

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 №3

**«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»**

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, random

def print_matrix(matr):
    for i in range(len(matr)):
        print("{}.".format(i+1), end = "")
        for j in range(len(matr[i])):
            print("{:7}".format(matr[i][j]), end="")
        print()

def start(m, n, q):
    print("Матриця кодових значень")
    x_code = np.array([[+1, -1, -1, -1],
                       [+1, -1, +1, +1],
                       [+1, +1, -1, +1],
                       [+1, +1, +1, -1]])
    print_matrix(x_code)

    print("Матриця іксів:")
    x = np.array([[x1_min, x2_min, x3_min],
                  [x1_min, x2_max, x3_max],
                  [x1_max, x2_min, x3_max],
                  [x1_max, x2_max, x3_min]])
    print_matrix(x)

    print("Матриця ігриків:")
    y = np.random.randint(y_min, y_max, size=(n, m))
    print_matrix(y)

    print("Середні значення функцій відгуку:")
    y_mid = np.sum(y, axis = 1)/len(y[0])
    y1, y2, y3, y4 = y_mid
    print(f"y1 = {y1:.3f}\ny2 = {y2:.3f}\ny3 = {y3:.3f}\ny4 = {y4:.3f}")
    len(x)
    mx1, mx2, mx3 = [i / len(x) for i in np.sum(x, axis=0)]
    my = sum(y_mid) / len(y_mid)

    a1 = sum([x[i][0] * y_mid[i] for i in range(len(x))]) / len(x)
    a2 = sum([x[i][1] * y_mid[i] for i in range(len(x))]) / len(x)
    a3 = sum([x[i][2] * y_mid[i] for i in range(len(x))]) / len(x)

    a11 = sum([x[i][0] ** 2 for i in range(len(x))]) / len(x)
    a22 = sum([x[i][1] ** 2 for i in range(len(x))]) / len(x)
    a33 = sum([x[i][2] ** 2 for i in range(len(x))]) / len(x)
    a12 = a21 = sum([x[i][0] * x[i][1] for i in range(len(x))]) / len(x)
    a13 = a31 = sum([x[i][0] * x[i][2] for i in range(len(x))]) / len(x)
    a23 = a32 = sum([x[i][1] * x[i][2] for i in range(len(x))]) / len(x)

    det = np.linalg.det([[1, mx1, mx2, mx3],
                         [mx1, a11, a12, a13],
                         [mx2, a12, a22, a32],
                         [mx3, a13, a23, a33]])

    det0 = np.linalg.det([[my, mx1, mx2, mx3],
                          [a1, a11, a12, a13],
                          [a2, a12, a22, a32],
                          [a3, a13, a23, a33]])

    det1 = np.linalg.det([[1, my, mx2, mx3],
                          [mx1, a1, a12, a13],
                          [mx2, a2, a22, a32],
```

```

[mx3, a3, a23, a33]])

det2 = np.linalg.det([[1, mx1, my, mx3],
                      [mx1, a11, a1, a13],
                      [mx2, a12, a2, a32],
                      [mx3, a13, a3, a33]])

det3 = np.linalg.det([[1, mx1, mx2, my],
                      [mx1, a11, a12, a1],
                      [mx2, a12, a22, a2],
                      [mx3, a13, a23, a3]])

b0, b1, b2, b3 = det0 / det, det1 / det, det2 / det, det3 / det
b = [b0, b1, b2, b3]
print("Нормоване рівняння регресії:")
print("\ny = {0} + {1}*x1 + {2}*x2 + {3}*x3\n".format(round(b0, 5), round(b1, 5),
round(b2, 5), round(b3, 5)))

print("Перевірка:")
y1_exp = b0 + b1 * x[0][0] + b2 * x[0][1] + b3 * x[0][2]
y2_exp = b0 + b1 * x[1][0] + b2 * x[1][1] + b3 * x[1][2]
y3_exp = b0 + b1 * x[2][0] + b2 * x[2][1] + b3 * x[2][2]
y4_exp = b0 + b1 * x[3][0] + b2 * x[3][1] + b3 * x[3][2]
print(f"y1 = {b0:.3f} + {b1:.3f} * {x[0][0]} + {b2:.3f} * {x[0][1]} + {b3:.3f} * {x[0][2]} = {y1_exp:.3f}")
print(f"y2 = {b0:.3f} + {b1:.3f} * {x[1][0]} + {b2:.3f} * {x[1][1]} + {b3:.3f} * {x[1][2]} = {y2_exp:.3f}")
print(f"y3 = {b0:.3f} + {b1:.3f} * {x[2][0]} + {b2:.3f} * {x[2][1]} + {b3:.3f} * {x[2][2]} = {y3_exp:.3f}")
print(f"y4 = {b0:.3f} + {b1:.3f} * {x[3][0]} + {b2:.3f} * {x[3][1]} + {b3:.3f} * {x[3][2]} = {y4_exp:.3f}")

print("\nКритерій Кохрена")
f1, f2 = m - 1, n
s1 = sum([(i - y1) ** 2 for i in y[0]]) / m
s2 = sum([(i - y2) ** 2 for i in y[1]]) / m
s3 = sum([(i - y3) ** 2 for i in y[2]]) / m
s4 = sum([(i - y4) ** 2 for i in y[3]]) / m
s_arr = np.array([s1, s2, s3, s4])
g_p = max(s_arr) / sum(s_arr)

table = {3: 0.6841, 4: 0.6287, 5: 0.5892, 6: 0.5598, 7: 0.5365, 8: 0.5175, 9: 0.5017,
10: 0.4884, range(11, 17): 0.4366, range(17, 37): 0.3720, range(37, 145):
0.3093}
g_t = table.get(m)
if(g_p < g_t):
    print("Дисперсія однорідна: Gp = {g_p:.5} < Gt = {g_t}")
else:
    print("Дисперсія не однорідна Gp = {g_p:.5} > Gt = {g_t}")
    m = m + 1
    start(m + 1, n, q)
    return

print("\nКритерій Стюдента")
s2_b = s_arr.sum() / n
s2_beta_s = s2_b / (n * m)
s_beta_s = pow(s2_beta_s, 1 / 2)

beta0 = sum([x_code[i][0] * y_mid[i] for i in range(len(x_code))]) / n
beta1 = sum([x_code[i][1] * y_mid[i] for i in range(len(x_code))]) / n
beta2 = sum([x_code[i][2] * y_mid[i] for i in range(len(x_code))]) / n
beta3 = sum([x_code[i][3] * y_mid[i] for i in range(len(x_code))]) / n

```

```

t = [abs(beta0) / s_beta_s, abs(beta1) / s_beta_s, abs(beta2) / s_beta_s, abs(beta3) /
s_beta_s ]

f3 = f1 * f2
t_table = {8: 2.306, 9: 2.262, 10: 2.228, 11: 2.201, 12: 2.179, 13: 2.160, 14: 2.145,
15: 2.131, 16: 2.120,
17: 2.110, 18: 2.101, 19: 2.093, 20: 2.086, 21: 2.08, 22: 2.074, 23: 2.069,
24: 2.064, 25: 2.06}
d = 4 #кількість значимих коефіцієнтів
for i in range(len(t)):
    if(t_table.get(f3) > t[i]):
        b[i] = 0
        d -= 1

print(f"Рівняння регресії:\ny = {b[0]:.3f} + {b[1]:.3f} * x1 + {b[2]:.3f} * x2 +
{b[3]:.3f} * x3")
check0 = b[0] + b[1] * x[0][0] + b[2] * x[0][1] + b[3] * x[0][2]
check1 = b[0] + b[1] * x[1][0] + b[2] * x[1][1] + b[3] * x[1][2]
check2 = b[0] + b[1] * x[2][0] + b[2] * x[2][1] + b[3] * x[2][2]
check3 = b[0] + b[1] * x[3][0] + b[2] * x[3][1] + b[3] * x[3][2]
ckeck_list = [check0, check1, check2, check3]
print("Значення у нормованих: ", ckeck_list)

print("\nКритерій Фішера")
f4 = n - d
s2_ad = m / f4 * sum([(ckeck_list[i] - y_mid[i]) ** 2 for i in range(len(y_mid))])
f_p = s2_ad / s2_b
f_t = [[164.4, 199.5, 215.7, 224.6, 230.2, 234], [18.5, 19.2, 19.2, 19.3, 19.3, 19.3],
[10.1, 9.6, 9.3, 9.1, 9, 8.9], [7.7, 6.9, 6.6, 6.4, 6.3, 6.2], [6.6, 5.8, 5.4,
5.2, 5.1, 5],
[6, 5.1, 4.8, 4.5, 4.4, 4.3], [5.5, 4.7, 4.4, 4.1, 4, 3.9], [5.3, 4.5, 4.1,
3.8, 3.7, 3.6],
[5.1, 4.3, 3.9, 3.6, 3.5, 3.4], [5, 4.1, 3.7, 3.5, 3.3, 3.2], [4.8, 4, 3.6,
3.4, 3.2, 3.1],
[4.8, 3.9, 3.5, 3.3, 3.1, 3], [4.7, 3.8, 3.4, 3.2, 3, 2.9], [4.6, 3.7, 3.3,
3.1, 3, 2.9],
[4.5, 3.7, 3.3, 3.1, 2.9, 2.8], [4.5, 3.6, 3.2, 3, 2.9, 2.7], [4.5, 3.6, 3.2,
3, 2.8, 2.7],
[4.4, 3.6, 3.2, 2.9, 2.8, 2.7], [4.4, 3.5, 3.1, 2.9, 2.7, 2.6], [4.4, 3.5,
3.1, 2.9, 2.7, 2.6]]
if(f_p > f_t[f3][f4]):
    print(f"f_p = {f_p} > f_t = {f_t[f3][f4]}.\nМатематична модель не адекватна
експериментальним даним")
else: print(f"f_p = {f_p} < f_t = {f_t}.\nМатематична модель адекватна експериментальним
даним")

x1_min = -10
x1_max = 50
x2_min = 20
x2_max = 60
x3_min = 50
x3_max = 55

x_mid_max = (x1_max + x2_max + x3_max) / 3
x_mid_min = (x1_min + x2_min + x3_min) / 3
y_max = 200 + x_mid_max
y_min = 200 + x_mid_min

m = 3
n = 4
q = 0.5
print("Рівняння регресії")
print("y = b0 + b1*x1 + b2*x2 + b3*x3")
start(m, n, q)

```

Результат роботи програми

Рівняння регресії

$$y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3$$

Матриця кодових значень

1.	1	-1	-1	-1
2.	1	-1	1	1
3.	1	1	-1	1
4.	1	1	1	-1

Матриця іксів:

1.	-10	20	50
2.	-10	60	55
3.	50	20	55
4.	50	60	50

Матриця ігриків:

1.	251	253	234
2.	232	249	240
3.	235	235	226
4.	254	241	252

Середні значення функцій відгуку:

$$y_1 = 246.000$$

$$y_2 = 240.333$$

$$y_3 = 232.000$$

$$y_4 = 249.000$$

Нормоване рівняння регресії:

$$y = 356.05556 + -0.04444 \cdot x_1 + 0.14167 \cdot x_2 + -2.26667 \cdot x_3$$

Перевірка:

$$y_1 = 356.056 + -0.044 \cdot -10 + 0.142 \cdot 20 + -2.267 \cdot 50 = 246.000$$

$$y_2 = 356.056 + -0.044 \cdot -10 + 0.142 \cdot 60 + -2.267 \cdot 55 = 240.333$$

$$y_3 = 356.056 + -0.044 \cdot 50 + 0.142 \cdot 20 + -2.267 \cdot 55 = 232.000$$

$$y_4 = 356.056 + -0.044 \cdot 50 + 0.142 \cdot 60 + -2.267 \cdot 50 = 249.000$$

Критерій Кохрена

$$\text{Дисперсія однорідна: } G_p = 0.42358 < G_t = 0.6841$$

Критерій Стюдента

Рівняння регресії:

$$y = 356.056 + 0.000 \cdot x_1 + 0.000 \cdot x_2 + -2.267 \cdot x_3$$

Значення у нормованих: [242.72222222223826, 231.38888888890642, 231.38888888890642, 242.72222222223826]

Критерій Фішера

$$f_p = 4.565198618284995 > f_t = 3.9.$$

Математична модель не адекватна експериментальним даним