

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 No 4**

Тема: Проведення трьохфакторного експерименту  
при використанні рівняння регресії з урахуванням ефекту взаємодії.

ВИКОНАВ:  
студент II курсу ФІОТ  
групи ІО-83  
Афанасьєв М. О.  
Варіант-303

ПЕРЕВІРИВ:  
Регіда П.Г.

Київ – 2020

## Мета роботи

Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

## Завдання на лабораторну роботу

1. Скласти матрицю планування для повного трьохфакторного експерименту.
2. Провести експеримент, повторивши N раз досліди у всіх точках факторного простору і знайти значення відгуку Y.
3. Знайти коефіцієнти рівняння регресії і записати його.
4. Провести 3 статистичні перевірки – за критеріями Кохрена, Стюдента, Фішера.
5. Зробити висновки по адекватності регресії та значимості окремих коефіцієнтів і записати скореговане рівняння регресії.
6. Написати комп'ютерну програму, яка усе це моделює.

```
import random
import numpy as np
import copy
from scipy.stats import t, f

def tss(prob, f3):
    x_vec = [i*0.0001 for i in range(int(5/0.0001))]
    par = 0.5 + prob/0.1*0.05
    for i in x_vec:
        if abs(t.cdf(i, f3) - par) < 0.000005:
            return i

def taf(prob, d, f3):
    x_vec = [i*0.001 for i in range(int(10/0.001))]
    for i in x_vec:
        if abs(f.cdf(i, 4-d, f3)-prob) < 0.0001:
            return i

#варіант 303

x1min = 20
```

```
x1max = 70
x2min = -15
x2max = 45
x3min = 20
x3max = 35
```

```
x01=(x1max + x1min)/2
x02=(x2max + x2min)/2
x03=(x3max + x3min)/2
```

```
#Average
xAveragemax = (x1max + x2max + x3max) / 3
xAveragemin = (x1min + x2min + x3min) / 3
ymax = int(200 + xAveragemax)
ymin = int(200 + xAveragemin)
```

```
# нормовані значення факторів
X11 = [-1, -1, -1, -1, +1, +1, +1, +1, -1.215, 1.215,
0, 0, 0, 0, 0]
X22 = [-1, -1, +1, +1, -1, -1, +1, +1, 0, 0, -1.215,
1.215, 0, 0, 0]
X33 = [-1, +1, -1, +1, -1, +1, -1, +1, 0, 0, 0, 0,
-1.215, 1.215, 0]
```

```
def totalkf2(x1, x2):
    xn = []
    for i in range(len(x1)):
        xn.append(x1[i]*x2[i])
    return xn

def totalkf3(x1, x2, x3):
    xn = []
    for i in range(len(x1)):
        xn.append(x1[i]*x2[i]*x3[i])
    return xn

def SK(x):
    xn = []
    for i in range(len(x)):
```

```

        xn.append(x[i]*x[i])
    return xn
X12 = totalkf2(X11, X22)
X13 = totalkf2(X11, X33)
X23 = totalkf2(X22, X33)
X123 = totalkf3(X11, X22, X33)
X8 = SK(X11)
X9 = SK(X22)
X10 = SK(X33)

X00=[1, 1, 1, 1, 1, 1, 1, 1]

Xnorm=[X00, X11, X22, X33, totalkf2(X11, X22),
totalkf2(X22, X33), totalkf2(X22, X33), totalkf3(X11,
X22, X33)]

for i in range(15):
    print("{:1} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6}
{:6} {:6}")

    .format(i+1,X11[i],X22[i],X33[i],X12[i],X13[i],X23[i],
X123[i],X8[i],X9[i],X10[i]))

print("matrix m=15")
print("{:3} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6}
{:6} {:6} {:6} {:6} {:6} {:6}")

    .format("N", "X0", "X1", "X2", "X3", "X4", "X5", "X6", "X7", "X
8", "X9", "X10", "Y1", "Y2", "Y3"))

X1 = [x1min, x1min, x1min, x1min, x1max, x1max, x1max,
x1max ]
X2 = [x2min, x2min, x2max, x2max, x2min, x2min, x2max,
x2max ]
X3 = [x3min, x3max, x3min, x3max, x3min, x3max, x3min,
x3max ]

Y1 = [random.randrange(138,247, 1) for i in range(8)]
Y2 = [random.randrange(138,247, 1) for i in range(8)]
Y3 = [random.randrange(138,247, 1) for i in range(8)]

X12=totalkf2(X1, X2)
X13=totalkf2(X1, X3)
X23=totalkf2(X2, X3)

```

```

X123=totalkf3(X1, X2, X3)
X0=[1]*8
Xall=[X0,X1,X2,X3,X12,X13,X23,X123]

for i in range(8):
    print("{:1} {:5} {:6} {:6} {:6} {:6} {:6} {:6} {:6}
{:6} {:6} {:6}
".format(i+1,X0[i],X1[i],X2[i],X3[i],X12[i],X13[i],X23
[i],X123[i],Y1[i],Y2[i],Y3[i]))

print("Середнє значення відгуку функції за рядками ")
def yav(i):
    return (Y1[i-1]+Y2[i-1]+Y3[i-1])/3
y1av1 = yav(1)
y2av2 = yav(2)
y3av3 = yav(3)
y4av4 = yav(4)
y5av5 = yav(5)
y6av6 = yav(6)
y7av7 = yav(7)
y8av8 = yav(8)

mx1 = sum(X1)/4
mx2 = sum(X2)/4
mx3 = sum(X3)/4

my = (y1av1 + y2av2 + y3av3 + y4av4)/4

a1 = (X1[0]*y1av1 + X1[1]*y2av2 + X1[2]*y3av3 +
X1[3]*y4av4)/4
a2 = (X2[0]*y1av1 + X2[1]*y2av2 + X2[2]*y3av3 +
X2[3]*y4av4)/4
a3 = (X3[0]*y1av1 + X3[1]*y2av2 + X3[2]*y3av3 +
X3[3]*y4av4)/4

a11 = (X1[0]*X1[0] + X1[1]*X1[1] + X1[2]*X1[2] +
X1[3]*X1[3])/4
a22 = (X2[0]*X2[0] + X2[1]*X2[1] + X2[2]*X2[2] +
X2[3]*X2[3])/4
a33 = (X3[0]*X3[0] + X3[1]*X3[1] + X3[2]*X3[2] +
X3[3]*X3[3])/4
a12 = a21 = (X1[0]*X2[0] + X1[1]*X2[1] + X1[2]*X2[2] +
X1[3]*X2[3])/4

```

```

a13 = a31 = (X1[0]*X3[0] + X1[1]*X3[1] + X1[2]*X3[2] +
X1[3]*X3[3])/4
a23 = a32 = (X2[0]*X3[0] + X2[1]*X3[1] + X2[2]*X3[2] +
X2[3]*X3[3])/4

```

```

b01 = np.array([[my, mx1, mx2, mx3], [a1, a11, a12,
a13], [a2, a12, a22, a32], [a3, a13, a23, a33]])
b02 = np.array([[1, mx1, mx2, mx3], [mx1, a11, a12,
a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]])
b0 = np.linalg.det(b01)/np.linalg.det(b02)

```

```

b11 = np.array([[1, my, mx2, mx3], [mx1, a1, a12,
a13], [mx2, a2, a22, a32], [mx3, a3, a23, a33]])
b12 = np.array([[1, mx1, mx2, mx3], [mx1, a11, a12,
a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]])
b1 = np.linalg.det(b11)/np.linalg.det(b12)

```

```

b21 = np.array([[1, mx1, my, mx3], [mx1, a11, a1,
a13], [mx2, a12, a2, a32], [mx3, a13, a3, a33]])
b22 = np.array([[1, mx1, mx2, mx3], [mx1, a11, a12,
a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]])
b2 = np.linalg.det(b21)/np.linalg.det(b22)

```

```

b31 = np.array([[1, mx1, mx2, my], [mx1, a11, a12,
a1], [mx2, a12, a22, a2], [mx3, a13, a23, a3]])
b32 = np.array([[1, mx1, mx2, mx3], [mx1, a11, a12,
a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]])
b3 = np.linalg.det(b31)/np.linalg.det(b32)

```

```

print("y1av1="+str(round(b0 + b1*X1[0] + b2*X2[0] +
b3*X3[0],2))+str(round(y1av1,2)))
print("y2av2="+str(round(b0 + b1*X1[1] + b2*X2[1] +
b3*X3[1],2))+str(round(y2av2,2)))
print("y3av3="+str(round(b0 + b1*X1[2] + b2*X2[2] +
b3*X3[2],2))+str(round(y3av3,2)))
print("y4av4="+str(round(b0 + b1*X1[3] + b2*X2[3] +
b3*X3[3],2))+str(round(y4av4,2)))
print("Значення співпадають")

```

#Дисперсія

```

d1 = ((Y1[0] - y1av1)**2 + (Y2[0] - y2av2)**2 + (Y3[0]
- y3av3)**2)/3

```

```

d2 = ((Y1[1] - y1av1)**2 + (Y2[1] - y2av2)**2 + (Y3[1]
- y3av3)**2)/3
d3 = ((Y1[2] - y1av1)**2 + (Y2[2] - y2av2)**2 + (Y3[2]
- y3av3)**2)/3
d4 = ((Y1[3] - y1av1)**2 + (Y2[3] - y2av2)**2 + (Y3[3]
- y3av3)**2)/3

dcup = [d1, d2, d3, d4]

m = 3
Gp = max(dcup) / sum(dcup)
f1 = m-1
f2 = N = 4
inscipy = taf(0.95, 1, f1 * 4)
Gt = inscipy / (inscipy + f1 - 2)

if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")
print("Критерій Стьюдента")
sb = sum(dcup) / N
sabs = sb / N * m
sbs = sabs ** 0.5

beta0 = (y1av1*1 + y2av2*1 + y3av3*1 + y4av4*1)/4
beta1 = (y1av1*(-1) + y2av2*(-1) + y3av3*1 +
y4av4*1)/4
beta2 = (y1av1*(-1) + y2av2*1 + y3av3*(-1) +
y4av4*1)/4
beta3 = (y1av1*(-1) + y2av2*1 + y3av3*1 +
y4av4*(-1))/4

t0 = abs(beta0)/sbs
t1 = abs(beta1)/sbs
t2 = abs(beta2)/sbs
t3 = abs(beta3)/sbs

f3 = f1*f2
tchart = tss(0.95, (m-1)*4)
if (t0<tchart):
    print("t0<tchart, b0 не значимий")
    b0=0

```

```

if (t1<tchart):
    print("t1<tchart, b1 не значимий")
    b1=0
if (t2<tchart):
    print("t2<tchart, b2 не значимий")
    b2=0
if (t3<tchart):
    print("t3<tchart, b3 не значимий")
    b3=0

yy1 = b0 + b1*x1min + b2*x2min + b3*x3min
yy2 = b0 + b1*x1min + b2*x2max + b3*x3max
yy3 = b0 + b1*x1max + b2*x2min + b3*x3max
yy4 = b0 + b1*x1max + b2*x2max + b3*x3min

print("Критерій Фішера")
d = 2

print(d," значимих коефіцієнтів")

f4 = N - d
sad = ((yy1 - y1av1)**2 + (yy2 - y2av2)**2 + (yy3 -
y3av3)**2 + (yy4 - y4av4)**2)*(m/(N-d))
Fp = sad/sb

print("d1=", round(d1,2), "d2=", round(d2,2), "d3=",
round(d3,2), "d4=", round(d4,2), "d5=", round(sb,2))
print("Fp=", round(Fp,2))

Ft = taf(0.95, d, (m - 1) * 4)

if Fp>Ft:
    print("Fp=", round(Fp,2), ">Ft", Ft, "Рівняння
неадекватно оригіналу")
    cont=1
else:
    print("Fp=", round(Fp,2), "<Ft", Ft, "Рівняння
адекватно оригіналу")
if cont==1:
    print("З взаємодією")
    Yall=[y1av1, y2av2, y3av3, y4av4, y5av5, y6av6,
y7av7, y8av8]

```



```
my = (y1av1 + y2av2 + y3av3 + y4av4 + y5av5 + y6av6  
+ y7av7 + y8av8)/8
```

```
def ni0(j):  
    s=0  
    for i in range(len(Xall[j])):  
        s=s+Xall[j][i]  
    return s/len(Xall[j])
```

```
def ni2(j, k):  
    s=0  
    for i in range(len(Xall[j])):  
        s=s+Xall[j][i]*Xall[k][i]  
    return s/len(Xall[j])
```

```
def ni3(j, k, l):  
    s=0  
    for i in range(len(Xall[j])):  
        s=s+Xall[j][i]*Xall[k][i]*Xall[l][i]  
    return s/len(Xall[j])
```

```
def ni4(j1, j2, j3, j4):  
    s=0  
    for i in range(len(Xall[j1])):
```

```
s=s+Xall[j1][i]*Xall[j2][i]*Xall[j3][i]*Xall[j4][i]  
    return s/len(Xall[j1])
```

```
def ni5(j1, j2, j3, j4, j5):  
    s=0  
    for i in range(len(Xall[j1])):
```

```
s=s+Xall[j1][i]*Xall[j2][i]*Xall[j3][i]*Xall[j4][i]*Xa  
ll[j5][i]  
    return s/len(Xall[j1])
```

```
def ni6(j1, j2, j3, j4, j5, j6):  
    s=0  
    for i in range(len(Xall[j1])):
```

```
s=s+Xall[j1][i]*Xall[j2][i]*Xall[j3][i]*Xall[j4][i]*Xa  
ll[j5][i]*Xall[j6][i]  
    return s/len(Xall[j1])
```

```
Yalls=sum(Yall)/8
```

```
def kmn(j):  
    s=0  
    for i in range(len(Yall)):
```

```

        s=s+yav(i)*Xall[j][i]
    return s/len(Yall)
def kmn2(j,k):
    s=0
    for i in range(len(Yall)):
        s=s+yav(i)*Xall[j][i]*Xall[k][i]
    return s/len(Yall)
def kmn3(j,k,l):
    s=0
    for i in range(len(Yall)):
        s=s+yav(i)*Xall[j][i]*Xall[k][i]*Xall[l][i]
    return s/len(Yall)

m=[ [0,0,0,0,0,0,0,0],
     [0,0,0,0,0,0,0,0],
     [0,0,0,0,0,0,0,0],
     [0,0,0,0,0,0,0,0],
     [0,0,0,0,0,0,0,0],
     [0,0,0,0,0,0,0,0],
     [0,0,0,0,0,0,0,0],
     [0,0,0,0,0,0,0,0]]

m[0][0]=1
m[1][0]=m[0][1]=ni0(1)
m[2][0]=m[0][2]=ni0(2)
m[3][0]=m[0][3]=ni0(3)
m[4][0]=m[0][4]=ni2(1, 2)
m[5][0]=m[0][5]=ni2(1, 3)
m[6][0]=m[0][6]=ni2(2, 3)

m[7][0]=m[0][7]=m[1][6]=m[2][5]=m[3][4]=m[4][3]=m[5][2]
]=m[6][1]=ni3(1, 2, 3)
    m[1][1]=ni2(1, 1)
    m[1][2]=m[2][1]=ni2(1, 2)
    m[1][3]=m[3][1]=ni2(1, 3)
    m[1][4]=m[4][1]=ni3(1, 1, 2)
    m[1][5]=m[5][1]=ni3(1, 1, 3)
    m[1][7]=m[7][1]=ni4(1, 1, 2, 3)
    m[2][2]=ni2(2, 2)
    m[2][3]=m[3][2]=ni2(2, 3)
    m[2][4]=m[4][2]=ni3(1, 2, 2)
    m[2][6]=m[6][2]=ni3(2, 2, 3)
    m[2][7]=m[7][2]=ni4(1, 2, 2, 3)
    m[3][3]=ni2(3, 3)

```

```

m[3][5]=m[5][3]=ni3(1, 3, 3)
m[3][6]=m[6][3]=ni3(2, 3, 3)
m[3][7]=m[7][3]=ni4(1, 2, 3, 3)
m[4][4]=ni4(1, 1, 2, 2)
m[4][5]=m[5][4]=ni4(1, 1, 2, 3)
m[4][6]=m[6][4]=ni4(1, 2, 2, 3)
m[4][7]=m[7][4]=ni5(1, 1, 2, 2, 3)
m[5][5]=ni4(1, 1, 3, 3)
m[5][6]=m[6][5]=ni4(1, 2, 3, 3)
m[5][7]=m[7][5]=ni5(1, 1, 2, 3, 3)
m[6][6]=ni4(2, 2, 3, 3)
m[6][7]=m[7][6]=ni5(1, 2, 2, 3, 3)
m[7][7]=ni6(1, 1, 2, 2, 3, 3)

```

```

k0 = Yalls
k1 = kmn(1)
k2 = kmn(2)
k3 = kmn(3)
k4 = kmn2(1, 2)
k5 = kmn2(1, 3)
k6 = kmn2(2, 3)
k7 = kmn3(1, 2, 3)
k = [k0, k1, k2, k3, k4, k5, k6, k7]

```

```

b00 = copy.copy(m)
b01 = copy.copy(m)
b02 = copy.copy(m)
b03 = copy.copy(m)
b04 = copy.copy(m)
b05 = copy.copy(m)
b06 = copy.copy(m)
b07 = copy.copy(m)
b00[0] = k
b01[1] = k
b02[2] = k
b03[3] = k
b04[4] = k
b05[5] = k
b06[6] = k
b07[7] = k

```

```

b0=np.linalg.det(b00)/np.linalg.det(m)

```

```

b1=np.linalg.det(b01)/np.linalg.det(m)
b2=np.linalg.det(b02)/np.linalg.det(m)
b3=np.linalg.det(b03)/np.linalg.det(m)
b4=np.linalg.det(b04)/np.linalg.det(m)
b5=np.linalg.det(b05)/np.linalg.det(m)
b6=np.linalg.det(b06)/np.linalg.det(m)
b7=np.linalg.det(b07)/np.linalg.det(m)

print("y1av1="+str(round(b0 +
b1*X1[0]+b2*X2[0]+b3*X3[0]+b4*X1[0]*X2[0]+b5*X1[0]*X3[
0]+b6*X2[0]*X3[0]+b7*X1[0]*X2[0]*X3[0],2))+)+"="+
str(round(y1av1,2)))
print("y2av2="+str(round(b0 +
b1*X1[1]+b2*X2[1]+b3*X3[1]+b4*X1[1]*X2[1]+b5*X1[1]*X3[
1]+b6*X2[1]*X3[1]+b7*X1[1]*X2[1]*X3[1],2))+)+"="+
str(round(y2av2,2)))
print("y3av3="+str(round(b0 +
b1*X1[2]+b2*X2[2]+b3*X3[2]+b4*X1[2]*X2[2]+b5*X1[2]*X3[
2]+b6*X2[2]*X3[2]+b7*X1[2]*X2[2]*X3[2],2))+)+"="+
str(round(y3av3,2)))
print("y4av4="+str(round(b0 +
b1*X1[3]+b2*X2[3]+b3*X3[3]+b4*X1[3]*X2[3]+b5*X1[3]*X3[
3]+b6*X2[3]*X3[3]+b7*X1[3]*X2[3]*X3[3],2))+)+"="+
str(round(y4av4,2)))
print("y5av5="+str(round(b0 +
b1*X1[4]+b2*X2[4]+b3*X3[4]+b4*X1[4]*X2[4]+b5*X1[4]*X3[
4]+b6*X2[4]*X3[4]+b7*X1[4]*X2[4]*X3[4],2))+)+"="+
str(round(y5av5,2)))
print("y6av6="+str(round(b0 +
b1*X1[5]+b2*X2[5]+b3*X3[5]+b4*X1[5]*X2[5]+b5*X1[5]*X3[
5]+b6*X2[5]*X3[5]+b7*X1[5]*X2[5]*X3[5],2))+)+"="+
str(round(y6av6,2)))
print("y7av7="+str(round(b0 +
b1*X1[6]+b2*X2[6]+b3*X3[6]+b4*X1[6]*X2[6]+b5*X1[6]*X3[
6]+b6*X2[6]*X3[6]+b7*X1[6]*X2[6]*X3[6],2))+)+"="+
str(round(y7av7,2)))
print("y8av8="+str(round(b0 +
b1*X1[7]+b2*X2[7]+b3*X3[7]+b4*X1[7]*X2[7]+b5*X1[7]*X3[
7]+b6*X2[7]*X3[7]+b7*X1[7]*X2[7]*X3[7],2))+)+"="+
str(round(y8av8,2)))

```

#Дисперсія

```

    d1 = ((Y1[0] - y1av1)**2 + (Y2[0] - y2av2)**2 +
(Y3[0] - y3av3)**2)/3
    d2 = ((Y1[1] - y1av1)**2 + (Y2[1] - y2av2)**2 +
(Y3[1] - y3av3)**2)/3
    d3 = ((Y1[2] - y1av1)**2 + (Y2[2] - y2av2)**2 +
(Y3[2] - y3av3)**2)/3
    d4 = ((Y1[3] - y1av1)**2 + (Y2[3] - y2av2)**2 +
(Y3[3] - y3av3)**2)/3
    d5 = ((Y1[4] - y1av1)**2 + (Y2[4] - y2av2)**2 +
(Y3[4] - y3av3)**2)/3
    d6 = ((Y1[5] - y1av1)**2 + (Y2[5] - y2av2)**2 +
(Y3[5] - y3av3)**2)/3
    d7 = ((Y1[6] - y1av1)**2 + (Y2[6] - y2av2)**2 +
(Y3[6] - y3av3)**2)/3
    d8 = ((Y1[7] - y1av1)**2 + (Y2[7] - y2av2)**2 +
(Y3[7] - y3av3)**2)/3

    print("d1=", round(d1,2), "d2=", round(d2,2), "d3=",
round(d3,2), "d4=", round(d4,2), "d5=",
round(d5,2), "d6=", round(d6,2), "d7=",
round(d7,2), "d8=", round(d8,2))

    dcup = [d1, d2, d3, d4, d5, d6, d7, d8]

    m = 3
    Gp = max(dcup) / sum(dcup)
    f1 = m-1
    f2 = N = 8
    inscipy = taf(0.95, 1, f1 * 4)
    Gt = inscipy / (inscipy + f1 - 2)
    if Gp < Gt:
        print("Дисперсія однорідна")
    else:
        print("Дисперсія неоднорідна")

    print("Критерій Стьюдента")
    sb = sum(dcup) / N
    sabs = sb / N * m
    sbs = sabs ** 0.5

    beta0 =
(y1av1*1+y2av2*1+y3av3*1+y4av4*1+y5av5*1+y6av6*1+y7av7
*1+y8av8*1)/8

```

```

    beta1 =
(y1av1*(-1)+y2av2*(-1)+y3av3*(-1)+y4av4*(-1)+y5av5*1+y
6av6*1+y7av7*1+y8av8*1)/8
    beta2 =
(y1av1*(-1)+y2av2*(-1)+y3av3*1+y4av4*1+y5av5*(-1)+y6av
6*(-1)+y7av7*1+y8av8*1)/8
    beta3 =
(y1av1*(-1)+y2av2*1+y3av3*(-1)+y4av4*1+y5av5*(-1)+y6av
6*1+y7av7*(-1)+y8av8*1)/8
    beta4 =
(y1av1*1+y2av2*1+y3av3*(-1)+y4av4*(-1)+y5av5*(-1)+y6av
6*(-1)+y7av7*1+y8av8*1)/8
    beta5 =
(y1av1*1+y2av2*(-1)+y3av3*1+y4av4*(-1)+y5av5*(-1)+y6av
6*1+y7av7*(-1)+y8av8*1)/8
    beta6 =
(y1av1*1+y2av2*(-1)+y3av3*(-1)+y4av4*1+y5av5*1+y6av6*(
-1)+y7av7*(-1)+y8av8*1)/8
    beta7 =
(y1av1*(-1)+y2av2*1+y3av3*1+y4av4*(-1)+y5av5*1+y6av6*(
-1)+y7av7*(-1)+y8av8*1)/8

```

```

t0 = abs(beta0)/sbs
t1 = abs(beta1)/sbs
t2 = abs(beta2)/sbs
t3 = abs(beta3)/sbs
t4 = abs(beta4)/sbs
t5 = abs(beta5)/sbs
t6 = abs(beta6)/sbs
t7 = abs(beta7)/sbs

```

```

f3 = f1*f2
tchart = tss(0.95, (m - 1) * 4)

```

```

d=8
if (t0<tchart):
    print("t0<tchart, b0 не значимий")
    b0=0
    d=d-1
if (t1<tchart):
    print("t1<tchart, b1 не значимий")
    b1=0

```

```

        d=d-1
    if (t2<tchart):
        print("t2<tchart, b2 не значимый")
        b2=0
        d=d-1
    if (t3<tchart):
        print("t3<tchart, b3 не значимый")
        b3=0
        d=d-1
    if (t4<tchart):
        print("t4<tchart, b4 не значимый")
        b4=0
        d=d-1
    if (t5<tchart):
        print("t5<tchart, b5 не значимый")
        b5=0
        d=d-1
    if (t6<tchart):
        print("t6<tchart, b6 не значимый")
        b6=0
        d=d-1
    if (t7<tchart):
        print("t7<tchart, b7 не значимый")
        b7=0
        d=d-1
    yy1 =
b0+b1*x1min+b2*x2min+b3*x3min+b4*x1min*x2min+b5*x1min*
x3min+b6*x2min*x3min+b7*x1min*x2min*x3min
    yy2 =
b0+b1*x1min+b2*x2min+b3*x3max+b4*x1min*x2min+b5*x1min*
x3max+b6*x2min*x3max+b7*x1min*x2min*x3max
    yy3 =
b0+b1*x1min+b2*x2max+b3*x3min+b4*x1min*x2max+b5*x1min*
x3min+b6*x2max*x3min+b7*x1min*x2max*x3min
    yy4 =
b0+b1*x1min+b2*x2max+b3*x3max+b4*x1min*x2max+b5*x1min*
x3max+b6*x2max*x3max+b7*x1min*x2max*x3max
    yy5 =
b0+b1*x1max+b2*x2min+b3*x3min+b4*x1max*x2min+b5*x1max*
x3min+b6*x2min*x3min+b7*x1max*x2min*x3min
    yy6 =
b0+b1*x1max+b2*x2min+b3*x3max+b4*x1max*x2min+b5*x1max*
x3max+b6*x2min*x3max+b7*x1max*x2min*x3max

```

```

yy7 =
b0+b1*x1max+b2*x2max+b3*x3min+b4*x1max*x2max+b5*x1max*
x3min+b6*x2max*x3min+b7*x1max*x2min*x3max
yy8 =
b0+b1*x1max+b2*x2max+b3*x3max+b4*x1max*x2max+b5*x1max*
x3max+b6*x2max*x3max+b7*x1max*x2max*x3max

print("Критерій Фішера")
print(d," значимих коефіцієнтів")

f4 = N - d

sad =
((yy1-y1av1)**2+(yy2-y2av2)**2+(yy3-y3av3)**2+(yy4-y4a
v4)**2+(yy5-y5av5)**2+(yy6-y6av6)**2+(yy7-y7av7)**2+(y
y8-y8av8)**2)*(m/(N-d))
Fp = sad/sb

Ft = taf(0.95, d, (m - 1) * 4)
if Fp > Ft:
    print("Fp=", round(Fp, 2), ">Ft", Ft, "Рівняння
неадекватно оригіналу")
else:
    print("Fp=", round(Fp, 2), "<Ft", Ft, "Рівняння
адекватно оригіналу")

```

**Висновок:**Провів повний трьохфакторний експеримент. Знайшов рівняння регресії адекватне об'єкту.