

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»
ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ

ВИКОНАВ:
Студент II курсу ФІОТ
Групи ІВ-93
Сержан Д. О.

ПЕРЕВІРИВ:
асистент
Регіда П.Г.

Київ 2021 р.

Мета:

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

Варіант завдання:

321	-30	20	-20	40
-----	-----	----	-----	----

$$Y_{\min} = (30 - 21) * 10 = 90$$

$$Y_{\max} = (20 - 21) * 10 = -10$$

Лістинг програми:

```
import random as rand
import math
variant = 21
m = 5
y_max = (30 - variant) * 10
y_min = (20 - variant) * 10
x1_min, x1_max, x2_min, x2_max = -30, 20, -20, 40
xn = [[-1, -1], [1, -1], [-1, 1]]
def averageY(list):
    avY = []
    for i in range(len(list)):
        s = 0
        for j in list[i]:
            s += j
        avY.append(s / len(list[i]))
    return avY
def dispersion(list):
    disp = []
    for i in range(len(list)):
        s = 0
        for j in list[i]:
            s += (j - averageY(list)[i]) * (j - averageY(list)[i])
        disp.append(s / len(list[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 = [[rand.randint(y_min, y_max) for j in range(6)] for i in range(3)]
avY = averageY(y)

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

Fuv = []
teta = []
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]))

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

Ruv.append(abs(teta[0] - 1) / sigmaTeta)
Ruv.append(abs(teta[1] - 1) / sigmaTeta)
Ruv.append(abs(teta[2] - 1) / sigmaTeta)

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 = (avY[0] + avY[1] + avY[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] * avY[0] + xn[1][0] * avY[1] + xn[2][0] * avY[2]) / 3
a22 = (xn[0][1] * avY[0] + xn[1][1] * avY[1] + xn[2][1] * avY[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)
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(koef0, 4), 'a1 =', round(koef1, 4),
'a2 =', round(koef2, 4))
print('У практичний ', round(y_pr1, 4), round(y_pr2, 4), round(y_pr3, 4),

```

```
'\nУ середній', round(avY[0], 4), round(avY[1], 4), round(avY[2], 4))  
print('У практичний норм.', round(yP1, 4), round(yP2, 4), round(yP3, 4))
```

Контрольні запитання:

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.

Опираючись на вимоги регресивного аналізу достовірно оброблення та використання вихідних даних експериментальних досліджень можливе лише тоді, коли дисперсії вимірювання функцій відгуку в кожній точці експерименту є однаковими. Дана властивість називається однорідністю дисперсії.

3. Що називається повним факторним експериментом?

ПФЕ – багатофакторний експеримент в якому використовуються всі можливі комбінації рівні факторів. $N_{\text{ПФЕ}} = 2^k$ або 3^k або 5^k .

Результат виконання роботи:

```
C:\Anaconda3\python.exe C:/Users/dan/.PyCharmCE2019.2/config/scratches/scratch_1.py
```

Матриця планування для $m = 5$

```
[-7, 76, -4, 80, 26, -10]
```

```
[11, -1, 23, 76, -2, 4]
```

```
[63, 17, 80, 85, 90, 50]
```

Експериментальні значення критерію Романовського:

```
0.10491986196917759
```

```
0.21291611858264442
```

```
0.1690488298960169
```

Натуралізовані коефіцієнти:

```
a0 = 34.2778 a1 = -0.1667 a2 = 0.6222
```

```
У практичний 26.8333 18.5 64.1667
```

```
У середній 26.8333 18.5 64.1667
```

```
У практичний норм. 26.8333 18.5 64.1667
```

```
Process finished with exit code 0
```

Висновок:

В даній лабораторній роботі я провів двофакторний експеримент з перевіркою дисперсій на однорідність за критерієм Романовського і отримав коефіцієнти рівняння регресії. Також провів натуралізацію рівняння регресії.