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

Желепа В. В.

Залікова – 9110

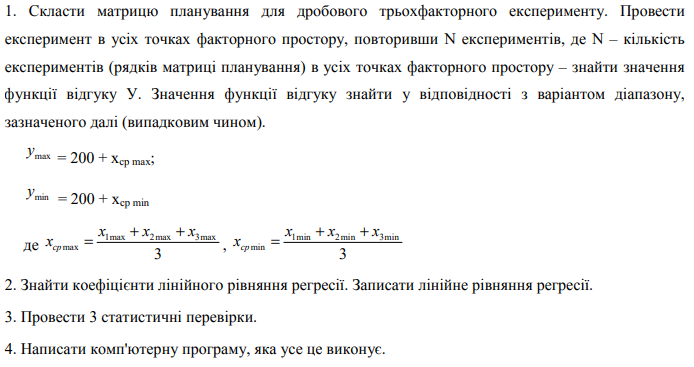
ПЕРЕВІРИВ:

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**

*from* random *import* \*  
*from* pprint *import* pprint  
*import* numpy *as* np  
*from* math *import* sqrt  
m, N, d = 3, 4, 4  
mat\_sX = [[-25, -5], [-30, 45], [-5, 5]]  
mat\_1X = [[1, -1, -1, -1], [1, -1, 1, 1], [1, 1, -1, 1], [1, 1, 1, -1]]  
tran1 = [*list*(i) *for* i *in zip*(\*mat\_1X)]  
x\_min\_max = [*sum*(mat\_sX[i][k] *for* i *in range*(3))/3 *for* k *in range*(2)]  
y\_min\_max = [*int*(200 + x\_min\_max[i]) *for* i *in range*(2)]  
*print*('Задана матриця Х:\n', mat\_sX, '\nНормована матриця Х:')  
pprint(mat\_1X, *width*=17)  
*print*('Xср min and max:\n', x\_min\_max, '\nY min and max:\n', y\_min\_max, '\nМатриця Y:')  
mat\_Y = [[randint(y\_min\_max[0], y\_min\_max[1]) *for* i *in range*(3)] *for* k *in range*(4)]  
pprint(mat\_Y, *width*=17)  
mat\_serY = [*sum*(mat\_Y[k1])/3 *for* k1 *in range*(4)]  
*print*('Середні значення Y:\n', mat\_serY, '\nМатриця Х:')  
mat\_X = [[-25, -30, -5], [-25, 45, 5], [-5, -30, 5], [-5, 45, -5]]  
pprint(mat\_X, *width*=17)  
mx = [*sum*(mat\_X[i][k] *for* i *in range*(4))/4 *for* k *in range*(3)]  
my = *sum*(mat\_serY)/4  
*print*('Значення mxi:\n', mx, '\nЗначення my:\n', my)  
tran = [*list*(i) *for* i *in zip*(\*mat\_X)]  
ai = [*sum*(tran[k][i] \* mat\_serY[i] *for* i *in range*(4)) / 4 *for* k *in range*(3)]  
aii = [*sum*(tran[k][i]\*\*2 *for* i *in range*(4)) / 4 *for* k *in range*(3)]  
*print*('Значення ai:\n', ai, '\nЗначення aii:\n', aii)  
a12 = a21 = (tran[0][0] \* tran[1][0] + tran[0][1] \* tran[1][1] + tran[0][2] \* tran[1][2] + tran[0][3] \* tran[1][3])/4  
a13 = a31 = (tran[0][0] \* tran[2][0] + tran[0][1] \* tran[2][1] + tran[0][2] \* tran[2][2] + tran[0][3] \* tran[2][3])/4  
a23 = a32 = (tran[1][0] \* tran[2][0] + tran[1][1] \* tran[2][1] + tran[1][2] \* tran[2][2] + tran[1][3] \* tran[2][3])/4  
znamen = np.linalg.det(np.matrix([[1, mx[0], mx[1], mx[2]], [mx[0], aii[0], a12, a13], [mx[1], a12, aii[1], a32], [mx[2], a13, a23, aii[2]]]))  
b0 = np.linalg.det(np.matrix([[my, mx[0], mx[1], mx[2]], [ai[0], aii[0], a12, a13], [ai[1], a12, aii[1], a32], [ai[2], a13, a23, aii[2]]]))/znamen  
b1 = np.linalg.det(np.matrix([[1, my, mx[1], mx[2]], [mx[0], ai[0], a12, a13], [mx[1], ai[1], aii[1], a32], [mx[2], ai[2], a23, aii[2]]]))/znamen  
b2 = np.linalg.det(np.matrix([[1, mx[0], my, mx[2]], [mx[0], aii[0], ai[0], a13], [mx[1], a12, ai[1], a32], [mx[2], a13, ai[2], aii[2]]]))/znamen  
b3 = np.linalg.det(np.matrix([[1, mx[0], mx[1], my], [mx[0], aii[0], a12, ai[0]], [mx[1], a12, aii[1], ai[1]], [mx[2], a13, a23, ai[2]]]))/znamen  
perevirka = [b0 + b1 \* tran[0][i] + b2 \* tran[1][i] + b3 \* tran[2][i] *for* i *in range*(4)]  
blist = [b0, b1, b2, b3]  
*print*("Перевірка порівнянням з середніми значеннями Y:\n", perevirka)  
mat\_disY = [*sum*([((k1 - mat\_serY[j]) \*\* 2) *for* k1 *in* mat\_Y[j]]) / m *for* j *in range*(4)]  
*print*("Дисперсії в рядках:\n", mat\_disY)  
*print*('-------------------------------------------------------------------------\nПЕРЕВІРКА ОДНОРІДНОСТІ ДИСПЕРСІЇ ЗА КРИТЕРІЄМ КОХРЕНА:\n\n...\n..')  
*if max*(mat\_disY)/*sum*(mat\_disY) < 0.7679:  
 *print*('Дисперсія однорідна')  
*else*:  
 *print*('Дисперсія неоднорідна')  
*print*('-------------------------------------------------------------------------\nПЕРЕВІРКА ЗНАЧУЩОСТІ КОЕФІЦІЄНТІВ ЗА КРИТЕРІЄМ СТЬЮДЕНТА:\n')  
S2b = *sum*(mat\_disY) / N  
S2bs = S2b / (m \* N)  
Sbs = sqrt(S2bs)  
*print*('Sbs:\n', Sbs)  
bb = [*sum*(mat\_serY[k] \* tran1[i][k] *for* k *in range*(N))/N *for* i *in range*(N)]  
t = [*abs*(bb[i])/Sbs *for* i *in range*(N)]  
*print*('bi:\n', bb, '\nti:\n', t, '\n...\n..')  
*for* i *in range*(N):  
 *if* t[i] < 2.306:  
 blist[i] = 0  
 d -= 1  
 *print*('Виключаємо з рівняння коефіціент b', i)  
y\_reg = [blist[0] + blist[1] \* mat\_X[i][0] + blist[2] \* mat\_X[i][1] + blist[3] \* mat\_X[i][2] *for* i *in range*(4)]  
*print*('Значення рівнянь регресій:\n', y\_reg)  
*print*('-------------------------------------------------------------------------\nПЕРЕВІРКА АДЕКВАТНОСТІ ЗА КРИТЕРІЄМ ФІШЕРА:\n')  
Sad = (m / (N - d)) \* *int*(*sum*(y\_reg[i] - mat\_serY[i] *for* i *in range*(N))\*\*2)  
Fp = Sad / S2b  
*print*('Кількість значимих коефіціентів:\n', d, '\nFp:\n', Fp, '\n...\n..')  
*if* Fp > 4.5:  
 *print*('Рівняння регресії неадекватно оригіналу при рівні значимості 0.05')  
*else*:  
 *print*('Рівняння регресії адекватно оригіналу при рівні значимості 0.05')

**Результат роботи програми**

