

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 ІО-93  
Бриль Владислав  
Залікова – 9303  
Номер у списку: 2

ПЕРЕВІРИВ:  
Асистент Регіда П. Г.

**Мета роботи:** 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.
3. Знайти коефіцієнти рівняння регресії і записати його.

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

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

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

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:**

602	20	60	10	50	30	35
-----	----	----	----	----	----	----

**Програмний код:**

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

x1min = 20
x1max = 60
x2min = 10
x2max = 50
x3min = 30
x3max = 35
n = 8
y_max = 200 + (x1max + x2max + x3max) / 3
y_min = 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, 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]

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)]

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]]]

print("Матриця планування y : \n")
for i in range(n):
    print(y_matrix[i])

x0 = [1, 1, 1, 1, 1, 1, 1, 1]
x1 = [20, 20, 60, 60, 20, 20, 60, 60]
x2 = [10, 50, 10, 50, 10, 50, 10, 50]
x3 = [30, 35, 35, 30, 35, 30, 30, 35]
x1x2, x1x3, x2x3, x1x2x3 = [0] * 8, [0] * 8, [0] * 8, [0] * 8
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]
Y_average = []
for i in range(len(y_matrix)):
    Y_average.append(np.mean(y_matrix[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(n):
    print(list_for_a[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))
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]))
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_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]))
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
coef_1 = []
coef_2 = []
m = 3
F3 = (m - 1) * n
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
print("Значущі коефіцієнти регресії:", coef_1)
print("Незначущі коефіцієнти регресії:", coef_2)

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)

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")

```

## Вивід програми:

```
C:\Users\Владислав\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Владислав/PycharmProjects/MOPE_LAB_4/MOPE_LAB_4.py
Матриця планування y :
```

```
[247, 236, 238]
[230, 247, 248]
[228, 247, 242]
[225, 222, 228]
[239, 227, 245]
[220, 228, 239]
[228, 221, 247]
[246, 241, 230]
```

```
Матриця планування X:
```

```
(1, 20, 10, 30, 200, 600, 300, 6000)
(1, 20, 50, 35, 1000, 700, 1750, 35000)
(1, 60, 10, 35, 600, 2100, 350, 21000)
(1, 60, 50, 30, 3000, 1800, 1500, 90000)
(1, 20, 10, 35, 200, 700, 350, 7000)
(1, 20, 50, 30, 1000, 600, 1500, 30000)
(1, 60, 10, 30, 600, 1800, 300, 18000)
(1, 60, 50, 35, 3000, 2100, 1750, 105000)
```

```
Рівняння регресії:
```

```
y = 329.625 + -2.123*x1 + -3.412*x2 + -2.725*x3 + 0.036*x1x2 + 0.063*x1x3 + 0.102*x2x3 + -0.001*x1x2x3
```

```
Рівняння регресії для нормованих факторів:
```

```
y = 235.375 + -1.625*x1 + -1.708*x2 + 3.792*x3 + -0.042*x1x2 + 1.458*x1x3 + 2.875*x2x3 + -1.125*x1x2x3
```

```
Перевірка за критерієм Кохрена
```

```
Середні значення відгуку за рядками:
```

```
240.33333333333334 241.66666666666666 239.0 225.0 237.0 229.0 232.0 239.0
```

```
Дисперсія однорідна
```

```
Перевірка значущості коефіцієнтів за критерієм Стюдента
```

```
Значущі коефіцієнти регресії: [235.375, 3.792]
```

```
Незначущі коефіцієнти регресії: [-1.625, -1.708, -0.042, 1.458, 2.875, -1.125]
```

```
Значення з отриманими коефіцієнтами:
```

```
[231.583, 239.167, 239.167, 231.583, 239.167, 231.583, 231.583, 239.167]
```

```
Перевірка адекватності за критерієм Фішера
```

```
Рівняння регресії адекватне при рівні значимості 0.05
```

```
Process finished with exit code 0
```