

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

ВИКОНАВ:
студент II курсу ФІОТ
групи ІВ-91
Красновський О. В.
Залікова - 9116

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Київ – 2021

Мета: 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 ($x_0=1$).
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.

Варіант завдання:

115	10	50	-20	60
-----	----	----	-----	----

Лістинг програми:

```
#include <bits/stdc++.h>

using namespace std;

int main() {
    srand(time(NULL));
    int x1min = 10, x1max = 50, x2min = -20, x2max = 60, ymin = -950, n = 5, p1sum = 0, p2sum = 0, p3sum = 0, Rkr = 2;
    double p1avg, p2avg, p3avg, m = 5, x1sum = 0, x2sum = 0;
    vector<int> p1, p2, p3, x1 = {-1, 1, -1}, x2 = {-1, -1, 1};
    double p1dispersion, p2dispersion, p3dispersion;
    for (int i = 0; i < n; i++) {
        p1.push_back((rand() % 101) + ymin);
        p1sum += p1[i];
        p2.push_back((rand() % 101) + ymin);
        p2sum += p2[i];
        p3.push_back((rand() % 101) + ymin);
        p3sum += p3[i];
    }
    p1avg = (double)p1sum / n;
    p2avg = (double)p2sum / n;
    p3avg = (double)p3sum / n;
    p1sum = 0, p2sum = 0, p3sum = 0;
    for (int i = 0; i < n; i++) {
        p1sum += (p1[i]-p1avg) * (p1[i]-p1avg);
        p2sum += (p2[i]-p2avg) * (p2[i]-p2avg);
        p3sum += (p3[i]-p3avg) * (p3[i]-p3avg);
    }
    p1dispersion = (double) p1sum / n;
    p2dispersion = (double) p2sum / n;
    p3dispersion = (double) p3sum / n;

    double romanovskiy_deviation = 1.79;
    double Fuv1 = p1dispersion / p2dispersion;
    double Fuv2 = p3dispersion / p1dispersion;
    double Fuv3 = p3dispersion / p2dispersion;
    double Tuv1 = ((m-2)/m)*Fuv1;
    double Tuv2 = ((m-2)/m)*Fuv2;
    double Tuv3 = ((m-2)/m)*Fuv3;
    double Ruv1 = abs(Tuv1-1) / romanovskiy_deviation;
    double Ruv2 = abs(Tuv2-1) / romanovskiy_deviation;
    double Ruv3 = abs(Tuv3-1) / romanovskiy_deviation;

    if (Ruv1 < Rkr && Ruv2 < Rkr && Ruv3 < Rkr) {
        cout << "The dispersion is homogeneous\n";
        double a1, a2, a3, a11, a22, an0, an1, an2;
        for (int i = 0; i < 3; i++) {
            x1sum += x1[i];
            x2sum += x2[i];
        }
        double mx1 = x1sum/x1.size();
        double mx2 = x2sum/x2.size();
        double pavg = (p1avg + p2avg + p3avg) / 3;

        a1 = (double)(x1[0] * x1[0] + x1[1] * x1[1] + x1[2] * x1[2]) / 3;
        a2 = (double)(x1[0] * x2[0] + x1[1] * x2[1] + x1[2] * x2[2]) / 3;
        a3 = (double)(x2[0] * x2[0] + x2[1] * x2[1] + x2[2] * x2[2]) / 3;
        a11 = (double)(x1[0]*p1avg + x1[1]*p2avg + x1[2]*p3avg)/3;
        a22 = (double)(x2[0]*p1avg + x2[1]*p2avg + x2[2]*p3avg)/3;
```

```

double d1 = (double)pavg*(a1*a3-a2*a2)-mx1*(a11*a3-a2*a22)+mx2*(a11*a2-a1*a22);
double d2 = (double)1*(a11*a3-a2*a22)-pavg*(mx1*a3-a2*mx2)+mx2*(mx1*a22-a11*mx2);
double d3 = (double)1*(a1*a22-a11*a2)-mx1*(mx1*a22-a11*mx2)+pavg*(mx1*a2-a1*mx2);
double d = (double)1*(a1*a3-a2*a2)-mx1*(mx1*a3-a2*mx2)+mx2*(mx1*a2-a1*mx2);

double b0 = d1/d;
double b1 = d2/d;
double b2 = d3/d;

double test1 = b0 + b1*x1[0] + b2*x2[0];
double test2 = b0 + b1*x1[1] + b2*x2[1];
double test3 = b0 + b1*x1[2] + b2*x2[2];

double dx1 = (double)abs(x1max-x1min)/2;
double dx2 = (double)abs(x2max-x2min)/2;
double x10 = (double)(x1max+x1min)/2;
double x20 = (double)(x2max+x2min)/2;

an0 = b0 - b1*(x10/dx1) - b2*(x20/dx2);
an1 = b1/dx1;
an2 = b2/dx2;

double test11 = an0 + an1*x1min + an2*x2min;
double test22 = an0 + an1*x1max + an2*x2min;
double test33 = an0 + an1*x1min + an2*x2max;

cout << "Experiment planning matrix:\n";
cout << "x1 x2 y1 y2 y3 y4 y5\n";
cout << x1min << " " << x2min << " " << p1[0] << " " << p1[1] << " " << p1[2] << " " << p1[3] << " " << p1[4] << "\n";
cout << x1max << " " << x2min << " " << p2[0] << " " << p2[1] << " " << p2[2] << " " << p2[3] << " " << p2[4] << "\n";
cout << x1min << " " << x2max << " " << p3[0] << " " << p3[1] << " " << p3[2] << " " << p3[3] << " " << p3[4] << "\n";
cout << "Normalized matrix of experiment planning:\n";
cout << "x1 x2\n";
cout << x1[0] << " " << x2[0] << "\n";
cout << " " << x1[1] << " " << x2[1] << "\n";
cout << x1[2] << " " << x2[2] << "\n\n";
cout << "Y average values: " << p1avg << ", " << p2avg << ", " << p3avg << "\n\n";
cout << "Normalized regression equation:\n";
cout << "y = " << b0 << " + " << b1 << " * x1 + " << b2 << " * x2\n";
cout << "Let's check:\n";
cout << b0 << " + " << b1 << " * " << x1[0] << " + " << b2 << " * " << x2[0] << " = " << test1 << "\n";
cout << b0 << " + " << b1 << " * " << x1[1] << " + " << b2 << " * " << x2[1] << " = " << test2 << "\n";
cout << b0 << " + " << b1 << " * " << x1[2] << " + " << b2 << " * " << x2[2] << " = " << test3 << "\n\n";
cout << "Naturalized regression equation:\n";
cout << "y = " << an0 << " + " << an1 << " * x1 + " << an2 << " * x2\n";
cout << "Let's check:\n";
cout << an0 << " + " << an1 << " * " << x1min << " + " << an2 << " * " << x2min << " = " << test11 << "\n";
cout << an0 << " + " << an1 << " * " << x1max << " + " << an2 << " * " << x2min << " = " << test22 << "\n";
cout << an0 << " + " << an1 << " * " << x1min << " + " << an2 << " * " << x2max << " = " << test33 << "\n\n";
cout << "Therefore, the coefficients of equations are correct.\n";
} else {
cout << "The dispersion is inhomogeneous\n";
}
}

```

Результат роботи програми:

```
C:\Users\demian\Desktop\II semestr\MND\lab2mnd>sol.exe
The dispersion is homogeneous
Experiment planning matrix:
x1  x2  y1  y2  y3  y4  y5
10 -20 -932 -903 -860 -902 -903
50 -20 -917 -882 -879 -930 -885
10  60 -934 -933 -909 -877 -857
Normalized matrix of experiment planning:
x1  x2
-1 -1
 1 -1
-1  1

Y average values: -900, -898.6, -902

Normalized regression equation:
y = -900.3 + 0.7 * x1 + -1 * x2
Let's check:
-900.3 + 0.7 * -1 + -1 * -1 = -900
-900.3 + 0.7 * 1 + -1 * -1 = -898.6
-900.3 + 0.7 * -1 + -1 * 1 = -902

Naturalized regression equation:
y = -900.85 + 0.035 * x1 + -0.025 * x2
Let's check:
-900.85 + 0.035 * 10 + -0.025 * -20 = -900
-900.85 + 0.035 * 50 + -0.025 * -20 = -898.6
-900.85 + 0.035 * 10 + -0.025 * 60 = -902

Therefore, the coefficients of equations are correct.

C:\Users\demian\Desktop\II semestr\MND\lab2mnd>
```

Контрольні питання:

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.

Обирають так названу «довірчу ймовірність» p – ймовірність, з якою вимагається підтвердити гіпотезу про однорідність дисперсій. У відповідності до p і кількості дослідів n обирають з таблиці критичне значення критерію. Кожне експериментальне значення критерію Романовського порівнюється зі значенням критерію Романовського за різних довірчих ймовірностей p і якщо кожне експериментальне значення менше за значення критерію, то гіпотеза про однорідність дисперсій підтверджується з ймовірністю p .

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.