Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №2**

**«Проведення трьохфакторного експеременту з використанням лінійного рівняння регресії»**

**Виконав:**

студент II курсу ФІОТ

групи ІВ-91

Бурбело Сергій

**Перевірив:**

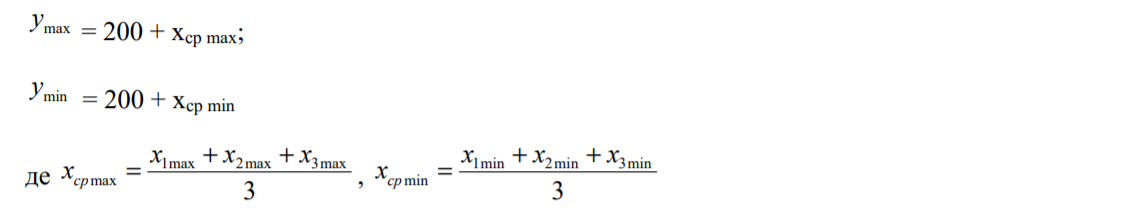
Регіда П.Г.

Київ – 2021

**Мета роботи:** 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 У. Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі (випадковим чином).



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

3. Провести 3 статистичні перевірки.

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

**Варіант завдання:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Варіант | X1 | | X2 | | X3 | |
| min | max | min | max | min | max |
| 104 | 15 | 45 | -25 | 10 | 45 | 50 |

Довірча ймовірність дорівнює 0.95, а рівень значимості q = 0.05.

**Роздруківка тексту програми:**

from prettytable import PrettyTable

from random import randint

import numpy as np

import math, os, sys

n = 4

m = 3

N = [i+1 for i in range(n+1)]

x\_min = [15, -25, 45]

x\_max = [45, 10, 50]

average\_x\_min = round(np.average(x\_min))

average\_x\_max = round(np.average(x\_max))

y\_min = 200 + average\_x\_min

y\_max = 200 + average\_x\_max

y = [[],[],[],[]]

y1 = [randint(y\_min, y\_max) for j in range(n)]

y2 = [randint(y\_min, y\_max) for j in range(n)]

y3 = [randint(y\_min, y\_max) for j in range(n)]

y[0] = [y1[0], y2[0], y3[0]]

y[1] = [y1[1], y2[1], y3[1]]

y[2] = [y1[2], y2[2], y3[2]]

y[3] = [y1[3], y2[3], y3[3]]

x0 = [1, 1, 1, 1]

x1 = [-1, -1, 1, 1]

x2 = [-1, 1, -1, 1]

x3 = [-1, 1, 1, -1]

x1\_m = [15, 15, 45, 45]

x2\_m = [-25, 10, -25, 10]

x3\_m = [45, 50, 50, 45]

# Cереднє значення функції відгуку в рядку:

av\_y = [round(sum(i) / len(i), 2) for i in y]

# Обчислюємо мат. очікування:

mx1 = np.average(x1\_m)

mx2 = np.average(x2\_m)

mx3 = np.average(x3\_m)

my = np.average(av\_y)

a1 = sum([x1\_m[i]\*av\_y[i] for i in range(n)])/n

a2 = sum([x2\_m[i]\*av\_y[i] for i in range(n)])/n

a3 = sum([x3\_m[i]\*av\_y[i] for i in range(n)])/n

a12 = sum([x1\_m[i]\*x2\_m[i] for i in range(n)])/n

a13 = sum([x1\_m[i]\*x3\_m[i] for i in range(n)])/n

a23 = sum([x2\_m[i]\*x3\_m[i] for i in range(n)])/n

a11 = sum([math.pow(i,2) for i in x1\_m])/n

a22 = sum([math.pow(i,2) for i in x2\_m])/n

a33 = sum([math.pow(i,2) for i in x3\_m])/n

a32, a31, a21 = a23, a13, a12

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, mx1, mx2, mx3,

                    mx1, a11, a12, a13,

                    mx2, a12, a22, a23,

                    mx3, a13, a23, a33)

B1 = determinant4(my, mx1, mx2, mx3,

                    a1, a11, a12, a13,

                    a2, a12, a22, a23,

                    a3, a13, a23, a33)

B2 = determinant4(1, my, mx2, mx3,

                    mx1, a1, a12, a13,

                    mx2, a2, a22, a23,

                    mx3, a3, a23, a33)

B3 = determinant4(1, mx1, my, mx3,

                    mx1, a11, a1, a13,

                    mx2, a12, a2, a23,

                    mx3, a13, a3, a33)

B4 = determinant4(1, mx1, mx2, my,

                    mx1, a11, a12, a1,

                    mx2, a12, a22, a2,

                    mx3, a13, a23, a3)

b0 = B1/B0

b1 = B2/B0

b2 = B3/B0

b3 = B4/B0

b = [b0, b1, b2, b3]

# Запишемо отримане рівняння регресії:

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

# Зробимо перевірку (підставимо значення факторів з матриці планування і порівняємо результат

# з середніми значеннями функції відгуку за рядками):

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(av\_y[i],5) == round(y\_pr[i],5):

        check1 = "Отримані значення збігаються з середніми значеннями функції відгуку за рядками"

    else:

        check1 = "Отримані значення НЕ збігаються з середніми значеннями функції відгуку за рядками"

# -------------------------------------------------------

# Перевірка однорідності дисперсії за критерієм Кохрена:

# -------------------------------------------------------

# Знайдемо дисперсії по рядках:

S1 = sum([math.pow((y[0][i] - av\_y[i]),2) for i in range(m)])/m

S2 = sum([math.pow((y[1][i] - av\_y[i]),2) for i in range(m)])/m

S3 = sum([math.pow((y[2][i] - av\_y[i]),2) for i in range(m)])/m

S4 = sum([math.pow((y[3][i] - av\_y[i]),2) for i in range(m)])/m

S = [S1, S2, S3, S4]

Gp = max(S)/sum(S)

# Рівень значимості приймемо 0.05.

Gt = 0.7679

if Gp < Gt:

    check2 = "Дисперсія однорідна з вірогідностю 95%"

else:

    # Виводжу повідомлення та просто перезапускаю програму.

    print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')

    print('Помилка, повторюємо експеремент заново.')

    print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')

    os.execl(sys.executable, sys.executable, \*sys.argv)

# ---------------------------------------------------------

# Перевірка однорідності дисперсії за критерієм Стьюдента:

# ---------------------------------------------------------

s\_beta = math.sqrt(sum(S)/(n\*m\*m))

s2\_b = sum(S)/n

t1 = abs(sum(([av\_y[i]\*x0[i] for i in range(n)])))/(s\_beta)

t2 = abs(sum(([av\_y[i]\*x1[i] for i in range(n)])))/(s\_beta)

t3 = abs(sum(([av\_y[i]\*x2[i] for i in range(n)])))/(s\_beta)

t4 = abs(sum(([av\_y[i]\*x3[i] for i in range(n)])))/(s\_beta)

t = [t1, t2, t3, t4]

T = 2.306

k = 0

for i in range(n):

    if t[i] < T:

        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]

# -------------------------------------------------------

# Перевірка однорідності дисперсії за критерієм Фішера:

# -------------------------------------------------------

# Кількість значимих коефіцієнтів:

d = n - k

f4 = n - d # f3 = 8

F = m\*sum([(av\_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]):

    check3 = "Рівняння регресії адекватне при рівні значимості 5%"

else:

    check3 = "Рівняння регресії неадекватне при рівні значимості 5%"

# -------------------------------------------------------

# Вивід даних:

# -------------------------------------------------------

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(x1)

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], y1[i], y2[i], y3[i]]

    td\_data = td[:]

    while td\_data:

        table.add\_row(td\_data[:columns])

        td\_data = td\_data[columns:]

print(table)

print("\nCередній Y:\n", round(av\_y[0],3), "\n", round(av\_y[1],3), \

    "\n", round(av\_y[2],3), "\n", round(av\_y[3],3))

print("\nОтримане рівняння регресії:", yr)

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(check1)

print("")

th = ["N", "X0", "X1", "X2", "X3", "Y1", "Y2", "Y3"]

columns = len(th)

rows = len(x1)

table = PrettyTable(th)

table.title = "Нормована матриця планування експерименту."

for i in range(rows):

    td = [N[i], x0[i], x1[i], x2[i], x3[i], y1[i], y2[i], y3[i]]

    td\_data = td[:]

    while td\_data:

        table.add\_row(td\_data[:columns])

        td\_data = td\_data[columns:]

print(table)

# Критерій Кохрена:

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(check2)

# Критерій Стьюдента:

print("\nКритерій Стьюдента:\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), \

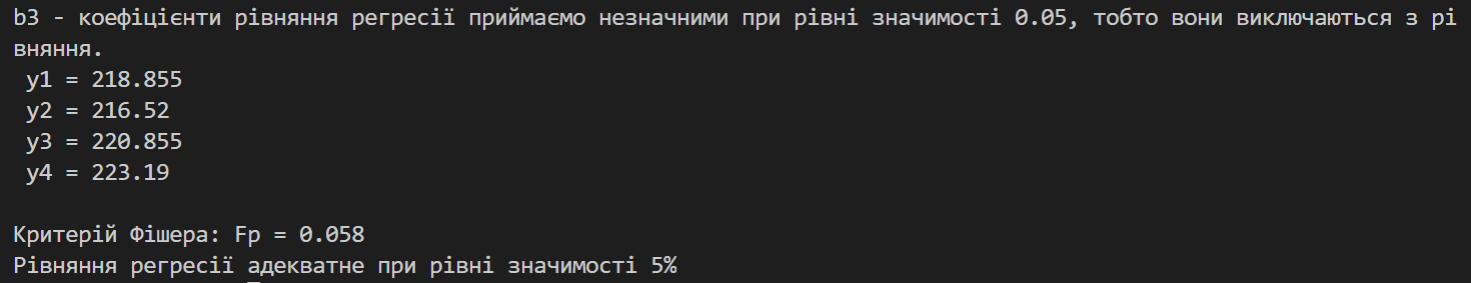
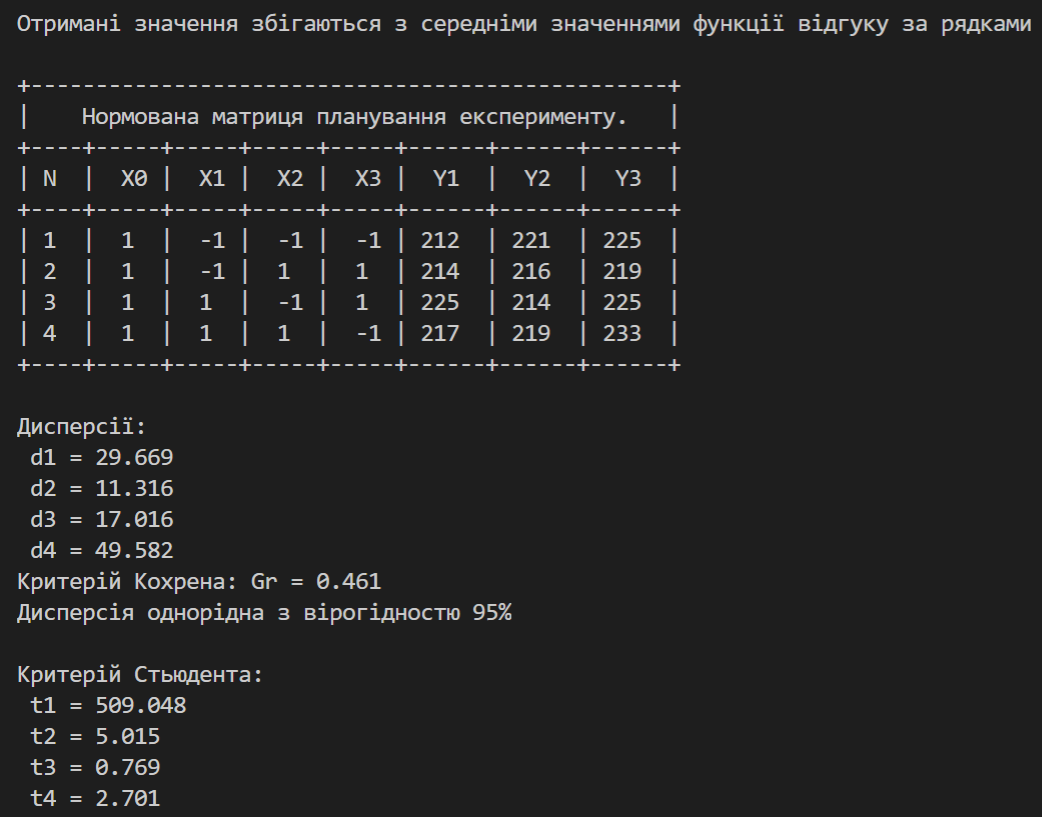
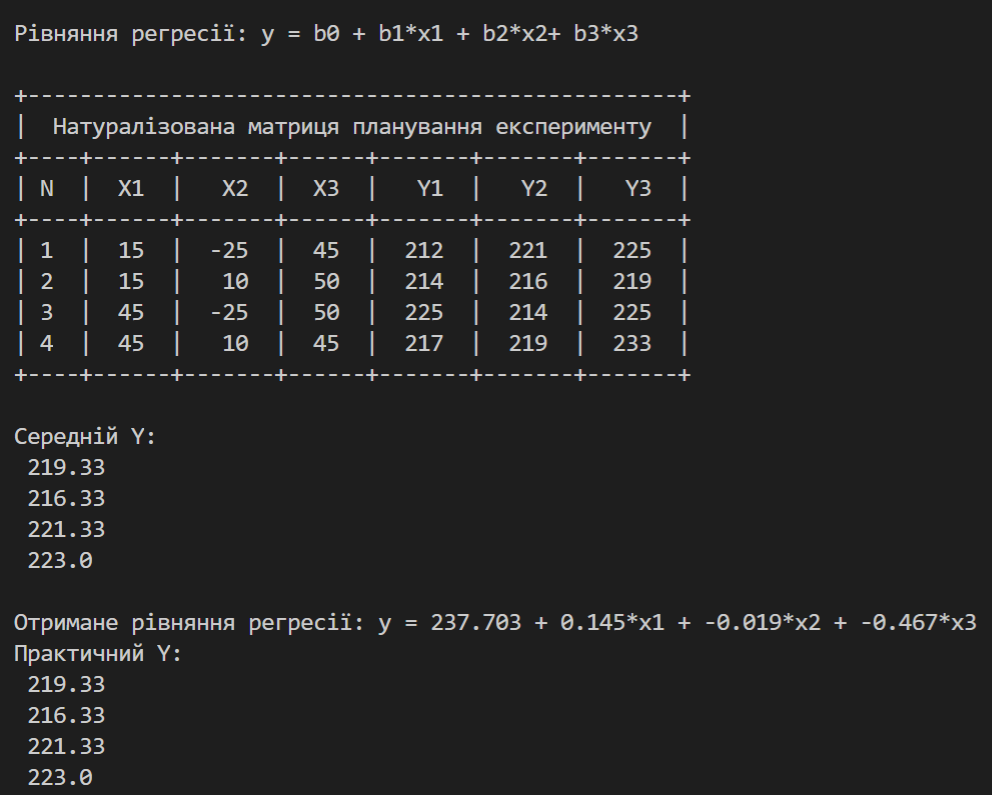
    "\n y3 =", round(y\_student[2],3), "\n y4 =", round(y\_student[3],3))

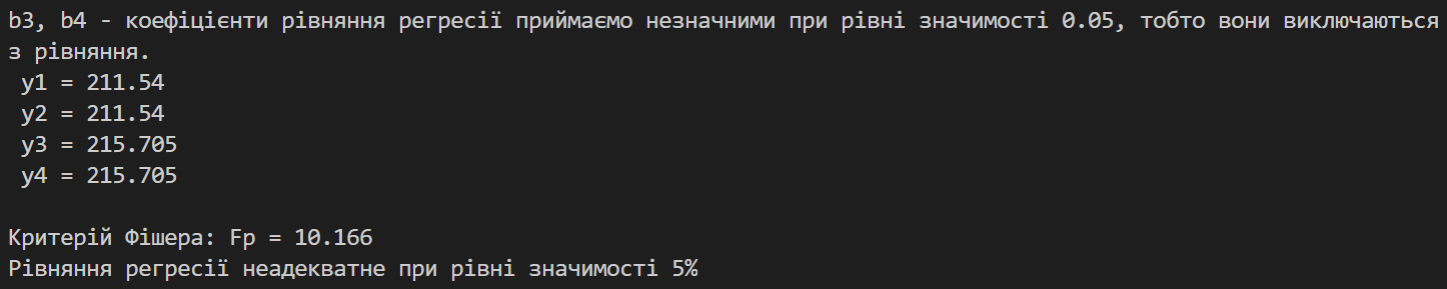
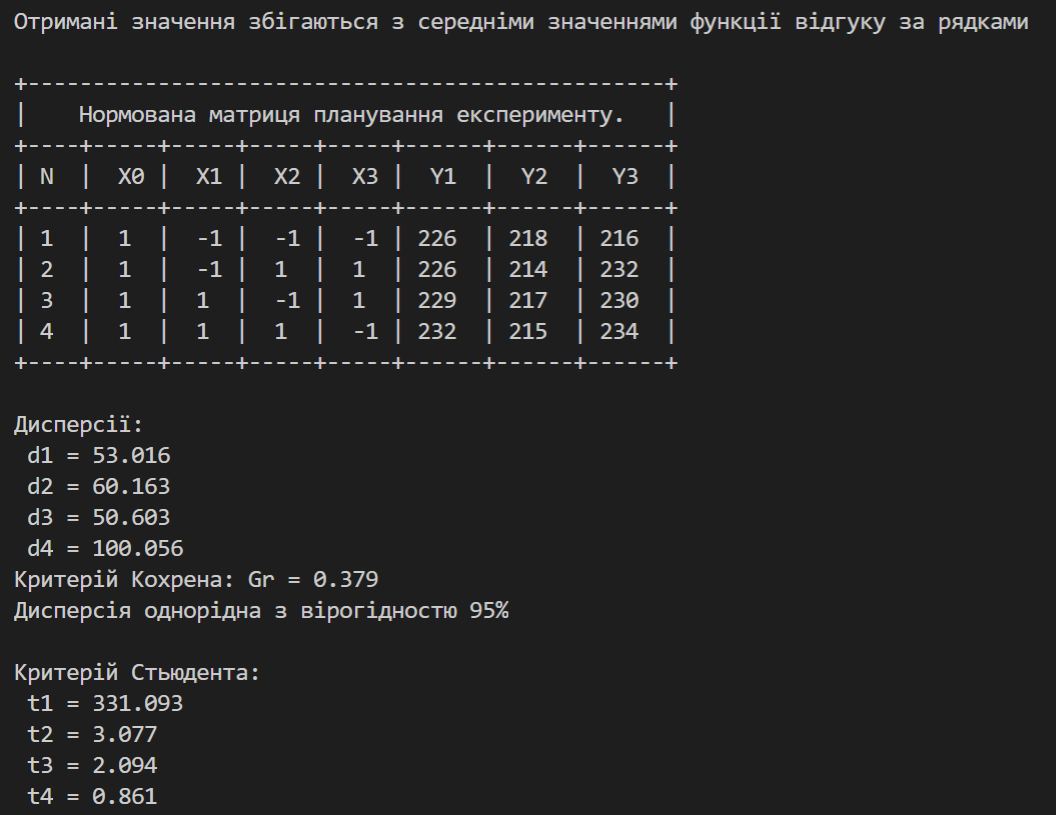
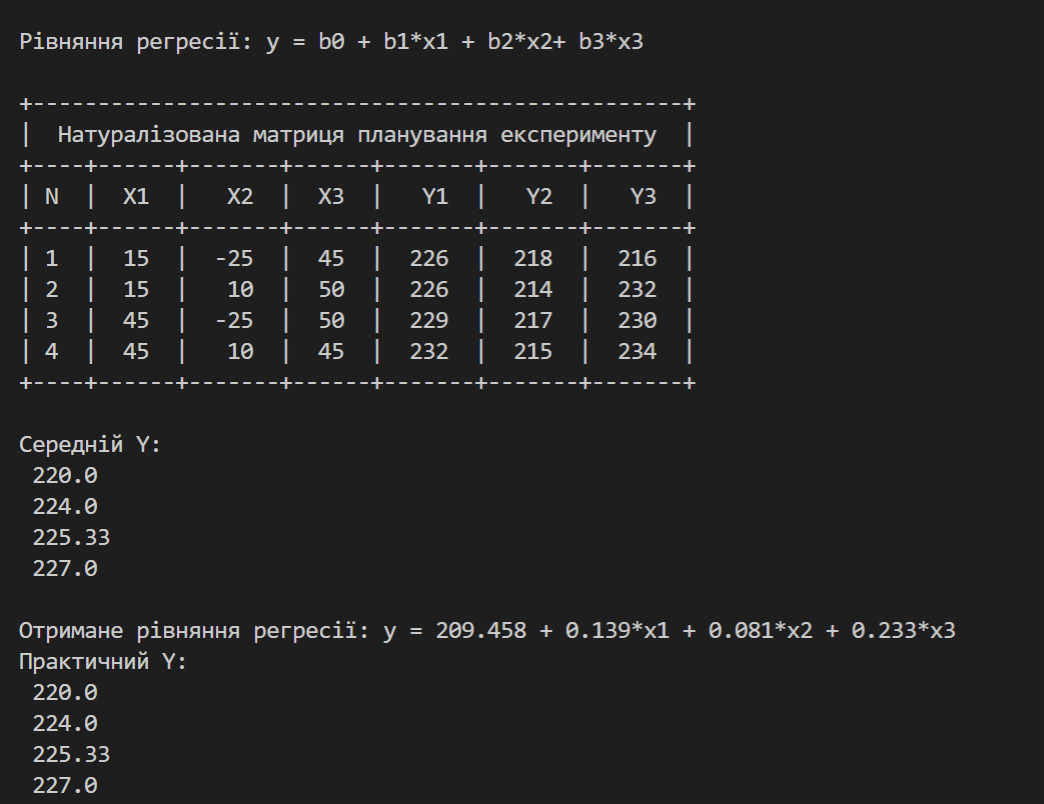
# Критерій Фішера:

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

print(check3)

**Результати роботи програми:**





**Висновок:** У ході лабораторної роботи я змоделював трьохфакторний експеримент з використанням лінійного рівняння регресії, склав матрицю планування експерименту, визначив коефіцієнти рівняння регресії, натуралізовані та нормовані, виконав перевірку правильності розрахунку коефіцієнтів рівняння регресії. Також я реалізував 3 статистичні перевіркиза критерієм Кохрена, Стьюдента та Фішера.

**Контрольні питання**

1. **Що називається дробовим факторним експериментом?**

У деяких випадках немає необхідності проводити повний факторний експеримент (ПФЕ). Якщо буде використовуватися лінійна регресія, то можливо зменшити кількість рядків матриці ПФЕ до кількості коефіцієнтів регресійної моделі. Кількість дослідів слід скоротити, використовуючи для планування так звані регулярні дробові репліки від повного факторного експерименту, що містять відповідну кількість дослідів і зберігають основні властивості матриці планування – це означає дробовий факторний експеримент (ДФЕ).

1. **Для чого потрібно розрахункове значення Кохрена?**

Статистична перевірка за критерієм Кохрена використовується для перевірки гіпотези про однорідність дисперсії з довірчою ймовірністю p. Якщо експериментальне значення G < Gкр, яке обирається з таблиці, то гіпотеза підтверджується, якщо ні, то відповідно не підтверджується.

1. **Для чого перевіряється критерій Стьюдента?**

Критерій Стьюдента використовується для перевірки значимості коефіцієнта рівняння регресії. Якщо з’ясувалось, що будь-який коефіцієнт рівняння регресії не значимий, то відповідний bi = 0 і відповідний член рівняння регресії треба викреслити. Іноді ця статистична перевірка має назву «нуль-гіпотеза». Якщо експериментальне значення t > tкр, тонуль-гіпотеза не підтверджується і даний коефіцієнт значимий, інакше нуль-гіпотеза підтверджується і даний коефіцієнт рівняння регресії не значимий.

1. **Чим визначається критерій Фішера і як його застосовувати?**

Критерій Фішера застосовується для перевірки адекватності моделі (рівняння регресії) оригіналу (експериментальним даним). Обчислюється експериментальне значення F, яке порівнюється з Fкр, взятим з таблиці залежно від кількості значимих коефіцієнтів та ступенів вільності. Якщо F < Fкр, то модель адекватна оригіналу, інакше – ні.