

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 6  
з дисципліни «Методи наукових досліджень»  
на тему «Проведення трьохфакторного експерименту при використанні  
рівняння регресії з квадратичними членами»

ВИКОНАВ:  
студент 2 курсу  
групи ІВ-91  
Карамшук Володимир  
Залікова – 9113

ПЕРЕВІРИВ:  
ас. Регіда П. Г.

**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; + $\bar{1}$ ; - $\bar{1}$ ; 0 для  $\bar{x}_1, \bar{x}_2, \bar{x}_3$ .
3. Значення функції відгуку знайти за допомогою підстановки в формулу:

$$y_1 = f(x_1, x_2, x_3) + \text{random}(10) - 5,$$

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.

113	-15	30	5	40	5	25	$6,7+2,8*x_1+1,3*x_2+8,9*x_3+6,3*x_1*x_1+0,6*x_2*x_2+8,1*x_3*x_3+5,4*x_1*x_2+0,2*x_1*x_3+4,9*x_2*x_3+5,2*x_1*x_2*x_3$
-----	-----	----	---	----	---	----	---

## Програмний код

```
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 113
x1min = -15
x1max = 30
x2min = 5
x2max = 40
x3min = 5
x3max = 25

def function(X1, X2, X3):
    y = 6.7 + 2.8 * X1 + 1.3 * X2 + 8.9 * X3 + 6.3 * X1 * X1 + 0.6 * X2 * X2 + 8.1 * X3 * X3 + 5.4 * X1 * X2 + \
        0.2 * X1 * X3 + 4.9 * X2 * X3 + 5.2 * X1 * X2 * X3 + randrange(0, 10) - 5
    return y

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, 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 = [[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("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 = []
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(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 -----\n-----")
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\nСтьюдента -----")
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\n-----")
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")

```

## Результат роботи програми

```
C:\Users\qazse\anaconda3\python.exe C:/Users/qazse/PycharmProjects/MND_6/main.py
Матриця планування з натуралізованими коефіцієнтами X:
  X1      X2      X3      X1X2      X1X3      X2X3      X1X2X3      X1X1      X2X2      X3X3
-15.000    5.000    5.000    -75.000    -75.000     25.000    -375.000    225.000    25.000    25.000
-15.000    5.000    25.000    -75.000    -375.000    125.000    -1875.000    225.000    25.000    625.000
-15.000    40.000    5.000    -600.000    -75.000    200.000    -3000.000    225.000    1600.000    25.000
-15.000    40.000    25.000    -600.000    -375.000    1000.000    -15000.000    225.000    1600.000    625.000
 30.000    5.000    5.000    150.000    150.000     25.000     750.000    900.000    25.000    25.000
 30.000    5.000    25.000    150.000    750.000    125.000    3750.000    900.000    25.000    625.000
 30.000    40.000    5.000    1200.000    150.000    200.000     6000.000    900.000    1600.000    25.000
 30.000    40.000    25.000    1200.000    750.000    1000.000    30000.000    900.000    1600.000    625.000
-31.425    22.500    15.000    -707.062    -471.375    337.500    -10605.937    987.531    506.250    225.000
 46.425    22.500    15.000    1044.562    696.375    337.500    15668.438    2155.281    506.250    225.000
  7.500    -7.775    15.000    -58.312    112.500    -116.625    -874.687    56.250     60.451    225.000
  7.500    52.775    15.000    395.812    112.500    791.625    5937.188    56.250    2785.201    225.000
  7.500    22.500    -2.300    168.750    -17.250    -51.750    -388.125    56.250    506.250     5.290
  7.500    22.500    32.300    168.750    242.250    726.750    5450.625    56.250    506.250    1043.290
  7.500    22.500    15.000    168.750    112.500    337.500    2531.250    56.250    506.250    225.000

Матриця планування Y:
  Y1      Y2      Y3
-596.800  -592.800  -597.800
-2926.800 -2931.800  -2926.800
-15229.800 -15235.800  -15230.800
-68731.800 -68731.800  -68737.800
10895.700  10895.700   10887.700
32141.700  32142.700  32135.700
45710.700  45704.700  45709.700
179585.700 179590.700  179590.700
-48982.385 -48985.385  -48979.385
104915.495 104912.495  104913.495
-3049.987  -3051.987   -3052.987
40990.028  40989.028  40990.028
-629.571   -624.571   -629.571
42321.219  42318.219   42312.219
18423.075  18424.075   18418.075

Середні значення відгуку за рядками:
-595.800 -2928.467 -15232.133 -68733.800 10893.033 32140.033 45708.367 179589.033 -48982.385 104913.829 -3051.654 40989.695 -627.904 42317.219 18421.742
Отримане рівняння регресії:
6.200 + 2.846 * X1 + 1.369 * X2 + 8.874 * X3 + 5.397 * X1X2 + 0.199 * X1X3 + 4.903 * X2X3+ 5.200 * X1X2X3 + 6.300 * X11^2 + 0.598 * X22^2 + 8.099 * X33^2 = y
Експериментальні значення:
-597.594 -2930.073 -15233.174 -68734.653 10892.173 32139.361 45708.260 179589.114 -48980.160 104913.894 -3049.638 40989.969 -626.543 42318.148 18421.725
----- Перевірка за критерієм Кохрена -----
Gr = 0.14645308924485123
Дисперсія однорідна

----- Перевірка значущості коефіцієнтів за критерієм Стьюдента -----
Значущі коефіцієнти регресії: [6.2, 2.846, 1.369, 8.874, 5.397, 0.199, 4.903, 5.2, 6.3, 0.598, 8.099]
Незначущі коефіцієнти регресії: []
Значення з отриманими коефіцієнтами:
-597.594 -2930.073 -15233.174 -68734.653 10892.173 32139.361 45708.260 179589.114 -48980.160 104913.894 -3049.638 40989.969 -626.543 42318.148 18421.725
----- Перевірка адекватності за критерієм Фішера -----
Fr = 2.389659447530584
Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0
```