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 х1, х2, х3. 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:

yi = f(х1, х2, х3) + random(10)-5,

де f(х1, х2, х3) вибирається по номеру в списку в журналі викладача.

1. Провести експерименти і аналізуючи значення статистичних перевірок, отримати адекватну модель рівняння регресії. При розрахунках

використовувати натуральні значення факторів.

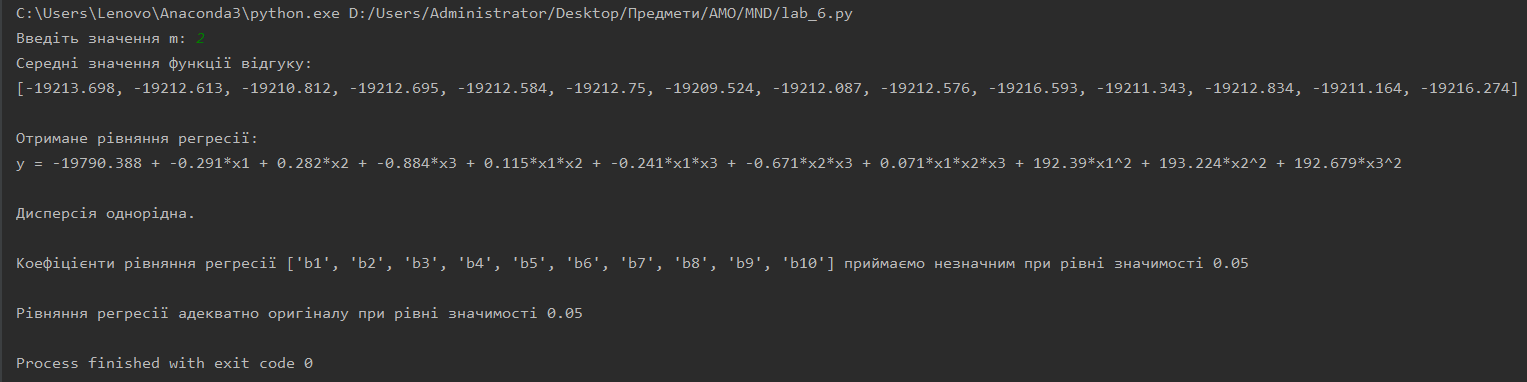
1. Зробити висновки по виконаній роботі.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Варіант | x1 | | x2 | | x3 | | f(х1, х2, х3) |
| 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, -l, l, 0, 0, 0, 0)  
x2\_n = (-1, -1, 1, 1, -1, -1, 1, 1, 0, 0, -l, l, 0, 0)  
x3\_n = (-1, 1, -1, 1, -1, 1, -1, 1, 0, 0, 0, 0, -l, l)  
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, -l), value(x1\_max, x1\_min, l),  
 (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, -l), value(x2\_max, x2\_min, l), (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, -l), value(x3\_max, x3\_min, l))  
  
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 експериметн 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:



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.