

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 2 курсу ФІОТ
Групи ІО-91
Варіант 15
Лазарєв М.О.

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 ₁		X ₂	
	min	max	min	max
115	10	50	-20	60

$$Y_{\max} = (30 - 15) * 10 = 150$$

$$Y_{\min} = (20 - 15) * 10 = 50$$

Код програми

```
import math
from random import randint
import numpy as np
maxYlim = 150
minYlim = 50
m = 5
X1min = 10
X1min_n = -1
X1max = 50
X1max_n = 1
X2min = -20
X2min_n = -1
X2max = 60
X2max_n = 1
p_list = (0.85, 0.84, 0.81, 0.76)
RKR_MATRIX = {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)}
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]])))

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 Dispers():
    m1 = min(RKR_MATRIX, key=lambda x: abs(x - m))
    p = 0
    for ruv in (ruv1, ruv2, ruv3):
        if ruv > RKR_MATRIX[m1][0]:
            return False
        for rkr in range(len(RKR_MATRIX[m1])):
            if ruv < RKR_MATRIX[m1][rkr]:
                p = rkr
    return p_list[p]
def line_equation(x1, x2):
    return AA0 + AA1 * x1 + AA2 * x2
for i in range(3):
    print("Y{}: {}, Average: {}".format(i + 1, Ymatr[i], Yavg[i]))
print("σ² y1:", sig1)
print("σ² y2:", sig2)
print("σ² y3:", sig3)
print("σθ =", sigT)
print("-"*60)
print("Fuv1 =", fuv1)
print("Fuv2 =", fuv2)
print("Fuv3 =", fuv3)
print("-"*60)
print("θuv1 =", Tuv1)
print("θuv2 =", Tuv2)
print("θuv3 =", Tuv3)
print("-"*60)
print("Ruv1 =", ruv1)
print("Ruv2 =", ruv2)
print("Ruv3 =", ruv3)
print("-"*60)

```

```

print("Однорідна дисперсія:", Dispers())
print("-"*60)
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("-"*60)
Ynr = [round(line_equation(Xlmin, X2min), 4),
        round(line_equation(Xlmax, X2min), 4),
        round(line_equation(Xlmin, X2max), 4)]
print("Натуралізоване рівняння регресії:", Ynr)
print("-"*60)
if Ynr == Yavg:
    print("Коефіцієнти натуралізованого рівняння регресії вірні")
else:
    print("Коефіцієнти натуралізованого рівняння регресії не вірні")
def check():
    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")
check()

```

Результати програми

```
C:\Users\MSI\anaconda3\python.exe "D:/2 курс 2 семестр/МОПЕ/МОПЕ_LAB2.py"
Y1: [93, 119, 148, 146, 105], Average: 122.2
Y2: [80, 106, 121, 66, 122], Average: 99.0
Y3: [82, 71, 110, 140, 58], Average: 92.2
 $\sigma^2$  y1: 478.16
 $\sigma^2$  y2: 502.4
 $\sigma^2$  y3: 864.96
 $\sigma_{\theta} = 1.7888543819998317$ 
-----
Fuv1 = 0.9517515923566879
Fuv2 = 1.8089342479504769
Fuv3 = 1.7216560509554142
-----
 $\theta_{uv1} = 0.5710509554140127$ 
 $\theta_{uv2} = 1.085360548770286$ 
 $\theta_{uv3} = 1.0329936305732486$ 
-----
Ruv1 = 0.23978980564446392
Ruv2 = 0.04771799741176139
Ruv3 = 0.018444000196574783
-----
Однорідна дисперсія: 0.76
-----
mx1: -0.3333333333333333
mx2: -0.3333333333333333
```

```

my: 104.46666666666665
a1: 1.0
a2: -0.3333333333333333
a3: 1.0
a11: -38.46666666666667
a22: -43.0
b0: 95.59999999999953
b1: -11.600000000000666
b2: -15.000000000000686
Натуралізація коефіцієнтів:
Δx1: 20.0
Δx2: 40.0
x10: 30.0
x20: 20.0
a0: 120.50000000000087
a1: -0.5800000000000333
a2: -0.37500000000001715
-----
Натуралізоване рівняння регресії: [122.2, 99.0, 92.2]
-----
Коефіцієнти натуралізованого рівняння регресії вірні
Значення перевірки нормованого рівняння регресії сходяться

Process finished with exit code 0

```

Відповіді на 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?
 - Повний факторний експеримент – багатфакторний експеримент в якому використовуються всі можливі комбінації рівні факторів. $N_{\text{ПФЕ}} = 2^k$ або 3^k або 5^k

Висновок:

Провів двофакторний експеримент, перевінив однорідність дисперсії за критерієм Романовського, отримав коефіцієнти рівняння регресії і провів натуралізацію рівняння регресії.