

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

ЛАБОРАТОРНА РОБОТА № 3

з дисципліни «Методи оптимізації та планування експерименту» на
тему

«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАВ:
студент II курсу ФІОТ
групи ІВ-91
Карамшук Володимир
Варіант: 113

ПЕРЕВІРИВ:
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 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.

Завдання по варіанту

113	-15	30	5	40	5	25
-----	-----	----	---	----	---	----

Код програми

```
import math
import random

# Функції для обчислення mx1, mx2, mx3, my, a1, a2, a3, a11, a22, a33, a12 =
a21, a13 = a31, a23 = a32
def Fmx(arr, main_arr):
    main_arr.append(sum(arr) / len(arr))

def F_a(arr, main_arr):
    main_arr.append(
        (arr[0] * averageY[0] + arr[1] * averageY[1] + arr[2] * averageY[2] +
         arr[3] * averageY[3]) / len(arr))

def Fa_ii(arr, main_arr):
    main_arr.append((arr[0] ** 2 + arr[1] ** 2 + arr[2] ** 2 + arr[3] ** 2) /
                     len(arr))

def Fa_ij(arr1, arr2, main_arr):
    main_arr.append((arr1[0] * arr2[0] + arr1[1] * arr2[1] + arr1[2] *
                     arr2[2] + arr1[3] * arr2[3]) / len(arr1))

m = 3
```

```

while True:
    GT = 0.7679 # Кохрена
    # матриця планування
    N = 4
    Tf = 2.306 # Розподіл Стюдента
    Ft = 4.5 # Розподіл Фішера
    # Дані відносно варіанту
    x1_min = -15
    x1_max = 30
    x2_min = 5
    x2_max = 40
    x3_min = 5
    x3_max = 25
    y_max = 200 + (x1_max + x2_max + x3_max) / 3
    y_min = 200 + (x1_min + x2_min + x3_min) / 3
    # матриця планування
    plan_matrix = [[1, 1, 1, 1],
                   [-1, -1, 1, 1],
                   [-1, 1, -1, 1],
                   [-1, 1, 1, -1]]

    # Заповнимо матрицю планування для m = 3.

    x1 = [random.randint(x1_min, x1_max + 1) for i in range(4)]
    x2 = [random.randint(x2_min, x2_max + 1) for i in range(4)]
    x3 = [random.randint(x3_min, x3_max + 1) for i in range(4)]
    y1 = [random.randint(int(y_min), int(y_max) + 1) for i in range(4)]
    y2 = [random.randint(int(y_min), int(y_max) + 1) for i in range(4)]
    y3 = [random.randint(int(y_min), int(y_max) + 1) for i in range(4)]

    # Знайдемо середні значення функції відгуку за рядками:
    averageY = [0, 0, 0, 0]
    for i in range(0, len(x1)):
        averageY[i] = (y1[i] + y2[i] + y3[i]) / 3
    print(f"average_Y: {averageY}")

    # обчислення mx1, mx2, mx3, my, a1, a2, a3, a11, a22, a33, a12 = a21, a13
    = a31, a23 = a32

    mx = []
    Fmx(x1, mx)
    Fmx(x2, mx)
    Fmx(x3, mx)
    print(f"MX: {mx}")

    my = (averageY[0] + averageY[1] + averageY[2] + averageY[3]) /
len(averageY)
    print(f"MY: {my}")

    a = []
    F_a(x1, a)
    F_a(x2, a)
    F_a(x3, a)
    print(f"A: {a}")

    a11 = []
    Fa_ii(x1, a11)
    print(f"A11: {a11}")

    a22 = []
    Fa_ii(x2, a22)
    print(f"A22: {a22}")

```

```

a33 = []
Fa_ii(x3, a33)
print(f"A33: {a33}")

a12 = a21 = []
Fa_ij(x1, x2, a12)
print(f"A12 = A21: {a12}")

a13 = a31 = []
Fa_ij(x1, x3, a13)
print(f"A13 = A31: {a13}")

a23 = a32 = []
Fa_ij(x2, x3, a23)
print(f"A23 = A32: {a23}")
# b0 b1 b2 b3
r01 = [1, mx[0], mx[1], mx[2]]
r02 = [mx[0], a11[0], a12[0], a13[0]]
r03 = [mx[1], a12[0], a22[0], a32[0]]
r04 = [mx[2], a13[0], a23[0], a33[0]]
temp0 = [r01, r02, r03, r04]
Determinanta = 1 * a11[0] * a22[0] * a33[0] + mx[0] * a12[0] * a32[0] *
mx[2] + mx[1] * a13[0] * mx[1] * a13[0] + mx[
2] * mx[0] * a12[0] * a23[0] - \
(mx[2] * a12[0] * a12[0] * mx[2] + a13[0] * a22[0] *
a13[0] * 1 + a23[0] * a32[0] * mx[0] * mx[0] + a33[
0] * mx[1] * a11[0] * mx[1])

r11 = [my, mx[0], mx[1], mx[2]]
r12 = [a[0], a11[0], a12[0], a13[0]]
r13 = [a[1], a12[0], a22[0], a32[0]]
r14 = [a[2], a13[0], a23[0], a33[0]]
temp1 = [r11, r12[0], r13[0], r14[0]]
Determinanta1 = my * a11[0] * a22[0] * a33[0] + a[0] * a12[0] * a32[0] *
mx[2] + a[1] * a13[0] * mx[1] * a13[0] + a[2] * \
mx[0] * a12[0] * a23[0] - \
(a[2] * a12[0] * a12[0] * mx[2] + a13[0] * a22[0] *
a13[0] * my + a23[0] * a32[0] * a[0] * mx[0] + a33[
0] * a[1] * a11[0] * mx[1])

r21 = [1, my, mx[1], mx[2]]
r22 = [mx[0], a[0], a12[0], a13[0]]
r23 = [mx[1], a[1], a22[0], a32[0]]
r24 = [mx[2], a[2], a23[0], a33[0]]
temp2 = [r21, r22, r23, r24]
Determinanta2 = 1 * a[0] * a22[0] * a33[0] + my * a12[0] * a32[0] * mx[2]
+ mx[1] * a13[0] * mx[1] * a[2] + mx[2] * mx[
0] * a[1] * a23[0] - \
(mx[2] * a[1] * a12[0] * mx[2] + a[2] * a22[0] * a13[0] *
1 + a23[0] * a32[0] * my * mx[0] + a33[0] *
mx[1] * a[0] * mx[1])

r31 = [1, mx[0], my, mx[2]]
r32 = [mx[0], a11[0], a[0], a13[0]]
r33 = [mx[1], a12[0], a[1], a32[0]]
r34 = [mx[2], a13[0], a[2], a33[0]]
temp3 = [r31, r32, r33, r34]
Determinanta3 = 1 * a11[0] * a[1] * a33[0] + mx[0] * a[0] * a32[0] *
mx[2] + my * a13[0] * mx[1] * a13[0] + mx[2] * mx[
0] * a12[0] * a[2] - \
(mx[2] * a12[0] * a[0] * mx[2] + a13[0] * a[1] * a13[0] *
1 + a[2] * a32[0] * mx[0] * mx[0] + a33[0] *
mx[1] * a11[0] * my)

```

```

r41 = [1, mx[0], mx[1], my]
r42 = [mx[0], a11[0], a12[0], a[0]]
r43 = [mx[1], a12[0], a22[0], a[1]]
r44 = [mx[2], a13[0], a23[0], a[2]]
temp4 = [r41, r42, r43, r44]
Determinanta4 = 1 * a11[0] * a22[0] * a[2] + mx[0] * a12[0] * a[1] *
mx[2] + mx[1] * a[0] * mx[1] * a13[0] + my * mx[
    0] * a12[0] * a23[0] - \
    (mx[2] * a12[0] * a12[0] * my + a13[0] * a22[0] * a[0] *
1 + a23[0] * a[1] * mx[0] * mx[0] + a[2] * mx[
    1] * a11[0] * mx[1])

b = [0, 0, 0, 0]
b[0] = Determinanta1 / Determinanta
b[1] = Determinanta2 / Determinanta
b[2] = Determinanta3 / Determinanta
b[3] = Determinanta4 / Determinanta

print("*****")
for i in range(4):
    print(f"b{i}: {b[i]}")
print("*****")

# Запишемо отримане рівняння регресії:
y = []
for i in range(len(x1)):
    y.append(b[0] + b[1] * x1[i] + b[2] * x2[i] + b[3] * x3[i])
    print(f"Y{i + 1}: {y[i]}")

Dispersiya = [0, 0, 0, 0]
for i in range(0, len(Dispersiya)):
    Dispersiya[i] = ((y1[i] - averageY[i]) ** 2 + (y2[i] - averageY[i])
** 2 + (y3[i] - averageY[i]) ** 2) / 3
    print(f"Dispersiya: {Dispersiya}")

print("*****")
print('Перевірка однорідності дисперсії за критерієм Кохрена:')
GP = max(Dispersiya) / sum(Dispersiya)
if GP < GT:
    print("Дисперсія однорідна")
    break
else:
    print('Дсперсія не однорідна. Потрібно збільшити m, поточний m={}'
збільшимо на 1'.format(m))
    m=m+1

# Далі оцінимо значимість коефіцієнтів регресії згідно критерію Стюдента
sb = sum(Dispersiya) / len(Dispersiya)
Odnoridna2 = sb / (N * m)
Odnoridna = math.sqrt(Odnoridna2)

beta = [0, 0, 0, 0]
beta[0] = (averageY[0] * 1 + averageY[1] * 1 + averageY[2] * 1 + averageY[3]
* 1) / N
beta[1] = (averageY[0] * (-1) + averageY[1] * (-1) + averageY[2] * 1 +
averageY[3] * 1) / N
beta[2] = (averageY[0] * (-1) + averageY[1] * 1 + averageY[2] * (-1) +
averageY[3] * 1) / N
beta[3] = (averageY[0] * (-1) + averageY[1] * 1 + averageY[2] * 1 +
averageY[3] * (-1)) / N
print(f"Beta: {beta}")

t = []
for i in range(len(beta)):

```

```

        t.append(abs(beta[i]) / Odnoridna)
print(f"t0: {t}")

print("*****")
print('\nОцінка значимості коефіцієнтів регресії згідно критерію Стьюдента:')
d = 0 # кількість значимих коефіцієнтів
temp = [0, 0, 0, 0]
for i in range(0, N):
    if t[i] <= Tf:
        print(f"t[{i}] = {t[i]} <= Tf = {Tf} >= b[{i}] = {b[i]} - не значний коефіцієнт")
        temp[i] = 0
    else:
        print(f"t[{i}] = {t[i]} > Tf = {Tf} >= b[{i}] = {b[i]} - значний коефіцієнт")
        temp[i] = b[i]
        d += 1

y_2 = []
for i in range(0, N):
    y_2.append(temp[0] + temp[1] * x1[i] + temp[2] * x2[i] + temp[3] * x3[i])
print(y_2)

# Критерій Фішера
print("*****")
print("Критерій Фішера")
s_ad = []
tmp = []
kof = m / (N - d)
for i in range(len(y_2)):
    tmp.append((y_2[i] - averageY[i]) ** 2)
s_ad.append(kof * sum(tmp))
print(f"S_ad: {s_ad}")

fp = s_ad[0] / Odnoridna2
print(f"Fp: {fp}")
if fp > Ft:
    print("рівняння регресії неадекватно оригіналу при рівні значимості 0.05")
else:
    print("рівняння регресії адекватно оригіналу при рівні значимості 0.05")
print(f"y = {b[0]} + {b[1]} * x1 + {b[2]} * x2 + {b[3]} * x3")

```

Приклад роботи програми

```
average_Y: [209.0, 216.0, 215.0, 218.66666666666666]
MX: [25.75, 25.25, 11.75]
MY: 214.66666666666666
A: [5517.166666666666, 5417.333333333333, 2540.833333333333]
A11: [678.75]
A22: [707.25]
A33: [182.25]
A12 = A21: [632.25]
A13 = A31: [276.75]
A23 = A32: [326.25]
*****
b0: 222.4789555257323
b1: 6.357516000080226
b2: -0.30975049951246286
b3: 4.3238826437030475
*****
Y1: 422.8781994058783
Y2: 438.15272097378477
Y3: 423.76694068532333
Y4: 431.87979284949046
Dispersiya: [52.666666666666664, 24.666666666666668, 100.66666666666667, 80.88888888888889]
*****
Перевірка однорідності дисперсії за критерієм Кохрена:
Дисперсія однорідна
Beta: [214.66666666666666, 2.1666666666666643, 2.6666666666666643, 0.8333333333333357]
t0: [92.43331432209128, 0.932944942691914, 1.1482399294669714, 0.3588249779584299]
*****
```

```
Оцінка значимості коефіцієнтів регресії згідно критерію Стюдента:
t[0] = 92.43331432209128 > Tf = 2.306 >= b[0] = 222.4789555257323 - значний коефіцієнт
t[1] = 0.932944942691914 <= Tf = 2.306 >= b[1] = 6.357516000080226 - не значний коефіцієнт
t[2] = 1.1482399294669714 <= Tf = 2.306 >= b[2] = -0.30975049951246286 - не значний коефіцієнт
t[3] = 0.3588249779584299 <= Tf = 2.306 >= b[3] = 4.3238826437030475 - не значний коефіцієнт
[222.4789555257323, 222.4789555257323, 222.4789555257323, 222.4789555257323]
*****
Критерій Фішера
S_ad: [294.12742886992345]
Fp: 54.53349754154803
рівняння регресії неадекватно оригіналу при рівні значимості 0.05
y = 222.4789555257323 + 6.357516000080226 * x1 + -0.30975049951246286 * x2 + 4.3238826437030475 * x3
```

Контрольні запитання

1. Що називається дробовим факторним експериментом?

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

2. Розрахункове значення Кохрена показує, яку частку в загальній сумі дисперсій у рядках має максимальна з них.

3. Критерій Стюдента використовується для перевірки значущості коефіцієнтів.
4. Критерій Фішера використовується для перевірки адекватності рівняння регресії.