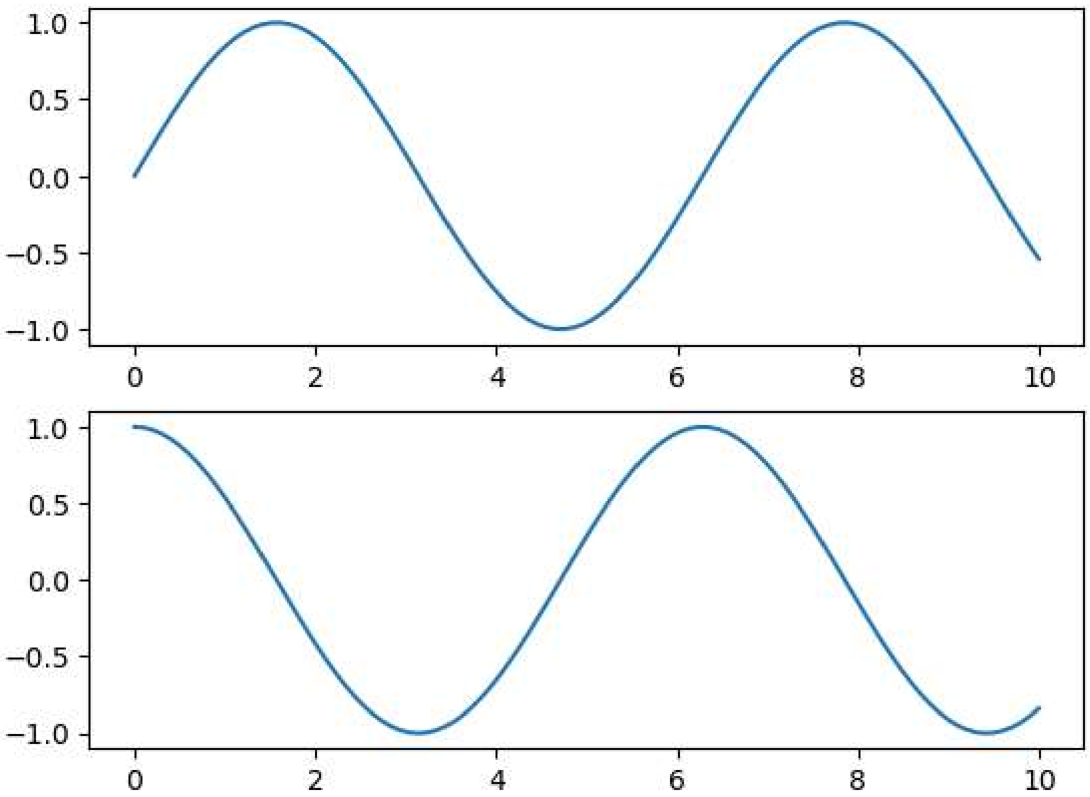


In [6]:

```
plt.figure()

plt.subplot(2,1,1)
plt.plot(x1,np.sin(x1))

plt.subplot(2,1,2)
plt.plot(x1,np.cos(x1))
```

Out[6]:

```
[<matplotlib.lines.Line2D at 0x175ae759730>]
```

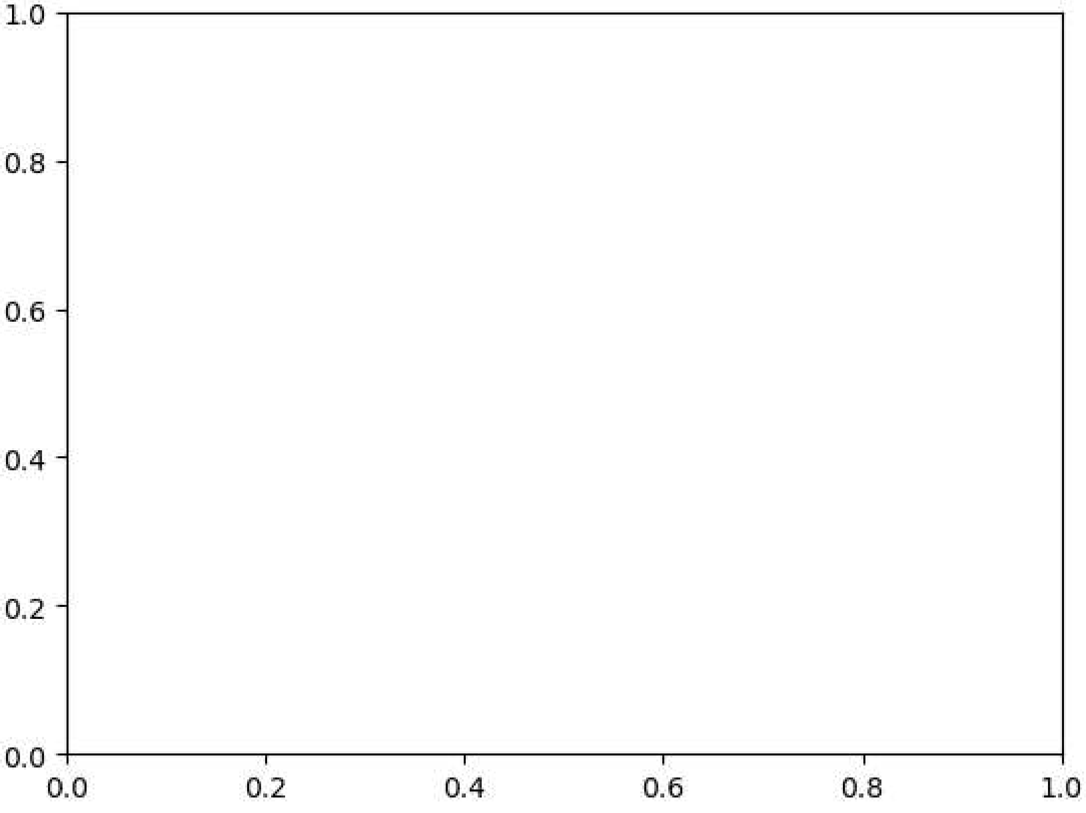


```
In [8]: print(plt.gcf())
```

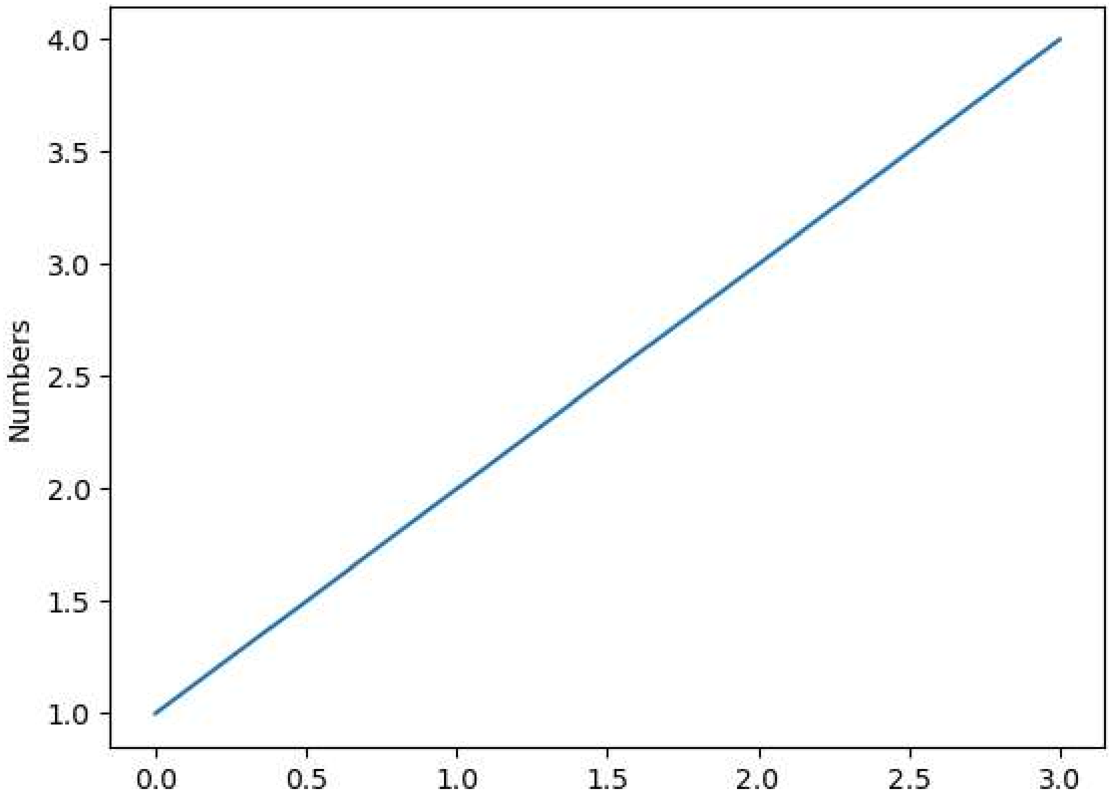
```
Figure(640x480)
<Figure size 640x480 with 0 Axes>
```

```
In [10]: print(plt.gca())
```

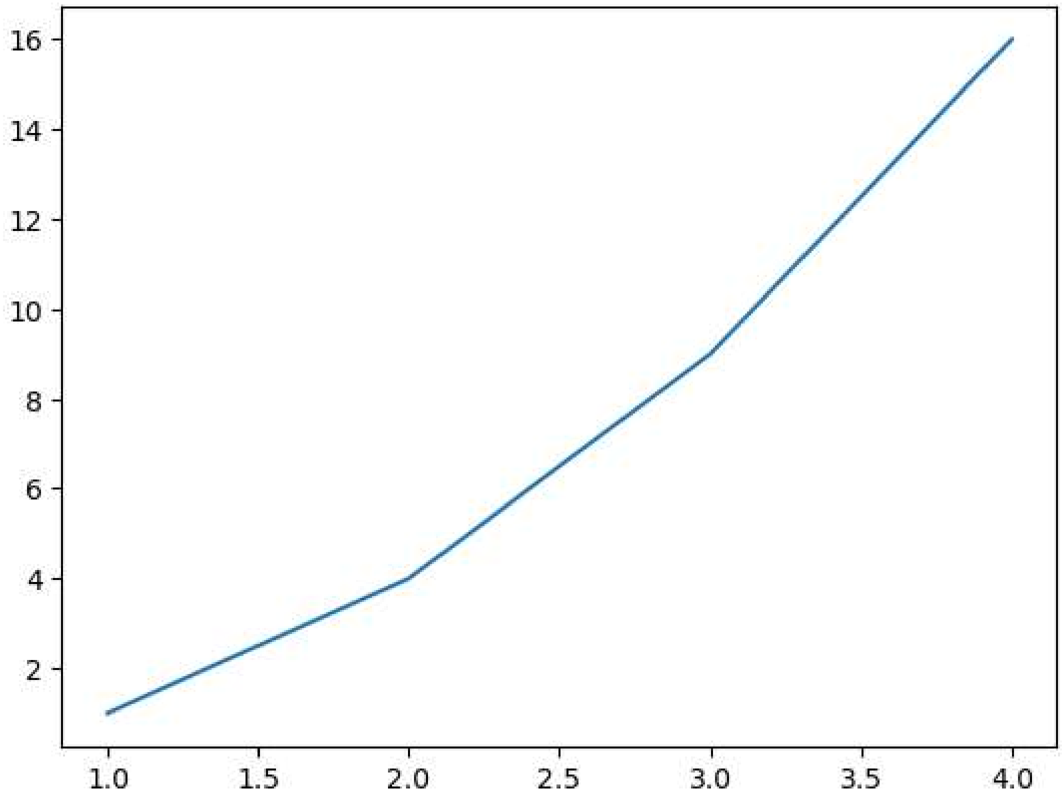
```
Axes(0.125,0.11;0.775x0.77)
```



In [12]:
```
plt.plot([1,2,3,4])
plt.ylabel('Numbers')
plt.show()
```

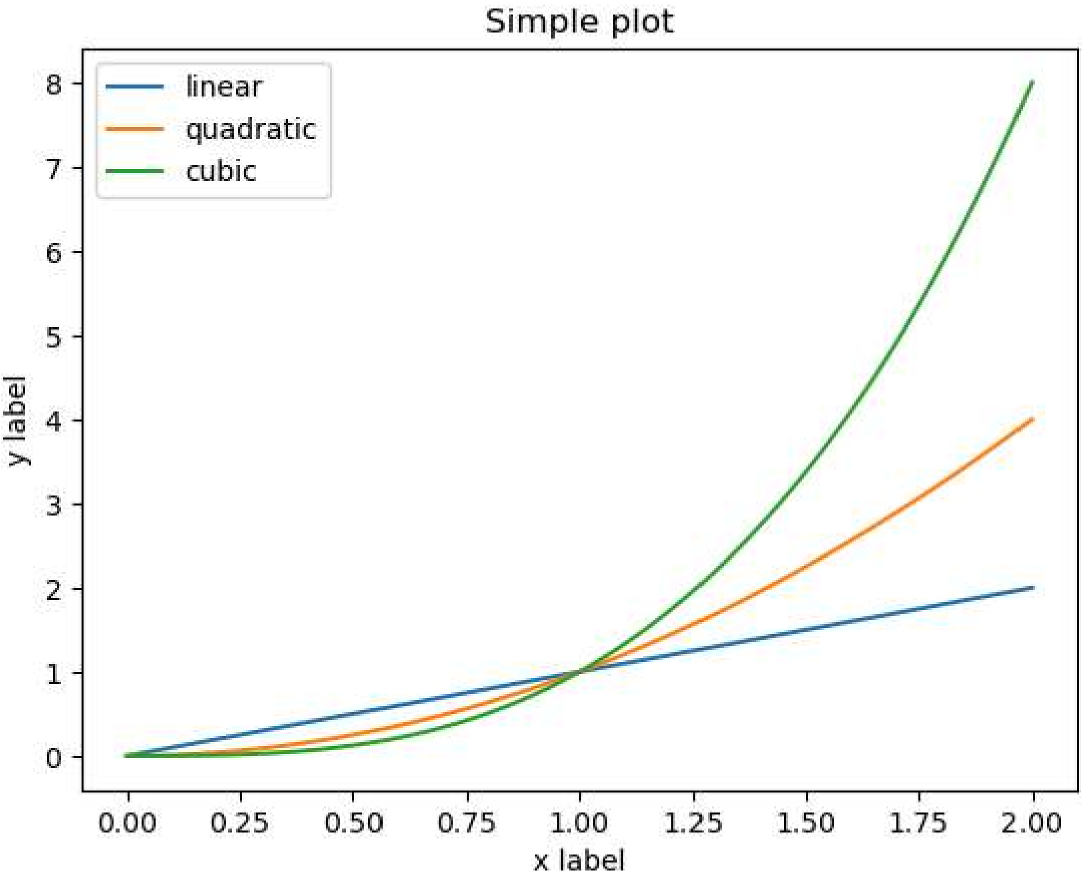


In [14]:
```
plt.plot([1,2,3,4],[1,4,9,16])
plt.show()
```

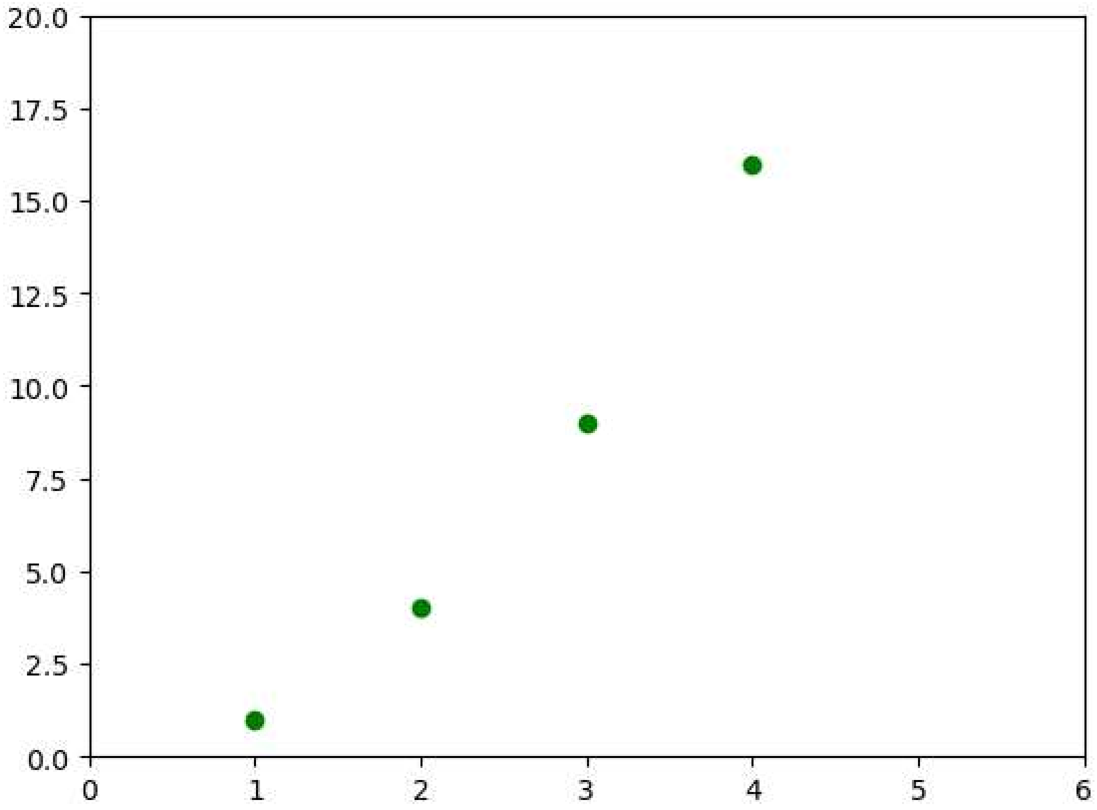


In [16]:
```
x= np.linspace(0,2,100)
```

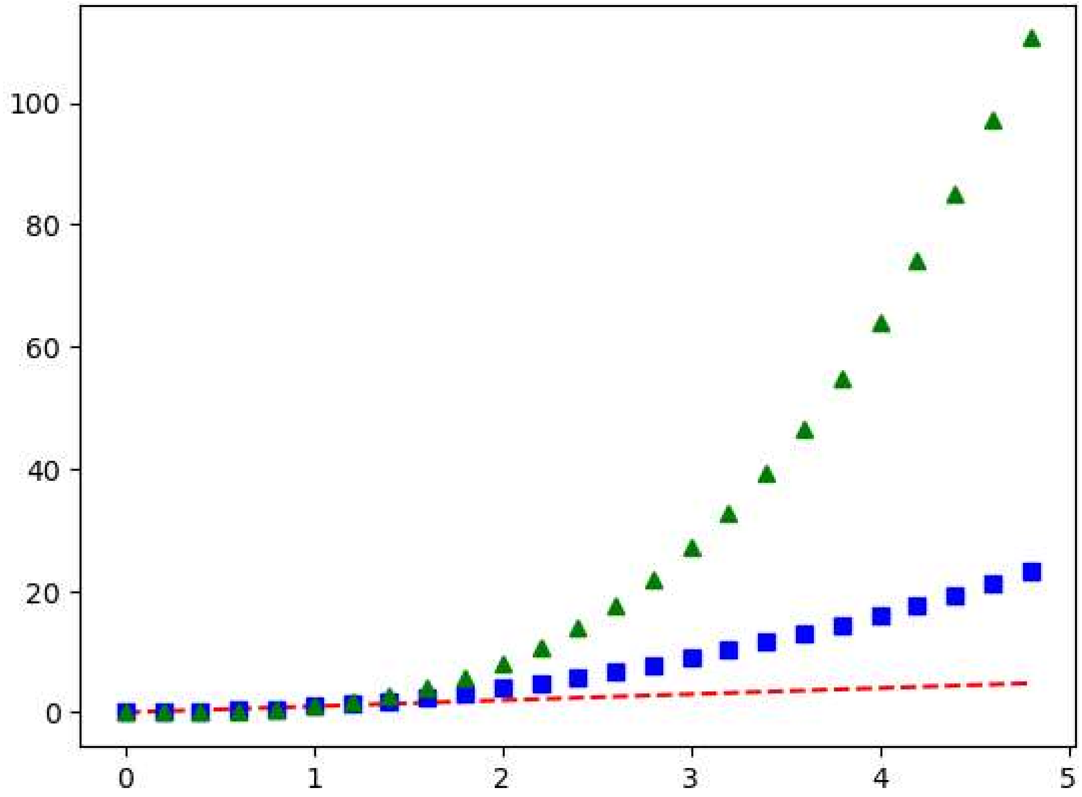
```
plt.plot(x,x,label='linear')
plt.plot(x,x**2,label='quadratic')
plt.plot(x,x**3,label='cubic')

plt.xlabel('x label')
plt.ylabel('y label')

plt.title("Simple plot")
plt.legend()
plt.show()
```



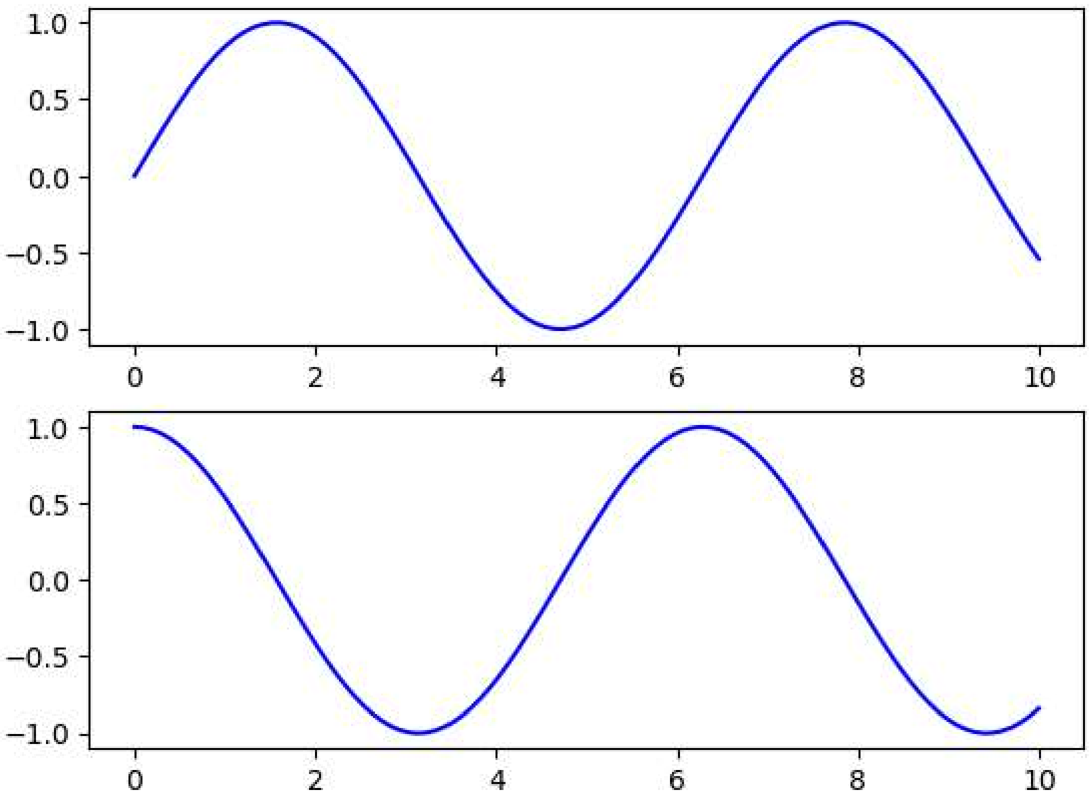
In [18]:
```
plt.plot([1,2,3,4],[1,4,9,16],'go')
plt.axis([0,6,0,20])
plt.show()
```



In [20]:
```
t = np.arange(0.,5.,0.2)

plt.plot(t,t,'r--',t,t**2,'bs',t,t**3,'g^')
plt.show()
```

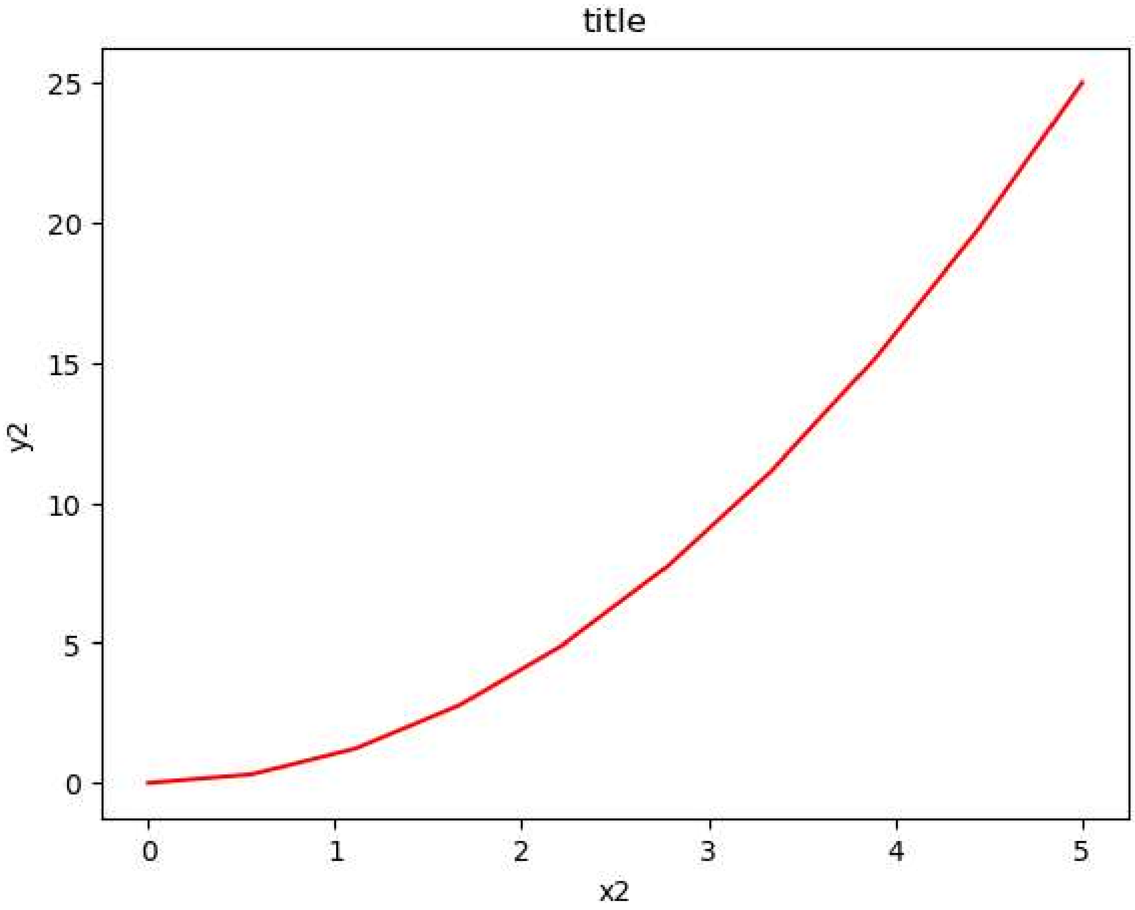


In [22]:
```
fig,ax = plt.subplots(2)

ax[0].plot(x1, np.sin(x1),'b-')
ax[1].plot(x1, np.cos(x1),'b-');
```

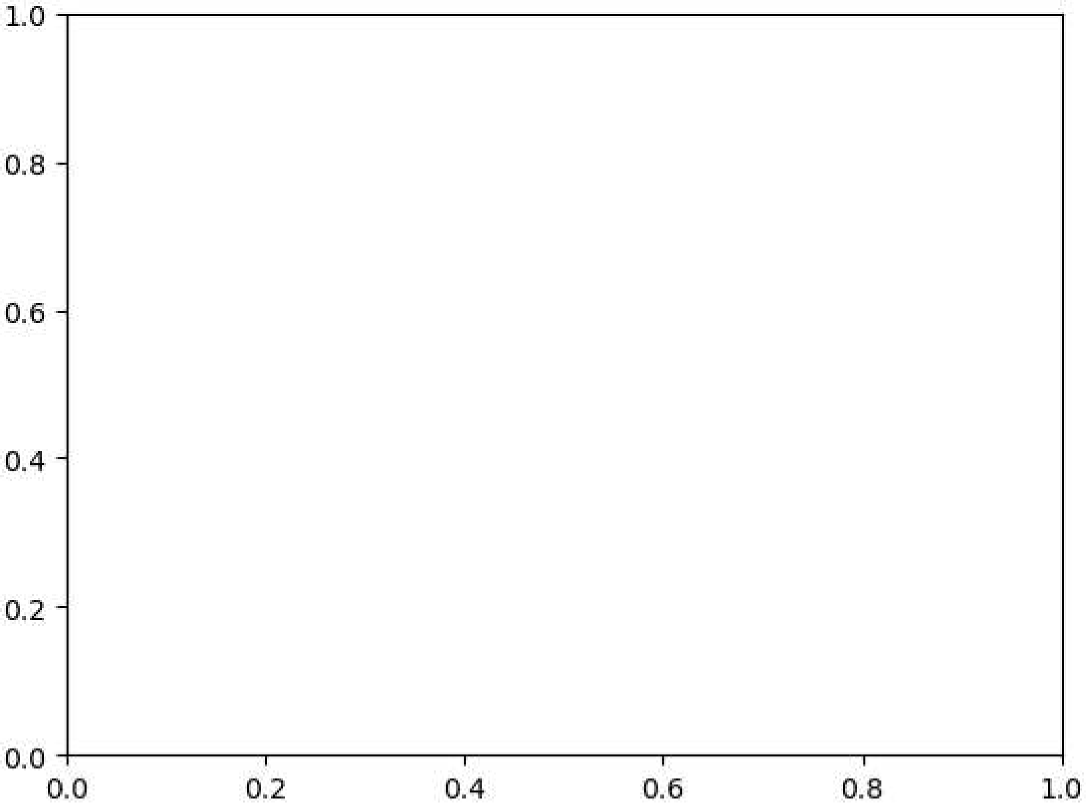


In [24]:

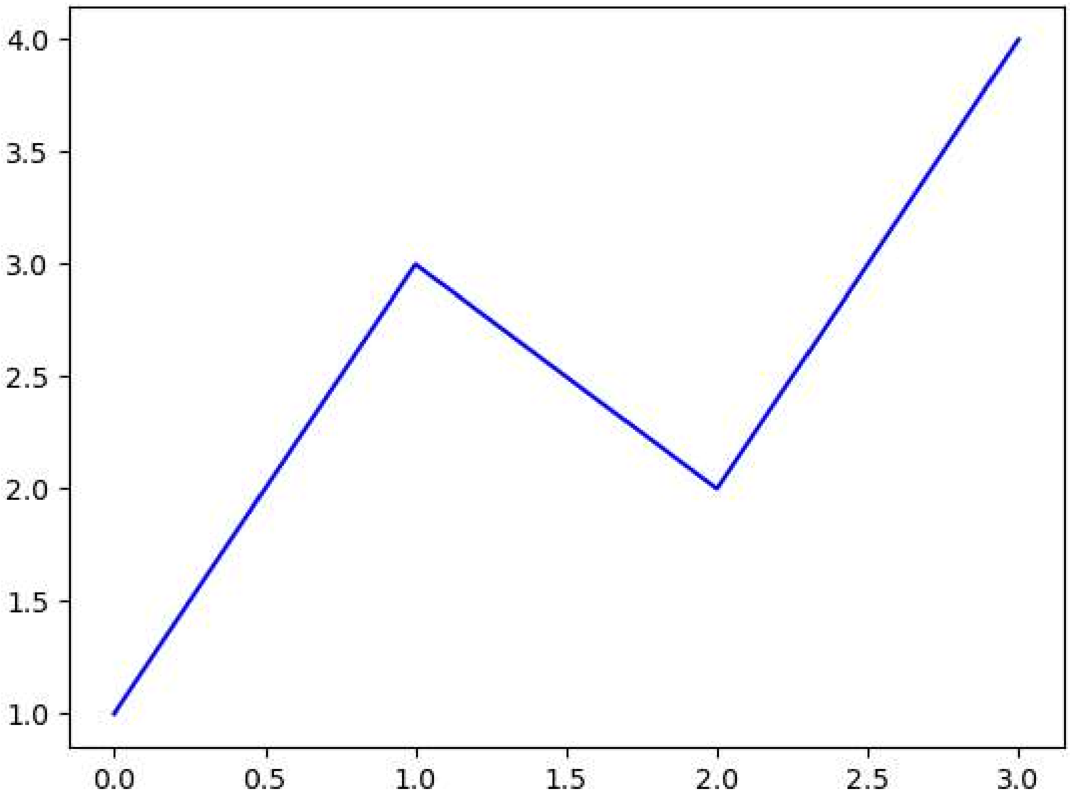
```
fig = plt.figure()

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');
```



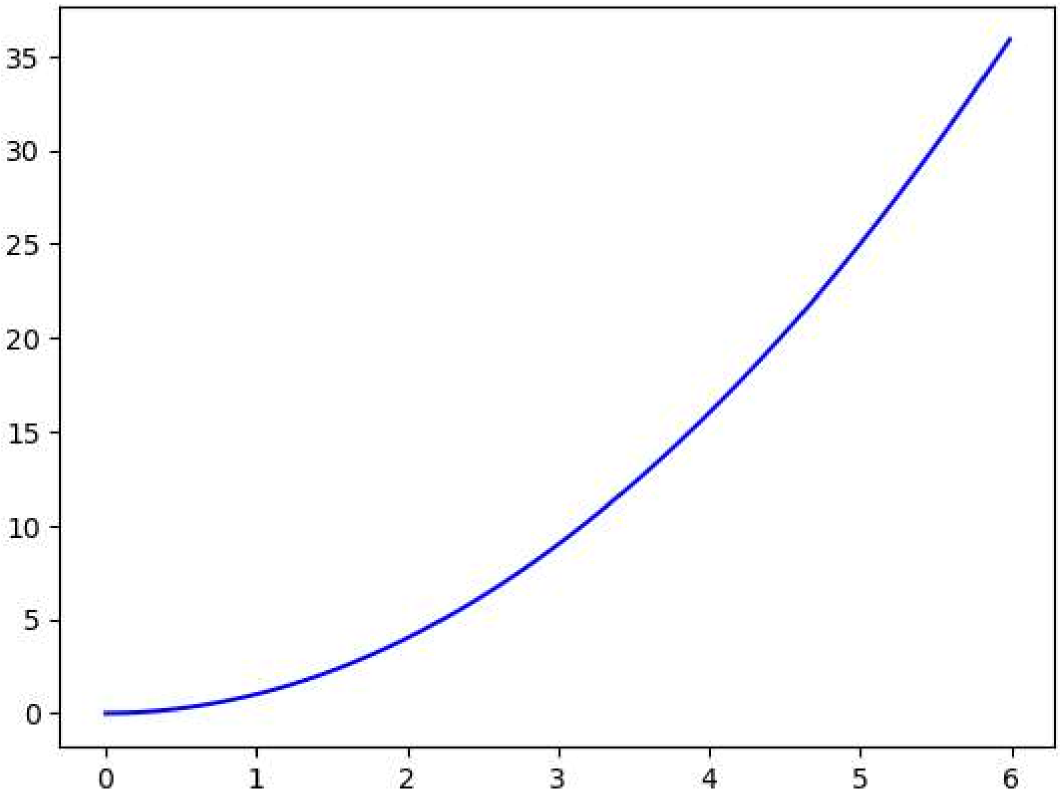
```
In [26]: fig = plt.figure()

         ax = plt.axes()
```



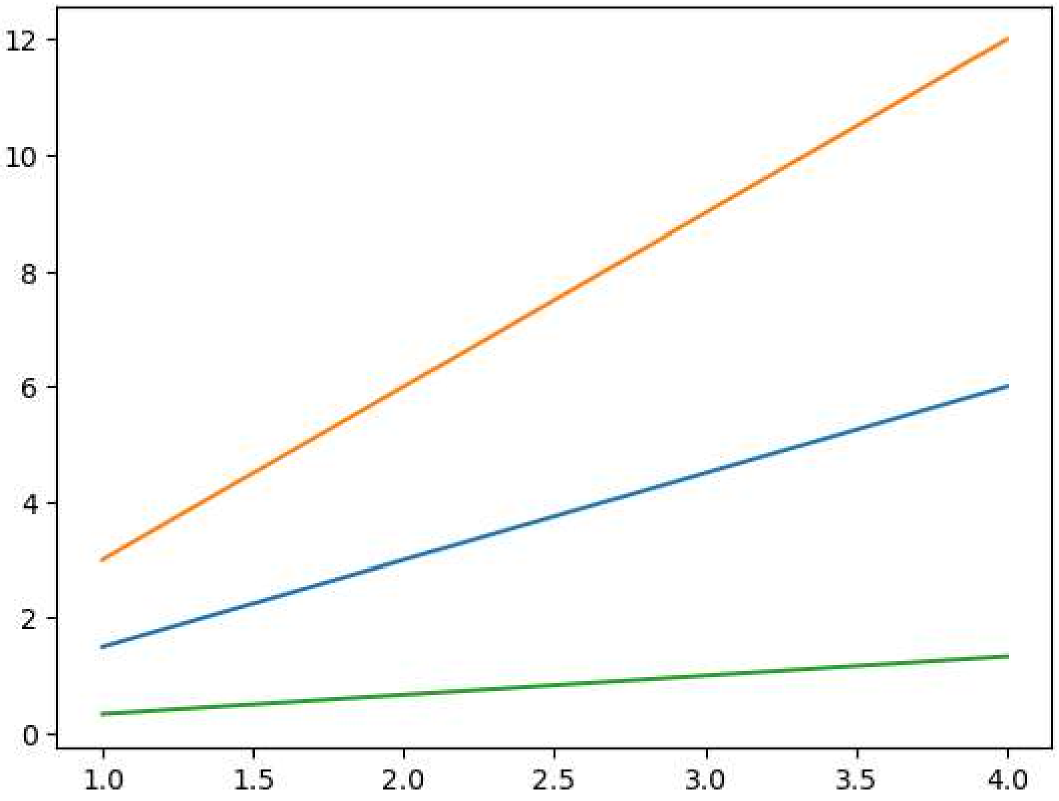
```
In [28]: plt.plot([1,3,2,4],'b-')
         plt.show()
```



In [30]:
```
x3 = np.arange(0.0,6.0,0.01)
plt.plot(x3,[xi**2 for xi in x3],'b-')
plt.show()
```



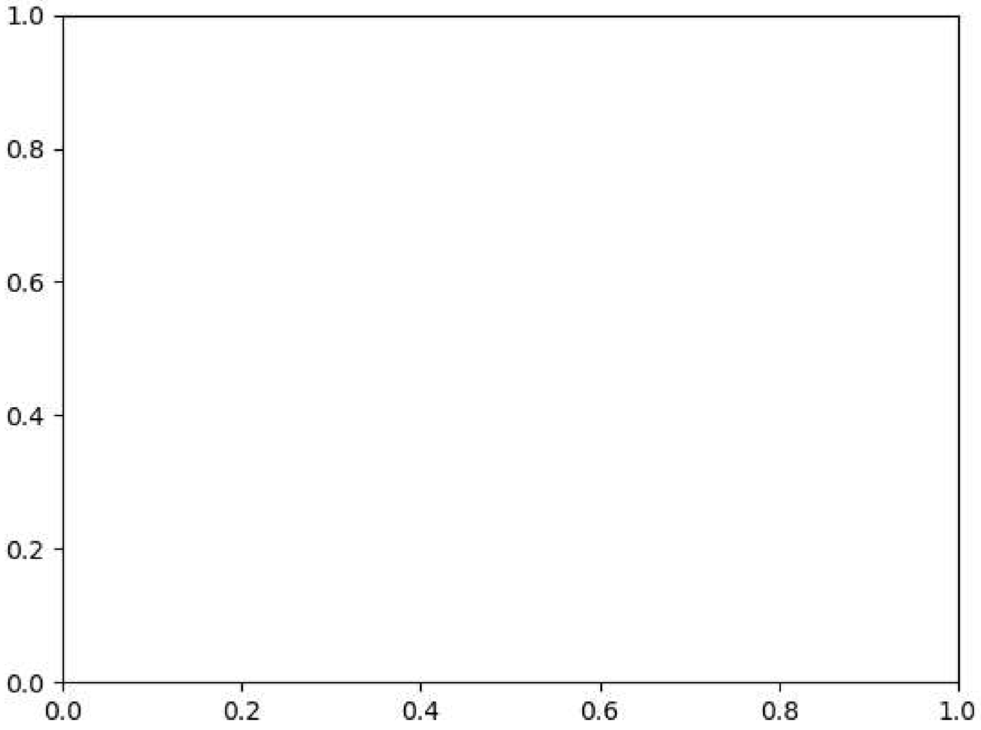
In [32]:
```
x4 = range(1,5)

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])
plt.show()
```



In [34]:
```
fig.savefig('plot1.png')
```

In [36]:
```
from IPython.display import Image
Image('plot1.png')
```

Out[36]:



In [38]:

```
fig.canvas.get_supported_filetypes()
```

Out[38]:

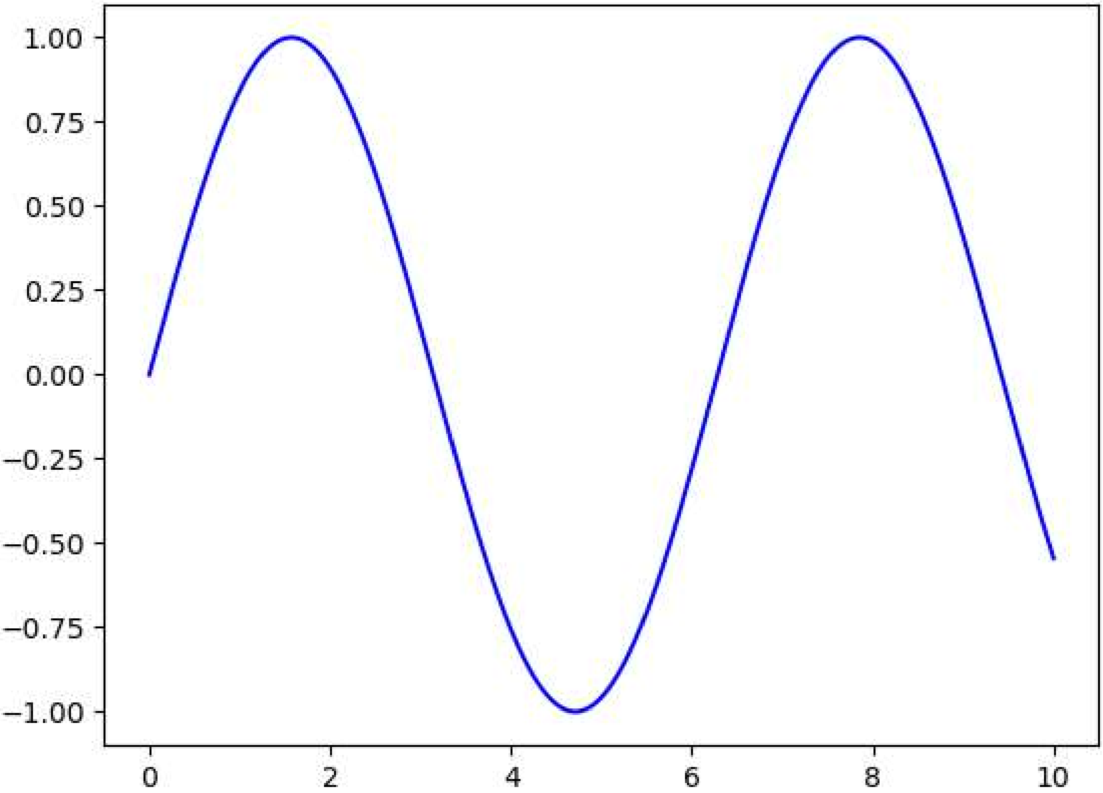
```
{'eps': 'Encapsulated Postscript',
 'jpg': 'Joint Photographic Experts Group',
 'jpeg': 'Joint Photographic Experts Group',
 'pdf': 'Portable Document Format',
 'pgf': 'PGF code for LaTeX',
 'png': 'Portable Network Graphics',
 'ps': 'Postscript',
 'raw': 'Raw RGBA bitmap',
 'rgba': 'Raw RGBA bitmap',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

In [40]:

```
fig = plt.figure()

ax = plt.axes()

x5 = np.linspace(0,10,1000)

ax.plot(x5,np.sin(x5),'b-')
```

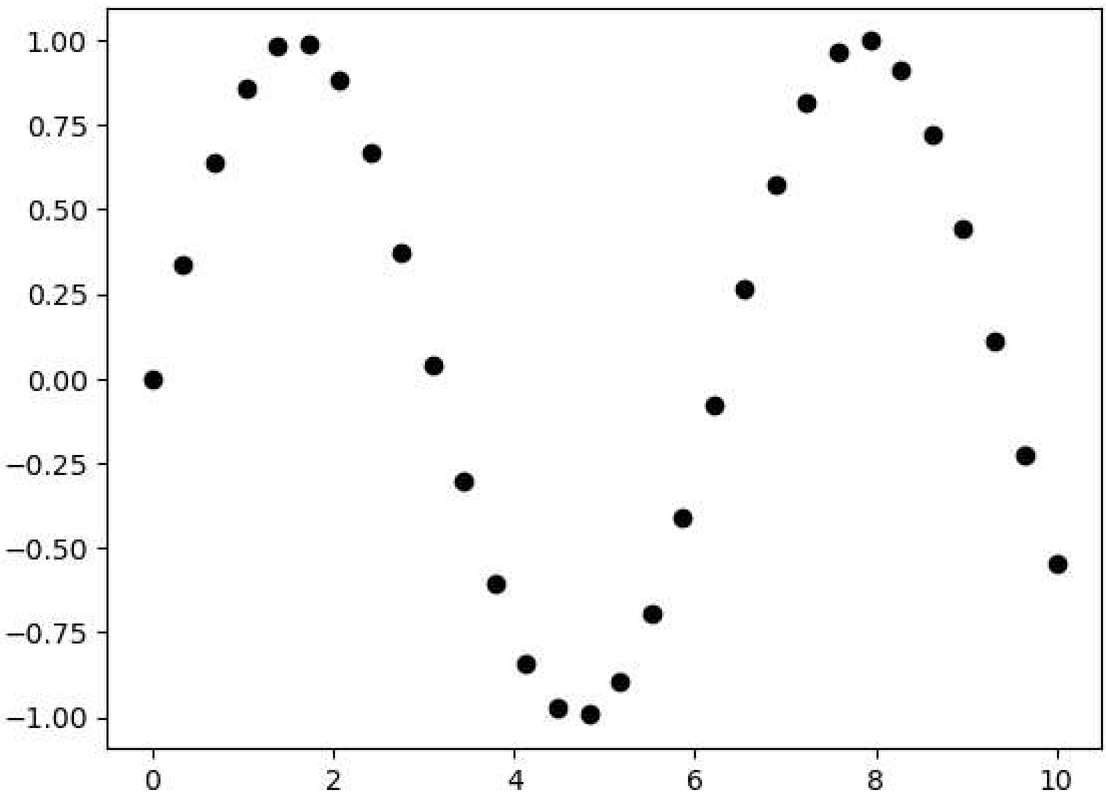
Out[40]:

```
[<matplotlib.lines.Line2D at 0x175afd5f140>]
```

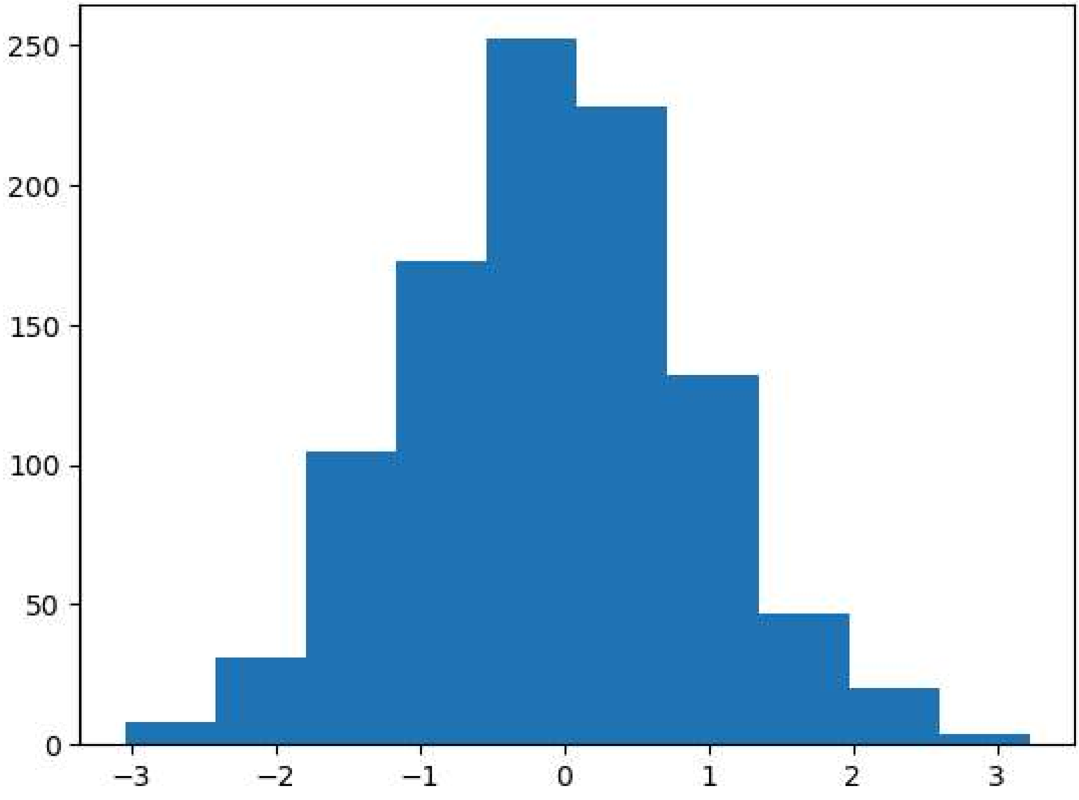


In [42]:

```
x7 = np.linspace(0,10,30)

y7 = np.sin(x7)

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

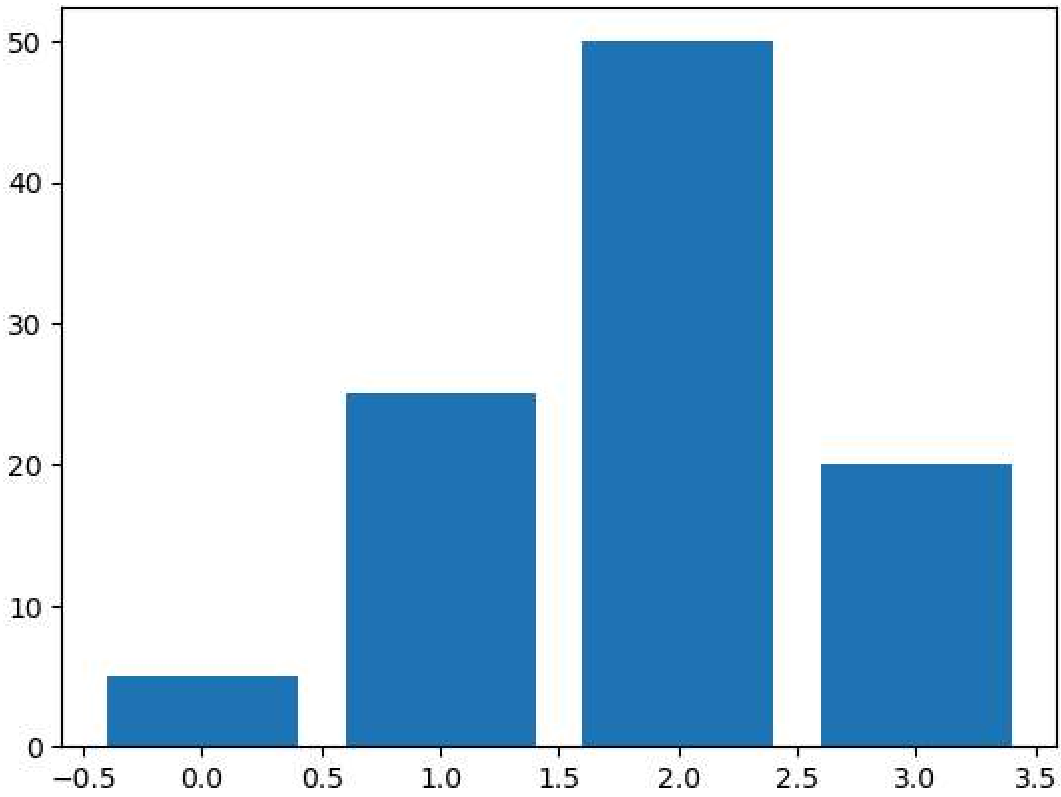


```
In [44]: data1 = np.random.randn(1000)

         plt.hist(data1)

Out[44]: (array([  8.,  31., 105., 173., 252., 228., 132.,  47.,  20.,   4.]),
          array([-3.04097229, -2.41453468, -1.78809707, -1.16165946, -0.53522185,
                  0.09121576,  0.71765337,  1.34409098,  1.97052859,  2.59696621,
                  3.22340382]),
          <BarContainer object of 10 artists>)
```



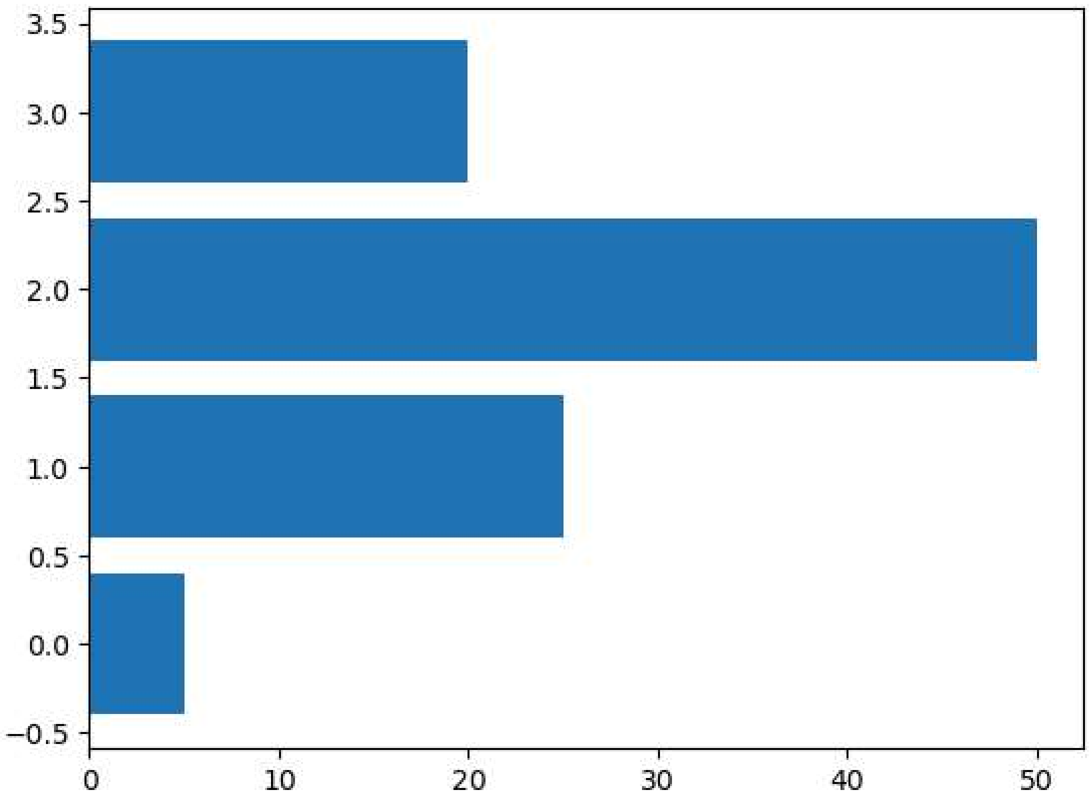
In [46]:

```
data2 = [5.,25.,50.,20.]

plt.bar(range(len(data2)),data2)

plt.show()
```



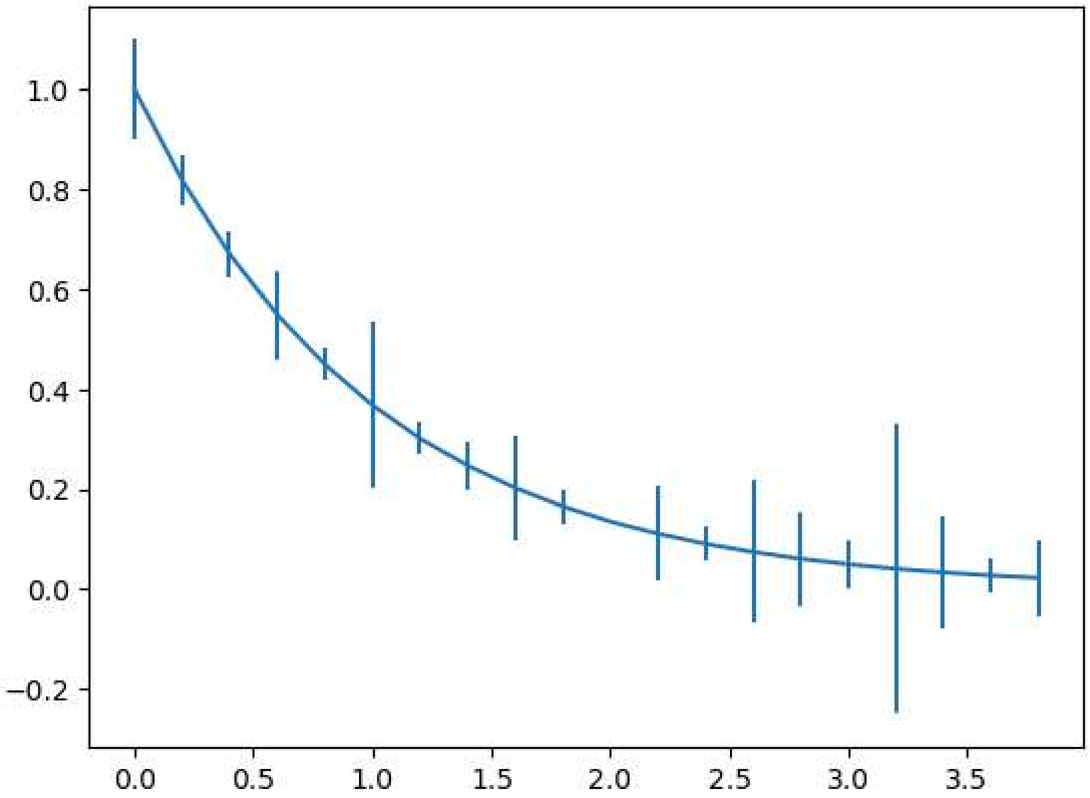
In [48]:

```
data2 = [5.,25.,50.,20.]
plt.barh(range(len(data2)),data2)

plt.show()
```



In [50]:

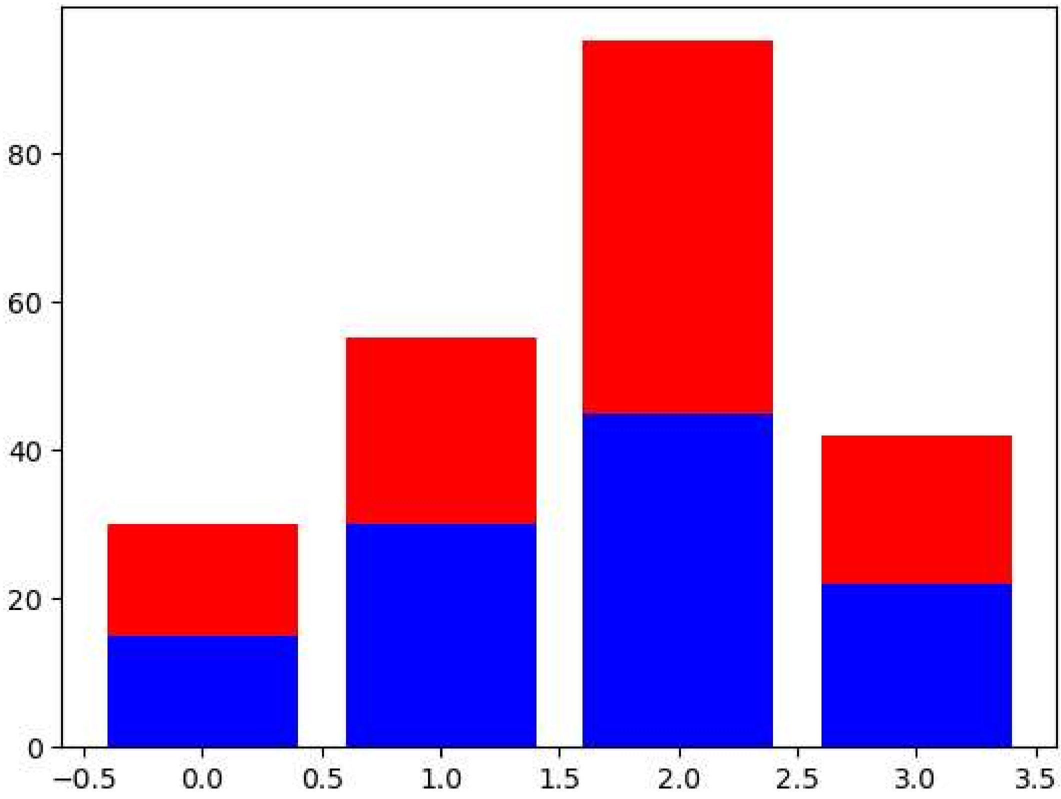
```
x9 = np.arange(0,4,0.2)
y9 = np.exp(-x9)

e1 = 0.1*np.abs(np.random.randn(len(y9)))
plt.errorbar(x9,y9,yerr =e1,fmt =',-')
plt.show()
```



In [52]:

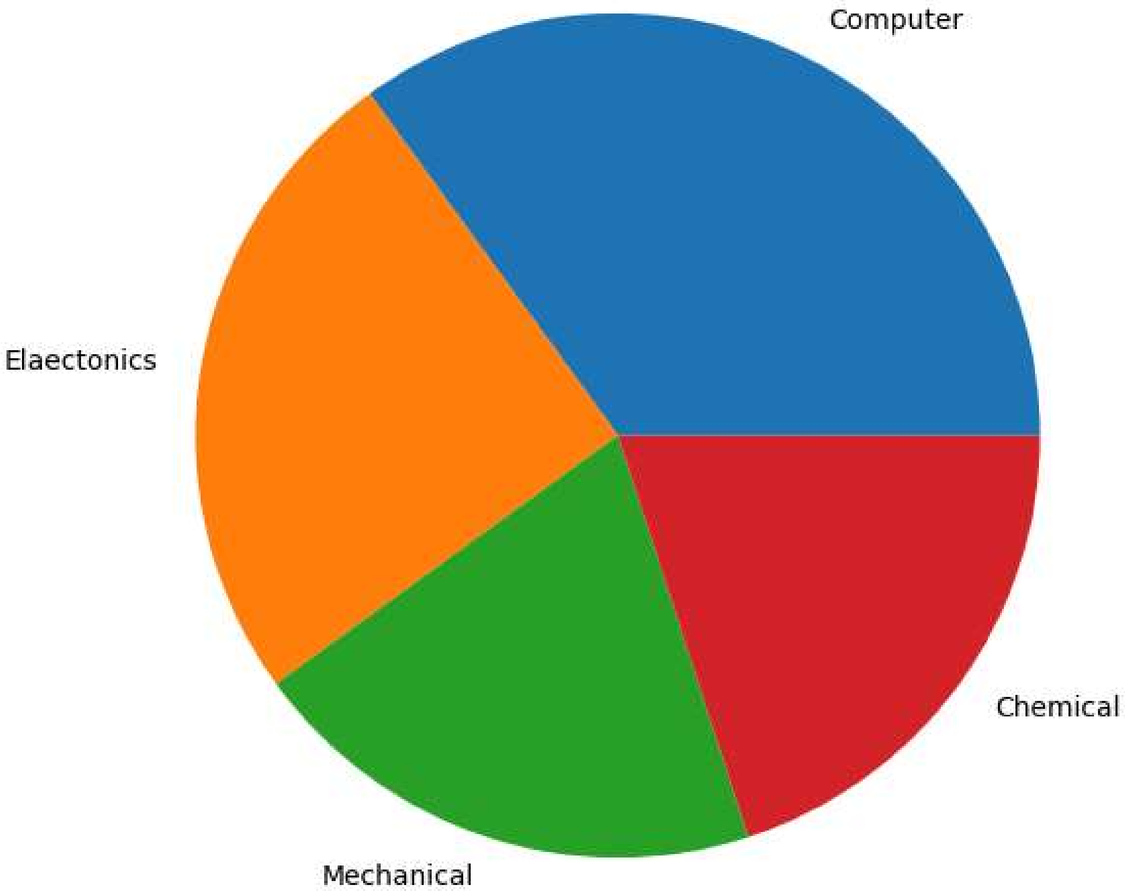
```
A= [15.,30.,45.,22.]
B= [15.,25.,50.,20.]
```

```
z2=range(4)
plt.bar(z2,A,color ='b')
plt.bar(z2,B,color ='r',bottom =A)
plt.show()
```

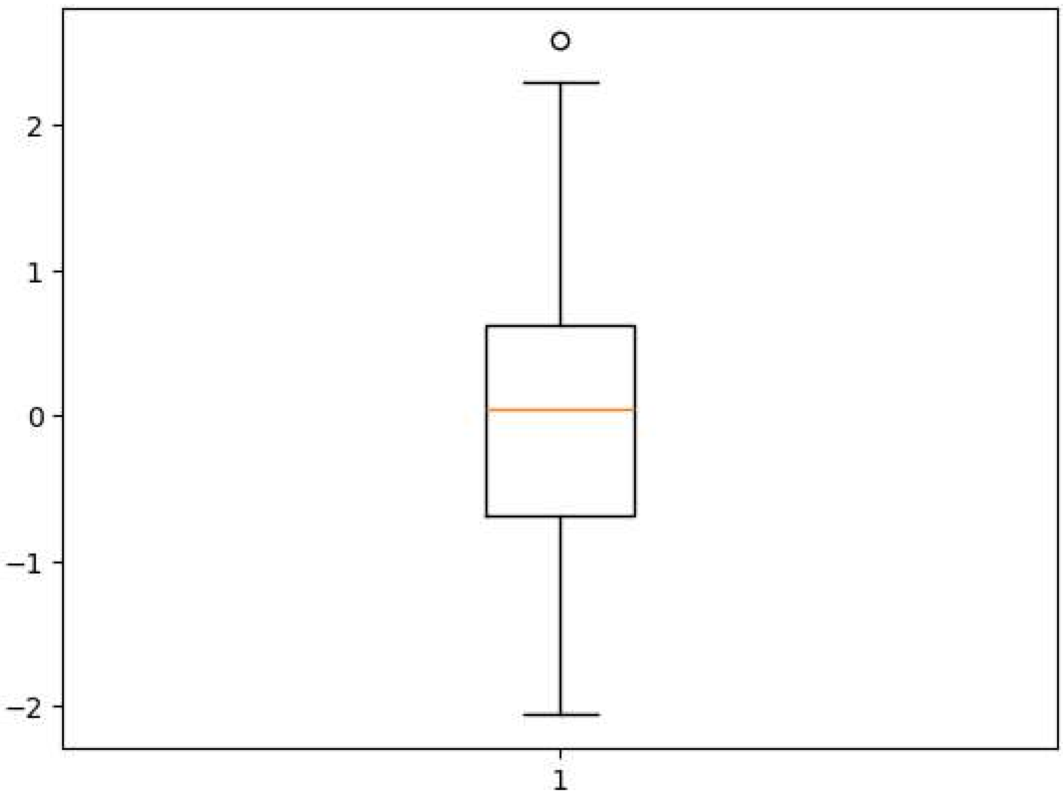


In [54]:

```
plt.figure(figsize = (7,7))
x10 = [35,25,20,20]
labels =['Computer','Elaectonics','Mechanical','Chemical']
plt.pie(x10,labels=labels);
plt.show()
```

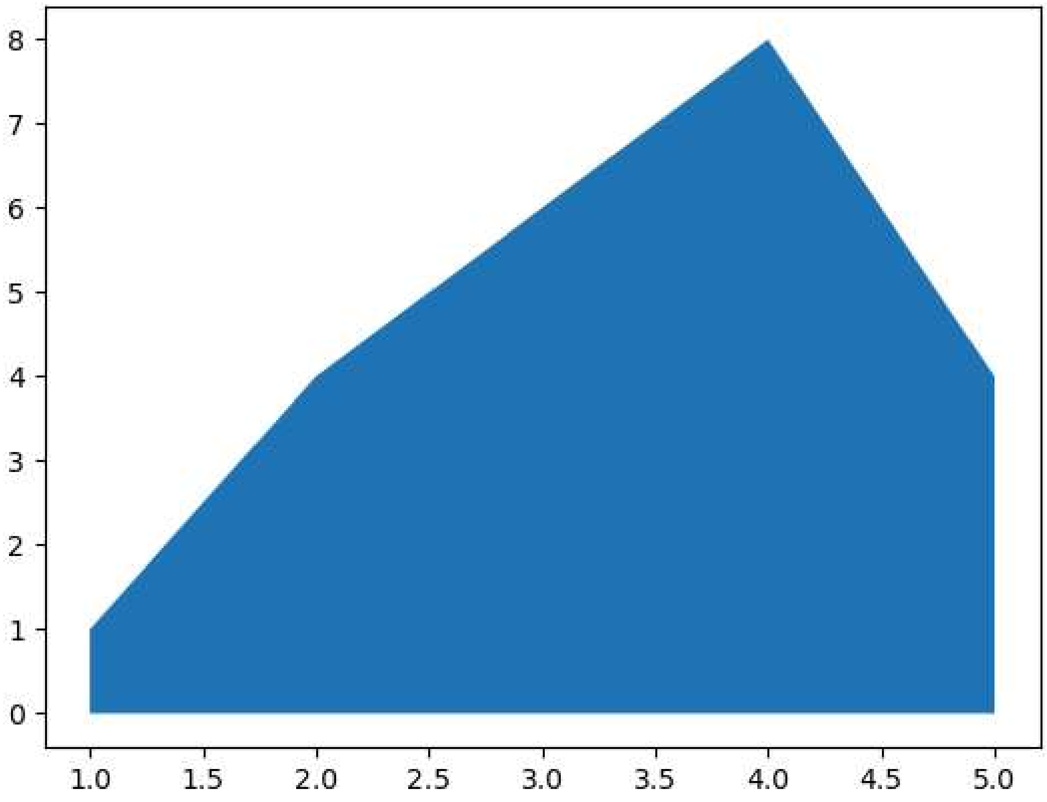


```
In [56]: date3 = np.random.randn(100)

         plt.boxplot(date3)

         plt.show()
```



In [58]:

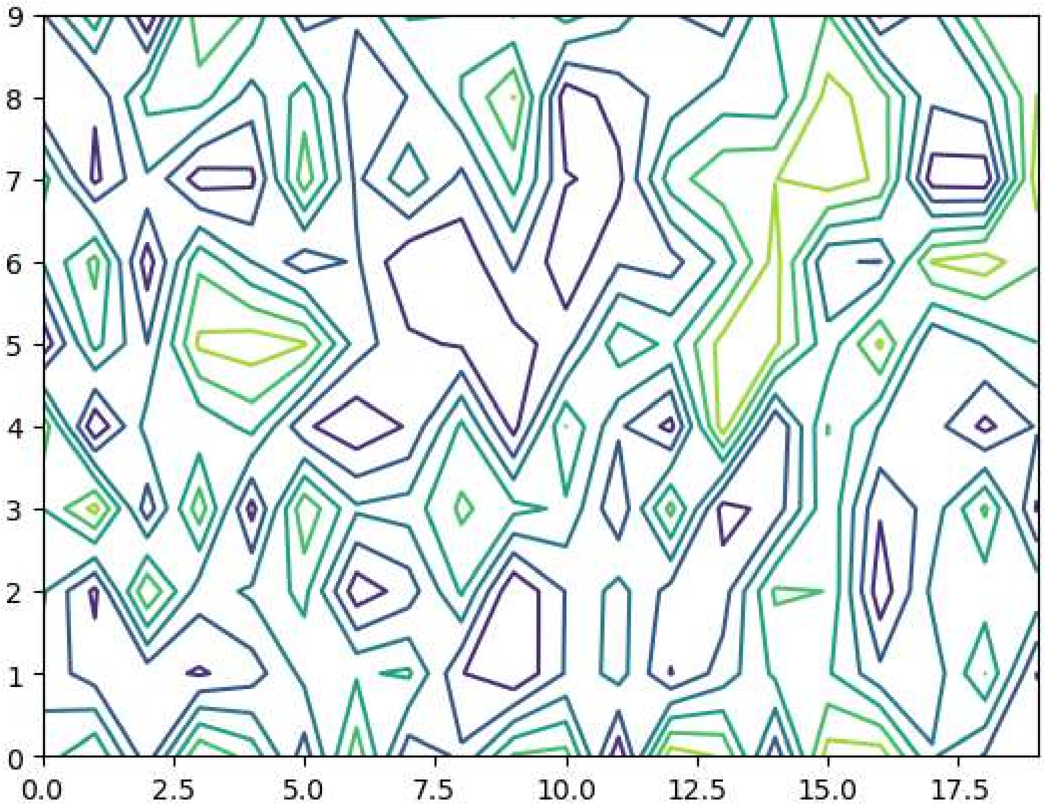
```
x12 = range(1,6)
y12 = [1,4,6,8,4]

plt.fill_between(x12,y12)
plt.show()
```



In [60]:

```
matrix1 = np.random.rand(10,20)

cp = plt.contour(matrix1)
```

```
plt.show()
```

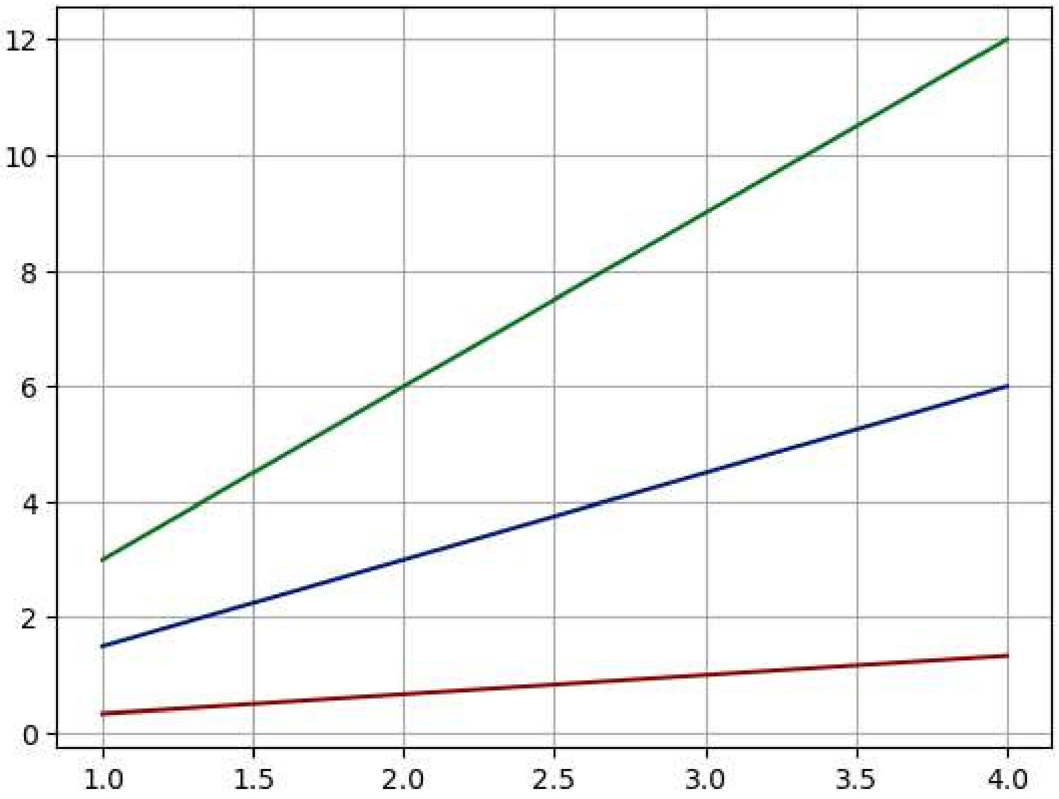


In [62]:
```
print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid',
'bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot', 'graysc
ale', '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-d
eep', 'seaborn-v0_8-muted', 'seaborn-v0_8-notebook', 'seaborn-v0_8-paper', 'seabo
rn-v0_8-pastel', 'seaborn-v0_8-poster', 'seaborn-v0_8-talk', 'seaborn-v0_8-tick
s', 'seaborn-v0_8-white', 'seaborn-v0_8-whitegrid', 'tableau-colorblind10']
```

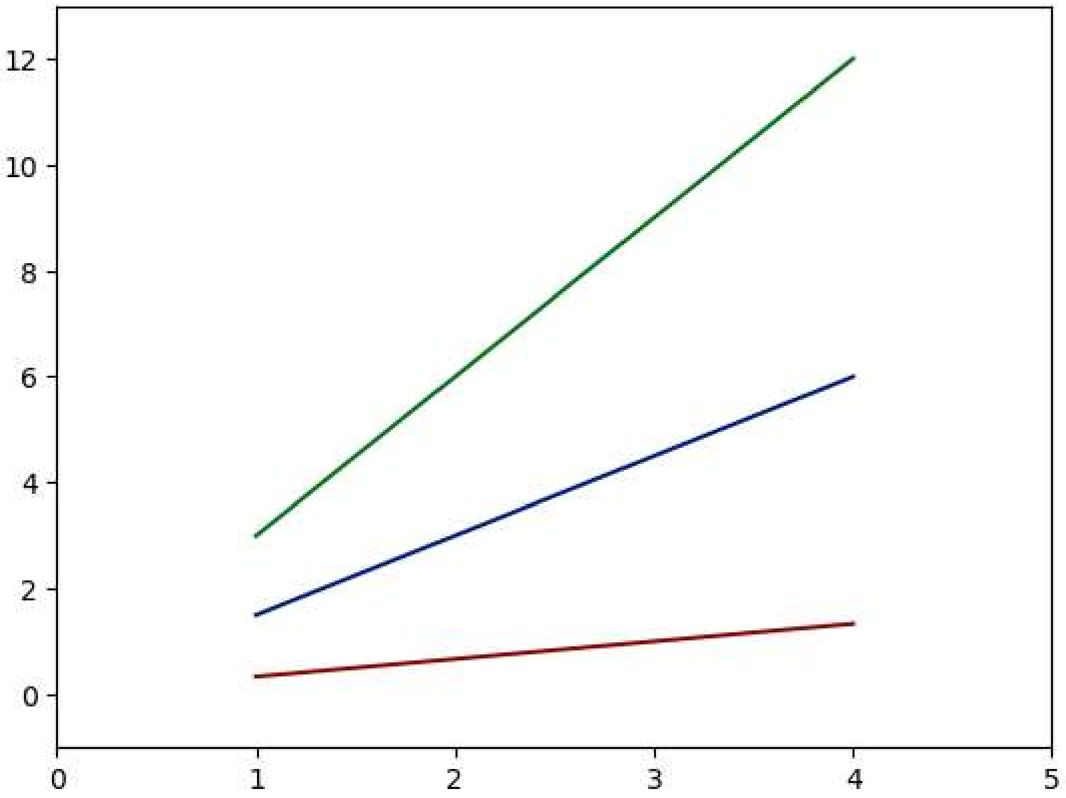
In [64]:
```
plt.style.use('seaborn-v0_8-dark-palette')
```

In [66]:
```
x15 =np.arange(1,5)

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

plt.grid(True)
plt.show()
```



In [68]:

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

plt.plot(x15,x15*1.5,x15,x15*3.0,x15,x15/3.0)
plt.axis()
plt.axis([0,5,-1,13])
plt.show()
```

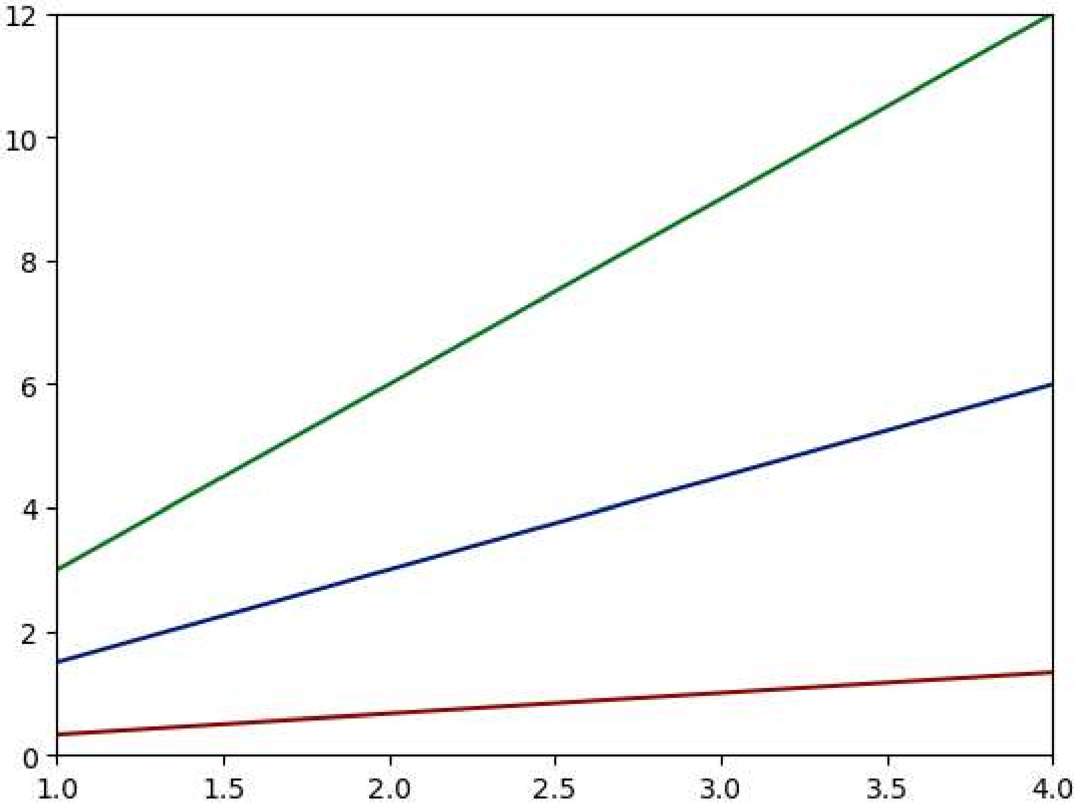


In [70]:

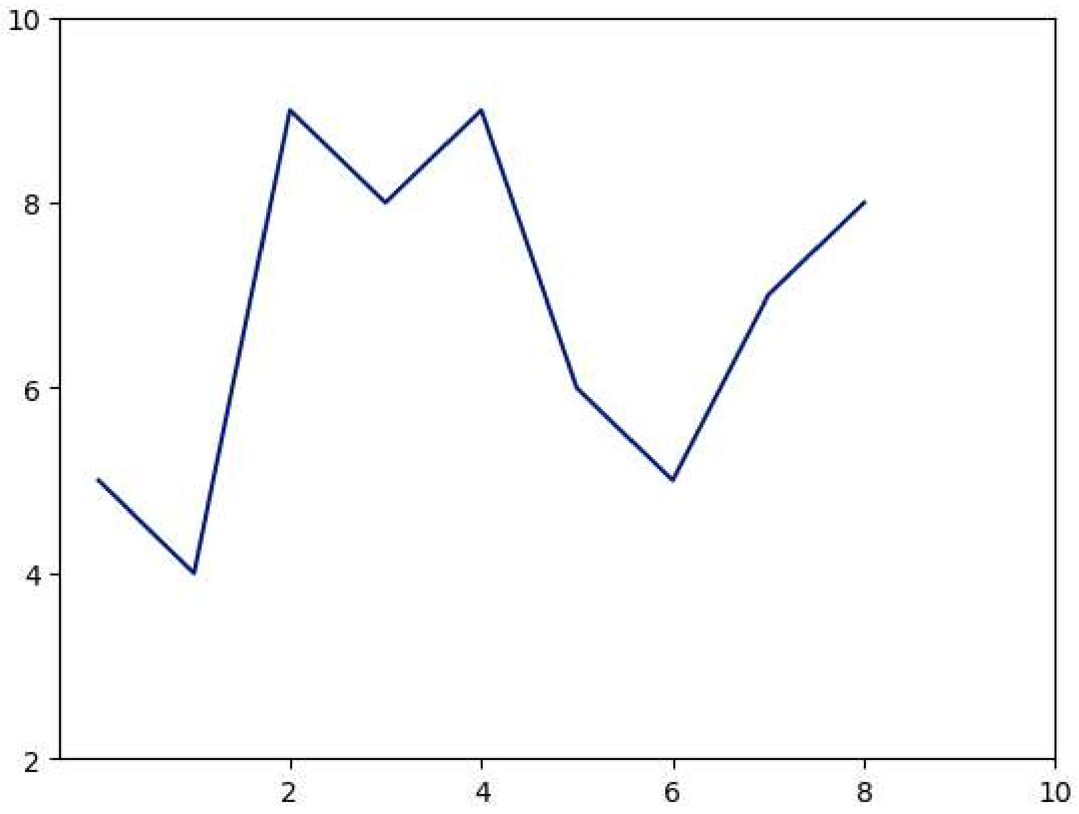
```
x15 =np.arange(1,5)
```

```
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])
```

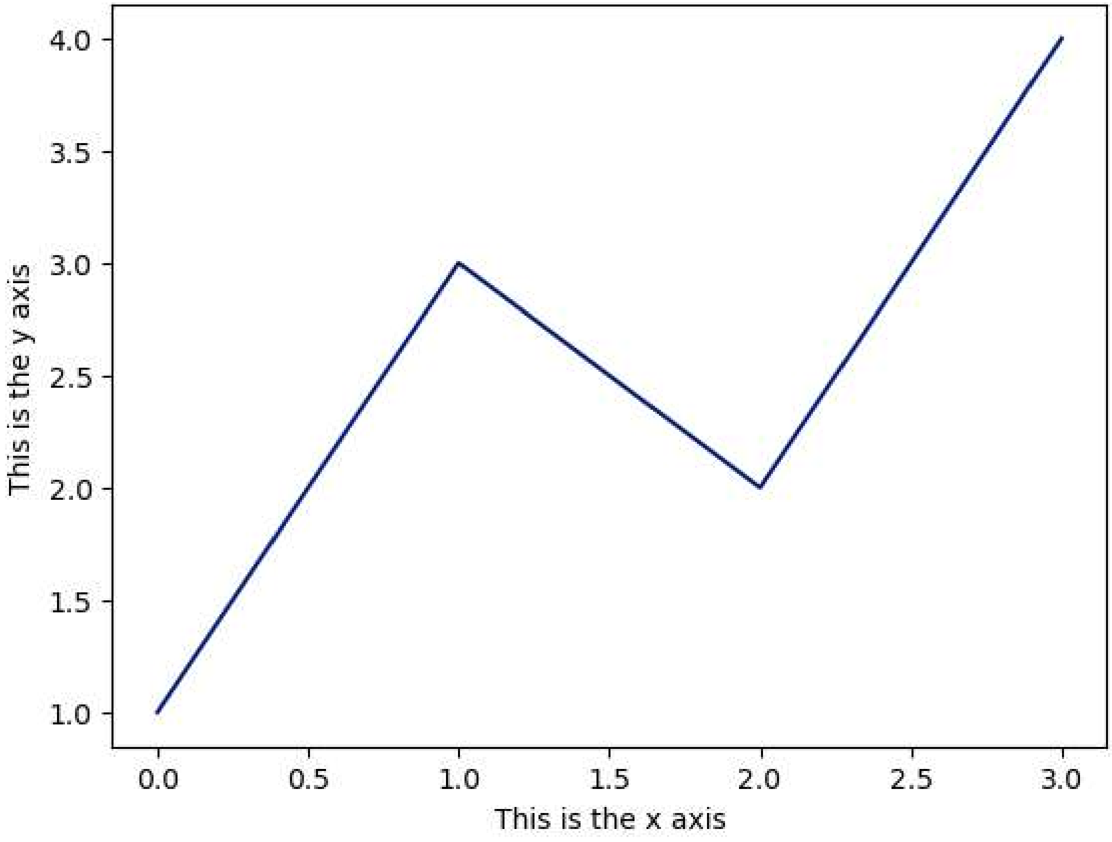
```
Out[70]:  (0.0, 12.0)
```



```
In [72]:  u = [5,4,9,8,9,6,5,7,8]

          plt.plot(u)

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

          plt.show()
```

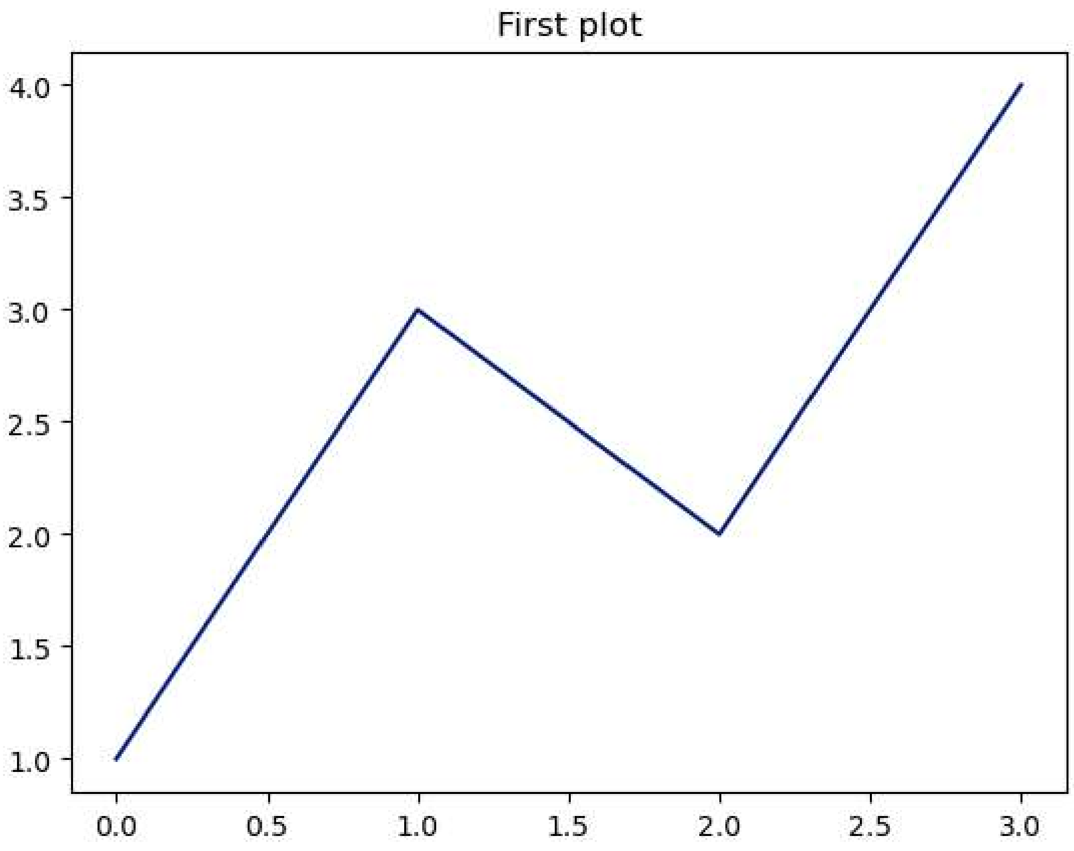


In [74]:
```
plt.plot([1,3,2,4])

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

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

plt.show()
```



In [76]:

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

plt.title('First plot')

plt.show()
```

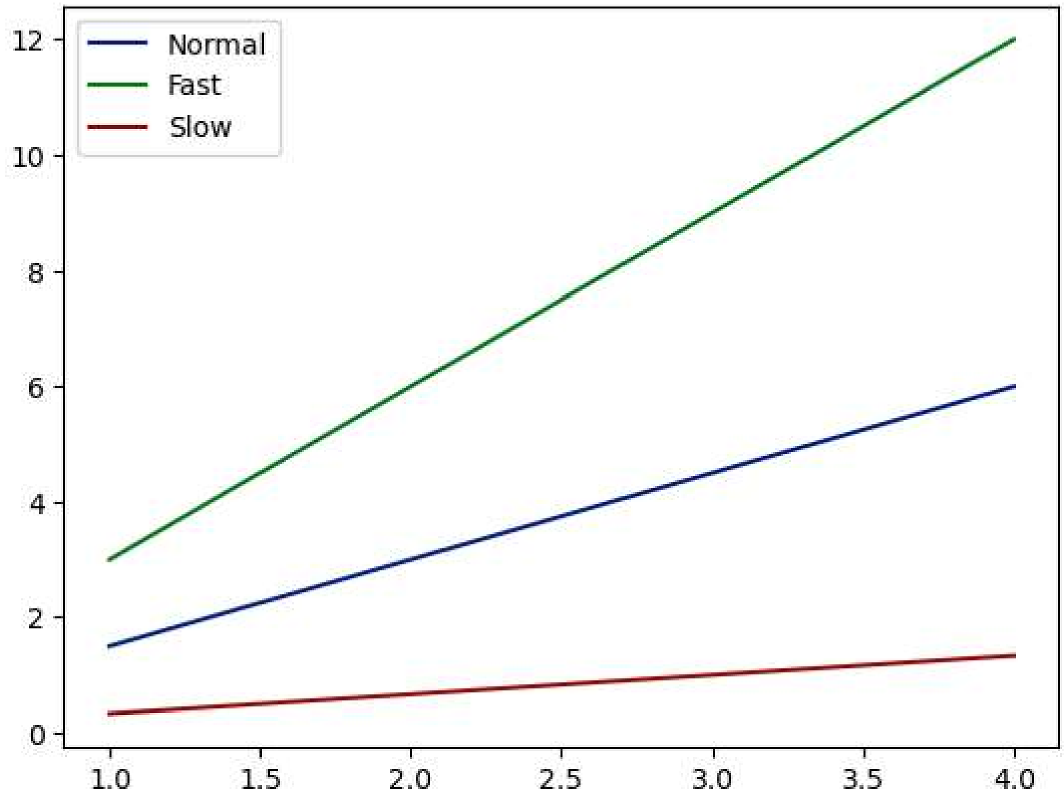


In [78]:

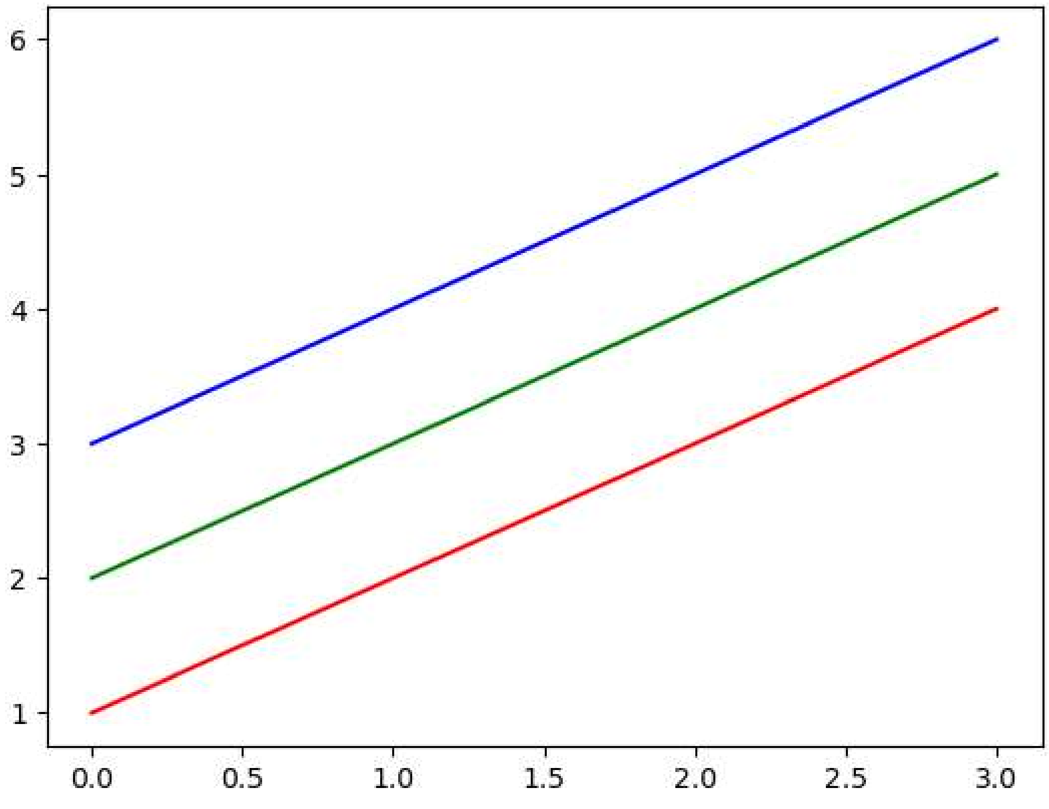
```
x15 = np.arange(1,5)

fig, ax = plt.subplots()

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

ax.legend(['Normal','Fast','Slow'])
```

Out[78]:

```
<matplotlib.legend.Legend at 0x175ae85b860>
```

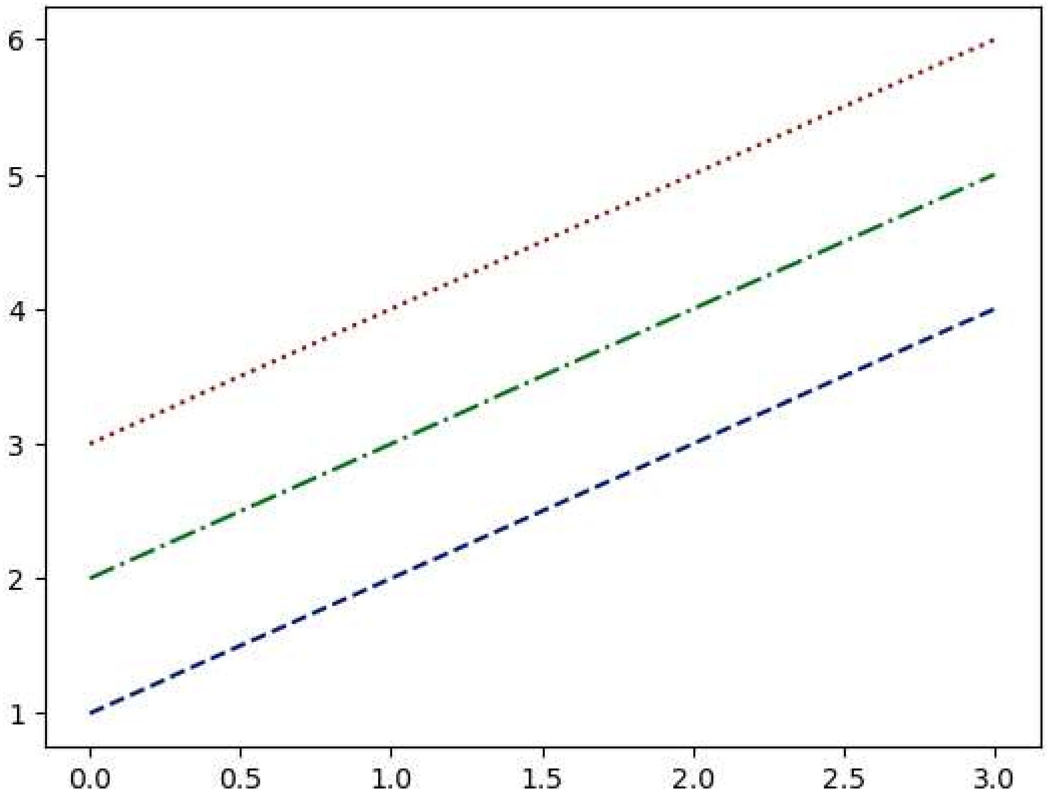


```
In [80]: x16 = np.arange(1,5)

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

         plt.show()
```



In [82]:

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

plt.plot(x16, '--' ,x16+1,'-.',x16+2,':')

plt.show()
```



35. Summary

In this project, I discuss Matplotlib (the basic plotting library in Python) and throw some light on various charts and customization techniques associated with it.

In particular, I discuss Matplotlib object hierarchy, Matplotlib architecture, Pyplot and Object-Oriented architecture. I also discuss subplots which is very important tool to create graphics in Matplotlib.

Then, I discuss various types of plots like line plot, scatter plot, histogram, bar chart, pie chart, box plot, area chart and contour plot.

Finally, I discuss various customization techniques. I discuss how to customize the graphics with styles. I discuss how to add a grid and how to handle axes and ticks. I discuss how to add labels, title and legend. I discuss how to customize the charts with colours and line styles.