

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

ЗВІТ
з лабораторної роботи №6
з навчальної дисципліни «Методи наукових досліджень»

Тема:
Проведення трьохфакторного експерименту при використанні рівняння
регресії з квадратичними членами

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

Перевірив:
Регіда П.Г.

Київ 2021

Мета: Провести трьохфакторний експеримент і отримати адекватну модель – рівняння регресії, використовуючи рототабельний композиційний план.

Завдання на лабораторну роботу:

1. Ознайомитися з теоретичними відомостями.
2. Вибрати з таблиці варіантів і записати в протокол інтервали значень x_1 , x_2 , x_3 . Обчислити і записати значення, відповідні кодованим значенням факторів +1; -1; + ; - ; 0 для 1, 2, 3.
3. Значення функції відгуку знайти за допомогою підстановки в формулу:
 $y_i = f(x_1, x_2, x_3) + \text{random}(10) - 5$,
де $f(x_1, x_2, x_3)$ вибирається по номеру в списку в журналі викладача.
4. Провести експерименти і аналізуючи значення статистичних перевірок, отримати адекватну модель рівняння регресії. При розрахунках використовувати натуральні значення факторів.
5. Зробити висновки по виконаній роботі.

Варіант	x1		x2		x3		f(x1, x2, x3)
215	min	max	min	max	min	max	7,2+5,5*x1+6,3*x2+3,1*x3+4,6*x1*x1+ +0,2*x2*x2+3,2*x3*x3+4,3*x1*x2+ +0,9*x1*x3+8,0*x2*x3+7,0*x1*x2*x3
	10	50	-20	60	10	15	

Код програми:

```
from random import uniform
import numpy as np
import scipy.stats
from sklearn import linear_model

x1_min = 10; x1_max = 15
x2_min = -20; x2_max = 60
x3_min = 10; x3_max = 15

def f(x1, x2, x3):
    return 7.2 +
5.5*x1+6.3*x2+3.1*x3+4.6*x1*x1+0.2*x2*x2+3.2*x3*x3+4.3*x1*x2+0.9*x1*x3+8.0*x2*x3+7.0*
x1*x2*x3

l = 1.73
n = 14

x0_n = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
x1_n = (-1, -1, -1, -1, 1, 1, 1, 1, 1, 1, -1, 1, 0, 0, 0)
x2_n = (-1, -1, 1, 1, -1, -1, 1, 1, 1, 1, 0, 0, -1, 1, 0)
x3_n = (-1, 1, -1, 1, -1, 1, -1, 1, 0, 0, 0, 0, -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_squared_n = [x1_n[i] ** 2 for i in range(n)]
x2_squared_n = [x2_n[i] ** 2 for i in range(n)]
x3_squared_n = [x3_n[i] ** 2 for i in range(n)]

# values of factors for stellar points
def value(x_max, x_min, l):
    x0 = (x_max + x_min) / 2
    delta_x = x_max - x0
    return l * delta_x + x0

x1 = (x1_min, x1_min, x1_min, x1_min, x1_max, x1_max, x1_max, x1_max, value(x1_max,
x1_min, -1), value(x1_max, x1_min, 1),
      (x1_max + x1_min) / 2, (x1_max + x1_min) / 2, (x1_max + x1_min) / 2, (x1_max +
x1_min) / 2)
x2 = (x2_min, x2_min, x2_max, x2_max, x2_min, x2_min, x2_max, x2_max, (x2_max +
x2_min) / 2, (x2_max + x2_min) / 2,
      value(x2_max, x2_min, -1), value(x2_max, x2_min, 1), (x2_max + x2_min) / 2,
(x2_max + x2_min) / 2)
x3 = (x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, (x3_max +
x3_min) / 2, (x3_max + x3_min) / 2,
      (x3_max + x3_min) / 2, (x3_max + x3_min) / 2, value(x3_max, x3_min, -1),
value(x3_max, x3_min, 1))

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)]
x1_squared = [x1[i] ** 2 for i in range(n)]
x2_squared = [x2[i] ** 2 for i in range(n)]
x3_squared = [x3[i] ** 2 for i in range(n)]

def experiment(m):
    y = [(f(x1[i], x2[i], x3[i]) + uniform(0, 10) - 5) 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))

    b = list(zip(x0_n, x1_n, x2_n, x3_n, x1x2_n, x1x3_n, x2x3_n, x1x2x3_n,
x1_squared_n, x2_squared_n, x3_squared_n))
    skm = linear_model.LinearRegression(fit_intercept=False)
    skm.fit(b, y_response)
    b = skm.coef_
    b = [round(i, 3) for i in b]

    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 + {8}*x1^2 + {9}*x2^2 + {10}*x3^2\n'.format(round(b[0], 3),
round(b[1], 3), round(b[2], 3),
                                                                    round(b[3], 3), round(b[4], 3),
round(b[5], 3), round(b[6], 3),
                                                                    round(b[7], 3), round(b[8], 3),
round(b[9], 3), round(b[10], 3)))

```

```

# checking the homogeneity of the variance according to the Cochren's criterion
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.\n')

# 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(11)]

f3 = f1 * f2

d = 0
b_significant = []
for i in range(11):
    if t[i] < scipy.stats.t.ppf(q=0.975, df=f3):
        b_significant.append('b{0}'.format(i))
        b[i] = 0
    else:
        d += 1
    print('Коефіцієнти рівняння регресії {0} приймаємо незначним при рівні
значимості 0.05'.format(b_significant))

# 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] * x3_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('\nРівняння регресії неадекватно оригіналу при рівні значимості
0.05')
else:
    print('\nРівняння 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:

```
C:\Users\Lenovo\Anaconda3\python.exe D:/Users/Administrator/Desktop/Предмети/ANO/MND/lab_6.py
Введіть значення m: 7
Середні значення функції відгуку:
[-19213.698, -19212.613, -19210.812, -19212.695, -19212.584, -19212.75, -19209.524, -19212.087, -19212.576, -19216.593, -19211.343, -19212.834, -19211.164, -19216.274]

Отримане рівняння регресії:
y = -19790.388 + -0.291*x1 + 0.282*x2 + -0.884*x3 + 0.115*x1*x2 + -0.241*x1*x3 + -0.671*x2*x3 + 0.071*x1*x2*x3 + 192.39*x1^2 + 193.224*x2^2 + 192.679*x3^2

Дисперсія однорідна.

Коефіцієнти рівняння регресії ['b1', 'b2', 'b3', 'b4', 'b5', 'b6', 'b7', 'b8', 'b9', 'b10'] 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 проведено трьохфакторний експеримент і отримано адекватну модель – рівняння регресії, використовуючи рототабельний композиційний план. Було написано програму з використанням можливостей алгоритмічної мови високого рівня Python, яка це все виконує. Результати роботи програми підтвердили правильність її виконання.