

Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ

ЗВІТ
з лабораторної роботи №4
з навчальної дисципліни «Методи наукових досліджень»

Тема:
Проведення трьохфакторного експерименту
при використанні рівняння регресії з урахуванням ефекту взаємодії

Виконала:
Студентка 2 курсу кафедри ОТ ФІОТ,
Навчальної групи ІВ-92
Орлова Ю.Д.
Номер у списку групи: 15

Перевірив:
Регіда П.Г.

Київ 2020

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

Завдання на лабораторну роботу:

1. Скласти матрицю планування для повного трьохфакторного експерименту.
2. Провести експеримент, повторивши N раз дослід 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.
3. Знайти коефіцієнти рівняння регресії і записати його.
4. Провести 3 статистичні перевірки – за критеріями Кохрена, Стьюдента, Фішера.
5. Зробити висновки по адекватності регресії та значимості окремих коефіцієнтів і записати скореговане рівняння регресії.
6. Написати комп'ютерну програму, яка усе це моделює.

№ варіанту	x 1		x 2		x 3	
215	min	max	min	max	min	max
	-25	75	-20	60	-25	-10

Код програми:

```
import random
import numpy as np
import scipy.stats

x1_min = -25;
x1_max = 75
x2_min = -20;
x2_max = 60
x3_min = -25;
x3_max = -10

y_min = 200 + (x1_min + x2_min + x3_min) / 3
y_max = 200 + (x1_max + x2_max + x3_max) / 3

n = 8

x0_n = (1, 1, 1, 1, 1, 1, 1, 1)
x1_n = (-1, -1, -1, -1, 1, 1, 1, 1)
x2_n = (-1, -1, 1, 1, -1, -1, 1, 1)
x3_n = (-1, 1, -1, 1, -1, 1, -1, 1)
x1x2_n = [x1_n[i] * x2_n[i] for i in range(n)]
x1x3_n = [x1_n[i] * x3_n[i] for i in range(n)]
x2x3_n = [x2_n[i] * x3_n[i] for i in range(n)]
x1x2x3_n = [x1_n[i] * x2_n[i] * x3_n[i] for i in range(n)]
```

```

x1 = (x1_min, x1_min, x1_min, x1_min, x1_max, x1_max, x1_max, x1_max)
x2 = (x2_min, x2_min, x2_max, x2_max, x2_min, x2_min, x2_max, x2_max)
x3 = (x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, x3_min, x3_max)

def experiment(m):
    y = [[random.uniform(y_min, y_max) for i in range(m)] for j in range(n)]

    # the average value of the response functions in the rows
    y_response = ([round(sum(y[j][i] for i in range(m)) / m, 3) for j in range(n)])

    print('Середні значення функції відгуку:\n{0}'.format(y_response))

    b0 = sum(y_response) / n
    b1 = sum([y_response[i] * x1_n[i] for i in range(n)]) / n
    b2 = sum([y_response[i] * x2_n[i] for i in range(n)]) / n
    b3 = sum([y_response[i] * x3_n[i] for i in range(n)]) / n
    b12 = sum([y_response[i] * x1_n[i] * x2_n[i] for i in range(n)]) / n
    b13 = sum([y_response[i] * x1_n[i] * x3_n[i] for i in range(n)]) / n
    b23 = sum([y_response[i] * x2_n[i] * x3_n[i] for i in range(n)]) / n
    b123 = sum([y_response[i] * x1_n[i] * x2_n[i] * x3_n[i] for i in range(n)]) / n
    b = [b0, b1, b2, b3, b12, b13, b23, b123]

    print(
        '\nОтримане рівняння регресії:\ny = {0} + {1}*x1 + {2}*x2 + {3}*x3 +
        {4}*x1*x2 + {5}*x1*x3 + {6}*x2*x3 + {7}*x1*x2*x3\n'
        .format(round(b0, 3), round(b1, 3), round(b2, 3), round(b3, 3), round(b12,
        3), round(b13, 3), round(b23, 3),
        round(b123, 3), ))

    dispersions = [sum([(y[j][i] - y_response[j]) ** 2 for i in range(m)]) / m for j
in range(n)]

    gp = max(dispersions) / sum(dispersions)

    f1 = m - 1; f2 = n; q = 0.05

    if 11 <= f1 <= 16: f1 = 11
    if 17 <= f1 <= 136: f1 = 17
    if f1 > 136: f1 = 137
    gt = {1: 0.9065, 2: 0.7679, 3: 0.6841, 4: 0.6287, 5: 0.5892, 6: 0.5598, 7:
0.5365, 8: 0.5365, 9: 0.5017, 10: 0.4884,
        11: 0.4366, 17: 0.3720, 137: 0.2500}

    if gp > gt[f1]:
        i = input('Дисперсія неоднорідна. Якщо ви хочете повторити експеримент при m
= m + 1 = {}, введіть 1: \n'
        .format(m + 1))
        if i == '1':
            experiment(m + 1)
            m += 1
        else:
            print('Дисперсія однорідна.')

    # assessment of the significance of regression coefficients according to
Student's criterion
    s_b = sum(dispersions) / n
    s = np.sqrt(s_b / (n * m))
    t = [abs(b[i]) / s for i in range(n)]

    f3 = f1 * f2

```

```

d = 0
for i in range(n):
    if t[i] < scipy.stats.t.ppf(q=0.975, df=f3):
        print('Коефіцієнт рівняння регресії b{0} приймаємо незначним при
рівні значимості 0.05'.format(i))
        b[i] = 0
    else: d += 1

# Fisher's criterion
f4 = n - d
s_ad = (m *
sum([(b[0]+b[1]*x1_n[i]+b[2]*x2_n[i]+b[3]*x3_n[i]+b[4]*x1_n[i]*x2_n[i]+b[5]*x1_n[i]*x
3_n[i]+b[6]*
x2_n[i]*x3_n[i]+b[7]*x1_n[i]*x2_n[i]*x3_n[i] -
y_response[i]) ** 2 for i in range(n)]) / f4)
f_p = s_ad / s_b

if f_p > scipy.stats.f.ppf(q=0.95, dfn=f4, dfd=f3):
    print('Рівняння регресії неадекватно оригіналу при рівні значимості
0.05')
else:
    print('Рівняння регресії адекватно оригіналу при рівні значимості 0.05')

try:
    m = int(input(("Введіть значення m: ")))
    experiment(m)
except:
    breakpoint()
    print("Ви ввели не ціле число. Спробуйте знову.")

```

Результат виконання програми:

```

Введіть значення m: 4
Середні значення функції відгуку:
[216.48, 205.395, 200.971, 225.927, 221.721, 211.063, 215.067, 221.205]

Отримане рівняння регресії:
y = 214.729 + 2.535*x1 + 1.064*x2 + 1.169*x3 + -0.192*x1*x2 + -2.299*x1*x3 + 6.605*x2*x3 + -2.406*x1*x2*x3

Дисперсія однорідна.
Коефіцієнт рівняння регресії b1 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b2 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b3 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b4 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b5 приймаємо незначним при рівні значимості 0.05
Коефіцієнт рівняння регресії b7 приймаємо незначним при рівні значимості 0.05
Рівняння регресії адекватно оригіналу при рівні значимості 0.05

Process finished with exit code 0

```

Висновки: під час виконання програми ми провели повний трьохфакторний експеримент. Було складено матрицю планування, знайдено рівняння регресії адекватне об'єкту. Було написано програму з використанням можливостей алгоритмічної мови високого рівня Python, яка це все виконує. Результати роботи програми підтвердили правильність її виконання.