Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 6

з дисципліни «Методи оптимізації та планування експерименту»

на тему «Проведення трьохфакторного експерименту при використанні рівняння регресії з квадратичними членами»

Варіант 110

ВИКОНАЛА:

студентка ІІ курсу ФІОТ

групи ІВ-71

Молчанова В.С.

ПЕРЕВІРИВ:

Асистент

Регіда П. Г.

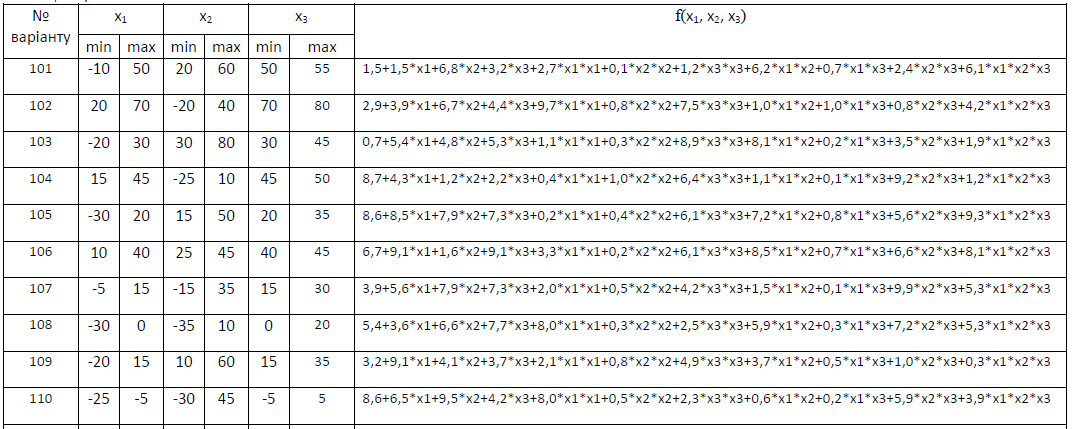
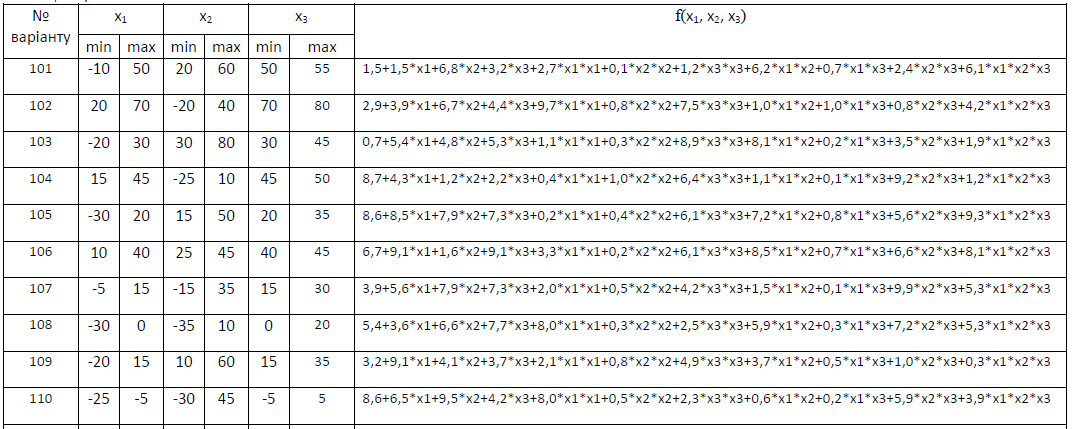
Київ – 2019

**Мета:**

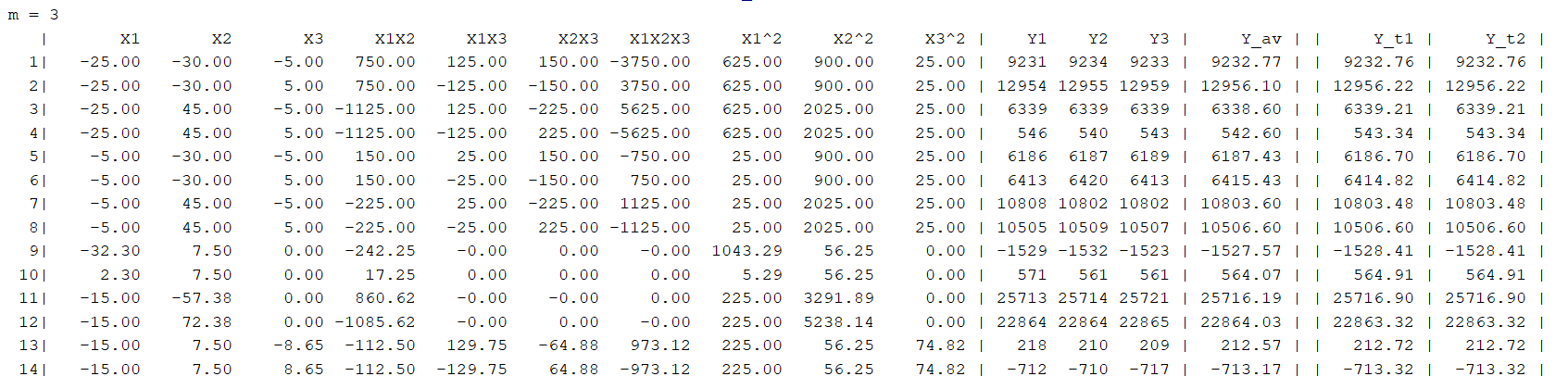
Провести трьохфакторний експеримент і отримати адекватну модель рівняння регресії, використовуючи рототабельний композиційний план.

**Хід роботи:**

**Варіант завдання:**



**Результати підготовки та виконання роботи:**



Рівняння регресії: у = -21.750 + 18.804\*x1 + 9.143\*x2 + 4.373\*x3 + 8.005\*x4 + 0.512\*x5 + 2.298\*x6 + 0.600\*x7 + 0.611\*x8 + 5.929\*x9 + 5.539\*x10

Дисперсії однорідні з ймовірнітю p = 0.95

Рівняння регресії із значущими коефіцієнтами: у = -21.750 + 18.804\*x1 + 9.143\*x2 + 4.373\*x3 + 8.005\*x4 + 0.512\*x5 + 2.298\*x6 + 0.600\*x7 + 0.611\*x8 + 5.929\*x9 + 5.539\*x10

Модель адекватна з ймовірнітю p = 0.95

**Код програми:**

**import** math  
**import** numpy **as** np  
**import** random  
**import** crit  
  
N = 14  
cur\_m = 3  
k\_simple = 3  
k = 10  
p = 0.95  
x = [[-25, -5],  
 [-30, 45],  
 [-5, 5]]  
  
b\_start = [8.6, 6.5, 9.5, 4.2, 8.0, 0.5, 2.3, 0.6, 0.2, 5.9, 3.9]  
  
  
**def** solve(k, N):  
 mx = [sum(x\_nat\_t[i]) / N **for** i **in** range(k)]  
 my = sum(y\_av) / N  
 a = []  
 **for** i **in** range(k):  
 a.append([])  
 **for** j **in** range(k):  
 a[i].append(sum([float(a \* b) **for** (a, b) **in** zip(x\_nat\_t[i], x\_nat\_t[j])]) / N)  
 a0 = [sum([a \* b **for** a, b **in** zip(x\_nat\_t[i], y\_av)]) / N **for** i **in** range(k)]  
 left = []  
 left.append(mx.copy())  
 left[0].insert(0, 1)  
 **for** i **in** range(k):  
 left.append([mx[i]])  
 **for** j **in** range(k):  
 left[i + 1].append(a[j][i])  
 right = a0.copy()  
 right.insert(0, my)  
 **return** np.linalg.solve(left, right)  
  
  
**def** print\_results(x\_nat, b, t\_main):  
 head = **" | {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} |"**.format(**"X1"**, **"X2"**, **"X3"**, **"X1X2"**, **"X1X3"**, **"X2X3"**, **"X1X2X3"**, **"X1^2"**, **"X2^2"**, **"X3^2"**)  
 **for** i **in** range(cur\_m):  
 head += **" {:>5}"**.format(**"Y{}"**.format(i + 1))  
 head += **" | {:>8} |"**.format(**"Y\_av"**)  
 head += **" | {:>8} | {:>8} |"**.format(**"Y\_t1"**, **"Y\_t2"**)  
 print(head)  
 **for** i **in** range(N):  
 s = **" {:>2}| "**.format(i+1)  
 **for** j **in** range(1, k+1):  
 s += **"{:8.2f} "**.format(x\_nat[i][j])  
 s += **"|"  
 for** j **in** range(cur\_m):  
 s += **" {:5.0f}"**.format(y[i][j])  
 s += **" | {:8.2f} |"**.format(y\_av[i])  
 s += **" | {:8.2f} |"**.format(y\_test1[i])  
 s += **" {:8.2f} |"**.format(y\_test2[i])  
 print(s)  
 rivn = **" {:6.3f}"**.format(b[0])  
 **for** i **in** range(1, k + 1):  
 **if** b[i] < 0:  
 sign = **"-"  
 else**:  
 sign = **"+"** rivn += **" {} {:.3f}\*x{}"**.format(sign, abs(b[i]), i)  
 print(**"\nРівняння регресії: у ="** + rivn)  
 print(**"Дисперсії однорідні з ймовірнітю p = {}"**.format(p))  
 rivn\_main = **" {:6.3f}"**.format(b[0]) **if** t\_main[0] > 0 **else ""  
 for** i **in** range(1, k + 1):  
 **if** b[i] < 0:  
 sign = **"-"  
 else**:  
 sign = **"+"  
 if** t\_main[i] == 1:  
 **if** i == 0:  
 rivn\_main += **" {} {:.3f}"**.format(sign, abs(b[i]))  
 **else**:  
 rivn\_main += **" {} {:.3f}\*x{}"**.format(sign, abs(b[i]), i)  
 **if** rivn\_main != **""**:  
 rivn\_main = rivn\_main[1:]  
 print(**"Рівняння регресії із значущими коефіцієнтами: у = "** + rivn\_main)  
 print(**"Модель адекватна з ймовірнітю p = {}"**.format(p))  
  
  
x\_norm = [[1, -1, -1, -1],  
 [1, -1, -1, 1],  
 [1, -1, 1, -1],  
 [1, -1, 1, 1],  
 [1, 1, -1, -1],  
 [1, 1, -1, 1],  
 [1, 1, 1, -1],  
 [1, 1, 1, 1],  
 [1, -1.73, 0, 0],  
 [1, 1.73, 0, 0],  
 [1, 0, -1.73, 0],  
 [1, 0, 1.73, 0],  
 [1, 0, 0, -1.73],  
 [1, 0, 0, 1.73],]  
  
x0 = [sum(row)/2 **for** row **in** x]  
delta\_x = [x[i][1] - x0[i] **for** i **in** range(k\_simple)]  
  
x\_nat = []  
**for** i **in** range(N):  
 x\_nat.append([1])  
 **for** j **in** range(1, k\_simple + 1):  
 **if** x\_norm[i][j] == -1:  
 x\_nat[i].append(x[j - 1][0])  
 **elif** x\_norm[i][j] == 1:  
 x\_nat[i].append(x[j - 1][1])  
 **elif** x\_norm[i][j] == 0:  
 x\_nat[i].append(x0[j - 1])  
 **else**:  
 x\_nat[i].append(x\_norm[i][j]\*delta\_x[j - 1] + x0[j-1])  
  
**for** i **in** range(N):  
 x\_norm[i].append(x\_norm[i][1] \* x\_norm[i][2])  
 x\_norm[i].append(x\_norm[i][1] \* x\_norm[i][3])  
 x\_norm[i].append(x\_norm[i][2] \* x\_norm[i][3])  
 x\_norm[i].append(x\_norm[i][1] \* x\_norm[i][2] \* x\_norm[i][3])  
 x\_norm[i].append(x\_norm[i][1] \* x\_norm[i][1])  
 x\_norm[i].append(x\_norm[i][2] \* x\_norm[i][2])  
 x\_norm[i].append(x\_norm[i][3] \* x\_norm[i][3])  
  
 x\_nat[i].append(x\_nat[i][1] \* x\_nat[i][2])  
 x\_nat[i].append(x\_nat[i][1] \* x\_nat[i][3])  
 x\_nat[i].append(x\_nat[i][2] \* x\_nat[i][3])  
 x\_nat[i].append(x\_nat[i][1] \* x\_nat[i][2] \* x\_nat[i][3])  
 x\_nat[i].append(x\_nat[i][1] \* x\_nat[i][1])  
 x\_nat[i].append(x\_nat[i][2] \* x\_nat[i][2])  
 x\_nat[i].append(x\_nat[i][3] \* x\_nat[i][3])  
  
x\_norm\_t = np.transpose(x\_norm)  
x\_nat\_t = list(np.transpose(x\_nat))  
x\_nat\_t.pop(0)  
  
**while** cur\_m < crit.max\_m:  
 print(**"m = {}"**.format(cur\_m))  
 y = [[sum([(a \* c) **for** a, c **in** zip(x\_nat[i], b\_start)]) + random.randint(0, 10) - 5 **for** j **in** range(cur\_m)] **for** i **in** range(N)]  
 y\_av = [sum(row) / cur\_m **for** row **in** y]  
  
 b = solve(k, N)  
  
 **if not** crit.Criteria.cohren(cur\_m, N, y\_av, y):  
 cur\_m += 1  
 print(**"Дисперсії неоднорідні, збільшуємо m"**)  
 **continue** t\_main = crit.Criteria.student(cur\_m, N, y\_av, y, x\_norm\_t, k)  
 b\_main = [b[i] \* t\_main[i] **for** i **in** range(k + 1)]  
  
 y\_test1 = [sum([(a \* c) **for** a, c **in** zip(x\_nat[i], b)]) **for** i **in** range(N)]  
 y\_test2 = [sum([(a \* b) **for** a, b **in** zip(x\_nat[i], b\_main)]) **for** i **in** range(N)]  
 **if** crit.Criteria.fisher(cur\_m, N, y\_av, y, t\_main, b\_main, x\_nat, k):  
 print\_results(x\_nat, b, t\_main)  
 **break** print(**"Модель неадекватна, збільшуємо m"**)  
 cur\_m += 1

**Висновок:**

Експериментальні значення функцій відгуку збігаються з середніми, отже коефіцієнти рівнянь регресії знайдені правильно.