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 № 3

з дисципліни «Методи оптимізації та планування експерименту»

на тему «Проведення трьохфакторного експерименту з використанням лінійного рівняння регресії»

Варіант 110

ВИКОНАЛА:

студентка ІІ курсу ФІОТ

групи ІВ-71

Молчанова В.С.

ПЕРЕВІРИВ:

Асистент

Регіда П. Г.

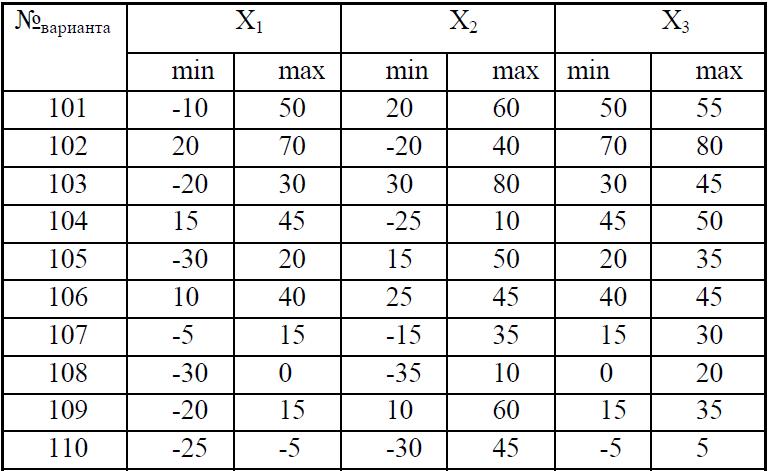
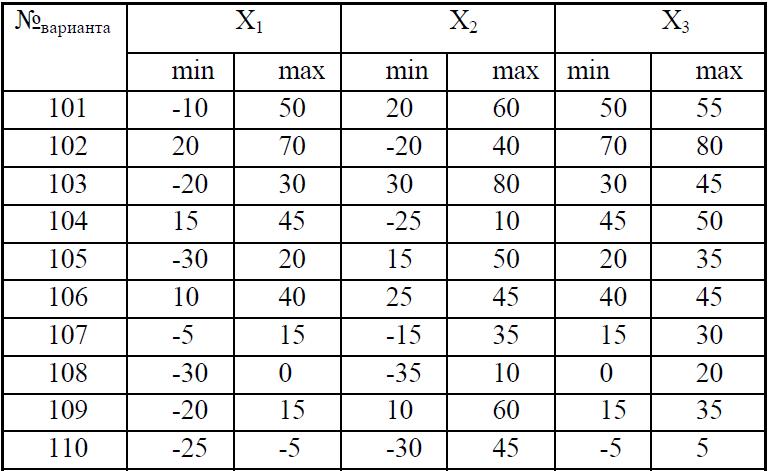
Київ – 2019

**Мета:**

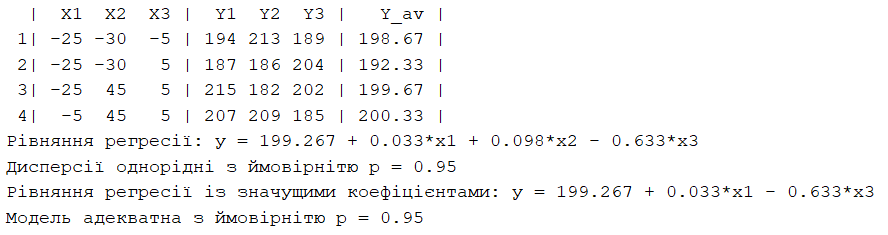
провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

**Хід роботи:**

**Варіант завдання:**



**Результати підготовки та виконання роботи:**



**Код програми:**

**import** math  
**import** numpy **as** np  
**import** random  
**import** crit  
  
k = 3  
N = 4  
m = 3  
p = 0.95  
x = [[-25, -5],  
 [-30, 45],  
 [-5, 5]]  
  
x\_av\_max = sum([row[1] **for** row **in** x])/3  
x\_av\_min = sum([row[0] **for** row **in** x])/3  
  
y\_max = 200+x\_av\_max  
y\_min = 200+x\_av\_min  
  
x\_norm = [[1, -1, -1, -1],  
 [1, -1, -1, 1],  
 [1, -1, 1, 1],  
 [1, 1, 1, 1]]  
  
x\_norm\_t = np.transpose(x\_norm)  
*#print("x\_norm:",x\_norm)*x\_nat = []  
**for** i **in** range(N):  
 x\_nat.append([1])  
 **for** j **in** range(1, k+1):  
 **if** x\_norm[i][j] < 0:  
 x\_nat[i].append(x[j - 1][0])  
 **else**:  
 x\_nat[i].append(x[j - 1][1])  
  
x\_nat\_t = list(np.transpose(x\_nat))  
x\_nat\_t.pop(0)  
  
y = [[random.randint(y\_min, y\_max) **for** j **in** range(m)] **for** i **in** range(N)]  
  
y\_av = [sum(row)/m **for** row **in** y]  
  
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([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 = [[1, mx[0], mx[1], mx[2]]]  
**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)  
  
b = np.linalg.solve(left, right)  
  
*# Print*head = **" | {:>3} {:>3} {:>3} |"**.format(**"X1"**, **"X2"**, **"X3"**)  
**for** i **in** range(m):  
 head += **" {:>3}"**.format(**"Y{}"**.format(i+1))  
head += **" | {:>6} |"**.format(**"Y\_av"**)  
print(head)  
**for** i **in** range(N):  
 s = **" {}| {:3} {:3} {:3} |"**.format(i+1, x\_nat[i][1], x\_nat[i][2], x\_nat[i][3])  
 **for** j **in** range(m):  
 s += **" {:.0f}"**.format(y[i][j])  
 s += **" | {:.2f} |"**.format(y\_av[i])  
 print(s)  
rivn = **" {:.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(**"Рівняння регресії: у ="**+rivn)  
  
*# Kohren*s2 = [sum([(y\_av[i]-y[i][j])\*\*2 **for** j **in** range(m)])/m **for** i **in** range(N)]  
  
s2\_max = max(s2)  
Gp = s2\_max/sum(s2)  
  
f1 = m-1  
f2 = N  
q = 1 - p  
G\_cr = 7.679  
**if** crit.Criteria.cohren(m, N, y\_av, y):  
 print(**"Дисперсії однорідні з ймовірнітю p = {}"**.format(p))  
**else**:  
 print(**"Дисперсії неоднорідні"**)  
  
*# Student*S2B = sum(s2)/N  
  
S2\_beta = S2B/(N\*m)  
S\_beta = math.sqrt(S2\_beta)  
  
beta = [sum([x\_norm\_t[i][j]\*y\_av[j] **for** j **in** range(N)])/N **for** i **in** range(k+1)]  
  
t = [abs(beta[i])/S\_beta **for** i **in** range(N)]  
f3 = (m-1)\*N  
t\_table = 2.306  
t\_main = crit.Criteria.student(m, N, y\_av, y, x\_norm\_t, k)  
b\_main = [b[i]\*t\_main[i] **for** i **in** range(k+1)]  
rivn = **""  
for** i **in** range(k+1):  
 **if** b[i] < 0:  
 sign = **"-"  
 else**:  
 sign = **"+"  
 if** t\_main[i] == 1:  
 **if** i == 0:  
 rivn += **" {} {:.3f}"**.format(sign, abs(b[i]))  
 **else**:  
 rivn += **" {} {:.3f}\*x{}"**.format(sign, abs(b[i]), i)  
**if** rivn != **""**:  
 rivn = rivn[2:]  
print(**"Рівняння регресії із значущими коефіцієнтами: у ="**+rivn)  
  
*# Fisher*d = sum(t\_main)  
y\_main = [sum([a\*b **for** (a,b) **in** zip(b\_main, x\_nat[i])]) **for** i **in** range(N)]  
**if** d < k+1:  
 F\_cr = [-1, 5.3, 4.5, 4.1, 3.8, 3.7, 3.6]  
 f4 = N - d  
 S2\_ad = m/f4\*sum([(y\_main[i]-y\_av[i])\*\*2 **for** i **in** range(N)])  
 F = S2\_ad/S2B  
 **if** F < F\_cr[f4]:  
 print(**"Модель адекватна з ймовірнітю p = {}"**.format(p))  
 **else**:  
 print(**"Модель неадекватна"**)  
**else**:  
 print(**"Модель адекватна з ймовірнітю p = {}"**.format(p))

**Висновок:**

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