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 № 6

з дисципліни «Методи планування експерименту»

# на тему «Проведення трьохфакторного експерименту при використанні рівняння регресії з квадратичними членами»

ВИКОНАЛА:

студентка 2 курсу

групи ІВ-93

Трибунська Кароліна Євгенівна

Залікова - 9331

ПЕРЕВІРИВ: ас. Регіда П.Г.

Київ - 2021

**Хід роботи**

**Мета**:

Провести трьохфакторний експеримент і отримати адекватну модель – рівняння регресії, використовуючи рототабельний композиційний план.

**Завдання до лабораторної роботи:**

1. Ознайомитися з теоретичними відомостями.

2. Вибрати з таблиці варіантів і записати в протокол інтервали значень х1, х2, х3. Обчислити і записати значення, відповідні кодованим значенням

факторів +1; -1;+ ; - ; 0 для 1, 2, 3.

3. Значення функції відгуку знайти за допомогою підстановки в формулу:

yi = f(х1, х2, х3) + random(10)-5,

де f(х1, х2, х3) вибирається по номеру в списку в журналі викладача.

4. Провести експерименти і аналізуючи значення статистичних перевірок, отримати адекватну модель рівняння регресії. При розрахунках

використовувати натуральні значення факторів.

5. Зробити висновки по виконаній роботі.

# Варіант 326

****

# Лістинг

from math import fabs  
from random import randrange  
import numpy as np  
from numpy.linalg import solve  
from scipy.stats import f, t  
  
m = 3  
n = 15  
  
# варіант 326  
x1min = -25  
x1max = -5  
x2min = -15  
x2max = 35  
x3min = -5  
x3max = 60  
  
def function(X1, X2, X3):  
 y = 8.4 + 0.4 \* X1 + 1.5 \* X2 + 9.3 \* X3 + 9.5 \* X1 \* X1 + 0.3 \* X2 \* X2 + 2.2 \* X3 \* X3 + 1.1 \* X1 \* X2 + \  
 0.7 \* X1 \* X3 + 9.3 \* X2 \* X3 + 3.7 \* X1 \* X2 \* X3 + randrange(0, 10) - 5  
 return y  
  
x01 = (x1max + x1min) / 2  
x02 = (x2max + x2min) / 2  
x03 = (x3max + x3min) / 2  
deltax1 = x1max - x01  
deltax2 = x2max - x02  
deltax3 = x3max - x03  
# матриця ПФЕ  
xn = [[-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, -1, -1, -1, +1, +1, +1],  
 [+1, +1, +1, +1, +1, +1, +1, +1, +1, +1],  
 [-1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],  
 [+1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],  
 [0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0],  
 [0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0],  
 [0, 0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929],  
 [0, 0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]  
  
x1 = [x1min, x1min, x1min, x1min, x1max, x1max, x1max, x1max, -1.73 \* deltax1 + x01, 1.73 \* deltax1 + x01, x01, x01,  
 x01, x01, x01]  
x2 = [x2min, x2min, x2max, x2max, x2min, x2min, x2max, x2max, x02, x02, -1.73 \* deltax2 + x02, 1.73 \* deltax2 + x02,  
 x02, x02, x02]  
x3 = [x3min, x3max, x3min, x3max, x3min, x3max, x3min, x3max, x03, x03, x03, x03, -1.73 \* deltax3 + x03,  
 1.73 \* deltax3 + x03, x03]  
# заповнення нулями х1х2, х1х3, х1х2х3  
# заповнення нулями х1kv, х2kv, х3kv  
x1x2, x1x3, x2x3, x1x2x3 = [0] \* n, [0] \* n, [0] \* n, [0] \* n  
x1kv, x2kv, x3kv = [0] \* n, [0] \* n, [0] \* n  
for i in range(15):  
 x1x2[i] = x1[i] \* x2[i]  
 x1x3[i] = x1[i] \* x3[i]  
 x2x3[i] = x2[i] \* x3[i]  
 x1x2x3[i] = x1[i] \* x2[i] \* x3[i]  
 x1kv[i] = x1[i] \*\* 2  
 x2kv[i] = x2[i] \*\* 2  
 x3kv[i] = x3[i] \*\* 2  
# формуємо список a  
list\_for\_a = list(zip(x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1kv, x2kv, x3kv))  
  
print("Матриця планування з натуралізованими коефіцієнтами X:")  
print(" X1 X2 X3 X1X2 X1X3 X2X3 X1X2X3 X1X1"  
 " X2X2 X3X3")  
for i in range(n):  
 print(end=' ')  
 for j in range(len(list\_for\_a[0])):  
 print("{:^12.3f}".format(list\_for\_a[i][j]), end=' ')  
 print("")  
# вивід матриці планування  
Y = [[function(list\_for\_a[j][0], list\_for\_a[j][1], list\_for\_a[j][2]) for i in range(m)] for j in range(15)]  
print("Матриця планування Y:")  
print(" Y1 Y2 Y3")  
for i in range(n):  
 print(end=' ')  
 for j in range(len(Y[0])):  
 print("{:^12.3f}".format(Y[i][j]), end=' ')  
 print("")  
# середні у  
Y\_average = []  
for i in range(len(Y)):  
 Y\_average.append(np.mean(Y[i], axis=0))  
print("Середні значення відгуку за рядками:")  
for i in range(n):  
 print("{:.3f}".format(Y\_average[i]), end=" ")  
# розрахунок дисперсій  
dispersions = []  
for i in range(len(Y)):  
 a = 0  
 for k in Y[i]:  
 a += (k - np.mean(Y[i], axis=0)) \*\* 2  
 dispersions.append(a / len(Y[i]))  
  
  
def find\_known(num):  
 a = 0  
 for j in range(n):  
 a += Y\_average[j] \* list\_for\_a[j][num - 1] / n  
 return a  
  
  
def a(first, second):  
 a = 0  
 for j in range(n):  
 a += list\_for\_a[j][first - 1] \* list\_for\_a[j][second - 1] / n  
 return a  
  
  
my = sum(Y\_average) / n  
mx = []  
for i in range(10):  
 number\_lst = []  
 for j in range(n):  
 number\_lst.append(list\_for\_a[j][i])  
 mx.append(sum(number\_lst) / len(number\_lst))  
  
det1 = [  
 [1, mx[0], mx[1], mx[2], mx[3], mx[4], mx[5], mx[6], mx[7], mx[8], mx[9]],  
 [mx[0], a(1, 1), a(1, 2), a(1, 3), a(1, 4), a(1, 5), a(1, 6), a(1, 7), a(1, 8), a(1, 9), a(1, 10)],  
 [mx[1], a(2, 1), a(2, 2), a(2, 3), a(2, 4), a(2, 5), a(2, 6), a(2, 7), a(2, 8), a(2, 9), a(2, 10)],  
 [mx[2], a(3, 1), a(3, 2), a(3, 3), a(3, 4), a(3, 5), a(3, 6), a(3, 7), a(3, 8), a(3, 9), a(3, 10)],  
 [mx[3], a(4, 1), a(4, 2), a(4, 3), a(4, 4), a(4, 5), a(4, 6), a(4, 7), a(4, 8), a(4, 9), a(4, 10)],  
 [mx[4], a(5, 1), a(5, 2), a(5, 3), a(5, 4), a(5, 5), a(5, 6), a(5, 7), a(5, 8), a(5, 9), a(5, 10)],  
 [mx[5], a(6, 1), a(6, 2), a(6, 3), a(6, 4), a(6, 5), a(6, 6), a(6, 7), a(6, 8), a(6, 9), a(6, 10)],  
 [mx[6], a(7, 1), a(7, 2), a(7, 3), a(7, 4), a(7, 5), a(7, 6), a(7, 7), a(7, 8), a(7, 9), a(7, 10)],  
 [mx[7], a(8, 1), a(8, 2), a(8, 3), a(8, 4), a(8, 5), a(8, 6), a(8, 7), a(8, 8), a(8, 9), a(8, 10)],  
 [mx[8], a(9, 1), a(9, 2), a(9, 3), a(9, 4), a(9, 5), a(9, 6), a(9, 7), a(9, 8), a(9, 9), a(9, 10)],  
 [mx[9], a(10, 1), a(10, 2), a(10, 3), a(10, 4), a(10, 5), a(10, 6), a(10, 7), a(10, 8), a(10, 9), a(10, 10)]]  
  
det2 = [my, find\_known(1), find\_known(2), find\_known(3), find\_known(4), find\_known(5), find\_known(6), find\_known(7),  
 find\_known(8), find\_known(9), find\_known(10)]  
  
beta = solve(det1, det2)  
print("\nОтримане рівняння регресії:")  
print("{:.3f} + {:.3f} \* X1 + {:.3f} \* X2 + {:.3f} \* X3 + {:.3f} \* Х1X2 + {:.3f} \* Х1X3 + {:.3f} \* Х2X3"  
 "+ {:.3f} \* Х1Х2X3 + {:.3f} \* X11^2 + {:.3f} \* X22^2 + {:.3f} \* X33^2 = ŷ"  
 .format(beta[0], beta[1], beta[2], beta[3], beta[4], beta[5], beta[6], beta[7], beta[8], beta[9], beta[10]))  
y\_i = [0] \* n  
print("Експериментальні значення:")  
for k in range(n):  
 y\_i[k] = beta[0] + beta[1] \* list\_for\_a[k][0] + beta[2] \* list\_for\_a[k][1] + beta[3] \* list\_for\_a[k][2] + \  
 beta[4] \* list\_for\_a[k][3] + beta[5] \* list\_for\_a[k][4] + beta[6] \* list\_for\_a[k][5] + beta[7] \* \  
 list\_for\_a[k][6] + beta[8] \* list\_for\_a[k][7] + beta[9] \* list\_for\_a[k][8] + beta[10] \* list\_for\_a[k][9]  
for i in range(n):  
 print("{:.3f}".format(y\_i[i]), end=" ")  
print("\n------------------------------- Перевірка за критерієм Кохрена -------------------------------")  
Gp = max(dispersions) / sum(dispersions)  
Gt = 0.3346  
print("Gp =", Gp)  
if Gp < Gt:  
 print("Дисперсія однорідна")  
else:  
 print("Дисперсія неоднорідна")  
  
print("------------------ Перевірка значущості коефіцієнтів за критерієм Стьюдента ------------------")  
sb = sum(dispersions) / len(dispersions)  
sbs = (sb / (n \* m)) \*\* 0.5  
  
F3 = (m - 1) \* n  
coefs1 = []  
coefs2 = []  
d = 11  
res = [0] \* 11  
for j in range(11):  
 t\_pract = 0  
 for i in range(15):  
 if j == 0:  
 t\_pract += Y\_average[i] / 15  
 else:  
 t\_pract += Y\_average[i] \* xn[i][j - 1]  
 res[j] = beta[j]  
 if fabs(t\_pract / sbs) < t.ppf(q=0.975, df=F3):  
 coefs2.append(beta[j])  
 res[j] = 0  
 d-=1  
 else:  
 coefs1.append(beta[j])  
print("Значущі коефіцієнти регресії:", [round(i, 3) for i in coefs1])  
print("Незначущі коефіцієнти регресії:", [round(i, 3) for i in coefs2])  
y\_st = []  
for i in range(n):  
 y\_st.append(res[0] + res[1] \* x1[i] + res[2] \* x2[i] + res[3] \* x3[i] + res[4] \* x1x2[i] + res[5] \*  
 x1x3[i] + res[6] \* x2x3[i] + res[7] \* x1x2x3[i] + res[8] \* x1kv[i] + res[9] \*  
 x2kv[i] + res[10] \* x3kv[i])  
print("Значення з отриманими коефіцієнтами:")  
for i in range(n):  
 print("{:.3f}".format(y\_st[i]), end=" ")  
  
print("\n------------------------- Перевірка адекватності за критерієм Фішера -------------------------")  
Sad = m \* sum([(y\_st[i] - Y\_average[i]) \*\* 2 for i in range(n)]) / (n - d)  
Fp = Sad / sb  
F4 = n - d  
print("Fp =", Fp)  
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):  
 print("Рівняння регресії адекватне при рівні значимості 0.05")  
else:  
 print("Рівняння регресії неадекватне при рівні значимості 0.05")

# Результати роботи програми

C:\Users\Karolina\AppData\Local\Programs\Python\Python38-32\python.exe "C:/Users/Karolina/Desktop/университет/four\_semestr/методи наукових досліджень/lab6/lab6.py"

Матриця планування з натуралізованими коефіцієнтами X:

X1 X2 X3 X1X2 X1X3 X2X3 X1X2X3 X1X1 X2X2 X3X3

-25.000 -15.000 -5.000 375.000 125.000 75.000 -1875.000 625.000 225.000 25.000

-25.000 -15.000 60.000 375.000 -1500.000 -900.000 22500.000 625.000 225.000 3600.000

-25.000 35.000 -5.000 -875.000 125.000 -175.000 4375.000 625.000 1225.000 25.000

-25.000 35.000 60.000 -875.000 -1500.000 2100.000 -52500.000 625.000 1225.000 3600.000

-5.000 -15.000 -5.000 75.000 25.000 75.000 -375.000 25.000 225.000 25.000

-5.000 -15.000 60.000 75.000 -300.000 -900.000 4500.000 25.000 225.000 3600.000

-5.000 35.000 -5.000 -175.000 25.000 -175.000 875.000 25.000 1225.000 25.000

-5.000 35.000 60.000 -175.000 -300.000 2100.000 -10500.000 25.000 1225.000 3600.000

-32.300 10.000 27.500 -323.000 -888.250 275.000 -8882.500 1043.290 100.000 756.250

2.300 10.000 27.500 23.000 63.250 275.000 632.500 5.290 100.000 756.250

-15.000 -33.250 27.500 498.750 -412.500 -914.375 13715.625 225.000 1105.562 756.250

-15.000 53.250 27.500 -798.750 -412.500 1464.375 -21965.625 225.000 2835.562 756.250

-15.000 10.000 -28.725 -150.000 430.875 -287.250 4308.750 225.000 100.000 825.126

-15.000 10.000 83.725 -150.000 -1255.875 837.250 -12558.750 225.000 100.000 7009.876

-15.000 10.000 27.500 -150.000 -412.500 275.000 -4125.000 225.000 100.000 756.250

Матриця планування Y:

Y1 Y2 Y3

245.400 247.400 252.400

88702.400 88701.400 88702.400

20048.400 20052.400 20044.400

-161903.600 -161899.600 -161897.600

-291.600 -297.600 -295.600

16921.400 16917.400 16914.400

2102.400 2106.400 2104.400

-10576.600 -10580.600 -10577.600

-19417.590 -19418.590 -19414.590

6990.400 6988.400 6988.400

46841.194 46844.194 46842.194

-63828.556 -63834.556 -63829.556

17140.596 17143.596 17141.596

-21341.794 -21344.794 -21338.794

-9050.350 -9050.350 -9050.350

Середні значення відгуку за рядками:

248.400 88702.067 20048.400 -161900.267 -294.933 16917.733 2104.400 -10578.267 -19416.923 6989.067 46842.527 -63830.890 17141.930 -21341.794 -9050.350

Отримане рівняння регресії:

5.287 + -0.310 \* X1 + 1.558 \* X2 + 9.383 \* X3 + 1.100 \* Х1X2 + 0.701 \* Х1X3 + 9.302 \* Х2X3+ 3.700 \* Х1Х2X3 + 9.478 \* X11^2 + 0.297 \* X22^2 + 2.199 \* X33^2 = ŷ

Експериментальні значення:

248.426 88700.519 20049.143 -161901.097 -294.482 16916.611 2105.569 -10578.671 -19416.178 6988.828 46843.610 -63831.466 17140.364 -21339.721 -9050.354

------------------------------- Перевірка за критерієм Кохрена -------------------------------

Gp = 0.16271186440677968

Дисперсія однорідна

------------------ Перевірка значущості коефіцієнтів за критерієм Стьюдента ------------------

Значущі коефіцієнти регресії: [5.287, -0.31, 1.558, 9.383, 1.1, 0.701, 9.302, 3.7, 9.478, 0.297, 2.199]

Незначущі коефіцієнти регресії: []

Значення з отриманими коефіцієнтами:

248.426 88700.519 20049.143 -161901.097 -294.482 16916.611 2105.569 -10578.671 -19416.178 6988.828 46843.610 -63831.466 17140.364 -21339.721 -9050.354

------------------------- Перевірка адекватності за критерієм Фішера -------------------------

Fp = 2.659521310199179

Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0

**Висновок:**

В даній лабораторній роботі проведено трьохфакторний експеримент та отримано адекватну модель – рівняння регресії, використовуючи ротатабельний композиційний план. Отримані значення статистичних перевірок.