

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 планування експерименту»

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

ВИКОНАВ:
студент II курсу ФІОТ
групи ІВ-92
Чередник Віталій Юрійович
Номер у списку групи: 26

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 на лабораторну роботу:

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.

Варіант завдання

№варіанта	X ₁		X ₂		X ₃	
	min	max	min	max	min	max
226	-5	4	-2	7	-1	2

Код програми

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

```
m = 3
```

```
x1min = -5
x1max = 4
x2min = -2
x2max = 7
x3min = -1
```

```

x3max = 2

ymax = 200 + (x1max + x2max + x3max) / 3
ymin = 200 + (x1min + x2min + x3min) / 3

xn = [[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],
      [-1, -1, 1, 1, -1, -1, 1, 1, 0, 0, -1.215, 1.215, 0, 0],
      [-1, 1, -1, 1, -1, 1, -1, 1, 0, 0, 0, 0, -1.215, 1.215]]

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] = round(xn[1][i] ** 2, 3)
    x2kv_norm[i] = round(xn[2][i] ** 2, 3)
    x3kv_norm[i] = round(xn[3][i] ** 2, 3)

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(15):
    print(Y[i])

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]
x2 = [-2, -2, 7, 7, -2, -2, 7, 7, x02, x02, -1.215 * deltax2 + x02, 1.215 * deltax2 + 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_for_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_for_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_for_b[i])

skm = linear_model.LinearRegression(fit_intercept=False)
skm.fit(list_for_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("----- Перевірка за критерієм Кохрена -----
-----")
print("Середні значення відгуку за рядками:", "\n", +Y_average[0], Y_average[1],
Y_average[2], Y_average[3],
Y_average[4], Y_average[5], Y_average[6], Y_average[7], Y_average[8],
Y_average[9], Y_average[10],
Y_average[11], Y_average[12], Y_average[13], Y_average[14])
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")
else:
    print("Дисперсія неоднорідна")

print("----- Перевірка значущості коефіцієнтів за критерієм Стьюдента --
-----")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (15 * m)) ** 0.5

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

d = 0
res = [0] * 11
coefs1 = []
coefs2 = []
n = 15
F3 = (m - 1) * n
for i in range(11):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coefs2.append(b[i])
        res[i] = 0
    else:
        coefs1.append(b[i])
        res[i] = b[i]
        d += 1

```

```

print("Значущі коефіцієнти регресії:", coefs1)
print("Незначущі коефіцієнти регресії:", coefs2)

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:", "\n", y_st)

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

```

Результати роботи програми

Матриця планування Y:

```

[201, 199, 198]
[201, 197, 201]
[199, 200, 200]
[197, 202, 201]
[202, 202, 203]
[199, 200, 197]
[200, 201, 204]
[200, 204, 197]
[204, 204, 198]
[198, 199, 202]
[198, 198, 199]
[197, 199, 199]
[198, 198, 201]
[198, 200, 200]
[203, 200, 203]

```

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 = 200.131 + 0.137*x1 + 0.152*x2 + -0.359*x3 + 0.042*x1x2 + -0.708*x1x3 + 0.291*x2x3 + 0.292*x1x2x3 - 0.892*x1^2 + -0.802*x2^2 + -0.238*x3^2

```

----- Перевірка за критерієм Кохрена -----

Середні значення відгуку за рядками:

```

199.333 199.667 199.667 200.0 202.333 198.667 201.667 200.333 202.0 199.667 198.333 198.333 199.0 199.333 202.0

```

Дисперсія однорідна

----- Перевірка значущості коефіцієнтів за критерієм Стюдента -----

Значущі коефіцієнти регресії: [200.131, -0.708, 0.892, -0.802]

Незначущі коефіцієнти регресії: [0.137, 0.152, -0.359, 0.042, 0.291, 0.292, -0.238]

Значення з отриманими коефіцієнтами:

```

[199.513, 200.929, 199.513, 200.929, 200.929, 199.513, 200.929, 199.513, 201.447592, 201.447592, 198.947248, 198.947248, 200.131, 200.131, 200.131]

```

----- Перевірка адекватності за критерієм Фішера -----

Fp = 1.6511788992823935

Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0