

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»  
**Проведення трьохфакторного експерименту при використанні рівняння  
регресії з урахуванням квадратичних членів  
(центральний ортогональний композиційний план)**

ВИКОНАВ:  
Студент II курсу ФІОТ  
Групи ІО-91  
Павлюк Владислав Володимирович  
Номер заліковки: 9120  
Номер у списку: 17

ПЕРЕВІРИВ:  
ас. Регіда П. Г.

Київ 2021 р.

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

### 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.

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

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

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

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

### Виконання роботи

№_варіанта	X <sub>1</sub>		X <sub>2</sub>		X <sub>3</sub>	
	min	max	min	max	min	max
117	-7	10	-4	8	-5	4

### Код програми

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

m = 3
x1_min = -7
x1_max = 10
x2_min = -4
x2_max = 8
x3_min = -5
x3_max = 4
print("x1_min = "+str(x1_min), "x2_min = "+str(x2_min), "x3_min = "+str(x3_min))
print("x1_max = "+str(x1_max), "x2_max = "+str(x2_max), "x3_max = "+str(x3_max))
print("")
```

```

y_min = 200 + (x1_min + x2_min + x3_min) / 3
y_max = 200 + (x1_max + x2_max + x3_max) / 3
print("y_min = "+str(y_min), "y_max = "+str(y_max))
print("")

mp = [[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]]
pprint(mp)
print("")

x1x2_norm = [0] * 15
x1x3_norm = [0] * 15
x2x3_norm = [0] * 15
x1x2x3_norm = [0] * 15
x1x1_norm = [0] * 15
x2x2_norm = [0] * 15
x3x3_norm = [0] * 15

for i in range(15):
    x1x2_norm[i] = mp[1][i] * mp[2][i]
    x1x3_norm[i] = mp[1][i] * mp[3][i]
    x2x3_norm[i] = mp[2][i] * mp[3][i]
    x1x2x3_norm[i] = mp[1][i] * mp[2][i] * mp[3][i]
    x1x1_norm[i] = round(mp[1][i] ** 2, 3)
    x2x2_norm[i] = round(mp[2][i] ** 2, 3)
    x3x3_norm[i] = round(mp[3][i] ** 2, 3)

Y = [[random.randint(int(y_min), int(y_max)) for i in range(m)] for j in
range(15)]
print("Матриця планування Y")
pprint(Y)
print("")

x1_aver = (x1_max + x1_min) / 2
x2_aver = (x2_max + x2_min) / 2
x3_aver = (x3_max + x3_min) / 2
x1_delt = x1_max - x1_aver
x2_delt = x2_max - x2_aver
x3_delt = x3_max - x3_aver

x0 = [1] * 15
x1 = [-7, -7, -7, -7, 10, 10, 10, 10, -1.215 * x1_delt + x1_aver, 1.215 *
x1_delt + x1_aver, x1_aver, x1_aver, x1_aver, x1_aver, x1_aver]
x2 = [-4, -4, 8, 8, -4, -4, 8, 8, x2_aver, x2_aver, -1.215 * x2_delt + x2_aver,
1.215 * x2_delt + x2_aver, x2_aver, x2_aver, x2_aver]
x3 = [-5, 4, -5, 4, -5, 4, -5, 4, x3_aver, x3_aver, x3_aver, x3_aver, x3_aver, -1.215 *
x3_delt + x3_aver, 1.215 * x3_delt + x3_aver, x3_aver]

x1x2 = [0] * 15
x1x3 = [0] * 15
x2x3 = [0] * 15
x1x2x3 = [0] * 15
x1x1 = [0] * 15
x2x2 = [0] * 15
x3x3 = [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)
    x1x1[i] = round(x1[i] ** 2, 3)

```

```

x2x2[i] = round(x2[i] ** 2, 3)
x3x3[i] = round(x3[i] ** 2, 3)

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

list_for_b = list(zip(mp[0], mp[1], mp[2], mp[3], x1x2_norm, x1x3_norm,
x2x3_norm, x1x2x3_norm, x1x1_norm, x2x2_norm, x3x3_norm))
list_for_a = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1x1, x2x2,
x3x3))
print("Матриця планування з нормованими коефіцієнтами X")
pprint(list_for_b)
print("")

skm = linear_model.LinearRegression(fit_intercept=False)
skm.fit(list_for_b, aver_y)
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 {}*x1x1 + {}*x2x2 + {}*x3x3".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("\033[1m\033[30m\033[43m{}\033[0m".format("Перевірка за критерієм
Кохрена"))
print("Середні значення відгуку за рядками\n", +aver_y[0], aver_y[1], aver_y[2],
aver_y[3],
    aver_y[4], aver_y[5], aver_y[6], aver_y[7], aver_y[8], aver_y[9],
aver_y[10],
    aver_y[11], aver_y[12], aver_y[13], aver_y[14])
disp = []
for i in range(len(Y)):
    a = 0
    for k in Y[i]:
        a += (k - np.mean(Y[i], axis=0)) ** 2
    disp.append(a / len(Y[i]))

Gp = max(disp) / sum(disp)
Gt = 0.3346
if Gp < Gt:
    print("\033[1m\033[30m\033[42m{}\033[0m".format("Дисперсія однорідна"))
else:
    print("\033[1m\033[30m\033[41m{}\033[0m".format("Дисперсія неоднорідна"))
print("")

print("\033[1m\033[30m\033[43m{}\033[0m".format("Перевірка значущості
коефіцієнтів за критерієм Стьюдента"))
sb = sum(disp) / len(disp)
sbs = (sb / (15 * m)) ** 0.5

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

d = 0
res = [0] * 11
coef_1 = []
coef_2 = []
n = 15
F3 = (m - 1) * n
for i in range(11):
    if t_list[i] >= t.ppf(q=0.975, df=F3):

```

```

        coef_1.append(b[i])
        res[i] = b[i]
        d += 1
    else:
        coef_2.append(b[i])
        res[i] = 0
print("Значущі коефіцієнти регресії\n", coef_1)
print("Незначущі коефіцієнти регресії\n", coef_2)

y_st = []
for i in range(15):
    y_st.append(res[0] + res[1] * mp[1][i] + res[2] * mp[2][i] + res[3] *
mp[3][i] + res[4] * x1x2_norm[i] + res[5] *
        x1x3_norm[i] + res[6] * x2x3_norm[i] + res[7] * x1x2x3_norm[i] +
res[8] * x1x1_norm[i] + res[9] *
        x2x2_norm[i] + res[10] * x3x3_norm[i])
print("Значення з отриманими коефіцієнтами:\n", y_st)
print("")

print("\033[1m\033[30m\033[43m{}\033[0m".format("Перевірка адекватності за
критерієм Фішера"))
Sad = m * sum([(y_st[i] - aver_y[i]) ** 2 for i in range(15)]) / (n - d)
Fp = Sad / sb
F4 = n - d
print("Fp =", Fp)
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("\033[1m\033[30m\033[42m{}\033[0m".format("При рівні значимості 0.05
рівняння регресії адекватне"))
else:
    print("\033[1m\033[30m\033[41m{}\033[0m".format("При рівні значимості 0.05
рівняння регресії неадекватне"))

```

### Результат роботи програми

```
C:\Users\38096\anaconda3\python.exe "C:/Z Крі/А МНД/Лаб/Лаб 5/main.py"
```

```
x1_min = -7 x2_min = -4 x3_min = -5
```

```
x1_max = 10 x2_max = 8 x3_max = 4
```

```
y_min = 194.66666666666666 y_max = 207.33333333333334
```

```
[[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]]
```

Матриця планування Y

```
[[199, 195, 198],  
 [197, 205, 201],  
 [202, 200, 197],  
 [196, 206, 200],  
 [207, 204, 199],  
 [204, 198, 194],  
 [204, 203, 199],  
 [201, 198, 198],  
 [203, 201, 204],  
 [197, 194, 207],  
 [196, 197, 196],  
 [194, 198, 197],  
 [198, 199, 200],  
 [196, 203, 207],  
 [203, 207, 201]]
```

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, 1, 1, 1, 1, 1, 1),  
 (1, -1.215, 0, 0, -0.0, -0.0, 0, -0.0, 1.476, 0, 0),  
 (1, 1.215, 0, 0, 0.0, 0.0, 0, 0.0, 1.476, 0, 0),  
 (1, 0, -1.215, 0, -0.0, 0, -0.0, -0.0, 0, 1.476, 0),  
 (1, 0, 1.215, 0, 0.0, 0, 0.0, 0.0, 0, 1.476, 0),  
 (1, 0, 0, -1.215, 0, -0.0, -0.0, -0.0, 0, 0, 1.476),
```

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, 1.476, 0, 0),
 (1, 1.215, 0, 0, 0.0, 0.0, 0, 0.0, 1.476, 0, 0),
 (1, 0, -1.215, 0, -0.0, 0, -0.0, -0.0, 0, 1.476, 0),
 (1, 0, 1.215, 0, 0.0, 0, 0.0, 0.0, 0, 1.476, 0),
 (1, 0, 0, -1.215, 0, -0.0, -0.0, -0.0, 0, 0, 1.476),
 (1, 0, 0, 1.215, 0, 0.0, 0.0, 0.0, 0, 0, 1.476),
 (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)]
```

Рівняння регресії зі знайденими коефіцієнтами:

$y = 200.666 + 0.026 \cdot x_1 + 0.091 \cdot x_2 + 0.059 \cdot x_3 - 0.375 \cdot x_1 x_2 - 1.542 \cdot x_1 x_3 - 0.125 \cdot x_2 x_3 + 0.542 \cdot x_1 x_2 x_3 - 0.893 \cdot x_1 x_1 - 2.269 \cdot x_2 x_2 + 0.554 \cdot x_3 x_3$

Перевірка за критерієм Кохрена

Середні значення відгуку за рядками

197.333 201.0 199.667 200.667 203.333 198.667 202.0 199.0 202.667 199.333 196.333 196.333 199.0 202.0 203.667

дисперсія однорідна

Перевірка значущості коефіцієнтів за критерієм Стьюдента

Значущі коефіцієнти регресії

[200.666, -1.542, -2.269]

Незначущі коефіцієнти регресії

[0.026, 0.091, 0.059, -0.375, -0.125, 0.542, 0.893, 0.554]

Значення з отриманими коефіцієнтами:

[196.855, 199.939, 196.855, 199.939, 199.939, 196.855, 199.939, 196.855, 200.666, 200.666, 197.316956, 197.316956, 200.666, 200.666, 200.666]

Перевірка адекватності за критерієм Фішера

Fp = 1.5519668299753697

при рівні значимості 0.05 рівняння регресії адекватне

Process finished with exit code 0

## Висновок

У ході виконання лабораторної роботи я провів трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план.

Знайшов рівняння регресії, яке буде адекватним для опису об'єкту.

Кінцева мета досягнута.