

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

ВИКОНАВ:

студент II курсу ФІОТ

групи ІО-92

Соболь Денис

Варіант: 218

ПЕРЕВІРИВ:

Резіда П. Г.

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

Варіант

218	-20	30	-35	15
-----	-----	----	-----	----

Ymin=20

Ymax=120

Приклад роботи програми

```
from math import *
from random import *

while True:

    m = 5
    max_y = (30 - 18)*10
    min_y = (20 - 18)*10
    min_x_1 = -20
    max_x_1 = 30
    min_x_2 = -35
    max_x_2 = 15

    x_arr = [[-1, 1, -1], [-1, -1, 1]]
    y_arr = []
    for i in range(3):
        ex_arr=[]
        for j in range(m):
            ex_arr.append(randrange(min_y, max_y))
        y_arr.append(ex_arr)

    y_arr_avg = []
    for i in y_arr:
        y_arr_avg.append(sum(i)/len(i))

    sigma = []
    for i in range(3):
        sum_sigma = 0
        for j in range(m):
            sum_sigma += (y_arr[i][j] - y_arr_avg[i]) ** 2
        sigma.append(sum_sigma/m)

    main_sigma = sqrt(2 * (2 * m - 2) / (m * (m - 4)))

    func=[]
    for i in range(3):
        ex = [sigma[int((i + 3) / 2)], sigma[int(i / 2)]]
        func.append(max(ex) / min(ex))

    t=[]
    for i in func:
        t.append(i*(m-2)/m)

    ruv=[]
    for i in t:
        ruv.append(fabs(i-1)/main_sigma)
```

```

m_x1 = sum(x_arr[0]) / 3
m_x2 = sum(x_arr[1]) / 3
M = sum(y_arr_avg)/len(y_arr_avg)

a = [sum([x_arr[0][i]**2 for i in range(3)])/3,
      sum([x_arr[0][i]*x_arr[1][i] for i in range(3)])/3,
      sum([x_arr[1][i]**2 for i in range(3)]) / 3]

aij = [sum([x_arr[j][i] * y_arr_avg[i] for i in range(3)]) / 3 for j in
range(2)]

def matrix(matr):
    return (matr[0][0] * matr[1][1] * matr[2][2] + matr[0][1] *
matr[1][2] * matr[2][0] + matr[1][0] * matr[2][1] * matr[0][2])-\
(matr[0][2] * matr[1][1] * matr[2][0] + matr[0][1] *
matr[1][0] * matr[2][2] + matr[0][0] * matr[1][2] * matr[2][1])

bz = matrix([[1, m_x1, m_x2], [m_x1, a[0], a[1]], [m_x2, a[1], a[2]]])

b0ch = matrix([M, m_x1, m_x2], [aij[0], a[0], a[1]], [aij[1], a[1],
a[2]])
b1ch = matrix([[1, M, m_x2], [m_x1, aij[0], a[1]], [m_x2, aij[1], a[2]]])
b2ch = matrix([[1, m_x1, M], [m_x1, a[0], aij[0]], [m_x2, a[1], aij[1]]])

b0=b0ch/bz
b1=b1ch/bz
b2=b2ch/bz

delta_x1 = fabs(max_x_1 - min_x_1) / 2
delta_x2 = fabs(max_x_2 - min_x_2) / 2

x10 = (max_x_1 + min_x_1) / 2
x20 = (max_x_2 + min_x_2) / 2

a0 = b0 - b1*x10/delta_x1 - b2*x20/delta_x2
a1 = b1/delta_x1
a2 = b2/delta_x2

b11 = b0 - b1 - b2
b22 = b0 + b1 - b2
b33 = b0 - b1 + b2

a11 = a0 + a1*min_x_1 + a2*min_x_2
a22 = a0 + a1*max_x_1 + a2*min_x_2
a33 = a0 + a1*min_x_1 + a2*max_x_2
if ruv[0] < 2 and ruv[1] < 2 and ruv[2] < 2:break

print("Y(max) = ", round(max_y, 2), "\nY(min) = ", round(min_y, 2), "\n")

print("Y:", y_arr)
print("Y(cep):", y_arr_avg, "\n")

print("σ²:", [round(sigma[0], 2), round(sigma[1], 2), round(sigma[2], 2)],
"\n")

print("F(uv): ", func)
print("O(uv): ", t)
print("R(uv): ", ruv, "\n")

print("mx1 =", round(m_x1, 2), ", mx2 =", round(m_x2, 2), "\n")

```

```

print("a:", [round(a[0], 2), round(a[1], 2), round(a[2], 2)])
print("aij:", [round(aij[0], 2), round(aij[1], 2)])
print("b0 = ", round(b0, 2), ", b1 = ", round(b1, 2), ", b2 = ", round(b2, 2), "\n")

print("Нормоване рівняння регресії виглядає так: y = ", round(b0, 2), "+", round(b1, 2), "*x1 +", round(b2, 2), "*x2 \n")

print("Перевірка отриманих результатів:")
print(round(b0, 2), "-", round(b1, 2), "-", round(b2, 2), "=", round(b11, 2))
print(round(b0, 2), "+", round(b1, 2), "-", round(b2, 2), "=", round(b22, 2))
print(round(b0, 2), "-", round(b1, 2), "+", round(b2, 2), "=", round(b33, 2), "\n")

print("a0 = ", round(a0, 2), ", a1 = ", round(a1, 2), ", a2 = ", round(a2, 2), "\n")

print("Натуралізоване рівняння регресії: y = ", round(a0, 2), "+", round(a1, 2), "*x1 +", round(a2, 2), "*x2")
print(round(a0, 2), "+", round(a1, 2), "*", round(min_x_1, 2), "+", round(a2, 2), "*", round(min_x_2, 2), "=", round(a11, 2))
print(round(a0, 2), "+", round(a1, 2), "*", round(max_x_1, 2), "+", round(a2, 2), "*", round(min_x_2, 2), "=", round(a22, 2))
print(round(a0, 2), "+", round(a1, 2), "*", round(min_x_1, 2), "+", round(a2, 2), "*", round(max_x_2, 2), "=", round(a33, 2), "\n")

```

Результат роботи

```
Y(max) = 120
Y(min) = 20

Y: [[92, 22, 119, 83, 97], [35, 69, 116, 102, 101], [37, 90, 97, 48, 21]]
Y(сep): [82.6, 84.6, 58.6]

 $\sigma^2$ : [1058.64, 852.24, 890.64]

F(uv): [1.242185299915517, 1.1886284020479654, 1.0450577302168405]
O(uv): [0.7453111799493102, 0.7131770412287792, 0.6270346381301042]
R(uv): [0.1423753786856385, 0.1603389083300173, 0.2084939755984738]

mx1 = -0.33 , mx2 = -0.33

a: [1.0, -0.33, 1.0]
aij: [-18.87, -36.2]
b0 = 71.6 , b1 = 1.0 , b2 = -12.0

Нормоване рівняння регресії виглядає так:  $y = 71.6 + 1.0 * x_1 + -12.0 * x_2$ 

Перевірка отриманих результатів:
71.6 - 1.0 - -12.0 = 82.6
71.6 + 1.0 - -12.0 = 84.6
71.6 - 1.0 + -12.0 = 58.6

a0 = 66.6 , a1 = 0.04 , a2 = -0.48

Натуралізоване рівняння регресії:  $y = 66.6 + 0.04 * x_1 + -0.48 * x_2$ 
66.6 + 0.04 * -20 + -0.48 * -35 = 82.6
66.6 + 0.04 * 30 + -0.48 * -35 = 84.6
66.6 + 0.04 * -20 + -0.48 * 15 = 58.6
```

Висновок:

Провів двофакторний експеримент, перевіряв однорідність дисперсії за критерієм Романовського, отримав коефіцієнти рівняння регресії, провів натуралізацію рівняння регресії.

Контрольні запитання

1. Що таке регресійні поліноми і де вони застосовуються?

Регресійний поліном – це апроксимуючі поліноми, за допомогою яких ми можемо описати функцію. Це рівняння регресії виду

$$y = b_0 + \sum_{i=1}^k b_i x_i + \sum_{i,j=1}^k b_{ij} x_i x_j + \sum_{i=1}^k b_{ii} x_i^2 + \sum_{i,j,n=1}^k b_{ijn} x_i x_j x_n + \dots$$

Воно використовується в ТПЕ для оцінки результатів вимірів.

2. Визначення однорідності дисперсій.

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

3. Що називається повним факторним експериментом?

Повний факторний експеримент (ПФЕ) – експеримент, в якому використовуються всі можливі комбінації рівнів факторів.