

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
Степанюк Р. В.
Залікова – 9127

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Мета: Провести повний трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Знайти рівняння регресії адекватне 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.

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

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

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

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

№ варіанту	x ₁		x ₂		x ₃	
	min	max	min	max	min	max
125	-1	6	-10	5	-8	2

Программный код

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

def average(list):
    average = 0
    for element in list:
        average += element / len(list)
    return average

def dispersion(list):
    list_average = average(list)
    dispersion = 0
    for element in list:
        dispersion += (element - list_average)**2 / len(list)
    return dispersion

def coef_b(x, y):
    for i in range(len(x)):
        x[i].insert(0, 1)
    skm = lm.LinearRegression(fit_intercept = False)
    skm.fit(x, y)
    b = skm.coef_
    return b

def cochrane_criteria():
    global k, N
    gp_denominator = 0
    for disp in dispersion_list:
        gp_denominator += disp
    gp = max(dispersion_list) / gp_denominator
    f1 = k - 1
    f2 = N
    gt = 0.3346
    if gp < gt: return True
    else: return False

def students_criteria(b):
    global k, N
    sb = average(dispersion_list)
    s_beta_2 = sb / (N * k)
    s_beta = math.sqrt(s_beta_2)
    beta = [sum(average_list[j] * plan_matrix[j][s] for j in range(N)) / N for s in range(m)]
    t = [abs(beta[i]) / s_beta for i in range(m)]
    f3 = (k - 1) * N
    tt = 2.042
    student_check = {}
    for i in range(m):
        if (t[i] > tt): student_check[i] = b[i]
        else:
            student_check[i] = 0
            b[i] = 0
    return student_check
```

```

def fisher_criteria():
    global k, N
    d = 0
    for key in students_criteria:
        if students_criteria[key] != 0: d += 1
    f1 = k - 1
    f2 = N
    f3 = (k - 1) * N
    f4 = N - d
    s2_ad = sum((regression_equation[i] - average_list[i])**2 for i in range(N))
    if (f4 == 0): s2_ad *= k / 10**-12
    else: s2_ad *= k / f4
    s2_b = average(dispersion_list)
    fp = s2_ad / s2_b
    if fp > f.ppf(q=0.95, dfn=f4, dfd=f3):
        print("\nPівняння регресії неадекватно оригіналу при рівні значимості 0.05")
        return True
    else:
        print("\nPівняння регресії адекватно оригіналу при рівні значимості 0.05")
        return False

x_min = [-1, -10, -8]
x_max = [6, 5, 2]
y_min = 200 + average(x_min)
y_max = 200 + average(x_max)
k = 3
N = 15
m = 10
l = 1.215
# Матриця планування ОЦКП для k = 3
plan_matrix = [
    [-1, -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],
    [1, 0, 0],
    [0, -1, 0],
    [0, 1, 0],
    [0, 0, -1],
    [0, 0, 1],
    [0, 0, 0]]
# Матриця планування ОЦКП із натуралізованими значеннями для k = 3
x0 = [(x_min[i] + x_max[i]) / 2 for i in range(k)]
delta_x = [x_max[i] - x0[i] for i in range(k)]
print(f"x\u2080: {x0}\n\u0394x: {delta_x}")
natur_matrix = [[plan_matrix[i][j] * delta_x[j] + x0[j] for j in range(k)] for i in range(N)]

```

```

for i in range(N):
    plan_matrix[i].append(plan_matrix[i][0] * plan_matrix[i][1])
    plan_matrix[i].append(plan_matrix[i][0] * plan_matrix[i][2])
    plan_matrix[i].append(plan_matrix[i][1] * plan_matrix[i][2])
    plan_matrix[i].append(plan_matrix[i][0] * plan_matrix[i][1] * plan_matrix[i][2])
    plan_matrix[i].append(plan_matrix[i][0] ** 2)
    plan_matrix[i].append(plan_matrix[i][1] ** 2)
    plan_matrix[i].append(plan_matrix[i][2] ** 2)
for i in range(N):
    natur_matrix[i].append(natur_matrix[i][0] * natur_matrix[i][1])
    natur_matrix[i].append(natur_matrix[i][0] * natur_matrix[i][2])
    natur_matrix[i].append(natur_matrix[i][1] * natur_matrix[i][2])
    natur_matrix[i].append(natur_matrix[i][0] * natur_matrix[i][1] * natur_matrix[i][2])
    natur_matrix[i].append(natur_matrix[i][0] ** 2)
    natur_matrix[i].append(natur_matrix[i][1] ** 2)
    natur_matrix[i].append(natur_matrix[i][2] ** 2)
print("\nМатриця планування ОЦКП із натуралізованими значеннями:")
for line in natur_matrix:
    print(line)
y_list = [[random.randint(int(y_min), int(y_max)) for _ in range(k)] for __ in range(N)]
print("\nФункції відгуку:")
for line in y_list:
    print(line)
dispersion_list = [dispersion(y_list[i]) for i in range(N)]
# Знайдемо середні значення функцій відгуку за рядками
average_list = [average(y_list[i]) for i in range(N)]
# Перевірка однорідності дисперсії за критерієм Кохрена
cochrane_criteria = cochrane_criteria()
if cochrane_criteria: print("\nДисперсія однорідна")
else:
    print("\nДисперсія неоднорідна")
    exit()
# Знайдемо коефіцієнти рівняння регресії методом найменших квадратів
b0 = sum(average_list) / N
b = coef_b(natur_matrix, average_list)
print("\nКоефіцієнти рівняння регресії:")
for i in range(len(b)):
    print(f"b{i} = {round(b[i], 3)}")
# Перевірка значущості коефіцієнтів за критерієм Стюдента
significant_coefficients = 0
students_criteria = students_criteria(b)
for key in students_criteria:
    if students_criteria[key] != 0:
        significant_coefficients += 1
# Перевірка адекватності за критерієм Фішера
regression_equation = [
    b0 +
    b[0] * natur_matrix[i][0] +
    b[1] * natur_matrix[i][1] +
    b[2] * natur_matrix[i][2] +
    b[3] * natur_matrix[i][3] +

```

```
        b[4] * natur_matrix[i][4] +  
        b[5] * natur_matrix[i][5] +  
        b[6] * natur_matrix[i][6] +  
        b[7] * natur_matrix[i][7] +  
        b[8] * natur_matrix[i][8] +  
        b[9] * natur_matrix[i][9] for i in range(N)]  
fisher_criteria()
```

Результати роботи програми

```
Х0: [2.5, -2.5, -3.0]
Δх: [3.5, 7.5, 5.0]

Матриця планування ОЦКП із натуралізованими значеннями:
[-1.0, -10.0, -8.0, 10.0, 8.0, 80.0, -80.0, 1.0, 100.0, €4.0]
[-1.0, -10.0, 2.0, 10.0, -2.0, -20.0, 20.0, 1.0, 100.0, 4.0]
[-1.0, 5.0, -8.0, -5.0, 8.0, -40.0, 40.0, 1.0, 25.0, €4.0]
[-1.0, 5.0, 2.0, -5.0, -2.0, 10.0, -10.0, 1.0, 25.0, 4.0]
[6.0, -10.0, -8.0, -60.0, -48.0, 80.0, 480.0, 36.0, 100.0, €4.0]
[6.0, -10.0, 2.0, -60.0, 12.0, -20.0, -120.0, 36.0, 100.0, 4.0]
[6.0, 5.0, -8.0, 30.0, -48.0, -40.0, -240.0, 36.0, 25.0, €4.0]
[6.0, 5.0, 2.0, 30.0, 12.0, 10.0, 60.0, 36.0, 25.0, 4.0]
[-1.7525000000000004, -2.5, -3.0, 4.381250000000001, 5.257500000000001, 7.5, -13.143750000000004, 3.0712562500000016, 6.25, 9.0]
[6.7525, -2.5, -3.0, -16.88125, -20.2575, 7.5, 50.€437500000000004, 45.59625625, 6.25, 9.0]
[2.5, -11.6125, -3.0, -29.03125, -7.5, 34.837500000000006, 87.09375, 6.25, 134.85015625000003, 9.0]
[2.5, 6.612500000000001, -3.0, 16.53125, -7.5, -19.837500000000002, -49.59375, 6.25, 43.72515625000001, 9.0]
[2.5, -2.5, -9.075, -6.25, -22.6875, 22.6875, 56.71874999999999, 6.25, 6.25, 82.35562499999999]
[2.5, -2.5, 3.075, -6.25, 7.6875, -7.6875, -19.21875, 6.25, 6.25, 9.455625000000001]
[2.5, -2.5, -3.0, -6.25, -7.5, 7.5, 18.75, 6.25, 6.25, 9.0]

Функції відгуку:
[204, 199, 196]
[198, 198, 194]
[201, 196, 201]
[200, 202, 195]
[202, 196, 195]
[204, 193, 201]
[197, 204, 202]
[198, 200, 204]
[204, 193, 198]
[202, 194, 198]
[199, 193, 196]
[193, 195, 204]
[204, 197, 195]
[195, 198, 202]
[198, 201, 197]

Дисперсія однорідна

Коефіцієнти рівняння регресії:
b0 = 197.811
b1 = -0.135
b2 = 0.09
b3 = 0.153
b4 = -0.001
b5 = 0.022
b6 = 0.013
b7 = -0.004
b8 = 0.065
b9 = -0.004
b10 = 0.041

Рівняння регресії адекватно оригіналу при рівні значимості 0.05
```

```
Х0: [2.5, -2.5, -3.0]
Δх: [3.5, 7.5, 5.0]

Матриця планування ОЦКП із натуралізованими значеннями:
[-1.0, -10.0, -8.0, 10.0, 8.0, 80.0, -80.0, 1.0, 100.0, €4.0]
[-1.0, -10.0, 2.0, 10.0, -2.0, -20.0, 20.0, 1.0, 100.0, 4.0]
[-1.0, 5.0, -8.0, -5.0, 8.0, -40.0, 40.0, 1.0, 25.0, €4.0]
[-1.0, 5.0, 2.0, -5.0, -2.0, 10.0, -10.0, 1.0, 25.0, 4.0]
[6.0, -10.0, -8.0, -60.0, -48.0, 80.0, 480.0, 36.0, 100.0, €4.0]
[6.0, -10.0, 2.0, -60.0, 12.0, -20.0, -120.0, 36.0, 100.0, 4.0]
[6.0, 5.0, -8.0, 30.0, -48.0, -40.0, -240.0, 36.0, 25.0, €4.0]
[6.0, 5.0, 2.0, 30.0, 12.0, 10.0, 60.0, 36.0, 25.0, 4.0]
[-1.7525000000000004, -2.5, -3.0, 4.381250000000001, 5.257500000000001, 7.5, -13.143750000000004, 3.0712562500000016, 6.25, 9.0]
[6.7525, -2.5, -3.0, -16.88125, -20.2575, 7.5, 50.€437500000000004, 45.59625625, 6.25, 9.0]
[2.5, -11.6125, -3.0, -29.03125, -7.5, 34.837500000000006, 87.09375, 6.25, 134.85015625000003, 9.0]
[2.5, 6.612500000000001, -3.0, 16.53125, -7.5, -19.837500000000002, -49.59375, 6.25, 43.72515625000001, 9.0]
[2.5, -2.5, -9.075, -6.25, -22.6875, 22.6875, 56.71874999999999, 6.25, 6.25, 82.35562499999999]
[2.5, -2.5, 3.075, -6.25, 7.6875, -7.6875, -19.21875, 6.25, 6.25, 9.455625000000001]
[2.5, -2.5, -3.0, -6.25, -7.5, 7.5, 18.75, 6.25, 6.25, 9.0]

Функції відгуку:
[196, 204, 198]
[194, 195, 201]
[194, 198, 201]
[194, 202, 202]
[199, 201, 196]
[195, 199, 196]
[201, 195, 200]
[204, 202, 196]
[198, 194, 198]
[194, 199, 193]
[199, 195, 200]
[204, 199, 196]
[196, 204, 197]
[195, 194, 196]
[201, 194, 195]

Дисперсія однорідна

Коефіцієнти рівняння регресії:
b0 = 196.606
b1 = 0.129
b2 = 0.29
b3 = 0.066
b4 = 0.013
b5 = 0.006
b6 = 0.029
b7 = -0.0
b8 = -0.015
b9 = 0.031
b10 = 0.02

Рівняння регресії неадекватно оригіналу при рівні значимості 0.05
```

Висновок:

Під час виконання даної лабораторної роботи я провів трьохфакторний експеримент, перевінив однорідність дисперсії за критерієм Кохрена, отримав коефіцієнти рівняння регресії, оцінив значимість знайдених коефіцієнтів за критеріями Стюдента та перевінив адекватність за критерієм Фішера.

Отже, мета лабораторної роботи була досягнута.