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”

Виконав:
студент групи ІВ-83
Щебетін Б. Ю.
Перевірив:
ас. Регіда П.Г.

Київ
2020 р.

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

Номер у списку: 22.

Варіант завдання: 322.

№$_{варіанта}$	x_1		x_2	
	min	max	min	max
322	10	40	30	80

1. Лістинг програми:

```
from random import randint
from math import sqrt


def r_crit(m):
    table_values = {2: 1.73, 6: 2.16, 8: 2.43, 10: 2.62, 12: 2.75, 15: 2.9, 20: 
3.08}
    for i in range(len(table_values.keys())):
        if m == list(table_values.keys())[i]:
            return list(table_values.values())[i]
        if m < list(table_values.keys())[i]:
            right_dist = abs(list(table_values.keys())[i]-m)
            left_dist = abs(list(table_values.keys())[i-1]-m)
            if left_dist < right_dist:
                return list(table_values.keys())[i-1]
            else:
                return list(table_values.keys())[i]


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


num_var = 22
y_min = (20 - num_var) * 10
y_max = (30 - num_var) * 10
x1_min = 10
x1_max = 40
x2_min = 30
x2_max = 80
m = 5


def main():
    global m
```

```
    rand_list1 = [randint(y_min, y_max) for i in range(m)]
    rand_list2 = [randint(y_min, y_max) for j in range(m)]
    rand_list3 = [randint(y_min, y_max) for k in range(m)]

    aver_arithm1 = sum(rand_list1) / len(rand_list1)
    aver_arithm2 = sum(rand_list2) / len(rand_list2)
    aver_arithm3 = sum(rand_list3) / len(rand_list3)

    dispersion1 = sum((i - aver_arithm1) ** 2 for i in rand_list1) /
len(rand_list1)
    dispersion2 = sum((i - aver_arithm2) ** 2 for i in rand_list2) /
len(rand_list2)
    dispersion3 = sum((i - aver_arithm3) ** 2 for i in rand_list3) /
len(rand_list3)

    main_deviation = sqrt((4 * m - 4) / (m * m - 4 * m))

    f12 = dispersion1 / dispersion2 if dispersion1 >= dispersion2 else
dispersion2 / dispersion1
    f23 = dispersion2 / dispersion3 if dispersion2 >= dispersion3 else
dispersion3 / dispersion2
    f13 = dispersion1 / dispersion3 if dispersion1 >= dispersion3 else
dispersion3 / dispersion1

    t12 = (m - 2) / m * f12
    t23 = (m - 2) / m * f23
    t13 = (m - 2) / m * f13

    r12 = abs(t12 - 1) / main_deviation
    r23 = abs(t23 - 1) / main_deviation
    r13 = abs(t13 - 1) / main_deviation

    r_kr = r_crit(m)

    print(f'y_min={y_min}')
    print(f'y_max={y_max}')

    print(f'\nЗн відг в діапазоні [{y_min}-{y_max}]:')
    print(*rand_list1, sep='\t')
    print(*rand_list2, sep='\t')
    print(*rand_list3, sep='\t')
    print('\nСереднє значення відгуку в кожній з точок плану:')
    print(aver_arithm1)
    print(aver_arithm2)
    print(aver_arithm3)
    print('\nДисперсії для кожної точки планування:')
    print(f'Дисперсія 1 = {dispersion1:.3f}\n' f' Дисперсія 2 =
{dispersion2:.3f}\n' f'Дисперсія 3 = {dispersion3:.3f}')
    print('\nОсновне відхилення:')
    print(f'{main_deviation:.3f}')
    print("\nКритерії Романовського")
    print(f'r12={r12:.3f}', '<' if r12 < r_kr else '>',f'r_kr={r_kr:.3f}')
    print(f'r23={r23:.3f}', '<' if r23 < r_kr else '>', f'r_kr={r_kr:.3f}')
    print(f'r13={r13:.3f}', '<' if r13 < r_kr else '>', f'r_kr={r_kr:.3f}')

    if r12 < r_kr and r23 < r_kr and r13 < r_kr:
        print('\n Ймовірність Однорідності 0.99\n')

        matrixNorm_x1_x2 = [
            [-1, -1],
            [-1, 1],
            [1, -1]
        ]
```

```
        mx_list = [sum(i) / len(i) for i in list(zip(matrixNorm_x1_x2[0],
matrixNorm_x1_x2[1], matrixNorm_x1_x2[2]))]
        my = sum([aver_arithm1, aver_arithm2, aver_arithm3]) / len([aver_arithm1,
aver_arithm2, aver_arithm3])
        a1 = sum(i[0] ** 2 for i in matrixNorm_x1_x2) / len(matrixNorm_x1_x2)
        a2 = sum(i[0] * i[1] for i in matrixNorm_x1_x2) / len(matrixNorm_x1_x2)
        a3 = sum(i[1] ** 2 for i in matrixNorm_x1_x2) / len(matrixNorm_x1_x2)
        a11 = sum(
            matrixNorm_x1_x2[i][0] * [aver_arithm1, aver_arithm2,
aver_arithm3][i] for i in range(len(matrixNorm_x1_x2))) / len(
            matrixNorm_x1_x2)
        a22 = sum(
            matrixNorm_x1_x2[i][1] * [aver_arithm1, aver_arithm2,
aver_arithm3][i] for i in range(len(matrixNorm_x1_x2))) / len(
            matrixNorm_x1_x2)
        matrix_b = [
            [1, mx_list[0], mx_list[1]],
            [mx_list[0], a1, a2],
            [mx_list[1], a2, a3]
        ]
        matrix_b1 = [
            [my, mx_list[0], mx_list[1]],
            [a11, a1, a2],
            [a22, a2, a3]
        ]
        matrix_b2 = [
            [1, my, mx_list[1]],
            [mx_list[0], a11, a2],
            [mx_list[1], a22, a3]
        ]
        matrix_b3 = [
            [1, mx_list[0], my],
            [mx_list[0], a1, a11],
            [mx_list[1], a2, a22]
        ]
        b0 = det(matrix_b1) / det(matrix_b)
        b1 = det(matrix_b2) / det(matrix_b)
        b2 = det(matrix_b3) / det(matrix_b)

        print('\nРозрахунок нормованих коефіцієнтів рівняння регресії:')

        for i in matrixNorm_x1_x2:
            print(
                f'y = b0 + b1 * x1 + b2 * x2 = {b0:.3f} + {b1:.3f} * {i[0]:2} +
{b2:.3f} * {i[1]:2}'
                f' = {b0 + b1 * i[0] + b2 * i[1]:.3f}')

        x10 = (x1_max + x1_min) / 2
        x20 = (x2_max + x2_min) / 2
        delta_x1 = (x1_max - x1_min) / 2
        delta_x2 = (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

        print('\nЗапишемо натуралізоване рівняння регресії:')
        print(
            f'y = a0 + a1 * x1 + a2 * x2 = {a_0:.3f} + {a_1:.3f} * {x1_min:3} +
{a_2:.3f} * {x2_min:3}'
            f' = {a_0 + a_1 * x1_min + a_2 * x2_min:.3f}')
        print(
            f'y = a0 + a1 * x1 + a2 * x2 = {a_0:.3f} + {a_1:.3f} * {x1_min:3} +
{a_2:.3f} * {x2_max:3}'
```

```
            f' = {a_0 + a_1 * x1_min + a_2 * x2_max:.3f}')
        print(
            f'y = a0 + a1 * x1 + a2 * x2 = {a_0:.3f} + {a_1:.3f} * {x1_max:3} +
{a_2:.3f} * {x2_min:3}'
            f' = {a_0 + a_1 * x1_max + a_2 * x2_min:.3f}')

    else:
        print('\nОднорідність не підтвердилася, підвищуємо m на 1\n')
        m += 1
        main()


main()
```

Результати виконання роботи

y_min=100

y_max=200

Значення відгуку в діапазоні [100-200]:

119 118 110 109 107

y_min=-20

y_max=80

Зн відг в діапазоні [-20-80]:

69	35	64	5	35
23	41	-19	38	-20
34	33	28	23	68

Середнє значення відгуку в кожній з точок плану:

41.6

12.6

37.2

Дисперсії для кожної точки планування:

Дисперсія 1 = 535.840

Дисперсія 2 = 724.240

Дисперсія 3 = 252.560

Основне відхилення:

1.789

Критерії Романовського

r12=0.106 < r_kr=6.000

r23=0.403 < r_kr=6.000

r13=0.153 < r_kr=6.000

Ймовірність Однорідності 0.99

Розрахунок нормованих коефіцієнтів рівняння регресії:

y = b0 + b1 * x1 + b2 * x2 = 24.900 + -2.200 * -1 + -14.500 * -1 = 41.600

y = b0 + b1 * x1 + b2 * x2 = 24.900 + -2.200 * -1 + -14.500 * 1 = 12.600

y = b0 + b1 * x1 + b2 * x2 = 24.900 + -2.200 * 1 + -14.500 * -1 = 37.200

Запишемо натуралізоване рівняння регресії:

y = a0 + a1 * x1 + a2 * x2 = 60.467 + -0.147 * 10 + -0.580 * 30 = 41.600

y = a0 + a1 * x1 + a2 * x2 = 60.467 + -0.147 * 10 + -0.580 * 80 = 12.600

y = a0 + a1 * x1 + a2 * x2 = 60.467 + -0.147 * 40 + -0.580 * 30 = 37.200

Process finished with exit code 0

Теоретичні відомості

- Що таке регресійні поліноми і де вони застосовуються?

В теорії планування експерименту найважливішою частиною є оцінка результатів вимірів. При цьому використовують апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. В ТПЕ ці поліноми отримали спеціальну назву - регресійні поліноми, а їх знаходження та аналіз - регресійний аналіз. Найчастіше в якості базисної функції використовується ряд Тейлора, який має скінченну кількість членів.

$$F(x) = F(a) + \frac{x-a}{1!}F'(a) + \frac{(x-a)^2}{2!}F''(a) + \ldots + \frac{(x-a)^N}{N!}F^{(N)}(a)$$

Але при використанні апроксимуючого полінома Тейлора в його початковому вигляді виникає ряд проблем, пов'язаних із знаходженням похідних, оскільки нам невідома функція, а відомий лише ряд її значень. Тому ми замінюємо поліном Тейлора аналогічним йому рівнянням регресії:

$$\hat{y} = b_0 + \sum_{i=1}^{k} b_i x_i + \sum_{i,j=1}^{k} b_{i,j} x_i x_j + \sum_{i=1}^{k} b_{i,i} x_i^2 + \sum_{i,j,n=1}^{k} b_{i,j,k} x_i x_j x_n + \ldots$$

де k –кількість факторів (кількість x)

Мета даної роботи – дослідити лінійну регресійну модель

$$\hat{y} = b_0 + \sum_{i=1}^{k} b_i x_i$$

- Визначення однорідності дисперсії.

  Однорідність дисперсії означає, що серед усіх дисперсій нема таких, які б значно перевищували одна одну. Перевірка однорідності проводиться за допомогою різних статистичних критеріїв.

- Що називається повним факторним експериментом

  Для знаходження коефіцієнтів у лінійному рівнянні регресії застосовують повний факторний експеримент (ПФЕ). Якщо в багатофакторному експерименті використані всі можливі комбінації рівнів факторів, то такий експеримент називається повним факторним експериментом.