Probability with Coding

Karthikeya Hanu Prakash Kanithi

IIT Hyd

October 18, 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

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

- and

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

- 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

Proof

- Let X and Y be independent standard normal variables

$$X, Y \sim \mathcal{N}(0,1) \quad and \quad X \perp Y \tag{3}$$

- The joint pdf of X and Y is given by

$$\begin{aligned} f_{XY}(X,Y) &= f(x)f(y) \qquad (4)\\ &= \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} . \frac{1}{\sqrt{2\pi}} e^{\frac{-y^2}{2}} \qquad (5)\\ &= \frac{1}{\sqrt{2\pi}} e^{\frac{-(x^2+y^2)}{2}} \qquad (6) \end{aligned}$$

Proof

- The relationship between Cartesian coordinates (x, y) and polar coordinates (r, θ) is as follows

$$x = r\cos\theta \tag{7}$$

$$y = r\sin\theta \tag{8}$$

- Change $f_{XY}(x, y)$ to polar coordinates :

$$f_{XY}(x, y)dxdy = f_{R\theta}(r, \theta)drd\theta \tag{9}$$

- i.e.,

$$f_{R\theta}(r, \theta) = f_{XY}(x, y)\frac{dxdy}{drd\theta} \tag{10}$$

Proof

- where J is the Jacobian

$$J = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{vmatrix} \tag{11}$$

$$= \begin{vmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{vmatrix} = r \tag{12}$$

- Change $f_{XY}(x, y)$ to polar coordinates :

$$f_{XY}(x, y)dxdy = f_{R\theta}(r, \theta)drd\theta \tag{13}$$

- i.e.,

$$f_{R\theta}(r, \theta) = f_{XY}(x, y)\frac{dxdy}{drd\theta} \tag{14}$$

C-code I

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

double sn() {
    double u1, u2;
    while (1) {
        u1 = ((double)rand() / RAND_MAX);
        if (u1 > 0 && u1 < 1)
            break;}
    while (1) {
        u2 = ((double)rand() / RAND_MAX);
        if (u2 > 0 && u2 < 1)
            break;}
```

C-code II

```c
    double z1 = sqrt(-2 * log(u1)) * cos(2 *
        M_PI * u2);
    return z1;
}

int main() {
    // Seed the random number generator
    srand(time(0));

    int numSamples = 100000; // You can change
        this to the desired number of samples
    FILE *file = fopen("uni.dat", "w");

    if (file == NULL) {
```

C-code III

```
        printf("Unable to open file for writing.\n");
        return 1;
    }

    for (int i = 0; i < numSamples; i++) {
        double sample = sn();
        fprintf(file, "%lf\n", sample);
    }

    fclose(file);

    return 0;
}
```

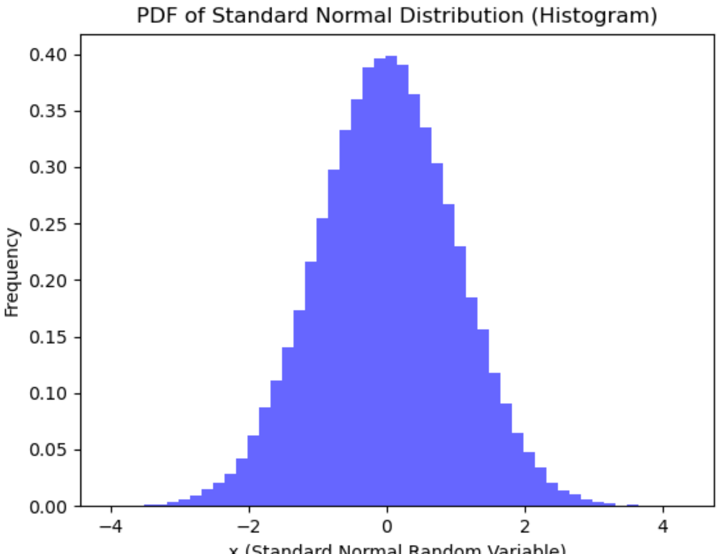
Python-code

```python
import numpy as np
import matplotlib.pyplot as plt

# Read data from the "uni.dat" file
x = np.genfromtxt("uni.dat")

# Create a histogram plot of the PDF
plt.hist(x, bins=50, density=True, alpha=0.6, color='b', label='PDF (Histogram)')
plt.xlabel('x (Standard Normal Random Variable)')
plt.ylabel('Frequency')
plt.title('PDF of Standard Normal Distribution (Histogram)')
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

$$F(x) = \begin{cases} \phi(x), & x < -1 \\ \phi(x+1), & x \geq -1 \end{cases} \tag{15}$$

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

```c
#include <stdio.h>
#include <math.h>
#define M_PI 3.14159265358979323846

double cdf(double x) {
    if (x < -1) {
        return 0.5 * (1.0 + erf(x / sqrt(2.0)));
    }
    else {
        return 0.5 * (1.0 + erf(x+1 / sqrt(2.0)));
    }
}

```

C-code II

```c
double pdf(double x) {
    if (x < -1) {
        return 1.0 / (sqrt(2.0 * M_PI)) * exp(-x * x / 2.0);
    } else {
        return 1.0 / (sqrt(2.0 * M_PI)) * exp(-(x + 1) * (x + 1) / 2.0);
    }
}

int main() {
    int num = 10000;
    double x_max = 4.0;
    double x_min = -4.0;
    double step = (x_max - x_min) / num;
```

C-code III

```c

    FILE *outfile = fopen("uni.dat", "w");

    for (int i = 0; i < num; i++) {
        double x = x_min + step * i;
        double sample = pdf(x);
        fprintf(outfile, "%lf %lf\n", x,
            sample);
    }

    fclose(outfile);

    return 0;
}
```

Python-code

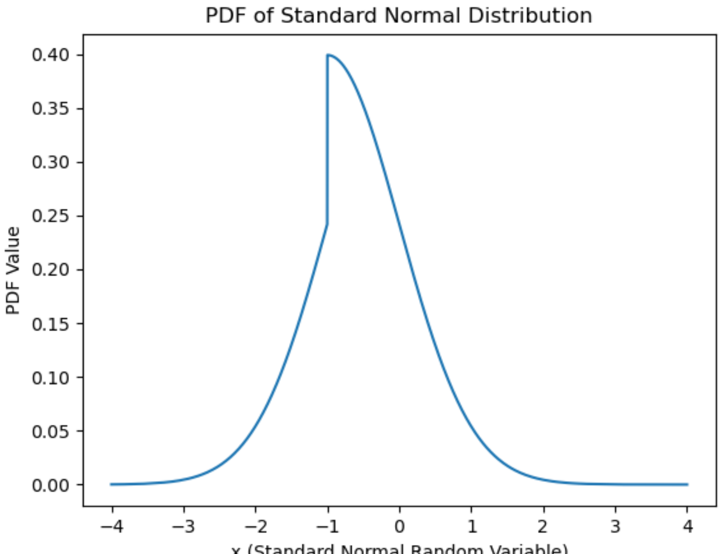
```python
import numpy as np
import matplotlib.pyplot as plt

# Read data from the "uni.dat" file
data = np.genfromtxt("uni.dat")

x = data[:, 0]  # Use correct index for columns (0 for x)
pdf = data[:, 1]  # Use correct index for columns (1 for pdf)

# Create a histogram plot of the PDF
plt.plot(x, pdf)
plt.xlabel('x (Standard Normal Random Variable)')
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)