

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 № 4

з дисципліни «Методи наукових досліджень» на тему
«Проведення трьохфакторного експерименту при використанні
рівняння регресії з урахуванням ефекту взаємодії.»

ВИКОНАВ:
студент II курсу ФІОТ
групи ІВ-92
Злочевський Нікіта Вікторович
Варіант: 209
ПЕРЕВІРИВ:
Регіда П. Г.

Хід роботи

Мета: Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

Завдання:

Завдання на лабораторну роботу

1. Скласти матрицю планування для повного трьохфакторного експерименту.
2. Провести експеримент, повторивши N раз досліди у всіх точках факторного простору і знайти значення відгуку Y. Знайти значення Y шляхом моделювання випадкових чисел у певному діапазоні відповідно варіанту. Варіанти вибираються за номером в списку в журналі викладача.

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

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

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

3. Знайти коефіцієнти рівняння регресії і записати його.
4. Провести 3 статистичні перевірки – за критеріями Кохрена, Ст'юдента, Фішера.
5. Зробити висновки по адекватності регресії та значимості окремих коефіцієнтів і записати скореговане рівняння регресії.
6. Написати комп'ютерну програму, яка усе це моделює.

№ варіанта	x_1		x_2		x_3	
	min	max	min	Max	min	max
209	-30	0	10	60	10	35

Лістинг програми

```
import random
import numpy as np
from numpy.linalg import solve
from scipy.stats import f, t

N = 8
x1_min, x1_max, x2_min, x2_max, x3_min, x3_max = [-30, 0, 10, 60, 10, 35]
avg_x_min = (x1_min + x2_min + x3_min)/3
avg_x_max = (x1_max + x2_max + x3_max)/3
y_min = round(200 + avg_x_min)
y_max = round(200 + avg_x_max)

xn = [[1, 1, 1, 1, 1, 1, 1, 1],
      [-1, -1, 1, 1, -1, -1, 1, 1],
      [-1, 1, -1, 1, -1, 1, -1, 1],
      [-1, 1, 1, -1, 1, -1, -1, 1]]

x1x2_norm, x1x3_norm, x2x3_norm, x1x2x3_norm = [0] * 8, [0] * 8, [0] * 8, [0] * 8

for i in range(N):
    x1x2_norm[i] = xn[1][i] * xn[2][i]
    x1x3_norm[i] = xn[1][i] * xn[3][i]
    x2x3_norm[i] = xn[2][i] * xn[3][i]
    x1x2x3_norm[i] = xn[1][i] * xn[2][i] * xn[3][i]

y1 = [random.randint(int(y_min), int(y_max)) for i in range(8)]
y2 = [random.randint(int(y_min), int(y_max)) for i in range(8)]
y3 = [random.randint(int(y_min), int(y_max)) for i in range(8)]

# матриця планування
y_matrix = [[y1[0], y2[0], y3[0]],
            [y1[1], y2[1], y3[1]],
            [y1[2], y2[2], y3[2]],
            [y1[3], y2[3], y3[3]],
            [y1[4], y2[4], y3[4]],
            [y1[5], y2[5], y3[5]],
            [y1[6], y2[6], y3[6]],
            [y1[7], y2[7], y3[7]]]

print("Матриця планування y :\n")
for i in range(N):
    print(y_matrix[i])

x0 = [1, 1, 1, 1, 1, 1, 1, 1]
x1 = [-30, -30, 0, 0, -30, -30, 0, 0]
x2 = [10, 60, 10, 60, 10, 60, 10, 60]
x3 = [10, 35, 35, 10, 35, 10, 10, 35]
x1x2, x1x3, x2x3, x1x2x3 = [0] * 8, [0] * 8, [0] * 8, [0] * 8

for i in range(N):
    x1x2[i] = x1[i] * x2[i]
    x1x3[i] = x1[i] * x3[i]
    x2x3[i] = x2[i] * x3[i]
    x1x2x3[i] = x1[i] * x2[i] * x3[i]

Y_average = []
for i in range(len(y_matrix)):
    Y_average.append(np.mean(y_matrix[i], axis=0))

list_for_b = [xn[0], xn[1], xn[2], xn[3], x1x2_norm, x1x3_norm, x2x3_norm, x1x2x3_norm]
```

```

list_for_a = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3))

print("Матриця планування X:")
for i in range(N):
    print(list_for_a[i])

ai = [round(i, 3) for i in solve(list_for_a, Y_average)]
print("Рівняння регресії: \n" "y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1x2 + {}*x1x3 + {}*x2x3 + {}*x1x2x3".format(ai[0],
    ai[1], ai[2], ai[3], ai[4], ai[5], ai[6], ai[7]))

bi = []
for k in range(N):
    S = 0
    for i in range(N):
        S += (list_for_b[k][i] * Y_average[i]) / N
    bi.append(round(S, 3))
print("Рівняння регресії для нормованих факторів: \n" "y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1x2 + {}*x1x3 + {}*x2x3 + {}*x1x2x3".format(bi[0], bi[1], bi[2], bi[3], bi[4], bi[5], bi[6],
bi[7]))

print("Перевірка за критерієм Кохрена")
print("Середні значення відгуку за рядками:", "\n", +Y_average[0], Y_average[1],
Y_average[2], Y_average[3],
    Y_average[4], Y_average[5], Y_average[6], Y_average[7])

# розрахунок дисперсій
dispersions = []
for i in range(len(y_matrix)):
    a = 0
    for k in y_matrix[i]:
        a += (k - np.mean(y_matrix[i], axis=0)) ** 2
    dispersions.append(a / len(y_matrix[i]))
Gp = max(dispersions) / sum(dispersions)
Gt = 0.5157

# однорідність дисперсій
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")

print("Перевірка коефіцієнтів за критерієм Стюдента")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (8 * 3)) ** 0.5

t_list = [abs(bi[i]) / sbs for i in range(0, 8)]

d = 0
res = [0] * 8
coefficient_1 = []
coefficient_2 = []
m = 3
F3 = (m - 1) * N

for i in range(N):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coefficient_2.append(bi[i])
        res[i] = 0
    else:
        coefficient_1.append(bi[i])

```

```

        res[i] = bi[i]
        d += 1

print("Значущі коефіцієнти регресії:", coefficient_1)
print("Незначущі коефіцієнти регресії:", coefficient_2)
y_st = []
for i in range(N):
    y_st.append(res[0] + res[1] * xn[1][i] + res[2] * xn[2][i] + res[3] * xn[3][i] +
res[4] * x1x2_norm[i]\
                + res[5] * x1x3_norm[i] + res[6] * x2x3_norm[i] + res[7] *
x1x2x3_norm[i])
print("Значення з отриманими коефіцієнтами:\n", y_st)

# критерій Фішера
print("\nПеревірка адекватності за критерієм Фішера\n")
S_ad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(8)]) / (N - d)
Fp = S_ad / sb
F4 = N - d
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("Рівняння регресії адекватне при рівні значимості 0.05")
else:
    print("Рівняння регресії неадекватне при рівні значимості 0.05")

```

Результат роботи програми

```

[201, 218, 222]
[228, 230, 201]
[231, 222, 197]
[201, 229, 200]
[225, 218, 223]
[206, 215, 202]
[217, 202, 199]
[227, 221, 200]
Матриця планування X:
(1, -30, 10, 10, -300, -300, 100, -3000)
(1, -30, 60, 35, -1800, -1050, 2100, -63000)
(1, 0, 10, 35, 0, 0, 350, 0)
(1, 0, 60, 10, 0, 0, 600, 0)
(1, -30, 10, 35, -300, -1050, 350, -10500)
(1, -30, 60, 10, -1800, -300, 600, -18000)
(1, 0, 10, 10, 0, 0, 100, 0)
(1, 0, 60, 35, 0, 0, 2100, 0)
Рівняння регресії:
y = 200.56 + -0.376*x1 + 0.117*x2 + 0.464*x3 + 0.009*x1x2 + 0.005*x1x3 + -0.004*x2x3 + -0.0*x1x2x3
Рівняння регресії для нормованих факторів:
y = 213.958 + -1.792*x1 + -0.625*x2 + 4.625*x3 + 1.458*x1x2 + -0.458*x1x3 + -0.125*x2x3 + -1.042*x1x2x3
Перевірка за критерієм Кохрена
Середні значення відгуку за рядками:
213.66666666666666 219.66666666666666 216.66666666666666 210.0 222.0 207.66666666666666 206.0 216.0
Дисперсія однорідна
Перевірка коефіцієнтів за критерієм Стьюдента
Значущі коефіцієнти регресії: [213.958, 4.625]
Незначущі коефіцієнти регресії: [-1.792, -0.625, 1.458, -0.458, -0.125, -1.042]
Значення з отриманими коефіцієнтами:
[209.333, 218.583, 218.583, 209.333, 218.583, 209.333, 209.333, 218.583]

Перевірка адекватності за критерієм Фішера

Рівняння регресії адекватне при рівні значимості 0.05

```

Висновок:

На цій лабораторній роботі ми провели повний трьохфакторний експеримент. Знайшли рівняння регресії адекватне об'єкту. Провели 3 статистичні перевірки за критеріями Кохрена, Стюдента, Фішера. Закріпили отримані знання їх практичним використанням при написанні програми у середовищі Rucharm на мові python, що реалізує завдання лабораторної роботи.