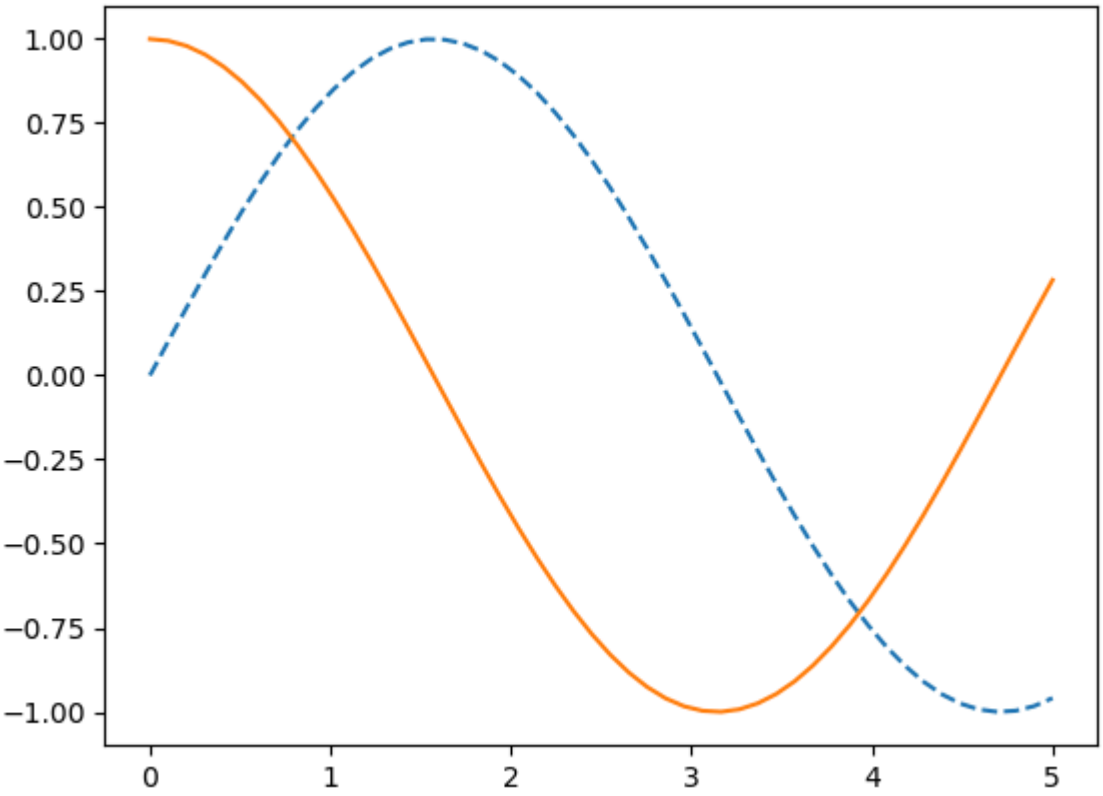
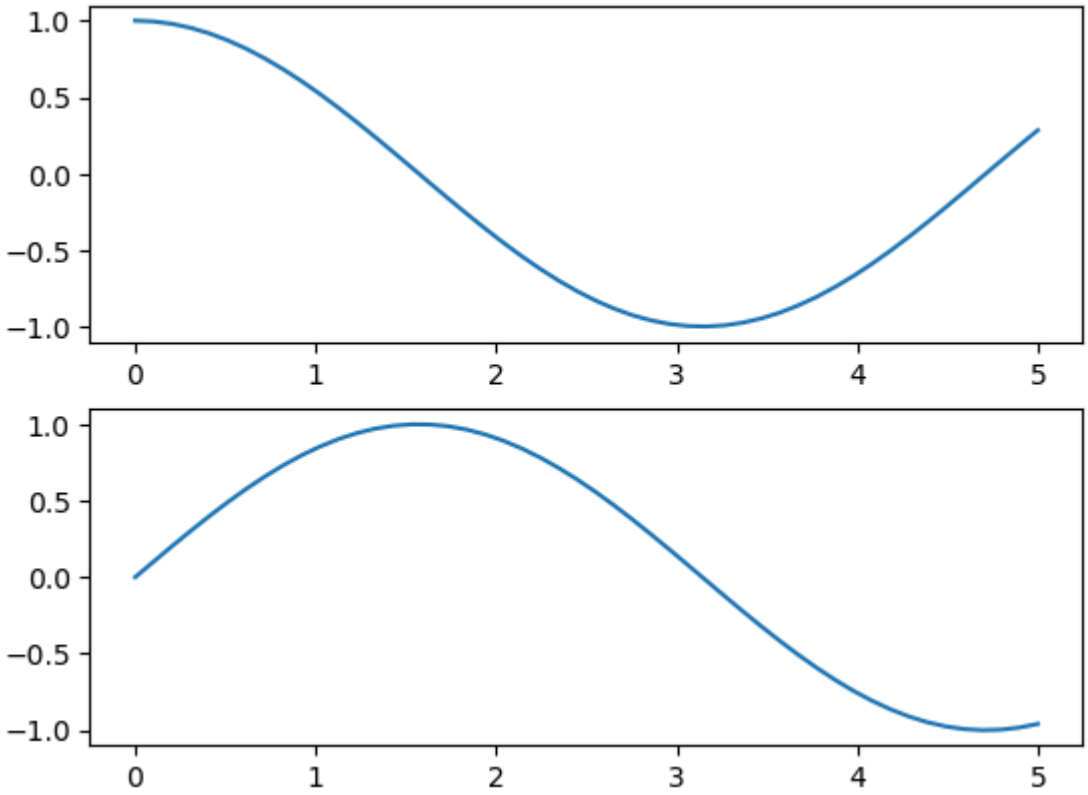
In [6]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [7]:
```
%matplotlib inline
x1 = np.linspace(0,5,50)

fig = plt.figure()

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



In [8]:
```
plt.figure()
plt.subplot(2,1,2)
plt.plot(x1,np.sin(x1))

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

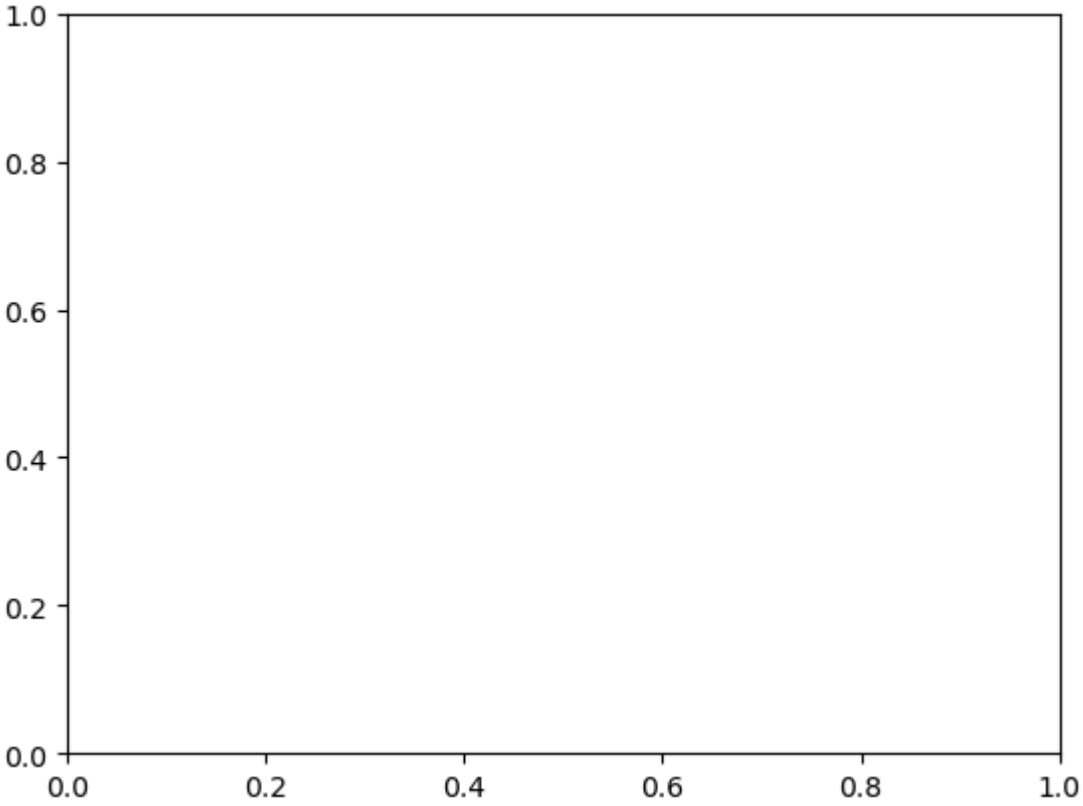


In [9]:
```
print(plt.gcf())
```

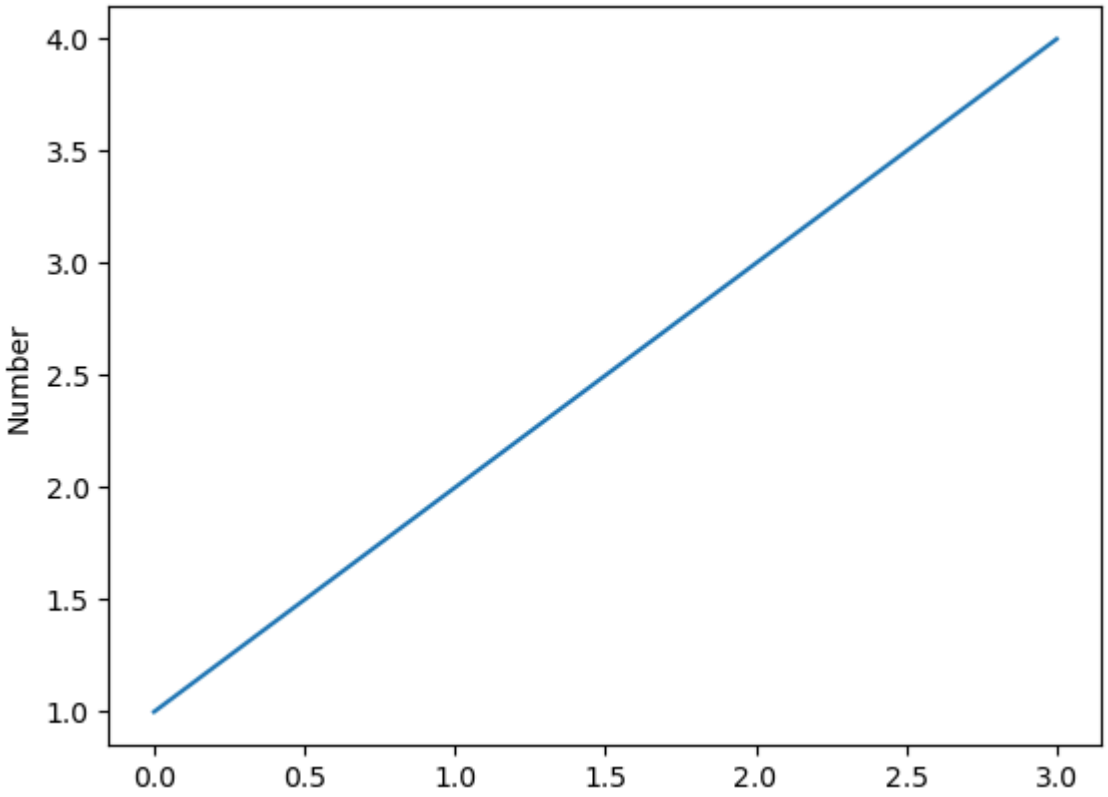
```
Figure(640x480)
```

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

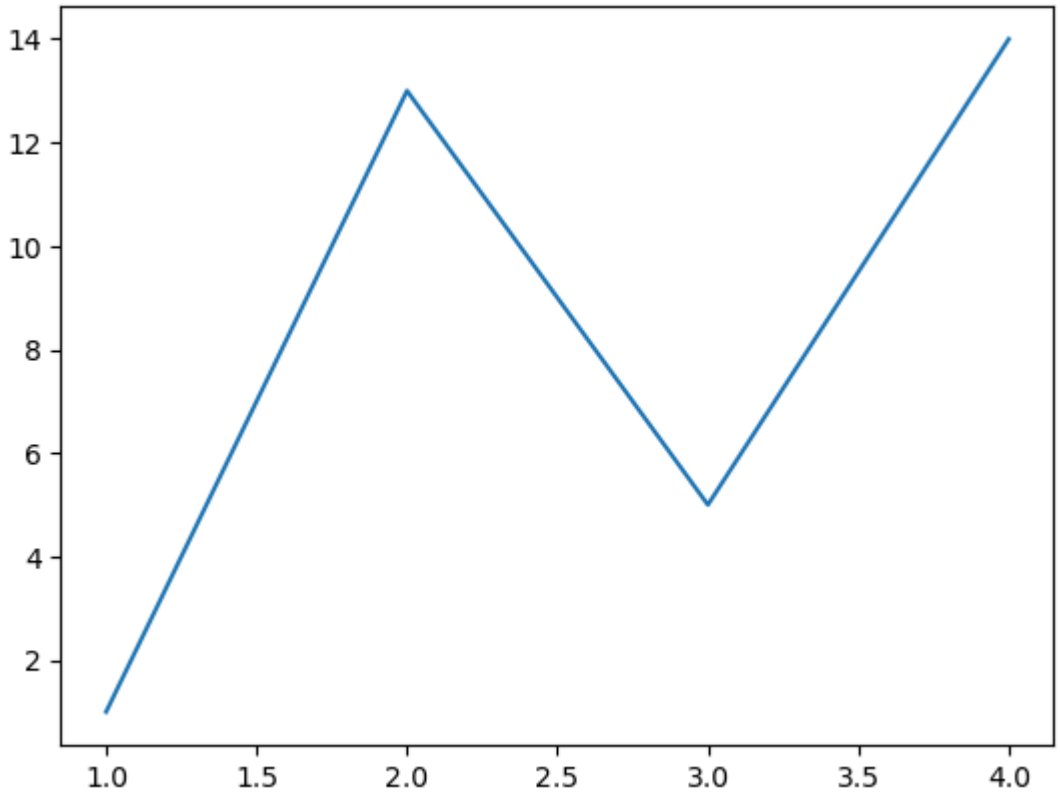
```
Axes(0.125,0.11;0.775x0.77)
```



In [11]:
```
plt.plot([1,2,3,4])
plt.ylabel('Number')
plt.show()
```

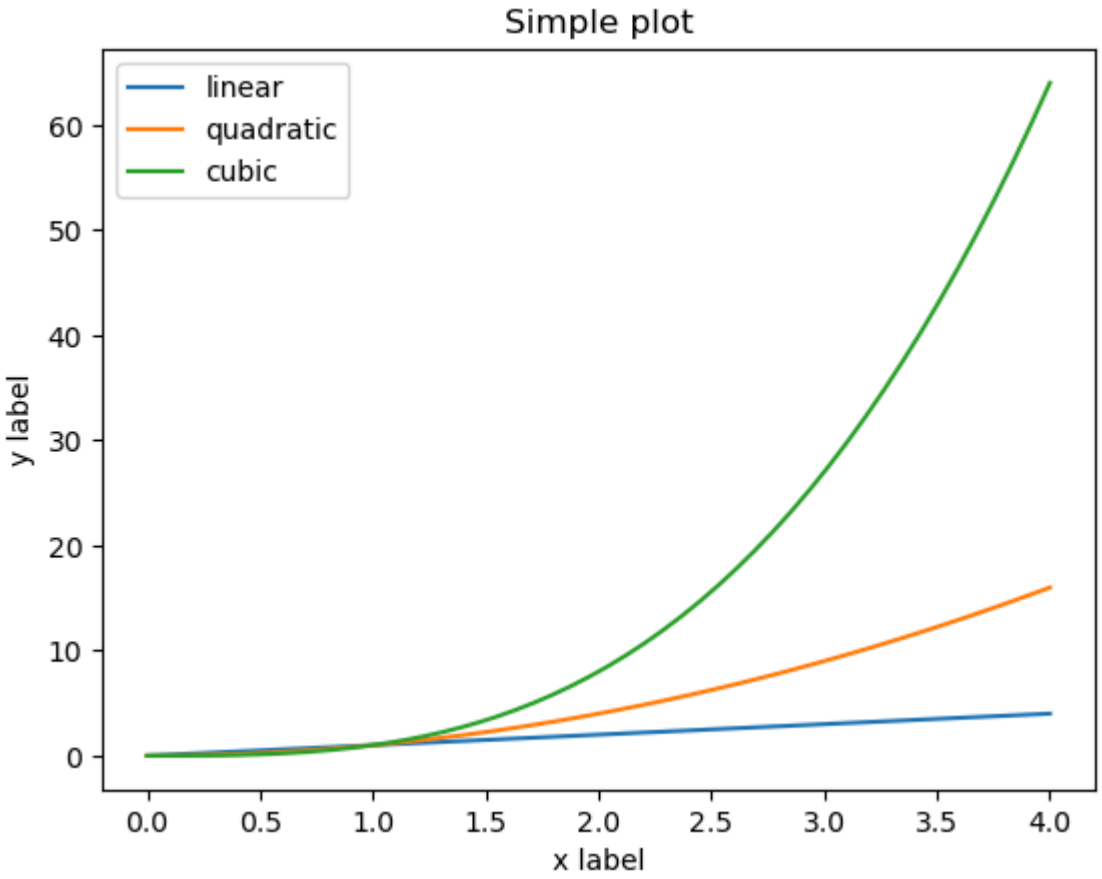


In [12]:
```
plt.plot([1,2,3,4],[1,13,5,14])
plt.show()
```

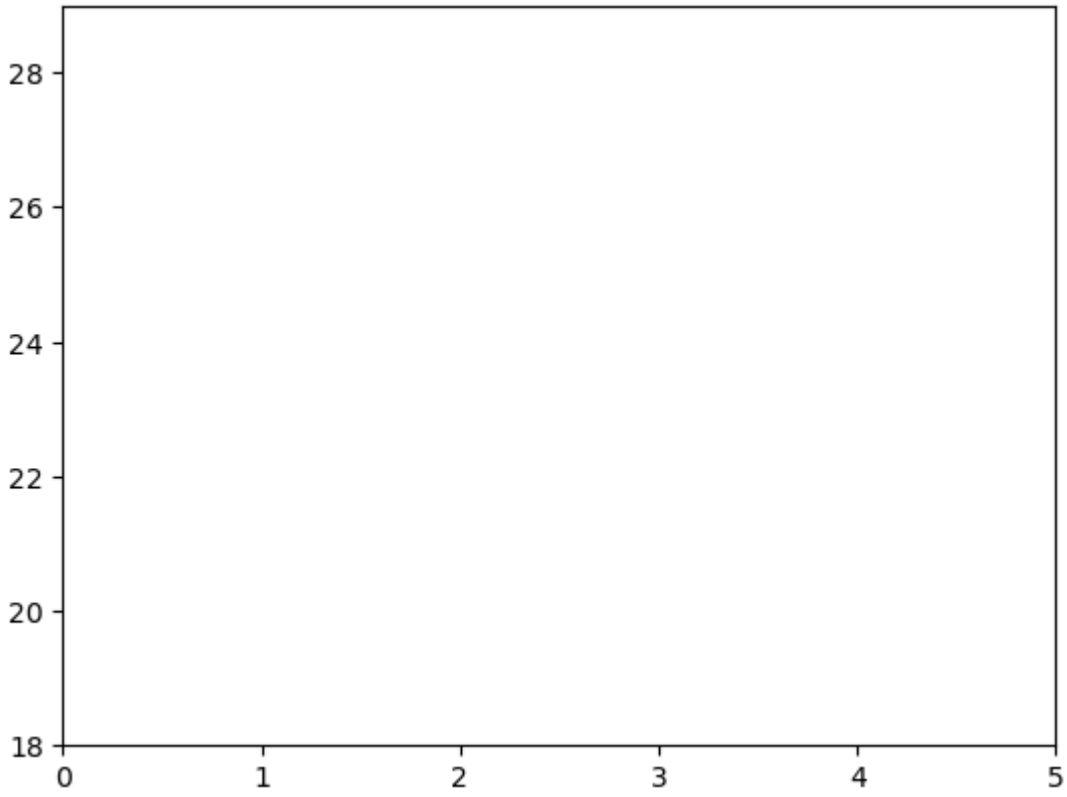


In [13]:
```
x = np.linspace(0,4,50)
plt.plot(x,x,label='linear')
```

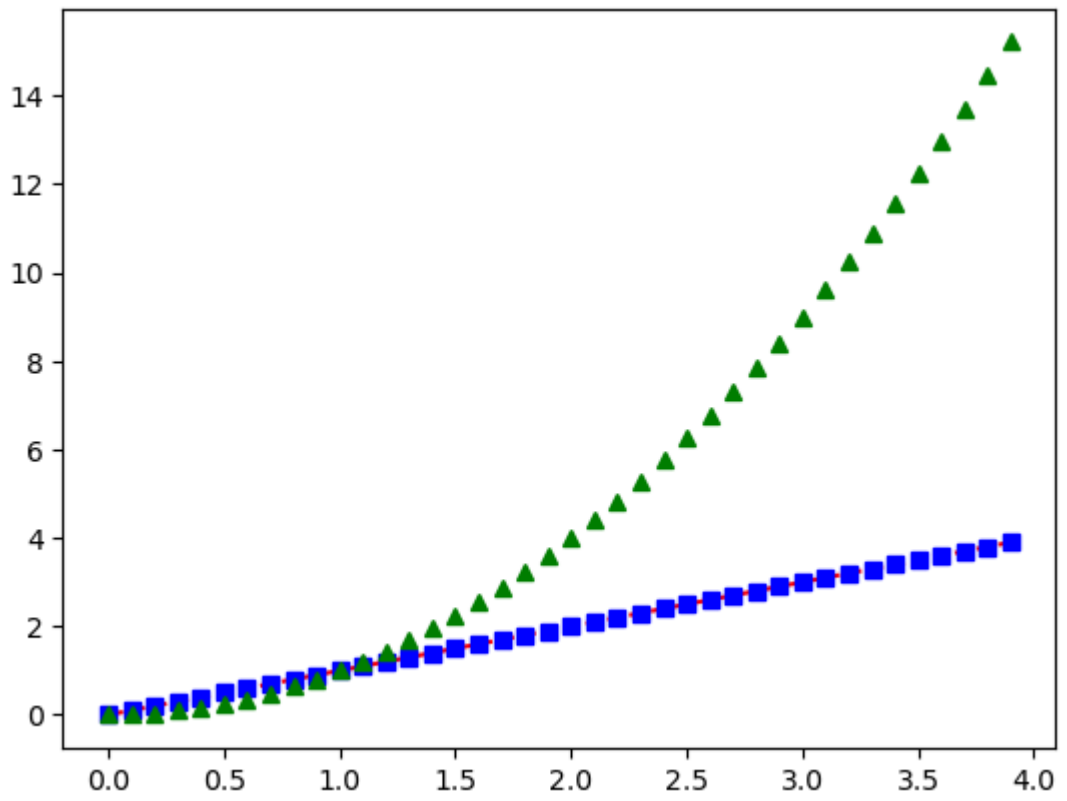
```
plt.plot(x,x**2,label='quadratic')
plt.plot(x,x**3,label='cubic')

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

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



In [14]:
```
plt.plot([1,2,3,4],[1,13,5,14],'go')
plt.axis([0,5,18,29])
plt.show()
```

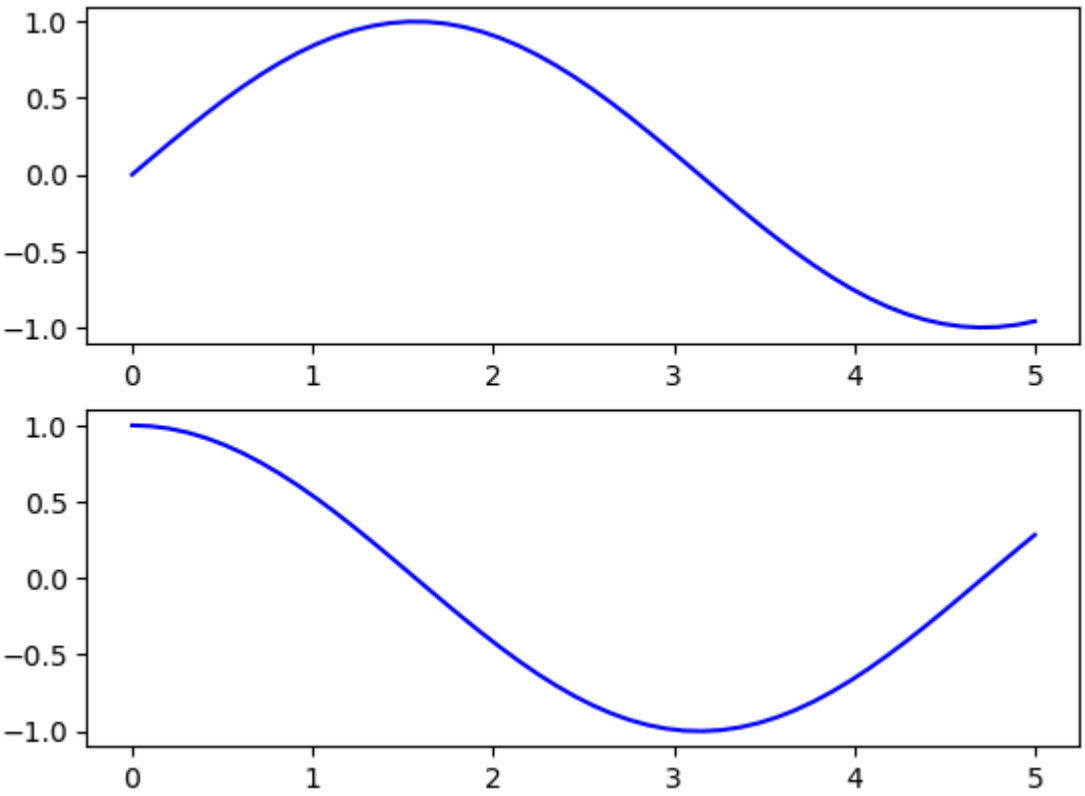


In [15]:
```
t = np.arange(0.,4.,0.1)

plt.plot(t,t,'r--',t,t**1,'bs',t,t**2,'g^')
plt.show()
```

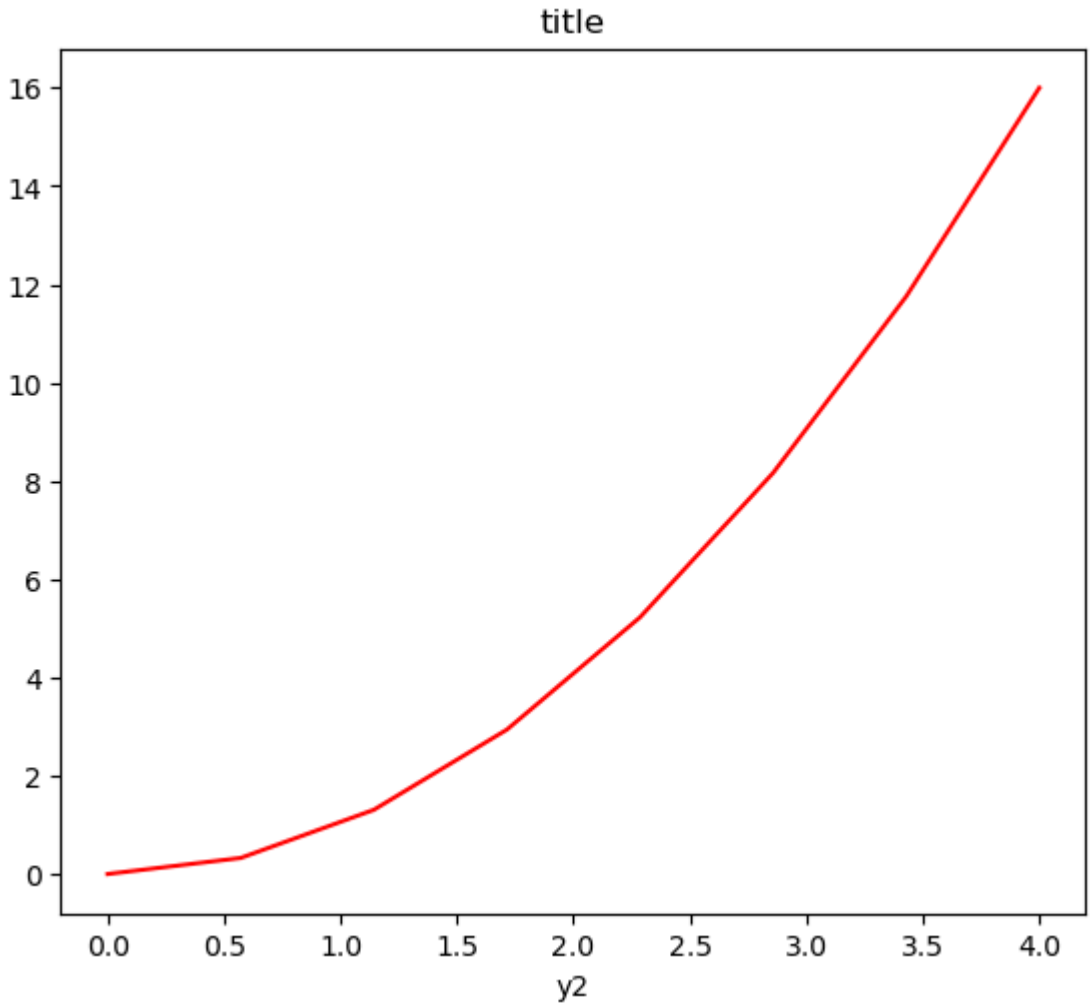


In [16]:
```
fig,ax = plt.subplots(2)

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

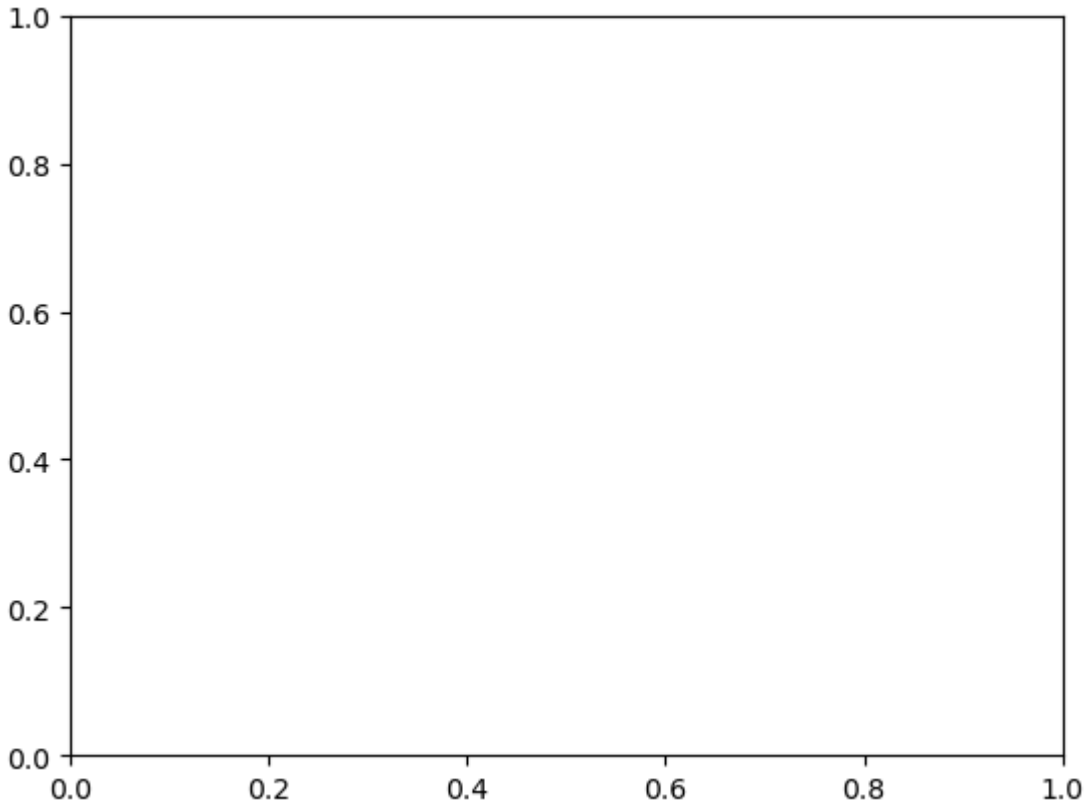
```
ax[1].plot(x1, np.cos(x1),'b-')
plt.show()
```



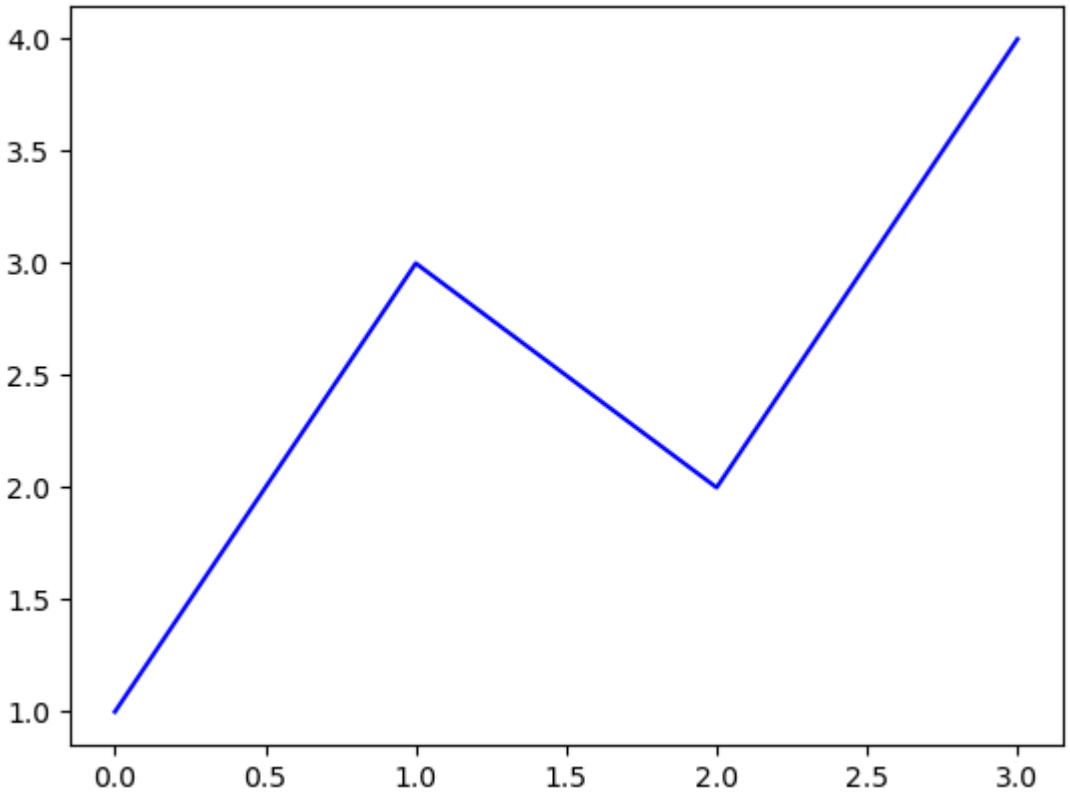
In [18]:
```
fig = plt.figure()

x2=np.linspace(0,4,8)
y2= x2 ** 2

axes = fig.add_axes([0.1,0.4,0.8,0.9])

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

axes.set_xlabel('x2')
axes.set_xlabel('y2')
axes.set_title('title')
plt.show()
```



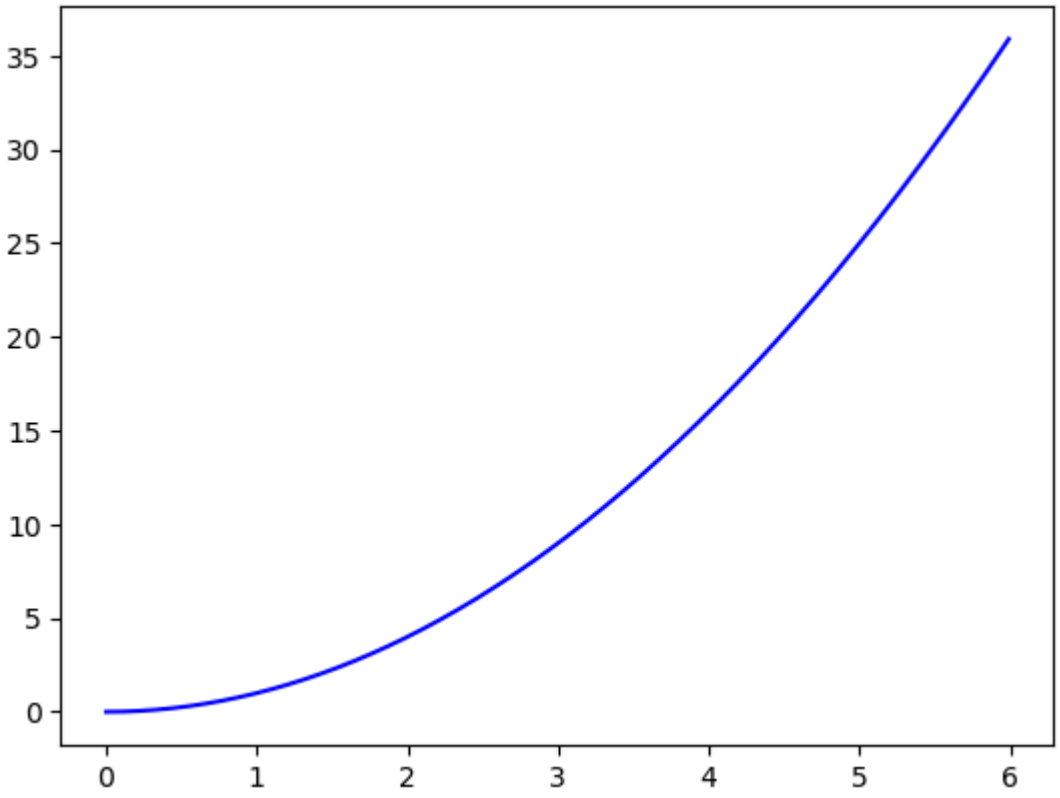
In [19]:
```
fig = plt.figure()
ax = plt.axes()
plt.show()
```



In [20]:
```
plt.plot([1,3,2,4], 'b-')
plt.show()
```

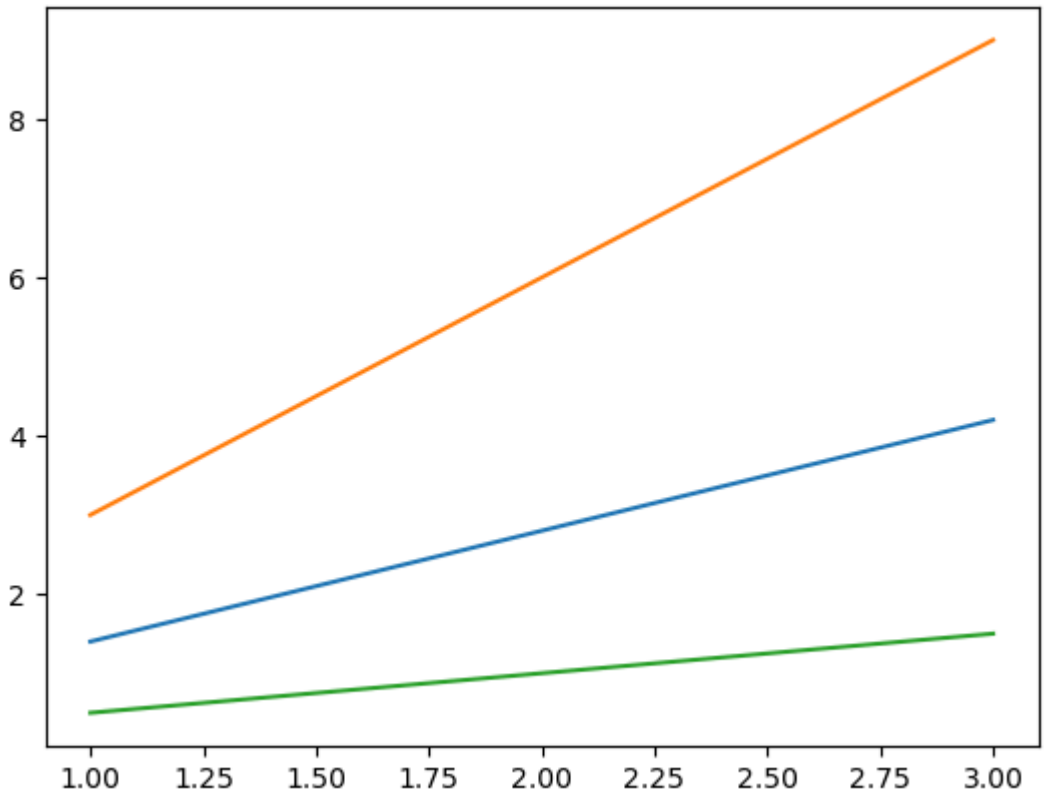


In [22]:
```
x3 = np.arange(0.0,6.0,0.01)
plt.plot(x3,[xi**2 for xi in x3],'b-')
plt.show()
```



In [23]:
```
x4 = range(1,4)
```

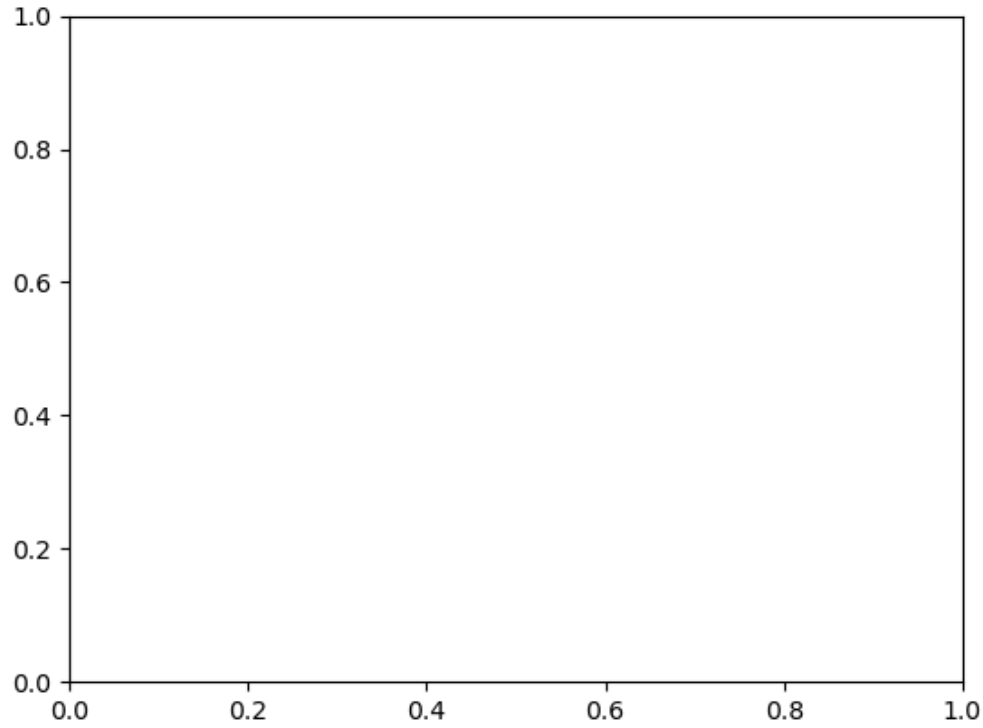
```
plt.plot(x4,[xi*1.4 for xi in x4])
plt.plot(x4,[xi*3 for xi in x4])
plt.plot(x4,[xi/2.0 for xi in x4])
plt.show()
```



In [25]:
```
fig.savefig('plot1.png')
```

In [29]:
```
from IPython.display import Image
Image('plot1.png')
```

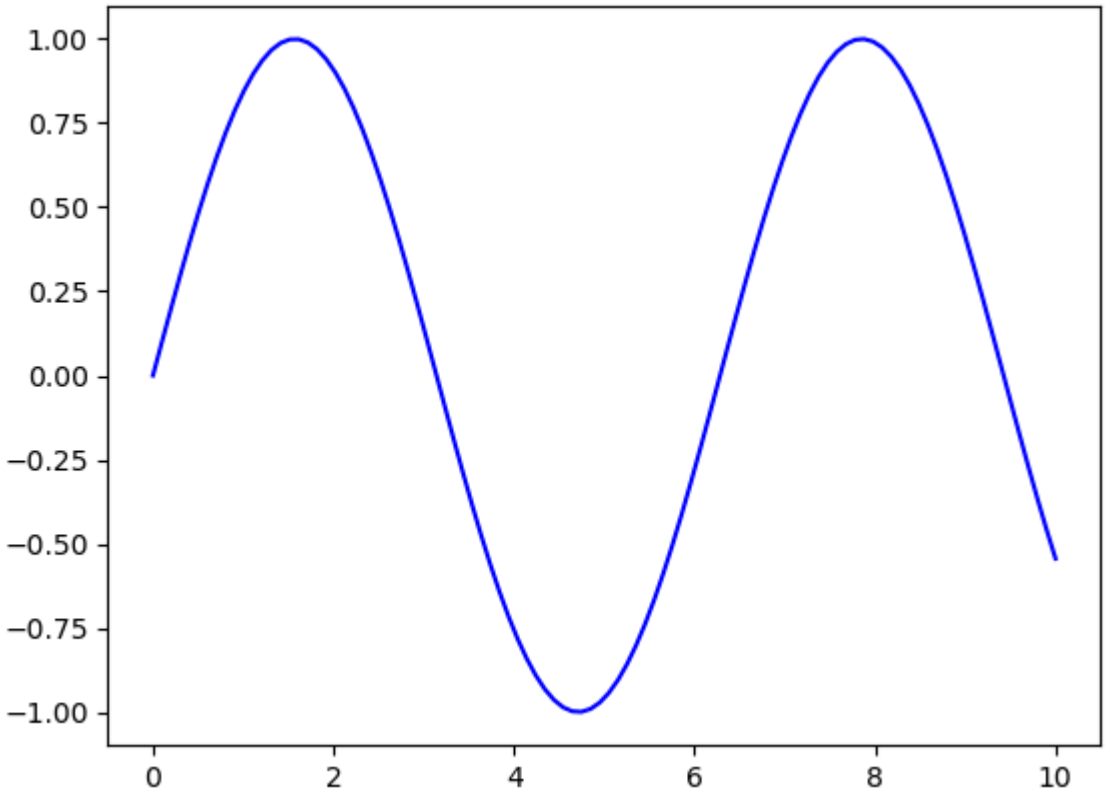
Out[29]:



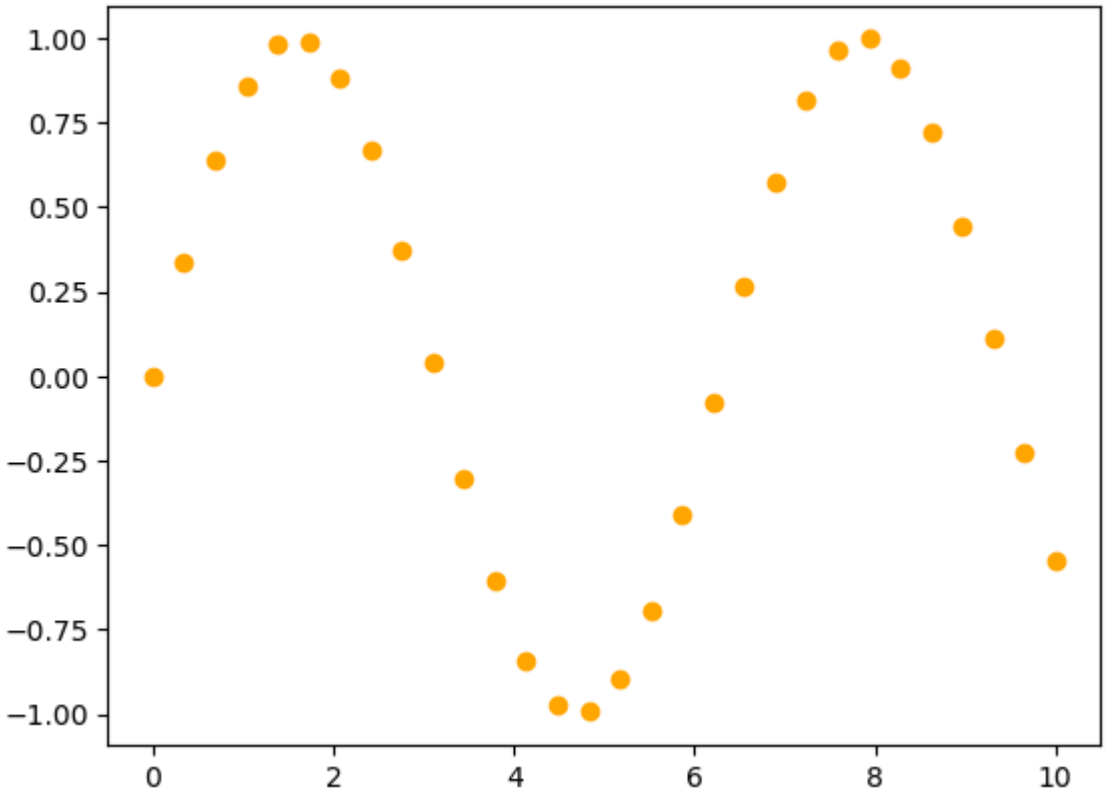
In [30]:
```
fig.canvas.get_supported_filetypes()
```

Out[30]:
```
{'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'}
```

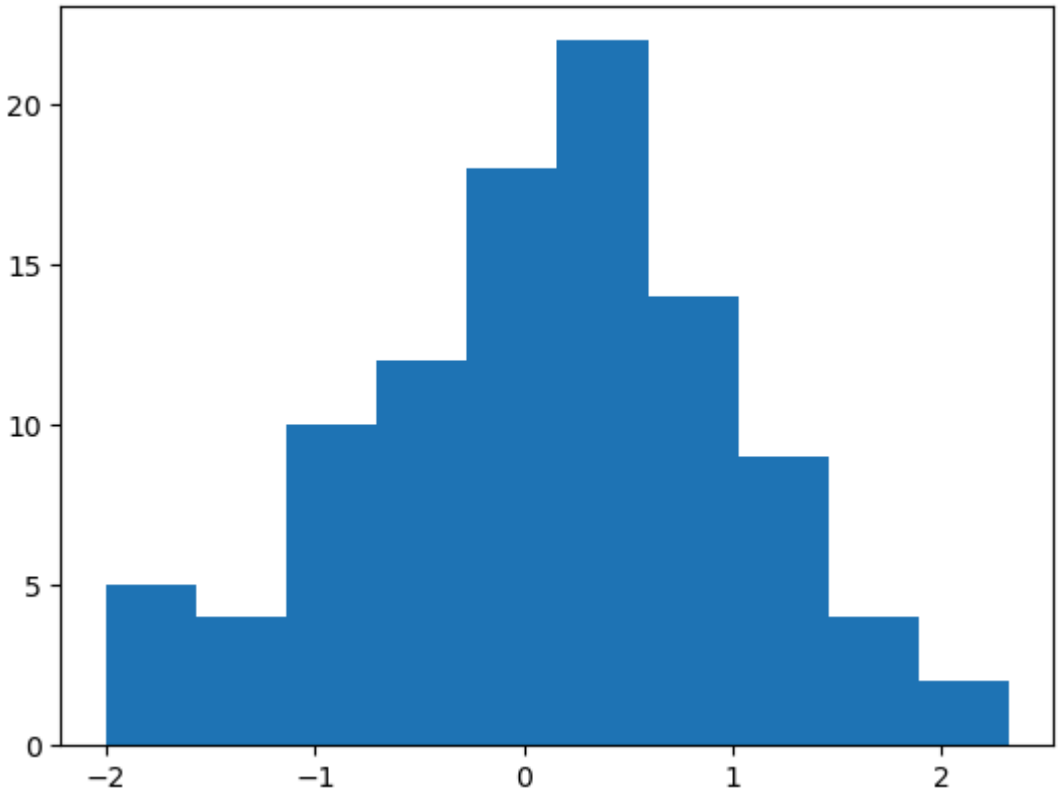
In [32]:
```
fig = plt.figure()
ax= plt.axes()
x5= np.linspace(0,10,100)
ax.plot(x5,np.sin(x5),'b-')
plt.show()
```



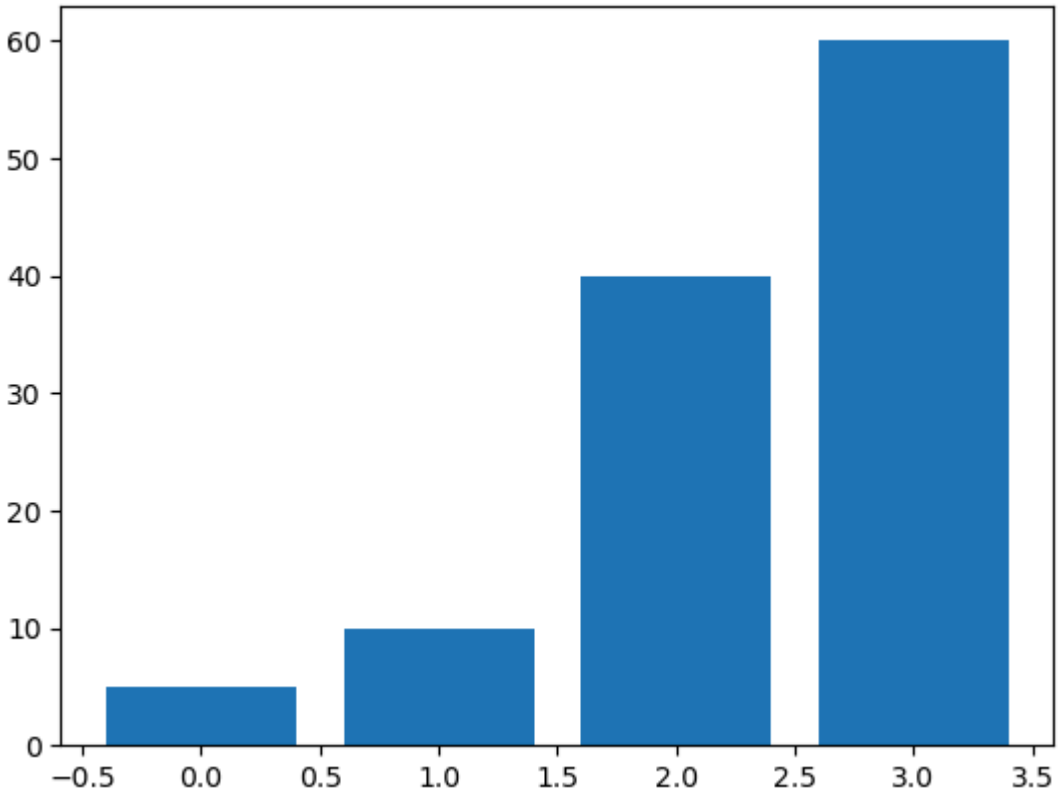
In [33]:
```
x6 = np.linspace(0,10,30)
y6 = np.sin(x6)
plt.plot(x6,y6,'o',color ='orange');
plt.show()
```



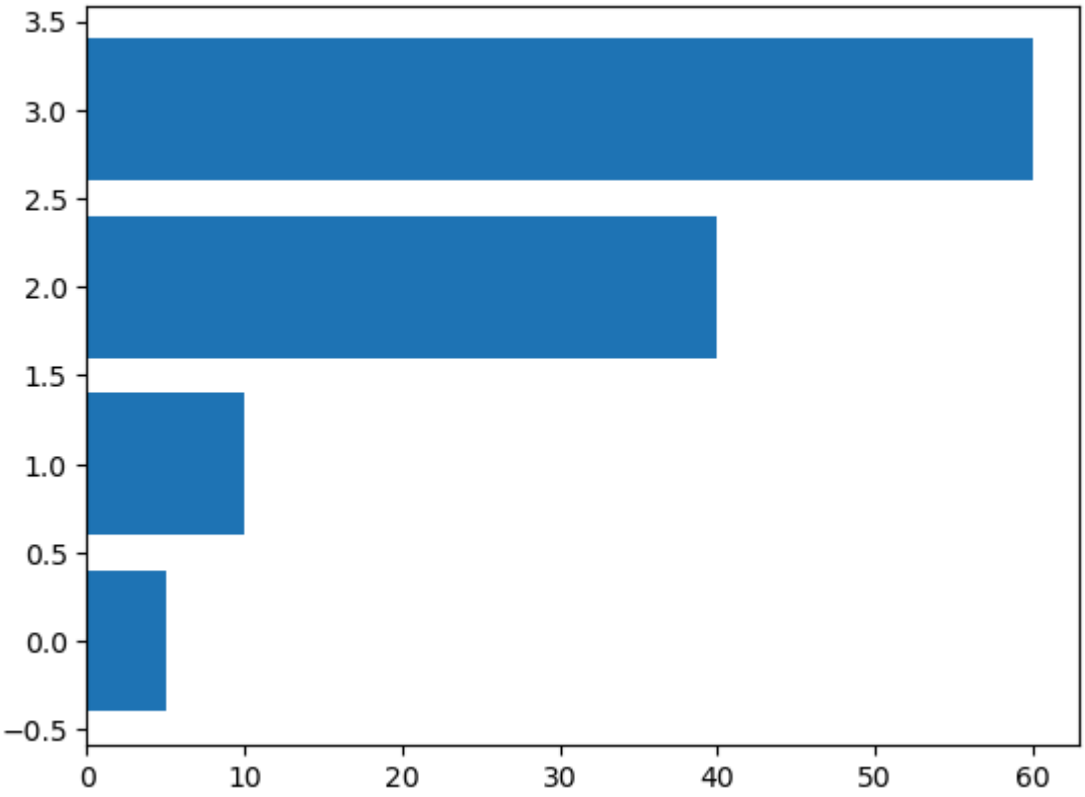
In [35]:
```
data1 = np.random.randn(100)
plt.hist(data1)
plt.show()
```



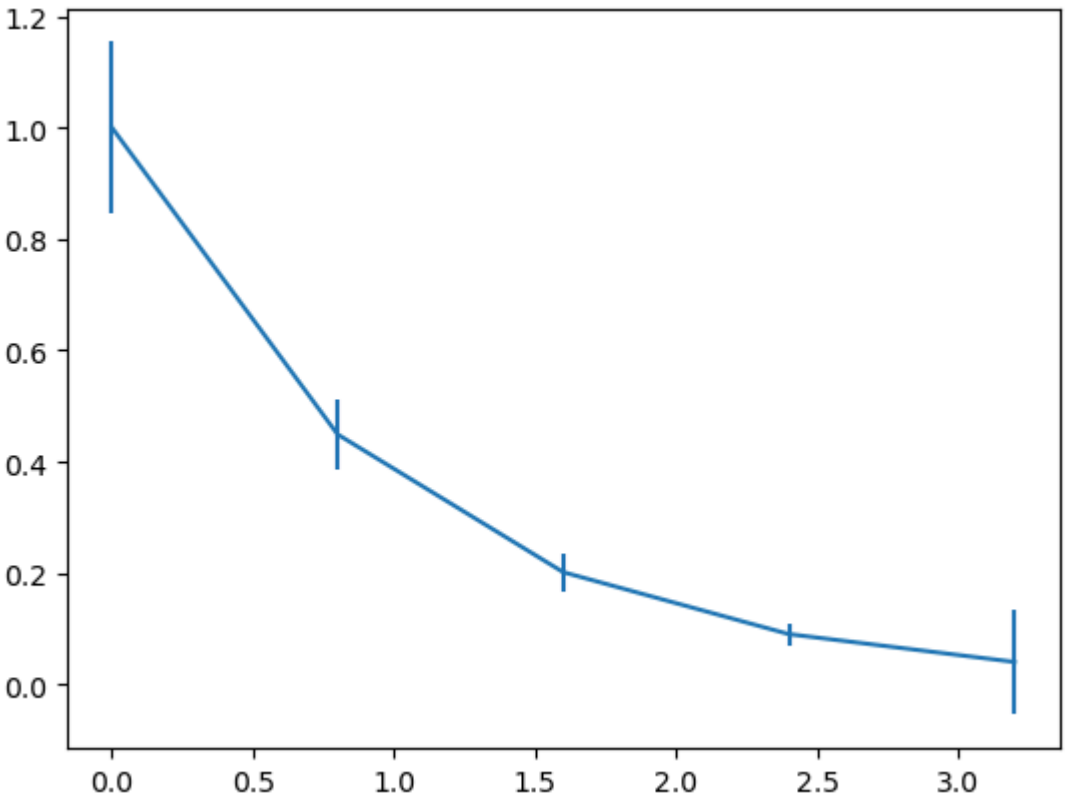
In [36]:
```
data2 = [5.,10.,40.,60.]
plt.bar(range(len(data2)),data2)
plt.show()
```



In [38]:
```
data2 = [5.,10.,40.,60.]
plt.barh(range(len(data2)),data2)
plt.show()
```

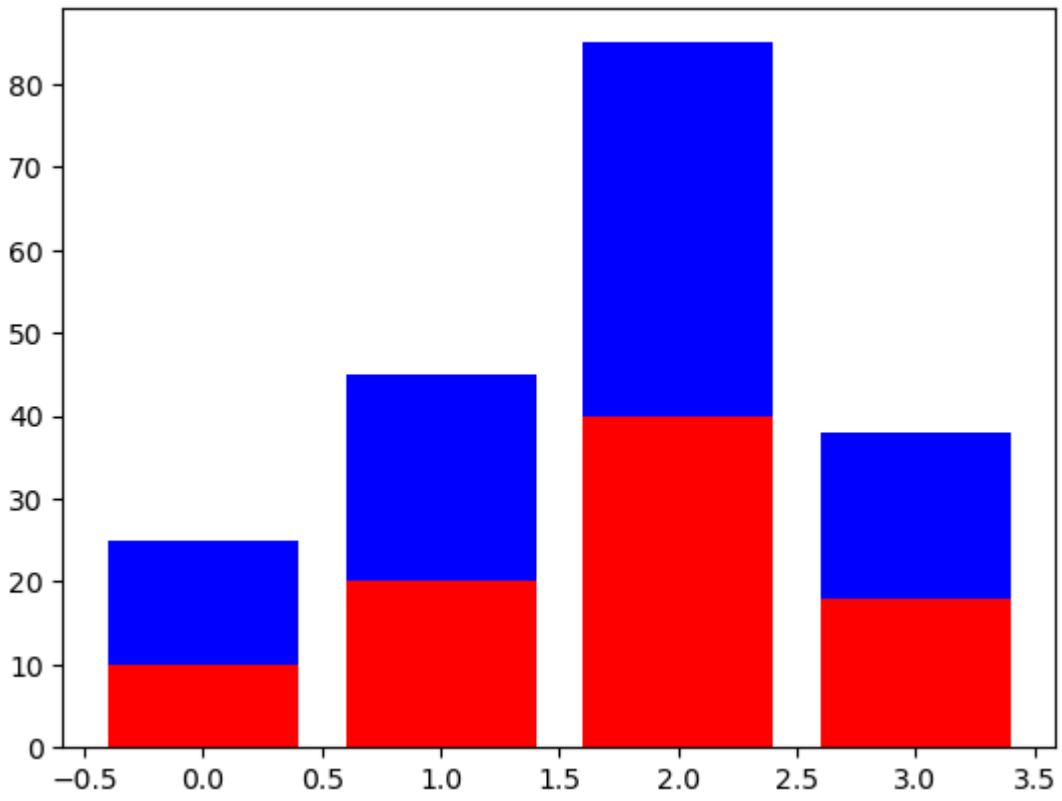


In [40]:
```
x9 = np.arange(0,4,0.8)
y9 = np.exp(-x9)

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

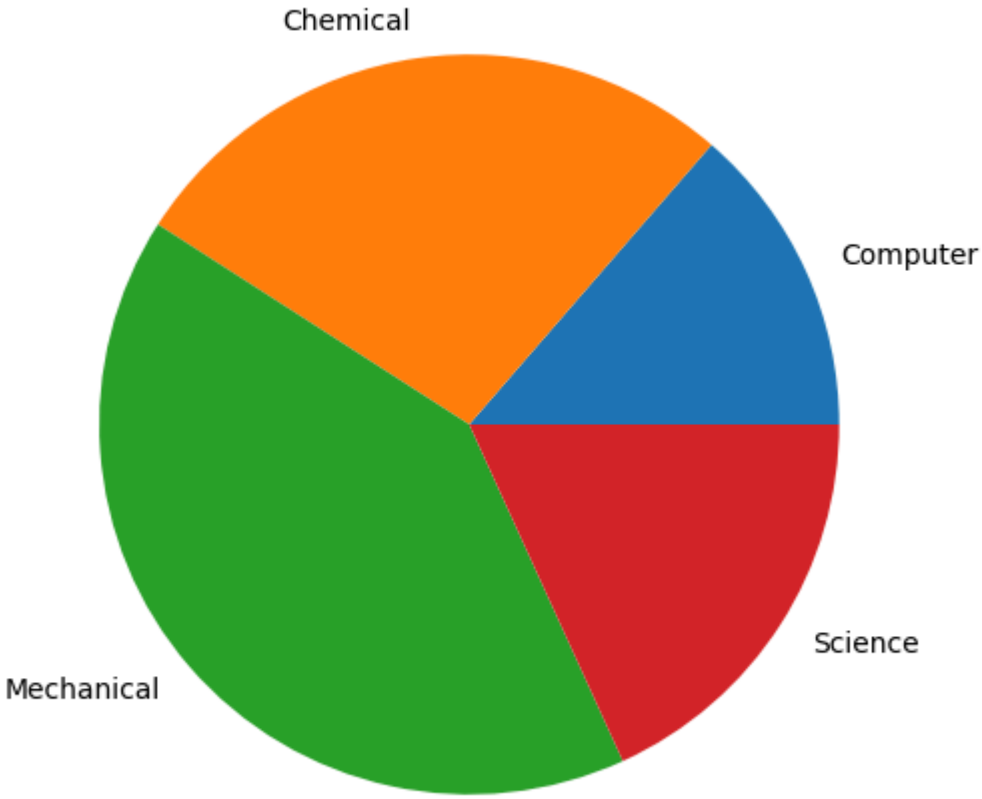


In [41]:
```
A = [ 10.,20.,40.,18.]
B = [15.,25.,45.,20.]
```

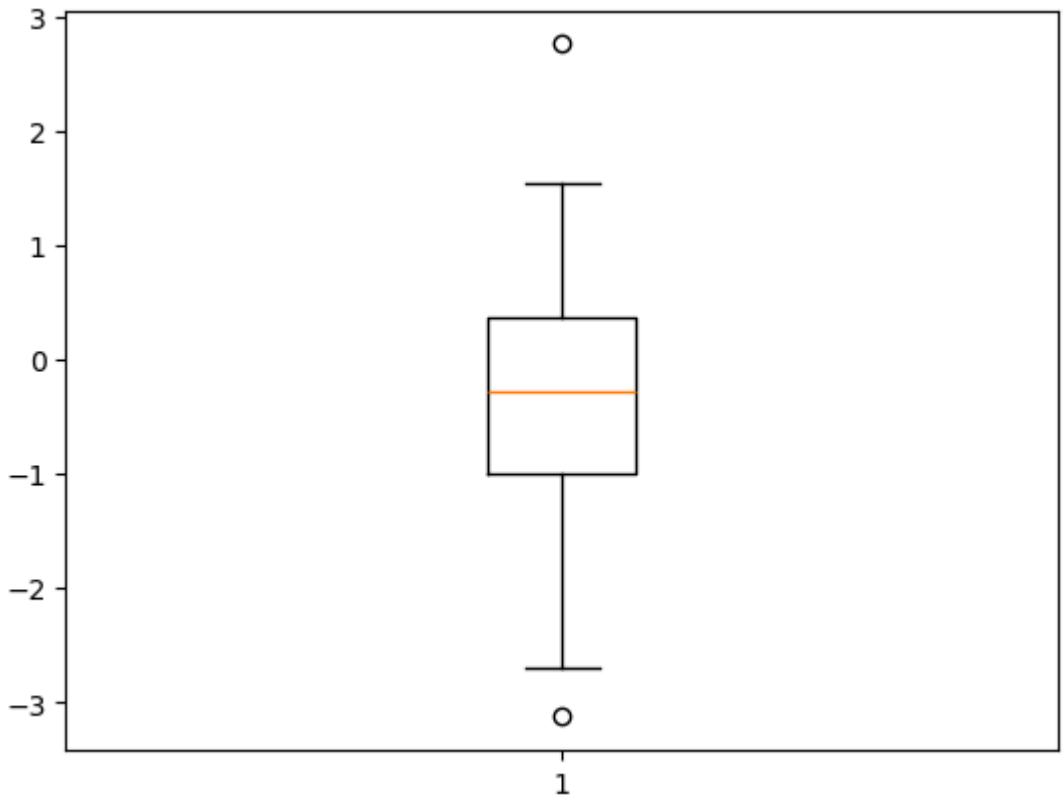
```
z2=range(4)
plt.bar(z2,A,color ='r')
plt.bar(z2,B,color ='b',bottom = A)
plt.show()
```



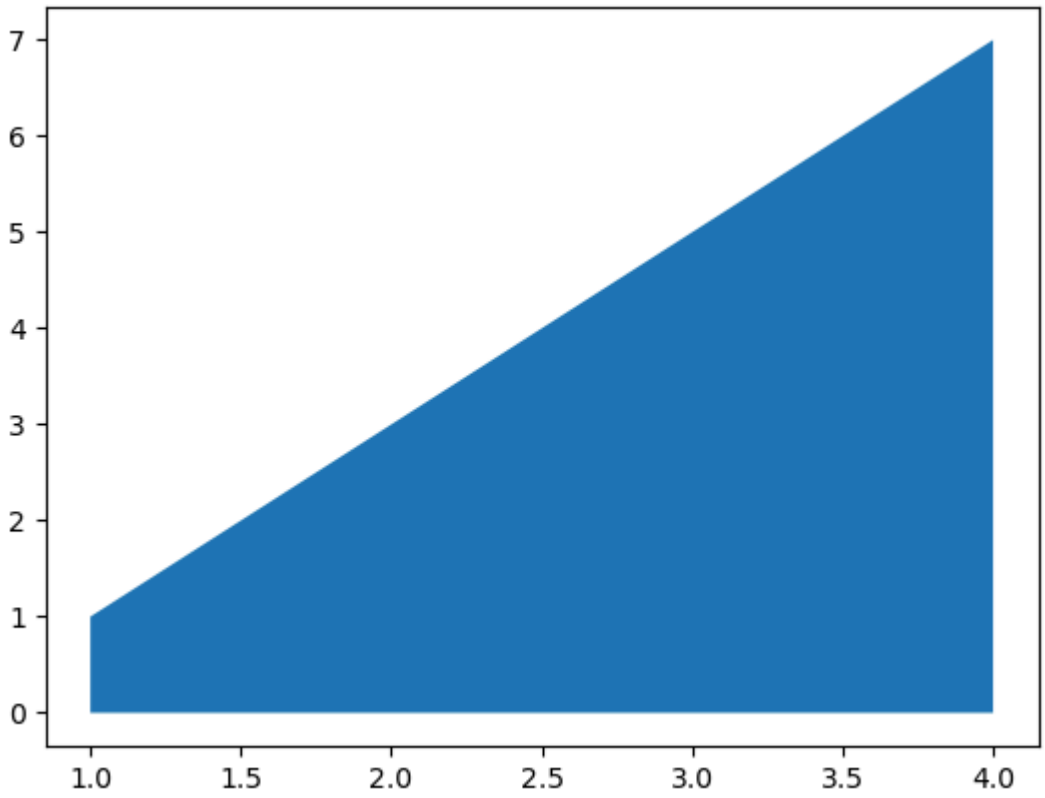
In [42]:
```
plt.figure(figsize = (6,6))
x10 =[15,30,45,20]
labels=['Computer','Chemical','Mechanical','Science']
plt.pie(x10,labels=labels);
plt.show()
```



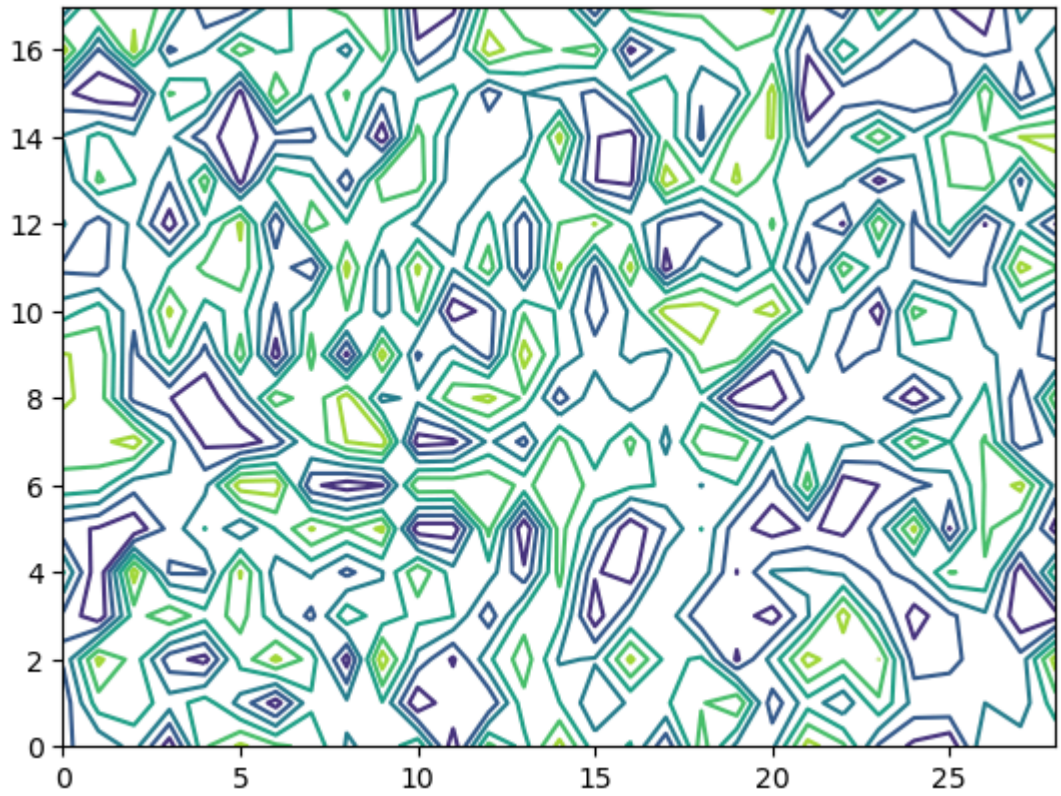
In [45]:
```
date3 = np.random.randn(50)
plt.boxplot(date3)
plt.show()
```



In [47]:
```
x12 = range(1,5)
y12 = [1,3,5,7]
plt.fill_between(x12,y12)
plt.show()
```



In [48]:
```
matrix1 = np.random.rand(18,29)

cp = plt.contour(matrix1)
plt.show()
```

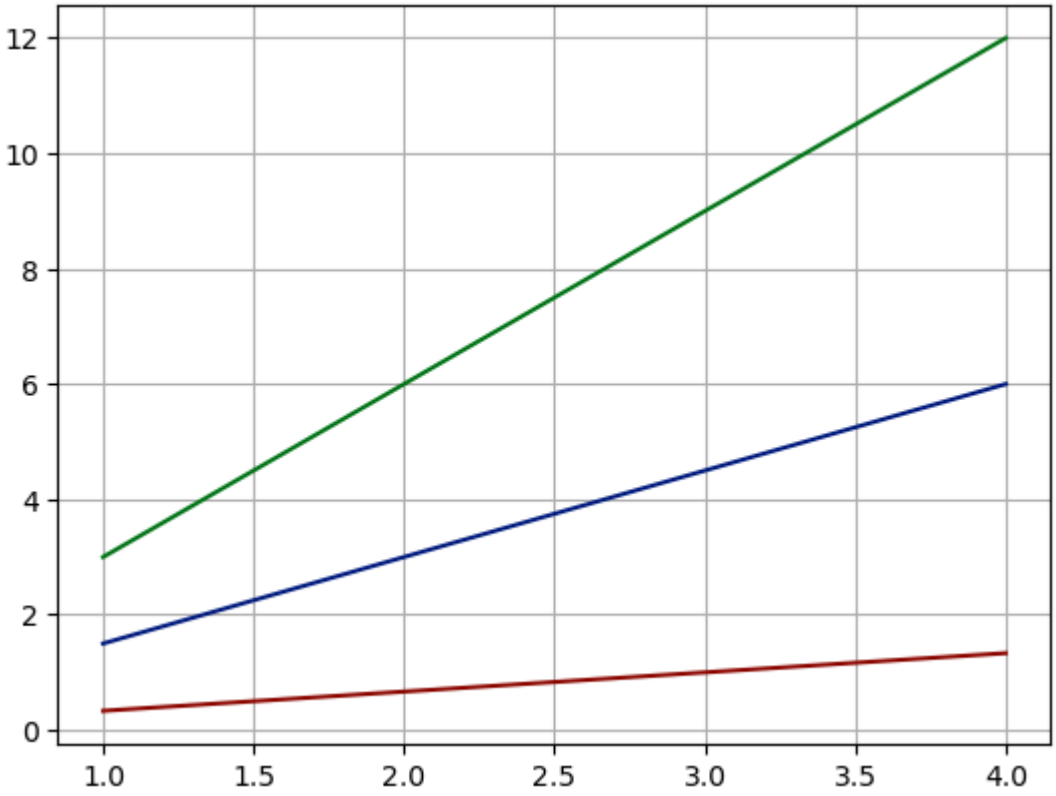


In [50]:
```
print(plt.style.available)
```

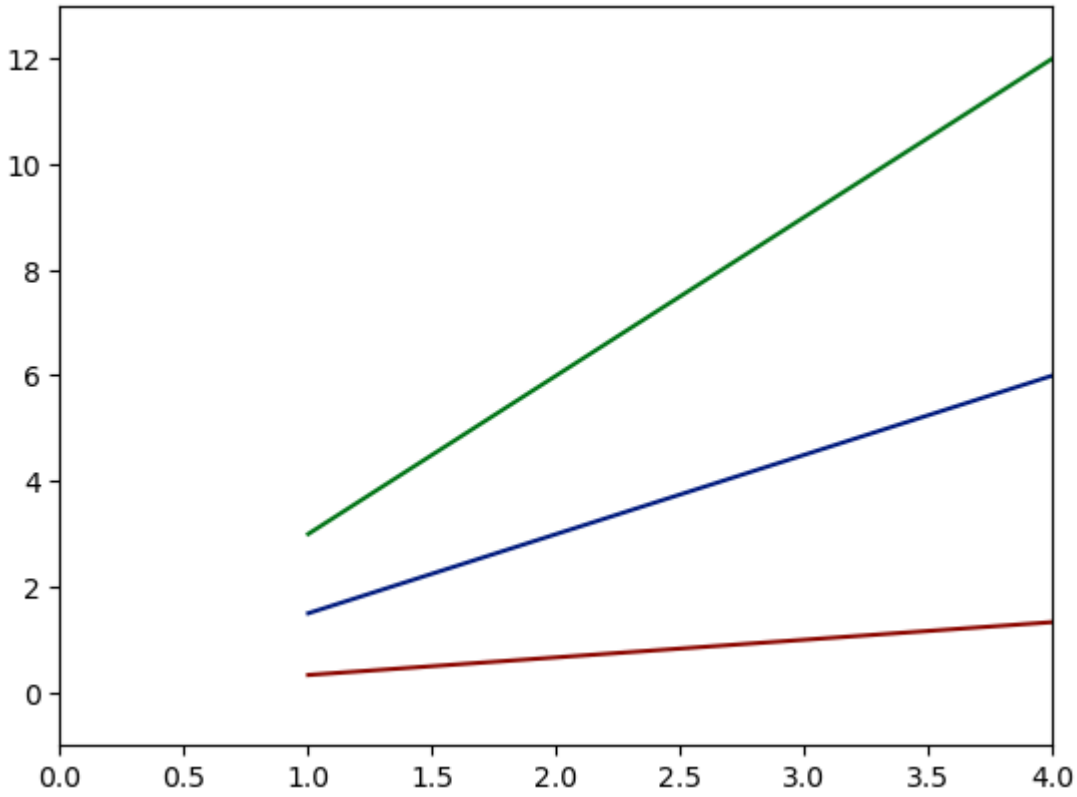
```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid',
'bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot', 'graysc
ale', 'seaborn-v0_8', 'seaborn-v0_8-bright', 'seaborn-v0_8-colorblind', 'seaborn-
v0_8-dark', 'seaborn-v0_8-dark-palette', 'seaborn-v0_8-darkgrid', '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 [51]:
```
plt.style.use('seaborn-v0_8-dark-palette')
```

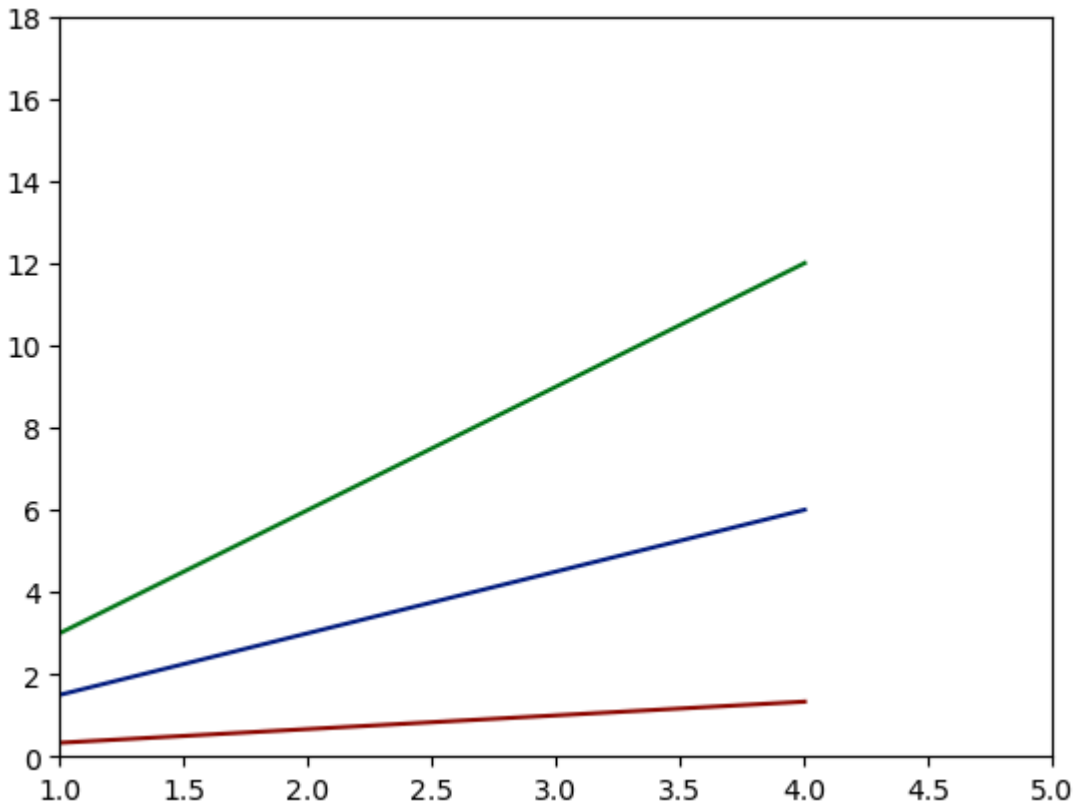
In [53]:
```
x15 =np.arange(1,5)

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

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



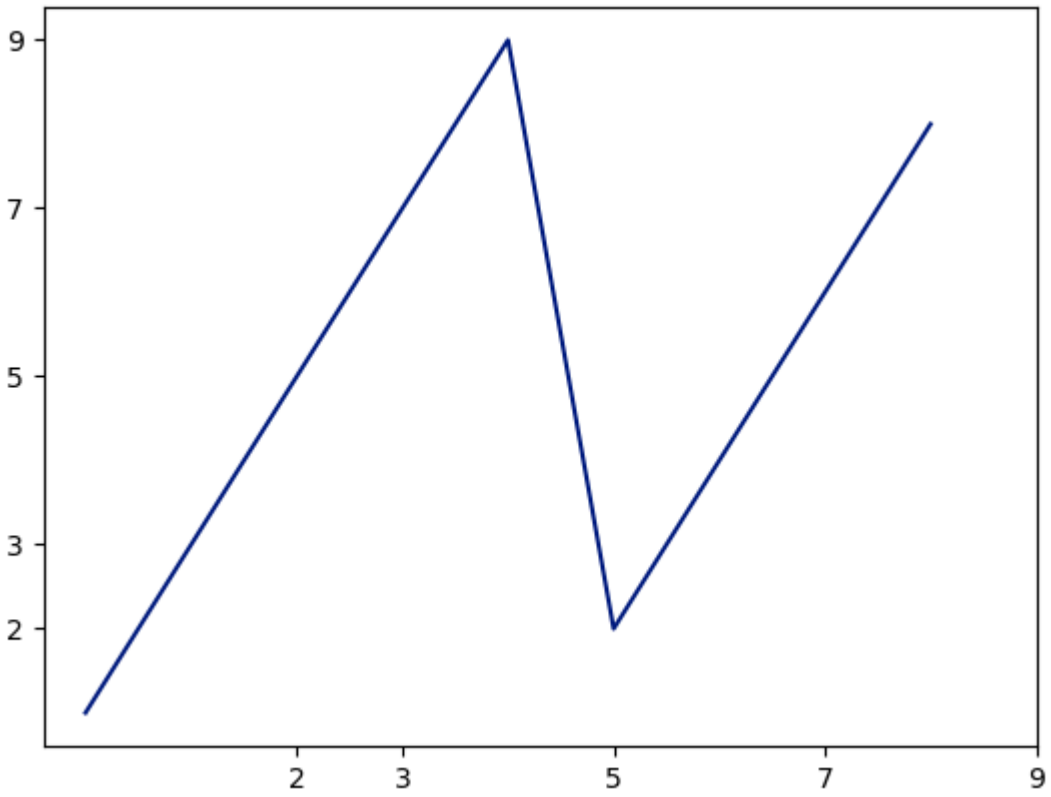
In [55]:
```
x16 =np.arange(1,5)

plt.plot(x15,x15*1.5,x15,x15*3.0,x15,x15/3.0)
plt.axis()
plt.axis([0,4,-1,13])
plt.show()
```



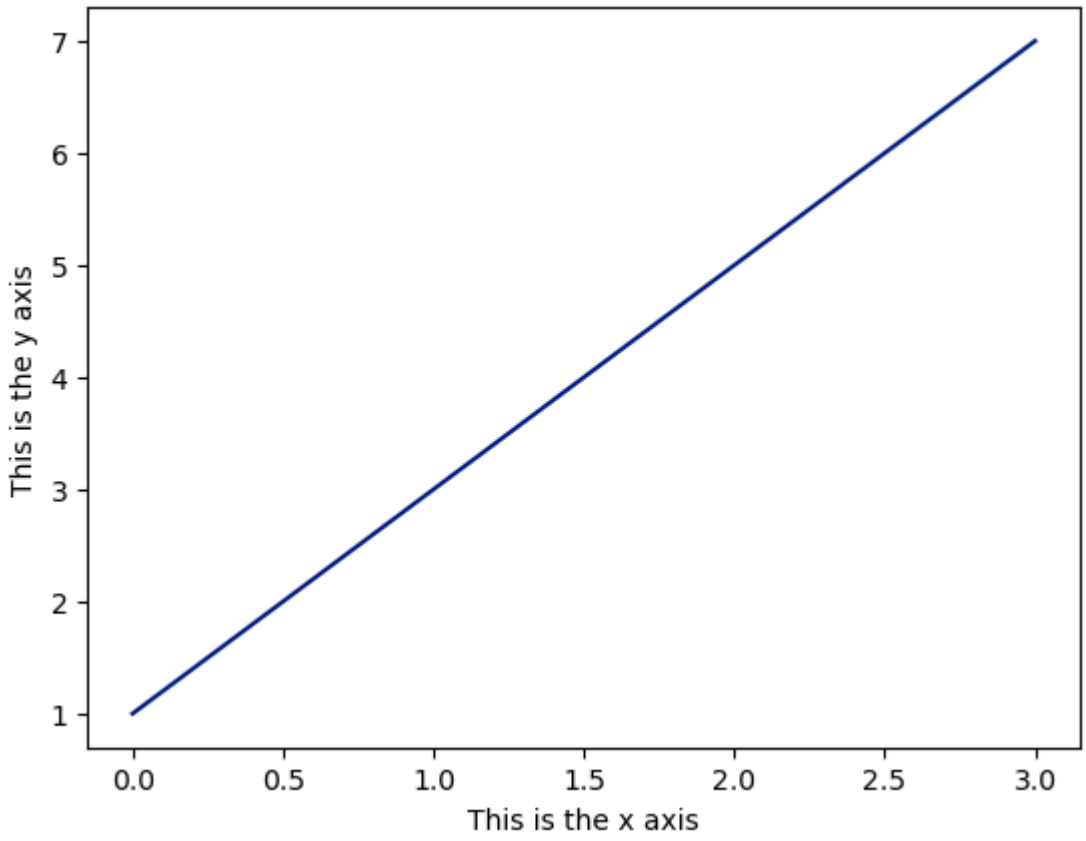
In [58]:
```
x17 =np.arange(1,5)

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

plt.xlim([1.0,5.0])
plt.ylim([0.0,18.0])
plt.show()
```



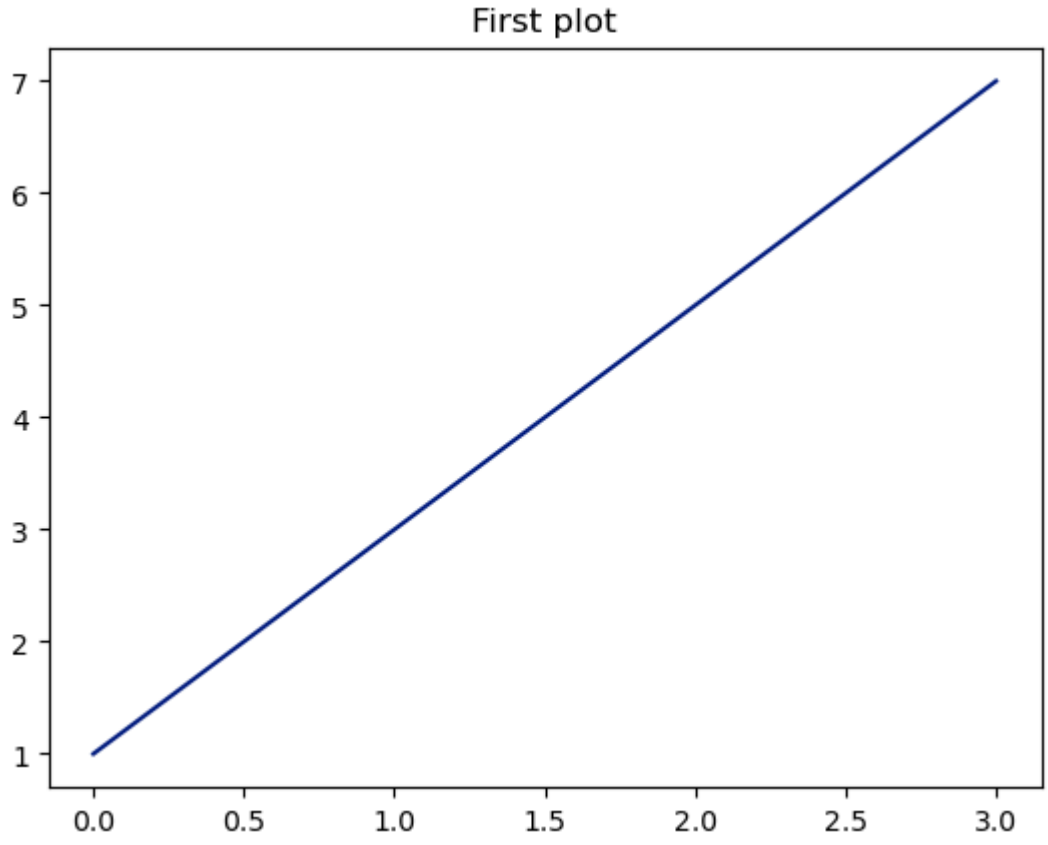
In [60]:
```
u = [1,3,5,7,9,2,4,6,8]
plt.plot(u)

plt.xticks([2,7,5,9,3])
plt.yticks([2,7,5,9,3])
plt.show()
```



In [62]:
```
plt.plot([1,3,5,7])
plt.xlabel('This is the x axis')
plt.ylabel('This is the y axis')
plt.show()
```

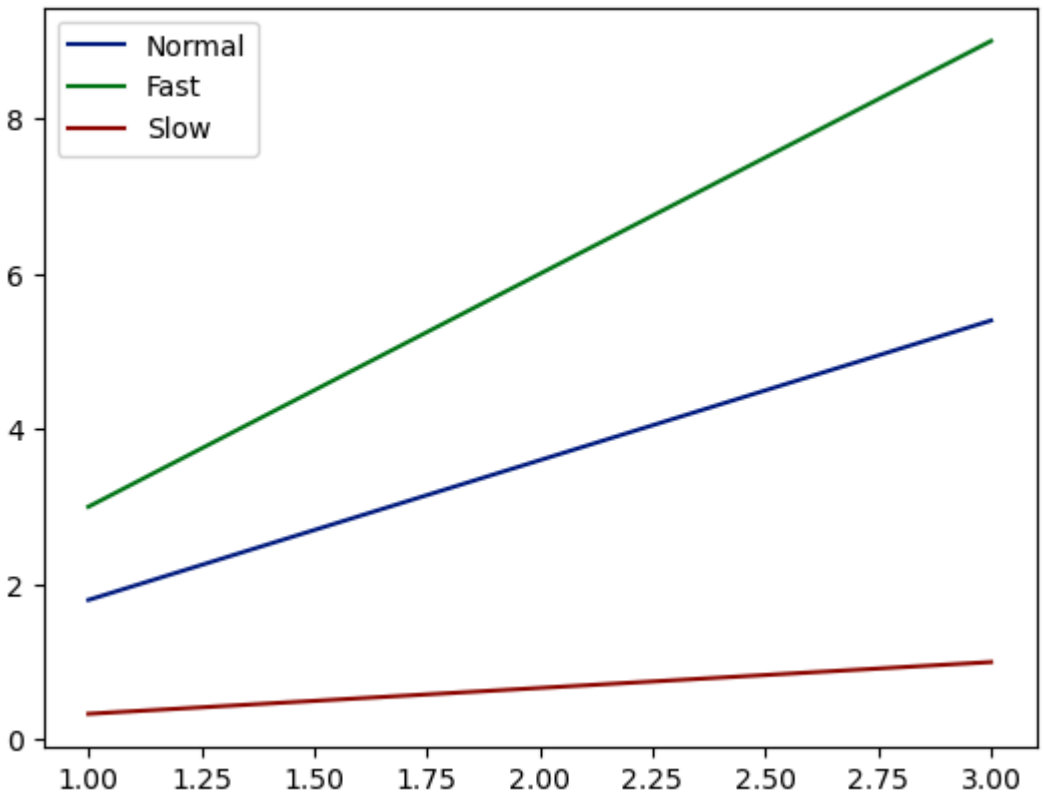


In [64]:
```
plt.plot([1,3,5,7])
plt.title('First plot')
plt.show()
```

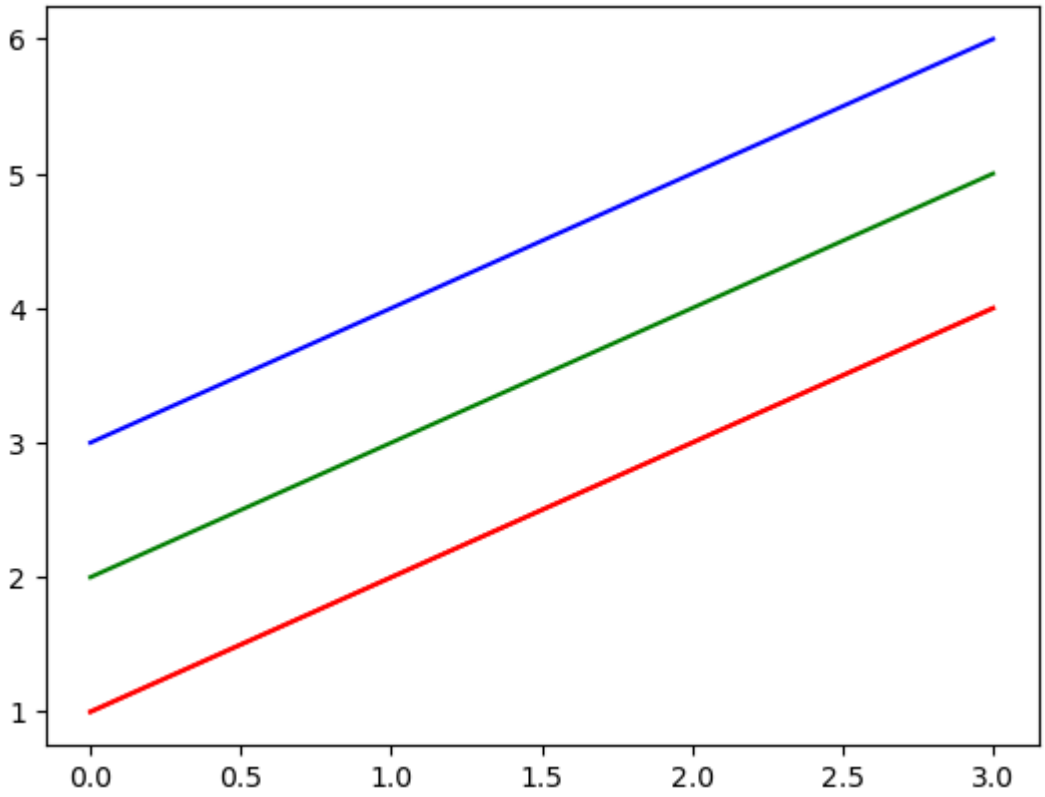


In [87]:
```
x18 = np.arange(1,4)
fig, ax = plt.subplots()
ax.plot(x18,x18*1.8)
```

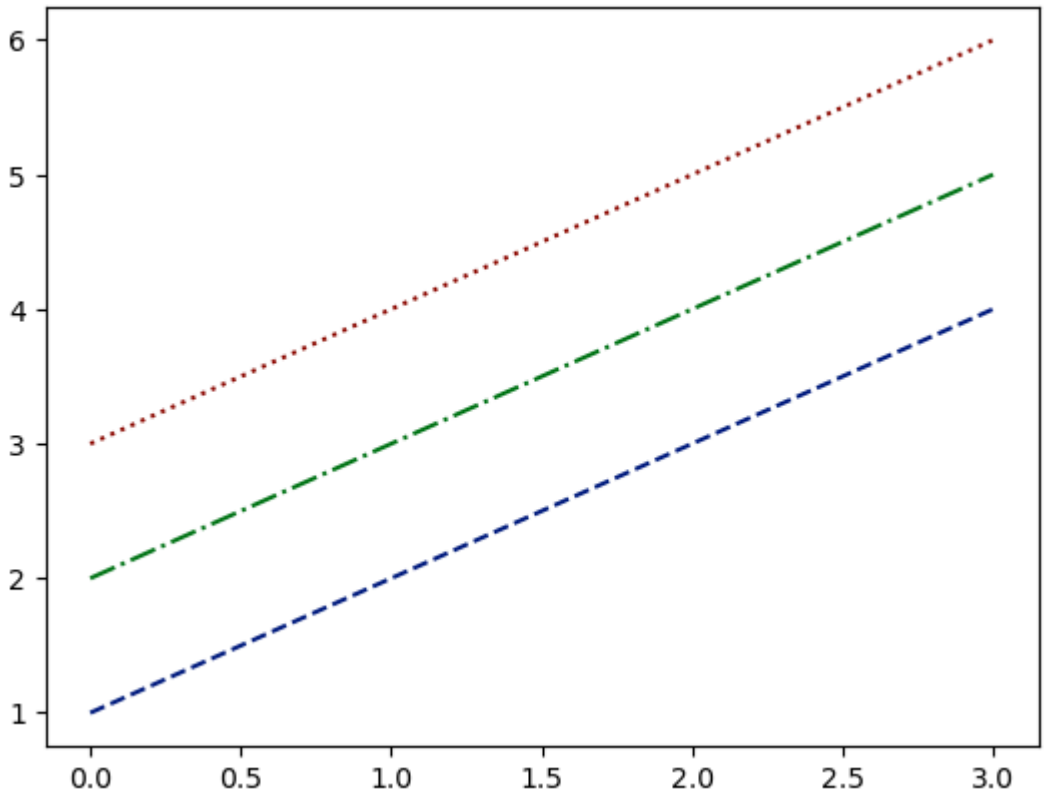
```
ax.plot(x18,x18*3.0)
ax.plot(x18,x18/3.0)

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



In [91]:
```
x19 =np.arange(1,5)

plt.plot(x19,'r')
plt.plot(x19+1,'g')
plt.plot(x19+2,'b')
plt.show()
```



In [93]:
```
x19 =np.arange(1,5)

plt.plot(x19, '--',x19+1, '-.',x19+2,':')

plt.show()
```



In [ ]: