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 ЕКСПЕРИМЕТНУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАЛА:
студентка 2 курсу
групи ІВ-91
Яременко В.Д.
Залікова – 9130

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Київ – 2021

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

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

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

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

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_1		x_2		x_3	
	min	max	min	max	min	max
130	10	50	20	60	20	25

Програмний код

```
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
    # 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

    # 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)
    s_beta_2 = sb / (N * m)

    s_beta = math.sqrt(s_beta_2)


    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

    t0 = abs(beta0) / s_beta
    t1 = abs(beta1) / s_beta
    t2 = abs(beta2) / s_beta
    t3 = abs(beta3) / s_beta


    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

    s2_b = average(dispersion_list)

    fp = s2_ad / s2_b


    ft = 4.5

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


x_min = [10, 20, 20]
x_max = [50, 60, 25]
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)}")

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

# 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

# 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\ny\u2081: {y1_average}\ny\u2082:
{y2_average}\ny\u2083: {y3_average}\ny\u2084: {y4_average}")
average_list = [y1_average, y2_average, y3_average, y4_average]

# 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 + plan_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 + plan_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 + plan_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]

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

# 6. Зробимо перевірку (підставимо значення факторів з матриці планування
#   і порівняємо результат з середнім значенням фунції 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")

# Перевірка однорідності дисперсії за критерієм Кохрена

print("-" * 42 + "\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")

# Далі оцінимо значимість коефіцієнтів регресії згідно критерію Стьюдента

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

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

print(f"\nРівняння регресії\ny = {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[1] * 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}")

# Критерій Фішера

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

fisher_criteria()
```

Результат роботи програми

```
Мінімальне значення ф-ції відгуку: 216
Максимальне значення ф-ції відгуку: 245

Матриця експерименту:
[50, 20, 20]
[10, 60, 25]
[50, 20, 25]
[50, 60, 20]

Функції відгуку
[233, 241, 220]
[220, 241, 238]
[233, 217, 235]
[226, 235, 227]

Середні значення ф-цій відгуку
y₁: 231.33333333333331
y₂: 233.0
y₃: 228.33333333333331
y₄: 229.33333333333331

Рівняння регресії
y = 252.66666666666973 + -0.16666666666669547*x1 + -0.050000000000017794*x2 + -0.5999999999999927*x3

Перевірку закінчено. Коефіцієнти розраховані ПРАВИЛЬНО
----------------------------------------
```

```
Перевірка за критерієм Кохрена

Дисперсії по рядках:
[74.88888888888889, 86.0, 64.88888888888889, 16.22222222222222]

За критерієм Кохрена дисперсія ОДНОРІДНА

----------------------------------------
Перевірка значущості коефіцієнтів за критерієм Стьюдента


{0: 252.66666666666973, 1: 0, 2: 0, 3: 0}

Рівняння регресії
y = 252.66666666666973 + 0*x1 + 0*x2 + 0*x3

y₁ = 252.66666666666973
y₂ = 252.66666666666973
y₃ = 252.66666666666973
y₄ = 252.66666666666973
----------------------------------------
Перевірка за критерієм Фішера

Fp > Fт
Отже, рівняння регресії неадекватно оригіналу при рівні значимості 0.05
```

Висновок:

Під час виконання даної лабораторної роботи я провела трьохфакторний експеримент, перевірила однорідність дисперсії за критерієм Кохрена, отримала коефіцієнти рівняння регресії, оцінила значимість знайдених коефіцієнтів за критеріями Стьюдента 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 Стьюдента?**

За допомогою критерія Стьюдента перевіряється значущість коефіцієнта рівняння регресії.

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

Якщо $t_s > t_{табл}$ то гіпотеза не підтверджується, тобто βs – 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 у, отриманого з рівняння регресії в тих самих точках факторного простору. Для цього використовують дисперсію адекватності.

Адекватність моделі перевіряють за F-критерієм Фішера.

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