

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 оптимізації та планування експерименту»
з теми «Проведення трьохфакторного експерименту з використанням
лінійного рівняння регресії»

Виконав:
студент II курсу ФІОТ
групи ІВ-93
Підгайний Данило
Варіант: 318

Перевірив:
Регіда П. Г.

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

Варіант

№_варіанта	X ₁		X ₂		X ₃	
	min	max	min	max	min	max
318	20	70	-15	45	20	35

Код програми

```
// Варіант 318

const X1_MIN = 20
const X1_MAX = 70
const X2_MIN = -15
const X2_MAX = 45
const X3_MIN = 20
const X3_MAX = 35

// <----- корисні функції ----->

const average = arr => arr.reduce((a, b) => a + b, 0) / arr.length;
const Xcp_max = average( arr: [X1_MAX, X2_MAX, X3_MAX])
const Xcp_min = average( arr: [X1_MIN, X2_MIN, X3_MIN])

const y_rand = () => parseInt(Math.random() * (Xcp_max - Xcp_min), 10) + 200;
const sum = arr => arr.reduce((a, b) => a + b, 0)

// детермінант
const det = m =>
  m.length === 1 ?
    m[0][0] :
    m.length === 2 ?
      m[0][0] * m[1][1] - m[0][1] * m[1][0] :
      m[0].reduce((r, e, i) =>
        r + (-1) ** (i + 2) * e * det(m.slice(1).map(c =>
          c.filter((_, j) => i !== j))), 0)
```

```
// <----- таблиця ----->
let Tab = {
  tab: [
    ["x1", "x2", "x3", "Y1", "Y2", "Y3"],
    [X1_MIN, X2_MIN, X3_MIN, y_rand(), y_rand(), y_rand()],
    [X1_MIN, X2_MAX, X3_MAX, y_rand(), y_rand(), y_rand()],
    [X1_MAX, X2_MIN, X3_MAX, y_rand(), y_rand(), y_rand()],
    [X1_MAX, X2_MAX, X3_MIN, y_rand(), y_rand(), y_rand()],
  ],
  get_x_col: n => Tab.tab.slice(1).map(arr => arr[n - 1]),
  get_x_line: n => Tab.tab[n].slice(0, 3),
  get_y_col: n => Tab.tab.slice(1).map(arr => arr[n + 2]),
  get_y_line: n => Tab.tab[n].slice(3)
}

console.table(Tab.tab)
```

```
// Знайдемо середні значення функції відгуку за рядками:
let y_aver = [1, 2, 3, 4].map(i => average(Tab.get_y_line(i)))
let mx1 = average(Tab.get_x_col(n: 1))
let mx2 = average(Tab.get_x_col(n: 2))
let mx3 = average(Tab.get_x_col(n: 3))

let my = average(y_aver)

let a1 = average(Tab.get_x_col(n: 1).map((x, i) => x * y_aver[i]))
let a2 = average(Tab.get_x_col(n: 2).map((x, i) => x * y_aver[i]))
let a3 = average(Tab.get_x_col(n: 3).map((x, i) => x * y_aver[i]))

let a11 = average(Tab.get_x_col(n: 1).map(x => x * x))
let a22 = average(Tab.get_x_col(n: 2).map(x => x * x))
let a33 = average(Tab.get_x_col(n: 3).map(x => x * x))

let a12 = average(Tab.get_x_col(n: 1).map((x, i) => Tab.get_x_col(n: 2)[i]))
let a13 = average(Tab.get_x_col(n: 1).map((x, i) => Tab.get_x_col(n: 3)[i]))
let a23 = average(Tab.get_x_col(n: 2).map((x, i) => Tab.get_x_col(n: 3)[i]))

let denominator = det(m: [
  [1, mx1, mx2, mx3],
  [mx1, a11, a12, a13],
  [mx2, a12, a22, a23],
  [mx3, a13, a23, a33]
])
```

```

let B = {
  b0 : det( m: [
    [my, mx1, mx2, mx3],
    [a1, a11, a12, a13],
    [a2, a12, a22, a23],
    [a3, a13, a23, a33]
  ]) / denominator,

  b1 : det( m: [
    [1, my, mx2, mx3],
    [mx1, a1, a12, a13],
    [mx2, a2, a22, a23],
    [mx3, a3, a23, a33]
  ]) / denominator,

  b2 : det( m: [
    [1, mx1, my, mx3],
    [mx1, a11, a1, a13],
    [mx2, a12, a2, a23],
    [mx3, a13, a3, a33]
  ]) / denominator,

  b3 : det( m: [
    [1, mx1, mx2, my],
    [mx1, a11, a12, a1],
    [mx2, a12, a22, a2],
    [mx3, a13, a23, a3]
  ]) / denominator
}

```

```

// отримали рівняння регресії
console.log(`y = ${B.b0} + ${B.b1} * x1 + ${B.b2} * x2 + ${B.b3} * x3`)

// перевірка рівняння регресії
const y = n => sum(
  Tab.get_x_line(n)
  .map((x : string , i : number) => Object.values(B)[i + 1] * x))
  + Object.values(B)[0]

// таблиця порівняння значення факторів з матриці планування
// і порівняємо результат з середніми значеннями функції
// відгуку за рядками
console.log("перевірка рівняння регресії")
console.table( tabularData: [
  ["y1", "y2", "y3", "y4"],
  y_aver,
  [y( n: 1), y( n: 2), y( n: 3), y( n: 4)]
])

```

```

console.log("<----- Перевірка однорідності дисперсії за критерієм Кохрена: ----->")

// Знайдемо дисперсії по рядках:
const s = n => average(
  Tab.get_y_line(n)
  .map(y => (y - y_aver[n - 1]) * (y - y_aver[n - 1])))

const S = [1, 2, 3, 4].map(s)

const Gp = Math.max(...S) / sum(S)

// перевірка однорідності дисперсії
if (Gp < 0.7679) {
  console.log("Дисперсія однорідна")
} else {
  console.log("Дисперсія не однорідна")
}

console.log("<---+----- Далі оцінимо значимість коефіцієнтів регресії згідно критерію Стюдента --->")
let S_b = Math.sqrt(x average(S) / 12)

const T = {
  t0 : Math.abs(x average([1, 1, 1, 1].map((x : number, i : number) => y_aver[i] * x)) / S_b),
  t1 : Math.abs(x average([-1, -1, 1, 1].map((x : number, i : number) => y_aver[i] * x)) / S_b),
  t2 : Math.abs(x average([-1, 1, -1, 1].map((x : number, i : number) => y_aver[i] * x)) / S_b),
  t3 : Math.abs(x average([-1, 1, 1, -1].map((x : number, i : number) => y_aver[i] * x)) / S_b)
}

console.table(T)

// значимі коефієнти
const cof = (Object.keys(T).filter(k => T[k] > 2.306));
console.log(`Значимі коефіцієнти: ${cof}`)

// творіння формули з значимими коефіцієнтами
const y_ = n => sum(
  cof
  .map(t => parseInt(t.slice(1)))
  .map(i => B[`b${i}`] * (Tab.get_x_col(n)[i] | 1)))

const y__ = [y_(n: 1), y_(n: 2), y_(n: 3), y_(n: 4)]

console.table({ tabularData: [
  ["y1^", "y2^", "y3^", "y4^"],
  y__
]})

console.log("<----- Критерій Фішера ----->")

const d = cof.length

const S_ad = 3 / (4 - d) * sum(
  y__.map((y_ : any, i : number) => {
    return (y_ - y_aver[i]) * (y_ - y_aver[i])
  }))

if (S_ad / S_b > 4.5) {
  console.log("рівняння регресії адекватно оригіналу" +
    " при рівні значимості 0.05")
} else {
  console.log(", рівняння регресії неадекватно оригіналу" +
    " при рівні значимості 0.05")
}

```

Результати

```
Run: tpe_lab_3.js ×
"C:\Program Files\nodejs\node.exe" G:\labs_js\tpe_lab_3.js



| (index) | 0    | 1    | 2    | 3    | 4    | 5    |
|---------|------|------|------|------|------|------|
| 0       | 'x1' | 'x2' | 'x3' | 'Y1' | 'Y2' | 'y3' |
| 1       | 20   | -15  | 20   | 235  | 233  | 212  |
| 2       | 20   | 45   | 35   | 200  | 201  | 210  |
| 3       | 70   | -15  | 35   | 215  | 224  | 214  |
| 4       | 70   | 45   | 20   | 225  | 227  | 222  |



y = 215.66944238566083 + 0.07097039092677844 * x1 + -0.0746934669712749 * x2 + 0.015416679777088962 * x3
перевірка рівняння регресії



| (index) | 0                  | 1                  | 2                  | 3                  |
|---------|--------------------|--------------------|--------------------|--------------------|
| 0       | 'y1'               | 'y2'               | 'y3'               | 'y4'               |
| 1       | 226.66666666666666 | 203.66666666666666 | 217.66666666666666 | 224.66666666666666 |
| 2       | 218.5175858043073  | 214.26722798268713 | 222.29735554730254 | 217.58449733236972 |



<----- Перевірка однорідності дисперсії за критерієм Кохрена: ----->
Дисперсія однорідна
<----- Далі оцінимо значимість коефіцієнтів регресії згідно критерію Стьюдента --->



| (index) | Values             |
|---------|--------------------|
| t0      | 122.24217052970694 |
| t1      | 1.680946577184664  |
| t2      | 2.241262102912885  |
| t3      | 4.20236644296166   |



Значимі коефіцієнти: t0,t3



| (index) | 0                  | 1                  | 2                 | 3                 |
|---------|--------------------|--------------------|-------------------|-------------------|
| 0       | 'y1^'              | 'y2^'              | 'y3^'             | 'y4^'             |
| 1       | 216.20902617785893 | 216.20902617785893 | 215.9931926609797 | 215.6848590654379 |



<----- Критерій Фішера ----->
рівняння регресії адекватно оригіналу при рівні значимості 0.05

Process finished with exit code 0
```

Відповіді на контрольні запитання

1. Що називається дробовим факторним експериментом?

Дробовий факторний експеримент – 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?

Розрахункове значення Кохрена показує, яку частку в загальній сумі дисперсій у рядках має максимальна з них.

3. Для чого перевіряється критерій Стюдента?

Критерій Стюдента використовується для перевірки значущості коефіцієнтів.

4. Чим визначається критерій Фішера і як його застосовувати?

Критерій Фішера використовується для перевірки адекватності рівняння регресії.

Висновки

Проведено дробовий трьохфакторний експеримент. Складено матрицю планування, знайдені коефіцієнти рівняння регресії, проведені 3 статистичні перевірки. Застосовані критерій Фішера, критерій Стюдента, розрахункове значення Кохрена.