Homework Assignment 2 – Data Analysis

DUE in class, Tuesday, September 26

Many physical phenomena follow a Gaussian or normal probability distribution. We will see this distribution often in this course. Therefore, it is useful to become familiar with the normal distribution.

Create a Python program that generates 10,000 values using the gauss random number generator. Use mu (mean) = 8 and sigma (standard deviation) = 3. The best way to write a program like this is to re-use code that you are familiar with. Look back at the code you used in class to do similar tasks and paste in snippets of that code to your program.

Please annotate your code! Points will be taken off for lack of annotation. Examples are:

```
“””
At the top of your program include a description of the program and its
output. This description should be between sets of triple quotation marks.
PUT YOUR NAME HERE. Unnamed code cannot receive credit.
“””
```

And for every block of code add a comment describing what the code is doing

```
# Set mu and sigma.
mu = 8
sigma = 3
```

Examine your output data with **xmgrace** (or plotting app of your choice) to ensure that it is reasonable and print out a labeled plot showing that your data approximates a Gaussian curve. Include the best fit curve in the plot. Include your name in the subtitle of the plot.

Include in your program code that will print out what % of your 10,000 values are greater than a cutoff value of mu+sigma, and what % are greater than mu+(2*sigma). The printed values should be labeled and should be floats formatted with %6.2f formatting. Below are snippets of code that, in addition to code you used in class, may be useful to you in this exercise.

Hand in a printed plot of your generated data (Don’t forget labels!) and a working version of your program **by email** to pat.fleming@jhu.edu. (Note: Please DO NOT attach an MSWord document of your program to the email; just attach the .py file so it can run "as is". It should be named [your_JHEDID].py.)

Your name (or JHEDID), must appear in at least three places: file name with .py extension, header or comment section of the python file, subtitle of your plot. No name, no credit.

Some hints for writing your program are on the next page.

Arrays (Vectors or Matrices) are useful for storing information; a list can be a vector array. You can "append" to an array using the following code.

```
# Define an empty array
A = []
# Fill the array with 9 values
for i in range(1,10):
      # add a value to the end of the array with each iteration
      A.append(i**2)
```

A is now an array of the form [1,4,8,16…81]

You can determine the length of an array.

```
B = [1,5,3,8,9,2,3,5,6,8]
print("Length of array B = ",len(B))
```

You can implement relational operations such as <,>,<=, >= or ==.

```
A = 1
B = 2
if A < B:
      print("A is smaller than B")
else:
      print("A is larger than or equal to B")
```

And you can iterate through the values in an array to check if they are above some cutoff value.

```
# Create filled array
C = [1,4,6,8,10]
# Initialize a variable
greater = 0
# Determine the number of values greater than a cutoff
for count in range(0,len(C)):
      if C[count] > 5:
            greater = greater +1
print("Number of values greater than 5 = ",greater)
```