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 № 6

з дисципліни «МНД» на тему

«Проведення трьохфакторного експерименту при використанні рівняння регресії з квадратичними членами»

ВИКОНАВ:
студент ІІ курсу ФІОТ
групи ІВ-91
Чопик Н.О.
Залікова - 9130

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Київ – 2021

Мета: Провести трьохфакторний експеримент і отримати адекватну модель – рівняння регресії, використовуючи рототабельний композиційний план.

Завдання:

1. Ознайомитися з теоретичними відомостями.
2. Вибрати з таблиці варіантів і записати в протокол інтервали значень x_1, x_2, x_3. Обчислити і записати значення, відповідні кодованим значенням факторів +1; -1;+ l; - l; 0 для $\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$.
3. Значення функції відгуку знайти за допомогою підстановки в формулу:

$$y_i = f(x_1, x_2, x_3) + \text{random}(10)-5,$$

де $f(x_1, x_2, x_3)$ вибирається по номеру в списку в журналі викладача.

4. Провести експерименти і аналізуючи значення статистичних перевірок, отримати адекватну модель рівняння регресії. При розрахунках використовувати натуральні значення факторів.
5. Зробити висновки по виконаній роботі.

Алгоритм отримання адекватної моделі рівняння регресії

1) *Вибір рівняння регресії (лінійна форма, рівняння з урахуванням ефекту взаємодії і з урахуванням квадратичних членів);*
2) *Вибір кількості повторів кожної комбінації (m = 2);*
3) *Складення матриці планування експерименту і вибір кількості рівнів (N)*
4) *Проведення експериментів;*
5) *Перевірка однорідності дисперсії. Якщо не однорідна – повертаємося на п. 2 і збільшуємо m на 1);*
6) *Розрахунок коефіцієнтів рівняння регресії. При розрахунку використовувати* ***натуральні*** *значення x_1, x_2 и x_3.*
7) *Перевірка нуль-гіпотези. Визначення значимих коефіцієнтів;*
8) *Перевірка адекватності моделі рівняння оригіналу. При неадекватності – повертаємося на п.1, змінивши при цьому рівняння регресії;*

Варіант: 128

№ варіанта	X1		X2		X3		$f(x_1, x_2, x_3)$
128	min	max	min	max	min	max	4,4+8,3*x1+3,5*x2+ +8,0*x3+2,9*x1*x1+0,3*x2*x2+ +2,3*x3*x3+3,4*x1*x2+ +0,3*x1*x3+9,3*x2*x3+ +8,3*x1*x2*x3
	-15	30	30	80	30	35	

Лістинг програми:

```
from math import fabs
from random import randrange
import numpy as np
from numpy.linalg import solve
from scipy.stats import f, t
from prettytable import PrettyTable


def round_matrix(matrix, n_to_round=3):
    for i in range(len(matrix)):
        matrix[i] = list(matrix[i])
        for j in range(len(matrix[i])):
            matrix[i][j] = round(matrix[i][j], n_to_round)
    return matrix


m = 3
n = 15

x1min = -15
x1max = 30
x2min = 30
x2max = 80
x3min = 30
x3max = 35

x01 = (x1max + x1min) / 2
x02 = (x2max + x2min) / 2
x03 = (x3max + x3min) / 2
deltax1 = x1max - x01
deltax2 = x2max - x02
deltax3 = x3max - x03

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, -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],
      [-1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],
      [+1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],
      [0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0],
      [0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0],
      [0, 0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929],
      [0, 0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

x1 = [x1min, x1min, x1min, x1min, x1max, x1max, x1max, x1max, -1.73 * deltax1 + x01, 1.73 * deltax1
+ x01, x01, x01,
      x01, x01, x01]
x2 = [x2min, x2min, x2max, x2max, x2min, x2min, x2max, x2max, x02, x02, -1.73 * deltax2 + x02, 1.73
* deltax2 + x02,
      x02, x02, x02]
x3 = [x3min, x3max, x3min, x3max, x3min, x3max, x3min, x3max, x03, x03, x03, x03, -1.73 * deltax3 +
x03,
      1.73 * deltax3 + x03, x03]
```

```
x1x2 = [0] * 15
x1x3 = [0] * 15
x2x3 = [0] * 15
x1x2x3 = [0] * 15
x1kv = [0] * 15
x2kv = [0] * 15
x3kv = [0] * 15

for i in range(15):
    x1x2[i] = x1[i] * x2[i]
    x1x3[i] = x1[i] * x3[i]
    x2x3[i] = x2[i] * x3[i]
    x1x2x3[i] = x1[i] * x2[i] * x3[i]
    x1kv[i] = x1[i] ** 2
    x2kv[i] = x2[i] ** 2
    x3kv[i] = x3[i] ** 2

list_for_a = round_matrix(list(zip(x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1kv, x2kv, x3kv)))

planning_matrix_with_naturalized_coeffs_x = PrettyTable()
planning_matrix_with_naturalized_coeffs_x.title = 'Матриця планування з натуралізованими
коефіцієнтами X'
planning_matrix_with_naturalized_coeffs_x.field_names = ['X1', 'X2', 'X3', 'X1X2', 'X1X3', 'X2X3',
'X1X2X3', 'X1X1', 'X2X2', 'X3X3']
planning_matrix_with_naturalized_coeffs_x.add_rows(list_for_a)
print(planning_matrix_with_naturalized_coeffs_x)


def function(X1, X2, X3):
    y = 4.4 + 8.3 * X1 + 3.5 * X2 + 8 * X3 + 2.9 * X1 * X1 + 0.3 * X2 * X2 + 2.3 * X3 * X3 + 3.4 *
X1 * X2 + \
        0.3 * X1 * X3 + 9.3 * X2 * X3 + 8.3 * X1 * X2 * X3 + randrange(0, 10) - 5
    return y


Y = round_matrix([[function(list_for_a[j][0], list_for_a[j][1], list_for_a[j][2]) for i in range(m)]
for j in range(15)])

planning_matrix_y = PrettyTable()
planning_matrix_y.title = 'Матриця планування Y'
planning_matrix_y.field_names = ['Y1', 'Y2', 'Y3']
planning_matrix_y.add_rows(Y)
print(planning_matrix_y)

Y_average = []
for i in range(len(Y)):
    Y_average.append(np.mean(Y[i], axis=0))
print("Середні значення відгуку за рядками:")
for i in range(15):
    print("{:.3f}".format(Y_average[i]), end=" ")

dispersions = []
for i in range(len(Y)):
    a = 0
    for k in Y[i]:
        a += (k - np.mean(Y[i], axis=0)) ** 2
    dispersions.append(a / len(Y[i]))


def find_known(num):
    a = 0
    for j in range(15):
```

```
        a += Y_average[j] * list_for_a[j][num - 1] / 15
    return a


def a(first, second):
    a = 0
    for j in range(15):
        a += list_for_a[j][first - 1] * list_for_a[j][second - 1] / 15
    return a


my = sum(Y_average) / 15
mx = []

for i in range(10):
    number_lst = []
    for j in range(15):
        number_lst.append(list_for_a[j][i])
    mx.append(sum(number_lst) / len(number_lst))

det1 = [
    [1, mx[0], mx[1], mx[2], mx[3], mx[4], mx[5], mx[6], mx[7], mx[8], mx[9]],
    [mx[0], a(1, 1), a(1, 2), a(1, 3), a(1, 4), a(1, 5), a(1, 6), a(1, 7), a(1, 8), a(1, 9), a(1,
10)],
    [mx[1], a(2, 1), a(2, 2), a(2, 3), a(2, 4), a(2, 5), a(2, 6), a(2, 7), a(2, 8), a(2, 9), a(2,
10)],
    [mx[2], a(3, 1), a(3, 2), a(3, 3), a(3, 4), a(3, 5), a(3, 6), a(3, 7), a(3, 8), a(3, 9), a(3,
10)],
    [mx[3], a(4, 1), a(4, 2), a(4, 3), a(4, 4), a(4, 5), a(4, 6), a(4, 7), a(4, 8), a(4, 9), a(4,
10)],
    [mx[4], a(5, 1), a(5, 2), a(5, 3), a(5, 4), a(5, 5), a(5, 6), a(5, 7), a(5, 8), a(5, 9), a(5,
10)],
    [mx[5], a(6, 1), a(6, 2), a(6, 3), a(6, 4), a(6, 5), a(6, 6), a(6, 7), a(6, 8), a(6, 9), a(6,
10)],
    [mx[6], a(7, 1), a(7, 2), a(7, 3), a(7, 4), a(7, 5), a(7, 6), a(7, 7), a(7, 8), a(7, 9), a(7,
10)],
    [mx[7], a(8, 1), a(8, 2), a(8, 3), a(8, 4), a(8, 5), a(8, 6), a(8, 7), a(8, 8), a(8, 9), a(8,
10)],
    [mx[8], a(9, 1), a(9, 2), a(9, 3), a(9, 4), a(9, 5), a(9, 6), a(9, 7), a(9, 8), a(9, 9), a(9,
10)],
    [mx[9], a(10, 1), a(10, 2), a(10, 3), a(10, 4), a(10, 5), a(10, 6), a(10, 7), a(10, 8), a(10,
9), a(10, 10)]]

det2 = [my, find_known(1), find_known(2), find_known(3), find_known(4), find_known(5),
find_known(6), find_known(7),
        find_known(8), find_known(9), find_known(10)]

beta = solve(det1, det2)
print("\nОтримане рівняння регресії:")
print("{:.3f} + {:.3f} * X1 + {:.3f} * X2 + {:.3f} * X3 + {:.3f} * X1X2 + {:.3f} * X1X3 + {:.3f} *
X2X3"
      "+ {:.3f} * X1X2X3 + {:.3f} * X11^2 + {:.3f} * X22^2 + {:.3f} * X33^2 = ŷ"
      .format(beta[0], beta[1], beta[2], beta[3], beta[4], beta[5], beta[6], beta[7], beta[8],
beta[9], beta[10]))
y_i = [0] * 15
print("Експериментальні значення:")
for k in range(15):
    y_i[k] = beta[0] + beta[1] * list_for_a[k][0] + beta[2] * list_for_a[k][1] + beta[3] *
list_for_a[k][2] + \
             beta[4] * list_for_a[k][3] + beta[5] * list_for_a[k][4] + beta[6] * list_for_a[k][5] +
beta[7] * \
             list_for_a[k][6] + beta[8] * list_for_a[k][7] + beta[9] * list_for_a[k][8] + beta[10] *
```

```
list_for_a[k][9]
for i in range(15):
    print("{:.3f}".format(y_i[i]), end=" ")
print("\n\nПеревірка за критерієм Кохрена")
Gp = max(dispersions) / sum(dispersions)
Gt = 0.3346
print("Gp =", Gp)
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")

print("\nПеревірка значущості коефіцієнтів за критерієм Стьюдента")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (15 * m)) ** 0.5

F3 = (m - 1) * n
coefs1 = []
coefs2 = []
d = 11
res = [0] * 11
for j in range(11):
    t_pract = 0
    for i in range(15):
        if j == 0:
            t_pract += Y_average[i] / 15
        else:
            t_pract += Y_average[i] * xn[i][j - 1]
        res[j] = beta[j]
    if fabs(t_pract / sbs) < t.ppf(q=0.975, df=F3):
        coefs2.append(beta[j])
        res[j] = 0
        d -= 1
    else:
        coefs1.append(beta[j])
print("Значущі коефіцієнти регресії:", [round(i, 3) for i in coefs1])
print("Незначущі коефіцієнти регресії:", [round(i, 3) for i in coefs2])
y_st = []
for i in range(15):
    y_st.append(res[0] + res[1] * x1[i] + res[2] * x2[i] + res[3] * x3[i] + res[4] * x1x2[i] +
res[5] *
                x1x3[i] + res[6] * x2x3[i] + res[7] * x1x2x3[i] + res[8] * x1kv[i] + res[9] *
                x2kv[i] + res[10] * x3kv[i])
print("Значення з отриманими коефіцієнтами:")
for i in range(15):
    print("{:.3f}".format(y_st[i]), end=" ")

print("\n\nПеревірка адекватності за критерієм Фішера")
Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(15)]) / (n - d)
Fp = Sad / sb
F4 = n - d
print("Fp =", Fp)
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")
```

Результат виконання роботи:

```
"C:\Users\Nazar\Desktop\КПІ\Методи наукових досліджень\venv\Scripts\python.exe" "C:/Users/Nazar/Desktop/КПІ/Метод
```

Матриця планування з натуралізованими коефіцієнтами X

X1	X2	X3	X1X2	X1X3	X2X3	X1X2X3	X1X1	X2X2	X3X3
-15	30	30	-450	-450	900	-13500	225	900	900
-15	30	35	-450	-525	1050	-15750	225	900	1225
-15	80	30	-1200	-450	2400	-36000	225	6400	900
-15	80	35	-1200	-525	2800	-42000	225	6400	1225
30	30	30	900	900	900	27000	900	900	900
30	30	35	900	1050	1050	31500	900	900	1225
30	80	30	2400	900	2400	72000	900	6400	900
30	80	35	2400	1050	2800	84000	900	6400	1225
-31.425	55.0	32.5	-1728.375	-1021.312	1787.5	-56172.187	987.531	3025.0	1056.25
46.425	55.0	32.5	2553.375	1508.812	1787.5	82984.688	2155.281	3025.0	1056.25
7.5	11.75	32.5	88.125	243.75	381.875	2864.062	56.25	138.062	1056.25
7.5	98.25	32.5	736.875	243.75	3193.125	23948.438	56.25	9653.062	1056.25
7.5	55.0	28.175	412.5	211.312	1549.625	11622.188	56.25	3025.0	793.831
7.5	55.0	36.825	412.5	276.188	2025.375	15190.313	56.25	3025.0	1356.081
7.5	55.0	32.5	412.5	243.75	1787.5	13406.25	56.25	3025.0	1056.25

Матриця планування Y

Y1	Y2	Y3
-102131.6	-102128.6	-102128.6
-118639.6	-118647.6	-118646.6
-275654.6	-275651.6	-275653.6
-320971.6	-320965.6	-320966.6
241350.4	241349.4	241344.4
280920.9	280927.9	280922.9
635721.4	635718.4	635724.4
739870.9	739877.9	739870.9
-449388.489	-449391.489	-449389.489
724964.191	724958.191	724962.191
30694.6	30700.6	30699.6
237204.538	237209.538	237204.538
115718.548	115726.548	115719.548
151139.298	151140.298	151147.298
133387.4	133393.4	133385.4

```
Середні значення відгуку за рядками:
-102129.600 -118644.600 -275653.267 -320967.933 241348.067 280923.900 635721.400 739873.233 -449389.822 724961.524 30698.267 237206.205 115721.548 151142.298 133388.733
Отримане рівняння регресії:
-36.788 + 8.634 * X1 + 3.360 * X2 + 10.723 * X3 + 3.397 * X1X2 + 0.291 * X1X3 + 9.302 * X2X3+ 8.300 * X1X2X3 + 2.901 * X11^2 + 0.301 * X22^2 + 2.255 * X33^2 = ŷ
Експериментальні значення:
-102128.876 -118643.909 -275652.616 -320967.316 241348.922 280924.722 635722.182 739873.982 -449390.655 724960.388 30697.198 237205.305 115720.525 151141.352 133388.747

Перевірка за критерієм Кохрена
Gp = 0.11242603550295856
Дисперсія однорідна

Перевірка значущості коефіцієнтів за критерієм Стьюдента
Значущі коефіцієнти регресії: [-36.788, 8.634, 3.36, 10.723, 3.397, 0.291, 9.302, 8.3, 2.901, 0.301, 2.255]
Незначущі коефіцієнти регресії: []
Значення з отриманими коефіцієнтами:
-102128.876 -118643.909 -275652.616 -320967.316 241348.922 280924.722 635722.182 739873.982 -449390.660 724960.383 30697.202 237205.301 115720.520 151141.347 133388.747

Перевірка адекватності за критерієм Фішера
Fp = 1.02890203825762
Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0
```