

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 № 4
з дисципліни «Методи планування експерименту»
на тему «**Проведення трьохфакторного експерименту при використанні
рівняння регресії з урахуванням ефекту взаємодії.**»

ВИКОНАВ: студент 2 курсу
групи ІВ-92
Подкур А. О.
Залікова – 9217
ПЕРЕВІРИВ: ас. Регіда П.Г.

Київ – 2021

Хід роботи

Мета:

Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

Завдання:

1. Скласти матрицю планування для повного трьохфакторного експерименту.
2. Провести експеримент, повторивши N раз досліди у всіх точках факторного простору і знайти значення відгуку Y. Знайти значення Y шляхом моделювання випадкових чисел у певному діапазоні відповідно варіанту. Варіанти вибираються за номером в списку в журналі викладача.

№ _{варіанта}	x_1		x_2		x_3	
	min	max	min	Max	min	max
217	-10	50	25	65	-10	15

$$y_{i\max} = 200 + x_{cp\max}$$

$$y_{i\min} = 200 + x_{cp\min}$$

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

3. Знайти коефіцієнти рівняння регресії і записати його.
4. Провести 3 статистичні перевірки – за критеріями Кохрена, Стьюдента, Фішера.
5. Зробити висновки по адекватності регресії та значимості окремих коефіцієнтів і записати скореговане рівняння регресії.
6. Написати комп'ютерну програму, яка усе це моделює.

Лістинг програми

```
using System;
using static System.Console;
using ConsoleTables;
using System.Linq;

namespace Lab4
{
    class Program
    {
        private static int[,] genRandomMatrix(int y_min, int y_max, int m, int n)
        {
            Random r = new Random();
            int[,] matrix = new int[n,m];
            for(int i=0; i < n; i++)
            {
                for(int j = 0; j < m; j++)
                {
                    matrix[i,j] = r.Next(y_min,y_max);
                }
            }

            return matrix;
        }

        static void Main(string[] args)
        {
            int m = 3;
            int n = 8;
            int x1_min = -10;
            int x1_max = -50;
            int x2_min = 25;
            int x2_max = 65;
            int x3_min = -10;
            int x3_max = 15;
            int y_min = 200 + (x1_min + x2_min + x3_min) / 3;
            int y_max = 200 + (x1_max + x2_max + x3_max) / 3;

            int[,] matrixY = genRandomMatrix(y_min, y_max, m,n);

            double[] averageY = new double[matrixY.GetLength(0)];
            int counter;
            for(int i = 0; i < matrixY.GetLength(0); i++)
            {
                counter = 0;
                for (int j = 0; j < matrixY.GetLength(1); j++)
                {
```

```

        counter += matrixY[i,j];
    }
    averageY[i] = Math.Round((double) counter / matrixY.GetLength(1),
3);
}

for(int i = 0; i< averageY.Length; i++){
    // Write($"{averageY[i]} ");
}

int[,] xN = {{-1, -1, -1},
              {-1, -1, 1},
              {-1, 1, -1},
              {-1, 1, 1},
              {1, -1, -1},
              {1, -1, 1},
              {1, 1, -1},
              {1, 1, 1}};

double b0;

double dcounter = 0;
for(int i = 0; i < averageY.Length; i++)
{
    dcounter += averageY[i];
}
b0 = dcounter / n;
dcounter = 0;

double b1;

for(int i = 0; i < averageY.Length; i++)
{
    dcounter += averageY[i] * xN[i,0];
}
b1 = dcounter / n;
dcounter = 0;

double b2;

for(int i = 0; i < averageY.Length; i++)
{
    dcounter += averageY[i] * xN[i,1];
}
b2 = dcounter / n;
dcounter = 0;

double b3;

for(int i = 0; i < averageY.Length; i++)
{

```

```

        dcounter += averageY[i] * xN[i,2];
    }
    b3 = dcounter / n;
    dcounter = 0;

    double b12;

    for(int i = 0; i < averageY.Length; i++)
    {
        dcounter += averageY[i] * xN[i,0] * xN[i,1];
    }
    b12 = dcounter / n;
    dcounter = 0;

    double b13;

    for(int i = 0; i < averageY.Length; i++)
    {
        dcounter += averageY[i] * xN[i,0] * xN[i,2];
    }
    b13 = dcounter / n;
    dcounter = 0;

    double b23;

    for(int i = 0; i < averageY.Length; i++)
    {
        dcounter += averageY[i] * xN[i,1] * xN[i,2];
    }
    b23 = dcounter / n;
    dcounter = 0;

    double b123;

    for(int i = 0; i < averageY.Length; i++)
    {
        dcounter += averageY[i] * xN[i,0] * xN[i,1] * xN[i,2];
    }
    b123 = dcounter / n;
    dcounter = 0;

    int[,] planMatrix = {{x1_min, x2_min, x3_min, x1_min * x2_min, x1_min
* x3_min, x2_min * x3_min, x1_min * x2_min * x3_min},
                        {x1_min, x2_min, x3_max, x1_min * x2_min, x1_min
* x3_max, x2_min * x3_max, x1_min * x2_min * x3_max},
                        {x1_min, x2_max, x3_min, x1_min * x2_max, x1_min
* x3_min, x2_max * x3_min, x1_min * x2_max * x3_min},
                        {x1_min, x2_max, x3_max, x1_min * x2_max, x1_min
* x3_max, x2_max * x3_max, x1_min * x2_max * x3_max},
                        {x1_max, x2_min, x3_min, x1_max * x2_min, x1_max
* x3_min, x2_min * x3_min, x1_max * x2_min * x3_min},

```

```

        {x1_max, x2_min, x3_max, x1_max * x2_min, x1_max
* x3_max, x2_min * x3_max, x1_max * x2_min * x3_max},
        {x1_max, x2_max, x3_min, x1_max * x2_max, x1_max
* x3_min, x2_max * x3_min, x1_max * x2_max * x3_min},
        {x1_max, x2_max, x3_max, x1_max * x2_max, x1_max
* x3_max, x2_max * x3_max, x1_max * x2_max * x3_max}};

        double[] resultY = new double[n];
        for( int i = 0; i< n; i++){
            resultY[i] = b0 + b1 * planMatrix[i,0] + b2 * planMatrix[i,1] + b
3 * planMatrix[i,2] +
                b12 * planMatrix[i,3] + b13 * planMatrix[i,4] + b23 * planMat
rix[i,5] +
                b123 * planMatrix[i,6];
        }

        double[] dispersion = new double[n];

        for(int i = 0; i < n; i++)
        {
            dcounter = 0;
            for(int j = 0; j < m; j++)
            {
                dcounter += (matrixY[i,j] - averageY[i])*(matrixY[i,j] - aver
ageY[i]);
            }
            dispersion[i] = Math.Round(dcounter/m,3);
        }

        ConsoleTable table;

        table = new ConsoleTable("", "", "", "", "", "", "Matrix of planning"
, "", "", "", "", "", "");
        table.AddRow("X0", "X1", "X2", "X3", "X12", "X13", "X23", "X123", "Y1
", "Y2", "Y3", "Y average", "S^2");
        int[] x0 = new int[n];
        for(int i = 0; i < n; i++)
            x0[i] = 1;

        for(int i = 0; i < n; i++)
        {
            table.AddRow(x0[i],planMatrix[i,0], planMatrix[i,1],planMatrix[i,
2],planMatrix[i,3],planMatrix[i,4],planMatrix[i,5],planMatrix[i,6],matrixY[i,0],
matrixY[i,1], matrixY[i,2], averageY[i],dispersion[i]);
        }

        WriteLine("Matrix of planning");
        table.Write();
        WriteLine();

```

```

// Cochran`s test
WriteLine("+++++++ Cochran`s test ++++++\n");
double gp = dispersion.Max() / dispersion.Sum();
double gt = 0.5157;
if (gp < gt) WriteLine($"According to the Cochran test, the dispersion is homogeneous - {Math.Round(gp,3)} < {Math.Round(gt,3)}\n");
else WriteLine($"According to the Cochran test, the dispersion is not homogeneous - {Math.Round(gp,3)} > {Math.Round(gt,3)}\n");
WriteLine();

WriteLine("+++++++ Student`s test ++++++\n");
int d = 8;
double sb = dispersion.Sum() / n;
double sBeta = Math.Sqrt(sb / (n * m));
double[] bb = {b0, b1, b2, b3, b12, b13, b23, b123};
double[] tList = new double[n];
for (int i = 0; i < n; i++)
    tList[i] = Math.Abs(bb[i]) / sBeta;
double tt = 2.120;
double[] bList = {b0, b1, b2, b3, b12, b13, b23, b123};
for (int i = 0; i < n; i++)
{
    if (tList[i] < tt)
    {
        bList[i] = 0;
        d--;
    }
}
for (int i = 0; i < tList.Length; i++)
{
    WriteLine($"\\tt{i} = {Math.Round(tList[i],3)}");
}
WriteLine();

// Fisher`s test
WriteLine("+++++++ Fisher`s test ++++++\n");
double[] yReg = new double[n];
for (int i = 0; i < n; i++)
    yReg[i] = b0 + b1 * planMatrix[i,0] + b2 * planMatrix[i,1] + b3 *
planMatrix[i,2] +
    b12 * planMatrix[i,3] + b13 * planMatrix[i,4] + b23 * planMatrix[i,5]
+
    b123 * planMatrix[i,6];

dcounter = 0;
for (int i = 0; i < n; i++)
{
    dcounter += (yReg[i] - averageY[i]) * (yReg[i] - averageY[i]);
}
double sad = (m / (n - d)) * (int) dcounter;

```

```

        double fp = sad/sb;

        if(fp < 4.5) WriteLine("The regression equation is adequate at 0.05")
;
        else WriteLine("The regression equation is adequate at 0.05");
        WriteLine();

        Write("Equation: ");
        WriteLine($"y = {Math.Round(b0,3)} + {Math.Round(b1,3)} * x1 + {Math.
Round(b2,3)} * x2 + {Math.Round(b3,3)} * x3 + {Math.Round(b12,3)} * x1x2 + {Math.
Round(b13,3)} * x1x3 + {Math.Round(b23,3)} * x2x3 + {Math.Round(b123,3)} * x1x2x3
");
        WriteLine();

        for(int i = 0; i< resultY.Length; i++)
            WriteLine($"\\ty{i+1} = {Math.Round(resultY[i], 3)}");
    }
}
}

```


Результати роботи програми

							Matrix of planning							
X0	X1	X2	X3	X12	X13	X23		X123	Y1	Y2	Y3	Y average	S^2	
1	-10	25	-10	-250	100	-250		2500	204	203	201	202.667	1.556	
1	-10	25	15	-250	-150	375		-3750	203	202	205	203.333	1.556	
1	-10	65	-10	-650	100	-650		6500	204	205	208	205.667	2.889	
1	-10	65	15	-650	-150	975		-9750	207	208	202	205.667	6.889	
1	-50	25	-10	-1250	500	-250		12500	206	209	203	206	6	
1	-50	25	15	-1250	-750	375		-18750	201	208	206	205	8.667	
1	-50	65	-10	-3250	500	-650		32500	208	207	202	205.667	6.889	
1	-50	65	15	-3250	-750	975		-48750	201	208	208	205.667	10.889	
Count: 9														
+++++++ Cochran`s test +++++++														
According to the Cochran test, the dispersion is homogeneous - 0.24 < 0.516														
+++++++ Student`s test +++++++														
t0 = 421.793														
t1 = 1.286														
t2 = 1.458														
t3 = 0.086														
t4 = 1.286														
t5 = 0.429														
t6 = 0.086														
t7 = 0.429														
+++++++ Fisher`s test +++++++														
The regression equation is adequate at 0.05														
Equation: y = 204.958 + 0.625 * x1 + 0.709 * x2 + -0.042 * x3 + -0.625 * x1x2 + -0.208 * x1x3 + 0.042 * x2x3 + 0.208 * x1x2x3														
y1 = 862.451														
y2 = -361.999														
y3 = 1957.091														
y4 = -1308.109														
y5 = 3461.651														
y6 = -2760.799														
y7 = 8888.291														
y8 = -7704.909														

Висновок: під час виконання лабораторної роботи було проведено повний трьохфакторний експеримент. Знайдено рівняння регресії адекватне об'єкту. Лабораторна робота виконана, кінцева мета досягнута.