Министерство науки и высшего образования Российской Федерации  
Рязанский государственный радиотехнический университет  
Кафедра вычислительной и прикладной математики

# ЛАБОРАТОРНАЯ РАБОТА №4 ГЕНЕРИРОВАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН С НОРМАЛЬНЫМ ЗАКОНОМ РАСПРЕДЕЛЕНИЯ

**Выполнил:**  
студент 4 курса группы 843  
Попов М.С.

**Проверил:** д-р техн. наук,

профессор каф. ВПМОвечкин Г.В.

Рязань, 2022 год

**Задание:**

СОСТАВИТЬ ПОДПРОГРАММУ ГЕНЕРИРОВАНИЯ СЛУЧАЙНЫХ ВЕЛИЧИН С НОРМАЛЬНЫМ ЗАКОНОМ РАСПРЕДЕЛЕНИЯ МЕТОДОМ, ОСНОВАННЫМ НА ЦЕНТРАЛЬНОЙ ПРЕДЕЛЬНОЙ ТЕОРЕМЕ, А ТАКЖЕ МЕТОДОМ, ОПРЕДЕЛЕННЫМ В СООТВЕТСТВИИ С ВАРИАНТОМ ЗАДАНИЯ (ТАБЛ. 4).  
1) ПО ПОЛУЧЕННОЙ С ПОМОЩЬЮ ПОДПРОГРАММЫ ВЫБОРКЕ ПОСТРОИТЬ И ПРОАНАЛИЗИРОВАТЬ ГИСТОГРАММУ ЧАСТОТ И СТАТИСТИЧЕСКУЮ ФУНКЦИЮ РАСПРЕДЕЛЕНИЯ  
2) ОЦЕНИТЬ МАТОЖИДАНИЕ И ДИСПЕРСИЮ СЛУЧАЙНОЙ ВЕЛИЧИНЫ.  
3) СООТВЕТСТВИЕ ЭМПИРИЧЕСКИХ ДАННЫХ ТЕОРЕТИЧЕСКОМУ РАСПРЕДЕЛЕНИЮ ПРОВЕРИТЬ С ПОМОЩЬЮ КРИТЕРИЯ ПИРСОНА ИЛИ КРИТЕРИЯ КОЛМОГОРОВА.  
ОБЪЕМ ВЫБОРКИ СЛУЧАЙНЫХ ВЕЛИЧИН НЕ МЕНЕЕ 1000.  
КОЛИЧЕСТВО ИНТЕРВАЛОВ РАЗБИЕНИЯ K = 15 ИЛИ K = 25.  
ТЕОРЕТИЧЕСКАЯ ЧАСТЬ ДЛЯ ДАННОЙ ЛАБОРАТОРНОЙ РАБОТЫ ПРЕДСТАВЛЕНА В УЧЕБНИКЕ [1] НА СТР. 76–83.

**C:\Users\matyh\Desktop\КМ\Лабараторные работы КМ\assets\img\varLabFouth.png**

**Выполнение:**

**Скрипт, для выполнения поставленных задач**

/\* Переменные \*/

let arrayNormalRasprX = []; // Массив с нормальным законом распределнения X

let arrayNormalRasprY = []; // Массив с нормальным законом распределнения Y

let firstHistogrammData = {}; // Данные для гистограммы

// Параметры выборки

const DELTA = 67; // Дельта

const SAMPLE\_SIZE = 1000; // Размер выборки

const AMOUNT\_INTERVALS = 15; // Количество интервалов

const SIGMA = -1.5; // Сигма

const MU = 1.7; // Мю

// Получение нормального закона распределения

function getNormalRaspr(arrayFirst, arraySecond, n) {

let i = 0;

let numb = 0;

let step = 0;

while (i < n) {

r1 = Math.random();

r2 = Math.random();

u1 = -1 + 2 \* r1;

u2 = -1 + 2 \* r2;

s = u1 \* u1 + u2 \* u2;

if (s <= 1) {

arrayFirst[i] = u1 \* Math.sqrt(-2 \* Math.log(s) / s);

arraySecond[i] = u2 \* Math.sqrt(-2 \* Math.log(s) / s);

arrayFirst[i] = -1.5 + (arrayFirst[i] \* Math.sqrt(0.8));

arraySecond[i] = 2.5 + (arraySecond[i] \* Math.sqrt(1.96));

// numb = (1 / (SIGMA \* Math.sqrt(2 \* Math.PI))) \* Math.E;

// step = -1 \* ((Math.pow((u1 - MU), 2)) / (2 \* (SIGMA \* SIGMA)));

// arrayFirst[i] = Math.pow(numb, step);

// console.log(arrayFirst[i], " ", numb, " ", step);

// console.log(i + 0, " ", arrayFirst[i], " ", arraySecond[i], " ");

i++;

}

}

}

// function getNormalRaspr(array, sigma, mu) {

// }

getNormalRaspr(arrayNormalRasprX, arrayNormalRasprY, SAMPLE\_SIZE);

arraySort(arrayNormalRasprX);

arraySort(arrayNormalRasprY);

/\*ГРАФИК ФУНКЦИИ\*/

//Готовим диаграмму

function Diagram() {

var ctx = document.getElementById("myChart");

var myChart = new Chart(ctx, {

type: 'line',

data: {

labels: [], //Подписи оси x

datasets: [

{

label: 'f(x)', //Метка

data: [], //Данные

borderColor: 'blue', //Цвет

borderWidth: 2, //Толщина линии

fill: false //Не заполнять под графиком

},

{

label: 'f(y)', //Метка

data: [], //Данные

borderColor: 'red', //Цвет

borderWidth: 2, //Толщина линии

fill: false //Не заполнять под графиком

},

//Можно добавить другие графики

]

},

options: {

responsive: false, //Вписывать в размер canvas

scales: {

xAxes: [{

display: true

}],

yAxes: [{

display: true

}]

}

}

});

//Заполняем данными

for (var i = 0.0; i < SAMPLE\_SIZE; i++) {

myChart.data.labels.push('' + arrayNormalRasprX[i].toFixed(2));

myChart.data.datasets[0].data.push(f(arrayNormalRasprX[i]).toFixed(2));

myChart.data.datasets[1].data.push(f(arrayNormalRasprY[i]).toFixed(2));

}

//Обновляем

myChart.update();

//Вычисление нужной функции

function f(x) {

return x;

}

}

// Вывод в консоль

function write(arg) {

console.log(arg);

}

// Сортировка массива

function arraySort(array) {

array.sort();

}

//Ставим загрузку диаграммы на событие загрузки страницы

window.addEventListener("load", Diagram);