

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

Методи наукових досліджень  
Лабораторна робота № 3  
**«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З  
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»**

Виконав:  
студент групи ІВ-93  
Манчук М.В.

Київ  
2021 р.

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

### Завдання на лабораторну роботу

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.

315	10	50	25	65	50	65
-----	----	----	----	----	----	----

### Код програми:

```
from math import pow, sqrt
import os
import sys
from random import randint
import numpy as np
from prettytable import PrettyTable
from scipy.stats import t

n = 4
m = 3
N = [i + 1 for i in range(n + 1)]
x_min = [10, 25, 50]
x_max = [50, 65, 65]
averageX_min = round(np.average(x_min)) # Середнє мінімальне значення
averageX_max = round(np.average(x_max)) # Середнє максимальне значення

y_min = 200 + averageX_min
y_max = 200 + averageX_max

y = [[], [], [], []]
y_1 = [randint(y_min, y_max) for _ in range(n)]
y_2 = [randint(y_min, y_max) for _ in range(n)]
y_3 = [randint(y_min, y_max) for _ in range(n)]
y[0] = [y_1[0], y_2[0], y_3[0]]
y[1] = [y_1[1], y_2[1], y_3[1]]
y[2] = [y_1[2], y_2[2], y_3[2]]
y[3] = [y_1[3], y_2[3], y_3[3]]

x_0 = [1, 1, 1, 1]
```

```

x_1 = [-1, -1, 1, 1]
x_2 = [-1, 1, -1, 1]
x_3 = [-1, 1, 1, -1]
x1_m = [10, 10, 50, 50]
x2_m = [25, 65, 25, 65]
x3_m = [50, 65, 65, 50]

average_y = [round(sum(i) / len(i), 2) for i in y]

mx_1 = np.average(x1_m)
mx_2 = np.average(x2_m)
mx_3 = np.average(x3_m)
my = np.average(average_y)

a_1 = sum([x1_m[i] * average_y[i] for i in range(n)]) / n
a_2 = sum([x2_m[i] * average_y[i] for i in range(n)]) / n
a_3 = sum([x3_m[i] * average_y[i] for i in range(n)]) / n
a_12 = sum([x1_m[i] * x2_m[i] for i in range(n)]) / n
a_13 = sum([x1_m[i] * x3_m[i] for i in range(n)]) / n
a_23 = sum([x2_m[i] * x3_m[i] for i in range(n)]) / n

a_11 = sum([pow(i, 2) for i in x1_m]) / n
a_22 = sum([pow(i, 2) for i in x2_m]) / n
a_33 = sum([pow(i, 2) for i in x3_m]) / n
a_32, a_31, a_21 = a_23, a_13, a_12

def determinant3(a11, a12, a13, a21, a22, a23, a31, a32, a33):
    determinant = a11 * a22 * a33 + a12 * a23 * a31 + a32 * a21 * \
        a13 - a13 * a22 * a31 - a32 * a23 * a11 - a12 * a21 * a33
    return determinant

def determinant4(a11, a12, a13, a14, a21, a22, a23, a24, a31, a32, a33, a34, a41, a42, a43,
a44):
    determinant = a11 * determinant3(a22, a23, a24, a32, a33, a34, a42, a43, a44) - \
        a12 * determinant3(a21, a23, a24, a31, a33, a34, a41, a43, a44) - \
        a13 * determinant3(a22, a21, a24, a32, a31, a34, a42, a41, a44) - \
        a14 * determinant3(a22, a23, a21, a32, a33, a31, a42, a43, a41)
    return determinant

B0 = determinant4(1, mx_1, mx_2, mx_3,
    mx_1, a_11, a_12, a_13,
    mx_2, a_12, a_22, a_23,
    mx_3, a_13, a_23, a_33)

B1 = determinant4(my, mx_1, mx_2, mx_3,
    a_1, a_11, a_12, a_13,
    a_2, a_12, a_22, a_23,
    a_3, a_13, a_23, a_33)

B2 = determinant4(1, my, mx_2, mx_3,
    mx_1, a_1, a_12, a_13,
    mx_2, a_2, a_22, a_23,
    mx_3, a_3, a_23, a_33)

B3 = determinant4(1, mx_1, my, mx_3,
    mx_1, a_11, a_1, a_13,
    mx_2, a_12, a_2, a_23,
    mx_3, a_13, a_3, a_33)

B4 = determinant4(1, mx_1, mx_2, my,
    mx_1, a_11, a_12, a_1,
    mx_2, a_12, a_22, a_2,
    mx_3, a_13, a_23, a_3)

b_0 = B1 / B0
b_1 = B2 / B0

```

```

b_2 = B3 / B0
b_3 = B4 / B0
b = [b_0, b_1, b_2, b_3]

y_r = "y = " + str(round(b[0], 3)) + " + " + str(round(b[1], 3)) + "*x1" + " + " + str(
    round(b[2], 3)) + "*x2" + " + " + str(round(b[3], 3)) + "*x3"

y_pr1 = b[0] + b[1] * x1_m[0] + b[2] * x2_m[0] + b[3] * x3_m[0]
y_pr2 = b[0] + b[1] * x1_m[1] + b[2] * x2_m[1] + b[3] * x3_m[1]
y_pr3 = b[0] + b[1] * x1_m[2] + b[2] * x2_m[2] + b[3] * x3_m[2]
y_pr4 = b[0] + b[1] * x1_m[3] + b[2] * x2_m[3] + b[3] * x3_m[3]
y_pr = [y_pr1, y_pr2, y_pr3, y_pr4]
for i in range(3):
    if round(average_y[i], 5) == round(y_pr[i], 5):
        check_1 = "Отримані значення збігаються " \
            "з середніми значеннями функції відгуку за рядками"
    else:
        check_1 = "Отримані значення НЕ збігаються " \
            "з середніми значеннями функції відгуку за рядками"

S_1 = sum([pow((y[0][i] - average_y[i]), 2) for i in range(m)]) / m
S_2 = sum([pow((y[1][i] - average_y[i]), 2) for i in range(m)]) / m
S_3 = sum([pow((y[2][i] - average_y[i]), 2) for i in range(m)]) / m
S_4 = sum([pow((y[3][i] - average_y[i]), 2) for i in range(m)]) / m
S = [S_1, S_2, S_3, S_4]

Gp = max(S) / sum(S)

Gt = 0.7679
if Gp < Gt:
    check_2 = "Дисперсія однорідна з вірогідністю 95%"
else:
    print('Помилка, повторіть експеримент заново!!!')
    os.execl(sys.executable, sys.executable, *sys.argv)

s_beta = sqrt(sum(S) / (n * m * m))
s2_b = sum(S) / n

t_1 = abs(sum([average_y[i] * x_0[i] for i in range(n)])) / s_beta
t_2 = abs(sum([average_y[i] * x_1[i] for i in range(n)])) / s_beta
t_3 = abs(sum([average_y[i] * x_2[i] for i in range(n)])) / s_beta
t_4 = abs(sum([average_y[i] * x_3[i] for i in range(n)])) / s_beta
T = [t_1, t_2, t_3, t_4]
T_table = t.ppf(q=0.975, df=9)

k = 0
for i in range(n):
    if T[i] < T_table:
        b[i] = 0
        k += 1

if k != 0:
    index_list = [str(i + 1) for i, x in enumerate(b) if x == 0]
    index_list = ["b" + i for i in index_list]
    deleted_koef = ', '.join(
        index_list) + " - коефіцієнти рівняння регресії приймаємо " \
        "незначними при рівні значимості 0.05, " \
        "тобто вони виключаються з рівняння. "
else:
    deleted_koef = "Всі b значимі коефіцієнти " \
        "і вони залишаються в рівнянні регресії."

ys1 = b[0] + b[1] * x1_m[0] + b[2] * x2_m[0] + b[3] * x3_m[0]
ys2 = b[0] + b[1] * x1_m[1] + b[2] * x2_m[1] + b[3] * x3_m[1]
ys3 = b[0] + b[1] * x1_m[2] + b[2] * x2_m[2] + b[3] * x3_m[2]
ys4 = b[0] + b[1] * x1_m[3] + b[2] * x2_m[3] + b[3] * x3_m[3]

y_student = [ys1, ys2, ys3, ys4]

```

```

d = n - k
f4 = n - d
F = m * sum([(average_y[i] - y_student[i]) ** 2 for i in range(n)]) / (n - d)
Fp = F / (sum(S) / n)
Fisher_table = [5.3, 4.5, 4.1, 3.8]

if Fp < Fisher_table[f4]:
    check_3 = "Рівняння регресії адекватне при рівні значимості 5%"
else:
    check_3 = "Рівняння регресії неадекватне при рівні значимості 5%"

print("\nРівняння регресії: y = b0 + b1*x1 + b2*x2+ b3*x3\n")
th = ["N", "X1", "X2", "X3", "Y1", "Y2", "Y3"]
columns = len(th)
rows = len(x_1)
table = PrettyTable(th)
table.title = "Натуралізована матриця планування експерименту"
for i in range(rows):
    td = [N[i], x1_m[i], x2_m[i], x3_m[i], y_1[i], y_2[i], y_3[i]]
    td_data = td[:]
    while td_data:
        table.add_row(td_data[:columns])
        td_data = td_data[columns:]
print(table)

print("\nСередній Y:\n", round(average_y[0], 3), "\n", round(average_y[1], 3),
      "\n", round(average_y[2], 3), "\n", round(average_y[3], 3))
print("\nОтримане рівняння регресії:", y_r)
print("Практичний Y:\n", round(y_pr[0], 3), "\n", round(y_pr[1], 3),
      "\n", round(y_pr[2], 3), "\n", round(y_pr[3], 3))
print(check_1)

print("")
th = ["N", "X0", "X1", "X2", "X3", "Y1", "Y2", "Y3"]
columns = len(th)
rows = len(x_1)
table = PrettyTable(th)
table.title = "Нормована матриця планування експерименту."
for i in range(rows):
    td = [N[i], x_0[i], x_1[i], x_2[i], x_3[i], y_1[i], y_2[i], y_3[i]]
    td_data = td[:]
    while td_data:
        table.add_row(td_data[:columns])
        td_data = td_data[columns:]
print(table)

print("\nДисперсії:\n d1 =", round(S[0], 3), "\n d2 =", round(S[1], 3),
      "\n d3 =", round(S[2], 3), "\n d4 =", round(S[3], 3))
print("Критерій Кохрена: Gr = " + str(round(Gp, 3)))
print(check_2)

print("\nКритерій Стюдента:\n t1 =", round(T[0], 3), "\n t2 =", round(T[1], 3),
      "\n t3 =", round(T[2], 3), "\n t4 =", round(T[3], 3))
print(deleted_koef)
print(" y1 =", round(y_student[0], 3), "\n y2 =", round(y_student[1], 3),
      "\n y3 =", round(y_student[2], 3), "\n y4 =", round(y_student[3], 3))

print("\nКритерій Фішера: Fp =", round(Fp, 3))
print(check_3)

```

## Результат виконання програми:

```
C:\Anaconda3\python.exe C:/PythonProjects/MND/Lab3.py
```

Рівняння регресії:  $y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3$

```
+-----+
|  Натуралізована матриця планування експерименту  |
+-----+-----+-----+-----+-----+-----+
|  N  |  X1  |  X2  |  X3  |  Y1  |  Y2  |  Y3  |
+-----+-----+-----+-----+-----+-----+
|  1  |  10  |  25  |  50  |  250 |  256 |  236 |
|  2  |  10  |  65  |  65  |  246 |  248 |  228 |
|  3  |  50  |  25  |  65  |  249 |  232 |  250 |
|  4  |  50  |  65  |  50  |  230 |  254 |  230 |
+-----+-----+-----+-----+-----+-----+
```

Середній Y:

247.33  
240.67  
243.67  
238.0

Отримане рівняння регресії:  $y = 253.624 + -0.079 \cdot x_1 + -0.154 \cdot x_2 + -0.033 \cdot x_3$

Практичний Y:

247.33  
240.67  
243.67  
238.0

Отримані значення збігаються з середніми значеннями функції відгуку за рядками

```
+-----+
|  Нормована матриця планування експерименту.  |
+-----+-----+-----+-----+-----+-----+
|  N  |  X0  |  X1  |  X2  |  X3  |  Y1  |  Y2  |  Y3  |
+-----+-----+-----+-----+-----+-----+
|  1  |  1  |  -1  |  -1  |  -1  |  250 |  256 |  236 |
|  2  |  1  |  -1  |  1  |  1  |  246 |  248 |  228 |
|  3  |  1  |  1  |  -1  |  1  |  249 |  232 |  250 |
|  4  |  1  |  1  |  1  |  -1  |  230 |  254 |  230 |
+-----+-----+-----+-----+-----+-----+
```

Дисперсії:

d1 = 100.322  
d2 = 100.349  
d3 = 39.342  
d4 = 221.629

Критерій Кохрена:  $G_r = 0.48$

Дисперсія однорідна з вірогідністю 95%

Критерій Стюдента:

t1 = 270.784  
t2 = 1.768  
t3 = 3.443  
t4 = 0.276

b2, b4 - коефіцієнти рівняння регресії приймаємо незначними при рівні значимості 0.05, тобто вони виключаються з рівняння.

y1 = 249.771  
y2 = 243.606  
y3 = 249.771  
y4 = 243.606

Критерій Фішера:  $F_p = 1.082$

Рівняння регресії адекватне при рівні значимості 5%

Process finished with exit code 0