

## EN4553\_Assignment\_1\_Q3

November 27, 2022

Q3:4 - Using the library functions `numpy.random.randn()`, and `numpy.random.uniform()` write code to simulate 10, 000 realizations of  $Y$ . Then use `seaborn.distplot()` to plot the distribution to verify that your sketch above is accurate. Include your code and the plot in the answer sheet.

```
[9]: # importing necessary libraries
import numpy as np
import seaborn as sns

realizations = 10000 # number of realizations of  $Y = IX_1 + (1-I)X_2$ 
p = 0.2 # probability of success of the bernoulli dist.
Y = np.zeros((realizations)) # numpy array to store samples of  $Y$  dist.

# Distribution of  $X_1$ 
mu1 = 0; sigma1 = 1
X1 = sigma1 * np.random.randn(realizations) + mu1

# Distribution of  $X_2$ 
mu2 = 10; sigma2 = 1
X2 = sigma2 * np.random.randn(realizations) + mu2

for trial in range(realizations):
    # get the bernoulli variable
    i = np.random.binomial(1, p)

    # sampling from the two distribution of  $X_1$ 
    index1 = int(np.random.uniform(0, realizations))
    x1 = X1[index1]

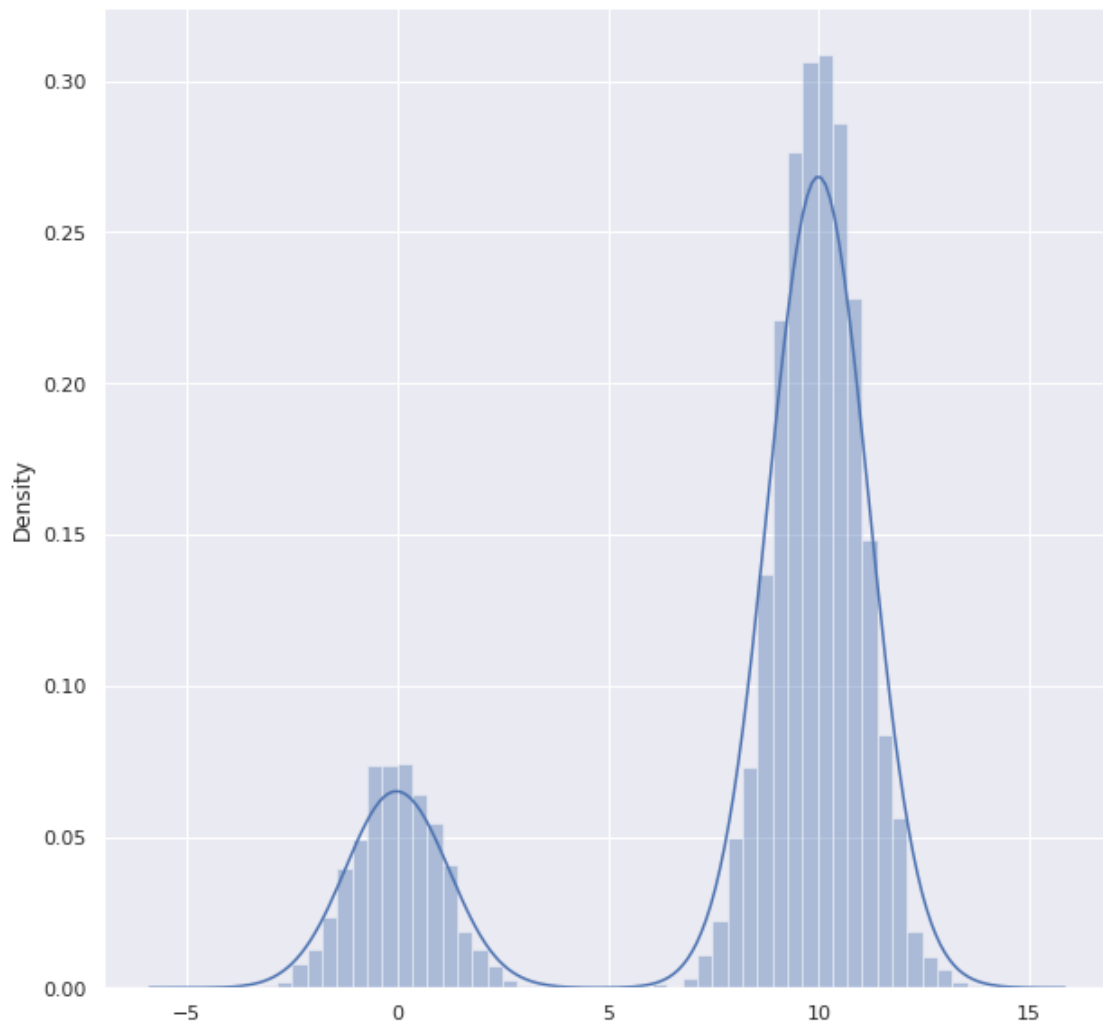
    # sampling from the two distribution of  $X_2$ 
    index2 = int(np.random.uniform(0, realizations))
    x2 = X2[index2]

    # claculating the y value using the above values
    y = i * x1 + (1 - i) * x2
    Y[trial] = y

# visualization of the  $Y$  distribution
sns.distplot(Y)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:  
FutureWarning: `distplot` is a deprecated function and will be removed in a  
future version. Please adapt your code to use either `displot` (a figure-level  
function with similar flexibility) or `histplot` (an axes-level function for  
histograms).  
warnings.warn(msg, FutureWarning)
```

```
[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd1b6ba4b10>
```



```
[1]:
```