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

# ЛАБОРАТОРНАЯ РАБОТА №3 ГЕНРИРОВАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН С ЗАДАННЫМ ЗАКОНОМ РАСПРЕДЕЛЕНИЯ ГЕНЕРАТОРОВ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

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

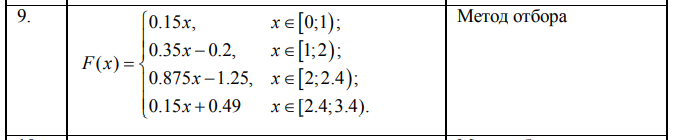
**Проверил:** д-р техн. наук,

профессор каф. ВПМОвечкин Г.В.

Рязань, 2022 год

**Задание:**

Составить подпрограмму генерирования случайных величин в соответствии с вариантом задания, определяемым таблицей 3. По полученной с помощью подпрограммы выборке построить и проанализировать гистограмму частот и статистическую функцию распределения, оценить матожидание и дисперсию случайной величины. 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. 65–76.

****

**Выполнение:**

**Скрипт, для выполнения поставленных задач**

/\*Переменные\*/

// Массивы случайных чискл

let arrayFirst = [];

let arraySecond = [];

let arrayThird = [];

let arrayFouth = [];

//Массивы правых границ интервалов

let arrayRightBorderFirst = [];

let arrayRightBorderSecond = [];

let arrayRightBorderThird = [];

let arrayRightBorderFouth = [];

//Массивы левых границ интервалов

let arrayLeftBorderFirst = [];

let arrayLeftBorderSecond = [];

let arrayLeftBorderThird = [];

let arrayLeftBorderFouth = [];

//

let delta = 67;

let n = 1000;

let k = 15;

// Ожидание прогрузки страницы

$(document).ready(function () {

});

// Первая функция

function getFirstRandomarray(array, leftBorder, rightBorder) {

for (let i = 0; i < n; i++) {

// максимальное значение плотности вероятности

let C = 0.15;

let f = true;

let r1 = 0.00;

let r2 = 0.00;

let z1 = 0.00;

let z2 = 0.00;

while (f) {

r1 = Math.random();

r2 = Math.random();

z1 = leftBorder + r1 \* (rightBorder - leftBorder);

z2 = C \* r2;

if (z2 < (0.15 \* z1)) {

array[i] = z1;

f = false;

}

}

}

}

// Вторая функция

function getSecondRandomarray(array, leftBorder, rightBorder) {

for (let i = 0; i < n; i++) {

// максимальное значение плотности вероятности

let C = 0.5;

let f = true;

let r1 = 0.00;

let r2 = 0.00;

let z1 = 0.00;

let z2 = 0.00;

while (f) {

r1 = Math.random();

r2 = Math.random();

z1 = leftBorder + r1 \* (rightBorder - leftBorder);

z2 = C \* r2;

if (z2 < ((0.35 \* z1) - 0.2)) {

array[i] = z1;

f = false;

}

}

}

}

// Третья функция

function getThirdRandomarray(array, leftBorder, rightBorder) {

for (let i = 0; i < n; i++) {

// максимальное значение плотности вероятности

let C = 0.85;

let f = true;

let r1 = 0.00;

let r2 = 0.00;

let z1 = 0.00;

let z2 = 0.00;

while (f) {

r1 = Math.random();

r2 = Math.random();

z1 = leftBorder + r1 \* (rightBorder - leftBorder);

z2 = C \* r2;

if (z2 < ((0.875 \* z1) - 1.25)) {

array[i] = z1;

f = false;

}

}

}

}

// Четвёртая функция

function getFouthRandomarray(array, leftBorder, rightBorder) {

for (let i = 0; i < n; i++) {

// максимальное значение плотности вероятности

let C = 1;

let f = true;

let r1 = 0.00;

let r2 = 0.00;

let z1 = 0.00;

let z2 = 0.00;

while (f) {

r1 = Math.random();

r2 = Math.random();

z1 = leftBorder + r1 \* (rightBorder - leftBorder);

z2 = C \* r2;

if (z2 < ((0.15 \* z1) + 0.49)) {

array[i] = z1;

f = false;

}

}

}

}

// Сортировка массива

function arraySort(array) {

array.sort();

}

// Получить величину одного интервала

function getInerval(array, part) {

return diffrent = (array[n - 1] - array[0]) / part;

}

// Получить массив левой границы интервалов

function getLeftBoardForNormArray(array, a, k, koefInt) {

for (let i = 0; i < k; i++) {

array[i] = (a \* i) + koefInt;

}

}

// Получить массив правой границы интервалов

function getRightBoardForNormArray(array, a, k, koefInt) {

for (let i = 0; i < k; i++) {

array[i] = (a + a \* i) + koefInt;

}

}

// Получить оценку вероятностей попадания случайной величины на всех интервалах

function getpSluchValue(amountArray, array, intervAmount, rightBoardForNormFunction, leftBoardForNormFunction) {

for (let i = 0; i < intervAmount; i++) {

amountArray[i] = 0;

for (let j = 0; j < n; j++) {

if ((array[j] < rightBoardForNormFunction[i]) & (array[j] >= leftBoardForNormFunction[i])) {

amountArray[i] = amountArray[i] + 1;

}

}

}

}

/\*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\*0.15)', //Метка

data: [], //Данные

borderColor: 'blue', //Цвет

borderWidth: 2, //Толщина линии

fill: false //Не заполнять под графиком

},

{

label: 'f(0.35\*x-0.2)', //Метка

data: [], //Данные

borderColor: 'red', //Цвет

borderWidth: 2, //Толщина линии

fill: false //Не заполнять под графиком

},

{

label: 'f(0.875\*x-1.25)', //Метка

data: [], //Данные

borderColor: 'green', //Цвет

borderWidth: 2, //Толщина линии

fill: false //Не заполнять под графиком

},

{

label: 'f(0.15\*x+0.49)', //Метка

data: [], //Данные

borderColor: 'yellow', //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

}]

}

}

});

getFirstRandomarray(arrayFirst, 0, 1);

getSecondRandomarray(arraySecond, 1, 2);

getThirdRandomarray(arrayThird, 2, 2.4);

getFouthRandomarray(arrayFouth, 2.4, 3.4);

//Заполняем данными

for (var i = 0.0; i < n; i++) {

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

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

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

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

myChart.data.datasets[3].data.push(f(arrayFouth[i]).toFixed(2));

}

//Обновляем

myChart.update();

//Вычисление нужной функции

function f(x) {

return x;

}

//Сортировка массивов

arraySort(arrayFirst);

arraySort(arraySecond);

arraySort(arrayThird);

arraySort(arrayFouth);

// console.log(arrayFouth);

// console.log(getInerval(arraySecond, k));

// Получаем правую границу интервалов

getRightBoardForNormArray(arrayRightBorderFirst, getInerval(arrayFirst, k), k, 0);

getRightBoardForNormArray(arrayRightBorderSecond, getInerval(arraySecond, k), k, 1);

getRightBoardForNormArray(arrayRightBorderThird, getInerval(arrayThird, k), k, 2);

getRightBoardForNormArray(arrayRightBorderFouth, getInerval(arrayFouth, k), k, 2.4);

// Получаем левую границу интервалов

getLeftBoardForNormArray(array, a, k, koefInt)

console.log(arrayRightBorderFirst);

console.log(arrayFirst);

// console.log(arrayRightBorderSecond);

// console.log(arraySecond);

// console.log(arrayRightBorderThird);

// console.log(arrayThird);

// console.log(arrayRightBorderFouth);

// console.log(arrayFouth);

// let arrayLeftBorderFirst = [];

// let arrayLeftBorderSecond = [];

// let arrayLeftBorderThird = [];

// let arrayLeftBorderFouth = [];

// fillFirstGistogrammData(keys, pSluchValue, k)

// Заполнить первую гистограмму данными

// fillFirstGistogrammData(keys, pSluchValue, k);

function fillFirstGistogrammData(keys, pSluchValue, k) {

for (let i = 0; i < k; i++) {

let key = keys[i];

let value = pSluchValue[i];

firstHistogrammData[key] = value;

}

}

}

//Ставим загрузку диаграммы на событие загрузки страницы

window.addEventListener("load", Diagram);

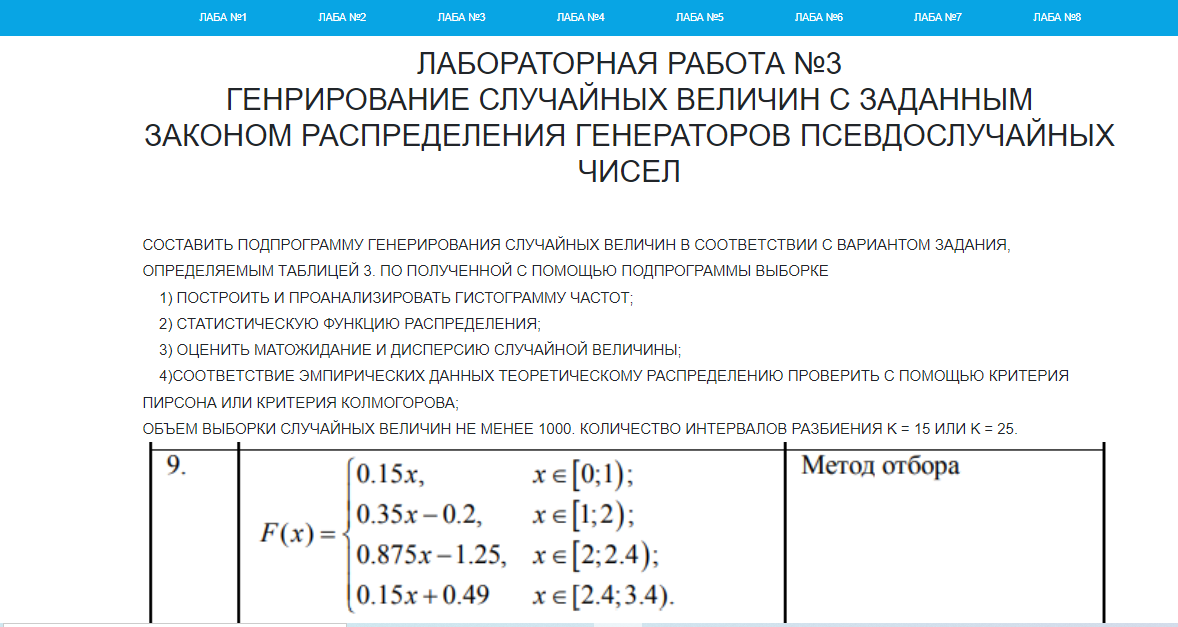
// Вывод значение в поле страницы

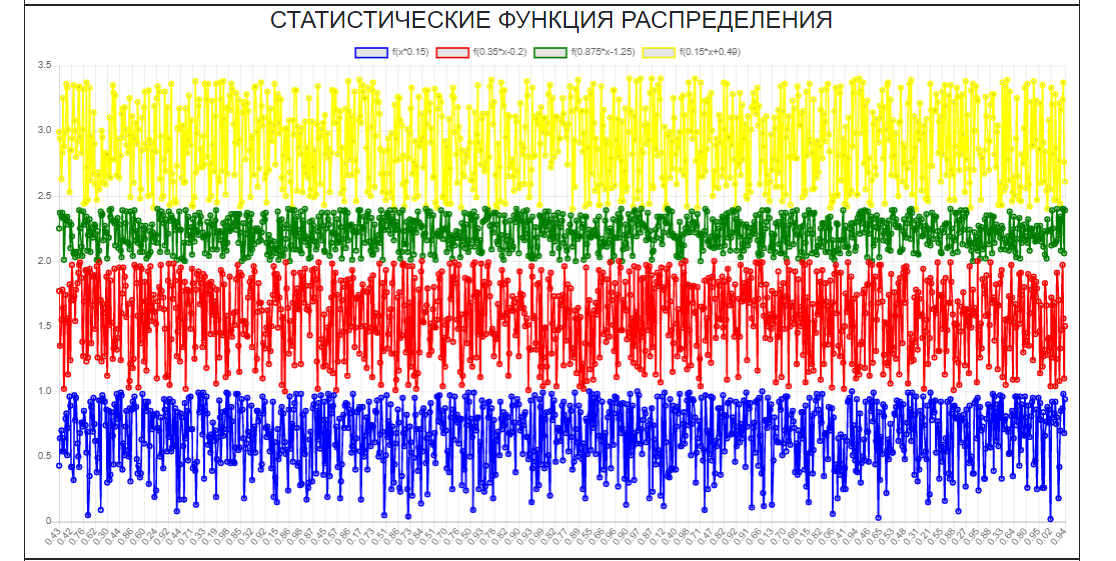
function writeInPage(value, field) {

$(field)[0].innerHTML = value;

}

**Результат:**

****

****