

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

Методи оптимізації та планування експерименту

Лабораторна робота №3

Виконав:
студент групи ІО-91
Дикун А.В.
Залікова книжка № 9110
Варіант № 8
Перевірив: Регіда П. Г.

Київ 2021р

Текст програми:

```
from random import randint
import numpy as np

# Find coefficients
def findA(a, b=None):
    if b is None:
        return sum([a[i] ** 2 for i in range(len(a))]) / len(a)
    else:
        return sum(a[i] * b[i] for i in range(len(a))) / len(a)

# Cramer's rule
def cramer(arr, ins, pos):
    matrix = np.insert(np.delete(arr, pos, 1), pos, ins, 1)

    return np.linalg.det(matrix) / np.linalg.det(arr)

# Method to get dispersion
def getDispersion(y, y_r):
    return [round(sum([(y[i][j] - y_r[i]) ** 2 for j in range(len(y[i]))]) /
3, 3) for i in range(len(y_r))]

# Cochran criteria
def cochrان(disр, m):
    Gp = max(disр) / sum(disр)
    Gt = [.9065, .7679, .6841, .6287, .5892, .5598, .5365, .5175, .5017,
.4884]

    if Gp < Gt[m - 2]:
        return [round(Gp, 4), Gt[m - 2]]
    else:
        return

# Student criteria
def student(disр, m, y_r, x_n):
    table = {
        8: 2.306,
        12: 2.179,
        16: 2.120,
        20: 2.086,
        24: 2.064,
        28: 2.048,
        'inf': 1.960
    }

    x_nT = x_n.T
    N = len(y_r)

    Sb = sum(disр) / len(y_r)
    Sbeta = (Sb / (m * N)) ** (1 / 2)

    beta = [sum([y_r[j] * x_nT[i][j] for j in range(N)]) / N for i in
range(N)]
    t = [abs(beta[i]) / Sbeta for i in range(len(beta))]

    f3 = N * (m - 1)

    if f3 > 30:
```

```

        t_t = table['inf']
    elif f3 > 0:
        t_t = table[f3]
    else:
        return

    result = []
    for i in t:
        if i < t_t:
            result.append(False)
        else:
            result.append(True)

    return result

# Fisher criteria
def fisher(y_r, y_st, b_det, disp, m):
    table = {
        8: [5.3, 4.5, 4.1, 3.8, 3.7, 3.6],
        12: [4.8, 3.9, 3.5, 3.3, 3.1, 3.0],
        16: [4.5, 3.6, 3.2, 3.0, 2.9, 2.7],
        20: [4.5, 3.5, 3.1, 2.9, 2.7, 2.6],
        24: [4.3, 3.4, 3.0, 2.8, 2.6, 2.5]
    }

    N = len(y_r)
    Sb = sum(disp) / N
    d = 0
    for b in b_det:
        if b:
            d += 1

    f4 = N - d
    f3 = N * (m - 1)
    Sad = (m / f4) * sum([(y_st[i] - y_r[i]) ** 2 for i in range(N)])
    Fap = Sad / Sb
    Ft = table[f3][f4 - 1]

    if Fap < Ft:
        return f"\nPівніняння регресії адекватно оригіналу:\nFap < Ft:
{round(Fap, 2)} < {Ft}"
    else:
        return f"\nPівніняння регресії неадекватно оригіналу: \nFap > Ft:
{round(Fap, 2)} > {Ft}"

# Main function
def experiment(m, min_x1, max_x1, min_x2, max_x2, min_x3, max_x3):
    y_min = round((min_x1 + min_x2 + min_x3) / 3) + 200
    y_max = round((max_x1 + max_x2 + max_x3) / 3) + 200

    x_norm = np.array([
        [1, -1, -1, -1],
        [1, -1, 1, 1],
        [1, 1, -1, 1],
        [1, 1, 1, -1]
    ])

    x = np.array([
        [min_x1, min_x2, min_x3],
        [min_x1, max_x2, max_x3],
        [max_x1, min_x2, max_x3],
        [max_x1, max_x2, min_x3]
    ])

```

```

    ])

    # Generate Y and calculate their Response
    y = [[randint(y_min, y_max) for _ in range(m)] for _ in range(len(x))]
    y_r = [round(sum(y[i]) / len(y[i]), 2) for i in range(len(y))]
    N = len(y_r)

    # Calculate Cochran's C test
    disp = getDispersion(y, y_r)
    cochran_cr = cochran(disp, m)

    if cochran_cr is None:
        raise Exception("Need more experiments")
    else:
        pass

    # Coefficients
    mx = [sum(i) / len(i) for i in x.T]
    my = sum(y_r) / N

    x_T = x.T

    a1 = findA(x_T[0], y_r)
    a2 = findA(x_T[1], y_r)
    a3 = findA(x_T[2], y_r)

    a11 = findA(x_T[0])
    a22 = findA(x_T[1])
    a33 = findA(x_T[2])

    a12 = a21 = findA(x_T[0], x_T[1])
    a13 = a31 = findA(x_T[0], x_T[2])
    a23 = a32 = findA(x_T[1], x_T[2])

    # Solve SoLE by Cramer's rule
    b_delta = np.array([
        [1, mx[0], mx[1], mx[2]],
        [mx[0], a11, a12, a13],
        [mx[1], a21, a22, a23],
        [mx[2], a31, a32, a33]
    ])

    b_set = np.array([my, a1, a2, a3])
    b = [cramer(b_delta, b_set, i) for i in range(N)]

    b_det = student(disp, m, y_r, x_norm)
    b_cut = b.copy()

    # Simplified equations
    if b_det is None:
        raise Exception("Need more experiments")
    else:
        for i in range(N):
            if not b_det[i]:
                b_cut[i] = 0

        y_st = [round(b_cut[0] + x[i][0] * b_cut[1] + x[i][1] * b_cut[2] +
x[i][2] * b_cut[3], 2) for i in range(N)]

    # Calculate F-test
    fisher_cr = fisher(y_r, y_st, b_det, disp, m)

    # Print out results
    print(f"\nМатриця планування для m = {m}:")

```

```

for i in range(m):
    print(f"Y{i + 1} = {np.array(y).T[i]}")

print(f"\nСередні значення функції відгуку за рядками:\nY_R: {y_r}")
print(f"\nКоефіцієнти рівняння регресії:")
for i in range(len(b)):
    print(f"b{i} = {round(b[i], 3)}")

print("\nДисперсії по рядках:\nS²{y} = ", disp, sep="")
print(f"\nЗа критерієм Кохрена дисперсія однорідна:\nGp < Gt -
{cochran_cr[0]} < {cochran_cr[1]}")

print(f"\nЗа критерієм Стьюдента коефіцієнти ", end="")
for i in range(len(b_det)):
    if not b_det[i]:
        print(f"b{i} ", end="")
print("приймаємо незначними")

print(f"\nОтримані функції відгуку зі спрощеними коефіцієнтами:\nY_St -
{y_st}")
print(fisher_cr)

def startExperiment(Min_x1, Max_x1, Min_x2, Max_x2, Min_x3, Max_x3):
    global M
    try:
        experiment(M, Min_x1, Max_x1, Min_x2, Max_x2, Min_x3, Max_x3)
    except:
        print("Збільшуємо кількість експериментів")
        M += 1
        startExperiment(Min_x1, Max_x1, Min_x2, Max_x2, Min_x3, Max_x3)

if __name__ == '__main__':
    Min_x1, Max_x1 = -30, 0
    Min_x2, Max_x2 = -35, 10
    Min_x3, Max_x3 = 0, 20

    M = 3
    startExperiment(Min_x1, Max_x1, Min_x2, Max_x2, Min_x3, Max_x3)

```

Результати роботи програми:

```
C:\Users\anvat\Anaconda3\envs\MOPE_labs\python.exe C:/Users/anvat
```

Матриця планування для $m = 3$:

Y_1 - [206 196 210 208]

Y_2 - [183 199 180 205]

Y_3 - [210 202 201 201]

Середні значення функції відгуку за рядками:

Y_R : [199.67, 199.0, 197.0, 204.67]

Коефіцієнти рівняння регресії:

$b_0 = 203.892$

$b_1 = 0.05$

$b_2 = 0.078$

$b_3 = -0.208$

Дисперсії по рядках:

$S^2\{y\} = [141.556, 6.0, 158.0, 8.222]$

За критерієм Кохрена дисперсія однорідна:

$G_p < G_t - 0.5035 < 0.7679$

За критерієм Стюдента коефіцієнти b_1 b_2 b_3 приймаємо незначними

Отримані функції відгуку зі спрощеними коефіцієнтами:

Y_St - [203.89, 203.89, 203.89, 203.89]

Рівняння регресії адекватно оригіналу:

$F_{ap} < F_t: 1.14 < 4.1$

Process finished with exit code 0