

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

**Методи оптимізації та планування експерименту**

**ЛАБОРАТОРНА РОБОТА № 6**

**«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ  
ВИКОРИСТАННІ РІВНЯННЯ РЕГРЕСІЇ З КВАДРАТИЧНИМИ  
ЧЛЕНАМИ»**

**ВИКОНАЛА:**

студентка II курсу ФІОТ

групи ІО-92

Варіант №211

Карнаухова Анастасія

**ПЕРЕВІРИВ:**

Доц. Порєв В.М.

Київ – 2021

## ЛАБОРАТОРНА РОБОТА № 5

### Тема: ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ ВИКОРИСТАННІ РІВНЯННЯ РЕГРЕСІЇ З КВАДРАТИЧНИМИ ЧЛЕНАМИ

**Мета:** провести трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Знайти рівняння регресії адекватне об'єкту.

#### Завдання на лабораторну роботу:

1. Ознайомитися з теоретичними відомостями.
2. Вибрати з таблиці варіантів і записати в протокол інтервали значень  $x_1, x_2, x_3$ . Обчислити і записати значення, відповідні кодованим значенням факторів  $+1; -1; +\frac{1}{2}; -\frac{1}{2}; 0$  для  $\bar{x}_1, \bar{x}_2, \bar{x}_3$ .
3. Значення функції відгуку знайти за допомогою підстановки в формулу:

$$y_i = f(x_1, x_2, x_3) + \text{random}(10) - 5,$$

де  $f(x_1, x_2, x_3)$  вибирається по номеру в списку в журналі викладача.

4. Провести експерименти і аналізуючи значення статистичних перевірок, отримати адекватну модель рівняння регресії. При розрахунках використовувати натуральні значення факторів.
5. Зробити висновки по виконаній роботі.

#### Варіант завдання:

211	10	60	-35	15	10	15	$3,8+6,4*x_1+4,8*x_2+6,9*x_3+9,0*x_1*x_1+0,2*x_2*x_2+5,2*x_3*x_3+2,6*x_1*x_2+1,0*x_1*x_3+0,6*x_2*x_3+1,8*x_1*x_2*x_3$
-----	----	----	-----	----	----	----	---

#### Код програми:

```
import random as r
import numpy as np
import pprint
from scipy.stats import t, f
import sklearn.linear_model as lm
from functools import partial
from prettytable import PrettyTable

table0 = PrettyTable()
table0.field_names = (["Студент", "Група"])
name = "Карнаухова Анастасія"
group = "IO-92"
table0.add_row([name, group])
print(table0)

x_range = [[10, 60], [-35, 15], [10, 15]]
x_sered_max = sum([x[1] for x in x_range]) / 3
x_sered_min = sum([x[0] for x in x_range]) / 3

x01 = (x_range[0][1] - x_range[0][0]) / 2
x02 = (x_range[1][1] - x_range[1][0]) / 2
x03 = (x_range[2][1] - x_range[2][0]) / 2
delta_x1 = x_range[0][1] - x01
delta_x2 = x_range[1][1] - x02
delta_x3 = x_range[2][1] - x03
```

```
y_max = 200 + x_sered_max
y_min = 200 + x_sered_min
```

```
def create_plan_matrix(n, m):
```

```
    x_matrix_norm = [
        [1, -1, -1, -1, 1, 1, 1, -1, 1, 1, 1],
        [1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1],
        [1, -1, 1, -1, -1, 1, -1, 1, 1, 1, 1],
        [1, -1, 1, 1, -1, -1, 1, -1, 1, 1, 1],
        [1, 1, -1, -1, -1, -1, 1, 1, 1, 1, 1],
        [1, 1, -1, 1, -1, 1, -1, -1, 1, 1, 1],
        [1, 1, 1, -1, 1, -1, -1, -1, 1, 1, 1],
        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
        [1, -1.73, 0, 0, 0, 0, 0, 0, 2,9929, 0, 0],
        [1, 1.73, 0, 0, 0, 0, 0, 0, 2,9929, 0, 0],
        [1, 0, -1.73, 0, 0, 0, 0, 0, 0, 2,9929, 0],
        [1, 0, 1.73, 0, 0, 0, 0, 0, 0, 2,9929, 0],
        [1, 0, 0, -1.73, 0, 0, 0, 0, 0, 0, 2,9929],
        [1, 0, 0, 1.73, 0, 0, 0, 0, 0, 0, 2,9929],
        [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    ]
```

```
    print('\nНормована матриця:')
    pprint.pprint(x_matrix_norm)
```

```
    x_matrix = [[] for x in range(n)]
```

```
    for i in range(len(x_matrix)):
```

```
        if i < 8:
```

```
            x1 = x_range[0][0] if x_matrix_norm[i][1] == -1 else x_range[0][1]
```

```
            x2 = x_range[1][0] if x_matrix_norm[i][2] == -1 else x_range[1][1]
```

```
            x3 = x_range[2][0] if x_matrix_norm[i][3] == -1 else x_range[2][1]
```

```
        else:
```

```
            x1 = x_matrix_norm[i][1] * delta_x1 + x01
```

```
            x2 = x_matrix_norm[i][2] * delta_x2 + x02
```

```
            x3 = x_matrix_norm[i][3] * delta_x3 + x03
```

```
        x_matrix[i] = [1, float(format(x1, '.2f')),
                       float(format(x2, '.2f')),
                       float(format(x3, '.2f')),
                       float(format(x1 * x2, '.2f')),
                       float(format(x1 * x3, '.2f')),
                       float(format(x2 * x3, '.2f')),
                       float(format(x1 * x2 * x3, '.2f')),
                       float(format(x1 ** 2, '.2f')),
                       float(format(x2 ** 2, '.2f')),
                       float(format(x3 ** 2, '.2f'))]
```

```
    print('\nНатуралізована матриця: ')
    pprint.pprint(x_matrix)
```

```
    y = np.zeros(shape=(n, m))
```

```
    for i in range(n):
```

```
        for j in range(m):
```

```
            y[i][j] = count_y(x_matrix[i])
```

```
    print("y =
```

```
3,8+6,4*x1+4,8*x2+6,9*x3+9,0*x1*x1+0,2*x2*x2+5,2*x3*x3+2,6*x1*x2+1,0*x1*x3+0,6*x2*x3+1,8*x1*x2
*x3 + random(10) - 5")
```

```
    print('Y:')
    pprint.pprint(y)
```

```
    y_avr = np.zeros(n)
```

```

for i in range(len(y)):
    for j in range(len(y[0])):
        y_avr[i] += y[i][j] / m
return [x_matrix_norm, x_matrix, y, y_avr]

def find_coefs(x, y):
    skm = lm.LinearRegression(fit_intercept=False) # знаходимо коефіцієнти рівняння регресії
    skm.fit(x, y)
    B = skm.coef_
    print('Коефіцієнти: ')
    print(B)
    return B

def count_y(x_arr):
    return 3.8 + 6.4 * x_arr[1] + 4.8 * x_arr[2] + 6.9 * x_arr[3] + 9.0 * x_arr[8] + 0.2 * x_arr[9] + 5.2 *
x_arr[10] + 2.6 * x_arr[4] + 1.0 * x_arr[5] + 0.6 * x_arr[6] + 1.8 * x_arr[7] + r.randint(0, 10) - 5

def perevirka(x, y, b):
    y_pract = np.zeros(len(y))
    for i in range(len(x)):
        for j in range(len(x[0])):
            y_pract[i] += b[j] * x[i][j]
    print("\nПеревірка:")
    print("\ny - real :", y)
    print('\ny - found:', y_pract)

def get_new_y(x, b):
    y_pract = np.zeros(len(y))
    for i in range(len(x)):
        for j in range(len(x[0])):
            y_pract[i] += b[j] * x[i][j]
    return y_pract

def get_cohren_critical(prob, f1, f2):
    f_crit = f.isf((1 - prob) / f2, f1, (f2 - 1) * f1)
    return f_crit / (f_crit + f2 - 1)

def cohren_crit(y, n, m):
    y_var = [np.var(i) for i in y]
    Gp = max(y_var) / sum(y_var)
    Gt = get_cohren_critical(0.95, m - 1, n)
    if (Gp < Gt):
        print("\nДисперсії однорідні")
        return True
    else:
        print("\nДисперсії не однорідні")
        return False

fisher_teor = partial(f.ppf, q=1 - 0.05)

student_teor = partial(t.ppf, q=1 - 0.025)

```

```

def kritერიy_studenta(x, y, y_aver, n, m, B):
    d = 0
    y_var = [np.var(i) for i in y]
    s_kv_aver = sum(y_var) / n
    s_aver = (s_kv_aver / (n * m)) ** 0.5
    b = np.zeros(len(x[0]))
    for i in range(len(x[0])):
        for j in range(len(y_aver)):
            b[i] += y_aver[j] * x[j][i] / n
    ts = []
    for bi in b:
        ts.append(abs(bi) / s_aver)
    Stud_teor = student_teor(df=(m - 1) * n)
    for i in range(len(ts)):
        if ts[i] < Stud_teor:
            B[i] = 0
        else:
            d += 1
    print("\nКоефіцієнти після перевірки нуль гіпотези: ")
    print(B)
    return [B, d]

def критерий_fishera(m, n, d, new_y_pract, y_avr, y):
    f4 = n - d
    f3 = (m - 1) * n
    y_var = [np.var(i) for i in y]
    Sa = (sum(y_var) / n)
    Sad = (m / (n - d)) * sum([(new_y_pract[i] - y_avr[i]) ** 2 for i in range(len(y_avr))])
    pract = Sad / Sa
    teor = fisher_teor(dfn=f4, dfd=f3)
    if pract > teor:
        print("\nПрактичне значення:", pract)
        print("Теоретичне значення:", teor)
        print("\nРівняння регресії неадекватне")
        return [False, False]
    else:
        print("\nРівняння регресії адекватне")
        return [True, True]

if __name__ == "__main__":
    odnorid = False
    adekvat = False
    n = 15
    m = 3
    while not adekvat:
        while not odnorid:
            x_matrix_norm, x_matrix, y, y_avr = create_plan_matrix(n, m)
            odnorid = cohren_crit(y, n, m)
            if odnorid == False:
                m += 1
        B = find_coefs(x_matrix, y_avr)
        perevirka(x_matrix, y_avr, B)
        new_B, d = критерий_studenta(x_matrix_norm, y, y_avr, n, m, B)
        new_y_pract = get_new_y(x_matrix, new_B)
        adekvat, odnorid = критерий_fishera(m, n, 4, new_y_pract, y_avr, y)

```

## Результат роботи програми:

```
+-----+-----+
|      Студент      | Группа |
+-----+-----+
| Карнаухова Анастасія | ІО-92  |
+-----+-----+

Нормована матриця:
[[1, -1, -1, -1, 1, 1, 1, -1, 1, 1, 1],
 [1, -1, -1, 1, 1, 1, -1, -1, 1, 1, 1],
 [1, -1, 1, -1, -1, 1, -1, 1, 1, 1, 1],
 [1, -1, 1, 1, -1, -1, 1, -1, 1, 1, 1],
 [1, 1, -1, -1, -1, -1, 1, 1, 1, 1, 1],
 [1, 1, -1, 1, -1, 1, -1, -1, 1, 1, 1],
 [1, 1, 1, -1, 1, -1, -1, -1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 [1, -1.73, 0, 0, 0, 0, 0, 0, 2, 9929, 0, 0],
 [1, 1.73, 0, 0, 0, 0, 0, 0, 2, 9929, 0, 0],
 [1, 0, -1.73, 0, 0, 0, 0, 0, 0, 2, 9929, 0],
 [1, 0, 1.73, 0, 0, 0, 0, 0, 0, 2, 9929, 0],
 [1, 0, 0, -1.73, 0, 0, 0, 0, 0, 0, 2, 9929],
 [1, 0, 0, 1.73, 0, 0, 0, 0, 0, 0, 2, 9929],
 [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

Натуралізована матриця:
[[1, 10.0, -35.0, 10.0, -350.0, 100.0, -350.0, -3500.0, 100.0, 1225.0, 100.0],
 [1, 10.0, -35.0, 15.0, -350.0, 150.0, -525.0, -5250.0, 100.0, 1225.0, 225.0],
 [1, 10.0, 15.0, 10.0, 150.0, 100.0, 150.0, 1500.0, 100.0, 225.0, 100.0],
 [1, 10.0, 15.0, 15.0, 150.0, 150.0, 225.0, 2250.0, 100.0, 225.0, 225.0],
 [1,
 60.0,
 ,
 -35.0,
 10.0,
 -2100.0,
 600.0,
 -350.0,
 -21000.0,
 3600.0,
 1225.0,
 100.0],
 [1,
 60.0,
 -35.0,
 15.0,
 -2100.0,
 900.0,
 -525.0,
 -31500.0,
 3600.0,
 1225.0,
 225.0],
 [1, 60.0, 15.0, 10.0, 900.0, 600.0, 150.0, 9000.0, 3600.0, 225.0, 100.0],
 [1, 60.0, 15.0, 15.0, 900.0, 900.0, 225.0, 13500.0, 3600.0, 225.0, 225.0],
 [1, -35.55, 25.0, 2.5, -888.75, -88.88, 62.5, -2221.87, 1263.8, 625.0, 6.25],
 [1, 85.55, 25.0, 2.5, 2138.75, 213.88, 62.5, 5346.88, 7318.8, 625.0, 6.25],
 [1, 25.0, 42.3, 2.5, 1057.5, 62.5, 105.75, 2643.75, 625.0, 1789.29, 6.25],
 [1, 25.0, 7.7, 2.5, 192.5, 62.5, 19.25, 481.25, 625.0, 59.29, 6.25],
 [1,
 25.0,
 25.0,
 -19.12,
```

```

625.0,
-478.12,
-478.12,
-11953.12,
625.0,
625.0,
365.77],
[1, 25.0, 25.0, 24.12, 625.0, 603.12, 603.12, 15078.12, 625.0, 625.0, 582.02],
[1, 25.0, 25.0, 2.5, 625.0, 62.5, 62.5, 1562.5, 625.0, 625.0, 6.25]]
y = 3,8+6,4*x1+4,8*x2+6,9*x3+9,0*x1*x1+0,2*x2*x2+5,2*x3*x3+2,6*x1*x2+1,0*x1*x3+0,6*x2*x3+1,8*x1*x2*x3 + random(10) - 5
Y :
array([[ -5686.2 , -5681.2 , -5690.2 ],
       [ -8201.7 , -8209.7 , -8211.7 ],
       [  4957.8 ,  4948.8 ,  4948.8 ],
       [  7080.3 ,  7079.3 ,  7087.3 ],
       [ -9412.2 , -9412.2 , -9419.2 ],
       [-27437.7 , -27434.7 , -27432.7 ],
       [ 52723.8 , 52722.8 , 52728.8 ],
       [ 61854.3 , 61853.3 , 61852.3 ],
       [  5078.734,  5084.734,  5078.734],
       [ 82146.784, 82150.784, 82153.784],
       [ 14031.648, 14034.648, 14036.648],
       [  7328.168,  7323.168,  7325.168],
       [-12850.732, -12851.732, -12849.732],
       [ 38960.34 , 38953.34 , 38958.34 ],
       [ 10625.05 , 10625.05 , 10616.05 ]])

Дисперсії однорідні
Коефіцієнти:
[2.07737148  6.44359002  4.8043026  6.79835228  2.59952106  1.002351
 0.59989815  1.80003398  8.99953049  0.20137318  5.20180627]

Перевірка:

y - real : [ -5685.86666667  -8207.7          4951.8          7082.3
 -9414.53333333 -27435.03333333  52725.13333333  61853.3
 5080.734        82150.45066667  14034.31466667  7325.50133333
 -12850.732       38957.34        10622.05        ]

y - found: [ -5686.51853606  -8207.22508338  4952.20292799  7081.55587054
 -9414.56332419 -27434.97945416  52723.81031518  61853.87843767
 5080.72618831  82150.73107815  14034.83738555  7327.18253125
 -12850.86921692  38957.54856606  10620.74098069]

Коефіцієнти після перевірки нуля гіпотези:
[2.07737148  6.44359002  4.8043026  6.79835228  2.59952106  1.002351
 0.59989815  1.80003398  8.99953049  0.20137318  5.20180627]

Рівняння регресії адекватне

```

## Висновок:

Проробивши лабораторну роботу, було проведено трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план.. Та знайдено рівняння регресії адекватне об'єкту. У ході виконання лабораторної роботи проблем не виникло. Результати виконання лабораторної висвітлені на роздруківках.