

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 5
з дисципліни «Методи наукових досліджень»
на тему «Проведення трьохфакторного експерименту при використанні
рівняння регресії з урахуванням квадратичних членів (центральний
ортогональний композиційний план)»

ВИКОНАВ:
студент 2 курсу
групи ІВ-91
Дерачиц Віталій
Залікова – 9109

ПЕРЕВІРИВ:
ас. Регіда П. Г.

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.

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

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

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

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

Варіанти обираються по номеру в списку в журналі викладача.

109	-3	7	0	9	-8	10
-----	----	---	---	---	----	----

Программний код

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

def main(m_tmp):
    m = m_tmp
    n = 15
    x1min = -3
    x1max = 7
    x2min = 0
    x2max = 9
    x3min = -8
    x3max = 10

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

    l = 1.215

    # 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, 1, 0, 0, 0, 0, 0],
          [-1, -1, 1, 1, -1, -1, 1, 1, 0, 0, -1, 1, 0, 0, 0],
          [-1, 1, -1, 1, -1, 1, -1, 1, 0, 0, 0, 0, -1, 1, 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)

    Y_matrix = [[random.randint(int(ymin), int(ymax)) for i in range(m)] for
j in range(n)]

    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 = [-4, -4, -4, -4, 4, 4, 4, 4, -1 * delta_x1 + x01, 1 * delta_x1 +
x01, x01, x01, x01, x01]
    x2 = [-10, -10, 4, 4, -10, -10, 4, 4, x02, x02, -1 * delta_x2 + x02, 1 *
delta_x2 + x02, x02, x02, x02]
```

```

x3 = [-5, 6, -5, 6, -5, 6, -5, 6, x03, x03, x03, x03, -1 * delta_x3 +
x03, 1 * delta_x3 + x03, x03]

x1x2, x1x3, x2x3, x1x2x3 = [0] * n, [0] * n, [0] * n, [0] * n

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)

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]

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("\nМатриця планування з нормованими коефіцієнтами X:")
for i in range(15):
    print(list_for_b[i])

skm = lm.LinearRegression(fit_intercept=False)
skm.fit(list_for_b, y_average)
b = skm.coef_
b = [round(i, 3) for i in b]

print("\nPівняння 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("\nПЕРЕВІРКА ОДНОРІДНОСТІ ДИСПЕРСІЇ 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])

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]))
print("\ndispersion: \n", dispersions)

Gp = max(dispersions) / sum(dispersions)
Gt = 0.3346

if Gp > Gt:
    print("Дисперсія неоднорідна")
    m += 1
    main(m)
else:

```

```

print("Дисперсія однорідна")

# критерій Стьюдента
print("\nПЕРЕВІРКА ЗНАЧУЩОСТІ КОЕФІЦІЄНТІВ ЗА КРИТЕРІЄМ СТЬЮДЕНТА")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (m * n)) ** (1/2)

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

d = 0
res = [0] * 11
coefficients = []
F3 = (m - 1) * n

for i in range(n-4):
    if t_array[i] < t.ppf(q=0.975, df=F3):
        res[i] = 0
        print('Виключаємо з рівняння статистично незначущий коефіцієнт
b', i)
    else:
        coefficients.append(b[i])
        res[i] = b[i]
        d += 1

print("\nЗначущі коефіцієнти регресії:", coefficients)

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("\nЗначення з отриманими коефіцієнтами:\n", y_st)

print("\nПЕРЕВІРКА АДЕКВАТНОСТІ ЗА КРИТЕРІЄМ ФІШЕРА")
Sad = m * sum([(y_st[i] - y_average[i]) ** 2 for i in range(n)]) / (n -
d)
Fp = Sad / sb
f4 = n - d

if Fp > f.ppf(q=0.95, dfn=f4, dfd=F3):
    print("Рівняння регресії неадекватно оригіналу при рівні значимості
0.05")
else:
    print("Рівняння регресії адекватно оригіналу при рівні значимості
0.05")

if __name__ == '__main__':
    main(3)

```

Результат роботи програми

```
C:\Users\derac\Anaconda3\python.exe "D:/Repo/MND/Laboratory work №5/main.py"
Матриця планування у:
[206, 200, 204]
[208, 199, 203]
[203, 207, 198]
[198, 203, 202]
[201, 200, 198]
[200, 198, 198]
[207, 207, 205]
[199, 203, 207]
[198, 206, 203]
[196, 208, 204]
[196, 204, 207]
[206, 199, 207]
[202, 199, 196]
[208, 196, 199]
[197, 204, 207]

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 = 202.027 + -0.206*x1 + 0.915*x2 + -0.326*x3 + 1.75*x1x2 + -0.333*x1x3 + -0.5*x2x3 + -0.083*x1x2x3 0.463*x1^2 + 0.914*x2^2 + -1.231*x3^2

ПЕРЕВІРКА ОДНОРІДНОСТІ ДИСПЕРСІЇ ЗА КРИТЕРІЄМ КОХРЕНА
Середні значення відгуку за рядками:
203.333 203.333 202.667 201.0 199.667 198.667 206.333 203.0

dispersion:
[6.222222222222222, 13.555555555555555, 13.555555555555557, 4.666666666666667, 1.5555555555555554, 0.8888888888888888, 0.8888888888888888, 10.666666666666666,
Дисперсія однорідна

ПЕРЕВІРКА ЗНАЧУЩОСТІ КОЕФІЦІЄНТІВ ЗА КРИТЕРІЄМ СТЬЮДЕНТА
Виключаємо з рівняння статистично незначущий коефіцієнт b 1
Виключаємо з рівняння статистично незначущий коефіцієнт b 2
Виключаємо з рівняння статистично незначущий коефіцієнт b 3
Виключаємо з рівняння статистично незначущий коефіцієнт b 5
Виключаємо з рівняння статистично незначущий коефіцієнт b 6
Виключаємо з рівняння статистично незначущий коефіцієнт b 7
Виключаємо з рівняння статистично незначущий коефіцієнт b 8
Виключаємо з рівняння статистично незначущий коефіцієнт b 9

Значущі коефіцієнти регресії: [202.027, 1.75, -1.231]

Значення з отриманими коефіцієнтами:
[203.777, 203.777, 200.277, 200.277, 200.277, 200.277, 203.777, 203.777, 202.027, 202.027, 202.027, 202.027, 202.027, 202.027]

ПЕРЕВІРКА АДЕКВАТНОСТІ ЗА КРИТЕРІЄМ ФІШЕРА
Рівняння регресії адекватно оригіналу при рівні значимості 0.05

Process finished with exit code 0
|
```

Висновок:

Виконуючи дану лабораторну роботу, я провів трьохфакторний експеримент при використанні рівняння регресії з урахуванням квадратичних членів (центральний ортогональний композиційний план). Склав матрицю планування та знайшов коефіцієнти рівняння регресії, провів статистичні перевірки.

Результати роботи програми наведені вище підтверджують правильність її виконання.