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 оптимізації та планування експерименту» з теми

«Проведення двофакторного експерименту з використанням лінійного рівняння регресії»

Виконала:

студентка ІІ курсу ФІОТ

групи ІВ-93

Трибунська Кароліна

Варіант: 326

Перевірив:

Регіда П. Г.

Київ 2021

**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 (х0=1).

3. Провести експеримент в усіх точках повного факторного простору (знайти

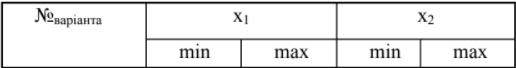
значення функції відгуку y). Значення функції відгуку задати випадковим

чином у відповідності до варіанту у діапазоні ymin ÷ ymax

ymax = (30 - Nваріанту)\*10,

ymin = (20 - Nваріанту)\*10.

Варіанти обираються по номеру в списку в журналі викладача.

****

****

**Код програми**

import random  
import math  
variant = 326  
m = 6  
y\_max = (30 - variant) \* 10  
y\_min = (20 - variant) \* 10  
  
x1\_min = -25  
x1\_max = -5  
x2\_min = -15  
x2\_max = 35  
  
xn = [[-1, -1], [-1, 1], [1, -1]]  
  
  
  
def average\_y(counting\_list):  
 average\_l\_y = []  
 for i in counting\_list:  
 average\_l\_y.append(sum(i)/len(i))  
 return average\_l\_y  
  
  
  
def dispersion(counting\_list):  
 d = []  
 for i in range(len(counting\_list)):  
 sum\_of\_y =0  
 for k in counting\_list[i]:  
 sum\_of\_y += (k - average\_y(counting\_list)[i]) \*\* 2  
 d.append(sum\_of\_y / len(counting\_list[i]))  
 return d  
  
  
def f\_uv(u, v):  
 if u >= v:  
 return u / v  
 else:  
 return v / u  
  
  
def determinant(x11, x12, x13, x21, x22, x23, x31, x32, x33):  
 det = x11 \* x22 \* x33 + x12 \* x23 \* x31 + x32 \* x21 \* x13 - x13 \* x22 \* x31 - x32 \* x23 \* x11 - x12 \* x21 \* x33  
 return det  
  
  
  
  
y = []  
for i in range(3):  
 y.append([random.randint(y\_min, y\_max),  
 random.randint(y\_min, y\_max),  
 random.randint(y\_min, y\_max),  
 random.randint(y\_min, y\_max),  
 random.randint(y\_min, y\_max),  
 random.randint(y\_min, y\_max)  
 ])  
  
print(y)  
  
avg\_y = average\_y(y)  
print(avg\_y)  
  
  
  
  
sigma\_tetta = math.sqrt((2 \* (2 \* m - 2)) / (m \* (m - 4)))  
  
fuv = []  
tetta = []  
ruv = []  
  
fuv.append(f\_uv(dispersion(y)[0], dispersion(y)[1]))  
fuv.append(f\_uv(dispersion(y)[2], dispersion(y)[0]))  
fuv.append(f\_uv(dispersion(y)[2], dispersion(y)[1]))  
  
tetta.append(((m - 2) / m) \* fuv[0])  
tetta.append(((m - 2) / m) \* fuv[1])  
tetta.append(((m - 2) / m) \* fuv[2])  
  
  
ruv.append(abs(tetta[0] - 1) / sigma\_tetta)  
ruv.append(abs(tetta[1] - 1) / sigma\_tetta)  
ruv.append(abs(tetta[2] - 1) / sigma\_tetta)  
  
r\_kr = 2  
  
  
for i in range(len(ruv)):  
 if ruv[i] > r\_kr:  
 print("Неоднорідна дисперсія")  
  
  
mx1 = (xn[0][0] + xn[1][0] + xn[2][0]) / 3  
mx2 = (xn[0][1] + xn[1][1] + xn[2][1]) / 3  
my = (avg\_y[0] + avg\_y[1] + avg\_y[2]) / 3  
  
  
a1 = (xn[0][0] \*\* 2 + xn[1][0] \*\* 2 + xn[2][0] \*\* 2) / 3  
a2 = (xn[0][0] \* xn[0][1] + xn[1][0] \* xn[1][1] + xn[2][0] \* xn[2][1]) / 3  
a3 = (xn[0][1] \*\* 2 + xn[1][1] \*\* 2 + xn[2][1] \*\* 2) / 3  
a11 = (xn[0][0] \* avg\_y[0] + xn[1][0] \* avg\_y[1] + xn[2][0] \* avg\_y[2]) / 3  
a22 = (xn[0][1] \* avg\_y[0] + xn[1][1] \* avg\_y[1] + xn[2][1] \* avg\_y[2]) / 3  
  
  
b0 = determinant(my, mx1, mx2, a11, a1, a2, a22, a2, a3) / determinant(1, mx1, mx2, mx1, a1, a2, mx2, a2, a3)  
b1 = determinant(1, my, mx2, mx1, a11, a2, mx2, a22, a3) / determinant(1, mx1, mx2, mx1, a1, a2, mx2, a2, a3)  
b2 = determinant(1, mx1, my, mx1, a1, a11, mx2, a2, a22) / determinant(1, mx1, mx2, mx1, a1, a2, mx2, a2, a3)  
  
y\_pr1 = b0 + b1 \* xn[0][0] + b2 \* xn[0][1]  
y\_pr2 = b0 + b1 \* xn[1][0] + b2 \* xn[1][1]  
y\_pr3 = b0 + b1 \* xn[2][0] + b2 \* xn[2][1]  
  
  
dx1 = abs(x1\_max - x1\_min) / 2  
dx2 = abs(x2\_max - x2\_min) / 2  
x10 = (x1\_max + x1\_min) / 2  
x20 = (x2\_max + x2\_min) / 2  
  
a\_0 = b0 - (b1 \* x10 / dx1) - (b2 \* x20 / dx2)  
a\_1 = b1 / dx1  
a\_2 = b2 / dx2  
  
yP1 = a\_0 + a\_1 \* x1\_min + a\_2 \* x2\_min  
yP2 = a\_0 + a\_1 \* x1\_max + a\_2 \* x2\_min  
yP3 = a\_0 + a\_1 \* x1\_min + a\_2 \* x2\_max  
  
print(f'Матриця планування при m = {m}')  
for i in range(3):  
 print(y[i])  
  
print('Експериментальні значення критерію Романовського:')  
for i in range(3):  
 print(ruv[i])  
  
print('Натуралізовані коефіцієнти: \na0 =', round(a\_0, 4), 'a1 =', round(a\_1, 4), 'a2 =', round(a\_2, 4))  
print('У практичний ', round(y\_pr1, 4), round(y\_pr2, 4), round(y\_pr3, 4),  
'\nУ середній', round(avg\_y[0], 4), round(avg\_y[1], 4), round(avg\_y[2], 4))  
print('У практичний норм.', round(yP1, 4), round(yP2, 4), round(yP3, 4))

**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 дисперсій – властивість, коли дисперсії вимірювання функцій відгуку є однаковими, або близькими.

**3) Що називається повним факторним експериментом?**

Однорідність дисперсій – властивість, коли дисперсії вимірювання функцій відгуку є однаковими, або близькими.