Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 4

з дисципліни «Методи наукових досліджень»

на тему «Проведення трьохфакторного експерименту при використанні рівняння регресії з урахуванням ефекту взаємодії»

ВИКОНАВ:
студент 2 курсу
групи ІВ-91
Дерачиц В. В.
Залікова – 9109

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Київ – 2021

Мета: проведення трьохфакторного експерименту при використанні рівняння регресії з урахуванням ефекту взаємодії.

Завдання:

Завдання на лабораторну роботу

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.

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

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

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.

Варіанти обираються по номеру в списку в журналі викладача.

№варианта	X_1		X_2		X_3	
	min	max	min	max	min	max
109	-30	0	-35	10	0	20

Програмний код

```
from random import randint


def main(m_tmp):
    m = m_tmp
    N = 8
    d = 8
    print("y' = b0 + b1*x1 + b2*x2 + b3*x3 + b12*x1*x2 + b13x1*x3 + b23*x2*x3
+ b123*x1*x2*x3")

    x1 = [-30, 0]
    x2 = [-35, 10]
    x3 = [0, 20]

    print("x1_min =", x1[0], ", x1_max = ", x1[1])
    print("x2_min =", x2[0], ", x2_max = ", x2[1])
    print("x3_min =", x3[0], ", x3_max = ", x3[1])

    xcp_min = (x1[0] + x2[0] + x3[0])/3
    xcp_max = (x1[1] + x2[1] + x3[1])/3

    print("\nXcp min = ", xcp_min)
    print("Xcp max = ", xcp_max)

    y_min = 200 + xcp_min
    y_max = 200 + xcp_max

    print("\nY min = ", y_min)
    print("Y max = ", y_max)

    y_matrix = [[randint(int(y_min), int(y_max)) for i in range(m)] for k in
range(N)]
    print("\nY matrix:\n", y_matrix)

    average_y = [round(sum(i) / len(i), 3) for i in y_matrix]
    print("\nAverage y:\n", average_y)

    normalized_x = [[-1, -1, -1],
                    [-1, -1, 1],
                    [-1, 1, -1],
                    [-1, 1, 1],
                    [1, -1, -1],
                    [1, -1, 1],
                    [1, 1, -1],
                    [1, 1, 1]]

    b0 = sum(average_y) / N
    b1 = sum([average_y[i] * normalized_x[i][0] for i in range(N)]) / N
    b2 = sum([average_y[i] * normalized_x[i][1] for i in range(N)]) / N
    b3 = sum([average_y[i] * normalized_x[i][2] for i in range(N)]) / N
    b12 = sum([average_y[i] * normalized_x[i][0] * normalized_x[i][1] for i
in range(N)]) / N
    b13 = sum([average_y[i] * normalized_x[i][0] * normalized_x[i][2] for i
in range(N)]) / N
    b23 = sum([average_y[i] * normalized_x[i][1] * normalized_x[i][2] for i
in range(N)]) / N
    b123 = sum([average_y[i] * normalized_x[i][0] * normalized_x[i][1] *
normalized_x[i][2] for i in range(N)]) / N

    plan_matrix = [[x1[0], x2[0], x3[0], x1[0] * x2[0], x1[0] * x3[0], x2[0]
```

```
* x3[0], x1[0] * x2[0] * x3[0]],
                    [x1[0], x2[0], x3[1], x1[0] * x2[0], x1[0] * x3[1], x2[0]
* x3[1], x1[0] * x2[0] * x3[1]],
                    [x1[0], x2[1], x3[0], x1[0] * x2[1], x1[0] * x3[0], x2[1]
* x3[0], x1[0] * x2[1] * x3[0]],
                    [x1[0], x2[1], x3[1], x1[0] * x2[1], x1[0] * x3[1], x2[1]
* x3[1], x1[0] * x2[1] * x3[1]],
                    [x1[1], x2[0], x3[0], x1[1] * x2[0], x1[1] * x3[0], x2[0]
* x3[0], x1[1] * x2[0] * x3[0]],
                    [x1[1], x2[0], x3[1], x1[1] * x2[0], x1[1] * x3[1], x2[0]
* x3[1], x1[1] * x2[0] * x3[1]],
                    [x1[1], x2[1], x3[0], x1[1] * x2[1], x1[1] * x3[0], x2[1]
* x3[0], x1[1] * x2[1] * x3[0]],
                    [x1[1], x2[1], x3[1], x1[1] * x2[1], x1[1] * x3[1], x2[1]
* x3[1], x1[1] * x2[1] * x3[1]]]

    print("\nМатриця планування: \n", plan_matrix)

    normalized_plan_matrix = [[-1, -1, -1, 1, 1, 1, -1],
                              [-1, -1, 1, 1, -1, -1, 1],
                              [-1, 1, -1, -1, 1, -1, 1],
                              [-1, 1, 1, -1, -1, 1, -1],
                              [1, -1, -1, -1, -1, 1, 1],
                              [1, -1, 1, -1, 1, -1, -1],
                              [1, 1, -1, 1, -1, -1, -1],
                              [1, 1, 1, 1, 1, 1, 1]]

    result_y = []
    for i in range(N):
        result_y.append(b0 + b1 * plan_matrix[i][0] + b2 * plan_matrix[i][1]
+ b3 * plan_matrix[i][2] +
                        b12 * plan_matrix[i][3] + b13 * plan_matrix[i][4] +
b23 * plan_matrix[i][5] +
                        b123 * plan_matrix[i][6])

    print("\nПЕРЕВІРКА ОДНОРІДНОСТІ ДИСПЕРСІЇ ЗА КРИТЕРІЄМ КОХРЕНА")
    matrix_dispersion_y = [sum([(y_matrix[j][i] - average_y[i]) ** 2 for i in
range(m)]) / m for j in range(N)]
    print("dispersion: \n", matrix_dispersion_y)

    gp = max(matrix_dispersion_y) / sum(matrix_dispersion_y)
    print("Gp = ", gp)

    # f1=m-1=2, f2=N=8, q=0.05 => Gт=0.5157 за таблицею
    if gp > 0.5157:
        print("Дисперсія неоднорідна!")
        m += 1
        main(m)
    else:
        print("Gp < 0.5157 => Дисперсія однорідна")

    print("\nПЕРЕВІРКА ЗНАЧУЩОСТІ КОЕФІЦІЄНТІВ ЗА КРИТЕРІЄМ СТЬЮДЕНТА")
    s2b = sum(matrix_dispersion_y) / N
    s2bs = s2b / (m * N)
    sbs = s2bs ** (1/2)
    print("sbs = ", sbs)

    b_array = [b0, b1, b2, b3, b12, b13, b23, b123]
    t_array = [abs(b_array[i]) / sbs for i in range(N)]

    print("beta: ", b_array)
    print("t: ", t_array, "\n")
    b_result = b_array
```

```
    f1 = m - 1
    f2 = N
    f3 = f1 * f2

    for i in range(N):
        if t_array[i] < 2.120:
            b_result[i] = 0
            d -= 1
            print('Виключаємо з рівняння статистично незначущий коефіцієнт
b', i)

    y_reg = [b_result[0] + b_result[1] * plan_matrix[i][0] + b_result[2] *
plan_matrix[i][1] + b_result[3] * plan_matrix[i][2] + b_result[4] *
plan_matrix[i][3] + b_result[5] * plan_matrix[i][4] + b_result[6] *
plan_matrix[i][5] + b_result[7] * plan_matrix[i][6] for i in range(N)]
    print("Значення рівнянь регресій:\n", y_reg)

    print("\nПЕРЕВІРКА АДЕКВАТНОСТІ ЗА КРИТЕРІЄМ ФІШЕРА")
    f4 = N - d
    sad = (m / (N - d)) * int(sum([(y_reg[i] - average_y[i]) ** 2 for i in
range(N)]))
    Fp = sad / s2b
    print("Кількість значимих коефіцієнтів:", d)
    print("\nFp = ", Fp)

    if Fp > 4.5:
        print("Рівняння регресії неадекватно оригіналу при рівні значимості
0.05")
    else:
        print("Рівняння регресії адекватно оригіналу при рівні значимості
0.05")

    print("Рівняння: \n "
          f"y = {b0} + {b1} * x1 + {b2} * x2 + {b3} * x3 + {b12} * x1x2 +
{b13} * x1x3 + {b23} * x2x3 + {b123} * x1x2x3")


if __name__ == '__main__':
    main(3)
```

Результат роботи програми

```
C:\Users\derac\Anaconda3\python.exe "D:/Repo/MND/Laboratory work №4/main.py"
y' = b0 + b1*x1 + b2*x2 + b3*x3 + b12*x1*x2 + b13x1*x3 + b23*x2*x3 + b123*x1*x2*x3
x1_min = -30 , x1_max =  0
x2_min = -35 , x2_max =  10
x3_min = 0 , x3_max =  20

Xcp min =  -21.666666666666668
Xcp max =  10.0

Y min =  178.33333333333334
Y max =  210.0

Y matrix:
 [[208, 194, 199], [186, 205, 188], [202, 191, 185], [203, 201, 204], [208, 184, 181], [200, 178, 210], [182, 185, 192], [184, 203, 198]]

Average y:
 [200.333, 193.0, 192.667, 202.667, 191.0, 196.0, 186.333, 192.333]

Матриця планування:
 [[-30, -35, 0, 1050, 0, 0, 0], [-30, -35, 20, 1050, -600, -700, 21000], [-30, 10, 0, -300, 0, 0, 0], [-30, 10, 20, -300, -600, 200, -6000], [0, -35, 0, 0, 0, 0, 0],

ПЕРЕВІРКА ОДНОРІДНОСТІ ДИСПЕРСІЇ ЗА КРИТЕРІЄМ КОХРЕНА
dispersion:
 [33.29659266666667, 123.73859266666665, 21.853926000000012, 66.51659266666665, 91.96725933333335, 175.18125933333332, 133.51459266666663, 124.62659266666664]
Gp =  0.22730284560529435
Gp < 0.5157 => Дисперсія однорідна

ПЕРЕВІРКА ЗНАЧУЩОСТІ КОЕФІЦІЄНТІВ ЗА КРИТЕРІЄМ СТЬЮДЕНТА
sbs =  2.003506571821848
beta:  [194.291625, -2.875125000000004, -0.7916249999999962, 1.7083750000000002, -1.2918750000000045, 1.0416249999999998, 2.291625, -2.041625]
t:  [96.97578622031915, 1.435046453271959, 0.39511974212300593, 0.8526924862774591, 0.6448069690259435, 0.5199009649630544, 1.143807079163288, 1.019025856323241]

Виключаємо з рівняння статистично незначущий коефіцієнт b 1
Виключаємо з рівняння статистично незначущий коефіцієнт b 2
Виключаємо з рівняння статистично незначущий коефіцієнт b 3
Виключаємо з рівняння статистично незначущий коефіцієнт b 4
Виключаємо з рівняння статистично незначущий коефіцієнт b 5
Виключаємо з рівняння статистично незначущий коефіцієнт b 6
Виключаємо з рівняння статистично незначущий коефіцієнт b 7
Значення рівнянь регресій:
 [194.291625, 194.291625, 194.291625, 194.291625, 194.291625, 194.291625, 194.291625, 194.291625]
ПЕРЕВІРКА АДЕКВАТНОСТІ ЗА КРИТЕРІЄМ ФІШЕРА
Кількість значимих коефіцієнтів: 1

Fp =  0.8496964378658175
Рівняння регресії адекватно оригіналу при рівні значимості 0.05
Рівняння:
 y = 194.291625 + -2.875125000000004 * x1 + -0.7916249999999962 * x2 + 1.7083750000000002 * x3 + -1.2918750000000045 * x1x2 + 1.0416249999999998 * x1x3 + 2.291625 * x2x3 + -2.041625 * x1x2x3

Process finished with exit code 0
```

Висновок: виконуючи дану лабораторну роботу, я провів трьохфакторний експеримент, склав матрицю планування та знайшов коефіцієнти рівняння регресії, провів статистичні перевірки. Результати роботи програми наведені вище пыдтверджують правильність виконання лабораторної роботи.