

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №6  
з дисципліни «Методи оптимізації планування експерименту»  
на тему: «Проведення трьохфакторного експерименту при використанні  
рівняння регресії з урахуванням квадратичних членів»

Виконала:  
студентка групи ІО-91  
Кійченко А. К.

Перевірив:  
Регіда П. Г.

Київ 2021

**Мета:** Провести трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Знайти рівняння регресії, яке буде адекватним для опису 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.

$$y_{i \max} = 200 + x_{i \max}$$

$$y_{i \min} = 200 + x_{i \min}$$

$$\text{где } x_{i \max} = \frac{x_{1 \max} + x_{2 \max} + x_{3 \max}}{3}, \quad x_{i \min} = \frac{x_{1 \min} + x_{2 \min} + x_{3 \min}}{3}$$

4. Розрахувати коефіцієнти рівняння регресії і записати його.
5. Провести 3 статистичні перевірки.

Завдання за варіантом:

111	-9	1	-2	3	-2	4
-----	----	---	----	---	----	---

## Лістинг програми

```
import random
import numpy as np
from pyDOE2 import *
import math

def get_avg(array):
    result = []
    for i in range(len(array[0])):
        result.append(0)
        for j in range(len(array)):
            result[i] += array[j][i]
        result[i] = result[i]/len(array)
    return result

def get_dispersion(array, avg_y):
    result = []
    for i in range(len(array[0])):
        result.append(0)
        for j in range(len(array)):
            result[i] += (array[j][i] - avg_y[i])**2
        result[i] = result[i]/3
    return result

def add_sq_nums(x):
    for i in range(len(x)):
        x[i][3] = x[i][0] * x[i][1]
        x[i][4] = x[i][0] * x[i][2]
        x[i][5] = x[i][1] * x[i][2]
        x[i][6] = x[i][0] * x[i][1] * x[i][2]
        x[i][7] = x[i][0] ** 2
        x[i][8] = x[i][1] ** 2
        x[i][9] = x[i][2] ** 2
    return x

def plan_matrix5(n, m, x_norm):
    l = 1.215
    x_norm = np.array(x_norm)
    x_norm = np.transpose(x_norm)
    x = np.ones(shape=(len(x_norm), len(x_norm[0])))
    for i in range(8):
        for j in range(3):
            if x_norm[i][j] == -1:
                x[i][j] = x_range[j][0]
            else:
                x[i][j] = x_range[j][1]
    for i in range(8, len(x)):
        for j in range(3):
            x[i][j] = float((x_range[j][0] + x_range[j][1]) / 2)
    dx = [x_range[i][1] - (x_range[i][0] + x_range[i][1]) / 2 for i in range(3)]
```

```

x[8][0] = (-1 * dx[0]) + x[9][0]
x[9][0] = (1 * dx[0]) + x[9][0]
x[10][1] = (-1 * dx[1]) + x[9][1]
x[11][1] = (1 * dx[1]) + x[9][1]
x[12][2] = (-1 * dx[2]) + x[9][2]
x[13][2] = (1 * dx[2]) + x[9][2]
x = add_sq_nums(x)
for i in range(len(x)):
    for j in range(len(x[i])):
        x[i][j] = round(x[i][j], 3)
return np.transpose(x).tolist()

def s_kv(y, y_aver, n, m):
    res = []
    for i in range(n):
        s = sum([(y_aver[i] - y[i][j]) ** 2 for j in range(m)]) / m
        res.append(round(s, 3))
    return res

def bs(x, y_aver, n):
    res = [sum(1 * y for y in y_aver) / n]

    for i in range(len(x[0])):
        b = sum(j[0] * j[1] for j in zip(x[:, i], y_aver)) / n
        res.append(b)
    return res

def student(x, y, y_aver, n, m):
    S_kv = s_kv(y, y_aver, n, m)
    s_kv_aver = sum(S_kv) / n

    s_Bs = (s_kv_aver / n / m) ** 0.5
    Bs = bs(x, y_aver, n)
    ts = [round(abs(B) / s_Bs, 3) for B in Bs]

    return ts

# todo: change to your variant
x1min = -9
x1max = 1
x2min = -2
x2max = 3
x3min = -2
x3max = 4
n = 15
m = 3
x_range = [[x1min, x1max], [x2min, x2max], [x3min, x3max]]

x_min_avg = sum([x1min, x2min, x3min]) / 3
x_max_avg = sum([x1max, x2max, x3max]) / 3

y_min = 200 + x_min_avg
y_max = 200 + x_max_avg

```

```

x_norm = [[-1, -1, -1, -1, 1, 1, 1, 1, -1.215, 1.215, 0, 0, 0, 0, 0],
          [-1, -1, 1, 1, -1, -1, 1, 1, 0, 0, -1.215, 1.215, 0, 0, 0],
          [-1, 1, -1, 1, -1, 1, -1, 1, 0, 0, 0, 0, -1.215, 1.215, 0],
          [1, 1, -1, -1, -1, -1, 1, 1, 0, 0, 0, 0, 0, 0, 0],
          [1, -1, 1, -1, -1, 1, -1, 1, 0, 0, 0, 0, 0, 0, 0],
          [1, -1, -1, 1, 1, -1, -1, 1, 0, 0, 0, 0, 0, 0, 0],
          [-1, 1, 1, -1, 1, -1, -1, 1, 0, 0, 0, 0, 0, 0, 0],
          [1, 1, 1, 1, 1, 1, 1, 1, 1.4623, 1.4623, 0, 0, 0, 0, 0],
          [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1.4623, 1.4623, 0, 0, 0],
          [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1.4623, 1.4623, 0]]

y1 = []
y2 = []
y3 = []
for i in range(15):
    y1.append(random.randint(int(y_min), int(y_max)))
    y2.append(random.randint(int(y_min), int(y_max)))
    y3.append(random.randint(int(y_min), int(y_max)))
# y1 = [196, 198, 196, 202, 202, 202, 198, 203, 203, 199, 201, 193, 203, 197, 195]
# y2 = [201, 195, 198, 203, 203, 196, 195, 198, 201, 200, 200, 202, 201, 203, 201]
# y3 = [194, 200, 193, 195, 200, 194, 193, 200, 198, 203, 195, 197, 198, 197, 203]

print("Матриця планування:\n X1 | X2 | X3 | X1*X2 | X1*X3 | X2*X3 | X1*X2*X3 | X1\u00b2 | X2\u00b2 | X3\u00b2")
for i in range(15):

print(f"{x_norm[0][i]:^6}|{x_norm[1][i]:^6}|{x_norm[2][i]:^6}|{x_norm[3][i]:^5}|{x_norm[4][i]:^5}|{x_norm[5][i]:^5}|{x_norm[6][i]:^8}|{x_norm[7][i]:^6}|{x_norm[8][i]:^6}|{x_norm[9][i]:^6}")

avg_y = get_avg([y1, y2, y3])
print("\n Y1 | Y2 | Y3 | Y\u00304")
for i in range(15):
    print(f"{y1[i]:^4}|{y2[i]:^4}|{y3[i]:^4}|{avg_y[i]:^6}")

x= plan_matrix5(n, m, x_norm)
print("Матриця планування з натуралізованими значеннями факторів:")
    "\n X1 | X2 | X3 | X1*X2 | X1*X3 | X2*X3 | X1*X2*X3 | X1\u00b2 | X2\u00b2 | X3\u00b2")
for i in range(15):

print(f"{x[0][i]:^6}|{x[1][i]:^8}|{x[2][i]:^6}|{x[3][i]:^7}|{x[4][i]:^7}|{x[5][i]:^7}|{x[6][i]:^8}|{x[7][i]:^6}|{x[8][i]:^8}|{x[9][i]:^6}")

y_sum = sum(avg_y)
b = [0] * 11
b[0] = y_sum
for i in range(15):
    b[1] += avg_y[i] * x_norm[0][i]
    b[2] += avg_y[i] * x_norm[1][i]
    b[3] += avg_y[i] * x_norm[2][i]
    b[4] += avg_y[i] * x_norm[0][i] * x_norm[1][i]
    b[5] += avg_y[i] * x_norm[0][i] * x_norm[2][i]
    b[6] += avg_y[i] * x_norm[1][i] * x_norm[2][i]
    b[7] += avg_y[i] * x_norm[0][i] * x_norm[1][i] * x_norm[2][i]

b[8] = (b[1]**2)/b[0]

```

```

b[9] = (b[2]**2)/b[0]
b[10] = (b[3]**2)/b[0]
for i in range(11):
    b[i] = b[i] / 15

result = []
for i in range(15):
    result.append(b[0] + b[1] * x_norm[0][i] + b[2] * x_norm[1][i] + b[3] * x_norm[2][i]
+ b[4] * x_norm[3][i] + b[5] * x_norm[4][i] + b[6] * x_norm[5][i] + b[7] * x_norm[6][i]
+ b[8] * x_norm[7][i] + b[9] * x_norm[8][i] + b[10] * x_norm[9][i])

print("b0 = {:.3f}, b1 = {:.3f}, b2 = {:.3f}, b3 = {:.3f},"
      " b12 = {:.3f}, b13 = {:.3f}, b23 = {:.3f}, b123 = {:.3f}, b11 = {:.3f}, b22 =
{:.3f}, b33 = {:.3f}".format(b[0], b[1], b[2], b[3], b[4], b[5], b[6], b[7], b[8], b[9],
b[10]))

print("Підставимо значення факторів з матриці планування")
for i in range(15):
    print("{:.3f} + x1 * {:.3f}"
          " + x2 * {:.3f} + x3 * {:.3f}"
          " + x12 * {:.3f} + x13 * {:.3f}"
          " + x23 * {:.3f} + x123 * {:.3f}"
          " + x11 * {:.3f} + x22 * {:.3f}"
          " + x33 * {:.3f} = {:.3f}".format(b[0], b[1], b[2], b[3], b[4], b[5], b[6],
b[7], b[8], b[9], b[10], result[i]))

# Перевірка однорідності дисперсії
sigma = get_dispersion([y1, y2, y3], avg_y)
print("Значення дисперсії по рядках: \u03c3\u00b2(y1) = {:.2f},"
      " \u03c3\u00b2(y2) = {:.2f},"
      " \u03c3\u00b2(y3) = {:.2f},"
      " \u03c3\u00b2(y4) = {:.2f},"
      " \u03c3\u00b2(y5) = {:.2f},"
      " \u03c3\u00b2(y6) = {:.2f},"
      " \u03c3\u00b2(y7) = {:.2f},"
      " \u03c3\u00b2(y8) = {:.2f}"
      ".format(sigma[0], sigma[1], sigma[2], sigma[3], sigma[4], sigma[5], sigma[6],
sigma[7]))
print(" \u03c3\u00b2(y9) = {:.2f}"
      " \u03c3\u00b2(y10) = {:.2f}"
      " \u03c3\u00b2(y11) = {:.2f}"
      " \u03c3\u00b2(y12) = {:.2f}"
      " \u03c3\u00b2(y13) = {:.2f}"
      " \u03c3\u00b2(y14) = {:.2f}"
      " \u03c3\u00b2(y15) = {:.2f}"
      ".format(sigma[8], sigma[9], sigma[10], sigma[11], sigma[12], sigma[13],
sigma[14]))
gp = max(sigma)/sum(sigma)
print("Gp = ", gp)
f1 = m-1
f2 = n
if gp < 0.3346:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")
    exit()

```

```

# оцінка значимості коефіцієнтів регресії за критерієм Стьюдента
sb = sum(sigma) / n
s2bs = sb / (n * m)
sbs = math.sqrt(s2bs)
print("S\u00b2b = {:.3f}, S\u00b2(\u03b2s) = {:.3f}, S(\u03b2s) = {:.3f}".format(sb,
s2bs, sbs))

betha = []
for i in range(4):
    betha.append(0)
    for j in range(15):
        betha[i] += avg_y[j] * x_norm[i][j]
print("\u03b20 = {:.3f}, \u03b21 = {:.3f}, \u03b22 = {:.3f}, \u03b23 =
{:.3f}".format(betha[0], betha[1], betha[2], betha[3]))

x_norm.insert(0, [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
f1 = m - 1
f2 = n
f3 = f1 * f2
t = student(np.transpose(np.array(x_norm))[:, 1:], np.transpose(np.array([y1, y2, y3])),
avg_y, 15, 3)
print("t0 = {:.3f}, t1 = {:.3f}, "
      " t2 = {:.3f}, t3 = {:.3f}, "
      " t4 = {:.3f}, t5 = {:.3f}, "
      " t6 = {:.3f}, t7 = {:.3f}, "
      " t8 = {:.3f}, t9 = {:.3f}, "
      " t10 = {:.3f}".format(t[0], t[1], t[2], t[3], t[4], t[5], t[6], t[7], t[8], t[9],
t[10]))
t_table = 2.042

y_ = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
d = 0
nm = []
for i in range(11):
    if t[i] < t_table:
        nm.append(i)
        d += 1
    else:
        for j in range(15):
            if i == 0:
                y_[j] += b[i]
            else:
                y_[j] += b[i] * x[2][j]
print("d = {}".format(d))
print("Підставимо значення факторів з матриці планування, y =", y_[0], y_[1], y_[2],
y_[3], y_[4], y_[5], y_[6], y_[7], y_[8], y_[9], y_[10])

# критерій Фішера
s2ad = (m / (n - d)) * sum((y_[i] - avg_y[i])**2 for i in range(4))
print("S\u00b2ад = {:.3f}".format(s2ad))

Fp = s2ad / sb
f4 = n - d
print("Критерій Фішера Fp =", Fp)

F8_table = 2.16

```

```
if Fp < F8_table:
    print("Рівняння регресії адекватно оригіналу при рівні значимості 0.05")
else:
    print("Рівняння регресії неадекватно оригіналу при рівні значимості 0.05")
```

## Результат виконання роботи

Запишемо лінійне рівняння регресії:

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 + b_{13}x_1x_3 + b_{23}x_2x_3 + b_{123}x_1x_2x_3 + b_{11}x_1^2 + b_{22}x_2^2 + b_{33}x_3^2$$

Матриця планування:

X1	X2	X3	X1*X2	X1*X3	X2*X3	X1*X2*X3	X1 <sup>2</sup>	X2 <sup>2</sup>	X3 <sup>2</sup>
-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
-1.215	0	0	0	0	0	0	1.4623	0	0
1.215	0	0	0	0	0	0	1.4623	0	0
0	-1.215	0	0	0	0	0	0	1.4623	0
0	1.215	0	0	0	0	0	0	1.4623	0
0	0	-1.215	0	0	0	0	0	0	1.4623
0	0	1.215	0	0	0	0	0	0	1.4623
0	0	0	0	0	0	0	0	0	0

Y1 | Y2 | Y3 |  $\bar{Y}$

199 | 202 | 196 | 199.0



195 | 200 | 199 | 198.0  
 196 | 198 | 196 | 196.66666666666666  
 200 | 197 | 196 | 197.66666666666666  
 202 | 200 | 195 | 199.0  
 200 | 199 | 199 | 199.33333333333334  
 195 | 202 | 198 | 198.33333333333334  
 199 | 202 | 200 | 200.33333333333334  
 196 | 201 | 200 | 199.0  
 196 | 196 | 196 | 196.0  
 201 | 199 | 200 | 200.0  
 196 | 200 | 201 | 199.0  
 197 | 198 | 198 | 197.66666666666666  
 197 | 197 | 196 | 196.66666666666666  
 201 | 198 | 201 | 200.0

Матриця планування з натуралізованими значеннями факторів:

X1 X3 <sup>2</sup>	X2	X3	X1*X2	X1*X3	X2*X3	X1*X2*X3	X1 <sup>2</sup>	X2 <sup>2</sup>	
-9.0 4.0	-2.0	-2.0	18.0	18.0	4.0	-36.0	81.0	4.0	
-9.0 16.0	-2.0	4.0	18.0	-36.0	-8.0	72.0	81.0	4.0	
-9.0 4.0	3.0	-2.0	-27.0	18.0	-6.0	54.0	81.0	9.0	
-9.0 16.0	3.0	4.0	-27.0	-36.0	12.0	-108.0	81.0	9.0	
1.0 4.0	-2.0	-2.0	-2.0	-2.0	4.0	4.0	1.0	4.0	
1.0 16.0	-2.0	4.0	-2.0	4.0	-8.0	-8.0	1.0	4.0	
1.0 4.0	3.0	-2.0	3.0	-2.0	-6.0	-6.0	1.0	9.0	

1.0 | 3.0 | 4.0 | 3.0 | 4.0 | 12.0 | 12.0 | 1.0 | 9.0 |  
16.0

-10.075 | 0.5 | 1.0 | -5.038 | -10.075 | 0.5 | -5.038 | 101.506 | 0.25  
| 1.0

2.075 | 0.5 | 1.0 | 1.038 | 2.075 | 0.5 | 1.038 | 4.306 | 0.25 |  
1.0

-4.0 | -2.538 | 1.0 | 10.15 | -4.0 | -2.538 | 10.15 | 16.0 | 6.439 |  
1.0

-4.0 | 3.538 | 1.0 | -14.15 | -4.0 | 3.538 | -14.15 | 16.0 | 12.514 |  
1.0

-4.0 | 0.5 | -2.645 | -2.0 | 10.58 | -1.323 | 5.29 | 16.0 | 0.25  
| 6.996

-4.0 | 0.5 | 4.645 | -2.0 | -18.58 | 2.323 | -9.29 | 16.0 | 0.25  
| 21.576

-4.0 | 0.5 | 1.0 | -2.0 | -4.0 | 0.5 | -2.0 | 16.0 | 0.25 |  
1.0

b0 = 198.444, b1 = 0.135, b2 = -0.237, b3 = 0.075, b12 = 0.200, b13 =  
0.156, b23 = 0.244, b123 = -0.022, b11 = 0.000, b22 = 0.000, b33 = 0.000

Підставимо значення факторів з матриці планування

$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 199.094$

$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.399$

$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 197.688$

$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.059$

$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.608$

$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.624$

$$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.091$$

$$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.995$$

$$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.281$$

$$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.608$$

$$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.732$$

$$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.157$$

$$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.354$$

$$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.535$$

$$198.444 + x_1 * 0.135 + x_2 * -0.237 + x_3 * 0.075 + x_{12} * 0.200 + x_{13} * 0.156 + x_{23} * 0.244 + x_{123} * -0.022 + x_{11} * 0.000 + x_{22} * 0.000 + x_{33} * 0.000 = 198.444$$

Значення дисперсії по рядках:  $\sigma^2(y_1) = 6.00$ ,  $\sigma^2(y_2) = 4.67$ ,  $\sigma^2(y_3) = 0.89$ ,  $\sigma^2(y_4) = 2.89$ ,  $\sigma^2(y_5) = 8.67$ ,  $\sigma^2(y_6) = 0.22$ ,  $\sigma^2(y_7) = 8.22$ ,  $\sigma^2(y_8) = 1.56$

$\sigma^2(y_9) = 4.67$   $\sigma^2(y_{10}) = 0.00$   $\sigma^2(y_{11}) = 0.67$   $\sigma^2(y_{12}) = 4.67$   $\sigma^2(y_{13}) = 0.22$   $\sigma^2(y_{14}) = 0.22$   $\sigma^2(y_{15}) = 2.00$

Gp = 0.1902439024390244

Дисперсія однорідна

$S^2b = 3.037$ ,  $S^2(\beta_s) = 0.067$ ,  $S(\beta_s) = 0.260$

$\beta_0 = 2.022$ ,  $\beta_1 = -3.548$ ,  $\beta_2 = 1.118$ ,  $\beta_3 = 3.000$

$t_0 = 763.859$ ,  $t_1 = 0.519$ ,  $t_2 = 0.911$ ,  $t_3 = 0.287$ ,  $t_4 = 0.770$ ,  $t_5 = 0.599$ ,  
 $t_6 = 0.941$ ,  $t_7 = 0.086$ ,  $t_8 = 555.814$ ,  $t_9 = 557.315$ ,  $t_{10} = 555.564$

$d = 7$

Підставимо значення факторів з матриці планування,  $y = 198.44364137725515$   
198.44605057882288 198.44364137725515 198.44605057882288  
198.44364137725515 198.44605057882288 198.44364137725515  
198.44605057882288 198.44484597803904 198.44484597803904  
198.44484597803904

$S^2_{ад} = 1.603$

Критерій Фішера  $F_p = 0.5276822098874243$

Рівняння регресії адекватно оригіналу при рівні значимості 0.05

## **Висновки:**

В результаті виконання роботи було досягнуто поставленої мети: проведено трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план та знайдено рівняння регресії, яке буде адекватним для опису об'єкту.

Результатом виконання програми є рівняння з 4 значимими коефіцієнтами, котре адекватне оригіналу.