Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 2

з дисципліни «Методи наукових досліджень»

на тему «ПРОВЕДЕННЯ ДВОФАКТОРНОГО 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

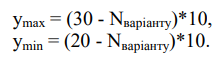
**Мета**: провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.

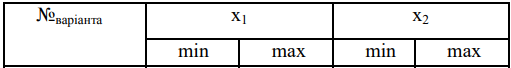
**Завдання:**

1. Записати лінійне рівняння регресії.

2. Обрати тип двофакторного експерименту і скласти матрицю планування для нього з використанням додаткового нульового фактору (хо=1).

3. Провести експеримент в усіх точках повного факторного простору (знайти значення функції відгуку y). Значення функції відгуку задати випадковим чином у відповідності до варіанту у діапазоні ymin ÷ ymax



Варіанти обираються по номеру в списку в журналі викладача. 



4. Перевірити однорідності дисперсії за критерієм Романовського

5. Знайти коефіцієнти нормованих рівнянь регресії і виконати перевірку (підставити значення нормованих факторів і коефіцієнтів у рівняння).

6. Провести натуралізацію рівняння регресії й виконати перевірку натуралізованого рівняння.

7. Написати комп'ютерну програму, яка все це виконує.

**Програмний код**

*from* random *import* randint  
*from* pprint *import* pprint  
*import* numpy *as* np  
*from* math *import* sqrt  
m = 8  
roman = {(2, 3, 4): 1.72, (5, 6, 7): 2.13, (8, 9): 2.37, (10, 11): 2.54, (12, 13): 2.66, (14, 15, 16, 17): 2.8, (18, 19, 20): 2.96}  
mat\_sX = [-25, -5, -30, 45]  
mat\_sY = [(30 - 110) \* 10, (20 - 110) \* 10]  
*print*("Задані Х:\n", mat\_sX, "\nЗадані Y:\n", mat\_sY, "\nНормована матриця планування експерименту:")  
mat\_X = [[-1, -1], [1, -1], [-1, 1]]  
  
  
*def* start(m):  
 *global* mat\_serY  
 mat\_Y = [[randint(mat\_sY[1], mat\_sY[0]) *for* i *in range*(m)] *for* k *in range*(3)]  
 *print*(mat\_X, "\nМатриця згенерованих значень Y:")  
 pprint(mat\_Y)  
 mat\_serY = [*sum*(mat\_Y[k1]) / m *for* k1 *in range*(3)]  
 *print*("Середні значення Y:", mat\_serY)  
 mat\_disY = [*sum*([((k1 - mat\_serY[j]) \*\* 2) *for* k1 *in* mat\_Y[j]]) / m *for* j *in range*(3)]  
 *print*("Дисперсії в рядках:", mat\_disY)  
 vidhil = sqrt((2 \* (2 \* m - 2)) / (m \* (m - 4)))  
 mat\_Fuv = [mat\_disY[0] / mat\_disY[1], mat\_disY[2] / mat\_disY[0], mat\_disY[2] / mat\_disY[1]]  
 *print*("Обчислеенні значення Fuv: ", mat\_Fuv)  
 mat\_Q = [mat\_Fuv[i2] \* ((m - 2) / m) *for* i2 *in range*(3)]  
 *print*("Обчислеенні значення Quv: ", mat\_Q)  
 mat\_R = [*abs*(mat\_Q[i3] - 1) / vidhil *for* i3 *in range*(3)]  
 *print*("Обчислеенні значення Ruv: ", mat\_R)  
 *for* i *in* roman.keys():  
 *if* m *in* i:  
 value = roman[i]  
 *break  
 if* m >= 21:  
 *print*("Занадто велике значення М")  
 *exit*()  
 *return max*(mat\_R) < value  
  
  
*if* start(m):  
 *print*("Однорідна")  
 *print*("M = ", m)  
*while not* start(m):  
 m += 1  
 *print*("Неоднорідна")  
tran = [*list*(i) *for* i *in zip*(\*mat\_X)]  
mx\_mat = [*sum*(tran[i]) / 3 *for* i *in range*(2)]  
mx1 = mx\_mat[0]  
mx2 = mx\_mat[1]  
my = *sum*(mat\_serY) / 3  
*print*("Нормовані коефіцієнти рівняння регресії:\nmx1, mx2: ", mx\_mat, "\nmy:", my)  
a1 = (tran[0][0] \*\* 2 + tran[0][1] \*\* 2 + tran[0][2] \*\* 2) / 3  
a2 = (tran[0][0] \* tran[1][0] + tran[0][1] \* tran[1][1] + tran[0][2] \* tran[1][2]) / 3  
a3 = (tran[1][0] \*\* 2 + tran[1][1] \*\* 2 + tran[1][2] \*\* 2) / 3  
ax1 = [(tran[i][0] \* mat\_serY[0] + tran[i][1] \* mat\_serY[1] + tran[i][2] \* mat\_serY[2]) / 3 *for* i *in range*(2)]  
a11 = ax1[0]  
a22 = ax1[1]  
*print*("a1: ", a1, "\na2: ", a2, "\na3: ", a3, "\na11: ", a11, "\na22: ", a22)  
znamen = np.linalg.det(np.array([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]]))  
b0 = np.linalg.det(np.array([[my, mx1, mx2], [a11, a1, a2], [a22, a2, a3]])) / znamen  
b1 = np.linalg.det(np.array([[1, my, mx2], [mx1, a11, a2], [mx2, a22, a3]])) / znamen  
b2 = np.linalg.det(np.array([[1, mx1, my], [mx1, a1, a11], [mx2, a2, a22]])) / znamen  
perevirka = [b0 + b1 \* tran[0][i] + b2 \* tran[1][i] *for* i *in range*(3)]  
*print*("Перевірка 1: ", perevirka)  
d\_x1 = *abs*(mat\_sX[1] - mat\_sX[0]) / 2  
d\_x2 = *abs*(mat\_sX[3] - mat\_sX[2]) / 2  
x10 = (mat\_sX[1] + mat\_sX[0]) / 2  
x20 = (mat\_sX[3] + mat\_sX[2]) / 2  
*print*("Дельта x1: ", d\_x1, "\nДельта x2: ", d\_x2, "\nx10: ", x10, "\nx20: ", x20)  
A0 = b0 - b1 \* x10 / d\_x1 - b2 \* x20 / d\_x2  
A1 = b1 / d\_x1  
A2 = b2 / d\_x2  
*print*("a0: ", A0, "\na1: ", A1, "\na2: ", A2)  
perevirka21 = A0 + A1 \* -25 + A2 \* -30  
perevirka22 = A0 + A1 \* -5 + A2 \* -30  
perevirka23 = A0 + A1 \* -25 + A2 \* 45  
perevirka2 = [perevirka21, perevirka22, perevirka23]  
*print*("Перевірка по рядкам 2: ", perevirka2)

**Результат роботи програми**

