

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

Лабораторна робота №3
З дисципліни «Методи наукових досліджень»
За темою:
«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ
ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАВ:
Студент II курсу ФІОТ
Групи ІВ-93
Дромашко Артем
Номер у списку – 6
Варіант - 306

ПЕРЕВІРИВ:
асистент
Регіда П.Г.

Київ 2021 р.

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

Завдання:

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.

№ варіанта	X ₁		X ₂		X ₃	
	min	max	min	max	min	max
306	10	40	15	50	10	30

Програмний код

```
from random import randint
from numpy.linalg import det
from functools import reduce

def naturalize(matrix_of_plan, min_max_arr):
    result = []
    for i in matrix_of_plan:
        result.append(min_max_arr[1]) if i == 1 else result.append(min_max_arr[0])
    return result

M = 4
x1 = [10, 40]
x2 = [15, 50]
x3 = [10, 30]
print(f'x1_min = {x1[0]}, x1_max = {x1[1]}')
print(f'x2_min = {x2[0]}, x2_max = {x2[1]}')
print(f'x3_min = {x3[0]}, x3_max = {x3[1]}')

# -----Матриця планування експерименту з +1,-1-----
x0_p1 = [1, 1, 1, 1]
x1_p1 = [-1, -1, 1, 1]
x2_p1 = [-1, 1, -1, 1]
x3_p1 = [-1 * (x1_p1[i] * x2_p1[i]) for i in range(len(x1_p1))]
print('x0:', x0_p1)
```

```

print('x1:', x1_p1)
print('x2:', x2_p1)
print('x3:', x3_p1)

# -----Матриця планування з натуралізованими значеннями
факторів-----
x1_p2 = naturalize(x1_p1, x1)
x2_p2 = naturalize(x2_p1, x2)
x3_p2 = naturalize(x3_p1, x3)
print('\nx1:', x1_p2)
print('x2:', x2_p2)
print('x3:', x3_p2)

xAverage_max = (max(x1_p2) + max(x2_p2) + max(x3_p2)) / 3
xAverage_min = (min(x1_p2) + min(x2_p2) + min(x3_p2)) / 3
print(f'\nx_avg_max = {xAverage_max}')
print(f'x_avg_min = {xAverage_min}')

# -----Діапазон y-----
-----
y_Max = int(200 + xAverage_max)
y_Min = int(200 + xAverage_min)
print(f'\ny_max = {y_Max}')
print(f'y_min = {y_Min}')

y1 = [randint(y_Min, y_Max) for _ in range(4)]
y2 = [randint(y_Min, y_Max) for _ in range(4)]
y3 = [randint(y_Min, y_Max) for _ in range(4)]
print('y1:', y1)
print('y2:', y2)
print('y3:', y3)

y_avg_arr = [(y1[i] + y2[i] + y3[i]) / 3 for i in range(4)]
print('y average:', y_avg_arr)

# -----Математичне очікування-----
-----
mx1 = reduce(lambda a, b: a + b, x1_p2) / 4
mx2 = reduce(lambda a, b: a + b, x2_p2) / 4
mx3 = reduce(lambda a, b: a + b, x3_p2) / 4
my = reduce(lambda a, b: a + b, y_avg_arr) / 4
print(f'\nmx1 = {mx1}')
print(f'mx2 = {mx2}')
print(f'mx3 = {mx3}')
print(f'my = {my}')

a1 = sum([x1_p2[i] * y_avg_arr[i] for i in range(4)]) / 4
a2 = sum([x2_p2[i] * y_avg_arr[i] for i in range(4)]) / 4
a3 = sum([x3_p2[i] * y_avg_arr[i] for i in range(4)]) / 4
print(f'\na1 = {a1}')
print(f'a2 = {a2}')
print(f'a3 = {a3}')

a11 = sum([i * i for i in x1_p2]) / 4
a22 = sum([i * i for i in x2_p2]) / 4
a33 = sum([i * i for i in x3_p2]) / 4
print(f'\na11 = {a11}')
print(f'a22 = {a22}')
print(f'a33 = {a33}')

a12 = sum([x1_p2[i] * x2_p2[i] for i in range(4)]) / 4
a13 = sum([x1_p2[i] * x3_p2[i] for i in range(4)]) / 4

```

```

a23 = sum([x2_p2[i] * x3_p2[i] for i in range(4)]) / 4
a21 = a12
a31 = a13
a32 = a23
print(f'\na12 = {a12}')
print(f'a13 = {a13}')
print(f'a23 = {a23}')
print(f'a21 = {a21}')
print(f'a31 = {a31}')
print(f'a32 = {a32}')

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]])

print(f'y = {b0} + {b1}*x1 + {b2}*x2 + {b3}*x3')

for i in range(4):
    y = b0 + b1 * x1_p2[i] + b2 * x2_p2[i] + b3 * x3_p2[i]
    print('y =', y)

# -----Перевірка однорідності дисперсії за критерієм
# Кохрена-----
dispersion = [(y1[i] - y_avg_arr[i]) ** 2 + (y2[i] - y_avg_arr[i]) ** 2 + (y3[i] -
y_avg_arr[i]) ** 2) / 3 for i in
              range(4)]
print('dispersion:', dispersion)

gp = max(dispersion) / sum(dispersion)
print('Gp =', gp)

# -----Рівень значимості q = 0.05; f1 = m - 1 = 2; f2 = N
# = 4-----
# Gт = 0.7679
if gp < 0.7679:
    print('Дисперсія однорідна')
# -----Оцінка значимості коефіцієнтів регресії згідно критерію

```

Студента-----

```
s2b = sum(dispersion) / 4
s2bs_avg = s2b / 4 * M
sb = s2bs_avg ** (1 / 2)
```

```
beta0 = sum([y_avg_arr[i] * x0_p1[i] for i in range(4)]) / 4
beta1 = sum([y_avg_arr[i] * x1_p1[i] for i in range(4)]) / 4
beta2 = sum([y_avg_arr[i] * x2_p1[i] for i in range(4)]) / 4
beta3 = sum([y_avg_arr[i] * x3_p1[i] for i in range(4)]) / 4
```

```
beta_arr = [beta0, beta1, beta2, beta3]
print('beta:', beta_arr)
t_arr = [abs(beta_arr[i]) / sb for i in range(4)]
print('tetta:', t_arr)
```

-----f3 = f1*f2-----

```
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')
```

```
b_list = [b0, b1, b2, b3]
print(f'y = b{indexes[0]}')
```

```
b_res = [b_list[indexes[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([(y_avg_arr[i] - b_res[i]) ** 2 for i in range(4)]) / 4 - d
fp = s2_ad / s2b
print(f'Fp = {fp}')
```

```
# Fт = 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')
```

```
else:
    print('Дисперсія неоднорідна')
```

Результати роботи програми

```
x1_min = 10, x1_max = 40
x2_min = 15, x2_max = 50
x3_min = 10, x3_max = 30
x0: [1, 1, 1, 1]
x1: [-1, -1, 1, 1]
x2: [-1, 1, -1, 1]
x3: [-1, 1, 1, -1]

x1: [10, 10, 40, 40]
x2: [15, 50, 15, 50]
x3: [10, 30, 30, 10]

x_avg_max = 40.0
x_avg_min = 11.666666666666666

y_max = 240
y_min = 211
y1: [228, 216, 240, 236]
y2: [222, 220, 224, 238]
y3: [240, 234, 229, 216]
y average: [230.0, 223.33333333333334, 231.0, 230.0]

mx1 = 25.0
mx2 = 32.5
mx3 = 20.0
my = 228.58333333333334

a11 = 850.0
a22 = 1362.5
a33 = 500.0

a12 = 812.5
a13 = 500.0
a23 = 650.0
a21 = 812.5
a31 = 500.0
a32 = 650.0
y = 231.78174603174654 + 0.127777777777778004*x1 + -0.10952380952380829*x2 + -0.14166666666666947*x3
y = 230.00000000000054
y = 223.33333333333383
y = 231.00000000000054
y = 230.00000000000063
```

```
dispersion: [56.0, 59.55555555555555, 44.666666666666664, 98.66666666666667]
Gr = 0.38111587982832623
Дисперсія однорідна
beta: [228.58333333333334, 1.9166666666666572, -1.916666666666643, -1.416666666666643]
t: [28.41304972807012, 0.23824285225869832, 0.2382428522586992, 0.17609254297382107]
Коефіцієнт b1 = 0.23824285225869832 приймаємо не значним
Коефіцієнт b2 = 0.2382428522586992 приймаємо не значним
Коефіцієнт b3 = 0.17609254297382107 приймаємо не значним
y = b0
y = 231.78174603174654
y = 231.78174603174654
y = 231.78174603174654
y = 231.78174603174654
Fr = 1.1948916333345032
Рівняння регресії адекватно оригіналу при рівні значимості 0.05
```

Контрольні запитання:

1. Дробовий факторний експеримент – частина ПФЕ, який мінімізує число дослідів, за рахунок тієї інформації, яка не дуже істотна для побудови моделі
2. Значення Кохрена використовують для перевірки однорідності дисперсії
3. Критерій Стюдента перевіряє значущість коефіцієнтів рівняння
4. Критерій Фішера використовують при перевірці отриманого рівняння регресії досліджуваному об'єкту