

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Методи наукових досліджень
Лабораторна робота № 5
**«Проведення трьохфакторного експерименту при використанні
рівняння регресії з урахуванням квадратичних членів
(центральний ортогональний композиційний план)»**

Виконав:
студент групи ІВ-93
Манчук М.В.

Київ
2021 р.

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.

$$y_{\max} = 200 + x_{\text{ср max}}$$

$$y_{\min} = 200 + x_{\text{ср min}}$$

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

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

315	-8	9	-8	6	-5	6
-----	----	---	----	---	----	---

Код програми:

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

x1_min = -8
x1_max = 9
x2_min = -8
x2_max = 6
x3_min = -5
x3_max = 6

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

n = 15

x0_n = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
x1_n = (-1, -1, -1, -1, 1, 1, 1, 1, -1.215, 1.215, 0, 0, 0, 0, 0)
x2_n = (-1, -1, 1, 1, -1, -1, 1, 1, 0, 0, -1.215, 1.215, 0, 0, 0)
x3_n = (-1, 1, -1, 1, -1, 1, -1, 1, 0, 0, 0, 0, -1.215, 1.215, 0)
x1x2_n = [x1_n[i] * x2_n[i] for i in range(n)]
x1x3_n = [x1_n[i] * x3_n[i] for i in range(n)]
x2x3_n = [x2_n[i] * x3_n[i] for i in range(n)]
x1x2x3_n = [x1_n[i] * x2_n[i] * x3_n[i] for i in range(n)]
x1_squared_n = [x1_n[i] ** 2 for i in range(n)]
x2_squared_n = [x2_n[i] ** 2 for i in range(n)]
x3_squared_n = [x3_n[i] ** 2 for i in range(n)]

def value(x_max, x_min, l):
    x0 = (x_max + x_min) / 2
    delta_x = x_max - x0
    return l * delta_x + x0

x1 = (
    x1_min, x1_min, x1_min, x1_min, x1_max, x1_max, x1_max, x1_max, value(x1_max, x1_min, -
1.215),
    value(x1_max, x1_min, 1.215), (x1_max + x1_min) / 2, (x1_max + x1_min) / 2, (x1_max +
x1_min) / 2,
    (x1_max + x1_min) / 2, (x1_max + x1_min) / 2)
x2 = (x2_min, x2_min, x2_max, x2_max, x2_min, x2_min, x2_max, x2_max, (x2_max + x2_min) / 2,
(x2_max + x2_min) / 2,
```

```

        value(x2_max, x2_min, -1.215), value(x2_max, x2_min, 1.215), (x2_max + x2_min) / 2,
(x2_max + x2_min) / 2,
        (x2_max + x2_min) / 2)
x3 = (x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, (x3_max + x3_min) / 2,
(x3_max + x3_min) / 2,
        (x3_max + x3_min) / 2, (x3_max + x3_min) / 2, value(x3_max, x3_min, -1.215),
value(x3_max, x3_min, 1.215),
        (x3_max + x3_min) / 2)

x1x2 = [x1[i] * x2[i] for i in range(n)]
x1x3 = [x1[i] * x3[i] for i in range(n)]
x2x3 = [x2[i] * x3[i] for i in range(n)]
x1x2x3 = [x1[i] * x2[i] * x3[i] for i in range(n)]
x1_squared = [x1[i] ** 2 for i in range(n)]
x2_squared = [x2[i] ** 2 for i in range(n)]
x3_squared = [x3[i] ** 2 for i in range(n)]

def experiment(m):
    y = [[random.uniform(y_min, y_max) for _ in range(m)] for _ in range(n)]

    y_response = ([round(sum(y[j][i] for i in range(m)) / m, 3) for j in range(n)])

    print('Середні значення функції відгук:\n{0}'.format(y_response))

    b = list(zip(x0_n, x1_n, x2_n, x3_n, x1x2_n, x1x3_n, x2x3_n, x1x2x3_n, x1_squared_n,
x2_squared_n, x3_squared_n))
    skm = linear_model.LinearRegression(fit_intercept=False)
    skm.fit(b, y_response)
    b = skm.coef_
    b = [round(i, 3) for i in b]

    print('\nОтримане рівняння регресії:\ny = {0} + {1}*x1 + {2}*x2 + {3}*x3 + {4}*x1*x2 +
{5}*x1*x3 + {6}*x2*x3 + '
        '{7}*x1*x2*x3 + {8}*x1^2 + {9}*x2^2 + {10}*x3^2\n'.format(round(b[0], 3),
round(b[1], 3), round(b[2], 3),
round(b[3], 3),
round(b[4], 3), round(b[5], 3),
round(b[6], 3),
round(b[7], 3),
round(b[8], 3), round(b[9], 3),
round(b[10], 3)))

    dispersions = [sum([(y[j][i] - y_response[j]) ** 2 for i in range(m)]) / m for j in
range(n)]
    gp = max(dispersions) / sum(dispersions)

    f1 = m - 1
    f2 = n

    if 11 <= f1 <= 16:
        f1 = 11
    if 17 <= f1 <= 136:
        f1 = 17
    if f1 > 136:
        f1 = 137

    gt = {1: 0.9065, 2: 0.7679, 3: 0.6841, 4: 0.6287, 5: 0.5892, 6: 0.5598, 7: 0.5365, 8:
0.5365, 9: 0.5017, 10: 0.4884,
        11: 0.4366, 17: 0.3720, 137: 0.2500}

    if gp > gt[f1]:
        i = input('Дисперсія неоднорідна. Якщо ви хочете повторити експеримент при m = m + 1 =
{}, введіть 1: \n'
        .format(m + 1))
        if i == '1':
            experiment(m + 1)
            m += 1
        else:
            print('Дисперсія однорідна.\n')

```

```

s_b = sum(dispersions) / n
s = np.sqrt(s_b / (n * m))

t = [abs(b[i]) / s for i in range(11)]

f3 = f1 * f2

d = 0
for i in range(11):
    if t[i] < scipy.stats.t.ppf(q=0.975, df=f3):
        print('Коефіцієнт рівняння регресії b{0} приймаємо незначним при рівні
значимості 0.05'.format(i))
        b[i] = 0
    else:
        d += 1

f4 = n - d

s_ad = (m * sum([(b[0] + b[1] * x1_n[i] + b[2] * x2_n[i] + b[3] * x3_n[i] + b[4] *
x1_n[i] * x2_n[i] + b[5] *
                    x1_n[i] * x3_n[i] + b[6] * x2_n[i] * x3_n[i] + b[7] * x1_n[i] *
x2_n[i] * x3_n[i] -
                    y_response[i]) ** 2 for i in range(n)])] / f4)
f_p = s_ad / s_b

if f_p > scipy.stats.f.ppf(q=0.95, dfn=f4, dfd=f3):
    print('\nРівняння регресії неадекватно '
          'оригіналу при рівні значимості 0.05')
else:
    print('\nРівняння регресії адекватно '
          'оригіналу при рівні значимості 0.05')

experiment(3)

```

Результат виконання програми:

```

C:\Anaconda3\python.exe C:/PythonProjects/MND/Lab5.py
Середні значення функції відгуку:
[197.021, 201.105, 198.782, 201.116, 199.782, 201.109, 198.98, 196.364, 200.738, 200.689, 198.288, 197.794, 198.647, 200.266, 202.424]

Отримане рівняння регресії:
y = 200.778 + -0.169*x1 + -0.399*x2 + 0.648*x3 + -0.915*x1*x2 + -0.963*x1*x3 + -0.712*x2*x3 + -0.274*x1*x2*x3 + 0.322*x1^2 + -1.488*x2^2 + -0.529*x3^2

Дисперсія однорідна.

Коефіцієнт рівняння регресії b1 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b2 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b3 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b4 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b5 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b6 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b7 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b8 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b10 приймаємо незначним при рівні значимості 0.05

Рівняння регресії адекватно оригіналу при рівні значимості 0.05

Process finished with exit code 0

```