

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 №3
з дисципліни
«Методи оптимізації та планування експерименту»

Виконав:

Гречаний Є.О.

Перевірив:

ас. Регіда П. Г.

Київ 2021

Тема: ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ

ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ.

Мета: провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти

коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

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.

Програмний код

```
import random, numpy

def rozpodil_kohren(S_list):
    return max(S_list) / sum(S_list)

def rozpodil_student(S_list, y_avr_list, x_n):
    disp_avr = sum(S_list) / 4
    S_b = (disp_avr / 12) ** 0.5
    b_list = []
    for i in range(4):
        b_list.append(sum([y_avr_list[j] * x_n[i][j] for j in range(4)]) / 4)
    t_list = [abs(b_list[i]) / S_b for i in range(4)]
    new_b = []
    for i in range(4):
        if t_list[i] > 2.306:
            new_b.append(B[i])
        else:
            new_b.append(0)
    return new_b

def rozpodil_fisher(new_B, new_Y, y_avr_list):
    amount_of_weig = 4 - new_B.count(0)
    S_ad = 3 / (4 - amount_of_weig) * sum([(new_Y[i] - y_avr_list[i]) ** 2
    for i in range(4)])
```

```

S_b = (sum(S_list) / 4 / 12)
F_p = (S_ad) / (S_b)
print(F_p)
if F_p > 4.5:
    return False
else:
    return True

# Variant 203
x_min = [-20, -25, -25]
x_max = [30, 10, -20]

x_avr_min = sum(x_min) / 3
x_avr_max = sum(x_max) / 3

y_min = 200 + int(x_avr_min)
y_max = 200 + int(x_avr_max)

x_list = [[random.randint(x_min[j], x_max[j]) for j in range(3)] for i in range(4)]
y_list = [[random.randint(y_min, y_max) for j in range(3)] for i in range(4)]

y_avr_list = [sum(i) / 3 for i in y_list]

a = [sum([x_list[j][i] * y_avr_list[j] for j in range(4)]) / 4 for i in range(3)]

mx1 = sum(x_list[i][0] for i in range(4)) / 4
mx2 = sum(x_list[i][1] for i in range(4)) / 4
mx3 = sum(x_list[i][2] for i in range(4)) / 4

my = sum(y_avr_list) / 4

a12 = sum([x_list[i][0] * x_list[i][1] for i in range(4)]) / 4
a13 = sum([x_list[i][0] * x_list[i][2] for i in range(4)]) / 4
a23 = sum([x_list[i][1] * x_list[i][2] for i in range(4)]) / 4
a11 = sum([x_list[i][0] ** 2 for i in range(4)]) / 4
a22 = sum([x_list[i][1] ** 2 for i in range(4)]) / 4
a33 = sum([x_list[i][2] ** 2 for i in range(4)]) / 4

X = [[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a23], [mx3, a13, a23, a33]]

Y = [my, *a]

B = [round(i, 3) for i in numpy.linalg.solve(X, Y)]

x_n = [[1, 1, 1, 1],
        [-1, -1, 1, 1],
        [-1, 1, -1, 1],
        [-1, 1, 1, -1]]

S_list = [sum([(y_avr_list[i] - y_list[i][j]) ** 2 for j in range(3)]) / 3 for i in range(4)]
new_B_values = rozpodil_student(S_list, y_avr_list, x_n)
new_Y_values = [
    new_B_values[0] + new_B_values[1] * x_list[i][0] + new_B_values[2] *
    x_list[i][1] + new_B_values[3] * x_list[i][2]
    for i in range(4)]

print("\nМатриця планування для m=3")
print("{}{:5}{:5}{:5}{:5}{:5}{:5}".format('X1', 'X2', 'X3', 'Y1', 'Y2', 'Y3'))

```

```

for i in zip(x_list, y_list):
    print("{:5}{:5}{:5}{:5}{:5}{:5}".format(*i[0], *i[1]))
print("Матриця планування експерименту\n")
print("    {:5}{:5}{:5}{:5}{:5}".format('X0', 'X1', 'X2', 'X3'))
for i in x_n:
    print("{:5}{:5}{:5}{:5}{:5}".format(*i))
print('mx1: ', mx1)
print('mx2: ', mx2)
print('mx3: ', mx3)
print("a12: ", a12)
print("a13: ", a13)
print("a23: ", a23)
print("a11: ", a11)
print("a22: ", a22)
print("a33: ", a33)
print("Отримані коефіцієнти B: ", *B, sep=" | ")
print("Перевірка: ")
Y = [B[0] + B[1] * x_list[i][0] + B[2] * x_list[i][1] + B[3] * x_list[i][2]
for i in range(4)]
print(Y)
print('Середні значення відгуку по рядках: ', end='')
print(*y_avr_list, sep=" | ")
print("Перевірка за критерієм Кохрена: ")
if rozpodil_kohren(S_list) < 0.7679:
    print('Дисперсія однорідна')
else:
    print('Дисперсія не однорідна')
print("Перевірка за критерієм Стюдента: ")
print('Значимі коефіцієнти(незначимі дорівнюють нулю):', *new_B_values)
print("Перевірка за критерієм Фішера при рівні значимості 0.05: ")
print('Fb:', )
fisher = rozpodil_fisher(new_B_values, new_Y_values, y_avr_list)
if fisher:
    print('Рівняння регресії адекватно оригіналу')
else:
    print('Рівняння регресії неадекватно оригіналу')

```

Результати виконання

Матриця планування для m=3

X1	X2	X3	Y1	Y2	Y3
-13	-13	-20	182	181	182
16	-11	-23	200	192	199
-13	7	-20	201	199	177
25	-14	-20	196	179	180

Матриця планування експерименту

X0	X1	X2	X3
1	1	1	1
-1	-1	1	1
-1	1	-1	1
-1	1	1	-1

mx1: 3.75

mx2: -7.75

mx3: -20.75

a12: -112.0

a13: -87.0

a23: 163.25

a11: 304.75

a22: 133.75

a33: 432.25

Отримані коефіцієнти B: | 114.484 | 0.102 | 0.533 | -3.772

Перевірка:

[181.66899999999998, 197.00900000000001, 192.329, 185.012]

Середні значення відгуку по рядках: 181.66666666666666 | 197.0 | 192.33333333333334 | 185.0

Перевірка за критерієм Кохрена:

Дисперсія однорідна

Перевірка за критерієм Ст'юдента:

Значимі коефіцієнти(незначимі дорівнюють нулю): 114.484 0 0 -3.772

Перевірка за критерієм Фішера при рівні значимості 0.05:

Fb:

43.62992893626876

Рівняння регресії неадекватно оригіналу