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 №3**

З дисципліни «Методи наукових досліджень»

ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ

ВИКОНАВ:

Студент ІІ курсу ФІОТ

Групи ІВ-91

Липчак Дмитро Олександрович

Номер заліковки: 9118

Номер у списку: 17

ПЕРЕВІРИВ:

ас. Регіда П. Г.

Київ 2021 р.

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

коефіцієнти рівняння регресії, провести 3 статистичні перевірки.



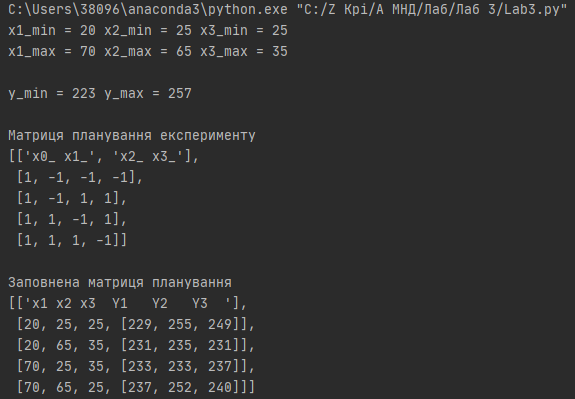


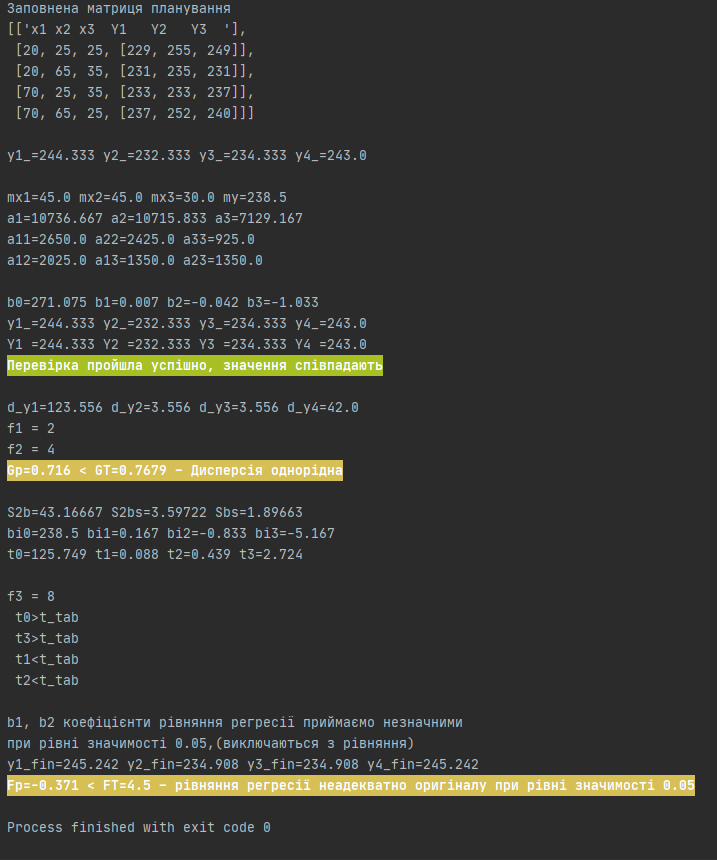
*y* = b0 + b1 x1 + b2 x2+ b3 x3

**Роздруківка коду програми**

import random as rand  
from numpy import linalg as lg  
import math as mt  
from pprint import pprint  
  
x1\_min = 20  
x1\_max = 70  
x2\_min = 25  
x2\_max = 65  
x3\_min = 25  
x3\_max = 35  
m = 3  
N = 4  
print(**"x1\_min = "**+str(x1\_min), **"x2\_min = "**+str(x2\_min), **"x3\_min = "**+str(x3\_min))  
print(**"x1\_max = "**+str(x1\_max), **"x2\_max = "**+str(x2\_max), **"x3\_max = "**+str(x3\_max))  
print(**""**)  
  
y\_min = 200 + round((x1\_min + x2\_min + x3\_min)/3)  
y\_max = 200 + round((x1\_max + x2\_max + x3\_max)/3)  
print(**"y\_min = "**+str(y\_min), **"y\_max = "**+str(y\_max))  
print(**""**)  
mp = [[**"x0\_ "** + **"x1\_"** , **"x2\_ "** + **"x3\_"** ],  
 [1, -1, -1, -1 ],  
 [1, -1, +1, +1 ],  
 [1, +1, -1, +1 ],  
 [1, +1, +1, -1 ]]  
print(**"Матриця планування експерименту"**)  
pprint(mp)  
print(**""**)  
  
e1 = [rand.randint(y\_min, y\_max) for i in range(m)]  
e2 = [rand.randint(y\_min, y\_max) for i in range(m)]  
e3 = [rand.randint(y\_min, y\_max) for i in range(m)]  
e4 = [rand.randint(y\_min, y\_max) for i in range(m)]  
mt2 = [[**"x1 "**+ **"x2 "**+ **"x3 "**+**" Y1 "**+**"Y2 "**+**"Y3 "**],  
 [x1\_min, x2\_min, x3\_min, e1],  
 [x1\_min,x2\_max,x3\_max,e2],  
 [x1\_max,x2\_min,x3\_max,e3],  
 [x1\_max,x2\_max,x3\_min,e4]]  
print(**"Заповнена матриця планування"**)  
pprint(mt2)  
print(**""**)  
  
y1\_ = sum(e1) / len(e1)  
y2\_ = sum(e2) / len(e2)  
y3\_ = sum(e3) / len(e3)  
y4\_ = sum(e4) / len(e4)  
print(**"y1\_="**+str(round(y1\_, 3))+**" y2\_="**+str(round(y2\_, 3))+**" y3\_="**+str(round(y3\_, 3))+**" y4\_="**+str(round(y4\_, 3)))  
print(**""**)  
mx1 = (mt2[1][0] + mt2[2][0] + mt2[3][0] + mt2[4][0])/4  
mx2 = (mt2[1][1] + mt2[2][1] + mt2[3][1] + mt2[4][1])/4  
mx3 = (mt2[1][2] + mt2[2][2] + mt2[3][2] + mt2[4][2])/4  
my = (y1\_ + y2\_ + y3\_ + y4\_)/4  
print(**"mx1="**+str(round(mx1, 3))+**" mx2="**+str(round(mx2, 3))+**" mx3="**+str(round(mx3, 3))+**" my="**+str(round(my,3)))  
  
a1 = ((mt2[1][0]\*y1\_ + mt2[2][0]\*y2\_ + mt2[3][0]\*y3\_ + mt2[4][0]\*y4\_)/4)  
a2 = ((mt2[1][1]\*y1\_ + mt2[2][1]\*y2\_ + mt2[3][1]\*y3\_ + mt2[4][1]\*y4\_)/4)  
a3 = ((mt2[1][2]\*y1\_ + mt2[2][2]\*y2\_ + mt2[3][2]\*y3\_ + mt2[4][2]\*y4\_)/4)  
print(**"a1="**+str(round(a1,3)) +**" a2="**+str(round(a2,3)) +**" a3="**+str(round(a3,3)))  
  
a11 = (mt2[1][0]\*mt2[1][0] + mt2[2][0]\*mt2[2][0] + mt2[3][0]\*mt2[3][0] + mt2[4][0]\*mt2[4][0])/4  
a22 = (mt2[1][1]\*mt2[1][1] + mt2[2][1]\*mt2[2][1] + mt2[3][1]\*mt2[3][1] + mt2[4][1]\*mt2[4][1])/4  
a33 = (mt2[1][2]\*mt2[1][2] + mt2[2][2]\*mt2[2][2] + mt2[3][2]\*mt2[3][2] + mt2[4][2]\*mt2[4][2])/4  
print(**"a11="**+str(round(a11,3)) +**" a22="**+str(round(a22,3)) +**" a33="**+str(round(a33,3)))  
a12 = (mt2[1][0]\*mt2[1][1] + mt2[2][0]\*mt2[2][1] + mt2[3][0]\*mt2[3][1] + mt2[4][0]\*mt2[4][1])/4  
a13 = (mt2[1][0]\*mt2[1][2] + mt2[2][0]\*mt2[2][2] + mt2[3][0]\*mt2[3][2] + mt2[4][0]\*mt2[4][2])/4  
a23 = (mt2[1][1]\*mt2[1][2] + mt2[2][1]\*mt2[2][2] + mt2[3][1]\*mt2[3][2] + mt2[4][1]\*mt2[4][2])/4  
print(**"a12="**+str(round(a12,3)) +**" a13="**+str(round(a13,3)) +**" a23="**+str(round(a23,3)))  
print(**""**)  
b0 = (lg.det([[my, mx1, mx2, mx3],  
 [a1, a11, a12, a13],  
 [a2, a12, a22, a23],  
 [a3, a13, a23, a33]]))/(lg.det([[1, mx1, mx2, mx3],  
 [mx1, a11, a12, a13],  
 [mx2, a12, a22, a23],  
 [mx3, a13, a23, a33]]))  
b1 = (lg.det([[1, my, mx2, mx3],  
 [mx1, a1, a12, a13],  
 [mx2, a2, a22, a23],  
 [mx3, a3, a23, a33]]))/(lg.det([[1, mx1, mx2, mx3],  
 [mx1, a11, a12, a13],  
 [mx2, a12, a22, a23],  
 [mx3, a13, a23, a33]]))  
b2 = (lg.det([[1, mx1, my, mx3],  
 [mx1, a11, a1, a13],  
 [mx2, a12, a2, a23],  
 [mx3, a13, a3, a33]]))/(lg.det([[1, mx1, mx2, mx3],  
 [mx1, a11, a12, a13],  
 [mx2, a12, a22, a23],  
 [mx3, a13, a23, a33]]))  
b3 = (lg.det([[1, mx1, mx2, my],  
 [mx1, a11, a12, a1],  
 [mx2, a12, a22, a2],  
 [mx3, a13, a23, a3]]))/(lg.det([[1, mx1, mx2, mx3],  
 [mx1, a11, a12, a13],  
 [mx2, a12, a22, a23],  
 [mx3, a13, a23, a33]]))  
print(**"b0="**+str(round(b0,3)) +**" b1="**+str(round(b1,3)) +**" b2="**+str(round(b2,3)) + **" b3="**+str(round(b3,3)))  
Y1 = b0 + b1 \* mt2[1][0] + b2 \* mt2[1][1] + b3 \* mt2[1][2]  
Y2 = b0 + b1 \* mt2[2][0] + b2 \* mt2[2][1] + b3 \* mt2[2][2]  
Y3 = b0 + b1 \* mt2[3][0] + b2 \* mt2[3][1] + b3 \* mt2[3][2]  
Y4 = b0 + b1 \* mt2[4][0] + b2 \* mt2[4][1] + b3 \* mt2[4][2]  
print(**"y1\_="**+str(round(y1\_, 3))+**" y2\_="**+str(round(y2\_, 3))+**" y3\_="**+str(round(y3\_, 3))+**" y4\_="**+str(round(y4\_, 3)))  
print(**"Y1 ="**+str(round(y1\_, 3))+**" Y2 ="**+str(round(Y2, 3))+**" Y3 ="**+str(round(Y3, 3))+**" Y4 ="**+str(round(Y4, 3)))  
if round(y1\_,3)==round(Y1,3) and round(y2\_,3)==round(Y2,3) and round(y3\_,3)==round(Y3,3) and round(y4\_,3)==round(Y4,3):  
 print(**"**\033**[1m**\033**[30m**\033**[42m{}**\033**[0m"** .format(**"Перевірка пройшла успішно, значення співпадають"**))  
print(**""**)  
d\_y1 = (((e1[0] - y1\_)\*\*2)+((e1[1] - y1\_)\*\*2)+((e1[2] - y1\_)\*\*2))/3  
d\_y2 = (((e2[0] - y2\_)\*\*2)+((e2[1] - y2\_)\*\*2)+((e2[2] - y2\_)\*\*2))/3  
d\_y3 = (((e3[0] - y3\_)\*\*2)+((e3[1] - y3\_)\*\*2)+((e3[2] - y3\_)\*\*2))/3  
d\_y4 = (((e4[0] - y4\_)\*\*2)+((e4[1] - y4\_)\*\*2)+((e4[2] - y4\_)\*\*2))/3  
print(**"d\_y1="**+str(round(d\_y1, 3))+**" d\_y2="**+str(round(d\_y2, 3))+**" d\_y3="**+str(round(d\_y3, 3))+**" d\_y4="**+str(round(d\_y4,3)))  
Gp = max(d\_y1, d\_y2, d\_y3, d\_y4)/(d\_y1 + d\_y2 + d\_y3 + d\_y4)  
f1 = m-1  
f2 = N  
print(**"f1 = "**+str(f1))  
print(**"f2 = "**+str(f2))  
GT = 0.7679  
if Gp < GT:  
 print(**"**\033**[1m**\033**[30m**\033**[43mGp={} < GT={} – Дисперсія однорідна**\033**[0m"**.format(round(Gp,3), GT))  
print(**""**)  
S2b = (d\_y1 + d\_y2 + d\_y3 + d\_y4)/N  
S2bs = (S2b)/(N\*m)  
Sbs = mt.sqrt(S2bs)  
print(**"S2b="**+str(round(S2b,5))+ **" S2bs="**+str(round(S2bs,5))+ **" Sbs="**+str(round(Sbs,5)))  
bi0 = (y1\_\*mp[1][0] + y2\_\*mp[2][0] + y3\_\*mp[3][0] + y4\_\*mp[4][0])/4  
bi1 = (y1\_\*mp[1][1] + y2\_\*mp[2][1] + y3\_\*mp[3][1] + y4\_\*mp[4][1])/4  
bi2 = (y1\_\*mp[1][2] + y2\_\*mp[2][2] + y3\_\*mp[3][2] + y4\_\*mp[4][2])/4  
bi3 = (y1\_\*mp[1][3] + y2\_\*mp[2][3] + y3\_\*mp[3][3] + y4\_\*mp[4][3])/4  
print(**"bi0="**+str(round(bi0, 3))+**" bi1="**+str(round(bi1, 3))+**" bi2="**+str(round(bi2, 3))+**" bi3="**+str(round(bi3,3)))  
t0 = abs(bi0)/Sbs  
t1 = abs(bi1)/Sbs  
t2 = abs(bi2)/Sbs  
t3 = abs(bi3)/Sbs  
print(**"t0="**+str(round(t0, 3))+**" t1="**+str(round(t1, 3))+**" t2="**+str(round(t2, 3))+**" t3="**+str(round(t3,3)))  
print(**""**)  
f3 = f1\*f2  
print(**"f3 = "**+str(f3))  
t\_tab = 2.306  
if t0>t\_tab:  
 print(**" t0>t\_tab"**)  
if t3>t\_tab:  
 print(**" t3>t\_tab"**)  
if t1<t\_tab:  
 print(**" t1<t\_tab"**)  
if t2<t\_tab:  
 print(**" t2<t\_tab"**)  
print(**""**)  
print(**"b1, b2 коефіцієнти рівняння регресії приймаємо незначними** \n**при рівні значимості 0.05,(виключаються з рівняння)"**)  
y1\_fin = b0 + b3\*mt2[1][2]  
y2\_fin = b0 + b3\*mt2[2][2]  
y3\_fin = b0 + b3\*mt2[3][2]  
y4\_fin = b0 + b3\*mt2[4][2]  
print(**"y1\_fin="**+str(round(y1\_fin, 3))+**" y2\_fin="**+str(round(y2\_fin, 3))+**" y3\_fin="**+str(round(y3\_fin, 3))+**" y4\_fin="**+str(round(y4\_fin,3)))  
d = 2  
f4 = N-d  
S2ad = (m/N-d)\*((y1\_fin - y1\_)\*\*2 + (y2\_fin - y2\_)\*\*2 + (y3\_fin - y3\_)\*\*2 + (y4\_fin - y4\_)\*\*2)  
Fp = S2ad/S2b  
FT = 4.5  
if Fp<FT:  
 print(**"**\033**[1m**\033**[30m**\033**[43mFp={} < FT={} – рівняння регресії неадекватно оригіналу при рівні значимості 0.05**\033**[0m"**.format(round(Fp,3), FT))

**Результат роботи програми**





**Відповіді на контрольні питання:**

**1. З чого складається план експерименту?**

Сукупність усіх точок плану - векторів Xi (для i = 1, 2, . . . , N) утворює план експерименту. Таким чином, план експерименту описується матрицею, яка містить N рядків і K стовбців. Кожен рядок матриці означає точку плану експерименту, а стовпчик – фактор експерименту.

**2. Що називається спектром плану?**

Сукупність усіх точок плану, що відрізняються рівнем хоча б одного фактора (різних строк матриці планування), називається спектром плану. Матриця, отримана із усіх різних строк плану називається матрицею спектра плану.

**3.Чим відрізняються активні та пасивні експерименти?**  Експерименти поділяють на пасивні та активні (керовані). В пасивному експерименті існують контрольовані, але некеровані вхідні параметри – ми не маємо можливості втручатись в хід проведення експерименту, і виступаємо в ролі пасивного користувача. В активному – існують керовані і контрольовані вхідні параметри – ми самі являємось адміністраторами нашої системи.

**4.Чим характеризується об’єкт досліджень? Дайте визначення факторному простору.**

Об’єкт досліджень розглядається як «чорний ящик». Аналізуються деякі властивості та якості, які можуть описуватися числовими значеннями. Вектор Х1…ХK представляє собою групу контрольованих та керованих величин, котрі можуть змінюватись необхідним чином при проведенні експерименту, Цю групу характеристик Х1…ХK також називають факторами або керованими впливами. Факторний простір - простір незалежних змінних(факторів), діапазон значень факторів.

**Висновок**

Виконуючи дану лабораторну роботу, я провів трьохфакторний есперимент. Склав матрицю планування та знайшов коефіцієнти рівняння регресії, провів статистичні перевірки. Результати роботи програми наведені вище. Під час виконання роботи проблем не виникло.