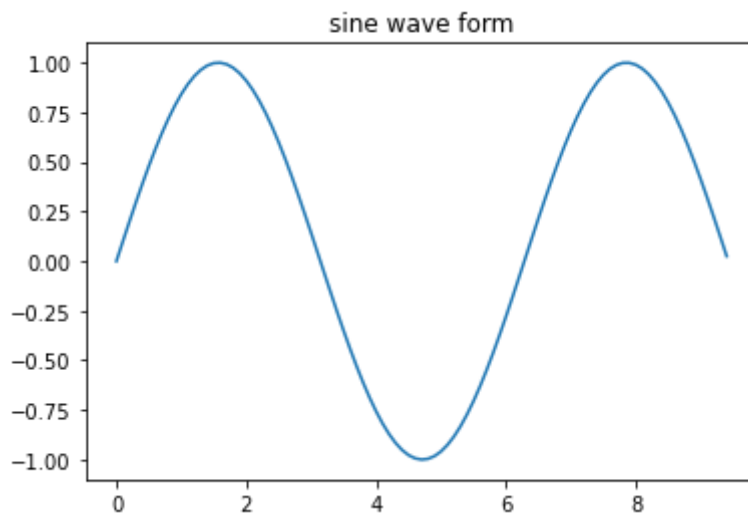


```

import numpy as np
import matplotlib.pyplot as plt
# Compute the x and y coordinates for points on a sine curve
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)
plt.title("sine wave form")
# Plot the points using matplotlib
plt.plot(x, y)
plt.show()

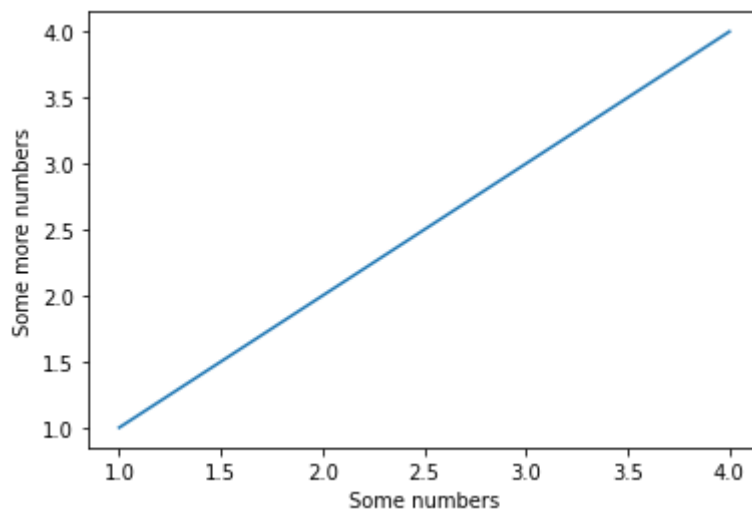
```



```

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# Plotting a Line
plt.plot([1,2,3,4], [1,2,3,4])
plt.xlabel('Some numbers')
plt.ylabel('Some more numbers')
plt.show()

```

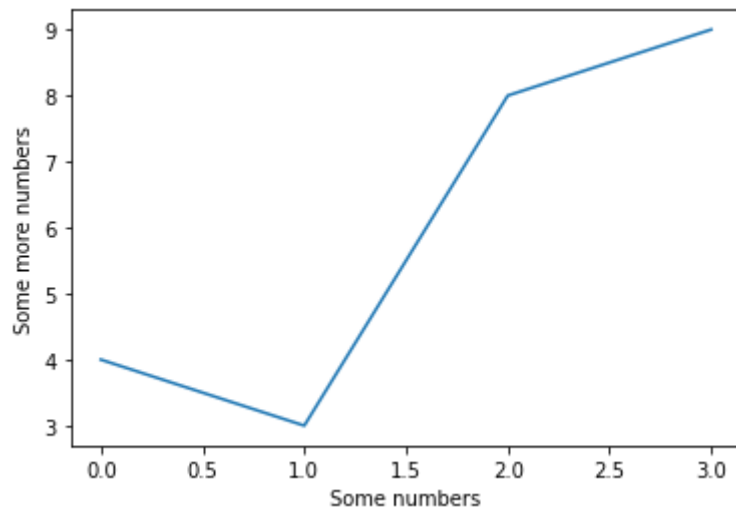


```

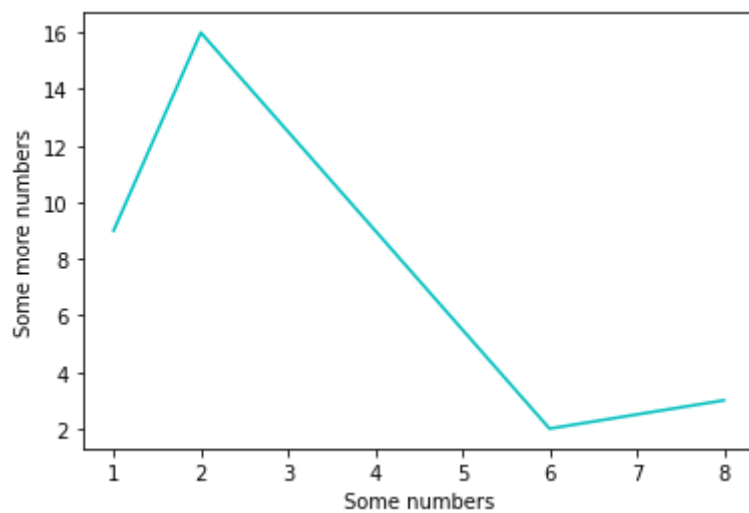
#Plotting using X axis only
plt.plot([4,3,8,9])
plt.xlabel('Some numbers')

```

```
plt.ylabel('Some more numbers')
plt.show()
```



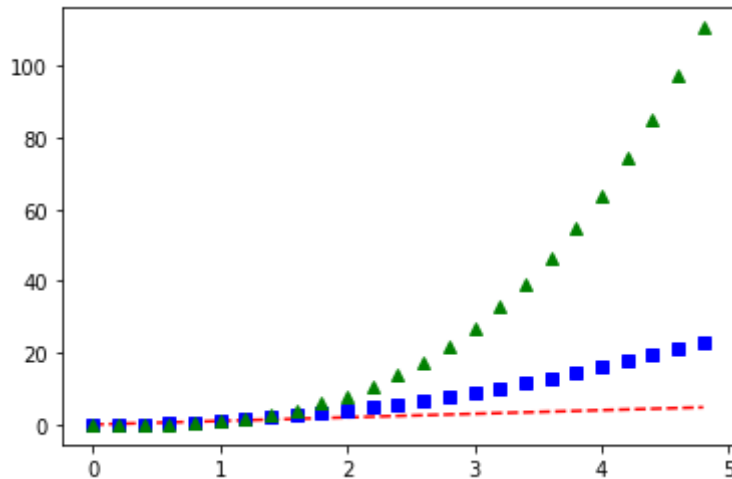
```
#Plotting with Tweaking Colors and Symbols
plt.plot([1,2,6,8], [9,16,2,3], 'c')
plt.xlabel('Some numbers')
plt.ylabel('Some more numbers')
plt.show()
```



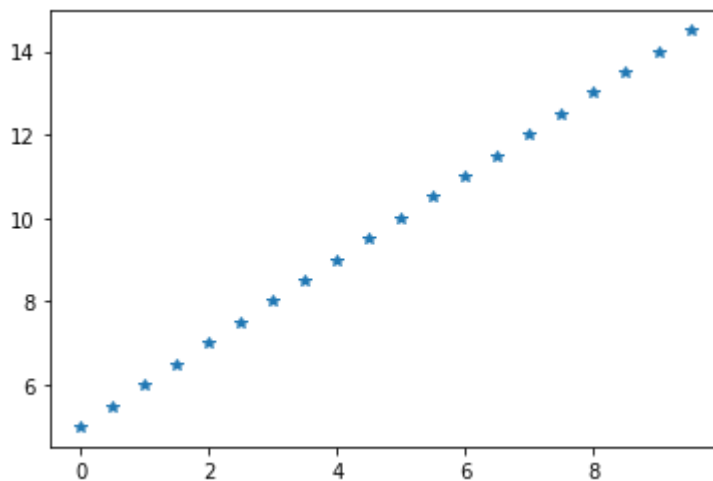
```
plt.plot([1,2,6,8], [9,16,2,3], 'm+')
plt.xlabel('Some numbers')
plt.ylabel('Some more numbers')
plt.show()
```



```
#Red dashes, blue squares and green triangles
t = np.arange(0., 5., 0.2)
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```

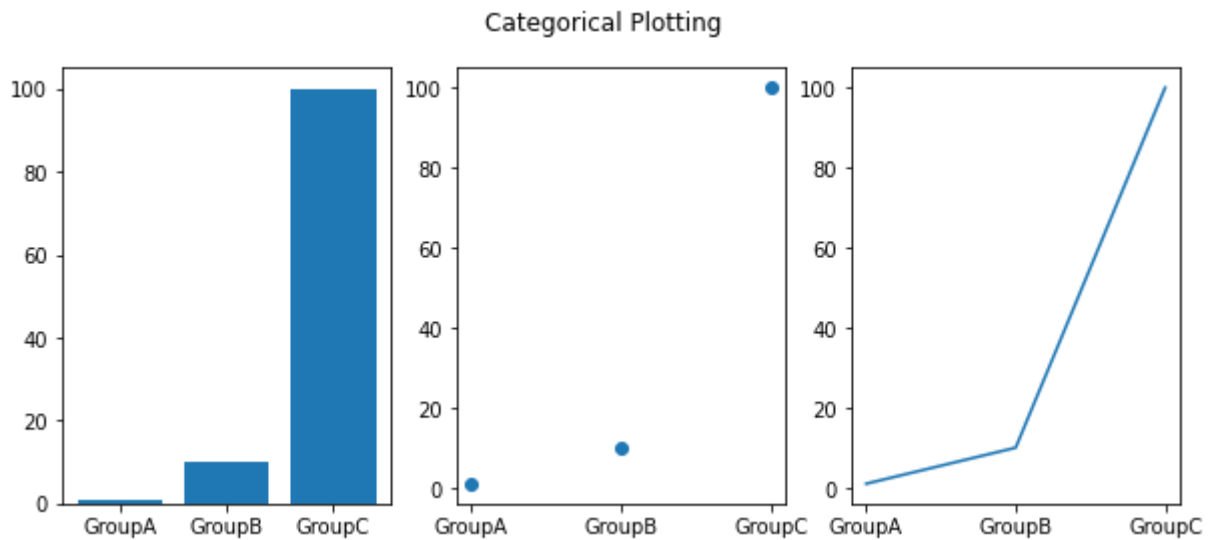


```
#By using Formula
data = np.arange(0, 10, 0.5)
y = 1 * data + 5
plt.plot(data, y, '*')
plt.show()
```

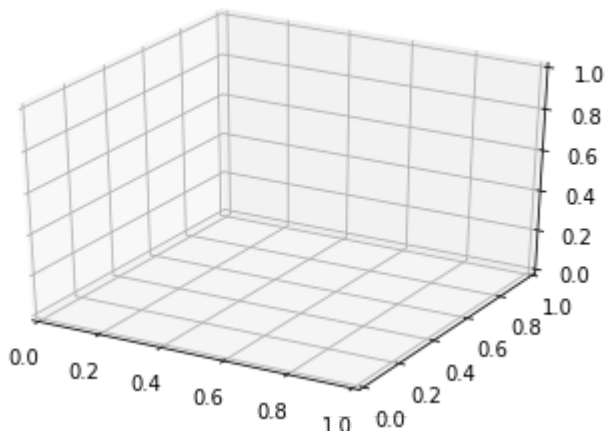


```
#Plotting with Categorical Data
names = ['GroupA', 'GroupB', 'GroupC']
values = [1, 10, 100]
plt.figure(1, figsize=(10,4))
plt.subplot(131)
plt.bar(names, values)
plt.subplot(132)
plt.scatter(names, values)
plt.subplot(133)
plt.plot(names, values)
```

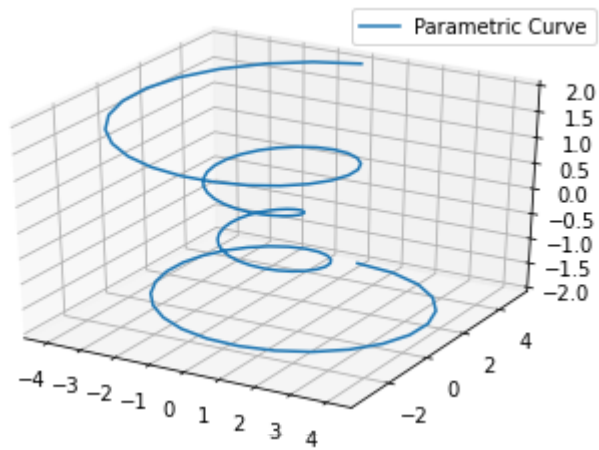
```
plt.suptitle('Categorical Plotting')
plt.show()
```



```
#Plotting over 3D Axes
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
mpl.rcParams['legend.fontsize'] = 10
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
```

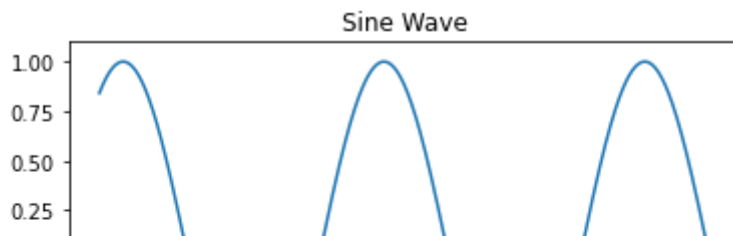


```
#Parametric Curve
mpl.rcParams['legend.fontsize'] = 10
fig = plt.figure()
ax = fig.gca(projection='3d')
theta = np.linspace(-4 * np.pi, 4 * np.pi, 100)
z = np.linspace(-2, 2, 100)
r = z**2 + 1
x = r * np.sin(theta)
y = r * np.cos(theta)
ax.plot(x, y, z, label='Parametric Curve')
ax.legend()
plt.show()
```



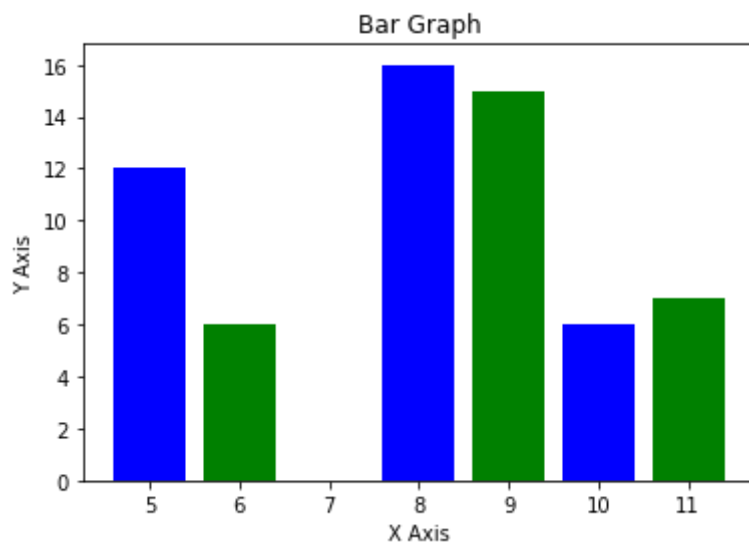
```
#Subplots of Sine and Cosine
x = np.arange(1, 5 * np.pi, 0.01)
y = np.sin(x)
plt.title('Sine Wave')
plt.plot(x, y)
plt.show()
x = np.arange(0, 3* np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)
plt.subplot(1, 2, 1)
plt.plot(x, y_sin)

plt.title('Sine Wave')
plt.subplot(1, 2, 2)
plt.plot(x, y_cos)
plt.title('Cosine Wave')
plt.suptitle('Waveforms')
plt.show()
```



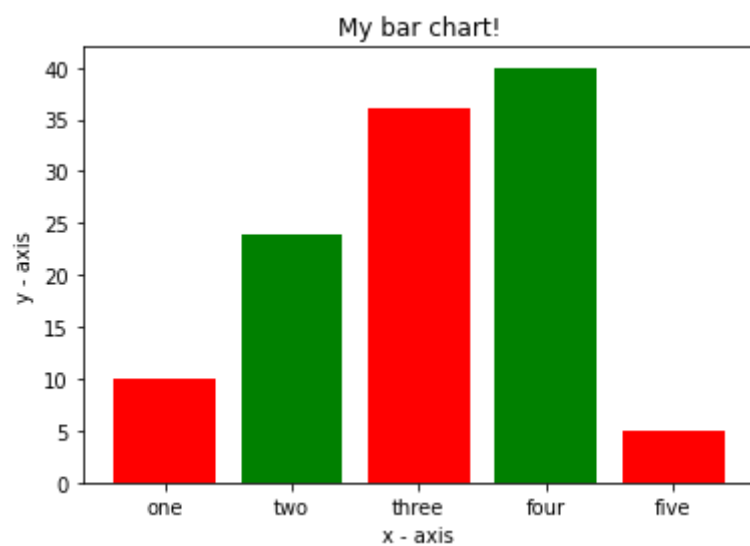
```
x1 = [5, 8, 10]
y1 = [12, 16, 6]
x2 = [6, 9, 11]
y2 = [6, 15, 7]
plt.bar(x1, y1, color = 'b')
plt.bar(x2, y2, color = 'g', align='center')
plt.title('Bar Graph')

plt.ylabel('Y Axis')
plt.xlabel('X Axis')
plt.show()
```



```
import matplotlib.pyplot as plt
# x-coordinates of left sides of bars
left = [1, 2, 3, 4, 5]
# heights of bars
height = [10, 24, 36, 40, 5]
# labels for bars
tick_label = ['one', 'two', 'three', 'four', 'five']

# plotting a bar chart
plt.bar(left, height, tick_label = tick_label,
width = 0.8, color = ['red', 'green'])
# naming the x-axis
plt.xlabel('x - axis')
# naming the y-axis
plt.ylabel('y - axis')
# plot title
plt.title('My bar chart!')
# function to show the plot
plt.show()
```



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 4:03 PM

