

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 планування експерименту»

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

ВИКОНАВ:  
студент II курсу ФІОТ  
групи ІВ-92  
Чередник Віталій Юрійович  
Номер у списку групи: 26

ПЕРЕВІРИВ:  
ас. Регіда П.Г.

## Хід роботи

**Мета:** Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

### 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.

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

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

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

3. Знайти коефіцієнти рівняння регресії і записати його.
4. Провести 3 статистичні перевірки – за критеріями Кохрена, Стюдента, 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.

### Варіант завдання

№ <sub>варіанта</sub>	$x_1$		$x_2$		$x_3$	
	min	max	min	Max	min	max
226	-25	-5	15	50	-25	-15

### Код програми

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

x1min = -25
x1max = -5
x2min = 15
x2max = 50
x3min = -25
```

```

x3max = -15

ymax = 200 + (x1max + x2max + x3max) / 3
ymin = 200 + (x1min + x2min + x3min) / 3

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 = [0] * 8
x1x3_norm = [0] * 8
x2x3_norm = [0] * 8
x1x2x3_norm = [0] * 8
for i in range(8):
    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]

y1 = [random.randint(int(ymin), int(ymax)) for i in range(8)]
y2 = [random.randint(int(ymin), int(ymax)) for i in range(8)]
y3 = [random.randint(int(ymin), int(ymax)) for i in range(8)]

Y = [[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]]]
print("Матриця планування Y (m=3):")
for i in range(8):
    print(Y[i])

x0 = [1, 1, 1, 1, 1, 1, 1, 1]
x1 = [-25, -25, -5, -5, -25, -25, -5, -5]
x2 = [15, 50, 15, 50, 15, 50, 15, 50]
x3 = [-25, -15, -15, -25, -15, -25, -25, -15]
x1x2 = [0] * 8
x1x3 = [0] * 8
x2x3 = [0] * 8
x1x2x3 = [0] * 8
for i in range(8):
    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]

Y_average = []
for i in range(len(Y)):
    Y_average.append(np.mean(Y[i], axis=0))

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))
print("Матриця планування X:")
for i in range(8):
    print(list_for_a[i])
bi = []
for k in range(8):

```

```

S = 0
for i in range(8):
    S += (list_for_b[k][i] * Y_average[i]) / 8
bi.append(round(S, 5))

ai = [round(i, 5) 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]))

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])
dispersions = []
for i in range(len(Y)):
    a = 0
    for k in Y[i]:
        a += (k - np.mean(Y[i], axis=0)) ** 2
    dispersions.append(a / len(Y[i]))

Gp = max(dispersions) / sum(dispersions)
Gt = 0.5157
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")

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
coefs1 = []
coefs2 = []
m = 3
n = 8
F3 = (m - 1) * n
for i in range(8):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coefs2.append(bi[i])
        res[i] = 0
    else:
        coefs1.append(bi[i])
        res[i] = bi[i]
        d += 1
print("Значущі коефіцієнти регресії:", coefs1)
print("Незначущі коефіцієнти регресії:", coefs2)

y_st = []
for i in range(8):
    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:", y_st)

print("----- Перевірка адекватності за критерієм Фішера -----
-----")
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")

```

## Результати роботи програми

```

Матриця планування Y (m=3):
[189, 210, 199]
[195, 198, 189]
[189, 190, 209]
[209, 202, 205]
[196, 188, 194]
[191, 206, 192]
[208, 210, 196]
[201, 189, 196]
Матриця планування X:
(1, -25, 15, -25, -375, 625, -375, 9375)
(1, -25, 50, -15, -1250, 375, -750, 18750)
(1, -5, 15, -15, -75, 75, -225, 1125)
(1, -5, 50, -25, -250, 125, -1250, 6250)
(1, -25, 15, -15, -375, 375, -225, 5625)
(1, -25, 50, -25, -1250, 625, -1250, 31250)
(1, -5, 15, -25, -75, 125, -375, 1875)
(1, -5, 50, -15, -250, 75, -750, 3750)
Рівняння регресії:
y = 185.35119 + 0.24167*x1 + -0.15119*x2 + -0.79881*x3 + -0.015*x1x2 + 0.00214*x1x3 + -0.00786*x2x3 + -0.00081*x1x2x3
Рівняння регресії для нормованих факторів:
y = 197.95833 + 2.375*x1 + -0.20833*x2 + -3.45833*x3 + 0.20833*x1x2 + -1.20833*x1x3 + 0.375*x2x3 + -0.70833*x1x2x3
----- Перевірка за критерієм Кохрена -----
Середні значення відгуку за рядками:
199.33333333333334 194.0 196.0 205.33333333333334 192.66666666666666 196.33333333333334 204.66666666666666 195.33333333333334
Дисперсія однорідна
----- Перевірка значущості коефіцієнтів за критерієм Стюдента -----
Значущі коефіцієнти регресії: [197.95833, -3.45833]
Незначущі коефіцієнти регресії: [2.375, -0.20833, 0.20833, -1.20833, 0.375, -0.70833]
Значення з отриманими коефіцієнтами: [201.41665999999998, 194.5, 194.5, 201.41665999999998, 194.5, 201.41665999999998, 201.41665999999998, 194.5]
----- Перевірка адекватності за критерієм Фішера -----
Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0

```