

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет України

«Київський Політехнічний Інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №2

з дисципліни

«Методи оптимізації та планування експерименту»

**на тему: «ЗАГАЛЬНІ ПРИНЦИПИ ОРГАНІЗАЦІЇ ЕКСПЕРИМЕНТІВ З
ДОВІЛЬНИМИ ЗНАЧЕННЯМИ ФАКТОРІВ»**

Виконав:

Студент 2-го курсу ФІОТ

групи ІО-92

Калашніков Ілля

Перевірив:

Регіда П. Г.

Київ – 2021

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

ВАРІАНТ

| | | | | |
|-----|-----|----|-----|-----|
| 210 | -25 | -5 | -70 | -10 |
|-----|-----|----|-----|-----|

```
import random
import numpy
import tkinter
root = tkinter.Tk()
variant = 210
x1_min = -25
x1_max = -5
x2_min = -70
x2_max = -10
y_min = (20 - variant) * 10
y_max = (30 - variant) * 10
xn = [[-1, -1], [-1, 1], [1, -1], [1, 1]]
x = [[-40, 30], [-40, 80], [20, 30], [20, 80]]
m = 5
k = 1
d = {5: 2, 6: 2, 7: 2.17, 8: 2.17, 9: 2.29, 10: 2.29, 11: 2.39, 12: 2.39, 13: 2.39, 14: 2.49, 15: 2.49, 16: 2.49, 17: 2.49, 18: 2.62, 19: 2.62, 20: 2.62}
def romanovski(m, d, y, y_mean):
    dispersion = []
    for i in range(len(y)):
        dispersion.append(0)
        for j in range(m):
            dispersion[i] += (y[i][j] - y_mean[i]) ** 2
        dispersion[i] /= m
    main_deviation = ((2 * (2 * m - 2)) / (m * (m - 4))) ** 0.5
    f = []
    for i in range(3):
        if dispersion[i - 1] >= dispersion[i]:
            f.append(dispersion[i - 1] / dispersion[i])
        else:
            f.append(dispersion[i] / dispersion[i - 1])
    teta = []
    for i in range(3):
        teta.append((m - 2) / m * f[i])
    # print(teta)
    r = []
    for i in range(3):
        r.append(abs(teta[i] - 1) / main_deviation)
    # print(r)
    if (r[0] < d[m]) & (r[1] < d[m]) & (r[2] < d[m]):
        return True
    return False
def normalized_multiplier(x, y_mean):
    mx1 = (x[0][0] + x[1][0] + x[2][0]) / 3
    mx2 = (x[0][1] + x[1][1] + x[2][1]) / 3
    my = sum(y_mean) / 3
    # print(mx1, mx2, my)
    a1 = (x[0][0] ** 2 + x[1][0] ** 2 + x[2][0] ** 2) / 3
    a2 = (x[0][0] * x[0][1] + x[1][0] * x[1][1] + x[2][0] * x[2][1]) / 3
    a3 = (x[0][1] ** 2 + x[1][1] ** 2 + x[2][1] ** 2) / 3
```

```

# print(a1, a2, a3)
a11 = (x[0][0] * y_mean[0] + x[1][0] * y_mean[1] + x[2][0] * y_mean[2])
/3
a22 = (x[0][1] * y_mean[0] + x[1][1] * y_mean[1] + x[2][1] * y_mean[2])
/3
# print(a11, a22)
a = numpy.array([[1, mx1, mx2],
                 [mx1, a1, a2],
                 [mx2, a2, a3]])
c = numpy.array([my], [a11], [a22])
b = numpy.linalg.solve(a, c)
return b
def naturalized_multipliers(x1_min, x1_max, x2_min, x2_max, b):
    dx1 = (x1_max - x1_min) / 2
    dx2 = (x2_max - x2_min) / 2
    x10 = (x1_max + x1_min) / 2
    x20 = (x2_max + x2_min) / 2
    a0 = b[0][0] - b[1][0] * x10 / dx1 - b[2][0] * x20 / dx2
    a1 = b[1][0] / dx1
    a2 = b[2][0] / dx2
    return a0, a1, a2
while k != 0:
    y = [[random.randint(y_min, y_max) for i in range(m)] for j in range(3)]
    print(y)
    y_mean = []
    for i in range(3):
        y_mean.append(sum(y[i]) / m)
    print(y_mean)
    if romanovski(m, d, y, y_mean):
        k = 0
    else:
        k = 1
        m += 1
b = normalized_multiplier(xn, y_mean)
print(b)
a = naturalized_multipliers(x1_min, x1_max, x2_min, x2_max, b)
print(a)
tkinter.Label(text="y = b0 + b1 * xH1 + b2 * xH2").grid(columnspan=m + 2)
tkinter.Label(text="xH1").grid()
tkinter.Label(text="xH2").grid(row=1, column=1)
for i in range(m):
    tkinter.Label(text="yi" + str(i + 1)).grid(row=1, column=i + 2)
tkinter.Label(text="-1").grid()
tkinter.Label(text="-1").grid(row=2, column=1)
tkinter.Label(text="-1").grid()
tkinter.Label(text="1").grid(row=3, column=1)
tkinter.Label(text="1").grid()
tkinter.Label(text="-1").grid(row=4, column=1)
for j in range(3):
    for i in range(m):
        tkinter.Label(text="{0:.2f}".format(y[j][i])).grid(row=j + 2,
column=i + 2)
tkinter.Label(text="Нормалізоване рівняння регресії:").grid(columnspan=m + 2)
tkinter.Label(text="y = {0:.2f} + {1:.2f} * xH1 + {2:.2f} *
xH2".format(b[0][0], b[1][0], b[2][0])).grid(columnspan=m + 2)
tkinter.Label(text="Натуралізоване рівняння регресії:").grid(columnspan=m +
2)
tkinter.Label(text="y = {0:.2f} + {1:.2f} * x1 + {2:.2f}
*x2".format(*a)).grid(columnspan=m + 2)
tkinter.Label(text="Перевірка рівнянь:").grid(columnspan=m + 2)
for i in range(3):
    tkinter.Label(text="yc" + str(i + 1) + " = " +
"{0:.2f}".format(y_mean[i])).grid(columnspan=m + 2)
    tkinter.Label(text="b0 + b1 * xH1" + str(i + 1) + " + b2 * xH2" + str(i +

```

```
1) + " = " + "{0:.2f}".format(b[0][0] + b[1][0] * xn[i][0] + b[2][0] *  
xn[i][1])).grid(columnspan=m + 2)  
    tkinter.Label(text="a0 + a1 * x1" + str(i + 1) + " + a2 * x2" + str(i +  
1) + " = " + "{0:.2f}".format(a[0] + a[1] * x[i][0] + a[2] *  
x[i][1])).grid(columnspan=m + 2)  
root.mainloop()
```