

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

Методи наукових досліджень

Лабораторна робота №2

«ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконав:

студент групи ІВ-91

Черних Б.І.

Варіант 127

Перевірив:

ас. 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 діапазоні $y_{\min} \div y_{\max}$
 $y_{\max} = (30 - N_{\text{варіанту}}) * 10$,
 $y_{\min} = (20 - N_{\text{варіанту}}) * 10$.
Варіанти обираються по номеру в списку в журналі викладача.

127	-40	20	-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.

Лістинг програми

```
from random import randint
import numpy as np

def normalize(array):
    x0 = (max(array) + min(array)) / 2
    dx = x0 - min(array)
    normalized = [(array[i] - x0) / dx for i in range(len(array))]
    return normalized

def main(m):
    var = 127
    y_max = (30 - var) * 10
    y_min = (20 - var) * 10
    print(f'y_max = {y_max}, y_min = {y_min}')

    x1 = [-40, randint(-40, 20), 20]
    x2 = [-25, randint(-25, 10), 10]
    print(f'X1: {x1}')
    print(f'X2: {x2}')

    x1_n = normalize(x1)
    x2_n = normalize(x2)
    print(f'X1 normalized: {x1_n}')
    print(f'X2 normalized: {x2_n}')

    matrix_of_experiments = [[randint(y_min, y_max) for _ in range(3)] for _ in range(m)]
    print(f'Y from 1 to m: {matrix_of_experiments}')
```

```

y_average = []
dispersions = []
for i in range(3):
    y_current = (matrix_of_experiments[0][i] + matrix_of_experiments[1][i] +
                 matrix_of_experiments[2][i] + matrix_of_experiments[3][i] +
                 matrix_of_experiments[4][i]) / len(matrix_of_experiments)
    y_average.append(y_current)

    disp = ((matrix_of_experiments[0][i] - y_current) ** 2 + (matrix_of_experiments[1][i] -
y_current) ** 2 +
            (matrix_of_experiments[2][i] - y_current) ** 2 + (matrix_of_experiments[3][i] -
y_current) ** 2 +
            (matrix_of_experiments[4][i] - y_current) ** 2) / m
    dispersions.append(disp)

print(f'Y average: {y_average}')
print(f'Dispersions: {dispersions}')

main_dispersion = ((2 * (2 * m - 2)) / (m * (m - 4))) ** (1 / 2)
print(f'Main dispersion = {main_dispersion}')

f_uv = []
theta_uv = []
r_uv = []
res = 0
for i in range(3):
    for j in range(i + 1, 3):
        if dispersions[i] >= dispersions[j]:
            res = dispersions[i] / dispersions[j]
        else:
            res = dispersions[j] / dispersions[i]
        f_uv.append(res)

    theta = ((m - 2) / m) * res
    theta_uv.append(theta)

    r_uv.append(abs(theta - 1) / main_dispersion)

print(f'Fuv : {f_uv}')
print(f'θuv: {theta_uv}')
print(f'Ruv: {r_uv}')

for i in r_uv:
    if i >= 2:
        print('Дисперсії неоднорідні')
        exit()
print('Дисперсії однорідні')

mx1 = sum(x1_n) / 3
mx2 = sum(x2_n) / 3
my = sum(y_average) / 3

print(f'mx1: {mx1}')
print(f'mx2: {mx2}')
print(f'my: {my}')

a1 = (x1_n[0] ** 2 + x1_n[1] ** 2 + x1_n[2] ** 2) / 3
a2 = (x1_n[0] * x2_n[0] + x1_n[1] * x2_n[1] + x1_n[2] * x2_n[2]) / 3
a3 = (x2_n[0] ** 2 + x2_n[1] ** 2 + x2_n[2] ** 2) / 3
print(f'a1 = {a1}')
print(f'a2 = {a2}')
print(f'a3 = {a3}')

a11 = (x1_n[0] * y_average[0] + x1_n[1] * y_average[1] + x1_n[2] * y_average[2]) / 3
a22 = (x2_n[0] * y_average[0] + x2_n[1] * y_average[1] + x2_n[2] * y_average[2]) / 3
print(f'a11 = {a11}')

```

```

print(f'a22 = {a22}')

b0 = np.linalg.det([[my, mx1, mx2],
                    [a11, a1, a2],
                    [a22, a2, a3]]) / \
    np.linalg.det([[1, mx1, mx2],
                    [mx1, a1, a2],
                    [mx2, a2, a3]])

b1 = np.linalg.det([[1, my, mx2],
                    [mx1, a11, a2],
                    [mx2, a22, a3]]) / \
    np.linalg.det([[1, mx1, mx2],
                    [mx1, a1, a2],
                    [mx2, a2, a3]])

b2 = np.linalg.det([[1, mx1, my],
                    [mx1, a1, a11],
                    [mx2, a2, a22]]) / \
    np.linalg.det([[1, mx1, mx2],
                    [mx1, a1, a2],
                    [mx2, a2, a3]])

print(f'Нормоване рівняння регресії: \ny = {b0} + {b1}*x1 + {b2}*x2')
for i in range(3):
    print(f'Для X1{i + 1} та X2{i + 1}')
    print(f'y = {b0 + b1 * x1_n[i] + b2 * x2_n[i]}')

dx1 = abs(x1[2] - x1[0]) / 2
dx2 = abs(x2[2] - x2[0]) / 2
x10 = (x1[2] + x1[0]) / 2
x20 = (x2[2] + x2[0]) / 2
print(f'dx1 = {dx1}')
print(f'dx2 = {dx2}')
print(f'x10 = {x10}')
print(f'x20 = {x20}')

a0_res = b0 - b1 * (x10 / dx1) - b2 * (x20 / dx2)
a1_res = b1 / dx1
a2_res = b2 / dx2
print(f'a0 = {a0_res}')
print(f'a1 = {a1_res}')
print(f'a2 = {a2_res}')

print(f'Натуралізоване рівняння регресії: \ny = {a0_res} + {a1_res}*x1 + {a2_res}*x2')
for i in range(3):
    print(f'Для X1{i + 1} та X2{i + 1}')
    print(f'y = {a0_res + a1_res * x1[i] + a2_res * x2[i]}')

if __name__ == '__main__':
    main(m=5)

```

Результат виконання роботи

```
y_max = -970, y_min = -1070
X1: [-40, -7, 20]
X2: [-25, 0, 10]
X1 normalized: [-1.0, 0.1, 1.0]
X2 normalized: [-1.0, 0.42857142857142855, 1.0]
Y from 1 to m: [[-1036, -978, -1013], [-1042, -1026, -984], [-976, -1001, -1027], [-993, -1012, -974], [-1062, -1030, -1027]]
Y average: [-1021.8, -1009.4, -1005.0]
Dispersions: [1030.56, 352.64000000000004, 486.8]
Main dispersion = 1.7888543819998317
Fuv : [2.922413793103448, 2.117009038619556, 1.3804446460980035]
0uv: [1.2702054231717335, 0.8282667876588021, 0.8282667876588021]
Ruv: [0.15104942352527323, 0.09600178419733106, 0.09600178419733106]
Дисперсії однорідні
mx1: 0.033333333333333326
mx2: 0.14285714285714288
my: -1012.0666666666666
a1 = 0.6699999999999999
a2 = 0.6809523809523809
a3 = 0.7278911564625851
a11 = -28.046666666666667
a22 = -138.6
Нормоване рівняння регресії:
y = -1013.3999999999999 + -1.2173913043506757*x1 + 9.617391304350598*x2
Для X11 та X21
y = -1021.7999999999998
Для X12 та X22
y = -1009.3999999999998
Для X13 та X23
y = -1004.9999999999999
dx1 = 30.0
dx2 = 17.5
x10 = -10.0
x20 = -7.5
a0 = -1009.6840579710132
a1 = -0.04057971014502252
a2 = 0.5495652173914627
Натуралізоване рівняння регресії:
y = -1009.6840579710132 + -0.04057971014502252*x1 + 0.5495652173914627*x2
Для X11 та X21
y = -1021.7999999999998
Для X12 та X22
y = -1009.3999999999998
Для X13 та X23
y = -1004.9999999999999
```

Відповіді на контрольні запитання:

1. Регресійний поліном – апроксимуючий поліном за допомогою якого можна описати нашу функцію. Використовується в теорії планування експерименту.
2. Однорідністю дисперсії є необхідна умова підтвердження гіпотези про забезпечення нормального закону розподілу вимірюваної величини.
3. Якщо в багатофакторному експерименті використані всі можливі комбінації рівнів факторів, то такий експеримент називається повним факторним експериментом

