

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

ЛАБОРАТОРНА РОБОТА № 3  
з дисципліни «Методи наукових досліджень»  
на тему «ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З  
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАВ:  
студент 2 курсу  
групи ІВ-91  
Бойко М. І.  
Залікова – 9102

ПЕРЕВІРИВ:  
ас. Регіда П. Г.

### Лабораторна робота №3

**Мета:** провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

**Завдання:**

1. Скласти матрицю планування для дробового трьохфакторного експерименту. Провести експеримент в усіх точках факторного простору, повторивши N експериментів, де N – кількість експериментів (рядків матриці планування) в усіх точках факторного простору – знайти значення функції відгуку Y. Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі (випадковим чином).

$$y_{\max} = 200 + x_{\text{ср max}};$$

$$y_{\min} = 200 + x_{\text{ср min}}$$

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

2. Знайти коефіцієнти лінійного рівняння регресії. Записати лінійне рівняння регресії.

3. Провести 3 статистичні перевірки.

4. Написати комп'ютерну програму, яка усе це виконує.

**Група:** ІВ-91

**Номер у списку:** 2

**Варіант – 102**

№ <sub>варіанта</sub>	X <sub>1</sub>		X <sub>2</sub>		X <sub>3</sub>	
	min	max	min	max	min	max
102	20	70	-20	40	70	80

## Роздруківка коду програми:

```
from random import *
from math import sqrt
from scipy import linalg
from scipy.stats import t, f

x_list = [[20, 70], [-20, 40], [70, 80]]

m = 3
N = 4
d = 4

mat_1X = [[1, -1, -1, -1],
           [1, -1, 1, 1],
           [1, 1, -1, 1],
           [1, 1, 1, -1]]

mat_X = [[x_list[0][0], x_list[1][0], x_list[2][0]],
          [x_list[0][1], x_list[1][1], x_list[2][1]],
          [x_list[0][2], x_list[1][2], x_list[2][2]],
          [x_list[0][3], x_list[1][3], x_list[2][3]]]

print("Матриця X:")
for i in range(len(mat_X)): print(mat_X[i])

tr_sx = [list(i) for i in zip(*mat_1X)]
tr_x = [list(i) for i in zip(*mat_X)]

x_minmax_av = [sum(x_list[i][k] for i in range(3)) / 3 for k in range(2)]
y_minmax = [int(200 + x_minmax_av[i]) for i in range(2)]

mat_Y = [[randint(y_minmax[0], y_minmax[1]) for _ in range(m)] for _ in
range(N)]

print("\nМатриця Y:")
for i in range(len(mat_Y)): print(mat_Y[i])

print("-"*65)
average_y = [sum(mat_Y[k1]) / m for k1 in range(N)]
print(f"\nСереднє y:\n{average_y}")

dispersion = [sum([(k1 - average_y[j]) ** 2) for k1 in mat_Y[j]]) / m for j
in range(N)]
print(f"\nДисперсія:\n{dispersion}")

mx = [sum(mat_X[i][k] for i in range(N)) / N for k in range(m)]
my = sum(average_y) / N

ai = [sum(tr_x[k][i] * average_y[i] for i in range(N)) / N for k in range(m)]
aii = [sum(tr_x[k][i] ** 2 for i in range(N)) / N for k in range(m)]

a12 = (tr_x[0][0] * tr_x[1][0] +
        tr_x[0][1] * tr_x[1][1] +
        tr_x[0][2] * tr_x[1][2] +
        tr_x[0][3] * tr_x[1][3]) / N

a13 = (tr_x[0][0] * tr_x[2][0] +
        tr_x[0][1] * tr_x[2][1] +
        tr_x[0][2] * tr_x[2][2] +
        tr_x[0][3] * tr_x[2][3]) / N

a23 = (tr_x[1][0] * tr_x[2][0] +
```

```

        tr_x[1][1] * tr_x[2][1] +
        tr_x[1][2] * tr_x[2][2] +
        tr_x[1][3] * tr_x[2][3]) / N

a32 = (tr_x[1][0] * tr_x[2][0] +
        tr_x[1][1] * tr_x[2][1] +
        tr_x[1][2] * tr_x[2][2] +
        tr_x[1][3] * tr_x[2][3]) / N

# Знайдемо коефіцієнти

zn = linalg.det([[1, mx[0], mx[1], mx[2]],
                 [mx[0], aii[0], a12, a13],
                 [mx[1], a12, aii[1], a32],
                 [mx[2], a13, a23, aii[2]]])

b0 = linalg.det([[my, mx[0], mx[1], mx[2]],
                 [ai[0], aii[0], a12, a13],
                 [ai[1], a12, aii[1], a32],
                 [ai[2], a13, a23, aii[2]]]) / zn

b1 = linalg.det([[1, my, mx[1], mx[2]],
                 [mx[0], ai[0], a12, a13],
                 [mx[1], ai[1], aii[1], a32],
                 [mx[2], ai[2], a23, aii[2]]]) / zn

b2 = linalg.det([[1, mx[0], my, mx[2]],
                 [mx[0], aii[0], ai[0], a13],
                 [mx[1], a12, ai[1], a32],
                 [mx[2], a13, ai[2], aii[2]]]) / zn

b3 = linalg.det([[1, mx[0], mx[1], my],
                 [mx[0], aii[0], a12, ai[0]],
                 [mx[1], a12, aii[1], ai[1]],
                 [mx[2], a13, a23, ai[2]]]) / zn

check = [b0 + b1 * tr_x[0][i] + b2 * tr_x[1][i] + b3 * tr_x[2][i] for i in
range(4)]

print(f"\nРівняння регресії:\ny = {b0} + {b1}*x1 + {b2}*x2 + {b3}*x3")
print(f"\nПорівняння з середнім y: {check}")
f1 = m - 1
f2 = N
f3 = f1 * f2
f4 = N - d

print("-"*65)
print('\nПеревіримо однорідність дисперсії за критерієм Кохрена')
if max(dispersion) / sum(dispersion) < 0.7679:
    print('Дисперсія однорідна:', max(dispersion) / sum(dispersion))
else:
    print('Дисперсія неоднорідна:', max(dispersion) / sum(dispersion))

print("-"*65)
print('\nПеревіримо на значимість за критерієм Стьюдента')
S2b = sum(dispersion) / N
S2bs = S2b / (m * N)
Sbs = sqrt(S2bs)
bb = [sum(average_y[k] * tr_sx[i][k] for k in range(N)) / N for i in
range(N)]
t_list = [abs(bb[i]) / Sbs for i in range(N)]
b_list = [b0, b1, b2, b3]
for i in range(N):
    if t_list[i] < t.ppf(q=0.975, df=f3):

```

```

        print('Незначний: ', b_list[i])
        b_list[i] = 0
        d -= 1
    else: print('Значний:   ', b_list[i])

y_reg = [b_list[0] + b_list[1] * mat_X[i][0] + b_list[2] * mat_X[i][1] +
b_list[3] * mat_X[i][2]
        for i in range(N)]
print("-"*65)
print('\nЗначення y')
for i in range(N):
    print(f"{b_list[0]} + {b_list[1]}*x1 + {b_list[2]}*x2 + {b_list[3]}*x3 ="
          f" {b_list[0] + b_list[1] * mat_X[i][0] + b_list[2] * mat_X[i][1] +
b_list[3] * mat_X[i][2]}")
print("-"*65)

print('\nПеревіримо адекватність моделі за критерієм Фішера')
Sad = (m / (N - d)) * int(sum(y_reg[i] - average_y[i] for i in range(N)) **
2)
Fp = Sad / S2b
q = 0.05
F_table = f.ppf(q=1-q, dfn=f4, dfd=f3)
print('Fp =', Fp)
if Fp > F_table:
    print('Модель неадекватна при 0.05')
else:
    print('Модель адекватна при 0.05')

```

## Приклад роботи програми:

Матриця X:

```

[20, -20, 70]
[20, 40, 80]
[70, -20, 80]
[70, 40, 70]

```

Матриця Y:

```

[257, 230, 228]
[236, 240, 251]
[253, 237, 228]
[228, 256, 227]

```

Середнє y:

```

[238.33333333333334, 242.33333333333334, 239.33333333333334, 237.0]

```

Дисперсія:

```

[174.88888888888889, 40.22222222222222, 106.88888888888887, 180.66666666666666]

```

Рівняння регресії:

```

y = 217.31111111109513 + -0.04333333333332803*x1 + 0.01388888888888742*x2 + 0.3166666666667567*x3

```

Порівняння з середнім y: [238.33333333332376, 242.33333333332465, 239.33333333332493, 236.99999999999068]

Перевіримо однорідність дисперсії за критерієм Кохрена  
Дисперсія однорідна: 0.3594164456233422

-----

Перевіримо на значимість за критерієм Стюдента  
Значний: 217.31111111109513  
Незначний: -0.04333333333332803  
Незначний: 0.01388888888888742  
Незначний: 0.3166666666667567

-----

Значення y  
 $217.31111111109513 + 0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 = 217.31111111109513$   
 $217.31111111109513 + 0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 = 217.31111111109513$   
 $217.31111111109513 + 0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 = 217.31111111109513$   
 $217.31111111109513 + 0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 = 217.31111111109513$

-----

Перевіримо адекватність моделі за критерієм Фішера  
FP = 61.28116710875332  
Модель адекватна при 0.05

Process finished with exit code 0

### Відповіді на контрольні запитання:

1. Дробовий факторний експеримент – це частина ПФЕ, яка використовується для мінімізації числа дослідів, за рахунок інформації, яка не є істотною для побудови моделі.
2. Розрахункове значення Кохрена потрібно для перевірки однорідності дисперсії.
3. Критерій Стюдента перевіряється для з'ясування значимості коефіцієнтів рівняння.
4. Критерій Фішера застосовують при перевірці отриманого рівняння регресії.