

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»

Виконав:  
студент II курсу ФІОТ  
групи ІВ-93  
Підгайний Д. Р.  
ПЕРЕВІРИВ:  
ас. Регіда П. Г.

Київ - 2021

**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; +1; -1; 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.

## Варіант: 318

|     |    |    |     |    |    |    |  |
|-----|----|----|-----|----|----|----|--|
| 318 | 20 | 70 | -15 | 45 | 20 | 35 | $2,4+0,5*x_1+6,0*x_2+10,0*x_3+4,2*x_1*x_1+0,9*x_2*x_2+2,2*x_3*x_3+0,8*x_1*x_2+0,9*x_1*x_3+7,2*x_2*x_3+4,7*x_1*x_2*x_3$ |
|-----|----|----|-----|----|----|----|--|

## Код 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

# варіант 318
x1min = 20
x1max = 70
x2min = -15
x2max = 45
x3min = 20
x3max = 35

def get_y(x1, x2, x3):
    return 2.4 + 0.5 * x1 + 6.0 * x2 + 10.0 * x3 + 4.2 * x1 * x1 + 0.9 * x2
* x2 + 2.2 * x3 * x3\
    + 0.8 * x1 * x2 + 0.9 * x1 * x3 + 7.2 * x2 * x3 + 4.7 * x1 * x2
* x3 + randrange(0, 10) - 5

x01 = (x1max + x1min) / 2
x02 = (x2max + x2min) / 2
x03 = (x3max + x3min) / 2
deltax1 = x1max - x01
deltax2 = x2max - x02
deltax3 = x3max - x03
# 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, -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]
# заповнення нулями x1x2, x1x3, x1x2x3
# заповнення нулями x1kv, x2kv, x3kv
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
# формуємо список a
list_for_a = list(zip(x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1kv, x2kv,
x3kv))

print("Матриця планування з натуралізованими коефіцієнтами X:")
print("      X1      X2      X3      X1X2      X1X3
X2X3      X1X2X3      X1X1"
      "      X2X2      X3X3")
for i in range(n):
    print(end=' ')
    for j in range(len(list_for_a[0])):
        print("{:^12.3f}".format(list_for_a[i][j]), end=' ')
    print("")

# вивід матриці планування
Y = [[get_y(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("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

```

```

Y_average = [np.mean(Y[i], axis=0) for i in range(len(Y))]

print("Середні значення відгуку за рядками:")
for i in range(n):
    print("{:.3f}".format(Y_average[i]), end=" ")
# 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_known(num):
    a = 0
    for j in range(n):
        a += Y_average[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(Y_average) / 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))

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] * n
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]), end=" ")
print("\n----- Перевірка 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("----- Перевірка значущості коефіцієнтів за критерієм
Стьюдента -----")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (n * 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(n):
    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(n):
    print("{:.3f}".format(y_st[i]), end=" ")

print("\n----- Перевірка 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
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")

```

## Роздруківка результатів виконання програми:

Матриця планування з натуралізованими коефіцієнтами X:

| X1     | X2      | X3     | X1X2      | X1X3     | X2X3      | X1X2X3     | X1X1     | X2X2     | X3X3     |
|--------|---------|--------|-----------|----------|-----------|------------|----------|----------|----------|
| 20.000 | -15.000 | 20.000 | -300.000  | 400.000  | -300.000  | -6000.000  | 400.000  | 225.000  | 400.000  |
| 20.000 | -15.000 | 35.000 | -300.000  | 700.000  | -525.000  | -10500.000 | 400.000  | 225.000  | 1225.000 |
| 20.000 | 45.000  | 20.000 | 900.000   | 400.000  | 900.000   | 18000.000  | 400.000  | 2025.000 | 400.000  |
| 20.000 | 45.000  | 35.000 | 900.000   | 700.000  | 1575.000  | 31500.000  | 400.000  | 2025.000 | 1225.000 |
| 70.000 | -15.000 | 20.000 | -1050.000 | 1400.000 | -300.000  | -21000.000 | 4900.000 | 225.000  | 400.000  |
| 70.000 | -15.000 | 35.000 | -1050.000 | 2450.000 | -525.000  | -36750.000 | 4900.000 | 225.000  | 1225.000 |
| 70.000 | 45.000  | 20.000 | 3150.000  | 1400.000 | 900.000   | 63000.000  | 4900.000 | 2025.000 | 400.000  |
| 70.000 | 45.000  | 35.000 | 3150.000  | 2450.000 | 1575.000  | 110250.000 | 4900.000 | 2025.000 | 1225.000 |
| 1.750  | 15.000  | 27.500 | 26.250    | 48.125   | 412.500   | 721.875    | 3.062    | 225.000  | 756.250  |
| 88.250 | 15.000  | 27.500 | 1323.750  | 2426.875 | 412.500   | 36403.125  | 7788.062 | 225.000  | 756.250  |
| 45.000 | -36.900 | 27.500 | -1660.500 | 1237.500 | -1014.750 | -45663.750 | 2025.000 | 1361.610 | 756.250  |
| 45.000 | 66.900  | 27.500 | 3010.500  | 1237.500 | 1839.750  | 82788.750  | 2025.000 | 4475.610 | 756.250  |
| 45.000 | 15.000  | 14.525 | 675.000   | 653.625  | 217.875   | 9804.375   | 2025.000 | 225.000  | 210.976  |
| 45.000 | 15.000  | 40.475 | 675.000   | 1821.375 | 607.125   | 27320.625  | 2025.000 | 225.000  | 1638.226 |
| 45.000 | 15.000  | 27.500 | 675.000   | 1237.500 | 412.500   | 18562.500  | 2025.000 | 225.000  | 756.250  |

Матриця планування Y:

| Y1          | Y2          | Y3          |
|-------------|-------------|-------------|
| -27356.100  | -27360.100  | -27351.100  |
| -47895.100  | -47886.100  | -47887.100  |
| 97022.900   | 97028.900   | 97026.900   |
| 167569.900  | 167567.900  | 167572.900  |
| -78626.100  | -78629.100  | -78627.100  |
| -151365.100 | -151366.100 | -151366.100 |
| 330151.900  | 330150.900  | 330149.900  |
| 559989.900  | 559994.900  | 559994.900  |
| 8671.513    | 8674.513    | 8677.513    |
| 212299.513  | 212290.513  | 212290.513  |
| -210667.776 | -210663.776 | -210665.776 |
| 420770.574  | 420770.574  | 420770.574  |

```

420770.574 420770.574 420770.574
58210.321 58208.321 58205.321
147790.721 147789.721 147784.721
102626.650 102624.650 102623.650
Середні значення відгуку за рядками:
-27355.767 -47889.433 97026.233 167570.233 -78627.433 -151365.767 330150.900 559993.233 8674.513 212293.513 -210665.776 420770.574 58207.988 147788.388 102624.983
Отримане рівняння регресії:
12.893 + 0.483 * X1 + 6.075 * X2 + 9.069 * X3 + 0.797 * X1X2 + 0.895 * X1X3 + 7.196 * X2X3 + 4.700 * X1X2X3 + 4.202 * X11^2 + 0.901 * X22^2 + 2.221 * X33^2 = y
Експериментальні значення:
-27355.889 -47888.702 97024.981 167569.835 -78628.015 -151365.494 330149.189 559992.376 8674.636 212294.698 -210666.428 420772.535 58209.630 147788.056 102624.974
----- Перевірка за критерієм Кохрена -----
Gr = 0.20558375634517767
Дисперсія однорідна
----- Перевірка значущості коефіцієнтів за критерієм Стюдента -----
Значущі коефіцієнти регресії: [12.893, 0.483, 6.075, 9.069, 0.797, 0.895, 7.196, 4.7, 4.202, 0.901, 2.221]
Незначущі коефіцієнти регресії: []
Значення з отриманими коефіцієнтами:
-27355.889 -47888.702 97024.981 167569.835 -78628.015 -151365.494 330149.189 559992.376 8674.636 212294.698 -210666.428 420772.535 58209.630 147788.056 102624.974
----- Перевірка адекватності за критерієм Фішера -----
Fr = 1.90774200842572
Рівняння регресії адекватне при рівні значимості 0.05

```

## Висновки:

Під час виконання лабораторної роботи було змодельовано трьохфакторний експеримент при використанні лінійного рівняння регресії, рівняння регресії з ефектом взаємодії та рівняння регресії з квадратичними членами, складено матрицю планування експерименту, було визначено коефіцієнти рівнянь регресії (натуралізовані та нормовані), для форми з квадратичними членами - натуралізовані, виконано перевірку правильності розрахунку коефіцієнтів рівнянь регресії. Також було проведено 3 статистичні перевірки(використання критеріїв Кохрена, Стюдента та Фішера) для кожної форми рівняння регресії . При виявленні неадекватності лінійного рівняння регресії оригіналу було застосовано ефект взаємодії факторів, при неадекватності і такого рівняння регресії було затосовано рівняння регресії з квадратичними членами. Довірча ймовірність в даній роботі дорівнює 0.95, відповідно рівень значимості  $\alpha = 0.05$ .