Probability with Coding

Karthikeya Hanu Prakash Kanithi

IIT Hyd

October 17, 2023

Objectives

- Generate standard normal random variable using C
- Solve our problem statement

Python vs C

- Python already has inbuilt libraries through which we can generate standard normal random variables. But In C we have to generate them using random numbers...
- Here is where we will learn about Box-muller transforms

Box-Muller Transform

- Suppose U1 and U2 are independent samples chosen from the uniform distribution on the unit interval (0, 1). Let

$$Z_0 = R\cos(\Theta) = \sqrt{-2\ln U_1}\cos(2\pi U_2) \tag{1}$$

- and

$$Z_1 = R\sin(\Theta) = \sqrt{-2\ln U_1}\cos(2\pi U_2) \tag{2}$$

- Then Z_0 and Z_1 are independent random variables with a standard normal distribution.
- So, now we will generate Z_0 using C Code as given below

C-code I

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include <time.h>
5
6  double sn() {
7      double u1, u2;
8      while (1) {
9          u1 = ((double)rand() / RAND_MAX);
10         if (u1 > 0 && u1 < 1)
11             break;}
12     while (1) {
13         u2 = ((double)rand() / RAND_MAX);
14         if (u2 > 0 && u2 < 1)
15             break;}
```

C-code II

```
16      double z1 = sqrt(-2 * log(u1)) * cos(2 *
            M_PI * u2);
17      return z1;
18  }
19
20  int main() {
21      // Seed the random number generator
22      srand(time(0));
23
24      int numSamples = 100000; // You can change
            this to the desired number of samples
25      FILE *file = fopen("uni.dat", "w");
26
27      if (file == NULL) {
```

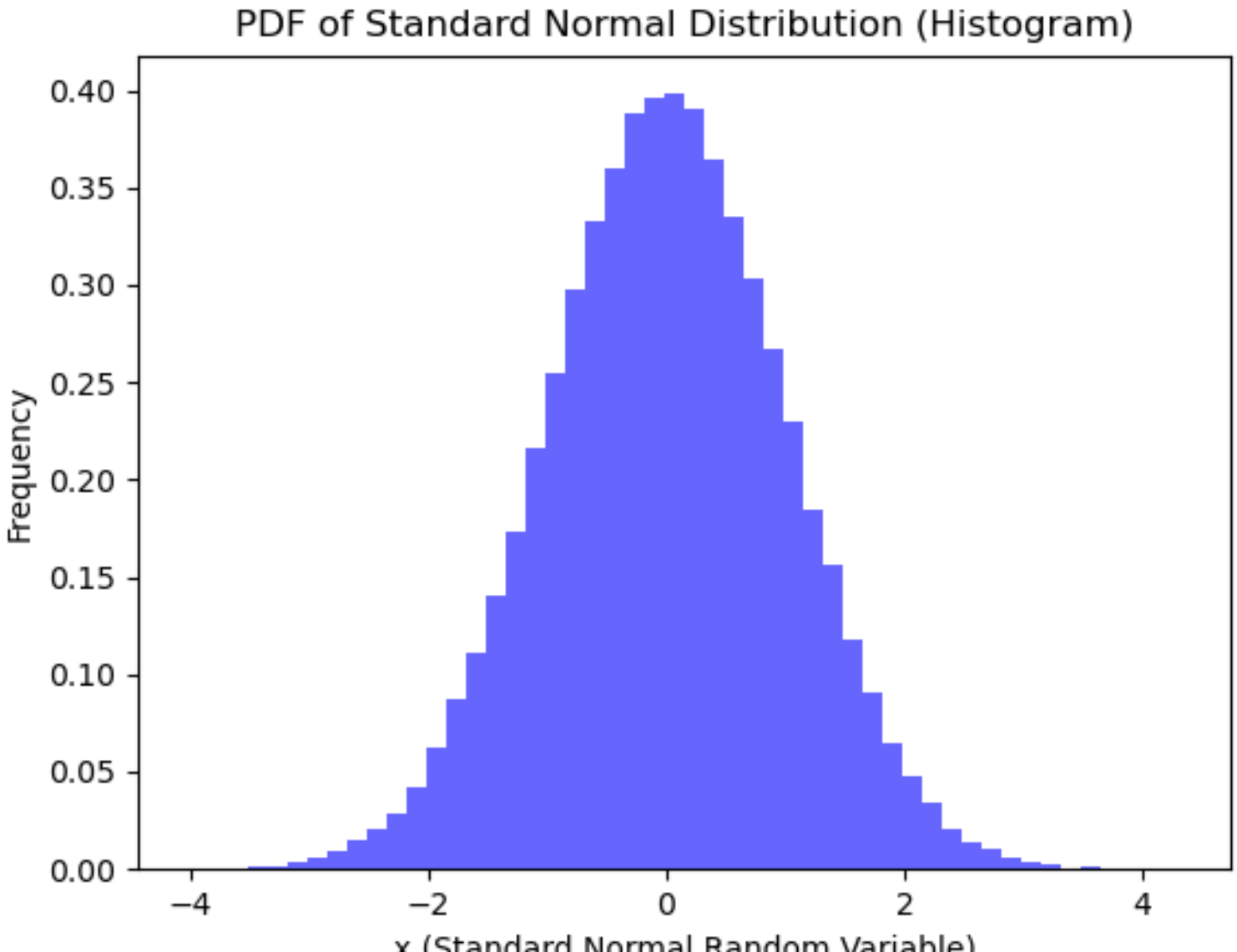
C-code III

```
28            printf("Unable to open file for
                  writing.\n");
29            return 1;
30        }
31
32        for (int i = 0; i < numSamples; i++) {
33            double sample = sn();
34            fprintf(file, "%lf\n", sample);
35        }
36
37        fclose(file);
38
39        return 0;
40    }
```

Python-code

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  # Read data from the "uni.dat" file
5  x = np.genfromtxt("uni.dat")
6
7  # Create a histogram plot of the PDF
8  plt.hist(x, bins=50, density=True, alpha=0.6,
       color='b', label='PDF (Histogram)')
9  plt.xlabel('x (Standard Normal Random Variable
       )')
10 plt.ylabel('Frequency')
11 plt.title('PDF of Standard Normal Distribution
       (Histogram)')
12 plt.savefig('/home/sayyam/KHP/figs/figure1.png
       ')
```

Python-code


PDF of Standard Normal Distribution (Histogram)
Frequency
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−2
0
2
4
x (Standard Normal Random Variable)

Problem Statement

Let $\phi(.)$ denote the cumulative distribution function of a standard normal random variable. If the random variable X has the cumulative distribution function

$$F(x) = \begin{cases} \phi(x), & x < -1 \\ \phi(x+1), & x \geq -1 \end{cases} \tag{3}$$

then which one of the following statements is true?

1. $P(X \leq -1) = \frac{1}{2}$
2. $P(X = -1) = \frac{1}{2}$
3. $P(X < -1) = \frac{1}{2}$
4. $P(X \leq 0) = \frac{1}{2}$

C-code I

```
1  #include <stdio.h>
2  #include <math.h>
3  #define M_PI 3.14159265358979323846
4
5  double cdf(double x) {
6      if (x < -1) {
7          return 0.5 * (1.0 + erf(x / sqrt(2.0))
               );
8      }
9      else {
10         return 0.5 * (1.0 + erf(x+1 / sqrt
               (2.0)));
11     }
12 }
13
```

C-code II

```
14 double pdf(double x) {
15     if (x < -1) {
16         return 1.0 / (sqrt(2.0 * M_PI)) * exp
               (-x * x / 2.0);
17     } else {
18         return 1.0 / (sqrt(2.0 * M_PI)) * exp
               (-(x + 1) * (x + 1) / 2.0);
19     }
20 }
21
22 int main() {
23     int num = 10000;
24     double x_max = 4.0;
25     double x_min = -4.0;
26     double step = (x_max - x_min) / num;
```

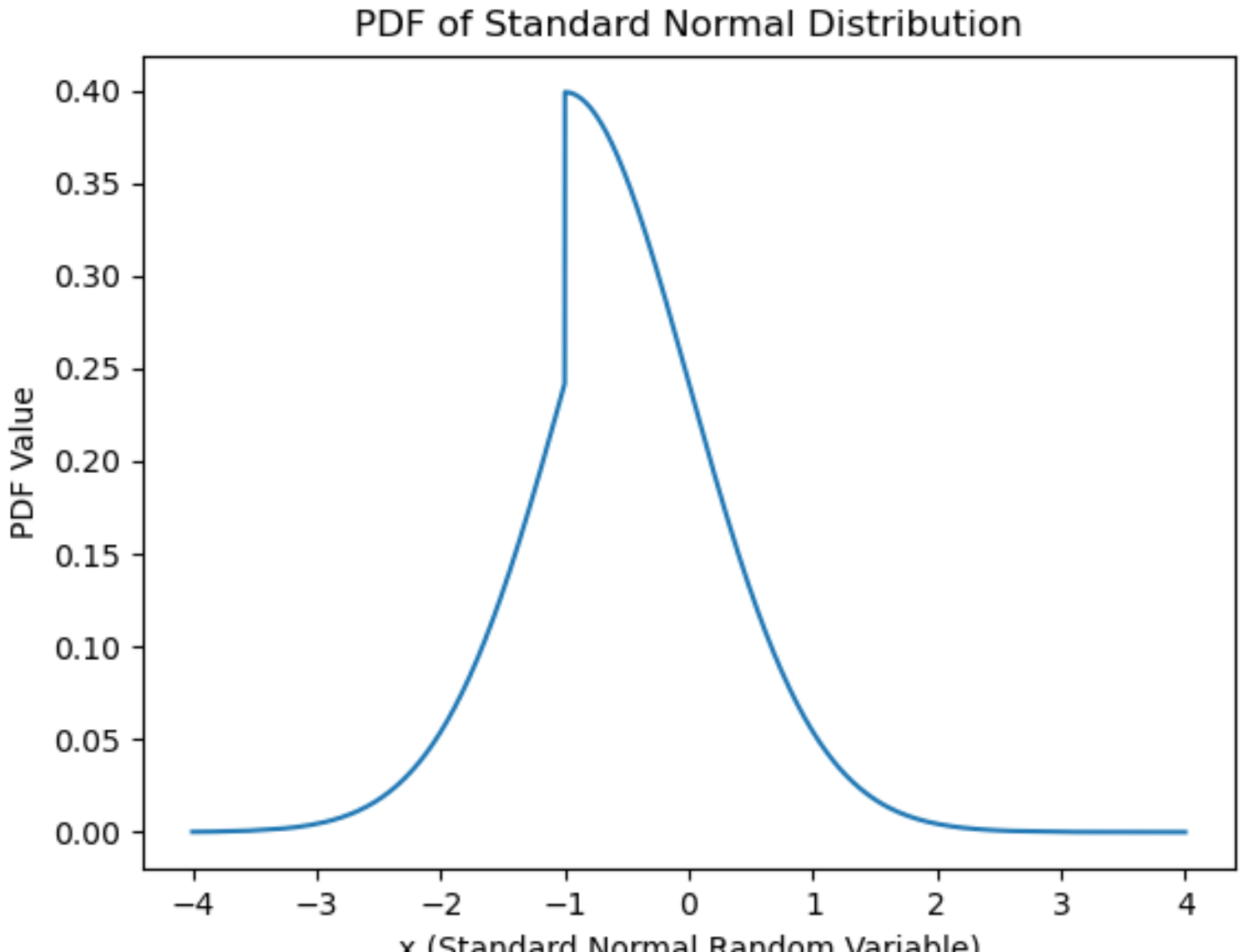
C-code III

```
27
28      FILE *outfile = fopen("uni.dat", "w");
29
30      for (int i = 0; i < num; i++) {
31          double x = x_min + step * i;
32          double sample = pdf(x);
33          fprintf(outfile, "%lf %lf\n", x,
                sample);
34      }
35
36      fclose(outfile);
37
38      return 0;
39  }
```

Python-code

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  # Read data from the "uni.dat" file
5  data = np.genfromtxt("uni.dat")
6
7  x = data[:, 0]  # Use correct index for
       columns (0 for x)
8  pdf = data[:, 1]  # Use correct index for
       columns (1 for pdf)
9
10 # Create a histogram plot of the PDF
11 plt.plot(x, pdf)
12 plt.xlabel('x (Standard Normal Random Variable
       )')
13 plt.ylabel('PDF Value')
```

Python-code


PDF of Standard Normal Distribution
PDF Value
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−3
−2
−1
0
1
2
3
4
x (Standard Normal Random Variable)