Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №2

з дисципліни « Методи оптимізації та планування » на тему

«ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕТНУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконала:

студентка II курсу ФІОТ

групи ІВ – 93

Стеценко Богдан

Група ІВ – 93, номер в списку - 24

Перевірив:

ас. Регіда П.Г.

Київ – 2021

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

**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 (хо=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

Варіанти обираються по номеру в списку в журналі викладача.

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



**Роздруківка тексту програми:**

**Script.js**

'use strict';

// Варіант

function main() {

    console.log('Стеценко|Богдан ІВ-93|Варіант - 324');

    // Задання за варіантом XnMin, XnMax

    let x1Min = -30;

    let x1Max = 0;

    let x2Min = 15;

    let x2Max = 50;

    let xN = [

        [-1, -1],

        [1, -1],

        [-1, 1]

    ];

    let varik = 324;

    let m = 6;

    let yMax = (30 - varik) \* 10;

    let yMin = (20 - varik) \* 10;

    console.log(`Максимальне Y: ${yMax}`);

    console.log(`Мінімальне Y: ${yMin}`);

    // Генерація числа в діапазоні

    function randomInteger(min, max) {

        let rand = min - 0.5 + Math.random() \* (max - min + 1);

        return Math.round(rand);

    }

    let y = [

        [],

        [],

        []

    ];

    // Заповнення матриці планування

    for (let index = 0; index < m; index++) {

        for (let i = 0; i < 3; i++) {

            y[i][index] = yMin + randomInteger(0, 100);

        }

    }

    // Матриця планування

    console.log(`Матриця планування при m = ${m}`);

    console.log(y[0]);

    console.log(y[1]);

    console.log(y[2]);

    //Знайдемо середнє значення Y

    let yAv = [];

    function sumY(arr) {

        let sum = 0;

        for (let index = 0; index < arr.length; index++) {

            sum += arr[index];

        }

        return sum / arr.length;

    }

    console.log('------------------Середнє значення функції відгуку------------------');

    yAv[0] = +sumY(y[0]).toFixed(5);

    yAv[1] = +sumY(y[1]).toFixed(5);

    yAv[2] = +sumY(y[2]).toFixed(5);

    for (let index = 0; index < 3; index++) {

        console.log(yAv[index]);

    }

    function despersion(i) {

        let dis = 0;

        for (let index = 0; index < m; index++) {

            dis += (y[i][index] - yAv[i]) \*\* 2;

        }

        return 1 / m \* dis;

    }

    //Знайдемо дисперсію по рядках

    console.log('------------------Знайдемо дисперсію по рядках------------------');

    let d = [];

    for (let index = 0; index < 3; index++) {

        d[index] = +despersion(index).toFixed(5);

    }

    for (let index = 0; index < 3; index++) {

        console.log(`Дисперсія №${index+1} = ${d[index]}`);

    }

    //Обчислимо основне відхилення

    let t = Math.sqrt((2 \* (2 \* m - 2)) / (m \* (m - 4)));

    console.log(`Основне відхилення = ${t}`);

    function fuv(u, v) {

        if (u > v) {

            return u / v;

        } else {

            return v / u;

        }

    }

    // Обчислимо Fuv

    let fUv = [];

    fUv[0] = (fuv(d[0], d[1])).toFixed(5);

    fUv[1] = (fuv(d[2], d[0])).toFixed(5);

    fUv[2] = (fuv(d[2], d[1])).toFixed(5);

    //Виведемо Fuv

    console.log('------------------Обчислимо Fuv------------------');

    for (let index = 0; index < 3; index++) {

        console.log(`Fuv${index+1} = ${fUv[index]}`);

    }

    let oUv = [];

    //Обчислимо θuv

    oUv[0] = (((m - 2) / m) \* fUv[0]).toFixed(5);

    oUv[1] = (((m - 2) / m) \* fUv[1]).toFixed(5);

    oUv[2] = (((m - 2) / m) \* fUv[2]).toFixed(5);

    //Виведемо θuv

    console.log('------------------Обчислимо θuv------------------');

    for (let index = 0; index < 3; index++) {

        console.log(`θuv${index+1} = ${oUv[index]}`);

    }

    let rUv = [];

    //Обчислимо Ruv

    rUv[0] = (Math.abs(oUv[0] - 1) / t).toFixed(5);

    rUv[1] = (Math.abs(oUv[1] - 1) / t).toFixed(5);

    rUv[2] = (Math.abs(oUv[2] - 1) / t).toFixed(5);

    //Виведемо Ruv

    console.log('------------------Обчислимо Ruv------------------');

    for (let index = 0; index < 3; index++) {

        console.log(`Ruv${index+1} = ${rUv[index]}`);

    }

    let rKr = 2.16;

    let p = 0.99;

    console.log(`Оскільки m=6, візьмемо значення Rkr = ${rKr} і довірчу ймовірність p = ${p}`);

    function test(r) {

        let ttt = 0;

        for (let index = 0; index < r.length; index++) {

            if (r[index] > rKr) {

                console.log('Дисперсія неоднорідна');

                m += 1;

                main();

            } else {

                ttt += 1;

                console.log(`${r[index]} > ${rKr}`);

                console.log(`Підтвердження №${index + 1},що дисперсія однорідна`);

            }

        }

        if (ttt == 3) {

            console.log('Розрахунок нормованих коефіцієнтів рівняння регресії.');

            let mx = [];

            mx[0] = (xN[0][0] + xN[1][0] + xN[2][0]) / 3;

            mx[1] = (xN[0][1] + xN[1][1] + xN[2][1]) / 3;

            for (let index = 0; index < 2; index++) {

                console.log(`Mx${index+1} = ${mx[index]}`);

            }

            function vuz(x11, x12, x13, x21, x22, x23, x31, x32, x33) {

                let det = x11 \* x22 \* x33 + x12 \* x23 \* x31 + x32 \* x21 \* x13 -

                    x13 \* x22 \* x31 - x32 \* x23 \* x11 - x12 \* x21 \* x33;

                return det;

            }

            let mY = (yAv[0] + yAv[1] + yAv[2]) / 3;

            console.log(`My = ${mY}`);

            let allA = [];

            allA[0] = (xN[0][0] \*\* 2 + xN[1][0] \*\* 2 + xN[2][0] \*\* 2) / 3;

            allA[1] = (xN[0][0] \* xN[0][1] + xN[1][0] \* xN[1][1] + xN[2][0] \* xN[2][1]) / 3;

            allA[2] = (xN[1][0] \*\* 2 + xN[1][1] \*\* 2 + xN[2][1] \*\* 2) / 3;

            for (let index = 0; index < allA.length; index++) {

                console.log(`A${index+1} = ${allA[index]}`);

            }

            let a11 = (xN[0][0] \* yAv[0] + xN[1][0] \* yAv[1] + xN[2][0] \* yAv[2]) / 3;

            console.log(`A11 = ${a11}`);

            let a22 = (xN[0][1] \* yAv[0] + xN[1][1] \* yAv[1] + xN[2][1] \* yAv[2]) / 3;

            console.log(`A22 = ${a11}`);

            let b0 = vuz(mY, mx[0], mx[1], a11, allA[0], allA[1], a22, allA[1], allA[2]) /

                vuz(1, mx[0], mx[1], mx[0], allA[0], allA[1], mx[1], allA[1], allA[2]);

            let b1 = vuz(1, mY, mx[1], mx[0], a11, allA[1], mx[1], a22, allA[2]) /

                vuz(1, mx[0], mx[1], mx[0], allA[0], allA[1], mx[1], allA[1], allA[2]);

            let b2 = vuz(1, mx[0], mY, mx[0], allA[0], a11, mx[1], allA[1], a22) /

                vuz(1, mx[0], mx[1], mx[0], allA[0], allA[1], mx[1], allA[1], allA[2]);

            console.log('b0 = ' + b0);

            console.log('b1 = ' + b1);

            console.log('b2 = ' + b2);

            console.log('Нормоване рівнянн регресії');

            console.log(`y = ${b0} + ${b1}\*x1 + ${b2}\*x2`);

            console.log('Перевірка');

            console.log(b0 + b1 \* xN[0][0] + b2 \* xN[0][1]);

            console.log(b0 + b1 \* xN[1][0] + b2 \* xN[1][1]);

            console.log(b0 + b1 \* xN[2][0] + b2 \* xN[2][1]);

            console.log('Проведемо натуралізацію коефіцієнтів:');

            let dx1 = Math.abs(x1Max - x1Min) / 2;

            console.log('∆x1 = ' + dx1.toFixed(5));

            let dx2 = Math.abs(x2Max - x2Min) / 2;

            console.log('∆x2 = ' + dx2.toFixed(5));

            let x10 = (x1Max + x1Min) / 2;

            console.log('x10 = ' + x10.toFixed(5));

            let x20 = (x2Max + x2Min) / 2;

            console.log('x20 = ' + x20.toFixed(5));

            let a0 = b0 - b1 \* (x10 / dx1) - b2 \* (x20 / dx2);

            console.log('a0 = ' + a0.toFixed(5));

            let a1 = b1 / dx1;

            console.log('a1 = ' + a1.toFixed(5));

            let a2 = b2 / dx2;

            console.log('a2 = ' + a2.toFixed(5));

            console.log('Запишемо натуралізоване рівняння регресії:');

            console.log(`y = ${a0.toFixed(5)} + ${a1.toFixed(5)}\*x1 + ${a2.toFixed(5)}\*x2`);

            console.log('Зробимо перевірку по рядках:');

            console.log((a0 + a1 \* (x1Min) + a2 \* x2Min).toFixed(5));

            console.log((a0 + a1 \* (x1Max) + a2 \* x2Min).toFixed(5));

            console.log((a0 + a1 \* (x1Min) + a2 \* x2Max).toFixed(5));

        }

    }

    test(rUv);

}

main();

**Результати роботи програми:**

[Running] node "c:\Users\Bohdan\Desktop\МНД\laba2\tempCodeRunnerFile.js"

Стеценко|Богдан ІВ-93|Варіант - 324

Максимальне Y: -2940

Мінімальне Y: -3040

Матриця планування при m = 6

[ -3010, -2980, -2973, -3029, -2973, -3017 ]

[ -2949, -2950, -3026, -3035, -2980, -3001 ]

[ -3016, -3019, -3024, -3033, -2943, -2957 ]

------------------Середнє значення функції відгуку------------------

-2997

-2990.16667

-2998.66667

------------------Знайдемо дисперсію по рядках------------------

Дисперсія №1 = 505.66667

Дисперсія №2 = 1137.13889

Дисперсія №3 = 1228.22222

Основне відхилення = 1.2909944487358056

------------------Обчислимо Fuv------------------

Fuv1 = 2.24879

Fuv2 = 2.42892

Fuv3 = 1.08010

------------------Обчислимо θuv------------------

θuv1 = 1.49919

θuv2 = 1.61928

θuv3 = 0.72007

------------------Обчислимо Ruv------------------

Ruv1 = 0.38667

Ruv2 = 0.47969

Ruv3 = 0.21683

Оскільки m=6, візьмемо значення Rkr = 2.16 і довірчу ймовірність p = 0.99

0.38667 > 2.16

Підтвердження №1,що дисперсія однорідна

0.47969 > 2.16

Підтвердження №2,що дисперсія однорідна

0.21683 > 2.16

Підтвердження №3,що дисперсія однорідна

Розрахунок нормованих коефіцієнтів рівняння регресії.

Mx1 = -0.3333333333333333

Mx2 = -0.3333333333333333

My = -2995.2777800000003

A1 = 1

A2 = -0.3333333333333333

A3 = 1

A11 = 1001.8333333333334

A22 = 1001.8333333333334

b0 = -2994.4166700000005

b1 = 3.4166649999998953

b2 = -0.8333349999998179

Нормоване рівнянн регресії

y = -2994.4166700000005 + 3.4166649999998953\*x1 + -0.8333349999998179\*x2

Перевірка

-2997.0000000000005

-2990.166670000001

-2998.66667

Проведемо натуралізацію коефіцієнтів:

∆x1 = 15.00000

∆x2 = 17.50000

x10 = -15.00000

x20 = 32.50000

a0 = -2989.45238

a1 = 0.22778

a2 = -0.04762

Запишемо натуралізоване рівняння регресії:

y = -2989.45238 + 0.22778\*x1 + -0.04762\*x2

Зробимо перевірку по рядках:

-2997.00000

-2990.16667

-2998.66667

[Done] exited with code=0 in 0.166 seconds

**Висновок:**

Під час лабораторної роботи я провів двофакторний експеримент, перевірив однорідність дисперсії за критерієм Романовського, отримав коефіцієнти рівняння регресії, провів натуралізацію рівняння регресії. Результат виконання програми наведений вище.

**Контрольні питання:**

**1. Що таке регресійні поліноми і де вони застосовуються?**

Регресійні поліноми – це апроксимуючі поліноми, за допомогою яких ми можемо описати функцію. Застосовуються в теорії планування експерименту.

**2. Визначення однорідності дисперсії.**

Однорідність дисперсії означає, що серед усіх дисперсій немає такої, яка б значно перевищували інші.

**3. Що називається повним факторним експериментом?**

ПФЕ (Повний факторний експеримент) – називається такий експеримент, при реалізації якого визначається значення параметра оптимізації при всіх можливих поєднаннях рівнів варіювання факторів.