In [1]:

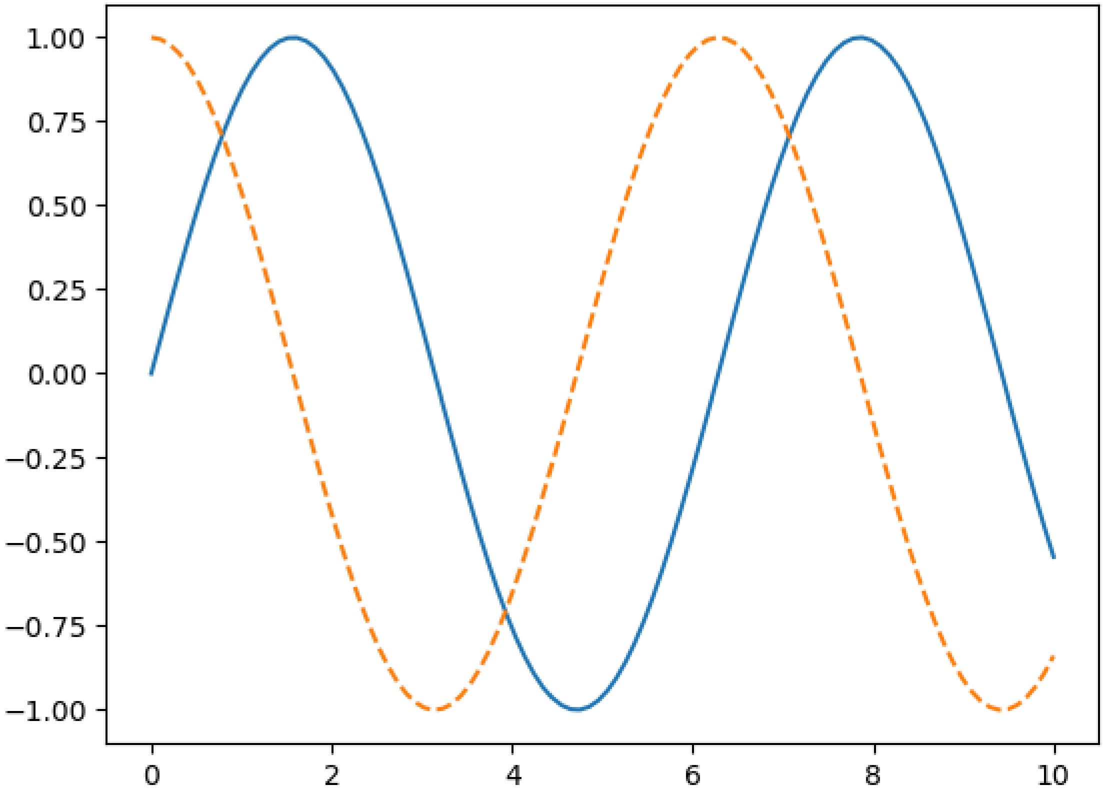
```
# Import dependencies

import numpy as np
import pandas as pd
```

In [2]:

```
# Import matplotlib

import matplotlib.pyplot as plt
```

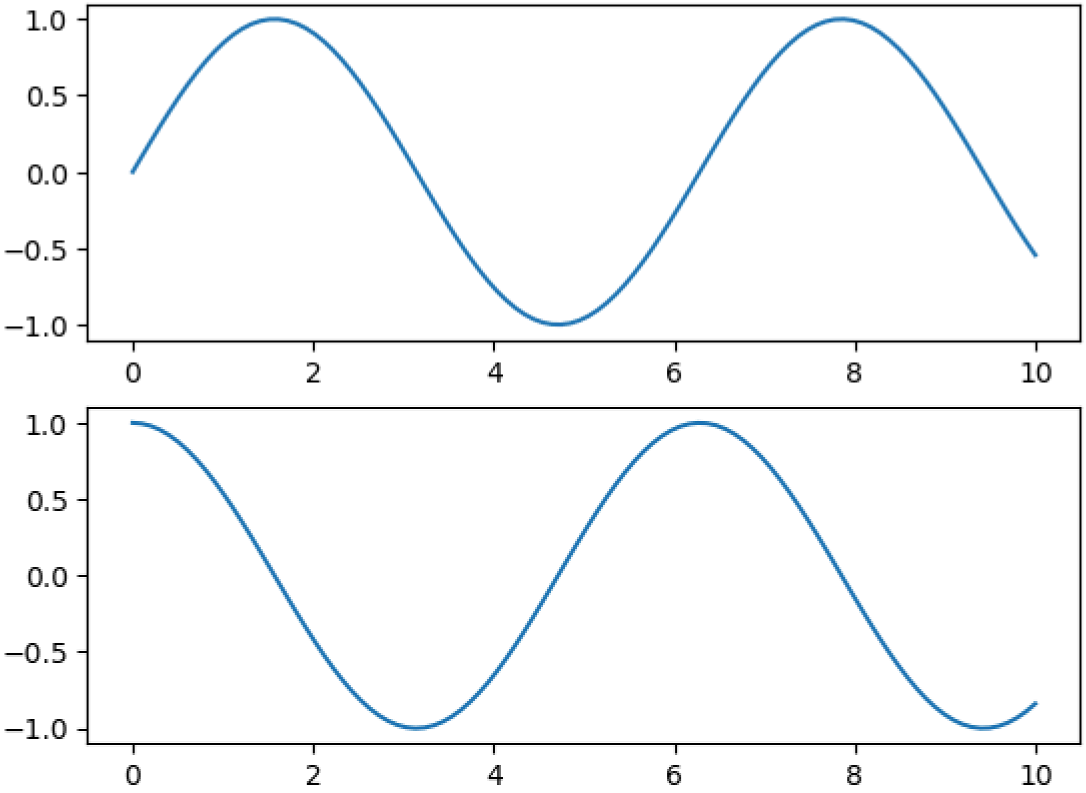
In [5]:

```
%matplotlib inline

x1 = np.linspace(0, 10, 100)

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

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



In [7]:

```
# creat a plot figure
plt.figure()

# creat the first of two panels and set current axis
plt.subplot(2, 1, 1) # (rows, columns, panel number)
plt.plot(x1, np.sin(x1))


# creat the second of two panels and set current axis
plt.subplot(2, 1, 2) # (rows, columns, panel number)
plt.plot(x1, np.cos(x1));
```

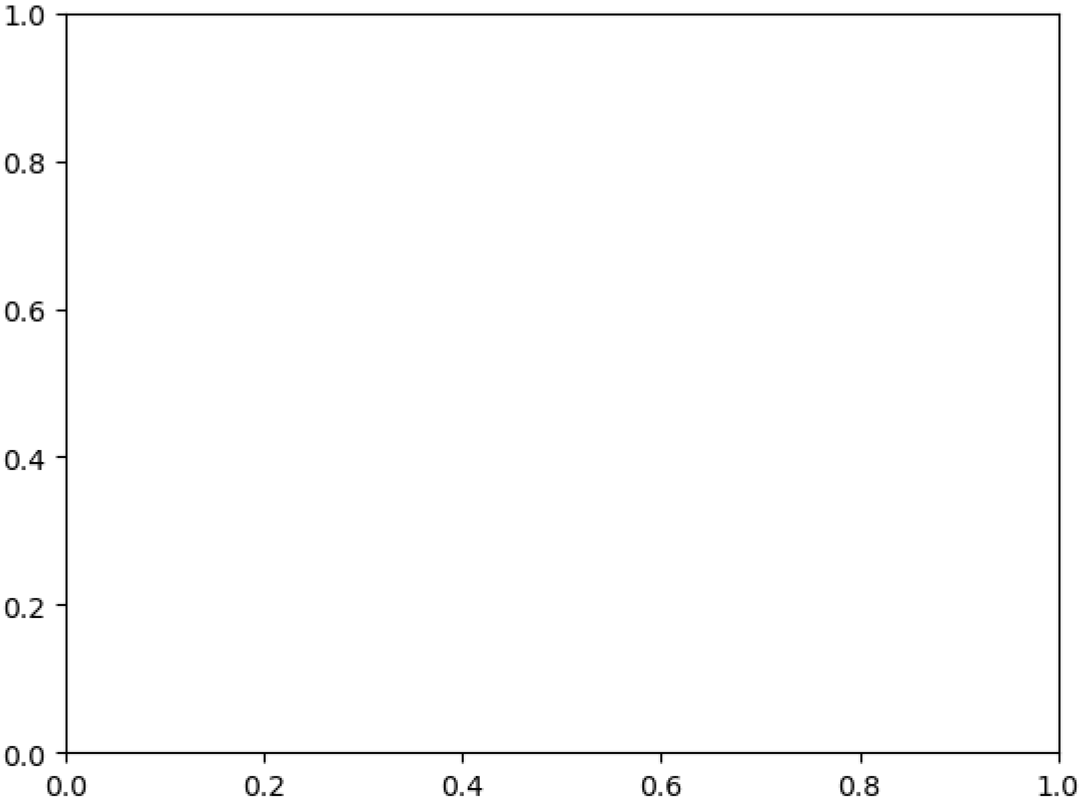


```
In [9]: # get current figure information

        print(plt.gcf())
```

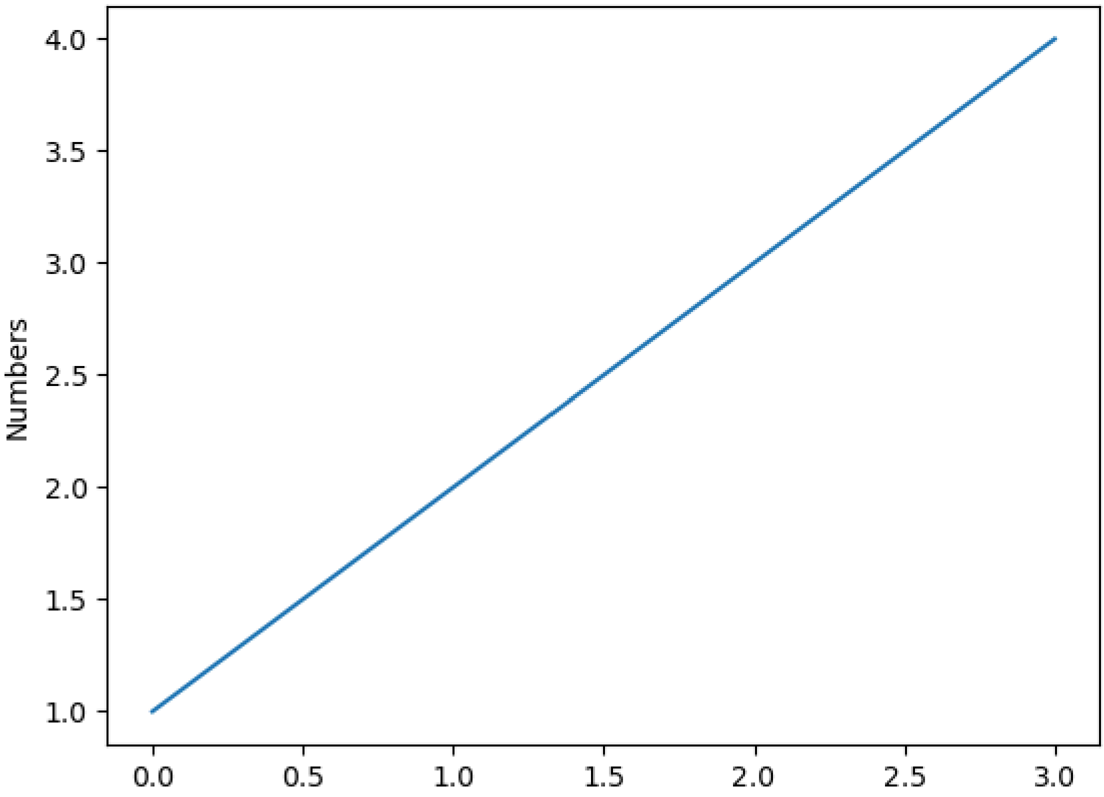
```
Figure(640x480)
<Figure size 640x480 with 0 Axes>
```

```
In [11]: # get current axis information

         print(plt.gca())
```

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

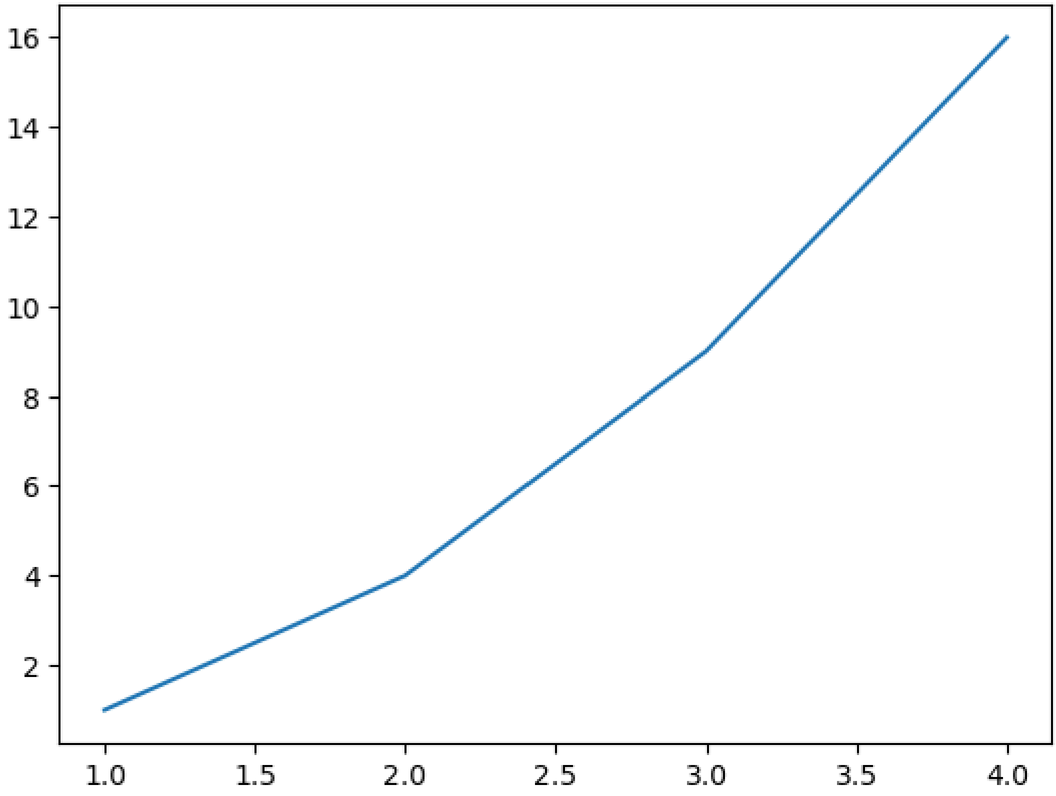


```
In [13]: plt.plot([1, 2, 3, 4])
         plt.ylabel('Numbers')
         plt.show()
```



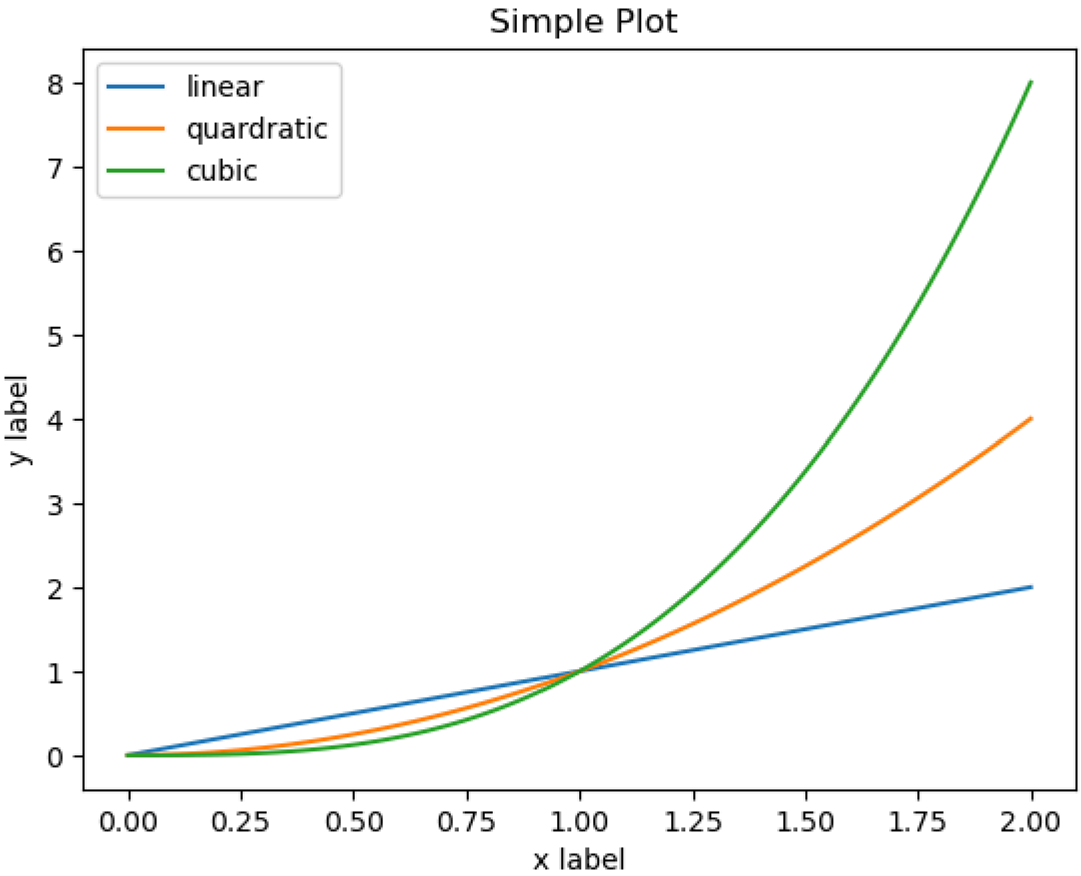
```
In [15]: plt.plot([1, 2, 3, 4], [1, 4, 9,16])
         plt.show
```

```
Out[15]: <function matplotlib.pyplot.show(close=None, block=None)>
```

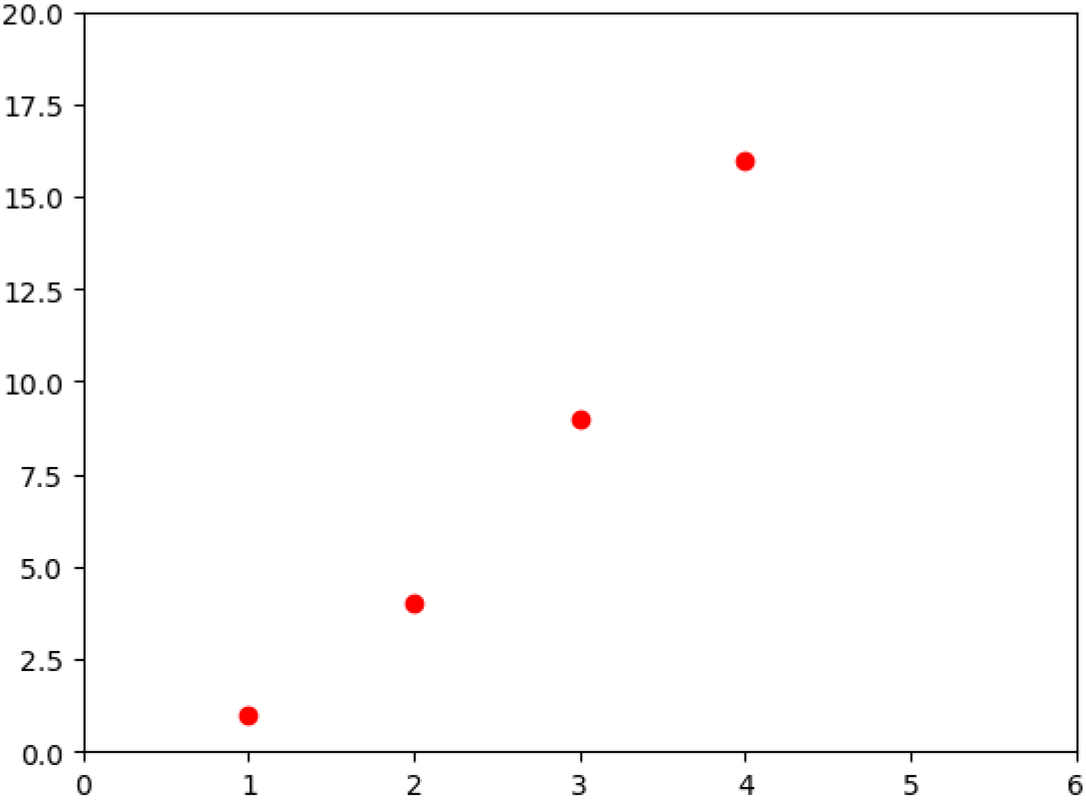


In [17]:

```
x = np.linspace(0, 2, 100)

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

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

plt.title(" Simple Plot")

plt.legend()

plt.show()
```

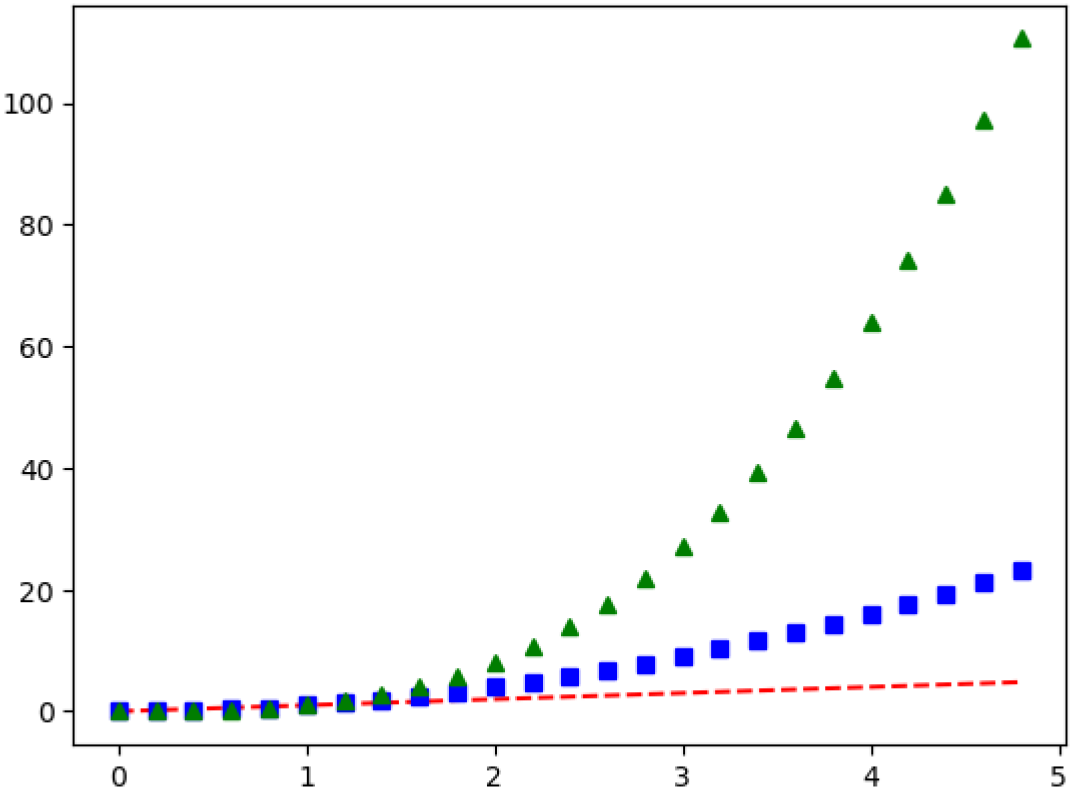


In [19]:
```
plt.plot([1, 2, 3, 4], [1,4,9,16],'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```



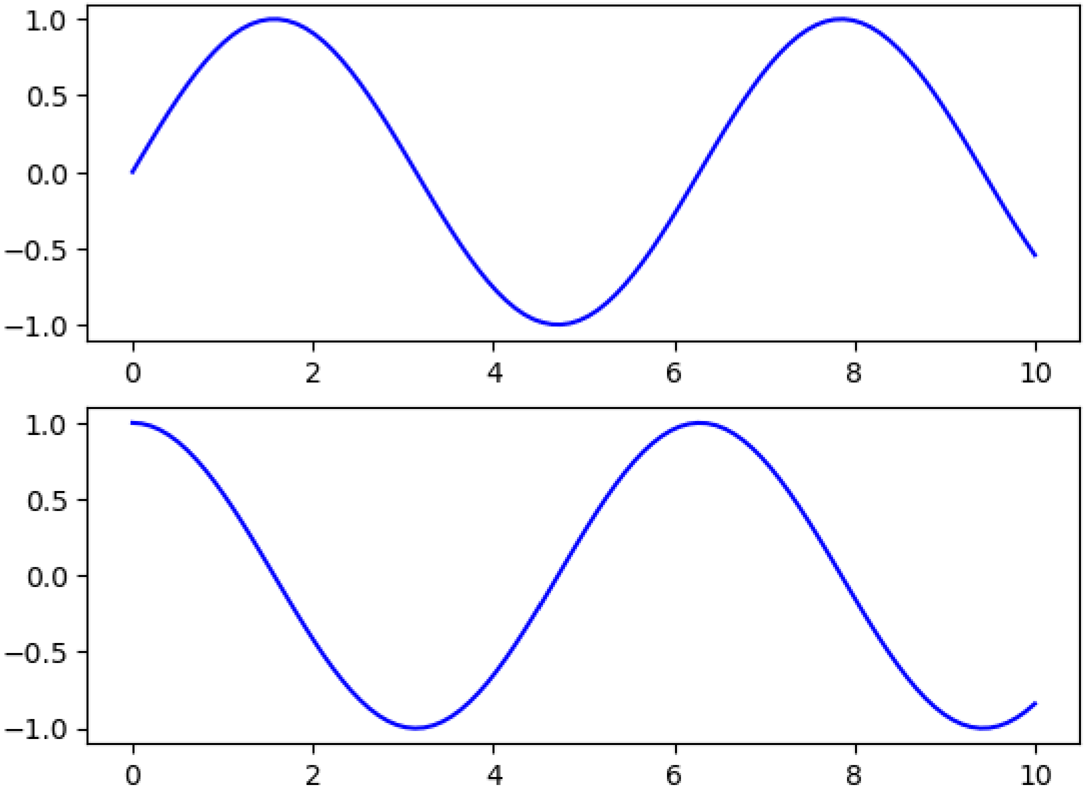
In [21]:
```
# evenly sample time at 200ms intervals
t = np.arange(0., 5., 0.2)
```

```
# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```



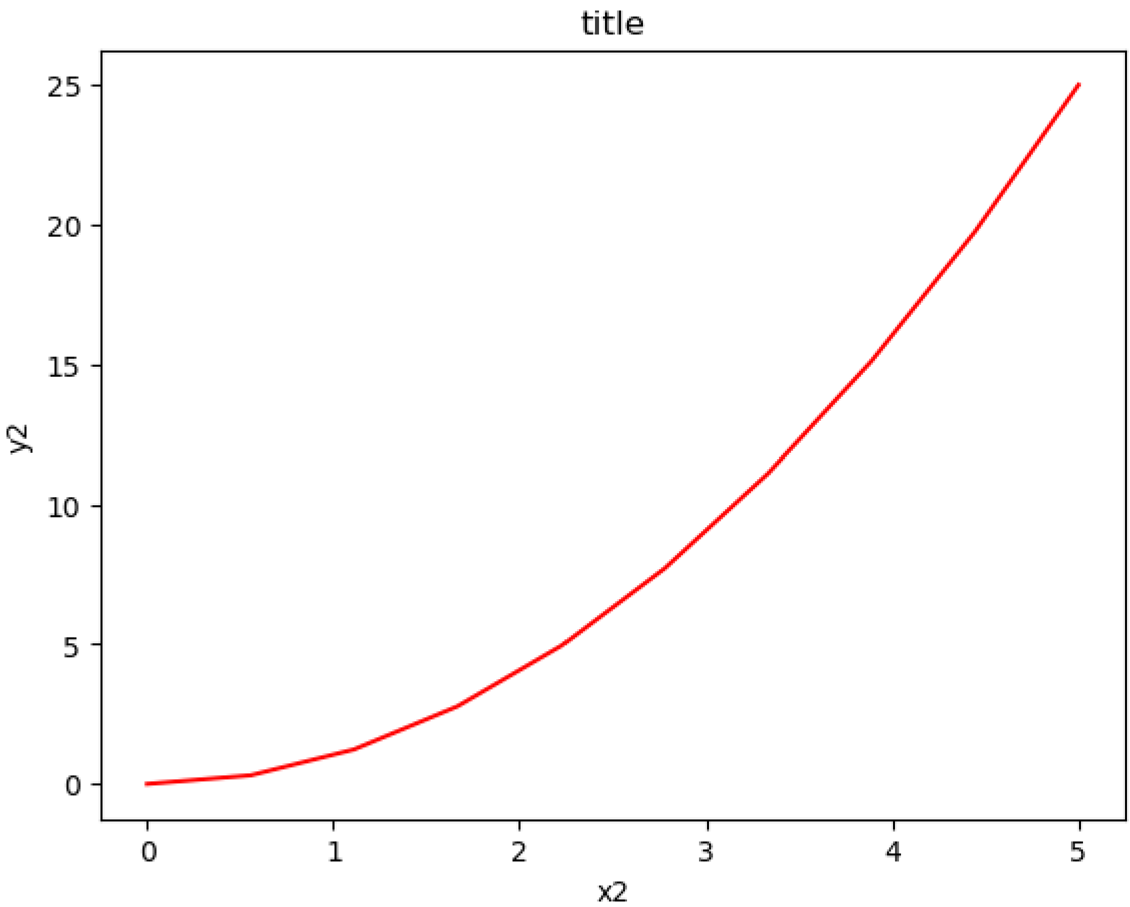
In [23]:

```
# first creat a grid of plots
# ax will be an array of two axes objects
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), 'b-');
```



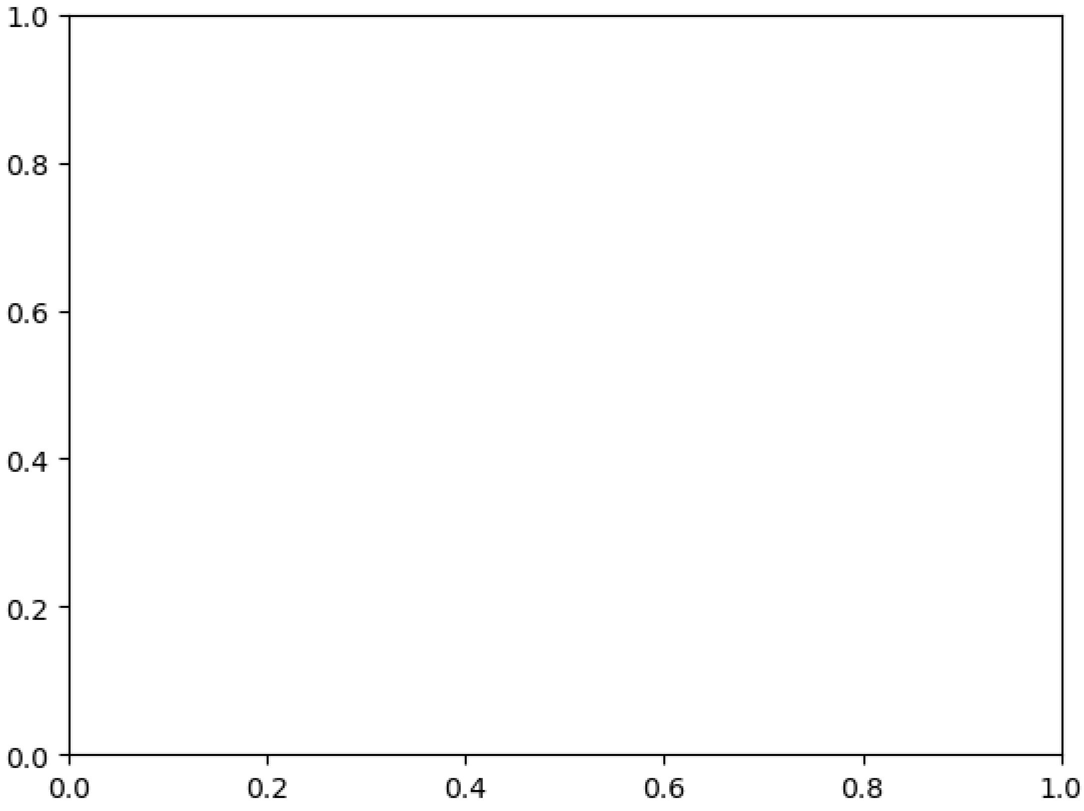
In [25]:

```
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 [27]:

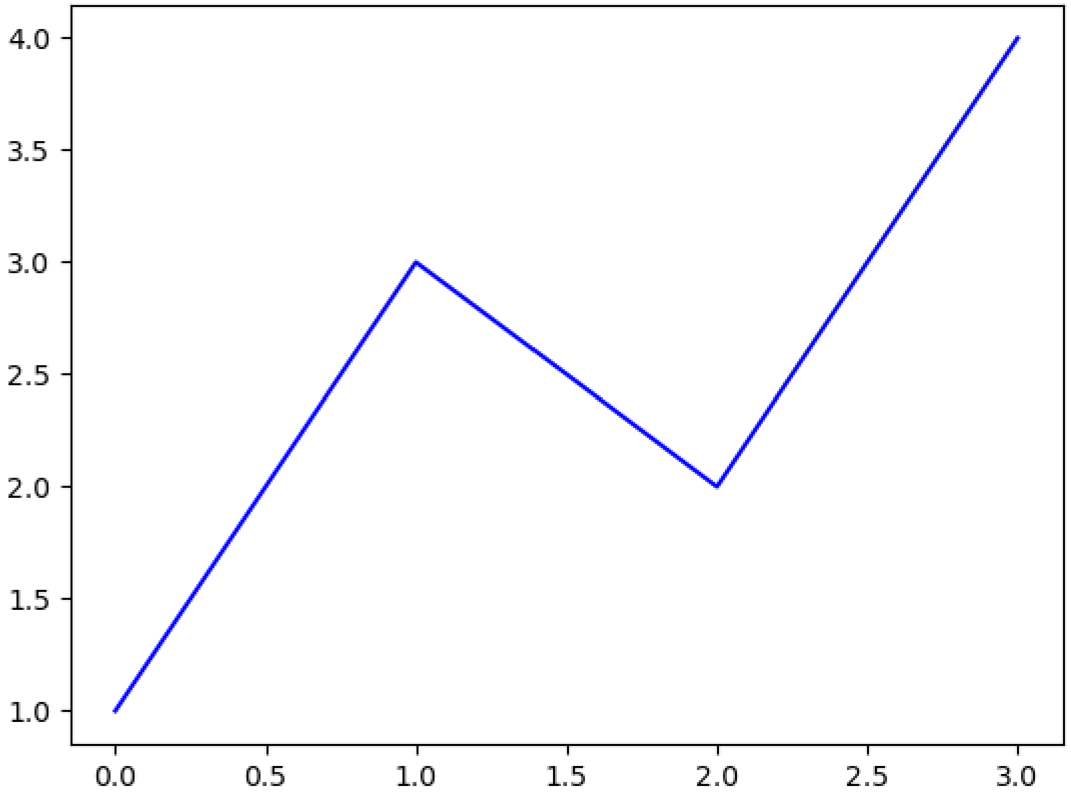
```
fig = plt.figure()

ax = plt.axes()
```



First plot with matplotlib

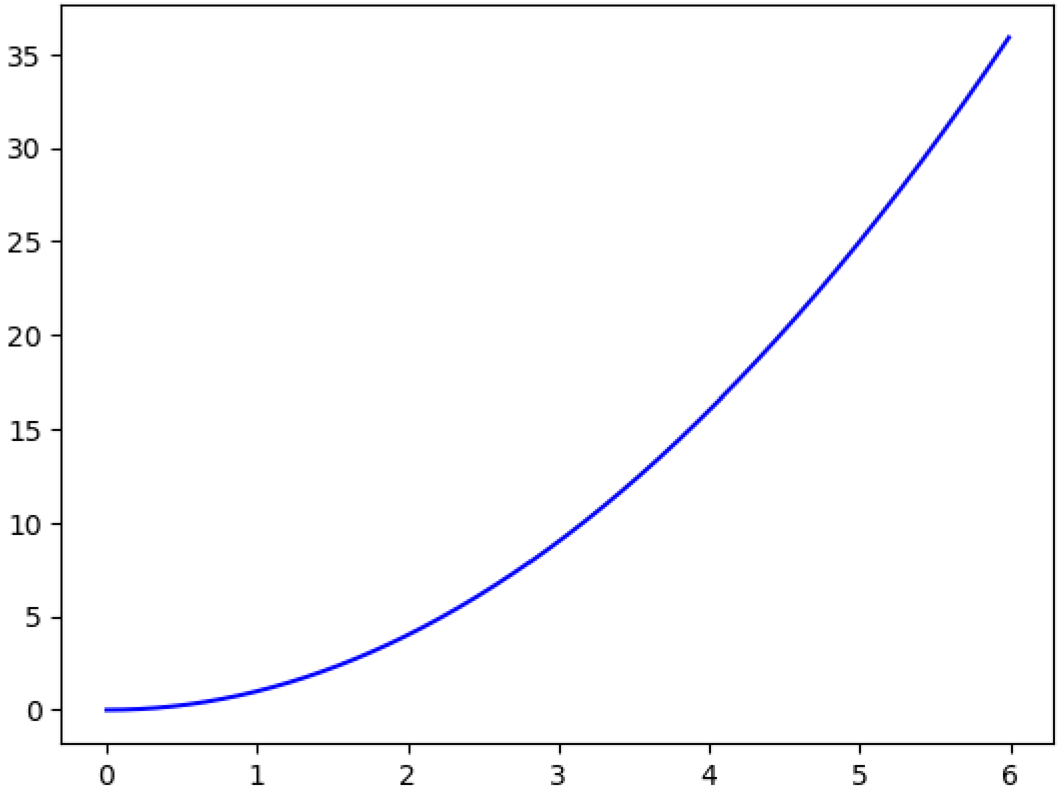
In [30]:

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

plt.show( )
```



In [32]:

```
x3 = np.arange (0.0, 6.0, 0.01)

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

plt.show()
```

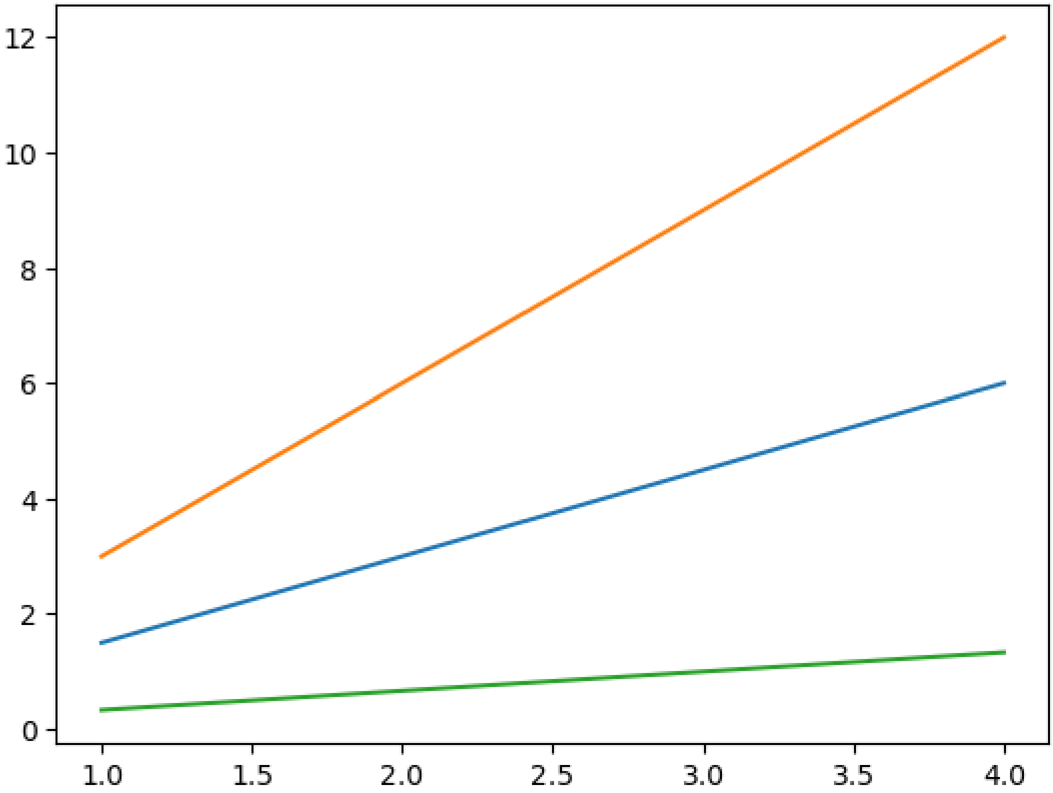


Multiline plots

_ multiline plots mean plotting more than one plot on the same figure.we can plot more than one plot on the same figure.

- it can be achieved by plotting all the lines before calling show(). it can be done as follows:-

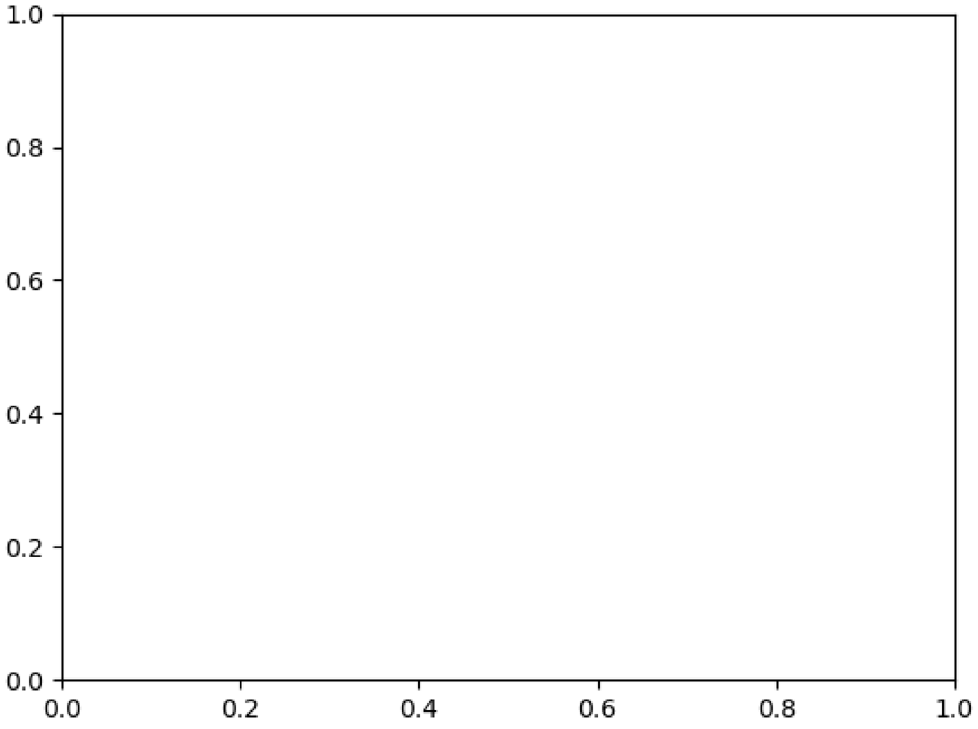
In [36]:
```
x4 = range(1, 5)

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 [38]:
```
# saving the figure

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

In [40]:
```
# Explore the contents of figure

from IPython.display import Image

Image('plot1.png')
```

Out[40]:



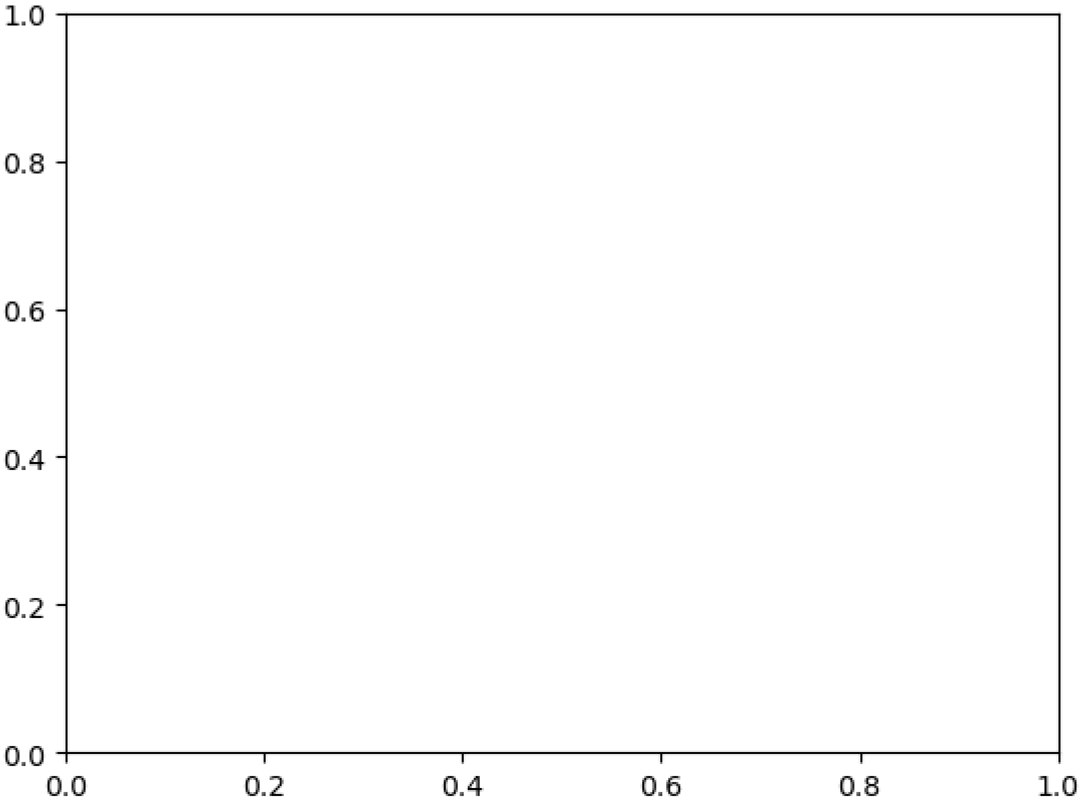
Line plot

- we can use the following commands to draw the simple sinusoid line plot:-

In [47]:

```
# Create figure and axes first
fig = plt.figure()

ax = plt.axes()

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

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[47], line 7
      4 ax = plt.axes()
      6 # plot the sinusoid function
----> 7 ax.plot(x5, np.sin(x5), 'b-')

NameError: name 'x5' is not defined
```



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```