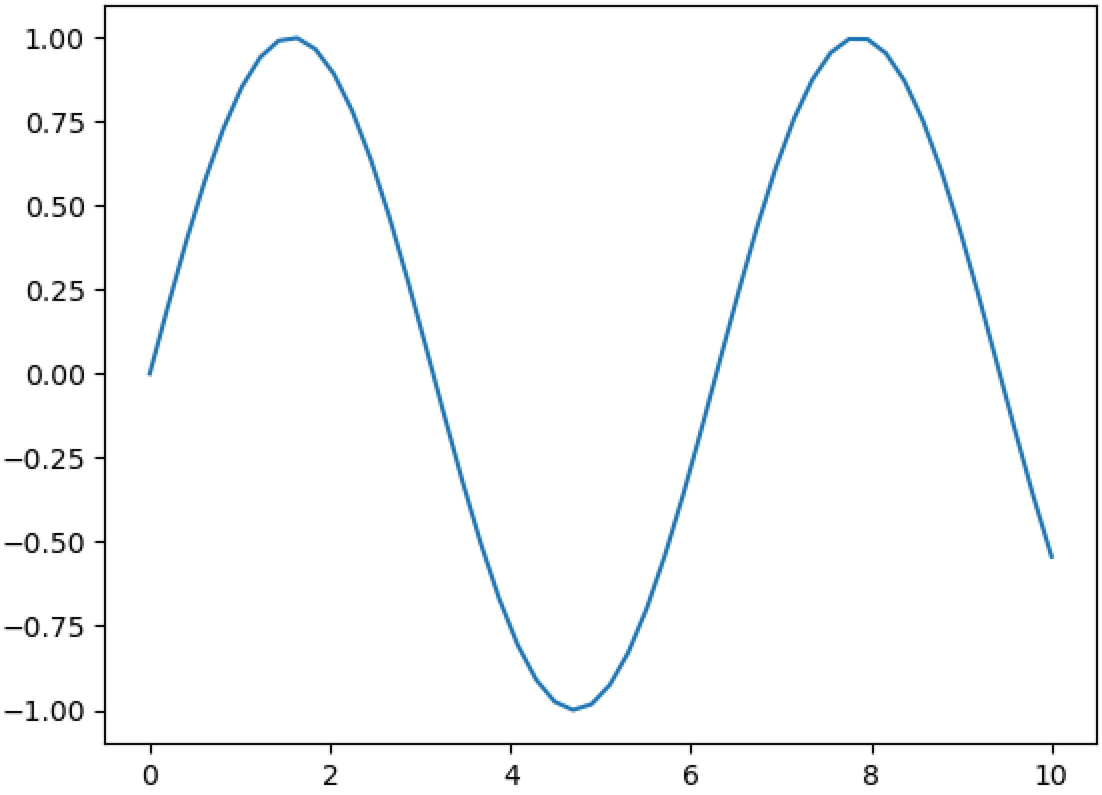
In [1]:

```
import numpy as np
import matplotlib.pyplot as plt #import matplotlib
```

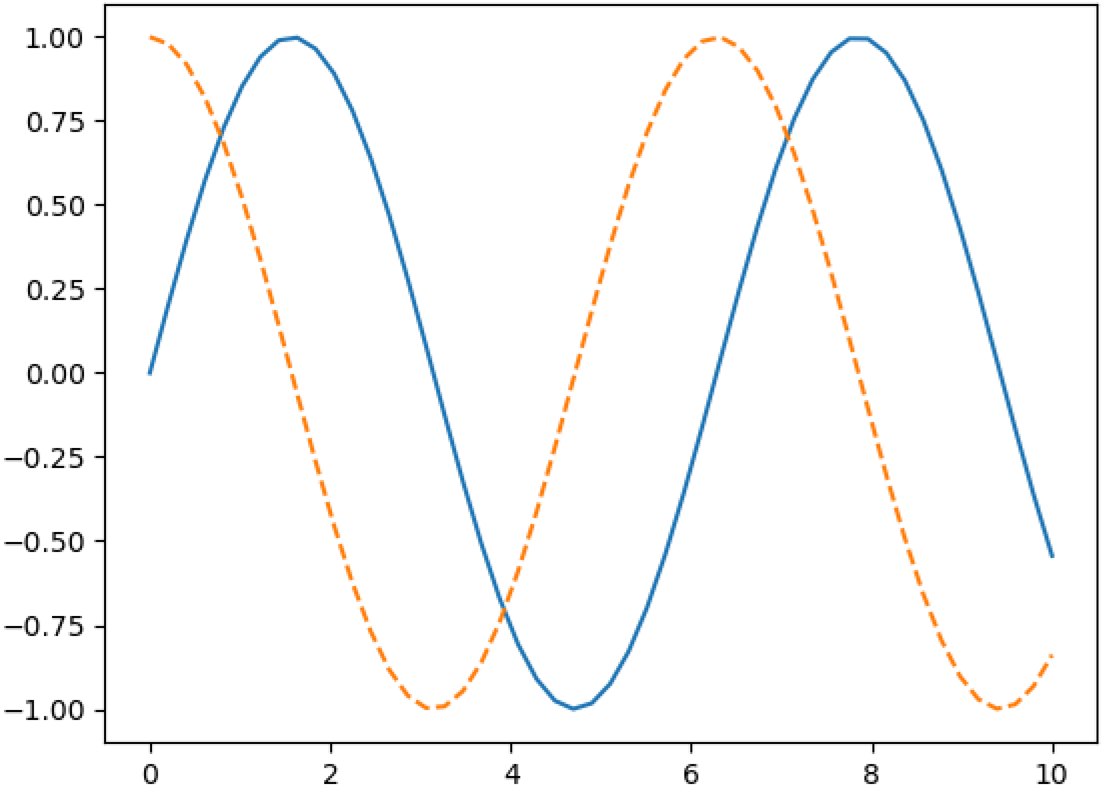
In [2]:

```
%matplotlib inline
x1=np.linspace(0,10,50)
#create a plot figure
#fig =plt.figure()

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



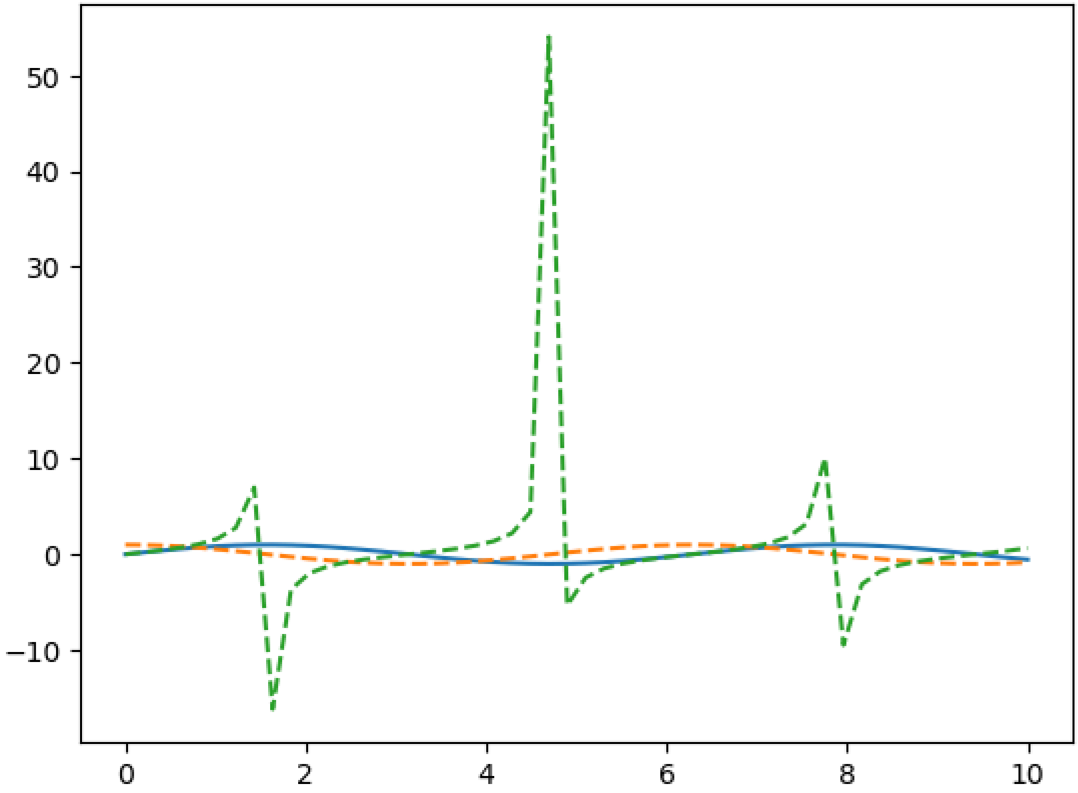
In [3]:

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



In [4]:

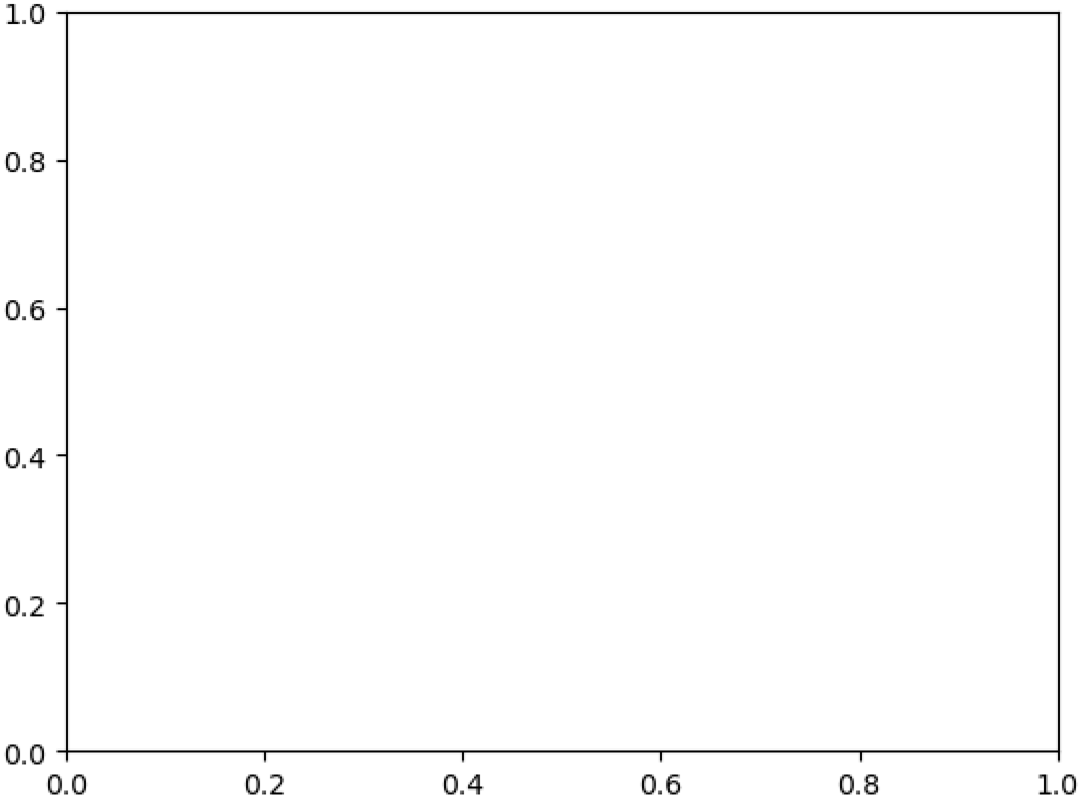
```
plt.plot(x1,np.sin(x1),'-')
plt.plot(x1, np.cos(x1), '--')
plt.plot(x1, np.tan(x1), '--')
plt.show()
```



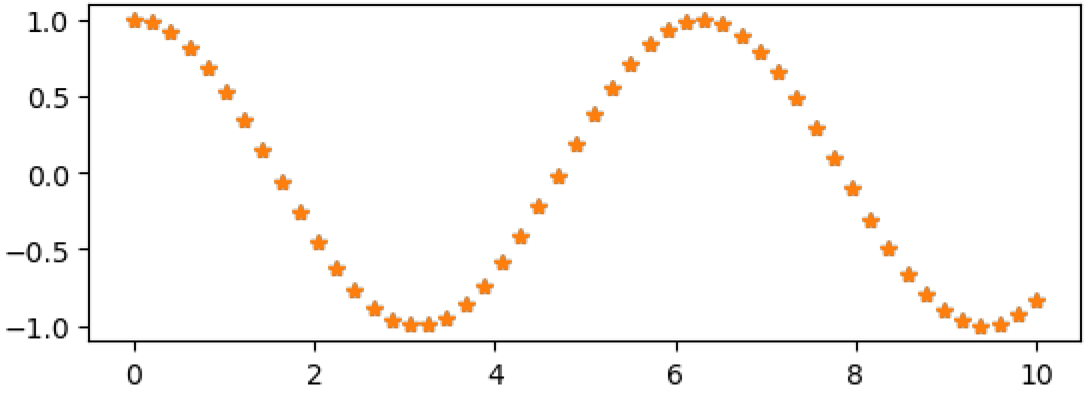
In [6]:
```
plt.gcf ()
```

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

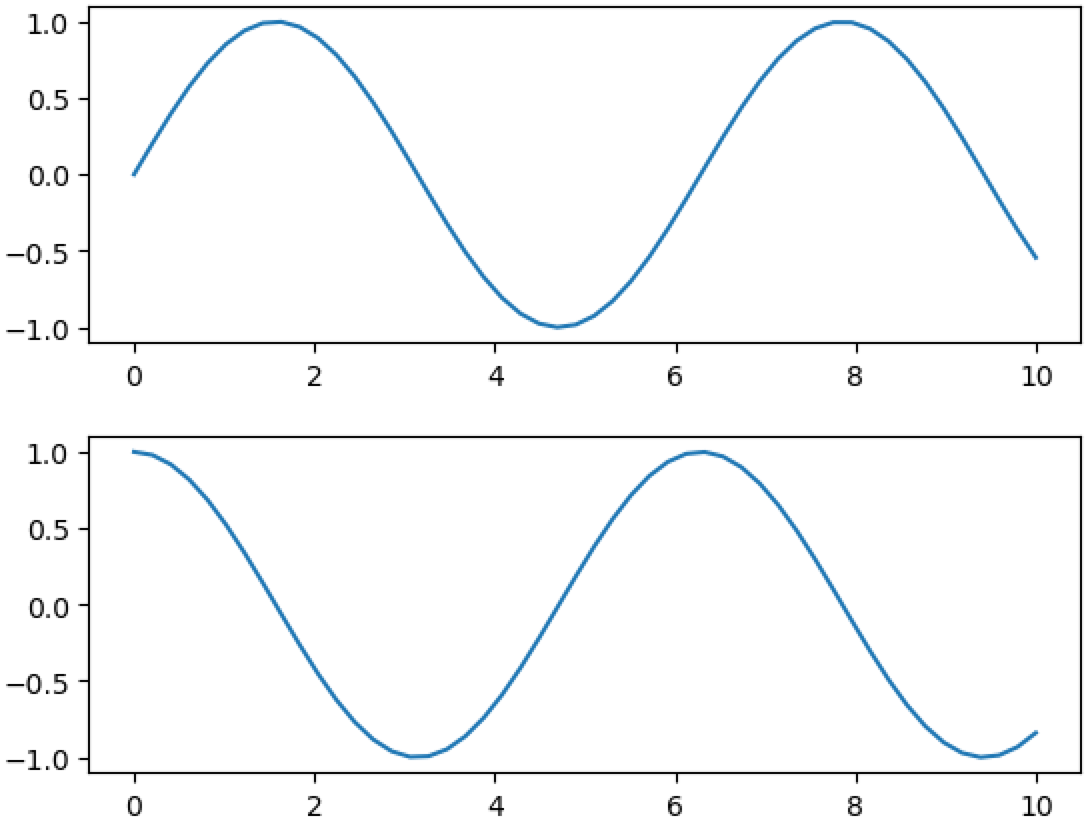
In [8]:
```
plt.gca()
plt.show()
```



In [10]:
```
plt.subplot(2,1,1) #(rows,columns,panel number)
plt.plot(x1,np.cos(x1),'*')
plt.show()
```



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

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



In [13]:

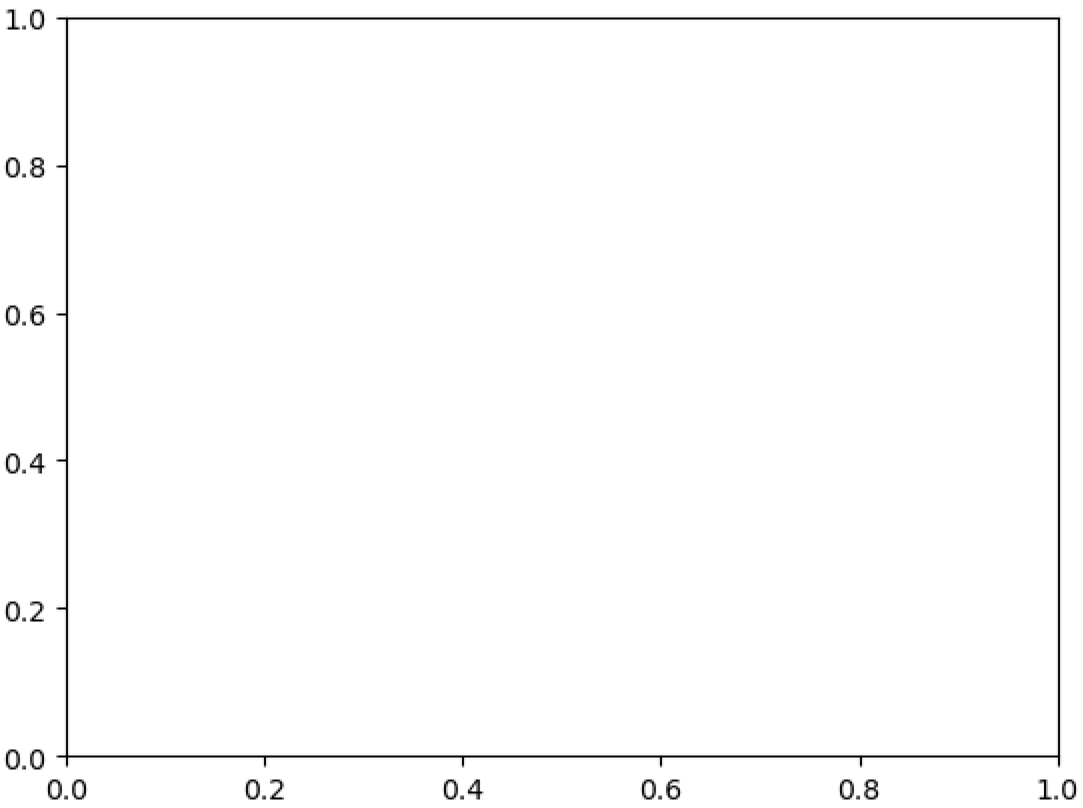
```
#get current figur information
print(plt.gcf())
```

```
Figure(640x480)
```

In [15]:

```
#get current axis information
print(plt.gca())
plt.show()
```

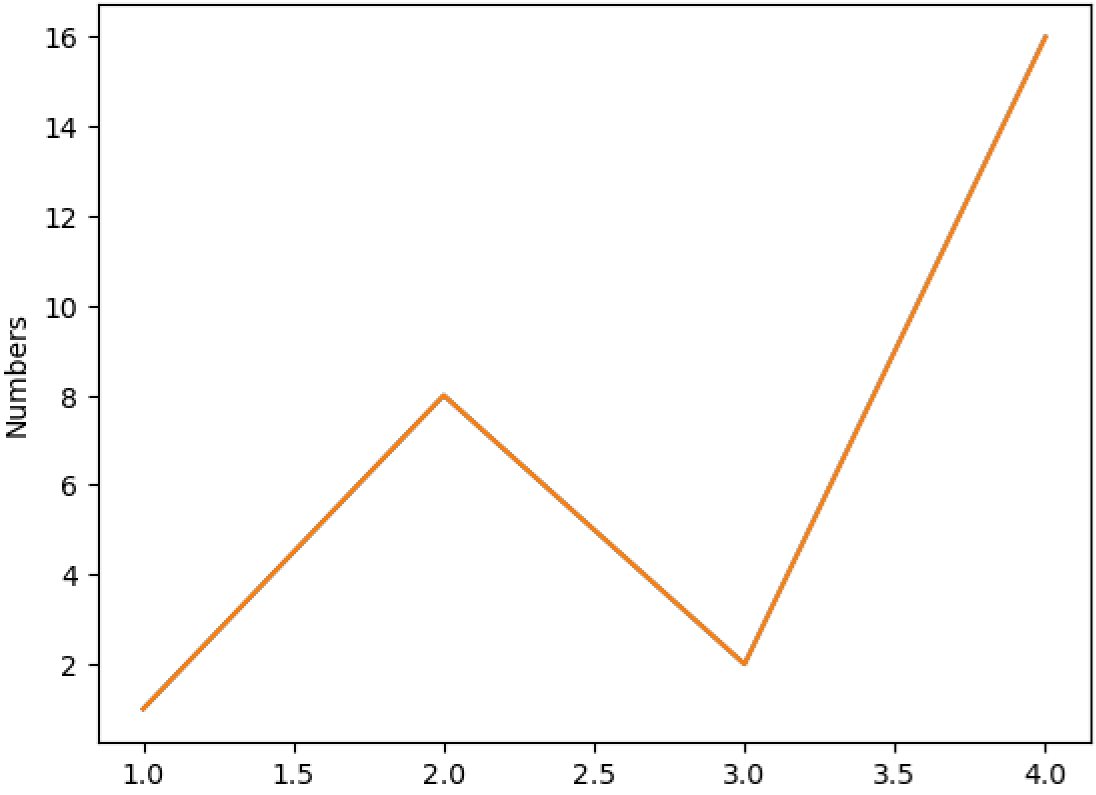
```
Axes(0.125,0.11;0.775x0.77)
```



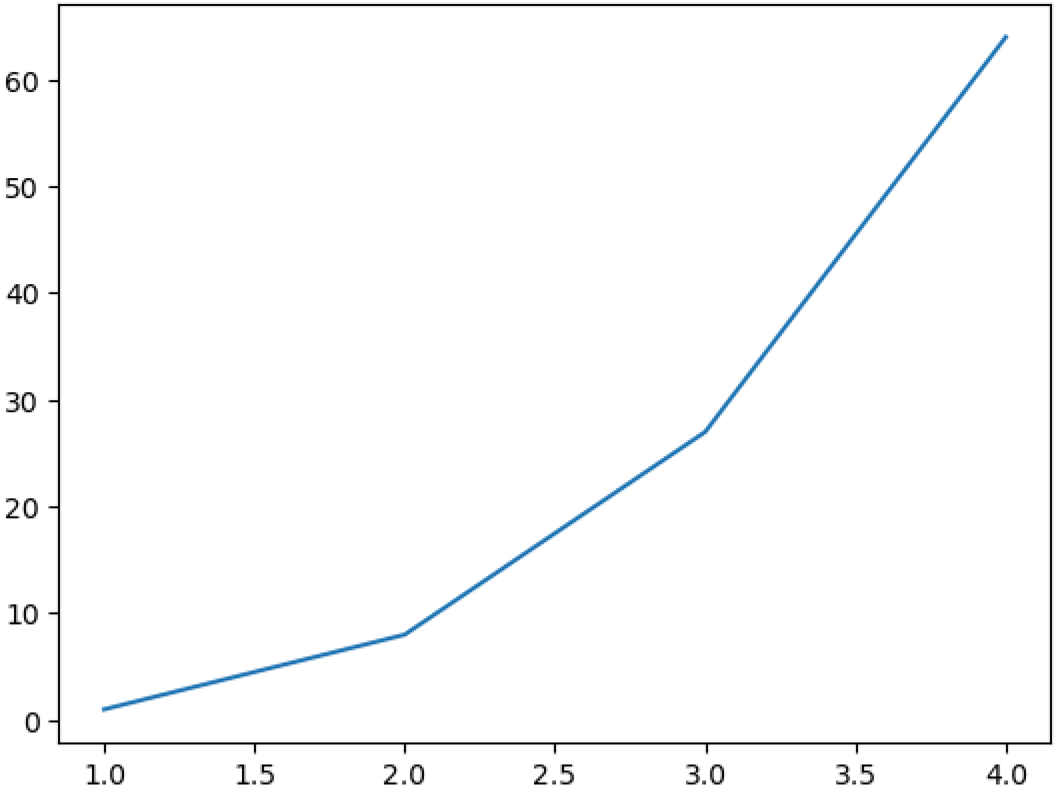
visualization with pyplot

In [17]:

```
plt.plot([1,2,3,4],[1,8,2,16])
plt.ylabel('Numbers')
plt.show()
```

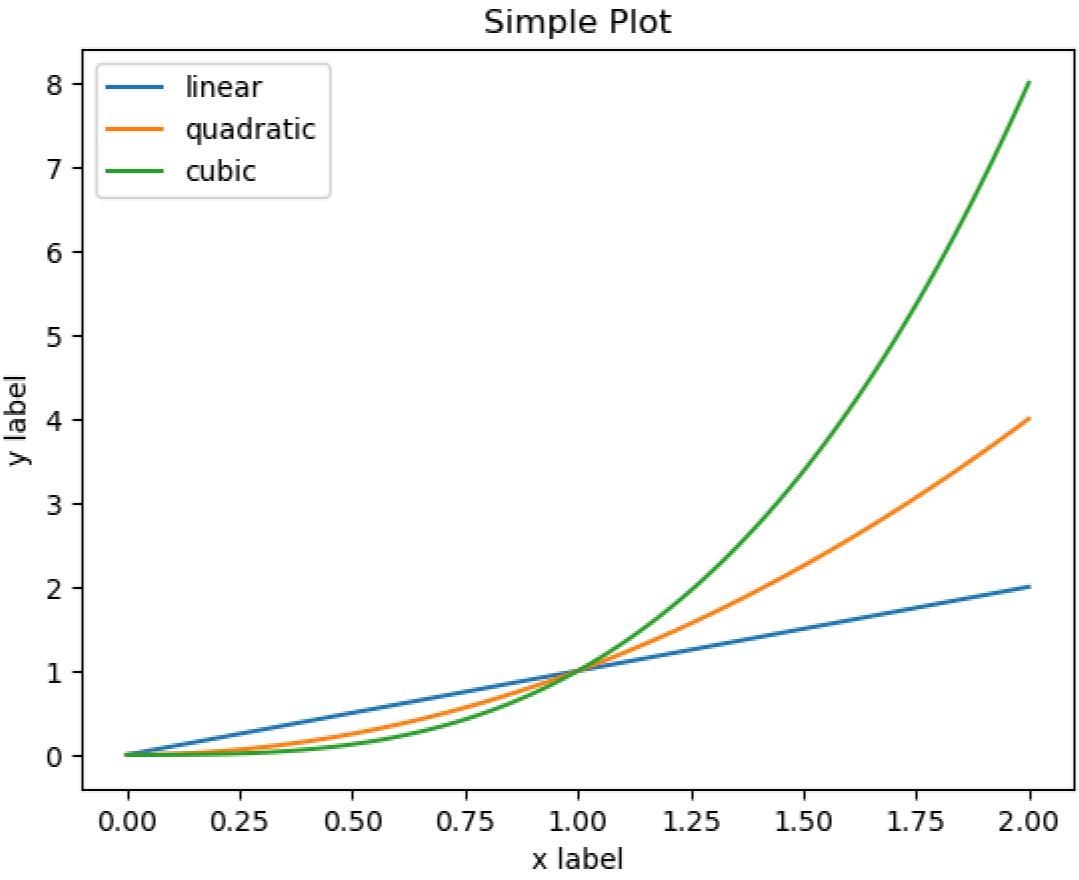


In [18]:
```
plt.plot([1,2,3,4],[1,8,27,64])
plt.show()
```

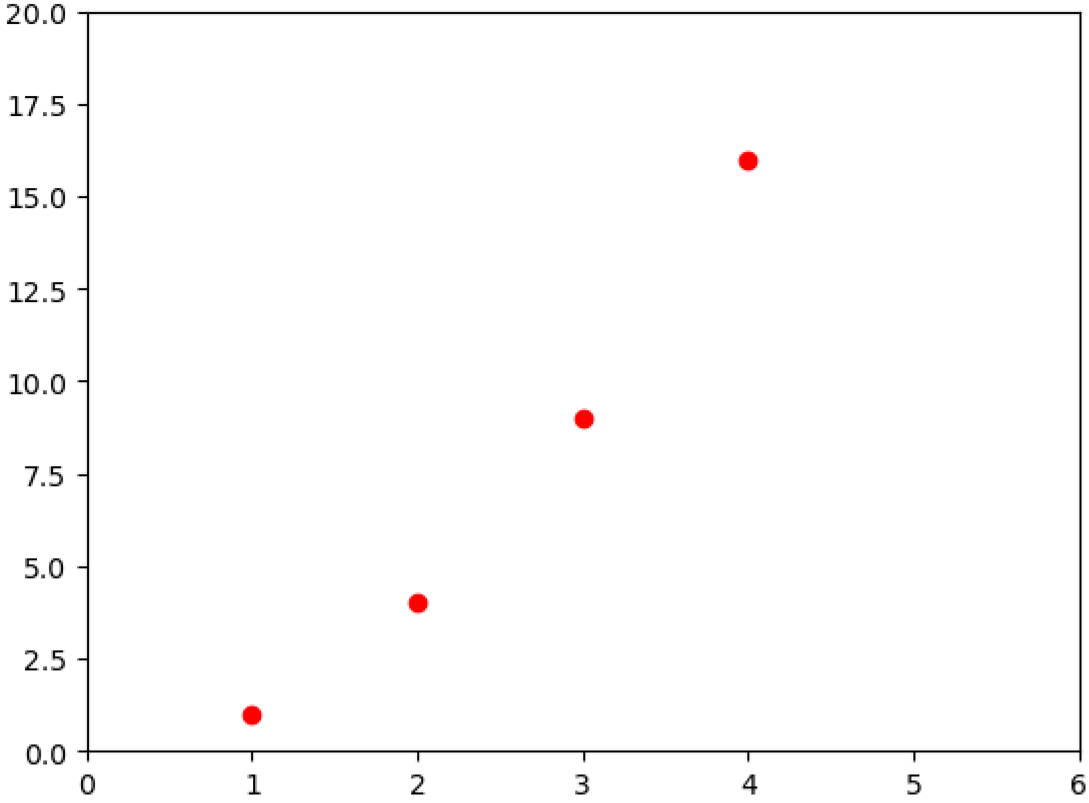


In [19]:
```
x = np.linspace(0, 2, 100)
```

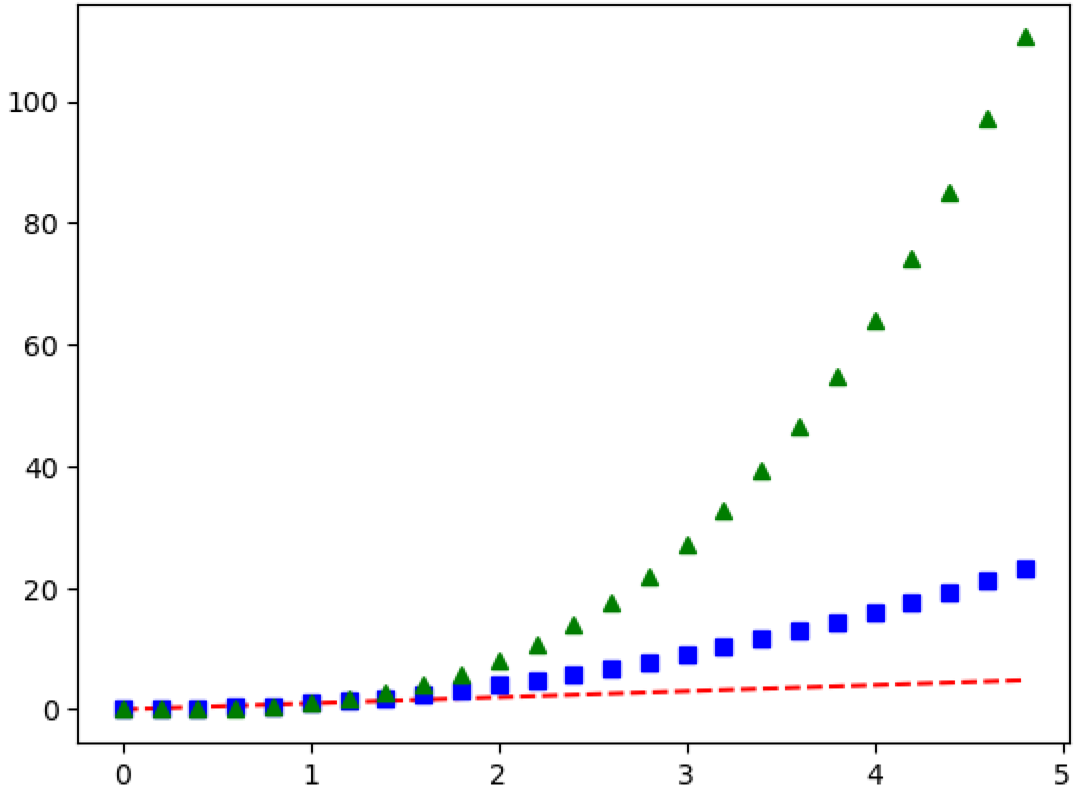
```
plt.plot(x, x, label='linear')
plt.plot(x, x**2, label='quadratic')
plt.plot(x, x**3, label='cubic')

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

plt.title("Simple Plot")

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



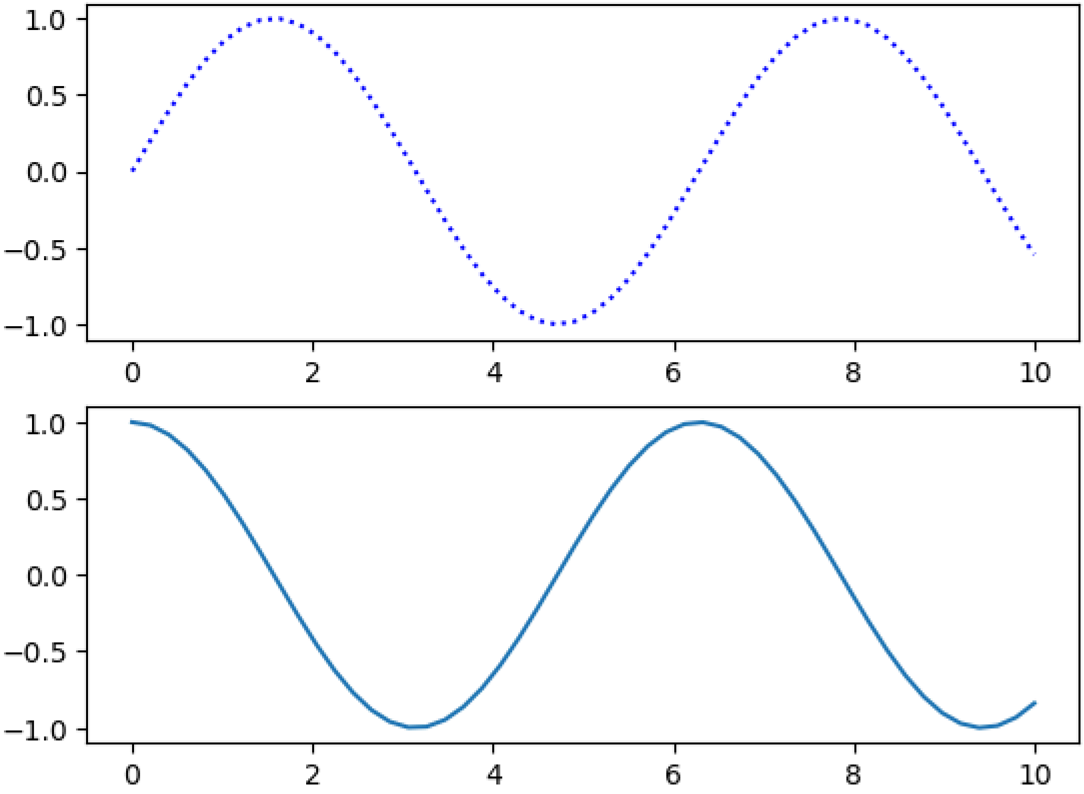
In [20]:
```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```



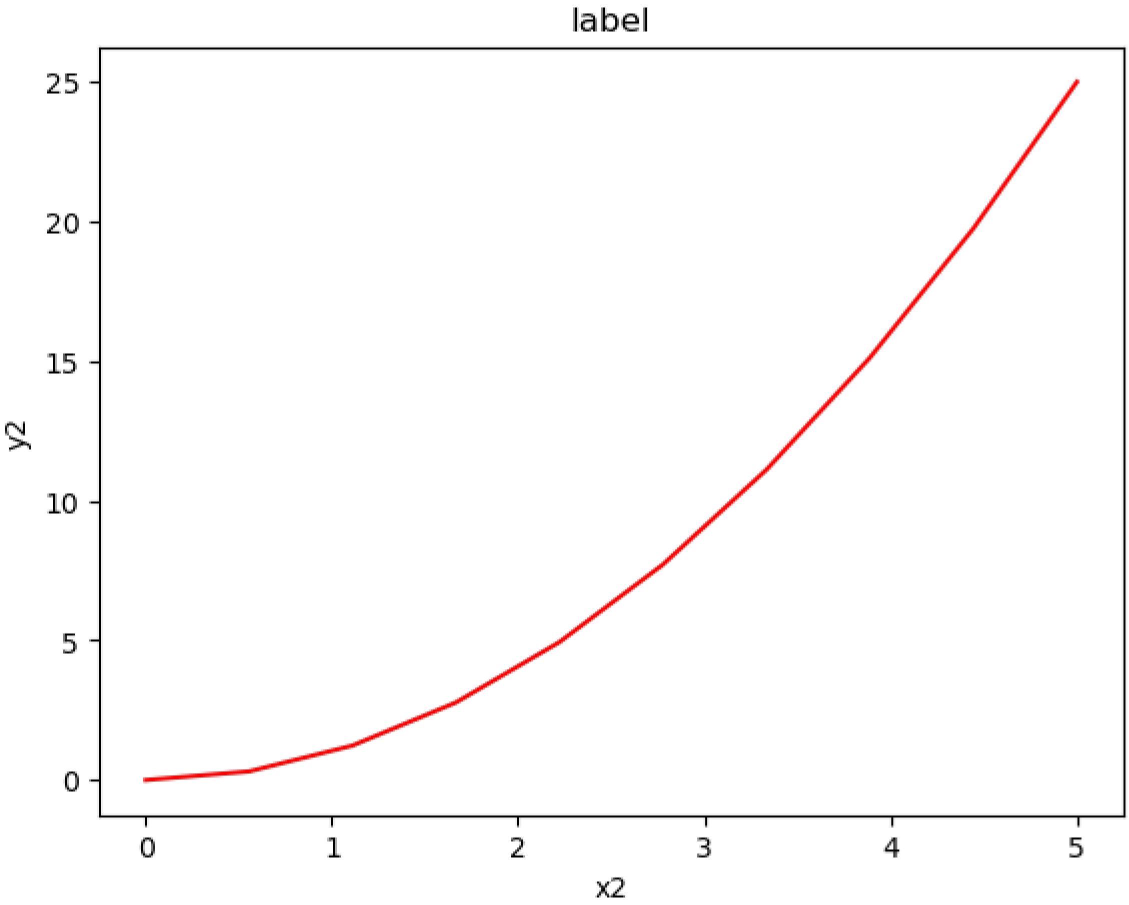
In [21]:
```
t = np.arange(0., 5., 0.2)


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



In [28]:
```
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), );
plt.show()
```



In [32]:
```
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('label');
plt.show()
```



In [35]:

```
fig = plt.figure()

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

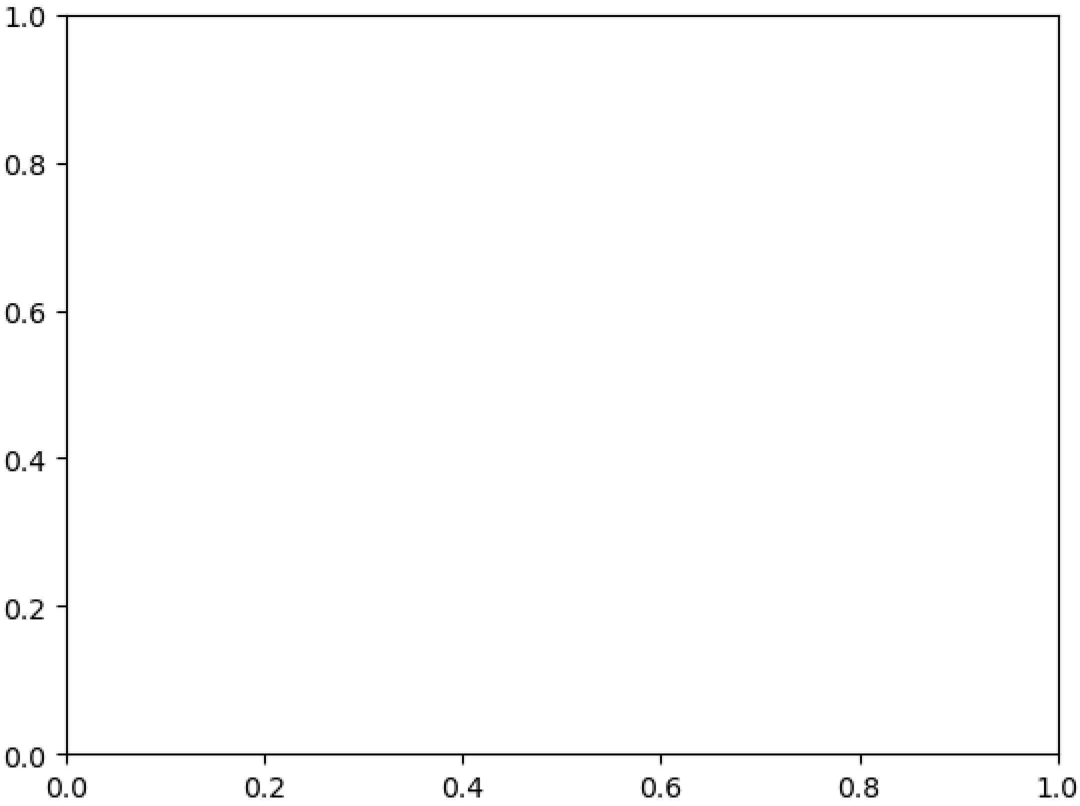
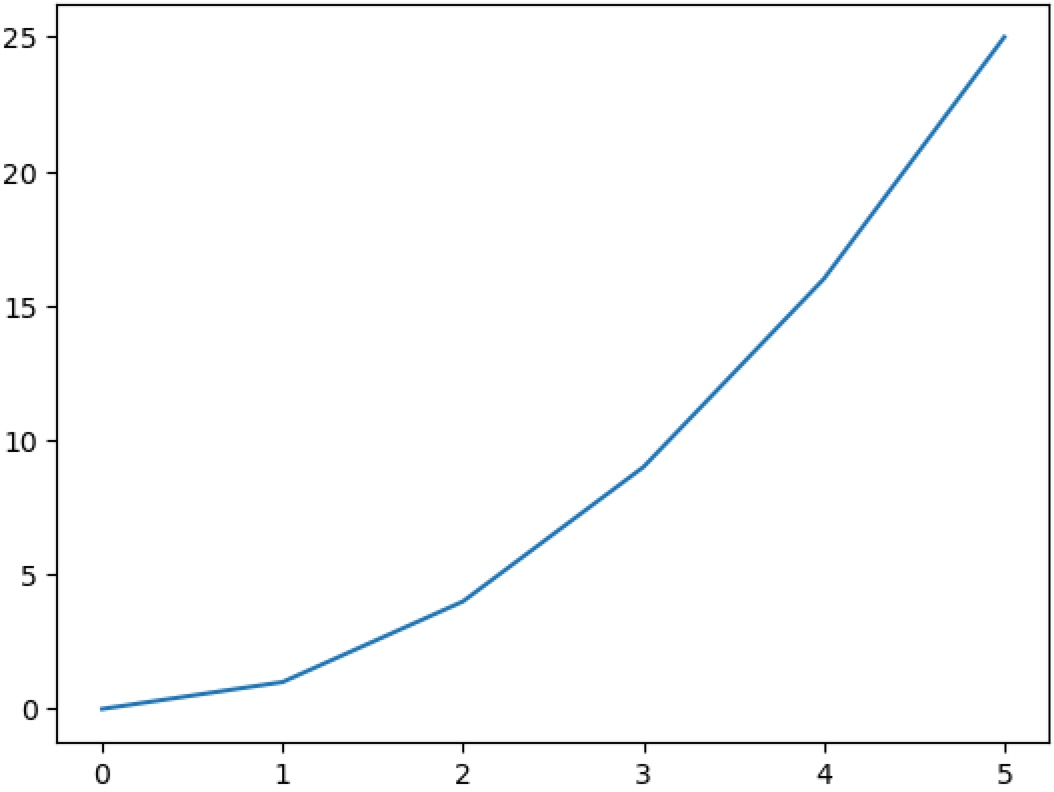
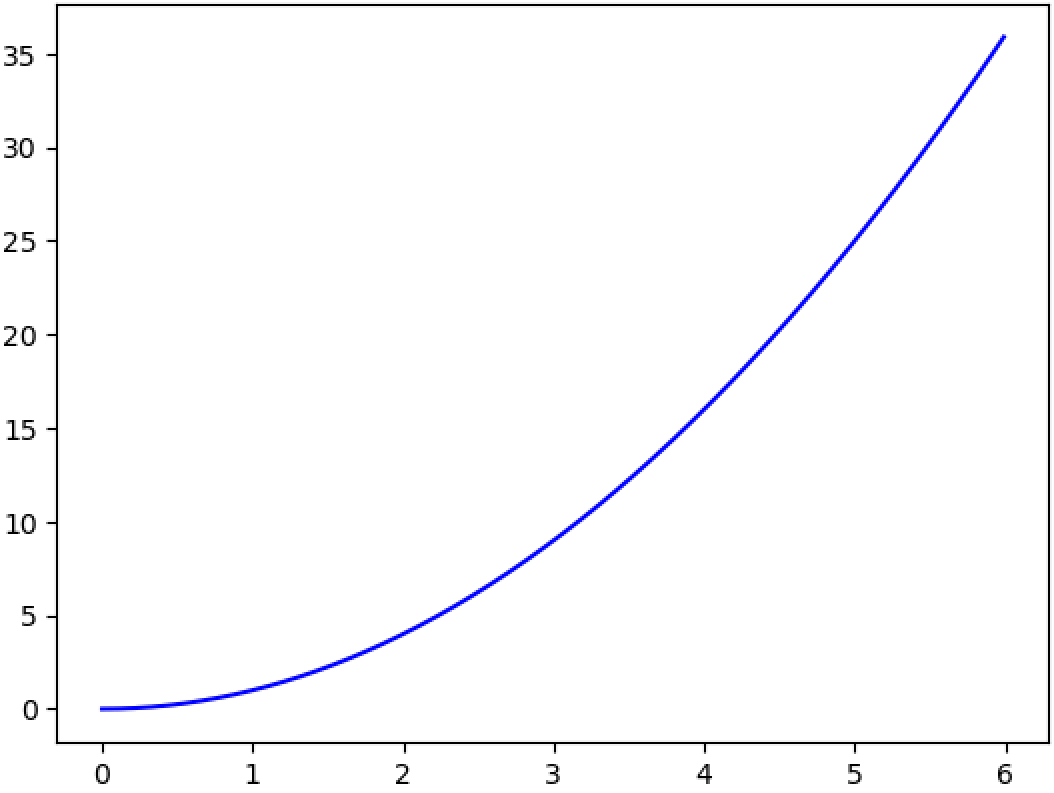


fig = plt.figure() ax1 = fig.add_subplot(2, 2, 1) ax2 = fig.add_subplot(2, 2, 2) ax3 = fig.add_subplot(2, 2, 3) plt.show()

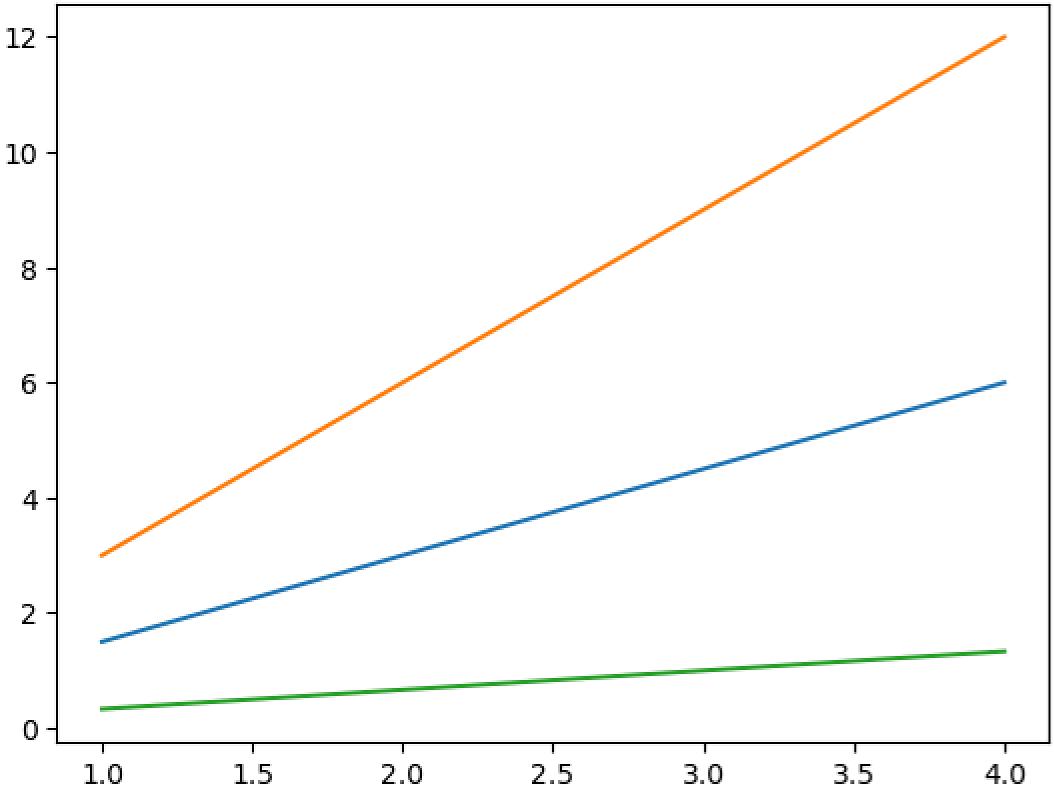
In [37]:
```
x3 = range(6)

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

plt.show()
```



In [38]:
```
x3 = np.arange(0.0, 6.0, 0.01)

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

plt.show()
```

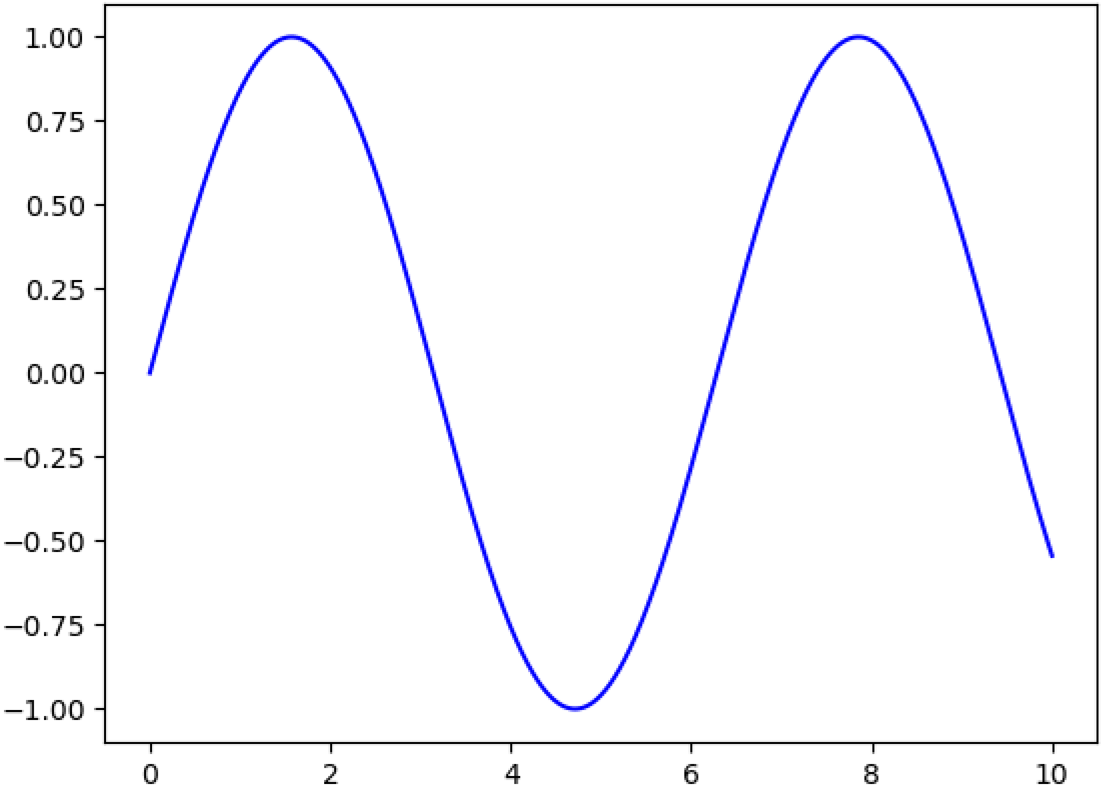


In [39]:
```
x4 = range(1, 5)

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()
```



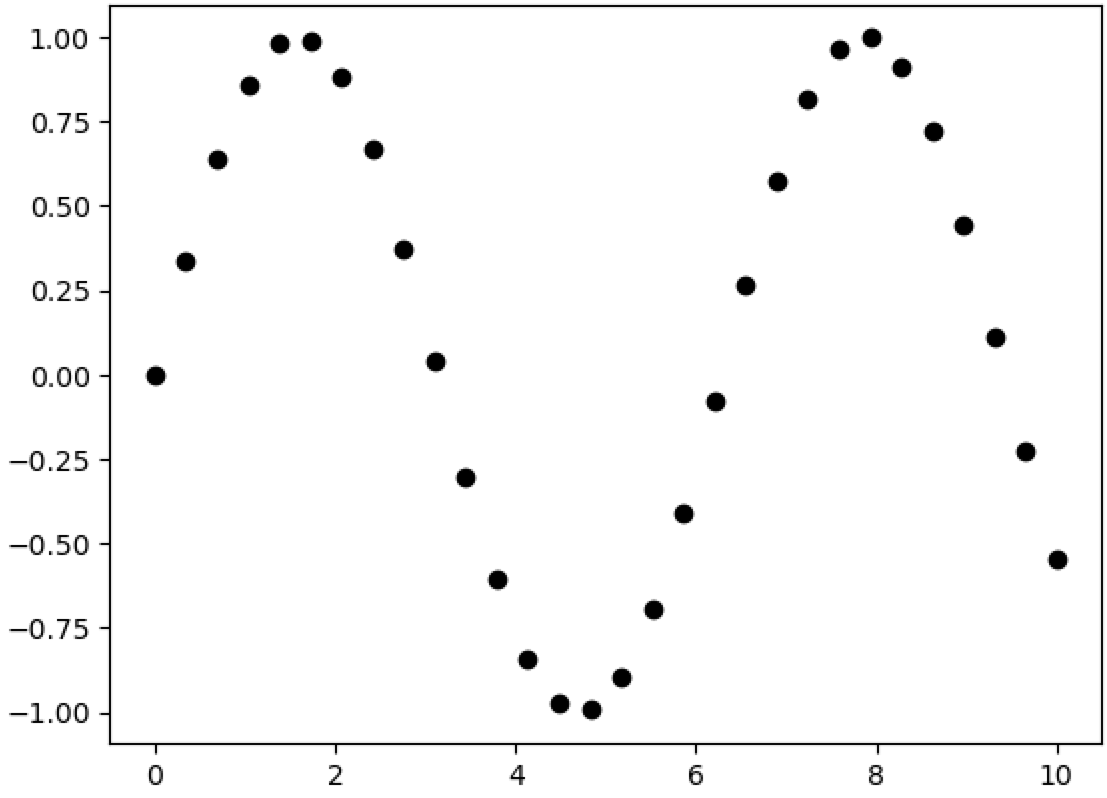
In [40]:
```
# Saving the figure

fig.savefig('plot1.png')
```

In [43]:
```
# Create figure and axes first
fig = plt.figure()

ax = plt.axes()

# Declare a variable x5
x5 = np.linspace(0, 10, 1000)


# Plot the sinusoid function
ax.plot(x5, np.sin(x5), 'b-');
plt.show()
```



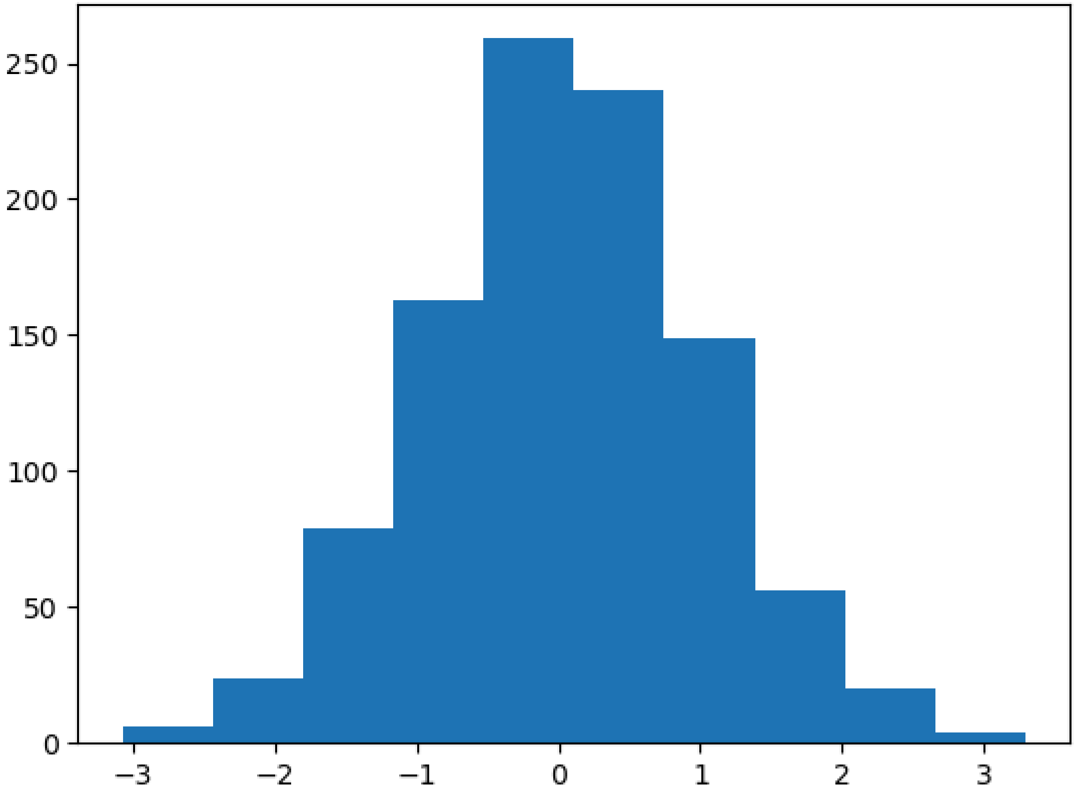
In [44]:
```
x7 = np.linspace(0, 10, 30)

y7 = np.sin(x7)

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

In [45]:
```
plt.show()
```

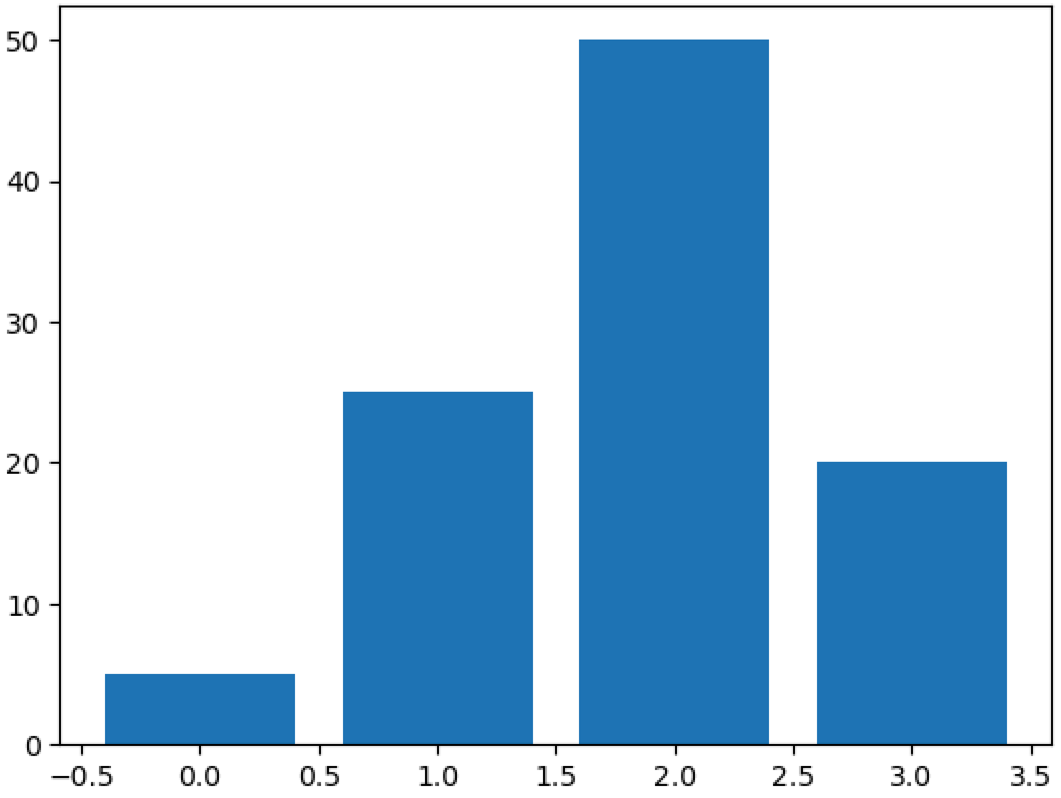


In [47]:

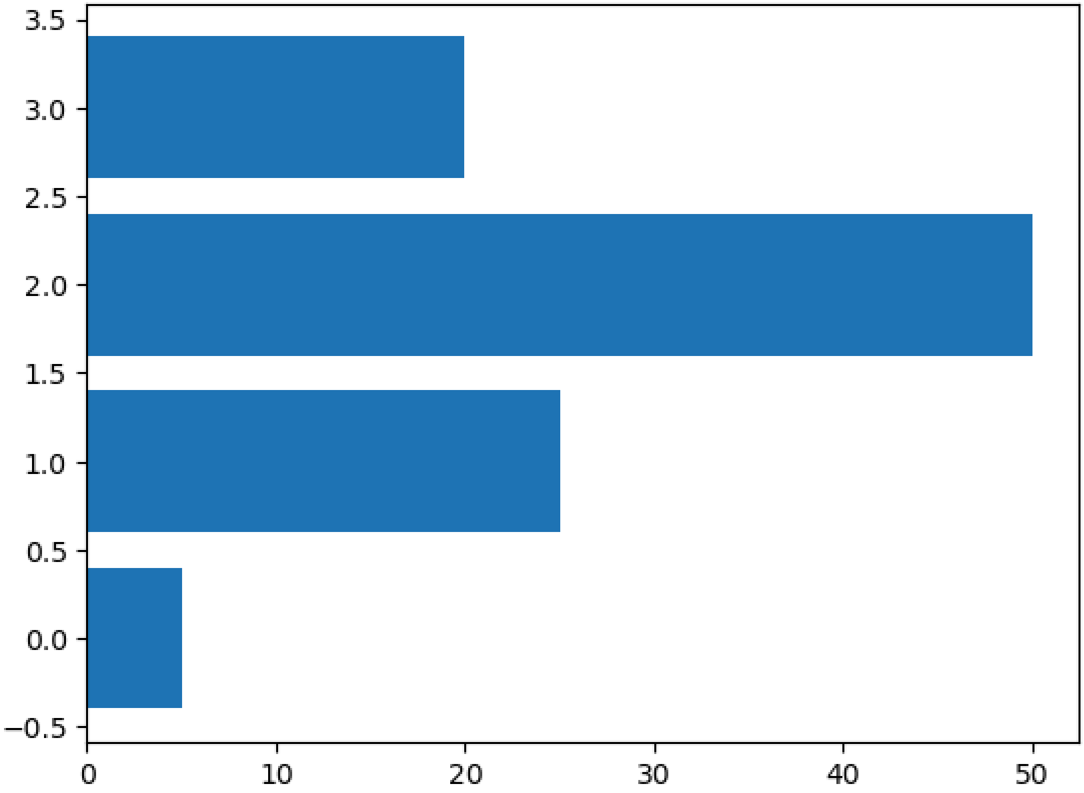
```
data1=np.random.randn(1000)
plt.hist(data1)
plt.show()
```



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

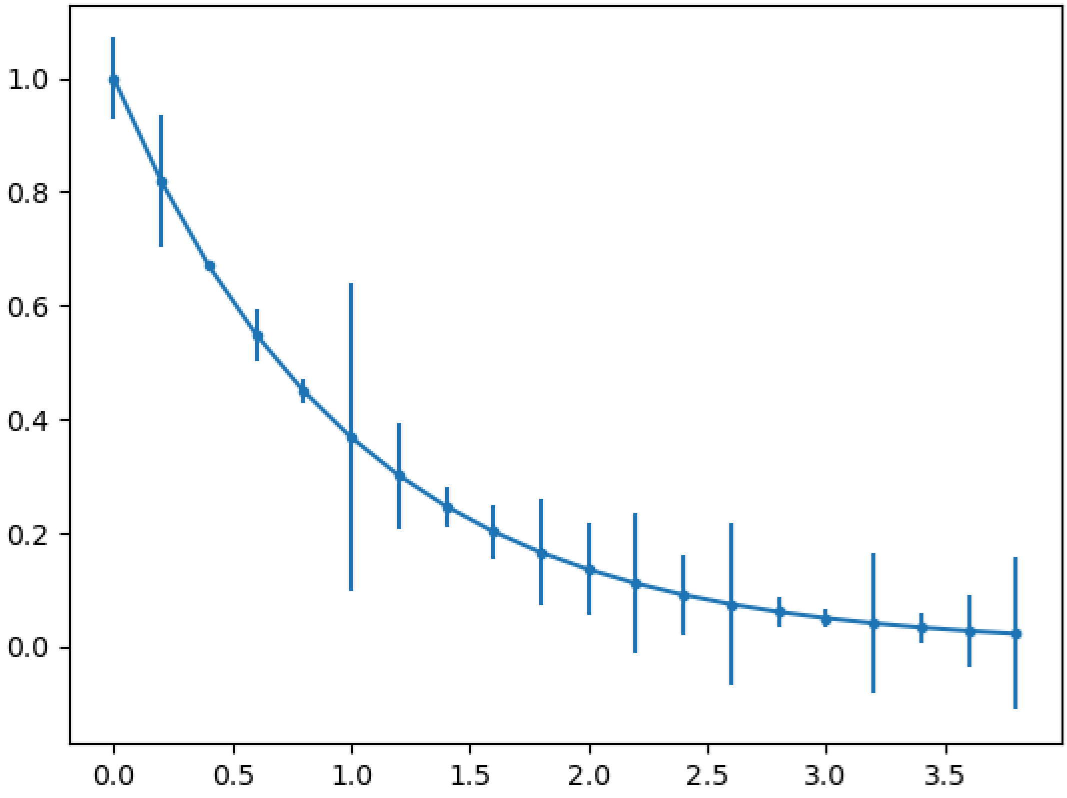


In [51]:
```
plt.barh(range(len(data2)),data2)
plt.show()
```



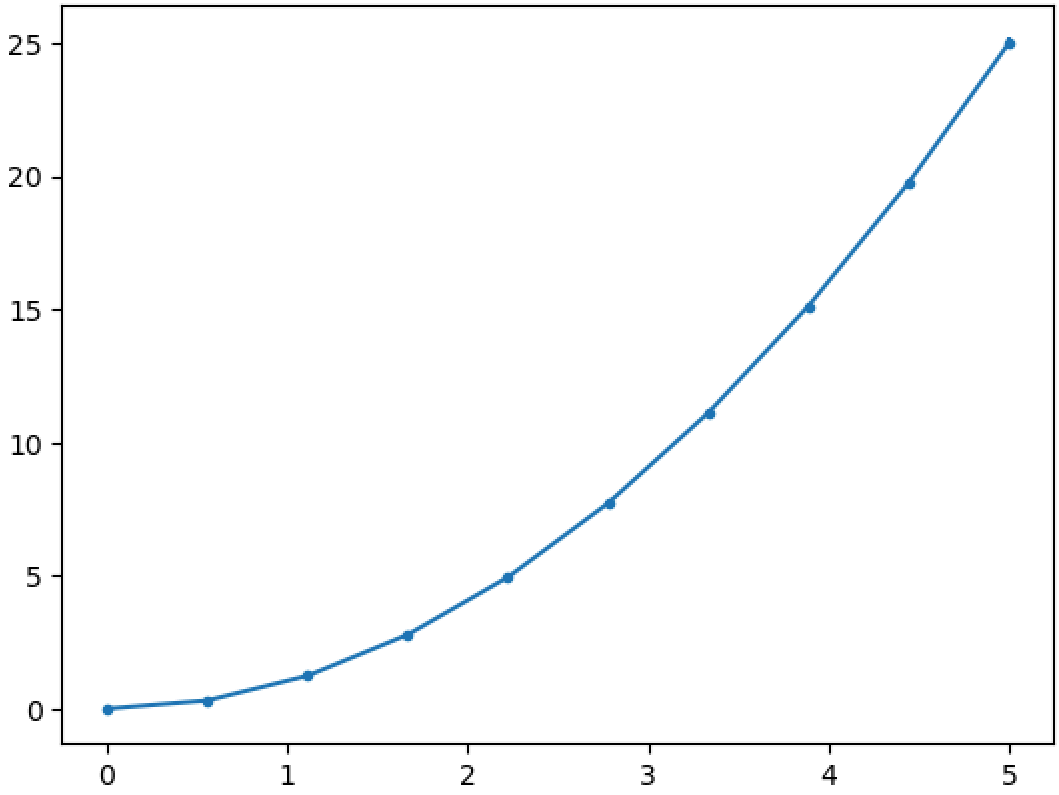
In [52]:

```
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();
```



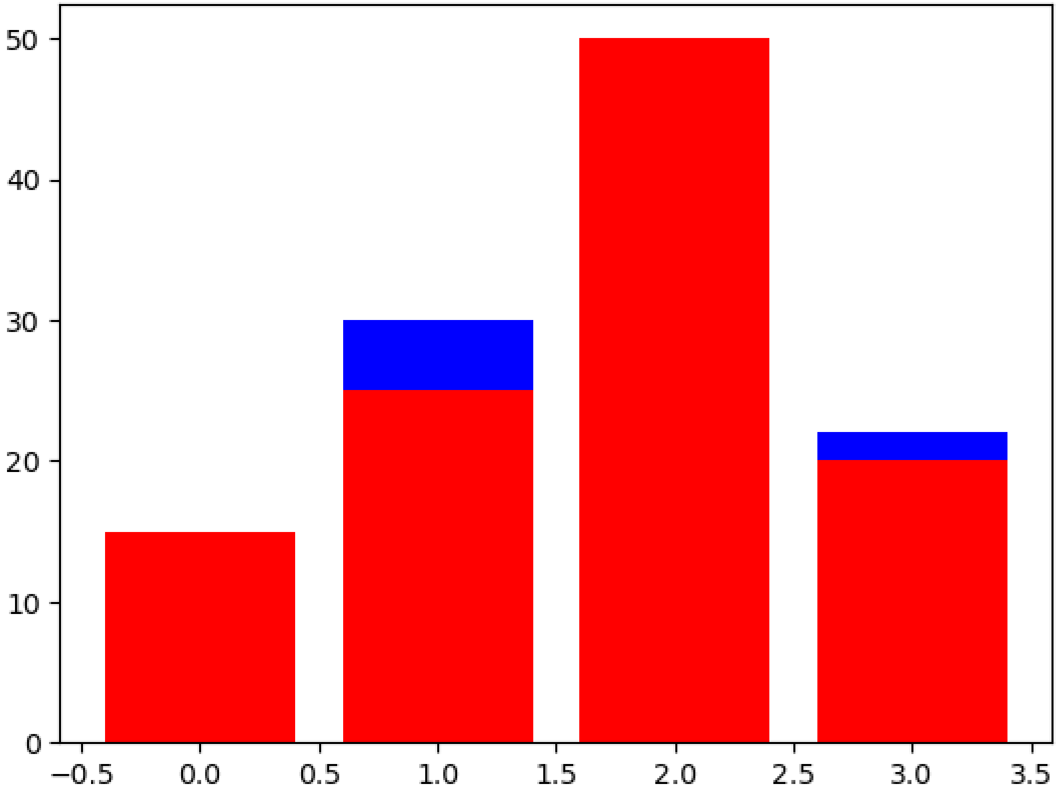
In [55]:

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

y9 = np.exp(-x2)

e1 = 0.1 * np.abs(np.random.randn(len(y2)))

plt.errorbar(x2, y2, yerr = e1, fmt = '.-')

plt.show();
```



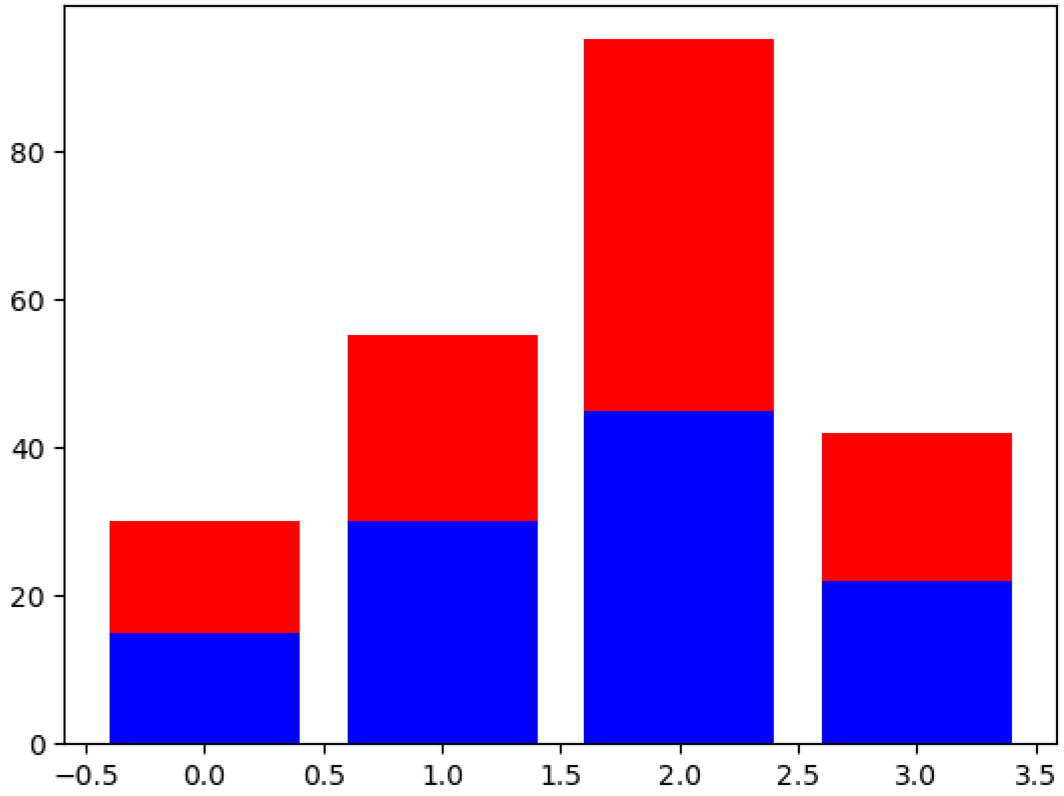
In [56]:

```
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',)

plt.show()
```



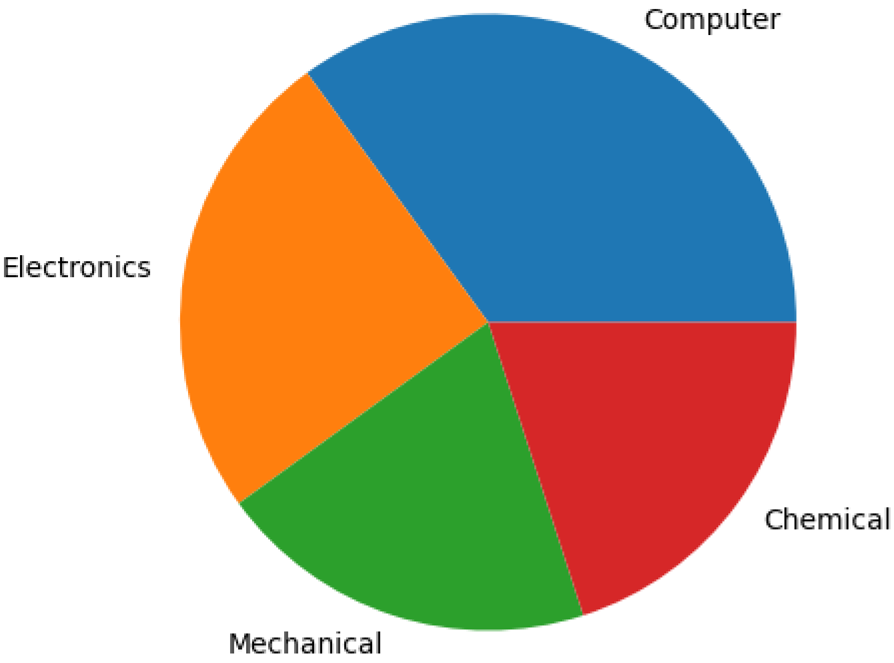
In [57]:

```
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()
```

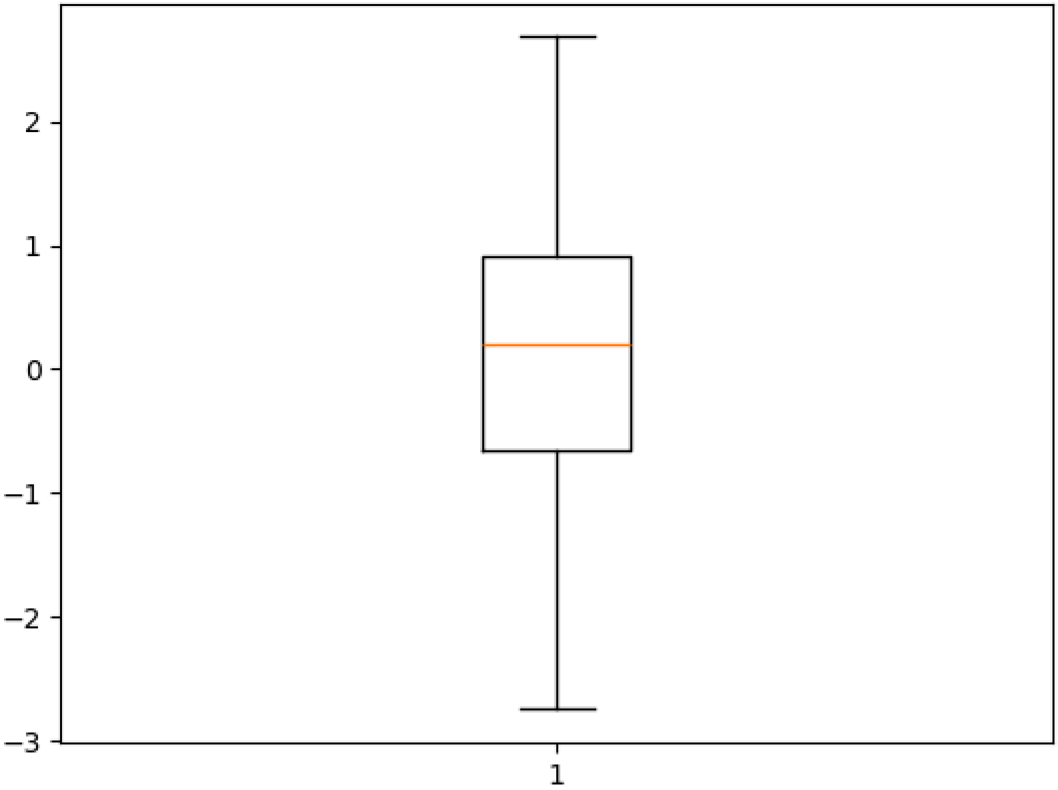


In [59]:

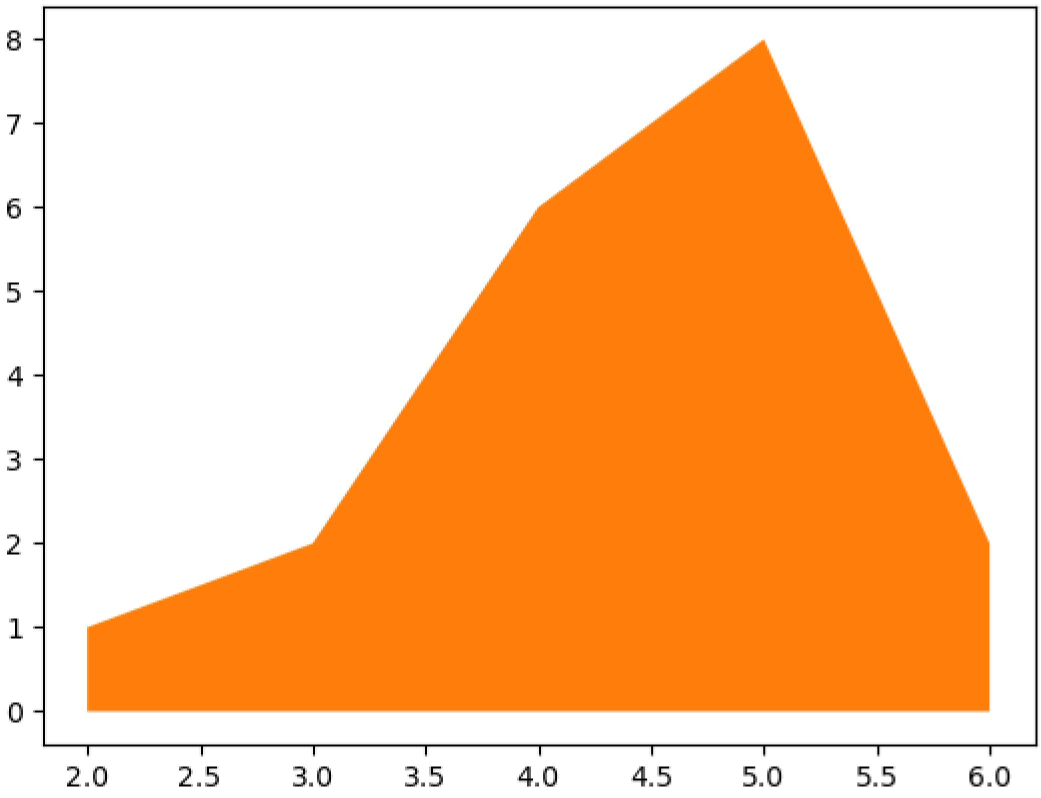
```
plt.figure(figsize=(5,5))

x10 = [35, 25, 20, 20]

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

plt.pie(x10, labels=labels);

plt.show()
```



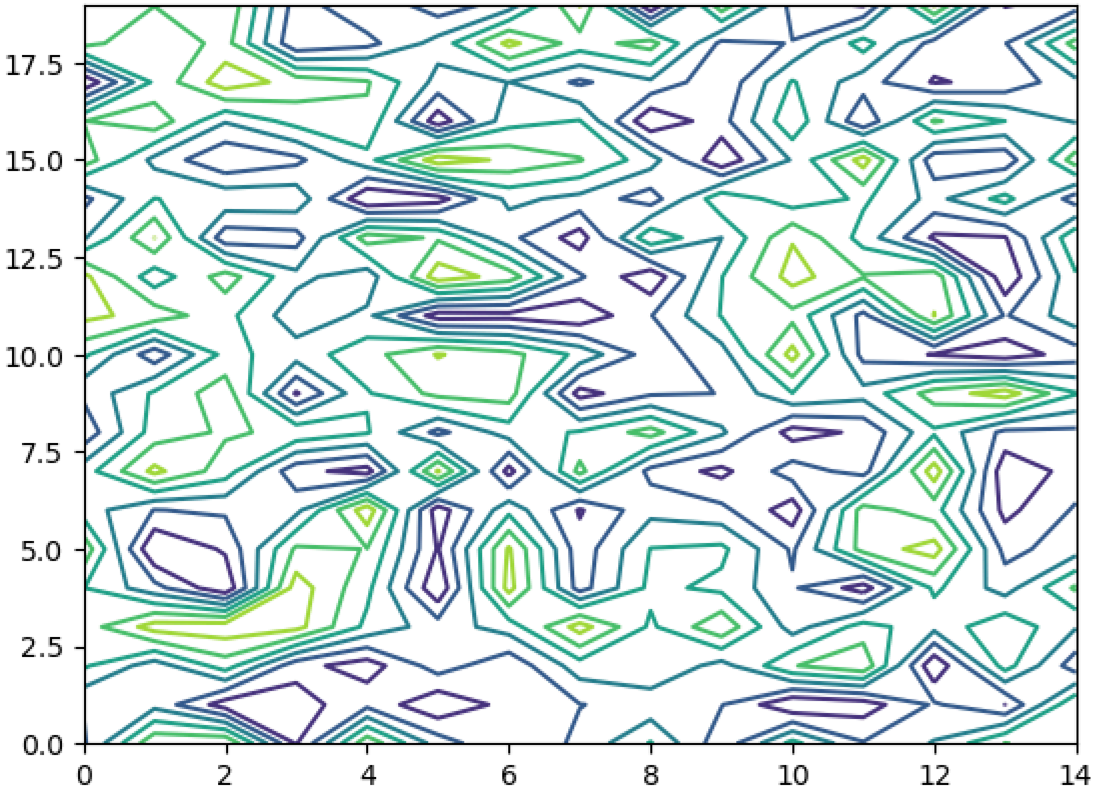
In [60]:
```
#box plot
data3 = np.random.randn(100)

plt.boxplot(data3)

plt.show();
```



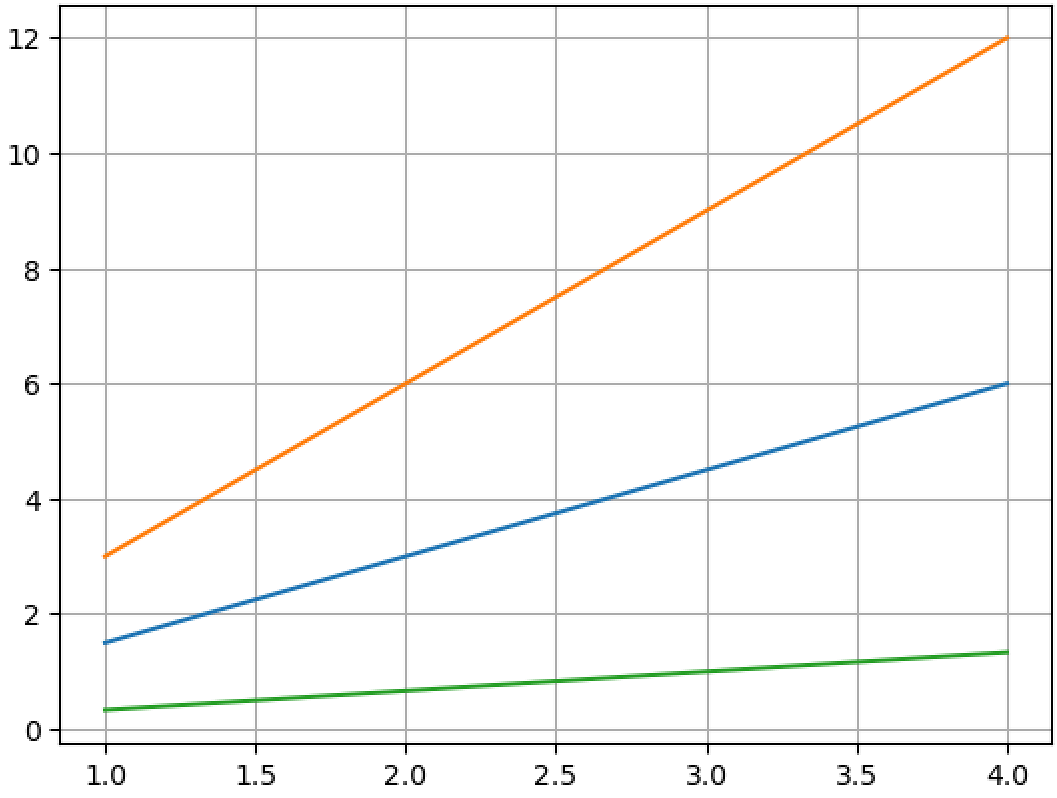
In [62]:
```
#area chart
x12=range(2,7)
y12=[1,2,6,8,2]

#area plot
plt.fill_between(x12,y12)
plt.show()
```



In [72]:
```
# Create a matrix
matrix1 = np.random.rand(20, 15)

cp = plt.contour(matrix1)
# Set styles for plots


plt.show()
```

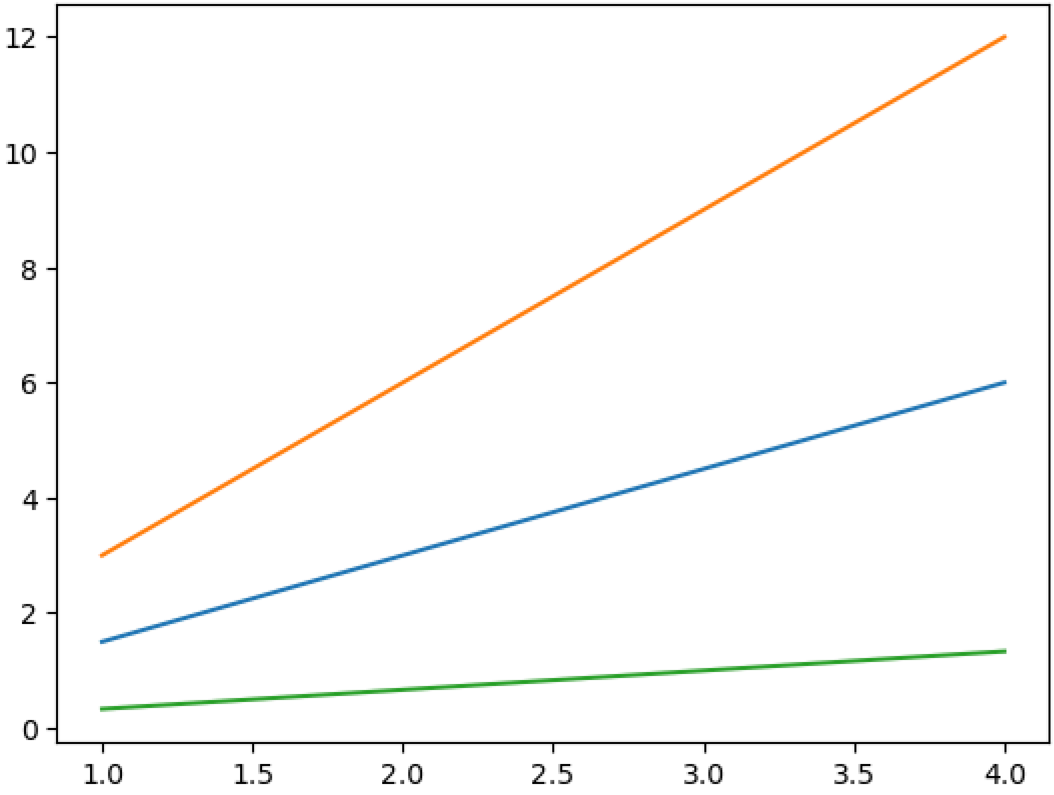


In [73]:
```
x15 = np.arange(1, 5)

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

plt.grid(True)

plt.show()
```

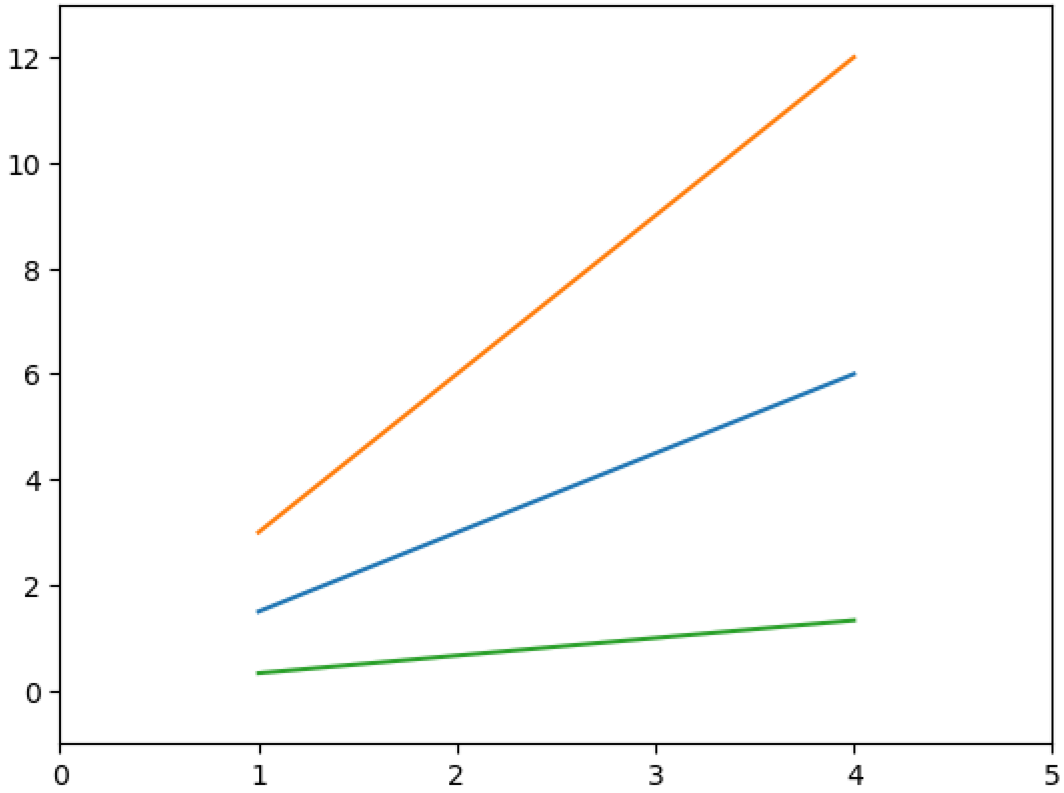


In [74]:

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

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

plt.grid(False)

plt.show()
```

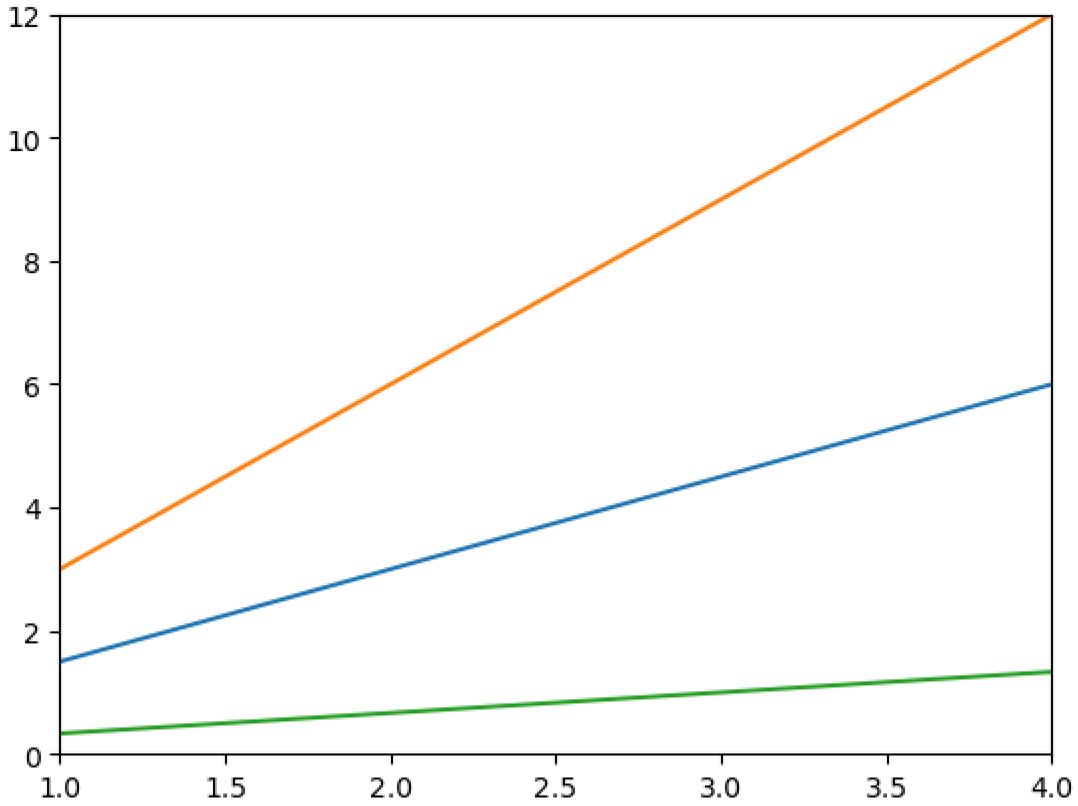


```
In [75]: x15 = np.arange(1, 5)

         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()
```

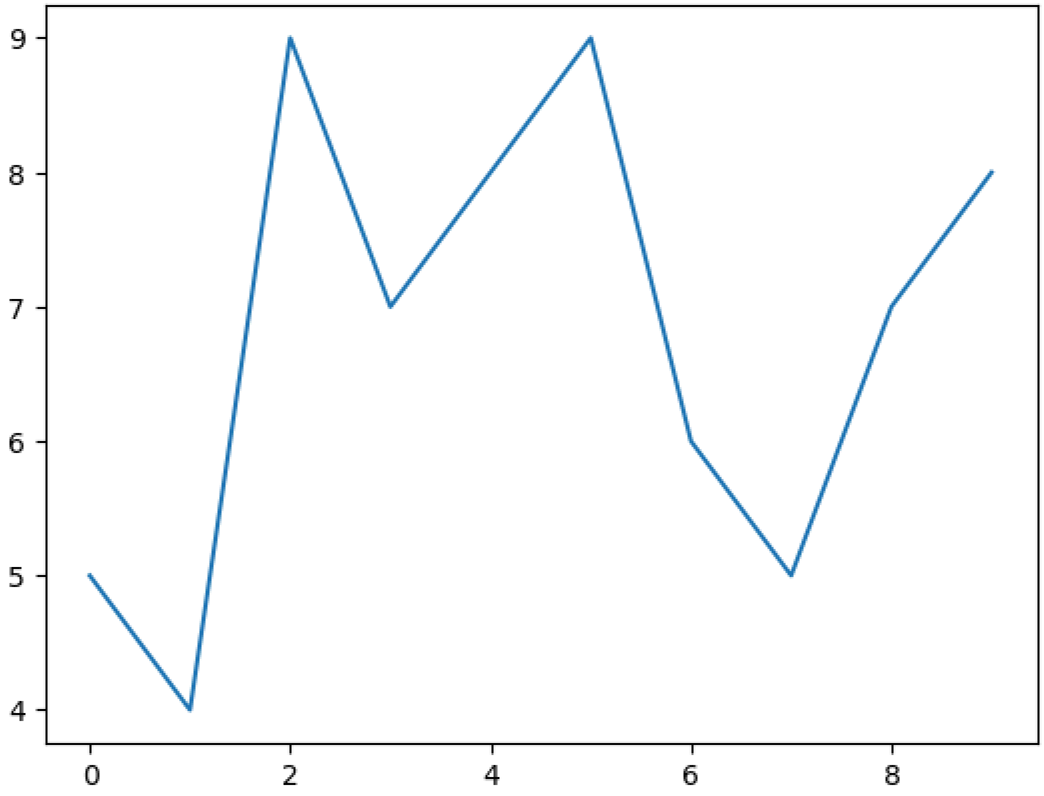


In [77]:

```
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.xlim([1.0, 4.0])

plt.ylim([0.0, 12.0])

plt.show()
```

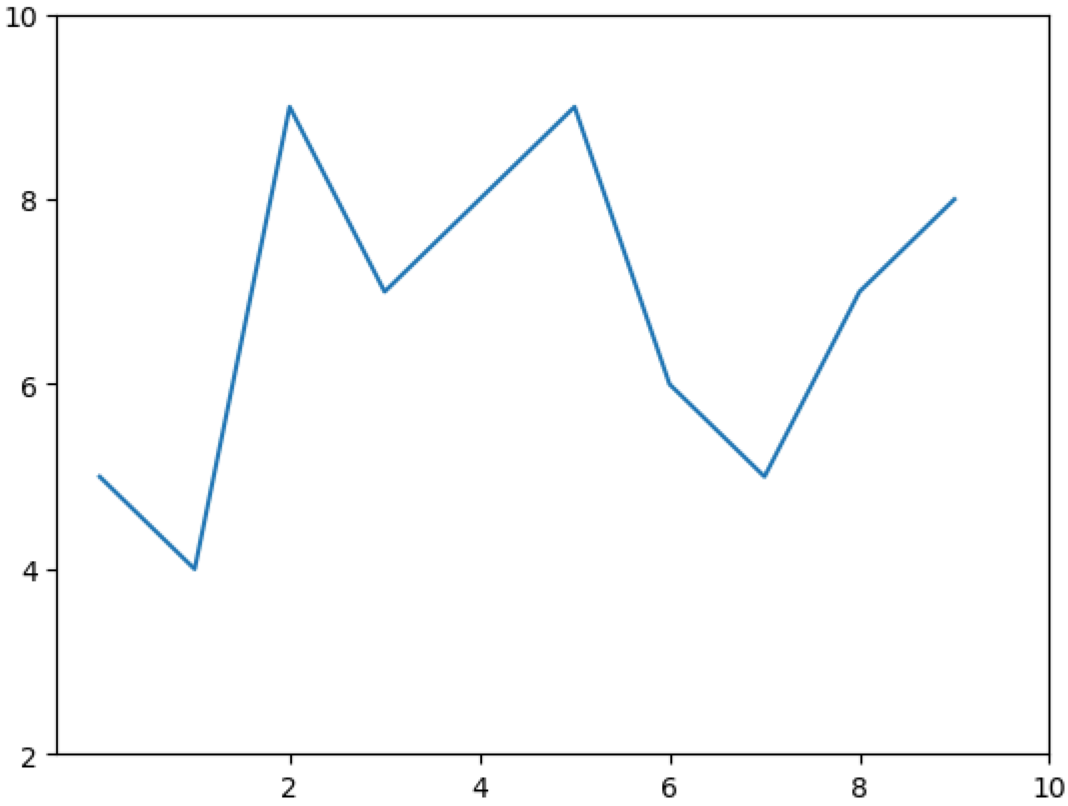


In [78]:
```
u = [5, 4, 9, 7, 8, 9, 6, 5, 7, 8]

plt.plot(u)



plt.show()
```



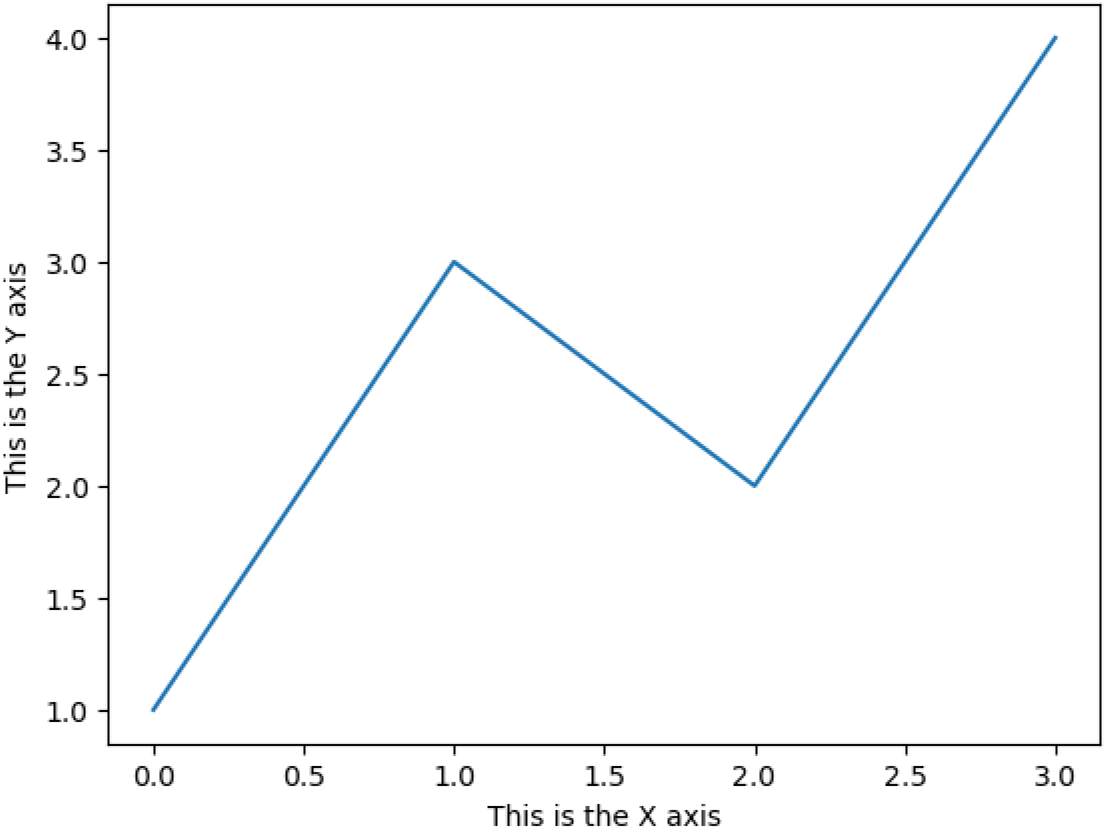
In [79]:

```
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()
```



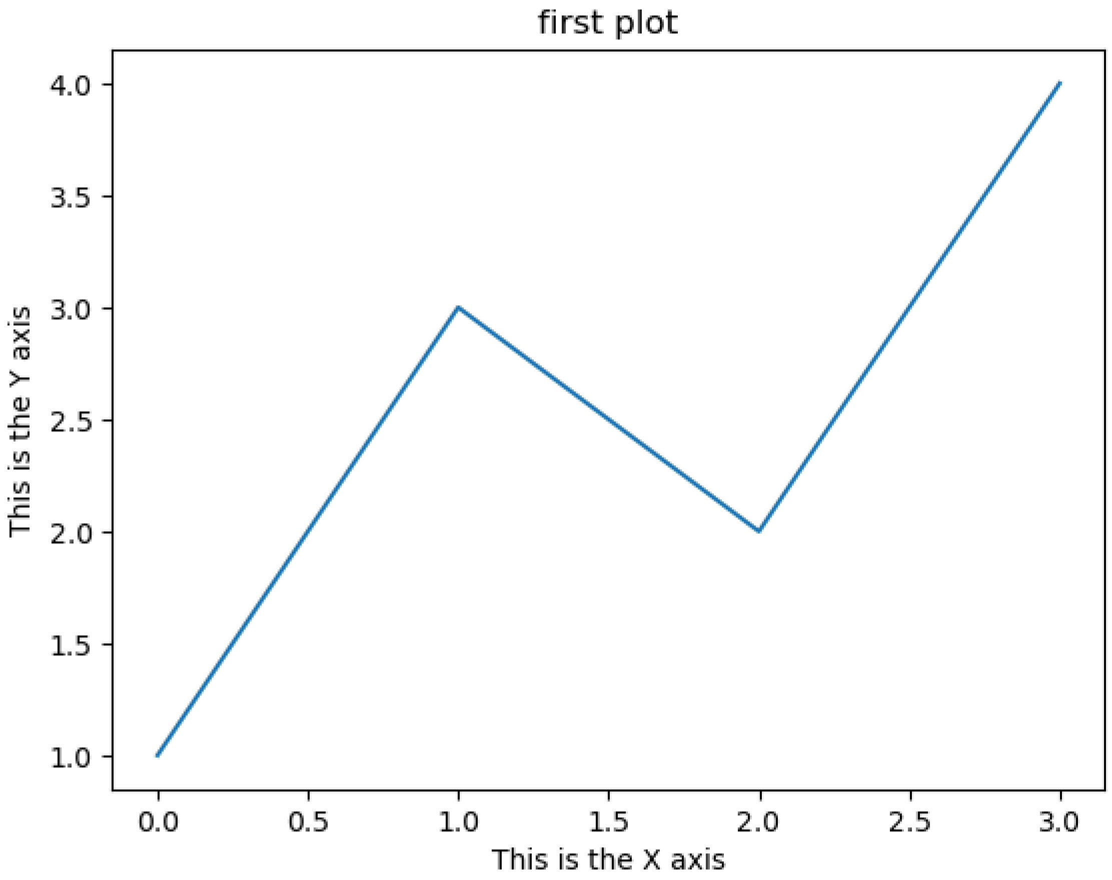
In [81]:

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

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

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

plt.show()
```



In [82]:

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

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

plt.ylabel('This is the Y axis')
plt.title('first plot')

plt.show()
```

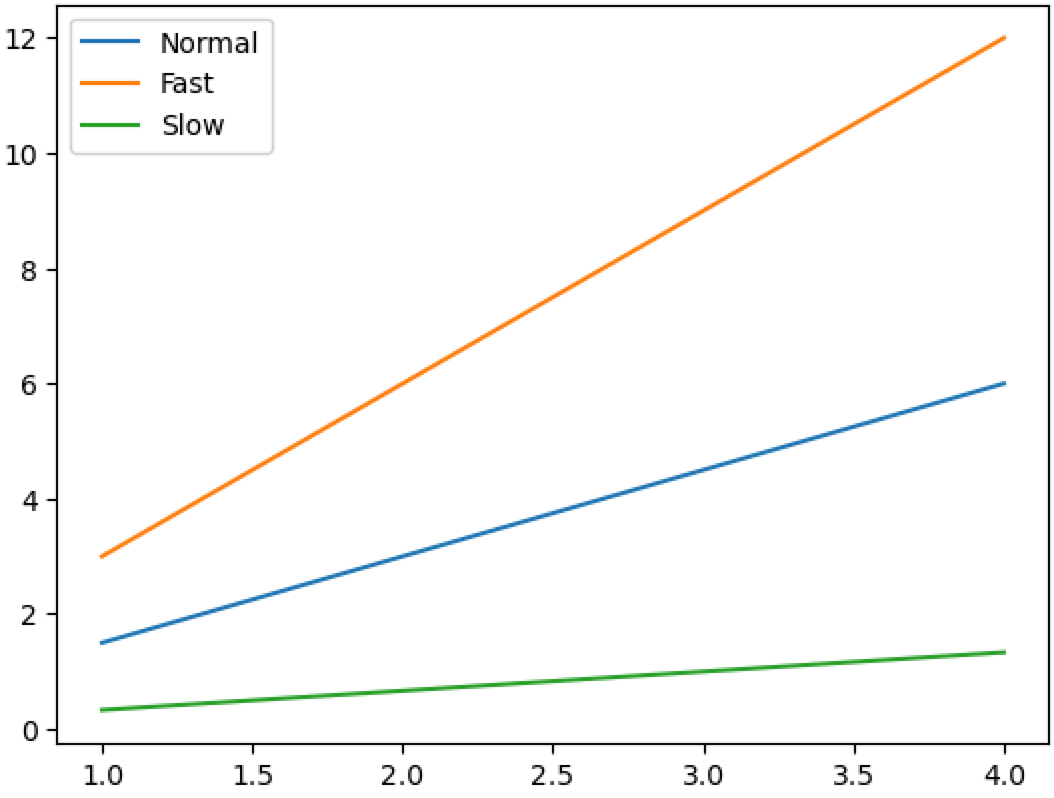


In [83]:

```
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']);
```

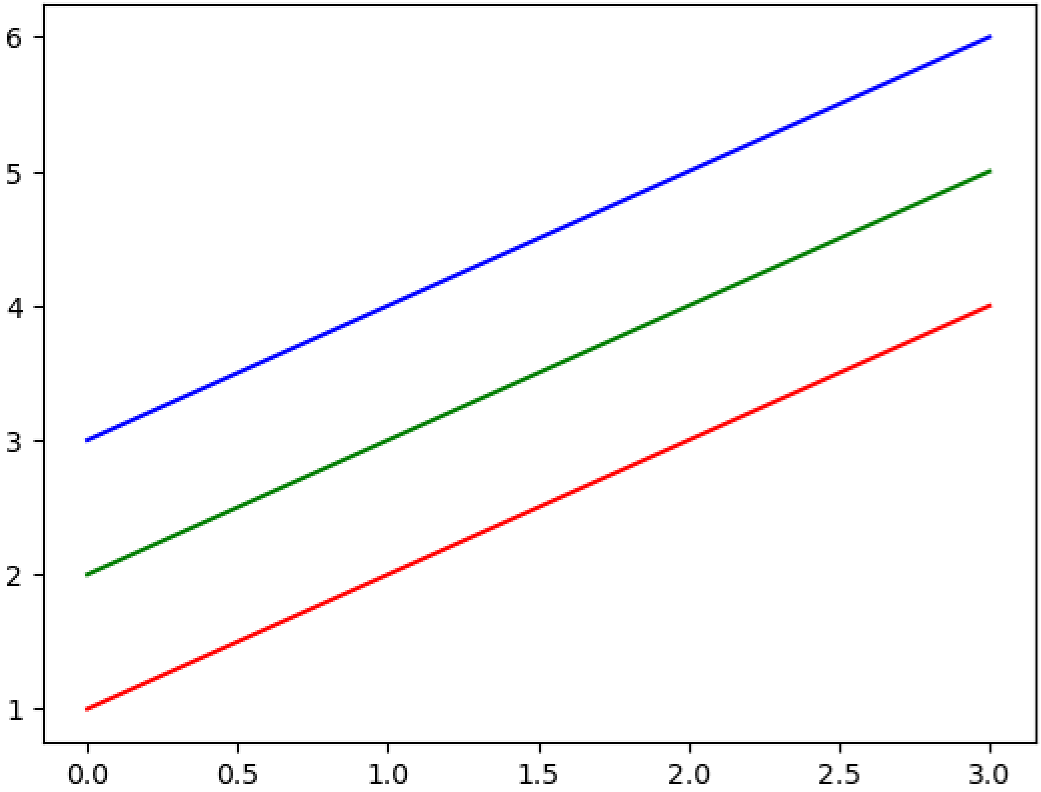
In [85]:

```
plt.show()
```



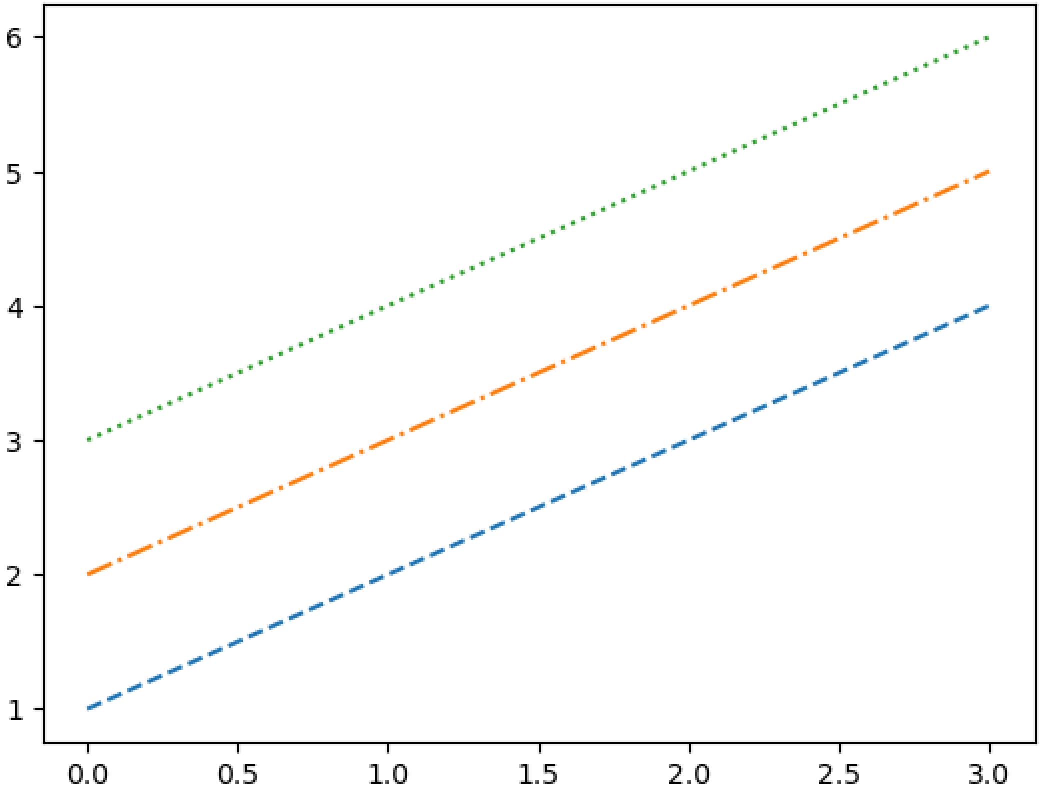
In [86]:

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

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

plt.show()
```



In [87]:

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

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

plt.show()
```



In [ ]: