

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
ортогональний композиційний план)»**

ВИКОНАЛА:
студентка 2 курсу
групи ІВ-92
Бабенко В.В.
Залікова - 9201

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:

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.

$$y_{\max} = 200 + x_{\text{срmax}}$$

$$y_{\min} = 200 + x_{\text{срmin}}$$

$$\text{де } x_{\text{срmax}} = \frac{x_{1\max} + x_{2\max} + x_{3\max}}{3}, \quad x_{\text{срmin}} = \frac{x_{1\min} + x_{2\min} + x_{3\min}}{3}$$

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

Варіант 201

201	-4	4	-10	4	-5	6
-----	----	---	-----	---	----	---

Лістинг

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

m = 3
n = 15
# варіант 201
x1min = -4
x1max = 4
x2min = -10
x2max = 4
x3min = -5
x3max = 6
# максимальне та мінімальне значення
ymax = 200 + (x1max + x2max + x3max) / 3
ymin = 200 + (x1min + x2min + x3min) / 3
# матриця ПФЕ
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, x1x3_norm, x2x3_norm, x1x2x3_norm, x1kv_norm, x2kv_norm, x3kv_norm =
[0] * n, [0] * n, [0] * n, [0] * n, \

[0] * n, [0] * n, [0] * n
```

```

for i in range(n):
    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)
# заповнення y(генерація)
Y_matrix = [[random.randint(int(ymin), int(ymax)) for i in range(m)] for j in range(n)]
# вивід даних за допомогою цикла
print("Матриця планування y:")
for i in range(15):
    print(Y_matrix[i])

x01 = (x1max + x1min) / 2
x02 = (x2max + x2min) / 2
x03 = (x3max + x3min) / 2

delta_x1 = x1max - x01
delta_x2 = x2max - x02
delta_x3 = x3max - x03
x0 = [1] * n
x1 = [-5, -5, -5, -5, 4, 4, 4, 4, -1.215 * delta_x1 + x01, 1.215 * delta_x1 + x01, x01, x01, x01, x01, x01, x01]
x2 = [-2, -2, 7, 7, -2, -2, 7, 7, x02, x02, -1.215 * delta_x2 + x02, 1.215 * delta_x2 + x02, x02, x02, x02, x02]
x3 = [-1, 2, -1, 2, -1, 2, -1, 2, x03, x03, x03, x03, -1.215 * delta_x3 + x03, 1.215 * delta_x3 + x03, x03, x03]
# заповнення нулями x1x2, x1x3, x1x2x3
x1x2, x1x3, x2x3, x1x2x3 = [0] * n, [0] * n, [0] * n, [0] * n
# заповнення нулями x1kv, x2kv, x3kv
x1kv, x2kv, x3kv = [0] * 15, [0] * 15, [0] * 15

for i in range(n):
    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)
# середні y
Y_average = []
for i in range(len(Y_matrix)):
    Y_average.append(np.mean(Y_matrix[i], axis=0))
    Y_average = [round(i, 3) for i in Y_average]
# формуємо списки b і a
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))
# вивід матриці планування X
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])
# розрахунок дисперсій
dispersions = []
for i in range(len(Y_matrix)):
    a = 0
    for k in Y_matrix[i]:
        a += (k - np.mean(Y_matrix[i], axis=0)) ** 2
    dispersions.append(a / len(Y_matrix[i]))
# експериментально
Gp = max(dispersions) / sum(dispersions)
# теоретично
Gt = 0.3346
# перевірка однорідності дисперсій
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")

# критерій Стюдента
print("Перевірка значущості коефіцієнтів за критерієм Стюдента")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (n * m)) ** 0.5

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

d = 0
res = [0] * 11
coef_1 = []
coef_2 = []
F3 = (m - 1) * n
# перевірка значущості коефіцієнтів(scipy)
for i in range(n-4):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coef_2.append(b[i])
        res[i] = 0
    else:
        coef_1.append(b[i])
        res[i] = b[i]
        d += 1

# вивід
print("Значущі коефіцієнти регресії:", coef_1)
print("Незначущі коефіцієнти регресії:", coef_2)

# значення y з коефіцієнтами регресії
y_st = []
for i in range(n):
    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])
print("Значення з отриманими коефіцієнтами:\n", y_st)

# критерій Фішера
print("\nПеревірка адекватності за критерієм Фішера\n")

```

```

Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(n)]) / (n - d)
Fp = Sad / sb
F4 = n - d
# перевірка за допомогою scipy
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

C:\Users\doma\Anaconda3\python.exe "D: /4 семестр/Методи наукових досліджень/Lab5.py"

Матриця планування у:

```

[194, 201, 198]
[194, 202, 202]
[196, 202, 194]
[203, 203, 195]
[194, 198, 200]
[194, 196, 193]
[195, 199, 201]
[197, 195, 195]
[202, 201, 199]
[202, 204, 199]
[203, 194, 203]
[195, 202, 199]
[194, 202, 203]
[195, 198, 201]
[195, 193, 197]

```

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:

$$y = 198.939 + -0.711 \cdot x_1 + 0.126 \cdot x_2 + -0.276 \cdot x_3 + 0.209 \cdot x_1 x_2 + -1.291 \cdot x_1 x_3 + 0.209 \cdot x_2 x_3 + -0.125 \cdot x_1 x_2 x_3 + 0.634 \cdot x_1^2 + -0.608 \cdot x_2^2 + -0.947 \cdot x_3^2$$

Перевірка за критерієм Кохрена

Середні значення відгуку за рядками:

197.667 199.333 197.333 200.333 197.333 194.333 198.333 195.667

Дисперсія однорідна

Перевірка значущості коефіцієнтів за критерієм Стюдента

Значущі коефіцієнти регресії: [198.939, -1.291, -0.947]

Незначущі коефіцієнти регресії: [-0.711, 0.126, -0.276, 0.209, 0.209, -0.125, 0.634, -0.608]

Значення з отриманими коефіцієнтами:

[197.648, 200.23, 197.648, 200.23, 200.23, 197.648, 200.23, 197.648, 198.939, 198.939, 198.939, 198.939, 198.939, 198.939]

Перевірка адекватності за критерієм Фішера

Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0

Висновок:

В даній лабораторній роботі проведено трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Знайдено рівняння регресії, яке буде адекватним для опису об'єкту. Кінцевої мети досягнуто.