

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

Лабораторна робота 4  
з дисципліни «Методи оптимізації та планування експерименту»

Виконав:  
Студент 2 курсу ФІОТ  
групи ІВ-93  
Трибушенко Артем

Перевірив:  
Регіда П.Г.

Київ 2021

**Мета:** Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

**Завдання на лабораторну роботу:**

1. Скласти матрицю планування для повного трьохфакторного експерименту.
2. Провести експеримент, повторивши N раз досліди у всіх точках факторного простору і знайти значення відгуку Y. Знайти значення Y шляхом моделювання випадкових чисел у певному діапазоні відповідно варіанту. Варіанти вибираються за номером в списку в журналі викладача.

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

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

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

3. Знайти коефіцієнти рівняння регресії і записати його.
4. Провести 3 статистичні перевірки – за критеріями Кохрена, Стюдента, Фішера.
5. Зробити висновки по адекватності регресії та значимості окремих коефіцієнтів і записати скореговане рівняння регресії.
6. Написати комп'ютерну програму, яка усе це моделює.

**Варіант: 327**

327	-25	-5	-15	35	-5	60
-----	-----	----	-----	----	----	----

**Роздруківка програми:**

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

# передача m як параметра функції(залежність від m)
def lab4(m, n):
    # варіант 327
    x1min = -25
    x1max = -5
    x2min = -15
    x2max = 35
    x3min = -5
```

```
x3max = 60
```

```
# максимальне та мінімальне значення
```

```
y_max = 200 + (x1max + x2max + x3max) / 3
```

```
y_min = 200 + (x1min + x2min + x3min) / 3
```

```
# матриця ПФЕ
```

```
xn = [[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]]
```

```
x1x2_norm, x1x3_norm, x2x3_norm, x1x2x3_norm = [0] * n, [0] * n, [0] * n, [0] *  
n
```

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

```
    x1x2_norm[i] = xn[1][i] * xn[2][i]
```

```
    x1x3_norm[i] = xn[1][i] * xn[3][i]
```

```
    x2x3_norm[i] = xn[2][i] * xn[3][i]
```

```
    x1x2x3_norm[i] = xn[1][i] * xn[2][i] * xn[3][i]
```

```
# заповнення y(генерація)
```

```
y1 = [random.randint(int(y_min), int(y_max)) for i in range(n)]
```

```
y2 = [random.randint(int(y_min), int(y_max)) for i in range(n)]
```

```
y3 = [random.randint(int(y_min), int(y_max)) for i in range(n)]
```

```
# матриця планування
```

```
y_matrix = [[y1[0], y2[0], y3[0]],  
            [y1[1], y2[1], y3[1]],  
            [y1[2], y2[2], y3[2]],  
            [y1[3], y2[3], y3[3]],  
            [y1[4], y2[4], y3[4]],  
            [y1[5], y2[5], y3[5]],  
            [y1[6], y2[6], y3[6]],  
            [y1[7], y2[7], y3[7]]]
```

```
# вивід даних за допомогою цикла
```

```
print("Матриця планування y : \n")
```

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

```
    print(y_matrix[i])
```

```
x0 = [1, 1, 1, 1, 1, 1, 1, 1]
```

```
# заміна -1 на x1_мін, 1 на x1_макс
```

```
x1 = [-25, -25, -5, -5, -25, -25, -5, -5]
```

```

# заміна -1 на x2_мін, 1 на x2_макс
x2 = [-15, 35, -15, 35, -15, 35, -15, 35]

# заміна -1 на x3_мін, 1 на x3_макс
x3 = [-5, 60, 60, -5, 60, -5, -5, 60]

# заповнення нулями x1x2, x1x3, x1x2x3
x1x2, x1x3, x2x3, x1x2x3 = [0] * n, [0] * n, [0] * n, [0] * n
# заповнення x1x2, x1x3, x1x2x3 добутками
for i in range(n):
    x1x2[i] = x1[i] * x2[i]
    x1x3[i] = x1[i] * x3[i]
    x2x3[i] = x2[i] * x3[i]
    x1x2x3[i] = x1[i] * x2[i] * x3[i]
# середні y
Y_average = []
for i in range(len(y_matrix)):
    Y_average.append(np.mean(y_matrix[i], axis=0))

# формуємо списки b і a
list_for_b = [xn[0], xn[1], xn[2], xn[3], x1x2_norm, x1x3_norm, x2x3_norm,
x1x2x3_norm]
list_for_a = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3))

# вивід матриці планування X
print("Матриця планування X:")
for i in range(n):
    print(list_for_a[i])
# нормовані фактори b_i
bi = []
for k in range(n):
    S = 0
    for i in range(n):
        S += (list_for_b[k][i] * Y_average[i]) / n
    bi.append(round(S, 3))
# розрахунок ai(система рівнянь) через функцію solve, вивід рівняння регресії
ai = [round(i, m) for i in solve(list_for_a, Y_average)]
print("Рівняння регресії: \n" "y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1x2 +
{}*x1x3 + {}*x2x3 "
      "+ {}*x1x2x3".format(ai[0],ai[1],ai[2],ai[3],ai[4],ai[5],ai[6],ai[7]))
# вивід даних
print("Рівняння регресії для нормованих факторів: \n" "y = {} + {}*x1 + {}*x2
+ {}*x3 + {}*x1x2 + {}*x1x3 +
" {}*x2x3 + {}*x1x2x3".format(bi[0], bi[1], bi[2], bi[3], bi[4], bi[5], bi[6],

```

```

bi[7]))

print("Перевірка за критерієм Кохрена")
print("Середні значення відгуку за рядками:", "\n", +Y_average[0],
Y_average[1], Y_average[2], Y_average[3],
      Y_average[4], Y_average[5], Y_average[6], Y_average[7])
# розрахунок дисперсій
dispersions = []
for i in range(len(y_matrix)):
    a = 0
    for k in y_matrix[i]:
        a += (k - np.mean(y_matrix[i], axis=0)) ** 2
    dispersions.append(a / len(y_matrix[i]))
# експериментально
Gp = max(dispersions) / sum(dispersions)
# теоретично
Gt = 0.5157
# перевірка однорідності дисперсій
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")
# критерій Стьюдента
print("Перевірка значущості коефіцієнтів за критерієм Стьюдента")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (n * m)) ** 0.5

t_list = [abs(bi[i]) / sbs for i in range(0, n)]

d = 0
res = [0] * n
coef_1 = []
coef_2 = []
F3 = (m - 1) * n
# перевірка значущості коефіцієнтів(scipy)
for i in range(n):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coef_2.append(bi[i])
        res[i] = 0
    else:
        coef_1.append(bi[i])
        res[i] = bi[i]
        d += 1

# вивід

```

```

print("Значущі коефіцієнти регресії:", coef_1)
print("Незначущі коефіцієнти регресії:", coef_2)

# значення y з коефіцієнтами регресії
y_st = []
for i in range(n):
    y_st.append(res[0] + res[1] * xn[1][i] + res[2] * xn[2][i] + res[3] * xn[3][i] +
res[4] * x1x2_norm[i] \
                + res[5] * x1x3_norm[i] + res[6] * x2x3_norm[i] + res[7] *
x1x2x3_norm[i])
    print("Значення з отриманими коефіцієнтами:\n", y_st)

# критерій Фішера
print("\nПеревірка адекватності за критерієм Фішера\n")
Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(n)]) / (n - d)
Fp = Sad / sb
F4 = n - d
# перевірка за допомогою scipy
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("Рівняння регресії адекватне при рівні значимості 0.05")
else:
    print("Рівняння регресії неадекватне при рівні значимості 0.05")

lab4(3, 8)

```

## Результати роботи програми:

Матриця планування у :

[222, 222, 203]  
[201, 230, 209]  
[193, 226, 224]  
[191, 230, 210]  
[225, 213, 200]  
[211, 229, 227]  
[224, 187, 219]  
[190, 212, 224]

Матриця планування X:

(1, -25, -15, -5, 375, 125, 75, -1875)  
(1, -25, 35, 60, -875, -1500, 2100, -52500)  
(1, -5, -15, 60, 75, -300, -900, 4500)  
(1, -5, 35, -5, -175, 25, -175, 875)  
(1, -25, -15, 60, 375, -1500, -900, 22500)  
(1, -25, 35, -5, -875, 125, -175, 4375)  
(1, -5, -15, -5, 75, 25, 75, -375)  
(1, -5, 35, 60, -175, -300, 2100, -10500)

Рівняння регресії:

$y = 208.544 + -0.35 \cdot x_1 + -0.034 \cdot x_2 + 0.067 \cdot x_3 + -0.006 \cdot x_1 x_2 + 0.006 \cdot x_1 x_3 + -0.002 \cdot x_2 x_3 + -0.0 \cdot x_1 x_2 x_3$

Рівняння регресії для нормованих факторів:

$y = 213.417 + -2.583 \cdot x_1 + 0.25 \cdot x_2 + -1.167 \cdot x_3 + -1.583 \cdot x_1 x_2 + 1.833 \cdot x_1 x_3 + -1.5 \cdot x_2 x_3 + -0.0 \cdot x_1 x_2 x_3$

Перевірка за критерієм Кохрена

Середні значення відгуку за рядками:

215.66666666666666 213.33333333333334 214.33333333333334 210.33333333333334 212.66666666666666 222.33333333333334 210.0 208.66666666666666

Дисперсія однорідна

Перевірка значущості коефіцієнтів за критерієм Стьюдента

Значущі коефіцієнти регресії: [213.417]

Незначущі коефіцієнти регресії: [-2.583, 0.25, -1.167, -1.583, 1.833, -1.5, -0.0]

Значення з отриманими коефіцієнтами:

[213.417, 213.417, 213.417, 213.417, 213.417, 213.417, 213.417]

Перевірка адекватності за критерієм Фішера

Рівняння регресії адекватне при рівні значимості 0.05

## **Висновки**

Проведений повний трьохфакторний експеримент. Знайдено рівняння регресії адекватне об'єкту. Складено матрицю планування для повного трьохфакторного експерименту, знайдено значення відгуку  $Y$ . Знайдено значення  $Y$  шляхом моделювання випадкових чисел у певному діапазоні відповідно варіанту. Знайдено коефіцієнти рівняння регресії. Проведені 3 статистичні перевірки – за критеріями Кохрена, Стюдента, Фішера. На екран виведені висновки по адекватності регресії та значимості окремих коефіцієнтів і скореговане рівняння регресії.

## **Контрольні запитання**

1. Що називається дробовим факторним експериментом?

Дробовим факторним експериментом називається експеримент з використанням частини повного факторного експерименту

2. Для чого потрібно розрахункове значення Кохрена?

Розрахункове значення Кохрена використовують для перевірки однорідності дисперсій.

3. Для чого перевіряється критерій Стюдента?

За допомогою критерію Стюдента перевіряється значущість коефіцієнтів рівняння

4. Чим визначається критерій Фішера і як його застосовувати?

Критерій Фішера використовують при перевірці отриманого рівняння регресії досліджуваного об'єкта.