Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 4

з дисципліни «Методи наукових досліджень»

на тему «Проведення трьохфакторного експерименту

при використанні рівняння регресії з урахуванням ефекту взаємодії.»

ВИКОНАВ:

студент 2 курсу

групи ІВ-91

Онищук Ю.І.

Залікова – 9122

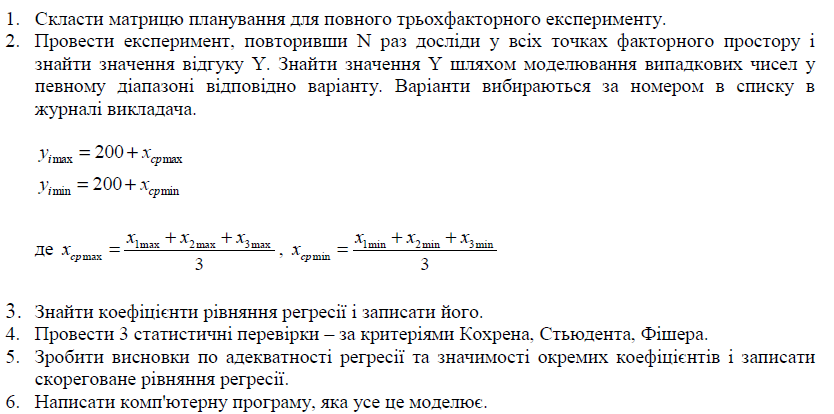
ПЕРЕВІРИВ:

ас. Регіда П. Г.

Київ – 2021

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

**Завдання на лабораторну роботу**

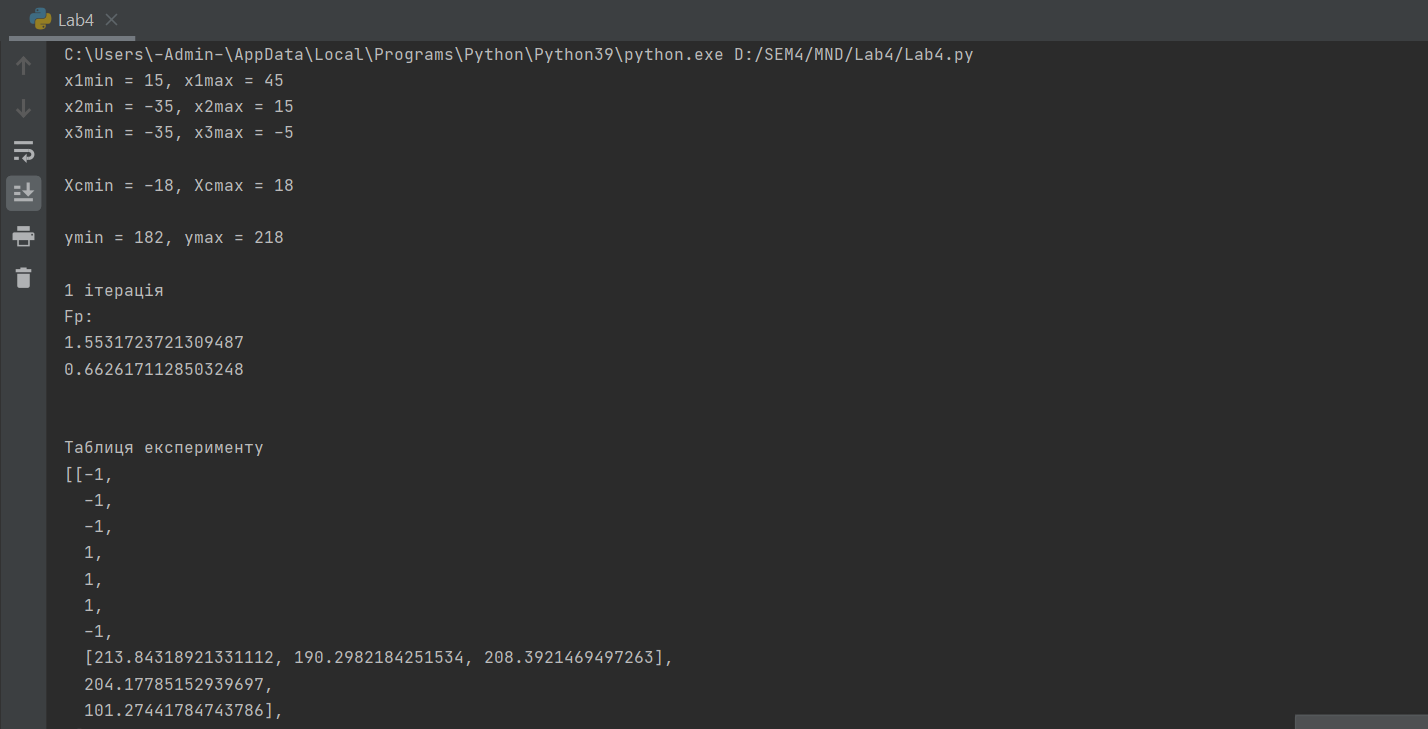


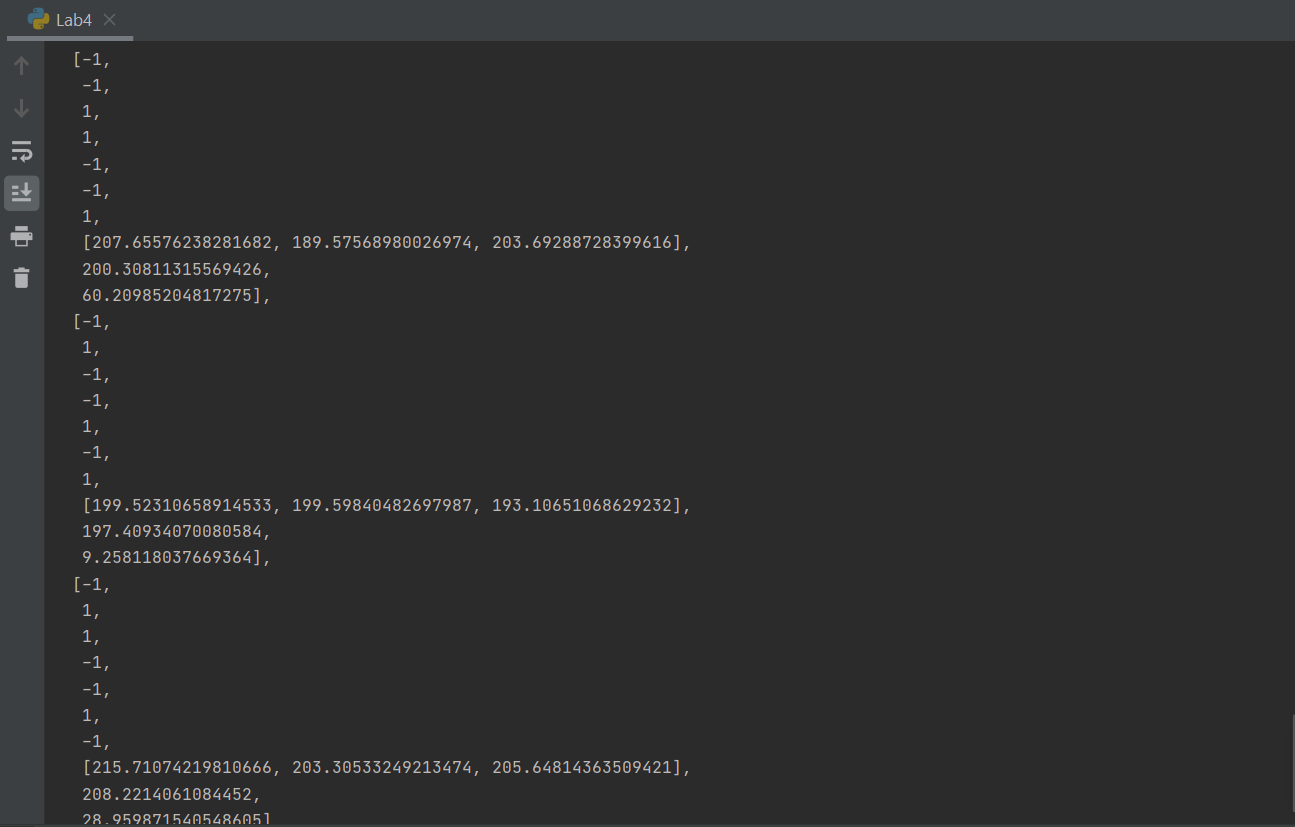
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № варіанту | x1 | | x2 | | x3 | |
| min | max | min | max | min | max |
| 120 | 15 | 45 | -35 | 15 | -35 | -5 |

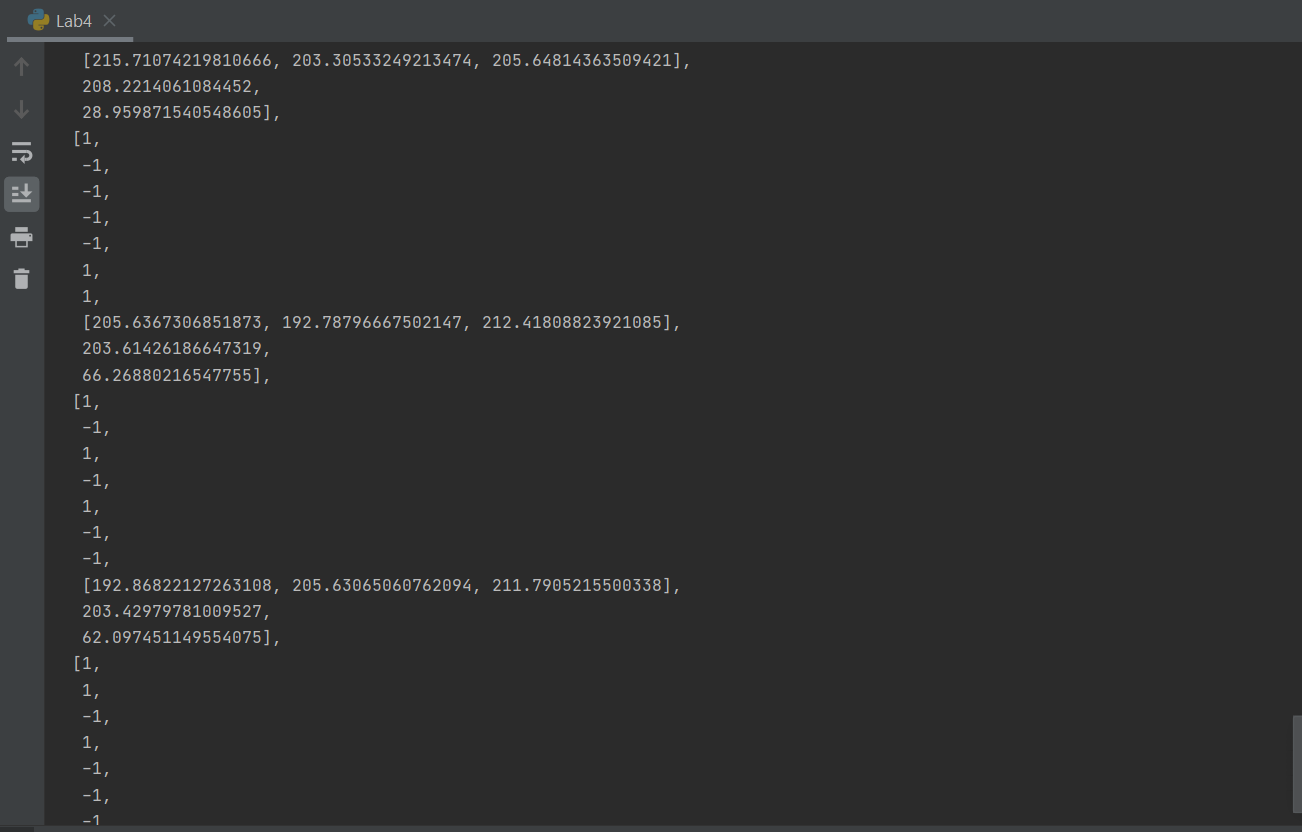
**Програмний код**

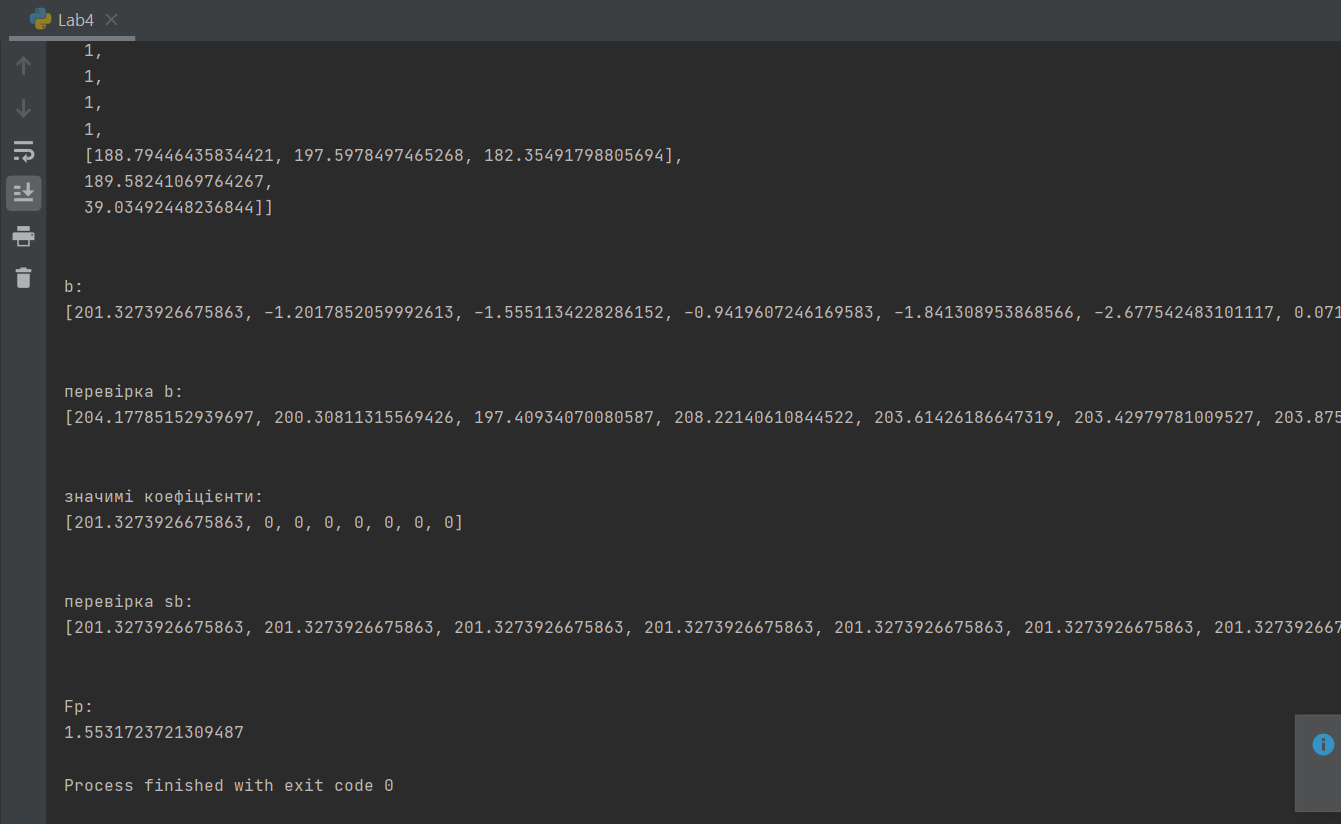
import random as rn  
from math import sqrt  
import pprint  
  
def exp\_row(x1\_num, x2\_num, x3\_num, y, m = 3):  
 y\_gen = [y[0] + (y[1]-y[0])\*rn.random() for i in range(m)]  
 y\_mid = sum(y\_gen)/m  
 sigma = 0  
 for i in range(len(y\_gen)):  
 sigma += ((y\_gen[i] - y\_mid)\*\*2)/m  
 return [x1\_num, x2\_num, x3\_num, x1\_num\*x2\_num, x1\_num\*x3\_num, x2\_num\*x3\_num, x1\_num\*x2\_num\*x3\_num, y\_gen, y\_mid, sigma]  
  
def experiment(x1, x2, x3, y, m = 3, N = 8):  
 flag = 1  
 counter = 1  
 while flag:  
 flag = 0  
 exp1 = exp\_row(x1[0], x2[0], x3[0], y)  
 exp2 = exp\_row(x1[0], x2[0], x3[1], y)  
 exp3 = exp\_row(x1[0], x2[1], x3[0], y)  
 exp4 = exp\_row(x1[0], x2[1], x3[1], y)  
 exp5 = exp\_row(x1[1], x2[0], x3[0], y)  
 exp6 = exp\_row(x1[1], x2[0], x3[1], y)  
 exp7 = exp\_row(x1[1], x2[1], x3[0], y)  
 exp8 = exp\_row(x1[1], x2[1], x3[1], y)  
  
 table = [exp1, exp2, exp3, exp4, exp5, exp6, exp7, exp8]  
 norm\_table = normalize\_table(table, x1, x2, x3)  
  
  
 cochrane = cochrane\_kriteria(table, 8)  
 cochrane\_check = cochrane[0]  
 Gp = cochrane[1]  
  
 if not cochrane\_check:  
 flag = 1  
 continue  
 else:  
 b0 = 0  
 b1 = 0  
 b2 = 0  
 b3 = 0  
 b12 = 0  
 b13 = 0  
 b23 = 0  
 b123 = 0  
  
 for i in range(N):  
 b0 += 1/N\*(norm\_table[i][-2])  
 b1 += 1/N\*(norm\_table[i][-2]\*norm\_table[i][0])  
 b2 += 1/N\*(norm\_table[i][-2]\*norm\_table[i][1])  
 b3 += 1/N\*(norm\_table[i][-2]\*norm\_table[i][2])  
 b12 += 1/N\*(norm\_table[i][-2]\*norm\_table[i][3])  
 b13 += 1/N\*(norm\_table[i][-2]\*norm\_table[i][4])  
 b23 += 1/N\*(norm\_table[i][-2]\*norm\_table[i][5])  
 b123 += 1/N\*(norm\_table[i][-2]\*norm\_table[i][6])  
  
 b = [b0, b1, b2, b3, b12, b13, b23, b123]  
  
 check\_b = check(b, table)  
  
 student = student\_kriteria(table, 8)  
 indexes = student[0]  
 SB = student[1]  
 Sbeta = student[2]  
 student\_b = list(map(lambda x: x if b.index(x) in indexes else 0, b))  
 student\_checks = check(student\_b, table)  
  
 fisher = fisher\_kriteria(table, student\_checks, SB)  
 fisher\_check = fisher[0]  
 Fp = fisher[1]  
 print('{} ітерація'.format(counter))  
 print('Fp:')  
 print(Fp)  
 print(Gp\*Sbeta\*Fp)  
 print('\n')  
 if fisher\_check:  
 return table, b, check\_b, student\_b, student\_checks, Fp  
 else:  
 flag = 1  
 counter += 1  
 x1 = list(map(lambda x: x/(Gp\*Sbeta\*Fp), x1))  
 x2 = list(map(lambda x: x/(Gp\*Sbeta\*Fp), x2))  
 x3 = list(map(lambda x: x/(Gp\*Sbeta\*Fp), x3))  
 xcmax = int(1/3\*(x1[1] + x2[1] + x3[1]))  
 xcmin = int(1/3\*(x1[0] + x2[0] + x3[0]))  
 ymax = 200 + xcmax  
 ymin = 200 + xcmin  
 y = [ymax, ymin]  
  
def normalize\_table(table, x1, x2, x3):  
 for row in table:  
 for j in range(3):  
 if j == 0:  
 if row[j] == x1[0]:  
 row[j] = -1  
 else:  
 row[j] = 1  
 elif j == 1:  
 if row[j] == x2[0]:  
 row[j] = -1  
 else:  
 row[j] = 1  
 elif j == 2:  
 if row[j] == x3[0]:  
 row[j] = -1  
 else:  
 row[j] = 1  
 row[3] = row[0]\*row[1]  
 row[4] = row[0]\*row[2]  
 row[5] = row[1]\*row[2]  
 row[6] = row[0]\*row[1]\*row[2]  
 return table  
  
def check(koeficients, table):  
 checks = [koeficients[0] for i in range(8)]  
  
 for i in range(len(table)):  
 for j in range(7):  
 checks[i] += koeficients[j+1]\*table[i][j]  
 return checks  
  
def cochrane\_kriteria(table, N = 8):  
 sigma = [table[i][-1] for i in range(N)]  
 Gt = 0.5157  
 Gp = max(sigma)/sum(sigma)  
 if Gp < Gt:  
 return 1, Gp  
 else:  
 return 0, Gp  
  
def student\_kriteria(table, N = 8, m = 3):  
 sigma = [table[i][-1] for i in range(N)]  
 y\_mid = [table[i][-2] for i in range(N)]  
 SB = sum(sigma)/N  
 Sb = SB/(N\*m)  
 Sbeta = sqrt(Sb)  
  
 beta = [0]\*N  
 for i in range(N):  
 beta[0] += 1/N\*(y\_mid[i]\*1)  
 for i in range(1, len(beta)):  
 for j in range(N):  
 beta[i] += 1/N\*(y\_mid[j]\*table[j][i-1])  
  
 ttab = 2.120  
  
 t = [0]\*N  
 for i in range(N):  
 t[i] = beta[i]/Sbeta  
  
 indexes = []  
 for i in t:  
 if i > ttab:  
 indexes.append(t.index(i))  
  
 return [indexes, SB, Sbeta]  
  
def fisher\_kriteria(table, checks, Sb, N = 8, m = 3, d = 1):  
 y\_mid = [table[i][-2] for i in range(N)]  
 Sad = 0  
 for i in range(N):  
 Sad += m/(N-d)\*(checks[i] - y\_mid[i])\*\*2  
 Ft = 2.7  
 Fp = Sad/Sb  
 if Fp > Ft:  
 return 0, Fp  
 else:  
 return 1, Fp  
  
  
  
x1 = [15, 45]  
print('x1min = {0}, x1max = {1}'.format(x1[0], x1[1]))  
x2 = [-35, 15]  
print('x2min = {0}, x2max = {1}'.format(x2[0], x2[1]))  
x3 = [-35, -5]  
print('x3min = {0}, x3max = {1}\n'.format(x3[0], x3[1]))  
  
  
xcmax = int(1/3\*(x1[1] + x2[1] + x3[1]))  
xcmin = int(1/3\*(x1[0] + x2[0] + x3[0]))  
print('Xcmin = {0}, Xcmax = {1}\n'.format(xcmin, xcmax))  
ymax = 200 + xcmax  
ymin = 200 + xcmin  
y = [ymin, ymax]  
print('ymin = {0}, ymax = {1}\n'.format(ymin, ymax))  
  
  
research = experiment(x1, x2, x3, y)  
table = research[0]  
koefs\_b = research[1]  
check\_b = research[2]  
koefs\_sb = research[3]  
check\_sb = research[4]  
Fp = research[5]  
  
print('Таблиця експерименту')  
pprint.pprint(table)  
print('\n')  
print('b:')  
print(koefs\_b)  
print('\n')  
print('перевірка b:')  
print(check\_b)  
print('\n')  
print('значимі коефіцієнти:')  
print(koefs\_sb)  
print('\n')  
print('перевірка sb:')  
print(check\_sb)  
print('\n')  
print('Fp:')  
print(Fp)

**Результат роботи програми**









**Висновок:**

Під час виконання даної лабораторної роботи я провів трьохфакторний експеримент, перевірив однорідність дисперсії за критерієм Кохрена, отримав коефіцієнти рівняння регресії, оцінив значимість знайдених коефіцієнтів за критеріями Стьюдента та Фішера.

Мета лабораторної роботи була досягнута.