

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

ЛАБОРАТОРНА РОБОТА №2

з дисципліни «Методи оптимізації та планування експерименту» на тему:

**«ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»**

Виконав:
студент II курсу ФІОТ
групи ІВ-81 :
Бухтій О. В.
Перевірив:
Регіда П. Г.

Київ-2020

ЛАБОРАТОРНА РОБОТА №2.

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

Завдання

	X ₁		X ₂	
	min	max	min	max
107	-5	15	-15	35

$$Y_{\max} = 230$$

$$Y_{\min} = 130$$

Лістинг програми

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

n_var = 7

ymax = (30-n_var)*10
ymin = (20-n_var)*10

x1min, x1max = -5, 15
x2min, x2max = -15, 35

m = 5

def main(m):

    x_norm = np.matrix([[[-1,-1],
                           [-1,+1],
                           [+1,-1]]])

    x_asis = np.matrix([[x1min, x2min],
                           [x1min, x2max],
                           [x1max, x2min]])

    tmp_matrix = []
    for i in x_asis:
        tmp_arr = []
        for i in range(m):
            tmp_arr.append(rnd.randint(ymin,ymax))
```

```

        tmp_matrix.append(tmp_arr)
tmp_matrix = np.matrix(tmp_matrix)
x_asis = np.append(x_asis, tmp_matrix, axis = 1)
print(x_asis)

x_norm = np.append(x_norm, tmp_matrix, axis = 1)

med = []
disp = []
for i in range(len(x_asis.tolist())):
    row = x_asis.tolist()[i][2:]
    print(row)
    row_med = sum(row)/len(row)
    med.append(row_med)
    row_disp = sum([(i-row_med)**2 for i in row])/len(row)
    disp.append(row_disp)
sigma_theta = math.sqrt(abs(2*(2*m-2)/(m*(m-4))))
tmp_disp = [i for i in disp]
print(tmp_disp)
F_uv = []
for i in range(len(tmp_disp)):
    for j in range(len(tmp_disp)):
        if (i != j) and (j < i):
            if (tmp_disp[i] >= tmp_disp[j]):
                F_uv.append(tmp_disp[i]/tmp_disp[j])
            else:
                F_uv.append(tmp_disp[j]/tmp_disp[i])

print(F_uv)
Theta_uv = [((m-2)/m)*i for i in F_uv]
R_uv = [abs(i-1)/sigma_theta for i in Theta_uv]
romanovsky_criteria_table = [[0, 2, 6, 8, 10, 12, 15, 20],
                                [0.99, 1.72, 2.16, 2.43, 2.62, 2.75, 2.90, 3.08],
                                [0.98, 1.72, 2.13, 2.37, 2.54, 2.66, 2.80, 2.96],
                                [0.95, 1.71, 2.10, 2.27, 2.41, 2.52, 2.64, 2.78],
                                [0.90, 1.69, 2.00, 2.17, 2.29, 2.39, 2.49, 2.62]]

index = 0
for i in romanovsky_criteria_table[0]:
    if i >= m :
        index = romanovsky_criteria_table[0].index(i)
        break
R_crit = romanovsky_criteria_table[1][index]
print(R_crit)
print(R_uv)
for i in R_uv:
    if i > R_crit:
        m += 1
        main(m)

print("Матриця планування : \n x1    x2    y1    y2    y3    y4    y5    ...")
for i in range(x_asis.shape[0]):
    for j in range(x_asis.shape[1]):
        print("{: ^5.1f}".format(x_asis.tolist()[i][j]), end = " ")
    print("\n")

print("Нормована матриця планування : \n x1    x2    y1    y2    y3    y4    y5    ...")
for i in range(x_norm.shape[0]):
    for j in range(x_norm.shape[1]):
        print("{: ^5.1f}".format(x_norm.tolist()[i][j]), end = " ")
    print("\n")

x_col1 = [ i[0] for i in x_norm.tolist()]
x_col2 = [ i[1] for i in x_norm.tolist()]
mx1 = sum(x_col1)/len(x_col1)

```

```

mx2 = sum(x_col2)/len(x_col2)
rows_med = []
for i in range(len(x_asis.tolist())):
    row = x_asis.tolist()[i][2:]
    row_med = sum(row)/len(row)
    rows_med.append(row_med)
my = sum(rows_med)/len(rows_med)
a1 = sum([ i[0]**2 for i in x_norm.tolist()])/len([ i[0]**2 for i in x_norm.tolist()])
a2 = sum([x_col1[i]*x_col2[i] for i in range(len(x_col1))])/len([x_col1[i]*x_col2[i] for i in
range(len(x_col1))])
a3 = sum([ i[1]**2 for i in x_norm.tolist()])/len([ i[1]**2 for i in x_norm.tolist()])
a11 = sum([x_col1[i]*rows_med[i] for i in range(len(x_col1))])/len([x_col1[i]*rows_med[i] for i in
range(len(x_col1))])
a22 = sum([x_col2[i]*rows_med[i] for i in range(len(x_col2))])/len([x_col2[i]*rows_med[i] for i in
range(len(x_col2))])
b0 = np.linalg.det(np.array([[my,mx1,mx2],[a11,a1,a2],[a22,a2,a3]]))\
    /np.linalg.det(np.array([[1,mx1,mx2],[mx1,a1,a2],[mx2,a2,a3]]))
b1 = np.linalg.det(np.array([[1,my,mx2],[mx1,a11,a2],[mx2,a22,a3]]))\
    /np.linalg.det(np.array([[1,mx1,mx2],[mx1,a1,a2],[mx2,a2,a3]]))
b2 = np.linalg.det(np.array([[1,mx1,my],[mx1,a1,a11],[mx2,a2,a22]]))\
    /np.linalg.det(np.array([[1,mx1,mx2],[mx1,a1,a2],[mx2,a2,a3]]))

print("\nРівняння регресії для нормованих факторів: y = "+str(round(b0,4))+ " + "+str(round(b1,4))+ "*x1 +
"+str(round(b2,4))+ "*x2")
print("Перевірка (теоретичні) : "+str(round(b0-b1-b2,5))+ " "+str(round(b0-b1+b2,5))+ " "+str(round(b0+b1-
b2,5)))
print("Перевірка (практичні) : "+str(rows_med[0])+ " "+str(rows_med[1])+ " "+str(rows_med[2]))

x_col1 = [ i[0] for i in x_asis.tolist()]
x_col2 = [ i[1] for i in x_asis.tolist()]

dx1 = abs(max(x_col1)-min(x_col1))/2
dx2 = abs(max(x_col2)-min(x_col2))/2
x10 = (max(x_col1)+min(x_col1))/2
x20 = (max(x_col2)+min(x_col2))/2
a0 = (b0 - (b1*x10)/(dx1) - (b2*x20)/(dx2))
a1 = (b1/dx1)
a2 = (b2/dx2)
print("\n\nРівняння регресії для натуралізованих факторів: y = "+str(round(a0,4))+ " + "+str(round(a1,4))
+ "*x1 + "+str(round(a2,4))+ "*x2")
print("Перевірка (теоретичні) : "+str(round(a0+a1*x_col1[0]+a2*x_col2[0],5))+ "
"+str(round(a0+a1*x_col1[1]+a2*x_col2[1],5))+ " "+str(round(a0+a1*x_col1[2]+a2*x_col2[2],5)))
print("Перевірка (практичні) : "+str(rows_med[0])+ " "+str(rows_med[1])+ " "+str(rows_med[2]))

main(m)

```

Результат виконання

```
Terminal -
File Edit View Terminal Tabs Help
Матриця планування :
x1    x2    y1    y2    y3    y4    y5    ...
-5.0  -15.0  205.0  164.0  185.0  185.0  143.0
-5.0   35.0  178.0  145.0  165.0  168.0  210.0
15.0  -15.0  178.0  203.0  190.0  170.0  203.0
Нормована матриця планування :
x1    x2    y1    y2    y3    y4    y5    ...
-1.0  -1.0  205.0  164.0  185.0  185.0  143.0
-1.0   1.0  178.0  145.0  165.0  168.0  210.0
 1.0  -1.0  178.0  203.0  190.0  170.0  203.0

Рівняння регресії для нормованих факторів:  y = 181.0 + 6.2*x1 + -1.6*x2
Перевірка (теоретичні) :  176.4  173.2  188.8
Перевірка (практичні ) :  176.4  173.2  188.8

Рівняння регресії для натуралізованих факторів:  y = 178.54 + 0.62*x1 + -0.064*x2
Перевірка (теоретичні) :  176.4  173.2  188.8
Перевірка (практичні ) :  176.4  173.2  188.8

-----
(program exited with code: 0)
```

Висновки

Підчас виконання лабораторної роботи було реалізовано завдання . Отримані результати збігаються , отже, експеримент було поставлено правильно.