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 №3:

«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ

ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконав:

студент групи ІО-83

Мітячкін Денис

Залікова книжка № 8320

Перевірив Регіда П. Г.

Київ 2020р.

**Лабораторна робота №3**

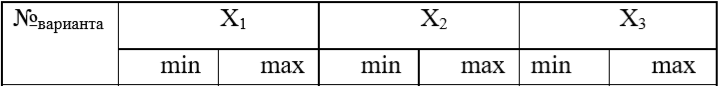
**Тема:** ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ

ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ

**Мета:** провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти

коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

**Виконання:**

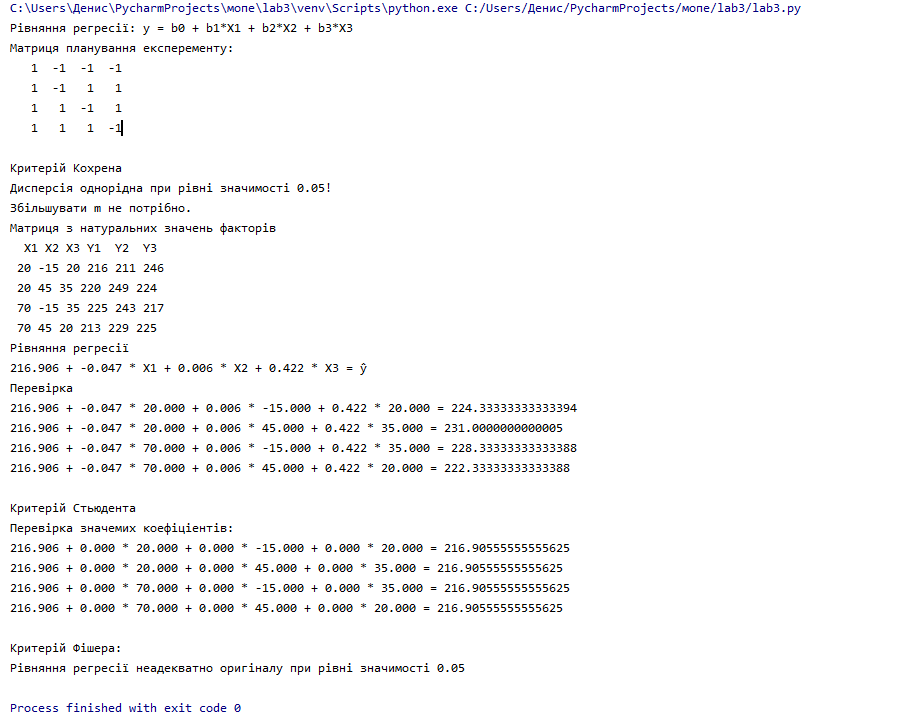




**Код програми:**

from numpy.linalg import det  
# визначник матриці  
  
from math import fabs, sqrt  
from random import randrange  
from \_pydecimal import Decimal  
from scipy.stats import f, t  
  
# Значення за варіантом  
m = 3  
N = 4  
p = 0.95  
  
min\_x1 = 20  
max\_x1 = 70  
min\_x2 = -15  
max\_x2 = 45  
min\_x3 = 20  
max\_x3 = 35  
  
min\_y = 200 + int((min\_x1 + min\_x2 + min\_x3) / 3)  
max\_y = 200 + int((max\_x1 + max\_x2 + max\_x3) / 3)  
  
  
def matrixGenerator():  
 # Генерує матрицю  
 matrix\_y = [[randrange(min\_y, max\_y)  
 for y in range(m)] for x in range(4)]  
 return matrix\_y  
  
  
class CritValues:  
 # Критичні значення  
 @staticmethod  
 def cohrenValue(selectionSize, selectionQty, significance):  
 # Значення критерію Кохрена  
 selectionSize += 1  
 partResult1 = significance / (selectionSize - 1)  
 params = [partResult1, selectionQty, (selectionSize - 1 - 1) \* selectionQty]  
 fisher = f.isf(\*params)  
 result = fisher / (fisher + (selectionSize - 1 - 1))  
 return Decimal(result).quantize(Decimal('.0001')).\_\_float\_\_()  
  
 @staticmethod  
 def studentValue(f3, significance):  
 # Значення критерію Стьюдента  
 return Decimal(abs(t.ppf(significance / 2, f3))).quantize(Decimal('.0001')).\_\_float\_\_()  
  
 @staticmethod  
 def fisherValue(f3, f4, significance):  
 # Значення критерію Фішера  
 return Decimal(abs(f.isf(significance, f4, f3))).quantize(Decimal('.0001')).\_\_float\_\_()  
  
  
def middle\_y(arr):  
 # Середнє значення у  
 middle = []  
 for k in range(len(arr)):  
 middle.append(sum(arr[k]) / len(arr[k]))  
 return middle  
  
  
def middle\_x(arr):  
 # Середнє значення х  
 middle = [0, 0, 0]  
 for k in range(4):  
 middle[0] += arr[k][0] / 4  
 middle[1] += arr[k][1] / 4  
 middle[2] += arr[k][2] / 4  
 return middle  
  
  
print("Рівняння регресії: у = b0 + b1\*X1 + b2\*X2 + b3\*X3")  
print("Матриця планування експеременту:")  
matrixExp = [[1, -1, -1, -1], [1, -1, 1, 1], [1, 1, -1, 1], [1, 1, 1, -1]]  
  
for i in range(len(matrixExp)):  
 for j in range(len(matrixExp[i])):  
 print("{:4d}".format(matrixExp[i][j]), end="")  
 print()  
  
tf = True  
# tf = true/false  
  
while tf:  
 matrix\_y = matrixGenerator()  
 matrix\_x = [[min\_x1, min\_x2, min\_x3], [min\_x1, max\_x2, max\_x3], [max\_x1, min\_x2, max\_x3], [max\_x1, max\_x2, min\_x3]]  
 a1, a2, a3, a11, a22, a33, a12, a13, a23 = 0, 0, 0, 0, 0, 0, 0, 0, 0  
 matrix = []  
  
 mean\_y = middle\_y(matrix\_y)  
 mean\_x = middle\_x(matrix\_x)  
  
 for i in range(4):  
 a1 += matrix\_x[i][0] \* mean\_y[i] / 4  
 a2 += matrix\_x[i][1] \* mean\_y[i] / 4  
 a3 += matrix\_x[i][2] \* mean\_y[i] / 4  
 a11 += matrix\_x[i][0] \*\* 2 / 4  
 a22 += matrix\_x[i][1] \*\* 2 / 4  
 a33 += matrix\_x[i][2] \*\* 2 / 4  
 a12 += matrix\_x[i][0] \* matrix\_x[i][1] / 4  
 a13 += matrix\_x[i][0] \* matrix\_x[i][2] / 4  
 a23 += matrix\_x[i][1] \* matrix\_x[i][2] / 4  
  
 a21 = a12  
 a31 = a13  
 a32 = a23  
  
 my = sum(mean\_y) / len(mean\_y)  
 numb0 = [[my, mean\_x[0], mean\_x[1], mean\_x[2]], [a1, a11, a12, a13], [a2, a21, a22, a23], [a3, a31, a32, a33]]  
 numb1 = [[1, my, mean\_x[1], mean\_x[2]], [mean\_x[0], a1, a12, a13], [mean\_x[1], a2, a22, a23],  
 [mean\_x[2], a3, a32, a33]]  
 numb2 = [[1, mean\_x[0], my, mean\_x[2]], [mean\_x[0], a11, a1, a13], [mean\_x[1], a21, a2, a23],  
 [mean\_x[2], a31, a3, a33]]  
 numb3 = [[1, mean\_x[0], mean\_x[1], my], [mean\_x[0], a11, a12, a1], [mean\_x[1], a21, a22, a2],  
 [mean\_x[2], a31, a32, a3]]  
 dividerB = [[1, mean\_x[0], mean\_x[1], mean\_x[2]], [mean\_x[0], a11, a12, a13],  
 [mean\_x[1], a21, a22, a23], [mean\_x[2], a31, a32, a33]]  
  
 b0 = det(numb0) / det(dividerB)  
 b1 = det(numb1) / det(dividerB)  
 b2 = det(numb2) / det(dividerB)  
 b3 = det(numb3) / det(dividerB)  
  
 f1 = m - 1  
 f2 = N  
 q = 1 - p  
 dispersion\_y = [0, 0, 0, 0]  
  
 for i in range(m):  
 dispersion\_y[0] += ((matrix\_y[0][i] - mean\_y[0]) \*\* 2) / 3  
 dispersion\_y[1] += ((matrix\_y[1][i] - mean\_y[1]) \*\* 2) / 3  
 dispersion\_y[2] += ((matrix\_y[2][i] - mean\_y[2]) \*\* 2) / 3  
 dispersion\_y[3] += ((matrix\_y[3][i] - mean\_y[3]) \*\* 2) / 3  
  
 Gp = max(dispersion\_y) / sum(dispersion\_y)  
 print("\nКритерій Кохрена")  
 Gt = CritValues.cohrenValue(f2, f1, q)  
  
 if Gt > Gp:  
 print("Дисперсія однорідна при рівні значимості {:.2f}!\nЗбільшувати m не потрібно.".format(q))  
 tf = False  
 else:  
 print("Дисперсія не однорідна при рівні значимості {:.2f}!".format(q))  
 m += 1  
  
for i in range(4):  
 matrix.append(matrix\_x[i] + matrix\_y[i])  
  
print("Матриця з натуральних значень факторів")  
print(" X1 X2 X3 Y1 Y2 Y3 ")  
for i in range(len(matrix)):  
 print("", end=" ")  
 for j in range(len(matrix[i])):  
 print(matrix[i][j], end=" ")  
 print("")  
  
print("Рівняння регресії")  
print("{:.3f} + {:.3f} \* X1 + {:.3f} \* X2 + {:.3f} \* X3 = ŷ".format(b0, b1, b2, b3))  
print("Перевірка")  
print("{:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} = ".format(b0, b1, min\_x1, b2, min\_x2, b3, min\_x3)  
 + str(b0 + b1 \* min\_x1 + b2 \* min\_x2 + b3 \* min\_x3))  
print("{:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} = ".format(b0, b1, min\_x1, b2, max\_x2, b3, max\_x3)  
 + str(b0 + b1 \* min\_x1 + b2 \* max\_x2 + b3 \* max\_x3))  
print("{:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} = ".format(b0, b1, max\_x1, b2, min\_x2, b3, max\_x3)  
 + str(b0 + b1 \* max\_x1 + b2 \* min\_x2 + b3 \* max\_x3))  
print("{:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} = ".format(b0, b1, max\_x1, b2, max\_x2, b3, min\_x3)  
 + str(b0 + b1 \* max\_x1 + b2 \* max\_x2 + b3 \* min\_x3))  
  
print("\nКритерій Стьюдента")  
f3 = f1 \* f2  
S\_2b = sum(dispersion\_y) / (N \* N \* m)  
S\_b = sqrt(S\_2b)  
beta\_0 = (mean\_y[0] + mean\_y[1] + mean\_y[2] + mean\_y[3]) / N  
beta\_1 = (-mean\_y[0] - mean\_y[1] + mean\_y[2] + mean\_y[3]) / N  
beta\_2 = (-mean\_y[0] + mean\_y[1] - mean\_y[2] + mean\_y[3]) / N  
beta\_3 = (-mean\_y[0] + mean\_y[1] + mean\_y[2] - mean\_y[3]) / N  
t\_0 = fabs(beta\_0) / S\_b  
t\_1 = fabs(beta\_1) / S\_b  
t\_2 = fabs(beta\_2) / S\_b  
t\_3 = fabs(beta\_3) / S\_b  
  
Tt = CritValues.studentValue(f1 \* f2, q)  
arr\_t = [t\_0, t\_1, t\_2, t\_3]  
arr\_b = [b0, b1, b2, b3]  
  
for i in range(4):  
 if arr\_t[i] > Tt:  
 continue  
 else:  
 arr\_t[i] = 0  
  
for j in range(4):  
 if arr\_t[j] != 0:  
 continue  
 else:  
 arr\_b[j] = 0  
  
print("Перевірка значемих коефіціентів:")  
yj1 = arr\_b[0] + arr\_b[1] \* min\_x1 + arr\_b[2] \* min\_x2 + arr\_b[3] \* min\_x3  
yj2 = arr\_b[0] + arr\_b[1] \* min\_x1 + arr\_b[2] \* max\_x2 + arr\_b[3] \* max\_x3  
yj3 = arr\_b[0] + arr\_b[1] \* max\_x1 + arr\_b[2] \* min\_x2 + arr\_b[3] \* max\_x3  
yj4 = arr\_b[0] + arr\_b[1] \* max\_x1 + arr\_b[2] \* max\_x2 + arr\_b[3] \* min\_x3  
  
print("{:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} = "  
 "".format(arr\_b[0], arr\_b[1], min\_x1, arr\_b[2], min\_x2, arr\_b[3], min\_x3) + str(yj1))  
print("{:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} = "  
 "".format(arr\_b[0], arr\_b[1], min\_x1, arr\_b[2], max\_x2, arr\_b[3], max\_x3) + str(yj2))  
print("{:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} = "  
 "".format(arr\_b[0], arr\_b[1], max\_x1, arr\_b[2], min\_x2, arr\_b[3], max\_x3) + str(yj3))  
print("{:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} + {:.3f} \* {:.3f} = "  
 "".format(arr\_b[0], arr\_b[1], max\_x1, arr\_b[2], max\_x2, arr\_b[3], min\_x3) + str(yj4))  
  
print("\nКритерій Фішера:")  
for i in range(3):  
 if arr\_b[i] == 0:  
 del arr\_b[i]  
  
d = len(arr\_b)  
f4 = N - d  
S\_2ad = m \* ((yj1 - mean\_y[0]) \*\* 2 + (yj2 - mean\_y[1]) \*\* 2 + (yj3 - mean\_y[2]) \*\* 2 + (  
 yj4 - mean\_y[3]) \*\* 2) / f4  
Fp = S\_2ad / S\_2b  
Ft = CritValues.fisherValue(f1 \* f2, f4, q)  
  
if Fp > Ft:  
 print("Рівняння регресії неадекватно оригіналу при рівні значимості {:.2f}".format(q))  
else:  
 print("Рівняння регресії адекватно оригіналу при рівні значимості {:.2f}".format(q))

**Результат:**



**Контрольні запитання**

1. Що називається дробовим факторним експериментом?

У деяких випадках немає необхідності проводити повний факторний експеримент (ПФЕ). Якщо буде використовуватися лінійна регресія, то можливо зменшити кількість рядків матриці ПФЕ до кількості коефіцієнтів регресійної моделі. Кількість дослідів слід скоротити, використовуючи для планування так звані регулярні дробові репліки від повного факторного експерименту, що містять відповідну кількість дослідів і зберігають основні властивості матриці планування – це означає дробовий факторний експеримент (ДФЕ).

1. Для чого потрібно розрахункове значення Кохрена?

Для перевірки дисперсії на однорідність.

1. Для чого перевіряється критерій Стьюдента?

Для перевірки значущості коефіцієнтів регресії. Тобто, Якщо виконується нерівність *ts*< *tтабл*, то приймається нуль-гіпотеза, тобто вважається, що знайдений коефіцієнт *βs* є статистично незначущим і його слід виключити з рівняння регресії. Якщо *ts* > *tтабл* то гіпотеза не підтверджується, тобто *βs* – значимий коефіцієнт і він залишається в рівнянні регресії.

1. Чим визначається критерій Фішера і як його застосовувати?

Отримане рівняння регресії необхідно перевірити на адекватність досліджуваному об'єкту. Для цієї мети необхідно оцінити, наскільки відрізняються середні значення у вихідної величини, отриманої в точках факторного простору, і значення у, отриманого з рівняння регресії в тих самих точках факторного простору. Для цього використовують дисперсію адекватності. Адекватність моделі перевіряють за F-критерієм Фішера, який дорівнює відношенню дисперсії адекватності до дисперсії відтворюваності.

**Висновок:** Отже, у ході виконання лабораторної роботи ми провели дробовий трьохфакторний експеримент. Склали матрицю планування, знайшли

коефіцієнти рівняння регресії, провели 3 статистичні перевірки. Була написана текстова програма. Результати співпадають із калькулятором.