

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”

Виконав:  
студент групи ІВ-83  
Щебетін Б. Ю.  
Перевірив:  
ас. Регіда П.Г.

Київ  
2020 р.

**Мета:** провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.

Номер у списку: 22.

Варіант завдання: 322.

№ <sub>варіанта</sub>	X <sub>1</sub>		X <sub>2</sub>		X <sub>3</sub>	
	min	max	min	max	min	max
322	10	40	30	80	10	20

### 1. Лістинг програми:

```
import numpy as np

x1_min = 10
x1_max = 40
x2_min = 30
x2_max = 80
x3_min = 10
x3_max = 20

x_aver_max = (x1_max + x2_max + x3_max) / 3
x_aver_min = (x1_min + x2_min + x3_min) / 3
y_max = 200 + x_aver_max
y_min = 200 + x_aver_min

m, n = 3, 4

def main(m, n):
    print("\nМатрица кодовых значений")
    norm_x = np.array([
        [+1, -1, -1, -1],
        [+1, -1, +1, +1],
        [+1, +1, -1, +1],
        [+1, +1, +1, -1]
    ])
    for i in range(len(norm_x)):
        print("{}.".format(i + 1), end="")
        for j in range(len(norm_x[i])):
            print("{:4}".format(norm_x[i][j]), end="")
        print()

    print("\nX-матрица:")
    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]
    ])

    for i in range(len(x)):
        print("{}.".format(i + 1), end="")
        for j in range(len(x[i])):
            print("{:4}".format(x[i][j]), end="")
        print()

    print("\nY-матрица:")
    y = np.random.randint(y_min, y_max, size=(n, m))

    print("\nСреднее из характеристик ответа:")
    y_av = np.sum(y, axis=1) / len(y[0])
    y_1, y_2, y_3, y_4 = y_av
    print(f"y_1 = {y_1:.2f}\ny_2 = {y_2:.2f}\ny_3 = {y_3:.2f}\ny_4 = {y_4:.2f}")
    mx_1, mx_2, mx_3 = [i / len(x) for i in np.sum(x, axis=0)]
    my = sum(y_av) / len(y_av)

    a_1 = sum([x[i][0] * y_av[i] for i in range(len(x))]) / len(x)
    a_2 = sum([x[i][1] * y_av[i] for i in range(len(x))]) / len(x)
    a_3 = sum([x[i][2] * y_av[i] for i in range(len(x))]) / len(x)

    a_11 = sum([x[i][0] ** 2 for i in range(len(x))]) / len(x)
    a_22 = sum([x[i][1] ** 2 for i in range(len(x))]) / len(x)
    a_33 = sum([x[i][2] ** 2 for i in range(len(x))]) / len(x)
    a_12 = sum([x[i][0] * x[i][1] for i in range(len(x))]) / len(x)
    a_13 = sum([x[i][0] * x[i][2] for i in range(len(x))]) / len(x)
    a_23 = a_32 = sum([x[i][1] * x[i][2] for i in range(len(x))]) / len(x)

    det = np.linalg.det(
        [[1, mx_1, mx_2, mx_3], [mx_1, a_11, a_12, a_13], [mx_2, a_12, a_22,
a_32], [mx_3, a_13, a_23, a_33]])
    det_0 = np.linalg.det(
        [[my, mx_1, mx_2, mx_3], [a_1, a_11, a_12, a_13], [a_2, a_12, a_22,
a_32], [a_3, a_13, a_23, a_33]])
    det_1 = np.linalg.det(
        [[1, my, mx_2, mx_3], [mx_1, a_1, a_12, a_13], [mx_2, a_2, a_22,
a_32], [mx_3, a_3, a_23, a_33]])
    det_2 = np.linalg.det(
        [[1, mx_1, my, mx_3], [mx_1, a_11, a_1, a_13], [mx_2, a_12, a_2,
a_32], [mx_3, a_13, a_3, a_33]])
    det_3 = np.linalg.det(
        [[1, mx_1, mx_2, my], [mx_1, a_11, a_12, a_1], [mx_2, a_12, a_22,
a_2], [mx_3, a_13, a_23, a_3]])

    b_0 = det_0 / det
    b_1 = det_1 / det
    b_2 = det_2 / det
    b_3 = det_3 / det
    b = [b_0, b_1, b_2, b_3]

    print(f"\nНормализованное уравнение регрессии: y = {b_0:.5f} + {b_1:.5f}
* x1 + {b_2:.5f} * x2 + {b_3:.5f} * x3\n")
    print("Аудит:")
    y_1_exp = b_0 + b_1 * x[0][0] + b_2 * x[0][1] + b_3 * x[0][2]
    y_2_exp = b_0 + b_1 * x[1][0] + b_2 * x[1][1] + b_3 * x[1][2]
    y_3_exp = b_0 + b_1 * x[2][0] + b_2 * x[2][1] + b_3 * x[2][2]
    y_4_exp = b_0 + b_1 * x[3][0] + b_2 * x[3][1] + b_3 * x[3][2]
    print(f"y_1 = {b_0:.3f} + {b_1:.3f} * {x[0][0]} + {b_2:.3f} * {x[0][1]} +
{b_3:.3f} * {x[0][2]} = {y_1_exp:.3f}")
    print(f"\ny_2 = {b_0:.3f} + {b_1:.3f} * {x[1][0]} + {b_2:.3f} * {x[1][1]}")

```

```

+ {b_3:.3f} * {x[1][2]} = {y_2_exp:.3f}"
    f"\ny_3 = {b_0:.3f} + {b_1:.3f} * {x[2][0]} + {b_2:.3f} * {x[2][1]}
+ {b_3:.3f} * {x[2][2]} = {y_3_exp:.3f}"
    f"\ny_4 = {b_0:.3f} + {b_1:.3f} * {x[3][0]} + {b_2:.3f} * {x[3][1]}
+ {b_3:.3f} * {x[3][2]} = {y_4_exp:.3f}")

print("\n[ Тест Керена ]")
f_1 = m - 1
f_2 = n
s_1 = sum([(i - y_1) ** 2 for i in y[0]]) / m
s_2 = sum([(i - y_2) ** 2 for i in y[1]]) / m
s_3 = sum([(i - y_3) ** 2 for i in y[2]]) / m
s_4 = sum([(i - y_4) ** 2 for i in y[3]]) / m
s_array = np.array([s_1, s_2, s_3, s_4])
gP = max(s_array) / sum(s_array)

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}
gT = table.get(m)

if (gP < gT):
    print(f"Дисперсия однородна: Gp = {gP:.5} < Gt = {gT}")
else:
    print(f"Дисперсия не является однородной Gp = {gP:.5} < Gt = {gT}")
    m = m + 1
    main(m + 1, n, q)
    return

print("\n[ Студенческий тест ]")
s2_B = s_array.sum() / n
s2_beta_S = s2_B / (n * m)
s_beta_S = pow(s2_beta_S, 1 / 2)

beta_0 = sum([norm_x[i][0] * y_av[i] for i in range(len(norm_x))]) / n
beta_1 = sum([norm_x[i][1] * y_av[i] for i in range(len(norm_x))]) / n
beta_2 = sum([norm_x[i][2] * y_av[i] for i in range(len(norm_x))]) / n
beta_3 = sum([norm_x[i][3] * y_av[i] for i in range(len(norm_x))]) / n

t = [abs(beta_0) / s_beta_S, abs(beta_1) / s_beta_S, abs(beta_2) /
s_beta_S, abs(beta_3) / s_beta_S]

f3 = f_1 * f_2
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

for i in range(len(t)):
    if (t_table.get(f3) > t[i]):
        b[i] = 0
        d -= 1

print(f"Уравнение регрессии: y = {b[0]:.3f} + {b[1]:.3f} * x1 +
{b[2]:.3f} * x2 + {b[3]:.3f} * x3")
check_0 = b[0] + b[1] * x[0][0] + b[2] * x[0][1] + b[3] * x[0][2]
check_1 = b[0] + b[1] * x[1][0] + b[2] * x[1][1] + b[3] * x[1][2]
check_2 = b[0] + b[1] * x[2][0] + b[2] * x[2][1] + b[3] * x[2][2]
check_3 = b[0] + b[1] * x[3][0] + b[2] * x[3][1] + b[3] * x[3][2]
ckeck_list = [check_0, check_1, check_2, check_3]
print("нормализованные значения: ", ckeck_list)

```

```

print("\n[ Тест Фишера ]")
f_4 = n - d
s2_ad = m / f_4 * sum([(ckeck_list[i] - y_av[i]) ** 2 for i in
range(len(y_av))])
fP = s2_ad / s2_B
fT = [
    [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 (fP > fT[f3][f_4]):
    print("fp = {fP} > ft = {fT[f3][f_4]}. \nМатематическая модель не
адекватна экспериментальным данным\n")
else:
    print("fP = {fP} < fT = {fT[f3][f_4]}. \nМатематическая модель
адекватна экспериментальным данным\n")

print("\nУравнение регрессии --- y = b_0 + b_1 * x1 + b_1 * x2 + b_3 * x3")
main(m, n)

```

## Результати виконання роботи

```
Уравнение регрессии ---  $y = b_0 + b_1 * x_1 + b_1 * x_2 + b_3 * x_3$ 

Матрица кодовых значений
1.  1  -1  -1  -1
2.  1  -1   1   1
3.  1   1  -1   1
4.  1   1   1  -1

X-матрица:
1.  10  30  10
2.  10  80  20
3.  40  30  20
4.  40  80  10

Y-матрица:

Среднее из характеристик ответа:
y_1 = 230.00
y_2 = 221.33
y_3 = 226.33
y_4 = 234.00

Нормализованное уравнение регрессии:  $y = 236.96667 + 0.15000 * x_1 + -0.01000 * x_2 + -0.81667 * x_3$ 

Аудит:
y_1 = 236.967 + 0.150 * 10 + -0.010 * 30 + -0.817 * 10 = 230.000
y_2 = 236.967 + 0.150 * 10 + -0.010 * 80 + -0.817 * 20 = 221.333
y_3 = 236.967 + 0.150 * 40 + -0.010 * 30 + -0.817 * 20 = 226.333
y_4 = 236.967 + 0.150 * 40 + -0.010 * 80 + -0.817 * 10 = 234.000

[ Тест Керена ]
Дисперсия однородна:  $G_p = 0.473 < G_t = 0.6841$ 

[ Студенческий тест ]
Уравнение регрессии:  $y = 236.967 + 0.150 * x_1 + -0.010 * x_2 + -0.817 * x_3$ 
Уравнение регрессии:  $y = 236.967 + 0.000 * x_1 + -0.010 * x_2 + -0.817 * x_3$ 
Уравнение регрессии:  $y = 236.967 + 0.000 * x_1 + 0.000 * x_2 + -0.817 * x_3$ 
Уравнение регрессии:  $y = 236.967 + 0.000 * x_1 + 0.000 * x_2 + 0.000 * x_3$ 
Нормализованные значения : [236.96666666666601, 236.96666666666601, 236.96666666666601, 236.96666666666601]

[ Тест Фишера ]
 $f_p = 6.5039024390236415 > f_t = 3.6.$ 
Математическая модель не адекватна экспериментальным данным

Process finished with exit code 0
```