Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №3**

*з дисципліни «Методи оптимізації та планування експерименту»*

# *на тему:* «ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ.»

**Виконала:**

студентка 2-го курсу ФІОТ

групи ІВ-81

**Дрозд С. В.**

Номер у списку: 11

Варiант: **111**

Перевірив:  
Регіда П.Г.

Київ – 2020

**Варіант завдання:**



**Код:**

import numpy as np

import scipy.stats

x1\_min, x1\_max = 10, 60

x2\_min, x2\_max = -70, -10

x3\_min, x3\_max = 60, 70

mx\_max = (x1\_max + x2\_max + x3\_max) / 3

mx\_min = (x1\_min + x2\_min + x3\_min) / 3

y\_max = mx\_max + 200

y\_min = mx\_min + 200

# Створити 4x3 random [y\_min; y\_max]

y\_list = np.random.randint(y\_min, y\_max, (4, 3))

x\_matrix = [

[x1\_min, x2\_min, x3\_min],

[x1\_min, x2\_max, x3\_max],

[x1\_max, x2\_min, x3\_max],

[x1\_max, x2\_max, x3\_min]

]

my\_list = []

mx1, mx2, mx3 = 0, 0, 0

for obj in y\_list:

my\_list.append((obj[0]+obj[1]+obj[2])/3)

for obj in x\_matrix:

mx1 += obj[0]

mx2 += obj[1]

mx3 += obj[2]

my1 = my\_list[0]

my2 = my\_list[1]

my3 = my\_list[2]

my4 = my\_list[3]

mx1 /= 4

mx2 /= 4

mx3 /= 4

my = (my1 + my2 + my3 + my4) / 4

# Коефіцієнти

a1 = (

x\_matrix[0][0] \* my1 +

x\_matrix[1][0] \* my2 +

x\_matrix[2][0] \* my3 +

x\_matrix[3][0] \* my4

) / 4

a2 = (

x\_matrix[0][1] \* my1 +

x\_matrix[1][1] \* my2 +

x\_matrix[2][1] \* my3 +

x\_matrix[3][1] \* my4

) / 4

a3 = (

x\_matrix[0][2] \* my1 +

x\_matrix[1][2] \* my2 +

x\_matrix[2][2] \* my3 +

x\_matrix[3][2] \* my4

) / 4

a11 = (

x\_matrix[0][0]\*\*2 +

x\_matrix[1][0]\*\*2 +

x\_matrix[2][0]\*\*2 +

x\_matrix[3][0]\*\*2

) / 4

a22 = (

x\_matrix[0][1]\*\*2 +

x\_matrix[1][1]\*\*2 +

x\_matrix[2][1]\*\*2 +

x\_matrix[3][1]\*\*2

) / 4

a33 = (

x\_matrix[0][2]\*\*2 +

x\_matrix[1][2]\*\*2 +

x\_matrix[2][2]\*\*2 +

x\_matrix[3][2]\*\*2

) / 4

a12 = a21 = (

x\_matrix[0][0] \* x\_matrix[0][1] + x\_matrix[1][0] \* x\_matrix[1][1] +

x\_matrix[2][0] \* x\_matrix[2][1] + x\_matrix[3][0] \* x\_matrix[3][1]

) / 4

a13 = a31 = (

x\_matrix[0][0] \* x\_matrix[0][2] + x\_matrix[1][0] \* x\_matrix[1][2] +

x\_matrix[2][0] \* x\_matrix[2][2] + x\_matrix[3][0] \* x\_matrix[3][2]

) / 4

a23 = a32 = (

x\_matrix[0][1] \* x\_matrix[0][2] + x\_matrix[1][1] \* x\_matrix[1][2] +

x\_matrix[2][1] \* x\_matrix[2][2] + x\_matrix[3][1] \* x\_matrix[3][2]

) / 4

denominator = np.linalg.det([

[1, mx1, mx2, mx3],

[mx1, a11, a12, a13],

[mx2, a12, a22, a32],

[mx3, a13, a23, a33]

])

numerator\_b0 = np.linalg.det([

[my, mx1, mx2, mx3],

[a1, a11, a12, a13],

[a2, a12, a22, a32],

[a3, a13, a23, a33]

])

numerator\_b1 = np.linalg.det([

[1, my, mx2, mx3],

[mx1, a1, a12, a13],

[mx2, a2, a22, a32],

[mx3, a3, a23, a33]

])

numerator\_b2 = np.linalg.det([

[1, mx1, my, mx3],

[mx1, a11, a1, a13],

[mx2, a12, a2, a32],

[mx3, a13, a3, a33]

])

numerator\_b3 = np.linalg.det([

[1, mx1, mx2, my],

[mx1, a11, a12, a1],

[mx2, a12, a22, a2],

[mx3, a13, a23, a3]

])

b0 = numerator\_b0 / denominator

b1 = numerator\_b1 / denominator

b2 = numerator\_b2 / denominator

b3 = numerator\_b3 / denominator

print("-"\*15 + "Рівняння регресії" + "-"\*15)

print(f"b0 = {b0:.3f}; b1 = {b1:.3f}; b2 = {b2:.3f}; b3 = {b3:.3f}")

print(f"Рівняння регресії: y = {b0:+.3f} {b1:+.3f}\*x1 {b2:+.3f}\*x2 {b3:+.3f}\*x3\n")

if (b0 + b1\*x\_matrix[0][0] + b2\*x\_matrix[0][1] + b3\*x\_matrix[0][2]) == my1:

print("b0 + b1\*X11 + b2\*X12 + b3\*X13 = my1")

print("b0 + b1\*X11 + b2\*X12 + b3\*X13 = " +

f"{b0 + b1\*x\_matrix[0][0] + b2\*x\_matrix[0][1] + b3\*x\_matrix[0][2]:.3f}" +

f"||| my1 = {my1:.3f}"

)

print("b0 + b1\*X21 + b2\*X22 + b3\*X23 = " +

f"{b0 + b1\*x\_matrix[1][0] + b2\*x\_matrix[1][1] + b3\*x\_matrix[1][2]:.3f}" +

f"||| my2 = {my2:.3f}"

)

print("b0 + b1\*X31 + b2\*X32 + b3\*X33 = " +

f"{b0 + b1\*x\_matrix[2][0] + b2\*x\_matrix[2][1] + b3\*x\_matrix[2][2]:.3f}" +

f"||| my3 = {my3:.3f}"

)

print("b0 + b1\*X41 + b2\*X42 + b3\*X43 = " +

f"{b0 + b1\*x\_matrix[3][0] + b2\*x\_matrix[3][1] + b3\*x\_matrix[3][2]:.3f}" +

f"||| my4 = {my4:.3f}"

)

x\_matrix\_normal = [

[1, -1, -1, -1],

[1, -1, 1, 1],

[1, 1, -1, 1],

[1, 1, 1, -1]

]

# Знайти дисперсію

S2 = []

for i in range(len(y\_list)):

S2.append(

(

(y\_list[i][0] - my\_list[i])\*\*2 +

(y\_list[i][1] - my\_list[i])\*\*2 +

(y\_list[i][2] - my\_list[i])\*\*2

) / 3

)

S2y1 = S2[0]

S2y2 = S2[1]

S2y3 = S2[2]

S2y4 = S2[3]

print("\n"+"-"\*15 + "Критерій Кохрена" + "-"\*16)

Gp = max(S2) / sum(S2)

m = len(y\_list[0])

print (f"m: {m}")

f1 = m - 1

f2 = N = len(x\_matrix)

print (f"n: {N}")

q = 0.05

# для q = 0.05, f1 = 2, f2 = 4, Gt = 0.7679

Gt = 0.7679

if Gp < Gt:

print("Дисперсія однорідна")

print("\n"+"-"\*15 + "Критерій Ст'юдента" + "-"\*14)

S2B = sum(S2) / N

S2beta = S2B / (N \* m)

Sbeta = np.sqrt(S2beta)

beta0 = (

my1 \* x\_matrix\_normal[0][0] +

my2 \* x\_matrix\_normal[1][0] +

my3 \* x\_matrix\_normal[2][0] +

my4 \* x\_matrix\_normal[3][0]

) / 4

beta1 = (

my1 \* x\_matrix\_normal[0][1] +

my2 \* x\_matrix\_normal[1][1] +

my3 \* x\_matrix\_normal[2][1] +

my4 \* x\_matrix\_normal[3][1]

) / 4

beta2 = (

my1 \* x\_matrix\_normal[0][2] +

my2 \* x\_matrix\_normal[1][2] +

my3 \* x\_matrix\_normal[2][2] +

my4 \* x\_matrix\_normal[3][2]

) / 4

beta3 = (

my1 \* x\_matrix\_normal[0][3] +

my2 \* x\_matrix\_normal[1][3] +

my3 \* x\_matrix\_normal[2][3] +

my4 \* x\_matrix\_normal[3][3]

) / 4

t0 = abs(beta0) / Sbeta

t1 = abs(beta1) / Sbeta

t2 = abs(beta2) / Sbeta

t3 = abs(beta3) / Sbeta

f3 = f1 \* f2

# t\_tab = 2.306 # для значення f3 = 8, t табличне = 2,306

# print("T\_tab:", t\_tab)

t\_tab = scipy.stats.t.ppf((1 + (1-q))/2, f3)

print(f"t табличне: {t\_tab:.3f}")

if t0 < t\_tab:

b0 = 0

print("t0 < t\_tab; b0=0")

if t1 < t\_tab:

b1 = 0

print("t1 < t\_tab; b1=0")

if t2 < t\_tab:

b2 = 0

print("t2 < t\_tab; b2=0")

if t3 < t\_tab:

b3 = 0

print("t3 < t\_tab; b3=0")

y1\_hat = b0 + b1\*x\_matrix[0][0] + b2\*x\_matrix[0][1] + b3\*x\_matrix[0][2]

y2\_hat = b0 + b1\*x\_matrix[1][0] + b2\*x\_matrix[1][1] + b3\*x\_matrix[1][2]

y3\_hat = b0 + b1\*x\_matrix[2][0] + b2\*x\_matrix[2][1] + b3\*x\_matrix[2][2]

y4\_hat = b0 + b1\*x\_matrix[3][0] + b2\*x\_matrix[3][1] + b3\*x\_matrix[3][2]

print(f"y1\_hat = {b0:.3f} {b1:+.3f}\*x11 {b2:+.3f}\*x12 {b3:+.3f}\*x13 "

f"= {y1\_hat:.3f}")

print(f"y2\_hat = {b0:.3f} {b1:+.3f}\*x21 {b2:+.3f}\*x22 {b3:+.3f}\*x23"

f" = {y2\_hat:.3f}")

print(f"y3\_hat = {b0:.3f} {b1:+.3f}\*x31 {b2:+.3f}\*x32 {b3:+.3f}\*x33 "

f"= {y3\_hat:.3f}")

print(f"y4\_hat = {b0:.3f} {b1:+.3f}\*x41 {b2:+.3f}\*x42 {b3:+.3f}\*x43"

f" = {y4\_hat:.3f}")

print("\n"+"-"\*15 + "Критерій Фішера" + "-"\*18)

d = 2

f4 = N - d

S2\_ad = (m / (N - d)) \* (

(y1\_hat - my1)\*\*2 +

(y2\_hat - my2)\*\*2 +

(y3\_hat - my3)\*\*2 +

(y4\_hat - my4)\*\*2

)

Fp = S2\_ad / S2B

Ft = scipy.stats.f.ppf(1 - q, f4, f3)

print(f"Fp:{Fp:.3f}")

print(f"Ft:{Ft:.3f}")

if Fp > Ft:

print("Рівняння регресії не адекватно оригіналу при рівні значимості 0,05")

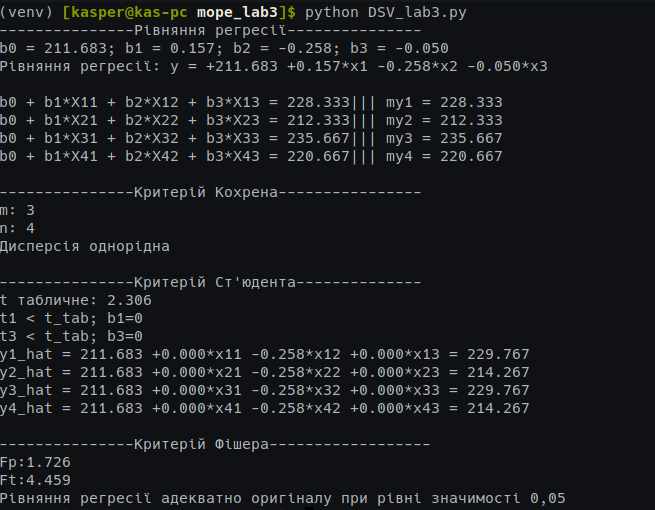
else:

print("Рівняння регресії адекватно оригіналу при рівні значимості 0,05")

else:

print("Дисперсія не однорідна, отже необхідно збільшити кількість дослідів")

**Вивід:**



**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).

2. Для чого потрібно розрахункове значення Кохрена?

Для перевірки дисперсії на однорідність.

3. 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 ts є статистично незначущим і його слід виключити з рівняння регресії. Якщо ts > tтабл то гіпотеза не підтверджується, тобто ts – значимий коефіцієнт і він залишається в рівнянні регресії.

4. 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.