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 № 2

з дисципліни «Методи наукових досліджень» на тему

«ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконав:

студент ІІ курсу ФІОТ

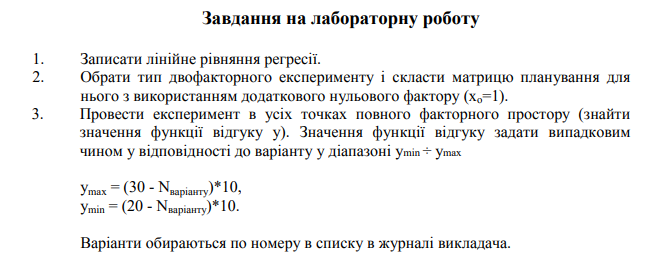
групи ІВ-93

Ільяш В. В.

ПЕРЕВІРИВ:

ас. Регіда П. Г.

Київ - 2021



**Варіант:**

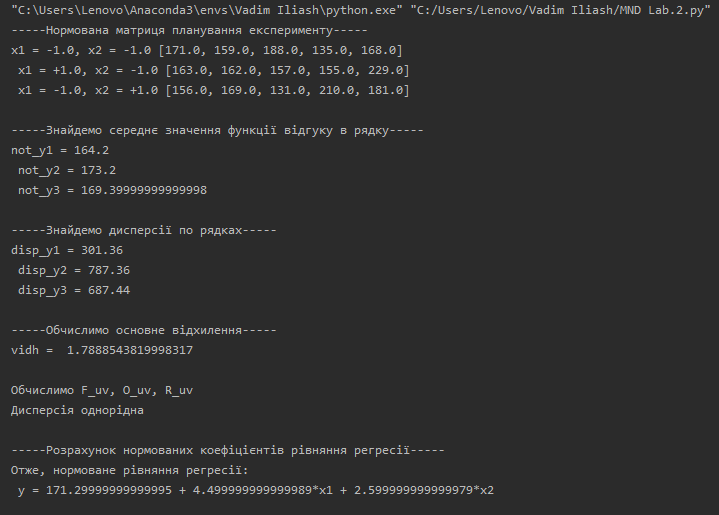
Nваріант = 7

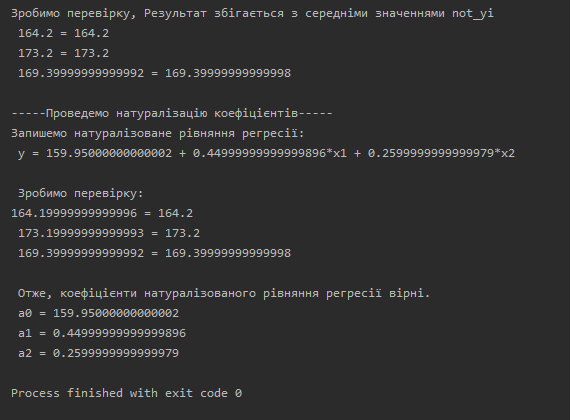


**Роздруківка коду програми:**

import random, math  
import numpy as np  
from numpy import linalg as LA  
  
print("-----Нормована матриця планування експерименту-----")  
y\_max = (30 - 7)\*10  
y\_min = (20 - 7)\*10  
  
x\_1 = [-5, 15] # перше значення min, друге значення max  
x\_2 = [25, 45] # перше значення min, друге значення max  
  
experement\_first = [] #Łx1 = -1.0,łx2 = -1.0  
experement\_second = [] #Łx1 = +1.0,łx2 = -1.0  
experement\_third = [] #Łx1 = -1.0,łx2 = +1.0  
  
  
for m in range(5):  
 experement\_first.append(float(random.randint(y\_min, y\_max)))  
 experement\_second.append(float(random.randint(y\_min, y\_max)))  
 experement\_third.append(float(random.randint(y\_min, y\_max)))  
  
print("x1 = -1.0,łx2 = -1.0", experement\_first, "\n", "x1 = +1.0,łx2 = -1.0", experement\_second,  
 "\n", "x1 = -1.0,łx2 = +1.0", experement\_third, "\n")  
  
print("-----Знайдемо середнє значення функції відгуку в рядку-----")  
not\_y1 = 0  
not\_y2 = 0  
not\_y3 = 0  
  
for i in range(5):  
 not\_y1 += experement\_first[i] / 5  
 not\_y2 += experement\_second[i] / 5  
 not\_y3 += experement\_third[i] / 5  
  
print("not\_y1 =", not\_y1, "\n", "not\_y2 =", not\_y2, "\n", "not\_y3 =", not\_y3, "\n")  
  
print("-----Знайдемо дисперсії по рядках-----")  
disp\_y1 = 0  
disp\_y2 = 0  
disp\_y3 = 0  
  
for i in range(5):  
 disp\_y1 += ((experement\_first[i] - not\_y1) \*\* 2) / 5  
 disp\_y2 += ((experement\_second[i] - not\_y2) \*\* 2) / 5  
 disp\_y3 += ((experement\_third[i] - not\_y3) \*\* 2) / 5  
  
print("disp\_y1 =", disp\_y1, "\n", "disp\_y2 =", disp\_y2, "\n", "disp\_y3 =", disp\_y3, "\n")  
  
print("-----Обчислимо основне відхилення-----")  
vidh = math.sqrt(2\*(2\*5-2)/5\*(5-4))  
print("vidh = ", vidh, "\n")  
  
print("Обчислимо F\_uv, O\_uv, R\_uv")  
F\_uv1 = disp\_y1 / disp\_y2  
F\_uv2 = disp\_y3 / disp\_y1  
F\_uv3 = disp\_y3 / disp\_y2  
  
O\_uv1 = (3 / 5) \* F\_uv1  
O\_uv2 = (3 / 5) \* F\_uv2  
O\_uv3 = (3 / 5) \* F\_uv3  
  
R\_uv1 = abs(O\_uv1 - 1) / vidh  
R\_uv2 = abs(O\_uv2 - 1) / vidh  
R\_uv3 = abs(O\_uv3 - 1) / vidh  
  
if R\_uv1 < 2 and R\_uv2 < 2 and R\_uv3 < 2:  
 print("Дисперсія однорідна", "\n")  
  
print("-----Розрахунок нормованих коефіцієнтів рівняння регресії-----")  
X1 = [-1.0, 1.0, -1.0]  
X2 = [-1.0, -1.0, 1.0]  
  
mx1 = (X1[0] +ŁX1[1] +ŁX1[2]) / 3  
mx2 = (X2[0] +łX2[1] +łX2[2]) / 3  
  
my = (not\_y1 + not\_y2 + not\_y3) / 3  
  
a1 = ((X1[0] \*\* 2) + (X1[1] \*\* 2) + (X1[2] \*\* 2)) / 3  
a2 = ((X1[0] \*łX2[0]) + (X1[1] \*łX2[1]) + (X1[2] \*łX2[2])) / 3  
a3 = ((X2[0] \*\* 2) + (X2[1] \*\* 2) + (X2[2] \*\* 2)) / 3  
  
a11 = ((X1[0] \* not\_y1) + (X1[1] \* not\_y2) + (X1[2] \* not\_y3)) / 3  
a22 = ((X2[0] \* not\_y1) + (X2[1] \* not\_y2) + (X2[2] \* not\_y3)) / 3  
  
b0 = LA.det([[my, mx1, mx2], [a11, a1, a2], [a22, a2, a3]]) / LA.det([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])  
b1 = LA.det([[1, my, mx2], [mx1, a11, a2], [mx2, a22, a3]]) / LA.det([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])  
b2 = LA.det([[1, mx1, my], [mx1, a1, a11], [mx2, a2, a22]]) / LA.det([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])  
  
print("Отже, нормоване рівняння регресії:", "\n", "y = {0} + {1}\*x1 + {2}\*x2".format(b0, b1, b2), "\n")  
  
  
y1 = b0 + b1\*X1[0] + b2\*X2[0]  
y2 = b0 + b1\*X1[1] + b2\*X2[1]  
y3 = b0 + b1\*X1[2] + b2\*X2[2]  
print("Зробимо перевірку, Результат збігається з середніми значеннями not\_yi", "\n",  
 y1, "=", not\_y1, "\n", y2, "=", not\_y2, "\n", y3, "=", not\_y3, "\n")  
  
print("-----Проведемо натуралізацію коефіцієнтів-----")  
delta\_x1 = abs(x\_1[1] - x\_1[0]) / 2  
delta\_x2 = abs(x\_2[1] - x\_2[0]) / 2  
  
x10 = (x\_1[1] + x\_1[0]) / 2  
x20 = (x\_2[1] + x\_2[0]) / 2  
  
a\_0 = b0 - b1\*(x10 / delta\_x1) - b2\*(x20 / delta\_x2)  
a\_1 = b1 / delta\_x1  
a\_2 = b2 / delta\_x2  
  
print("Запишемо натуралізоване рівняння регресії:", "\n", "y = {0} + {1}\*x1 + {2}\*x2".format(a\_0, a\_1, a\_2), "\n", "\n", "Зробимо перевірку:")  
y\_1 = a\_0 + a\_1\*x\_1[0] + a\_2\*x\_2[0]  
y\_2 = a\_0 + a\_1\*x\_1[1] + a\_2\*x\_2[0]  
y\_3 = a\_0 + a\_1\*x\_1[0] + a\_2\*x\_2[1]  
print(y\_1, "=", not\_y1, "\n", y\_2, "=", not\_y2, "\n", y\_3, "=", not\_y3, "\n", "\n",  
 "Отже, коефіцієнти натуралізованого рівняння регресії вірні.", "\n", "a0 =", a\_0, "\n", "a1 =", a\_1, "\n", "a2 =", a\_2)

**Результат виконання коду:**





**Відповіді на контрольні питання:**

1) Що таке регресійні поліноми і де вони застосовуються?

В теорії планування експерименту найважливішою частиною є оцінка результатів вимірів. При цьому використовують апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. В ТПЕ ці поліноми отримали спеціальну назву - регресійні поліноми, а їх знаходження та аналіз - регресійний аналіз.

2) Визначення однорідності дисперсії.

Однорідність дисперсії означає, що серед усіх дисперсій нема таких, які б значно перевищували одна одну.

3) Що називається повним факторним експериментом?

Повним факторним експериментом називається такий [експеримент](https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D0%B8%D0%BC%D0%B5%D0%BD%D1%82), при реалізації якого визначається значення параметра оптимізації при всіх можливих поєднаннях рівнів варіювання факторів.