Scuola universitaria professionale
della Svizzera italiana
SUPSI

Scientific Computing with Python

Kamini Garg
kamini.garg@supsi.ch

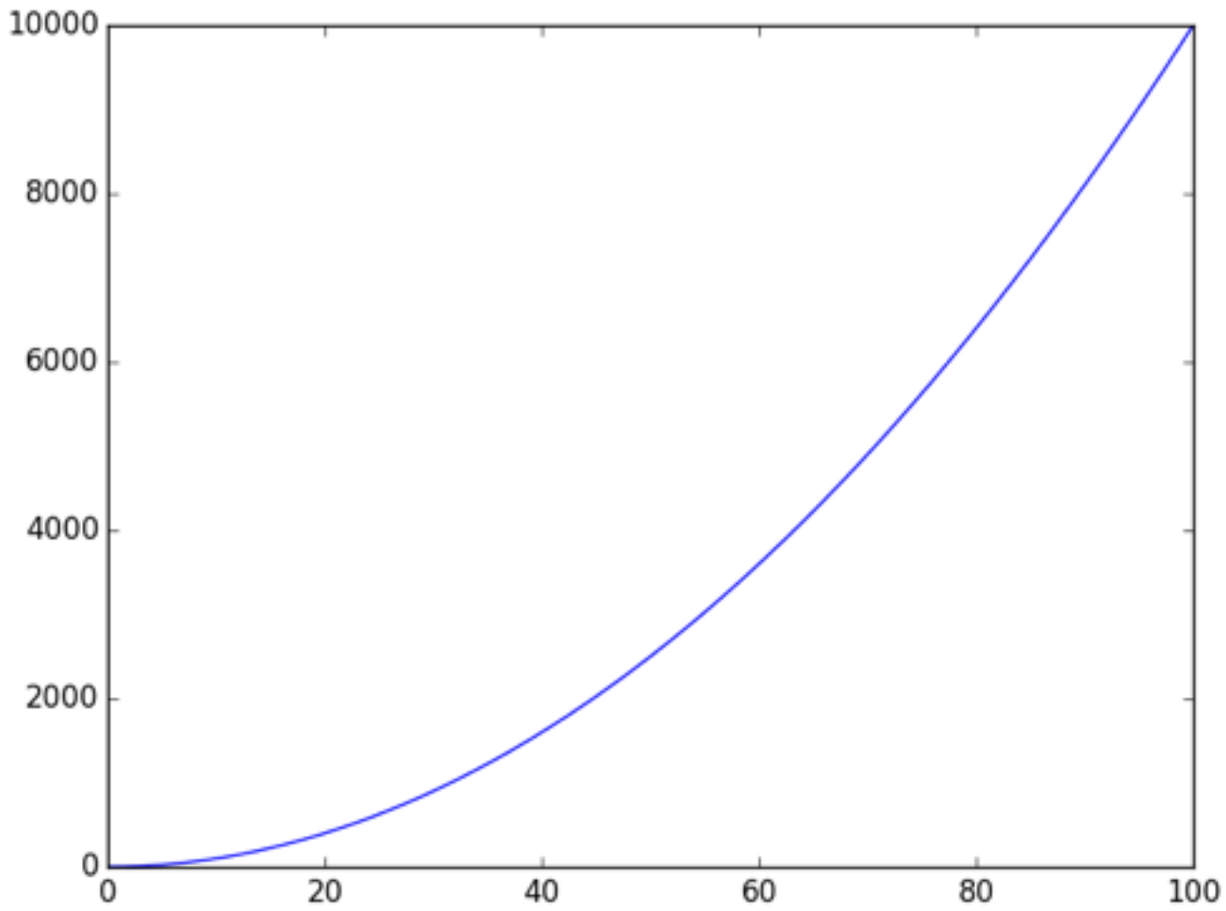
University of Applied Sciences and Arts of Southern Switzerland (SUPSI)

Matplotlib

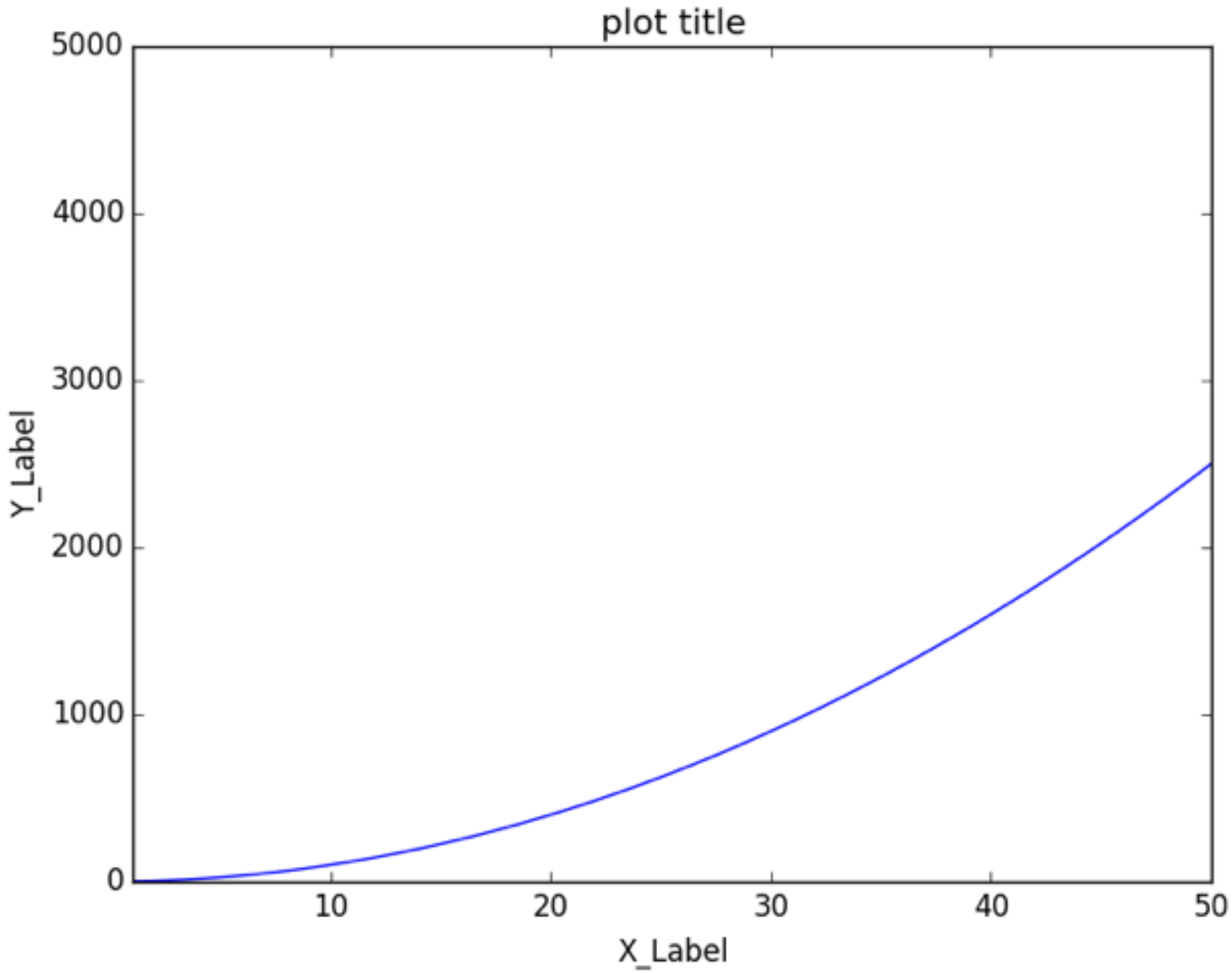
- matplotlib is a python 2D plotting library that produces publication quality figures.
- matplotlib is the standard Python plotting library
- We will focus on matplotlib.pyplot for data analysis
- matplotlib is designed for both interactive and script-based use.

Line plot

```
import numpy as np
import matplotlib . pyplot as plt
x = np. linspace (0, 100, 1000)
y = np. power(x, 2)
plt . plot (x, y)
plt.savefig("lineplot_1.png")
```



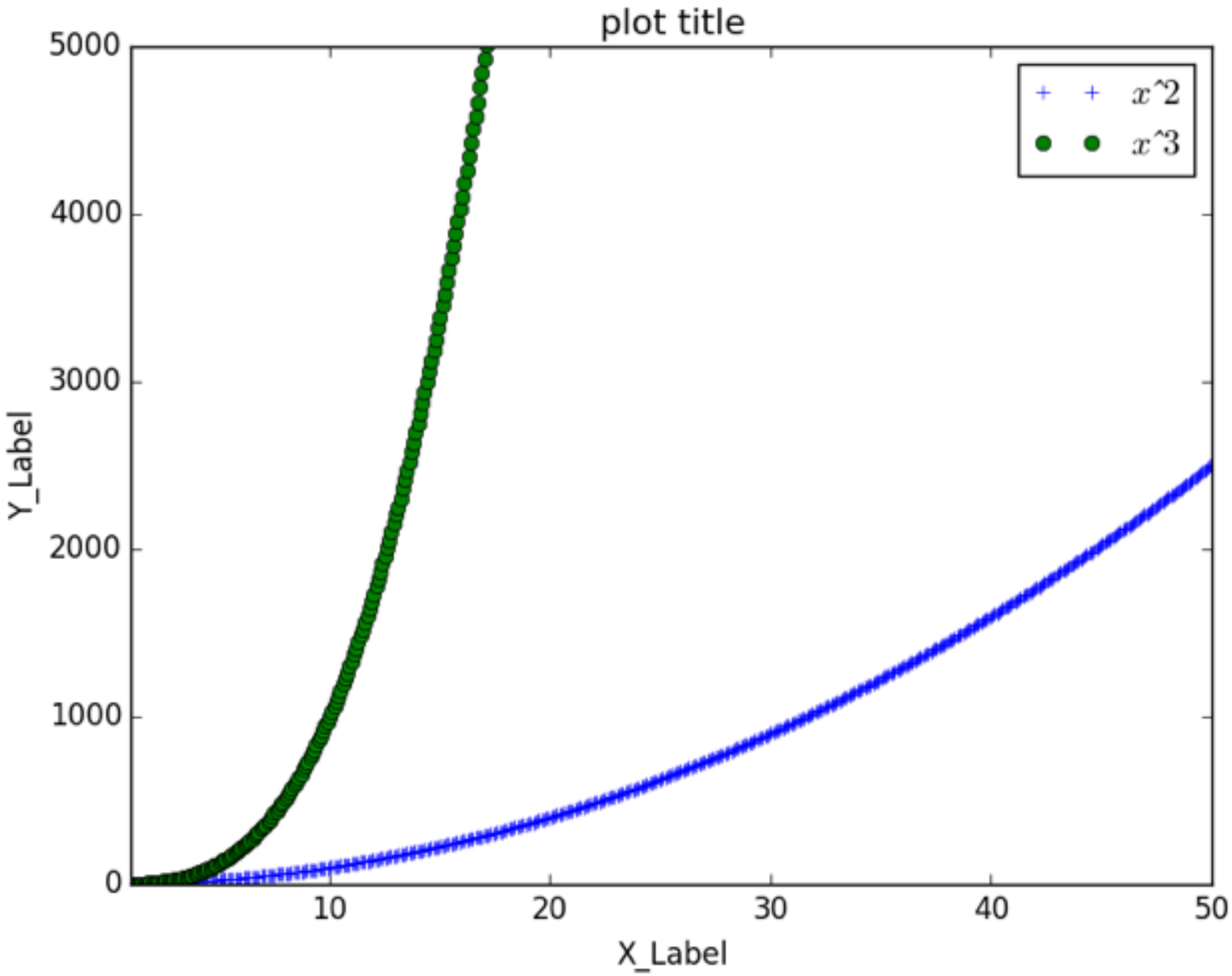
Add some more details



Add some more details

```
import numpy as np
import matplotlib . pyplot as plt
x = np. linspace (0, 100, 1000)
y = np. power(x, 2)
plt . plot (x, y)

# add limits
plt.xlim((1, 50))
plt.ylim((0, 5000))

# add labels and title
plt.xlabel("X_Label")
plt.ylabel("Y_Label")
plt.title("plot title")

plt.savefig("lineplot_2.png")
```

Lets add even more



Lets add even more

```
import numpy as np
import matplotlib . pyplot as plt
x = np. linspace (0, 100, 1000)
y = np. power(x, 2)

# add another data points
z = np. power(x, 3)

# plot both data points together

plt.plot(x, y, 'b-', x, z, 'go')

# add limits
plt.xlim((1, 50))
plt.ylim((0, 5000))

# add labels and title
plt.xlabel("X_Label")
plt.ylabel("Y_Label")
plt.title("plot title")

# add legend
plt.legend(('$x^2$','$x^3$'))


plt.savefig("lineplot_2.png")
```

Controlling line properties

```
# to change line width
plt.plot(x, y, linewidth=2.0)
```

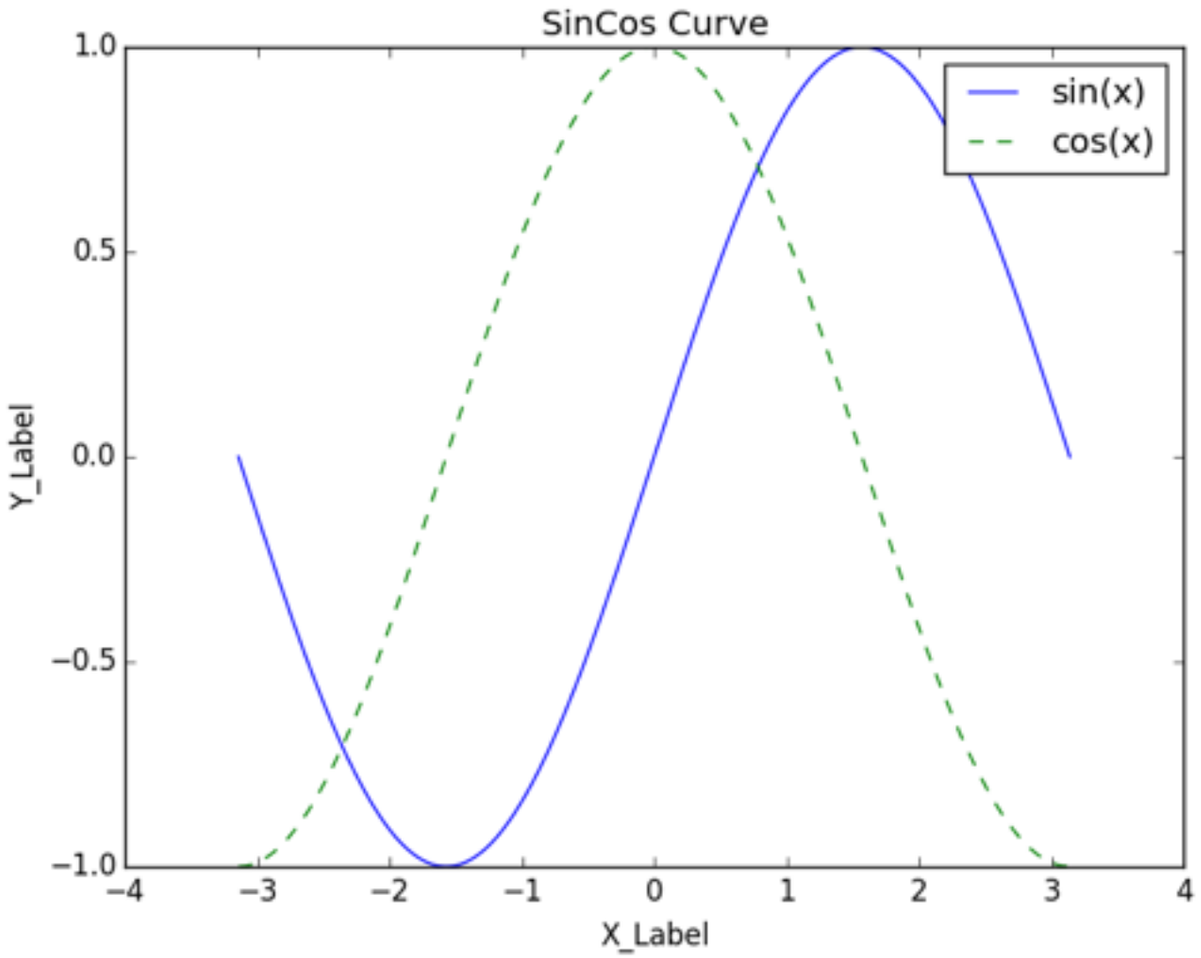
Here are the available `Line2D` properties.

Property	Value Type
alpha	float
animated	[True | False]
antialiased or aa	[True | False]
clip_box	a matplotlib.transform.Bbox instance
clip_on	[True | False]
clip_path	a Path instance and a Transform instance, a Patch
color or c	any matplotlib color
contains	the hit testing function
dash_capstyle	`['butt' | 'round' | 'projecting']`
dash_joinstyle	`['miter' | 'round' | 'bevel']`
dashes	sequence of on/off ink in points
data	(np.array xdata, np.array ydata)
figure	a matplotlib.figure.Figure instance
label	any string
linestyle or ls	`[ '-' | '--' | '-.' | ':' | 'steps' | ...]`
linewidth or lw	float value in points
lod	[True | False]
marker	`[ '+' | ',' | '.' | '1' | '2' | '3' | '4' ]`
markeredgecolor or mec	any matplotlib color
markeredgewidth or mew	float value in points
markerfacecolor or mfc	any matplotlib color
markersize or ms	float

For more information : http://matplotlib.org/users/pyplot_tutorial.html

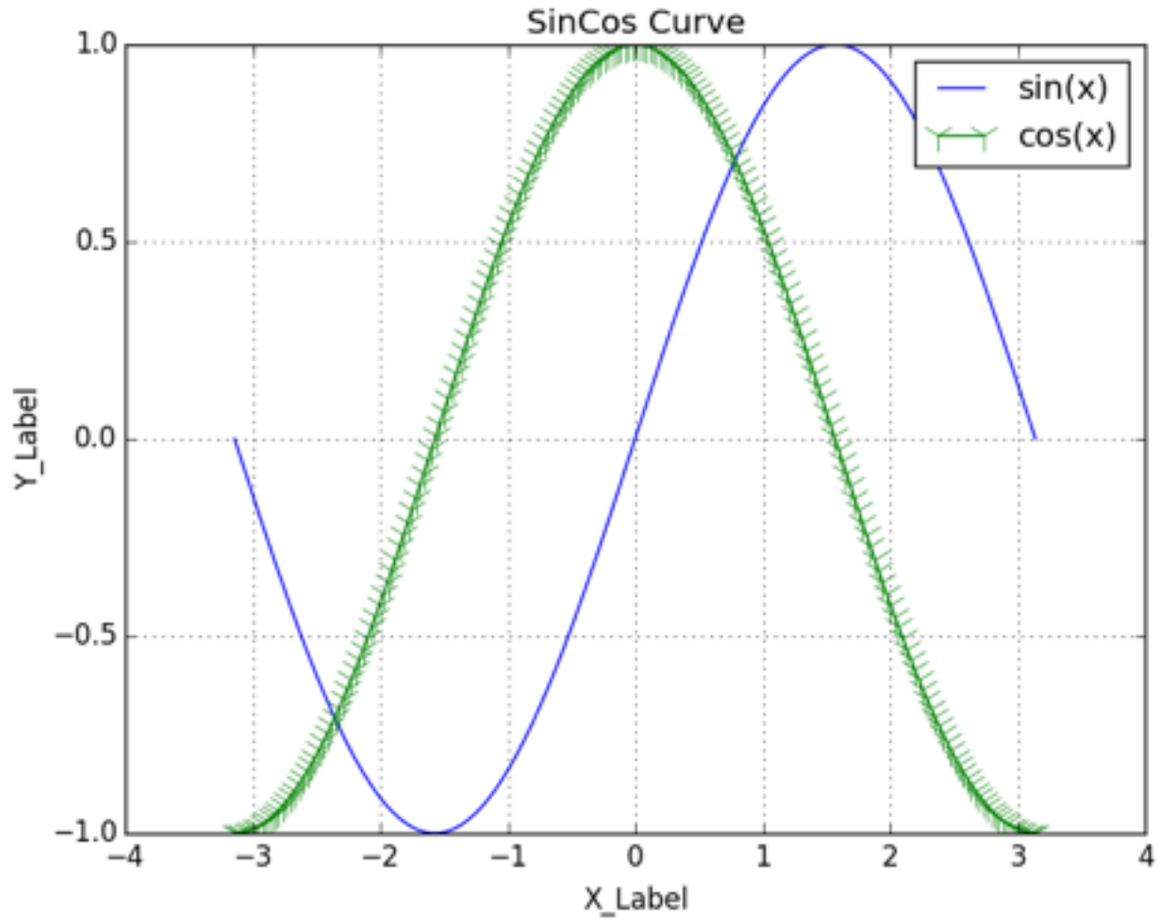
Assignment 7(a) (20 mins)

1. Plot the following sincos curve with 200 data points between -pi to +pi (3pts)



Assignment 7(a) (20 mins)

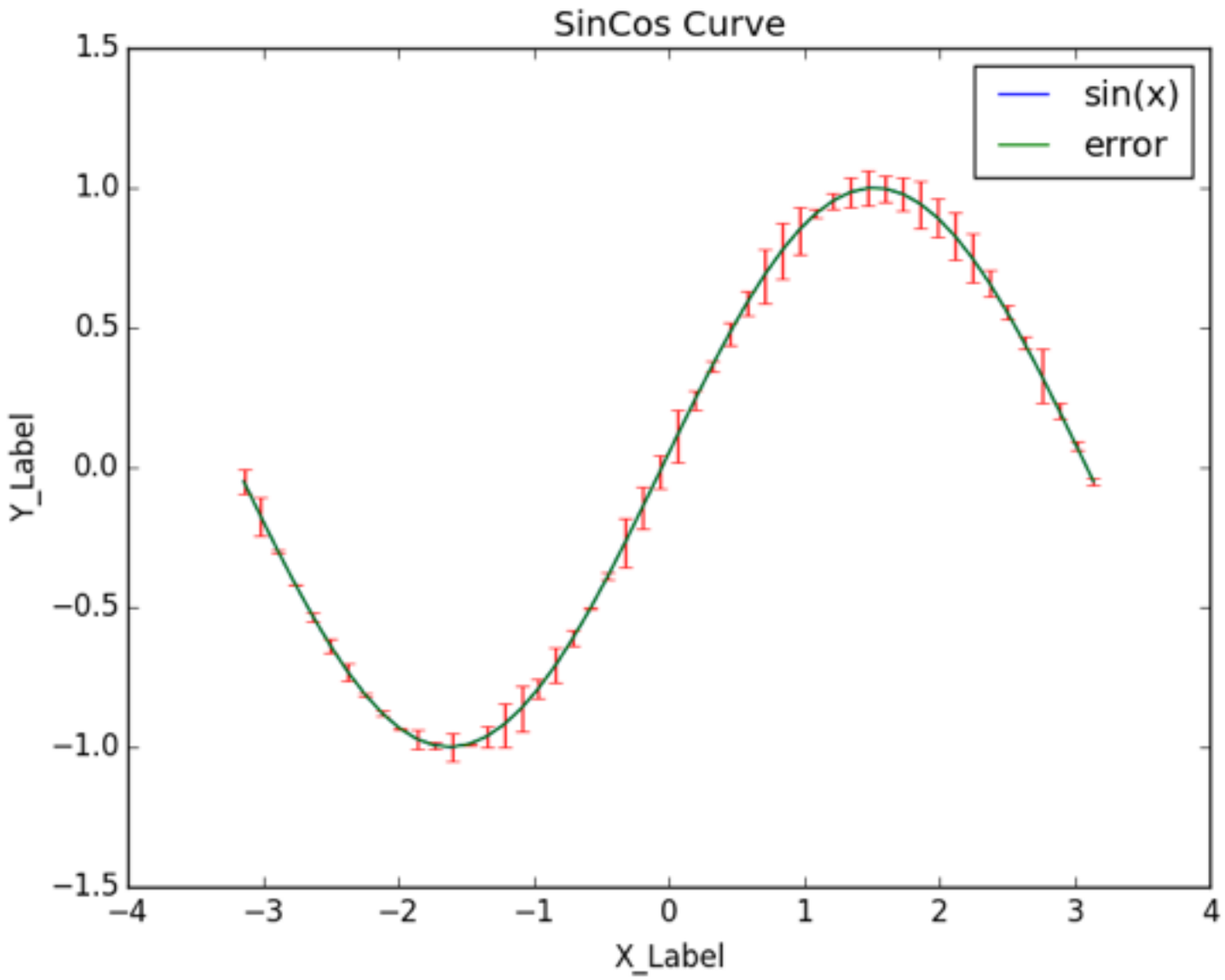
2. Try to change the plot as follows (2pt)



1. File name : **as7a_yoursurname_name.py**
2. Upload your solution at Moodle under A07 folder after completion.

Error bars

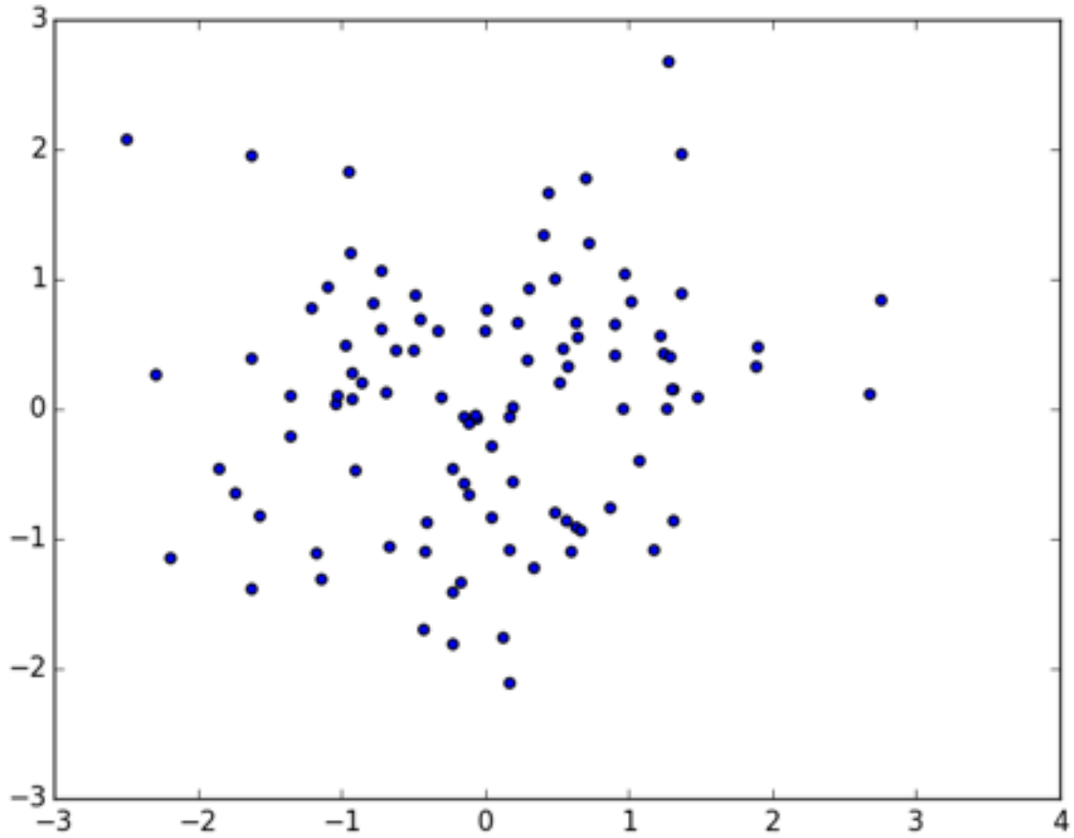
```
error = 0.1*np.random.random(len(x))
plt.errorbar(x,y,error,ecolor='r')
```



Scatter Plots

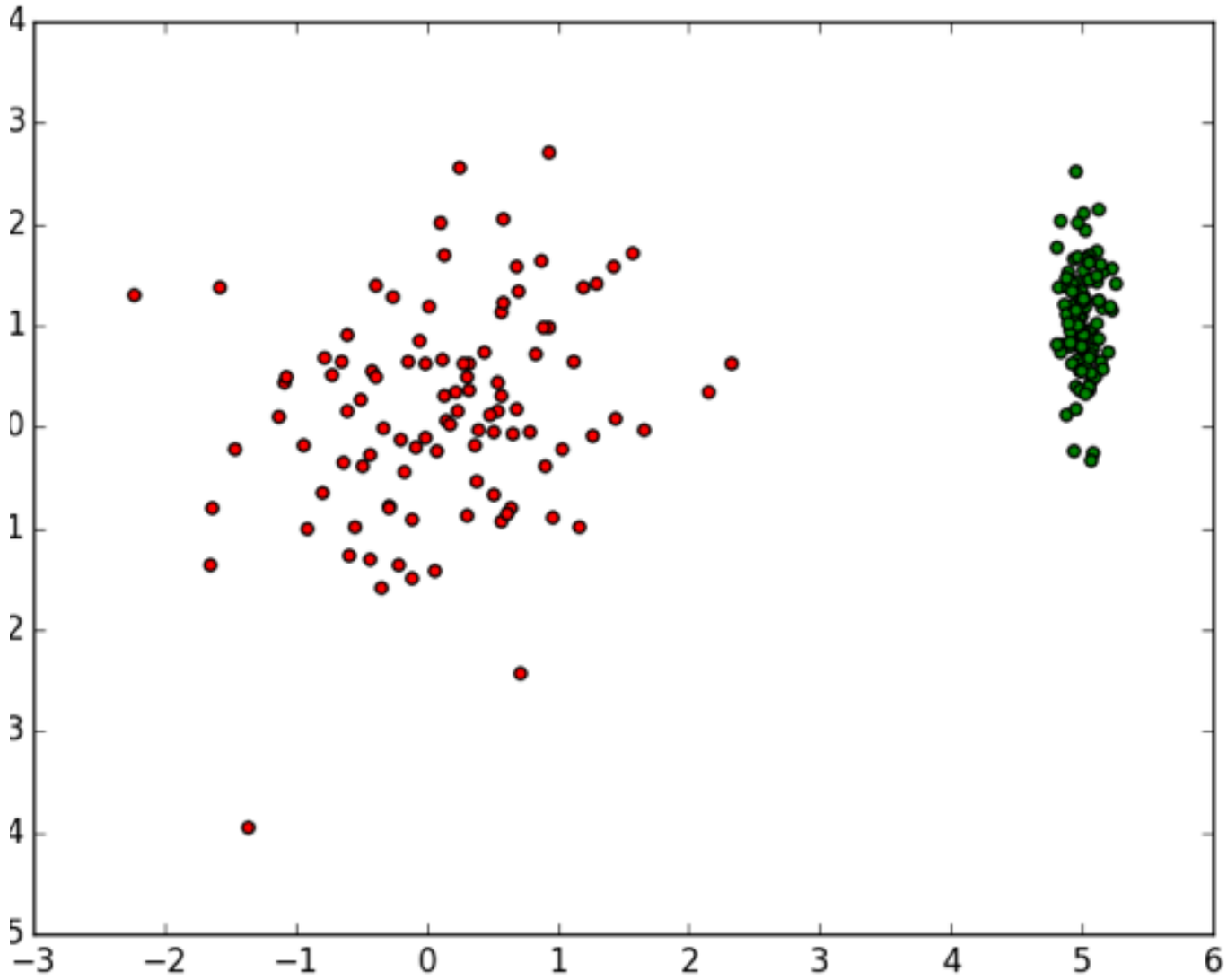
A scatter plot just shows one point for each dataset entry

```
# generate points from normal distribution

npoints = 100
x = np.random.standard_normal(npoints)
y = np.random.standard_normal(npoints)
plt.scatter(x,y)
plt.savefig("scatter_1.png")
```



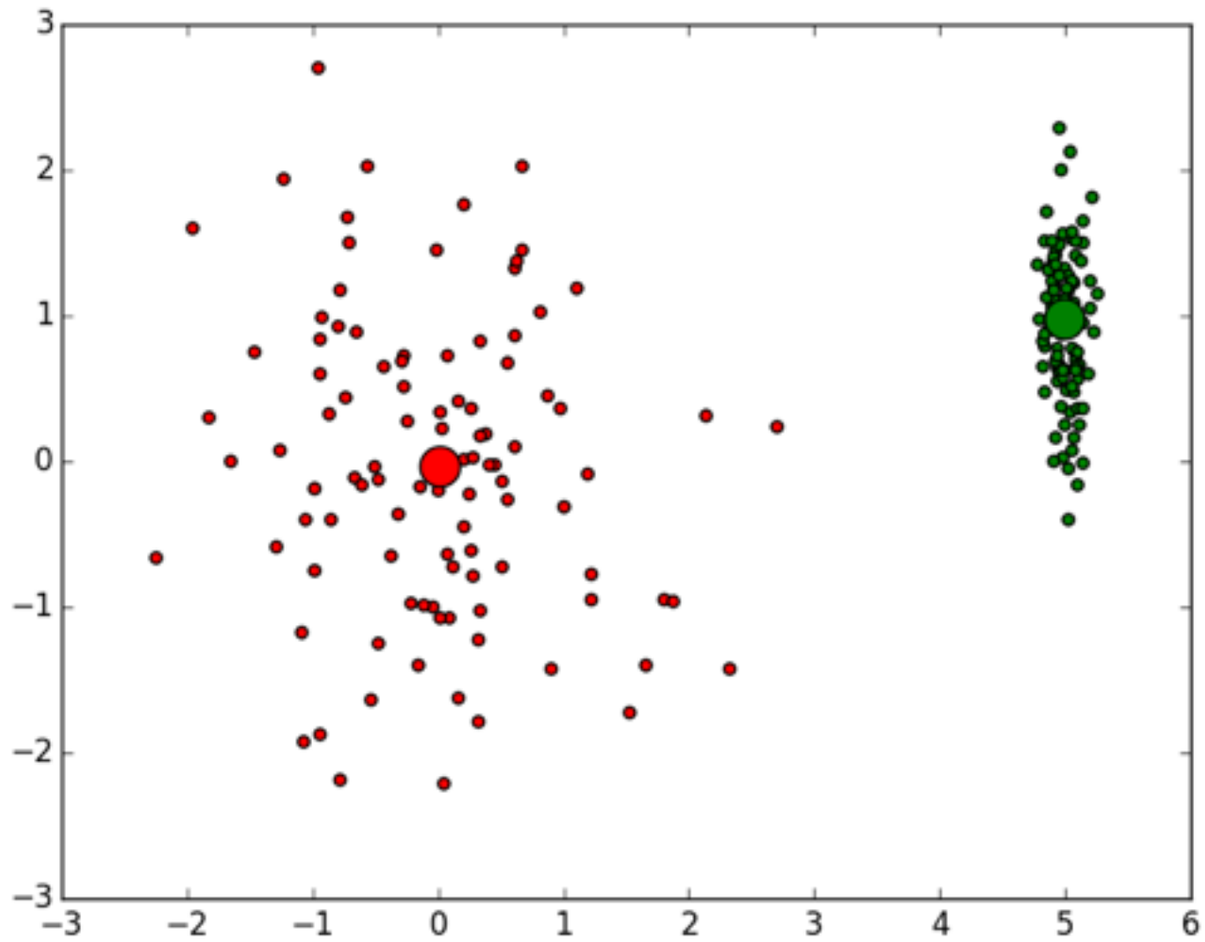
Assignment 7(b) (20 mins)

1. Plot the following scatter plot (3pts)



Assignment 7(b) (20 mins)

1. Plot the centre of data points as follows (2pts) Hint: you can use median

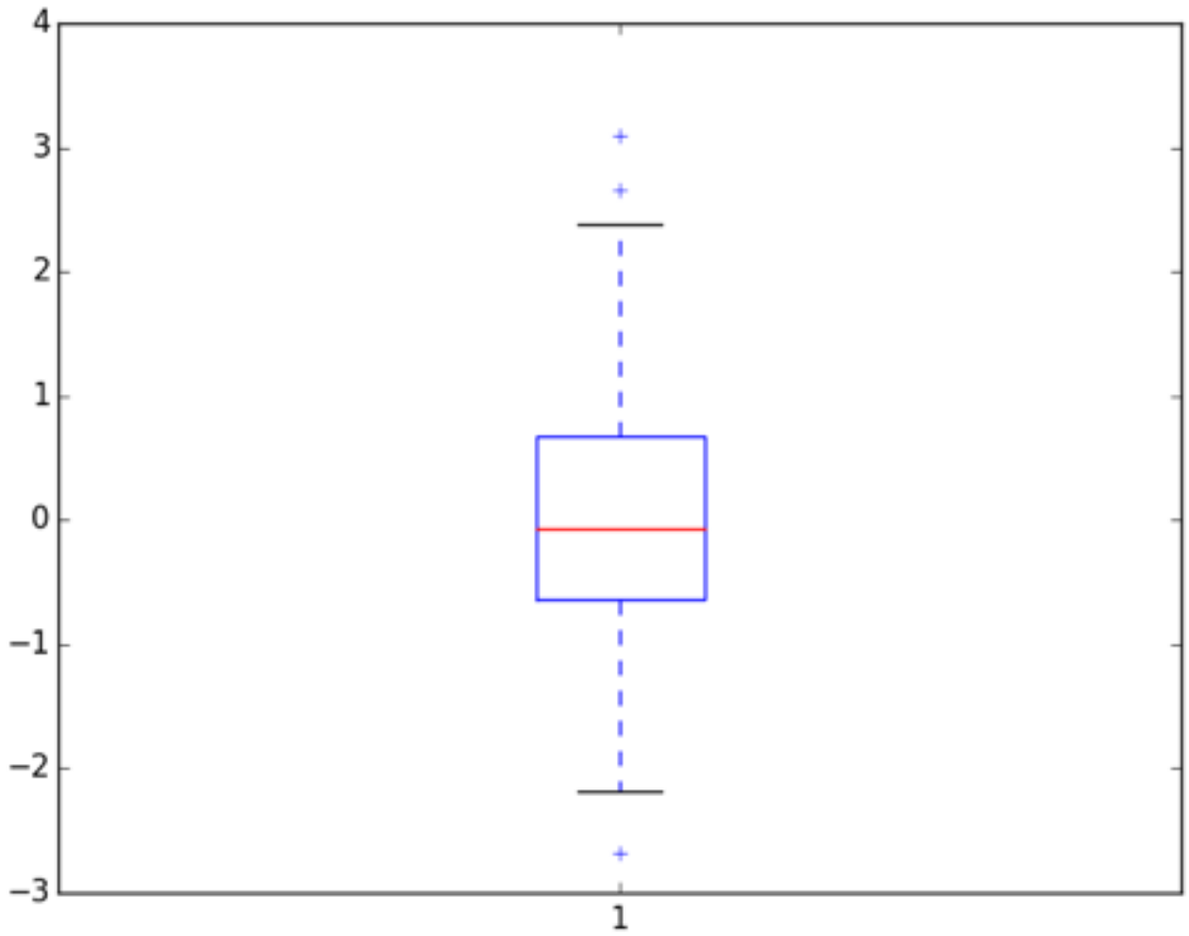


1. File name : **as7b_yoursurname_name.py**
2. Upload your solution at Moodle under A07 folder after completion.

Box Plots

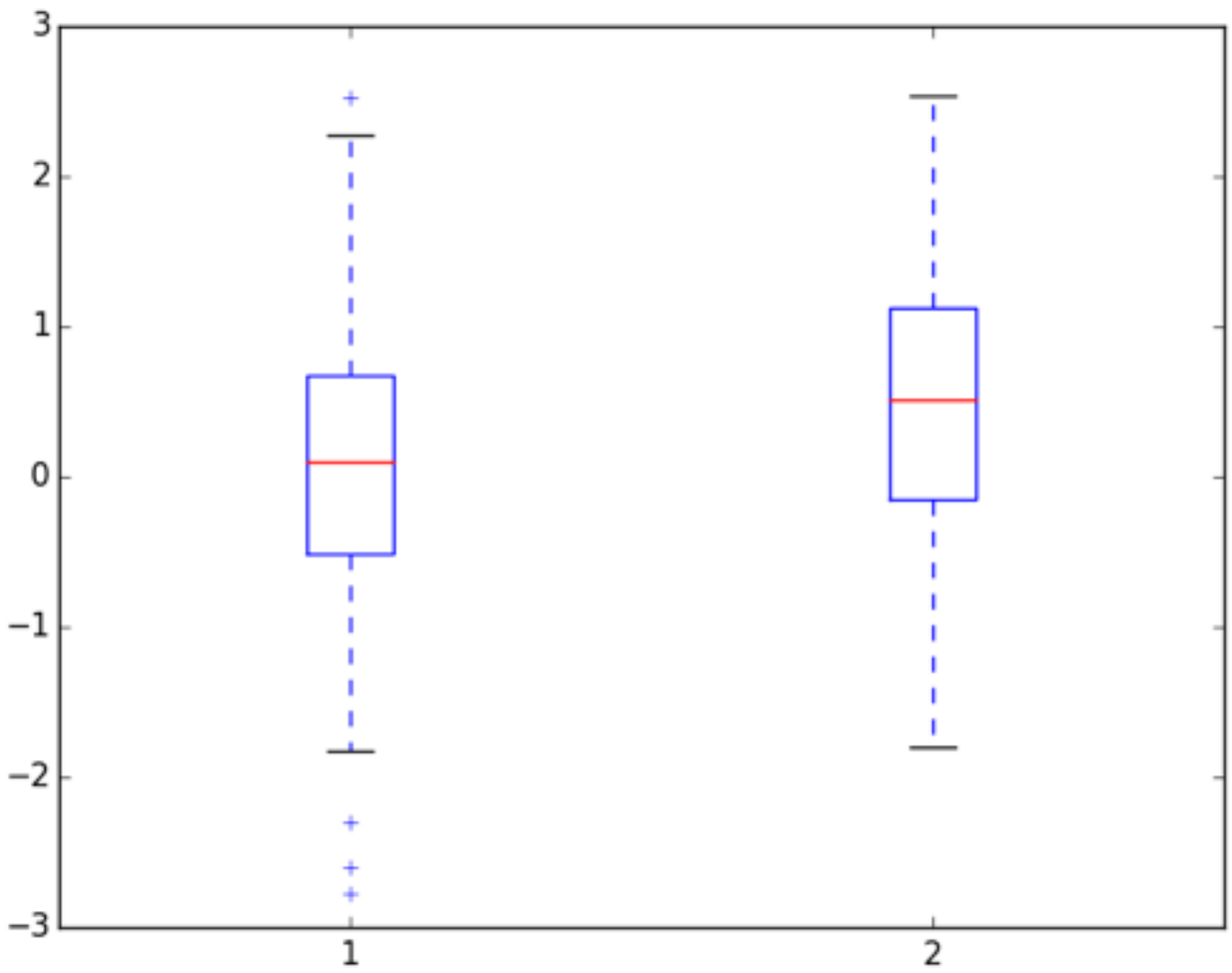
Boxplot is a convenient way of graphically depicting groups of numerical data through their quartiles.

```
# generate points from normal distribution

npoints = 200
x = np.random.standard_normal(npoints)
plt.boxplot(x)
plt.savefig("box_1.png")
```



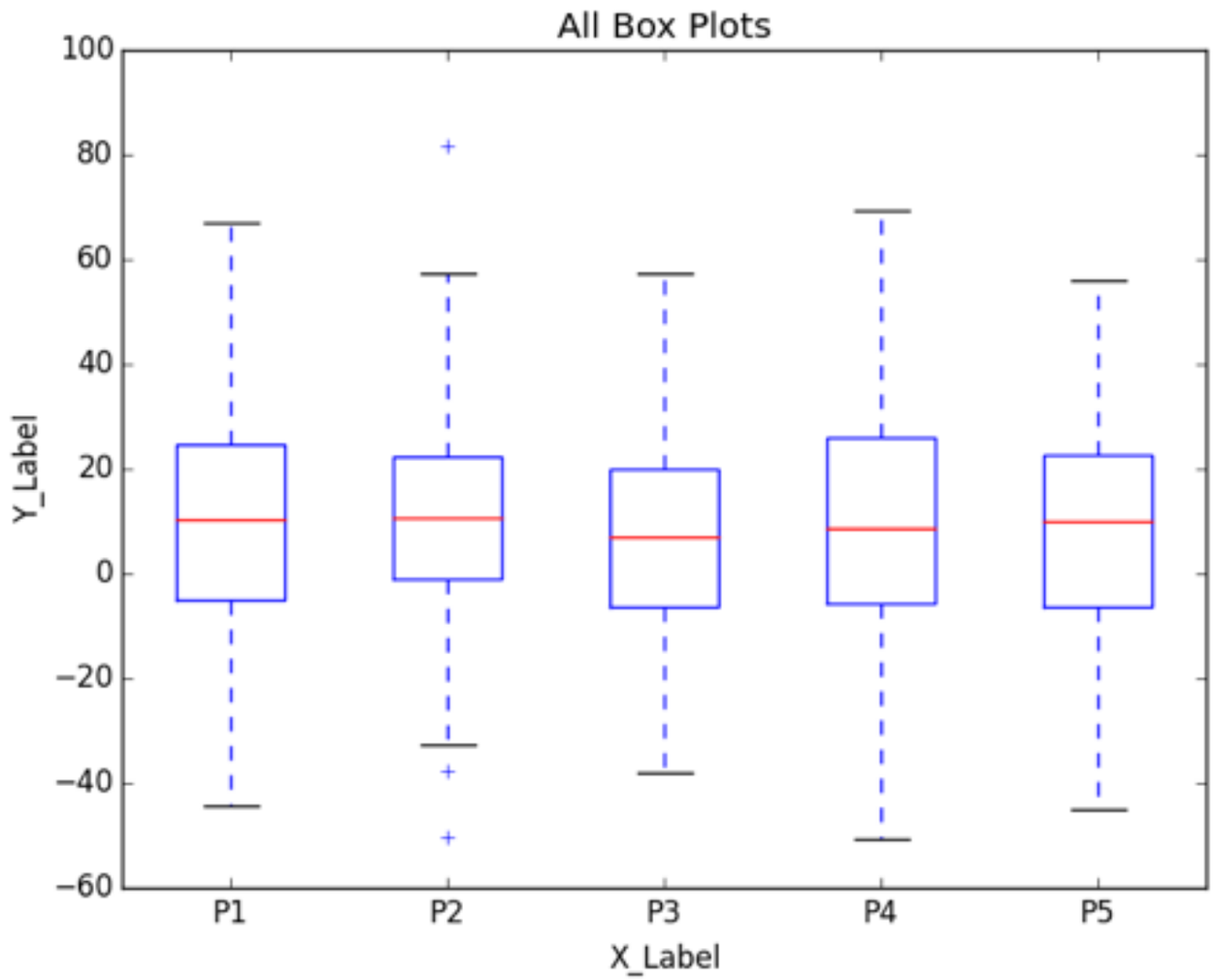
Multiple Box Plots

```
# generate points from normal distribution

npoints = 200
x = np.random.standard_normal(npoints)
y = 0.5 + np.random.standard_normal(npoints)
plt.boxplot((x,y))
```



Assignment 7(c) (15mins)

1. Plot the following box plot (2pts)

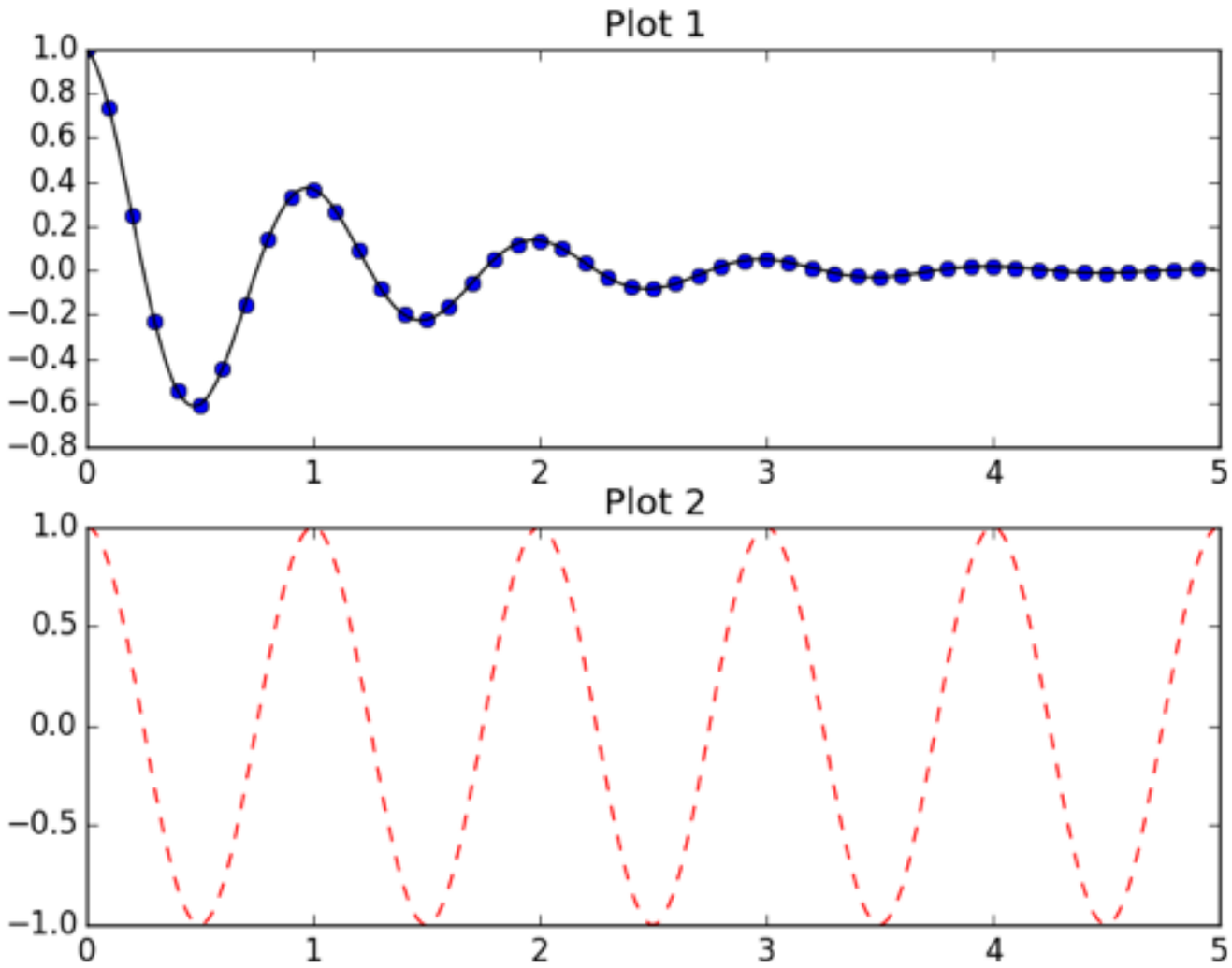


1. File name : **as7c_yoursurname_name.py**
2. Upload your solution at Moodle under A07 folder after completion.

Working with multiple figures and axes

Similar to MATLAB, pyplot also have the concept of the current figure and the current axes.

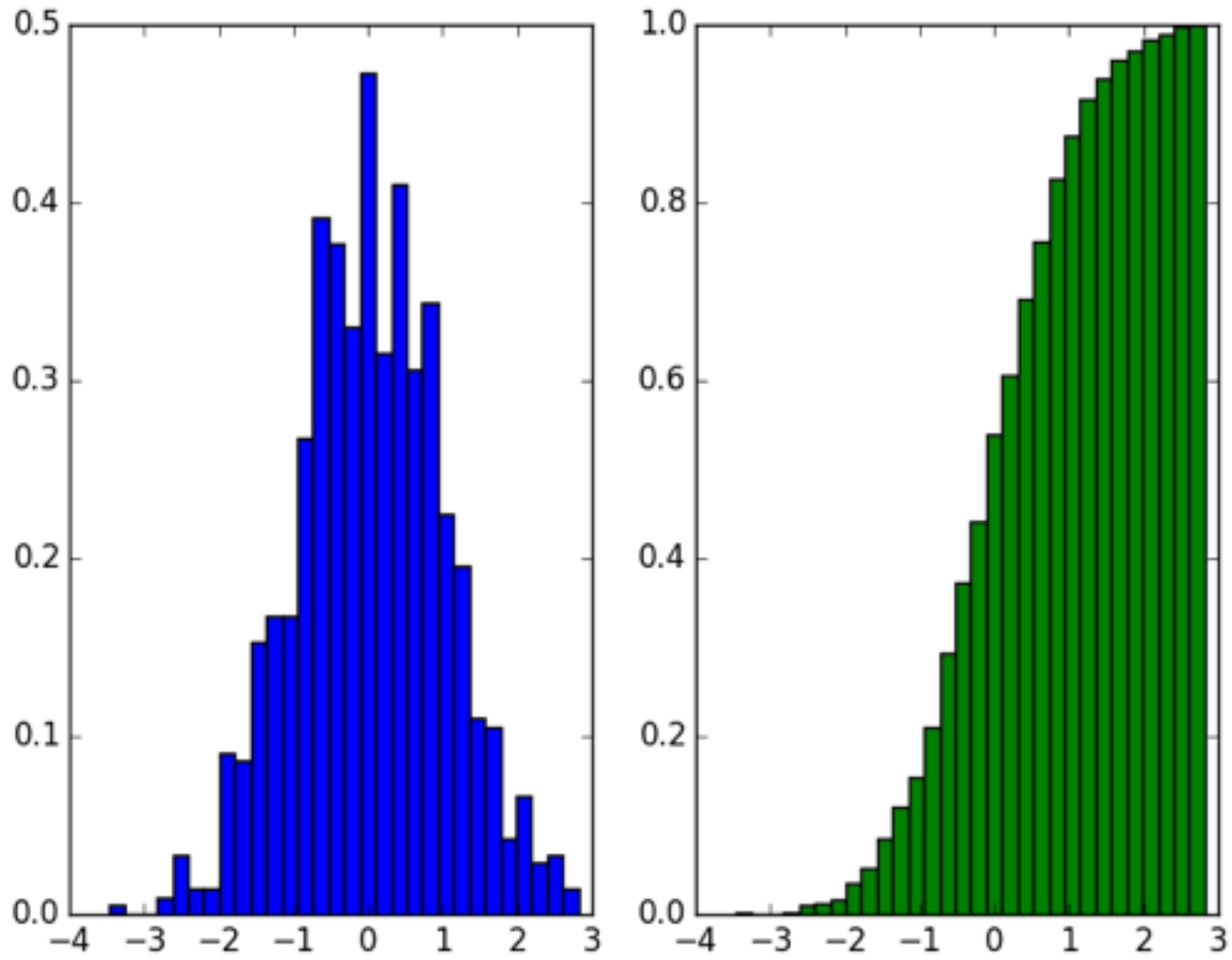
```
def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

# create the canvas for figure
fig = plt.figure()

#subplot(numrows, numcols, fignum)
axis1 = fig.add_subplot(211)
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')
axis1.set_title("Plot 1")

axis2 = fig.add_subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')
axis2.set_title("Plot 2")
plt.savefig("multiple_plots.png")
```



Histograms

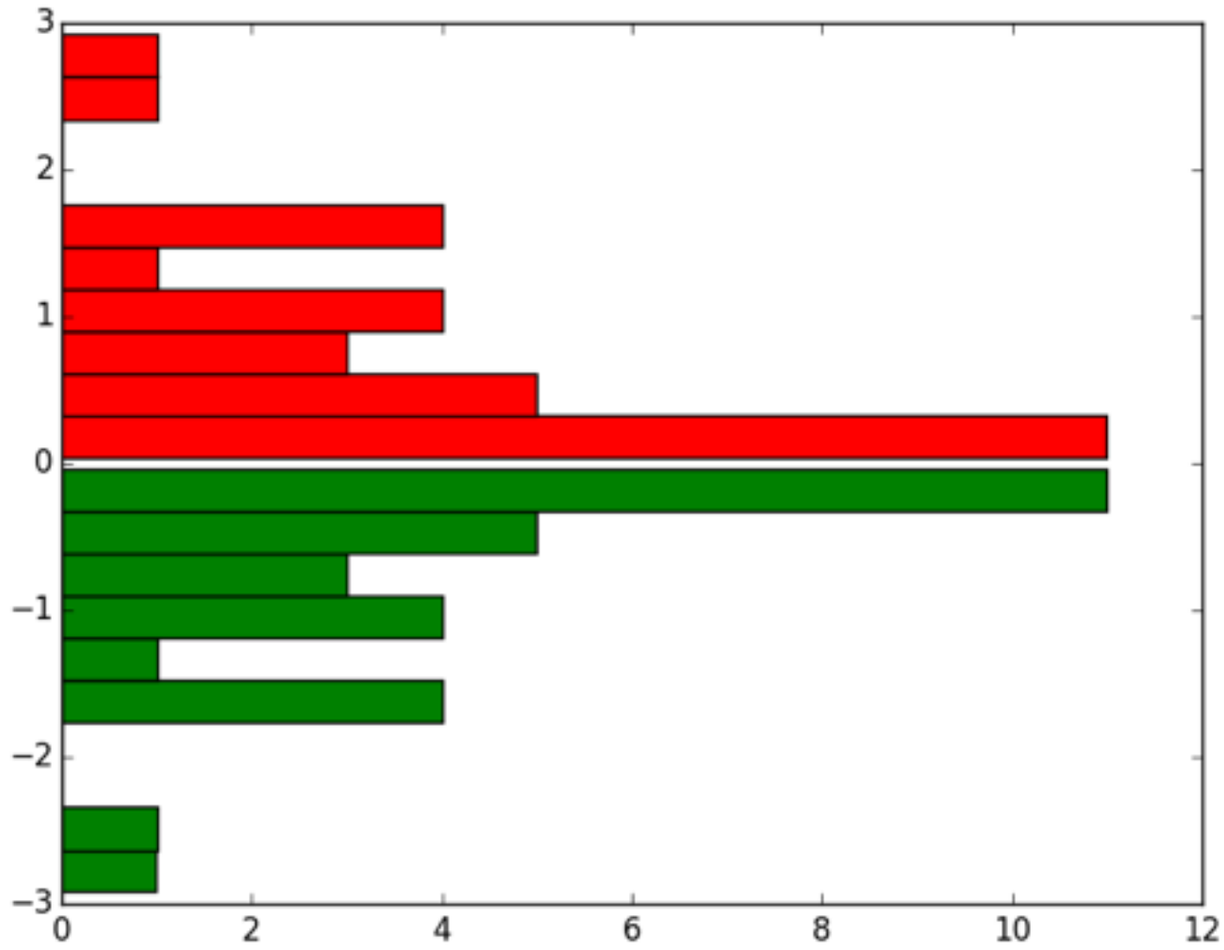
Histogram are convenient to sum-up results

```
data = np. random. randn(1000)
# histogram (pdf)
plt . subplot (1, 2, 1)
plt . hist (data , bins=30, normed=True, facecolor='b')
# empirical cdf
plt . subplot (1, 2, 2)
plt . hist (data , bins=30, normed=True,
color='g',cumulative=True)
plt . savefig ('histogram.png')
```



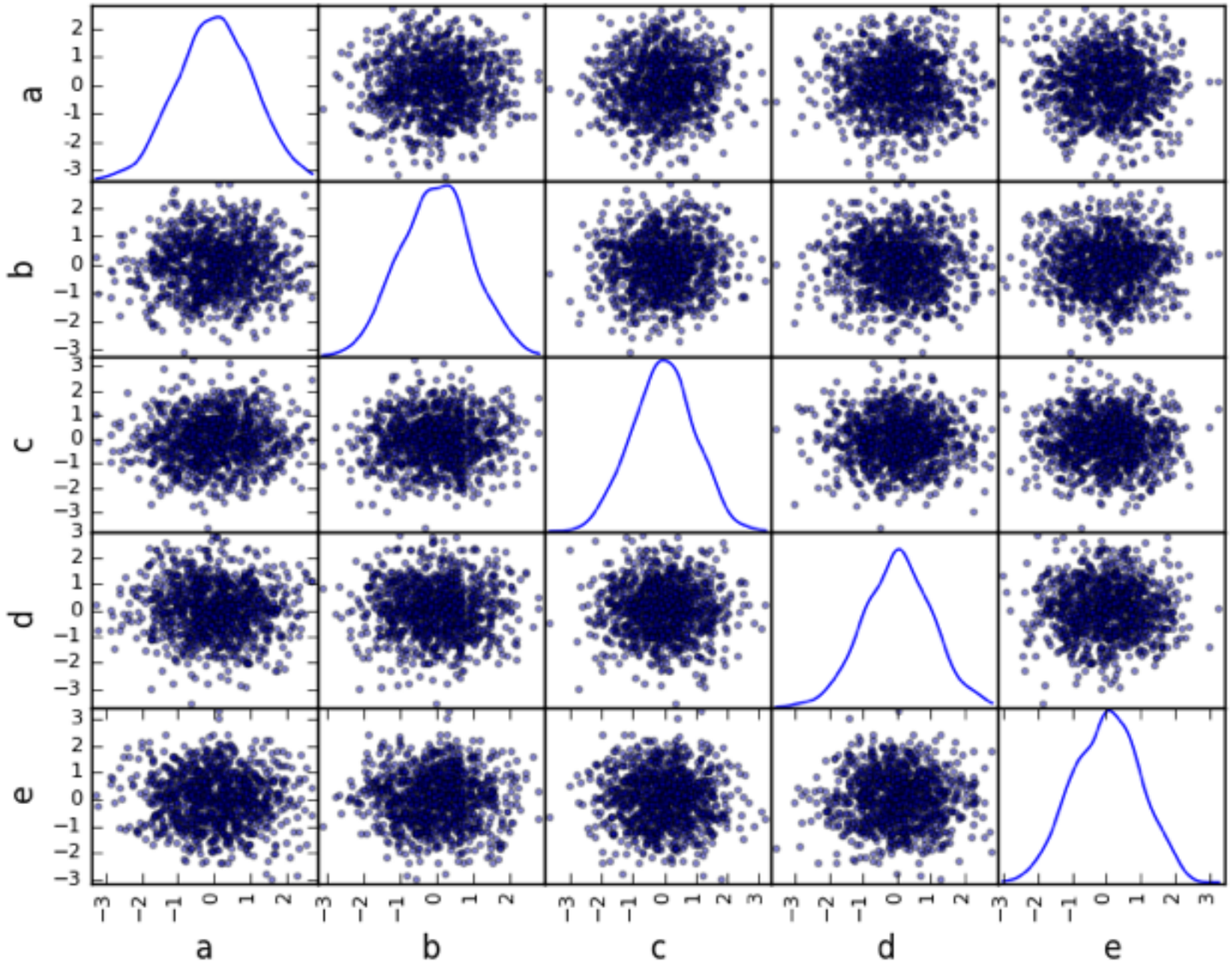
Assignment 7(d) (15 mins)

1. Plot the following histogram, generate data using np.abs(np.random.standard_normal(30)) (5pts)



1. File name : **as7d_yoursurname_name.py**
2. Upload your solution at Moodle under A07 folder after completion.

Scatter Matrix



Scatter Matrix

matplotlib doesn't have everything, especially functions that are designed to act on more than one axis at once.

Pandas come for your rescue :)

```
from pandas.tools.plotting import scatter_matrix
from pandas import DataFrame
df=DataFrame(np.random.normal(loc=0.,scale=1.,size=(1000,
5)),columns=['a', 'b', 'c', 'd', 'e'])
scatter_matrix(df, alpha=0.4, diagonal='kde')
plt.savefig('scattermatrix.png')
```

Try to draw the same thing with histogram in the diagonal!!

Pie Chart

```
n = 20
Z = np.random.uniform(0,1,n)
plt.pie(Z)
plt.savefig('pie.png')
```

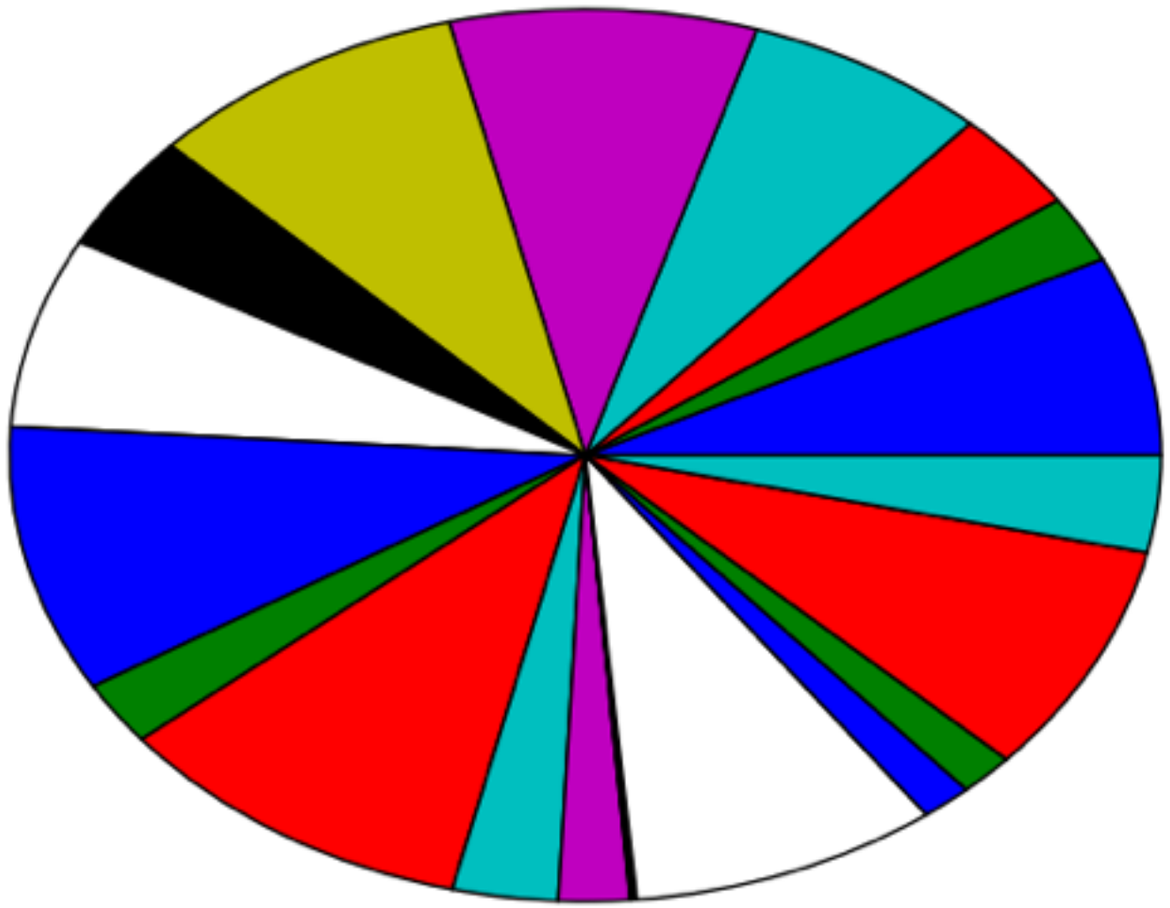
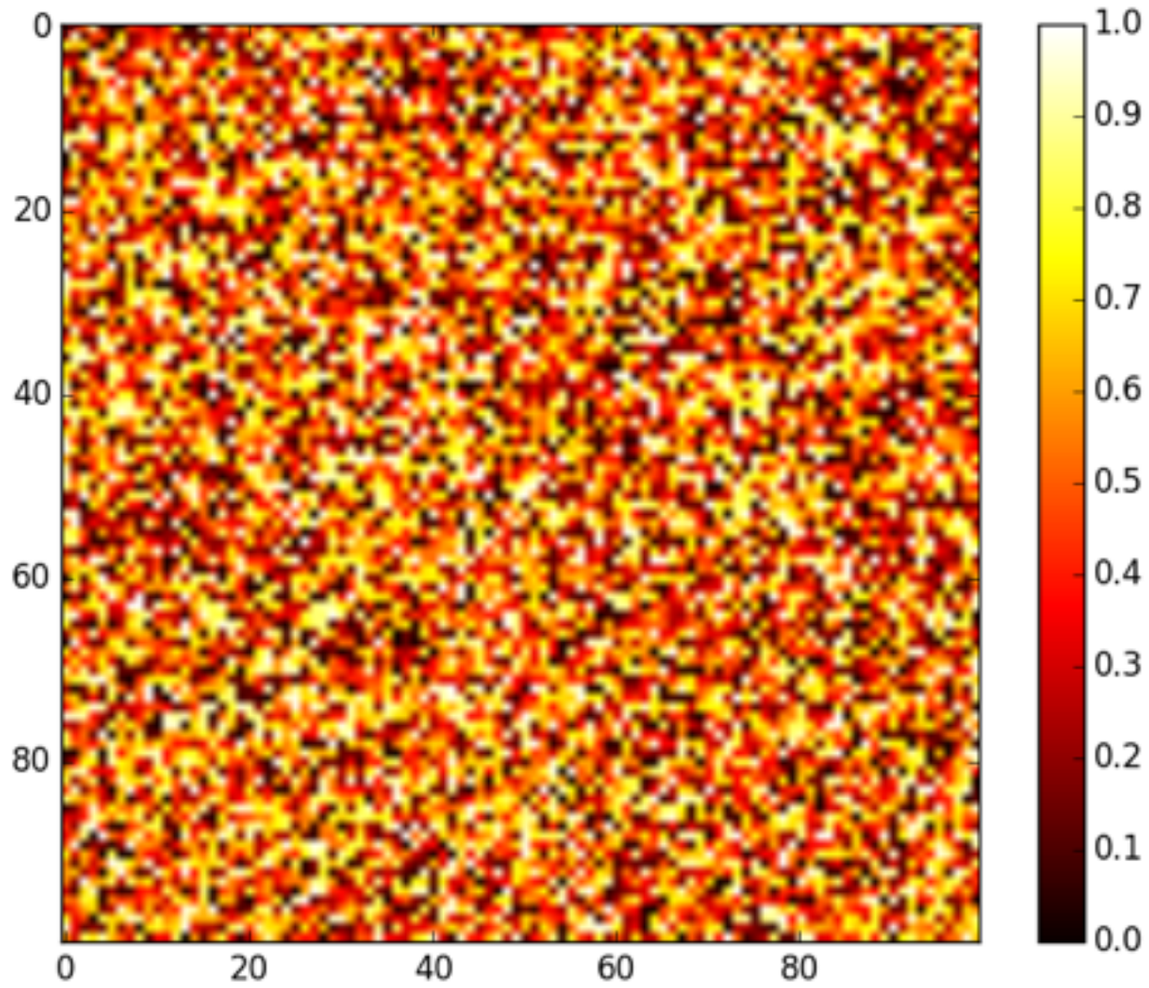


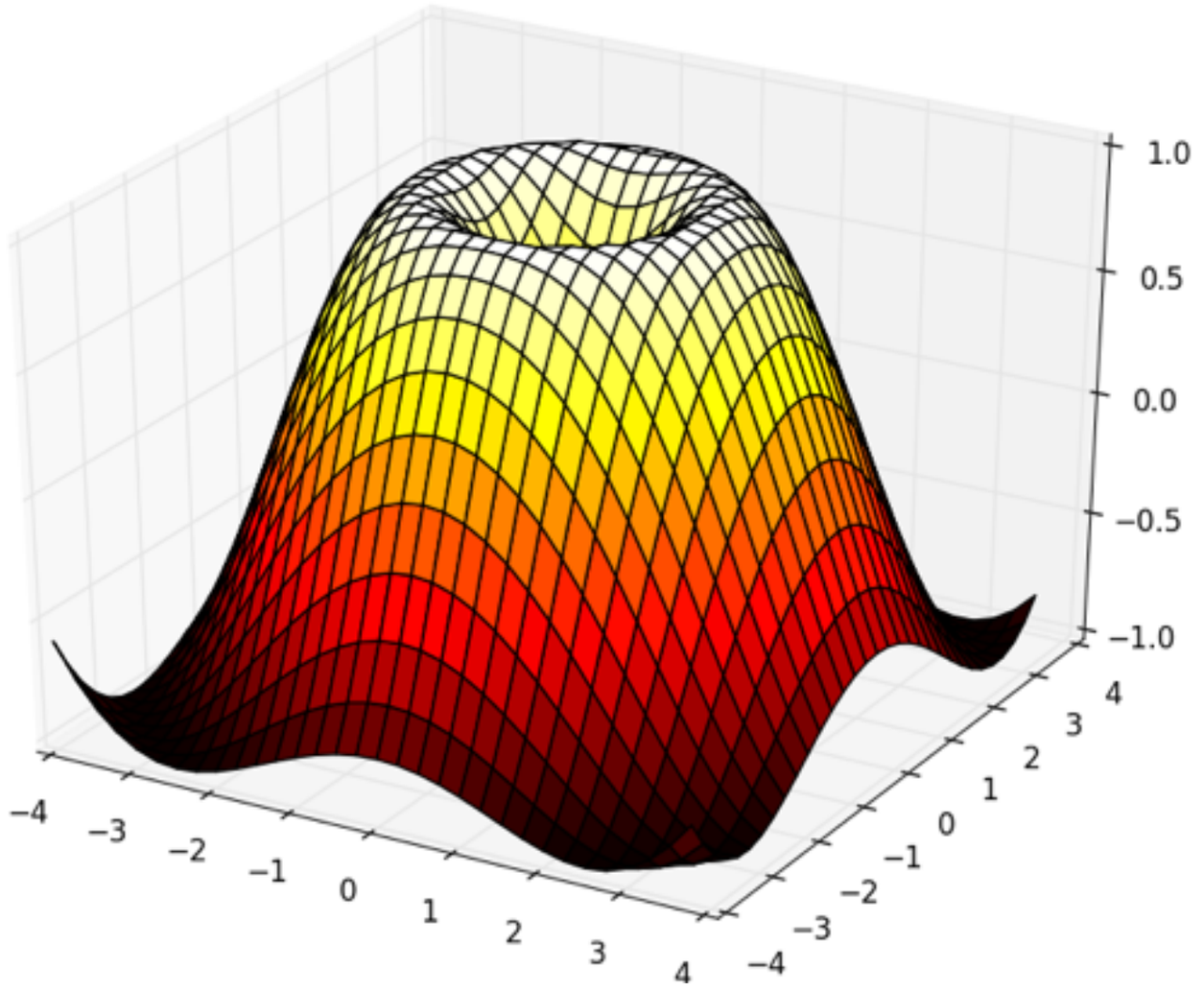
Image Plot

```
A = np.random.random((100, 100)
plt.imshow(A)
plt.hot()
plt.colorbar()
plt.savefig('image.png')
```



3D-Plot

```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = Axes3D(fig)
X = np.arange(-4, 4, 0.25)
Y = np.arange(-4, 4, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
Z = np.sin(R)

ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
cmap='hot')

plt.savefig('3d.png')
```

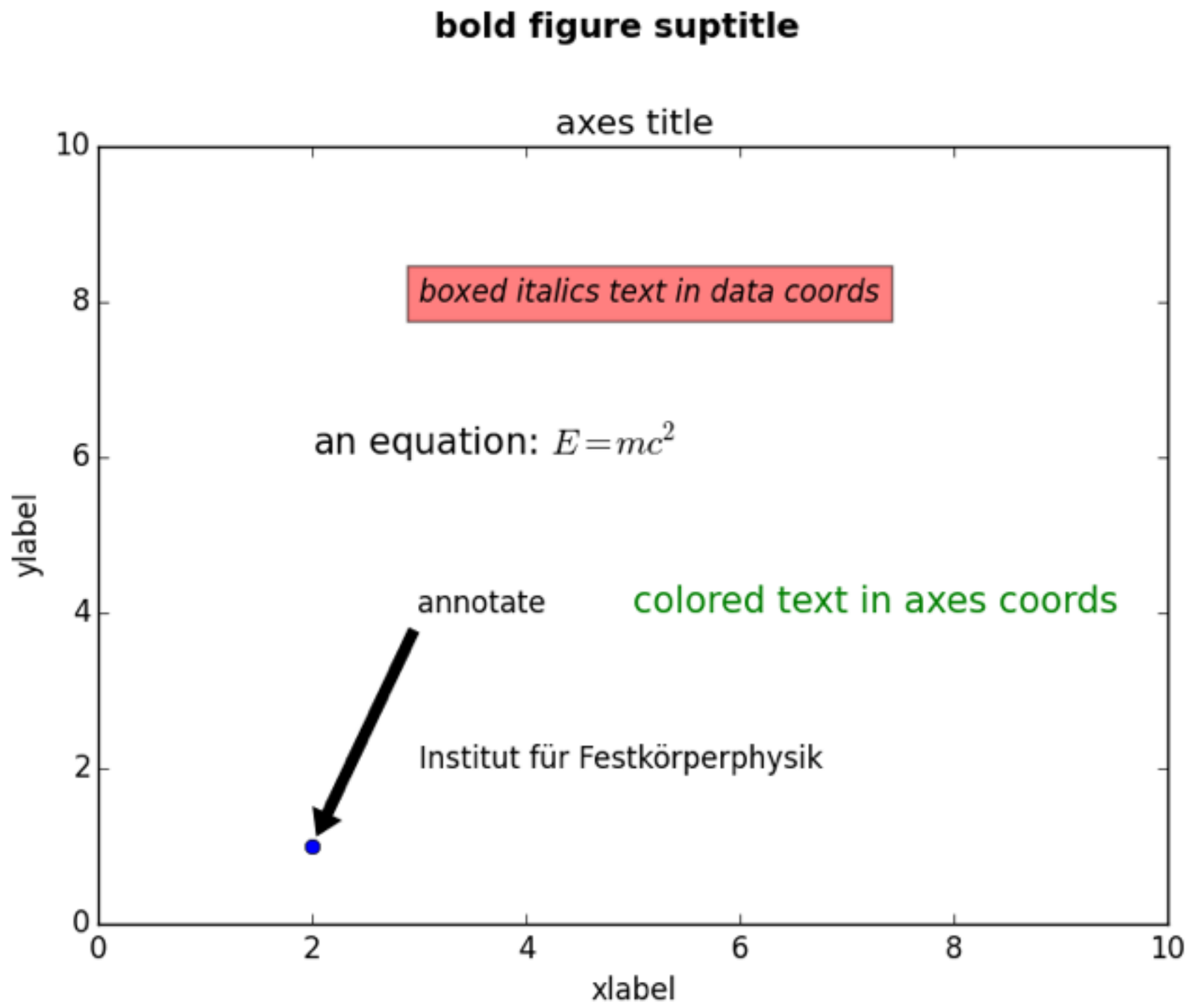


Assignment 7(e) (25 mins)

1. Download the file “bill_of_rights.txt” from A07 folder, find 10 most and least frequent words and plot them. (5pts)

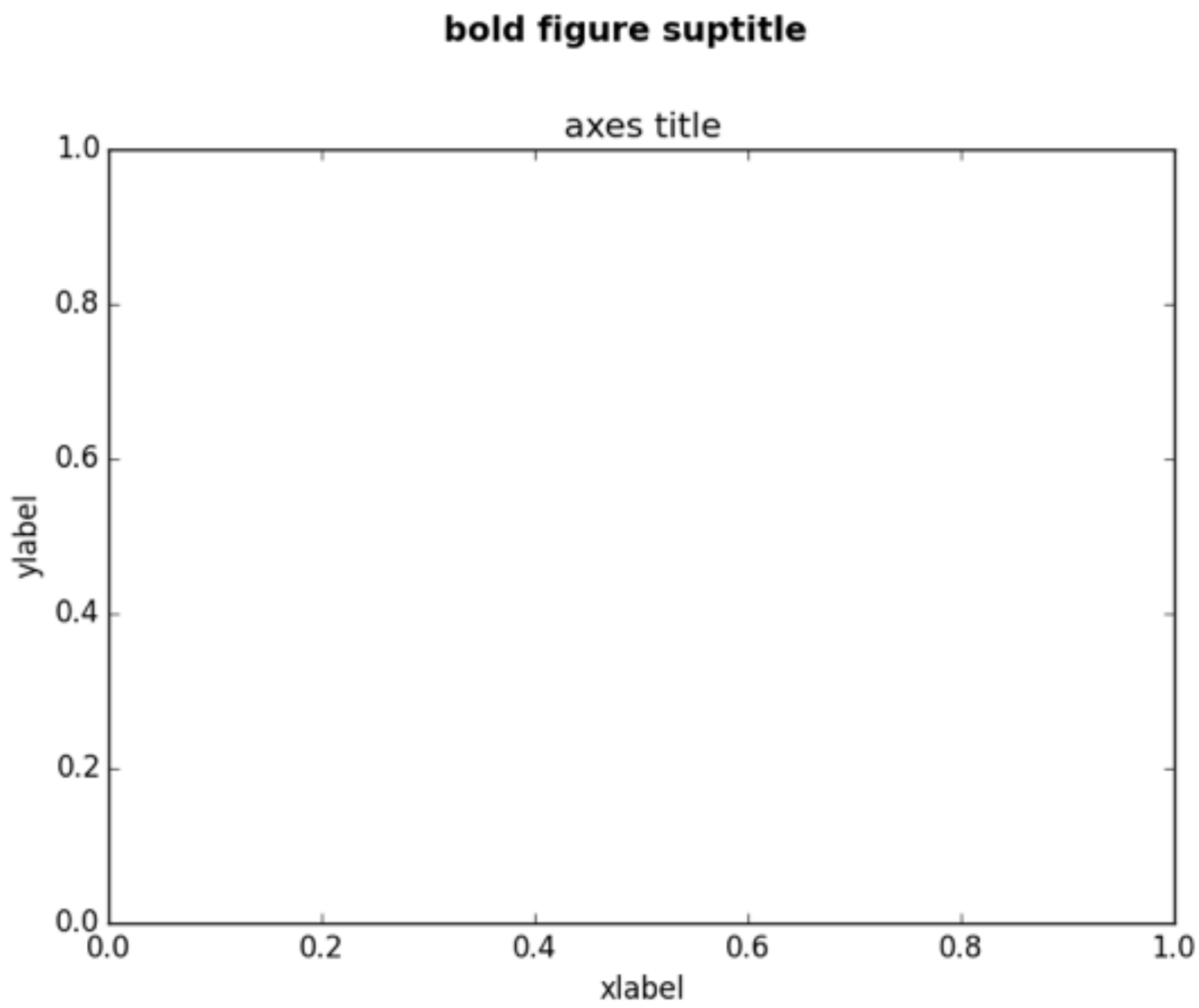
1. File name : **as7e_yoursurname_name.py**
2. Upload your solution at Moodle under A07 folder after completion.

Text on Plots



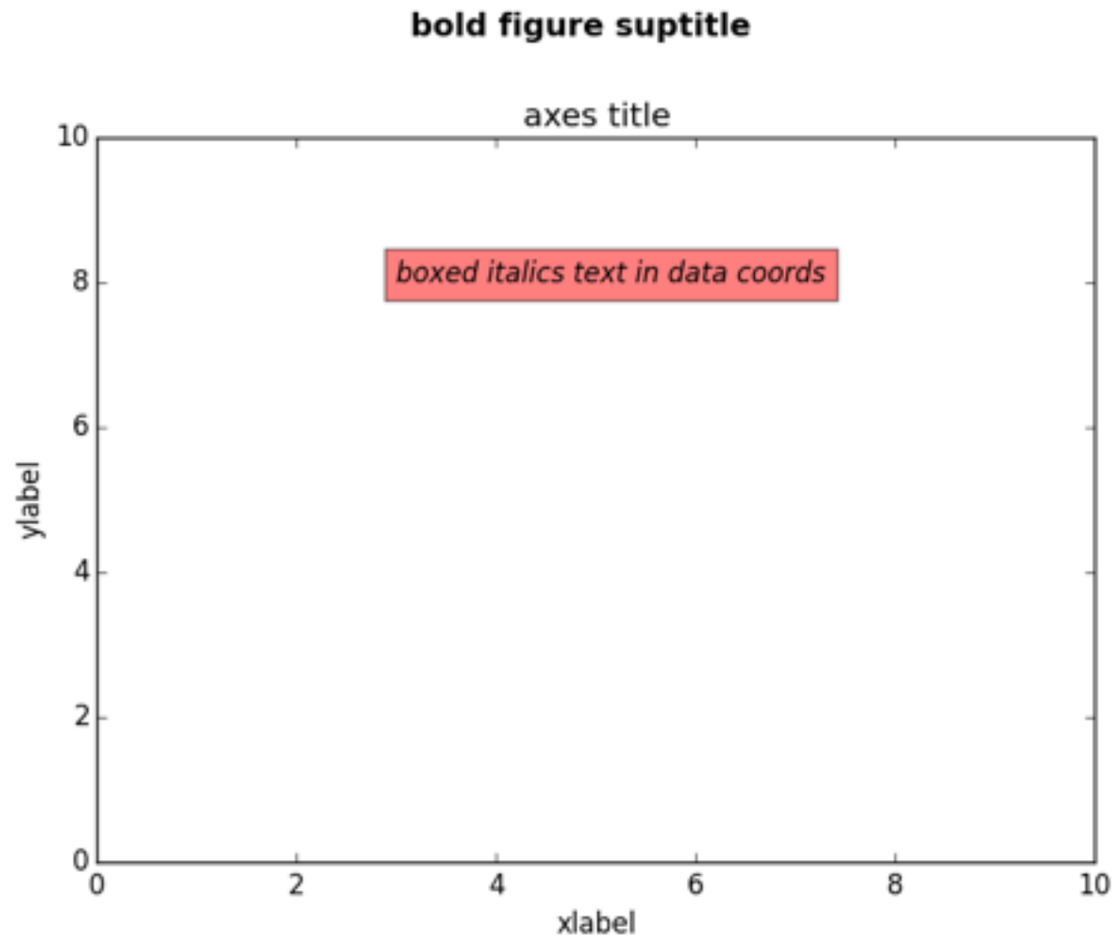
Text on Plots

```
# initialise a figure

fig = plt.figure()
fig.suptitle('bold figure suptitle', fontsize=14, fontweight='bold')

ax = fig.add_subplot(111)
fig.subplots_adjust(top=0.85)
ax.set_title('axes title')

ax.set_xlabel('xlabel')
ax.set_ylabel('ylabel')
ax.axis([0, 10, 0, 10])
```



Text on Plots

```
# lets add some text box surrounded by box

ax.text(3, 8, 'boxed italics text in data coords', style='italic',
        bbox={'facecolor':'red', 'alpha':0.5, 'pad':10})
```

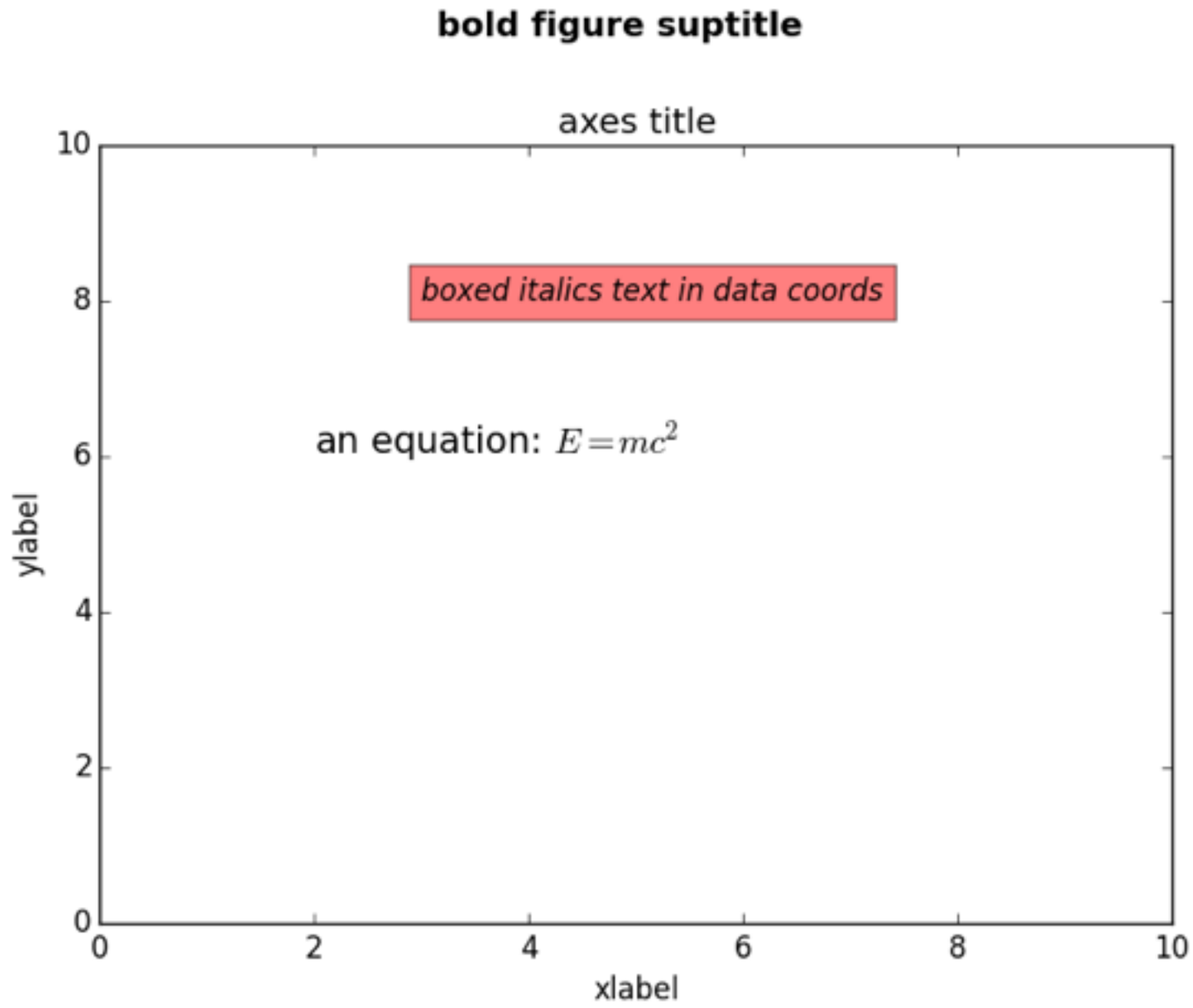


text() - add text at an arbitrary location to the Axes

Text on Plots

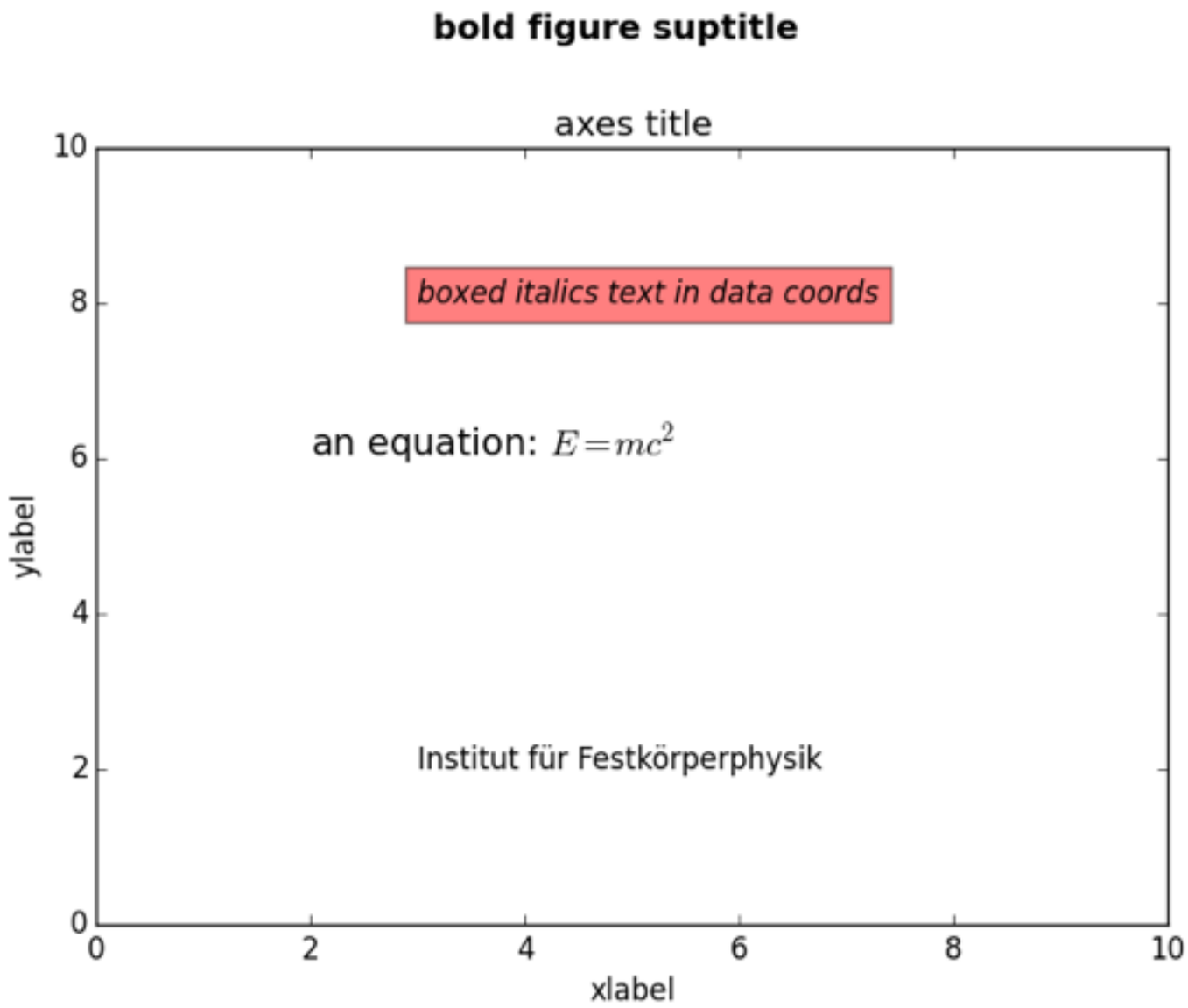
```
# lets add an equation

ax.text(2, 6, r'an equation: $E=mc^2$', fontsize=15)
```

Latex like style



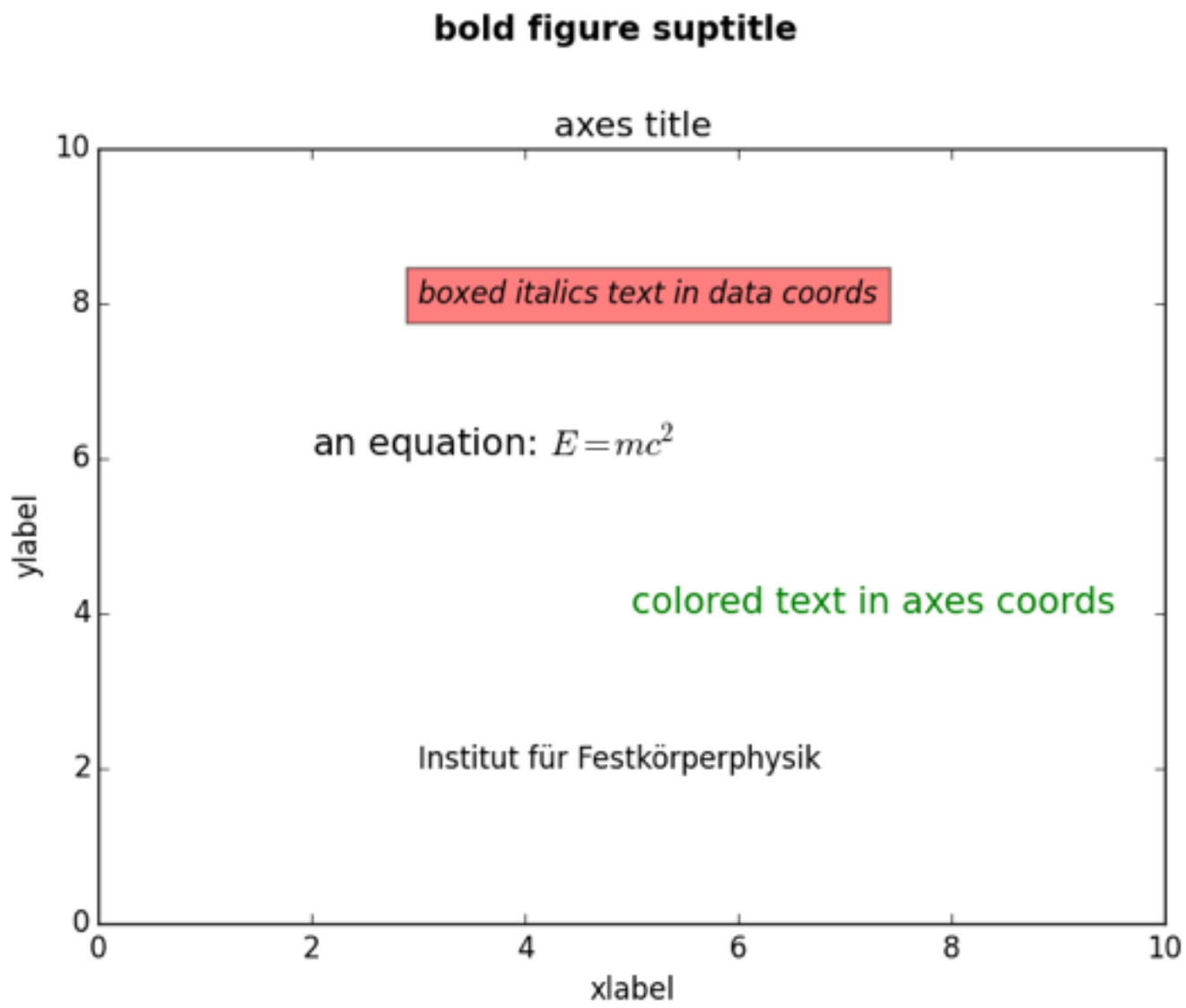
Text on Plots

```
# You can also add unicodes
ax.text(3, 2, u'Institut für Festkörperphysik')
```



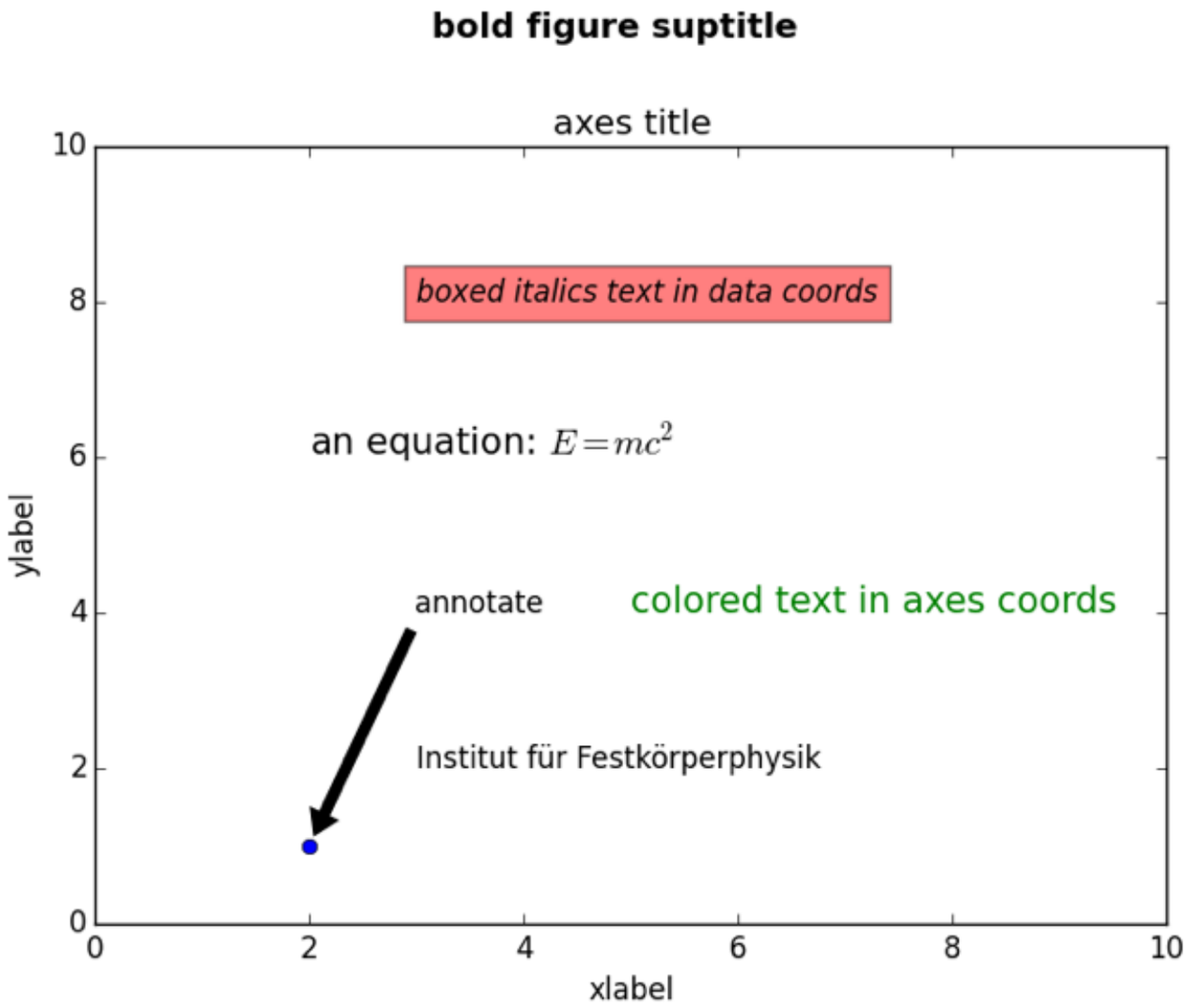
Text on Plots

```
# Add colored text
ax.text(5, 4, 'colored text in axes coords',
        color='green', fontsize=15)
```

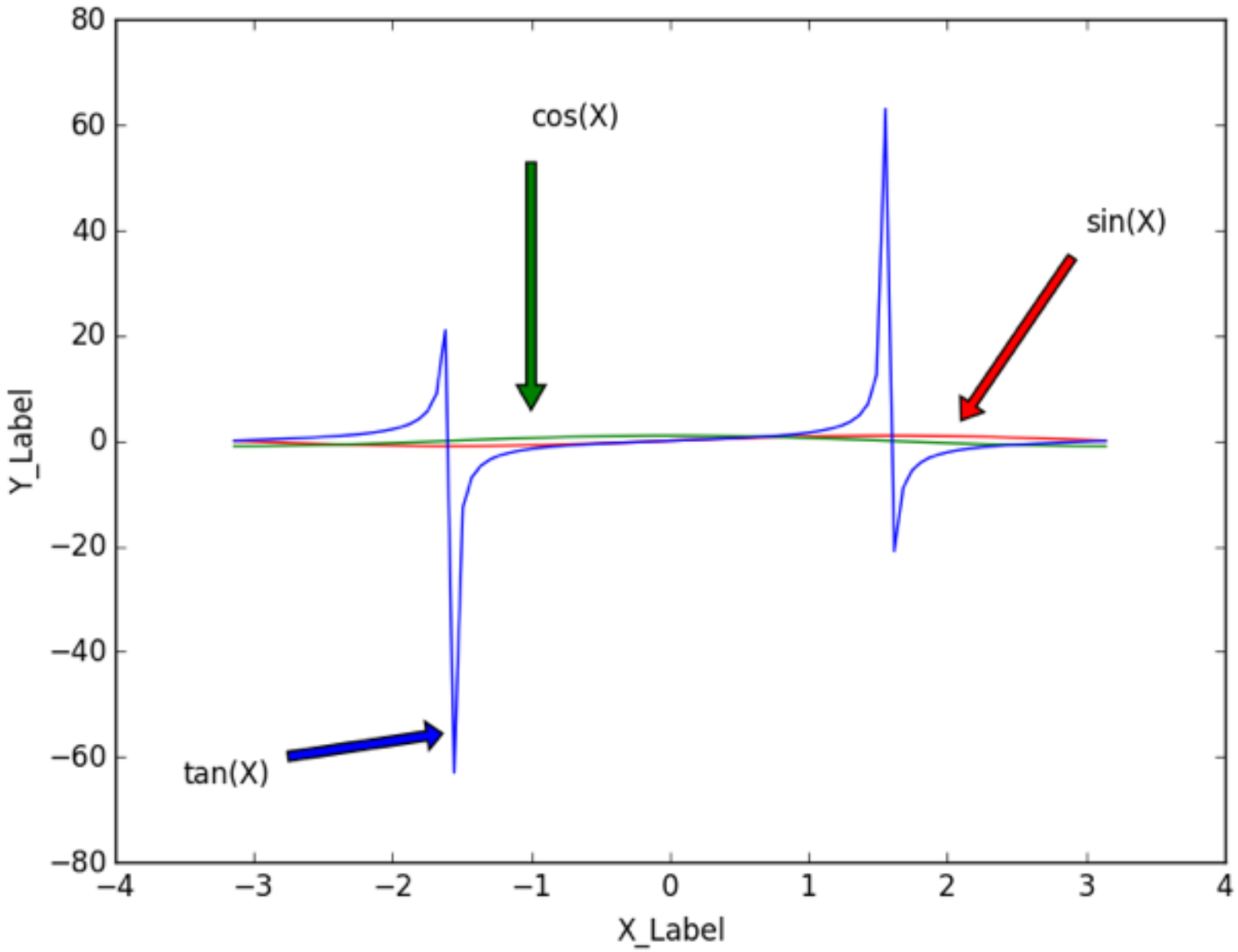


Annotation on Plots

```
# Now annotate at point
ax.plot([2], [1], 'o')
ax.annotate('annotate', xy=(2, 1), xytext=(3, 4),
            arrowprops=dict(facecolor='black', shrink=0.05))
```



Assignment 7(f) (15 mins)



1. File name : **as7f_yoursurname_name.py**
2. Upload your solution at Moodle under A07 folder after completion.