

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 № 5

з дисципліни «Методи наукових досліджень»

на тему «Проведення трьохфакторного експерименту при використанні  
рівняння регресії з урахуванням квадратичних членів (центральный  
ортогональний композиційний план).»

Виконав:  
студент 2 курсу  
групи ІВ-91  
Охочий Р.О.  
Номер у списку групи: 21

ПЕРЕВІРИВ:  
ас. Регіда П.Г.

## Хід роботи

**Мета:** Провести трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Знайти рівняння регресії, яке буде адекватним для опису об'єкту.

### Завдання:

#### Завдання

1. Взяти рівняння з урахуванням квадратичних членів.
2. Скласти матрицю планування для ОЦКП
3. Провести експеримент у всіх точках факторного простору (знайти значення функції відгуку Y). Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі. Варіанти вибираються по номеру в списку в журналі викладача.

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

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

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

4. Розрахувати коефіцієнти рівняння регресії і записати його.
5. Провести 3 статистичні перевірки.

121	-3	6	0	10	-7	10
-----	----	---	---	----	----	----

### Код програми

```
import random
import numpy as np
from sklearn import linear_model
from scipy.stats import f, t

def Lab5(k):
    m = k
    n = 15
    d = 0

    X1min = -3
    X1max = 6
    X2min = 0
    X2max = 10
    X3min = -7
    X3max = 10

    print("x1 min =", X1min, ", x1 max =", X1max)
    print("x2 min =", X2min, ", x2 max =", X2max)
    print("x3 min =", X3min, ", x3 max =", X3max)
    print("-----")

    Ymax = 200 + (X1max + X2max + X3max) / 3
    Ymin = 200 + (X1min + X2min + X3min) / 3
    print("Y min =", Ymin)
    print("Y max =", Ymax)
    print("-----")

    xn = [[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
```

```

        [-1, -1, -1, -1, 1, 1, 1, 1, -1.215, 1.215, 0, 0, 0, 0, 0],
        [-1, -1, 1, 1, -1, -1, 1, 1, 0, 0, -1.215, 1.215, 0, 0, 0],
        [-1, 1, -1, 1, -1, 1, -1, 1, 0, 0, 0, 0, -1.215, 1.215, 0]]

x1x2_norm = [0]*15
x1x3_norm = [0]*15
x2x3_norm = [0]*15
x1x2x3_norm = [0]*15
x1kv_norm = [0]*15
x2kv_norm = [0]*15
x3kv_norm = [0]*15
for i in range(15):
    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]
    x1kv_norm[i] = xn[1][i] ** 2
    x2kv_norm[i] = xn[2][i] ** 2
    x3kv_norm[i] = xn[3][i] ** 2

Y = [[random.randint(int(Ymin), int(Ymax)) for i in range(m)] for j in range(15)]
print("Матриця планування Y:")
for i in range(len(Y)):
    print(Y[i])
print("-----")

x01 = (X1max + X1min) / 2
x02 = (X2max + X2min) / 2
x03 = (X3max + X3min) / 2
deltaX1 = X1max - x01
deltaX2 = X2max - x02
deltaX3 = X3max - x03
x0 = [1] * 15
x1 = [-5, -5, -5, -5, 4, 4, 4, 4, -1.215 * deltaX1 + x01, 1.215 * deltaX1 + x01, x01,
x01, x01, x01, x01]
x2 = [-2, -2, 7, 7, -2, -2, 7, 7, x02, x02, -1.215 * deltaX2 + x02, 1.215 * deltaX2 +
x02, x02, x02, x02]
x3 = [-1, 2, -1, 2, -1, 2, -1, 2, x03, x03, x03, x03, -1.215 * deltaX3 + x03, 1.215 *
deltaX3 + x03, x03]
x1x2 = [0] * 15
x1x3 = [0] * 15
x2x3 = [0] * 15
x1x2x3 = [0] * 15
x1kv = [0] * 15
x2kv = [0] * 15
x3kv = [0] * 15
for i in range(15):
    x1x2[i] = round(x1[i] * x2[i], 3)
    x1x3[i] = round(x1[i] * x3[i], 3)
    x2x3[i] = round(x2[i] * x3[i], 3)
    x1x2x3[i] = round(x1[i] * x2[i] * x3[i], 3)
    x1kv[i] = round(x1[i] ** 2, 3)
    x2kv[i] = round(x2[i] ** 2, 3)
    x3kv[i] = round(x3[i] ** 2, 3)

Y_average = []
for i in range(len(Y)):
    Y_average.append(np.mean(Y[i], axis=0))
    Y_average = [round(i,3) for i in Y_average]

list_b = list(zip(xn[0], xn[1], xn[2], xn[3], x1x2_norm, x1x3_norm, x2x3_norm,
x1x2x3_norm, x1kv_norm, x2kv_norm, x3kv_norm))

```

```

list_a = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1kv, x2kv, x3kv))

print("Матриця планування з нормованими коефіцієнтами X:")
for i in range(15):
    print(list_b[i])
print("-----")

skm = linear_model.LinearRegression(fit_intercept=False)
skm.fit(list_b, Y_average)
b = skm.coef_
b = [round(i, 3) for i in b]

print("Рівняння регресії зі знайденими коефіцієнтами: \n" "y = {} + {}*x1 + {}*x2 +
{}*x3 + {}*x1x2 + {}*x1x3 + {}*x2x3 + {}*x1x2x3 {}*x1^2 + {}*x2^2 + {}*x3^2".format(b[0],
b[1], b[2], b[3], b[4], b[5], b[6], b[7], b[8], b[9], b[10]))
print("-----")

print("Перевірка однорідності за критерієм Кохрена")
print("-----")
print("Середні значення відгуку за рядками:")
for i in range(len(Y_average)):
    print(Y_average[i], end=', ')
print()
print("-----")

dispersions = []
for i in range(len(Y)):
    a = 0
    for k in Y[i]:
        a += (k - np.mean(Y[i], axis=0)) ** 2
    dispersions.append(a / len(Y[i]))

Gp = max(dispersions) / sum(dispersions)
Gt = 0.3346
if Gp < Gt:
    print("Дисперсія однорідна")
    print("-----")
else:
    print("Дисперсія неоднорідна")
    print("-----")
    m += 1
    Lab5(m)

print("Оцінка значимості коефіцієнтів за критерієм Стюдента")
print("-----")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (15 * m)) ** 0.5

t_list = [abs(b[i]) / sbs for i in range(0, 11)]

res = [0] * 11
S1 = []
S2 = []
F3 = (m - 1) * n

for i in range(11):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        S2.append(b[i])
        res[i] = 0
    else:
        S1.append(b[i])
        res[i] = b[i]

```

```

        d += 1

    print("Значущі коефіцієнти регресії:", S1)
    print("-----")
    print("Незначущі коефіцієнти регресії:", S2)
    print("-----")

    y_st = []
    for i in range(15):
        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] + res[8] * x1kv_norm[i] + res[9] * x2kv_norm[i] + res[10] * x3kv_norm[i])

    print("Значення з отриманими коефіцієнтами:")

    for i in range(len(y_st)):
        print(y_st[i], end=', ')
    print()
    print("-----")

    print("Перевірка на адекватність за критерієм Фішера")
    print("-----")
    Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(15)]) / (n - d)
    Fp = Sad / sb
    F4 = n - d
    print("Fp =", Fp)
    print("-----")

    if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
        print("Рівняння регресії адекватне при рівні значимості 0.05")
        print("-----")
    else:
        print("Рівняння регресії неадекватне при рівні значимості 0.05")
        print("-----")

```

## Результати роботи програми

```
x1 min = -3 , x1 max = 6
x2 min = 0 , x2 max = 10
x3 min = -7 , x3 max = 10
```

```
Y min = 196.66666666666666
Y max = 208.66666666666666
```

Матриця планування Y:

```
[200, 202, 199]
[208, 199, 205]
[207, 196, 196]
[203, 200, 200]
[198, 197, 201]
[208, 197, 208]
[208, 205, 207]
[204, 203, 197]
[200, 201, 197]
[198, 202, 203]
[201, 199, 207]
[203, 199, 206]
[196, 203, 201]
[197, 204, 206]
[207, 197, 204]
```

Матриця планування з нормованими коефіцієнтами X:

```
(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, 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, 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, -1.215, 0, 0, -0.0, -0.0, 0, -0.0, 1.4762250000000001, 0, 0)
(1, 1.215, 0, 0, 0.0, 0.0, 0, 0.0, 1.4762250000000001, 0, 0)
(1, 0, -1.215, 0, -0.0, 0, -0.0, -0.0, 0, 1.4762250000000001, 0)
(1, 0, 1.215, 0, 0.0, 0, 0.0, 0.0, 0, 1.4762250000000001, 0)
(1, 0, 0, -1.215, 0, -0.0, -0.0, -0.0, 0, 0, 1.4762250000000001)
(1, 0, 0, 1.215, 0, 0.0, 0.0, 0.0, 0, 0, 1.4762250000000001)
(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
```

Рівняння регресії зі знайденими коефіцієнтами:

```
y = 201.482 + 0.733*x1 + 0.159*x2 + 0.746*x3 + 1.083*x1x2 + -0.583*x1x3 + -1.667*x2x3 + -1.083*x1x2x3 -0.628*x1^2 + 0.953*x2^2 + 0.049*x3^2
```

Перевірка однорідності за критерієм Кохрена

Середні значення відгуку за рядками:

```
200.333, 204.0, 199.667, 201.0, 198.667, 204.333, 206.667, 201.333, 199.333, 201.0, 202.333, 202.667, 200.0, 202.333, 202.667,
```

Дисперсія однорідна

Оцінка значимості коефіцієнтів за критерієм Стюдента

Значущі коефіцієнти регресії: [201.482, 1.083, -1.667, -1.083]

Незначущі коефіцієнти регресії: [0.733, 0.159, 0.746, -0.583, -0.628, 0.953, 0.049]

Значення з отриманими коефіцієнтами:

```
201.981, 203.149, 200.983, 199.815, 197.649, 203.149, 205.315, 199.815, 201.482, 201.482, 201.482, 201.482, 201.482, 201.482, 201.482,
```

Перевірка на адекватність за критерієм Фішера

Fp = 0.6504446581056229

Рівняння регресії адекватне при рівні значимості 0.05