

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

«Методи оптимізації та планування експерименту»
Лабораторна робота №2

**«ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»**

Виконала:
студентка групи ІО-91
Тимошенко Діана
Варіант: 123
Перевірив Регіда П. Г.

Київ
2021 р.

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 y в діапазоні $y_{\min} \div y_{\max}$
 $y_{\max} = (30 - 23) \cdot 10 = 70$
 $y_{\min} = (20 - 23) \cdot 10 = -30$

№ _{варіанта}	X ₁		X ₂	
	min	max	min	max
123	-30	0	-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.

Роздруківка тексту програми:

```
from random import randint
from math import sqrt
import numpy as np

m = 5

# p = 0.95
Rkr_table = {2: 1.7,
              3: 1.87,
              4: 1.99,
              5: 2.1,
              6: 2.17,
              7: 2.24,
              8: 2.29,
              9: 2.34,
              10: 2.39}

x1_min = -30
x1_max = 0
x2_min = -25
x2_max = 10

y_min = -30
y_max = 70

x1 = [-1, -1, 1]
x2 = [-1, 1, -1]

flag = True

while flag:
    y = [[randint(y_min, y_max) for i in range(m)] for j in range(3)]

    y_aver = [sum(y[i])/m for i in range(3)]

    dispersion = []
    for i in range(3):
        z = 0
        for j in range(m):
            z += (y[i][j] - y_aver[i])**2
        dispersion.append(round(z/m, 3))

    sigma_teta = round(sqrt((2 * (2*m-2)) / (m * (m-4))), 3)

    Fuv1 = round(max(dispersion[0], dispersion[1])/min(dispersion[0], dispersion[1]),
3)
    Fuv2 = round(max(dispersion[0], dispersion[2])/min(dispersion[0], dispersion[2]),
3)
    Fuv3 = round(max(dispersion[1], dispersion[2])/min(dispersion[1], dispersion[2]),
3)

    teta_uv1 = round((m - 2)/m * Fuv1, 3)
    teta_uv2 = round((m - 2)/m * Fuv2, 3)
    teta_uv3 = round((m - 2)/m * Fuv3, 3)

    Ruv1 = round(abs(teta_uv1 - 1)/sigma_teta, 3)
    Ruv2 = round(abs(teta_uv2 - 1)/sigma_teta, 3)
    Ruv3 = round(abs(teta_uv3 - 1)/sigma_teta, 3)
```

```

Rkr = Rkr_table[m]
if Ruv1 < Rkr and Ruv2 < Rkr and Ruv3 < Rkr:
    flag = False
else:
    m += 1

print("m =", m)
print("x1", x1)
print("x2", x2)
print("y", y)
print("Середні значення y", y_aver)
print("Дисперсії", dispersion)
print("Основне відхилення", sigma_teta)
print("Fuv", [Fuv1, Fuv2, Fuv3])
print("teta_uv", [teta_uv1, teta_uv2, teta_uv3])
print("Ruv", [Ruv1, Ruv2, Ruv3])

#----- нормовані коефіцієнти -----

mx1 = round(sum(x1)/3, 3)
mx2 = round(sum(x2)/3, 3)
my = round(sum(y_aver)/3, 3)

a1 = (x1[0]**2 + x1[1]**2 + x1[2]**2)/3
a2 = 0
for i in range(3):
    a2 += x1[i] * x2[i]
a2 = round(a2/3, 3)
a3 = round((x2[0]**2 + x2[1]**2 + x2[2]**2)/3, 3)

a11 = 0
a22 = 0
for i in range(3):
    a11 += x1[i] * y_aver[i]
    a22 += x2[i] * y_aver[i]
a11 = round((a11/3), 3)
a22 = round((a22/3), 3)

matr_01 = np.array([[my, mx1, mx2], [a11, a1, a2], [a22, a2, a3]])
matr_02 = np.array([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])
matr_11 = np.array([[1, my, mx2], [mx1, a11, a2], [mx2, a22, a3]])
matr_21 = np.array([[1, mx1, my], [mx1, a1, a11], [mx2, a2, a22]])

b0 = np.linalg.det(matr_01)/np.linalg.det(matr_02)
b1 = np.linalg.det(matr_11)/np.linalg.det(matr_02)
b2 = np.linalg.det(matr_21)/np.linalg.det(matr_02)

print("\nНормоване рівняння регресії: y = {0:.2f} {1:+.2f}*x1 {2:+.2f}*x2".format(b0,
b1, b2))

print("Зробимо перевірку")
test1_y1 = round(b0 + b1*x1[0] + b2*x2[0], 3)
test1_y2 = round(b0 + b1*x1[1] + b2*x2[1], 3)
test1_y3 = round(b0 + b1*x1[2] + b2*x2[2], 3)

print("{0:.2f} {1:+.2f}*{4} {2:+.2f}*{5} = {3:.2f}".format(b0, b1, b2, test1_y1,
x1[0], x2[0]))
print("{0:.2f} {1:+.2f}*{4} {2:+.2f}*{5} = {3:.2f}".format(b0, b1, b2, test1_y2,
x1[1], x2[1]))
print("{0:.2f} {1:+.2f}*{4} {2:+.2f}*{5} = {3:.2f}".format(b0, b1, b2, test1_y3,
x1[2], x2[2]))

```

```
#----- натуралізація коефіцієнтів -----

delta_x1 = abs(x1_max - x1_min)/2
delta_x2 = abs(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

print("\nНатуралізоване рівняння регресії: y = {0:.2f} {1:+.2f}*x1
{2:+.2f}*x2".format(a0, a1, a2))

print("Зробимо перевірку")
test2_y1 = round(a0 + a1*x1_min + a2*x2_min, 3)
test2_y2 = round(a0 + a1*x1_min + a2*x2_max, 3)
test2_y3 = round(a0 + a1*x1_max + a2*x2_min, 3)

print("{0:.2f} {1:+.2f}*{4} {2:+.2f}*{5} = {3}".format(a0, a1, a2, test2_y1, x1_min,
x2_min))
print("{0:.2f} {1:+.2f}*{4} {2:+.2f}*{5} = {3}".format(a0, a1, a2, test2_y2, x1_min,
x2_max))
print("{0:.2f} {1:+.2f}*{4} {2:+.2f}*{5} = {3}".format(a0, a1, a2, test2_y3, x1_max,
x2_min))
```

Результат виконання програми:

```
m = 5
x1 [-1, -1, 1]
x2 [-1, 1, -1]
y [[-16, -8, 21, 46, 7], [69, 16, 18, 40, 1], [57, 61, 52, 41, 13]]
Середні значення y [10.0, 28.8, 44.8]
Дисперсії [485.2, 558.96, 297.76]
Основне відхилення 1.789
Fuv [1.152, 1.63, 1.877]
teta_uv [0.691, 0.978, 1.126]
Ruv [0.173, 0.012, 0.07]

Нормоване рівняння регресії: y = 36.76 +17.36*x1 +9.35*x2
Зробимо перевірку
36.76 +17.36*-1 +9.35*-1 = 10.05
36.76 +17.36*-1 +9.35*1 = 28.76
36.76 +17.36*1 +9.35*-1 = 44.76

Натуралізоване рівняння регресії: y = 58.13 +1.16*x1 +0.53*x2
Зробимо перевірку
58.13 +1.16*-30 +0.53*-25 = 10.051
58.13 +1.16*-30 +0.53*10 = 28.76
58.13 +1.16*0 +0.53*-25 = 44.764

Process finished with exit code 0
```

Відповіді на контрольні питання:

1. Що таке регресійні поліноми і де вони застосовуються?

Регресійні поліноми – це апроксимуючі поліноми, які використовуються в ТПЕ для знаходження функції, що описує даний об'єкт.

2. Визначення однорідності дисперсії.

В цій лабораторній роботі для оцінки однорідності дисперсії ми використовуємо критерій Романовського:

1. Для кожної комбінації знайти – дисперсію.

2. Обчислити основне відхилення:

$$\sigma_{\theta} = \sqrt{\frac{2(2m-2)}{m(m-4)}}$$

3. Для кожної пари комбінацій u, v ($u=\overline{1,N}; v=\overline{1,N}$) обчислити:

$$F_{uv} = \begin{cases} \frac{\sigma^2\{y_u\}}{\sigma^2\{y_v\}}, & \text{якщо } \sigma^2\{y_u\} \geq \sigma^2\{y_v\} \\ \frac{\sigma^2\{y_v\}}{\sigma^2\{y_u\}}, & \text{якщо } \sigma^2\{y_u\} < \sigma^2\{y_v\} \end{cases}$$

$$\theta_{uv} = \frac{m-2}{m} F_{uv}$$

$$R_{uv} = \frac{|\theta_{uv}-1|}{\sigma_{\theta}}$$

4. 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 $R_{кр}$.

5. Кожне R_{uv} (експериментальне значення критерію Романовського) порівнюється з $R_{кр}$ (значення критерію Романовського за різних довірчих ймовірностей p) і, якщо для усіх кожне $R_{uv} < R_{кр}$, то гіпотеза про однорідність дисперсій підтверджується з ймовірністю p .

Якщо хоча б для одної пари u, v має місце $R_{uv} > R_{кр}$, то гіпотеза про однорідність дисперсій не підтверджується. В цьому випадку розбіжність між дисперсіями експериментальних значень u -її v -ї комбінацій є значною. Необхідно збільшити кількість дослідів $m=m+1$, провести нові досліді і заново перевірити критерій.

3. Що називається повним факторним експериментом?

Повний факторний експеримент – експеримент, в якому використані всі можливі комбінації рівнів факторів.