

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №3
з дисципліни «Методи оптимізації планування експерименту»
на тему: «Проведення трьохфакторного експерименту з використанням
рівняння регресії»

Виконала:
студентка групи ІО-91
Кійченко А. К.

Перевірив:
Регіда П. Г.

Київ 2021

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

Завдання на лабораторну роботу

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.

Завдання за варіантом:

111	10	60	-70	-10	60	70
-----	----	----	-----	-----	----	----

Лістинг програми

```
import random
import math
import numpy as np

def get_avg(array):
    result = []
    for i in range(len(array[0])):
        result.append(0)
        for j in range(len(array)):
            result[i] += array[j][i]
        result[i] = result[i]/len(array)
    return result

def get_dispersion(array, avg_y):
    result = []
    for i in range(len(array[0])):
        result.append(0)
        for j in range(len(array)):
            result[i] += (array[j][i] - avg_y[i])**2
        result[i] = result[i]/3
    return result

def determinant(array):
    a = np.array(array)
    return np.linalg.det(a)

m = 3
n = 4
x0 = [1, 1, 1, 1]
x1 = [-1, -1, 1, 1]
x2 = [-1, 1, -1, 1]
x3 = [-1, 1, 1, -1]

x = [[10, 10, 60, 60], [-70, -10, -70, -10], [60, 70, 70, 60]]
y_min = 200
y_max = 240

y1 = []
y2 = []
y3 = []
for i in range(4):
    y1.append(random.randint(y_min, y_max))
    y2.append(random.randint(y_min, y_max))
    y3.append(random.randint(y_min, y_max))

# y1 = [15, 10, 11, 16]
# y2 = [18, 19, 14, 19]
# y3 = [16, 13, 12, 16]
```

```

print("матриця планування:\n X1 | X2 | X3 | Y1 | Y2 | Y3")
for i in range(4):
    print(f"{{x1[i]:^4}}|{{x2[i]:^4}}|{{x3[i]:^4}}|{{y1[i]:^4}}|{{y2[i]:^4}}|{{y3[i]:^4}}")

avg_y = get_avg([y1, y2, y3])
print("Середнє значення функції відгуку в рядку:\ny\u0304 = {:.3f},"
      " y\u03042 = {:.3f},"
      " y\u03043 = {:.3f},"
      " y\u03044 = {:.3f}".format(avg_y[0], avg_y[1], avg_y[2], avg_y[3]))

mx1 = sum(x[0])/4
mx2 = sum(x[1])/4
mx3 = sum(x[2])/4
my = sum(avg_y)/4
print("mx1 = {:.3f}, mx2 = {:.3f}, mx3 = {:.3f}, my = {:.3f}".format(mx1, mx2, mx3, my))

a = []
for i in range(3):
    a.append(0)
    for j in range(4):
        a[i] += (x[i][j] * avg_y[j])
    a[i] = a[i]/4
print("a1 = {:.3f}, a2 = {:.3f}, a3 = {:.3f}".format(a[0], a[1], a[2]))

matr_a = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
for i in range(4):
    matr_a[0][0] += x[0][i] * x[0][i] / 4
    matr_a[1][1] += x[1][i] * x[1][i] / 4
    matr_a[2][2] += x[2][i] * x[2][i] / 4
    matr_a[0][1] += x[0][i] * x[1][i] / 4
    matr_a[1][0] += x[0][i] * x[1][i] / 4
    matr_a[0][2] += x[0][i] * x[2][i] / 4
    matr_a[2][0] += x[0][i] * x[2][i] / 4
    matr_a[2][1] += x[2][i] * x[1][i] / 4
    matr_a[1][2] += x[2][i] * x[1][i] / 4
print("a11 = {}, a12 = {}, a13 = {}".format(matr_a[0][0], matr_a[0][1], matr_a[0][2]))
print("a21 = {}, a22 = {}, a23 = {}".format(matr_a[1][0], matr_a[1][1], matr_a[1][2]))
print("a31 = {}, a32 = {}, a33 = {}".format(matr_a[2][0], matr_a[2][1], matr_a[2][2]))

b01 = determinant([[my, mx1, mx2, mx3],
                    [a[0], matr_a[0][0], matr_a[0][1], matr_a[0][2]],
                    [a[1], matr_a[1][0], matr_a[1][1], matr_a[1][2]],
                    [a[2], matr_a[2][0], matr_a[2][1], matr_a[2][2]]])
b02 = determinant([[1, mx1, mx2, mx3],
                    [mx1, matr_a[0][0], matr_a[0][1], matr_a[0][2]],
                    [mx2, matr_a[1][0], matr_a[1][1], matr_a[1][2]],
                    [mx3, matr_a[2][0], matr_a[2][1], matr_a[2][2]]])
b0 = b01/b02
b11 = determinant([[1, my, mx2, mx3],
                    [mx1, a[0], matr_a[0][1], matr_a[0][2]],
                    [mx2, a[1], matr_a[1][1], matr_a[2][1]],
                    [mx3, a[2], matr_a[1][2], matr_a[2][2]]])
b12 = determinant([[1, mx1, mx2, mx3],
                    [mx1, matr_a[0][0], matr_a[0][1], matr_a[0][2]],
                    [mx2, matr_a[0][1], matr_a[1][1], matr_a[2][1]],
                    [mx3, matr_a[0][2], matr_a[1][2], matr_a[2][2]]])
b1 = b11/b12

```

```

b21 = determinant([[1, mx1, my, mx3],
                  [mx1, matr_a[0][0], a[0], matr_a[0][2]],
                  [mx2, matr_a[1][0], a[1], matr_a[1][2]],
                  [mx3, matr_a[2][0], a[2], matr_a[2][2]]])
b22 = determinant([[1, mx1, mx2, mx3],
                  [mx1, matr_a[0][0], matr_a[0][1], matr_a[0][2]],
                  [mx2, matr_a[1][0], matr_a[1][1], matr_a[1][2]],
                  [mx3, matr_a[2][0], matr_a[2][1], matr_a[2][2]]])
b2 = b21/b22
b31 = determinant([[1, mx1, mx2, my],
                  [mx1, matr_a[0][0], matr_a[0][1], a[0]],
                  [mx2, matr_a[1][0], matr_a[1][1], a[1]],
                  [mx3, matr_a[2][0], matr_a[2][1], a[2]]])
b32 = determinant([[1, mx1, mx2, mx3],
                  [mx1, matr_a[0][0], matr_a[0][1], matr_a[0][2]],
                  [mx2, matr_a[1][0], matr_a[1][1], matr_a[1][2]],
                  [mx3, matr_a[2][0], matr_a[2][1], matr_a[2][2]]])
b3 = b31/b32
print("b0 = {:.3f}, b1 = {:.3f}, b2 = {:.3f}, b3 = {:.3f}".format(b0, b1, b2, b3))

print("Підставимо значення факторів з матриці планування")
print("{:.3f} + {:.3f} * {:.3f} + {:.3f} * {:.3f} + {:.3f} * {:.3f} = {:.3f}".format(b0,
b1, x[0][0], b2, x[1][0], b3, x[2][0], b0 + b1 * x[0][0] + b2 * x[1][0] + b3 * x[2][0]))
print("{:.3f} + {:.3f} * {:.3f} + {:.3f} * {:.3f} + {:.3f} * {:.3f} = {:.3f}".format(b0,
b1, x[0][1], b2, x[1][1], b3, x[2][1], b0 + b1 * x[0][1] + b2 * x[1][1] + b3 * x[2][1]))
print("{:.3f} + {:.3f} * {:.3f} + {:.3f} * {:.3f} + {:.3f} * {:.3f} = {:.3f}".format(b0,
b1, x[0][2], b2, x[1][2], b3, x[2][2], b0 + b1 * x[0][2] + b2 * x[1][2] + b3 * x[2][2]))
print("{:.3f} + {:.3f} * {:.3f} + {:.3f} * {:.3f} + {:.3f} * {:.3f} = {:.3f}".format(b0,
b1, x[0][3], b2, x[1][3], b3, x[2][3], b0 + b1 * x[0][3] + b2 * x[1][3] + b3 * x[2][3]))

sigma = get_dispersion([y1, y2, y3], avg_y)
print("Значення дисперсії по рядках: \u03c3\u00b2(y1) = {:.2f},"
      " \u03c3\u00b2(y2) = {:.2f},"
      " \u03c3\u00b2(y3) = {:.2f},"
      "\u03c3\u00b2(y3) = {:.2f}".format(sigma[0], sigma[1], sigma[2], sigma[3]))

gp = max(sigma)/sum(sigma)
print("Gp = ", gp)
f1 = m-1
f2 = n
if gp < 0.7679:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")
    exit()

# оцінка значимості коефіцієнтів регресії за критерієм Стьюдента
sb = sum(sigma) / n
s2bs = sb / (n * m)
sbs = math.sqrt(s2bs)
print("S\u00b2b = {:.3f}, S\u00b2(\u03b2s) = {:.3f}, S(\u03b2s) = {:.3f}".format(sb,
s2bs, sbs))

betha0 = (avg_y[0] * x0[0] + avg_y[1] * x0[1] + avg_y[2] * x0[2] + avg_y[3] * x0[3]) / 4
betha1 = (avg_y[0] * x1[0] + avg_y[1] * x1[1] + avg_y[2] * x1[2] + avg_y[3] * x1[3]) / 4
betha2 = (avg_y[0] * x2[0] + avg_y[1] * x2[1] + avg_y[2] * x2[2] + avg_y[3] * x2[3]) / 4
betha3 = (avg_y[0] * x3[0] + avg_y[1] * x3[1] + avg_y[2] * x3[2] + avg_y[3] * x3[3]) / 4

```

```

print("\u03b20 = {:.3f}, \u03b21 = {:.3f}, \u03b22 = {:.3f}, \u03b23 = {:.3f}".format(betha0, betha1, betha2, betha3))

t0 = abs(betha0)/sbs
t1 = abs(betha1)/sbs
t2 = abs(betha2)/sbs
t3 = abs(betha3)/sbs
print("t0 = {:.3f}, t1 = {:.3f}, t2 = {:.3f}, t3 = {:.3f}".format(t0, t1, t2, t3))

f3 = f1 * f2
t_table = 2.306

y = [0, 0, 0, 0]
t = [t0, t1, t2, t3]
b = [b0, b1, b2, b3]
d = 0
for j in range(4):
    for i in range(4):
        if t[i] > t_table:
            if i == 0:
                y[j] += b[i]
            else:
                y[j] += b[i] * x[i - 1][j]
        d += 1/4
print("d = {}".format(d))
print("Підставимо значення факторів з матриці планування, y =", y[0], y[1], y[2], y[3])

# критерій Фішера
s2ad = (m / (n - d)) * sum((y[i] - avg_y[i])**2 for i in range(4))
print("S\u00b2ад = {:.3f}".format(s2ad))

Fp = s2ad / sb
f4 = n - d
print("Критерій Фішера Fp =", Fp)

F8_table = [5.3, 4.5, 4.1, 3.8, 3.7, 3.6, 3.3, 3.1, 2.9]
if Fp < F8_table[int(f4-1)]:
    print("Рівняння регресії адекватно оригіналу при рівні значимості 0.05")
else:
    print("Рівняння регресії неадекватно оригіналу при рівні значимості 0.05")

```

Результат виконання роботи

Запишемо лінійне рівняння регресії:

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3$$

матриця планування:

X1	X2	X3	Y1	Y2	Y3
-1	-1	-1	203	229	219
-1	1	1	210	215	201
1	-1	1	240	203	215
1	1	-1	205	240	231

Середнє значення функції відгуку в рядку:

$\bar{y}_1 = 217.000$, $\bar{y}_2 = 208.667$, $\bar{y}_3 = 219.333$, $\bar{y}_4 = 225.333$

$m_{x1} = 35.000$, $m_{x2} = -40.000$, $m_{x3} = 65.000$, $m_y = 217.583$

$a_1 = 7734.167$, $a_2 = -8720.833$, $a_3 = 14125.000$

$a_{11} = 1850.0$, $a_{12} = -1400.0$, $a_{13} = 2275.0$

$a_{21} = -1400.0$, $a_{22} = 2500.0$, $a_{23} = -2600.0$

$a_{31} = 2275.0$, $a_{32} = -2600.0$, $a_{33} = 4250.0$

$b_0 = 256.739$, $b_1 = 0.190$, $b_2 = -0.019$, $b_3 = -0.717$

Підставимо значення факторів з матриці планування

$256.739 + 0.190 * 10.000 + -0.019 * -70.000 + -0.717 * 60.000 = 217.000$

$256.739 + 0.190 * 10.000 + -0.019 * -10.000 + -0.717 * 70.000 = 208.667$

$256.739 + 0.190 * 60.000 + -0.019 * -70.000 + -0.717 * 70.000 = 219.333$

$256.739 + 0.190 * 60.000 + -0.019 * -10.000 + -0.717 * 60.000 = 225.333$

Значення дисперсії по рядках: $\sigma^2(y_1) = 114.67$, $\sigma^2(y_2) = 33.56$, $\sigma^2(y_3) = 237.56$, $\sigma^2(y_4) = 220.22$

$G_p = 0.39200586725339204$

Дисперсія однорідна

$S^2_b = 151.500$, $S^2(\beta_s) = 12.625$, $S(\beta_s) = 3.553$

$\beta_0 = 217.583$, $\beta_1 = 4.750$, $\beta_2 = -0.583$, $\beta_3 = -3.583$

$t_0 = 61.236$, $t_1 = 1.337$, $t_2 = 0.164$, $t_3 = 1.008$

$d = 1.0$

Підставимо значення факторів з матриці планування, $y = 256.73888888889144$
 256.73888888889144 256.73888888889144 256.73888888889144

$S^2_{ад} = 6275.602$

Критерій Фішера $F_p = 41.42311779326608$

Рівняння регресії неадекватно оригіналу при рівні значимості 0.05

Відповіді на контрольні запитання

1. Що називається дробовим факторним експериментом?
Дробовим факторним експериментом називають експеримент який використовує тільки частину ПФЕ.
2. Для чого потрібно розрахункове значення Кохрена?
Це значення використовується для перевірки однорідності дисперсій.
3. Для чого перевіряється критерій Стюдента?
Щоб перевірити значимість коефіцієнтів рівняння
4. Чим визначається критерій Фішера і як його застосовувати?
Критерій Фішера потрібен для перевірки отриманого рівняння регресії.