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 наукових досліджень» на тему

«Проведення трьохфакторного експерименту при використанні рівняння регресії з урахуванням ефекту взаємодії.»

Виконав:

студент ІІ курсу ФІОТ

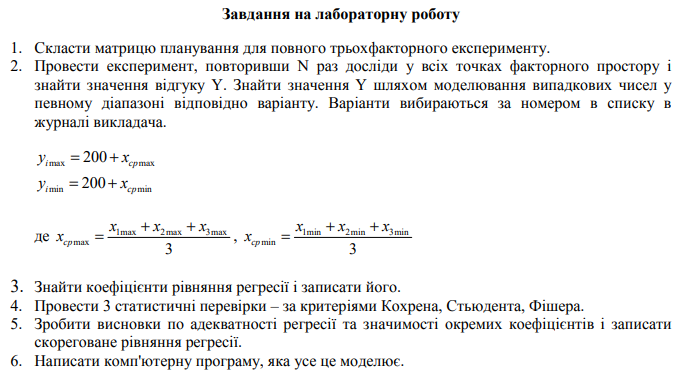
групи ІВ-93

Ільяш В. В.

ПЕРЕВІРИВ:

ас. Регіда П. Г.

Київ - 2021



**Варіант:**

Nваріант = 7





**Роздруківка коду програми:**

import math  
import numpy as np  
from numpy.linalg import solve  
from scipy.stats import f, t  
from functools import partial  
from random import randint  
  
  
while True:  
 fisher\_teor = partial(f.ppf, q=1 - 0.05)  
 student\_teor = partial(t.ppf, q=1 - 0.025)  
  
 ŁX1min = 10  
 X1max = 40  
 X2min = 15  
 X2max = 50  
 X3min = 10  
 X3max = 30  
  
 Xmax\_average = (X1max +łX2max +ŃX3max) / 3  
 Xmin\_average = (X1min +łX2min +ŃX3min) / 3  
  
 y\_max = round(200 + Xmax\_average)  
 y\_min = round(200 + Xmin\_average)  
  
 # Матриця ПФЕ  
 x0\_factor = [1, 1, 1, 1, 1, 1, 1, 1]  
 Łx1\_factor = [-1, -1, 1, 1, -1, -1, 1, 1]  
 łx2\_factor = [-1, 1, -1, 1, -1, 1, -1, 1]  
 Ńx3\_factor = [-1, 1, 1, -1, 1, -1, -1, 1]  
 Łx1x2\_factor = [a \* b for a, b in zip(x1\_factor, x2\_factor)]  
 Łx1x3\_factor = [a \* b for a, b in zip(x1\_factor, x3\_factor)]  
 łx2x3\_factor = [a \* b for a, b in zip(x2\_factor, x3\_factor)]  
 Łx1x2x3\_factor = [a \* b \* c for a, b, c in zip(x1\_factor, x2\_factor, x3\_factor)]  
  
 m = 3  
  
 y1, y2, y3 = [], [], []  
 for i in range(0, 8):  
 y1.append(randint(y\_min, y\_max))  
 y2.append(randint(y\_min, y\_max))  
 y3.append(randint(y\_min, y\_max))  
  
 Y\_average = [round(np.average([y1[0], y2[0], y3[0]]), 3), round(np.average([y1[1], y2[1], y3[1]]), 3),  
 round(np.average([y1[2], y2[2], y3[2]]), 3), round(np.average([y1[3], y2[3], y3[3]]), 3),  
 round(np.average([y1[4], y2[4], y3[4]]), 3), round(np.average([y1[5], y2[5], y3[5]]), 3),  
 round(np.average([y1[6], y2[6], y3[6]]), 3), round(np.average([y1[7], y2[7], y3[7]]), 3)]  
  
 ŀx0 = [1, 1, 1, 1, 1, 1, 1, 1]  
 Łx1 = [10, 10, 40, 40, 10, 10, 40, 40]  
 łx2 = [15, 50, 15, 50, 15, 50, 15, 50]  
 Ńx3 = [10, 30, 30, 10, 30, 10, 10, 30]  
 Łx1x2 = [a \* b for a, b in zip(x1, x2)]  
 Łx1x3 = [a \* b for a, b in zip(x1, x3)]  
 łx2x3 = [a \* b for a, b in zip(x2, x3)]  
 Łx1x2x3 = [a \* b \* c for a, b, c in zip(x1, x2, x3)]  
  
 list\_for\_solve\_b = [x0\_factor, x1\_factor, x2\_factor, x3\_factor, x1x2\_factor, x1x3\_factor, x2x3\_factor, x1x2x3\_factor]  
 list\_for\_solve\_a = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3))  
  
 N = 8  
 list\_bi = []  
 for k in range(N):  
 S = 0  
 for i in range(N):  
 S += (list\_for\_solve\_b[k][i] \* Y\_average[i]) / N  
 list\_bi.append(round(S, 5))  
  
  
 Disp\_list = [0, 0, 0, 0, 0, 0, 0, 0]  
  
 Y\_row1 = [y1[0], y2[0], y3[0]]  
 Y\_row2 = [y1[1], y2[1], y3[1]]  
 Y\_row3 = [y1[2], y2[2], y3[2]]  
 Y\_row4 = [y1[3], y2[3], y3[3]]  
 Y\_row5 = [y1[4], y2[4], y3[4]]  
 Y\_row6 = [y1[5], y2[5], y3[5]]  
 Y\_row7 = [y1[6], y2[6], y3[6]]  
 Y\_row8 = [y1[7], y2[7], y3[7]]  
  
 for i in range(m):  
 Disp\_list[0] += ((Y\_row1[i] - np.average(Y\_row1)) \*\* 2) / m  
 Disp\_list[1] += ((Y\_row2[i] - np.average(Y\_row2)) \*\* 2) / m  
 Disp\_list[2] += ((Y\_row3[i] - np.average(Y\_row3)) \*\* 2) / m  
 Disp\_list[3] += ((Y\_row4[i] - np.average(Y\_row4)) \*\* 2) / m  
 Disp\_list[4] += ((Y\_row5[i] - np.average(Y\_row5)) \*\* 2) / m  
 Disp\_list[5] += ((Y\_row6[i] - np.average(Y\_row6)) \*\* 2) / m  
 Disp\_list[6] += ((Y\_row7[i] - np.average(Y\_row7)) \*\* 2) / m  
 Disp\_list[7] += ((Y\_row8[i] - np.average(Y\_row8)) \*\* 2) / m  
  
 sum\_dispersion = sum(Disp\_list)  
  
  
 print("|ŀx0 |Łx1 |łx2 |Ńx3 |Łx1x2 |Łx1x3 |łx2x3 |Łx1x2x3 | y1 | y2 | y3 | y | s^2 |\n"  
 "|x0 = ", x0\_factor, "\n"  
 "|x1 = ", x1\_factor, "\n"  
 "|x2 = ", x2\_factor, "\n"  
 "|x3 = ", x3\_factor, "\n"  
 "|x1x2 = ", x1x2\_factor, "\n"  
 "|x1x3 = ", x1x3\_factor, "\n"  
 "|x2x3 = ", x2x3\_factor, "\n"  
 "|x1x2x3 = ", x1x2x3\_factor, "\n"  
 "|y1 = ", y1, "\n"  
 "|y2 = ", y2, "\n"  
 "|y3 = ", y3, "\n"  
 "|y = ", Y\_average, "\n"  
 "|s^2 = ", Disp\_list, "\n")  
  
 # рівняння регресії з ефектом взаємодії  
 print("y = {} + {}\*x1 + {}\*x2 + {}\*x3 + {}\*x1x2 + {}\*x1x3 + {}\*x2x3 + {}\*x1x2x3 \n".format(list\_bi[0], list\_bi[1],  
 list\_bi[2], list\_bi[3],  
 list\_bi[4], list\_bi[5],  
 list\_bi[6], list\_bi[7]))  
  
 print("###################################################################################################################")  
  
 print("|ŀx0 |Łx1 |łx2 |Ńx3 |Łx1x2 |Łx1x3 |łx2x3 |Łx1x2x3 | y1 | y2 | y3 | y | s^2 |\n"  
 "|x0 = ", x0, "\n"  
 "|x1 = ", x1, "\n"  
 "|x2 = ", x2, "\n"  
 "|x3 = ", x3, "\n"  
 "|x1x2 = ", x1x2,  
 "\n"  
 "|x1x3 = ", x1x3, "\n"  
 "|x2x3 = ", x2x3, "\n"  
 "|x1x2x3 = ", x1x2x3, "\n"  
 "|y1 = ", y1, "\n"  
 "|y2 = ", y2,  
 "\n"  
 "|y3 = ", y3, "\n"  
 "|y = ", Y\_average, "\n"  
 "|s^2 = ", Disp\_list, "\n")  
  
 list\_ai = [round(i, 5) for i in solve(list\_for\_solve\_a, Y\_average)]  
 print("y = {} + {}\*x1 + {}\*x2 + {}\*x3 + {}\*x1x2 + {}\*x1x3 + {}\*x2x3 + {}\*x1x2x3".format(list\_ai[0], list\_ai[1],  
 list\_ai[2], list\_ai[3],  
 list\_ai[4], list\_ai[5],  
 list\_ai[6], list\_ai[7]))  
  
 print("###################################################################################################################")  
  
 Gp = max(Disp\_list) / sum\_dispersion  
 F1 = m - 1  
 N = len(y1)  
 F2 = N  
 q1 = 0.05 / F1  
 fisher\_value = f.ppf(q=1 - q1, dfn=F2, dfd=(F1 - 1) \* F2)  
 Gt = fisher\_value / (fisher\_value + F1 - 1)  
 print("\nGp = ", Gp, " Gt = ", Gt)  
  
  
 if Gp < Gt:  
 print("\_\_\_\_\_Дисперсія однорідна!\_\_\_\_\_\n")  
  
 Dispersion\_B = sum\_dispersion / N  
 Dispersion\_beta = Dispersion\_B / (m \* N)  
 S\_beta = math.sqrt(abs(Dispersion\_beta))  
  
 beta\_list = [0, 0, 0, 0, 0, 0, 0, 0]  
 for i in range(len(x0\_factor)):  
 beta\_list[0] += (Y\_average[i] \*ŀx0\_factor[i]) / N  
 beta\_list[1] += (Y\_average[i] \*Łx1\_factor[i]) / N  
 beta\_list[2] += (Y\_average[i] \*łx2\_factor[i]) / N  
 beta\_list[3] += (Y\_average[i] \*Ńx3\_factor[i]) / N  
 beta\_list[4] += (Y\_average[i] \*Łx1x2\_factor[i]) / N  
 beta\_list[5] += (Y\_average[i] \*Łx1x3\_factor[i]) / N  
 beta\_list[6] += (Y\_average[i] \*łx2x3\_factor[i]) / N  
 beta\_list[7] += (Y\_average[i] \*Łx1x2x3\_factor[i]) / N  
 t\_list = [abs(beta\_list[i])/S\_beta for i in range(0, 8)]  
  
 F3 = F1 \* F2  
 d = 0  
 T = student\_teor(df=F3)  
 print("t\_табличне = ", T)  
  
 for i in range(len(t\_list)):  
 if T < t\_list[i]:  
 beta\_list[i] = 0  
 print("+++++ Гіпотеза підтверджена, beta{} = 0 +++++".format(i))  
 else:  
 print("----- Гіпотеза не підтверджена, beta{} = {} -----".format(i, beta\_list[i]))  
 d += 1  
  
 y\_1 = beta\_list[0] + beta\_list[1] \*Łx1[0] + beta\_list[2] \*łx2[0] + beta\_list[3] \*Ńx3[0] + beta\_list[4] \*Łx1x2[0] \  
 + beta\_list[5] \*Łx1x3[0] + beta\_list[6] \*łx2x3[0] + beta\_list[7] \*Łx1x2x3[0]  
 y\_2 = beta\_list[0] + beta\_list[1] \*Łx1[1] + beta\_list[2] \*łx2[1] + beta\_list[3] \*Ńx3[1] + beta\_list[4] \*Łx1x2[1] \  
 + beta\_list[5] \*Łx1x3[1] + beta\_list[6] \*łx2x3[1] + beta\_list[7] \*Łx1x2x3[1]  
 y\_3 = beta\_list[0] + beta\_list[1] \*Łx1[2] + beta\_list[2] \*łx2[2] + beta\_list[3] \*Ńx3[2] + beta\_list[4] \*Łx1x2[2] \  
 + beta\_list[5] \*Łx1x3[2] + beta\_list[6] \*łx2x3[2] + beta\_list[7] \*Łx1x2x3[2]  
 y\_4 = beta\_list[0] + beta\_list[1] \*Łx1[3] + beta\_list[2] \*łx2[3] + beta\_list[3] \*Ńx3[3] + beta\_list[4] \*Łx1x2[3] \  
 + beta\_list[5] \*Łx1x3[3] + beta\_list[6] \*łx2x3[3] + beta\_list[7] \*Łx1x2x3[3]  
 y\_5 = beta\_list[0] + beta\_list[1] \*Łx1[4] + beta\_list[2] \*łx2[4] + beta\_list[3] \*Ńx3[4] + beta\_list[4] \*Łx1x2[4] \  
 + beta\_list[5] \*Łx1x3[4] + beta\_list[6] \*łx2x3[4] + beta\_list[7] \*Łx1x2x3[4]  
 y\_6 = beta\_list[0] + beta\_list[1] \*Łx1[5] + beta\_list[2] \*łx2[5] + beta\_list[3] \*Ńx3[5] + beta\_list[4] \*Łx1x2[5] \  
 + beta\_list[5] \*Łx1x3[5] + beta\_list[6] \*łx2x3[5] + beta\_list[7] \*Łx1x2x3[5]  
 y\_7 = beta\_list[0] + beta\_list[1] \*Łx1[6] + beta\_list[2] \*łx2[6] + beta\_list[3] \*Ńx3[6] + beta\_list[4] \*Łx1x2[6] \  
 + beta\_list[5] \*Łx1x3[6] + beta\_list[6] \*łx2x3[6] + beta\_list[7] \*Łx1x2x3[6]  
 y\_8 = beta\_list[0] + beta\_list[1] \*Łx1[7] + beta\_list[2] \*łx2[7] + beta\_list[3] \*Ńx3[7] + beta\_list[4] \*Łx1x2[7] \  
 + beta\_list[5] \*Łx1x3[7] + beta\_list[6] \*łx2x3[7] + beta\_list[7] \*Łx1x2x3[7]  
  
 Y\_counted\_for\_Student = [y\_1, y\_2, y\_3, y\_4, y\_5, y\_6, y\_7, y\_8]  
 F4 = N - d  
 Dispersion\_ad = 0  
 for i in range(len(Y\_counted\_for\_Student)):  
 Dispersion\_ad += ((Y\_counted\_for\_Student[i] - Y\_average[i]) \*\* 2) \* m / (N - d)  
 Fp = Dispersion\_ad / Dispersion\_beta  
 Ft = fisher\_teor(dfn=F4, dfd=F3)  
  
  
 if Ft > Fp:  
 print("\_\_\_\_\_Рівняння регресії адекватне!\_\_\_\_\_")  
 break  
 else:  
 print("\_\_\_\_\_Рівняння регресії неадекватне\_\_\_\_\_")  
 break  
  
 else:  
 print("!!!!! Дисперсія неоднорідна. Збільшуємо m, m = m + 1. І спробуємо знову !!!!!")  
 m += 1

**Результат виконання коду:**

