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 №1
«Проведення двофакторного експерименту з використанням лінійного рівняння регресії»

Виконав:
студент групи ІВ-83
Грисюк М. О.
Залікова книжка №8306
Перевірив Регіда П. Г.

Київ - 2020 р.

№варіанта	x_1		x_2	
	min	max	min	max
307	-5	15	25	45

y_{max}=(30-7)*10=230

y_{min}=(20-7)*10=130

```
from random import randint
from math import sqrt
n_variant = 7
y_min = (20 - n_variant) * 10
y_max = (30 - n_variant) * 10
x1_min = -5
x1_max = 15
x2_min = 25
x2_max = 45
p = float(input("Введіть дочірну ймовірніст p = "))
m = int(input('Введіть кількість дослідів у за однієї і тієї ж комбінації факторів m = 
'))
print("-" * 100)
print('Значення за варіантом:')
print('x1_min = ',x1_min)
print('x1_max = ',x1_max)
print('x2_min = ',x2_min)
print('x2_max = ',x2_max)
print('y_min = ',y_min)
print('y_max = ',y_max)
print("-" * 100)

def get_r_kr(m):
    if p == 0.99:
        table_values = {2: 1.73, 6: 2.16, 8: 2.43, 10: 2.62, 12: 2.75, 15: 2.9, 20: 
3.08}
    elif p == 0.98:
        table_values = {2: 1.72, 6: 2.13, 8: 2.37, 10: 2.54, 12: 2.66, 15: 2.8, 20: 
2.96}
    elif p == 0.95:
        table_values = {2: 1.71, 6: 2.10, 8: 2.27, 10: 2.41, 12: 2.52, 15: 2.64, 20: 
2.78}
    elif p == 0.9:
        table_values = {2: 1.69, 6: 2, 8: 2.17, 10: 2.29, 12: 2.39, 15: 2.49, 20: 2.62}
    else:
        print("Введіть значення довірчої ймовірності з таблиці (0.99, 0.98, 0.95 або 
0.9).")
    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]:
            less_than_m_key = list(table_values.keys())[i]
            less_than_m = list(table_values.values())[i]
            more_than_m_key = list(table_values.keys())[i + 1]
            more_than_m = list(table_values.values())[i + 1]
            return less_than_m + (more_than_m - less_than_m) * (m - less_than_m_key) /
```

```
(
                    more_than_m_key - less_than_m_key)


def determinant(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]


def main():
    global m
    response_list1 = [randint(y_min, y_max) for i in range(m)]
    response_list2 = [randint(y_min, y_max) for j in range(m)]
    response_list3 = [randint(y_min, y_max) for k in range(m)]

    average1 = sum(response_list1) / len(response_list1)
    average2 = sum(response_list2) / len(response_list2)
    average3 = sum(response_list3) / len(response_list3)

    dispersion1 = sum((i - average1) ** 2 for i in response_list1) /
len(response_list1)
    dispersion2 = sum((i - average2) ** 2 for i in response_list2) /
len(response_list2)
    dispersion3 = sum((i - average3) ** 2 for i in response_list3) /
len(response_list3)

    major_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) / major_deviation
    r23 = abs(t23 - 1) / major_deviation
    r13 = abs(t13 - 1) / major_deviation

    r_kr = get_r_kr(m)

    print(f'\nЗначення відгуку в діапазоні [{y_min} - {y_max}]:')
    print(*response_list1, sep='\t')
    print(*response_list2, sep='\t')
    print(*response_list3, sep='\t')
    print("-" * 100)
    print('\nСереднє значення відгуку в кожній з точок плану:')
    print("ȳ1 = " + str(average1))
    print("ȳ2 = " + str(average2))
    print("ȳ3 = " + str(average3))
    print("-" * 100)
    print('\nДисперсії для кожної точки планування:')
    print("σ{y1} = " + "{:.3f}".format(dispersion1))
    print("σ{y2} = " + "{:.3f}".format(dispersion2))
    print("σ{y3} = " + "{:.3f}".format(dispersion3))
```

```
    print("-" * 100)
    print('\nОсновне відхилення:')
    print(f'{major_deviation:.3f}')
    print("-" * 100)
    print(f'\nr12 = {r12:.3f} ', ' < ' if r12 < r_kr else ' > ', f'r_kr = {r_kr:.3f}')
    print(f'\nr23 = {r23:.3f} ', ' < ' if r23 < r_kr else ' > ', f'r_kr = {r_kr:.3f}')
    print(f'\nr13 = {r13:.3f} ', ' < ' if r13 < r_kr else ' > ', f'r_kr = {r_kr:.3f}')
    print("-" * 100)

    if r12 < r_kr and r23 < r_kr and r13 < r_kr:
        print('\nОднорідність підтверджується з ймовірністю ' + str(p))
        print("-" * 100)
        normalized_x1_x2 = [
            [-1, -1],
            [-1, 1],
            [1, -1]
        ]

        mx_list = [sum(i) / len(i) for i in list(zip(normalized_x1_x2[0],
normalized_x1_x2[1], normalized_x1_x2[2]))]
        my = sum([average1, average2, average3]) / len([average1, average2, average3])
        a1 = sum(i[0] ** 2 for i in normalized_x1_x2) / len(normalized_x1_x2)
        a2 = sum(i[0] * i[1] for i in normalized_x1_x2) / len(normalized_x1_x2)
        a3 = sum(i[1] ** 2 for i in normalized_x1_x2) / len(normalized_x1_x2)
        a11 = sum(
            normalized_x1_x2[i][0] * [average1, average2, average3][i] for i in
range(len(normalized_x1_x2))) / len(
            normalized_x1_x2)
        a22 = sum(
            normalized_x1_x2[i][1] * [average1, average2, average3][i] for i in
range(len(normalized_x1_x2))) / len(
            normalized_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 = determinant(matrix_b1) / determinant(matrix_b)
        b1 = determinant(matrix_b2) / determinant(matrix_b)
        b2 = determinant(matrix_b3) / determinant(matrix_b)
        print('\nРозрахунок нормованих коефіцієнтів рівняння регресії:')

        for i in normalized_x1_x2:
            print(
                f'ŷ = {b0:.3f} + {b1:.3f} * {i[0]:2} + {b2:.3f} * {i[1]:2}'
                f' = {b0 + b1 * i[0] + b2 * i[1]:.3f}')
```

```python
        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("-" * 100)
        print('\nЗапишемо натуралізоване рівняння регресії:')
        print(
            f'ŷ = {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'ŷ = {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'ŷ = {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')
        print("-" * 100)
        m += 1
        main()
main()
```

```
Введіть дочірну ймовірніст p = 0.9
Введіть кількість дослідів у за однієї і тієї ж комбінації факторів m = 5
----------------------------------------------------------------------
Значення за варіантом:
x1_min =  -5
x1_max =  15
x2_min =  25
x2_max =  45
y_min =  130
y_max =  230
----------------------------------------------------------------------


Значення відгуку в діапазоні [130 - 230]:
199 202 152 167 183
208 206 166 224 198
222 171 191 208 214
----------------------------------------------------------------------


Середнє значення відгуку в кожній з точок плану:
ȳ1 = 180.6
ȳ2 = 200.4
ȳ3 = 201.2
----------------------------------------------------------------------


Дисперсії для кожної точки планування:
σ{y1} = 361.040
σ{y2} = 367.040
σ{y3} = 331.760
----------------------------------------------------------------------
```

```
Основне відхилення:
1.789
---------------------------------------------------------------

r12 = 0.218   <  r_kr = 1.922

r23 = 0.188   <  r_kr = 1.922

r13 = 0.194   <  r_kr = 1.922
---------------------------------------------------------------

Однорідність підтверджується з ймовірністю 0.9
---------------------------------------------------------------

Розрахунок нормованих коефіцієнтів рівняння регресії:
ŷ = 200.800 + 10.300 * -1 + 9.900 * -1 = 180.600
ŷ = 200.800 + 10.300 * -1 + 9.900 *  1 = 200.400
ŷ = 200.800 + 10.300 *  1 + 9.900 * -1 = 201.200
---------------------------------------------------------------

Запишемо натуралізоване рівняння регресії:
ŷ = 161.000 + 1.030 *  -5 + 0.990 *  25 = 180.600
ŷ = 161.000 + 1.030 *  -5 + 0.990 *  45 = 200.400
ŷ = 161.000 + 1.030 *  15 + 0.990 *  25 = 201.200

Process finished with exit code 0
```

Відповіді на контрольні запитання:

1. .

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.

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

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

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

2. Однорідність дисперсії означає, що серед усіх дисперсій нема таких, які б значно перевищували одна одну. Перевірка однорідності проводиться за допомогою різніх статистичних критеріїв.
3. 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.