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 № 4

з дисципліни «МНД» на тему
«Проведення трьохфакторного експерименту
при використанні рівняння регресії з урахуванням ефекту взаємодії»

ВИКОНАЛА:
студентка ІІ курсу ФІОТ
групи ІВ-91
Бузулук М.В.
№ заліковки - 9103

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Київ – 2021

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

Варіант завдання:

№варіанта	x_1		x_2		x_3	
	min	max	min	Max	min	max
103	20	70	-20	40	70	80

Лістинг програми:

```
import random
from prettytable import PrettyTable

m = 3
n = 8

x1_min = 20
x1_max = 70

x2_min = -20
x2_max = 40

x3_min = 70
x3_max = 80

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

plan_matrix_normal = [[-1, -1, -1, 1, 1, 1, -1],
       [-1, -1, 1, 1, -1, -1, 1],
       [-1, 1, -1, -1, 1, -1, 1],
       [-1, 1, 1, -1, -1, 1, -1],
       [1, -1, -1, -1, -1, 1, 1],
       [1, -1, 1, -1, 1, -1, -1],
       [1, 1, -1, 1, -1, -1, -1],
       [1, 1, 1, 1, 1, 1, 1]]

plan_matrix = [[x1_min, x2_min, x3_min, x1_min * x2_min, x1_min * x3_min, x2_min
* x3_min, x1_min * x2_min * x3_min],
              [x1_min, x2_min, x3_max, x1_min * x2_min, x1_min * x3_max, x2_min *
 x3_max, x1_min * x2_min * x3_max],
              [x1_min, x2_max, x3_min, x1_min * x2_max, x1_min * x3_min, x2_max *
 x3_min, x1_min * x2_max * x3_min],
              [x1_min, x2_max, x3_max, x1_min * x2_max, x1_min * x3_max, x2_max *
 x3_max, x1_min * x2_max * x3_max],
              [x1_max, x2_min, x3_min, x1_max * x2_min, x1_max * x3_min, x2_min *
 x3_min, x1_max * x2_min * x3_min],
              [x1_max, x2_min, x3_max, x1_max * x2_min, x1_max * x3_max, x2_min *
 x3_max, x1_max * x2_min * x3_max],
```

```
            [x1_max, x2_max, x3_min, x1_max * x2_max, x1_max * x3_min, x2_max *
 x3_min, x1_max * x2_max * x3_min],
            [x1_max, x2_max, x3_max, x1_max * x2_max, x1_max * x3_max, x2_max *
 x3_max, x1_max * x2_max * x3_max]]
while True:
    y_matrix = [[random.randint(int(y_min), int(y_max)) for _ in range(m)] for _
in range(n)]

    average_y = [round(sum(i) / len(i), 3) for i in y_matrix]

    b0 = sum(average_y) / n
    b1 = sum([average_y[i] * plan_matrix_normal[i][0] for i in range(n)]) / n
    b2 = sum([average_y[i] * plan_matrix_normal[i][1] for i in range(n)]) / n
    b3 = sum([average_y[i] * plan_matrix_normal[i][2] for i in range(n)]) / n
    b12 = sum([average_y[i] * plan_matrix_normal[i][0] * plan_matrix_normal[i][1]
 for i in range(n)]) / n
    b13 = sum([average_y[i] * plan_matrix_normal[i][0] * plan_matrix_normal[i][2]
 for i in range(n)]) / n
    b23 = sum([average_y[i] * plan_matrix_normal[i][1] * plan_matrix_normal[i][2]
 for i in range(n)]) / n
    b123 = sum([average_y[i] * plan_matrix_normal[i][0] * plan_matrix_normal[i][1
] * plan_matrix_normal[i][2] for i in range(n)]) / n

    s = [sum([(y_matrix[j][i] -
 average_y[i]) ** 2 for i in range(m)]) / m for j in range(n)]
    gp = max(s) / sum(s)
    gt = 0.5157

    if gp > gt:
        m += 1
    else: break

d = 8

sb = sum(s) / n
s_beta_2 = sb / (n * m)
s_beta = s_beta_2 ** (1 / 2)

bb = [b0, b1, b2, b3, b12, b13, b23, b123]
t = [abs(bb[i]) / s_beta for i in range(n)]
tt = 2.120
blist = [b0, b1, b2, b3, b12, b13, b23, b123]

for i in range(n):
    if t[i] < tt:
        blist[i] = 0
        d -= 1

y_reg = [b0 + b1 * plan_matrix[i][0] + b2 * plan_matrix[i][1] + b3 * plan_matrix[
i][2] +
```

```
            b12 * plan_matrix[i][3] + b13 * plan_matrix[i][4] + b23 * plan_matri
x[i][5] +
            b123 * plan_matrix[i][6] for i in range(n)]
sad = (m / (n - d)) * int(sum([(y_reg[i] -
 average_y[i]) ** 2 for i in range(n)]))
fp = sad / sb
if fp > 4.5:
    toPrint = 'неадекватно оригіналу при рівні значимості 0.05'
else:
    toPrint = 'адекватно оригіналу при рівні значимості 0.05'

table = PrettyTable()

headers_x = ['X{}'.format(i) for i in range(0, m+1)]
headers_x.extend(['X12', 'X13', 'X23', 'X123'])
headers_y = ['Y{}'.format(i) for i in range(1, m+1)]
headers_y.extend(['av_Y', 'S^2'])

table.field_names = [*headers_x, *headers_y]
x0 = [[1] for _ in range(n)]
for i in range(n):
    table.add_row([*x0[i], *plan_matrix_normal[i], *y_matrix[i], average_y[i], s[
i]])

print(table)
print("Дисперсія однорідна за критерієм Кохрена")
print("Кількість значущих коефіцієнтів за критерієм Стьюдента: ", d)
print("За критерієм Фішера рівняння регресії ", toPrint)
print('Рівняння')
print('y = {} + {} * x1 + {} * x2 + {} * x3 + {} * x1x2 + {} * x1x3 + {} * x2x3 +
 {} * x1x2x3'
      .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)))
```

Результат роботи програми:

```
+----+----+----+----+-----+-----+-----+------+-----+-----+-----+---------+--------------------+
| X0 | X1 | X2 | X3 | X12 | X13 | X23 | X123 |  Y1 |  Y2 |  Y3 |   av_Y  |        S^2         |
+----+----+----+----+-----+-----+-----+------+-----+-----+-----+---------+--------------------+
| 1  | -1 | -1 | -1 |  1  |  1  |  1  |  -1  | 238 | 257 | 225 |  240.0  | 134.74592600000003 |
| 1  | -1 | -1 | 1  |  1  |  -1 |  -1 |  1   | 242 | 263 | 255 | 253.333 |     68.073926      |
| 1  | -1 | 1  | -1 |  -1 |  1  |  -1 |  1   | 253 | 252 | 229 | 244.667 | 138.74392600000002 |
| 1  | -1 | 1  | 1  |  -1 |  -1 |  1  |  -1  | 236 | 231 | 240 | 235.667 | 178.84792599999994 |
| 1  | 1  | -1 | -1 |  -1 |  -1 |  1  |  1   | 247 | 260 | 232 | 246.333 | 84.63392600000002  |
| 1  | 1  | -1 | 1  |  -1 |  1  |  -1 |  -1  | 247 | 244 | 249 | 246.667 | 51.626592666666646 |
| 1  | 1  | 1  | -1 |  1  |  -1 |  -1 |  -1  | 260 | 237 | 249 | 248.667 |     228.513926     |
| 1  | 1  | 1  | 1  |  1  |  1  |  1  |  1   | 230 | 241 | 260 | 243.667 | 162.4012593333333  |
+----+----+----+----+-----+-----+-----+------+-----+-----+-----+---------+--------------------+
Дисперсія однорідна за критерієм Кохрена
Кількість значущих коефіцієнтів за критерієм Стьюдента:  1
За критерієм Фішера рівняння регресії  неадекватно оригіналу при рівні значимості 0.05
Рівняння
y = 244.875 + 1.458 * x1 + -1.708 * x2 + -0.042 * x3 + 1.542 * x1x2 + -1.125 * x1x3 + -3.458 * x2x3 + 2.125 * x1x2x3
```