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 №2

з дисципліни « Методи оптимізації та планування » на тему

«Проведення двофакторного експерименту

з використанням лінійного рівняння регресії»

Виконала:

студентка II курсу ФІОТ

групи ІВ – 92

Поморова Марія

Номер залікової книжки: ІВ - 9221

Перевірив:

ас. Регіда П.Г.

Київ – 2021

**Мета роботи:** провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.

**Завдання на лабораторну роботу:**

1. Записати лінійне рівняння регресії.

2. Обрати тип двофакторного експерименту і скласти матрицю планування для нього з використанням додаткового нульового фактору (хо=1).

3. Провести експеримент в усіх точках повного факторного простору (знайти значення функції відгуку y). Значення функції відгуку задати випадковим чином у відповідності до варіанту у діапазоні ymin ÷ ymax

4. Перевірити однорідності дисперсії за критерієм Романовського

5. Знайти коефіцієнти нормованих рівнянь регресії і виконати перевірку (підставити значення нормованих факторів і коефіцієнтів у рівняння).

6. Провести натуралізацію рівняння регресії й виконати перевірку натуралізованого рівняння.

7. Написати комп'ютерну програму, яка все це виконує.

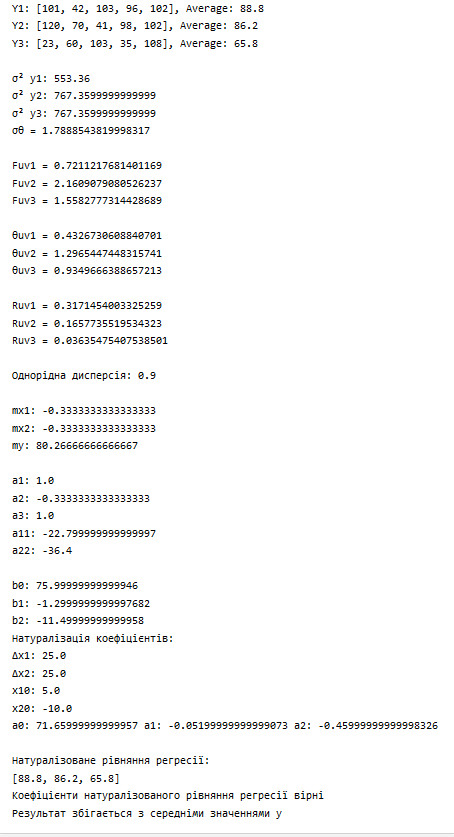
**Варіант завдання:**



**Роздруківка тексту програми:**

import math  
import numpy as np  
import random as rnd  
  
m = 5  
y\_min, y\_max = 20, 120  
  
x1\_min, x1\_max = -20, 30  
x2\_min, x2\_max = -35, 15  
x1\_min\_norm, x1\_max\_norm = -1, 1  
x2\_min\_norm, x2\_max\_norm = -1, 1  
  
p\_prob = (0.99, 0.98, 0.95, 0.90)  
rkr\_table = {2: (1.73, 1.72, 1.71, 1.69),  
 6: (2.16, 2.13, 2.10, 2.00),  
 8: (2.43, 4.37, 2.27, 2.17),  
 10: (2.62, 2.54, 2.41, 2.29),  
 12: (2.75, 2.66, 2.52, 2.39),  
 15: (2.9, 2.8, 2.64, 2.49),  
 20: (3.08, 2.96, 2.78, 2.62)}  
  
matrix\_of\_y = [[rnd.randint(y\_min, y\_max) for i in range(m)] for j in range(3)]  
average\_y = [sum(matrix\_of\_y[i][j] for j in range(m)) / m for i in range(3)]  
  
quadric\_sigma1 = sum([(j - average\_y[0]) \*\* 2 for j in matrix\_of\_y[0]]) / m  
quadric\_sigma2 = sum([(j - average\_y[1]) \*\* 2 for j in matrix\_of\_y[1]]) / m  
quadric\_sigma3 = sum([(j - average\_y[2]) \*\* 2 for j in matrix\_of\_y[2]]) / m  
  
teta\_sigma = math.sqrt((2 \* (2 \* m - 2)) / (m \* (m - 4)))  
  
Fuv1 = quadric\_sigma1 / quadric\_sigma2  
Fuv2 = quadric\_sigma3 / quadric\_sigma1  
Fuv3 = quadric\_sigma3 / quadric\_sigma2  
  
TetaUV1 = ((m - 2) / m) \* Fuv1  
TetaUV2 = ((m - 2) / m) \* Fuv2  
TetaUV3 = ((m - 2) / m) \* Fuv3  
  
Ruv1 = abs(TetaUV1 - 1) / teta\_sigma  
Ruv2 = abs(TetaUV2 - 1) / teta\_sigma  
Ruv3 = abs(TetaUV3 - 1) / teta\_sigma  
  
mx1 = (-1 + 1 - 1) / 3  
mx2 = (-1 - 1 + 1) / 3  
my = sum(average\_y) / 3  
a1 = (1 + 1 + 1) / 3  
a2 = (1 - 1 - 1) / 3  
a3 = (1 + 1 + 1) / 3  
a11 = (-1 \* average\_y[0] + 1 \* average\_y[1] - 1 \* average\_y[2]) / 3  
a22 = (-1 \* average\_y[0] - 1 \* average\_y[1] + 1 \* average\_y[2]) / 3  
  
b0 = np.linalg.det(np.dot([[my, mx1, mx2], [a11, a1, a2], [a22, a2, a3]],  
 np.linalg.inv([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])))  
  
b1 = np.linalg.det(np.dot([[1, my, mx2], [mx1, a11, a2], [mx2, a22, a3]],  
 np.linalg.inv([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])))  
  
b2 = np.linalg.det(np.dot([[1, mx1, my], [mx1, a1, a11], [mx2, a2, a22]],  
 np.linalg.inv([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])))  
  
  
def self\_dispersion():  
 M = 0  
 M = min(rkr\_table, key=lambda x: abs(x - M))  
 p = 0  
 for ruv in (Ruv1, Ruv2, Ruv3):  
 if ruv > rkr\_table[M][0]:  
 return False  
 for rkr in range(len(rkr\_table[M])):  
 if ruv < rkr\_table[M][rkr]:  
 p = rkr  
 return p\_prob[p]  
  
  
def regressionCheck():  
 y\_norm1 = round(b0 - b1 - b2, 2)  
 y\_norm2 = round(b0 + b1 - b2, 2)  
 y\_norm3 = round(b0 - b1 + b2, 2)  
  
 if y\_norm1 == average\_y[0] and y\_norm2 == average\_y[1] and y\_norm3 == average\_y[2]:  
 print(**"Результат збігається з середніми значеннями y"**)  
 else:  
 print(**"Результат НЕ збігається з середніми значеннями y"**)  
  
  
delta\_x1 = math.fabs(x1\_max - x1\_min) / 2  
delta\_x2 = math.fabs(x2\_max - x2\_min) / 2  
x10 = (x1\_max + x1\_min) / 2  
x20 = (x2\_max + x2\_min) / 2  
  
A0 = b0 - b1 \* x10 / delta\_x1 - b2 \* x20 / delta\_x2  
A1 = b1 / delta\_x1  
A2 = b2 / delta\_x2  
  
  
def naturalized\_regression(x1, x2):  
 return A0 + A1 \* x1 + A2 \* x2  
  
  
*# output*for i in range(3):  
 print(**"Y{}: {}, Average: {}"**.format(i + 1, matrix\_of\_y[i], average\_y[i]))  
print()  
print(**"σ² y1:"**, quadric\_sigma1, **"**\n**σ² y2:"**, quadric\_sigma2, **"**\n**σ² y3:"**, quadric\_sigma2)  
print(**"σθ ="**, teta\_sigma, **'**\n**'**)  
print(**"Fuv1 ="**, Fuv1, **"**\n**Fuv2 ="**, Fuv2, **"**\n**Fuv3 ="**, Fuv3, **'**\n**'**)  
print(**"θuv1 ="**, TetaUV1, **"**\n**θuv2 ="**, TetaUV2, **"**\n**θuv3 ="**, TetaUV3, **'**\n**'**)  
print(**"Ruv1 ="**, Ruv1, **"**\n**Ruv2 ="**, Ruv2, **"**\n**Ruv3 ="**, Ruv3, **'**\n**'**)  
D = self\_dispersion()  
if D is False:  
 print(**'Дисперсія не однорідна!'**)  
 m += 1  
else:  
 print(**"Однорідна дисперсія:"**, D, **'**\n**'**)  
print(**"mx1:"**, mx1, **"**\n**mx2:"**, mx2, **"**\n**my:"**, my, **'**\n**'**)  
print(**"a1:"**, a1, **"**\n**a2:"**, a2, **"**\n**a3:"**, a3)  
print(**"a11:"**, a11, **"**\n**a22:"**, a22, **'**\n**'**)  
print(**"b0:"**, b0, **"**\n**b1:"**, b1, **"**\n**b2:"**, b2)  
print(**"Натуралізація коефіцієнтів:"**)  
print(**"Δx1:"**, delta\_x1, **"**\n**Δx2:"**, delta\_x2)  
print(**"x10:"**, x10, **"**\n**x20:"**, x20)  
print(**"a0:"**, A0, **"a1:"**, A1, **"a2:"**, A2, **'**\n**'**)  
print(**"Натуралізоване рівняння регресії:"**)  
naturReg\_Y = [round(naturalized\_regression(x1\_min, x2\_min), 2),  
 round(naturalized\_regression(x1\_max, x2\_min), 2),  
 round(naturalized\_regression(x1\_min, x2\_max), 2)]  
print(naturReg\_Y)  
  
if naturReg\_Y == average\_y:  
 print(**"Коефіцієнти натуралізованого рівняння регресії вірні"**)  
else:  
 print(**"Коефіцієнти натуралізованого рівняння регресії НЕ вірні"**)  
  
regressionCheck()

**Результати роботи програми:**



**Відповіді на контрольні запитання:**

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.

1. **Визначення однорідності дисперсії.**

Обирають так названу «довірчу ймовірність» p – ймовірність, з якою вимагається підтвердити гіпотезу про однорідність дисперсій. У відповідності до p і кількості дослідів m обирають з таблиці критичне значення критерію . Кожне експериментальне значення Ruv критерію Романовського порівнюється з Rкр. (значення критерію Романовського за різних довірчих ймовірностей p) і якщо для усіх кожне Ruv < Rкр., то гіпотеза про однорідність дисперсій підтверджується з ймовірністю p.

1. **Що називається повним факторним експериментом?**

Для знаходження коефіцієнтів у лінійному рівнянні регресії застосовують повний факторний експеримент (ПФЕ). Якщо в багатофакторному експерименті використані всі можливі комбінації рівнів факторів, то такий експеримент називається повним факторним експериментом.

**Висновки:**

Під час виконання даної роботи було змодельовано двофакторний експеримент, перевірено однорідність дисперсії за критерієм Романовського, отримано коефіцієнти рівняння регресії та проведено натуралізацію рівняння регресії.