

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

Методи планування експерименту

Лабораторна робота №2

«Проведення двофакторного експерименту з
використанням лінійного рівняння регресії»

Виконав:

студент II курсу ФІОТ

групи ІВ-92

Салун Кирило

номер у списку групи – 20

Перевірив:

ас. Регіда П. Г.

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

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

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

Варіанти обираються по номеру в списку в журналі викладача.

Варіант завдання:

220	-30	20	-30	45
-----	-----	----	-----	----

Лістинг програми:

```
import math
import random

import numpy
```

```

def regression_naturalization(x_1, x_2):
    return a_0 + a_1 * x_1 + a_2 * x_2

def uniform_dispersion():
    p = 0
    minimum = min(romanovsky_table, key=lambda x: abs(x - m))
    for ruv in (ruv1, ruv2, ruv3):
        if ruv > romanovsky_table[minimum][0]:
            return False
        for rkr in range(len(romanovsky_table[minimum])):
            if ruv < romanovsky_table[minimum][rkr]:
                p = rkr
    return p_list[p]

# Блок даних, заданих за варіантом 220
m = 5
x1 = [-1, 1, -1]
x2 = [-1, -1, 1]
y_min, y_max = 0, 100
x1_min, x1_max_normalized = -30, -1
x1_max, x1_max_normalized = 20, 1
x2_min, x2_min_normalized = -30, -1
x2_max, x2_max_normalized = 45, 1

p_list = (0.99, 0.98, 0.95, 0.90)
romanovsky_table = {2: (1.73, 1.72, 1.71, 1.69),
                     6: (2.16, 2.13, 2.10, 2.00),
                     8: (2.43, 4.37, 2.27, 2.17),
                     10: (2.62, 2.54, 2.41, 2.29),
                     12: (2.75, 2.66, 2.52, 2.39),
                     15: (2.9, 2.8, 2.64, 2.49),
                     20: (3.08, 2.96, 2.78, 2.62)}

y_matrix = [[random.randint(y_min, y_max) for _ in range(m)] for _ in range(3)] # достатньо
# провести 3 експеримента

y_average_value = [sum(y_matrix[i][j] for j in range(m)) / m for i in range(3)] # середнє
# значення функції відгуку

sigma = [sum([(element - y_average_value[i]) ** 2 for element in y_matrix[i]]) / m for i in
range(3)] # Пошук дисперсій

sigma_theta = math.sqrt((2 * (2 * m - 2)) / (m * (m - 4))) # основне відхилення

def f_uv_calc(u,v):
    if sigma[u] >= sigma[v]:
        return round(sigma[u] / sigma[v], 3)
    else:
        return round(sigma[v] / sigma[u], 3)

fuv1 = f_uv_calc(0,1)
fuv2 = f_uv_calc(0,2)
fuv3 = f_uv_calc(1,2)

theta_uv1 = ((m - 2) / m) * fuv1
theta_uv2 = ((m - 2) / m) * fuv2
theta_uv3 = ((m - 2) / m) * fuv3

ruv1 = abs(theta_uv1 - 1) / sigma_theta
ruv2 = abs(theta_uv2 - 1) / sigma_theta

```

```

ruv3 = abs(theta_uv3 - 1) / sigma_theta

# Розрахунок нормованих коефіцієнтів
mx1 = sum(x1) / 3
mx2 = sum(x2) / 3
m_y = sum(y_average_value) / 3
a1 = sum([element ** 2 for element in x1]) / 3
a2 = sum([x1[i] * x2[i] for i in range(3)]) / 3
a3 = sum([element ** 2 for element in x2]) / 3
a11 = sum([x1[i] * y_average_value[i] for i in range(3)]) / 3
a22 = sum([x2[i] * y_average_value[i] for i in range(3)]) / 3

denominator_determinant = numpy.linalg.det([[1, mx1, mx2],
                                             [mx1, a1, a2],
                                             [mx2, a2, a3]])

b0 = numpy.linalg.det([[m_y, mx1, mx2],
                       [a11, a1, a2],
                       [a22, a2, a3]]) / denominator_determinant
b1 = numpy.linalg.det([[1, m_y, mx2],
                       [mx1, a11, a2],
                       [mx2, a22, a3]]) / denominator_determinant
b2 = numpy.linalg.det([[1, mx1, m_y],
                       [mx1, a1, a11],
                       [mx2, a2, a22]]) / denominator_determinant

# Натуралізація коефіцієнтів
delta_x1 = math.fabs(x1_max - x1_min) / 2
delta_x2 = math.fabs(x2_max - x2_min) / 2
x10 = (x1_max + x1_min) / 2
x20 = (x2_max + x2_min) / 2
a_0 = b0 - b1 * x10 / delta_x1 - b2 * x20 / delta_x2
a_1 = b1 / delta_x1
a_2 = b2 / delta_x2

equation_coefficients = [round(regression_naturalization(x1_min, x2_min), 2),
                          round(regression_naturalization(x1_max, x2_min), 2),
                          round(regression_naturalization(x1_min, x2_max), 2)]

# Вивід результуючих даних
for i in range(3):
    print(f"y{i + 1} = {y_matrix[i]}; середнє значення = {y_average_value[i]}")
print(f"σ²(y1) = {sigma[0]}")
print(f"σ²(y2) = {sigma[1]}")
print(f"σ²(y3) = {sigma[2]}")
print(f"σ(θ) = {sigma_theta}")
print(f"Fuv1 = {fuv1}")
print(f"Fuv2 = {fuv2}")
print(f"Fuv3 = {fuv3}")
print(f"θuv1 = {theta_uv1}")
print(f"θuv2 = {theta_uv2}")
print(f"θuv3 = {theta_uv3}")
print(f"Ruv1 = {ruv1}")
print(f"Ruv2 = {ruv2}")
print(f"Ruv3 = {ruv3}")
print(f"Однорідна дисперсія = {uniform_dispersion()}")
print(f"mx1 = {mx1}")
print(f"mx2 = {mx2}")
print(f"my = {m_y}")
print(f"a1 = {a1}")
print(f"a2 = {a2}")
print(f"a3 = {a3}")
print(f"a11 = {a11}")
print(f"a22 = {a22}")
print(f"b0 = {b0}")

```

```
print(f"b1 = {b1}")
print(f"b2 = {b2}")

print("Натуралізація коефіцієнтів:")
print(f" $\Delta x_1$  = {delta_x1}")
print(f" $\Delta x_2$  = {delta_x2}")
print(f"x10 = {x10}")
print(f"x20 = {x20}")
print(f"a0 = {a_0}")
print(f"a1 = {a_1}")
print(f"a2 = {a_2}")

print(f"Натуралізоване рівняння регресії = {equation_coefficients}")
print("Коефіцієнти натуралізованого рівняння розраховано коректно.") if equation_coefficients
== y_average_value \
    else print("Коефіцієнти натуралізованого рівняння розраховано не правильно!")
```

Результати виконання роботи:

$y_1 = [3, 7, 35, 21, 39]$; середнє значення = 21.0
 $y_2 = [67, 28, 31, 28, 74]$; середнє значення = 45.6
 $y_3 = [28, 81, 49, 37, 9]$; середнє значення = 40.8
 $\sigma^2(y_1) = 208.0$
 $\sigma^2(y_2) = 419.43999999999994$
 $\sigma^2(y_3) = 574.56000000000001$
 $\sigma(\theta) = 1.7888543819998317$
 $F_{uv1} = 2.017$
 $F_{uv2} = 2.762$
 $F_{uv3} = 1.37$
 $\theta_{uv1} = 1.2102$
 $\theta_{uv2} = 1.6572$
 $\theta_{uv3} = 0.8220000000000001$
 $R_{uv1} = 0.11750537221761392$
 $R_{uv2} = 0.36738596870321544$
 $R_{uv3} = 0.0995050249987406$
Однорідна дисперсія = 0.9
 $m_{x1} = -0.3333333333333333$
 $m_{x2} = -0.3333333333333333$
 $m_y = 35.8$
 $a_1 = 1.0$
 $a_2 = -0.3333333333333333$
 $a_3 = 1.0$

a11 = -5.399999999999999

a22 = -8.6

b0 = 43.199999999999998

b1 = 12.299999999999997

b2 = 9.899999999999997

Натуралізація коефіцієнтів:

$\Delta x_1 = 25.0$

$\Delta x_2 = 37.5$

$x_{10} = -5.0$

$x_{20} = 7.5$

a0 = 43.679999999999986

a1 = 0.4919999999999999

a2 = 0.2639999999999999

Натуралізоване рівняння регресії = [21.0, 45.6, 40.8]

Коефіцієнти натуралізованого рівняння розраховано коректно.

Process finished with exit code 0

Контрольні запитання: 1. Що таке регресійні поліноми і де

вони застосовуються?

Регресійні поліноми – це апроксимуючі поліноми, за допомогою яких можна описати функцію. Застосовуються в теорії планування експерименту для оцінки результатів вимірів.

2. Визначення однорідності дисперсії.

Однорідність дисперсії означає, що серед усіх дисперсій не може знайтись така дисперсія, яка б значно перевищувала всі інші.

3. Що називається повним факторним експериментом?

Повний факторний експеримент – це багатофакторний експеримент, у якому використовуються всі можливі комбінації рівнів факторів.

Висновок:

У ході виконання лабораторної роботи проведено двофакторний експеримент. В ході дослідження було розроблено відповідну програму мовою програмування Python, яка моделює проведення двофакторного експерименту, перевіряє однорідність дисперсії за критерієм Романовського, розраховує коефіцієнти рівняння регресії, проводить натуралізацію рівняння регресії та відображає коефіцієнти для цього рівняння. Результати роботи, наведені у протоколі, підтверджують правильність виконання – кінцеву мету роботи було досягнуто.