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 наукових досліджень» на тему

«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ.»

Виконав:

студент ІІ курсу ФІОТ

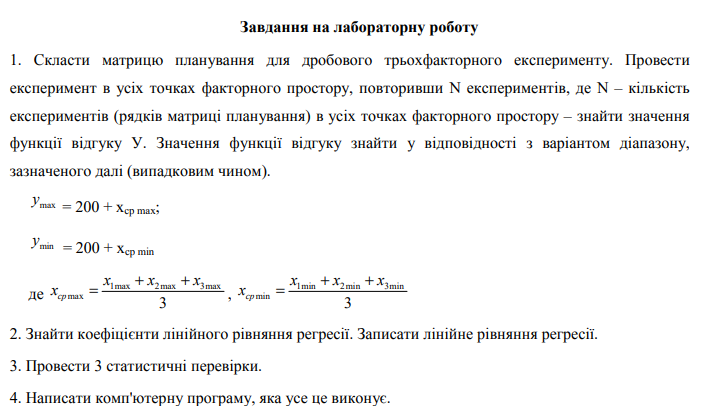
групи ІВ-93

Ільяш В. В.

ПЕРЕВІРИВ:

ас. Регіда П. Г.

Київ - 2021



**Варіант:**

Nваріант = 7

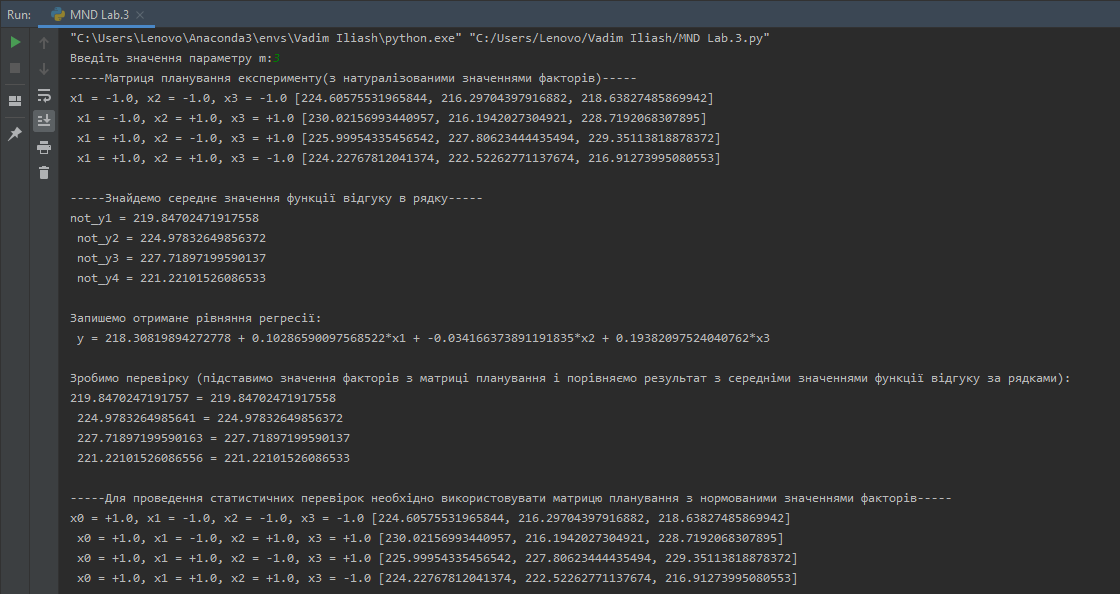


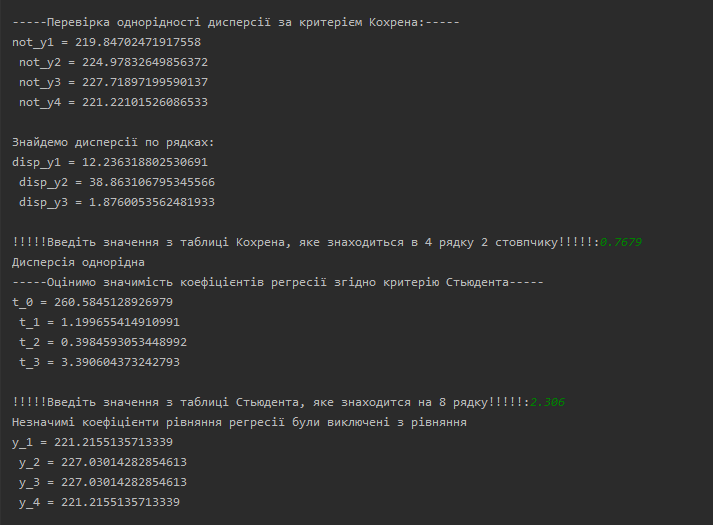


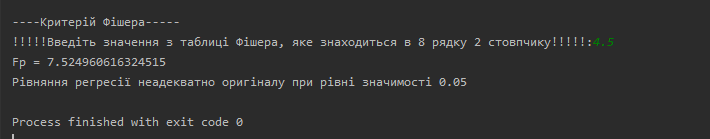
**Роздруківка коду програми:**

import random, math  
import numpy as np  
from numpy import linalg as LA  
  
N = 4  
  
m = int(input("Введіть значення параметру m:"))  
  
x\_1 = [-5, 15] # перше значення min, друге значення max  
x\_2 = [25, 45] # перше значення min, друге значення max  
x\_3 = [15, 45] # перше значення min, друге значення max  
  
y\_max = 200 + ((x\_1[1] + x\_2[1] + x\_3[1]) / 3)  
y\_min = 200 + ((x\_1[0] + x\_2[0] + x\_3[0]) / 3)  
  
  
print("-----Матриця планування експерименту(з натуралізованими значеннями факторів)-----")  
  
experement\_first = [] #Łx1 = -1.0,łx2 = -1.0,Ńx3 = -1.0  
experement\_second = [] #Łx1 = -1.0,łx2 = +1.0,Ńx3 = +1.0  
experement\_third = [] #Łx1 = +1.0,łx2 = -1.0,Ńx3 = +1.0  
experement\_fourth = [] #Łx1 = +1.0,łx2 = +1.0,Ńx3 = -1.0  
  
for l in range(m):  
 experement\_first.append(float(random.uniform(y\_min, y\_max)))  
 experement\_second.append(float(random.uniform(y\_min, y\_max)))  
 experement\_third.append(float(random.uniform(y\_min, y\_max)))  
 experement\_fourth.append(float(random.uniform(y\_min, y\_max)))  
  
print("x1 = -1.0,łx2 = -1.0,Ńx3 = -1.0", experement\_first, "\n", "x1 = -1.0,łx2 = +1.0,Ńx3 = +1.0", experement\_second,  
 "\n", "x1 = +1.0,łx2 = -1.0,Ńx3 = +1.0", experement\_third, "\n", "x1 = +1.0,łx2 = +1.0,Ńx3 = -1.0", experement\_fourth, "\n")  
  
  
print("-----Знайдемо середнє значення функції відгуку в рядку-----")  
not\_y1 = 0  
not\_y2 = 0  
not\_y3 = 0  
not\_y4 = 0  
  
for i in range(m):  
 not\_y1 += experement\_first[i] / m  
 not\_y2 += experement\_second[i] / m  
 not\_y3 += experement\_third[i] / m  
 not\_y4 += experement\_fourth[i] / m  
  
print("not\_y1 =", not\_y1, "\n", "not\_y2 =", not\_y2, "\n", "not\_y3 =", not\_y3, "\n", "not\_y4 =", not\_y4, "\n")  
  
mx1 = (x\_1[0] + x\_1[0] + x\_1[1] + x\_1[1]) / 4  
mx2 = (x\_2[0] + x\_2[1] + x\_2[0] + x\_2[1]) / 4  
mx3 = (x\_3[0] + x\_3[1] + x\_3[1] + x\_3[0]) / 4  
  
my = (not\_y1 + not\_y2 + not\_y3 + not\_y4) / 4  
  
a1 = (x\_1[0] \* not\_y1 + x\_1[0] \* not\_y2 + x\_1[1] \* not\_y3 + x\_1[1] \* not\_y4) / 4  
a2 = (x\_2[0] \* not\_y1 + x\_2[1] \* not\_y2 + x\_2[0] \* not\_y3 + x\_2[1] \* not\_y4) / 4  
a3 = (x\_3[0] \* not\_y1 + x\_3[1] \* not\_y2 + x\_3[1] \* not\_y3 + x\_3[0] \* not\_y4) / 4  
  
a11 = (2 \* (x\_1[0] \*\* 2) + 2 \* (x\_1[1] \*\* 2)) / 4  
a22 = (2 \* (x\_2[0] \*\* 2) + 2 \* (x\_2[1] \*\* 2)) / 4  
a33 = (2 \* (x\_3[0] \*\* 2) + 2 \* (x\_3[1] \*\* 2)) / 4  
  
a12 = (x\_1[0] \* x\_2[0] + x\_1[0] \* x\_2[1] + x\_1[1] \* x\_2[0] + x\_1[1] \* x\_2[1]) / 4  
a13 = (x\_1[0] \* x\_3[0] + x\_1[0] \* x\_3[1] + x\_1[1] \* x\_3[1] + x\_1[1] \* x\_3[0]) / 4  
a23 = (x\_2[0] \* x\_3[0] + x\_2[1] \* x\_3[1] + x\_2[0] \* x\_3[1] + x\_2[1] \* x\_3[0]) / 4  
  
b0 = LA.det([[my, mx1, mx2, mx3], [a1, a11, a12, a13], [a2, a12, a22, a23], [a3, a13, a23, a33]]) / LA.det([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a23], [mx3, a13, a23, a33]])  
b1 = LA.det([[1, my, mx2, mx3], [mx1, a1, a12, a13], [mx2, a2, a22, a23], [mx3, a3, a23, a33]]) / LA.det([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a23], [mx3, a13, a23, a33]])  
b2 = LA.det([[1, mx1, my, mx3], [mx1, a11, a1, a13], [mx2, a12, a2, a23], [mx3, a13, a3, a33]]) / LA.det([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a23], [mx3, a13, a23, a33]])  
b3 = LA.det([[1, mx1, mx2, my], [mx1, a11, a12, a1], [mx2, a12, a22, a2], [mx3, a13, a23, a3]]) / LA.det([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a23], [mx3, a13, a23, a33]])  
  
print("Запишемо отримане рівняння регресії:", "\n", "y = {0} + {1}\*x1 + {2}\*x2 + {3}\*x3".format(b0, b1, b2, b3), "\n")  
  
print("Зробимо перевірку (підставимо значення факторів з матриці планування і порівняємо результат з середніми значеннями функції відгуку за рядками): ")  
  
y1 = b0 + b1 \* x\_1[0] + b2 \* x\_2[0] + b3 \* x\_3[0]  
y2 = b0 + b1 \* x\_1[0] + b2 \* x\_2[1] + b3 \* x\_3[1]  
y3 = b0 + b1 \* x\_1[1] + b2 \* x\_2[0] + b3 \* x\_3[1]  
y4 = b0 + b1 \* x\_1[1] + b2 \* x\_2[1] + b3 \* x\_3[0]  
  
print(y1, "=", not\_y1, "\n", y2, "=", not\_y2, "\n", y3, "=", not\_y3, "\n", y4, "=", not\_y4, "\n")  
  
  
print("-----Для проведення статистичних перевірок необхідно використовувати матрицю планування з нормованими значеннями факторів-----")  
print("x0 = +1.0,Łx1 = -1.0,łx2 = -1.0,Ńx3 = -1.0", experement\_first, "\n", "x0 = +1.0,Łx1 = -1.0,łx2 = +1.0,Ńx3 = +1.0", experement\_second,  
 "\n", "x0 = +1.0,Łx1 = +1.0,łx2 = -1.0,Ńx3 = +1.0", experement\_third, "\n", "x0 = +1.0,Łx1 = +1.0,łx2 = +1.0,Ńx3 = -1.0", experement\_fourth, "\n")  
  
print("-----Перевірка однорідності дисперсії за критерієм Кохрена:-----")  
print("not\_y1 =", not\_y1, "\n", "not\_y2 =", not\_y2, "\n", "not\_y3 =", not\_y3, "\n", "not\_y4 =", not\_y4, "\n")  
  
print("Знайдемо дисперсії по рядках:")  
disp\_y1 = 0  
disp\_y2 = 0  
disp\_y3 = 0  
disp\_y4 = 0  
disp\_list = [disp\_y1, disp\_y2, disp\_y3, disp\_y4]  
  
for i in range(m):  
 disp\_y1 += ((experement\_first[i] - not\_y1) \*\* 2) / m  
 disp\_y2 += ((experement\_second[i] - not\_y2) \*\* 2) / m  
 disp\_y3 += ((experement\_third[i] - not\_y3) \*\* 2) / m  
 disp\_y4 += ((experement\_fourth[i] - not\_y4) \*\* 2) / m  
  
Gp = max(disp\_list) / (disp\_y1 + disp\_y2 + disp\_y3 + disp\_y4)  
print("disp\_y1 =", disp\_y1, "\n", "disp\_y2 =", disp\_y2, "\n", "disp\_y3 =", disp\_y3, "\n")  
  
f1 = m - 1  
f2 = N  
  
Gt = float(input("!!!!!Введіть значення з таблиці Кохрена, яке знаходиться в {0} рядку {1} стовпчику!!!!!:".format(f2, f1)))  
  
if Gp < Gt:  
 print("Дисперсія однорідна")  
 print("-----Оцінимо значимість коефіцієнтів регресії згідно критерію Стьюдента-----")  
 s\_2\_b = (disp\_y1 + disp\_y2 + disp\_y3 + disp\_y4) / N  
  
 s\_2\_b\_s = s\_2\_b / (N \* m)  
  
 s\_b\_s = math.sqrt(s\_2\_b\_s)  
  
 b\_0 = ((not\_y1 \* 1) + (not\_y2 \* 1) + (not\_y3 \* 1) + (not\_y4 \* 1)) / m  
 b\_1 = ((not\_y1 \* -1) + (not\_y2 \* -1) + (not\_y3 \* 1) + (not\_y4 \* 1)) / m  
 b\_2 = ((not\_y1 \* -1) + (not\_y2 \* 1) + (not\_y3 \* -1) + (not\_y4 \* 1)) / m  
 b\_3 = ((not\_y1 \* -1) + (not\_y2 \* 1) + (not\_y3 \* 1) + (not\_y4 \* -1)) / m  
  
 t\_0 = math.fabs(b\_0) / s\_b\_s  
 t\_1 = math.fabs(b\_1) / s\_b\_s  
 t\_2 = math.fabs(b\_2) / s\_b\_s  
 t\_3 = math.fabs(b\_3) / s\_b\_s  
  
 f3 = (m - 1) \* N  
 print("t\_0 =", t\_0, "\n", "t\_1 =", t\_1, "\n", "t\_2 =", t\_2, "\n", "t\_3 =", t\_3, "\n")  
 t\_tabl = float(input("!!!!!Введіть значення з таблиці Стьюдента, яке знаходится на {0} рядку!!!!!:".format(f3)))  
  
 y\_1 = 0  
 y\_2 = 0  
 y\_3 = 0  
 y\_4 = 0  
  
 t\_list = [t\_0, t\_1, t\_2, t\_3]  
 t\_list\_new = []  
 for i in range(len(t\_list)):  
 if t\_list[i] > t\_tabl:  
 t\_list\_new.append(t\_list[i])  
  
 for j in t\_list\_new:  
 n = t\_list.index(j)  
 if n == 0:  
 y\_1 += b0  
 y\_2 += b0  
 y\_3 += b0  
 y\_4 += b0  
 elif n == 1:  
 y\_1 += b1 \* x\_1[0]  
 y\_2 += b1 \* x\_1[0]  
 y\_3 += b1 \* x\_1[1]  
 y\_4 += b1 \* x\_1[1]  
 elif n == 2:  
 y\_1 += b2 \* x\_2[0]  
 y\_2 += b2 \* x\_2[1]  
 y\_3 += b2 \* x\_2[0]  
 y\_4 += b2 \* x\_2[1]  
 elif n == 3:  
 y\_1 += b3 \* x\_3[0]  
 y\_2 += b3 \* x\_3[1]  
 y\_3 += b3 \* x\_3[1]  
 y\_4 += b3 \* x\_3[0]  
  
 print("Незначимі коефіцієнти рівняння регресії були виключені з рівняння")  
 print("y\_1 =", y\_1, "\n", "y\_2 =", y\_2, "\n", "y\_3 =", y\_3, "\n", "y\_4 =", y\_4, "\n")  
  
 print("----Критерій Фішера-----")  
 d = len(t\_list\_new)  
 f4 = N - d  
  
 Ft = float(input("!!!!!Введіть значення з таблиці Фішера, яке знаходиться в {0} рядку {1} стовпчику!!!!!:".format(f3, f4)))  
  
 s\_2\_ad = (m / (N - d)) \* (((y\_1 - not\_y1) \*\* 2) + ((y\_2 - not\_y2) \*\* 2) + ((y\_3 - not\_y3) \*\* 2) + ((y\_4 - not\_y4) \*\* 2))  
 Fp = s\_2\_ad / s\_2\_b\_s  
 print("Fp =", Fp)  
 if Fp < Ft:  
 print("Рівняння регресії адекватно оригіналу при рівні значимості 0.05")  
 else:  
 print("Рівняння регресії неадекватно оригіналу при рівні значимості 0.05")  
  
else:  
 print("Дисперсія не однорідна")

**Результат виконання коду:**







**Відповіді на контрольні питання:**

1. *Що називається дробовим факторним експериментом?*

Дробовим факторним експериментом називається експеримент з використанням частини повного факторного експерименту.

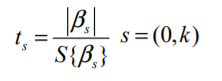
1. *Для чого потрібно розрахункове значення Кохрена?*

Розрахункове значення Кохрена використовують для перевірки однорідності дисперсій.

1. *Для чого перевіряється критерій Стьюдента?*

За допомогою критерію Стьюдента перевіряється значущість коефіцієнтів рівняння регресії.

Знайдені таким чином коефіцієнти рівняння регресії необхідно оцінити на статистичну значимість. Оцінка проводитися за t-критерієм Стьюдента. Для кожного коефіцієнта βs обчислюється коефіцієнт ts



Якщо виконується нерівність ts< tтабл, то приймається нуль-гіпотеза, тобто вважається, що знайдений коефіцієнт βs є статистично незначущим і його слід виключити з рівняння регресії. Якщо ts > tтабл то гіпотеза не підтверджується, тобто βs – значимий коефіцієнт і він залишається в рівнянні регресії.

1. *Чим визначається критерій Фішера і як його застосовувати?*

Критерій Фішера використовують при перевірці отриманого рівняння регресії досліджуваному об’єкту.

Отримане рівняння регресії необхідно перевірити на адекватність досліджуваному об'єкту. Для цієї мети необхідно оцінити, наскільки відрізняються середні значення у вихідної величини, отриманої в точках факторного простору, і значення у, отриманого з рівняння регресії в тих самих точках факторного простору. Для цього використовують дисперсію адекватності.