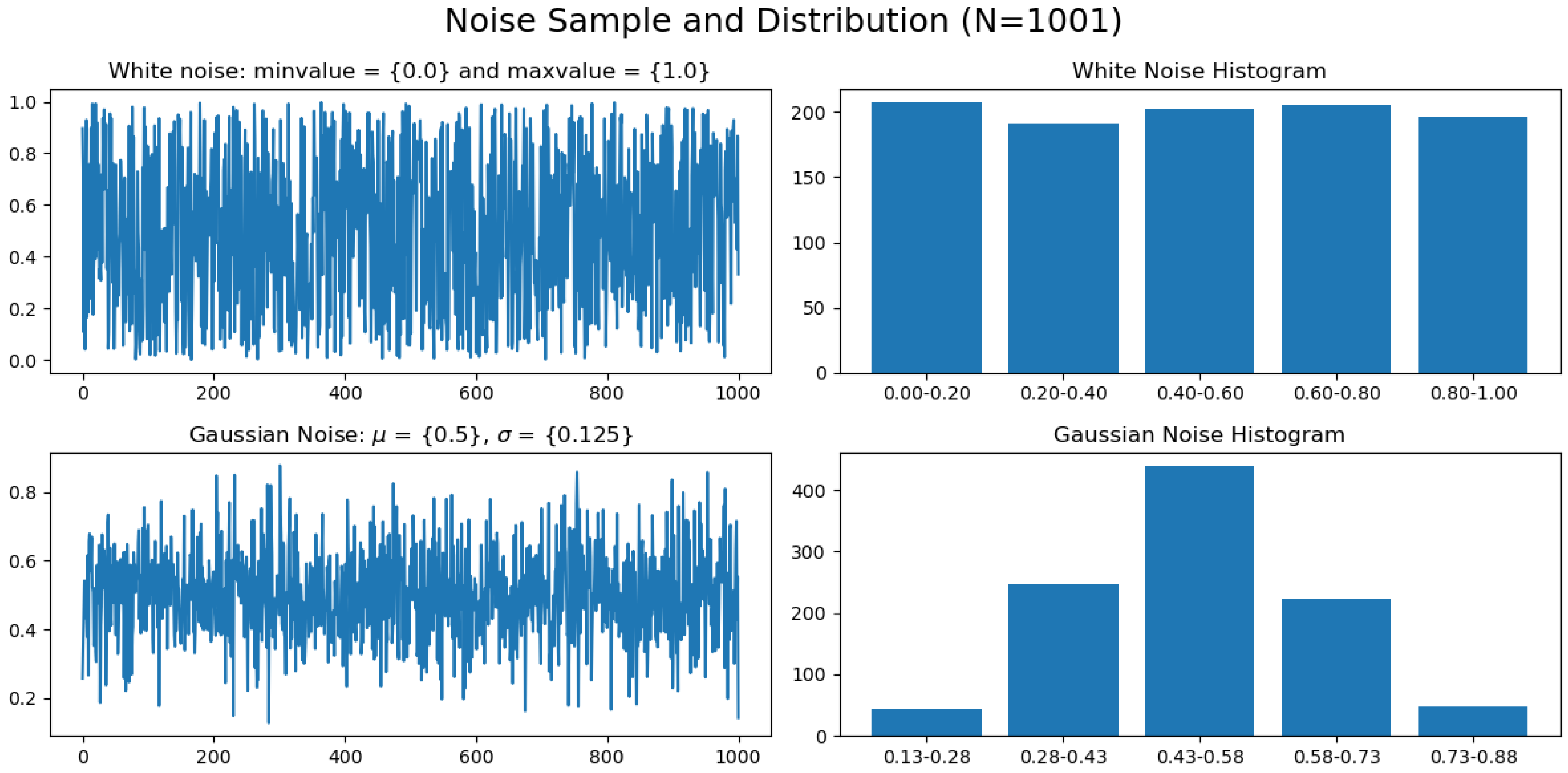
In [ ]:

```
#Copyright (C) 2021, Sydney Nwakanma
#April 15th, 2021
# Description: Hello world Python program that will generate white noise
              #  and gaussian samplings with plots
#Inputs: input from the keyboard
#Outputs: displays ASCII text to stdout
#Assumptions: written/tested with Python 3.9.1 on Windows
#Dependencies: Python plots and NumPy modules

import numpy as np
import random
import matplotlib.pyplot as plt
from datetime import datetime as dt

#Generate a function for White noise, returning a NumPy array to the caller.
def make_white_noise(nSamples, minValue, maxValue):
    a = np.random.uniform(minValue, maxValue, nSamples)
    return a

#Generate a function for Gaussian noise, returning a NumPy array to the caller.
def make_gaussian_noise(nSamples, mu, sigma):
    a = np.random.normal(mu, sigma, nSamples)
    return a



#start of main

def main():

#print out name and date using print()
   print("Sydney Nwakanma")
   t = dt.today()
   print(t)

#hard coded values
   nSamples = 1001 #number of white noise samplings to produce
   minValue = 0 #minimum value for white noise
   maxValue = 1 #maximum value for white noise
   mu = 0.5 #mean for gaussian noise
   sigma = 0.125 #standard deviation for gaussian noise

#call the functions for white noise and gaussian noise
   a_wn = make_white_noise(nSamples, minValue, maxValue)
   a_gn = make_gaussian_noise(nSamples, mu, sigma)



#create the top level figure and 4 subplots for white noise
   fig, sp = plt.subplots(nrows = 2, ncols = 2, figsize = (12,6))
   fig.suptitle("Noise Sample and Distribution (N={0})".format(nSamples),
                fontsize = 18)



#histogram computations for noise values
   nbins = 5
   counts_wn, edges_wn = np.histogram(a_wn, nbins)
   counts_gn, edges_gn = np.histogram(a_gn, nbins)
```

```
#Make a line plot of white noise

   sp[0,0].set_title("White noise: minvalue = {0.0} and maxvalue = {1.0}")
   sp[0,0].plot(a_wn)




#make the axis labels for the histogram plot for white noise
   axis_labels_wn = list()
   for i in range(len(edges_wn)-1):
       axis_labels_wn.append("%4.2f-%4.2f" % (edges_wn[i], edges_wn[i+1]))

#plot histogram for white noise
   sp[0,1].set_xticklabels(axis_labels_wn)
   sp[0,1].set_xticks(np.arange(nbins+1))

   sp[0,1].set_title("White Noise Histogram")
   sp[0,1].bar(np.arange(nbins), counts_wn)

#Make a line plot of gaussian noise
   sp[1,0].set_title(r"Gaussian Noise: $\mu$ = {0.5}, $\sigma$ = {0.125}")
   sp[1,0].plot(a_gn)



#make the axis labels for the histogram plot for gaussian noise
   axis_labels_gn = list()
   for i in range(len(edges_gn)-1):
       axis_labels_gn.append("%4.2f-%4.2f" % (edges_gn[i], edges_gn[i+1]))

        #plot histogram for gaussian noise
   sp[1,1].set_xticklabels(axis_labels_gn)
   sp[1,1].set_xticks(np.arange(nbins+1))

   sp[1,1].set_title("Gaussian Noise Histogram")
   sp[1,1].bar(np.arange(nbins), counts_gn)

   plt.tight_layout()
   plt.show()



if __name__=='__main__':
    main()

#end of file
```


Noise Sample and Distribution (N=1001)
White noise: minvalue = {0.0} and maxvalue = {1.0}
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
White Noise Histogram
0
50
100
150
200
0.00-0.20
0.20-0.40
0.40-0.60
0.60-0.80
0.80-1.00
Gaussian Noise: μ = {0.5}, σ = {0.125}
0.2
0.4
0.6
0.8
0
200
400
600
800
1000
Gaussian Noise Histogram
0
100
200
300
400
0.13-0.28
0.28-0.43
0.43-0.58
0.58-0.73
0.73-0.88