```python
"""
Registration : xxxx
Description  : Basics of MatPlotLib
Author       : AKB
"""

import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")

# Integer case switch for different problems to choose from
plot2D=0; subplt=0; statprob=0; shm=1;

if(plot2D):
    #==========================================================================#
    print                   ('~~~ 2D PLOT ~~~')                               #
    #==========================================================================#

    help(plt.plot)            # Read the complete help
    plt.figure(1)             # Figure-1
    plt.plot()                # Blank Plot without display on screen
    plt.plot(range(10), 'bo') # plot y using x (automatic) with blue circles
    plt.plot(range(10), 'r+') # plot y using x (automatic) with red plus on top
    #plt.show()               # Display on-screen

    x = np.linspace(10, 100, 10) # array([10.,20.,30.,...,100.])
    y = np.linspace(1,  10,  10) # array([1., 2., 3.,..., 10.])
    y = np.square(y)
    plt.figure(2)
    plt.plot(x,y, 'bo-')
    #plt.show()

    y1 = [i*i for i in y]
    plt.figure(3)
    plt.plot(x,y,x,y1)
    plt.show()

if(subplt):
    #==========================================================================#
    print         ('~~~ SUBPLOT : MULTIPLE FIGURES COLLAGE ~~~')              #
    #==========================================================================#

    x = np.arange(0, 4, 0.01) # subplot(nrow, ncolumn, plot_number)
    plt.figure(4)             # x vs y
    plt.subplot(221)  # left upper
    plt.plot(x, x**0.5*np.exp(-x), 'k<', lw=2, ms=2)
    plt.text(2.5, 0.35, r'$\sqrt{x}e^{-x}$', size=12)
    plt.ylabel('f(x)', size = 16); plt.yticks(size=14)
    plt.subplot(222)  # right upper
    plt.plot(x, x**0.25*np.exp(-x**2), 'ro', lw=2, ms=2);
    plt.text(2.5, 0.55, r'$x^{1/4}e^{-x}$', size=12)
    plt.subplot(223)  # left lower
    plt.plot(x, x**2*np.exp(-x), 'b+', lw=2, ms=2);
    plt.text(2.5, 0.1, r'$x^2e^{-x}$', size=12)
    plt.subplot(224)  # right lower
    plt.plot(x, x**4*np.exp(-x**2), 'cx', lw=2, ms=2);
    plt.text(2.5, 0.1, r'$x^4e^{-x^2}$', size=12)
    plt.xlabel('x', size=16); plt.xticks(size=14)

    plt.figure(5)              # Semilogx vs y
    plt.subplot(221)
    plt.semilogx(x, x**0.5*np.exp(-x), 'k<', lw=2, ms=2);
    plt.text(0.02, 0.35, r'$\sqrt{x}e^{-x}$', size=12)
    plt.ylabel('f(x)', size = 16); plt.yticks(size=14)
    plt.subplot(222)
```

```
    plt.semilogx(x, x**0.25*np.exp(-x**2), 'ro', lw=2, ms=2);
    plt.text(0.012, 0.55, r'$x^{1/4}e^{-x}$', size=12)
    plt.subplot(223)
    plt.semilogx(x, x**2*np.exp(-x), 'b+', lw=2, ms=2);
    plt.text(0.02, 0.45, r'$x^2e^{-x}$', size=12)
    plt.subplot(224)
    plt.semilogx(x, x**4*np.exp(-x**2), 'cx', lw=2, ms=2);
    plt.text(0.02, 0.45, r'$x^4e^{-x^2}$', size=12)
    plt.xlabel('x', size=16); plt.xticks(size=14)

    plt.figure(6)               # logx vs logy
    plt.subplot(221)
    plt.loglog(x, x**0.5*np.exp(-x), 'k<', lw=2, ms=2);
    plt.text(0.02, 0.35, r'$\sqrt{x}e^{-x}$', size=12)
    plt.ylabel('f(x)', size = 16); plt.yticks(size=14)
    plt.subplot(222)
    plt.loglog(x, x**0.25*np.exp(-x**2), 'ro', lw=2, ms=2)
    plt.text(0.015, 0.01, r'$x^{1/4}e^{-x}$', size=12)
    plt.subplot(223)
    plt.loglog(x, x**2*np.exp(-x), 'b+', lw=2, ms=2);
    plt.text(0.02, 0.1, r'$x^2e^{-x}$', size=12)
    plt.subplot(224)
    plt.loglog(x, x**4*np.exp(-x**2), 'cx', lw=2, ms=2);
    plt.text(0.02, 0.01, r'$x^4e^{-x^2}$', size=12)
    plt.xlabel('x', size=16); plt.xticks(size=14)
    plt.show()

if(statprob):
    #==========================================================================#
    print           ('~~~ STATISTICS & PROBABILITY ~~~')                       #
    #==========================================================================#

    #=================== Histogram ==================#
    npts = 900000;  # Total Number of Points
    nbin = 100;     # Total Number of Bins
    mean = 0;       # Mean
    std  = 2;       # Standard Deviation
    x = np.random.normal(mean, std, npts) # Gaussian Distribution

    plt.figure(7)
    plt.subplot(2,2,1);
    plt.hist(x, nbin, color='tan', label=r'$N=9\times10^5$')
    plt.legend(loc='best',prop={'size':10})
    plt.xlabel('x', size=12); plt.xticks(size=14); plt.xlim([-10, 10])
    plt.ylabel('P(x)', size=12); plt.yticks(size=10)
    plt.title(r'$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(x-\mu)^2/{2\sigma^2}}$',
size=12);

    #=================== Barchart ===================#
    men = (80, 50, 75, 60)
    women = (90, 62, 50, 65) # Data to plot
    x = np.arange(len(men))
    bar_width = 0.35

    plt.subplot(2,2,2)
    plt.bar(x+bar_width, men, bar_width, label='Men', color='red')
    plt.bar(x, women, bar_width, label='Women', color='green')
    plt.legend(loc='best', prop={'size':12})
    plt.xlabel('x', size=12);    plt.xticks(size=14)
    plt.ylabel('P(x)', size=12); plt.yticks(size=14)

    #========== User-predefined Errorbar ============#
    x = range(5)
    y = [1, 4, 16, 28, 42]

    plt.subplot(2,2,3)
```

```
    plt.errorbar(x,y,fmt='o',xerr=0.2,yerr=4.8,color='magenta')
    plt.xlabel('x', size=16); plt.xticks(size = 14)
    plt.ylabel('y', size=16); plt.yticks(size=14)
    plt.grid()

    #==================== Pie-Chart ==================#
    areas = [12.25, 29.75, 38.42, 19.58]  # Total = 100
    names = "Fortran", "Java", "Python", "Pearl"
    graycolors = "0.1", "0.8", "0.3", "0.6"
    somecolors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']
    slice = (0, 0, 0.05, 0)
    plt.subplot(2,2,4)
    plt.pie(areas, autopct='%0.2f', explode=slice, labels=names, colors=somecolors)
    #plt.savefig('plot/01_statprob.pdf')
    plt.show()

if(shm):
    #===========================================================================#
    print            ('~~~ SHM : dy/dx = z, dz/dx = -y ~~~')                   #
    #===========================================================================#

    def f(x,y,z): return z
    def g(x,y,z): return -y

    x, y, z, h = 0, 0, 1.0, 0.01  # initial values
    X, Y, Z = [],[],[]            # empty lists
    for i in range(1000):
        y += h*f(x,y,z)
        z += h*g(x,y,z)
        x += h
        X.append(x)
        Y.append(y)
        Z.append(z)

    plt.figure(8)
    plt.plot(X, Y, 'r+-', label='X-Y', lw=3, ms=8); # lw=linewidth, ms=markersize
    plt.plot(Y, Z, 'kx-', label='Y-Z', lw=3, ms=8);
    plt.legend(loc='best', prop={'size':12})
    plt.grid()
    plt.axis([-2, 10, -1.5, 1.5])
    plt.title('Simple Harmonic Motion', size = 12)
    plt.suptitle('MatPlotLib tutorial',size = 12)
    plt.text(2,1.2,r'$\frac{dy}{dx} = z, \frac{dz}{dx} = -y$',size = 12)
    plt.xlabel('Horizontal axis', size = 12); plt.xticks(size = 14)
    plt.ylabel('Vertical axis',   size = 12); plt.yticks(size = 14)
    #plt.savefig('plot/01_shm.pdf')
    plt.show()
```