भारतीय प्रौद्योगिकी संस्थान तिरुपति

TIRUPATI

Python Matplotlib

Panchatcharam Mariappan

Assistant Professor

**Department of Mathematics and Statistics,
IIT Tirupati**

- Plotting library
- Huge number of examples for tacking unique problems

Matplotlib-plot function

- `import matplotlib as mpl`
- `import matplotlib.pyplot as plt`
- `plt.plot(x_values, y_values, format_string [, x, y, format, ])`

Matplotlib-plot function

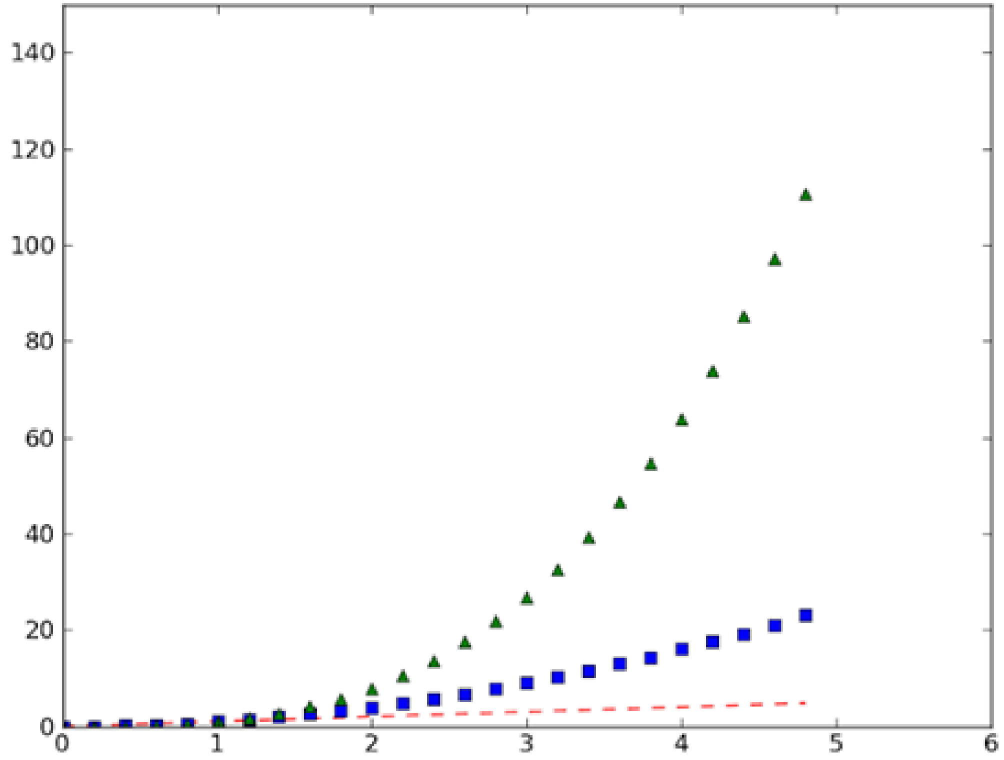
- `import matplotlib as mpl`
- `import matplotlib.pyplot as plt`
- `plt.plot(x_values, y_values, format_string [, x, y, format, ])`
- `x_values` and `y_values`: numpy arrays, if not, internal conversion to numpy array
- `format_string`: the color and line type of the plot
  - ‘bs’ blue squares
  - ‘ro’: red circles
- Line properties: Set them via keyword arguments to the plot function.
  - `label`, `linewidth`, `animated`, `color`, etc…

Matplotlib-plot function

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

# evenly sampled time at .2 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.axis([0, 6, 0, 150]) # x and y range of axis
plt.show()
```

Matplotlib-plot function

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

# evenly sampled time at .2 intervals
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
plt.axis([0, 6, 0, 150]) # x and y range of ax
plt.show()
```



Matplotlib-plot function

```
import numpy as np
import matplotlib.figure as figure

t = np.arange(0, 5, .2)

f = figure.Figure()
axes = f.add_subplot(111)
axes.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
axes.axis([0, 6, 0, 150])
```

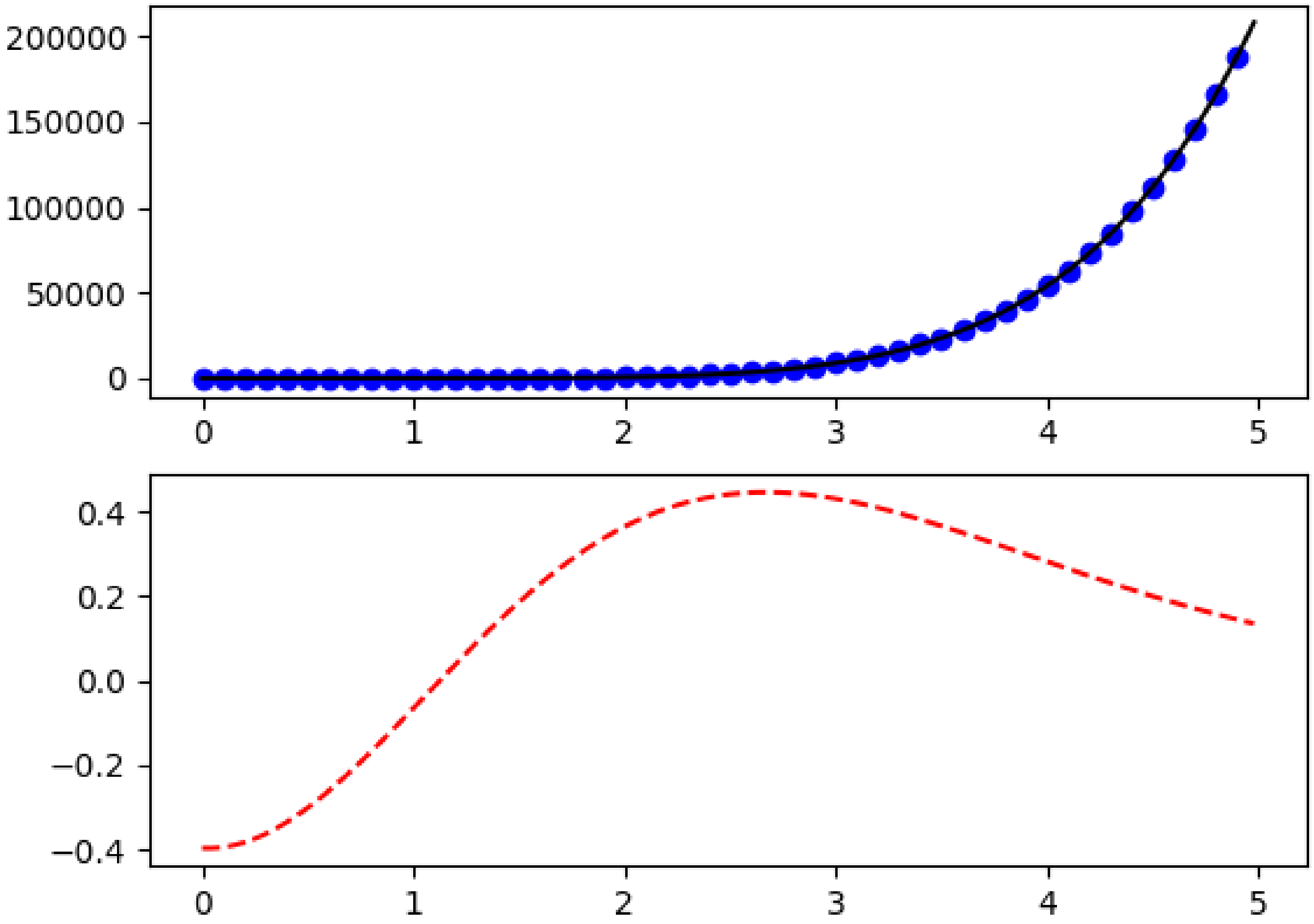
Matplotlib-subplot function

- A script can generate multiple figures, but typically you'll only have one.
- To create multiple plots within a figure, either use the subplot() function which manages the layout of the figure or use add_axes().

Matplotlib-plot function

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.special

def Legendre(t,n):
    p=scipy.special.legendre(n)
    return p(t)
def Bessel(t,n):
    return scipy.special.jv(t,n)
t1 = np.arange(0.0, 5.0, 0.1) #50
t2 = np.arange(0.0, 5.0, 0.02) #250
plt.figure(1)
n=6
plt.subplot(211) # 2 rows, 1 column, 1st plot
plt.plot(t1, Legendre(t1,n), 'bo', t2, Legendre(t2,n), 'k')
plt.subplot(212) # 2 rows, 1 column, 2nd plot
plt.plot(t2, Bessel(t2,4), 'r--')
plt.show()
```

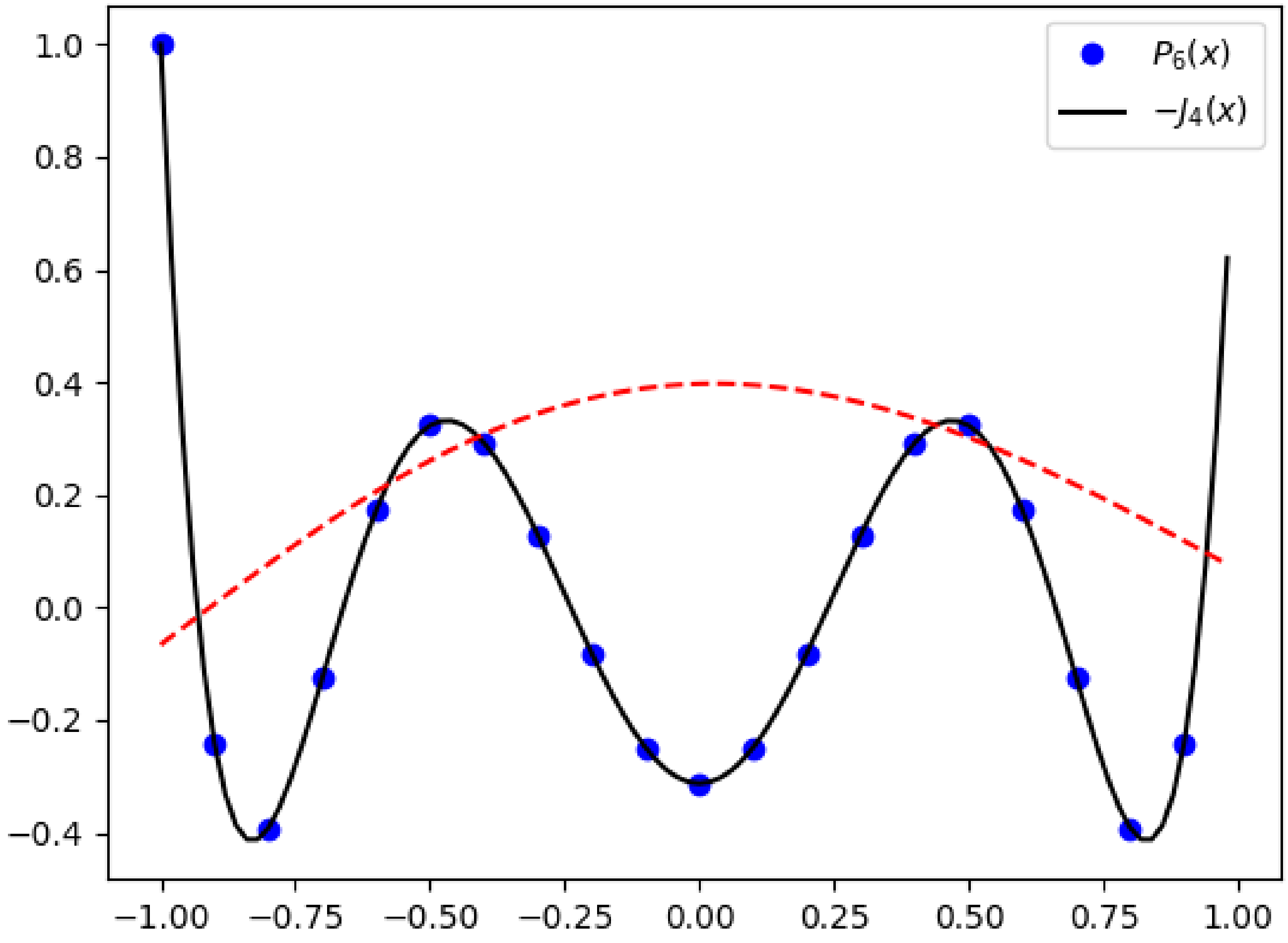
Matplotlib-plot function



Matplotlib-plot function

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.special

def Legendre(t,n):
    p=scipy.special.legendre(n)
    return p(t)
def Bessel(t,n):
    return scipy.special.jv(t,n)
t1 = np.arange(-1.0, 1.0, 0.1)
t2 = np.arange(-1.0, 1.0, 0.02)
plt.figure(1)
n=6
plt.plot(t1, Legendre(t1,n), 'bo', t2, Legendre(t2,n), 'k')
plt.plot(t2, -Bessel(t2,4), 'r--')
plt.legend(['$P_6(x)$','$-J_4(x)$'])
plt.show()
```

Matplotlib-plot function



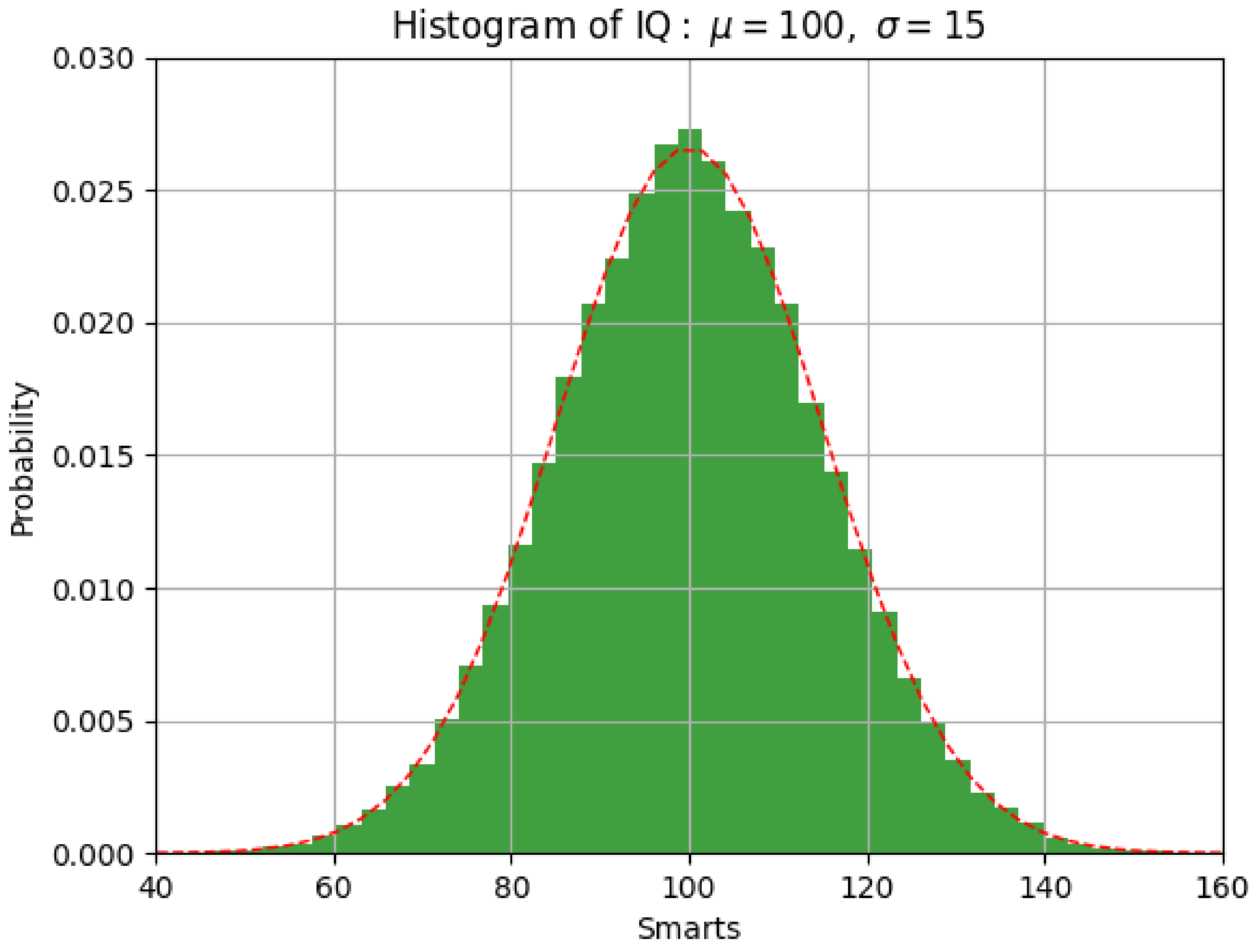
Matplotlib-plot function

- `text()` : to add text in an arbitrary location
- `xlabel()`, `ylabel()`, `title()`, **axes labels and title**
- `clear()` removes all plots from the axes.

Matplotlib-plot function

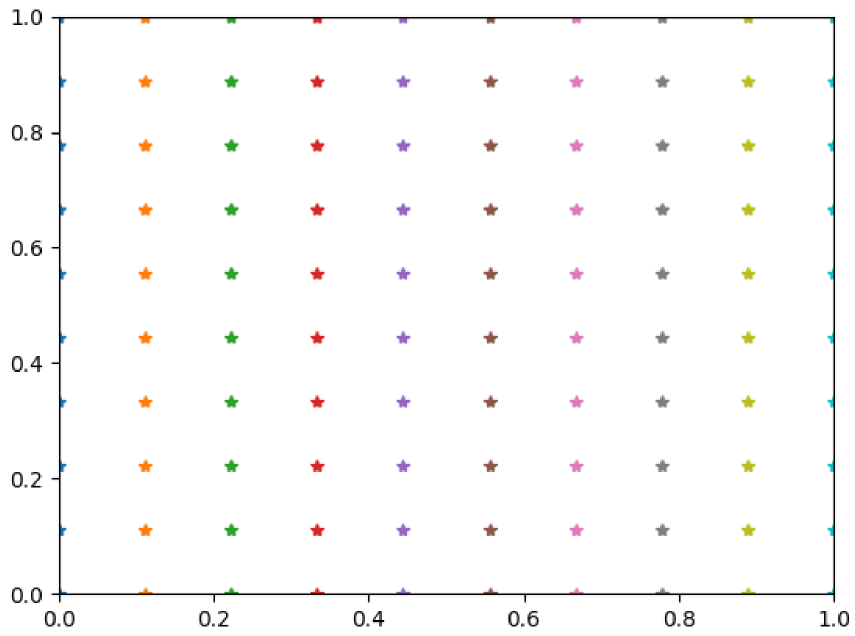
```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
mu, sigma = 100, 15
x = mu + sigma*np.random.randn(100000)
# the histogram of the data
n, bins, patches = plt.hist(x, 50, density=True, facecolor='green', alpha=0.75)
# add a 'best fit' line
y = norm.pdf( bins, mu, sigma)
l = plt.plot(bins, y, 'r--', linewidth=1)
plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title(r'$\mathrm{Histogram\ of\ IQ:}\ \mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.show()
```

Matplotlib-plot function

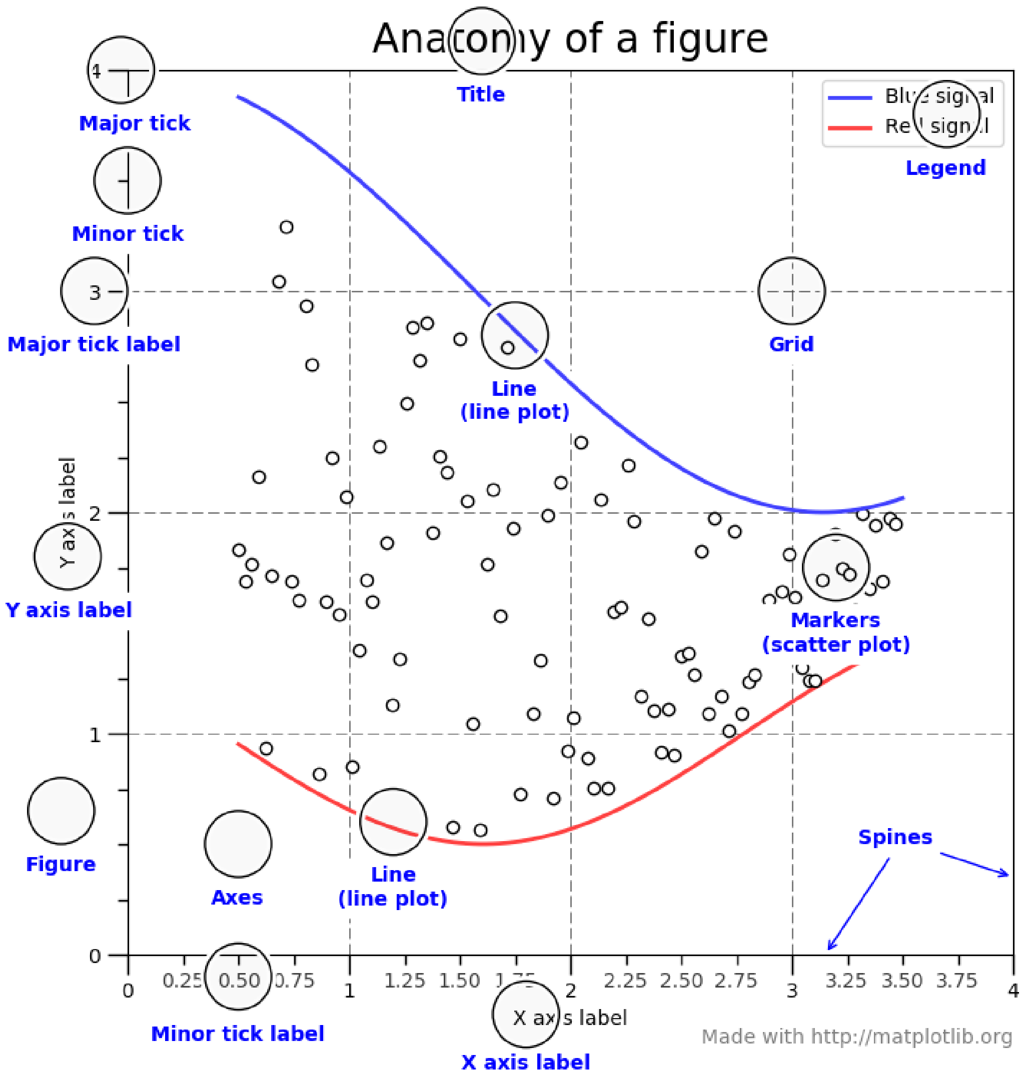


Matplotlib-plot function

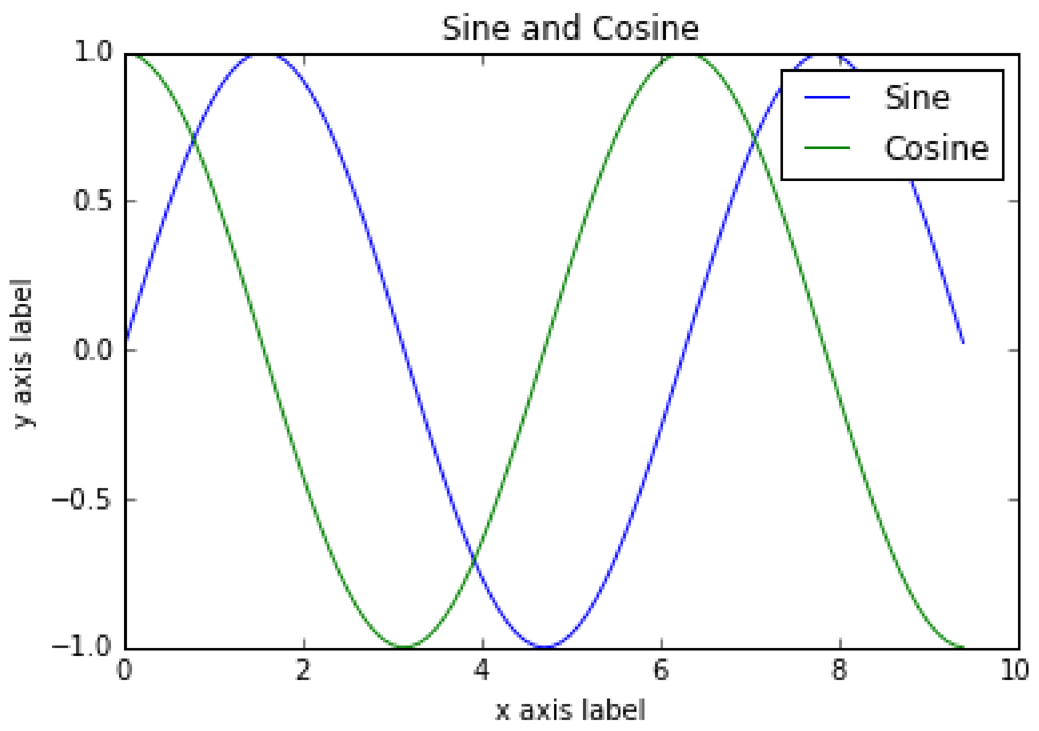
```
#FDM Discretization
import numpy as np
x=np.linspace(0,1,num=10,endpoint=True)
y=np.linspace(0,1,num=10,endpoint=True)
X,Y=np.meshgrid(x,y)
import matplotlib.pyplot as plt
plt.plot(X,Y,'*')
plt.axis([0,1,0,1])
plt.show()
```



Matplotlib-plot function

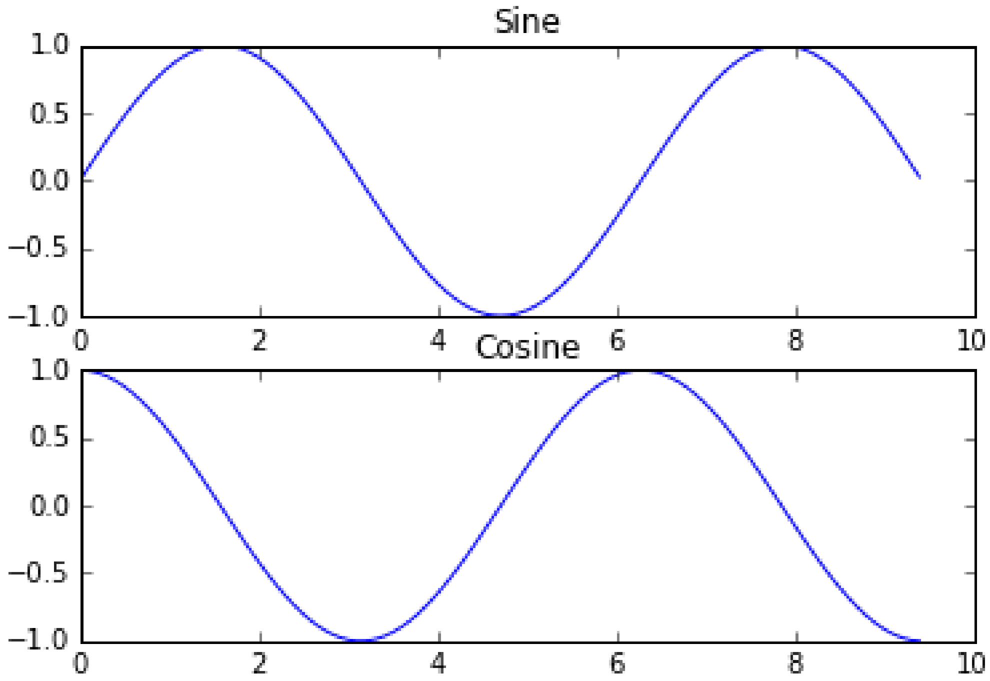


Produce the following Graph using Matplotlib

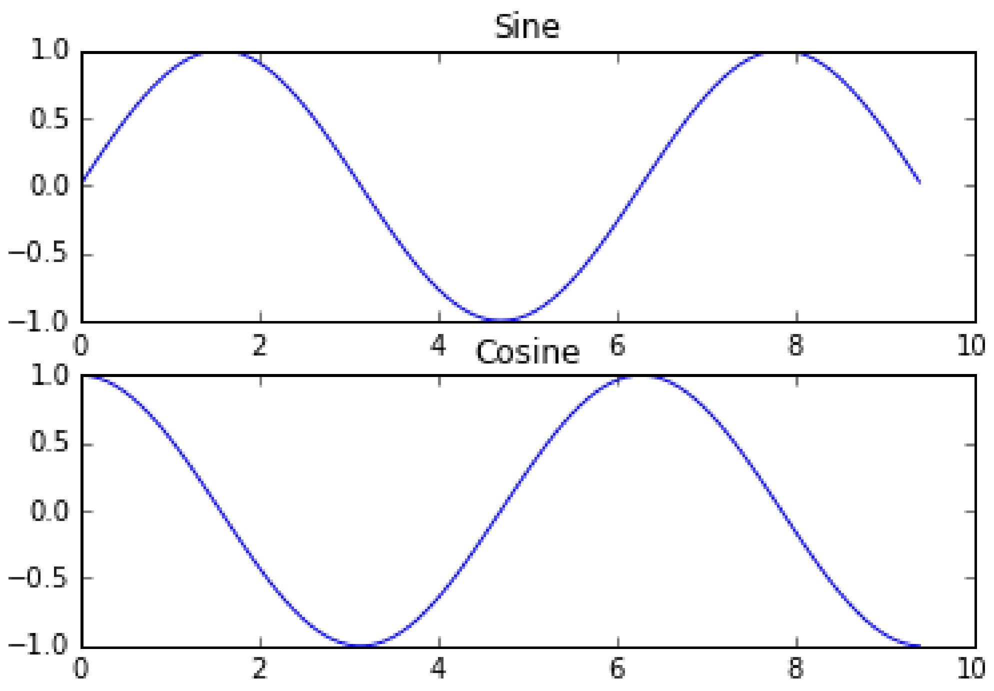


Exercises

Produce the following Graph using Matplotlib

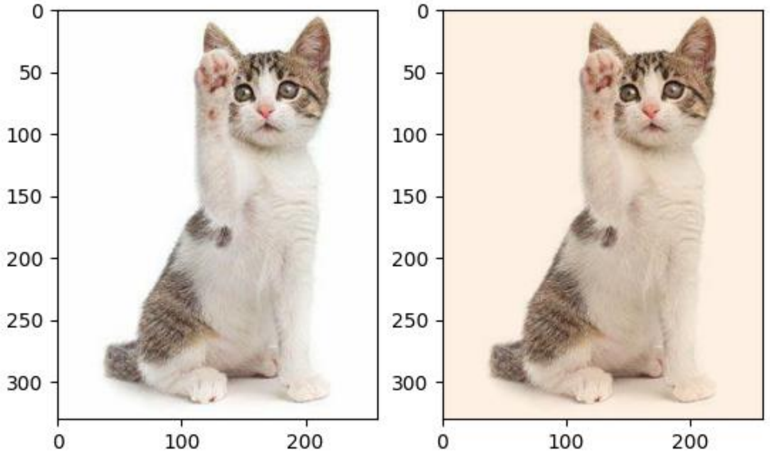


Produce the following Graph using Matplotlib



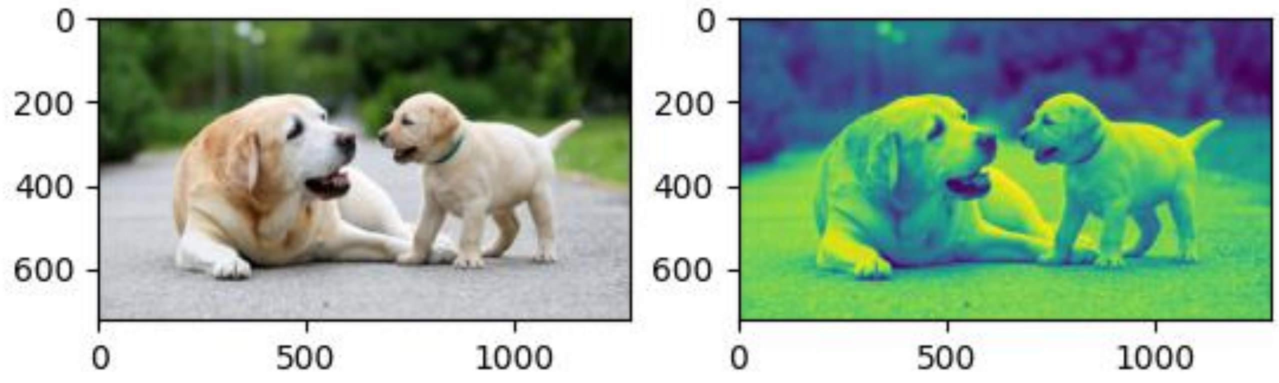
Matplotlib-plot function

```
import numpy as np
from matplotlib.pyplot import imread
import matplotlib.pyplot as plt

img = imread('cat.jpg')
img_tinted = img * [1, 0.95, 0.9]

plt.subplot(1, 2, 1)
plt.imshow(img)


plt.subplot(1, 2, 2)
plt.imshow(np.uint8(img_tinted))
plt.show()
```



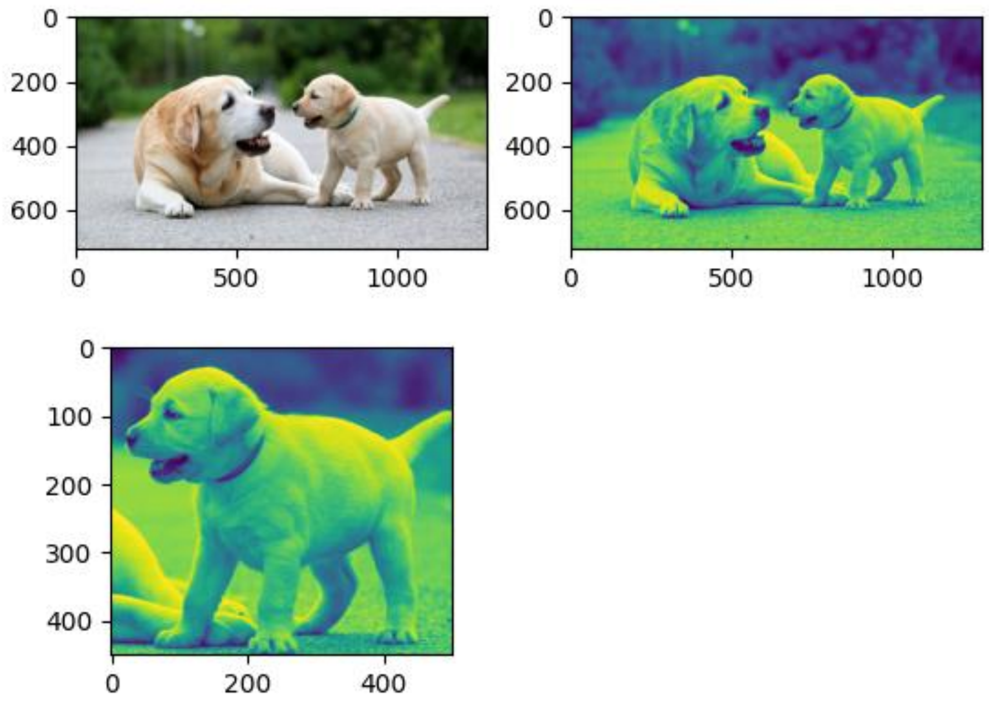
Matplotlib-plot function

```
##Dog
import numpy as np
from matplotlib.pyplot import imread
import matplotlib.pyplot as plt

def rgb2gray(rgb):
    r, g, b = rgb[:,:,0], rgb[:,:,1], rgb[:,:,2]
    return 0.2989 * r + 0.5870 * g + 0.1140 * b
img = imread('dog.jpg')
grayscale=rgb2gray(img)
print(grayscale.shape)
plt.subplot(1, 2, 1)
plt.imshow(img)
plt.subplot(1, 2, 2)
plt.imshow(grayscale)
plt.show()
```



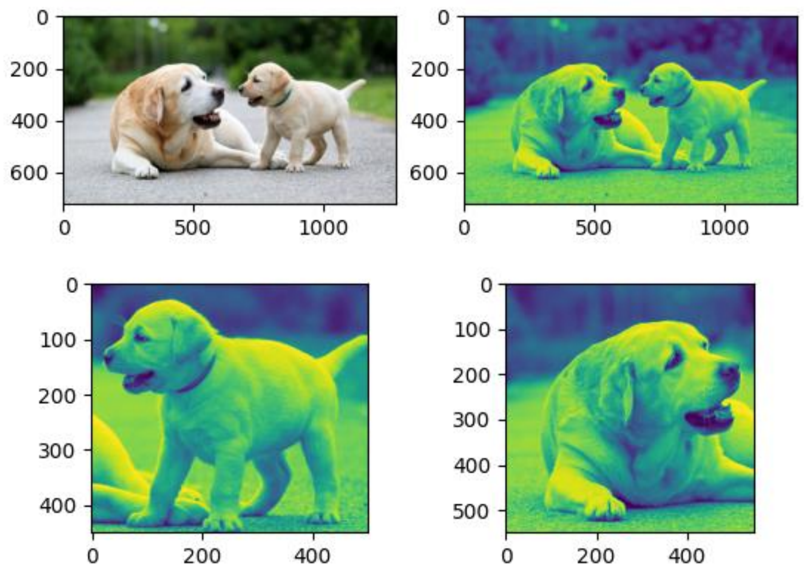
Matplotlib-plot Slicing

```
plt.subplot(2, 2, 1)
plt.imshow(img)

plt.subplot(2, 2, 2)
plt.imshow(grayscale)

plt.subplot(2, 2, 3)
plt.imshow(grayscale[150:600,650:1150])
plt.show()
```



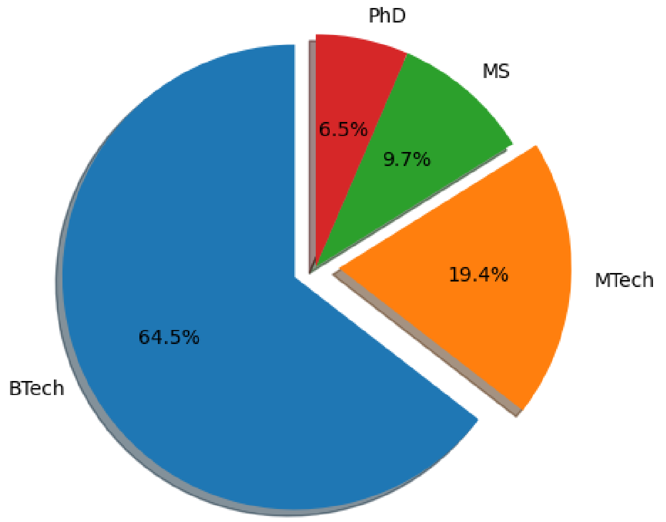
Exercises

Produce the following Graph using Matplotlib



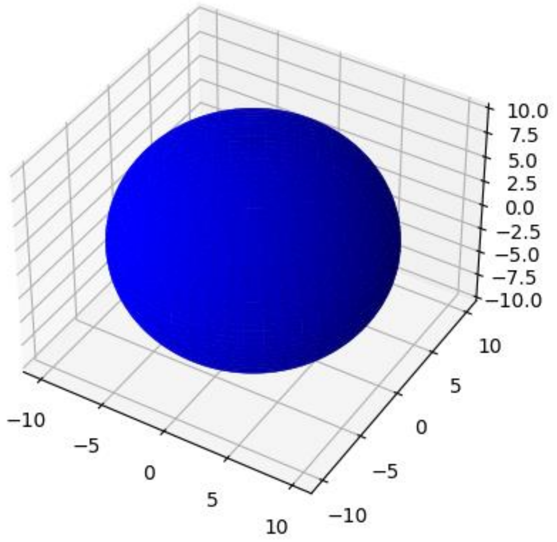
Matplotlib-plot function

```
import matplotlib.pyplot as plt

labels = 'BTech', 'MTech', 'MS', 'PhD'
sizes = [100, 30, 15, 10]
explode = (0.1, 0.1, 0, 0)  # only "explode"

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode,
labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')

plt.show()
```



Matplotlib-plot function

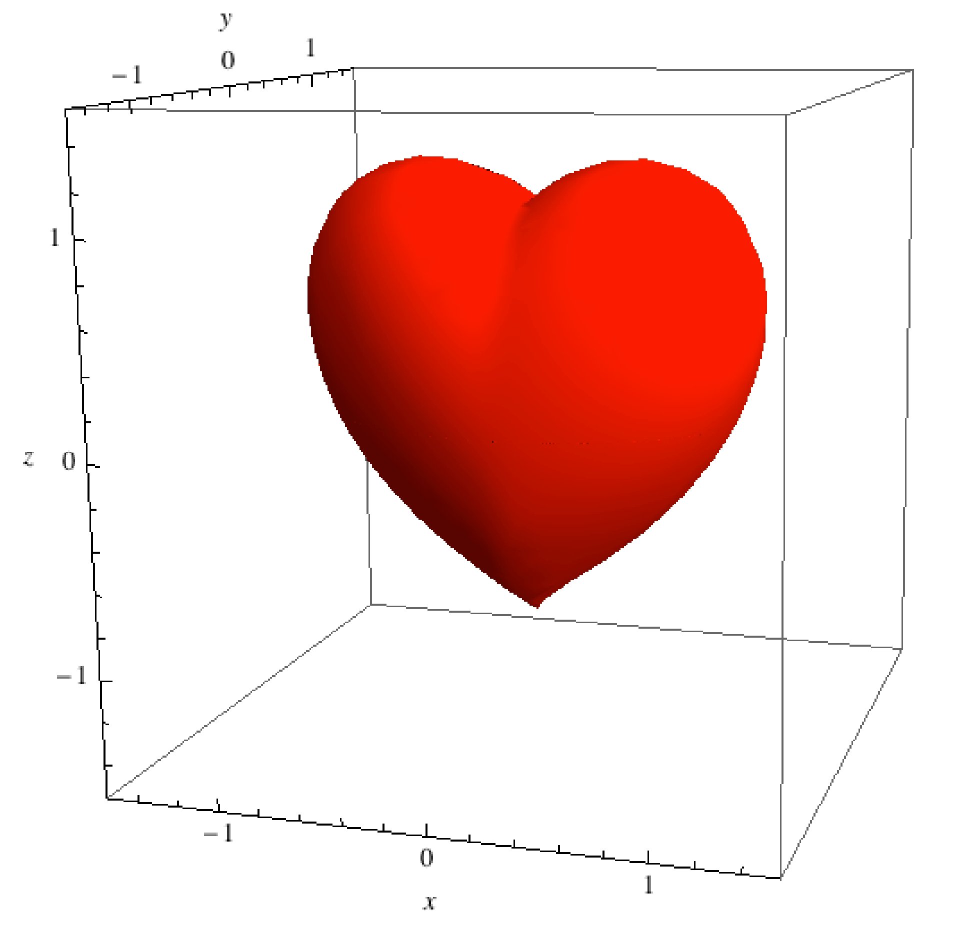
```
import matplotlib.pyplot as plt
import numpy as np
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Make data
u = np.linspace(0, 2 * np.pi, 100)
v = np.linspace(0, np.pi, 100)
x = 10 * np.outer(np.cos(u), np.sin(v))
y = 10 * np.outer(np.sin(u), np.sin(v))
z = 10 * np.outer(np.ones(np.size(u)),
np.cos(v))

# Plot the surface
ax.plot_surface(x, y, z, color='b')
plt.show()
```



Matplotlib Challenge

Reproduce the following 3D Heart Equation

$$(x^2 + \frac{9}{4}y^2 + z^2 - 1)^3 - x^2 z^3 - \frac{9}{200}y^2 z^3 = 0$$



End of Python Matplotlib