

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Лабораторная работа 1

Выполнил:

Анищенко Арсений

4 курс 3 группа

Преподаватель:

Кирлица Валерий Петрович

М и н с к

2019

Условие

Вариант 9 (задание 4 страница 52)

Контролируемый признак ξ распределен по закону $N_1(\mu\sigma^2)$. Множество допустимых значений признака ограничено полем допуска $[a,b]$ ($\mu=(a+b)/2$). Оценить вероятность выхода за поле допуска по $n=1000$ реализациям СВ ξ . Рассмотреть случаи:

- A. Увеличение степени рассеяния. $a=57$, $b=63$, $\sigma=1,2,3$
- B. Смещение средних $a=57$, $b=63$, $\sigma=1$, $\mu=58,65$

Ход работы

Теория

Базовую случайную величину можем смоделировать используя генератор Макларена — Марсальи, основанного на мультипликативном конгруэнтном методе:

$$\alpha_i = \alpha_i' / M$$

$$\alpha_i' = \beta \alpha_{i-1}' - 1' \bmod M$$

$$i = 1, 2, \dots$$

$$\alpha_0' = \beta = 65539$$

$$M = 2147483648$$

Нормальное распределение получим при помощи моделирования случайной базовой величины:

$$\eta = \sqrt{-2 \ln a_1} \cos(2\pi a_2)$$

Реализация

```

/// <summary>
/// MacLaren-Marsaglia generator for base random variable
/// </summary>
3 references
public class MGenerator
{
    private double _alpha;
    private double _beta;
    private double _m;

    1 reference
    public MGenerator(int seed = 65539)
    {
        _alpha = seed;
        _beta = 65539;
        _m = 2147483648;
    }

    1 reference
    public double NextRand()
    {
        _alpha = (_alpha * _beta) % _m;

        return _alpha / _m;
    }
}

```

Класс реализующий генерацию базовой случайной величины методом Макларена — Марсальи.

```

/// <summary>
/// Normally distributed random generator
/// </summary>
2 references
public class NormalGenerator
{
    private readonly MGenerator _mGenerator;

    2 references
    public double StandardDeviation { get; private set; }
    2 references
    public double Expectation { get; private set; }

    1 reference
    public NormalGenerator(double expectation, double standardDeviation, int seed = 65539)
    {
        _mGenerator = new MGenerator(seed);

        Expectation = expectation;
        StandardDeviation = standardDeviation;
    }

    1 reference
    public double NextRand()
    {
        var baseVariable = _mGenerator.NextRand();

        return Expectation + StandardDeviation * Math.Sqrt(2)
            * AdvancedMath.InverseErf(2 * baseVariable - 1);
    }
}

```

Класс реализующий генерацию СВ обладающую нормальным распределением, основываясь на базовой СВ.

```

0 references
class Program
{
    2 references
    static double ConductExperiment(int a, int b, double expectation, double standardDeviation, int n = 1000)
    {
        var errors = 0;
        var generator = new NormalGenerator(expectation, standardDeviation);

        for (int i = 0; i < n; ++i)
        {
            var rand = generator.NextRand();
            if (rand < a || rand > b)
                ++errors;
        }

        return (double)errors / n;
    }

    0 references
    static void Main()
    {
        int a = 57;
        int b = 63;
        var deviations = new List<int> { 1, 2, 3 };

        foreach (var deviation in deviations)
        {
            Console.WriteLine($"Standard deviation {deviation}: {ConductExperiment(a, b, (a + b) / 2, deviation)}");
        }

        var expectations = new List<int> { 58, 65 };

        foreach (var expectation in expectations)
        {
            Console.WriteLine($"Expectation {expectation}: {ConductExperiment(a, b, expectation, 1)}");
        }

        Console.ReadKey();
    }
}

```

Тестовый класс.

Результаты

A

Standard deviation 1: 0,005

Standard deviation 2: 0,133

Standard deviation 3: 0,304

B

Expectation 58: 0,15

Expectation 65: 0,974

Оценка

Нетрудно видеть, что при увеличении степени рассеяния вероятность выйти за пределы заданного промежутка увеличивается, и при смещении среднего значения вероятность выйти за пределы заданного промежутка также увеличивается.