

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Методи оптимізації та планування експерименту

Лабораторна робота №4

**«Проведення трьохфакторного експерименту при
використанні рівняння регресії з урахуванням ефекту
взаємодії»**

Виконала:

студентка групи ІВ-81

Базова Л.Г.

Номер залікової книжки № 8101

Перевірив: Регіда П.Г.

Київ 2020 р.

Код програми:

```
import numpy as np
from scipy.stats import f, t
from tabulate import tabulate
def mult(x1, x2, x3 = np.array([1, 1, 1, 1, 1, 1, 1, 1])):
    xn = []
    for i in range(N):
        xn.append(x1[i] * x2[i] * x3[i])
    return xn
x1_min = -10
x1_max = 50
x2_min = 20
x2_max = 60
x3_min = 50
x3_max = 55
print("y=b0+b1*x1+b2*x2+b3*x3+b12*x1*x2+b13*x1*x3+b23*x2*x3+b123*x1*x2*x3\n")

x_av_max = (x1_max + x2_max + x3_max) / 3
x_av_min = (x1_min + x2_min + x3_min) / 3
y_max = int(200 + x_av_max)
y_min = int(200 + x_av_min)

m = 4
N = 8
q = 0.05

X11 = np.array([-1, -1, -1, -1, 1, 1, 1, 1])
X22 = np.array([-1, -1, 1, 1, -1, -1, 1, 1])
X33 = np.array([-1, 1, -1, 1, -1, 1, -1, 1])
X00 = np.array([1, 1, 1, 1, 1, 1, 1, 1])
X12 = mult(X11, X22)
X13 = mult(X11, X33)
X23 = mult(X22, X33)
X123 = mult(X11, X22, X33)

print("Кодовані значення X")
header_table = ["M", "x1", "x2", "x3", "x1x2", "x1x3", "x2x3", "x1x2x3"]
code_table = []
for i in range(N):
    code_table.append([i+1, X11[i], X22[i], X33[i], X12[i], X13[i], X23[i], X123[i]])
print(tabulate(code_table, headers=header_table, tablefmt="fancy_grid"))

X1 = np.array([x1_min, x1_min, x1_min, x1_min, x1_max, x1_max, x1_max, x1_max])
X2 = np.array([x2_min, x2_min, x2_max, x2_max, x2_min, x2_min, x2_max, x2_max])
X3 = np.array([x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, x3_min, x3_max])
X12 = mult(X1, X2)
X13 = mult(X1, X3)
X23 = mult(X2, X3)
X123 = mult(X1, X2, X3)
y = np.random.randint(y_min, y_max, size=(N, m))

while 1:
    if(m > 4):
        next_int = np.random.randint(y_min, y_max, size=(N, 1))
        y = np.append(y, next_int, axis=1)
        del header_table[8:]
    y_mid = np.sum(y, axis=1) / m
    disper = np.zeros(N)
    for i in range(N):
        for j in range(m):
```

```

        disper[i] += (y[i][j] - y_mid[i]) ** 2
    disper[i] /= m

print("Матриця планування:")
table = []
for i in range(N):
    table.append([i+1, X1[i], X2[i], X3[i], X12[i], X13[i], X23[i], X123[i]])
    for j in range(m):
        table[i].append(y[i][j])
    table[i].append(y_mid[i])
    table[i].append(disper[i])
    if(i < m):
        header_table.append("Y" + str(i + 1))
header_table.append("Y")
header_table.append("S^2")
print(tabulate(table, headers=header_table, tablefmt="fancy_grid"))

b = [i for i in np.linalg.solve(list(zip(X00, X1, X2, X3, X12, X13, X23, X123))),
y_mid]]
print(f"y={b[0]:.3f} + {b[1]:.3f}*x1 + {b[2]:.3f}*x2 + {b[3]:.3f}*x3 +
{b[4]:.3f}*x1*x2 + {b[5]:.3f}*x1*x3 + {b[6]:.3f}*x2*x3 + {b[7]:.3f}*x1*x2*x3\n")

print("Критерій Кохрена")
Gp = max(disper) / sum(disper)
f1 = m - 1
f2 = N
fisher = f.isf(*[q / f2, f1, (f2 - 1) * f1])
Gt = round(fisher / (fisher + (f2 - 1)), 4)
print("Gp = " + str(Gp) + ", Gt = " + str(Gt))
if Gp > Gt:
    print("Дисперсія неоднорідна , потрібно збільшити m")
    m = m + 1
    continue

print("Gp < Gt -> Дисперсія однорідна\n")
print("Критерій Стюдента")
sb = sum(disper) / N
ssbs = sb / N * m
sbs = ssbs ** 0.5

beta = np.zeros(len(code_table[0]))
t_exp = []
for j in range(len(code_table[0])):
    for i in range(N):
        if(j == 0):
            beta[j] += y_mid[i]
        else:
            beta[j] += y_mid[i] * code_table[i][j]
    beta[j] /= N
    t_exp.append(abs(beta[j]) / sbs)

f3 = f1 * f2
ttabl = round(abs(t.ppf(q / 2, f3)), 4)

d = 8
for i in range(len(t_exp)):
    if (t_exp[i] < ttabl):
        print(f"Коефіцієнт b{i:} не значимий")
        b[i] = 0
        d = d - 1
yy1 = b[0] + b[1] * x1_min + b[2] * x2_min + b[3] * x3_min + b[4] * x1_min * x2_min +
b[5] * x1_min * x3_min + \

```

```

        b[6] * x2_min * x3_min + b[7] * x1_min * x2_min * x3_min
yy2 = b[0] + b[1] * x1_min + b[2] * x2_min + b[3] * x3_max + b[4] * x1_min * x2_min +
b[5] * x1_min * x3_max + \
        b[6] * x2_min * x3_max + b[7] * x1_min * x2_min * x3_max
yy3 = b[0] + b[1] * x1_min + b[2] * x2_max + b[3] * x3_min + b[4] * x1_min * x2_max +
b[5] * x1_min * x3_min + \
        b[6] * x2_max * x3_min + b[7] * x1_min * x2_max * x3_min
yy4 = b[0] + b[1] * x1_min + b[2] * x2_max + b[3] * x3_max + b[4] * x1_min * x2_max +
b[5] * x1_min * x3_max + \
        b[6] * x2_max * x3_max + b[7] * x1_min * x2_max * x3_max
yy5 = b[0] + b[1] * x1_max + b[2] * x2_min + b[3] * x3_min + b[4] * x1_max * x2_min +
b[5] * x1_max * x3_min + \
        b[6] * x2_min * x3_min + b[7] * x1_max * x2_min * x3_min
yy6 = b[0] + b[1] * x1_max + b[2] * x2_min + b[3] * x3_max + b[4] * x1_max * x2_min +
b[5] * x1_max * x3_max + \
        b[6] * x2_min * x3_max + b[7] * x1_max * x2_min * x3_max
yy7 = b[0] + b[1] * x1_max + b[2] * x2_max + b[3] * x3_min + b[4] * x1_max * x2_max +
b[5] * x1_max * x3_min + \
        b[6] * x2_max * x3_min + b[7] * x1_max * x2_max * x3_min
yy8 = b[0] + b[1] * x1_max + b[2] * x2_max + b[3] * x3_max + b[4] * x1_max * x2_max +
b[5] * x1_max * x3_max + \
        b[6] * x2_max * x3_max + b[7] * x1_max * x2_max * x3_max
print("Критерій Фішера")
print("Значимих коефіцієнтів(Фішера): ", d)
f4 = N - d
sad = ((yy1 - y_mid[0]) ** 2 + (yy2 - y_mid[1]) ** 2 + (yy3 - y_mid[2]) ** 2 + (yy4 -
y_mid[3]) ** 2 + (
        yy5 - y_mid[4]) ** 2 + (yy6 - y_mid[5]) ** 2 + (yy7 - y_mid[6]) ** 2 + (yy8 -
y_mid[7]) ** 2) * (m / (N - d))
Fp = sad / sb
print("Fp=", round(Fp, 2))

Ft = round(abs(f.isf(q, f4, f3)), 4)

print("Fp = " + str(round(Fp, 2)) + ", Ft = " + str(Ft))
if Fp > Ft:
    print("Fp > Ft -> Рівняння неадекватне оригіналу")
else:
    print("Fp < Ft -> Рівняння адекватне оригіналу")
break

```

Результат роботи програми

$$y=b_0+b_1*x_1+b_2*x_2+b_3*x_3+b_{12}*x_1*x_2+b_{13}*x_1*x_3+b_{23}*x_2*x_3+b_{123}*x_1*x_2*x_3$$

Кодовані значення X

№	x1	x2	x3	x1x2	x1x3	x2x3	x1x2x3
1	-1	-1	-1	1	1	1	-1
2	-1	-1	1	1	-1	-1	1
3	-1	1	-1	-1	1	-1	1
4	-1	1	1	-1	-1	1	-1
5	1	-1	-1	-1	-1	1	1
6	1	-1	1	-1	1	-1	-1
7	1	1	-1	1	-1	-1	-1
8	1	1	1	1	1	1	1

Матриця планування:

№	x1	x2	x3	x1x2	x1x3	x2x3	x1x2x3	Y1	Y2	Y3	Y4	Y	S^2
1	-10	20	50	-200	-500	1000	-10000	249	235	227	230	235.25	71.1875
2	-10	20	55	-200	-550	1100	-11000	229	243	236	224	233	51.5
3	-10	60	50	-600	-500	3000	-30000	224	225	247	239	233.75	93.6875
4	-10	60	55	-600	-550	3300	-33000	246	229	230	231	234	48.5
5	50	20	50	1000	2500	1000	50000	224	245	229	237	233.75	63.6875
6	50	20	55	1000	2750	1100	55000	237	250	237	230	238.5	52.25
7	50	60	50	3000	2500	3000	150000	249	250	221	250	242.5	154.25
8	50	60	55	3000	2750	3300	165000	236	248	242	236	240.5	24.75

$$y=250.521 + -2.048*x_1 + -0.234*x_2 + -0.312*x_3 + 0.043*x_1*x_2 + 0.039*x_1*x_3 + 0.005*x_2*x_3 + -0.001*x_1*x_2*x_3$$

Критерій Кохрена

$$Gr = 0.2755386848275092, Gt = 0.4377$$

Gr < Gt -> Дисперсія однорідна

Критерій Стюдента

Коефіцієнт b1 не значимий

Коефіцієнт b2 не значимий

Коефіцієнт b3 не значимий

Коефіцієнт b4 не значимий

Коефіцієнт b5 не значимий

Коефіцієнт b6 не значимий

Коефіцієнт b7 не значимий

Критерій Фішера

Значимих коефіцієнтів(Фішера): 1

$$Fr = 13.76$$

$$Fr = 13.76, Ft = 2.4226$$

Fr > Ft -> Рівняння неадекватне оригіналу