Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Методи оптимізації та планування експерименту

Лабораторна робота №2

“ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ”

Виконав:

студент II курсу гр. ІВ-83

Герасімов С.С.

Перевірив:

ас. Регіда П.Г.

Варiант: 305

Київ-2020 р.

ymax = (30 - 5)\*10 = 25\*10 = 250

ymin = (20 - 5)\*10 = 15\*10 = 150

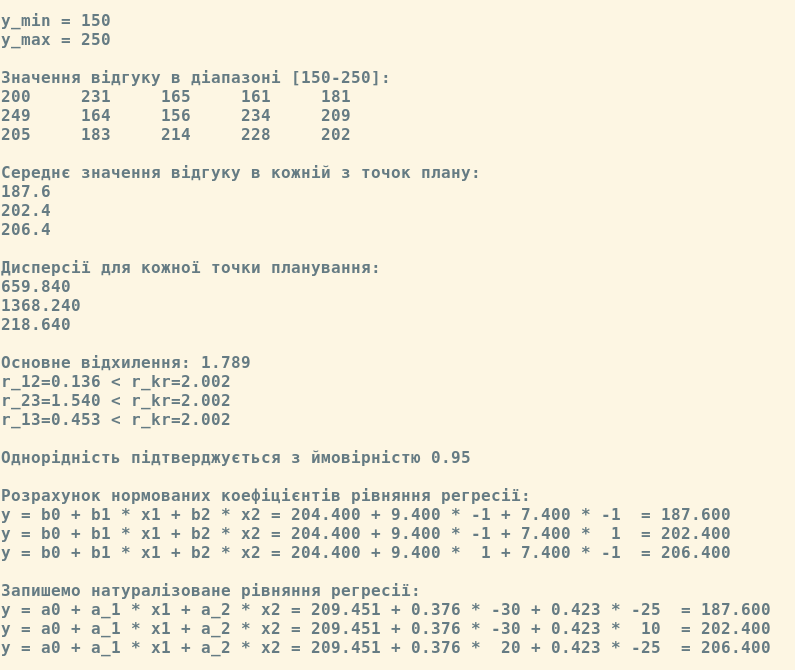




**Лістинг програми:**

# Нехай довірча ймовірність = 0.95  
  
from math import \*  
from random import \*  
  
m = 5  
  
def determinant(matrix):  
 return matrix[0][0] \* matrix[1][1] \* matrix[2][2] + matrix[0][1] \* matrix[1][2] \* matrix[2][0] + matrix[0][2] \* \  
 matrix[1][0] \* matrix[2][1] - matrix[0][2] \* matrix[1][1] \* matrix[2][0] - matrix[0][1] \* matrix[1][0] \* \  
 matrix[2][2] - matrix[0][0] \* matrix[1][2] \* matrix[2][1]  
  
def rkr(m):  
 table = {2: 1.71, 6: 2.10, 8: 2.27, 10: 2.41, 12: 2.52, 15: 2.64, 20: 2.78}  
  
 for i in range(len(table.keys())):  
 if m == list(table.keys())[i]:  
 return list(table.values())[i]  
 if m > list(table.keys())[i]:  
 smaller\_than\_mkey = list(table.keys())[i]  
 smaller\_than\_m = list(table.values())[i]  
 more\_than\_mkey = list(table.keys())[i+1]  
 more\_than\_m = list(table.values())[i+1]  
 return smaller\_than\_m + (more\_than\_m - smaller\_than\_m) \* (m - smaller\_than\_mkey) / (more\_than\_mkey - smaller\_than\_mkey)  
  
variant = 5  
y\_max = (30 - variant) \* 10  
y\_min = (20 - variant) \* 10  
  
def main():  
 global m  
  
 response\_list\_1 = [randint(y\_min, y\_max) for i in range(m)]  
 response\_list\_2 = [randint(y\_min, y\_max) for j in range(m)]  
 response\_list\_3 = [randint(y\_min, y\_max) for k in range(m)]  
  
 average\_1 = sum(response\_list\_1) / len(response\_list\_1)  
 average\_2 = sum(response\_list\_2) / len(response\_list\_2)  
 average\_3 = sum(response\_list\_3) / len(response\_list\_3)  
  
 dispersion\_1 = sum((i - average\_1) \*\* 2 for i in response\_list\_1) / len(response\_list\_1)  
 dispersion\_2 = sum((i - average\_2) \*\* 2 for i in response\_list\_2) / len(response\_list\_2)  
 dispersion\_3 = sum((i - average\_3) \*\* 2 for i in response\_list\_3) / len(response\_list\_3)  
  
 major\_deviation = sqrt((4 \* m - 4) / (m \* m - 4 \* m))  
  
 f\_12 = dispersion\_1 / dispersion\_2 if dispersion\_1 >= dispersion\_2 else dispersion\_2 / dispersion\_1  
 f\_23 = dispersion\_2 / dispersion\_3 if dispersion\_2 >= dispersion\_3 else dispersion\_3 / dispersion\_2  
 f\_13 = dispersion\_1 / dispersion\_3 if dispersion\_1 >= dispersion\_3 else dispersion\_3 / dispersion\_1  
  
 t\_12 = (m - 2) / m \* f\_12  
 t\_23 = (m - 2) / m \* f\_23  
 t\_13 = (m - 2) / m \* f\_13  
  
 r\_12 = abs(t\_12 - 1) / major\_deviation  
 r\_23 = abs(t\_23 - 1) / major\_deviation  
 r\_13 = abs(t\_13 - 1) / major\_deviation  
  
 r\_kr = rkr(m)  
  
 print('\ny\_min =', y\_min, '\ny\_max =', y\_max)  
  
 print(f'\nЗначення відгуку в діапазоні [{y\_min}-{y\_max}]:')  
 print(\*response\_list\_1, sep='\t')  
 print(\*response\_list\_2, sep='\t')  
 print(\*response\_list\_3, sep='\t')  
  
 print('\nСереднє значення відгуку в кожній з точок плану:', '\n' + f'{average\_1}', '\n' + f'{average\_2}', '\n' + f'{average\_3}')  
 print('\nДисперсії для кожної точки планування:', '\n' + f'{dispersion\_1:.3f}', '\n' + f'{dispersion\_2:.3f}', '\n' + f'{dispersion\_3:.3f}')  
 print('\nОсновне відхилення:', f'{major\_deviation:.3f}')  
  
 print('r\_12=' + f'{r\_12:.3f}', '<' if r\_12 < r\_kr else '>', 'r\_kr=' + f'{r\_kr:.3f}')  
 print('r\_23=' + f'{r\_23:.3f}', '<' if r\_23 < r\_kr else '>', 'r\_kr=' + f'{r\_kr:.3f}')  
 print('r\_13=' + f'{r\_13:.3f}', '<' if r\_13 < r\_kr else '>', 'r\_kr=' + f'{r\_kr:.3f}')  
  
 if r\_12 < r\_kr and r\_23 < r\_kr and r\_13 < r\_kr:  
 print('\nОднорідність підтверджується з ймовірністю 0.95')  
 normalized\_x1\_x2 = [[-1, -1], [-1, 1], [1, -1]]  
 mx\_list = [sum(i) / len(i) for i in list(zip(normalized\_x1\_x2[0], normalized\_x1\_x2[1], normalized\_x1\_x2[2]))]  
 my = sum([average\_1, average\_2, average\_3]) / len([average\_1, average\_2, average\_3])  
   
 a\_1 = sum(i[0] \*\* 2 for i in normalized\_x1\_x2) / len(normalized\_x1\_x2)  
 a\_2 = sum(i[0] \* i[1] for i in normalized\_x1\_x2) / len(normalized\_x1\_x2)  
 a\_3 = sum(i[1] \*\* 2 for i in normalized\_x1\_x2) / len(normalized\_x1\_x2)  
 a\_11 = sum(normalized\_x1\_x2[i][0] \* [average\_1, average\_2, average\_3][i] for i in range(len(normalized\_x1\_x2))) / len(normalized\_x1\_x2)  
 a\_22 = sum(normalized\_x1\_x2[i][1] \* [average\_1, average\_2, average\_3][i] for i in range(len(normalized\_x1\_x2))) / len(normalized\_x1\_x2)  
   
 matrix\_b = [ [1, mx\_list[0], mx\_list[1]], [mx\_list[0], a\_1, a\_2], [mx\_list[1], a\_2, a\_3] ]  
 matrix\_b\_1 = [ [my, mx\_list[0], mx\_list[1]], [a\_11, a\_1, a\_2], [a\_22, a\_2, a\_3] ]  
 matrix\_b\_2 = [ [1, my, mx\_list[1]], [mx\_list[0], a\_11, a\_2], [mx\_list[1], a\_22, a\_3] ]  
 matrix\_b\_3 = [ [1, mx\_list[0], my], [mx\_list[0], a\_1, a\_11], [mx\_list[1], a\_2, a\_22] ]  
  
 b0 = determinant(matrix\_b\_1) / determinant(matrix\_b)  
 b1 = determinant(matrix\_b\_2) / determinant(matrix\_b)  
 b2 = determinant(matrix\_b\_3) / determinant(matrix\_b)  
  
 print('\nРозрахунок нормованих коефіцієнтів рівняння регресії:')  
 for i in normalized\_x1\_x2:  
 print(f'y = b0 + b1 \* x1 + b2 \* x2 = {b0:.3f} + {b1:.3f} \* {i[0]:2} + {b2:.3f} \* {i[1]:2}', f' = {b0 + b1 \* i[0] + b2 \* i[1]:.3f}')  
  
 x1\_min = -30  
 x1\_max = 20  
 x2\_min = -25  
 x2\_max = 10  
  
 x\_10 = (x1\_max + x1\_min) / 2  
 x\_20 = (x2\_max + x2\_min) / 2  
  
 delta\_x1 = (x1\_max - x1\_min) / 2  
 delta\_x2 = (x2\_max - x2\_min) / 2  
  
 a\_0 = b0 - b1 \* (x\_10 / delta\_x1) - b2 \* (x\_20 / delta\_x2)  
 a\_1 = b1 / delta\_x1  
 a\_2 = b2 / delta\_x2  
  
 print('\nЗапишемо натуралізоване рівняння регресії:')  
 print(f'y = a0 + a\_1 \* x1 + a\_2 \* x2 = {a\_0:.3f} + {a\_1:.3f} \* {x1\_min:3} + {a\_2:.3f} \* {x2\_min:3}', f' = {a\_0 + a\_1 \* x1\_min + a\_2 \* x2\_min:.3f}')  
 print(f'y = a0 + a\_1 \* x1 + a\_2 \* x2 = {a\_0:.3f} + {a\_1:.3f} \* {x1\_min:3} + {a\_2:.3f} \* {x2\_max:3}', f' = {a\_0 + a\_1 \* x1\_min + a\_2 \* x2\_max:.3f}')  
 print(f'y = a0 + a\_1 \* x1 + a\_2 \* x2 = {a\_0:.3f} + {a\_1:.3f} \* {x1\_max:3} + {a\_2:.3f} \* {x2\_min:3}', f' = {a\_0 + a\_1 \* x1\_max + a\_2 \* x2\_min:.3f}')  
 print()  
  
 else:  
 print('\nОднорідність не підтвердилася, підвищуємо m на 1\n')  
 m += 1  
 main()  
  
main()

**Результат роботи програми:**



**Контрольні запитання:**

1. Що таке регресійні поліноми і де вони застосовуються?

Регресійний поліном – це рівняння регресії виду



використовується в ТПЕ для оцінки результатів вимірів.

1. Визначення однорідності дисперсії.

Однорідність дисперсій – властивість, коли дисперсії вимірювання функцій відгуку є однаковими, або близькими.

1. Що називається повним факторним експериментом?

ПФЕ – експеримент, в якому використовуються всі можливі комбінації рівнів факторів.