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 №4**

З дисципліни «Методи оптимізації та планування»

**Проведення трьохфакторного експерименту**

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

ВИКОНАВ:

Студент ІІ курсу ФІОТ

Групи ІВ-91

Богомол В.Ю. - 9101

ПЕРЕВІРИВ:

асистент

Регіда П.Г.

Київ 2021 р.

**Мета:**

Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

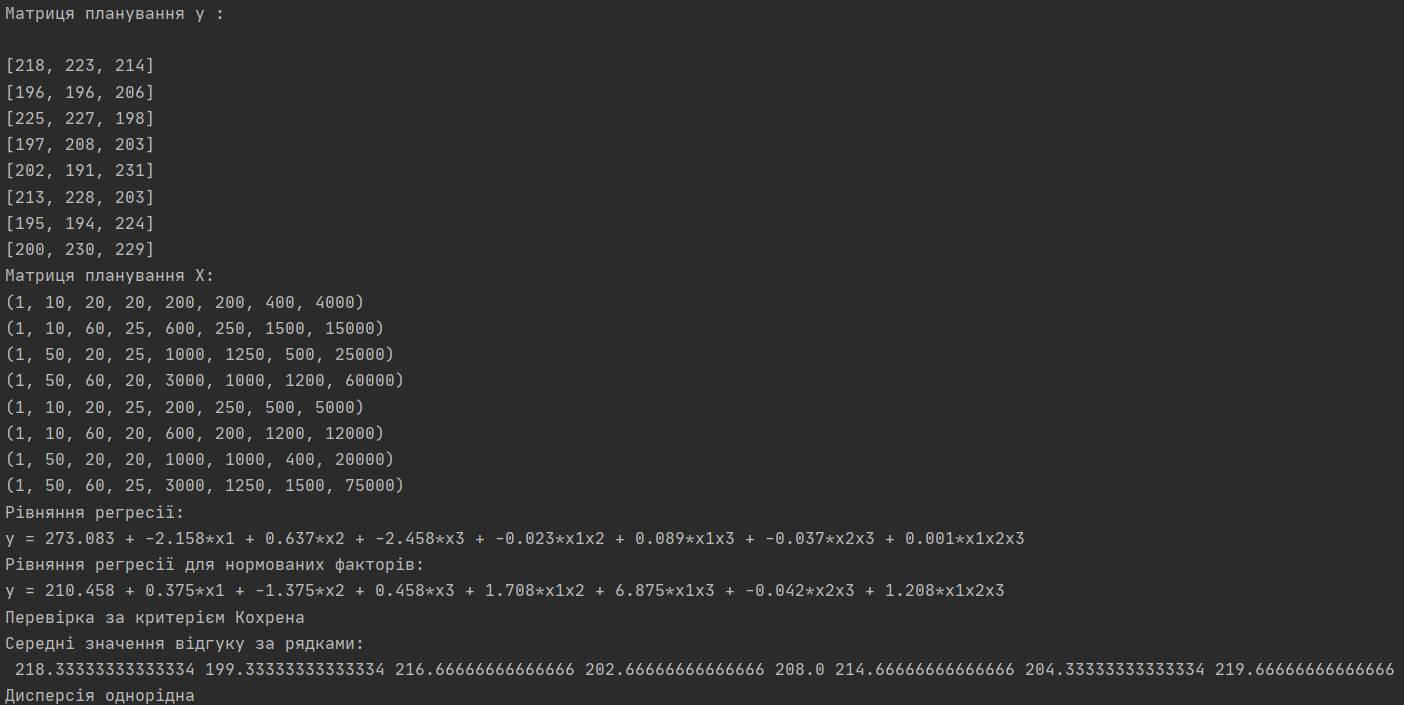
**Варіант завдання:**

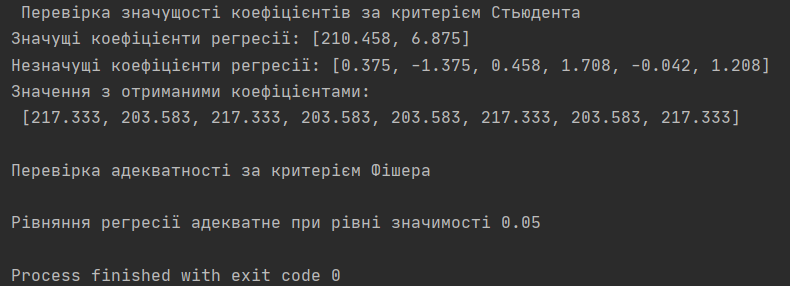
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Варіант | Х1 | | Х2 | | Х3 | |
| min | max | min | max | min | max |
| 101 | -40 | 20 | 10 | 60 | -20 | 20 |

**Лістинг програми:**

 import random  
import numpy as np  
from numpy.linalg import solve  
from scipy.stats import f, t  
  
# кількість повторення дослідів  
n = 8  
  
# варіант 101  
x1min = -40  
x1max = 20  
x2min = 10  
x2max = 60  
x3min = -20  
x3max = 20  
  
# максимальне та мінімальне значення  
y\_max = 200 + (x1max + x2max + x3max) / 3  
y\_min = 200 + (x1min + x2min + x3min) / 3  
  
# матриця ПФЕ  
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],  
 [-1, 1, 1, -1, 1, -1, -1, 1]]  
  
x1x2\_norm, x1x3\_norm, x2x3\_norm, x1x2x3\_norm = [0] \* 8, [0] \* 8, [0] \* 8, [0] \* 8  
  
for i in range(n):  
 x1x2\_norm[i] = xn[1][i] \* xn[2][i]  
 x1x3\_norm[i] = xn[1][i] \* xn[3][i]  
 x2x3\_norm[i] = xn[2][i] \* xn[3][i]  
 x1x2x3\_norm[i] = xn[1][i] \* xn[2][i] \* xn[3][i]  
  
# заповнення у(генерація)  
y1 = [random.randint(int(y\_min), int(y\_max)) for i in range(8)]  
y2 = [random.randint(int(y\_min), int(y\_max)) for i in range(8)]  
y3 = [random.randint(int(y\_min), int(y\_max)) for i in range(8)]  
  
# матриця планування  
y\_matrix = [[y1[0], y2[0], y3[0]],  
 [y1[1], y2[1], y3[1]],  
 [y1[2], y2[2], y3[2]],  
 [y1[3], y2[3], y3[3]],  
 [y1[4], y2[4], y3[4]],  
 [y1[5], y2[5], y3[5]],  
 [y1[6], y2[6], y3[6]],  
 [y1[7], y2[7], y3[7]]]  
  
# вивід данних за допомогою цикла  
print("Матриця планування y : \n")  
for i in range(n):  
 print(y\_matrix[i])  
  
x0 = [1, 1, 1, 1, 1, 1, 1, 1]  
  
# заміна -1 на х1\_мін, 1 на х1\_макс  
x1 = [10, 10, 50, 50, 10, 10, 50, 50]  
  
# заміна -1 на х2\_мін, 1 на х2\_макс  
x2 = [20, 60, 20, 60, 20, 60, 20, 60]  
  
# заміна -1 на х3\_мін, 1 на х3\_макс  
x3 = [20, 25, 25, 20, 25, 20, 20, 25]  
  
# заповнення нулями х1х2, х1х3, х1х2х3  
x1x2, x1x3, x2x3, x1x2x3 = [0] \* 8, [0] \* 8, [0] \* 8, [0] \* 8  
# заповнення х1х2, х1х3, х1х2х3 добутками  
for i in range(n):  
 x1x2[i] = x1[i] \* x2[i]  
 x1x3[i] = x1[i] \* x3[i]  
 x2x3[i] = x2[i] \* x3[i]  
 x1x2x3[i] = x1[i] \* x2[i] \* x3[i]  
# середні у  
Y\_average = []  
for i in range(len(y\_matrix)):  
 Y\_average.append(np.mean(y\_matrix[i], axis=0))  
  
# формуємо списки b i a  
list\_for\_b = [xn[0], xn[1], xn[2], xn[3], x1x2\_norm, x1x3\_norm, x2x3\_norm, x1x2x3\_norm]  
list\_for\_a = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3))  
  
# вивід матриці планування Х  
print("Матриця планування X:")  
for i in range(n):  
 print(list\_for\_a[i])  
# нормовані фактори b\_i  
bi = []  
for k in range(n):  
 S = 0  
 for i in range(n):  
 S += (list\_for\_b[k][i] \* Y\_average[i]) / n  
 bi.append(round(S, 3))  
# розрахунок аі(система рівнянь) через функцію solve, вивід рівняння регресії  
ai = [round(i, 3) for i in solve(list\_for\_a, Y\_average)]  
print("Рівняння регресії: \n" "y = {} + {}\*x1 + {}\*x2 + {}\*x3 + {}\*x1x2 + {}\*x1x3 + {}\*x2x3 + {}\*x1x2x3".format(ai[0],  
 ai[1],  
 ai[2],  
 ai[3],  
 ai[4],  
 ai[5],  
 ai[6],  
 ai[7]))  
# вивід даних  
print("Рівняння регресії для нормованих факторів: \n" "y = {} + {}\*x1 + {}\*x2 + {}\*x3 + {}\*x1x2 + {}\*x1x3 +"  
 " {}\*x2x3 + {}\*x1x2x3".format(bi[0], bi[1], bi[2], bi[3], bi[4], bi[5], bi[6], bi[7]))  
  
print("Перевірка за критерієм Кохрена")  
print("Середні значення відгуку за рядками:", "\n", +Y\_average[0], Y\_average[1], Y\_average[2], Y\_average[3],  
 Y\_average[4], Y\_average[5], Y\_average[6], Y\_average[7])  
# розрахунок дисперсій  
dispersions = []  
for i in range(len(y\_matrix)):  
 a = 0  
 for k in y\_matrix[i]:  
 a += (k - np.mean(y\_matrix[i], axis=0)) \*\* 2  
 dispersions.append(a / len(y\_matrix[i]))  
# експериментально  
Gp = max(dispersions) / sum(dispersions)  
# теоретично  
Gt = 0.5157  
# перевірка однорідності дисперсій  
if Gp < Gt:  
 print("Дисперсія однорідна")  
else:  
 print("Дисперсія неоднорідна")  
# критерій Стьюдента  
print(" Перевірка значущості коефіцієнтів за критерієм Стьюдента")  
sb = sum(dispersions) / len(dispersions)  
sbs = (sb / (8 \* 3)) \*\* 0.5  
  
t\_list = [abs(bi[i]) / sbs for i in range(0, 8)]  
  
d = 0  
res = [0] \* 8  
coef\_1 = []  
coef\_2 = []  
# кількість повторень кожної комбінації  
m = 3  
F3 = (m - 1) \* n  
# перевірка значущості коефіцієнтів  
for i in range(n):  
 if t\_list[i] < t.ppf(q=0.975, df=F3):  
 coef\_2.append(bi[i])  
 res[i] = 0  
 else:  
 coef\_1.append(bi[i])  
 res[i] = bi[i]  
 d += 1  
  
# вивід  
print("Значущі коефіцієнти регресії:", coef\_1)  
print("Незначущі коефіцієнти регресії:", coef\_2)  
  
# значення y з коефіцієнтами регресії  
y\_st = []  
for i in range(n):  
 y\_st.append(res[0] + res[1] \* xn[1][i] + res[2] \* xn[2][i] + res[3] \* xn[3][i] + res[4] \* x1x2\_norm[i] \  
 + res[5] \* x1x3\_norm[i] + res[6] \* x2x3\_norm[i] + res[7] \* x1x2x3\_norm[i])  
print("Значення з отриманими коефіцієнтами:\n", y\_st)  
  
# критерій Фішера  
print("\nПеревірка адекватності за критерієм Фішера\n")  
Sad = m \* sum([(y\_st[i] - Y\_average[i]) \*\* 2 for i in range(8)]) / (n - d)  
Fp = Sad / sb  
F4 = n - d  
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):  
 print("Рівняння регресії адекватне при рівні значимості 0.05")  
else:  
 print("Рівняння регресії неадекватне при рівні значимості 0.05")

**Результат виконання роботи:**





**Висновок:**

В даній лабораторній роботі я провів повний трьохфакторний експеримент з трьома статистичними.