Practical - 8

AIM

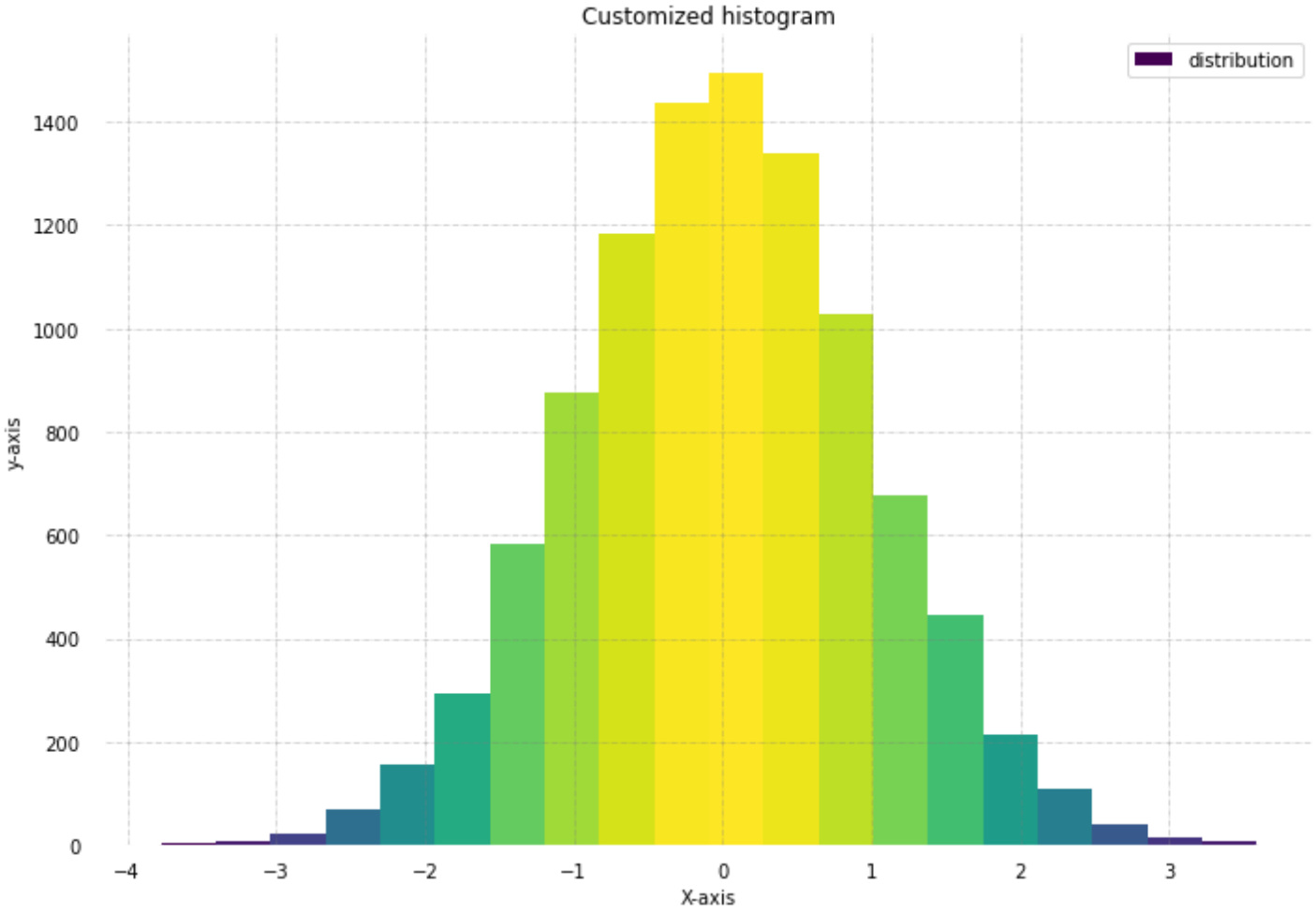
Create various types of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix. Further label different axes in a plot and data in a plot.

CODE & OUTPUT

In [3]:

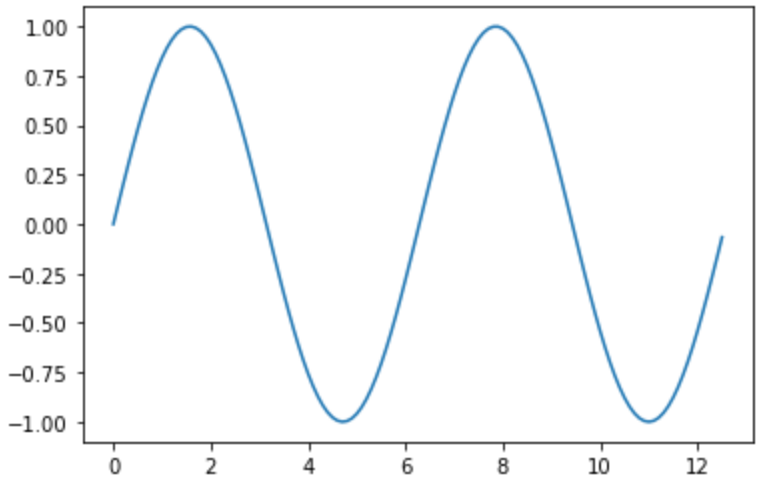
```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import colors
from matplotlib.ticker import PercentFormatter


# Creating dataset
np.random.seed(23685752)
N_points = 10000
n_bins = 20

# Creating distribution
x = np.random.randn(N_points)
y = .8 ** x + np.random.randn(10000) + 25
legend = ['distribution']

# Creating histogram
fig, axs = plt.subplots(1, 1,
                        figsize =(10, 7),
                        tight_layout = True)


# Remove axes splines
for s in ['top', 'bottom', 'left', 'right']:
    axs.spines[s].set_visible(False)

# Remove x, y ticks
axs.xaxis.set_ticks_position('none')
axs.yaxis.set_ticks_position('none')

# Add padding between axes and labels
axs.xaxis.set_tick_params(pad = 5)
axs.yaxis.set_tick_params(pad = 10)

# Add x, y gridlines
axs.grid(b = True, color ='grey',
         linestyle ='-.', linewidth = 0.5,
         alpha = 0.6)

# Creating histogram
N, bins, patches = axs.hist(x, bins = n_bins)

# Setting color
fracs = ((N**(1 / 5)) / N.max())
norm = colors.Normalize(fracs.min(), fracs.max())

for thisfrac, thispatch in zip(fracs, patches):
    color = plt.cm.viridis(norm(thisfrac))
    thispatch.set_facecolor(color)

# Adding extra features
plt.xlabel("X-axis")
```

```
plt.ylabel("y-axis")
plt.legend(legend)
plt.title('Customized histogram')

# Show plot
plt.show()
```



In [5]:

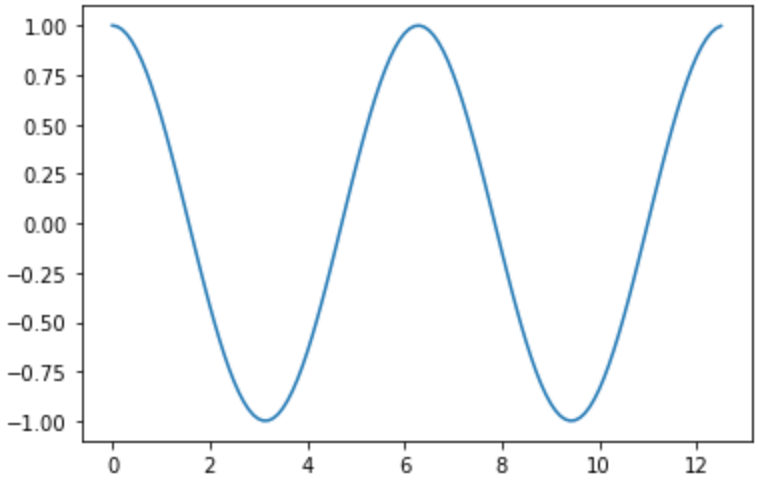
```
# creating a sine plot
x = np.arange(0, 4 * np.pi, 0.1)
y = np.sin(x)
plt.plot(x, y)
plt.show()
```



In [7]:

```
# creating a cosine plot
y1 = np.cos(x)
```

```
plt.plot(x, y1)
plt.show()
```

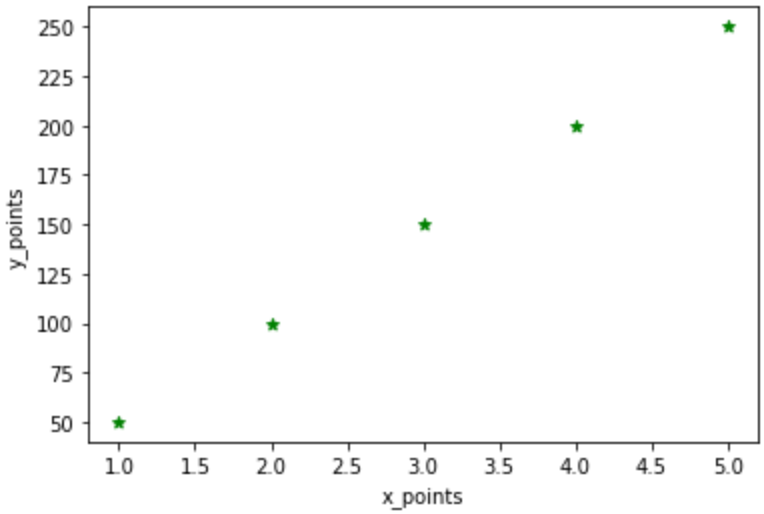


In [9]:

```
x_points = np.array([1, 2, 3, 4, 5])
y_points = np.array([50, 100, 150, 200, 250])
plt.xlabel('x_points')
plt.ylabel('y_points')
plt.scatter(x_points, y_points, color = "green", marker="*")
```

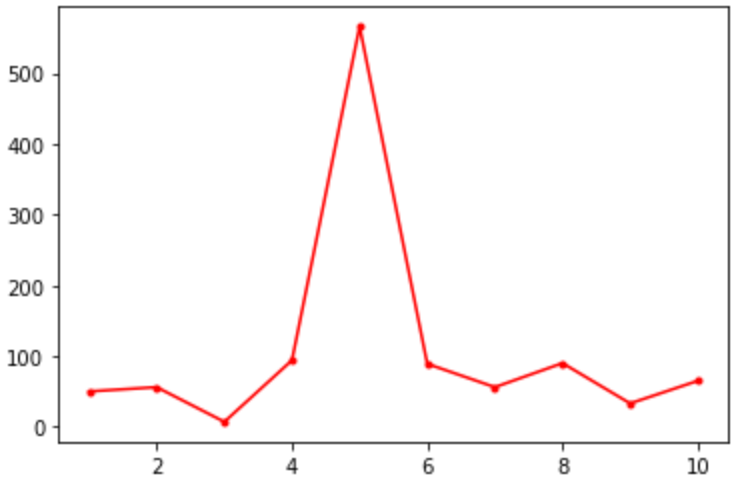
Out[9]:

```
<matplotlib.collections.PathCollection at 0x210f6af1640>
```



In [14]:

```
x_points_1 = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
y_points_1 = np.array([50, 56, 7, 94, 566, 89, 56, 90, 33, 65])
plt.plot(x_points_1, y_points_1, color="red", marker=".")
plt.show()
```


500
400
300
200
100
0
2
4
6
8
10