Лабораторна робота

з теорії ймовірності №4

Виконала: студент

ФІОТ групи ІО-33

Шуркіна Анастасія

Перевірив:

Марковський О. П.

Київ 2014

Тип розподілу: 25 Cімпсонів.

package lab4;

import java.util.\*;

import java.util.Random;

public class Simpson\_4 {

public static void main(String[] args) {

int l = 10, m = 1, a = 1, n = 1000, nsim = 25, k = 0;

double ss = 0, sd = 0, SS = 0;

double[] A = new double[nsim];

double[] R = new double[n];

double[] G = new double[l - 1];

double[] P = new double[l];

int[] K = new int[l];

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

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

Random r = new Random();

double x = r.nextFloat();

if (x > 0.5)

A[i] = a \* Math.sqrt(2 \* x) + (m - a);

else if (x < 0.5)

A[i] = (m + a) - a \* Math.sqrt(2 \* (1 - x));

else if (x == 0.5)

A[i] = m;

// System.out.println(A[i]);

ss += A[i];

}

R[j] += ss;

ss = 0;

SS += R[j];

}

Arrays.sort(R);

/\*

\* System.out.println(); double mos = ss / nsim;

\* System.out.println("MATH=" + mos); for (int i = 0; i < A.length; i++)

\* sd += Math.pow(A[i] - mos, 2); double disp = sd / nsim;

\* System.out.println("DISP=" + disp); double serkv = Math.sqrt(disp);

\* System.out.println("SER\_KV=" + serkv);}}

\*/

double MR = SS / n;

System.out.println("MATH MAIN=" + MR);

double SD = 0;

for (int j = 0; j < n; j++)

SD += Math.pow(R[j] - MR, 2);

double DR = SD / n;

System.out.println("DISP MAIN=" + DR);

double BR = Math.sqrt(DR);

System.out.println("SER\_KV MAIN=" + BR);

// for (int j = 0; j < n; j++)

// System.out.println(R[j]);

System.out.println("===========BOUND================");

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

G[i] = (R[(i + 1) \* 100 - 1] + R[(i + 1) \* 100]) / 2;

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

System.out.println("i is " + i + " bound is: " + G[i]);

System.out.println("========================");

System.out.println("=========PROBABILITY==========");

for (int i = 0; i < l; i++)

P[i] = 0;

double SP = 0;

P[0] = Laplace.laplace((G[0] - MR) / BR) + 0.5;

P[l - 1] = 0.5 - Laplace.laplace((G[l - 2] - MR) / BR);

for (int i = 1; i < l - 1; i++)

P[i] = Laplace.laplace((G[i] - MR) / BR)

- Laplace.laplace((G[i - 1] - MR) / BR);

for (int i = 0; i < l; i++) {

System.out.println(P[i]);

SP = SP + P[i];

}

System.out.println();

System.out.println("SP=" + SP);

System.out.println();

System.out.println("=================");

double xi = 0;

int amint = 100;

for (int i = 0; i < l; i++)

xi += Math.pow(amint, 2) / n / P[i];

xi = xi - n;

System.out.println("xi2\_I=" + xi);

System.*out*.println("=========Part II==========");

int N = 10, L = 3;

// double[] Arr = new double[N];

double Arr[] = { 25, 25.23, 25.06, 25.15, 25.2, 24.86, 24.75, 24.98,

24.79, 25.1 };

Arrays.*sort*(Arr);

for (int j = 0; j < N; j++)

System.*out*.println(Arr[j]);

double SumArr = 0;

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

SumArr += Arr[i];

double MathArr = SumArr / N;

System.*out*.println("MATH MAIN=" + MathArr);

double SumSqr = 0;

for (int i = 0; i < N; i++)

SumSqr += Math.*pow*(Arr[i] - MathArr, 2);

double DispArr = SumSqr / N;

System.*out*.println("DISP MAIN=" + DispArr);

double DevArr = Math.*sqrt*(DispArr);

System.*out*.println("SER\_KV MAIN=" + DevArr);

double[] Bound = new double[L + 1];

double[] Prob = new double[L];

int[] Amount = new int[L];

System.*out*.println("===========ГРАНИЦІ================");

double u = (Arr[N - 1] - Arr[0]) / L;

Bound[0] = Arr[0];

Bound[L] = Arr[N - 1];

for (int i = 1; i < Bound.length - 1; i++) {

Bound[i] = Arr[0] + i \* u;

}

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

System.*out*.println("i is " + i + " bound is: " + Bound[i]);

System.*out*.println("========================");

System.*out*.println("===========AMOUNT================");

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

for (int j = 0; j < Arr.length; j++) {

if ((Arr[j] >= Bound[i]) && (Arr[j] <= Bound[i + 1])) {

k = k + 1;

}

}

Amount[i] = k;

System.*out*.println("K=" + Amount[i]);

k = 0;

}

System.*out*.println("========================");

System.*out*.println("=========PROBABILITY==========");

double SumProb = 0;

for (int i = 0; i < L; i++) {

Prob[i] = Laplace.*laplace*((Bound[i + 1] - MathArr) / DevArr)

- Laplace.*laplace*((Bound[i] - MathArr) / DevArr);

System.*out*.println(Prob[i]);

// SumProb = SumProb + Prob[i];

}

System.*out*.println();

// System.out.println("SumProb=" + SumProb);

System.*out*.println();

System.*out*.println("=================");

xi = 0;

for (int i = 0; i < L; i++)

if (Prob[i] != 0)

xi = xi + Math.*pow*(Amount[i] - N \* Prob[i], 2) / (N \* Prob[i]);

System.*out*.println("xi2=" + xi);

}

}