

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 «Методи наукових досліджень»  
**Проведення трьохфакторного експерименту при використанні рівняння  
регресії з квадратичними членами**

ВИКОНАВ:  
Студент II курсу ФІОТ  
Групи ІО-91  
Павлюк Владислав Володимирович  
Номер заліковки: 9120  
Номер у списку: 17

ПЕРЕВІРИВ:  
ас. Регіда П. Г.

Київ 2021 р.

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

### Завдання до лабораторної роботи:

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; + ; - ; 0 для 1, 2, 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;

### Виконання роботи

Таблиця варіантів

№ варіанту	$x_1$		$x_2$		$x_3$		$f(x_1, x_2, x_3)$
	min	max	min	max	min	max	
117	20	70	25	65	25	35	$0,6+4,0*x_1+2,8*x_2+4,7*x_3+3,1*x_1*x_1+0,4*x_2*x_2+5,4*x_3*x_3+5,7*x_1*x_2+0,1*x_1*x_3+8,8*x_2*x_3+0,1*x_1*x_2*x_3$

### 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

m = 3
n = 15

x1_min = 20
x1_max = 70
x2_min = 25
x2_max = 65
x3_min = 25
```

```

x3_max = 35

def function(x1, x2, x3):
    y = 0.6 + 4.0 * x1 + 2.8 * x2 + 4.7 * x3 + 3.1 * x1 * x1 + 0.4 * x2 * x2 +
5.4 * x3 * x3 + 5.7 * x1 * x2 + \
        0.1 * x1 * x3 + 8.8 * x2 * x3 + 0.1 * x1 * x2 * x3 + randrange(0, 10) -
5
    return y

x01 = (x1_max + x1_min) / 2
x02 = (x2_max + x2_min) / 2
x03 = (x3_max + x3_min) / 2
x1_delt = x1_max - x01
x2_delt = x2_max - x02
x3_delt = x3_max - 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 = [x1_min, x1_min, x1_min, x1_min, x1_max, x1_max, x1_max, x1_max, -1.73 *
x1_delt + x01, 1.73 * x1_delt + x01, x01, x01,
      x01, x01, x01]
x2 = [x2_min, x2_min, x2_max, x2_max, x2_min, x2_min, x2_max, x2_max, x02, x02,
-1.73 * x2_delt + x02, 1.73 * x2_delt + x02,
      x02, x02, x02]
x3 = [x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, x03, x03,
x03, x03, -1.73 * x3_delt + x03,
      1.73 * x3_delt + x03, x03]

x1x2, x1x3, x2x3, x1x2x3 = [0] * n, [0] * n, [0] * n, [0] * n
x1kv, x2kv, x3kv = [0] * n, [0] * n, [0] * n

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

print("Матриця планування з натуралізованими коефіцієнтами X")
print("      x1      x2      x3      x1x2      x1x3      x2x3
x1x2x3      x1x1"
      "      x2x2      x3x3")
for i in range(n):

```

```

        for j in range(len(list_for_a[0])):
            print("{:^12.3f}".format(list_for_a[i][j]), end='')
        print("")
# вивід матриці планування
Y = [[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)]
print("")

print("Матриця планування Y")
print("      Y1      Y2      Y3")
for i in range(n):
    print(end='')
    for j in range(len(Y[0])):
        print("{:^12.3f}".format(Y[i][j]), end='')
    print("")
# середні y
aver_y = []
for i in range(len(Y)):
    aver_y.append(np.mean(Y[i], axis=0))
print("")

print("Середні значення відгуку за рядками")
for i in range(n):
    print("{:.3f}".format(aver_y[i]))
# розрахунок дисперсій
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_kn(num):
    a = 0
    for j in range(n):
        a += aver_y[j] * list_for_a[j][num - 1] / n
    return a

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

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

det 1 = [
    [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)]]

det_2 = [my, find_kn(1), find_kn(2), find_kn(3), find_kn(4), find_kn(5),
find_kn(6), find_kn(7),
        find_kn(8), find_kn(9), find_kn(10)]

beta = solve(det_1, det_2)
print("")

print("Отримане рівняння регресії")
print("{:.3f} + {:.3f} * X1 + {:.3f} * X2 + {:.3f} * X3 + {:.3f} * X1X2 + {:.3f}
* X1X3 +\n + {:.3f} * X2X3"
      "+ {:.3f} * X1X2X3 + {:.3f} * X11^2 + {:.3f} * X22^2 + {:.3f} * X33^2
= y"
      .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] * n
print("")
print("Експериментальні значення")
for k in range(n):
    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(n):
    print("{:.3f}".format(y_i[i]))
print("")

print("\033[1m\033[30m\033[43m{}\033[0m".format("Перевірка за критерієм
Кохрена"))
Gp = max(dispersions) / sum(dispersions)
GT = 0.3346
print("Gp =", Gp)
if Gp < GT:
    print("\033[1m\033[30m\033[42mGp={} < GT={} - Дисперсія
однорідна\033[0m".format(round(Gp, 3), GT))
else:
    print("\033[1m\033[30m\033[41m{}\033[0m".format("Дисперсія неоднорідна"))
print("")
print("\033[1m\033[30m\033[43m{}\033[0m".format("Перевірка значущості
коефіцієнтів за критерієм Стьюдента"))
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (n * m)) ** 0.5

F3 = (m - 1) * n
coef_1 = []
coef_2 = []
d = 11
res = [0] * 11
for j in range(11):

```

```

pract_t = 0
for i in range(15):
    if j == 0:
        pract_t += aver_y[i] / 15
    else:
        pract_t += aver_y[i] * xn[i][j - 1]
    res[j] = beta[j]
if fabs(pract_t / sbs) < t.ppf(q=0.975, df=F3):
    coef_2.append(beta[j])
    res[j] = 0
    d -= 1
else:
    coef_1.append(beta[j])
print("Значущі коефіцієнти регресії", [round(i, 3) for i in coef_1])
print("Незначущі коефіцієнти регресії", [round(i, 3) for i in coef_2])
st_y = []
for i in range(n):
    st_y.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("")

print("Значення з отриманими коефіцієнтами")
for i in range(n):
    print("{:.3f}".format(st_y[i]))
print("")

print("\033[1m\033[30m\033[43m{}\033[0m".format("Перевірка адекватності за
критерієм Фішера"))
Sad = m * sum([(st_y[i] - aver_y[i]) ** 2 for i in range(n)]) / (n - d)
Fp = Sad / sb
F4 = n - d
print("Fp =", Fp)
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("\033[1m\033[30m\033[42m{}\033[0m".format("При рівні значимості 0.05
рівняння регресії адекватне"))
else:
    print("\033[1m\033[30m\033[41m{}\033[0m".format("При рівні значимості 0.05
рівняння регресії неадекватне"))

```

## Результат роботи програми

C:\Users\38096\anaconda3\python.exe "C:/Z Крі/А МНД/Лаб/Лаб 6/main.py"

Матриця планування з натуралізованими коефіцієнтами X

X1	X2	X3	X1X2	X1X3	X2X3	X1X2X3	X1X1	X2X2
20.000	25.000	25.000	500.000	500.000	625.000	12500.000	400.000	625.000
20.000	25.000	35.000	500.000	700.000	875.000	17500.000	400.000	625.000
20.000	65.000	25.000	1300.000	500.000	1625.000	32500.000	400.000	4225.000
20.000	65.000	35.000	1300.000	700.000	2275.000	45500.000	400.000	4225.000
70.000	25.000	25.000	1750.000	1750.000	625.000	43750.000	4900.000	625.000
70.000	25.000	35.000	1750.000	2450.000	875.000	61250.000	4900.000	625.000
70.000	65.000	25.000	4550.000	1750.000	1625.000	113750.000	4900.000	4225.000
70.000	65.000	35.000	4550.000	2450.000	2275.000	159250.000	4900.000	4225.000
1.750	45.000	30.000	78.750	52.500	1350.000	2362.500	3.062	2025.000
88.250	45.000	30.000	3971.250	2647.500	1350.000	119137.500	7788.062	2025.000
45.000	10.400	30.000	468.000	1350.000	312.000	14040.000	2025.000	108.160
45.000	79.600	30.000	3582.000	1350.000	2388.000	107460.000	2025.000	6336.160
45.000	45.000	21.350	2025.000	960.750	960.750	43233.750	2025.000	2025.000
45.000	45.000	38.650	2025.000	1739.250	1739.250	78266.250	2025.000	2025.000
45.000	45.000	30.000	2025.000	1350.000	1350.000	60750.000	2025.000	2025.000

# Матриця планування Y

Y1	Y2	Y3
14779.100	14786.100	14780.100
20791.100	20787.100	20794.100
31699.100	31699.100	31694.100
42020.100	42022.100	42022.100
39304.100	39309.100	39304.100
46616.100	46612.100	46613.100
72615.100	72617.100	72621.100
86246.100	86251.100	86244.100
18527.469	18526.469	18523.469
77129.219	77127.219	77123.219
18487.684	18483.684	18480.684
66525.244	66528.244	66532.244
34373.437	34375.437	34376.437
50493.846	50486.846	50487.846
42030.600	42031.600	42030.600

## Середні значення відгуку за рядками

14781.767  
 20790.767  
 31697.433  
 42021.433  
 39305.767  
 46613.767  
 72617.767  
 86247.100  
 18525.802  
 77126.552  
 18484.017  
 66528.577  
 34375.103  
 50489.513  
 42030.933

## Отримане рівняння регресії

$$-60.949 + 4.463 * X1 + 3.678 * X2 + 7.764 * X3 + 5.690 * X1X2 + 0.090 * X1X3 + 8.781 * X2X3 + 0.100 * X1X2X3 + 3.098 * X11^2 + 0.397 * X22^2 + 5.359 * X33^2 = y$$

## Експериментальні значення

14783.014  
 20790.775  
 31698.041  
 42020.802  
 39307.021  
 46613.782  
 72618.381  
 86246.475  
 18525.394  
 77126.128  
 18482.862  
 66528.901  
 34373.254  
 50490.530  
 42030.939

#### Перевірка за критерієм Кохрена

$G_p = 0.11315789473684208$

$G_p = 0.113 < G_T = 0.3346$  - Дисперсія однорідна

#### Перевірка значущості коефіцієнтів за критерієм Стьюдента

Значущі коефіцієнти регресії  $[-60.949, 4.463, 3.678, 7.764, 5.69, 0.09, 8.781, 0.1, 3.098, 0.397, 5.359]$

Незначущі коефіцієнти регресії  $[]$

Значення з отриманими коефіцієнтами

14783.014

20790.775

31698.041

42020.802

39307.021

46613.782

72618.381

86246.475

18525.394

77126.128

18482.862

66528.901

34373.254

50490.530

42030.939

#### Перевірка адекватності за критерієм Фішера

$F_p = 1.452591796700626$

При рівні значимості 0.05 рівняння регресії адекватне

Process finished with exit code 0