

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

Лабораторна робота №3  
з дисципліни «Методи оптимізації та планування експерименту»

«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З  
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконав:  
студент групи ІВ-92  
Коптюх Н.Є  
Залікова книжка № ІВ-9214  
Номер у списку: 12  
Варіант: 212  
Перевірив:  
ас. Регіда П.Г.

Київ 2021

**Мета:** провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

## Завдання на лабораторну роботу:

1. Скласти матрицю планування для дробового трьохфакторного експерименту. Провести експеримент в усіх точках факторного простору, повторивши N експериментів, де N – кількість експериментів (рядків матриці планування) в усіх точках факторного простору – знайти значення функції відгуку Y. Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі (випадковим чином).

$$y_{\max} = 200 + x_{\text{ср max}};$$

$$y_{\min} = 200 + x_{\text{ср min}}$$

$$\text{де } x_{\text{ср max}} = \frac{x_{1\max} + x_{2\max} + x_{3\max}}{3}, \quad x_{\text{ср min}} = \frac{x_{1\min} + x_{2\min} + x_{3\min}}{3}$$

2. Знайти коефіцієнти лінійного рівняння регресії. Записати лінійне рівняння регресії.

3. Провести 3 статистичні перевірки.

4. Написати комп'ютерну програму, яка усе це виконує.

## Варіант:

№_варіанта	X <sub>1</sub>		X <sub>2</sub>		X <sub>3</sub>	
	min	max	min	max	min	max
212	-40	20	5	40	-40	-20

## Лістинг програми:

```
import numpy as np
from scipy.stats import t,f

def table_student(prob, f3):
    x_vec = [i*0.0001 for i in range(int(5/0.0001))]
    par = 0.5 + prob/0.1*0.05
    for i in x_vec:
        if abs(t.cdf(i, f3) - par) < 0.000005:
            return i

def table_fisher(prob, d, f3):
```

```

x_vec = [i*0.001 for i in range(int(10/0.001))]
for i in x_vec:
    if abs(f.cdf(i, 4-d, f3)-prob) < 0.0001:
        return i

def make_norm_plan_matrix(plan_matrix, matrix_of_min_and_max_x):
    X0 = np.mean(matrix_with_min_max_x, axis=1)
    interval_of_change = np.array([(matrix_of_min_and_max_x[i, 1] -
X0[i]) for i in range(len(plan_matrix[0]))])
    X_norm = np.array(
        [[round((plan_matrix[i, j] - X0[j]) / interval_of_change[j], 3)
for j in range(len(plan_matrix[i]))]
        for i in range(len(plan_matrix))])
    return X_norm

def cochran_check(Y_matrix):
    fisher = table_fisher(0.95, 1, (m - 1) * 4)
    mean_Y = np.mean(Y_matrix, axis=1)
    dispersion_Y = np.mean((Y_matrix.T - mean_Y) ** 2, axis=0)
    Gp = np.max(dispersion_Y) / (np.sum(dispersion_Y))
    if Gp < fisher/(fisher+(m-1)-2):
        return True
    return False

def students_t_test(norm_matrix, Y_matrix):
    mean_Y = np.mean(Y_matrix, axis=1)
    dispersion_Y = np.mean((Y_matrix.T - mean_Y) ** 2, axis=0)
    mean_dispersion = np.mean(dispersion_Y)
    sigma = np.sqrt(mean_dispersion / (N * m))
    betta = np.mean(norm_matrix.T * mean_Y, axis=1)
    f3 = (m - 1) * 4
    t = np.abs(betta) / sigma
    return np.where(t > table_student(0.95, f3))

def phisher_criterion(Y_matrix, d):
    if d == N:
        return False
    Sad = m / (N - d) * np.mean(check1 - mean_Y)
    mean_dispersion = np.mean(np.mean((Y_matrix.T - mean_Y) ** 2,
axis=0))
    Fp = Sad / mean_dispersion
    f3 = (m - 1) * 4
    if Fp > table_fisher(0.95, d, f3):
        return False
    return True

matrix_with_min_max_x = np.array([[-40, 20], [5, 40], [-40, -20]])

```

```

m = 6
N = 4
plan_matr = np.array(
    [np.random.randint(-40, 20, size=N), np.random.randint(5, 40,
size=N), np.random.randint(-40, -20, size=N)]).T
norm_matrix = make_norm_plan_matrix(plan_matr, matrix_with_min_max_x)
plan_matr = np.insert(plan_matr, 0, 1, axis=1)
norm_matrix = np.insert(norm_matrix, 0, 1, axis=1)
Y_matrix = np.random.randint(200 + np.mean(matrix_with_min_max_x,
axis=0)[0],
                             200 + np.mean(matrix_with_min_max_x,
axis=0)[1], size=(N, m))
mean_Y = np.mean(Y_matrix, axis=1)
if cochrane_check(Y_matrix):
    b_natura = np.linalg.lstsq(plan_matr, mean_Y, rcond=None)[0]
    b_norm = np.linalg.lstsq(norm_matrix, mean_Y, rcond=None)[0]
    check1 = np.sum(b_natura * plan_matr, axis=1)
    check2 = np.sum(b_norm * norm_matrix, axis=1)
    indexes = students_t_test(norm_matrix, Y_matrix)
    print("Матриця плану експерименту: \n", plan_matr)
    print("Нормована матриця: \n", norm_matrix)
    print("Матриця відгуків: \n", Y_matrix)
    print("Середні значення Y: ", mean_Y)
    print("Натуралізовані коефіцієнти: ", b_natura)
    print("Нормовані коефіцієнти: ", b_norm)
    print("Перевірка 1: ", check1)
    print("Перевірка 2: ", check2)
    print("Індекси коефіцієнтів, які задовольняють критерію Стьюдента:
", np.array(indexes)[0])
    print("Критерій Стьюдента: ", np.sum(np.sum(b_natura[indexes] *
plan_matr[:, indexes], axis=1), axis=1))
    if phisher_criterion(Y_matrix, np.size(indexes)):
        print("Рівняння регресії адекватно оригіналу.")
    else:
        print("Рівняння регресії неадекватно оригіналу.")
else:
    print("Дисперсія неоднорідна!")

```

**Результат підготовки та виконання програми:**

Матриця плану експерименту:

```
[[ 1  4 34 -21]
 [ 1 -39 18 -24]
 [ 1 -11 34 -31]
 [ 1 10 20 -36]]
```

Нормована матриця:

```
[[ 1.      0.467  0.657  0.9   ]
 [ 1.     -0.967 -0.257  0.6   ]
 [ 1.     -0.033  0.657 -0.1   ]
 [ 1.      0.667 -0.143 -0.6   ]]
```

Матриця відгуків:

```
[[211 185 181 212 198 181]
 [204 208 175 211 189 183]
 [182 193 196 181 176 212]
 [183 193 183 176 189 188]]
```

Середні значення Y: [194.66666667 195. 190. 185.33333333]

Натуралізовані коефіцієнти: [ 2.05282814e+02 -6.19095477e-02 4.06365159e-02 5.59530988e-01]

Нормовані коефіцієнти: [190.03053724 -1.8566781 0.71186143 5.59500572]

Перевірка 1: [194.66666667 195. 190. 185.33333333]

Перевірка 2: [194.66666667 195. 190. 185.33333333]

Індекси коефіцієнтів, які задовольняють критерію Стюдента: [0 1 2 3]

Критерій Стюдента: [194.66666667 195. 190. 185.33333333]

Рівняння регресії неадекватно оригіналу.

## Відповіді на контрольні запитання:

- 1) У деяких випадках немає необхідності проводити повний факторний експеримент (ПФЕ). Якщо буде використовуватися лінійна регресія, то можливо зменшити кількість рядків матриці ПФЕ до кількості коефіцієнтів регресійної моделі. Кількість дослідів слід скоротити, використовуючи для планування так звані регулярні дробові репліки від повного факторного експерименту, що містять відповідну кількість дослідів і зберігають основні властивості матриці планування – це означає дробовий факторний експеримент (ДФЕ). Репліка, що включає тільки половину експериментів ПФЕ, називається напівреплікою, що включає четверту частину дослідів - чвертьреплікою і т. д.
- 2) Розрахункове значення Кохрена потрібне для порівняння цього значення з табличним, задля перевірки однорідності дисперсій.
- 3) Критерій Стюдента перевіряється для того, щоб залишити в рівняння регресії лише суттєві коефіцієнти задля спрощення моделі об'єкта. Спочатку розраховується експериментальне значення критерія для кожного коефіцієнта, після чого вони порівнюються з табличним значенням. Якщо виконується нерівність  $t_s < t_{\text{табл}}$ , то то приймається нуль - гіпотеза, тобто вважається, що знайдений коефіцієнт  $\beta_s$  є статистично незначущим і його слід виключити з рівняння регресії.

Якщо  $t_s > t_{\text{табл}}$ , то гіпотеза не підтверджується, тобто  $\beta_s$  – значимий коефіцієнт і він залишається в рівнянні регресії.

- 4) Отримане рівняння регресії необхідно перевірити на адекватність досліджуваному об'єкту. Для цієї мети необхідно оцінити, наскільки відрізняються середні значення у вихідної величини, отриманої в точках факторного простору, і значення у, отриманого з рівняння регресії в тих самих точках факторного простору. Для цього використовують дисперсію адекватності. Адекватність моделі перевіряють за F-критерієм Фішера, який дорівнює відношенню дисперсії адекватності до дисперсії відтворюваності

## **Висновок:**

Отже, ми провели дробовий трьохфакторний експеримент. Була складена матриця планування, знайдені коефіцієнти рівняння регресії та проведені 3 статистичні перевірки. Під час виконання роботи проблем не виникало. Отримані результати збігаються з очікуваними.