

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 3
з дисципліни «Методи наукових досліджень»
на тему «ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАЛА:
студентка 2 курсу
групи ІВ-91

Сайко С. А.
Залікова – 9126

ПЕРЕВІРИВ:
ас. Регіда П. Г.

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).

$$y_{max} = 200 + x_{cp\ max}$$

$$y_{min} = 200 + x_{cp\ min}$$

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

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

3. Провести 3 статистичні перевірки.

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

№ варіанту	X ₁		X ₂		X ₃	
	min	max	min	max	min	max
	-20	15	-15	35	-15	-10

Програмний код

```
from functools import reduce
from random import randint
from math import sqrt
import numpy as np
from numpy.linalg import det
from prettytable import PrettyTable

def average(list):
    avrg = 0
    for element in list:
        avrg += element
    avrg = avrg / len(list)
    return avrg

def dispersion(list):
    list_average = average(list)
    dispersion = 0
    for element in list:
        dispersion += (element - list_average) ** 2 / len(list)
    return dispersion

def main():
```

```

minX = [-20, -15, -15]
maxX = [15, 35, -10]

maxY = int(200 + average(maxX))
minY = int(200 + average(minX))

m = 3
N = 2**(m-1)
gener_x = []
for i in range(N):
    gener_x.append([])
    gener_x[i].append(randint(minX[0], maxX[0]))
    gener_x[i].append(randint(minX[1], maxX[1]))
    gener_x[i].append(randint(minX[2], maxX[2]))

plan_matrix = []
for str_ind in range(N):
    plan_matrix.append([])
    plan_matrix[str_ind].append(1)
    for xi in gener_x[str_ind]:
        index = gener_x[str_ind].index(xi)
        if average([minX[index], maxX[index]]) < xi:
            plan_matrix[str_ind].append(1)
        else:
            plan_matrix[str_ind].append(-1)
#Матриця планування з натуралізованими значеннями факторів
experement_matrix = []
for i in range(len(plan_matrix)):
    experement_matrix.append([])
    for j in range(1, len(plan_matrix[i])):
        if plan_matrix[i][j] == 1:
            experement_matrix[i].append(maxX[j-1])
        else:
            experement_matrix[i].append(minX[j - 1])

gener_y = []
for i in range(N):
    gener_y.append([])
    gener_y[i].append(randint(minY, maxY))
    gener_y[i].append(randint(minY, maxY))
    gener_y[i].append(randint(minY, maxY))
b = np.array(experement_matrix)
Tgener_y = b.T
avrg_y = []
for i in range(len(gener_y)):
    avrg_y.append(average(gener_y[i]))

# Математичне очікування
a = np.array(experement_matrix)
Texperement_matrix = a.T

mx1 = reduce(lambda a, b: a + b, Texperement_matrix[0]) / 4
mx2 = reduce(lambda a, b: a + b, Texperement_matrix[1]) / 4
mx3 = reduce(lambda a, b: a + b, Texperement_matrix[2]) / 4
my = reduce(lambda a, b: a + b, avrg_y) / 4

a1 = sum([Texperement_matrix[0][i] * avrg_y[i] for i in range(4)]) / 4
a2 = sum([Texperement_matrix[1][i] * avrg_y[i] for i in range(4)]) / 4
a3 = sum([Texperement_matrix[2][i] * avrg_y[i] for i in range(4)]) / 4

a11 = sum([i * i for i in Texperement_matrix[0]]) / 4
a22 = sum([i * i for i in Texperement_matrix[1]]) / 4
a33 = sum([i * i for i in Texperement_matrix[2]]) / 4

a12 = sum([Texperement_matrix[0][i] * Texperement_matrix[1][i] for i in
range(4)]) / 4
a13 = sum([Texperement_matrix[1][i] * Texperement_matrix[2][i] for i in
range(4)]) / 4
a23 = sum([Texperement_matrix[2][i] * Texperement_matrix[0][i] for i in

```

```

range(4)]) / 4
    a21 = a12
    a31 = a13
    a32 = a23

    b0 = det([[my, mx1, mx2, mx3],
               [a1, a11, a12, a13],
               [a2, a21, a22, a23],
               [a3, a31, a32, a33]]) / det([[1, mx1, mx2, mx3],
                                               [mx1, a11, a12, a13],
                                               [mx2, a21, a22, a23],
                                               [mx3, a31, a32, a33]])

    b1 = det([[1, my, mx2, mx3],
               [mx1, a1, a12, a13],
               [mx2, a2, a22, a23],
               [mx3, a3, a32, a33]]) / det([[1, mx1, mx2, mx3],
                                               [mx1, a11, a12, a13],
                                               [mx2, a21, a22, a23],
                                               [mx3, a31, a32, a33]])

    b2 = det([[1, mx1, my, mx3],
               [mx1, a11, a1, a13],
               [mx2, a21, a2, a23],
               [mx3, a31, a3, a33]]) / det([[1, mx1, mx2, mx3],
                                               [mx1, a11, a12, a13],
                                               [mx2, a21, a22, a23],
                                               [mx3, a31, a32, a33]])

    b3 = det([[1, mx1, mx2, my],
               [mx1, a11, a12, a1],
               [mx2, a21, a22, a2],
               [mx3, a31, a32, a3]]) / det([[1, mx1, mx2, mx3],
                                               [mx1, a11, a12, a13],
                                               [mx2, a21, a22, a23],
                                               [mx3, a31, a32, a33]])

    c = np.array(plan_matrix)
    Tplan_matrix = c.T
    plan_table = PrettyTable()
    plan_table.field_names = ['№', 'X0', 'X1', 'X2', 'X3', *[f"Y{i}" for i in
range(1, N)]]
    for i in range(N):
        plan_table.add_row([i + 1, Tplan_matrix[0][i], Tplan_matrix[1][i],
Tplan_matrix[2][i], Tplan_matrix[3][i], *gener_y[i]])
    print(plan_table)
    experement_table = PrettyTable()
    experement_table.field_names = ['№', 'X1', 'X2', 'X3', *[f"Y{i}" for i in
range(1, N)]]
    for i in range(N):
        experement_table.add_row(
            [i + 1, *experement_matrix[i], *gener_y[i]])

    print(experement_table)
    print(avrg_y)
    print(f'y = {b0} + {b1}*x1 + {b2}*x2 + {b3}*x3')
    # dispersion_list = []
    # for i in range(N):
    #     dispersion_list.append(dispersion(gener_y[i]))
    for i in range(4):
        y = b0 + b1 * Texperement_matrix[0][i] + b2 * Texperement_matrix[1][i] +
b3 * Texperement_matrix[2][i]
        print('y =', y)
        dispersion = (((gener_y[0][i] - avrg_y[i])**2 + (gener_y[1][i] -
avrg_y[i])**2 + (gener_y[2][i] - avrg_y[i])**2 + (gener_y[3][i] - avrg_y[i])**2
) / 4 for i in
                        range(3))
    print('dispersion:', dispersion)

    gp = max(dispersion) / sum(dispersion)

```

```

print('Gp =', gp)

# Рівень значимості  $\alpha = 0.05$ ;  $f_1 = m - 1 = 2$ ;  $f_2 = N = 4$ 
# За таблицею  $G_t = 0.7679$ 
if gp < 0.7679:
    print('Дисперсія однорідна')
else:
    print('Дисперсія неоднорідна')
    exit()

# Оцінка значимості коефіцієнтів регресії згідно критерію Стюдента
s2b = sum(dispersion) / N
s2bs_avg = s2b / (N * m)
sb = s2bs_avg**(1/2)

beta0 = sum([(avrg_y[i] * Tplan_matrix[0][i]) for i in range(4)]) / 4
beta1 = sum([(avrg_y[i] * Tplan_matrix[1][i]) for i in range(4)]) / 4
beta2 = sum([(avrg_y[i] * Tplan_matrix[2][i]) for i in range(4)]) / 4
beta3 = sum([(avrg_y[i] * Tplan_matrix[3][i]) for i in range(4)]) / 4

beta_arr = [beta0, beta1, beta2, beta3]
t_arr = [abs(beta_arr[i]) / sb for i in range(4)]

#  $f_3 = f_1 * f_2 = 2 * 4 = 8$ 
# З таблиці беремо значення 2.306
indexes = []
for i, v in enumerate(t_arr):
    if t_arr[i] > 2.306:
        indexes.append(i)
    else:
        print(f'Коефіцієнт  $b\{i\} = \{v\}$  приймаємо не значним')
ind = [i for i in range(N)]

b_list = [b0, b1, b2, b3]
print(f'y =  $b\{ind[0]\}$ ')

b_res = [b_list[ind[0]] for _ in range(4)]
for i in b_res:
    print(f'y =  $\{i\}$ ')

# Критерій Фішера
# кількість значимих коефіцієнтів
d = 1
s2_ad = m * sum([(avrg_y[i] - b_res[i]) ** 2 for i in range(4)]) / 4 - d
fp = s2_ad / s2b
print(f'Fp =  $\{fp\}$ ')

#  $F_t = 4.5$ 
if fp > 4.5:
    print('Рівняння регресії неадекватно оригіналу при рівні значимості 0.05')
else:
    print('Рівняння регресії адекватно оригіналу при рівні значимості 0.05')

if __name__ == '__main__':
    main()

```

Результат роботи програми

```
+---+---+---+---+---+---+---+---+
| № | X0 | X1 | X2 | X3 | Y1 | Y2 | Y3 |
+---+---+---+---+---+---+---+---+
| 1 | 1 | -1 | -1 | 1 | 194 | 204 | 185 |
| 2 | 1 | -1 | -1 | -1 | 193 | 188 | 185 |
| 3 | 1 | 1 | 1 | 1 | 196 | 205 | 204 |
| 4 | 1 | 1 | 1 | -1 | 185 | 206 | 192 |
+---+---+---+---+---+---+---+---+
+---+---+---+---+---+---+---+---+
| № | X1 | X2 | X3 | Y1 | Y2 | Y3 |
+---+---+---+---+---+---+---+---+
| 1 | -20 | -15 | -10 | 194 | 204 | 185 |
| 2 | -20 | -15 | -15 | 193 | 188 | 185 |
| 3 | 15 | 35 | -10 | 196 | 205 | 204 |
| 4 | 15 | 35 | -15 | 185 | 206 | 192 |
+---+---+---+---+---+---+---+---+
[194.33333333333334, 188.66666666666666, 201.66666666666666, 194.33333333333334]
y = 193.88588235294065 + 0.04588235294117612*x1 + 0.09788235294117692*x2 + -0.0*x3
y = 191.49999999999995
y = 191.49999999999995
y = 197.99999999999995
y = 197.99999999999995
dispersion: [22.94444444444449, 200.69444444444468, 163.61111111111092]
Gr = 0.5182555053439499
Дисперсія однорідна
Коефіцієнт b1 = 1.1442166588663933 приймаємо не значним
Коефіцієнт b2 = 1.1442166588663933 приймаємо не значним
Коефіцієнт b3 = 1.1442166588663933 приймаємо не значним
y = b0
y = 193.88588235294065
y = 193.88588235294065
y = 193.88588235294065
y = 193.88588235294065
Fr = 0.6728049710234879
Рівняння регресії адекватно оригіналу при рівні значимості 0.05
```

Висновок:

Під час виконання даної лабораторної роботи я провела трьохфакторний експеримент, перевірила однорідність дисперсії за критерієм Кохрена, отримала коефіцієнти рівняння регресії, оцінила значимість знайдених коефіцієнтів за критеріями Стюдента та Фішера. Отже, мета лабораторної роботи була досягнута.

Контрольні питання

1. Що називається дробовим факторним експериментом?

ДФЕ – це частина ПФЕ, який мінімізує число дослідів, за рахунок тієї інформації, яка не дуже істотна для побудови лінійної моделі.

2. Для чого потрібно розрахункове значення Кохрена?

Критерій Кохрена використовують для порівняння трьох і більше виборок однакового обсягу n

3. Для чого перевіряється критерій Стюдента?

За допомогою критерія Стюдента перевіряється значущість коефіцієнта рівняння регресії.

Тобто, якщо виконується нерівність $t_s < t_{\text{табл}}$, то приймається нуль-гіпотеза, тобто вважається, що знайдений коефіцієнт β_s є статистично незначущим і його слід виключити з рівняння регресії.

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

4. Чим визначається критерій Фішера і як його застосовувати?

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 y , отриманого з рівняння регресії в тих самих точках факторного простору. Для цього використовують дисперсію адекватності.

Адекватність моделі перевіряють за F-критерієм Фішера.

Якщо $F_{\text{прак}} < F_{\text{теор}}$, то отримана математична модель з прийнятим рівнем статистичної значимості q адекватна експериментальним даним.