

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 № 4**

з дисципліни «Методи оптимізації та планування експерименту» на  
тему

**«Проведення трьохфакторного експерименту  
при використанні рівняння регресії з урахуванням  
ефекту взаємодії.»**

ВИКОНАВ:  
студент II курсу ФІОТ  
групи ІВ-91  
Карамшук Володимир  
Варіант: 113

ПЕРЕВІРИВ:  
Регіда П. Г.

Київ – 2021

## Хід роботи

### Мета:

Провести повний трьохфакторний експеримент. Знайти рівняння регресії

адекватне об'єкту.

### Завдання:

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.

113	-40	20	-35	15	20	25
-----	-----	----	-----	----	----	----

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

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

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

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.

## Програмний код

```
import random
import numpy as np
from numpy.linalg import solve
from scipy.stats import f,t

# кількість повторення дослідів
n = 8

# варіант 113
x1min = -40
x1max = 20
x2min = -35
x2max = 15
x3min = 20
x3max = 25

# максимальне та мінімальне значення
y_max = 200 + (x1max + x2max + x3max) / 3
y_min = 200 + (x1min + x2min + x3min) / 3

# матриця ПФЕ
xn = [[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]]

x1x2_norm, x1x3_norm, x2x3_norm, x1x2x3_norm = [0] * 8, [0] * 8, [0] * 8, [0] * 8

for i in range(n):
    x1x2_norm[i] = xn[1][i] * xn[2][i]
    x1x3_norm[i] = xn[1][i] * xn[3][i]
    x2x3_norm[i] = xn[2][i] * xn[3][i]
    x1x2x3_norm[i] = xn[1][i] * xn[2][i] * xn[3][i]

# заповнення y (генерація)
y1 = [random.randint(int(y_min), int(y_max)) for i in range(8)]
y2 = [random.randint(int(y_min), int(y_max)) for i in range(8)]
y3 = [random.randint(int(y_min), int(y_max)) for i in range(8)]

# матриця планування
y_matrix = [[y1[0], y2[0], y3[0]],
            [y1[1], y2[1], y3[1]],
            [y1[2], y2[2], y3[2]],
            [y1[3], y2[3], y3[3]],
            [y1[4], y2[4], y3[4]],
            [y1[5], y2[5], y3[5]],
            [y1[6], y2[6], y3[6]],
            [y1[7], y2[7], y3[7]]]

# вивід даних за допомогою цикла
print("Матриця планування y : \n")
for i in range(n):
    print(y_matrix[i])

x0 = [1, 1, 1, 1, 1, 1, 1, 1]

# заміна -1 на x1_мін, 1 на x1_макс
x1 = [-40, -40, 20, 20, -40, -40, 20, 20]

# заміна -1 на x2_мін, 1 на x2_макс
x2 = [20, -35, 20, -35, 20, -35, 20, -35]
```

```

# заміна -1 на x3_мін, 1 на x3_макс
x3 = [20, 25, 25, 20, 25, 20, 20, 25]

# заповнення нулями x1x2, x1x3, x1x2x3
x1x2, x1x3, x2x3, x1x2x3 = [0] * 8, [0] * 8, [0] * 8, [0] * 8
# заповнення x1x2, x1x3, x1x2x3 добутками
for i in range(n):
    x1x2[i] = x1[i] * x2[i]
    x1x3[i] = x1[i] * x3[i]
    x2x3[i] = x2[i] * x3[i]
    x1x2x3[i] = x1[i] * x2[i] * x3[i]

# середні y
Y_average = []
for i in range(len(y_matrix)):
    Y_average.append(np.mean(y_matrix[i], axis=0))

# формуємо списки b і a
list_for_b = [xn[0], xn[1], xn[2], xn[3], x1x2_norm, x1x3_norm, x2x3_norm,
x1x2x3_norm]
list_for_a = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3))

# вивід матриці планування X
print("Матриця планування X:")
for i in range(n):
    print(list_for_a[i])
# нормовані фактори b_i
bi = []
for k in range(n):
    S = 0
    for i in range(n):
        S += (list_for_b[k][i] * Y_average[i]) / n
    bi.append(round(S, 3))

# розрахунок ai (система рівнянь) через функцію solve, вивід рівняння регресії
ai = [round(i, 3) for i in solve(list_for_a, Y_average)]
print("Рівняння регресії: \n" "y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1x2 +
{}*x1x3 + {}*x2x3 + {}*x1x2x3".format(ai[0],
ai[1], ai[2], ai[3], ai[4], ai[5], ai[6], ai[7]))
# вивід даних
print("Рівняння регресії для нормованих факторів: \n" "y = {} + {}*x1 + {}*x2
+ {}*x3 + {}*x1x2 + {}*x1x3 +
" {}*x2x3 + {}*x1x2x3".format(bi[0], bi[1], bi[2], bi[3], bi[4], bi[5],
bi[6], bi[7]))

print("Перевірка за критерієм Кохрена")
print("Середні значення відгуку за рядками:", "\n", +Y_average[0],
Y_average[1], Y_average[2], Y_average[3],
Y_average[4], Y_average[5], Y_average[6], Y_average[7])
# розрахунок дисперсій
dispersions = []
for i in range(len(y_matrix)):
    a = 0
    for k in y_matrix[i]:
        a += (k - np.mean(y_matrix[i], axis=0)) ** 2
    dispersions.append(a / len(y_matrix[i]))
# експериментально
Gp = max(dispersions) / sum(dispersions)
# теоретично
Gt = 0.5157
# перевірка однорідності дисперсій
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")

```

```

# критерій Стюдента
print("Перевірка значущості коефіцієнтів за критерієм Стюдента")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (8 * 3)) ** 0.5

t_list = [abs(bi[i]) / sbs for i in range(0, 8)]

d = 0
res = [0] * 8
coef_1 = []
coef_2 = []
# кількість повторень кожної комбінації
m = 3
F3 = (m - 1) * n
# перевірка значущості коефіцієнтів
for i in range(n):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coef_2.append(bi[i])
        res[i] = 0
    else:
        coef_1.append(bi[i])
        res[i] = bi[i]
        d += 1

# вивід
print("Значущі коефіцієнти регресії:", coef_1)
print("Незначущі коефіцієнти регресії:", coef_2)

# значення y з коефіцієнтами регресії
y_st = []
for i in range(n):
    y_st.append(res[0] + res[1] * xn[1][i] + res[2] * xn[2][i] + res[3] *
xn[3][i] + res[4] * x1x2_norm[i] \
                + res[5] * x1x3_norm[i] + res[6] * x2x3_norm[i] + res[7] *
x1x2x3_norm[i])
print("Значення з отриманими коефіцієнтами:\n", y_st)

# критерій Фішера
print("\nПеревірка адекватності за критерієм Фішера\n")
Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(8)]) / (n - d)
Fp = Sad / sb
F4 = n - d
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("Рівняння регресії адекватне при рівні значимості 0.05")
else:
    print("Рівняння регресії неадекватне при рівні значимості 0.05")

```

## Приклад роботи програми

```
C:\Users\qazse\anaconda3\python.exe C:/Users/qazse/PycharmProjects/MND_4/main.py
Матриця планування y :

[207, 196, 219]
[206, 199, 189]
[191, 202, 207]
[205, 206, 206]
[198, 200, 182]
[206, 192, 217]
[214, 200, 201]
[192, 200, 192]
Матриця планування X:
(1, -40, 20, 20, -800, -800, 400, -16000)
(1, -40, -35, 25, 1400, -1000, -875, 35000)
(1, 20, 20, 25, 400, 500, 500, 10000)
(1, 20, -35, 20, -700, 400, -700, -14000)
(1, -40, 20, 25, -800, -1000, 500, -20000)
(1, -40, -35, 20, 1400, -800, -700, 28000)
(1, 20, 20, 20, 400, 400, 400, 8000)
(1, 20, -35, 25, -700, 500, -875, -17500)

Рівняння регресії:
y = 240.081 + -0.306*x1 + -0.115*x2 + -1.721*x3 + -0.017*x1x2 + 0.014*x1x3 + 0.006*x2x3 + 0.001*x1x2x3
Рівняння регресії для нормованих факторів:
y = 201.125 + 0.208*x1 + -0.292*x2 + -4.625*x3 + -0.875*x1x2 + 0.625*x1x3 + 0.125*x2x3 + -1.625*x1x2x3
Перевірка за критерієм Кохрена
Середні значення відгуку за рядками:
207.33333333333334 198.0 200.0 205.66666666666666 193.33333333333334 205.0 205.0 194.66666666666666
Дисперсія однорідна
Перевірка значущості коефіцієнтів за критерієм Стюдента
Значущі коефіцієнти регресії: [201.125, -4.625]
Незначущі коефіцієнти регресії: [0.208, -0.292, -0.875, 0.625, 0.125, -1.625]
Значення з отриманими коефіцієнтами:
[205.75, 196.5, 196.5, 205.75, 196.5, 205.75, 205.75, 196.5]

Перевірка адекватності за критерієм Фішера

Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0
```

### Висновок:

В даній лабораторній роботі проведено повний трьохфакторний експеримент.

Знайдено рівняння регресії адекватне об'єкту. Кінцеву мети досягнуто.