Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 2

з дисципліни «Методи наукових досліджень»

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

ВИКОНАЛА:

студентка 2 курсу

групи ІВ-91

Яременко В.Д.

Залікова – 9132

ПЕРЕВІРИВ:

ас. Регіда П. Г.

Київ – 2021

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

Завдання:

1. Записати лінійне рівняння регресії.
2. Обрати тип двофакторного експерименту і скласти матрицю планування для нього з використанням додаткового нульового фактору (xo=1).
3. Провести експеримент в усіх точках повного факторного простору (знайти значення функції відгуку у). Значення функції відгуку задати випадковим чином у відповідності до варіанту у діапазоні $y_{min} \div y_{max}$

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

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

Варіанти обираються по номеру в списку в журналі викладача.

№ варіанту	x_1		x_2	
	min	max	min	max
130	10	50	20	60

Програмний код:

```
from random import randint
from math import sqrt

print("""
Лабораторна робота 2 з МОПЕ
Варіант: 130
""")


variant = 30
m = 6
y_max = (30 - variant) * 10
y_min = (20 - variant) * 10
x1_min, x1_max, x2_min, x2_max = 10, 50, 20, 60
xn = [[-1, -1], [1, -1], [-1, 1]]


def average_y(lst):
    av_y = []
    for i in range(len(lst)):
        s = 0
        for j in lst[i]:
            s += j
        av_y.append(s / len(lst[i]))
    return av_y


def dispersion(lst):
```

```
    disp = []
    for i in range(len(lst)):
        s = 0
        for j in lst[i]:
            s += (j - average_y(lst)[i]) * (j - average_y(lst)[i])
        disp.append(s / len(lst[i]))
    return disp


def fuv(u, v):
    if u >= v:
        return u / v
    else:
        return v / u


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


y = [[randint(y_min, y_max) for j in range(6)] for i in range(3)]
av_y = average_y(y)
sigma_t = sqrt((2 * (2 * m - 2)) / (m * (m - 4)))
Fuv = []
t = []
Ruv = []

Fuv.append(fuv(dispersion(y)[0], dispersion(y)[1]))
Fuv.append(fuv(dispersion(y)[2], dispersion(y)[0]))
Fuv.append(fuv(dispersion(y)[2], dispersion(y)[1]))
t.append(((m - 2) / m) * Fuv[0])
t.append(((m - 2) / m) * Fuv[1])
t.append(((m - 2) / m) * Fuv[2])

Ruv.append(abs(t[0] - 1) / sigma_t)
Ruv.append(abs(t[1] - 1) / sigma_t)
Ruv.append(abs(t[2] - 1) / sigma_t)
Rkr = 2

for i in range(len(Ruv)):
    if Ruv[i] > Rkr:
        print('Помилка, повторіть експеримент')

mx1 = (xn[0][0] + xn[1][0] + xn[2][0]) / 3
mx2 = (xn[0][1] + xn[1][1] + xn[2][1]) / 3
my = (av_y[0] + av_y[1] + av_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] * av_y[0] + xn[1][0] * av_y[1] + xn[2][0] * av_y[2]) / 3
a22 = (xn[0][1] * av_y[0] + xn[1][1] * av_y[1] + xn[2][1] * av_y[2]) / 3

b0 = discriminant(my, mx1, mx2, a11, a1, a2, a22, a2, a3) / discriminant(1, mx1, mx2,
mx1, a1, a2, mx2, a2, a3)
b1 = discriminant(1, my, mx2, mx1, a11, a2, mx2, a22, a3) / discriminant(1, mx1, mx2,
mx1, a1, a2, mx2, a2, a3)
b2 = discriminant(1, mx1, my, mx1, a1, a11, mx2, a2, a22) / discriminant(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

koef0 = b0 - (b1 * x10 / dx1) - (b2 * x20 / dx2)
koef1 = b1 / dx1
koef2 = b2 / dx2

yP1 = koef0 + koef1 * x1_min + koef2 * x2_min
yP2 = koef0 + koef1 * x1_max + koef2 * x2_min
yP3 = koef0 + koef1 * x1_min + koef2 * x2_max

print('Матриця планування для m =', m)
print(y[0])
print(y[1])
print(y[2], "\n")

print('Експериментальні значення критерію Романовського:')
print(Ruv[0])
print(Ruv[1])
print(Ruv[2], "\n")

print('Натуралізовані коефіцієнти: \na0 =', round(koef0, 4), 'a1 =', round(koef1, 4),
'a2 =', round(koef2, 4), "\n")
print('У практичний ', round(y_pr1, 4), round(y_pr2, 4), round(y_pr3, 4),
      '\nУ середній', round(av_y[0], 4), round(av_y[1], 4), round(av_y[2], 4))
print('У практичний норм.', round(yP1, 4), round(yP2, 4), round(yP3, 4))
```

Результат роботи програми:

```
Лабораторна робота 2 з МОПЕ
Варіант: 130

Матриця планування для m = 6
[-62, -68, -67, -24, -37, -95]
[-81, -88, -15, -39, -59, -49]
[-6, -87, -38, -16, -95, -33]

Експериментальні значення критерію Романовського:
0.17417151930657201
0.3394743910520499
0.18356409700361986

Натуралізовані коефіцієнти:
a0 = -66.25 a1 = 0.0917 a2 = 0.325

У практичний  -58.8333 -55.1667 -45.8333
У середній -58.8333 -55.1667 -45.8333
У практичний норм. -58.8333 -55.1667 -45.8333
```

Висновок:

Під час виконання даної лабораторної роботи, провела двофакторний експеримент, перевірив однорідність дисперсії за критерієм Романовського, отримав коефіцієнти рівняння регресії, провів натуралізацію рівняння регресії. Зробивши перевірку я впевнилася в правильності коефіцієнтів Мета лабораторної роботи досягнута.

Відповіді на контрольні питання:

1. Що таке регресійні поліноми і де вони застосовуються?

Регресійні поліноми – це апроксимуючі поліноми, за допомогою яких ми можемо описати функцію. Застосовуються в теорії планування експерименту.

2. Визначення однорідності дисперсії.

Однорідність дисперсії означає, що серед усіх дисперсій немає такої, яка б значно перевищували інші.

3. Що називається повним факторним експериментом?

ПФЕ (Повний факторний експеримент) – називається такий експеримент, при реалізації якого визначається значення параметра оптимізації при всіх можливих поєднаннях рівнів варіювання факторів.