

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

**Лабораторна робота №2**  
З дисципліни «Методи оптимізації та планування»  
**ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З**  
**ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ**

ВИКОНАВ:  
Студент II курсу ФІОТ  
Групи ІО-91  
Лаппо М.О. - 9119

ПЕРЕВІРИВ:  
Регіда П.Г.

Київ 2021 р.

**Мета:**

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

**Варіант завдання:**

Варіант	X <sub>1</sub>		X <sub>2</sub>	
	min	max	min	max
116	-10	50	-20	60

$$Y_{\max} = (30 - 16) * 10 = 140$$

$$Y_{\min} = (20 - 16) * 10 = 40$$

**Код:**

```
import math
from random import randint
import numpy as np

p_list = (0.99, 0.98, 0.95, 0.90)

RKtable = {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: (2.90, 2.80, 2.64, 2.49),
             20: (3.08, 2.96, 2.78, 2.62)}

minYlim = 40
maxYlim = 140
m = 5
X1min = -10
X1min_n = -1
X1max = 50
X1max_n = 1
X2min = -20
X2min_n = -1
X2max = 60
X2max_n = 1

Ymatr = [[randint(minYlim, maxYlim) for i in range(m)] for j in range(3)]
Yavg = [sum(Ymatr[i][j] for j in range(m)) / m for i in range(3)]

sig1 = sum([(j - Yavg[0]) ** 2 for j in Ymatr[0]]) / m
sig2 = sum([(j - Yavg[1]) ** 2 for j in Ymatr[1]]) / m
sig3 = sum([(j - Yavg[2]) ** 2 for j in Ymatr[2]]) / m

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

fuv1 = sig1 / sig2
fuv2 = sig3 / sig1
fuv3 = sig3 / sig2

Tuv1 = ((m - 2) / m) * fuv1
Tuv2 = ((m - 2) / m) * fuv2
Tuv3 = ((m - 2) / m) * fuv3
```

```

ruv1 = abs(Tuv1 - 1) / sigT
ruv2 = abs(Tuv2 - 1) / sigT
ruv3 = abs(Tuv3 - 1) / sigT

MX1 = (-1 + 1 - 1) / 3
MX2 = (-1 - 1 + 1) / 3
MY = sum(Yavg) / 3
A1 = (1 + 1 + 1) / 3
A2 = (1 - 1 - 1) / 3
A3 = (1 + 1 + 1) / 3
A11 = (-1 * Yavg[0] + 1 * Yavg[1] - 1 * Yavg[2]) / 3
A22 = (-1 * Yavg[0] - 1 * Yavg[1] + 1 * Yavg[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 checkRegression():
    NY1 = round(b0 - b1 - b2, 1)
    NY2 = round(b0 + b1 - b2, 1)
    NY3 = round(b0 - b1 + b2, 1)
    if NY1 == Yavg[0] and NY2 == Yavg[1] and NY3 == Yavg[2]:
        print("Значення перевірки нормованого рівняння регресії сходяться")
    else:
        print("Значення перевірки нормованого рівняння регресії НЕ сходяться")

NORM_Y = b0 - b1 + b2

DX1 = math.fabs(X1max - X1min) / 2
DX2 = math.fabs(X2max - X2min) / 2
X10 = (X1max + X1min) / 2
X20 = (X2max + X2min) / 2

AA0 = b0 - b1 * X10 / DX1 - b2 * X20 / DX2
AA1 = b1 / DX1
AA2 = b2 / DX2

def odnor_disp():
    m1 = min(RKRtable, key=lambda x: abs(x - m))
    p = 0
    for ruv in (ruv1, ruv2, ruv3):
        if ruv > RKRtable[m1][0]:
            return False
        for rkr in range(len(RKRtable[m1])):
            if ruv < RKRtable[m1][rkr]:
                p = rkr
    return p_list[p]

```

```

def nat_reg(x1, x2):
    return AA0 + AA1 * x1 + AA2 * x2

# output
for i in range(3):
    print("Y{}: {}, Average: {}".format(i + 1, Ymatr[i], Yavg[i]))
print()
print("σ² y1:", sig1)
print("σ² y2:", sig2)
print("σ² y3:", sig3)
print("σθ =", sigT)
print("-----")
print("Fuv1 =", fuv1)
print("Fuv2 =", fuv2)
print("Fuv3 =", fuv3)
print("-----")
print("θuv1 =", Tuv1)
print("θuv2 =", Tuv2)
print("θuv3 =", Tuv3)
print("-----")
print("Ruv1 =", ruv1)
print("Ruv2 =", ruv2)
print("Ruv3 =", ruv3)
print("-----")
print("Однорідна дисперсія:", odnor_disp())
print("-----")
print("mx1:", MX1)
print("mx2:", MX2)
print("my:", MY)
print("a1:", A1)
print("a2:", A2)
print("a3:", A3)
print("a11:", A11)
print("a22:", A22)
print("b0:", b0)
print("b1:", b1)
print("b2:", b2)
print("Натуралізація коефіцієнтів:")
print("Δx1:", DX1)
print("Δx2:", DX2)
print("x10:", X10)
print("x20:", X20)
print("a0:", AA0)
print("a1:", AA1)
print("a2:", AA2)
print("-----")
print("Натуралізоване рівняння регресії:")

Ynr = [round(nat_reg(X1min, X2min), 2),
        round(nat_reg(X1max, X2min), 2),
        round(nat_reg(X1min, X2max), 2)]
print(Ynr)
if Ynr == Yavg:
    print("Коефіцієнти натуралізованого рівняння регресії вірні")
else:
    print("Коефіцієнти натуралізованого рівняння регресії НЕ вірні")
checkRegression()

```

## Результат виконання роботи:

```
C:\TeoriyaUmovirnostey\venv\Scripts\python.exe C:/TeoriyaUmovirnostey/venv/
Y1: [50, 83, 76, 52, 70], Average: 66.2
Y2: [69, 125, 86, 140, 42], Average: 92.4
Y3: [122, 130, 79, 126, 41], Average: 99.6

 $\sigma^2$  y1: 171.36
 $\sigma^2$  y2: 1291.44
 $\sigma^2$  y3: 1196.24
 $\sigma\theta = 1.7888543819998317$ 
-----
Fuv1 = 0.1326890912469801
Fuv2 = 6.980859010270774
Fuv3 = 0.9262838381961221
-----
 $\theta_{uv1} = 0.07961345474818807$ 
 $\theta_{uv2} = 4.188515406162464$ 
 $\theta_{uv3} = 0.5557703029176733$ 
-----
Ruv1 = 0.5145117201898095
Ruv2 = 1.7824342988711555
Ruv3 = 0.24833195007505565
-----
Однорідна дисперсія: 0.9
-----
mx1: -0.3333333333333333
mx2: -0.3333333333333333
my: 86.06666666666668
a1: 1.0
a2: -0.3333333333333333
a3: 1.0
a11: -24.466666666666665
a22: -19.666666666666675
b0: 96.000000000000033
b1: 13.1000000000000673
b2: 16.700000000000017
Натуралізація коефіцієнтів:
 $\Delta x1$ : 30.0
 $\Delta x2$ : 40.0
x10: 20.0
x20: 20.0
a0: 78.91666666666653
a1: 0.4366666666666891
a2: 0.4175000000000004
-----
Натуралізоване рівняння регресії:
[66.2, 92.4, 99.6]
Коефіцієнти натуралізованого рівняння регресії вірні
Значення перевірки нормованого рівняння регресії сходяться

Process finished with exit code 0
```

## Контрольні запитання:

1. Що таке регресійні поліноми і де вони застосовуються?  
Регресійні поліноми – це апроксимуючі поліноми, за допомогою яких ми можемо описати функцію. Застосовуються в теорії планування експерименту.
2. Визначення однорідності дисперсії.  
Опираючись на вимоги регресивного аналізу достовірне оброблення та використання вихідних даних експериментальних досліджень можливе лише тоді, коли дисперсії вимірювання функцій відгуку в кожній точці експерименту є однаковими. Дана властивість називається однорідністю дисперсії.
3. Що називається повним факторним експериментом?  
ПФЕ – багатофакторний експеримент в якому використовуються всі можливі комбінації рівні факторів.  $N_{\text{ПФЕ}} = 2^k$  або  $3^k$  або  $5^k$ .

## Висновок:

В даній лабораторній роботі я провела двофакторний експеримент з перевіркою дисперсій на однорідність за критерієм Романовського і отримала коефіцієнти рівняння регресії. Також провела натуралізацію рівняння регресії.