

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 планування експерименту»

на тему «ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАВ:
студент II курсу ФІОТ
групи ІВ-92
Чередник Віталій Юрійович
Номер у списку групи: 26

ПЕРЕВІРИВ:
ас. Регіда П.Г.

Хід роботи

Мета: 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 ($x_0=1$).
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 $y_{\min} \div y_{\max}$

$$y_{\max} = (30 - N_{\text{варіанту}}) * 10,$$

$$y_{\min} = (20 - N_{\text{варіанту}}) * 10.$$

Варіанти обираються по номеру в списку в журналі викладача.

226	10	60	-25	10
-----	----	----	-----	----

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.

Код програми

```
import random
import numpy as np
import math

var = 226
m = 5

ymax = (30 - var) * 10
ymin = (20 - var) * 10
x1min = 10
x1max = 60
x2min = -25
x2max = 10

xn = [[-1, -1],
      [1, -1],
```

```

    [-1, 1]]

Y = [[random.randint(ymin, ymax) for i in range(5)] for j in range(3)]
print("Матриця планування (m=5):")
for i in range(3):
    print(Y[i])

Y_average = []
for i in range(len(Y)):
    Y_average.append(np.mean(Y[i], axis=0))

dispersions = []
for i in range(len(Y)):
    sum = 0
    for k in Y[i]:
        sum += (k - np.mean(Y[i], axis=0)) ** 2
    dispersions.append(sum / len(Y[i]))
print("Дисперсії:", dispersions)

def determinant(x11, x12, x13, x21, x22, x23, x31, x32, x33):
    deter = x11 * x22 * x33 + x12 * x23 * x31 + x32 * x21 * x13 - x13 * x22 * x31 -
    x32 * x23 * x11 - x12 * x21 * x33
    return deter

sigma_teta = math.sqrt((2 * (2 * m - 2)) / (m * (m - 4)))

fuv = []
fuv.append(max(dispersions[0], dispersions[1]) / min(dispersions[0], dispersions[1]))
fuv.append(max(dispersions[2], dispersions[0]) / min(dispersions[2], dispersions[0]))
fuv.append(max(dispersions[2], dispersions[1]) / min(dispersions[2], dispersions[1]))

teta = []
teta.append(((m - 2) / m) * fuv[0])
teta.append(((m - 2) / m) * fuv[1])
teta.append(((m - 2) / m) * fuv[2])

ruv = []
ruv.append(abs(teta[0] - 1) / sigma_teta)
ruv.append(abs(teta[1] - 1) / sigma_teta)
ruv.append(abs(teta[2] - 1) / sigma_teta)
print("Експериментальні значення критерію Романовського:")
for i in range(3):
    print(ruv[i])

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 = (Y_average[0] + Y_average[1] + Y_average[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] * Y_average[0] + xn[1][0] * Y_average[1] + xn[2][0] * Y_average[2]) /
3
a22 = (xn[0][1] * Y_average[0] + xn[1][1] * Y_average[1] + xn[2][1] * Y_average[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)

yNorm1 = b0 + b1 * xn[0][0] + b2 * xn[0][1]
yNorm2 = b0 + b1 * xn[1][0] + b2 * xn[1][1]
yNorm3 = b0 + b1 * xn[2][0] + b2 * xn[2][1]

dx1 = abs(x1max - x1min) / 2
dx2 = abs(x2max - x2min) / 2
x10 = (x1max + x1min) / 2
x20 = (x2max + x2min) / 2

a_0 = b0 - (b1 * x10 / dx1) - (b2 * x20 / dx2)
a_1 = b1 / dx1
a_2 = b2 / dx2

yNat1 = a_0 + a_1 * x1min + a_2 * x2min
yNat2 = a_0 + a_1 * x1max + a_2 * x2min
yNat3 = a_0 + a_1 * x1min + a_2 * x2max

print("Середні значення:", Y_average[0], Y_average[1], Y_average[2])
print("Значення з нормованими коефіцієнтами:", round(yNorm1, 4), round(yNorm2, 4),
round(yNorm3, 4))
print("Значення з натуралізованими коефіцієнтами:", round(yNat1, 4), round(yNat2, 4),
round(yNat3, 4))

```

Відповіді на контрольні запитання

1. Що таке регресійні поліноми і де вони застосовуються?

Регресійні поліноми – це апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. Вони використовуються в ТПЕ для оцінки результатів вимірів.

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?

Якщо в багатофакторному експерименті використані всі можливі комбінації рівнів факторів, то такий експеримент називається повним факторним експериментом.

Нормована матриця планування

```
xn = [[-1, -1],  
      [1, -1],  
      [-1, 1]]
```

Результати роботи програми

```
"C:\Program Files\Anaconda\python.exe" C:/Users/Компик/PycharmProjects/MND/lab2.py
```

Матриця планування (m=5):

[-2009, -2058, -2030, -2025, -1993]

[-2019, -2045, -2046, -2012, -1970]

[-2004, -2028, -1997, -1985, -2047]

Дисперсії: [474.8, 770.64, 499.76000000000005]

Експериментальні значення критерію Романовського:

0.014618270855432535

0.20597445042950888

0.04180770606242616

Середні значення: -2023.0 -2018.4 -2012.2

Значення з нормованими коефіцієнтами: -2023.0 -2018.4 -2012.2

Значення з натуралізованими коефіцієнтами: -2023.0 -2018.4 -2012.2

Process finished with exit code 0