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 ЕКСПЕРИМЕТНУ 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:

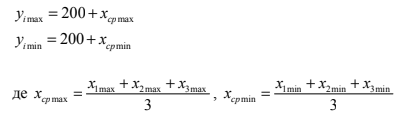
ас. Регіда П. Г.

Київ – 2021

**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.



1. Розрахувати коефіцієнти рівняння регресії і записати його.
2. Провести 3 статистичні перевірки

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № варіанту | x1 | | x2 | | x3 | |
| min | max | min | max | min | max |
| 125 | -1 | 6 | -10 | 5 | -8 | 2 |

**Програмний 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("\nРівняння регресії неадекватно оригіналу при рівні значимості 0.05")

        return True

    else:

        print("\nРівняння 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],

    [-l,     0,      0],

    [ l,     0,      0],

    [ 0,    -l,      0],

    [ 0,     l,      0],

    [ 0,     0,     -l],

    [ 0,     0,      l],

    [ 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}\nΔx: {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 Стьюдента

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] +

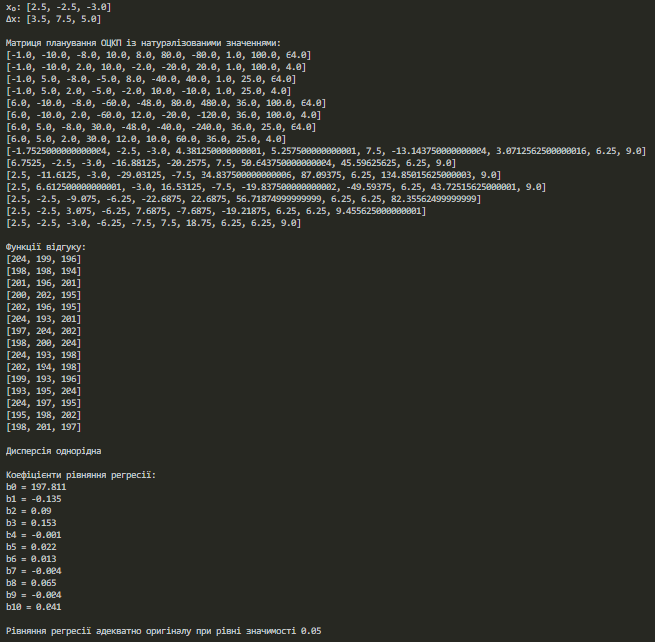
            b[7] \* natur\_matrix[i][7] +

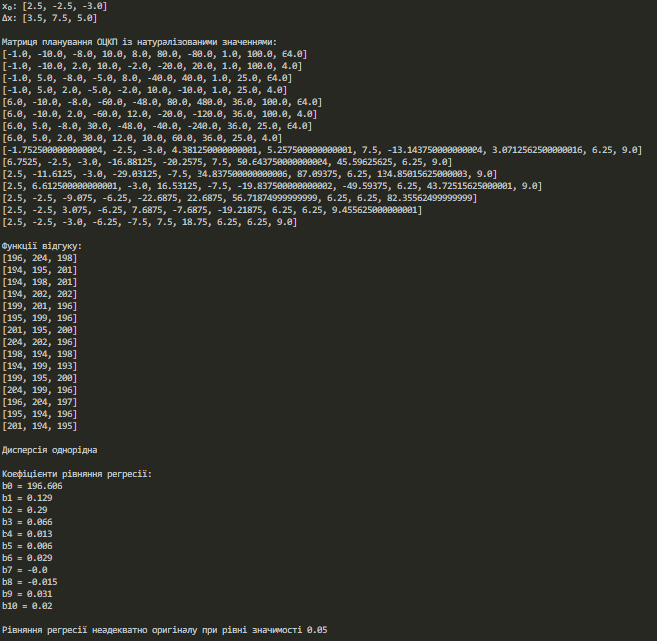
            b[8] \* natur\_matrix[i][8] +

            b[9] \* natur\_matrix[i][9] for i in range(N)]

fisher\_criteria()

**Результати роботи програми**

****



**Висновок:**

Під час виконання даної лабораторної роботи я провів трьохфакторний експеримент, перевірив однорідність дисперсії за критерієм Кохрена, отримав коефіцієнти рівняння регресії, оцінив значимість знайдених коефіцієнтів за критеріями Стьюдента та перевірив адекватність за критерієм Фішера.  
Отже, мета лабораторної роботи була досягнута.