Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Методи оптимізації та планування експерименту**

**Лабораторна робота №3**

«Проведення трьохфакторного експерименту з використанням лінійного рівняння регресії»

Виконав:

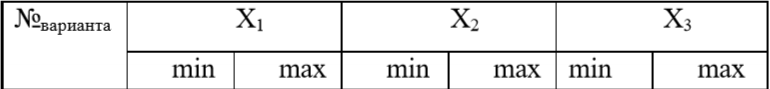
студент групи ІВ-82

Кузьмич А.А.

Залікова книжка №8215

Перевірив Регіда П. Г.

Київ - 2020 р.





**import** random  
**import** numpy **as** np  
**from** prettytable **import** PrettyTable  
**from** scipy.stats **import** f ,t  
  
  
x1min = -15  
x1max = 30  
x2min = 25  
x2max = 65  
x3min = -15  
x3max = -5  
xAvmax =(x1max+x2max+x3max)/3  
xAvmin = (x1min+x2min+x3min)/3  
ymax = int(200+xAvmax)  
ymin = int(200+xAvmin)  
  
m = 3  
  
print(**"Кодоване значення X"**)  
table1 = PrettyTable()  
X11 = (**"-1"**, **"-1"**, **"+1"**, **"+1"**)  
X22 = [**"-1"**, **"+1"**, **"-1"**, **"+1"**]  
X33 = [**"-1"**, **"+1"**, **"+1"**, **"-1"**]  
table1.add\_column( **"№"**, (1,2,3,4))  
table1.add\_column( **"X1"**, X11)  
table1.add\_column( **"X2"**, X22)  
table1.add\_column( **"X3"**, X33)  
print(table1)  
  
print(**"Матриця для m="** , m)  
X1 = [x1min, x1min, x1max, x1max]  
X2 = [x2min, x2max, x2min, x2max]  
X3 = [x3min, x3max, x3max, x3min]  
**for** i **in** range(1 , m+1):  
 globals()[**'Y%s'** % i] = [random.randrange(ymin,ymax, 1) **for** k **in** range(4)]  
  
table2 = PrettyTable()  
table2.add\_column(**"№"** , (1,2,3,4))  
table2.add\_column(**"X1"** , X1)  
table2.add\_column(**"X2"** , X2)  
table2.add\_column(**"X3"** , X3)  
**for** i **in** range(1 , m+1):  
 table2.add\_column(**"Y"**+str(i) , globals()[**'Y%s'** % i])  
  
  
print(table2)  
print(**"Середнє значення відгуку функції за рядками "**)  
  
s1 , s2 ,s3 ,s4=0 ,0,0,0  
  
**for** i **in** range(1 ,m+1):  
 s1 += globals()[**'Y%s'** % i][0]  
 s2 += globals()[**'Y%s'** % i][1]  
 s3 += globals()[**'Y%s'** % i][2]  
 s4 += globals()[**'Y%s'** % i][3]  
y1av1 = s1/m  
y2av2 = s2/m  
y3av3 = s3/m  
y4av4 = s4/m  
  
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)))  
**if** round(b0 + b1\*X1[0] + b2\*X2[0] + b3\*X3[0],2) == round(y1av1,2) **and** round(b0 + b1\*X1[1] + b2\*X2[1] + b3\*X3[1],2) == round(y2av2,2) **and** round(b0 + b1\*X1[2] + b2\*X2[2] + b3\*X3[2],2) == round(y3av3,2) **and** round(b0 + b1\*X1[3] + b2\*X2[3] + b3\*X3[3],2) == round(y4av4,2):  
 print(**"Значення співпадають"**)  
**else**:  
 print(**"Значення не співпадають"**)  
  
print(**"Дисперсія по рядкам"**)  
sd1 ,sd2 ,sd3 ,sd4 = 0,0,0,0  
**for** i **in** range(1, m+1):  
 sd1 += ((globals()[**'Y%s'** % i][0])-y1av1)\*\*2  
 sd2 += ((globals()[**'Y%s'** % i][1]) - y2av2) \*\* 2  
 sd3 += ((globals()[**'Y%s'** % i][2]) - y3av3) \*\* 2  
 sd4 += ((globals()[**'Y%s'** % i][3]) - y4av4) \*\* 2  
  
d1 = sd1/m  
d2 = sd2/m  
d3 = sd3/m  
d4 = sd4/m  
print(**"d1="**, round(d1,2),**"d2="**, round(d2,2),**"d3="**, round(d3,2),**"d4="**, round(d4,2))  
  
dcouple = [d1, d2, d3, d4]  
  
  
  
Gp = max(dcouple)/sum(dcouple)  
  
q = 0.05  
f1 = m-1  
f2 = N = 4  
  
fisher = f.isf(\*[q/f2, f1, (f2-1)\*f1])  
Gt = round(fisher/(fisher+(f2-1)),4)  
  
**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  
  
  
  
f3 = f1\*f2  
ttabl = round(abs(t.ppf(q / 2, f3)),4)  
  
**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  
Ft = round(abs(f.isf(q,f4,f3)),4)  
  
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))  
print(**"Fp="**, round(Fp,2))  
  
**if** Fp>Ft:  
 print(**"Fp="**,round(Fp,2),**">Ft"**,Ft,**"Рівняння неадекватно оригіналу"**)  
**else**:  
 print(**"Fp="**,round(Fp,2),**"<Ft"**,Ft,**"Рівняння адекватно оригіналу"**)

Кодоване значення X

+---+----+----+----+

| № | X1 | X2 | X3 |

+---+----+----+----+

| 1 | -1 | -1 | -1 |

| 2 | -1 | +1 | +1 |

| 3 | +1 | -1 | +1 |

| 4 | +1 | +1 | -1 |

+---+----+----+----+

Матриця для m= 3

+---+-----+----+-----+-----+-----+-----+

| № | X1 | X2 | X3 | Y1 | Y2 | Y3 |

+---+-----+----+-----+-----+-----+-----+

| 1 | -15 | 25 | -15 | 218 | 225 | 198 |

| 2 | -15 | 65 | -5 | 200 | 204 | 224 |

| 3 | 30 | 25 | -5 | 213 | 202 | 200 |

| 4 | 30 | 65 | -15 | 223 | 220 | 206 |

+---+-----+----+-----+-----+-----+-----+

Середнє значення відгуку функції за рядками

y1av1=213.67=213.67

y2av2=209.33=209.33

y3av3=205.0=205.0

y4av4=216.33=216.33

Значення співпадають

Дисперсія по рядкам

d1= 130.89 d2= 110.22 d3= 32.67 d4= 54.89

Дисперсія однорідна

Критерій Стьюдента

t1<ttabl, b1 не значимий

t2<ttabl, b2 не значимий

t3<ttabl, b3 не значимий

Критерій Фішера

2 значимих коефіцієнтів

d1= 130.89 d2= 110.22 d3= 32.67 d4= 54.89

Fp= 11.24

Fp= 11.24 >Ft 4.459 Рівняння неадекватно оригіналу