

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  
Степанюк Р. В.  
Залікова – 9127

ПЕРЕВІРИВ:  
ас. Регіда П. Г.

**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).

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

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

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

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

3. Провести 3 статистичні перевірки.

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

№ варіанту	x <sub>1</sub>		x <sub>2</sub>		x <sub>3</sub>	
	min	max	min	max	min	max
125	-25	-5	25	45	25	30

### Програмний код

```
# Методи наукових досліджень
#
# Степанюк Роман Вікторович ІВ-91 ФІОТ
#
# Варіант 225

import random, math, numpy as np

def average(list):
    average = 0
    for element in list:
        average += element / len(list)
    return average

def dispersion(list):
    list_average = average(list)
```

```

dispersion = 0
for element in List:
    dispersion += (element - list_average)**2 / len(List)
return dispersion

def cochrane_criteria():
    global m, N
    # 7.1. Знайдемо дисперсії по рядках
    dispersion_list = [dispersion(y1), dispersion(y2), dispersion(y3), dispersion(y4)]
    print(f"Дисперсії по рядках:\n{dispersion_list}")
    gp_denominator = 0
    for disp in dispersion_list:
        gp_denominator += disp
    gp = max(dispersion_list) / gp_denominator
    # print(f"Gp = {round(gp, 3)}")

    # 7.2. Рівень значимості приймемо за 0.05
    f1 = m - 1
    f2 = N
    gt = 0.7679

    if gp < gt: return True
    else: return False

def students_criteria():
    global m, N
    sb = average(dispersion_list)
    # print(f"\nSβ = {sb}")
    s_beta_2 = sb / (N * m)
    # print("Sβ = ", s_beta_2)

    s_beta = math.sqrt(s_beta_2)
    # print("Sβ = ", s_beta)

    beta0 = 0
    for i in range(N):
        beta0 += average_list[i] * x0[i] / N
    beta1 = 0
    for i in range(N):
        beta1 += average_list[i] * x1[i] / N
    beta2 = 0
    for i in range(N):
        beta2 += average_list[i] * x2[i] / N
    beta3 = 0
    for i in range(N):
        beta3 += average_list[i] * x3[i] / N
    # print(f"β = {beta0}\nβ = {beta1}\nβ = {beta2}\nβ = {beta3}")

    t0 = abs(beta0) / s_beta
    t1 = abs(beta1) / s_beta
    t2 = abs(beta2) / s_beta

```

```

t3 = abs(beta3) / s_beta

# print(f"\nt0 = {t0}\nt1 = {t1}\nt2 = {t2}\nt3 = {t3}")

f3 = (m - 1) * N
tt = 2.306

student_check = {}
if (t0 > tt): student_check[0] = b[0]
else: student_check[0] = 0
if (t1 > tt): student_check[1] = b[1]
else: student_check[1] = 0
if (t2 > tt): student_check[2] = b[2]
else: student_check[2] = 0
if (t3 > tt): student_check[3] = b[3]
else: student_check[3] = 0

return student_check

def fisher_criteria():
    global m, N
    d = 0
    for key in std_ch:
        if std_ch[key] != 0: d += 1

    f1 = m - 1
    f2 = N
    f3 = (m - 1) * N
    f4 = N - d

    s2_ad = 0
    for i in range(N):
        s2_ad += (y_std[i] - average_list[i])**2
    s2_ad *= m / f4
    # print(s2_ad)

    s2_b = average(dispersion_list)

    fp = s2_ad / s2_b
    # print(f"Fp = {fp}")

    ft = 4.5

    if (fp > ft): print("Fp > Ft\nОтже, рівняння регресії неадекватно оригіналу при рівні значимості 0.05"
)
    else: print("Fp < Ft\nРівняння регресії адекватно оригіналу")

x_min = [-25, 25, 25]
x_max = [-5, 45, 30]
y_min = 200 + average(x_min)
y_max = 200 + average(x_max)

```

```

x0 = [1, 1, 1, 1]
x1 = [-1, -1, 1, 1]
x2 = [-1, 1, -1, 1]
x3 = [-1, 1, 1, -1]

print(f"Мінімальне значення ф-ції відгуку: {int(y_min)}")
print(f"Максимальне значення ф-ції відгуку: {int(y_max)}")

# 1. Матриця планування експерименту
# -1 -1 -1
# -1 +1 +1
# +1 -1 +1
# +1 +1 -1

# 2. Заповнимо матрицю планування для m = 3
plan_matrix = []
plan_matrix.append([x_max[0], x_min[1], x_min[2]])
plan_matrix.append([x_min[0], x_max[1], x_max[2]])
plan_matrix.append([x_max[0], x_min[1], x_max[2]])
plan_matrix.append([x_max[0], x_max[1], x_min[2]])

print(f"\nМатриця експерименту:")
for line in plan_matrix:
    print(line)

m = 3
N = 4

y1 = [random.randint(int(y_min), int(y_max)) for _ in range(m)]
y2 = [random.randint(int(y_min), int(y_max)) for _ in range(m)]
y3 = [random.randint(int(y_min), int(y_max)) for _ in range(m)]
y4 = [random.randint(int(y_min), int(y_max)) for _ in range(m)]

print(f"\nФункції відгуку\n{y1}\n{y2}\n{y3}\n{y4}")

dispersion_list = [dispersion(y1), dispersion(y2), dispersion(y3), dispersion(y4)]

# Рівняння регресії
# y = b0 + b1*x1 + b2*x2 + b3*x3

# 3. Знайдемо середні значення функцій відгуку за рядками
y1_average = average(y1)
y2_average = average(y2)
y3_average = average(y3)
y4_average = average(y4)
print(f"\nСередні значення ф-цій відгуку\n\u2081: {y1_average}\n\u2082: {y2_average}\n\u2083: {y3_average}\n\u2084: {y4_average}")
average_list = [y1_average, y2_average, y3_average, y4_average]

# 4. Знайдемо коефіцієнти рівняння регресії
mx1 = (plan_matrix[0][0] + plan_matrix[1][0] + plan_matrix[2][0] + plan_matrix[3][0]) / 4
mx2 = (plan_matrix[0][1] + plan_matrix[1][1] + plan_matrix[2][1] + plan_matrix[3][1]) / 4

```

```

mx3 = (plan_matrix[0][2] + plan_matrix[1][2] + plan_matrix[2][2] + plan_matrix[3][2]) / 4

my = average(average_list)

a1 = (plan_matrix[0][0] * y1_average + plan_matrix[1][0] * y2_average + plan_matrix[2][0] * y3_average + p
lan_matrix[3][0] * y4_average) / 4
a2 = (plan_matrix[0][1] * y1_average + plan_matrix[1][1] * y2_average + plan_matrix[2][1] * y3_average + p
lan_matrix[3][1] * y4_average) / 4
a3 = (plan_matrix[0][2] * y1_average + plan_matrix[1][2] * y2_average + plan_matrix[2][2] * y3_average + p
lan_matrix[3][2] * y4_average) / 4

a11 = (plan_matrix[0][0]**2 + plan_matrix[1][0]**2 + plan_matrix[2][0]**2 + plan_matrix[3][0]**2) / 4
a22 = (plan_matrix[0][1]**2 + plan_matrix[1][1]**2 + plan_matrix[2][1]**2 + plan_matrix[3][1]**2) / 4
a33 = (plan_matrix[0][2]**2 + plan_matrix[1][2]**2 + plan_matrix[2][2]**2 + plan_matrix[3][2]**2) / 4

a12 = (plan_matrix[0][0]*plan_matrix[0][1] + plan_matrix[1][0]*plan_matrix[1][1] + plan_matrix[2][0]*plan_
matrix[2][1] + plan_matrix[3][0]*plan_matrix[3][1]) / 4
a13 = (plan_matrix[0][0]*plan_matrix[0][2] + plan_matrix[1][0]*plan_matrix[1][2] + plan_matrix[2][0]*plan_
matrix[2][2] + plan_matrix[3][0]*plan_matrix[3][2]) / 4
a23 = (plan_matrix[0][1]*plan_matrix[0][2] + plan_matrix[1][1]*plan_matrix[1][2] + plan_matrix[2][1]*plan_
matrix[2][2] + plan_matrix[3][1]*plan_matrix[3][2]) / 4
a21 = a12
a31 = a13
a32 = a23

b0_numerator = np.array([[my, mx1, mx2, mx3],
                          [a1, a11, a12, a13],
                          [a2, a12, a22, a32],
                          [a3, a13, a23, a33]])
b0_denominator = np.array([[1, mx1, mx2, mx3],
                           [mx1, a11, a12, a13],
                           [mx2, a12, a22, a32],
                           [mx3, a13, a23, a33]])
b0 = np.linalg.det(b0_numerator) / np.linalg.det(b0_denominator)

b1_numerator = np.array([[1, my, mx2, mx3],
                          [mx1, a1, a12, a13],
                          [mx2, a2, a22, a32],
                          [mx3, a3, a23, a33]])
b1_denominator = np.array([[1, mx1, mx2, mx3],
                           [mx1, a11, a12, a13],
                           [mx2, a12, a22, a32],
                           [mx3, a13, a23, a33]])
b1 = np.linalg.det(b1_numerator) / np.linalg.det(b1_denominator)

b2_numerator = np.array([[1, mx1, my, mx3],
                          [mx1, a11, a1, a13],
                          [mx2, a12, a2, a32],
                          [mx3, a13, a3, a33]])
b2_denominator = np.array([[1, mx1, mx2, mx3],
                           [mx1, a11, a12, a13],

```

```

        [mx2, a12, a22, a32],
        [mx3, a13, a23, a33]))
b2 = np.linalg.det(b2_numerator) / np.linalg.det(b2_denominator)

b3_numerator = np.array([[1, mx1, mx2, my],
                          [mx1, a11, a12, a1],
                          [mx2, a12, a22, a2],
                          [mx3, a13, a23, a3]])
b3_denominator = np.array([[1, mx1, mx2, mx3],
                           [mx1, a11, a12, a13],
                           [mx2, a12, a22, a32],
                           [mx3, a13, a23, a33]])
b3 = np.linalg.det(b3_numerator) / np.linalg.det(b3_denominator)

b = [b0, b1, b2, b3]

# 5. Запишемо отримане рівняння регресії
print(f"\nРівняння регресії\nty = {b0} + {b1}*x1 + {b2}*x2 + {b3}*x3")

# 6. Зробимо перевірку (підставимо значення факторів з матриці планування
# і порівняємо результат з середнім значенням функції відгуку за рядками)
y1_reg = b0 + b1 * plan_matrix[0][0] + b2 * plan_matrix[0][1] + b3 * plan_matrix[0][2]
y2_reg = b0 + b1 * plan_matrix[1][0] + b2 * plan_matrix[1][1] + b3 * plan_matrix[1][2]
y3_reg = b0 + b1 * plan_matrix[2][0] + b2 * plan_matrix[2][1] + b3 * plan_matrix[2][2]
y4_reg = b0 + b1 * plan_matrix[3][0] + b2 * plan_matrix[3][1] + b3 * plan_matrix[3][2]

if (round(y1_reg, 0) == round(y1_average, 0)
    and round(y2_reg, 0) == round(y2_average, 0)
    and round(y3_reg, 0) == round(y3_average, 0)
    and round(y4_reg, 0) == round(y4_average, 0)):
    print(f"\nПеревірку закінчено. Коефіцієнти розраховані ПРАВИЛЬНО")
else: print(f"\nПеревірку закінчено. Коефіцієнти розраховані НЕПРАВИЛЬНО")

# 7. Перевірка однорідності дисперсії за критерієм Кохрена

print("\nПеревірка за критерієм Кохрена.....\n")

if cochrane_criteria(): print(f"\nЗа критерієм Кохрена дисперсія ОДНОРІДНА\n")
else: print(f"\nЗа критерієм Кохрена дисперсія НЕОДНОРІДНА\n")

# 8. Далі оцінимо значимість коефіцієнтів регресії згідно критерію Стюдента

print("-"*42 + "\nПеревірка значущості коефіцієнтів за критерієм Стюдента.....\n")

std_ch = students_criteria()
print(f"\n{std_ch}")

print(f"\nРівняння регресії\nty = {std_ch[0]} + {std_ch[1]}*x1 + {std_ch[2]}*x2 + {std_ch[3]}*x3")

y1_std = std_ch[0] + std_ch[1] * plan_matrix[0][0] + std_ch[2] * plan_matrix[0][1] + std_ch[3] * plan_matrix[0][2]

```

```

y2_std = std_ch[0] + std_ch[1] * plan_matrix[1][0] + std_ch[2] * plan_matrix[1][1] + std_ch[3] * plan_matrix[1][2]
y3_std = std_ch[0] + std_ch[1] * plan_matrix[2][0] + std_ch[2] * plan_matrix[2][1] + std_ch[3] * plan_matrix[2][2]
y4_std = std_ch[0] + std_ch[1] * plan_matrix[3][0] + std_ch[2] * plan_matrix[3][1] + std_ch[3] * plan_matrix[3][2]
y_std = [y1_std, y2_std, y3_std, y4_std]
print(f"\ny\u2081 = {y1_std}\ny\u2082 = {y2_std}\ny\u2083 = {y3_std}\ny\u2084 = {y4_std}")

# 9. Критерій Фішера

print("-"*42 + "\nПеревірка за критерієм Фішера.....\n")

fisher_criteria()

```



# Результат роботи програми

## Позитивний результат

```
Мінімальне значення ф-ції відгуку: 208
Максимальне значення ф-ції відгуку: 223

Матриця експерименту:
[-5, 25, 25]
[-25, 45, 30]
[-5, 25, 30]
[-5, 45, 25]

Функції відгуку
[211, 213, 222]
[208, 219, 221]
[216, 208, 219]
[220, 222, 220]

Середні значення ф-цій відгуку
y1: 215.33333333333331
y2: 216.0
y3: 214.33333333333331
y4: 220.66666666666663

Рівняння регресії
y = 214.58333333332055 + 0.1833333333339547*x1 + 0.2666666666666664*x2 + -0.19999999999985982*x3

Перевірку закінчено. Коefіцієнти розраховані ПРАВИЛЬНО

Перевірка за критерієм Кохрена.....

Дисперсії по рядках:
[22.88888888888893, 32.666666666666664, 21.555555555555557, 0.8888888888888888]

За критерієм Кохрена дисперсія ОДНОРІДНА

.....
Перевірка значущості коefіцієнтів за критерієм Стюдента.....

{0: 214.58333333332055, 1: 0, 2: 0, 3: 0}

Рівняння регресії
y = 214.58333333332055 + 0*x1 + 0*x2 + 0*x3

y1 = 214.58333333332055
y2 = 214.58333333332055
y3 = 214.58333333332055
y4 = 214.58333333332055
.....
Перевірка за критерієм Фішера.....

Fp < Ft
Рівняння регресії адекватно оригіналу
```

## Негативний результат

```
Мінімальне значення ф-ції відгуку: 208
Максимальне значення ф-ції відгуку: 223

Матриця експерименту:
[-5, 25, 25]
[-25, 45, 30]
[-5, 25, 30]
[-5, 45, 25]

Функції відгуку
[212, 218, 218]
[220, 218, 218]
[209, 212, 218]
[223, 219, 217]

Середні значення ф-цій відгуку
y1: 216.0
y2: 216.0
y3: 210.33333333333334
y4: 219.66666666666663

Рівняння регресії
y = 239.24999999999488 + -0.1000000000005577*x1 + 0.1833333333331728*x2 + -1.1333333333336963*x3

Перевірку закінчено. Коефіцієнти розраховані ПРАВИЛЬНО

Перевірка за критерієм Кохрена.....

Дисперсії по рядках:
[7.999999999999999, 18.666666666666664, 1.5555555555555556, 6.222222222222222]

За критерієм Кохрена дисперсія ОДНІРІДНА

-----
Перевірка значущості коефіцієнтів за критерієм Стюдента.....

{0: 239.24999999999488, 1: 0, 2: 0.1833333333331728, 3: -1.1333333333336963}

Рівняння регресії
y = 239.24999999999488 + 0*x1 + 0.1833333333331728*x2 + -1.1333333333336963*x3

y1 = 215.49999999998542
y2 = 213.4999999999833
y3 = 209.83333333331694
y4 = 218.91666666665247
-----
Перевірка за критерієм Фішера.....

Fr > Ft
Отже, рівняння регресії неадекватно оригіналу при рівні значимості 0.05
```

## **Висновок:**

Під час виконання даної лабораторної роботи я провів трьохфакторний експеримент, перевінив однорідність дисперсії за критерієм Кохрена, отримав коефіцієнти рівняння регресії, оцінив значимість знайдених коефіцієнтів за критеріями Стюдента та Фішера.

Отже, мета лабораторної роботи була досягнута.

## **Контрольні питання**

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

ДФЕ – це частина ПФЕ, який мінімізує число дослідів, за рахунок тієї інформації, яка не дуже істотна для побудови лінійної моделі.

### **2. Для чого потрібно розрахункове значення Кохрена?**

Критерій Кохрена використовують для порівняння трьох і більше виборок однакового обсягу  $n$

### **3. Для чого перевіряється критерій Стюдента?**

За допомогою критерія Стюдента перевіряється значущість коефіцієнта рівняння регресії.

Тобто, якщо виконується нерівність  $t_s < t_{\text{табл}}$ , то приймається нуль-гіпотеза, тобто вважається, що знайдений коефіцієнт  $\beta_s$  є статистично незначущим і його слід виключити з рівняння регресії.

Якщо  $t_s > t_{\text{табл}}$  то гіпотеза не підтверджується, тобто  $\beta_s$  – значимий коефіцієнт і він залишається в рівнянні регресії.

### **4. Чим визначається критерій Фішера і як його застосовувати?**

Отримане рівняння регресії необхідно перевірити на адекватність досліджуваному об'єкту. Для цієї мети необхідно оцінити, наскільки відрізняються середні значення у вихідної величини, отриманої в точках факторного простору, і значення  $y$ , отриманого з рівняння регресії в тих самих точках факторного простору. Для цього використовують дисперсію адекватності.

Адекватність моделі перевіряють за F-критерієм Фішера.

Якщо  $F_{\text{прак}} < F_{\text{теор}}$ , то отримана математична модель з прийнятим рівнем статистичної значимості  $q$  адекватна експериментальним даним.