

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 №2
З дисципліни «Методи наукових досліджень»
За темою:
«Проведення двофакторного експерименту з
використанням лінійного рівняння регресії»

ВИКОНАВ:
Студент II курсу ФІОТ
Групи ІВ-91
Гутов В.В.
Номер у списку - 8

ПЕРЕВІРИВ:
асистент
Регіда П.Г.

Київ 2021 р.

Мета: провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.

Завдання:

№ варіанта	X1		X2	
	min	max	min	max
108	-30	0	-35	10

Програмний код

```
import random
from beautifultable import BeautifulTable
from numpy.linalg import det

# Гутов Віталій
# Варіант 108: x1_min = -30, x1_max = 0, x2_min = -35, x2_max = 10
# y_min = (20 - 108) * 10 = -880
# y_max = (30 - 108) * 10 = -780

x1_min = -30
x1_max = 0
x2_min = -35
x2_max = 10
y_min = (20 - 108) * 10
y_max = (30 - 108) * 10
x0 = 1
m = 5
xn = [[-1, -1], [1, -1], [-1, 1]]

def romanovskii():
    global matrix_y
    global average_y
    global dispersion_y
    global main_deviation
    global f_uv
    global theta_uv
    global r_uv
    global m

    # Для довірчої ймовірності 0.9
    romanovskii_table = {(5, 6, 7): 2.0, (8, 9): 2.17, (10, 11): 2.29,
                          (12, 13): 2.39, (14, 15, 16, 17): 2.49, (18, 19,
20): 2.62}

    for key in romanovskii_table.keys():
        if m in key:
            rom_value = romanovskii_table[key]
            break

    matrix_y = [[random.randint(y_min, y_max) for i in range(m)] for j in
range(len(xn))]

    average_y = [sum(matrix_y[i])/len(matrix_y[i]) for i in range(len(xn))]

    tmp_dispersion_y = [[round(((matrix_y[j][i] - average_y[j]) ** 2), 3) for
i in range(m)] for j in range(len(xn))]
```

```

dispersion_y = [sum(tmp_dispersion_y[i]) / m for i in range(len(xn))]

main_deviation = ((2 * (2 * m - 2)) / (m * (m - 4))) ** (1 / 2)

def f_uv_adder(u, v):
    if dispersion_y[u] >= dispersion_y[v]:
        value = dispersion_y[u] / dispersion_y[v]
    else:
        value = dispersion_y[v] / dispersion_y[u]
    return value

f_uv = [f_uv_adder(0, 1), f_uv_adder(0, 2), f_uv_adder(1, 2)]

theta_uv = [((m - 2) / m) * f_uv[i] for i in range(len(f_uv))]

r_uv = [(abs(theta_uv[i] - 1) / main_deviation) for i in
range(len(theta_uv))]

    if not r_uv[0] <= rom_value or not r_uv[1] <= rom_value or not r_uv[2] <=
rom_value:
        m += 1
        romanovskii()

romanovskii()

mx1 = (xn[0][0] + xn[1][0] + xn[2][0]) / 3
mx2 = (xn[0][1] + xn[1][1] + xn[2][1]) / 3
my = (sum(average_y)) / 3

a1 = (xn[0][0] ** 2 + xn[1][0] ** 2 + xn[2][0] ** 2) / 3
a2 = (xn[0][0] * xn[0][1] + xn[1][0] * xn[1][1] + xn[2][0] * xn[2][1]) / 3
a3 = (xn[0][1] ** 2 + xn[1][1] ** 2 + xn[2][1] ** 2) / 3

a11 = (xn[0][0] * average_y[0] + xn[1][0] * average_y[1] + xn[2][0] *
average_y[2]) / 3
a22 = (xn[0][1] * average_y[0] + xn[1][1] * average_y[1] + xn[2][1] *
average_y[2]) / 3

main_det = det([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])
b0 = det([[my, mx1, mx2], [a11, a1, a2], [a22, a2, a3]]) / main_det
b1 = det([[1, my, mx2], [mx1, a11, a2], [mx2, a22, a3]]) / main_det
b2 = det([[1, mx1, my], [mx1, a1, a11], [mx2, a2, a22]]) / main_det

dx1 = abs(x1_max - x1_min) / 2
dx2 = abs(x2_max - x2_min) / 2
x10 = (x1_max + x1_min) / 2
x20 = (x2_max + x2_min) / 2

a0_naturalized = b0 - (b1 * x10 / dx1) - (b2 * x20 / dx2)
a1_naturalized = b1 / dx1
a2_naturalized = b2 / dx2

# Роздрукування нормованої матриці планування
norm_table = BeautifulTable()
headers_y = ['Y{}'.format(i) for i in range(1, m+1)]
norm_table.columns.header = ['Normalized X1', 'Normalized X2', *headers_y]
for i in range(len(xn)):
    norm_table.rows.append([*xn[i], *matrix_y[i]])
print(norm_table, '\n')

# Роздрукування результатів обчислень
criterion_table = BeautifulTable()
criterion_table.columns.header = ['Average Y', 'Dispersion Y', 'Fuv',

```

```

'THETAuv', 'Ruv', 'Main deviation']
for i in range(len(xn)):
    criterion_table.rows.append([average_y[i], dispersion_y[i], f_uv[i],
theta_uv[i], r_uv[i], main_deviation])
print(criterion_table)

print('Normalized equations:')
for i in range(len(xn)):
    result_y = b0 + b1 * xn[i][0] + b2 * xn[i][1]
    print('y = {} + {} * x1 + {} * x2 = {}'.format(round(b0, 3), round(b1,
3), round(b2, 3), round(result_y, 3)))

x_list = [[], [], []]
for i in range(len(xn)):
    if xn[i][0] > 0:
        x_list[i].append(x1_max)
    else:
        x_list[i].append(x1_min)

    if xn[i][1] > 0:
        x_list[i].append(x2_max)
    else:
        x_list[i].append(x2_min)

print('Naturalized equations:')
for i in range(len(xn)):
    result_y = a0_naturalized + a1_naturalized * x_list[i][0] +
a2_naturalized * x_list[i][1]
    print('y = {} + {} * x1 + {} * x2 = {}'.format(round(a0_naturalized, 3),
round(a1_naturalized, 3),
round(a2_naturalized, 3),
round(result_y, 3)))

```

Результати роботи програми

Normalized X1	Normalized X2	Y1	Y2	Y3	Y4	Y5
-1	-1	-854	-818	-792	-823	-869
1	-1	-842	-860	-852	-819	-805
-1	1	-834	-788	-824	-872	-783

Average Y	Dispersion Y	Fuv	THETAuv	Ruv	Main deviation
-831.2	745.36	1.76	1.056	0.031	1.789
-835.6	423.44	1.424	0.855	0.081	1.789
-820.2	1061.76	2.507	1.504	0.282	1.789

Normalized equations:

$$y = -827.9 + -2.2 * x_1 + 5.5 * x_2 = -831.2$$

$$y = -827.9 + -2.2 * x_1 + 5.5 * x_2 = -835.6$$

$$y = -827.9 + -2.2 * x_1 + 5.5 * x_2 = -820.2$$

Naturalized equations:

$$y = -827.044 + -0.147 * x_1 + 0.244 * x_2 = -831.2$$

$$y = -827.044 + -0.147 * x_1 + 0.244 * x_2 = -835.6$$

$$y = -827.044 + -0.147 * x_1 + 0.244 * x_2 = -820.2$$

Контрольні запитання:

1) Що таке регресійні поліноми і де вони застосовуються?

В теорії планування експерименту найважливішою частиною є оцінка результатів вимірів. При цьому використовують апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. В ТПЕ ці поліноми отримали спеціальну назву - регресійні поліноми, а їх знаходження та аналіз - регресійний аналіз.

2) Визначення однорідності дисперсії.

Однорідність дисперсії – необхідна умова підтвердження гіпотези про забезпечення нормального закону розподілу вимірюваної величини при обраній кількості повторів m та ймовірності p .

3) Що називається повним факторним експериментом?

Повний факторний експеримент – експеримент, у якому використовуються всі можливі комбінації рівнів факторів (r^k)

Висновок

Виконуючи дану лабораторну роботу, я провів двофакторний експеримент, перевіряв однорідність дисперсії за критерієм Романовського, отримав коефіцієнти рівняння регресії та провів натуралізацію рівняння регресії.

Результати роботи програми наведені вище. Під час виконання роботи проблем не виникло.