# Laboratory 6: Random Variate Generator using lnverse-Transfonn Technique

1. Introduction and Purpose of Experiment
2. Aim and Objectives
3. Experimental Procedure

Design and implement a Java program to determine a sequence of 10 random variates by generating a sequence of random numbers using the flowing distributions:

* + 1. Uniform distribution in the interval [10, 20]
    2. Exponential distribution with mean value 5
    3. Normal distribution with mean 3.1 and sigma 0.6

1. Algorithms
2. Presentation of Results

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package lab06;

import java.util.Random;

import java.util.Scanner;

/\*\*

 \*

 \* @author shadowleaf

 \*/

public class Lab06 {

    /\*\*

     \* @param args the command line arguments

     \*/

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        Random random = new Random();

        int N = 10;

        double []R = new double[N];

        double []X = new double[N];

        for (int i = 0 ; i < R.length ; i++) {

            R[i] = random.nextDouble();

        }

        boolean done = false;

        System.out.print("Random Variate Generator\n"

                + "1.\tUniform Distribution\n"

                + "2.\tExponential Distribution\n"

                + "3.\tNormal Distribution\n"

                + "Your Choice : ");

        switch(input.nextInt()) {

            case 1: {

                int a, b;

                System.out.print("Enter value of a : ");

                a = input.nextInt();

                System.out.print("Enter value of b : ");

                b = input.nextInt();

                for (int i = 0 ; i < X.length ; i++) {

                    X[i] = a + (b-a) \* R[i];

                }

                done = true;

            }

                break;

            case 2: {

                double lambda;

                System.out.print("Enter the value of lambda : ");

                lambda = input.nextDouble();

                for (int i = 0 ; i < X.length ; i++) {

                    X[i] = Math.log(R[i]) / (-lambda);

                }

                done = true;

            }

                break;

            case 3: {

                double meu, sigma;

                System.out.print("Enter the value of meu : ");

                meu = input.nextDouble();

                System.out.print("Enter the value of sigma : ");

                sigma = input.nextDouble();

                for (int i = 0 ; i < X.length ; i++) {

                    X[i] = (Math.pow(R[i], 0.135) - Math.pow(1-R[i], 0.135) ) / 0.1975;

                    X[i] = X[i] \* sigma + meu;

                }

                done = true;

            }

                break;

        }

        if (done) {

            System.out.println("\nThe Random Variates are  : ");

            for (int i = 0 ; i < X.length ; i++) {

                System.out.print(X[i]+"\n");

            }

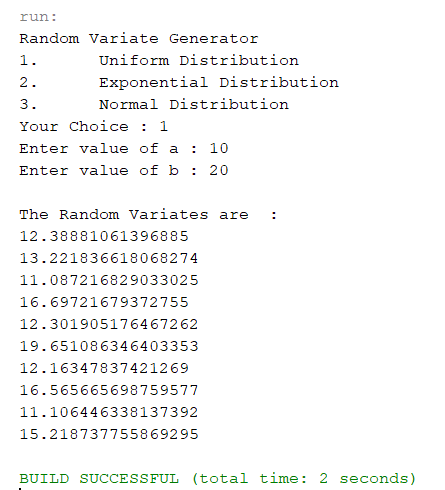
            System.out.println();

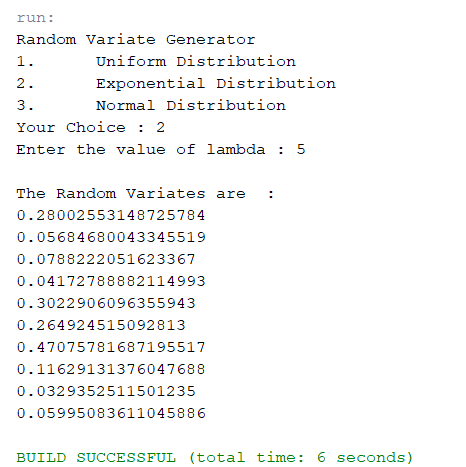
        }

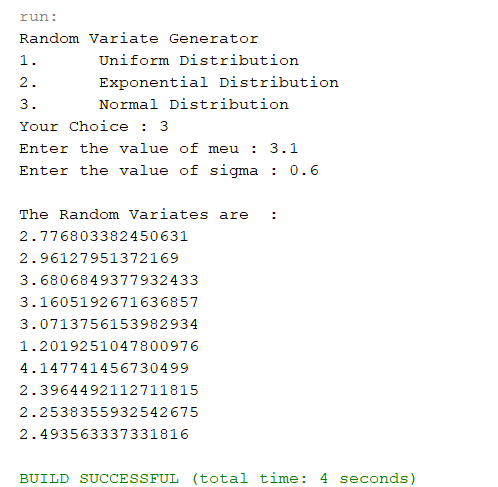
    }

}

1. Analysis and Discussions







1. Conclusions

Standard Normal Distribution

The inverse CDF of the standard normal distribution is approximated as:

Example usage for ,

Then take

Taking example of and

Then

Exponential Distribution

Or

Uniform Distribution