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 №4

**«**Проведення трьохфакторного експерименту

при використанні рівняння регресії з урахуванням ефекту взаємодії.

**»**

Виконав:

студент групи ІВ-83

Головенець Р.С.

Перевірив:

ас. Регіда П.Г.

Київ-2020 р.

**Лістинг програми**

**import** random, math  
**import** scipy.stats  
x1min,x2min,x3min,x1max,x2max,x3max,N=-30,-25,-30,20,10,-15,8  
neadekvat=1  
Average\_max=(x1max+x2max+x3max)/3  
Average\_min=(x1min+x2min+x3min)/3  
ymin=round(200+Average\_min)  
ymax=round(200+Average\_max)  
X = [[-1.0, -1.0, -1.0],  
 [-1.0, -1.0, 1.0],  
 [-1.0, 1.0, -1.0],  
 [-1.0, 1.0, 1.0],  
 [1.0, -1.0, -1.0],  
 [1.0, -1.0, 1.0],  
 [1.0, 1.0, -1.0],  
 [1.0, 1.0, 1.0]]  
MatrixX = [ [ x1min , x2min , x3min ] ,  
 [ x1min , x2min , x3max ] ,  
 [ x1min , x2max , x3min ] ,  
 [ x1min , x2max , x3max ] ,  
 [ x1max , x2min , x3min ] ,  
 [ x1max , x2min , x3max ] ,  
 [ x1max , x2max , x3min ] ,  
 [ x1max , x2max , x3max ] ]  
print(**"Матриця X: "**)  
**for** i **in** range(len(MatrixX)):  
 print(MatrixX[i])  
**while True**:  
 **if** neadekvat==1:  
 m=3  
 print(**"Рівняння регресії: \n y=b0+b1\*x1+b2\*x2+b3\*x3+b12\*x1\*x2+b13\*x1\*x3+b23\*x2\*x3+b123\*x1\*x2\*x3"**)  
 neadekvat=0  
 MatrixY, Average, Dispersion, Beta, t = [], [], [], [], []  
 print(**'Матриця Y с діапазоном('**, ymin, **','**, ymax, **')'**)  
 **for** i **in** range(0, 8):  
 MatrixY.append([random.randint(ymin, ymax) **for** j **in** range(0, m)])  
 Average.append(sum(MatrixY[i]) / len(MatrixY[i]))  
 Dispersion.append(sum((k - Average[i]) \*\* 2 **for** k **in** MatrixY[i]) / len(MatrixY[i]))  
 print(MatrixY[i])  
 print(**"Середні значення Y: \n"**, round(Average[0], 3), round(Average[1], 3), round(Average[2], 3),  
 round(Average[3], 3),round(Average[4], 3),round(Average[5], 3),round(Average[6], 3),round(Average[7], 3))  
 b0 = sum([Average[i] **for** i **in** range(len(MatrixX))]) / len(MatrixX)  
 b1 = sum([X[i][0] \* Average[i] **for** i **in** range(len(MatrixX))]) / len(MatrixX)  
 b2 = sum([X[i][1] \* Average[i] **for** i **in** range(len(MatrixX))]) / len(MatrixX)  
 b3 = sum([X[i][2] \* Average[i] **for** i **in** range(len(MatrixX))]) / len(MatrixX)  
 b12 = sum([X[i][0] \* X[i][1] \* Average[i] **for** i **in** range(len(MatrixX))]) / len(MatrixX)  
 b13 = sum([X[i][0] \* X[i][2] \* Average[i] **for** i **in** range(len(MatrixX))]) / len(MatrixX)  
 b23 = sum([X[i][1] \* X[i][2] \* Average[i]**for** i **in** range(len(MatrixX))]) / len(MatrixX)  
 b123 = sum([X[i][0] \* X[i][1] \* X[i][2] \* Average[i] **for** i **in** range(len(MatrixX))]) / len(MatrixX)  
 print(**'Отримане рівняння регресії: \n'**, round(b0, 3), **' + '**, round(b1, 3), **' \* x1 +'**, round(b2, 3),  
 **' \* x2 +'**, round(b3, 3), **' \* x3 +'**, round(b12,3),**' \* x1\*x2 +'**, round(b13,3),**' \* x1\*x3 +'**,round(b23,3)  
 ,**' \* x2\*x3 +'**, round(b123,3),**' \* x1\*x2\*x3'**)  
 **for** i **in** range(len(MatrixX)):  
 print(**"y"**,i+1,**" = b0+b1\*x1+b2\*x2+b3\*b12\*x1\*x2+b13\*x1\*x3+b23\*x2\*x3+b123\*x1\*x2\*x3 = "**,  
 round(b0 + b1 \* X[i][0] + b2 \* X[i][1] + b3 \* X[i][2] + b12\*  
 X[i][0]\*X[i][1]+b13\*X[i][0]\*X[i][2]+b23\*  
 X[i][1]\*X[i][2]+b123\*X[i][0]\*X[i][1]\*X[i][2]  
 , 3))  
 print(**"Результат збігається з середніми значеннями"**)  
 print(**"Перевіримо критерії Кохрена"**)  
 Gp = max(Dispersion) / sum(Dispersion)  
 f1 = m - 1  
 f2 = N  
 q = 0.05  
 tableGt = {2: 7679, 3: 0.6841, 4: 0.6287, 5: 0.5892, 6: 0.5598, 7: 0.5365, 8: 0.5175, 9: 0.5017, 10: 0.4884}  
 tableGt2 = [(range(11, 17), 0.4366), (range(17, 37), 0.3720), (range(37, 145), 0.3093)]  
 **if** m<11:  
 Gt= tableGt.get(m)  
 **else**:  
 **for** i **in** range(len(tableGt2)):  
 **if** m **in** tableGt2[i][0]:  
 Gt = tableGt2[i][1]  
 **break** print(**'Gp='**, round(Gp, 3), **", Gt="**, Gt)  
 **if** Gp < Gt:  
 print(Gp, **"<="**, Gt)  
 print(**"Дисперсія однорідна"**)  
 **else**:  
 print(Gp, **">="**, Gt)  
 print(**"Дисперсія не однорідна"**)  
 m += 1  
 **continue** print(**"перевіримо критерій Стьюдента"**)  
 S2betaSum = sum(Dispersion) / N  
 S2beta = S2betaSum / (N \* m)  
 Sbeta = math.sqrt(S2beta)  
 MatrixCodeX = [[1.0, -1.0, -1.0, -1.0, 1.0, 1.0, 1.0, -1.0],  
 [1.0, -1.0, -1.0, 1.0, 1.0, -1.0, -1.0, 1.0],  
 [1.0, -1.0, 1.0, -1.0, -1.0, 1.0, -1.0, 1.0],  
 [1.0, -1.0, 1.0, 1.0, -1.0, -1.0, 1.0, -1.0],  
 [1.0, 1.0, -1.0, -1.0, -1.0, -1.0, 1.0, 1.0],  
 [1.0, 1.0, -1.0, 1.0, -1.0, 1.0, -1.0, -1.0],  
 [1.0, 1.0, 1.0, -1.0, 1.0, -1.0, -1.0, -1.0],  
 [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]]  
 **for** i **in** range(N):  
 Beta.append(round(sum([MatrixCodeX[j][i] \* Average[j] **for** j **in** range(len(MatrixCodeX))])/N,3))  
 t.append(round(abs(Beta[i]/Sbeta),3))  
 print(**"t: "**, t)  
 f3 = f1 \* f2  
 print(**"f3="**, f3)  
 tableS = round(scipy.stats.t.ppf((1 + (1 - q)) / 2, f3),3)  
 print(**"Табличне значення ="**,tableS)  
 b = [b0, b1, b2, b3, b12, b13, b23, b123]  
 **for** i **in** range(N):  
 **if** t[i] < tableS:  
 b[i] = 0  
 print(t[i], **"<"**, tableS)  
 y = []  
 print(**"y="**,round(b[0],3),**"+"**,round(b[1],3),**"\*x1+"**,round(b[2],3),**"\*x2+"**,round(b[3],3),**"\*x3+"**,round(b[4],3)  
 ,**"\*x1\*x2+"**,round(b[5],3),**"\*x1\*x3"**,round(b[6],3),**"\*x2\*x3"**,round(b[7],3),**"\*x1\*x2\*x3"**)  
 **for** i **in** range(N):  
 y.append(b[0] + b[1] \* X[i][0] + b[2] \* X[i][1] + b[3] \* X[i][2] + b[4] \* X[i][0]\* X[i][1] +  
 b[5] \* X[i][0]\* X[i][2] + b[6] \* X[i][1]\* X[i][2] + b[7] \* X[i][0]\* X[i][1]\* X[i][2])  
 y[i]=round((y[i]),3)  
 print(**"y: "**, y)  
 **for** i **in** range(N):  
 print(i+1,**"."**,y[i], **"=="**, round((Average[i]),3))  
 print(**"Нуль гіпотеза виконується"**)  
 print(**"Перевіримо критерій Фішера"**)  
 d = 0  
 **for** i **in** range(len(b)):  
 **if** b[i] != 0:  
 d += 1  
 print(**"d="**, d)  
 f4 = N - d  
 print(**"f4="**, f4)  
 Sum = 0  
 **for** i **in** range(len(y)):  
 Sum += pow((y[i] - Average[i]), 2)  
 Sad = (m / (N - d)) \* Sum  
 Fp = Sad / S2betaSum  
 print(**"Fp="**, round(Fp, 3))  
 Ft = round(scipy.stats.f.ppf(1 - q, f4, f3), 3)  
 print(**"Ft="**, Ft)  
 **if** Fp > Ft:  
 print(**"Рівняння регресії неадекватно оригіналу при рівні значимості 0.05"**)  
 neadekvat=1  
 **continue  
 else**:  
 print(**"Рівняння регресії адекватно оригіналу при рівні значимості 0.05"**)  
 **break**

**Результат виконання роботи:**



