

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.»**

Виконав:
студент 2-го курсу ФІОТ
групи ІВ-81
Юхимчук Я. М.
Номер у списку: 28
Варіант: **128**
Перевірив:
Регіда П.Г.

Варіант завдання:

128	-15	30	30	80	30	35
-----	-----	----	----	----	----	----

Код:

```
import random
import numpy as np
import scipy.stats

x1min = -15
x1max = 30
x2min = 30
x2max = 80
x3min = 30
x3max = 35

xAvmx = (x1max+x2max+x3max)/3
xAvmin = (x1min+x2min+x3min)/3
ymax = int(200+xAvmx)
ymin = int(200+xAvmin)

print("Кодоване значення X")
print("{:<5} {:<5} {:<5} {:<5}".format("№", "X1", "X2", "X3"))
X11 = ["-1", "-1", "+1", "+1"]
X22 = ["-1", "+1", "-1", "+1"]
X33 = ["-1", "+1", "+1", "-1"]
for i in range(4):
    print("{:<5} {:<5} {:<5} {:<5}".format(i+1,X11[i],X22[i],X33[i]))

print("Матриця для m=3")
print("{:<5} {:<5} {:<5} {:<5} {:<5} {:<5}"
      "{:<5}".format("№", "X1", "X2", "X3", "Y1", "Y2", "Y3"))
X1 = [x1min, x1min, x1max, x1max]
X2 = [x2min, x2max, x2min, x2max]
X3 = [x3min, x3max, x3max, x3min]
Y1 = [random.randrange(ymin,ymax, 1) for i in range(4)]
Y2 = [random.randrange(ymin,ymax, 1) for i in range(4)]
Y3 = [random.randrange(ymin,ymax, 1) for i in range(4)]
for i in range(4):
    print("{:<5} {:<5} {:<5} {:<5} {:<5} {:<5}"
          "{:<5}".format(i+1,X1[i],X2[i],X3[i],Y1[i],Y2[i],Y3[i]))

ylav1 = (Y1[0]+Y2[0]+Y3[0])/3
y2av2 = (Y1[1]+Y2[1]+Y3[1])/3
y3av3 = (Y1[2]+Y2[2]+Y3[2])/3
y4av4 = (Y1[3]+Y2[3]+Y3[3])/3

mx1 = sum(X1)/4
mx2 = sum(X2)/4
mx3 = sum(X3)/4

my = (ylav1 + y2av2 + y3av3 + y4av4)/4

a1 = (X1[0]*ylav1 + X1[1]*y2av2 + X1[2]*y3av3 + X1[3]*y4av4)/4
a2 = (X2[0]*ylav1 + X2[1]*y2av2 + X2[2]*y3av3 + X2[3]*y4av4)/4
a3 = (X3[0]*ylav1 + X3[1]*y2av2 + X3[2]*y3av3 + X3[3]*y4av4)/4

a11 = (X1[0]*X1[0] + X1[1]*X1[1] + X1[2]*X1[2] + X1[3]*X1[3])/4
a22 = (X2[0]*X2[0] + X2[1]*X2[1] + X2[2]*X2[2] + X2[3]*X2[3])/4
a33 = (X3[0]*X3[0] + X3[1]*X3[1] + X3[2]*X3[2] + X3[3]*X3[3])/4
a12 = a21 = (X1[0]*X2[0] + X1[1]*X2[1] + X1[2]*X2[2] + X1[3]*X2[3])/4
a13 = a31 = (X1[0]*X3[0] + X1[1]*X3[1] + X1[2]*X3[2] + X1[3]*X3[3])/4
a23 = a32 = (X2[0]*X3[0] + X2[1]*X3[1] + X2[2]*X3[2] + X2[3]*X3[3])/4
```

```

b01 = np.array([[my, mx1, mx2, mx3], [a1, a11, a12, a13], [a2, a12, a22, a32], [a3, a13, a23, a33]])
b02 = np.array([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]])
b0 = np.linalg.det(b01)/np.linalg.det(b02)

b11 = np.array([[1, my, mx2, mx3], [mx1, a1, a12, a13], [mx2, a2, a22, a32], [mx3, a3, a23, a33]])
b12 = np.array([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]])
b1 = np.linalg.det(b11)/np.linalg.det(b12)

b21 = np.array([[1, mx1, my, mx3], [mx1, a11, a1, a13], [mx2, a12, a2, a32], [mx3, a13, a3, a33]])
b22 = np.array([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]])
b2 = np.linalg.det(b21)/np.linalg.det(b22)

b31 = np.array([[1, mx1, mx2, my], [mx1, a11, a12, a1], [mx2, a12, a22, a2], [mx3, a13, a23, a3]])
b32 = np.array([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]])
b3 = np.linalg.det(b31)/np.linalg.det(b32)

print('Запишемо отримане рівняння регресії:  $y = \{\} + (\{\})x_1 + (\{\})x_2 + (\{\})x_3$ '.format(round(b0,2), round(b1,2), round(b2,2), round(b3,2)))

print("Середнє значення відгуку функції за рядками:")
print("y1av1="+str(round(b0 + b1*X1[0] + b2*X2[0] + b3*X3[0],2))+""+str(round(y1av1,2)))
print("y2av2="+str(round(b0 + b1*X1[1] + b2*X2[1] + b3*X3[1],2))+""+str(round(y2av2,2)))
print("y3av3="+str(round(b0 + b1*X1[2] + b2*X2[2] + b3*X3[2],2))+""+str(round(y3av3,2)))
print("y4av4="+str(round(b0 + b1*X1[3] + b2*X2[3] + b3*X3[3],2))+""+str(round(y4av4,2)))
print("Значення співпадають")

print("Дисперсія по рядкам:")
d1 = ((Y1[0] - y1av1)**2 + (Y2[0] - y2av2)**2 + (Y3[0] - y3av3)**2)/3
d2 = ((Y1[1] - y1av1)**2 + (Y2[1] - y2av2)**2 + (Y3[1] - y3av3)**2)/3
d3 = ((Y1[2] - y1av1)**2 + (Y2[2] - y2av2)**2 + (Y3[2] - y3av3)**2)/3
d4 = ((Y1[3] - y1av1)**2 + (Y2[3] - y2av2)**2 + (Y3[3] - y3av3)**2)/3
print("d1=", round(d1,2), "d2=", round(d2,2), "d3=", round(d3,2), "d4=", round(d4,2))

dcouple = [d1, d2, d3, d4]

print("Критерій Кохрена:")
# Рівень значимості приймемо 0.05
Gp = max(dcouple)/sum(dcouple)
q = 0.05
m = 3
f1 = m-1
f2 = N = 4
# для q = 0.05, f1 = 2, f2 = 4, Gt = 0.7679
Gt = 0.7679
print("Gp = ", round(Gp, 4))
print("Gt = ", round(Gt, 4))
if Gp < Gt:
    print("Gp < Gt, дисперсія однорідна")
    print("Критерій Стюдента:")
    sb = sum(dcouple) / N

```

```

ssbs = sb / N * m
sbs = ssbs ** 0.5

beta0 = (ylav1 * 1 + y2av2 * 1 + y3av3 * 1 + y4av4 * 1) / N
beta1 = (ylav1 * (-1) + y2av2 * (-1) + y3av3 * 1 + y4av4 * 1) / N
beta2 = (ylav1 * (-1) + y2av2 * 1 + y3av3 * (-1) + y4av4 * 1) / N
beta3 = (ylav1 * (-1) + y2av2 * 1 + y3av3 * 1 + y4av4 * (-1)) / N

t0 = abs(beta0) / sbs
t1 = abs(beta1) / sbs
t2 = abs(beta2) / sbs
t3 = abs(beta3) / sbs

f3 = f1 * f2
# t_tab = 2.306 # для значення f3 = 8, t табличне = 2,306
ttabl = scipy.stats.t.ppf((1 + (1 - q)) / 2, f3)
print("f3 = f1*f2, з таблиці tтабл =", round(ttabl, 2))
if (t0 < ttabl):
    print("t0<ttabl, b0 не значимий")
    b0 = 0
if (t1 < ttabl):
    print("t1<ttabl, b1 не значимий")
    b1 = 0
if (t2 < ttabl):
    print("t2<ttabl, b2 не значимий")
    b2 = 0
if (t3 < ttabl):
    print("t3<ttabl, b3 не значимий")
    b3 = 0

yy1 = b0 + b1 * x1min + b2 * x2min + b3 * x3min
yy2 = b0 + b1 * x1min + b2 * x2max + b3 * x3max
yy3 = b0 + b1 * x1max + b2 * x2min + b3 * x3max
yy4 = b0 + b1 * x1max + b2 * x2max + b3 * x3min

print("Критерій Фішера:")
# Рівень значимості прийmemo 0.05
d = 2
f4 = N - d
sad = ((yy1 - ylav1) ** 2 + (yy2 - y2av2) ** 2 + (yy3 - y3av3) ** 2 +
(yy4 - y4av4) ** 2) * (m / (N - d))
Fp = sad / sb
print("d1 =", round(d1, 2), "d2=", round(d2, 2), "d3=", round(d3, 2),
"d4=", round(d4, 2), "d5=", round(sb, 2))
print("Fp =", round(Fp, 2))
# Ft = 4.5 # для f3=8; f4=2
Ft = scipy.stats.f.ppf(1 - q, f4, f3)
print("Ft =", round(Ft, 2))
if Fp > Ft:
    print("Fp > Ft, рівняння неадекватно оригіналу")
else:
    print("Fp < Ft, рівняння адекватно оригіналу")
else:
    print("Gp > Gt, дисперсія неоднорідна, необхідно збільшити кількість
дослідів")

```

Вивід:

```
Кодоване значення X
№      X1      X2      X3
1       -1      -1      -1
2       -1      +1      +1
3       +1      -1      +1
4       +1      +1      -1

Матриця для m=3
№      X1      X2      X3      Y1      Y2      Y3
1      -15     30     30     222     215     226
2      -15     80     35     226     222     228
3       30     30     35     236     239     215
4       30     80     30     232     234     223

Запишемо отримане рівняння регресії:  $y = 208.02 + (0.15)*x_1 + (0.04)*x_2 + (0.47)*x_3$ 
Середнє значення відгуку функції за рядками:
y1av1=221.0=221.0
y2av2=225.33=225.33
y3av3=230.0=230.0
y4av4=229.67=229.67
Значення співпадають
Дисперсія по рядкам:
d1= 41.26 d2= 13.37 d3= 212.26 d4= 81.7
Критерій Кохрена:
Gr = 0.6089
Gt = 0.7679
Gr < Gt, дисперсія однорідна
Критерій Стьюдента:
f3 = f1*f2, з таблиці tтабл = 2.31
t1<tтабл, b1 не значимий
t2<tтабл, b2 не значимий
t3<tтабл, b3 не значимий
Критерій Фішера:
d1 = 41.26 d2= 13.37 d3= 212.26 d4= 81.7 d5= 87.15
Fr = 24.43
Ft = 4.46
Fr > Ft, рівняння неадекватно оригіналу
```

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 β_s є статистично незначущим і його слід виключити з рівняння регресії. Якщо $t_s > t_{\text{табл}}$ то гіпотеза не підтверджується, тобто β_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 y , 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.