

Національний технічний університет України
«Київський політехнічний інститут імені 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 ІО-82
Скібінський А.О.
Номер у списку: 19
Перевірив:
Регіда П.Г.

Київ – 2020

Варіант:

219	15	45	-70	-10	15	30
-----	----	----	-----	-----	----	----

Лістинг коду програми:

```
import numpy as np

X1max = 45
X1min = 15
X2max = -10
X2min = -70
X3max = 30
X3min = 15
Ymax = 200 + (X1max + X2max + X3max)/3
Ymin = 200 + (X1min + X2min + X3min)/3
X = [[X1min, X2min, X3min],
      [X1min, X2max, X3max],
      [X1max, X2min, X3max],
      [X1max, X2max, X3min]]
x = [[1, -1, -1, -1],
      [1, -1, 1, 1],
      [1, 1, -1, 1],
      [1, 1, 1, -1]]
m = 3
N = 4
print("Кодовані значення факторів: ")
for row in x:
    for i in row:
        print("{:4d}".format(int(i)), end = " |")
    print()
print("Натуральні значення факторів: ")
for row in X:
    for i in row:
        print("{:4d}".format(int(i)), end = " |")
    print()

def lab3(m, N):
    A = np.random.randint(Ymin, Ymax, (N, m))
    print("Згенерована матриця значень Y: ")
    for row in A:
        for i in row:
            print("{:4d}".format(int(i)), end=" |")
        print()

    print("Середні значення функції відгуку по рядкам:")
    Yall = np.sum(A, axis=1) / m
    Y1, Y2, Y3, Y4 = Yall
    print("Y1 = ", round(Y1, 3))
    print("Y2 = ", round(Y2, 3))
    print("Y3 = ", round(Y3, 3))
    print("Y4 = ", round(Y4, 3))

    print("Середні значення натуральних значень факторів по стовпчикам:")
    mx1 = (X[0][0] + X[1][0] + X[2][0] + X[3][0]) / N
    mx2 = (X[0][1] + X[1][1] + X[2][1] + X[3][1]) / N
    mx3 = (X[0][2] + X[1][2] + X[2][2] + X[3][2]) / N
    print("mx1 = ", mx1)
    print("mx2 = ", mx2)
    print("mx3 = ", mx3)
```

```

print("Середнє значення Y:")
my = (Y1 + Y2 + Y3 + Y4) / N
print("my = ", round(my, 3))

a1 = (X[0][0] * Y1 + X[1][0] * Y2 + X[2][0] * Y3 + X[3][0] * Y4) / N
a2 = (X[0][1] * Y1 + X[1][1] * Y2 + X[2][1] * Y3 + X[3][1] * Y4) / N
a3 = (X[0][2] * Y1 + X[1][2] * Y2 + X[2][2] * Y3 + X[3][2] * Y4) / N

a11 = (X[0][0] * X[0][0] + X[1][0] * X[1][0] + X[2][0] * X[2][0] + X[3][0] *
X[3][0]) / N
a22 = (X[0][1] * X[0][1] + X[1][1] * X[1][1] + X[2][1] * X[2][1] + X[3][1] *
X[3][1]) / N
a33 = (X[0][2] * X[0][2] + X[1][2] * X[1][2] + X[2][2] * X[2][2] + X[3][2] *
X[3][2]) / N

a12 = a21 = (X[0][0] * X[0][1] + X[1][0] * X[1][1] + X[2][0] * X[2][1] + X[3][0]
* X[3][1]) / N
a13 = a31 = (X[0][0] * X[0][2] + X[1][0] * X[1][2] + X[2][0] * X[2][2] + X[3][0]
* X[3][2]) / N
a23 = a32 = (X[0][1] * X[0][2] + X[1][1] * X[1][2] + X[2][1] * X[2][2] + X[3][1]
* X[3][2]) / N

B = [[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a21, a22, a23], [mx3, a31,
a32, a33]]
B0 = [[my, mx1, mx2, mx3], [a1, a11, a12, a13], [a2, a21, a22, a23], [a3, a31,
a32, a33]]
B1 = [[1, my, mx2, mx3], [mx1, a1, a12, a13], [mx2, a2, a22, a23], [mx3, a3, a32,
a33]]
B2 = [[1, mx1, my, mx3], [mx1, a11, a1, a13], [mx2, a21, a2, a23], [mx3, a31, a3,
a33]]
B3 = [[1, mx1, mx2, my], [mx1, a11, a12, a1], [mx2, a21, a22, a2], [mx3, a31,
a32, a3]]

print("Коефіцієнти рівняння регресії:")
b0 = np.linalg.det(B0) / np.linalg.det(B)
b1 = np.linalg.det(B1) / np.linalg.det(B)
b2 = np.linalg.det(B2) / np.linalg.det(B)
b3 = np.linalg.det(B3) / np.linalg.det(B)
print("b0 = ", round(b0, 3))
print("b1 = ", round(b1, 3))
print("b2 = ", round(b2, 3))
print("b3 = ", round(b3, 3))

print("Підставимо значення факторів з матриці планування в рівняння регресії:")
y1 = b0 + b1 * X[0][0] + b2 * X[0][1] + b3 * X[0][2]
y2 = b0 + b1 * X[1][0] + b2 * X[1][1] + b3 * X[1][2]
y3 = b0 + b1 * X[2][0] + b2 * X[2][1] + b3 * X[2][2]
y4 = b0 + b1 * X[3][0] + b2 * X[3][1] + b3 * X[3][2]
print("y1 = ", round(y1, 3))
print("y2 = ", round(y2, 3))
print("y3 = ", round(y3, 3))
print("y4 = ", 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("Перевіркою переконуємось, що коефіцієнти рівняння регресії знайдено
правильно")
else:
    print("Перевіркою переконуємось, що коефіцієнти рівняння регресії знайдено
неправильно")

```

```

print("Перевірка однорідності дисперсії за критерієм Кохрена:")

print("Знайдемо дисперсії по рядках:")
D1 = sum([(i - Y1) ** 2 for i in A[0]]) / m
D2 = sum([(i - Y2) ** 2 for i in A[1]]) / m
D3 = sum([(i - Y3) ** 2 for i in A[2]]) / m
D4 = sum([(i - Y4) ** 2 for i in A[3]]) / m
print("D1 = ", round(D1, 3))
print("D2 = ", round(D2, 3))
print("D3 = ", round(D3, 3))
print("D4 = ", round(D4, 3))

Dmax = max(D1, D2, D3, D4)
Dsum = D1 + D2 + D3 + D4
Gp = Dmax / Dsum
print("Коефіцієнт Gp = ", round(Gp, 5))

print("Ступені свободи: ")
f1 = m - 1
f2 = N
print("f1 = ", f1)
print("f2 = ", f2)

Gtable = {3: 0.6841, 4: 0.6287, 5: 0.5892, 6: 0.5598, 7: 0.5365, 8: 0.5175, 9:
0.5017,
          10: 0.4884, range(11, 17): 0.4366, range(17, 37): 0.3720, range(37,
145): 0.3093}
Gt = Gtable.get(m)
print("За таблицею Gt = ", Gt)

if (Gp < Gt):
    print("Gp < Gt, отже дисперсія однорідна. Критерій Кохрена виконується")
else:
    print("Gp > Gt, отже дисперсія неоднорідна. Збільшуємо кількість дослідів на
1 ")
    m = m + 1
    lab3(m, N)
    return

print("Оцінимо значимість коефіцієнтів регресії згідно критерію Стюдента:")

print("Середнє значення дисперсії:")
mD = Dsum / N
print("mD = ", round(mD, 3))

print("Статистична оцінка дисперсії:")
Db = mD / (m * N)
sD = Db ** 0.5
print("Дисперсія відносності Db = ", round(Db, 3))
print("sD = ", round(sD, 3))

print("Визначення оцінок коефіцієнтів:")
beta0 = (Y1 * x[0][0] + Y2 * x[1][0] + Y3 * x[2][0] + Y4 * x[3][0]) / N
beta1 = (Y1 * x[0][1] + Y2 * x[1][1] + Y3 * x[2][1] + Y4 * x[3][1]) / N
beta2 = (Y1 * x[0][2] + Y2 * x[1][2] + Y3 * x[2][2] + Y4 * x[3][2]) / N
beta3 = (Y1 * x[0][3] + Y2 * x[1][3] + Y3 * x[2][3] + Y4 * x[3][3]) / N
print("beta0 = ", round(beta0, 3))
print("beta1 = ", round(beta1, 3))
print("beta2 = ", round(beta2, 3))
print("beta3 = ", round(beta3, 3))

```

```

print("Оцінка за t-критерієм Стюдента:")
t0 = abs(beta0) / sD
t1 = abs(beta1) / sD
t2 = abs(beta2) / sD
t3 = abs(beta3) / sD
print("t0 = ", round(t0, 3))
print("t1 = ", round(t1, 3))
print("t2 = ", round(t2, 3))
print("t3 = ", round(t3, 3))

print("Ступені свободи: ")
f3 = f1 * f2
print("f3 = ", f3)

Ttabl = 2.306
print("За таблицю в 8 рядку Ttabl = ", Ttabl)

d = 0

if (t0 < Ttabl):
    print("Коефіцієнт b0 є статистично незначущим, виключаємо його з рівняння регресії")
    b0 = 0
else:
    d += 1
    print("Гіпотеза не підтверджується, тобто b0 – значимий коефіцієнт і він залишається в рівнянні регресії.")
if (t1 < Ttabl):
    print("Коефіцієнт b1 є статистично незначущим, виключаємо його з рівняння регресії")
    b1 = 0
else:
    d += 1
    print("Гіпотеза не підтверджується, тобто b1 – значимий коефіцієнт і він залишається в рівнянні регресії.")
if (t2 < Ttabl):
    print("Коефіцієнт b2 є статистично незначущим, виключаємо його з рівняння регресії")
    b2 = 0
else:
    d += 1
    print("Гіпотеза не підтверджується, тобто b2 – значимий коефіцієнт і він залишається в рівнянні регресії.")
if (t3 < Ttabl):
    print("Коефіцієнт b3 є статистично незначущим, виключаємо його з рівняння регресії")
    b3 = 0
else:
    d += 1
    print("Гіпотеза не підтверджується, тобто b3 – значимий коефіцієнт і він залишається в рівнянні регресії.")

y_1 = b0 + b1 * X[0][0] + b2 * X[0][1] + b3 * X[0][2]
y_2 = b0 + b1 * X[1][0] + b2 * X[1][1] + b3 * X[1][2]
y_3 = b0 + b1 * X[2][0] + b2 * X[2][1] + b3 * X[2][2]
y_4 = b0 + b1 * X[3][0] + b2 * X[3][1] + b3 * X[3][2]
print("y_1 = ", round(y_1, 3))
print("y_2 = ", round(y_2, 3))
print("y_3 = ", round(y_3, 3))
print("y_4 = ", round(y_4, 3))

print("Перевірка адекватності за критерієм Фішера:")

```

```

    print("Кількість значущих коефіцієнтів d = ", d)

    Dad = (m / (N - d)) * ((y_1 - Y1) ** 2 + (y_2 - Y2) ** 2 + (y_3 - Y3) ** 2 + (y_4
- Y4) ** 2)
    print("Дисперсія адекватності Dad = ", round(Dad, 3))

    Fp = Dad / Db
    print("Перевірка адекватності Fp = ", round(Fp, 3))

    print("Ступені свободи: ")
    f4 = N - d
    print("f4 = ", f4)

    if (f4 == 1):
        Ft = 5.3
    elif (f4 == 2):
        Ft = 4.5
    elif (f4 == 3):
        Ft = 4.1
    elif (f4 == 4):
        Ft = 3.8
    print("За таблицею Ft = ", Ft)

    if (Fp < Ft):
        print("Fp < Ft, отримана математична модель адекватна експериментальним
даним.")
    else:
        print("Fp > Ft, отже, рівняння регресії неадекватно оригіналу")

lab3(m, N)

```

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 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.