

ENCE 460 – Digital Signal Processing
Lecture 1B: Programming in Python for Projects

Liang Zhang, Ph.D.
Dept. of Eng. and Aviation Sciences
University of Maryland Eastern Shore
Princess Anne, MD 21853, USA
lzhang1@umes.edu

February 21, 2025

Save Data

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

- `np.random.seed(1)`
- `bs = []`
- `bs = np.random.rand(10,10)`
- `data_len=np.size(bs)`
- `bs=np.reshape(bs,data_len)`

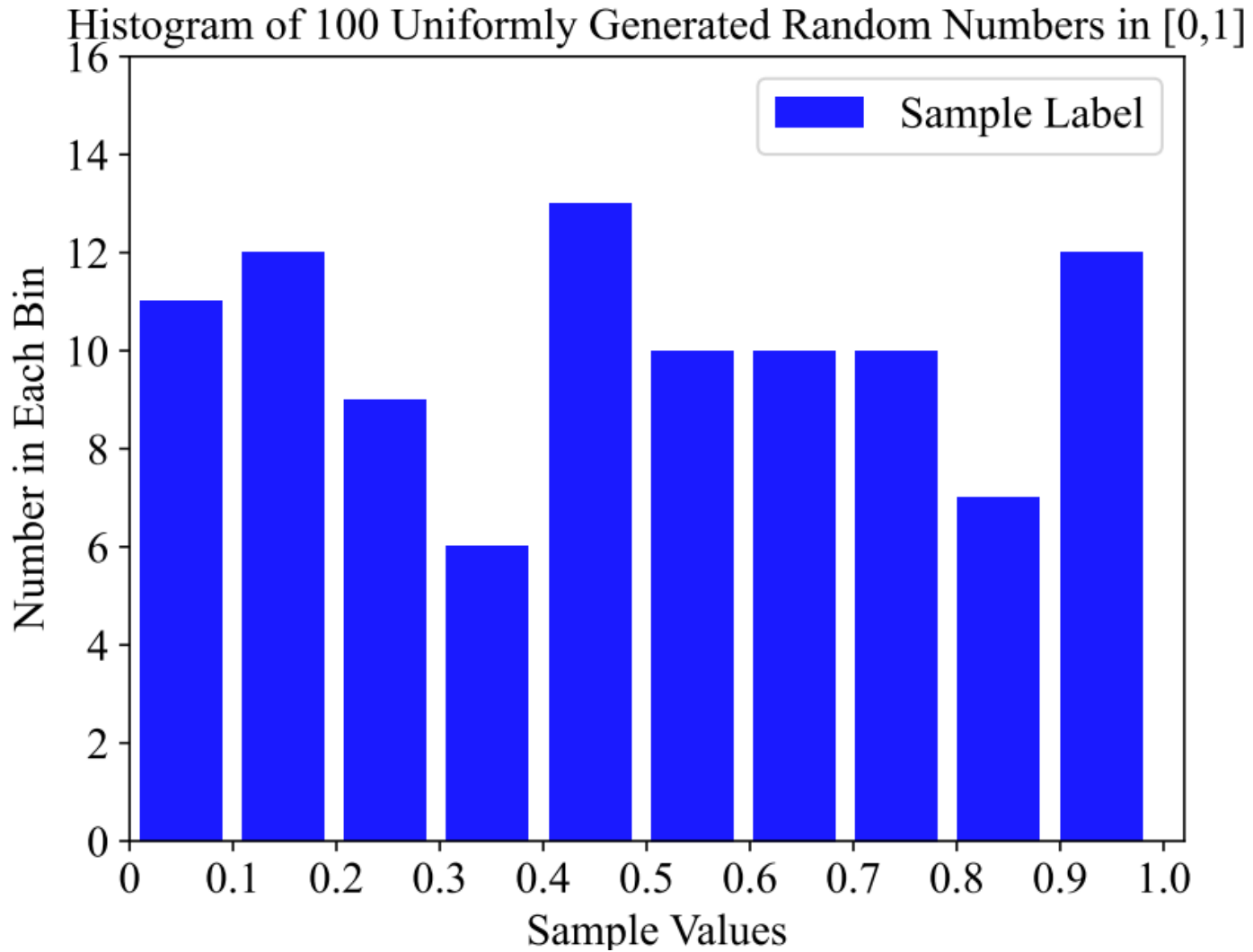
- `##### save data #####`
- `filename = ('bs'+'.pkl')`
- `pyfile = open(filename, 'wb')`
- `dill.dump(bs, pyfile)`
- `pyfile.close()`

Load Data

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

- `##### Load data #####`
- `bs = []`
- `filename='bs+'.pkl'`
- `bs = dill.load(open(filename, "rb"))`

Python Programming Hist



Python Programming Hist

- `num_fig=1`
- `##### This is figure 1 #####`
- `plt.hist(bs, bins=10,rwidth=0.8, align='mid',color='blue', alpha=0.9,label='Sample Label',)`

- `font1 = {'family' : 'Times New Roman','weight' : 'normal','size': 15,}`
- `tick_label=["0","0.1","0.2","0.3","0.4","0.5","0.6","0.7","0.8","0.9","1.0"]`
- `plt.xticks(np.arange(0,1.1,0.1),tick_label,fontproperties=font1)`
- `plt.yticks(fontproperties=font1)`
- `plt.title('Histogram of 10000 Uniformly Generated Random Numbers in [0,1]',fontproperties=font1)`
- `plt.legend(prop=font1,loc= 'best')`
- `plt.xlabel('Sample Values',font1)`
- `plt.ylabel('Number in Each Bin',font1)`
- `plt.axis([0, 1.02, 0, 16])`

- `##### save figure as a pdf #####`
- `name = "histogram"`
- `filename = str(num_fig) + '_' + str(name) + '.pdf'`
- `plt.savefig(filename)`
- `num_fig += 1`
- `plt.ion() # plt.show()`
- `plt.pause(15) # set up the time of figure on the screen`
- `plt.close()`

Initialize Variables in Python

- Define an int variable:

- `slots = 20`

- Define an array:

- `N_ue_set = np.array([4,8,12,16,20,24,28])`

- `N_bs_set = (np.ones(np.size(N_ue_set))*3).astype(int)`

- Define a two-dimensional array:

- `Readout = np.zeros((Alg_num, N_bs))`

- `Readout = np.ones((Alg_num, N_bs))`

- Define a range list:

- `load = range(100, 900, 100)`

Initialize Variables in Python

- `slots = 20`
- `load = range(100, 900, 100)`
- `N_ue_set = np.array([4,8,12,16,20,24,28])`
- `N_bs_set = (np.ones(np.size(N_ue_set))*3).astype(int)`

- `test = 2`
- `N_bs = int(N_bs_set[test])`
- `N_ue = int(N_ue_set[test])`
- `capacity = np.ones(N_bs)`

- `Alg_num = 4`
- `Prob = np.zeros((Alg_num, N_bs))`
- `Readout = np.zeros((Alg_num, N_bs))`

Python range and arange

- arange needs numpy and range does not need it.
- ```
>>> range(5)
[0,1,2,3,4]
```
- ```
>>> np.arange(5)  
array([0,1,2,3,4])
```
- ```
>>>range(1,10,2)
[1,3,5,7,9]
```
- ```
>>>np.arange(1,10,2)  
array([1,3,5,7,9])
```


Define a Class

```
● 17 class Timeslot:
    18     def __init__(self, N_bs, N_ue, slots):
    19         self.N_bs = N_bs
    20         self.N_ue = N_ue
    21         self.slots = slots
    22     def displayTimeslot(self):
    23         print("Number of wearable device is N_bs=%s ", self.N_bs, "\n")
    24         print("Number of biosensor is N_ue= %s", self.N_ue)
```

Matplotlib Color

Alias	Color
'b'	Blue
'r'	Red
'g'	Green
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'w'	White

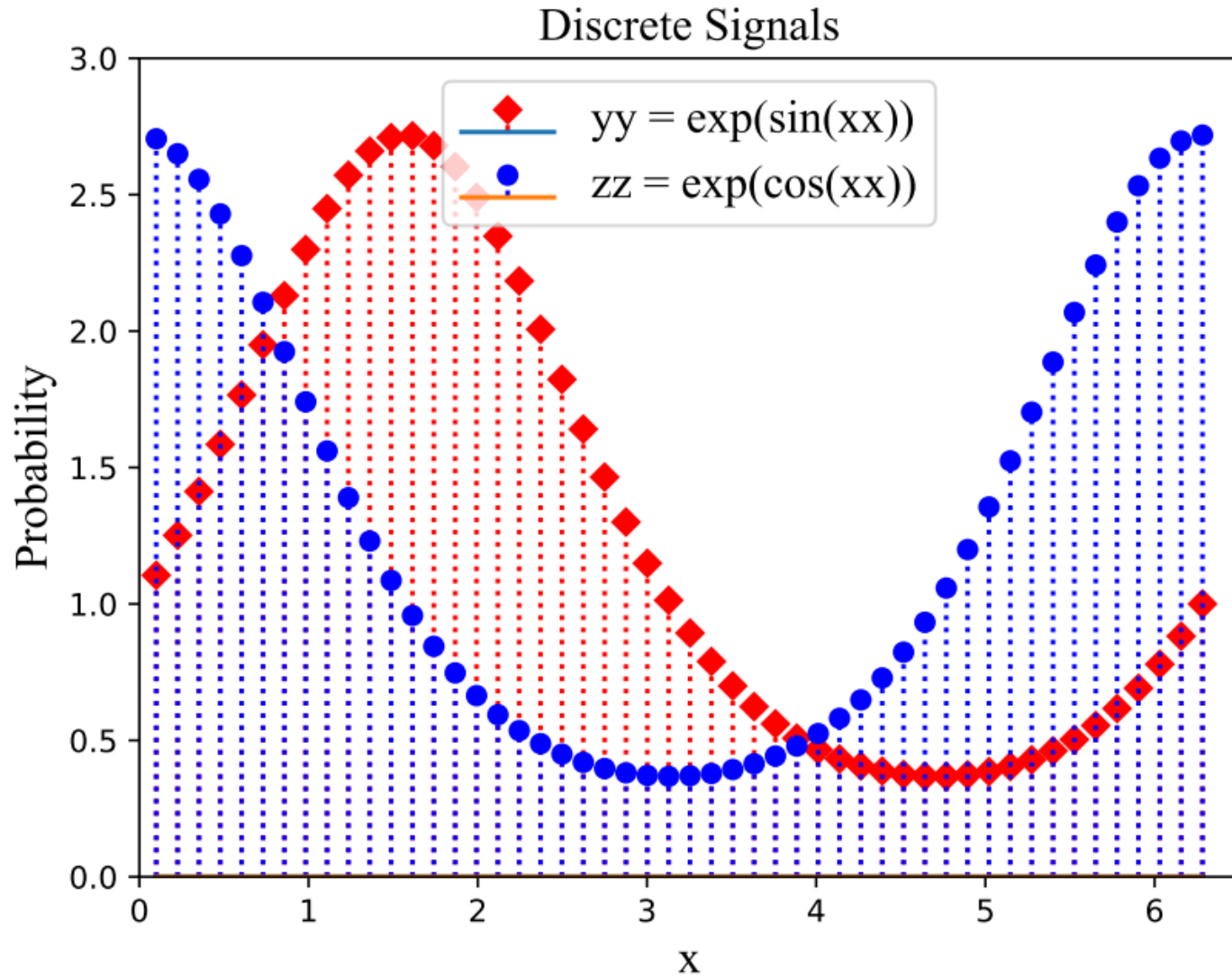
Source: <https://www.geeksforgeeks.org/matplotlib-pyplot-colors-in-python/>

Matplotlib Markers

Marker	Description	Marker	Description
'o'	Circle	'H'	Hexagon
'*'	Star	'h'	Hexagon
'.'	Point	'v'	Triangle Down
','	Pixel	'^'	Triangle Up
'x'	X	'<'	Triangle Left
'X'	X (filled)	'>'	Triangle Right
'+'	Plus	'1'	Tri Down
'P'	Plus (filled)	'2'	Tri Up
's'	Square	'3'	Tri Left
'D'	Diamond	'4'	Tri Right
'd'	Diamond (thin)	' '	Vline
'p'	Pentagon	'_'	Hline

Source: https://www.w3schools.com/python/matplotlib_markers.asp

Python Programming Plot



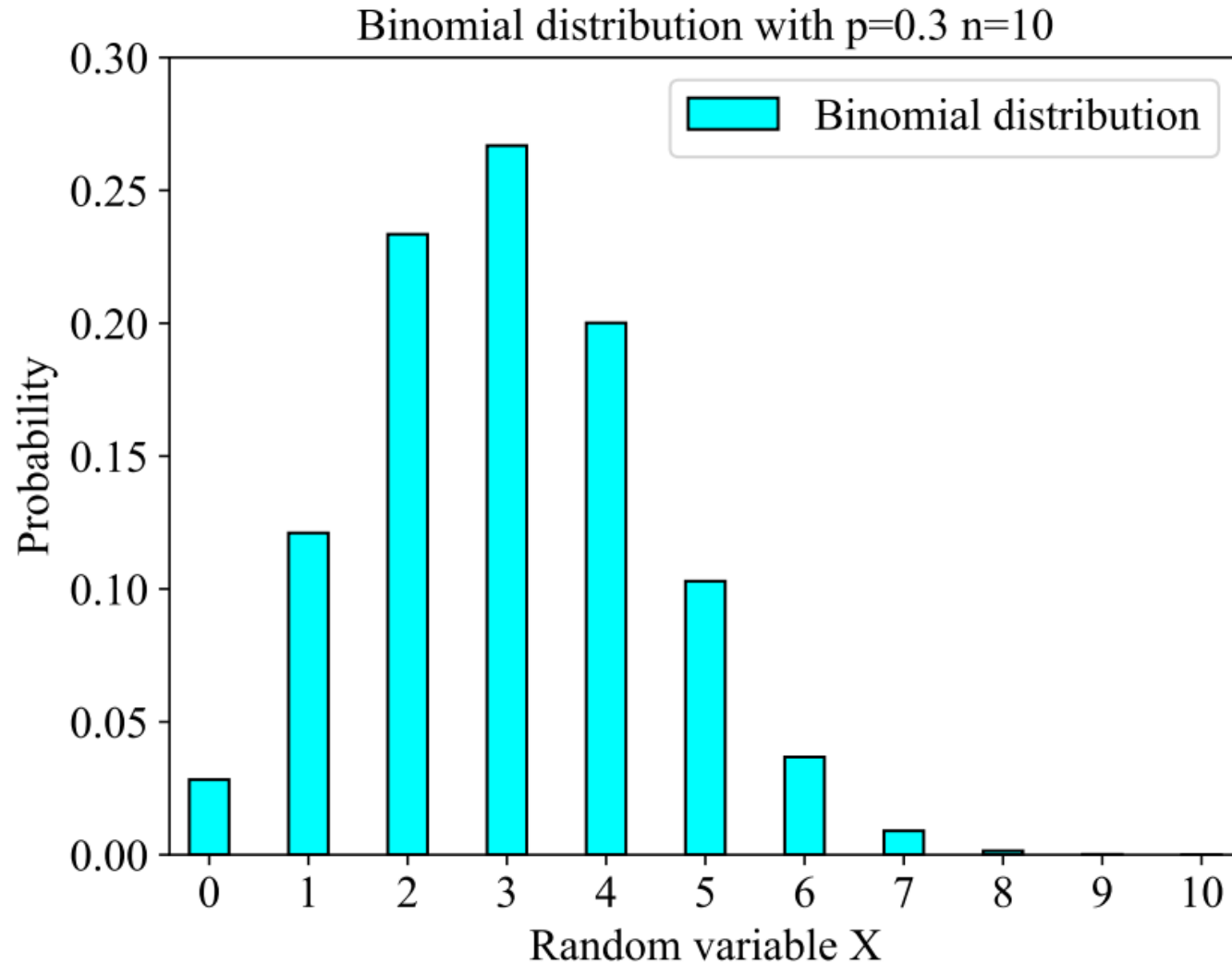
Python Programming Plot

- `import matplotlib.pyplot as plt`
- `import numpy as np`
- `num_fig=1`
- `font1 = {'family' : 'Times New Roman','weight' : 'normal','size': 15,}`
- `##### This is figure 1 #####`
- `plt.figure(num_fig)`
- `xx = np.linspace(0.1, 2 * np.pi, 50)`
- `yy = np.exp(np.sin(xx))`
- `zz = np.exp(np.cos(xx))`
- `markerline, stemlines, baseline = plt.stem(xx,yy,linewidth="--", markerfmt="rD",basefmt="--", use_line_collection=True,label='yy = exp(sin(xx))',)`
- `plt.setp(stemlines, 'color', plt.getp(markerline,'color'))`
- `plt.setp(stemlines, 'linestyle', 'dotted')`

- `markerline, stemlines, baseline = plt.stem(xx,zz,linewidth="--", markerfmt="bo",basefmt="--", use_line_collection=True,label='zz = exp(cos(xx))',)`
- `plt.setp(stemlines, 'color', plt.getp(markerline,'color'))`
- `plt.setp(stemlines, 'linestyle', 'dotted')`

- `plt.title('Discrete Signals',fontproperties=font1)`
- `plt.xlabel('x',font1)`
- `plt.ylabel('Probability',font1)`
- `plt.legend(prop=font1,loc='best')`
- `plt.axis([0, 6.5,0, 3])`

Binomial Distribution Plot



Binomial Distribution Plot

- `import matplotlib.pyplot as plt`
- `import numpy as np`
- `import scipy.stats as stats`
- `num_fig=2`
- `##### This is figure 2 #####`
- `plt.figure(num_fig)`
- `n=10`
- `p=0.3`
- `x=np.arange(0,n+1,1)`
- `y=stats.binom.pmf(x,n,p)`
- `bar_width=0.4`
- `plt.bar(x,y,bar_width,color="#00FFFF",edgecolor='black',label="Binomial distribution",hatch="")`

- `plt.legend(prop=font1,loc= 'best')`
- `plt.title('Binomial distribution with p=0.3 n=10',font1)`
- `plt.xlabel('Random variable X',font1)`
- `plt.ylabel('Probability',font1)`
- `plt.xticks(x,fontproperties=font1)`
- `plt.yticks(fontproperties=font1)`
- `plt.axis([-0.4, 10.4,0, 0.35])`

Python Tutorial

- The sample codes of python have been uploaded to Canvas.

- Open google and search the content below:
“Python Cheat Sheet Mosh Hamedani PDF”