Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

з дисципліни МОПЕ

на тему:

# «Проведення трьохфакторного експерименту

# при використанні рівняння регресії з урахуванням ефекту взаємодії.»

Виконав:

студент групи ІВ-83

Кочерук Давид

Варіант: 314

Перевірив:

Регіда П.Г.

Київ 2020



Варіант:

X1min = -15 X2min = -35 X3min = -25

X1max = 30 X2max = 15 X3max = 5

**Код програми:**

import random  
import numpy as np  
import copy  
x1min = -15  
x1max = 30  
x2min = -35  
x2max = 15  
x3min = -25  
x3max = 5  
x01=(x1max + x1min)/2  
x02=(x2max + x2min)/2  
x03=(x3max + x3min)/2  
xAvmax = (x1max + x2max + x3max)/3  
xAvmin = (x1min + x2min + x3min)/3  
ymax = int(200+xAvmax)  
ymin = int(200+xAvmin)  
#print(round(xAvmin,2))  
#print(round(xAvmax,2))  
  
#print(round(ymax,2))  
#print(round(ymin,2))  
  
print("y=b0+b1\*x1+b2\*x2+b3\*x3+b12\*x1\*x2+b13\*x1\*x3+b23\*x2\*x3+b123\*x1\*x2\*x3+b11\*x1^2+b22\*x2^2+b33\*x3^2")  
print("Кодованє значення X")  
print("{:5} {:5} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6}".format("№","X1","X2","X3","X4","X5","X6","X7","X8","X9","X10"))  
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 sumkf2(x1,x2):  
 xn = []  
 for i in range(len(x1)):  
 xn.append(x1[i]\*x2[i])  
 return xn  
def sumkf3(x1,x2,x3):  
 xn = []  
 for i in range(len(x1)):  
 xn.append(x1[i]\*x2[i]\*x3[i])  
 return xn  
def kv(x):  
 xn = []  
 for i in range(len(x)):  
 xn.append(x[i]\*x[i])  
 return xn  
X12 = sumkf2(X11,X22)  
X13 = sumkf2(X11,X33)  
X23 = sumkf2(X22,X33)  
X123 = sumkf3(X11,X22,X33)  
X8 = kv(X11)  
X9 = kv(X22)  
X10 = kv(X33)  
  
X00=[1, 1, 1, 1, 1, 1, 1, 1]  
Xnorm=[X00,X11,X22,X33,sumkf2(X11,X22),sumkf2(X22,X33),sumkf2(X22,X33),sumkf3(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("Матриця для m=15")  
print("{:3} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6} {:6}".format("№","X0","X1","X2","X3","X4","X5","X6","X7","X8","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=sumkf2(X1,X2)  
X13=sumkf2(X1,X3)  
X23=sumkf2(X2,X3)  
X123=sumkf3(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")  
  
print("Дисперсія по рядкам")  
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  
print("d1=", round(d1,2),"d2=", round(d2,2),"d3=", round(d3,2),"d4=", round(d4,2))  
  
dcouple = [d1, d2, d3, d4]  
  
m = 3  
Gp = max(dcouple)/sum(dcouple)  
f1 = m-1  
f2 = N = 4  
Gt = 0.7679  
if Gp < Gt:  
 print("Дисперсія однорідна")  
else:  
 print("Дисперсія неоднорідна")  
print("Критерій Стьюдента")  
sb = sum(dcouple)/N  
ssbs = sb/N\*m  
sbs = ssbs\*\*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  
  
#print(t0,t1,t2,t3)  
  
f3 = f1\*f2  
ttabl = 2.306  
print("f3 = f1\*f2, з таблиці tтабл = 2.306")  
#print(t0,t1,t2,t3)  
if (t0<ttabl):  
 print("t0<ttabl, b0 не значимий")  
 b0=0  
if (t1<ttabl):  
 print("t1<ttabl, b1 не значимий")  
 b1=0  
if (t2<ttabl):  
 print("t2<ttabl, b2 не значимий")  
 b2=0  
if (t3<ttabl):  
 print("t3<ttabl, 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  
#print(f4)  
#print(f3)  
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))  
print('Ft берем із таблиці 8 рядяк 2 стовпець Ft = 4.5')  
Ft=4.5  
cont = 0  
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  
 #print(Xall)  
 def mni(j):  
 s=0  
 for i in range(len(Xall[j])):  
 s=s+Xall[j][i]  
 return s/len(Xall[j])  
 #print(mni(0))  
 def mni2(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 mni3(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 mni4(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 mni5(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]\*Xall[j5][i]  
 return s/len(Xall[j1])  
 def mni6(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]\*Xall[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]=mni(1)  
 m[2][0]=m[0][2]=mni(2)  
 m[3][0]=m[0][3]=mni(3)  
 m[4][0]=m[0][4]=mni2(1,2)  
 m[5][0]=m[0][5]=mni2(1,3)  
 m[6][0]=m[0][6]=mni2(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]=mni3(1,2,3)  
 m[1][1]=mni2(1,1)  
 m[1][2]=m[2][1]=mni2(1,2)  
 m[1][3]=m[3][1]=mni2(1,3)  
 m[1][4]=m[4][1]=mni3(1,1,2)  
 m[1][5]=m[5][1]=mni3(1,1,3)  
 m[1][7]=m[7][1]=mni4(1,1,2,3)  
 m[2][2]=mni2(2,2)  
 m[2][3]=m[3][2]=mni2(2,3)  
 m[2][4]=m[4][2]=mni3(1,2,2)  
 m[2][6]=m[6][2]=mni3(2,2,3)  
 m[2][7]=m[7][2]=mni4(1,2,2,3)  
 m[3][3]=mni2(3,3)  
 m[3][5]=m[5][3]=mni3(1,3,3)  
 m[3][6]=m[6][3]=mni3(2,3,3)  
 m[3][7]=m[7][3]=mni4(1,2,3,3)  
 m[4][4]=mni4(1,1,2,2)  
 m[4][5]=m[5][4]=mni4(1,1,2,3)  
 m[4][6]=m[6][4]=mni4(1,2,2,3)  
 m[4][7]=m[7][4]=mni5(1,1,2,2,3)  
 m[5][5]=mni4(1,1,3,3)  
 m[5][6]=m[6][5]=mni4(1,2,3,3)  
 m[5][7]=m[7][5]=mni5(1,1,2,3,3)  
 m[6][6]=mni4(2,2,3,3)  
 m[6][7]=m[7][6]=mni5(1,2,2,3,3)  
 m[7][7]=mni6(1,1,2,2,3,3)  
 #print(m)  
  
 #print(np.linalg.det(m))  
 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)))  
  
 print("Значення приблизно співпадають")  
  
 print("Дисперсія по рядкам")  
 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))  
  
 dcouple = [d1, d2, d3, d4, d5, d6, d7, d8]  
  
 m = 3  
 Gp = max(dcouple)/sum(dcouple)  
 f1 = m-1  
 f2 = N = 8  
 Gt = 0.5157  
 if Gp < Gt:  
 print("Дисперсія однорідна")  
 else:  
 print("Дисперсія неоднорідна")  
  
 print("Критерій Стьюдента")  
 sb = sum(dcouple)/N  
 ssbs = sb/N\*m  
 sbs = ssbs\*\*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+y6av6\*1+y7av7\*1+y8av8\*1)/8  
 beta2 = (y1av1\*(-1)+y2av2\*(-1)+y3av3\*1+y4av4\*1+y5av5\*(-1)+y6av6\*(-1)+y7av7\*1+y8av8\*1)/8  
 beta3 = (y1av1\*(-1)+y2av2\*1+y3av3\*(-1)+y4av4\*1+y5av5\*(-1)+y6av6\*1+y7av7\*(-1)+y8av8\*1)/8  
 beta4 = (y1av1\*1+y2av2\*1+y3av3\*(-1)+y4av4\*(-1)+y5av5\*(-1)+y6av6\*(-1)+y7av7\*1+y8av8\*1)/8  
 beta5 = (y1av1\*1+y2av2\*(-1)+y3av3\*1+y4av4\*(-1)+y5av5\*(-1)+y6av6\*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  
  
 #print(t0,t1,t2,t3)  
  
 f3 = f1\*f2  
 ttabl = 2.12  
 print("f3 = f1\*f2, з таблиці tтабл = 2.306")  
 #print(t0,t1,t2,t3)  
 d=8  
 if (t0<ttabl):  
 print("t0<ttabl, b0 не значимий")  
 b0=0  
 d=d-1  
 if (t1<ttabl):  
 print("t1<ttabl, b1 не значимий")  
 b1=0  
 d=d-1  
 if (t2<ttabl):  
 print("t2<ttabl, b2 не значимий")  
 b2=0  
 d=d-1  
 if (t3<ttabl):  
 print("t3<ttabl, b3 не значимий")  
 b3=0  
 d=d-1  
 if (t4<ttabl):  
 print("t4<ttabl, b4 не значимий")  
 b4=0  
 d=d-1  
 if (t5<ttabl):  
 print("t5<ttabl, b5 не значимий")  
 b5=0  
 d=d-1  
 if (t6<ttabl):  
 print("t6<ttabl, b6 не значимий")  
 b6=0  
 d=d-1  
 if (t7<ttabl):  
 print("t7<ttabl, b7 не значимий")  
 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  
 #print(f4)  
 #print(f3)  
 sad = ((yy1-y1av1)\*\*2+(yy2-y2av2)\*\*2+(yy3-y3av3)\*\*2+(yy4-y4av4)\*\*2+(yy5-y5av5)\*\*2+(yy6-y6av6)\*\*2+(yy7-y7av7)\*\*2+(yy8-y8av8)\*\*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))  
 F=[4.5,3.6,3.2,3.0,2.9,2.7,2.4,2.2]  
 Fi=[1,2,3,4,5,6,12,24]  
 dif=100  
 counter=0  
 for i in range(len(F)):  
 if abs(Fi[i]-(i+1))<dif:  
 dif= abs(Fi[i]-i)  
 counter=i  
 Ft=F[counter]  
 print("Ft берем із таблиці 16 рядяк ",f4," стовпець Ft = ",Ft)  
 cont=0  
 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")