EGR 103L – Spring 2022

Laboratory 3 - Functions and Random Numbers

Malcolm Rodgers (mlr81)
Lab Section 03, Thursday 12:00 - 3:00 PM
1/30/22

I understand and have adhered to all the tenets of the Duke Community Standard in completing every part of this assignment. I understand that a violation of any part of the Standard on any part of this assignment can result in failure of this assignment, failure of this course, and/or suspension from Duke University.

Contents

List of Figures

1 PE 1.35 - Triangles

```
Running tests for user mlr81
Test 1: runs: triangles( 9,  2,  8) returns (1.989e+00, 2.045e-01, 9.480e-01)
Test 2: runs: triangles( 4,  9,  7) returns (4.400e-01, 1.861e+00, 8.411e-01)
Test 3: runs: triangles( 2,  8,  7) returns (2.320e-01, 1.975e+00, 9.351e-01)
Test 4: runs: triangles( 8,  4,  5) returns (2.183e+00, 4.214e-01, 5.368e-01)
```

2 Random Numbers

```
NetID = mlr81
how many numbers? 10000

Uniform: Min: +2.365e-04 Avg: +4.995e-01 Max: +9.999e-01
Normal: Min: -3.579e+00 Avg: -7.495e-03 Max: +3.927e+00
```

A Codes

A.1 tri_calc.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Feb  6 19:27:06 2022

@author: malcolmrodgers
"""

#import modules
import numpy as np
import matplotlib.pyplot as plt

#define 'Triangles' function
def triangles( a, b, c, draw=False, fnum=1):
    cosA = ((b*b) + (c*c) - (a*a)) / (2*b*c)
    cosB = ((a*a) + (c*c) - (b*b)) / (2*a*c)
    cosC = ((a*a) + (b*b) - (c*c)) / (2*a*b)

    #calculate interior angles
    A = np.arccos(cosA)
    B = np.arccos(cosB)
    C = np.arccos(cosC)

    #generate plots if draw=True
    if draw:
        fig = plt.figure(num=fnum, clear=True)
        ax = fig.add_subplot(1, 1, 1)

        #h = horizontal distance, v = vertical
        h = b * np.cos(C)
        v = b * np.sin(C)

        x = [0, a, (a-h), 0]
        y = [0, 0, v, 0]

        ax.plot(x, y, '-')
        ax.set(title = "Triangle (mlr81)")
        ax.axis("equal")
        fig.tight_layout()

        #return angles
        return A, B, C

if __name__ == "__main__":
    print(triangles(3, 6, 4, True, 5))
```

A.2 gen_rand.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Feb  6 19:52:53 2022

@author: malcolmrodgers
"""

#import modules
import numpy as np
import matplotlib.pyplot as plt
import math as m

#seed based on NetID
NetID = input("NetID = ")
seed = 0
for code in map(ord, NetID):
    seed = seed + code

np.random.seed(seed)

#user inputs desired # of numbers
nums = int(input("how many numbers? "))

#distribution calculations
ud = np.random.uniform(0, 1, size = nums)
nd = np.random.normal(0, 1, size = nums)

#generate plots
num_bins = m.ceil(10 * m.log10(nums))

fig = plt.figure(num=1, clear=True)
ax = fig.add_subplot(1, 1, 1)
ax.hist(ud, num_bins)
ax.set(title="Uniform")
fig.tight_layout()
fig.savefig("UniformPlot.png")

fig = plt.figure(num=2, clear=True)
ax = fig.add_subplot(1, 1, 1)
ax.hist(nd, num_bins)
ax.set(title="Normal")
fig.tight_layout()
fig.savefig("NormalPlot.png")

#print statistics
a = np.min(ud)
b = np.mean(ud)
c = np.max(ud)

d = np.min(nd)
e = np.mean(nd)
f = np.max(nd)

print("Min: {:+.3e}".format(a), "Avg: {:+.3e}".format(b), "Max: {:+.3e}".format(c))
print("Min: {:+.3e}".format(d), "Avg: {:+.3e}".format(e), "Max: {:+.3e}".format(f))
```

B Figures

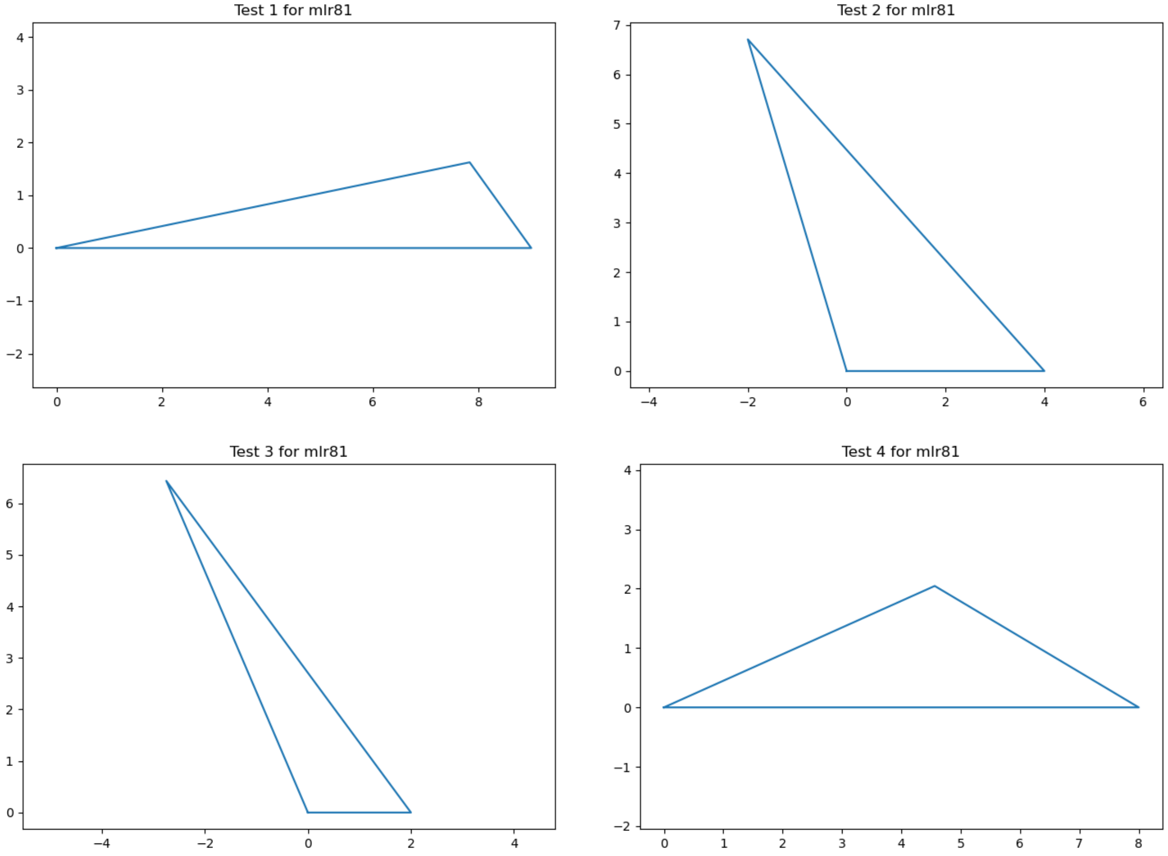

Figure 1: Test Triangles.

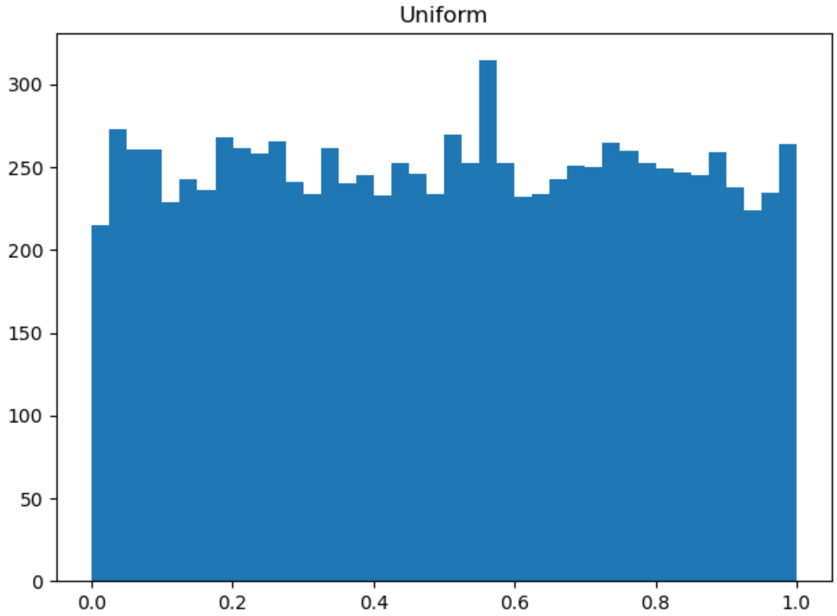

Figure 2: Histogram of Uniformly Distributed Random Numbers.

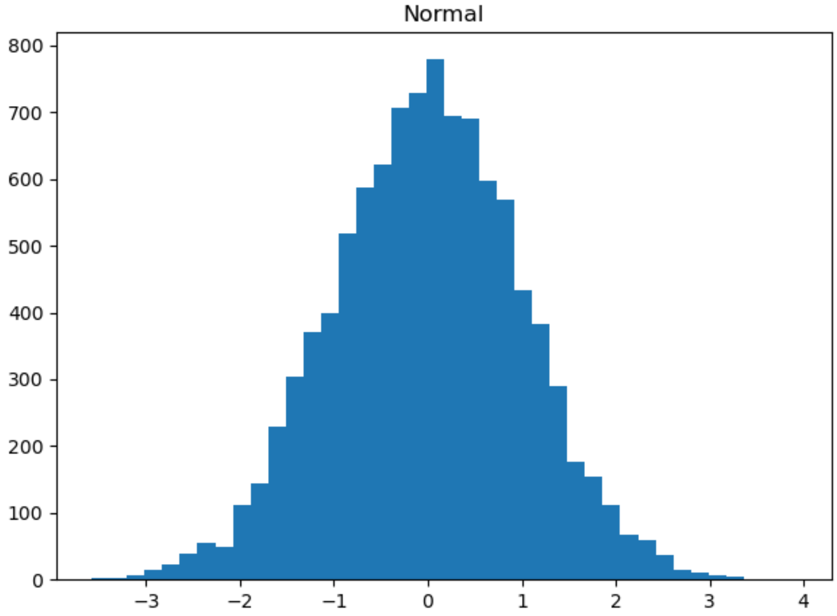

Figure 3: Histogram of Normally Distributed Random Numbers.