

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
Варіант: **111**
Перевірив:
Регіда П.Г.

Варіант завдання:

111	10	60	-70	-10	60	70
-----	----	----	-----	-----	----	----

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:

```

(venv) [kasper@kas-pc more_lab3]$ python DSV_lab3.py
-----Рівняння регресії-----
b0 = 211.683; b1 = 0.157; b2 = -0.258; b3 = -0.050
Рівняння регресії: y = +211.683 +0.157*x1 -0.258*x2 -0.050*x3

b0 + b1*X11 + b2*X12 + b3*X13 = 228.333||| my1 = 228.333
b0 + b1*X21 + b2*X22 + b3*X23 = 212.333||| my2 = 212.333
b0 + b1*X31 + b2*X32 + b3*X33 = 235.667||| my3 = 235.667
b0 + b1*X41 + b2*X42 + b3*X43 = 220.667||| my4 = 220.667

-----Критерій Кохрена-----
m: 3
n: 4
Дисперсія однорідна

-----Критерій Ст'юдента-----
t табличне: 2.306
t1 < t_tab; b1=0
t3 < t_tab; b3=0
y1_hat = 211.683 +0.000*x11 -0.258*x12 +0.000*x13 = 229.767
y2_hat = 211.683 +0.000*x21 -0.258*x22 +0.000*x23 = 214.267
y3_hat = 211.683 +0.000*x31 -0.258*x32 +0.000*x33 = 229.767
y4_hat = 211.683 +0.000*x41 -0.258*x42 +0.000*x43 = 214.267

-----Критерій Фішера-----
Fp:1.726
Ft:4.459
Рівняння регресії адекватно оригіналу при рівні значимості 0,05

```

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 Стюдента?

Для перевірки значущості коефіцієнтів регресії. Тобто, Якщо виконується нерівність $t_s < t_{\text{табл}}$, то приймається нуль-гіпотеза, тобто вважається,

що знайдений коефіцієнт t_s є статистично незначущим і його слід виключити з рівняння регресії. Якщо $t_s > t_{\text{табл}}$ то гіпотеза не підтверджується, тобто t_s – 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.