Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 3

з дисципліни «Методи наукових досліджень»

на тему «ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАВ:

студент 2 курсу

групи ІВ-91

Онищук Ю. І.

Залікова – 9122

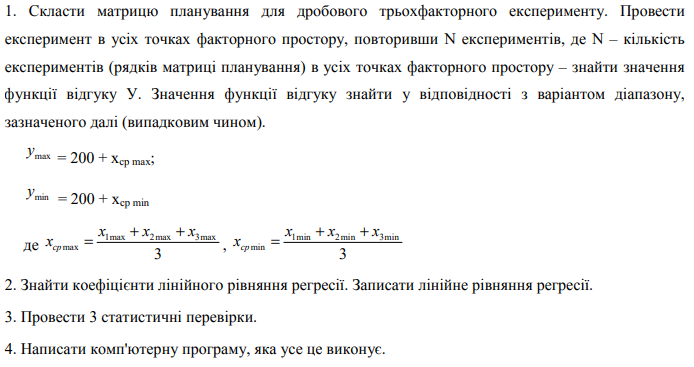
ПЕРЕВІРИВ:

ас. Регіда П. Г.

Київ – 2021

**Мета**: провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

**Завдання:**



Варіанти обираються по номеру в списку в журналі викладача.





**Програмний код**

**import** random **as** rn  
**from** math **import** sqrt  
**from** numpy **import** linalg **as** lg  
**import** pprint  
  
**def** exp\_raw(x1\_num, x2\_num, x3\_num, y, m = 3):  
 y\_gen = [rn.randint(y[0], y[1]) **for** i **in** range(m)]  
 y\_mid = sum(y\_gen)/m  
 sigma = 0  
 **for** i **in** range(len(y\_gen)):  
 sigma += ((y\_gen[i] - y\_mid)\*\*2)/m  
 **return** [x1\_num, x2\_num, x3\_num, y\_gen, y\_mid, sigma]  
  
**def** experiment(x1, x2, x3, y):  
 exp1 = exp\_raw(x1[0], x2[0], x3[0], y)  
 exp2 = exp\_raw(x1[0], x2[1], x3[1], y)  
 exp3 = exp\_raw(x1[1], x2[0], x3[1], y)  
 exp4 = exp\_raw(x1[1], x2[1], x3[0], y)  
  
 table = [exp1, exp2, exp3, exp4]  
 print(**"Таблиця експерименту:"**)  
 pprint.pprint(table)  
 print(**'\n'**)  
  
 **if not** cochrane\_kriteria(table):  
 **raise** Exception(**"Дисперсія неоднорідна перезапустіть програму"**)  
 **else**:  
  
 X1 = [exp1[0], exp2[0], exp3[0], exp4[0]]  
 X2 = [exp1[1], exp2[1], exp3[1], exp4[1]]  
 X3 = [exp1[2], exp2[2], exp3[2], exp4[2]]  
  
 y1mid = exp1[-2]  
 y2mid = exp2[-2]  
 y3mid = exp3[-2]  
 y4mid = exp4[-2]  
  
 mx1 = sum(X1)/4  
 mx2 = sum(X2)/4  
 mx3 = sum(X3)/4  
  
 my = (y1mid + y2mid + y3mid + y4mid)/4  
  
 a1 = (X1[0]\*y1mid + X1[1]\*y2mid + X1[2]\*y3mid + X1[3]\*y4mid)/4  
 a2 = (X2[0]\*y1mid + X2[1]\*y2mid + X2[2]\*y3mid + X2[3]\*y4mid)/4  
 a3 = (X3[0]\*y1mid + X3[1]\*y2mid + X3[2]\*y3mid + X3[3]\*y4mid)/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 = (X1[0]\*X2[0] + X1[1]\*X2[1] + X1[2]\*X2[2] + X1[3]\*X2[3])/4  
 a21 = a12  
 a13 = (X1[0]\*X3[0] + X1[1]\*X3[1] + X1[2]\*X3[2] + X1[3]\*X3[3])/4  
 a31 = a13  
 a23 = (X2[0]\*X3[0] + X2[1]\*X3[1] + X2[2]\*X3[2] + X2[3]\*X3[3])/4  
 a32 = a23  
  
 b0 = (lg.det([[my, mx1, mx2, mx3],  
 [a1, a11, a12, a13],  
 [a2, a12, a22, a32],  
 [a3, a13, a23, a33]]))/(lg.det([[1, mx1, mx2, mx3],  
 [mx1, a11, a21, a31],  
 [mx2, a12, a22, a32],  
 [mx3, a13, a23, a33]]))  
 b1 = (lg.det([[1, my, mx2, mx3],  
 [mx1, a1, a12, a13],  
 [mx2, a2, a22, a32],  
 [mx3, a3, a23, a33]]))/(lg.det([[1, mx1, mx2, mx3],  
 [mx1, a11, a21, a31],  
 [mx2, a12, a22, a32],  
 [mx3, a13, a23, a33]]))  
 b2 = (lg.det([[1, mx1, my, mx3],  
 [mx1, a11, a1, a13],  
 [mx2, a12, a2, a32],  
 [mx3, a13, a3, a33]]))/(lg.det([[1, mx1, mx2, mx3],  
 [mx1, a11, a21, a31],  
 [mx2, a12, a22, a32],  
 [mx3, a13, a23, a33]]))  
 b3 = (lg.det([[1, mx1, mx2, my],  
 [mx1, a11, a12, a1],  
 [mx2, a12, a22, a2],  
 [mx3, a13, a23, a3]]))/(lg.det([[1, mx1, mx2, mx3],  
 [mx1, a11, a21, a31],  
 [mx2, a12, a22, a32],  
 [mx3, a13, a23, a33]]))  
  
 b = [b0, b1, b2, b3]  
 print(**'Коефіцієнти bx:'**)  
 print(b)  
 print(**'\n'**)  
  
 check1 = check(b, x1, x2, x3)  
 print(**"Перевірка коефіцієнтів:"**)  
 print(check1)  
 print(**'\n'**)  
  
 student\_krit = student\_kriteria(table)  
 indexes = student\_krit[0]  
 Sb = student\_krit[1]  
 student\_b = list(map(**lambda** x: x **if** b.index(x) **in** indexes **else** 0, b))  
  
 print(**'Значимі коефіцієнти за Стюдентом:'**)  
 print(student\_b)  
 print(**'\n'**)  
  
 student\_checks = check(student\_b, x1, x2, x3)  
 print(**'y^ :'**)  
 print(student\_checks)  
 print(**'\n'**)  
  
 message = **''  
 if** fisher\_kriteria(table, student\_checks, Sb):  
 message = **'Рівняння регресії адекватно оригіналу при рівні значимості 0.05'  
 else**:  
 message = **'Рівняння регресії неадекватно оригіналу при рівні значимості 0.05'** print(message)  
  
**def** check(koeficients, x1 = [-1, 1], x2 = [-1, 1], x3 = [-1, 1]):  
 check1 = koeficients[0] + x1[0]\*koeficients[1] + x2[0]\*koeficients[2] + x3[0]\*koeficients[3]  
 check2 = koeficients[0] + x1[0]\*koeficients[1] + x2[1]\*koeficients[2] + x3[1]\*koeficients[3]  
 check3 = koeficients[0] + x1[1]\*koeficients[1] + x2[0]\*koeficients[2] + x3[1]\*koeficients[3]  
 check4 = koeficients[0] + x1[1]\*koeficients[1] + x2[1]\*koeficients[2] + x3[0]\*koeficients[3]  
 **return** [ check1, check2, check3, check4]  
  
**def** cochrane\_kriteria(table, N = 4):  
 sigma = [table[i][-1] **for** i **in** range(N)]  
 Gt = 0.7679  
 Gp = max(sigma)/sum(sigma)  
 **if** Gp < Gt:  
 **return** 1  
 **else**:  
 **return** 0  
  
**def** student\_kriteria(table, m = 3, N = 4):  
 sigma = [table[i][-1] **for** i **in** range(N)]  
 y\_mid = [table[i][-2] **for** i **in** range(N)]  
 SB = sum(sigma)/N  
 Sb = SB/(N\*m)  
 Sbeta = sqrt(Sb)  
  
 beta0 = 1/4\*(y\_mid[0]\*1 + y\_mid[1]\*1 + y\_mid[2]\*1 + y\_mid[3]\*1)  
 beta1 = 1/4\*(y\_mid[0]\*(-1) + y\_mid[1]\*(-1) + y\_mid[2]\*1 + y\_mid[3]\*1)  
 beta2 = 1/4\*(y\_mid[0]\*(-1) + y\_mid[1]\*1 + y\_mid[2]\*(-1) + y\_mid[3]\*1)  
 beta3 = 1/4\*(y\_mid[0]\*(-1) + y\_mid[1]\*1 + y\_mid[2]\*1 + y\_mid[3]\*(-1))  
  
 ttab = 2.306  
  
 t0 = abs(beta0)/Sbeta  
 t1 = abs(beta1)/Sbeta  
 t2 = abs(beta2)/Sbeta  
 t3 = abs(beta3)/Sbeta  
  
 t = [t0, t1, t2, t3]  
 indexes = []  
 **for** i **in** t:  
 **if** i > ttab:  
 indexes.append(t.index(i))  
  
 **return** [indexes, Sb]  
  
**def** fisher\_kriteria(table, checks, Sb, d = 2, N = 4, m = 3):  
 y\_mid = [table[i][-2] **for** i **in** range(N)]  
 Sad = 0  
 **for** i **in** range(N):  
 Sad += m/(N-d)\*(checks[i] - y\_mid[i])\*\*2  
 Ft = 4.5  
 Fp = Sad/Sb  
 print(**'Fp :'**)  
 print(Fp)  
 print(**'\n'**)  
 **if** Fp > Ft:  
 **return** 0  
 **else**:  
 **return** 1  
  
  
  
x1 = [-30, 20]  
print(**'x1min = {0}, x1max = {1}'**.format(x1[0], x1[1]))  
x2 = [-70, -10]  
print(**'x2min = {0}, x2max = {1}'**.format(x2[0], x2[1]))  
x3 = [-70, -40]  
print(**'x3min = {0}, x3max = {1}\n'**.format(x3[0], x3[1]))  
  
  
xcmax = int(1/3\*(x1[1] + x2[1] + x3[1]))  
xcmin = int(1/3\*(x1[0] + x2[0] + x3[0]))  
print(**'Xcmin = {0}, Xcmax = {1}\n'**.format(xcmin, xcmax))  
ymax = 200 + xcmax  
ymin = 200 + xcmin  
y = [ymin, ymax]  
print(**'ymin = {0}, ymax = {1}\n'**.format(ymin, ymax))  
  
  
experiment(x1, x2, x3, y)

**Результат роботи програми**

