

Національний технічний університет України  
«Київський політехнічний інститут ім. І. Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

## **Методи оптимізації та планування експерименту**

### **Лабораторна робота №3**

«Проведення трьохфакторного експерименту з використанням лінійного рівняння  
регресії»

Виконав:

студент групи ІО-93

Варченко Є. В.

Номер у списку групи – 3

Перевірив:

ас. Регіда П. Г.

Київ – 2021

**Тема:** «Проведення трьохфакторного експерименту з використанням лінійного рівняння регресії».

**Мета:** провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

### Завдання

1. Скласти матрицю планування для дробового трьохфакторного експерименту. Провести експеримент в усіх точках факторного простору, повторивши N експериментів, де N – кількість експериментів (рядків матриці планування) в усіх точках факторного простору – знайти значення функції відгуку Y. Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі (випадковим чином).

$$y_{\max} = 200 + x_{\text{ср max}};$$

$$y_{\min} = 200 + x_{\text{ср min}}$$

$$\text{де } x_{\text{ср max}} = \frac{x_{1\max} + x_{2\max} + x_{3\max}}{3}, \quad x_{\text{ср min}} = \frac{x_{1\min} + x_{2\min} + x_{3\min}}{3}$$

2. Знайти коефіцієнти лінійного рівняння регресії. Записати лінійне рівняння регресії.
3. Провести 3 статистичні перевірки.
4. Написати комп'ютерну програму, яка усе це виконує.

### Варіант

№ варіанта	X <sub>1</sub>		X <sub>2</sub>		X <sub>3</sub>	
	min	max	min	max	min	max
303	-20	30	-20	40	-20	-10

## Код програми

```
import java.util.*;

class Experiment {
    private final int X1min, X1max, X2min, X2max, X3min, X3max;

    public Experiment(int x1min, int x1max, int x2min, int x2max, int x3min, int x3max) {
        X1min = x1min;
        X1max = x1max;
        X2min = x2min;
        X2max = x2max;
        X3min = x3min;
        X3max = x3max;
    }

    public void calculate() {
        double Ymin = (X1min + X2min + X3min) / 3.0 + 200;
        double Ymax = (X1max + X2max + X3max) / 3.0 + 200;

        int[][] normalizePlan = new int[4][8];

        normalizePlan[0][0] = X1min;
        normalizePlan[0][1] = X2min;
        normalizePlan[0][2] = X3min;
        normalizePlan[1][0] = X1min;
        normalizePlan[1][1] = X2max;
        normalizePlan[1][2] = X3max;
        normalizePlan[2][0] = X1max;
        normalizePlan[2][1] = X2min;
        normalizePlan[2][2] = X3max;
        normalizePlan[3][0] = X1max;
        normalizePlan[3][1] = X2max;
        normalizePlan[3][2] = X3min;
    }
}
```

```

Random random = new Random();

for (int i = 0; i < 4; i++) {
    for (int j = 3; j < 8; j++)
        normalizePlan[i][j] = (int)(Ymin + (Ymax - Ymin) * random.nextDouble());
}

System.out.println("Натуралізований план:");
System.out.println(" №   X1max X2max X3max   Y1     Y2     Y3     Y4     Y5");

for (int i = 0; i < 4; i++) {
    System.out.print("   " + (i + 1) + "   ");

    for (int j = 0; j < 8; j++)
        System.out.print(normalizePlan[i][j] + "   ");

    System.out.println();
}

double[] averageY = new double[4];
double mY = 0;

for (int i = 0; i < 4; i++) {
    for (int j = 3; j < 8; j++) {
        averageY[i] += normalizePlan[i][j];
    }

    averageY[i] /= 5.0;
    mY += averageY[i];
}

mY /= 4.0;

double mX1 = (normalizePlan[0][0] + normalizePlan[1][0] + normalizePlan[2][0] +

```

```

normalizePlan[3][0]) / 4.0;
    double mX2 = (normalizePlan[0][1] + normalizePlan[1][1] + normalizePlan[2][1] +
normalizePlan[3][1]) / 4.0;
    double mX3 = (normalizePlan[0][2] + normalizePlan[1][2] + normalizePlan[2][2] +
normalizePlan[3][2]) / 4.0;

    double a1 = (normalizePlan[0][0] * averageY[0] + normalizePlan[1][0] * averageY[1]
+ normalizePlan[2][0] * averageY[2] + normalizePlan[3][0] * averageY[3]) / 4.0;
    double a2 = (normalizePlan[0][1] * averageY[0] + normalizePlan[1][1] * averageY[1]
+ normalizePlan[2][1] * averageY[2] + normalizePlan[3][1] * averageY[3]) / 4.0;
    double a3 = (normalizePlan[0][2] * averageY[0] + normalizePlan[1][2] * averageY[1]
+ normalizePlan[2][2] * averageY[2] + normalizePlan[3][2] * averageY[3]) / 4.0;

    double a11 = (normalizePlan[0][0] * normalizePlan[0][0] + normalizePlan[1][0] *
normalizePlan[1][0]) / 4.0;
    a11 += (normalizePlan[2][0] * normalizePlan[2][0] + normalizePlan[3][0] *
normalizePlan[3][0]) / 4.0;
    double a22 = (normalizePlan[0][1] * normalizePlan[0][1] + normalizePlan[1][1] *
normalizePlan[1][1]) / 4.0;
    a22 += (normalizePlan[2][1] * normalizePlan[2][1] + normalizePlan[3][1] *
normalizePlan[3][1]) / 4.0;
    double a33 = (normalizePlan[0][2] * normalizePlan[0][2] + normalizePlan[1][2] *
normalizePlan[1][2]) / 4.0;
    a33 += (normalizePlan[2][2] * normalizePlan[2][2] + normalizePlan[3][2] *
normalizePlan[3][2]) / 4.0;
    double a12 = (normalizePlan[0][0] * normalizePlan[0][1] + normalizePlan[1][0] *
normalizePlan[1][1]) / 4.0;
    a12 += (normalizePlan[2][0] * normalizePlan[2][1] + normalizePlan[3][0] *
normalizePlan[3][1]) / 4.0;
    double a13 = (normalizePlan[0][0] * normalizePlan[0][2] + normalizePlan[1][0] *
normalizePlan[1][2]) / 4.0;
    a13 += (normalizePlan[2][0] * normalizePlan[2][2] + normalizePlan[3][0] *
normalizePlan[3][2]) / 4.0;
    double a23 = (normalizePlan[0][1] * normalizePlan[0][2] + normalizePlan[1][1] *
normalizePlan[1][2]) / 4.0;
    a23 += (normalizePlan[2][1] * normalizePlan[2][2] + normalizePlan[3][1] *
normalizePlan[3][2]) / 4.0;

```

```

double det = determinant4
    (1, mX1, mX2, mX3,
        mX1, a11, a12, a13,
        mX2, a12, a22, a23,
        mX3, a13, a23, a33);

```

```

double det1 = determinant4
    (mY, mX1, mX2, mX3,
        a1, a11, a12, a13,
        a2, a12, a22, a23,
        a3, a13, a23, a33);

```

```

double det2 = determinant4
    (1, mY, mX2, mX3,
        mX1, a1, a12, a13,
        mX2, a2, a22, a23,
        mX3, a3, a23, a33);

```

```

double det3 = determinant4
    (1, mX1, mY, mX3,
        mX1, a11, a1, a13,
        mX2, a12, a2, a23,
        mX3, a13, a3, a33);

```

```

double det4 = determinant4
    (1, mX1, mX2, mY,
        mX1, a11, a12, a1,
        mX2, a12, a22, a2,
        mX3, a13, a23, a3);

```

```

double b0 = det1 / det;
double b0_round = Math.round(b0 * 100) / 100.0;

```

```

double b1 = det2 / det;
double b1_round = Math.round(b1 * 100) / 100.0;

```

```

double b2 = det3 / det;
double b2_round = Math.round(b2 * 100) / 100.0;

double b3 = det4 / det;
double b3_round = Math.round(b3 * 100) / 100.0;

System.out.println();
System.out.println("Рівняння регресії:");
System.out.println("Y = " + b0_round + " + " + b1_round + "*X1max + " + b2_round +
"*X2max + " + b3_round + "*X3max\n");
System.out.println("Нові значення:");
System.out.println("Y1 = " + Math.round(100 * (b0 + b1 * X1min + b2 * X2min + b3 *
X3min)) / 100.0);
System.out.println("Y2 = " + Math.round(100 * (b0 + b1 * X1min + b2 * X2max + b3 *
X3max)) / 100.0);
System.out.println("Y3 = " + Math.round(100 * (b0 + b1 * X1max + b2 * X2min + b3 *
X3max)) / 100.0);
System.out.println("Y4 = " + Math.round(100 * (b0 + b1 * X1max + b2 * X2max + b3 *
X3min)) / 100.0);

int[][] plan = normalizePlan;
plan[0][0] = plan[1][0] = plan[2][1] = plan[3][2] = plan[0][1] = plan[0][2] = -1;
plan[1][1] = plan[1][2] = plan[2][0] = plan[2][2] = plan[3][0] = plan[3][1] = 1;

System.out.println();
System.out.println("Нормалізований план:");
System.out.println(" № X1max X2max X3max Y1 Y2 Y3 Y4 Y5");

for (int i = 0; i < 4; i++) {
    System.out.print(" " + (i + 1) + " ");

    for (int j = 0; j < 8; j++)
        System.out.print(normalizePlan[i][j] + " ");
}

```

```

        System.out.println();
    }

    double[] Sigma = new double[4];
    double[] Sigma_round = Sigma;

    for (int i = 0; i < 4; i++) {
        for (int j = 3; j < 8; j++)
            Sigma[i] += (averageY[i] - plan[i][j]) * (averageY[i] - plan[i][j]);

        Sigma[i] /= 5.0;
        Sigma_round[i] = Math.round(Sigma[i] * 100) / 100.0;
    }

    System.out.println("\nСереднє Yi:");

    for (int i = 0; i < 4; i++)
        System.out.printf("Y%d = %f\n", i + 1, averageY[i]);

    System.out.print("\nДисперсії: ");

    for (int i = 0; i < 4; i++)
        System.out.printf("D(Y%d) = %f", i + 1, Sigma_round[i]);

    Sigma_round = Sigma;

    Arrays.sort(Sigma_round);

    double Gp = Sigma_round[3] / (Sigma[0] + Sigma[1] + Sigma[2] + Sigma[3]);

    System.out.println("\nКритерій Кохрена: " + Math.round(Gp * 10000) / 10000.0);

    if (Gp <= 0.6287)
        System.out.println("\nДисперсія однорідна з вірогідністю 95%.");

```



**else**

System.out.println("\nДисперсія неоднорідна");

**double** Sb = (Sigma[0] + Sigma[1] + Sigma[2] + Sigma[3]) / 4.0;

**double** Sbs = Math.sqrt(Sb / 20.0);

**double** beta0 = (averageY[0] + averageY[1] + averageY[2] + averageY[3]) / 4.0;

**double** beta1 = (averageY[0] \* plan[0][0] + averageY[1] \* plan[1][0] + averageY[2] \* plan[2][0] + averageY[3] \* plan[3][0]) / 4.0;

**double** beta2 = (averageY[0] \* plan[0][1] + averageY[1] \* plan[1][1] + averageY[2] \* plan[2][1] + averageY[3] \* plan[3][1]) / 4.0;

**double** beta3 = (averageY[0] \* plan[0][2] + averageY[1] \* plan[1][2] + averageY[2] \* plan[2][2] + averageY[3] \* plan[3][2]) / 4.0;

**double**[] t = new **double**[4];

**double**[] t\_round = t;

t[0] = Math.abs(beta0) / Sbs;

t\_round[0] = Math.round(t[0] \* 100000) / 100000.0;

t[1] = Math.abs(beta1) / Sbs;

t\_round[1] = Math.round(t[1] \* 100000) / 100000.0;

t[2] = Math.abs(beta2) / Sbs;

t\_round[2] = Math.round(t[2] \* 100000) / 100000.0;

t[3] = Math.abs(beta3) / Sbs;

t\_round[3] = Math.round(t[3] \* 100000) / 100000.0;

System.out.println();

System.out.println("Критерій Стьюдента:");

**for** (**int** i = 0; i < 4; i++)

System.out.printf("t%d = %f", i + 1, t\_round[i]);

**double**[] b = {b0, b1, b2, b3};

**for** (**int** i = 0; i < 4; i++)

```

{
    if (t[i] < 2.12)
        b[i] = 0;
}

System.out.println();
System.out.println("\nPівняння регресії:");
System.out.print("Y = ");

int f4 = 4;

if (b[0] != 0) {
    System.out.print(b0_round);
    f4--;
}

if (b[1] != 0) {
    System.out.print(" + " + b1_round + "*X1max");
    f4--;
}

if (b[2] != 0) {
    System.out.print(" + " + b2_round + "*X2max");
    f4--;
}

if (b[3] != 0) {
    System.out.print(" + " + b3_round + "*X3max");
    f4--;
}

System.out.println();
System.out.println("\nНові значення:");

double[] Yj = new double[4];

```

```

Yj[0] = b[0] + b[1] * X1min + b[2] * X2min + b[3] * X3min;
Yj[1] = b[0] + b[1] * X1min + b[2] * X2max + b[3] * X3max;
Yj[2] = b[0] + b[1] * X1max + b[2] * X2min + b[3] * X3max;
Yj[3] = b[0] + b[1] * X1max + b[2] * X2max + b[3] * X3min;

for (int i = 0; i < 4; i++)
    System.out.printf("Y%d = %f\n", (i + 1), Math.round(100 * (Yj[i])) / 100.0);

double[] fisher = { 4.5, 3.6, 3.2, 3.0 };

double Sad = 0;

for (int i = 0; i < 4; i++)
    Sad += (averageY[i] - Yj[i]) * (averageY[i] - Yj[i]);

Sad *= 5.0 / f4;

double F = Sad / Sb;

System.out.println("\nКритерій Фішера: " + Math.round(F * 100) / 100.0);

if (F < fisher[f4])
    System.out.println("\nРівняння регресії адекватне при рівні значимості 5%.");
else
    System.out.println("\nРівняння регресії неадекватне при рівні значимості 5%");
}

public double determinant3(
    double a11, double a12, double a13,
    double a21, double a22, double a23,
    double a31, double a32, double a33) {
return a11 * a22 * a33 - a13 * a22 * a31 +
    a12 * a23 * a31 - a12 * a21 * a33 +
    a13 * a21 * a32 - a11 * a23 * a32;
}

```

```
}
```

```
public double determinant4(  
    double a11, double a12, double a13, double a14,  
    double a21, double a22, double a23, double a24,  
    double a31, double a32, double a33, double a34,  
    double a41, double a42, double a43, double a44) {  
    return a11 * determinant3(a22, a23, a24, a32, a33, a34, a42, a43, a44) -  
        a12 * determinant3(a21, a23, a24, a31, a33, a34, a41, a43, a44) -  
        a13 * determinant3(a22, a21, a24, a32, a31, a34, a42, a41, a44) -  
        a14 * determinant3(a22, a23, a21, a32, a33, a31, a42, a43, a41);  
}
```

```
}
```

```
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Experiment experiment = new Experiment(-20, 30, -20, 40, -20, -10);  
        experiment.calculate();  
    }  
}
```

```
}
```

## Результати роботи програми

Натуралізований план:

№	X1max	X2max	X3max	Y1	Y2	Y3	Y4	Y5
1	-20	-20	-20	185	211	180	204	206
2	-20	40	-10	215	210	195	211	206
3	30	-20	-10	202	201	188	192	180
4	30	40	-20	197	213	192	213	218

Рівняння регресії:

$$Y = 196.35 + -0.05 \cdot X1_{\max} + 0.2 \cdot X2_{\max} + -0.19 \cdot X3_{\max}$$

Нові значення:

$$Y1 = 197.2$$

$$Y2 = 207.4$$

$$Y3 = 192.6$$

$$Y4 = 206.6$$

Нормалізований план:

№	X1max	X2max	X3max	Y1	Y2	Y3	Y4	Y5
1	-1	-1	-1	185	211	180	204	206
2	-1	1	1	215	210	195	211	206
3	1	-1	1	202	201	188	192	180
4	1	1	-1	197	213	192	213	218

Середнє  $Y_i$ :

$$Y1 = 197.200000$$

$$Y2 = 207.400000$$

$$Y3 = 192.600000$$

$$Y4 = 206.600000$$

$$\text{Дисперсії: } D(Y1) = 151.760000 D(Y2) = 46.640000 D(Y3) = 67.840000 D(Y4) = 103.440000$$

Критерій Кохрена: 0.4105

Дисперсія однорідна з вірогідністю 95%.

```
Критерій Стюдента:  
t1 = 93.480280t2 = 0.628010t3 = 2.814410t4 = 0.441930  
  
Рівняння регресії:  
Y = 196.35 + 0.2*X2max  
  
Нові значення:  
Y1 = 192.320000  
Y2 = 204.420000  
Y3 = 192.320000  
Y4 = 204.420000  
  
Критерій Фішера: 1.02  
  
Рівняння регресії адекватне при рівні значимості 5%.  
  
Process finished with exit code 0
```

### Висновки

- Ознайомилися з темою роботи.
- Були здобуті необхідні навички для виконання завдань.
- Розроблено програму, яка виконує поставлену задачу.
- Вище приведені результати свідчать про успішне виконання умов завдань.
- Основну мету лабораторної роботи було досягнуто.