

scratch (/github/ryo0921/scratch/tree/master) / 01 (/github/ryo0921/scratch/tree/master/01)

# Stochastic Processes: Data Analysis and Computer Simulation

## Python programming for beginners

## 2. Making graphs with matplotlib

### 2.1. Import libraries

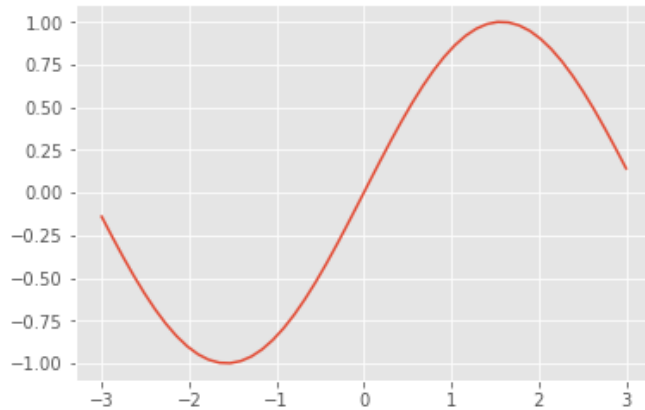
#### A code example

```
In [1]: % matplotlib inline
import numpy as np # import numpy library as np
import matplotlib.pyplot as plt # import pyplot library as plt
plt.style.use('ggplot') # use "ggplot" style for graphs
```

### 2.2. plot $\sin(x)$

#### A code example

```
In [2]: x = np.linspace(-3, 3) # create array of x from -3 to 3
        y = np.sin(x) # create array of sin(x)
        plt.plot(x, y) # plot y vs. x
        plt.show() # display plot
```



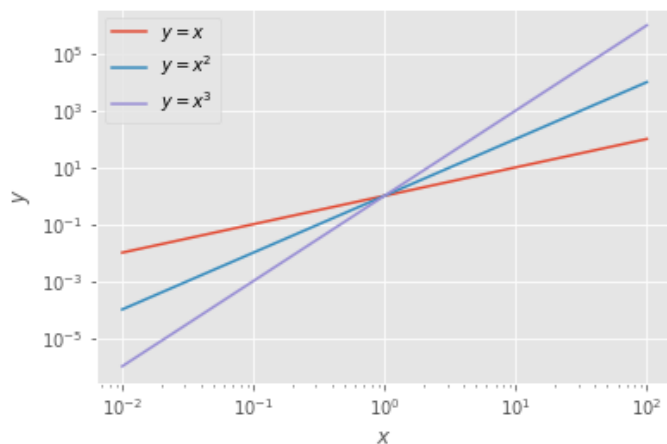
## 2.3. plot $x$ , $x^2$ , $x^3$ , ... in log-log scale.

### A code example

```
In [3]: def func (x,n):
        y = x**n
        return y
```

### A code example

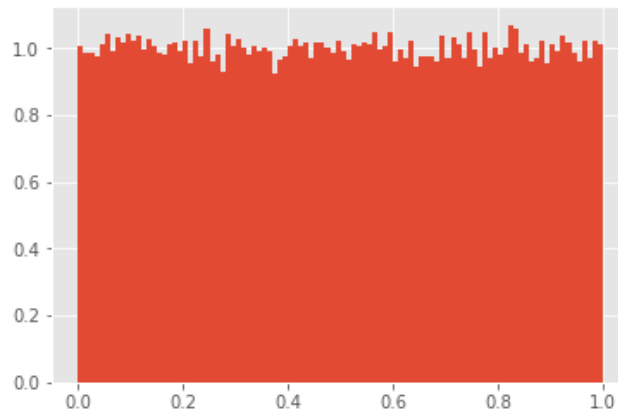
```
In [4]: x = np.logspace(-2,2) # create array of x from
        # 10^-2 to 10^2 equally spaced in log scale
        plt.plot(x,func(x,1),label='$y=x$') # plot y=x
        plt.plot(x,func(x,2),label='$y=x^2$') # plot y=x^2
        plt.plot(x,func(x,3),label='$y=x^3$') # plot y=x^3
        plt.legend() # display plot legends
        plt.xscale("log") # set log scale for x axis
        plt.yscale("log") # set log scale for y axis
        plt.xlabel("$x$") # display name of x axis
        plt.ylabel("$y$") # display name of y axis
        plt.show() # display plots
```



## 2.4. Make histogram plot

### A code example

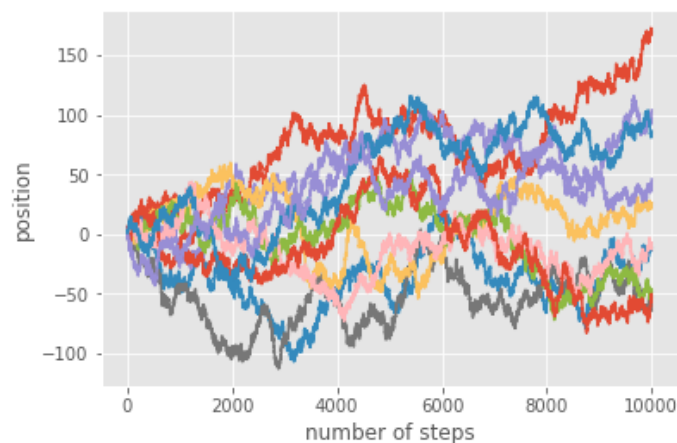
```
In [5]: N = 100000 # size of R
np.random.seed(0) # initialization of the random number generator
R = np.random.rand(N) # generate random sequence and store it as R
# plot normalized histogram of R using 100 bins
plt.hist(R,bins=100,normed=True)
plt.show() # show plot
```



## 2.5. Trajectory plot, position(step)

### A code example

```
In [6]: NSTEP = 10000 # number of random steps
plt.xlabel("number of steps") # name of x axis
plt.ylabel("position") # name of y axis
for nseed in range(10): # generate 10 random walks
    # initialize random step generator with different nseeds
    np.random.seed(nseed)
    # generate random sequences of NSTEP +1/-1 steps
    step = np.random.choice([-1,1],NSTEP)
    # calculate position of random walk at each step
    position = np.cumsum(step)
    plt.plot(position) # plot position(step) vs. step
```



## References

- matplotlib website, <http://matplotlib.org/> (<http://matplotlib.org/>)
- matplotlib gallery page, <http://matplotlib.org/gallery.html> (<http://matplotlib.org/gallery.html>)
- Python website, <https://www.python.org/> (<https://www.python.org/>)