Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №4**

«Проведення трьохфакторного експерименту

при використанні рівняння регресії з урахуванням ефекту взаємодії**»**

**Виконав:**

студент II курсу ФІОТ

групи ІО-93

Довгаль Богдан

**Перевірив:**

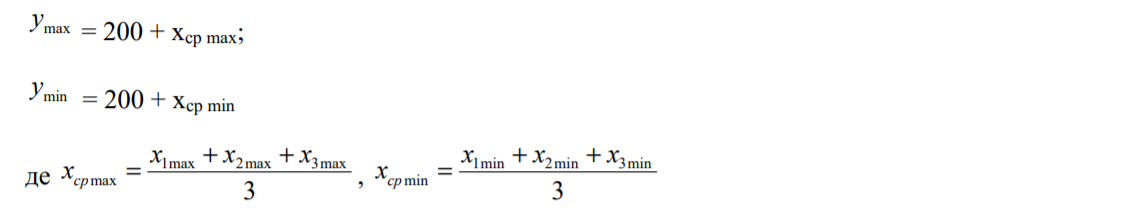
Регіда П.Г.

Київ – 2021

**Мета роботи:** Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

**Завдання на лабораторну роботу:**

1. Скласти матрицю планування для повного трьохфакторного експерименту.

2. Провести експеримент, повторивши N раз досліди у всіх точках факторного простору і знайти значення відгуку Y. Знайти значення Y шляхом моделювання випадкових чисел у певному діапазоні відповідно варіанту. Варіанти вибираються за номером в списку в журналі викладача.

2. Знайти коефіцієнти лінійного рівняння регресії. Записати лінійне рівняння регресії.

3. Провести 3 статистичні перевірки.

4. Зробити висновки по адекватності регресії та значимості окремих коефіцієнтів і записати скореговане рівняння регресії.

5. Написати комп'ютерну програму, яка усе це виконує.

**Варіант завдання:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Варіант | X1 | | X2 | | X3 | |
| min | max | min | max | min | max |
| 308 | -5 | 15 | 25 | 45 | 15 | 45 |

Довірча ймовірність дорівнює 0.95, а рівень значимості q = 0.05.

**Роздруківка тексту програми:**

**import** random  
**from** prettytable **import** PrettyTable  
  
m = 3  
n = 8  
x1\_min = -5  
x1\_max = 15  
x2\_min = 25  
x2\_max = 45  
x3\_min = 15  
x3\_max = 45  
y\_min = 200 + (x1\_min + x2\_min + x3\_min) / 3  
y\_max = 200 + (x1\_max + x2\_max + x3\_max) / 3  
y\_matrix = [[random.randint(int(y\_min), int(y\_max)) **for** \_ **in** range(m)] **for** \_ **in** range(n)]  
y\_average = [round(sum(i) / len(i), 3) **for** i **in** y\_matrix]  
xn = [[-1, -1, -1],  
 [-1, -1, 1],  
 [-1, 1, -1],  
 [-1, 1, 1],  
 [1, -1, -1],  
 [1, -1, 1],  
 [1, 1, -1],  
 [1, 1, 1]]  
  
b0 = sum(y\_average) / n  
b1 = sum([y\_average[i] \* xn[i][0] **for** i **in** range(n)]) / n  
b2 = sum([y\_average[i] \* xn[i][1] **for** i **in** range(n)]) / n  
b3 = sum([y\_average[i] \* xn[i][2] **for** i **in** range(n)]) / n  
b12 = sum([y\_average[i] \* xn[i][0] \* xn[i][1] **for** i **in** range(n)]) / n  
b13 = sum([y\_average[i] \* xn[i][0] \* xn[i][2] **for** i **in** range(n)]) / n  
b23 = sum([y\_average[i] \* xn[i][1] \* xn[i][2] **for** i **in** range(n)]) / n  
b123 = sum([y\_average[i] \* xn[i][0] \* xn[i][1] \* xn[i][2] **for** i **in** range(n)]) / n  
  
plan\_matrix = [[x1\_min, x2\_min, x3\_min, x1\_min \* x2\_min, x1\_min \* x3\_min, x2\_min \* x3\_min, x1\_min \* x2\_min \* x3\_min],  
 [x1\_min, x2\_min, x3\_max, x1\_min \* x2\_min, x1\_min \* x3\_max, x2\_min \* x3\_max, x1\_min \* x2\_min \* x3\_max],  
 [x1\_min, x2\_max, x3\_min, x1\_min \* x2\_max, x1\_min \* x3\_min, x2\_max \* x3\_min, x1\_min \* x2\_max \* x3\_min],  
 [x1\_min, x2\_max, x3\_max, x1\_min \* x2\_max, x1\_min \* x3\_max, x2\_max \* x3\_max, x1\_min \* x2\_max \* x3\_max],  
 [x1\_max, x2\_min, x3\_min, x1\_max \* x2\_min, x1\_max \* x3\_min, x2\_min \* x3\_min, x1\_max \* x2\_min \* x3\_min],  
 [x1\_max, x2\_min, x3\_max, x1\_max \* x2\_min, x1\_max \* x3\_max, x2\_min \* x3\_max, x1\_max \* x2\_min \* x3\_max],  
 [x1\_max, x2\_max, x3\_min, x1\_max \* x2\_max, x1\_max \* x3\_min, x2\_max \* x3\_min, x1\_max \* x2\_max \* x3\_min],  
 [x1\_max, x2\_max, x3\_max, x1\_max \* x2\_max, x1\_max \* x3\_max, x2\_max \* x3\_max, x1\_max \* x2\_max \* x3\_max]]  
  
y\_result = []  
**for** i **in** range(n):  
 y\_result.append(b0 + b1 \* plan\_matrix[i][0] + b2 \* plan\_matrix[i][1] + b3 \* plan\_matrix[i][2] +  
 b12 \* plan\_matrix[i][3] + b13 \* plan\_matrix[i][4] + b23 \* plan\_matrix[i][5] +  
 b123 \* plan\_matrix[i][6])  
  
dispersion = [round(sum([(y\_matrix[j][i] - y\_average[i]) \*\* 2 **for** i **in** range(m)]) / m, 3) **for** j **in** range(n)]  
  
table1 = PrettyTable()  
table1.field\_names = [**"X0"**, **"X1"**, **"X2"**, **"X3"**, **"X12"**, **"X13"**, **"X23"**, **"X123"**, **"Y1"**, **"Y2"**, **"Y3"**, **"Y average"**, **"S^2"**]  
x0 = [[1] **for** \_ **in** range(n)]  
**for** i **in** range(n):  
 table1.add\_row([\*x0[i], \*plan\_matrix[i], \*y\_matrix[i], y\_average[i], dispersion[i]])  
print(**'Матриця планування:'**)  
print(table1)  
print()  
  
*# Критерій Кохрена*print(**"Перевірка за критерієм Кохрена:"**)  
gp = max(dispersion) / sum(dispersion)  
gt = 0.5157  
**if** gp < gt:  
 print(**"За критерієм Кохрена дисперсія однорідна"**)  
 print(**"{} < {}"**.format(round(gp, 3), round(gt, 3)))  
**else**:  
 print(**"За критерієм Кохрена дисперсія однорідна"**)  
 print(**"{} > {}"**.format(round(gp, 3), round(gt, 3)))  
print()  
  
*# Критерій Стьюденса*print(**"Перевірка значущості коефіцієнтів за критерієм Стьюдента"**)  
d = 8  
sb = sum(dispersion) / n  
s\_beta\_2 = sb / (n \* m)  
s\_beta = s\_beta\_2 \*\* (1 / 2)  
bb = [b0, b1, b2, b3, b12, b13, b23, b123]  
t\_list = [abs(bb[i]) / s\_beta **for** i **in** range(n)]  
tt = 2.120  
b\_list = [b0, b1, b2, b3, b12, b13, b23, b123]  
**for** i **in** range(n):  
 **if** t\_list[i] < tt:  
 b\_list[i] = 0  
 d -= 1  
**for** i **in** range(len(t\_list)):  
 print(**'t{} = {}'**.format(i, round(t\_list[i], 3)))  
print()  
  
*# Критерій Фішера*print(**"Перевірка адекватності за критерієм Фішера"**)  
y\_reg = [b0 + b1 \* plan\_matrix[i][0] + b2 \* plan\_matrix[i][1] + b3 \* plan\_matrix[i][2] +  
 b12 \* plan\_matrix[i][3] + b13 \* plan\_matrix[i][4] + b23 \* plan\_matrix[i][5] +  
 b123 \* plan\_matrix[i][6] **for** i **in** range(n)]  
sad = (m / (n - d)) \* int(sum([(y\_reg[i] - y\_average[i]) \*\* 2 **for** i **in** range(n)]))  
fp = sad / sb  
**if** fp < 4.5:  
 print(**'Рівняння регресії адекватне оригіналу на рівні 0.05'**)  
**else**:  
 print(**'Рівняння регресії неадекватне оригіналу на рівні 0.05'**)  
print()  
  
print(**'Рівняння:'**)  
print(**'y = {} + {} \* x1 + {} \* x2 + {} \* x3 + {} \* x1x2 + {} \* x1x3 + {} \* x2x3 + {} \* x1x2x3'** .format(round(b0, 3), round(b1, 3), round(b2, 3), round(b3, 3), round(b12, 3), round(b13, 3), round(b23, 3),  
 round(b123, 3)))  
print()  
**for** i **in** range(len(y\_result)):  
 print(**'y{} = {}'**.format(i+1, round(y\_result[i], 3)))

**Висновок:**

У ході лабораторної роботи я змоделював трьохфакторний експеримент. Також я реалізував 3 статистичні перевірки за критерієм Кохрена, Стьюдента та Фішера. Знайшов рівняння регресії адекватне об'єкту.