

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 оптимізації та планування»
Проведення трьохфакторного експерименту
при використанні рівняння регресії з урахуванням ефекту взаємодії

ВИКОНАВ:
Студент II курсу ФІОТ
Групи ІВ-91
Богомол В.Ю. - 9101

ПЕРЕВІРИВ:
асистент
Регіда П.Г.

Київ 2021 р.

Мета:

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

Варіант завдання:

Варіант	X ₁		X ₂		X ₃	
	min	max	min	max	min	max
101	-40	20	10	60	-20	20

Лістинг програми:

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

# кількість повторення дослідів
n = 8

# варіант 101
x1min = -40
x1max = 20
x2min = 10
x2max = 60
x3min = -20
x3max = 20

# максимальне та мінімальне значення
y_max = 200 + (x1max + x2max + x3max) / 3
y_min = 200 + (x1min + x2min + x3min) / 3

# матриця ПФЕ
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]

# заповнення y (генерація)
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]]]

# вивід даних за допомогою цикла
print("Матриця планування y : \n")
for i in range(n):
    print(y_matrix[i])

x0 = [1, 1, 1, 1, 1, 1, 1, 1]

# заміна -1 на x1_мін, 1 на x1_макс
x1 = [10, 10, 50, 50, 10, 10, 50, 50]

# заміна -1 на x2_мін, 1 на x2_макс
x2 = [20, 60, 20, 60, 20, 60, 20, 60]

# заміна -1 на x3_мін, 1 на x3_макс
x3 = [20, 25, 25, 20, 25, 20, 20, 25]

# заповнення нулями x1x2, x1x3, x1x2x3
x1x2, x1x3, x2x3, x1x2x3 = [0] * 8, [0] * 8, [0] * 8, [0] * 8
# заповнення x1x2, x1x3, x1x2x3 добутками
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]
# середні y
Y_average = []
for i in range(len(y_matrix)):
    Y_average.append(np.mean(y_matrix[i], axis=0))

# формуємо списки b і a
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))

# вивід матриці планування X
print("Матриця планування X:")
for i in range(n):
    print(list_for_a[i])
# нормовані фактори b_i
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))
# розрахунок ai(система рівнянь) через функцію solve, вивід рівняння регресії
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]))
# вивід даних
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]))
# експериментально
Gp = max(dispersions) / sum(dispersions)
# теоретично
Gt = 0.5157
# перевірка однорідності дисперсій
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")
# критерій Стюдента
print("Перевірка значущості 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
coef_1 = []
coef_2 = []
# кількість повторень кожної комбінації
m = 3
F3 = (m - 1) * n
# перевірка значущості коефіцієнтів
for i in range(n):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coef_2.append(bi[i])
        res[i] = 0
    else:
        coef_1.append(bi[i])
        res[i] = bi[i]
        d += 1

# вивід
print("Значущі коефіцієнти регресії:", coef_1)
print("Незначущі коефіцієнти регресії:", coef_2)

# значення y з коефіцієнтами регресії
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")
Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(8)]) / (n - d)
Fp = Sad / 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 виконання роботи:

```
Матриця планування y :

[218, 223, 214]
[196, 196, 206]
[225, 227, 198]
[197, 208, 203]
[202, 191, 231]
[213, 228, 203]
[195, 194, 224]
[200, 230, 229]
Матриця планування X:
(1, 10, 20, 20, 200, 200, 400, 4000)
(1, 10, 60, 25, 600, 250, 1500, 15000)
(1, 50, 20, 25, 1000, 1250, 500, 25000)
(1, 50, 60, 20, 3000, 1000, 1200, 60000)
(1, 10, 20, 25, 200, 250, 500, 5000)
(1, 10, 60, 20, 600, 200, 1200, 12000)
(1, 50, 20, 20, 1000, 1000, 400, 20000)
(1, 50, 60, 25, 3000, 1250, 1500, 75000)
Рівняння регресії:
y = 273.083 + -2.158*x1 + 0.637*x2 + -2.458*x3 + -0.023*x1x2 + 0.089*x1x3 + -0.037*x2x3 + 0.001*x1x2x3
Рівняння регресії для нормованих факторів:
y = 210.458 + 0.375*x1 + -1.375*x2 + 0.458*x3 + 1.708*x1x2 + 6.875*x1x3 + -0.042*x2x3 + 1.208*x1x2x3
Перевірка за критерієм Кохрена
Середні значення відгуку за рядками:
218.33333333333334 199.33333333333334 216.66666666666666 202.66666666666666 208.0 214.66666666666666 204.33333333333334 219.66666666666666
Дисперсія однорідна

Перевірка значущості коефіцієнтів за критерієм Стюдента
Значущі коефіцієнти регресії: [210.458, 6.875]
Незначущі коефіцієнти регресії: [0.375, -1.375, 0.458, 1.708, -0.042, 1.208]
Значення з отриманими коефіцієнтами:
[217.333, 203.583, 217.333, 203.583, 203.583, 217.333, 203.583, 217.333]

Перевірка адекватності за критерієм Фішера

Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0
```

Висновок:

В даній лабораторній роботі я провів повний трьохфакторний експеримент з трьома статистичними.