

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

ВИКОНАВ:
студент II курсу ФІОТ
групи ІВ-92
Подкур Антон Олександрович
Варіант: 217
ПЕРЕВІРИВ:
Регіда П. Г.

Хід роботи

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

Завдання:

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 y діапазоні $y_{\min} \div y_{\max}$

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

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

№ _{варіанта}	X ₁		X ₂	
	min	max	min	max
217	20	70	5	40

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

$$Y_{\max} = (30 - 17) * 10 = 130$$

$$Y_{\min} = (20 - 17) * 10 = 30$$

Лістинг програми

Experiment.cs

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

namespace Lab2
{
    class Experiment
    {
        ConsoleTable table;
        private static double R_critical = 2.10; //number = 6, p = 0.95

        private static double X11 = -1.0;
        private static double X12 = -1.0;
        private static double X13 = +1.0;
        private static double X21 = -1.0;
        private static double X22 = +1.0;
        private static double X23 = -1.0;

        private int MinX1;
        private int MaxX1;
        private int MinX2;
        private int MaxX2;
        private int MinY;
        private int MaxY;

        private double[] averageY;

        private double b0;
        private double b1;
        private double b2;

        private int numberOfY;
        private double[,] Y;

        public Experiment(int MinX1, int MaxX1, int MinX2, int MaxX2, int MinY, int
MaxY, int numberOfY)
        {
            this.MinX1 = MinX1;
            this.MaxX1 = MaxX1;
            this.MinX2 = MinX2;
            this.MaxX2 = MaxX2;
            this.MinY = MinY;
            this.MaxY = MaxY;
            this.numberOfY = numberOfY;
            randomY();
            romanovskiY();
        }
    }
}
```

```

        normalize();
        naturalize();
    }

    private void randomY()
    {
        Random rnd = new Random();
        Y = new double[3, numberOfY];
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < numberOfY; j++)
            {
                Y[i, j] = rnd.NextDouble() * (MaxY - MinY) + MinY;
            }
        }
    }

    private void romanovskiy()
    {
        averageY = new double[3];
        for (int i = 0; i < numberOfY; i++)
        {
            averageY[0] += Y[0, i];
            averageY[1] += Y[1, i];
            averageY[2] += Y[2, i];
        }
        averageY[0] /= numberOfY;
        averageY[1] /= numberOfY;
        averageY[2] /= numberOfY;

        table = new ConsoleTable(" ", "AVERAGE Ys", " ");
        table.AddRow("Average y1", "Average y2", "Average y3");
        table.AddRow(averageY[0], averageY[1], averageY[2]);
        table.Configure(o => o.NumberAlignment = Alignment.Right).Write(Format.
Alternative);

        WriteLine();

        double sigmaY1 = 0;
        double sigmaY2 = 0;
        double sigmaY3 = 0;
        for (int i = 0; i < numberOfY; i++)
        {
            sigmaY1 += Math.Pow(Y[0, i] - averageY[0], 2);
            sigmaY2 += Math.Pow(Y[1, i] - averageY[1], 2);
            sigmaY3 += Math.Pow(Y[2, i] - averageY[2], 2);
        }
        sigmaY1 /= numberOfY;
        sigmaY2 /= numberOfY;
        sigmaY3 /= numberOfY;
    }

```

```

        table = new ConsoleTable(" ", "SIGMAS OF Ys", " ");
        table.AddRow("Sigma y1", "Sigma y2", "Sigma y3");
        table.AddRow(sigmaY1, sigmaY2, sigmaY3);
        table.Configure(o => o.NumberAlignment = Alignment.Right).Write(Format.
Alternative);
        WriteLine();

        double teta0 = Math.Pow(2 * (2 * numberOfY - 2) / numberOfY / (numberOf
Y - 4), 1 / 2);
        WriteLine("Main deviation teta0 = " + teta0);
        WriteLine();

        double Fuv1 = Math.Max(sigmaY1, sigmaY2) / Math.Min(sigmaY1, sigmaY2);
        double Fuv2 = Math.Max(sigmaY1, sigmaY3) / Math.Min(sigmaY1, sigmaY3);
        double Fuv3 = Math.Max(sigmaY3, sigmaY2) / Math.Min(sigmaY3, sigmaY2);

        table = new ConsoleTable(" ", "FUVs", " ");
        table.AddRow("Fuv1", "Fuv2", "Fuv3");
        table.AddRow(Fuv1, Fuv2, Fuv3);
        table.Configure(o => o.NumberAlignment = Alignment.Right).Write(Format.
Alternative);
        WriteLine();

        double tetauv1 = (numberOfY - 2.0) / (double)numberOfY * Fuv1;
        double tetauv2 = (numberOfY - 2.0) / (double)numberOfY * Fuv2;
        double tetauv3 = (numberOfY - 2.0) / (double)numberOfY * Fuv3;

        table = new ConsoleTable(" ", "TETA UVs", " ");
        table.AddRow("Teta uv1", "Teta uv2", "Teta uv3");
        table.AddRow(tetauv1, tetauv2, tetauv3);
        table.Configure(o => o.NumberAlignment = Alignment.Right).Write(Format.
Alternative);
        WriteLine();

        double Ruv1 = Math.Abs(tetauv1 - 1) / teta0;
        double Ruv2 = Math.Abs(tetauv2 - 1) / teta0;
        double Ruv3 = Math.Abs(tetauv3 - 1) / teta0;

        table = new ConsoleTable(" ", "RUVs", " ");
        table.AddRow("Ruv1", "Ruv2", "Ruv3");
        table.AddRow(Ruv1, Ruv2, Ruv3);
        table.Configure(o => o.NumberAlignment = Alignment.Right).Write(Format.
Alternative);
        WriteLine();

        if (Ruv1 < R_critical && Ruv2 < R_critical && Ruv3 < R_critical)
        {
            WriteLine("Ruv < Rkp, dispersion is homogeneous");
        }

```

```

    }
    else
    {
        WriteLine("Ruv > Rkp, dispersion is not homogeneous");
    }
    WriteLine();
}
private void normalize()
{
    double mx1 = (X11 + X12 + X13) / 3;
    double mx2 = (X21 + X22 + X23) / 3;
    double my = (averageY[0] + averageY[1] + averageY[2]) / 3;
    double a1 = (X11 * X11 + X12 * X12 + X13 * X13) / 3;
    double a2 = (X11 * X21 + X12 * X22 + X13 * X23) / 3;
    double a3 = (X21 * X21 + X22 * X22 + X23 * X23) / 3;
    double a11 = (X11 * averageY[0] + X12 * averageY[1] + X13 * averageY[2]
) / 3;
    double a22 = (X21 * averageY[0] + X22 * averageY[1] + X23 * averageY[2]
) / 3;

    double det = (a1 * a3 + mx1 * a2 * mx2 + mx2 * mx1 * a2 - mx2 * mx2 * a
1 - mx1 * mx1 * a3 - a2 * a2);
    b0 = (my * a1 * a3 + mx1 * a2 * a22 + mx2 * a11 * a2 - a22 * a1 * mx2 -
mx1 * a11 * a3 - my * a2 * a2) / det;
    b1 = (a3 * a11 + a22 * mx1 * mx2 + a2 * my * mx2 - mx2 * mx2 * a11 - a2
2 * a2 - mx1 * my * a3) / det;
    b2 = (a1 * a22 + a2 * mx1 * my + mx1 * mx2 * a11 - mx2 * my * a1 - mx1
* mx1 * a22 - a2 * a11) / det;

    table = new ConsoleTable(" ", "Bs", " ");
    table.AddRow("b0", "b1", "b2");
    table.AddRow(b0, b1, b2);
    table.Configure(o => o.NumberAlignment = Alignment.Right).Write(Format.
Alternative);
    WriteLine();

    table = new ConsoleTable("Normalized regression equation: ");
    table.AddRow("y = (" + b0 + ") + (" + b1 + ") * X1 + (" + b2 + ") * X2"
);
    table.Configure(o => o.NumberAlignment = Alignment.Right).Write(Format.
Alternative);
    WriteLine();

    WriteLine("Checking:");
    WriteLine((b0 + X11 * b1 + X21 * b2) + " = " + averageY[0]);
    WriteLine((b0 + X12 * b1 + X22 * b2) + " = " + averageY[1]);
    WriteLine((b0 + X13 * b1 + X23 * b2) + " = " + averageY[2]);
    WriteLine();
}
private void naturalize()
{

```

```

        double deltaX1 = (MaxX1 - MinX1) / 2;
        double deltaX2 = (MaxX2 - MinX2) / 2;
        double X10 = (MaxX1 + MinX1) / 2;
        double X20 = (MaxX2 + MinX2) / 2;
        double a0 = b0 - b1 * X10 / deltaX1 - b2 * X20 / deltaX2;
        double a1 = b1 / deltaX1;
        double a2 = b2 / deltaX2;

        table = new ConsoleTable(" ", "As", " ");
        table.AddRow("a0", "a1", "a2");
        table.AddRow(a0, a1, a2);
        table.Configure(o => o.NumberAlignment = Alignment.Right).Write(Format.
Alternative);
        WriteLine();

        table = new ConsoleTable("Naturalized regression equation: ");
        table.AddRow("y = (" + a0 + ") + (" + a1 + ") * X1 + (" + a2 + ") * X2"
);
        table.Configure(o => o.NumberAlignment = Alignment.Right).Write(Format.
Alternative);
        WriteLine();

        WriteLine("Checking:");
        WriteLine((a0 + MinX1 * a1 + MinX2 * a2) + " = " + averageY[0]);
        WriteLine((a0 + MinX1 * a1 + MaxX2 * a2) + " = " + averageY[1]);
        WriteLine((a0 + MaxX1 * a1 + MinX2 * a2) + " = " + averageY[2]);
    }
}
}

```

Program.cs

```

using System;

namespace Lab2
{
    // Variant #217

    // 20 70 4 40
    // Ymax = (30 - Nvar)*10 = 130
    // Ymin = (20 - Nvar)*10 = 30
    class Program
    {
        static void Main(string[] args)
        {
            int minX1 = 20;

```

```
        int maxX1 = 70;
        int minX2 = 5;
        int maxX2 = 40;
        int minY = 30;
        int maxY = 130;
        int yAmount = 6;

        Experiment experiment = new Experiment(minX1, maxX1, minX2, maxX2, minY
, maxY, yAmount);
    }

}
```


Результат роботи програми

```
PS C:\Users\Anton\Documents\KPI\Semestr 2\MND\Lab2> dotnet run
```

AVERAGE Ys		
Average y1	Average y2	Average y3
77.14140698646263	53.00789399988697	65.8338287499891

SIGMAS OF Ys		
Sigma y1	Sigma y2	Sigma y3
498.6662699914021	638.196388611108	370.58543069010824

Main deviation teta0 = 1

FUVs		
Fuv1	Fuv2	Fuv3
1.2798066101846264	1.3456175788205713	1.7221302721551996

TETA UVs		
Teta uv1	Teta uv2	Teta uv3
0.8532044067897508	0.8970783858803808	1.1480868481034663

RUVs		
Ruv1	Ruv2	Ruv3
0.14679559321024915	0.1029216141196192	0.14808684810346628

Ruv < Rkp, dispersion is homogeneous

Bs		
b0	b1	b2
59.42086137493805	-5.653789118236775	-12.066756493287835

Normalized regression equation:

$$y = (59.42086137493805) + (-5.653789118236775) * X1 + (-12.066756493287835) * X2$$

Checking:

77.14140698646267 = 77.14140698646263
53.007893999886996 = 53.00789399988697
65.83382874998911 = 65.8338287499891

As		
a0	a1	a2
85.21348430848968	-0.22615156472947098	-0.7098092054875197

Naturalized regression equation:

$$y = (85.21348430848968) + (-0.22615156472947098) * X1 + (-0.7098092054875197) * X2$$

Checking:

77.14140698646267 = 77.14140698646263
52.29808479439948 = 53.00789399988697
65.83382874998911 = 65.8338287499891

Відповіді на контрольні запитання

1. Що таке регресійні поліноми і де вони застосовуються?

В теорії планування експерименту найважливішою частиною є оцінка результатів вимірів. При цьому використовують апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. В ТПЕ ці поліноми отримали спеціальну назву - регресійні поліноми, а їх знаходження та аналіз - регресійний аналіз.

2. Визначення однорідності дисперсії.

Однорідність дисперсій – властивість, коли дисперсії вимірювання функцій відгуку є однаковими, або близькими.

3. Що називається повним факторним експериментом?

Для знаходження коефіцієнтів у лінійному рівнянні регресії застосовують повний факторний експеримент (ПФЕ). Якщо в багатофакторному експерименті використані всі можливі комбінації рівнів факторів, то такий експеримент називається повним факторним експериментом.

Висновок: під час виконання лабораторної роботи було провели двофакторний експеримент, перевірено однорідність дисперсій за критерієм Романовського, отримано коефіцієнти рівняння регресії, проведено натуралізацію рівняння регресії.