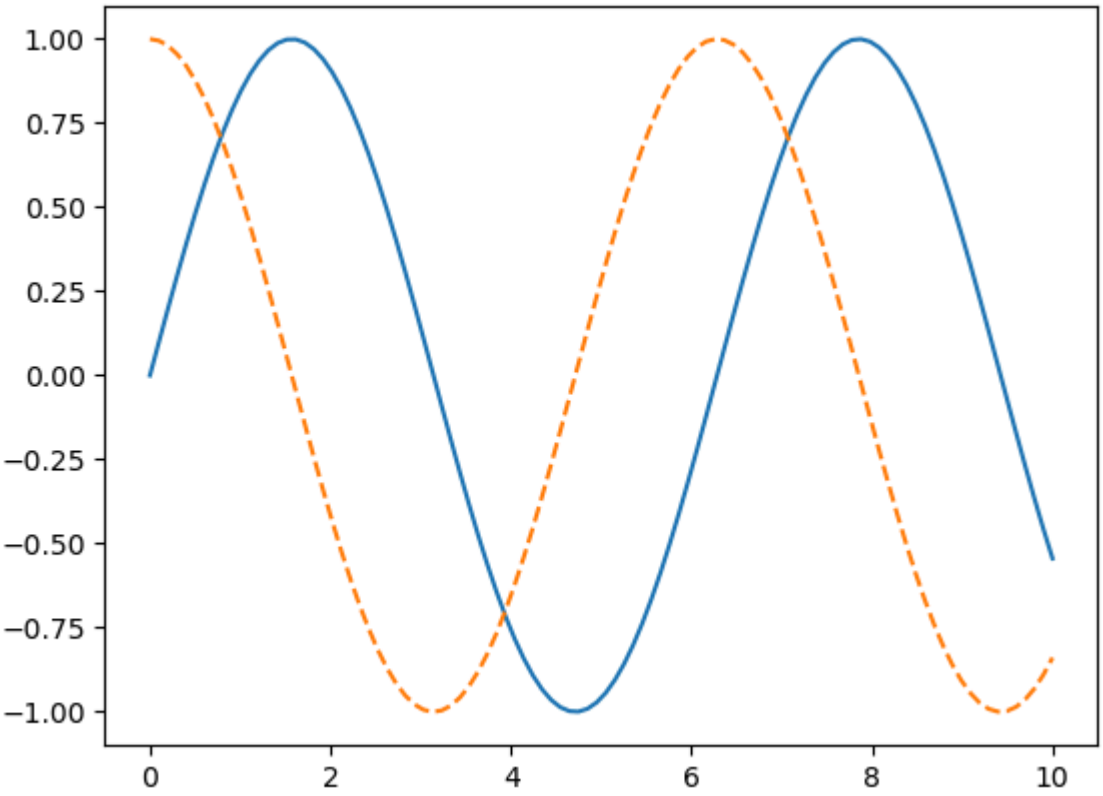
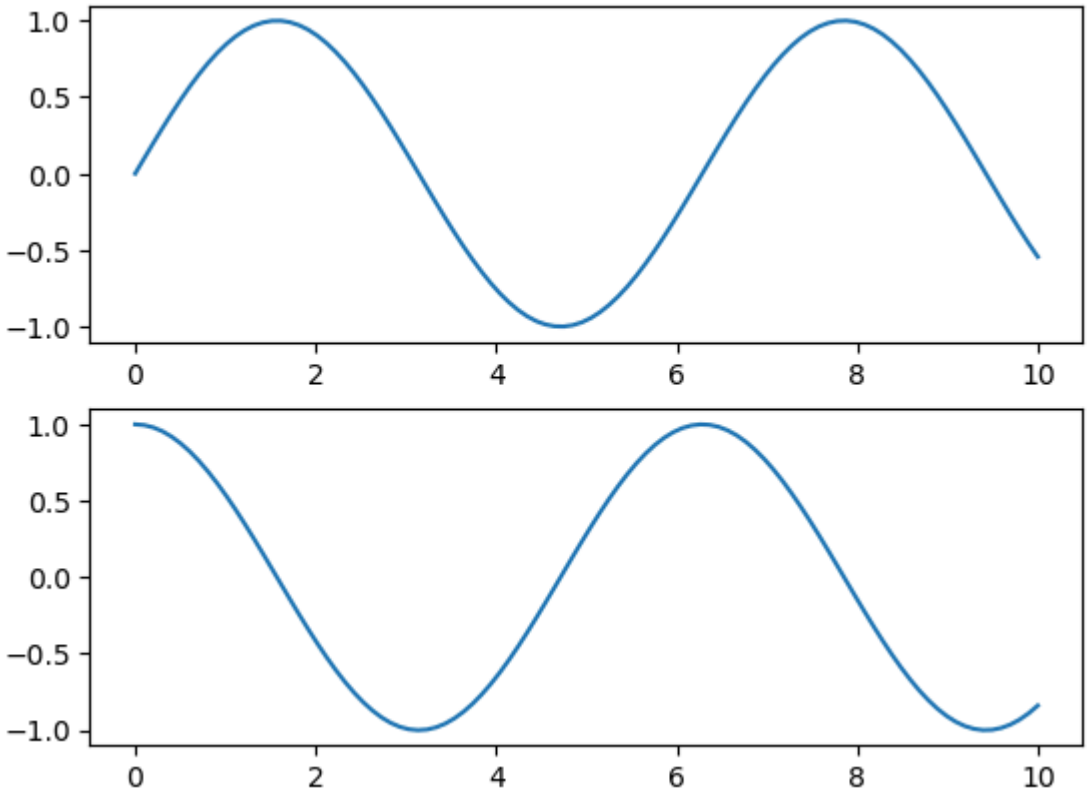
In [1]:
```
import numpy as np
import pandas as pd
```

In [3]:
```
import matplotlib.pyplot as plt
```

In [9]:
```
%matplotlib inline


X1 = np.linspace(0, 10, 100)


# create a plot figure
fig = plt.figure()

plt.plot(x1, np.sin(X1), '-')
plt.plot(x1, np.cos(X1), '--');
```



In [11]:
```
plt.figure()
plt.subplot(2,1,1) #create first of two panels and set current axis
plt.plot(X1,np.sin(X1))
plt.subplot(2,1,2) #create seconf of two panels and set current axis
plt.plot(X1,np.cos(X1));
```

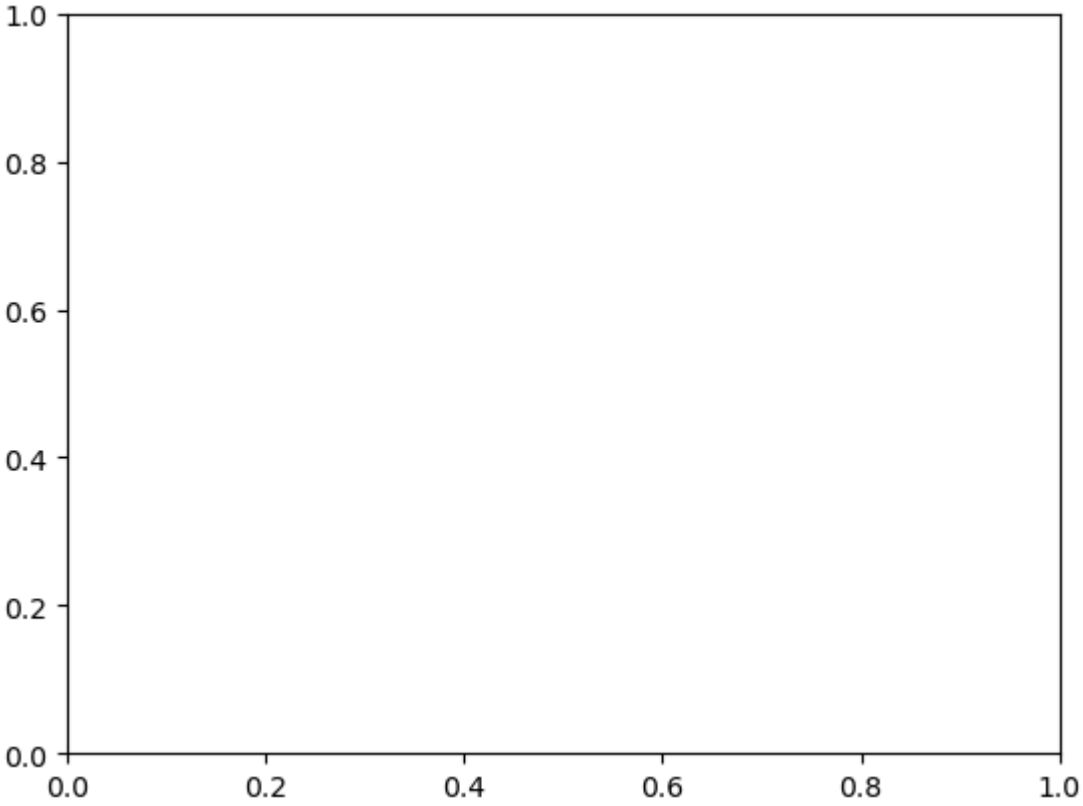


In [13]:
```
print(plt.gcf())
```

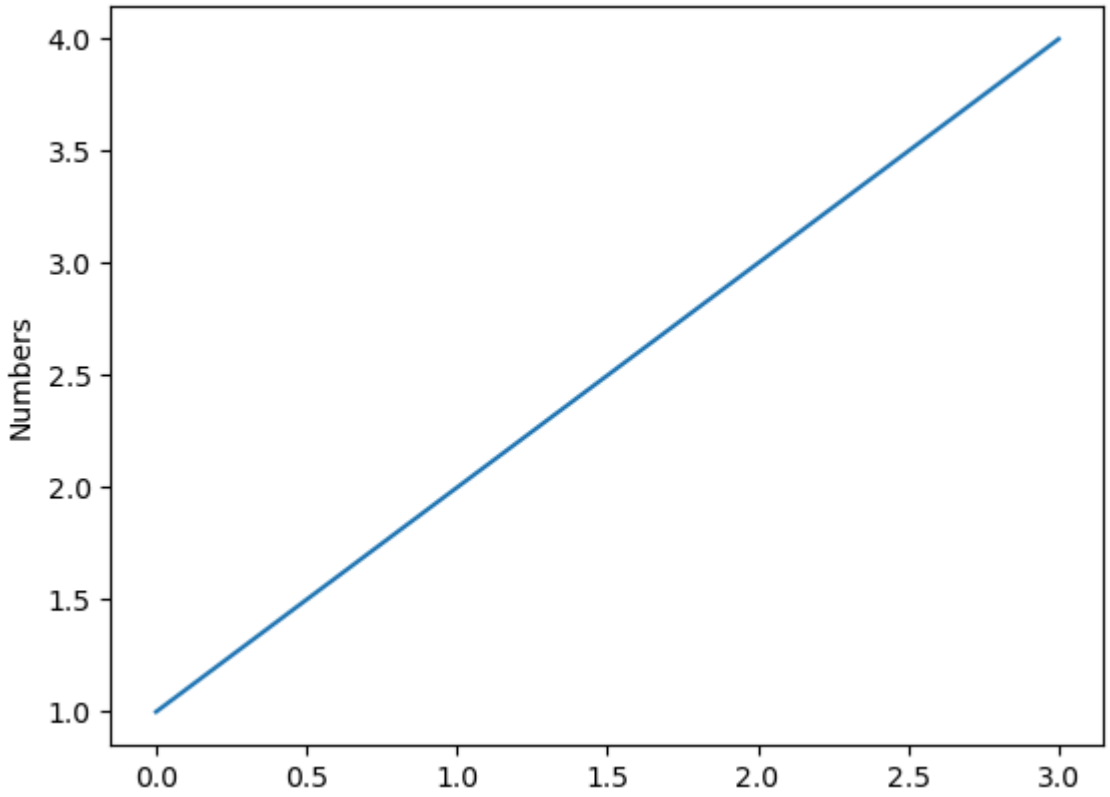
```
Figure(640x480)
<Figure size 640x480 with 0 Axes>
```

In [15]:
```
print(plt.gca())
```

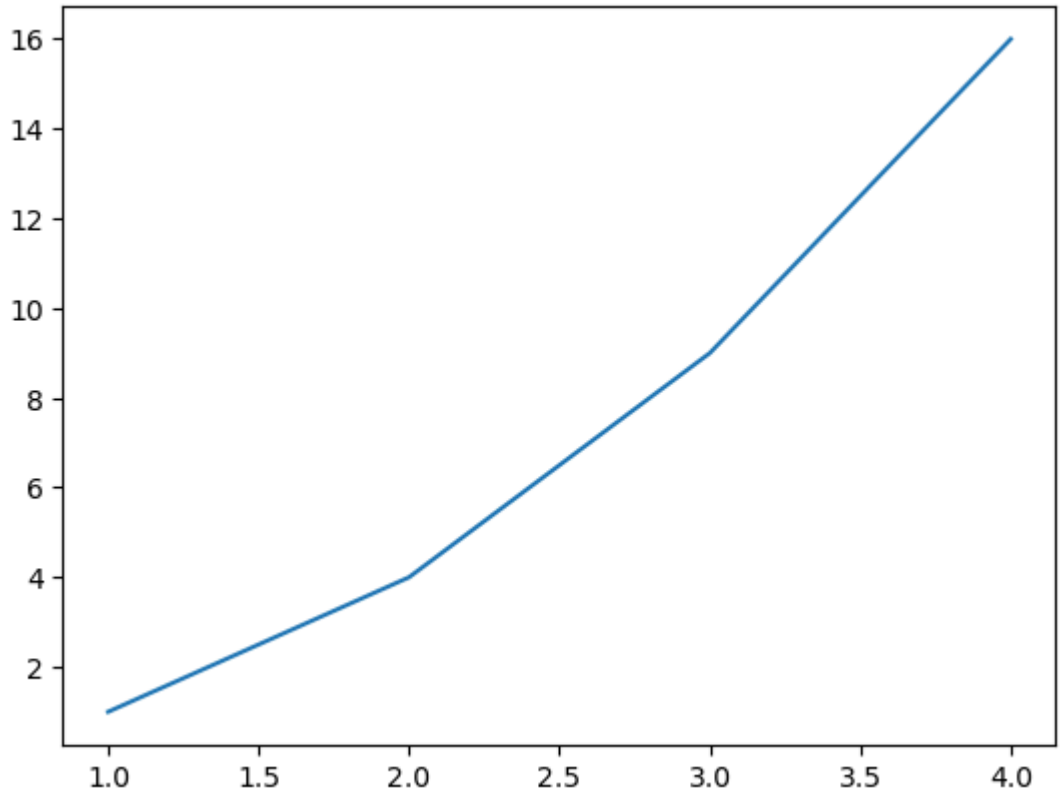
```
Axes(0.125,0.11;0.775x0.77)
```



In [17]:
```
plt.plot([1,2,3,4])
plt.ylabel('Numbers')
plt.show()
```

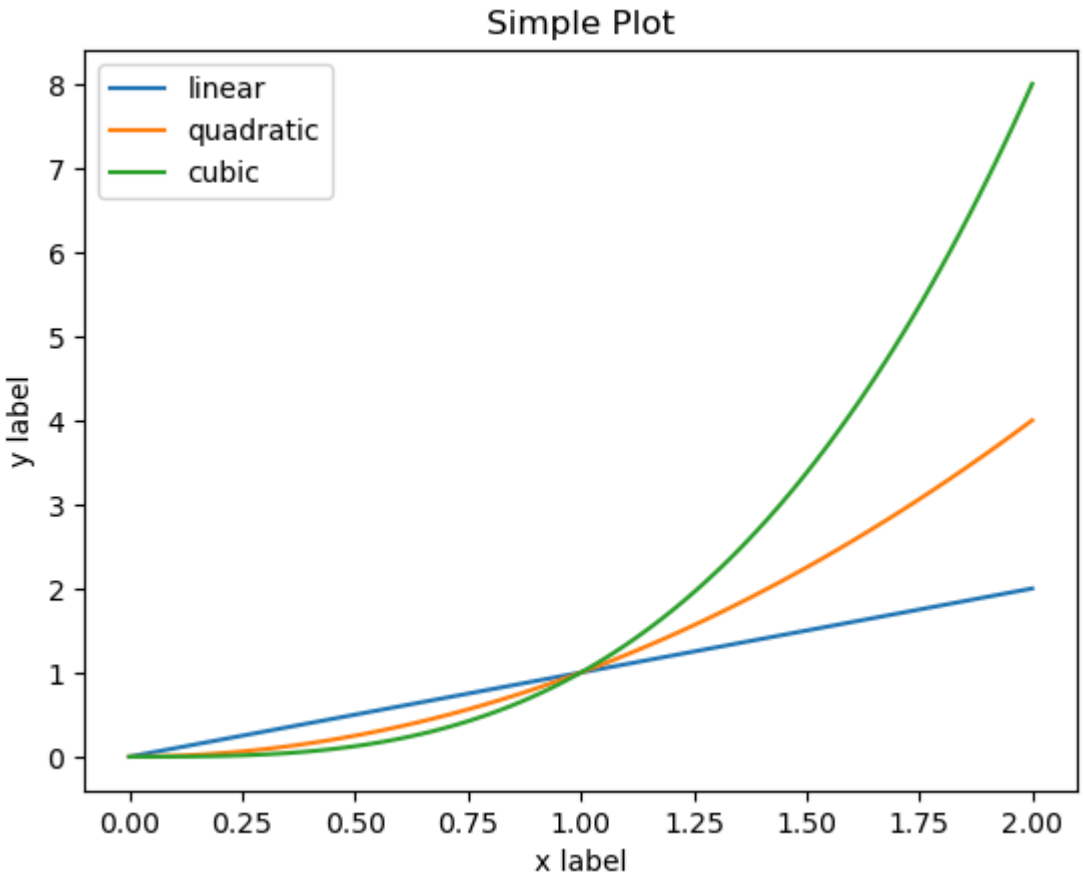


In [19]:
```
plt.plot([1,2,3,4],[1,4,9,16])
plt.show()
```

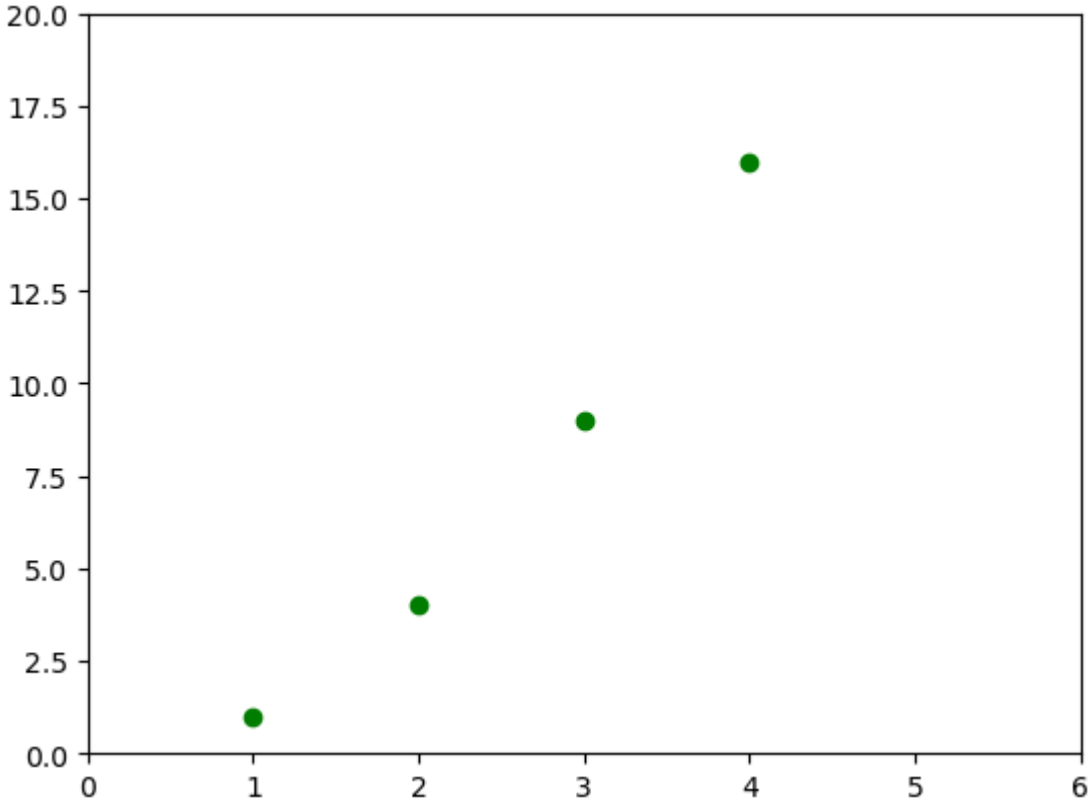


In [21]:
```
x = np.linspace(0, 2, 100)
```

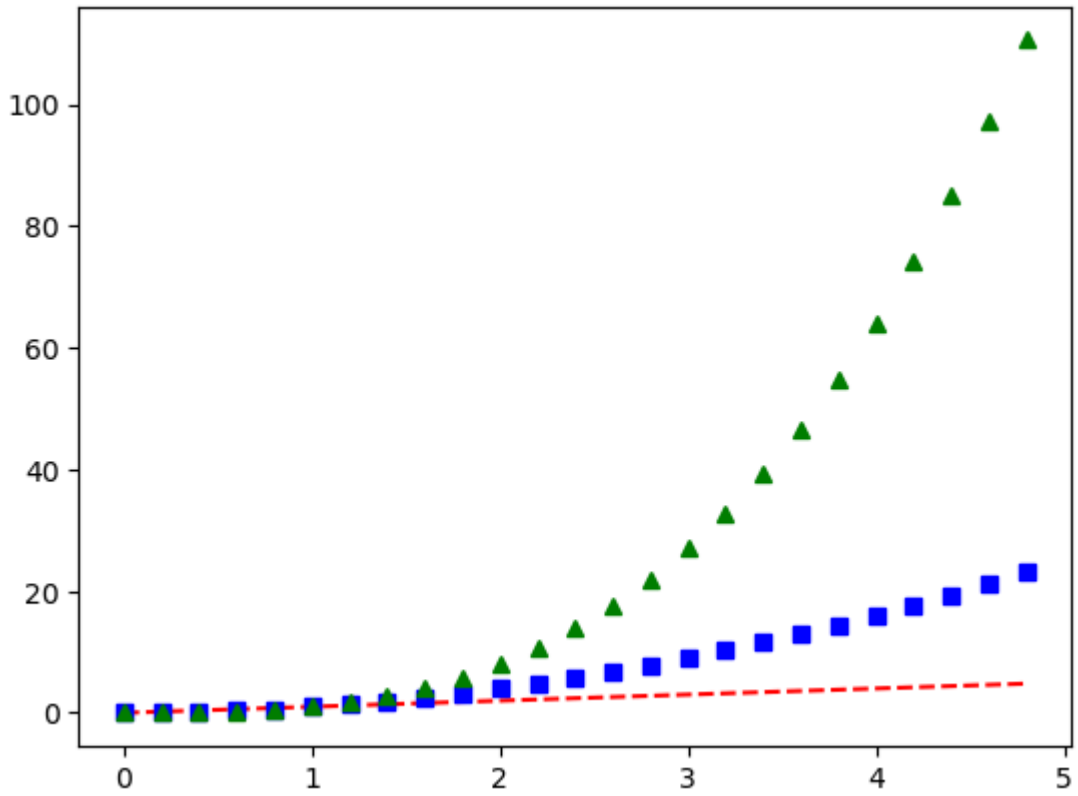
```
plt.plot(x, x, label='linear')
plt.plot(x, x**2, label='quadratic')
plt.plot(x, x**3, label='cubic')

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

plt.title("Simple Plot")

plt.legend()

plt.show()
```



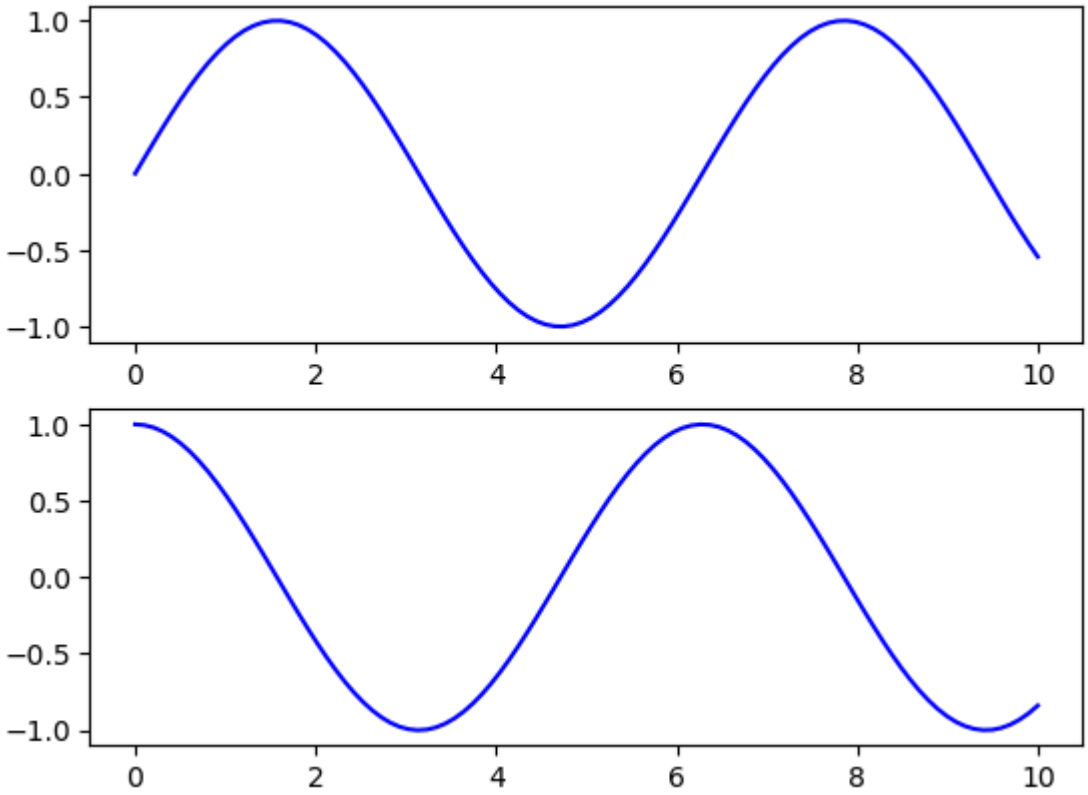
In [23]:
```
plt.plot([1,2,3,4],[1,4,9,16],'go')
plt.axis([0,6,0,20])
plt.show()
```



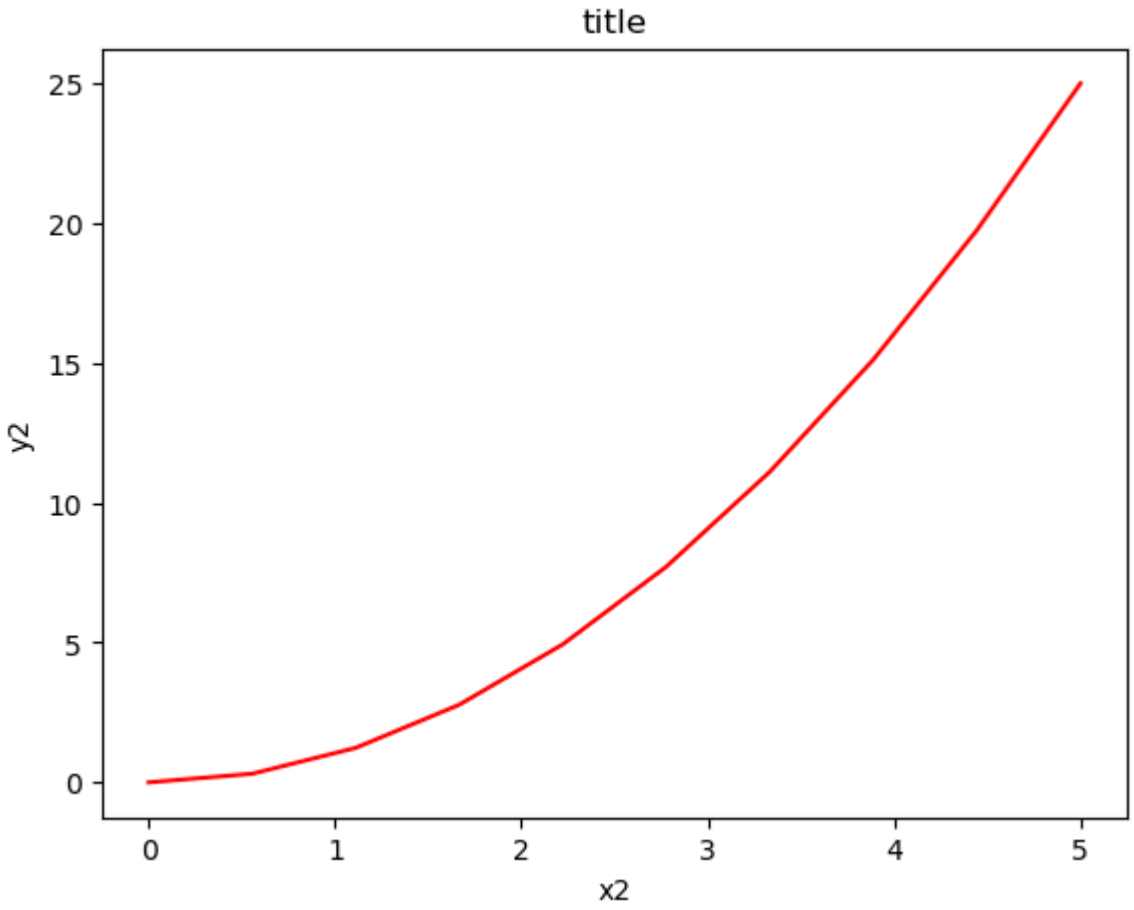
In [25]:
```
t = np.arange(0.,5.,0.2)
plt.plot(t,t,'r--',t,t**2,'bs',t,t**3,'g^')
plt.show()
```



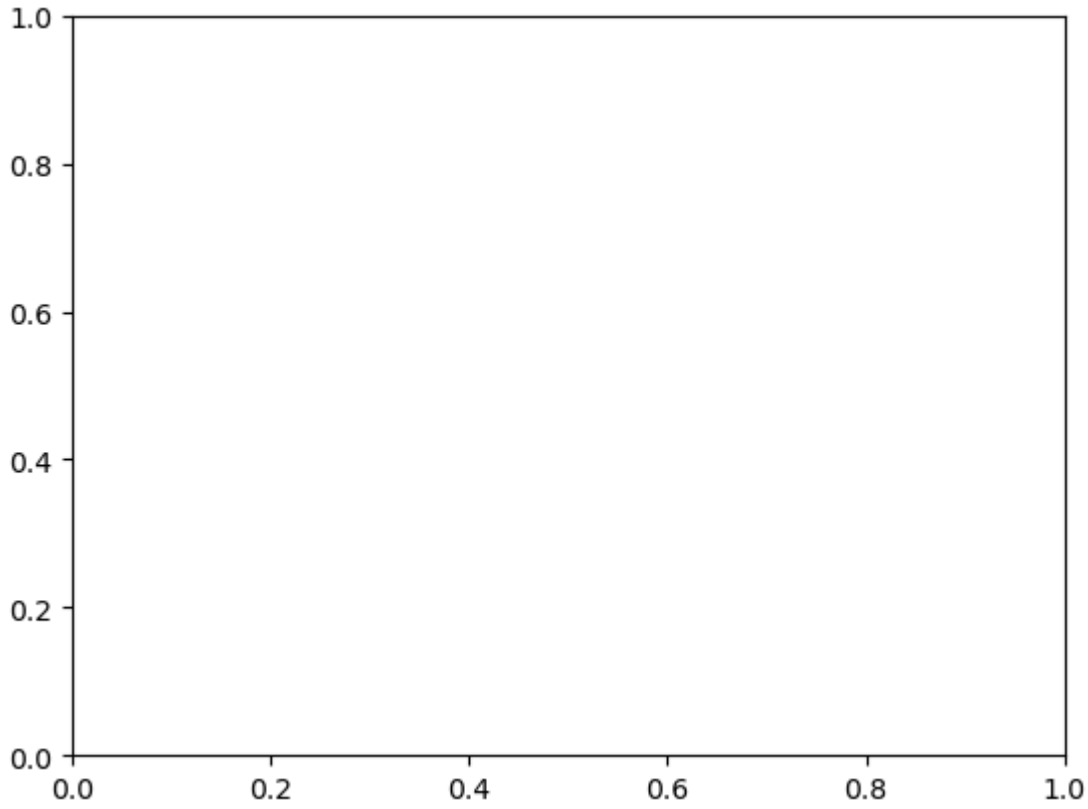
In [27]:
```
fig, ax = plt.subplots(2)
ax[0].plot(x1, np.sin(x1), 'b-')
ax[1].plot(x1, np.cos(x1), 'b-');
```



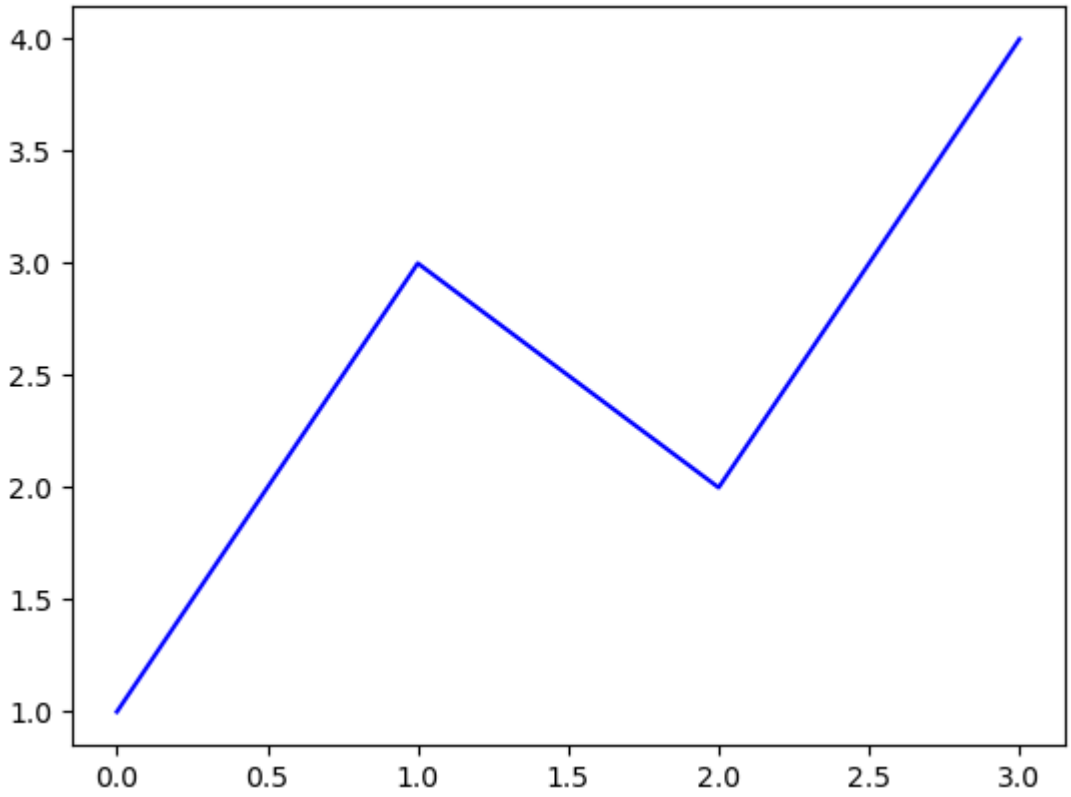
In [29]:
```
fig = plt.figure()

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');
```



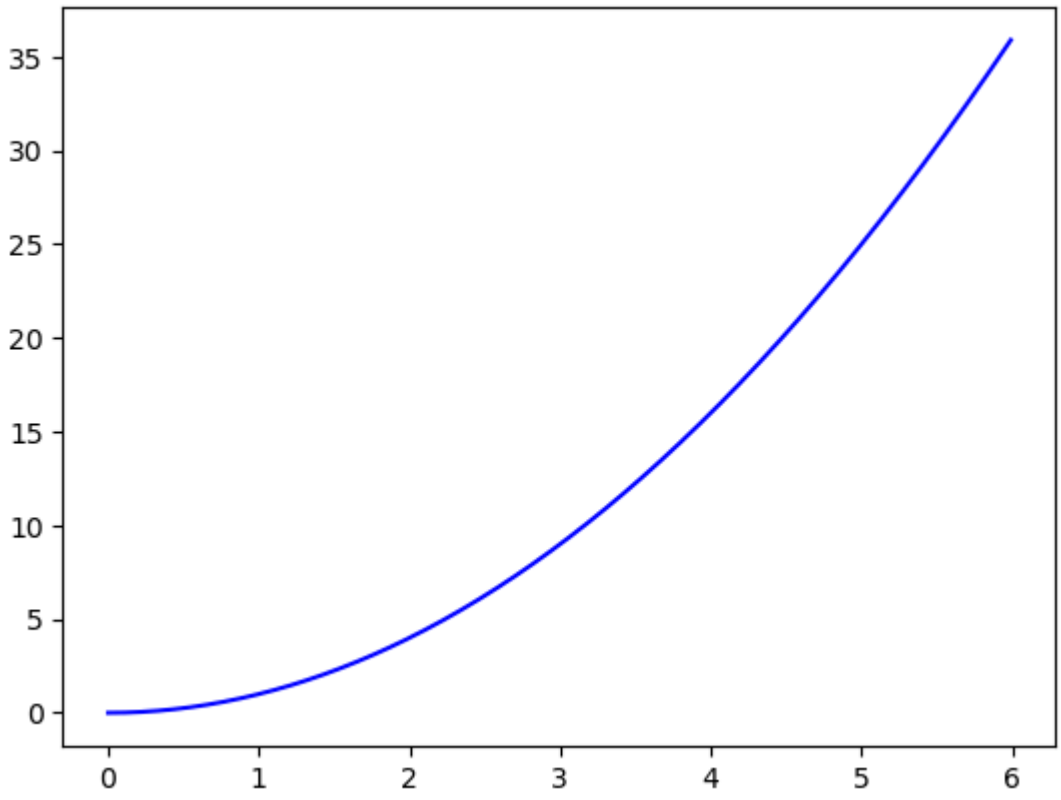
In [31]:
```
fig = plt.figure()
ax = plt.axes()
```



In [37]:
```
plt.plot([1, 3, 2, 4], 'b-') #first plot with matplotlib

plt.show( )
```

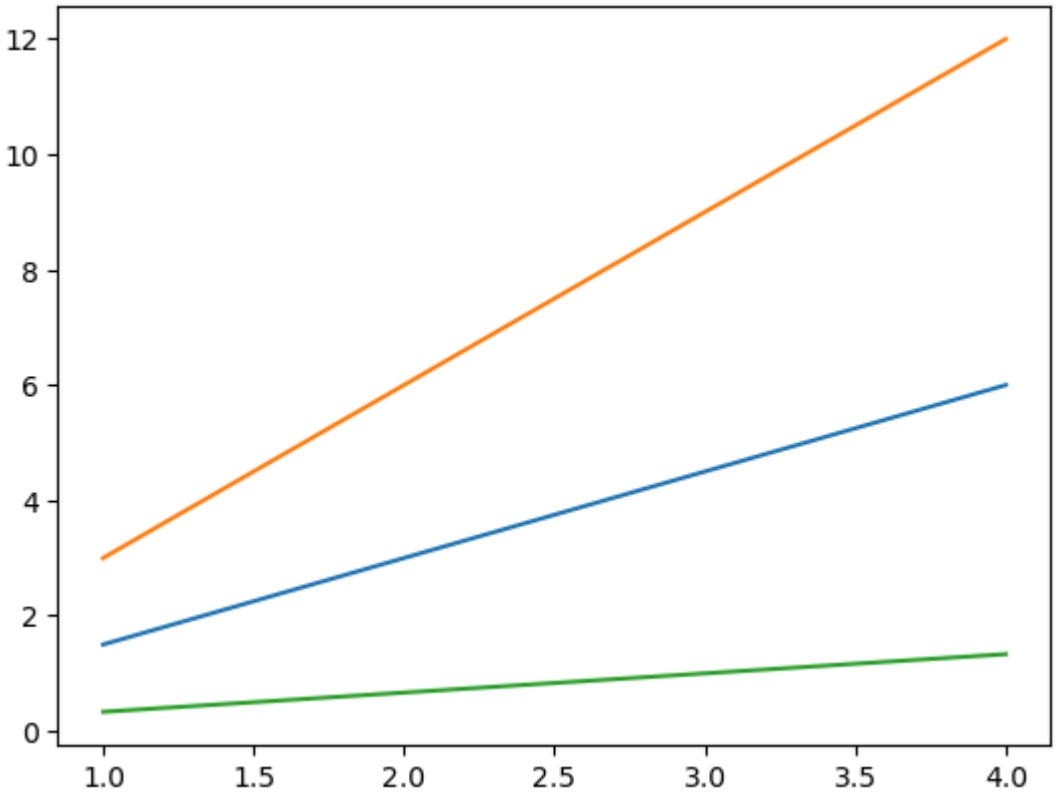


In [35]:
```
x3 = np.arange(0.0, 6.0, 0.01)

plt.plot(x3, [xi**2 for xi in x3], 'b-')

plt.show()
```



In [39]:
```
x4 = range(1, 5) #multiline plots

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

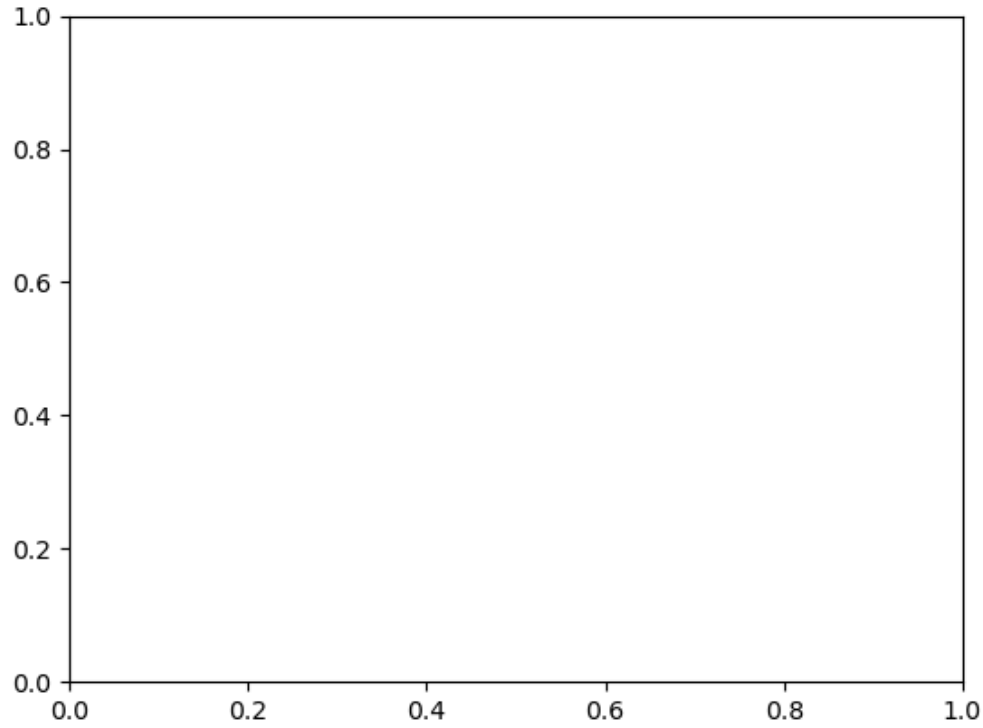
```
plt.plot(x4, [xi*3 for xi in x4])

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

plt.show()
```



In [41]:
```
fig.savefig('plot1.png')
```

In [43]:
```
from IPython.display import Image

Image('plot1.png')
```

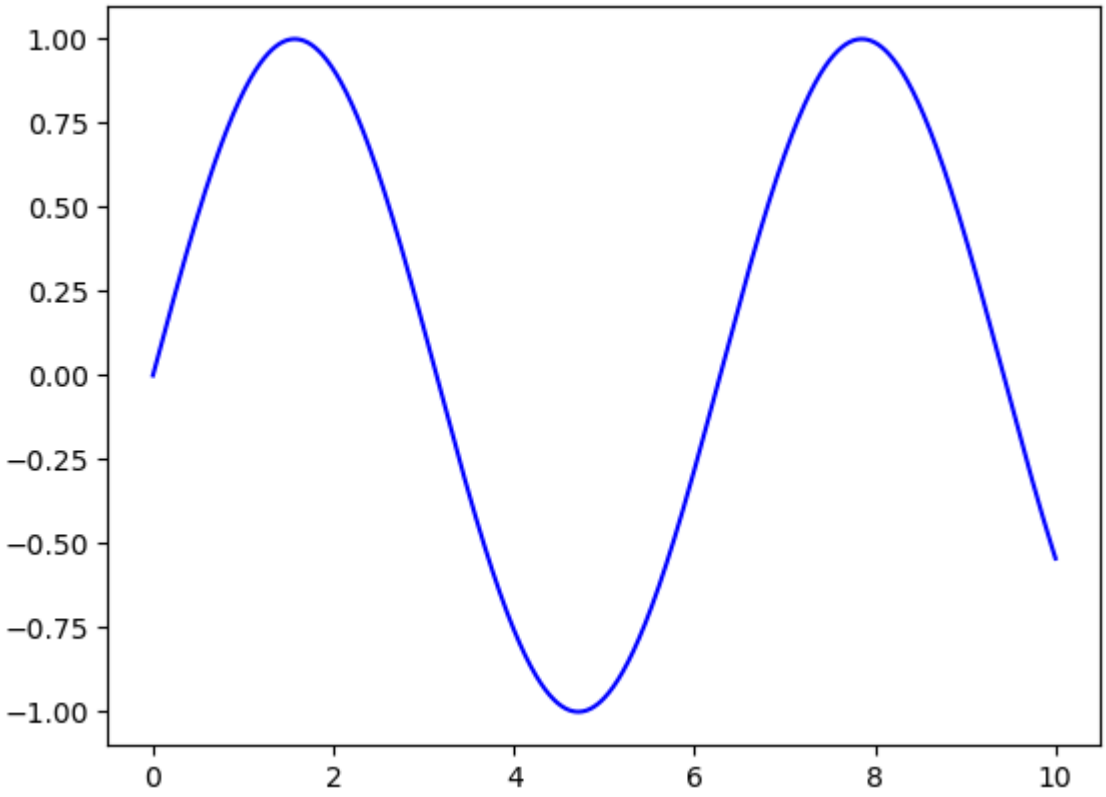
Out[43]:



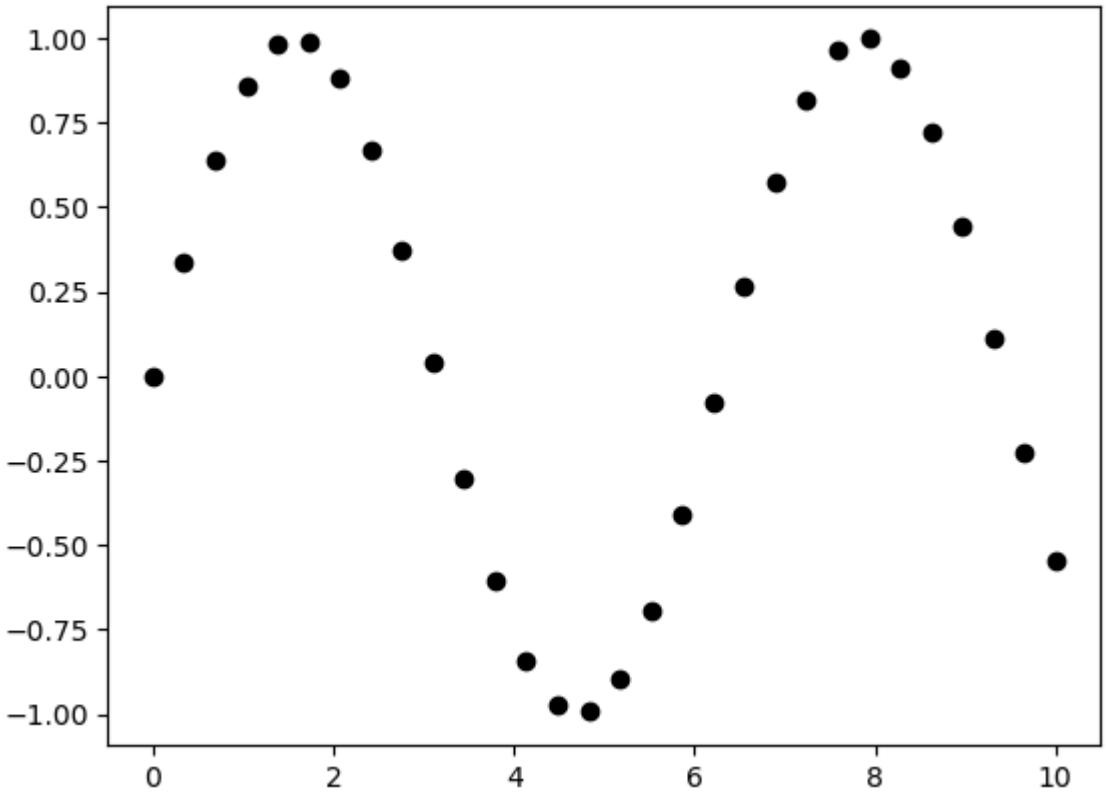
In [45]:
```
fig.canvas.get_supported_filetypes()
```

Out[45]:
```
{'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'}
```

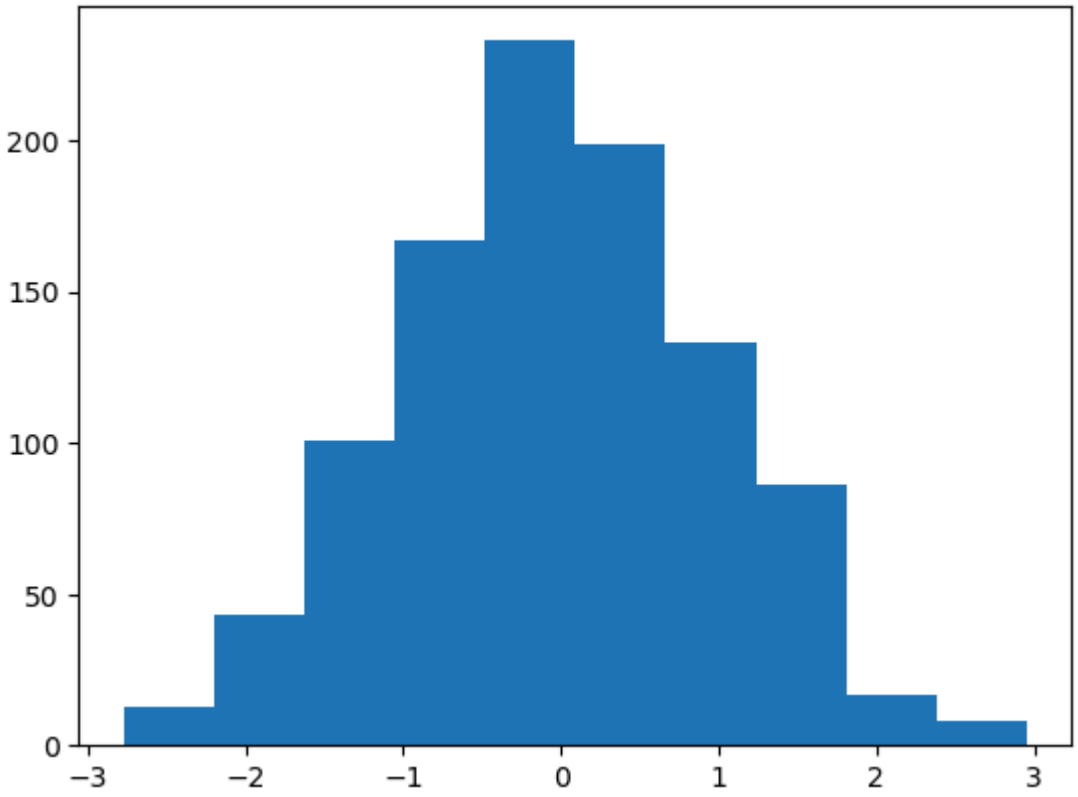
In [47]:
```
fig = plt.figure() #Line plot
ax = plt.axes()
x5 = np.linspace(0, 10, 1000)
ax.plot(x5, np.sin(x5), 'b-');
```



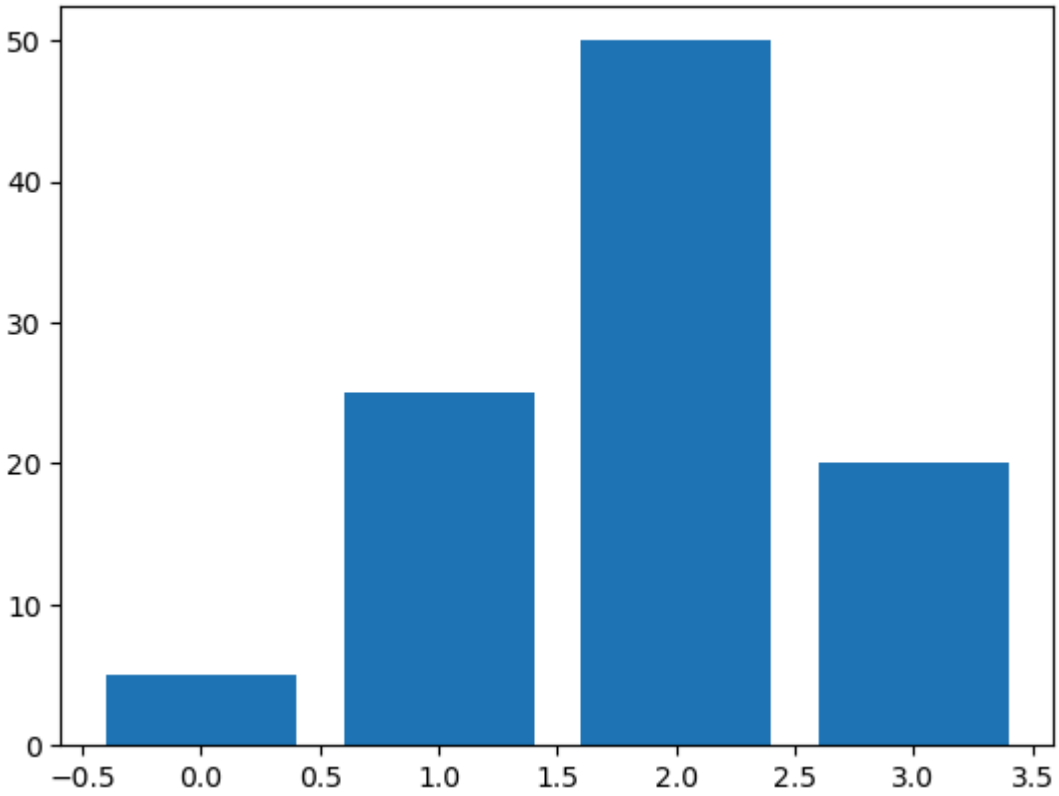
In [51]:
```
x7 = np.linspace(0, 10, 30) #scatter plot
y7 = np.sin(x7)
plt.plot(x7, y7, 'o', color = 'black');
```



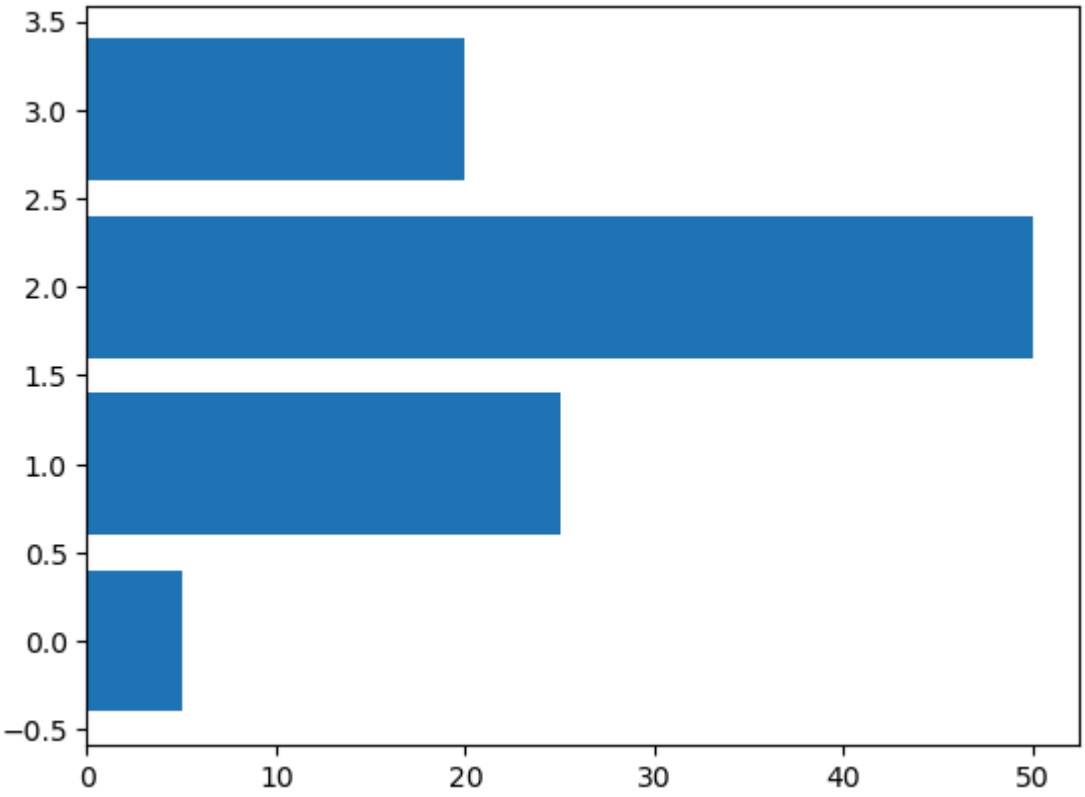
In [53]:
```
data1 = np.random.randn(1000) #histogram
plt.hist(data1);
```



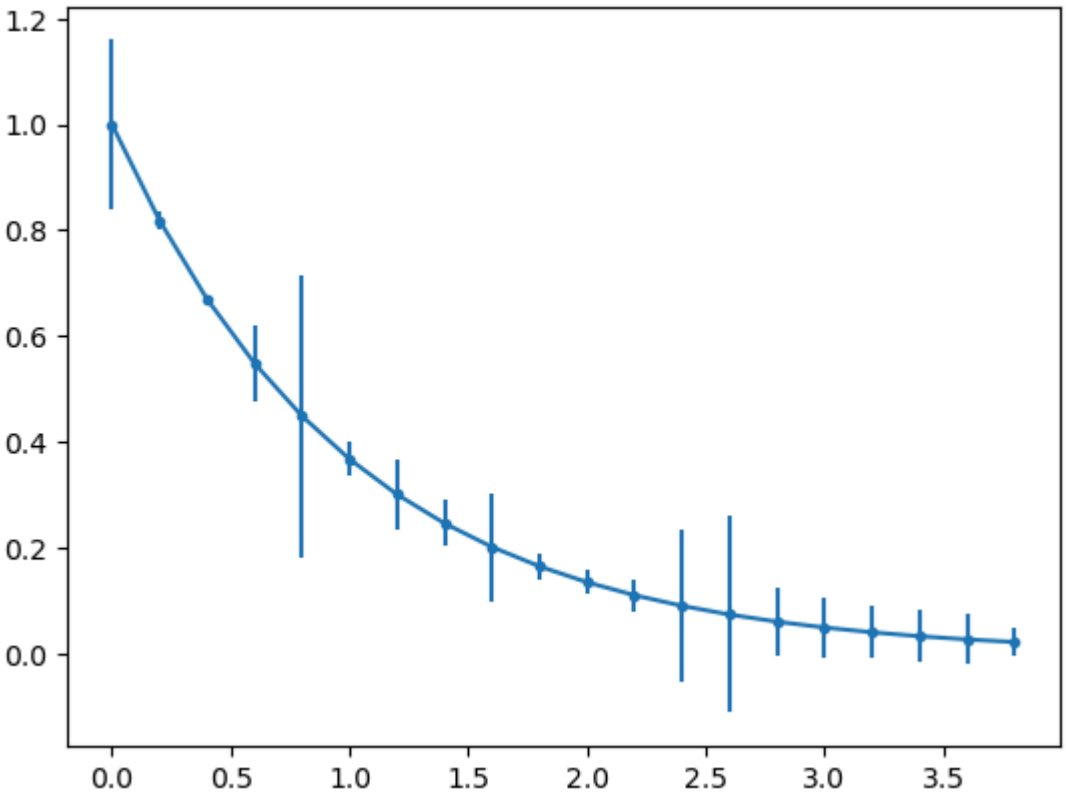
In [55]:
```
data2 = [5. , 25. , 50. , 20.] #bar chart
plt.bar(range(len(data2)), data2)
plt.show()
```



In [57]:
```
data2 = [5. , 25. , 50. , 20.] #horizontal bar chart
plt.barh(range(len(data2)), data2)
plt.show()
```

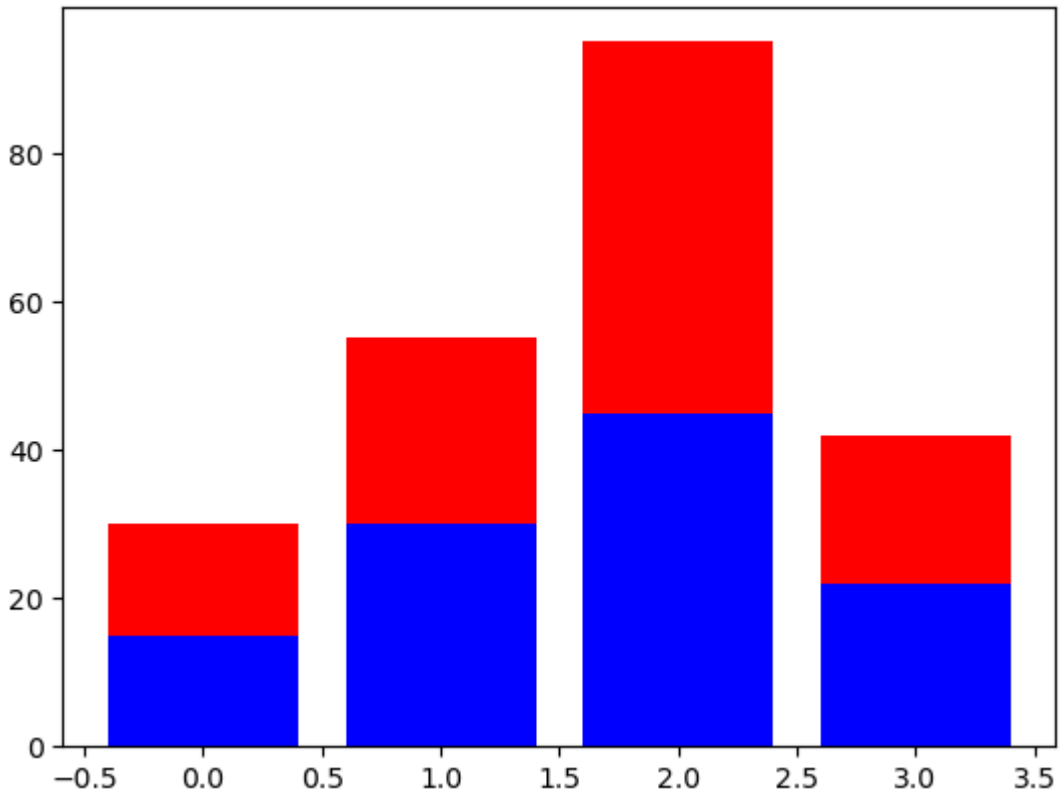


In [59]:
```
x9 = np.arange(0, 4, 0.2) #error bar chart
y9 = np.exp(-x9)
e1 = 0.1 * np.abs(np.random.randn(len(y9)))
plt.errorbar(x9, y9, yerr = e1, fmt = '.-')
plt.show();
```

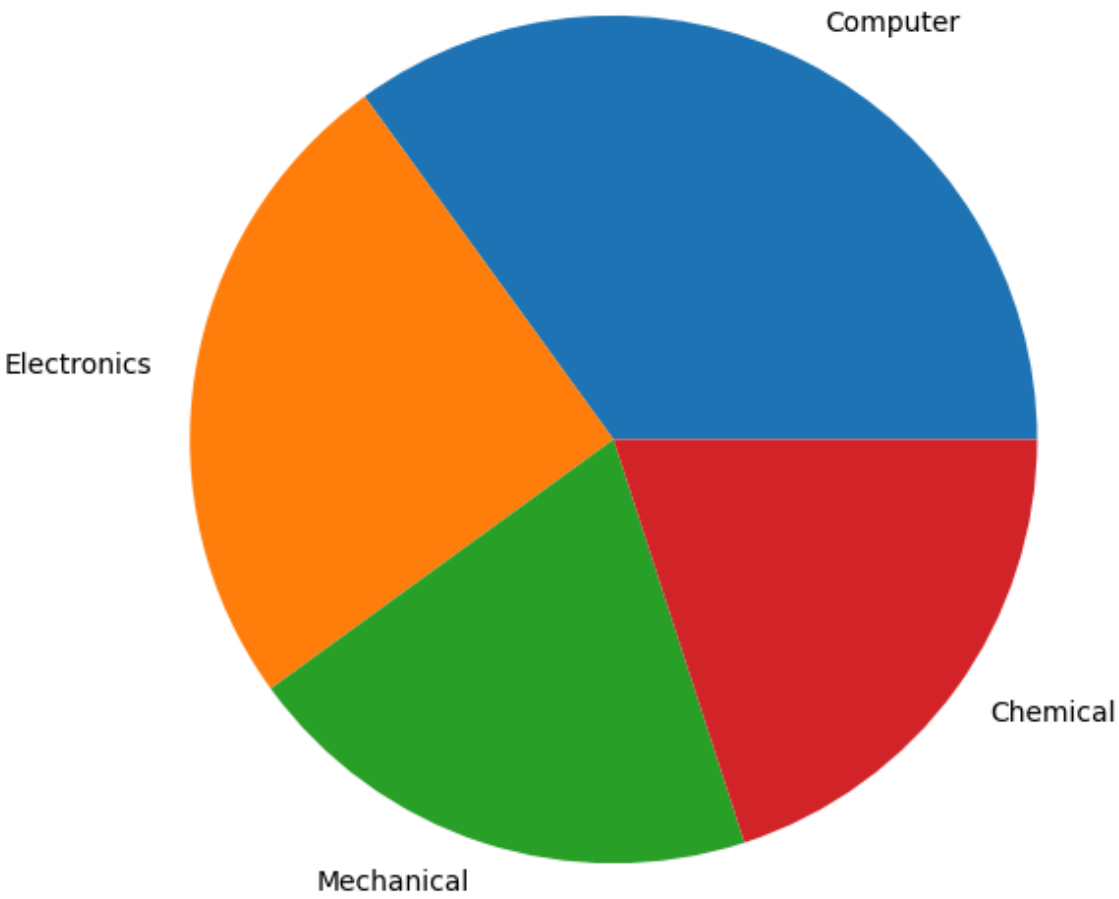


In [61]:
```
A = [15., 30., 45., 22.] #stacked bar chart
B = [15., 25., 50., 20.]
z2 = range(4)
```

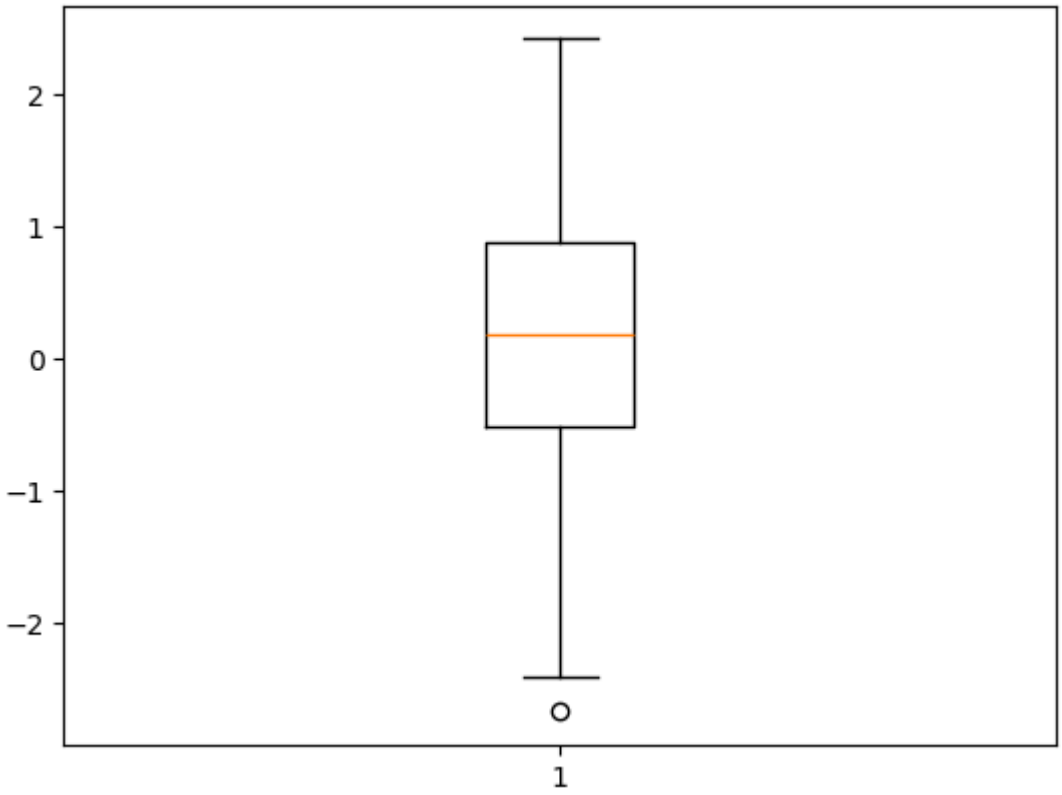
```
plt.bar(z2, A, color = 'b')
plt.bar(z2, B, color = 'r', bottom = A)
plt.show()
```



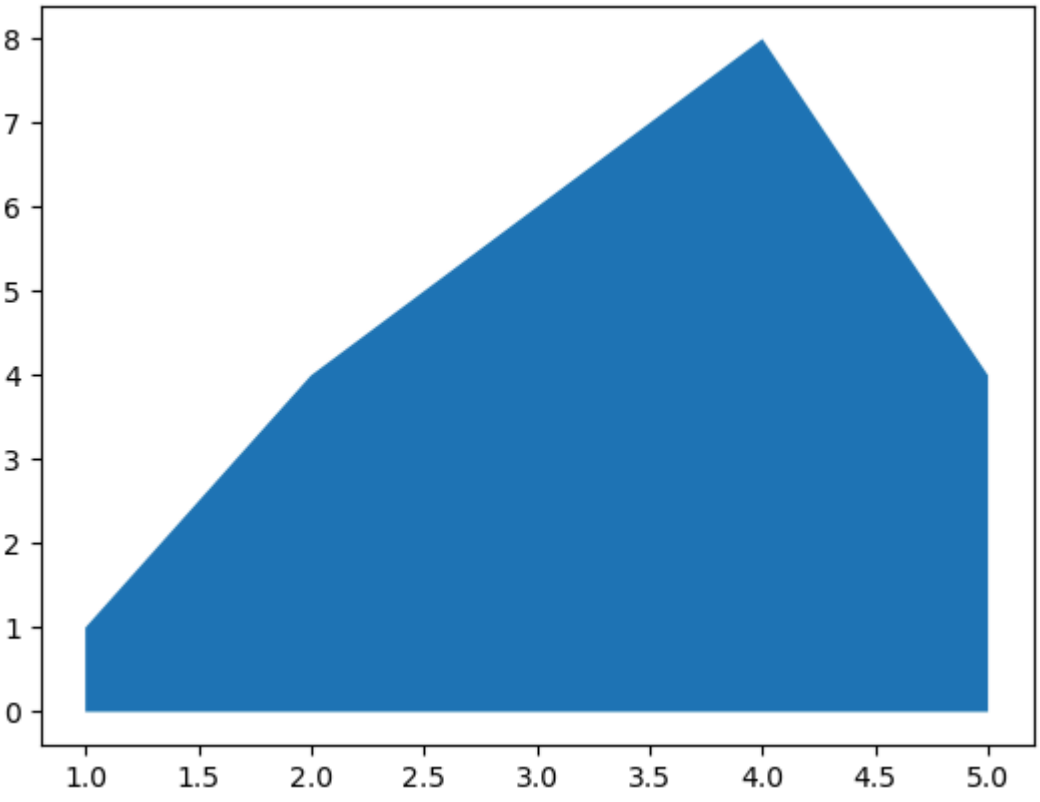
In [63]:
```
plt.figure(figsize=(7,7)) #piechart
x10 = [35, 25, 20, 20]
labels = ['Computer', 'Electronics', 'Mechanical', 'Chemical']
plt.pie(x10, labels=labels);
plt.show()
```



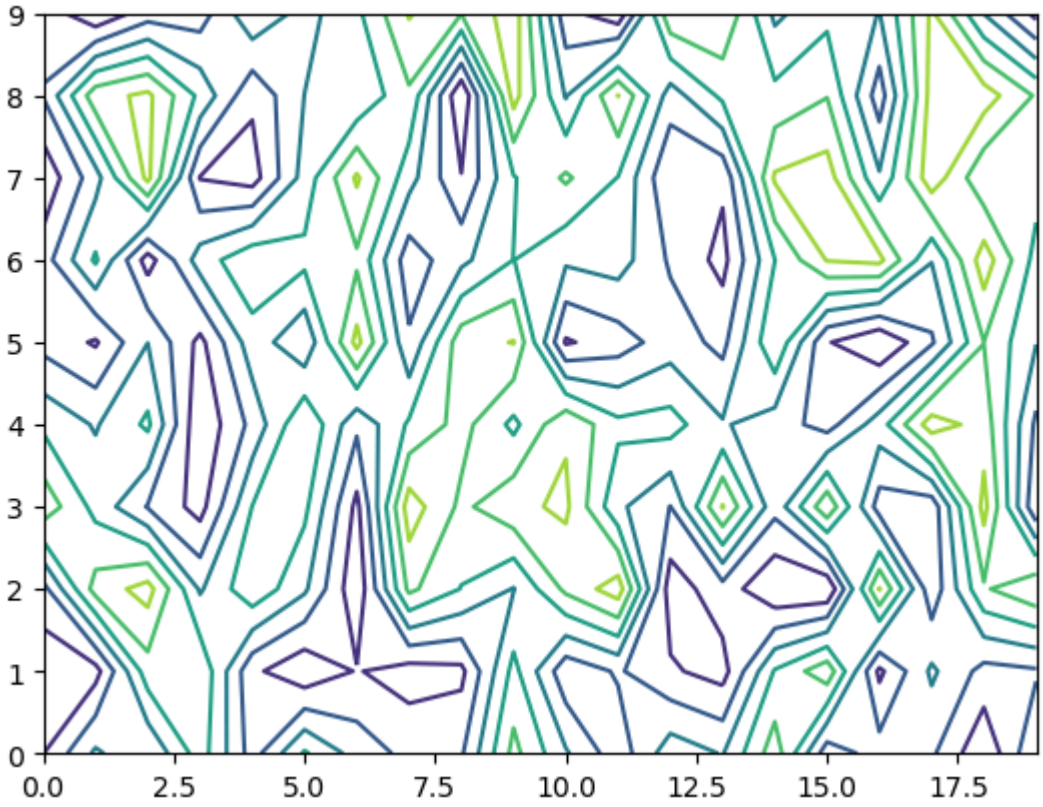
In [65]:
```
data3 = np.random.randn(100) #boxplot
plt.boxplot(data3)
plt.show();
```



In [67]:
```
x12 = range(1, 6) # area plot
y12 = [1, 4, 6, 8, 4]
plt.fill_between(x12, y12)
plt.show()
```



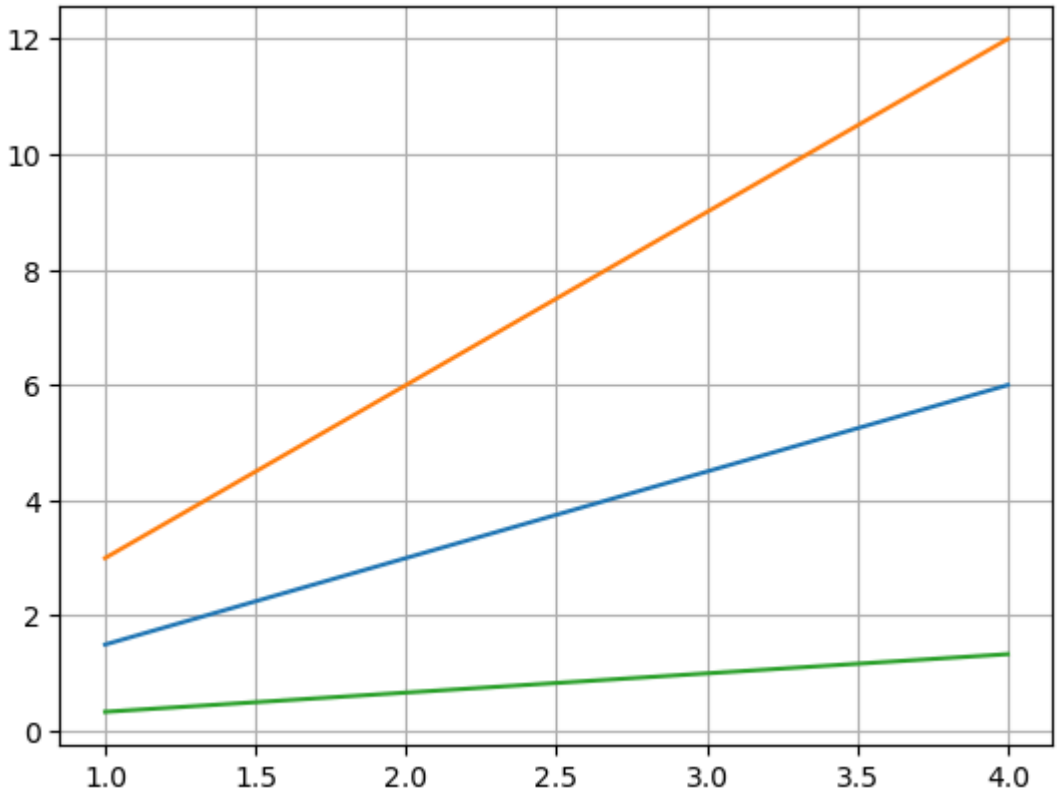
In [69]:
```
matrix1 = np.random.rand(10, 20) #create a matrix
cp = plt.contour(matrix1)
plt.show()
```



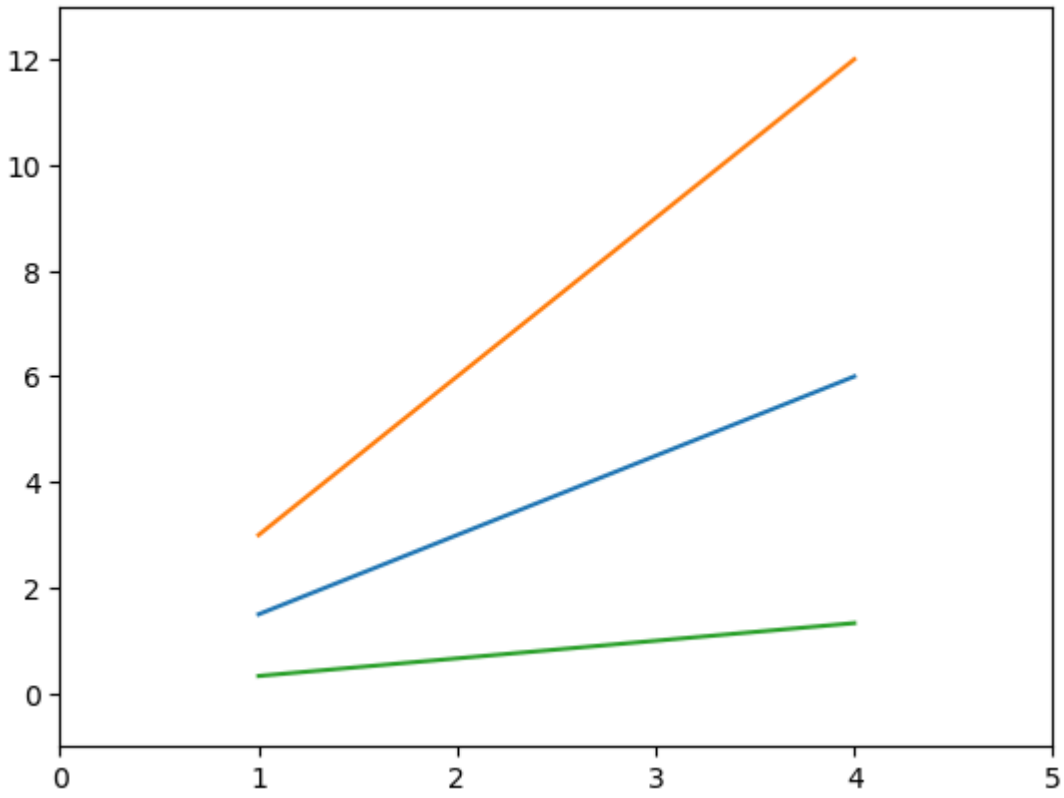
In [71]:
```
print(plt.style.available)
```

```
['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 [89]:
```
x15 = np.arange(1, 5) #adding grid
plt.plot(x15, x15*1.5, x15, x15*3.0, x15, x15/3.0)
plt.grid(True)
plt.show()
```



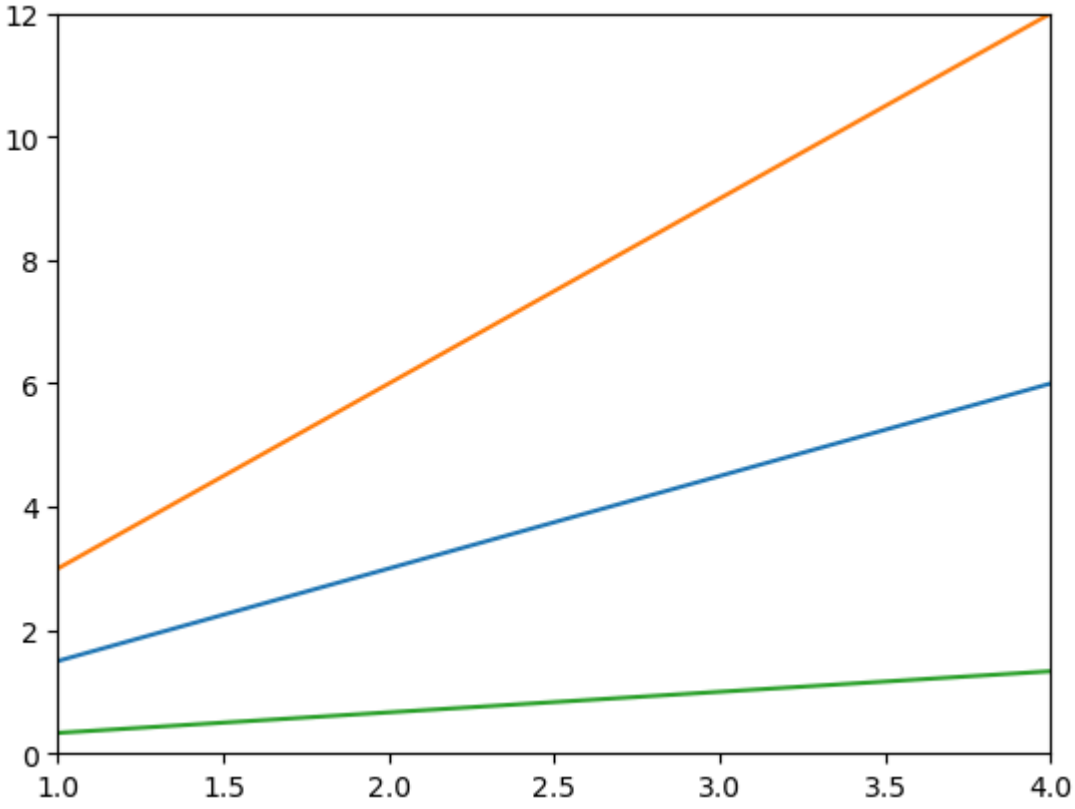
In [91]:
```
x15 = np.arange(1, 5) #handling axes
plt.plot(x15, x15*1.5, x15, x15*3.0, x15, x15/3.0)
plt.axis()   # shows the current axis limits values
plt.axis([0, 5, -1, 13])
plt.show()
```



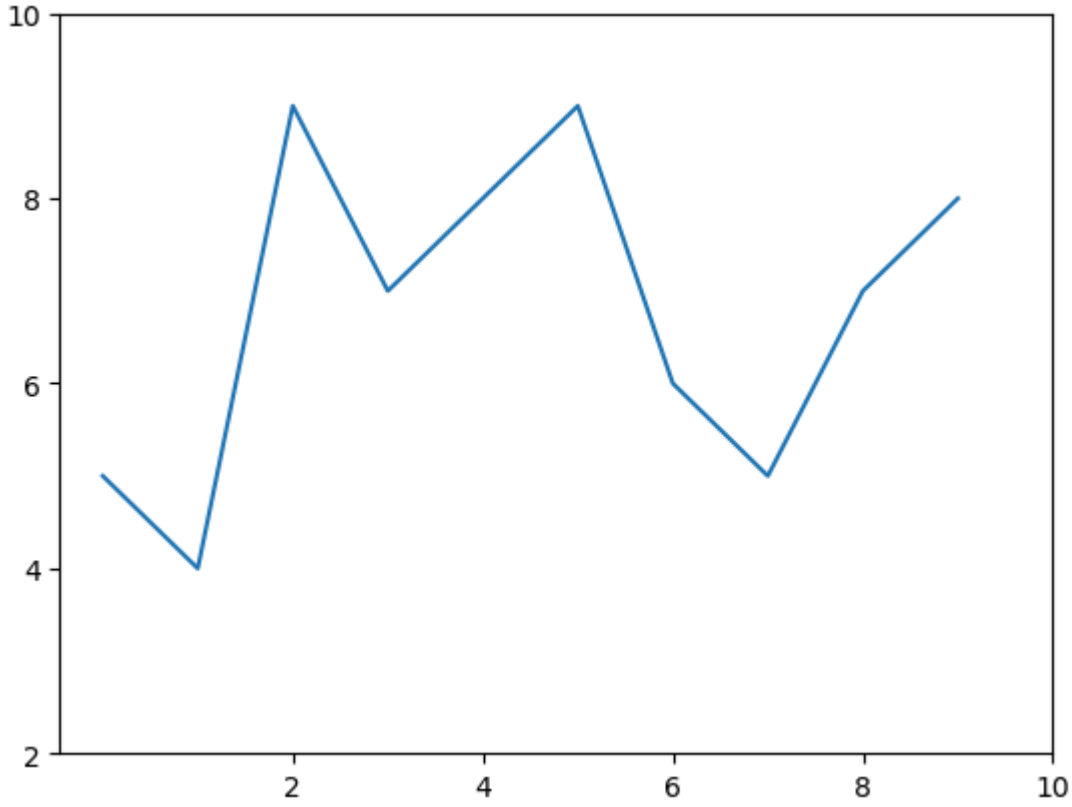
In [93]:
```
x15 = np.arange(1, 5)
plt.plot(x15, x15*1.5, x15, x15*3.0, x15, x15/3.0)
```

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

Out[93]: (0.0, 12.0)

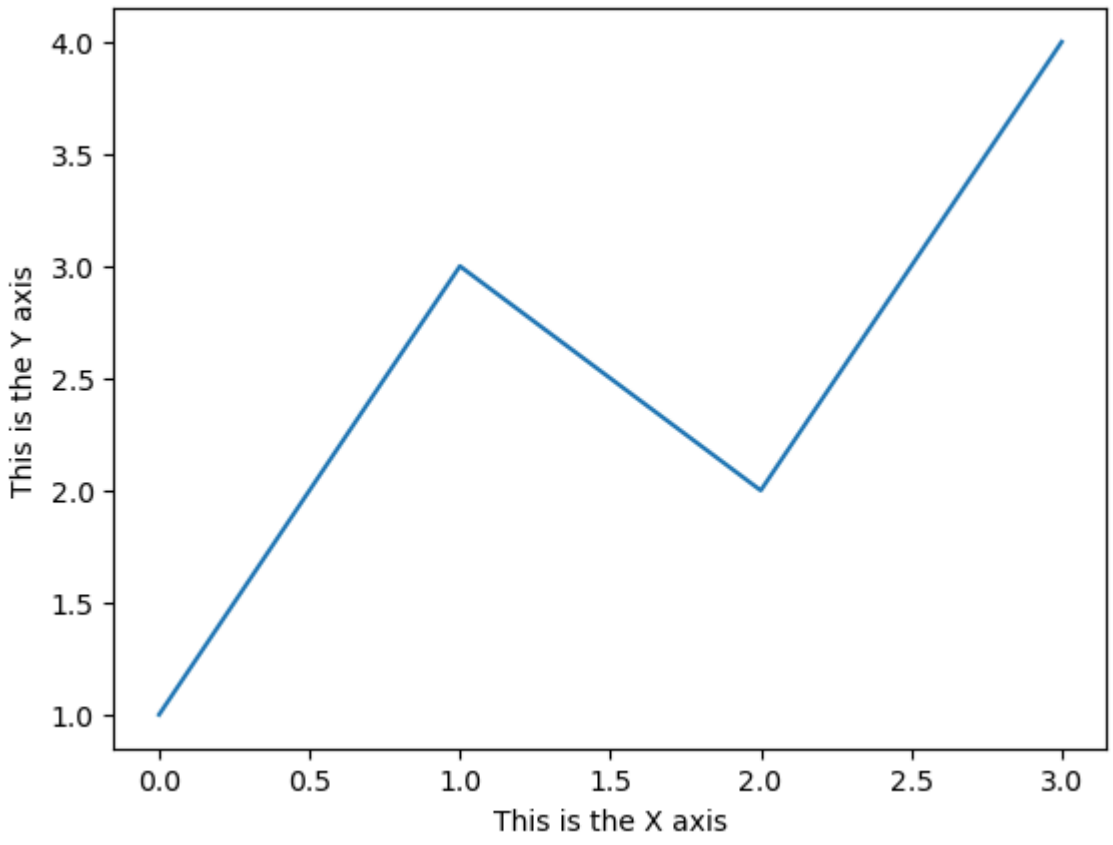


In [97]:
```
u = [5, 4, 9, 7, 8, 9, 6, 5, 7, 8]
plt.plot(u)
plt.xticks([2, 4, 6, 8, 10])
plt.yticks([2, 4, 6, 8, 10])
plt.show()
```



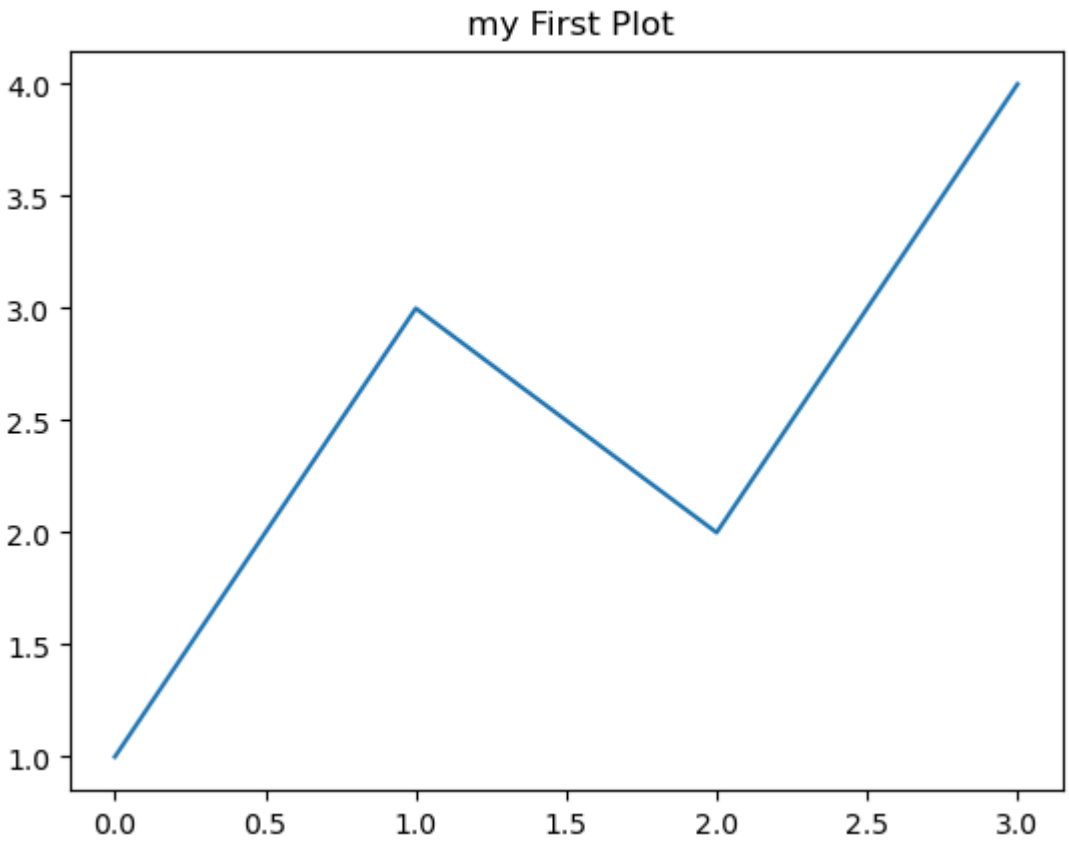
In [101…

```
plt.plot([1, 3, 2, 4]) #adding labels
plt.xlabel('This is the X axis')
plt.ylabel('This is the Y axis')
plt.show()
```



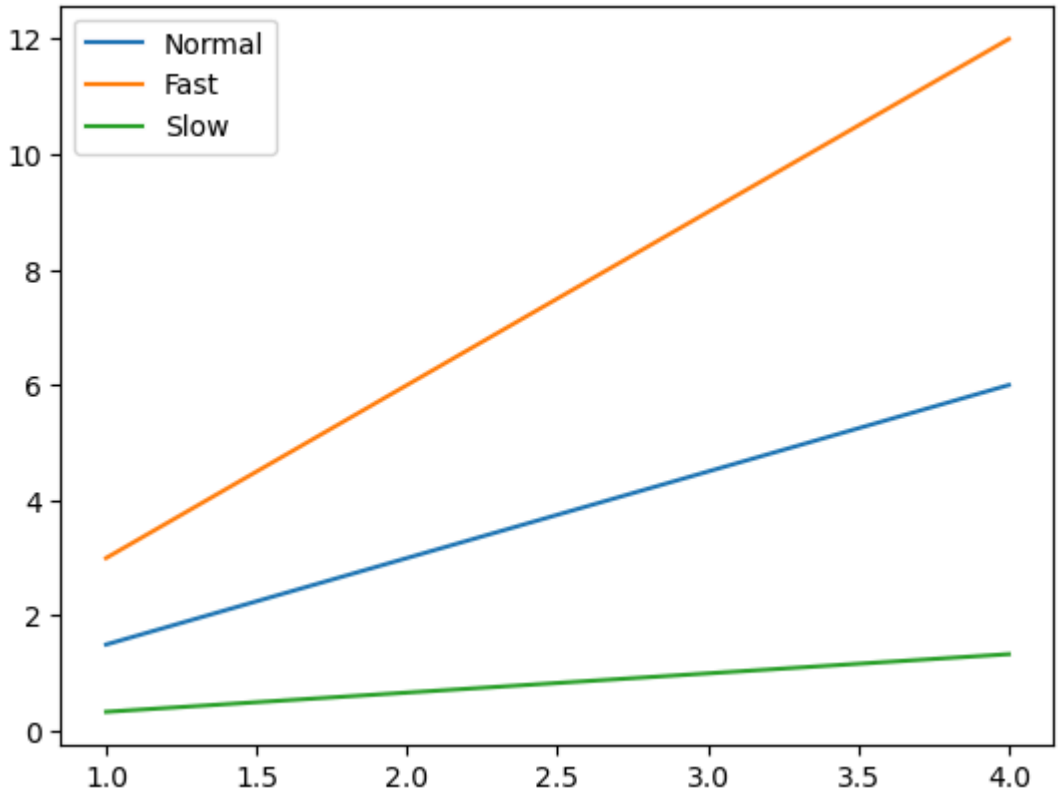
In [105…

```
plt.plot([1, 3, 2, 4]) # adding title
plt.title(' my First Plot')
plt.show()
```



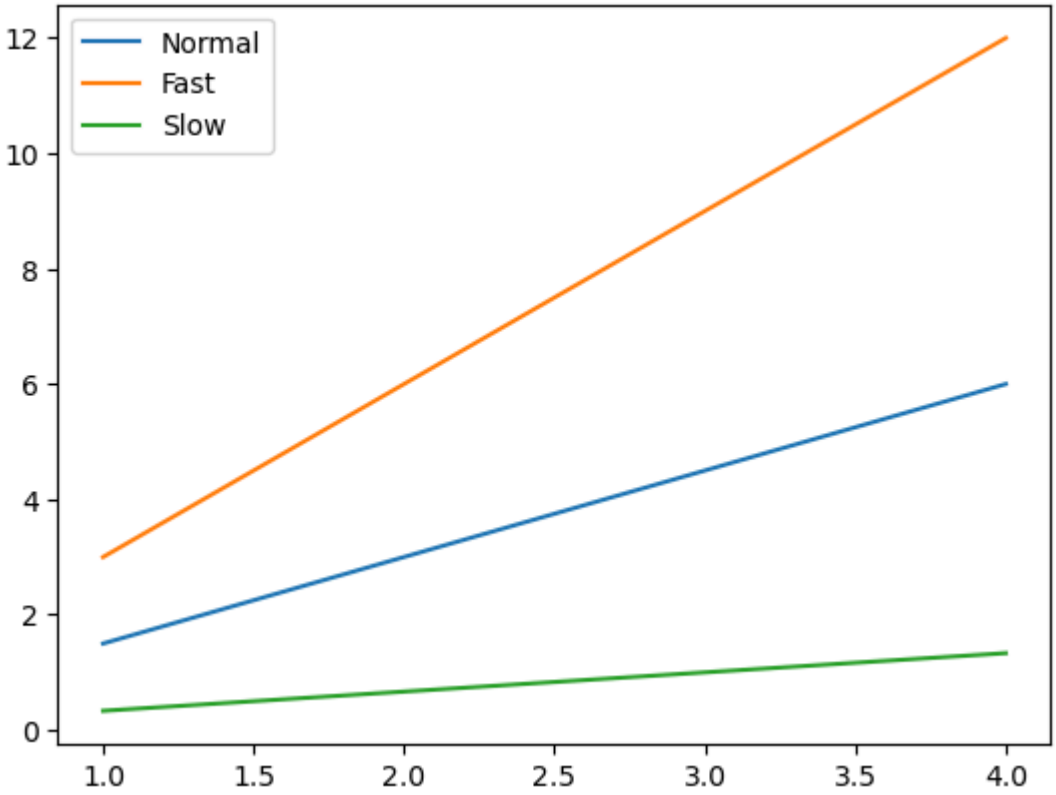
In [107…

```
x15 = np.arange(1, 5) #adding Legend
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']);
```



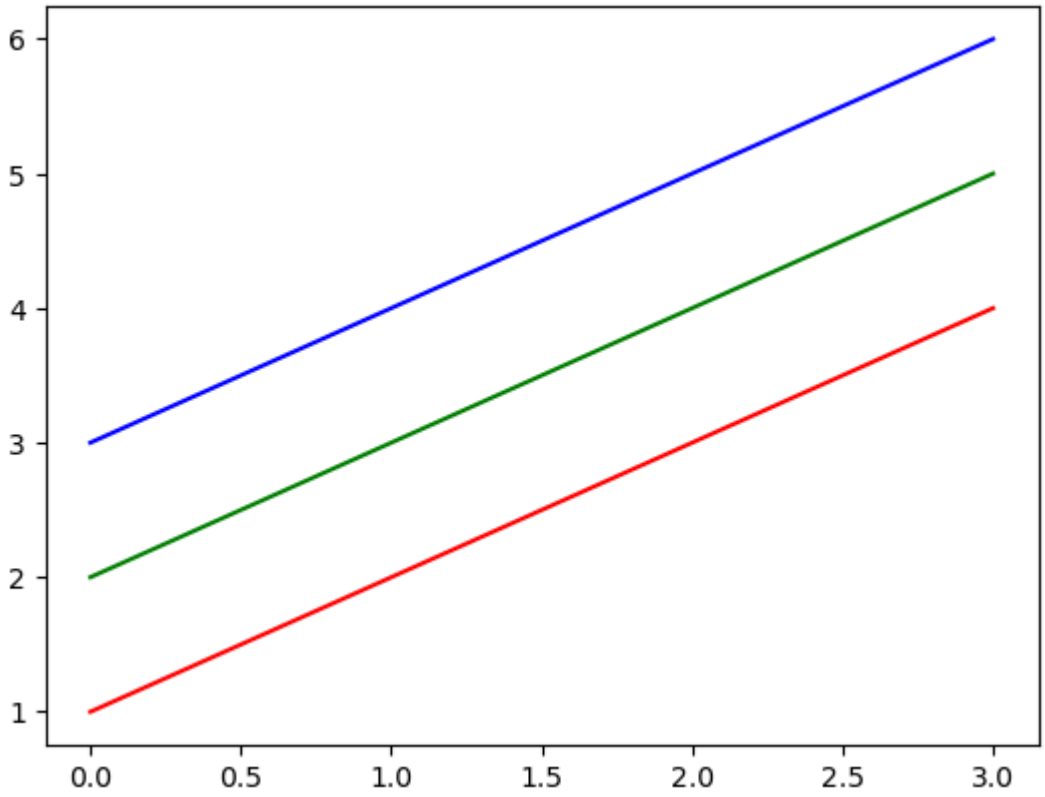
In [111…

```
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();
```



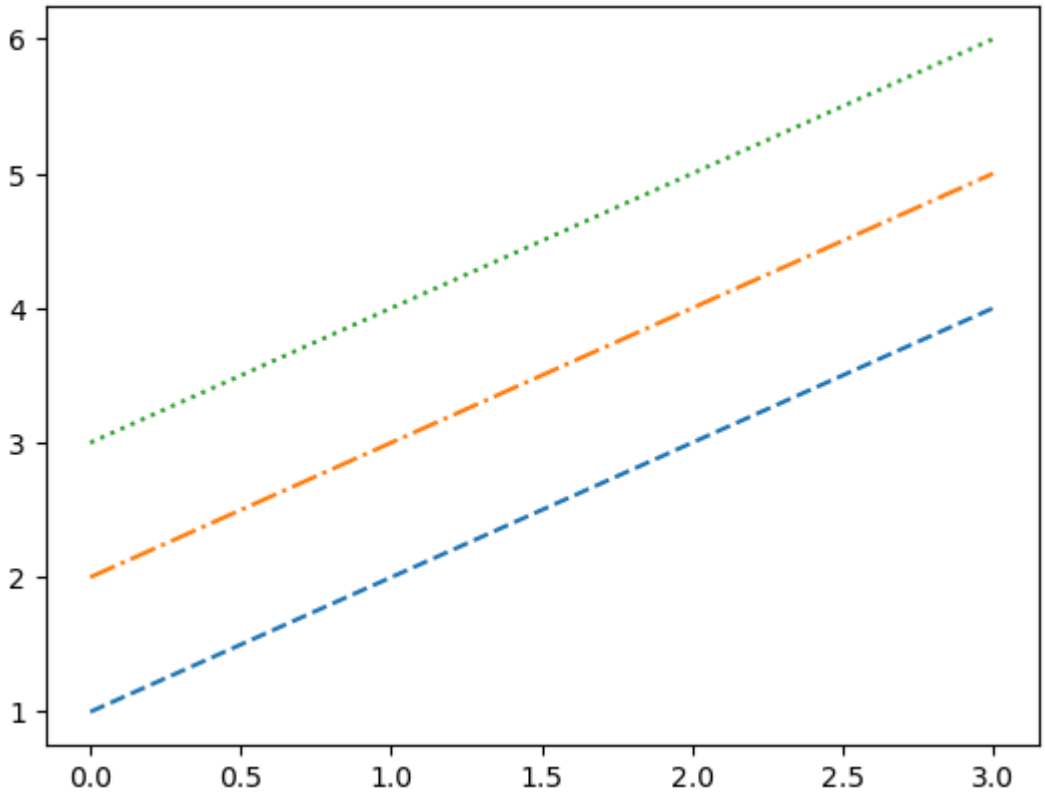
In [113…

```
x16 = np.arange(1, 5) # control colours
plt.plot(x16, 'r')
plt.plot(x16+1, 'g')
plt.plot(x16+2, 'b')
plt.show()
```



In [115…

```
x16 = np.arange(1, 5) # control line styles
plt.plot(x16, '--', x16+1, '-.', x16+2, ':')
plt.show()
```



In [ ]: