Plotting with Matplotlib

Quick note

- I am going to be using abbreviations for numpy and matplotlib throughout this presentation

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Basic Plotting

- Syntax

In [4]:
```
X = np.linspace(-np.pi, np.pi, 256)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C)
plt.plot(X,S)

plt.show()
```

Basic Plotting

- Syntax

In [4]:
```
X = np.linspace(-np.pi, np.pi, 256)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C)
plt.plot(X,S)

plt.show()
```

X axis

Basic Plotting

- Syntax

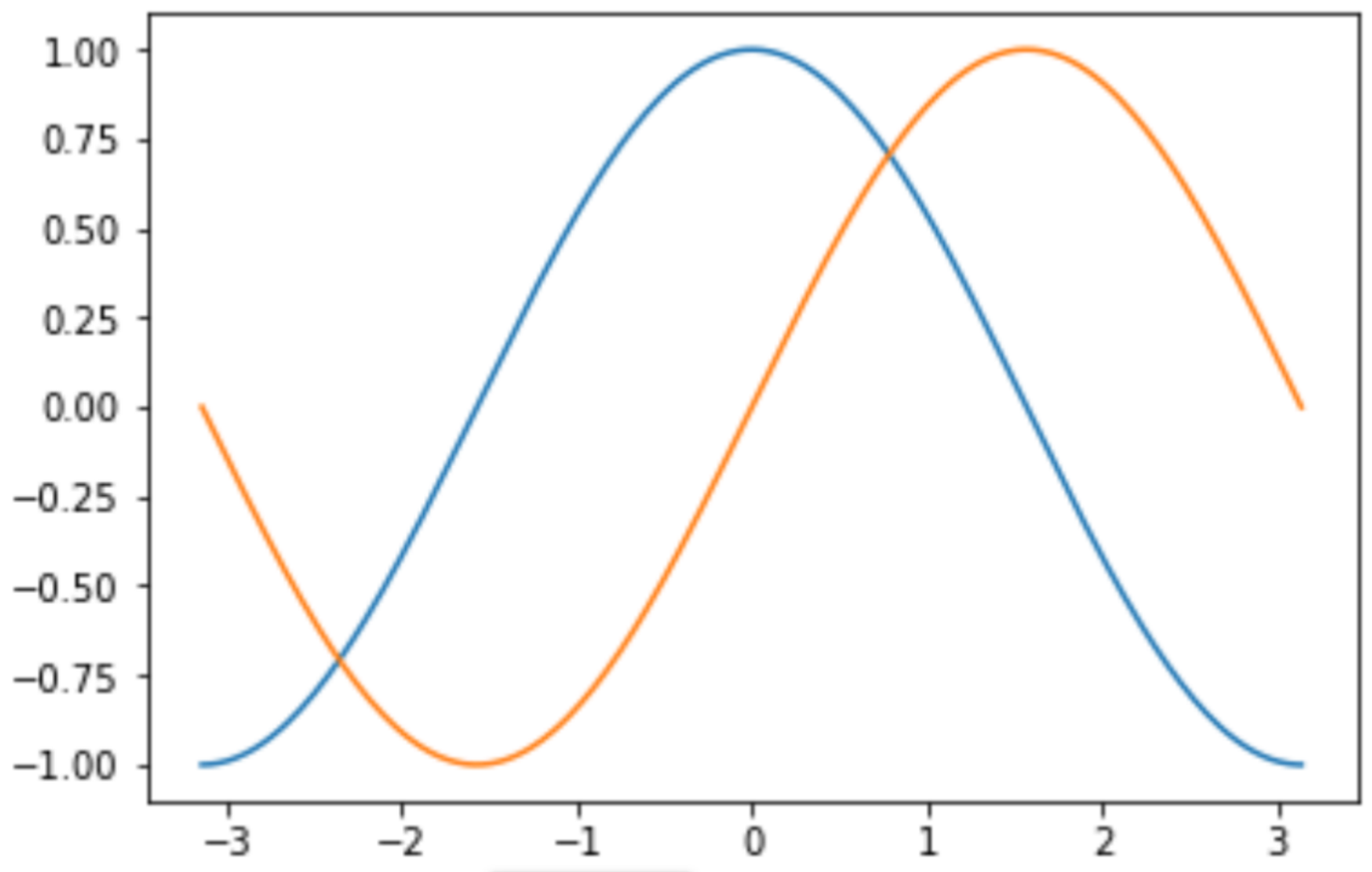
In [4]:
```
X = np.linspace(-np.pi, np.pi, 256)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C)
plt.plot(X,S)

plt.show()
```

Y axis

Basic Plotting

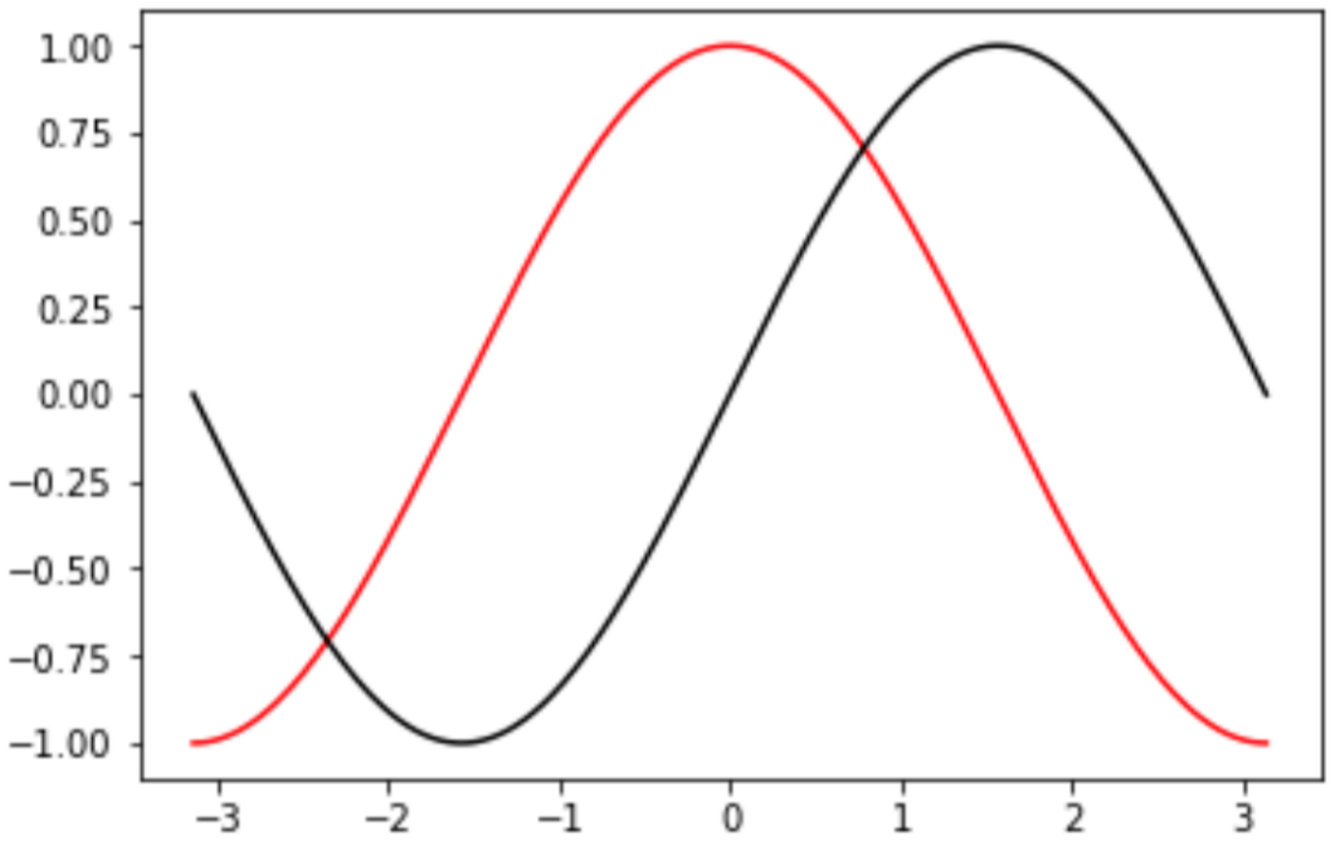
- Syntax

In [4]:
```
X = np.linspace(-np.pi, np.pi, 256)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C)
plt.plot(X,S)

plt.show()
```



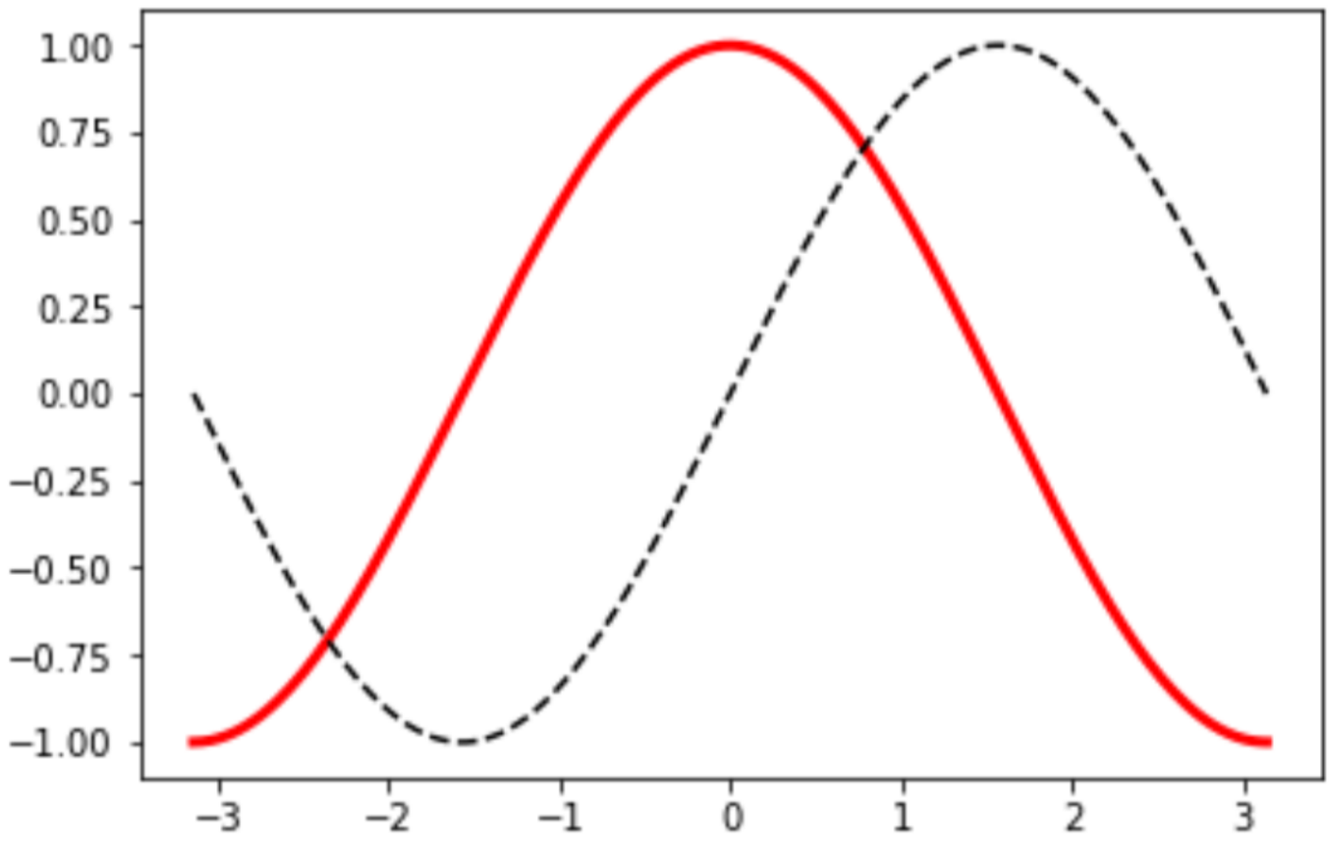
Basic Plotting

In [5]:
```
X = np.linspace(-np.pi, np.pi, 256)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C,color='r')
plt.plot(X,S,color='k')

plt.show()
```



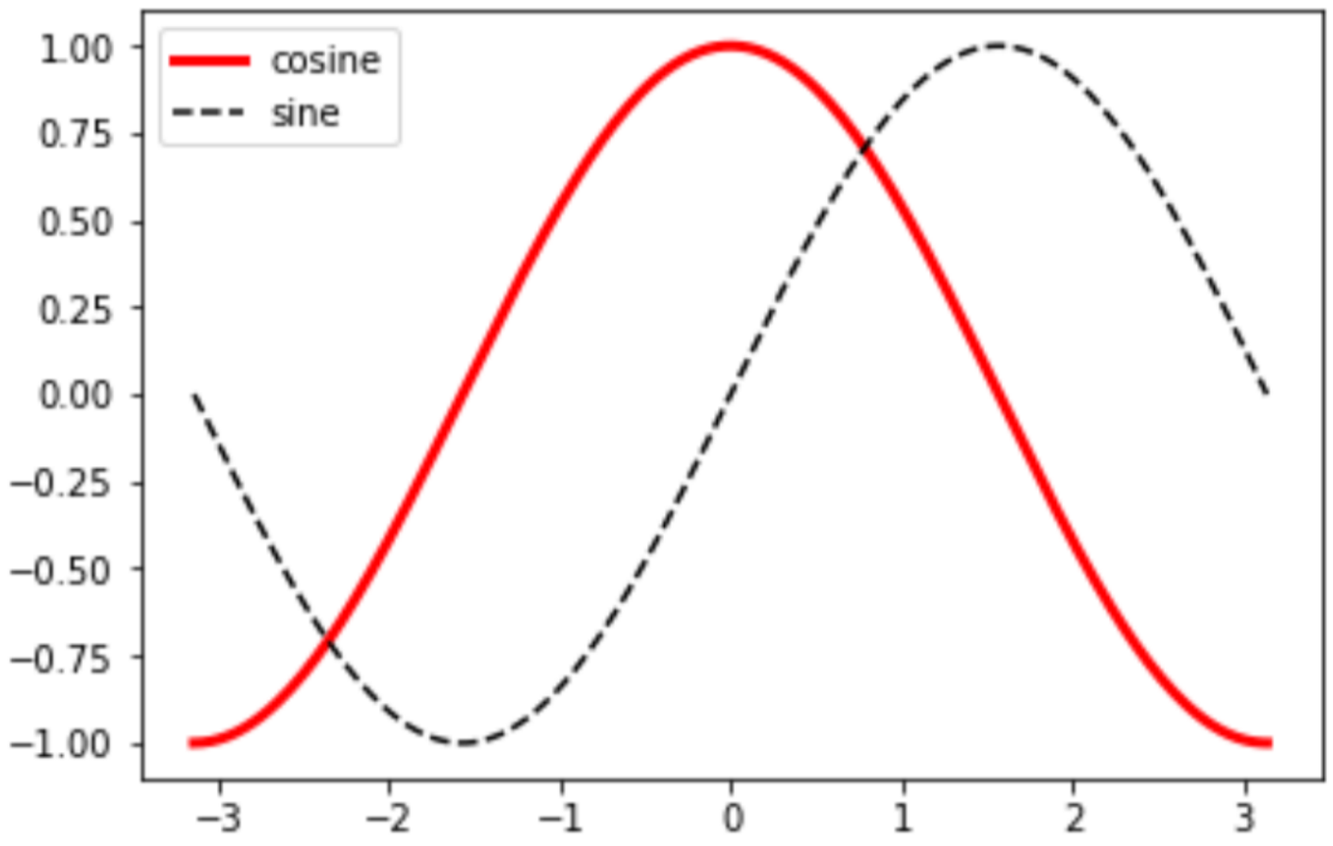
Basic Plotting

In [6]:
```
X = np.linspace(-np.pi, np.pi, 256)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C,color='r',linewidth=3)
plt.plot(X,S,color='k',linestyle='--')

plt.show()
```



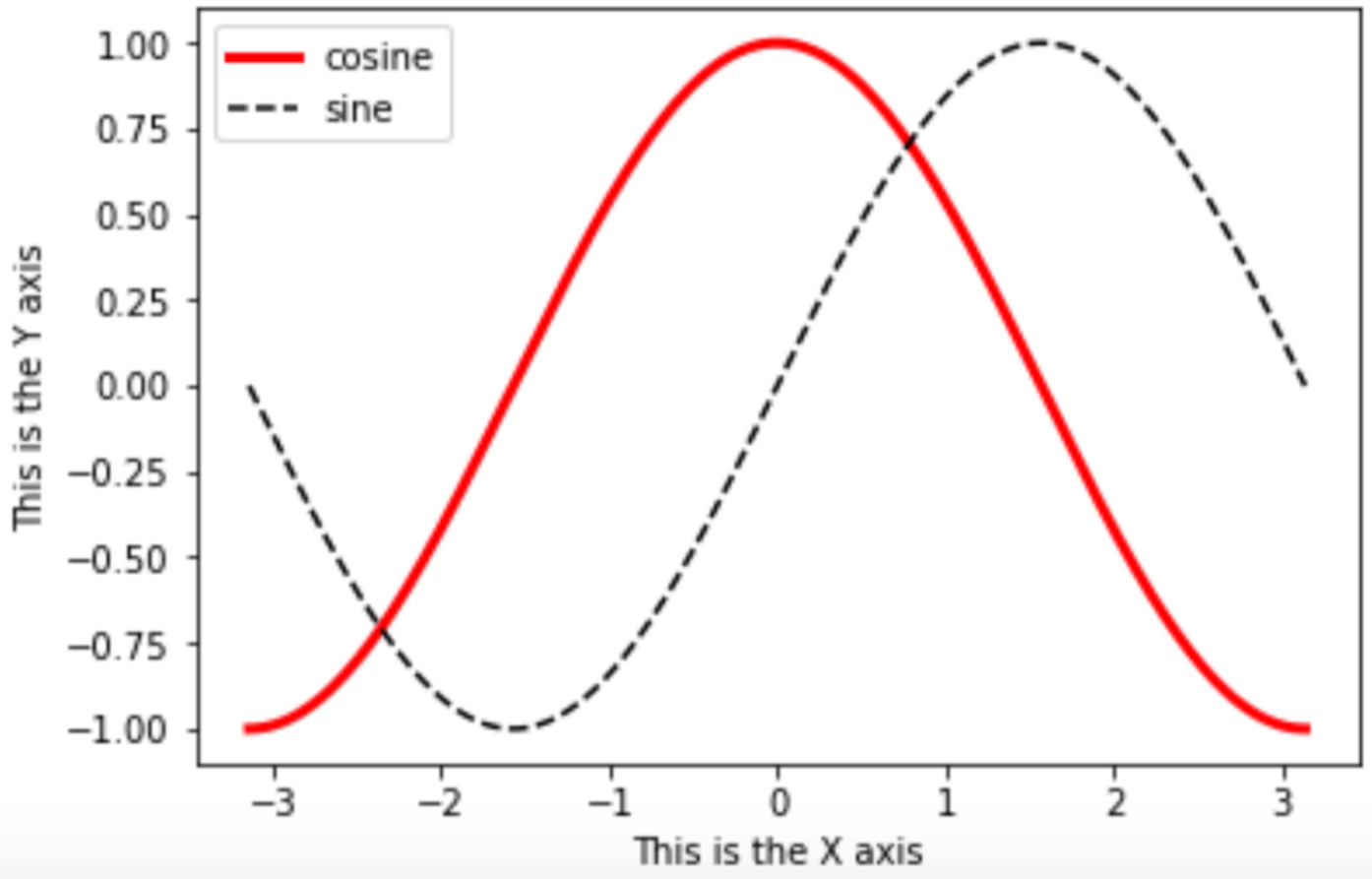
Basic Plotting

In [7]:
```
X = np.linspace(-np.pi, np.pi, 256,endpoint=True)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C,color='r',linewidth=3,label = 'cosine')
plt.plot(X,S,color='k',linestyle='--', label = 'sine')

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



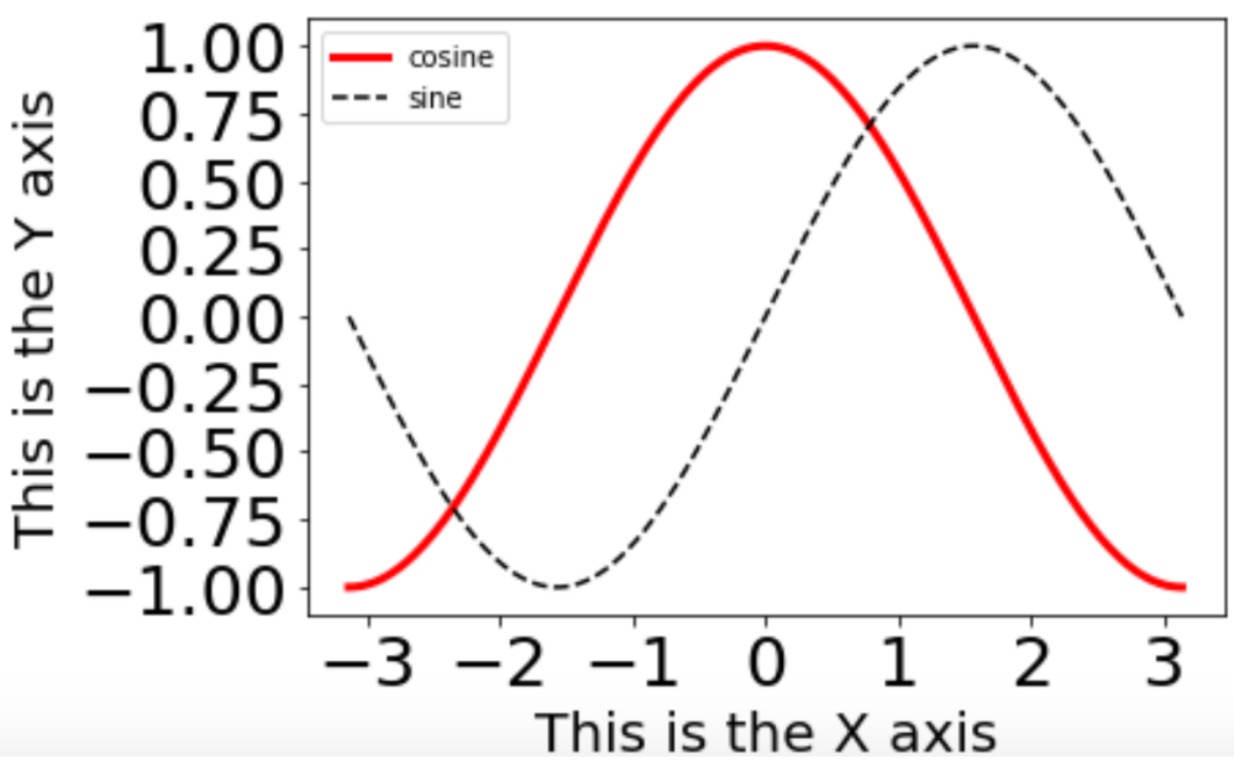
Basic Plotting

In [10]:
```
X = np.linspace(-np.pi, np.pi, 256,endpoint=True)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C,color='r',linewidth=3,label = 'cosine')
plt.plot(X,S,color='k',linestyle='--', label = 'sine')

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

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



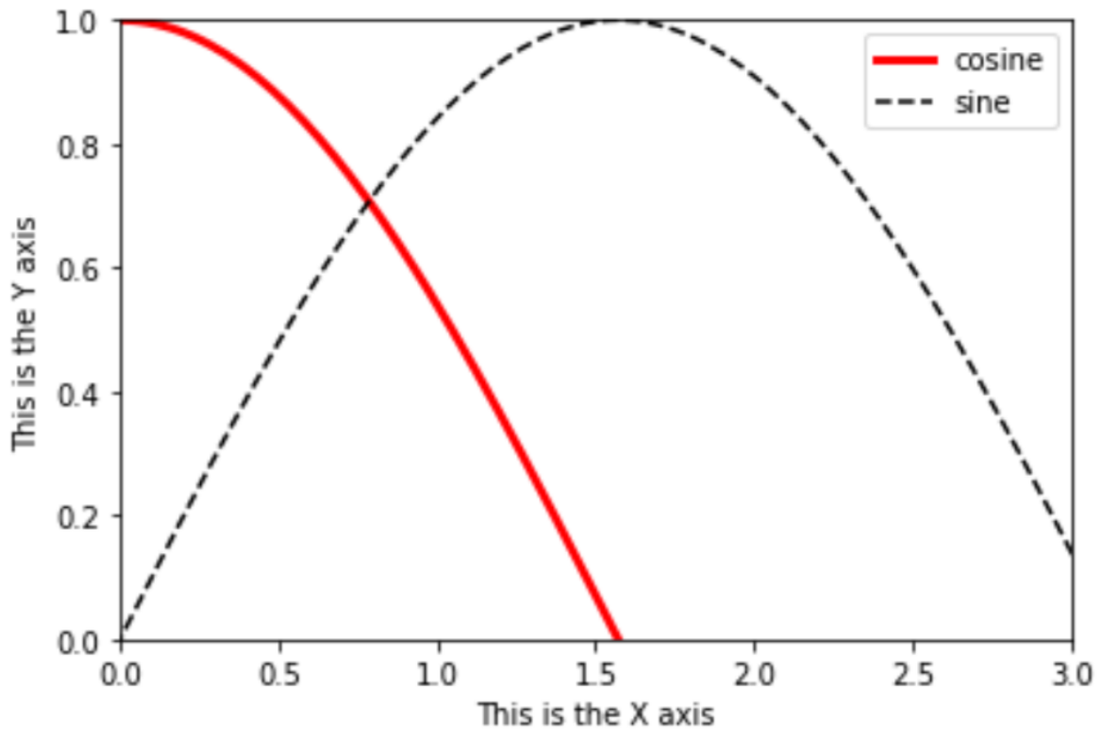
Basic Plotting

In [12]:
```
X = np.linspace(-np.pi, np.pi, 256,endpoint=True)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C,color='r',linewidth=3,label = 'cosine')
plt.plot(X,S,color='k',linestyle='--', label = 'sine')

plt.xlabel('This is the X axis',fontsize=20)
plt.ylabel('This is the Y axis',fontsize=20)

plt.xticks(fontsize=25)
plt.yticks(fontsize=25)

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



Basic Plotting

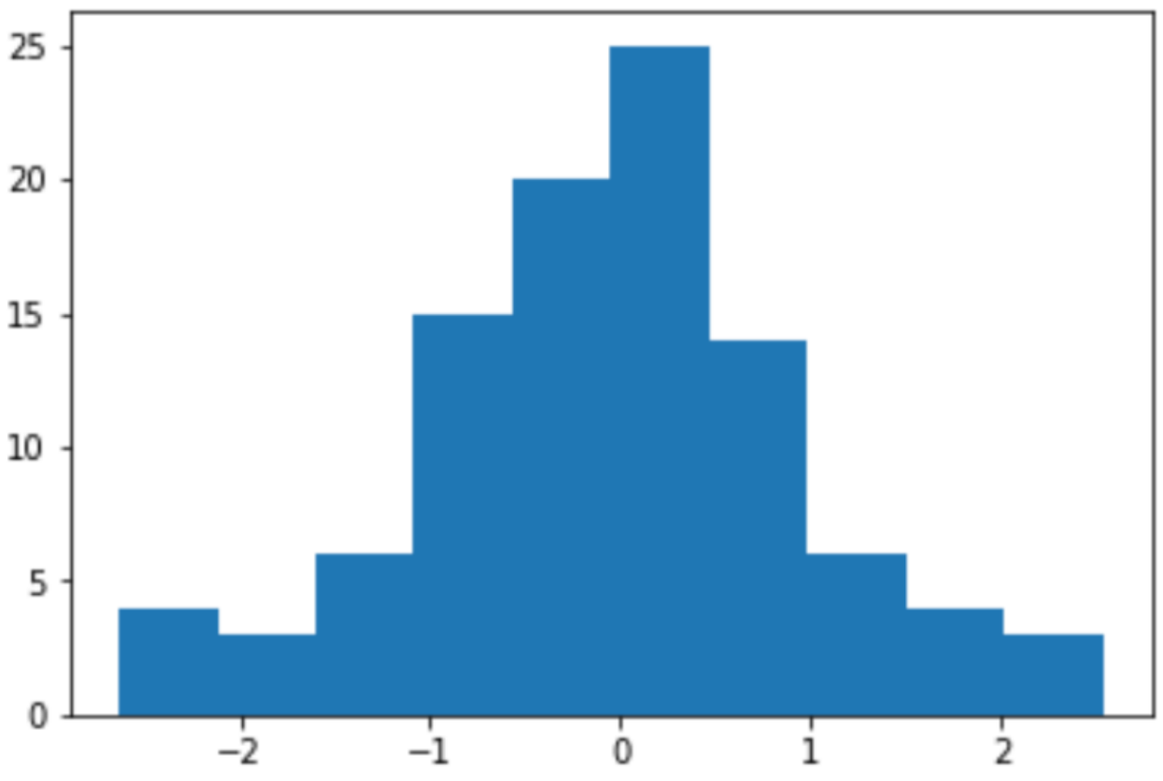
In [148]:
```
X = np.linspace(-np.pi, np.pi, 256,endpoint=True)
C,S = np.cos(X), np.sin(X)

plt.plot(X,C,color='r',linewidth=3,label = 'cosine')
plt.plot(X,S,color='k',linestyle='--', label = 'sine')

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

plt.xlim([0.0,3.0])
plt.ylim([0.0,1.0])

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



Basic Plotting

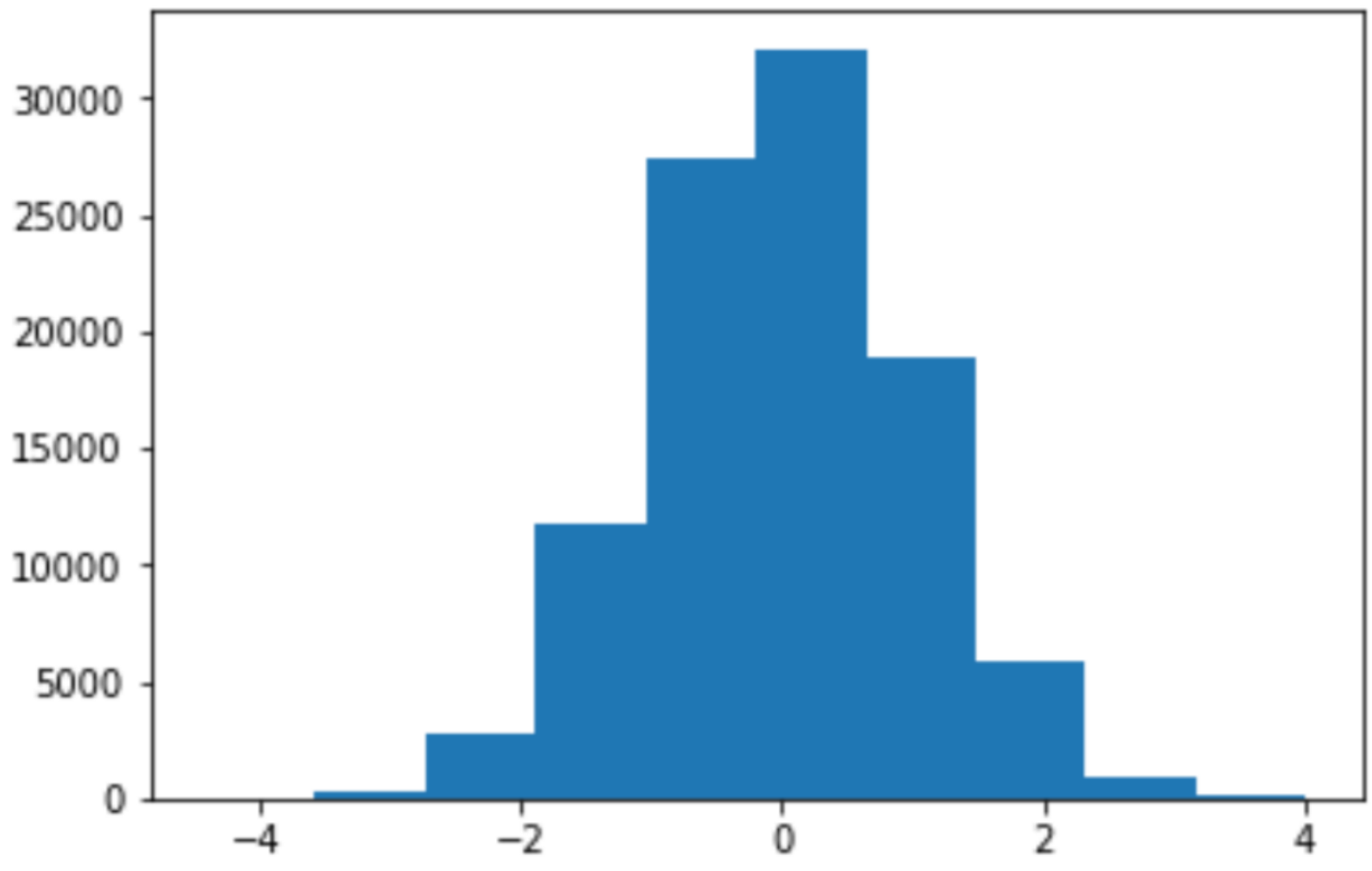
In [31]:
```
gaussian_data = np.random.normal(size=100)

plt.hist(gaussian_data)
```

Out[31]:
```
(array([  4.,   3.,   6.,  15.,  20.,  25.,  14.,   6.,   4.,   3.]),
 array([-2.63453856, -2.11759251, -1.60064646, -1.08370041, -0.56675437,
        -0.04980832,  0.46713773,  0.98408377,  1.50102982,  2.01797587,
         2.53492192]),
 <a list of 10 Patch objects>)
```



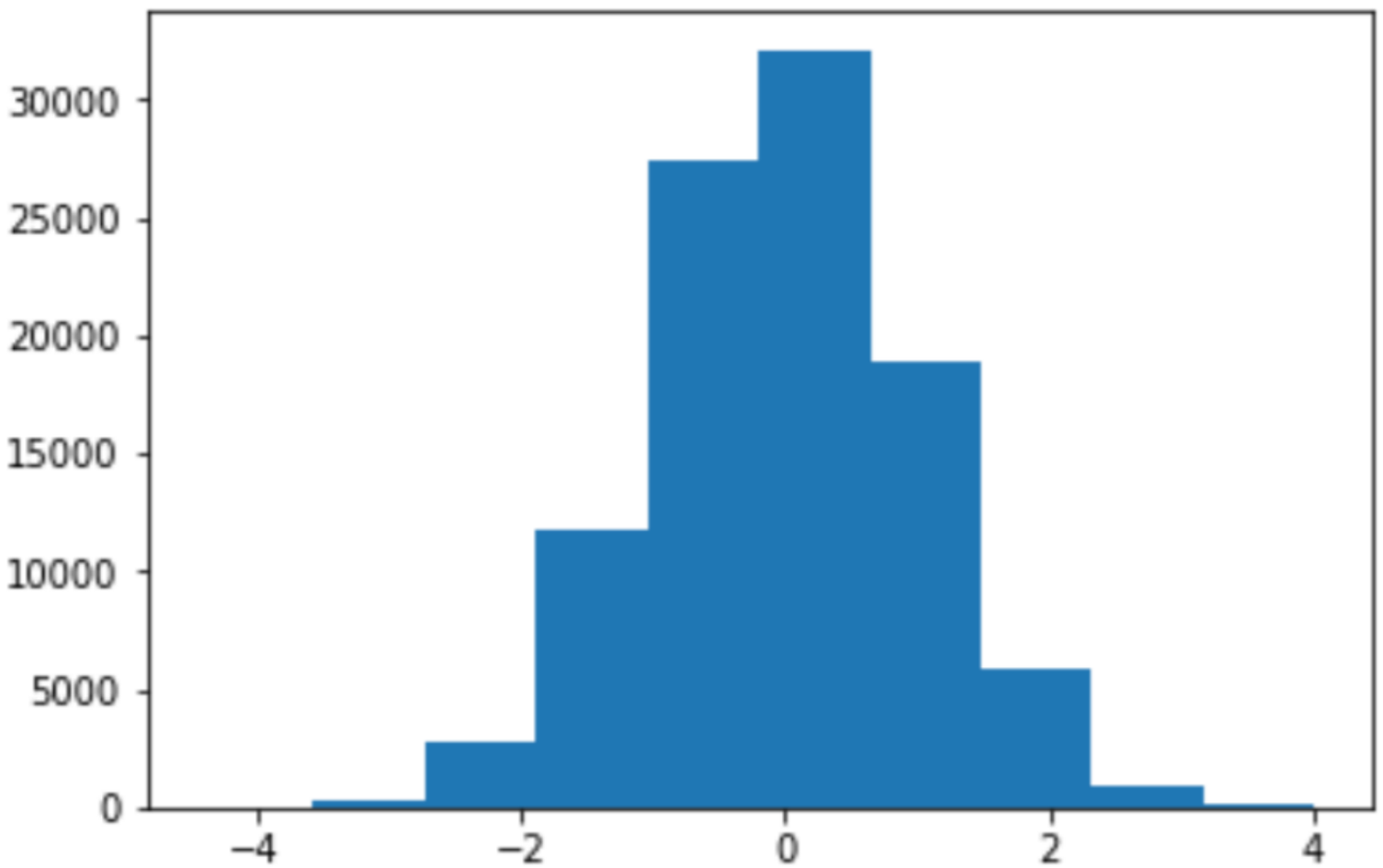
Basic Plotting

In [41]:
```
gaussian_data = np.random.normal(size=100000)

n,bins,patches = plt.hist(gaussian_data)
```



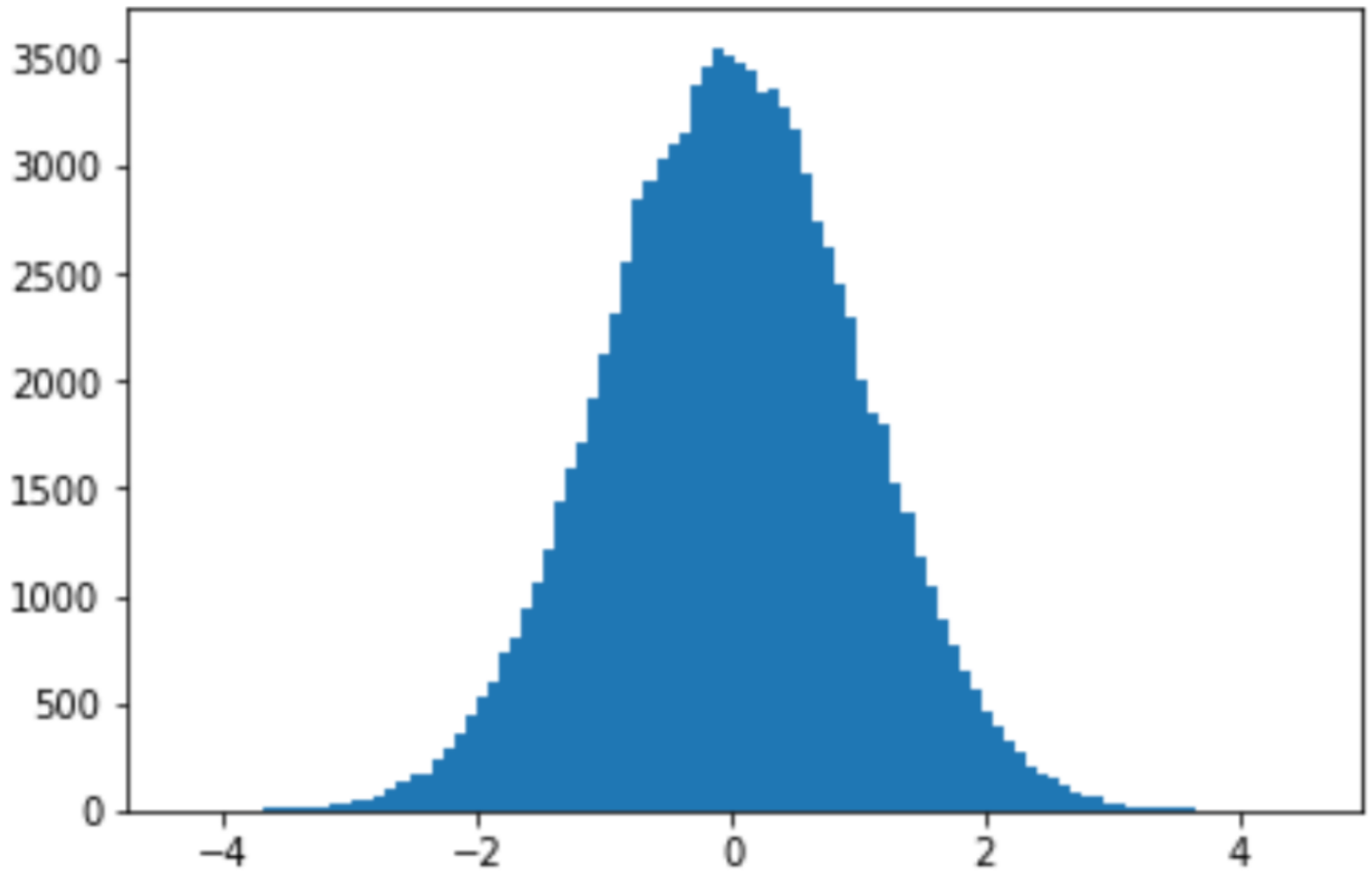
Basic Plotting

In [41]:
```
gaussian_data = np.random.normal(size=100000)

n,bins,patches = plt.hist(gaussian_data)
```



Basic Plotting

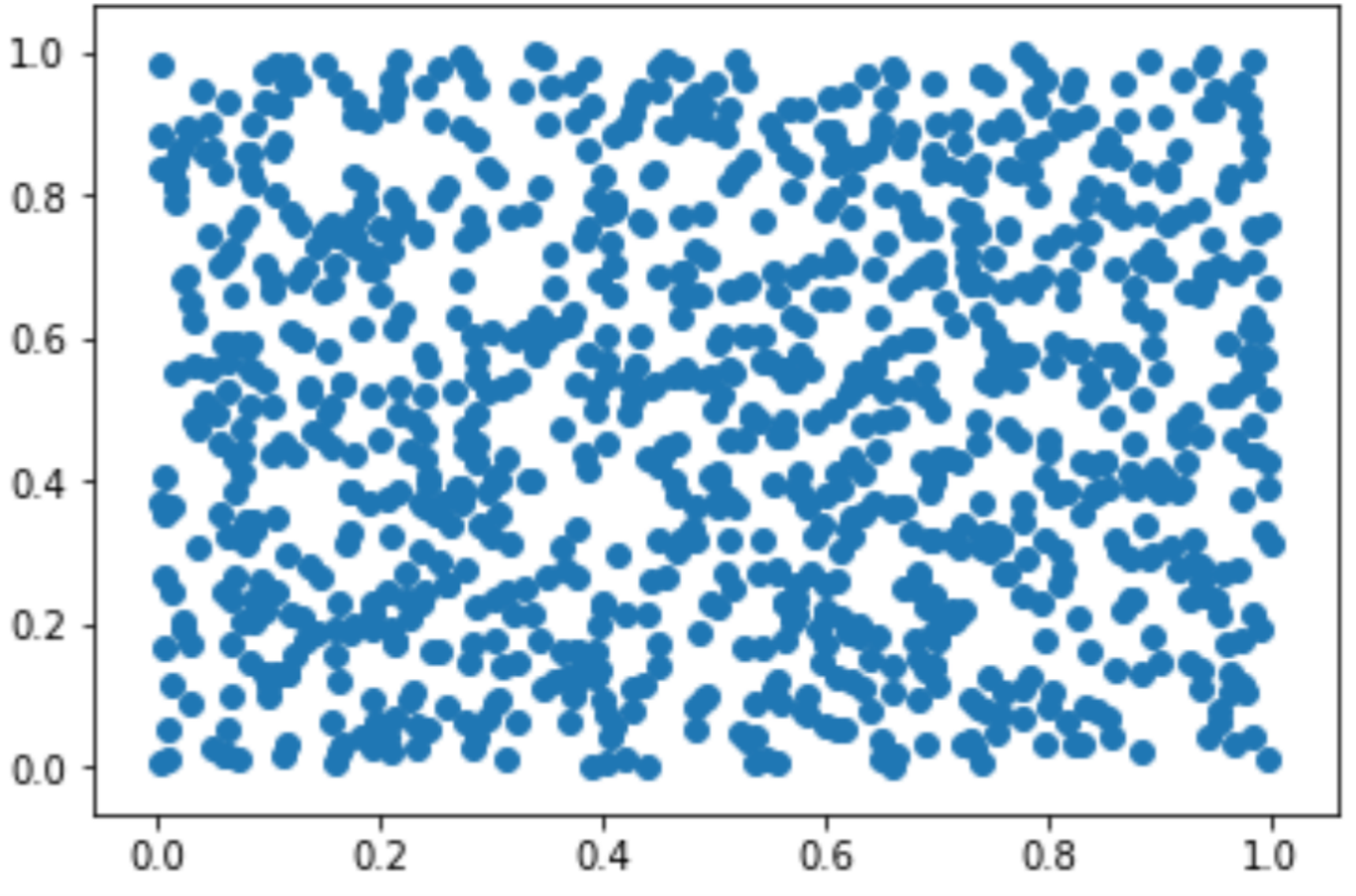
In [42]:
```
gaussian_data = np.random.normal(size=100000)

n,bins,patches = plt.hist(gaussian_data,bins=100)
```



Basic Plotting

In [46]:
```
n = 1000

x = np.random.random_sample(n)
y = np.random.random_sample(n)

plt.scatter(x,y)
```

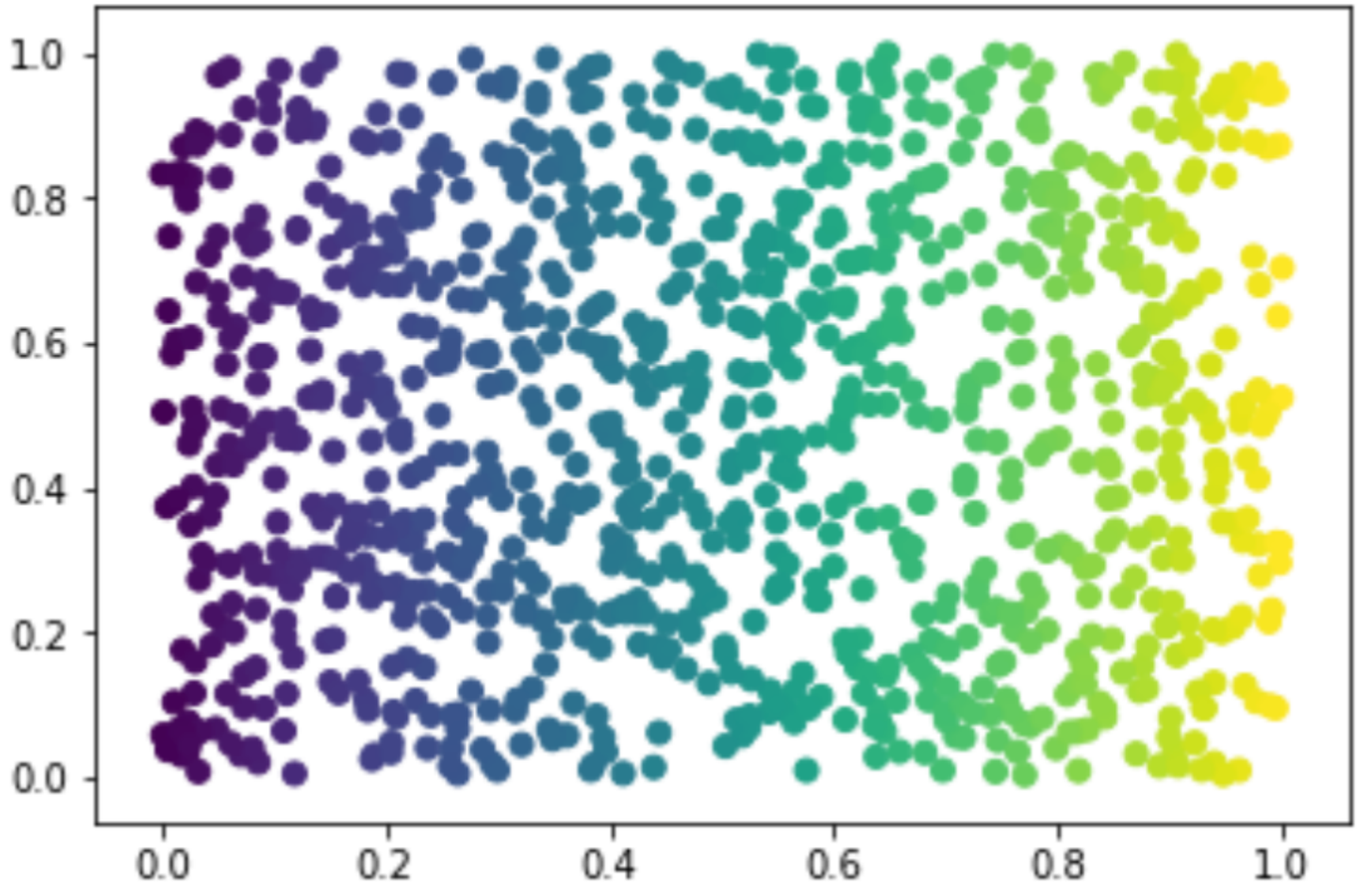
Out[46]: <matplotlib.collections.PathCollection at 0x1167c37d0>



Basic Plotting

In [51]:
```
n = 1000

x = np.random.random_sample(n)
y = np.random.random_sample(n)

plt.scatter(x,y,c=x)
```

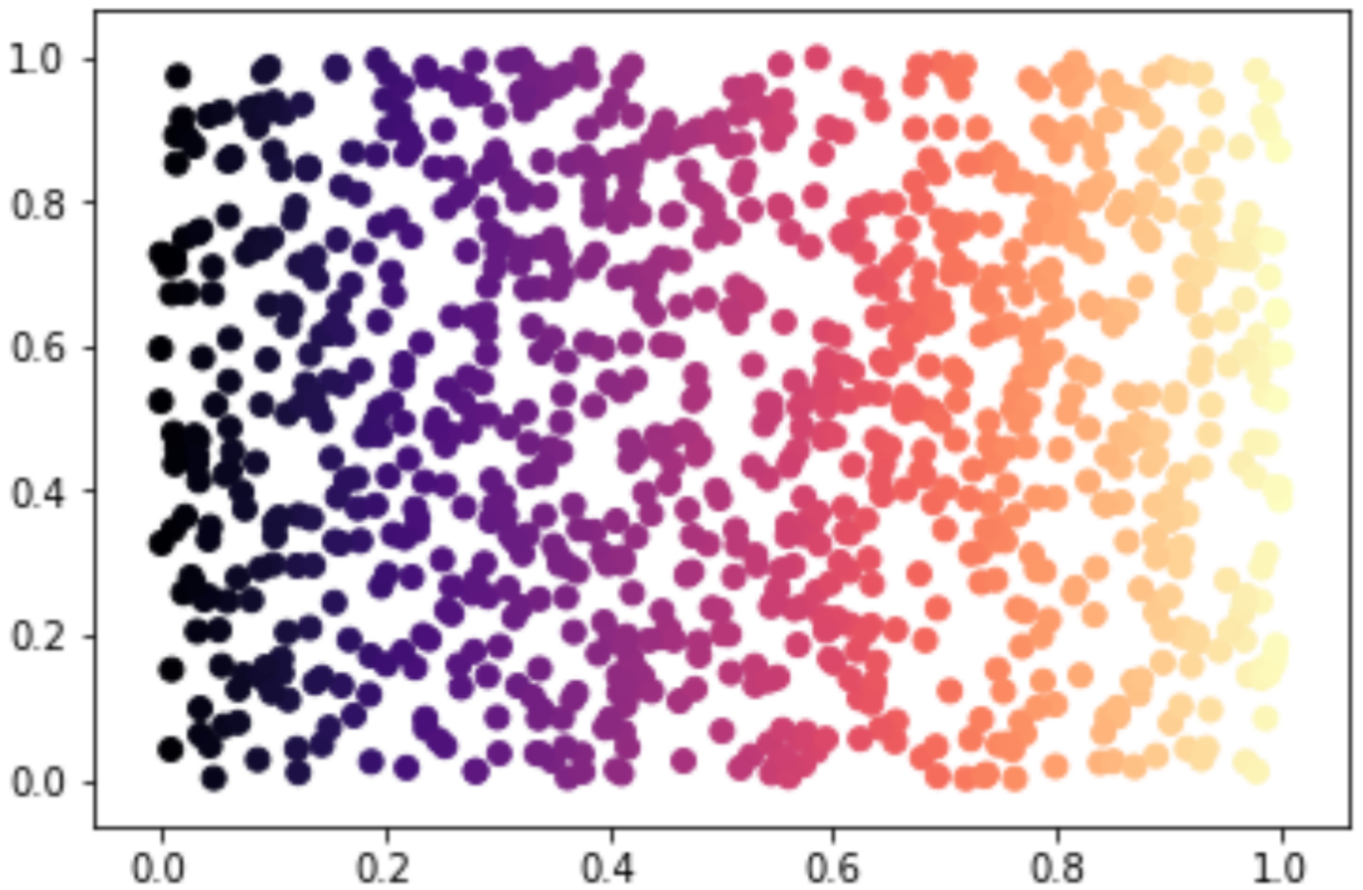
Out[51]: <matplotlib.collections.PathCollection at 0x115ce5bd0>



Basic Plotting

In [53]:
```
n = 1000

x = np.random.random_sample(n)
y = np.random.random_sample(n)

plt.scatter(x,y,c=x,cmap='magma')
```

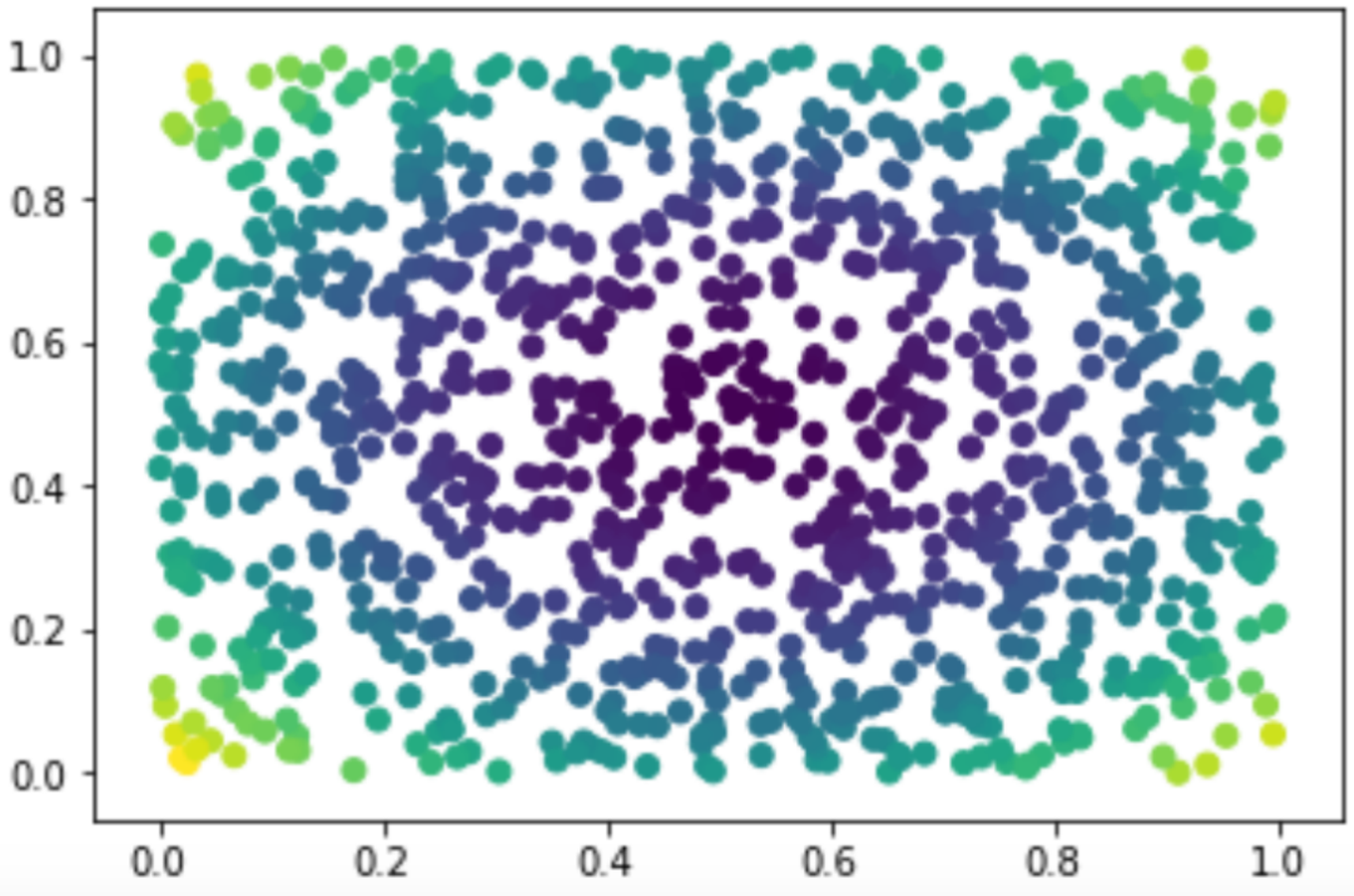
Out[53]: <matplotlib.collections.PathCollection at 0x115d94810>



Basic Plotting

In [58]:
```
n = 1000

x = np.random.random_sample(n)
y = np.random.random_sample(n)

z = (x-np.median(x))**2.0+(y-np.median(y))**2.0

plt.scatter(x,y,c=z)
```

Out[58]: <matplotlib.collections.PathCollection at 0x116f0b390>



To Do

- Use the data file to make a map of the stars and a color magnitude diagram (CMD)
- Remember:
    - array[:,n] gives you the nth column of an array
    - For this data file the columns are:
        - 0 - x position
        - 1 - y position
        - 2 - V band magnitude
        - 3 - B band magnitude
- Also magnitudes are INVERTED smaller numbers are BRIGHTER!