

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА 2
з дисципліни «Методи оптимізації та планування експерименту»

Виконав:
Студентка 2 курсу ФІОТ
групи ІВ-92
Гайдукевич М. А.

Перевірив:
Регіда П.Г.

Мета: провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.

$$y_{\max} = (30 - N_{\text{варіанту}}) * 10 = 260,$$

$$y_{\min} = (20 - N_{\text{варіанту}}) * 10 = 160.$$

Варіант:

204	15	45	15	50
-----	----	----	----	----

Код програми

```
import math
import numpy as np
import random as rnd

m = 5
y_min, y_max = 160, 260

x1_min, x1_max = 15, 45
x2_min, x2_max = 15, 50
x1_min_norm, x1_max_norm = -1, 1
x2_min_norm, x2_max_norm = -1, 1

p_prob = (0.99, 0.98, 0.95, 0.90)
rkr_table = {2: (1.73, 1.72, 1.71, 1.69),
              6: (2.16, 2.13, 2.10, 2.00),
              8: (2.43, 4.37, 2.27, 2.17),
              10: (2.62, 2.54, 2.41, 2.29),
              12: (2.75, 2.66, 2.52, 2.39),
              15: (2.9, 2.8, 2.64, 2.49),
              20: (3.08, 2.96, 2.78, 2.62)}

matrix_of_y = [[rnd.randint(y_min, y_max) for i in range(m)] for j in range(3)]
average_y = [sum(matrix_of_y[i][j]) for j in range(m)] / m for i in range(3)]

quadric_sigma1 = sum([(j - average_y[0]) ** 2 for j in matrix_of_y[0]]) / m
quadric_sigma2 = sum([(j - average_y[1]) ** 2 for j in matrix_of_y[1]]) / m
quadric_sigma3 = sum([(j - average_y[2]) ** 2 for j in matrix_of_y[2]]) / m

teta_sigma = math.sqrt((2 * (2 * m - 2)) / (m * (m - 4)))

Fuv1 = quadric_sigma1 / quadric_sigma2
Fuv2 = quadric_sigma3 / quadric_sigma1
Fuv3 = quadric_sigma3 / quadric_sigma2

TetaUV1 = ((m - 2) / m) * Fuv1
TetaUV2 = ((m - 2) / m) * Fuv2
TetaUV3 = ((m - 2) / m) * Fuv3

Ruv1 = abs(TetaUV1 - 1) / teta_sigma
Ruv2 = abs(TetaUV2 - 1) / teta_sigma
Ruv3 = abs(TetaUV3 - 1) / teta_sigma

mx1 = (-1 + 1 - 1) / 3
mx2 = (-1 - 1 + 1) / 3
my = sum(average_y) / 3
a1 = (1 + 1 + 1) / 3
```

```

a2 = (1 - 1 - 1) / 3
a3 = (1 + 1 + 1) / 3
a11 = (-1 * average_y[0] + 1 * average_y[1] - 1 * average_y[2]) / 3
a22 = (-1 * average_y[0] - 1 * average_y[1] + 1 * average_y[2]) / 3

b0 = np.linalg.det(np.dot([[my, mx1, mx2], [a11, a1, a2], [a22, a2, a3]],
                           np.linalg.inv([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])))

b1 = np.linalg.det(np.dot([[1, my, mx2], [mx1, a11, a2], [mx2, a22, a3]],
                           np.linalg.inv([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])))

b2 = np.linalg.det(np.dot([[1, mx1, my], [mx1, a1, a11], [mx2, a2, a22]],
                           np.linalg.inv([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])))

def self_dispersion():
    M = 0
    M = min(rkr_table, key=lambda x: abs(x - M))
    p = 0
    for ruv in (Ruv1, Ruv2, Ruv3):
        if ruv > rkr_table[M][0]:
            return False
        for rkr in range(len(rkr_table[M])):
            if ruv < rkr_table[M][rkr]:
                p = rkr
    return p_prob[p]

def regressionCheck():
    y_norm1 = round(b0 - b1 - b2, 2)
    y_norm2 = round(b0 + b1 - b2, 2)
    y_norm3 = round(b0 - b1 + b2, 2)

    if y_norm1 == average_y[0] and y_norm2 == average_y[1] and y_norm3 == average_y[2]:
        print("Результат збігається з середніми значеннями y")
    else:
        print("Результат НЕ збігається з середніми значеннями y")

delta_x1 = math.fabs(x1_max - x1_min) / 2
delta_x2 = math.fabs(x2_max - x2_min) / 2
x10 = (x1_max + x1_min) / 2
x20 = (x2_max + x2_min) / 2

A0 = b0 - b1 * x10 / delta_x1 - b2 * x20 / delta_x2
A1 = b1 / delta_x1
A2 = b2 / delta_x2

def naturalized_regression(x1, x2):
    return A0 + A1 * x1 + A2 * x2

# output
for i in range(3):
    print("Y{:} = {}, Average: {}".format(i + 1, matrix_of_y[i], average_y[i]))
print()
print("σ² y1:", quadric_sigma1, "\nσ² y2:", quadric_sigma2, "\nσ² y3:", quadric_sigma2)
print("σθ =", teta_sigma)
print()
print("Fuv1 =", Fuv1, "\nFuv2 =", Fuv2, "\nFuv3 =", Fuv3)
print()
print("θuv1 =", TetaUV1, "\nθuv2 =", TetaUV2, "\nθuv3 =", TetaUV3)
print()
print("Ruv1 =", Ruv1, "\nRuv2 =", Ruv2, "\nRuv3 =", Ruv3)

```

```

print()
print("Однорідна дисперсія:", self_dispersion())
print()
print("mx1:", mx1, "\nmx2:", mx2, "\nmy:", my)
print()
print("a1:", a1, "\na2:", a2, "\na3:", a3)
print("a11:", a11, "\na22:", a22)
print()
print("b0:", b0, "\nb1:", b1, "\nb2:", b2)
print("Натуралізація коефіцієнтів:")
print("Δx1:", delta_x1, "\nΔx2:", delta_x2)
print("x10:", x10, "\nx20:", x20)
print("a0:", A0, "a1:", A1, "a2:", A2)
print()
print("Натуралізоване рівняння регресії:")
naturReg_Y = [round(naturalized_regression(x1_min, x2_min), 2),
               round(naturalized_regression(x1_max, x2_min), 2),
               round(naturalized_regression(x1_min, x2_max), 2)]
print(naturReg_Y)

if naturReg_Y == average_y:
    print("Коефіцієнти натуралізованого рівняння регресії вірні")
else:
    print("Коефіцієнти натуралізованого рівняння регресії НЕ вірні")

regressionCheck()

```

Результати:

C:\Python\python.exe C:/labs/MND/lab2.py

Y1: [232, 191, 255, 255, 191], Average: 224.8

Y2: [225, 193, 231, 187, 224], Average: 212.0

Y3: [183, 182, 227, 184, 256], Average: 206.4

σ^2 y1: 832.1599999999999

σ^2 y2: 332.0

σ^2 y3: 332.0

σ_0 = 1.7888543819998317

Fuv1 = 2.5065060240963852

Fuv2 = 1.0885406652566816

Fuv3 = 2.7284337349397587

θuv1 = 1.503903614457831

θuv2 = 0.6531243991540089

θuv3 = 1.6370602409638553

Ruv1 = 0.281690684008889

Ruv2 = 0.19390935580692992

Ruv3 = 0.3561275011393941

Однорідна дисперсія: 0.9

mx1: -0.3333333333333333

mx2: -0.3333333333333333

my: 214.4

a1: 1.0

a2: -0.3333333333333333

a3: 1.0

a11: -73.06666666666668

a22: -76.8

```
b0: 209.1999999999984
b1: -6.399999999997996
b2: -9.199999999997662
Натуралізація коефіцієнтів:
Δx1: 15.0
Δx2: 17.5
x10: 30.0
x20: 32.5
a0: 239.08571428570434 a1: -0.426666666666533 a2: -0.5257142857141521
```

```
Натуралізоване рівняння регресії:
[224.8, 212.0, 206.4]
Коефіцієнти натуралізованого рівняння регресії вірні
Результат збігається з середніми значеннями у
```

```
Process finished with exit code 0
```

Висновок:

У процесі виконання даної лабораторної роботи був проведений багатофакторний експеримент, перевірена однорідність дисперсії за критерієм Романовського, отримані коефіцієнти рівняння регресії та проведена натуралізація рівняння регресії.

Мета лабораторної роботи досягнена.

Результати виконання завдання наведені вище.

Відповіді на контрольні питання:

1) Що таке регресійні поліноми і де вони застосовуються?

Регресійні поліноми - апроксимуючі поліноми, за допомогою яких ми можемо описати функцію. Застосовуються для оцінки результатів вимірів.

2) Визначення однорідності дисперсії.

Кожне R_{uv} (експериментальне значення критерію Романовського) порівнюється з $R_{кр}$ (значення критерію Романовського за різних довірчих ймовірностей p) і якщо для усіх $u = \overline{1, N}$; $v = \overline{1, N}$ кожне $R_{uv} < R_{кр}$, то гіпотеза про однорідність дисперсій підтверджується з ймовірністю p .

3) Що називається повним факторним експериментом?

Повний факторний експеримент - багатофакторний експеримент, де використані всі можливі комбінації рівнів факторів.