

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський Політехнічний Інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №3

з дисципліни «Методи оптимізації і планування експерименту»

*на тему “Проведення трьохфакторного
експерименту з використанням лінійного
рівняння регресії”*

Виконав:
студент 2-го курсу ФІОТ
групи ІО-82
Шевчук О. Б.
Номер у списку: 24

Перевірив:
Регіда П. Г.

Варіант

224	-20	15	25	45	-20	-15
-----	-----	----	----	----	-----	-----

Лістинг коду програми

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

def normalise(factors, def_matrix):
    X0 = np.mean(def_matrix, axis=1)
    delta = np.array([(def_matrix[i, 1] - X0[i]) for i in range(len(factors[0]))])
    X_norm = np.array(
        [[round((factors[i, j] - X0[j]) / delta[j], 3) for j in range(len(factors[i]))]
         for i in range(len(factors))])
    return X_norm

def cohran(Y_matrix):
    fish = fisher(0.95, 1, (6 - 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 < fish / (fish + (6 - 1) - 2):
        return True
    return False

def 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 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 / (4 * 6))
    betta = np.mean(norm_matrix.T * mean_Y, axis=1)
    f3 = (6 - 1) * 4
    t = np.abs(betta) / sigma
    return np.where(t > student(0.95, f3))

def 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
```

Лабораторна робота №4

```
def fisher_criteria(Y_matrix, d):
    if d == 4:
        return False
    sad = 6 / (4 - 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 = (6 - 1) * 4
    if Fp > fisher(0.95, d, f3):
        return False
    return True

x1min, x1max = -20, 15
x2min, x2max = 25, 45
x3min, x3max = -20, -15
def_matrix = np.array([[x1min, x1max], [x2min, x2max], [x3min, x3max]])
factors = np.array([np.random.randint(x1min, x1max, size=4), np.random.randint(x2min, x2max, size=4),
np.random.randint(x3min, x3max, size=4)]).T
norm_factors = normalise(factors, def_matrix)
factors = np.insert(factors, 0, 1, axis=1)
norm_factors = np.insert(norm_factors, 0, 1, axis=1)
Y_matrix = np.random.randint(200 + np.mean(def_matrix, axis=0)[0], 200 + np.mean(def_matrix, axis=0)[1], size=(4,
6))
mean_Y = np.mean(Y_matrix, axis=1)

if cohran(Y_matrix):
    b_natural = np.linalg.lstsq(factors, mean_Y, rcond=None)[0]
    b_norm = np.linalg.lstsq(norm_factors, mean_Y, rcond=None)[0]
    check1 = np.sum(b_natural * factors, axis=1)
    check2 = np.sum(b_norm * norm_factors, axis=1)
    indexes = students_t_test(norm_factors, Y_matrix)
    print("Фактори: \n", factors)
    print("Нормована матриця факторів: \n", norm_factors)
    print("Функції відгуку: \n", Y_matrix)
    print("Середні значення Y: ", mean_Y)
    print("Натуралізовані коефіцієнти: ", b_natural)
    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_natural[indexes] * factors[:, indexes], axis=1), axis=1))
    if fisher_criteria(Y_matrix, np.size(indexes)):
        print("Рівняння регресії є адекватним оригіналу.")
    else:
        print("Рівняння регресії не є адекватним оригіналу.")
else:
    print("Дисперсія неоднорідна!")
```