

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»

**Загальні принципи організації експериментів з
довільними значеннями факторів**

ВИКОНАВ: Ралло Ігор

Студент 2 курсу

ФІОТ гр. ІО-92

ПЕРЕВІРИВ:

асистент

Регіда П.Г.

Київ 2021 р.

Мета:

Вивчити основні поняття, визначення, принципи теорії планування експерименту, на основі яких вивчити побудову формалізованих алгоритмів проведення експерименту і отримання формалізованої моделі об'єкта. Закріпити отримані знання практичним їх використанням при написанні програми, що реалізує завдання на лабораторну роботу.

Варіант завдання:

216	-10	50	25	65
-----	-----	----	----	----

Лістинг програми:

```
import random
import numpy

def Ruv(m, trust_p, y, mean):
    dispersion = []
    for i in range(len(y)):
        dispersion.append(0)
        for j in range(m):
            dispersion[i] += (y[i][j] - mean[i]) ** 2
        dispersion[i] /= m

    main_deviation = ((2 * (2 * m - 2)) / (m * (m - 4))) ** 0.5
    fuvs = [dispersion[i - 1] / dispersion[i] if dispersion[i - 1] >=
dispersion[i] else dispersion[i] / dispersion[i - 1] for i in range(3)]
    teta = [(m - 2) / m * fuvs[i] for i in range(3)]
    ruvs = [abs(teta[i] - 1) / main_deviation for i in range(3)]

    return True if (ruvs[0] < trust_p[m]) & (ruvs[1] < trust_p[m]) &
(ruvs[2] < trust_p[m]) else False

def get_normalized_coefficients(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

    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

    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
```

```
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 get_naturalized_coefficients(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
```

```
x1_min, x1_max = -10, 50
x2_min, x2_max = 25, 65
y_min, y_max = (20 - 225) * 10, (30 - 225) * 10
m = 5
```

```
xn = [[-1, -1],
       [-1, 1],
       [1, -1],
       [1, 1]]
```

```
x = [[x1_min, x2_min],
      [x1_min, x2_max],
      [x1_max, x2_min],
      [x1_max, x2_max]]
```

```
trust_p = {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}
```

```
while True:
    y = [[random.randint(y_min, y_max) for i in range(m)] for j in
```

```

range(3)]
    y_mean = [sum(y[i]) / m for i in range(3)]
    if Ruv(m, trust_p, y, y_mean): break
    else: m += 1

b = get_normalized_coefficients(xn, y_mean)
a = get_naturalized_coefficients(x1_min, x1_max, x2_min, x2_max, b)

table_values = ["-X1", "-X2"]
for i in range(m): table_values.append("Y" + str(i + 1))
row_format = "|{: ^15}" * (len(table_values))
header_separator_format = "+{0: ^15s}" * (len(table_values))
separator_format = "+{0: ^15s}" * (len(table_values))

print("\n\tЛінійне рівняння регресії:  $y = b_0 + b_1 \cdot -x_1 + b_2 \cdot -x_2$ \n\n" +
header_separator_format.format("=") + "\n" +
    row_format.format(*table_values) + "\n" +
header_separator_format.format("=") + "+")

for i in range(3):
    print("|{0: ^15}|{1: ^15}".format(xn[i][0], xn[i][1]), end="")
    for j in range(m):
        print("|{: ^15.2f}".format(y[i][j]), end="")
    print("|")

print(separator_format.format("-") + "+\n\n\tНормалізоване рівняння
регресії: " +
    "y = {0:.2f} + {1:.2f} * -x1 + {2:.2f} * -x2".format(b[0][0],
b[1][0], b[2][0]) + "\n\tНатуралізоване рівняння регресії: " +
    "y = {0:.2f} + {1:.2f} * x1 + {2:.2f} * x2".format(*a) +
"\n\n\tПеревірка рівнянь:\n")

for i in range(3):
    print("Yc" + str(i + 1) + " = " + "{0:.2f}".format(y_mean[i]) +
        "\n\tb0 + b1 * -x1" + str(i + 1) + " + b2 * -x2" + str(i + 1) +
" = " +
        "{0:.2f}".format(b[0][0] + b[1][0] * xn[i][0] + b[2][0] *
xn[i][1]) +
        "\n\ta0 + 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]))

```

Результат виконання роботи:

```
Лінійне рівняння регресії: y = b0 + b1*x1 + b2*x2

=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+
|      ~X1      |      ~X2      |      Y1      |      Y2      |      Y3      |      Y4      |      Y5      |      Y6      |      Y7      |      Y8      |      Y9      |
+=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+=====+
|      -1      |      -1      |    -1973.00   |    -1981.00   |    -1952.00   |    -2044.00   |    -1967.00   |    -1988.00   |    -2031.00   |    -1956.00   |    -2018.00   |
|      -1      |       1      |    -1979.00   |    -1953.00   |    -2017.00   |    -2045.00   |    -1958.00   |    -2031.00   |    -2015.00   |    -2035.00   |    -1978.00   |
|       1      |      -1      |    -2030.00   |    -2046.00   |    -1965.00   |    -1984.00   |    -1990.00   |    -1955.00   |    -1989.00   |    -1979.00   |    -2035.00   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

Нормалізоване рівняння регресії: y = -1999.11 + -3.50 * ~x1 + -5.61 * ~x2
Натуралізоване рівняння регресії: y = -1984.15 + -0.12 * x1 + -0.28 *x2

Перевірка рівнянь:

Yc1 = -1990.00
b0 + b1 * ~x11 + b2 * ~x21 = -1990.00
a0 + a1 * x11 + a2 * x21 = -1990.00
Yc2 = -2001.22
b0 + b1 * ~x12 + b2 * ~x22 = -2001.22
a0 + a1 * x12 + a2 * x22 = -2001.22
Yc3 = -1997.00
b0 + b1 * ~x13 + b2 * ~x23 = -1997.00
a0 + a1 * x13 + a2 * x23 = -1997.00

Process finished with exit code 0
```