

**Міністерство освіти і науки України
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:

Гречаний Є.О.

Перевірив:

ас. Регіда П. Г.

Київ 2021

Тема: Проведення трьохфакторного експерименту

при використанні рівняння регресії з урахуванням ефекту взаємодії.

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

Завдання на лабораторну роботу

- Скласти матрицю планування для повного трьохфакторного експерименту.
- 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.

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

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

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

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

Програмний код

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

#base settings
print("Варіант 203")
x_max = [70, 80, 35]
x_min = [20, 30, 30]
n = 8
y_max = 200 + sum(x_max) / 3
y_min = 200 + sum(x_min) / 3

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 = [xn[1][i] * xn[2][i] for i in range(n)]
x1x3_norm = [xn[1][i] * xn[3][i] for i in range(n)]
x2x3_norm = [xn[2][i] * xn[3][i] for i in range(n)]
x1x2x3_norm = [xn[1][i] * xn[2][i] * xn[3][i] for i in range(n)]

y1 = [random.randint(int(y_min), int(y_max)) for i in range(8)]
y2 = [random.randint(int(y_min), int(y_max)) for i in range(8)]
y3 = [random.randint(int(y_min), int(y_max)) for i in range(8)]

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]]]

x0 = [1, 1, 1, 1, 1, 1, 1]
x1 = [20, 20, 70, 70, 20, 20, 70, 70]
x2 = [30, 80, 30, 80, 30, 80, 30, 80]
x3 = [30, 35, 35, 30, 35, 30, 30, 35]

x1x2 = [x1[i] * x2[i] for i in range(n)]
x1x3 = [x1[i] * x3[i] for i in range(n)]
x2x3 = [x2[i] * x3[i] for i in range(n)]
x1x2x3 = [x1[i] * x2[i] * x3[i] for i in range(n)]

Y_average = []
for i in range(len(y_matrix)):
    Y_average.append(np.mean(y_matrix[i], axis=0))

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))

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))
ai = [round(i, 3) for i in solve(list_for_a, Y_average)]

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]))
Gp = max(dispersions) / sum(dispersions)
Gt = 0.5157

sb = sum(dispersions) / len(dispersions)
sbs = (sb / (8 * 3)) ** 0.5
t_list = [abs(bi[i]) / sbs for i in range(0, 8)]
d = 0
res = [0] * 8
coef_matter = []
coef_nomatter = []
m = 3
F3 = (m - 1) * n
for i in range(n):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coef_nomatter.append(bi[i])
        res[i] = 0
    else:
        coef_matter.append(bi[i])
        res[i] = bi[i]
        d += 1

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])

Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(8)]) / (n - d)
Fp = Sad / sb
F4 = n - d

print("Y : \n")
for i in range(n):
    print("|", *y_matrix[i], "|")

print("X:")
for i in range(n):
    print("|", *list_for_a[i], "|")

print("Рівняння регресії: \n" "y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1*x2 +
{}*x1*x3 + {}*x2*x3 + {}*x1*x2*x3".format(*ai))
print("Рівняння регресії для нормованих факторів: \n" "y = {} + {}*x1 + {}*x2 +
{}*x3 + {}*x1*x2 + {}*x1*x3 +
" "{}*x2*x3 + {}*x1*x2*x3".format(*bi))

print("Перевірка за критерієм Кохрена:")
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")
print("Середні значення відгуку за рядками:", "\n", *Y_average)
print("Перевірка значущості коефіцієнтів за критерієм Стьюдента")
print("Значущі коефіцієнти регресії:", *coef_matter)
print("Незначущі коефіцієнти регресії:", *coef_nomatter)

print("Значення з отриманими коефіцієнтами:", *y_st)

print("\nПеревірка адекватності за критерієм Фішера")

if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("Рівняння регресії адекватне при рівні значимості 0.05\n")
else:
    print("Рівняння регресії не є адекватним при рівні значимості 0.05\n")
print("Виконав студент групи ІО-92 Гречаний Євгеній")

```

Результати виконання

Варіант 203

Y :

251	237	244
245	231	227
248	242	251
251	231	229
253	227	236
261	254	245
256	243	227
243	230	246

X:

1	20	30	30	600	600	900	18000
1	20	80	35	1600	700	2800	56000
1	70	30	35	2100	2450	1050	73500
1	70	80	30	5600	2100	2400	168000
1	20	30	35	600	700	1050	21000
1	20	80	30	1600	600	2400	48000
1	70	30	30	2100	2100	900	63000
1	70	80	35	5600	2450	2800	196000

Рівняння регресії:

$$y = 227.04 + -0.292*x_1 + 2.485*x_2 + 0.291*x_3 + -0.033*x_1*x_2 + 0.014*x_1*x_3 + -0.073*x_2*x_3 + 0.001*x_1*x_2*x_3$$

Рівняння регресії для нормованих факторів:

$$y = 242.0 + -0.583*x_1 + -0.917*x_2 + -2.083*x_3 + -2.167*x_1*x_2 + 4.0*x_1*x_3 + -2.0*x_2*x_3 + 1.417*x_1*x_2*x_3$$

Перевірка за критерієм Кохрена:

Дисперсія однорідна

Середні значення відгуку за рядками:

244.0 234.3333333333334 247.0 237.0 238.6666666666666 253.3333333333334 242.0 239.6666666666666

Перевірка значущості коефіцієнтів за критерієм Стьюдента

Значущі коефіцієнти регресії: 242.0 4.0

Незначущі коефіцієнти регресії: -0.583 -0.917 -2.083 -2.167 -2.0 1.417

Значення з отриманими коефіцієнтами: 246.0 238.0 246.0 238.0 238.0 246.0 238.0 246.0

Перевірка адекватності за критерієм Фишера

Рівняння регресії адекватне при рівні значимості 0.05

Виконав студент групи ІО-92 Гречаний Євгеній