

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

## ЛАБОРАТОРНА РОБОТА № 2

з дисципліни «Методи планування експерименту»

на тему «ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З  
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконав:  
студент 2 курсу  
групи ІВ-92  
Дмитришин А. Д.  
Номер у списку групи: 6

ПЕРЕВІРИВ:  
ас. Регіда П.Г.

## Хід роботи

**Мета:** провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.

### Завдання:

#### Завдання на лабораторну роботу

1. Записати лінійне рівняння регресії.
2. Обрати тип двофакторного експерименту і скласти матрицю планування для нього з використанням додаткового нульового фактору ( $x_0=1$ ).
3. Провести експеримент в усіх точках повного факторного простору (знайти значення функції відгуку  $y$ ). Значення функції відгуку задати випадковим чином у відповідності до варіанту у діапазоні  $y_{\min} \div y_{\max}$

$$y_{\max} = (30 - N_{\text{варіанту}}) * 10,$$

$$y_{\min} = (20 - N_{\text{варіанту}}) * 10.$$

Варіанти обираються по номеру в списку в журналі викладача.

№ варіанта	X <sub>1</sub>		X <sub>2</sub>	
	min	max	min	max
206	10	40	-15	35

4. Перевірити однорідності дисперсії за критерієм Романовського
5. Знайти коефіцієнти нормованих рівнянь регресії і виконати перевірку (підставити значення нормованих факторів і коефіцієнтів у рівняння).
6. Провести натуралізацію рівняння регресії й виконати перевірку натуралізованого рівняння.
7. Написати комп'ютерну програму, яка все це виконує.

## Лістинг

```
package com.company;

import java.util.Random;

public class Lab2 {
    static double avarege(int[] y) {
        double sum = 0;
        for (int i = 0; i < y.length; i++) {
            sum += y[i];
        }
        return sum / y.length;
    }

    static double[] dispersion(int y[][]) {
        double d[] = new double[y.length];
        for (int i = 0; i < y.length; i++) {
            double sum = 0;
            for (int j = 0; j < y[i].length; j++) {
                sum += Math.pow(y[i][j] - avarege(y[i]), 2);
            }
            d[i] = sum / y[i].length;
        }
        return d;
    }

    static double determinant(double a[][]) {
        return a[0][0] * a[1][1] * a[2][2] + a[0][1] * a[1][2] * a[2][0] + a[1][0] *
a[0][2] * a[2][1] - a[0][2] * a[1][1] * a[2][0] - a[0][1] *
        a[1][0] * a[2][2] - a[0][0] * a[1][2] * a[2][1];
    }

    public static void main(String[] args) {
        double ymax = (30 - 206) * 10;
        double ymin = (20 - 206) * 10;
        double x[][] = {{1, 1},
            {-1, 1},
            {1, -1}};
        double x1min = 10;
        double x1max = 40;
        double x2min = -15;
        double x2max = 35;
        double m = 5;
        Random random = new Random();
        int y[][] = new int[3][5];
        for (int i = 0; i < y.length; i++) {
            for (int j = 0; j < y[i].length; j++) {
                y[i][j] = random.nextInt((int) ymax - (int) ymin + 1) + (int) ymin;
            }
        }
        double yavr[] = {avarege(y[0]), avarege(y[1]), avarege(y[2])};

        double sigmateta = Math.sqrt((2 * (2 * m - 2)) / (m * (m - 4)));

        double Fuv[] = {Math.max(dispersion(y)[0], dispersion(y)[1]) /
Math.min(dispersion(y)[0], dispersion(y)[1]),
            Math.max(dispersion(y)[2], dispersion(y)[0]) /
Math.min(dispersion(y)[2], dispersion(y)[0]),
            Math.max(dispersion(y)[2], dispersion(y)[1]) /
Math.min(dispersion(y)[2], dispersion(y)[1])};

        double Teta[] = {(m - 2) / m * Fuv[0], (m - 2) / m * Fuv[1], (m - 2) / m *
Fuv[2]};
    }
}
```

```

        double Ruv[] = {Math.abs(Teta[0] - 1) / sigmateta, Math.abs(Teta[1] - 1) /
sigmateta,
        Math.abs(Teta[2] - 1) / sigmateta};

        if ((Ruv[0] > 2) || (Ruv[1] > 2) || (Ruv[2] > 2)) {
            System.out.println("Неоднорідна дисперсія");
            System.exit(1337);
        }
        double mx1 = (x[0][0] + x[1][0] + x[2][0]) / 3;
        double mx2 = (x[0][1] + x[1][1] + x[2][1]) / 3;
        double my = (yavr[0] + yavr[1] + yavr[2]) / 3;
        double a1 = (Math.pow(x[0][0], 2) + Math.pow(x[1][0], 2) + Math.pow(x[2][0],
2)) / 3;
        double a2 = (x[0][0] * x[0][1] + x[1][0] * x[1][1] + x[2][0] * x[2][1]) / 3;
        double a3 = (Math.pow(x[0][1], 2) + Math.pow(x[1][1], 2) + Math.pow(x[2][1],
2)) / 3;
        double a11 = (x[0][0] * yavr[0] + x[1][0] * yavr[1] + x[2][0] * yavr[2]) /
3;
        double a22 = (x[0][1] * yavr[0] + x[1][1] * yavr[1] + x[2][1] * yavr[2]) /
3;

        double k1[][] = {{my, mx1, mx2}, {a11, a1, a2}, {a22, a2, a3}};
        double k2[][] = {{1, my, mx2}, {mx1, a11, a2}, {mx2, a22, a3}};
        double k3[][] = {{1, mx1, my}, {mx1, a1, a11}, {mx2, a2, a22}};
        double z[][] = {{1, mx1, mx2}, {mx1, a1, a2}, {mx2, a2, a3}};

        double b0 = determinant(k1) / determinant(z);
        double b1 = determinant(k2) / determinant(z);
        double b2 = determinant(k3) / determinant(z);

        double yp1 = b0 + b1 * x[0][0] + b2 * x[0][1];
        double yp2 = b0 + b1 * x[1][0] + b2 * x[1][1];
        double yp3 = b0 + b1 * x[2][0] + b2 * x[2][1];

        double dx1 = Math.abs(x1max - x1min) / 2;
        double dx2 = Math.abs(x2max - x2min) / 2;

        double x10 = Math.abs(x1max + x1min) / 2;
        double x20 = Math.abs(x2max + x2min) / 2;

        double a0n = b0 - (b1 * x10 / dx1) - (b2 * x20 / dx2);
        double a1n = b1 / dx1;
        double a2n = b2 / dx2;

        double eq1 = a0n + a1n * x1max + a2n * x2max;
        double eq2 = a0n + a1n * x1min + a2n * x2max;
        double eq3 = a0n + a1n * x1max + a2n * x2min;
        System.out.println("Нормована матриця планування:");
        for (int i = 0; i < x.length; i++) {
            for (int j = 0; j < x[i].length; j++) {
                System.out.print(x[i][j] + ", ");
            }
            System.out.println("");
        }
        System.out.print("Середні значення: ");
        for (int i = 0; i < yavr.length; i++) {
            System.out.print(yavr[i] + " ");
        }
        System.out.println("\n" + "З нормованими коефіцієнтами: " + yp1 + " " + yp2
+ " " + yp3);
        System.out.println("З натуралізованими: " + eq1 + " " + eq2 + " " + eq3);
    }
}

```

## Відповіді на контрольні запитання

- 1) Що таке регресійні поліноми і де вони застосовуються?

Регресійні поліноми – це апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. Вони використовуються в ТПЕ для оцінки результатів вимірів.

- 2) Визначення однорідності дисперсії.

Однорідність дисперсії означає, що серед усіх дисперсій нема таких, які б значно перевищували одна одну.

- 3) Що називається повним факторним експериментом?

Якщо в багатофакторному експерименті використані всі можливі комбінації рівнів факторів, то такий експеримент називається повним факторним експериментом.

## Результати роботи програми

Нормована матриця планування:

1.0, 1.0,

-1.0, 1.0,

1.0, -1.0,

Середні значення: -1809.0 -1815.8 -1798.2

3 нормованими коефіцієнтами: -1809.0000000000002 -1815.8 -1798.2

3 натуралізованими: -1809.0 -1815.8 -1798.1999999999998