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 №4:

«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ ВИКОРИСТАННІ РІВНЯННЯ РЕГРЕСІЇ З УРАХУВАННЯМ ЕФЕКТУ ВЗАЄМОДІЇ»

Виконав:

студент групи ІВ-81

Соколов В.С.

Залікова книжка № 8126

Перевірив:

Регіда П. Г.

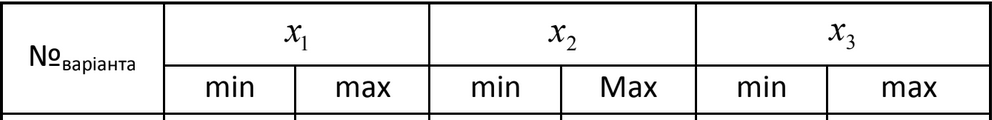
Київ 2020р.

**Тема:** проведення трьохфакторного експерименту при використанні рівняння регресії з урахуванням ефекту взаємодії.

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

**Виконання:**

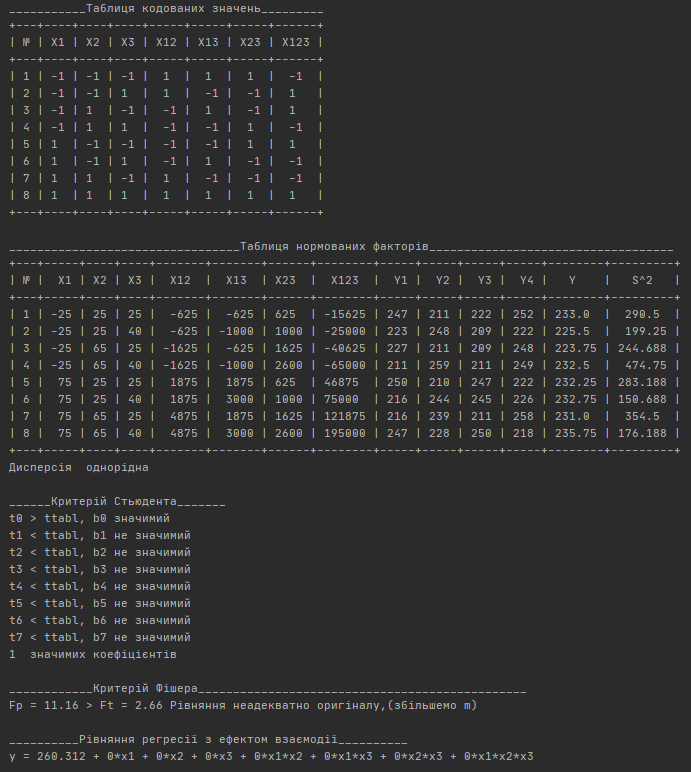
Варіант – 115.



**Код програми**

import random  
import scipy.stats  
from prettytable import PrettyTable  
from numpy.linalg import solve  
  
x1min = -25  
x1max = 75  
x2min = 25  
x2max = 65  
x3min = 25  
x3max = 40  
  
xAvmax = (x1max + x2max + x3max) / 3  
xAvmin = (x1min + x2min + x3min) / 3  
ymax = int(200 + xAvmax)  
ymin = int(200 + xAvmin)  
  
m = 4  
  
Xi = [[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],  
 [-1, 1, 1, -1, 1, -1, -1, 1]]  
  
  
def sumkf2(x1, x2):  
 return [x1[i] \* x2[i] for i in range(len(x1))]  
  
  
def sumkf3(x1, x2, x3):  
 return [x1[i] \* x2[i] \* x3[i] for i in range(len(x1))]  
  
  
X12 = sumkf2(Xi[1], Xi[2])  
X13 = sumkf2(Xi[1], Xi[3])  
X23 = sumkf2(Xi[2], Xi[3])  
X123 = sumkf3(Xi[1], Xi[2], Xi[3])  
X8 = list(map(lambda el: el \* el, Xi[1]))  
X9 = list(map(lambda el: el \* el, Xi[2]))  
X10 = list(map(lambda el: el \* el, Xi[3]))  
  
  
print("\_\_\_\_\_\_\_\_\_\_\_Таблиця кодованих значень\_\_\_\_\_\_\_\_\_")  
table1 = PrettyTable()  
table1.add\_column("№", (1, 2, 3, 4, 5, 6, 7, 8))  
table1.add\_column("X1", Xi[1])  
table1.add\_column("X2", Xi[2])  
table1.add\_column("X3", Xi[3])  
table1.add\_column("X12", X12)  
table1.add\_column("X13", X13)  
table1.add\_column("X23", X23)  
table1.add\_column("X123", X123)  
print(table1)  
  
Y = [[random.randrange(ymin, ymax, 1) for \_ in range(8)] for \_\_ in range(m)]  
  
X1 = [x1min, x1min, x1min, x1min, x1max, x1max, x1max, x1max]  
X2 = [x2min, x2min, x2max, x2max, x2min, x2min, x2max, x2max]  
X3 = [x3min, x3max, x3min, x3max, x3min, x3max, x3min, x3max]  
X12 = sumkf2(X1, X2)  
X13 = sumkf2(X1, X3)  
X23 = sumkf2(X2, X3)  
X123 = sumkf3(X1, X2, X3)  
X0 = [1] \* 8  
  
s = [sum([Y[i][j] for i in range(m)]) for j in range(8)]  
  
yav = [round(s[i] / m, 3) for i in range(8)]  
  
sd = [sum([((Y[i][j]) - yav[j]) \*\* 2 for i in range(m)]) for j in range(8)]  
  
d = [sd[i] / m for i in range(8)]  
  
disper = [round(d[i], 3) for i in range(8)]  
  
  
print("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Таблиця нормованих факторів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
table2 = PrettyTable()  
table2.add\_column("№", (1, 2, 3, 4, 5, 6, 7, 8))  
table2.add\_column("X1", X1)  
table2.add\_column("X2", X2)  
table2.add\_column("X3", X3)  
table2.add\_column("X12", X12)  
table2.add\_column("X13", X13)  
table2.add\_column("X23", X23)  
table2.add\_column("X123", X123)  
for i in range(m):  
 table2.add\_column("Y" + str(i+1), Y[i])  
table2.add\_column("Y", yav)  
table2.add\_column("S^2", disper)  
print(table2)  
  
b = [round(i, 3) for i in solve(list(zip(X0, X1, X2, X3, X12, X13, X23, X123)), yav)]  
  
m = 3  
Gp = max(d) / sum(d)  
q = 0.05  
f1 = m - 1  
f2 = N = 8  
fisher = scipy.stats.f.isf(\*[q / f2, f1, (f2 - 1) \* f1])  
Gt = fisher / (fisher + (f2 - 1))  
  
  
if Gp < Gt:  
 print("Дисперсія однорідна")  
 print("\n\_\_\_\_\_\_Критерій Стьюдента\_\_\_\_\_\_\_")  
 sb = sum(d) / N  
 ssbs = sb / N \* m  
 sbs = ssbs \*\* 0.5  
  
 beta = [sum([yav[j] \* Xi[i][j] for j in range(8)]) / 8 for i in range(8)]  
  
 t = [abs(beta[i]) / sbs for i in range(8)]  
  
 f3 = f1 \* f2  
 ttabl = round(abs(scipy.stats.t.ppf(q / 2, f3)), 4)  
  
 d\_ = 8  
  
 for i in range(8):  
 if t[i] < ttabl:  
 print(f"t{i} < ttabl, b{i} не значимий")  
 b[i] = 0  
 d\_ -= 1  
 else:  
 print(f"t{i} > ttabl, b{i} значимий")  
  
 print(d\_, " значимих коефіцієнтів")  
  
 yy1 = b[0] + b[1] \* x1min + b[2] \* x2min + b[3] \* x3min + b[4] \* x1min \* x2min + b[5] \* x1min \* x3min + b[6] \* x2min \* x3min + b[7] \* x1min \* x2min \* x3min  
 yy2 = b[0] + b[1] \* x1min + b[2] \* x2min + b[3] \* x3max + b[4] \* x1min \* x2min + b[5] \* x1min \* x3max + b[6] \* x2min \* x3max + b[7] \* x1min \* x2min \* x3max  
 yy3 = b[0] + b[1] \* x1min + b[2] \* x2max + b[3] \* x3min + b[4] \* x1min \* x2max + b[5] \* x1min \* x3min + b[6] \* x2max \* x3min + b[7] \* x1min \* x2max \* x3min  
 yy4 = b[0] + b[1] \* x1min + b[2] \* x2max + b[3] \* x3max + b[4] \* x1min \* x2max + b[5] \* x1min \* x3max + b[6] \* x2max \* x3max + b[7] \* x1min \* x2max \* x3max  
  
 yy5 = b[0] + b[1] \* x1max + b[2] \* x2min + b[3] \* x3min + b[4] \* x1max \* x2min + b[5] \* x1max \* x3min + b[6] \* x2min \* x3min + b[7] \* x1max \* x2min \* x3min  
 yy6 = b[0] + b[1] \* x1max + b[2] \* x2min + b[3] \* x3max + b[4] \* x1max \* x2min + b[5] \* x1max \* x3max + b[6] \* x2min \* x3max + b[7] \* x1max \* x2min \* x3max  
 yy7 = b[0] + b[1] \* x1max + b[2] \* x2max + b[3] \* x3min + b[4] \* x1max \* x2max + b[5] \* x1max \* x3min + b[6] \* x2max \* x3min + b[7] \* x1max \* x2min \* x3max  
 yy8 = b[0] + b[1] \* x1max + b[2] \* x2max + b[3] \* x3max + b[4] \* x1max \* x2max + b[5] \* x1max \* x3max + b[6] \* x2max \* x3max + b[7] \* x1max \* x2max \* x3max  
 print("\n\_\_\_\_\_\_\_\_\_\_\_\_Критерій Фішера\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
 f4 = N - d\_  
 sad = ((yy1 - yav[0]) \*\* 2 + (yy2 - yav[1]) \*\* 2 + (yy3 - yav[2]) \*\* 2 + (yy4 - yav[3]) \*\* 2 + (yy5 - yav[4]) \*\* 2 + (  
 yy6 - yav[5]) \*\* 2 + (yy7 - yav[6]) \*\* 2 + (yy8 - yav[7]) \*\* 2) \* (m / (N - d\_))  
 Fp = sad / sb  
  
 Ft = abs(scipy.stats.f.isf(q, f4, f3))  
  
 if Fp > Ft:  
 print("Fp = {:.2f} > Ft = {:.2f} Рівняння неадекватно оригіналу,(збільшемо m)".format(Fp, Ft))  
 m += 1  
 else:  
 print("Fp = {:.2f} < Ft = {:.2f} Рівняння адекватно оригіналу".format(Fp, Ft))  
  
else:  
 print("Дисперсія неоднорідна (збільшемо кількість дослідів)")  
 m += 1  
print("\n\_\_\_\_\_\_\_\_\_\_Рівняння регресії з ефектом взаємодії\_\_\_\_\_\_\_\_\_\_")  
print("y = {} + {}\*x1 + {}\*x2 + {}\*x3 + {}\*x1\*x2 + {}\*x1\*x3 + {}\*x2\*x3 + {}\*x1\*x2\*x3".format(\*b))

**Результат виконання роботи програми:**



**Висновок**

Отже, у ході виконання лабораторної роботи № 4 ми провели повний трьохфакторний експеримент при використанні рівняння з ефектом взаємодії. Склали матрицю планування, знайшли коефіцієнти рівняння регресії, провели 3 статистичні перевірки. Була написана тестова програма, результати наведені вище.