

Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ

ЗВІТ
з лабораторної роботи №2
з навчальної дисципліни «МЕТОДИ ОБЧИСЛЕННЯ ТА ПЛАНУВАННЯ
ЕКСПЕРИМЕНТУ»

Тема:
« Проведення двофакторного експерименту з використанням лінійного
рівняння регресії »

Виконав:
студент 2-го курсу ФІОТ
групи ІО-91
Денисенко М. О.
Варіант - 7

Перевірив:
Регіда П. Г.

Київ 2021

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

Завдання:

107	-5	15	-15	35
-----	----	----	-----	----

Код програми:

```
1 import java.util.ArrayList;
2 import java.util.List;
3 import java.util.Scanner;
4
5 public class MopeLab2 {
6     public static void main(String[] args) {
7
8         final int MinX1 = -5;
9         final int MaxX1 = 15;
10        final int MinX2 = -15;
11        final int MaxX2 = 35;
12        final int MinY = 110;
13        final int MaxY = 210;
14        int m;
15        int[][] X = new int[][]{
16            {-1, -1},
17            {1, -1},
18            {-1, 1}};
19        List<String> symbols = new ArrayList<>();
20        double[] mx = new double[2];
21        double my = 0;
22        double[] a = new double[3];
23        double a11;
24        double a22;
25        double[] b = new double[3];
26        double[] y_average = new double[3];
27        final double[] RkrValue = {1.73, 2.16, 2.43, 2.62, 2.75, 2.9, 3.08};
28        double[] dispersion = new double[3];
29        double deviation;
30        double[] Fuv = new double[3];
31        double[] Theta_uv = new double[3];
32        double[] Ruv = new double[3];
33        double Rkr;
34        boolean work;
35
36        // определение кол-ва экспериментов
37
38        Scanner st = new Scanner(System.in);
39        System.out.println("Задайте значения m: ");
40        try {
41            m = st.nextInt();
42            if (m > 0) {
43                symbols.add("\u2219");
44                for (int i = 0; i <= m; i++) {
45                    symbols.add(" "+i);
46                }
47                work = true;
48            }
49            else{
50                System.out.println("Потрібно задати ціле значення...");
51                work = false;
52            }
53        }
54        catch (Exception e){
55            System.out.println("Потрібно задати додатне значення...");
56            m = 0;
57            work = false;
58        }
59
60
61        //проверка по критерию Романовского
62
```

```

while (work) {
    System.out.printf("Лінійне рівняння перспекії має вигляд: y = b%s + b%s*sx%s + b%s*sx%s",
        symbols.get(1), symbols.get(2), symbols.get(0), symbols.get(2), symbols.get(3), symbols.get(0),
        symbols.get(3));

    System.out.println();

    System.out.println("Нормована матриця планування експерименту : ");
    System.out.printf("X%s\tX%s\t", symbols.get(2), symbols.get(3));
    for (int i = 0; i < m; i++) {
        System.out.print("Y" + symbols.get(i + 2) + "\t");
    }
    System.out.println();

    List<int[]> Y = new ArrayList<>();

    for (int[] x : X) {
        int[] y = new int[m];
        for (int j = 0; j < 2; j++) {
            if (x[j] > 0){
                System.out.print(" ");
            }
            System.out.print(x[j] + "\t");
        }
        for (int j = 0; j < m; j++) {
            y[j] = (int) (Math.random() * (MaxY - MinY)) + MinY;
            System.out.print(y[j] + "\t");
        }
        Y.add(y);
        System.out.println();
    }

    for (int i = 0; i < 3; i++) {
        int[] y = Y.get(i);
        for (int j = 0; j < m; j++) {
            y_average[i] += (double) y[j] / m;
        }
    }

    for (int i = 0; i < 3; i++) {
        int[] y = Y.get(i);
        for (int j = 0; j < m; j++) {
            dispersion[i] += Math.pow((y[j] - y_average[i]), 2);
        }
    }

    deviation = Math.sqrt((2 * (2 * m - 2)) / (double) (m * (m - 4)));

    Fuv[0] = dispersion[0] / dispersion[1];
    Fuv[1] = dispersion[2] / dispersion[0];
    Fuv[2] = dispersion[2] / dispersion[1];

    if (m <= 4) {
        Rkr = RkrValue[0];
    }
    else if (m <= 6) Rkr = RkrValue[1];
    else if (m <= 8) Rkr = RkrValue[2];
    else if (m <= 10) Rkr = RkrValue[3];
    else if (m <= 13) Rkr = RkrValue[4];
    else if (m <= 17) Rkr = RkrValue[5];
    else if (m <= 20) Rkr = RkrValue[6];
    else Rkr = 3.1;

    System.out.println("\nRkr = " + Rkr);
    System.out.println("\nRuv:");
    boolean[] comparison = new boolean[3];
    for (int i = 0; i < 3; i++) {
        Theta_uv[i] = ((m - 2) / (double) m) * Fuv[i];
        Ruv[i] = Math.abs((Theta_uv[i] - 1) / deviation);
        if (Ruv[i] < Rkr){
            comparison[i] = true;
            System.out.println(Ruv[i] + " < " + Rkr);
        }
        else {
            comparison[i] = false;
            System.out.println(Ruv[i] + " > " + Rkr);
        }
    }
}

```

```

    }
    if (Ruv[0] < Rkr && Ruv[1] < Rkr && Ruv[2] < Rkr ) System.out.println("Дисперсії однорідні\n");
    work = !comparison[0] || !comparison[1] || !comparison[2];
    m++;
    symbols.add(" "+m);
    if (work){
        System.out.println("Ruv > Rkr\nЗБІЛЬШУЄМО КІЛЬКІСТЬ ДОСЛІДІВ: m + 1 ... \n");
    }
}

// расчет нормированных коэффициентов уравнения

for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 3; j++) {
        mx[i] += (double) X[j][i]/3;
    }
}

for (double y : y_average) {
    my += y/3;
}

a[0] = (Math.pow(X[0][0],2) + Math.pow(X[1][0],2) + Math.pow(X[2][0],2))/3;
a[1] = (double) (X[0][0]*X[0][1] + X[1][0]*X[1][1] + X[2][0]*X[2][1])/3;
a[2] = (Math.pow(X[0][1],2) + Math.pow(X[1][1],2) + Math.pow(X[2][1],2))/3;

a11 = (X[0][0]*y_average[0] + X[1][0]*y_average[1] + X[2][0]*y_average[2])/3;
a22 = (X[0][1]*y_average[0] + X[1][1]*y_average[1] + X[2][1]*y_average[2])/3;

double det11,det12,det21,det22,det31,det32;

det11 = (my*a[0]*a[2]) + (mx[0]*a[1]*a22) + (mx[1]*a11*a[1]) -
        (a22*a[0]*mx[1]) - (my*a[1]*a[1]) - (mx[0]*a11*a[2]);

det12 = (1*a[0]*a[2]) + (mx[0]*a[1]*mx[1]) + (mx[1]*mx[0]*a[1]) -
        (mx[1]*mx[1]*a[0]) - (mx[0]*mx[0]*a[2]) - (1*a[1]*a[1]);

det21 = (1*a11*a[2]) + (my*a[1]*mx[1]) + (mx[0]*a22*mx[1]) -
        (mx[1]*a11*mx[1]) - (mx[0]*my*a[2]) - (1*a22*a[1]);

det22 = (1*a[0]*a[2]) + (mx[0]*a[1]*mx[1]) + (mx[1]*mx[0]*a[1]) -
        (mx[1]*mx[1]*a[0]) - (mx[0]*mx[0]*a[2]) - (a[1]*a[1]*1);

det31 = (1*a[0]*a22) + (mx[0]*a11*mx[1]) + (mx[0]*a[1]*my) -
        (mx[1]*a[0]*my) - (mx[0]*mx[0]*a22) - (1*a[1]*a11);

det32 = (1*a[0]*a[2]) + (mx[0]*a[1]*mx[1]) + (mx[0]*a[1]*mx[1]) -
        (mx[1]*a[0]*mx[1]) - (mx[0]*mx[0]*a[2]) - (a[1]*a[1]*1);

b[0] = det11/det12;
b[1] = det21/det22;
b[2] = det31/det32;

System.out.println("Нормоване рівняння регресії:");
System.out.printf("y = %.2f",b[0]);
if (b[1] < 0 ) System.out.print(" - "); else System.out.print(" + ");
System.out.printf("%.2f%s%s", Math.abs(b[1]), symbols.get(0), symbols.get(2));
if (b[2] < 0 ) System.out.print(" - "); else System.out.print(" + ");
System.out.printf("%.2f%s%s\n", Math.abs(b[2]), symbols.get(0), symbols.get(3));

System.out.println("\nПеревірка:");
boolean[] correctly = new boolean[3];
for (int i = 0; i < 3; i++) {
    correctly[i] = (float) (b[0] + b[1] * X[i][0] + b[2] * X[i][1]) == (float) y_average[i];
    System.out.printf("%.2f = %.2f\n", (b[0] + b[1]*X[i][0] + b[2]*X[i][1]), y_average[i]);
}
if (correctly[0] && correctly[1] && correctly[2]){
    System.out.printf("\nНормовані коефіцієнти рівняння регресії b%s, b%s, b%s визначено правильно",
        symbols.get(1), symbols.get(2), symbols.get(3));
    System.out.println();
}
else{
    System.out.printf("Нормовані коефіцієнти рівняння регресії b%s, b%s, b%s визначено неправильно",
        symbols.get(1), symbols.get(2), symbols.get(3));
    System.out.println();
}
}

```

```

215
216 // натуралізація коефіцієнтів
217
218 double deltaX1, deltaX2, x10, x20, a0, a1, a2;
219
220 deltaX1 = (double) Math.abs(MaxX1 - MinX1)/2;
221 deltaX2 = (double) Math.abs(MaxX2 - MinX2)/2;
222 x10 = (double) (MaxX1 + MinX1)/2;
223 x20 = (double) (MaxX2 + MinX2)/2;
224
225 a0 = b[0] - b[1]*x10/deltaX1 - b[2]*x20/deltaX2;
226 a1 = b[1]/deltaX1;
227 a2 = b[2]/deltaX2;
228
229 System.out.println();
230 System.out.println("Натуралізоване рівняння регресії:");
231
232 System.out.printf("y = %.2f", a0);
233 if (a1 < 0 ){
234     System.out.print(" - ");
235 } else System.out.print(" + ");
236 System.out.printf("%.2f%s%s", Math.abs(a1), symbols.get(0), symbols.get(2));
237 if (a2 < 0 ){
238     System.out.print(" - ");
239 } else System.out.print(" + ");
240 System.out.printf("%.2f%s%s\n", Math.abs(a2), symbols.get(0), symbols.get(3));
241
242 System.out.println();
243 System.out.println("Перевірка:");
244
245 System.out.printf("%.2f = %.2f\n", (a0 + a1*MinX1 + a2*MinX2), y_average[0]);
246 System.out.printf("%.2f = %.2f\n", (a0 + a1*MaxX1 + a2*MinX2), y_average[1]);
247 System.out.printf("%.2f = %.2f\n", (a0 + a1*MinX1 + a2*MaxX2), y_average[2]);
248
249 if ((float)(a0 + a1*MinX1 + a2*MinX2) == (float)y_average[0] &&
250     (float)(a0 + a1*MaxX1 + a2*MinX2) == (float)y_average[1] &&
251     (float)(a0 + a1*MinX1 + a2*MaxX2) == (float)y_average[2]){
252     System.out.println();
253     System.out.printf("Коефіцієнти натуралізованого рівняння регресії a%s, a%s, a%s визначено правильно",
254                       symbols.get(1), symbols.get(2), symbols.get(3));
255 }
256 else{
257     System.out.printf("Коефіцієнти натуралізованого рівняння регресії a%s, a%s, a%s визначено неправильно",
258                       symbols.get(1), symbols.get(2), symbols.get(3));
259 }
260 }
261 }

```

Результат роботи:

Задайте значення m:

6

Лінійне рівняння регресії має вигляд: $y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2$

Нормована матриця планування експерименту :

X1	X2	Y1	Y2	Y3	Y4	Y5	Y6
-1	-1	143	198	164	171	161	164
1	-1	145	198	122	132	122	188
-1	1	204	135	159	163	135	150

Rkr = 2.16

Ruv:

0.627912983077117 < 2.16

0.27416056781886644 < 2.16

0.4766953545336424 < 2.16

Дисперсії однорідні

Нормоване рівняння регресії:

$y = 154,42 - 7,83 \cdot x_1 - 4,58 \cdot x_2$

Перевірка:

166,83 = 166,83

151,17 = 151,17

157,67 = 157,67

Нормовані коефіцієнти рівняння регресії b_0 , b_1 , b_2 визначено правильно

Натуралізоване рівняння регресії:

$y = 160,17 - 0,78 \cdot x_1 - 0,18 \cdot x_2$

Перевірка:

166,83 = 166,83

151,17 = 151,17

157,67 = 157,67

Коефіцієнти натуралізованого рівняння регресії a_0 , a_1 , a_2 визначено правильно

Process finished with exit code 0

|

Відповіді на теоретичні 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 m обирають з таблиці критичне значення критерію $R_{кр}$. Кожне експериментальне значення $R_{ув}$ критерію Романовського порівнюється з $R_{кр}$. (значення критерію Романовського за різних довірчих ймовірностей p) і якщо для усіх кожне $R_{ув} < R_{кр}$, 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 багатofакторному експерименті використані всі можливі комбінації рівнів факторів, то такий експеримент називається повним факторним експериментом.