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 №6

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:

Провести трьохфакторний експеремент і отримати адекватну модель-рівняння регресії використовуючи рототабельний композиційний план.

Варіант завдання:

216	-10	50	25	65	-10	15	5,6+8,0*x1+4,8*x2+6,2*x3+5,9*x1*x1+1,0*x2*x2+8,7*x3*x3+2,0*x1*x2+0,8*x1*x3+1,0*x2*x3+3,0*x1*x2*x3

Лістинг програми:

```
from math import sqrt
from time import process_time
from scipy.stats import f, t
from functools import partial
from random import randrange
from numpy.linalg import solve

x1, x2, x3 = [-10, 50], [25, 65], [-10, 15]
m, N, l = 2, 15, 1.73  # кількість повторень кожної комбінації &
кількість повторення дослідів

x_avg = [(max(x1) + max(x2) + max(x3)) / 3, (min(x1) + min(x2) +
min(x3)) / 3]  # Xcp(max) & Xcp(min)
xo = [(min(x1) + max(x1)) / 2, (min(x2) + max(x2)) / 2, (min(x3) +
max(x3)) / 2]  # Xoi
delta_x = [max(x1) - xo[0], max(x1) - xo[1], max(x1) - xo[2]]  #
delta Xi

y_range = [200 + int(max(x_avg)), 200 + int(min(x_avg))]  # Yi(max)
& Yi(min)

xn = [[-1, -1, -1, -1, +1, +1, +1, +1, -1.73, 1.73, 0, 0, 0, 0, 0],
# нормовані значення факторів
      [-1, -1, +1, +1, -1, -1, +1, +1, 0, 0, -1.73, 1.73, 0, 0, 0],
      [-1, +1, -1, +1, -1, +1, -1, +1, 0, 0, 0, 0, -1.73, 1.73, 0]]

xx = [[int(x * y) for x, y in zip(xn[0], xn[1])],  # нормовані
значення факторів для ефекту взаємодії
      [int(x * y) for x, y in zip(xn[0], xn[2])],
      [int(x * y) for x, y in zip(xn[1], xn[2])]]

xxx = [int(x * y * z) for x, y, z in zip(xn[0], xn[1], xn[2])]
```

```
x_xn = [[round(xn[j][i] ** 2, 3) for i in range(N)] for j in
range(3)]  # нормовані знач. факторів для квад. членів

x = [[min(x1), min(x1), min(x1), min(x1), max(x1), max(x1), max(x1),
max(x1), round(-l * delta_x[0] + xo[0], 3),
      round(l * delta_x[0] + xo[0], 3), xo[0], xo[0], xo[0], xo[0],
xo[0]],  # натуральні значення факторів
     [min(x2), min(x2), max(x2), max(x2), min(x2), min(x2), max(x2),
max(x2), xo[1], xo[1],
      round(-l * delta_x[1] + xo[1], 3), round(l * delta_x[1] +
xo[1], 3), xo[1], xo[1], xo[1]],
     [min(x3), max(x3), min(x3), max(x3), max(x3), min(x3), max(x3),
min(x3), xo[2], xo[2], xo[2], xo[2],
      round(-l * delta_x[2] + xo[2], 3), round(l * delta_x[2] +
xo[2], 3), xo[2]]]

xx2 = [[round(x * y, 3) for x, y in zip(x[0], x[1])],  # натуральні
значення факторів для ефекту взаємодії
       [round(x * y, 3) for x, y in zip(x[0], x[2])],
       [round(x * y, 3) for x, y in zip(x[1], x[2])]]

xxx2 = [round(x * y * z, 3) for x, y, z in zip(x[0], x[1], x[2])]

x_x = [[round(x[j][i] ** 2, 3) for i in range(N)] for j in range(3)]
# натуральні значення факторів для квадрат. членів

while True:
    start_time = process_time()
    # формування Y
    y = [[round(5.6 + 8.0 * x[0][j] + 4.8 * x[1][j] + 6.2 * x[2][j]
+ 5.9 * x[0][j] * x[0][j] + 1.0 * x[1][j] * x[1][j] +
                8.7 * x[2][j] * x[2][j] + 2.0 * x[0][j] * x[1][j] +
0.8 * x[0][j] * x[2][j] + 1.0 * x[1][j] * x[2][j] +
                3.0 * x[0][j] * x[1][j] * x[2][j] + randrange(0, 10)
- 5, 2) for i in range(m)] for j in range(N)]
    arr_avg = lambda arr: round(sum(arr) / len(arr), 4)
    y_avg = list(map(arr_avg, y))  # середнє значення Y

    dispersions = [sum([((y[i][j] - y_avg[i]) ** 2) / m for j in
range(m)]) for i in range(N)]  # дисперсії по рядках
    x_matrix = x + xx2 + [xxx2] + x_x  # повна матриця з
натуральними значеннями факторів
    norm_matrix = xn + xx + [xxx] + x_xn  # повна матриця з
нормованими значеннями факторів

    mx = list(map(arr_avg, x_matrix))  # середні значення x по
```

```
колонкам
    my = sum(y_avg) / N  # середнє значення Y_avg

    # ======================================== Форматування таблиці
==================================================

    table_factors_1 = ["X1", "X2", "X3"]
    table_factors_2 = ["X1X2", "X1X3", "X2X3", "X1X2X3", "X1^2",
"X2^2", "X3^2"]
    table_y = ["Y{}".format(i + 1) for i in range(m)]
    other = ["#", "Y"]

    header_format_norm = "+{0:=^3}" + "+{0:=^8}" *
(len(table_factors_1)) + "+{0:=^8s}" * (len(table_factors_2))
    header_format = "+{0:=^3}" + "+{0:=^8}" * (len(table_factors_1))
+ "+{0:=^10s}" * (len(table_factors_2)) + "+{0:=^10s}" *
(len(table_y)) + "+{0:=^10s}"
    row_format_norm = "|{:^3}" + "|{:^8}" * (len(table_factors_1)) +
"|{:^8}" * (len(table_factors_2))
    row_format = "|{:^3}" + "|{:^8}" * (len(table_factors_1)) +
"|{:^10}" * (len(table_factors_2)) + "|{:^10}" * (len(table_y)) +
"|{:^10}"
    separator_format_norm = "+{0:-^3s}" + "+{0:-^8s}" *
(len(table_factors_1)) + "+{0:-^8s}" * (len(table_factors_2))
    separator_format = "+{0:-^3s}" + "+{0:-^8s}" *
(len(table_factors_1)) + "+{0:-^10s}" * (len(table_factors_2)) +
"+{0:-^10s}" * (len(table_y)) + "+{0:-^10s}"
    my_sep_norm = "|{:^93s}|\n"
    my_sep = "|{:^140s}|\n" if m == 2 else "|{:^151s}|\n"
    # ======================================== Нормальні значення
==================================================
    print(header_format_norm.format("=") + "+\n" +
my_sep_norm.format("Матриця ПФЕ (нормальні значення факторів)") +
          header_format_norm.format("=") + "+\n" +
row_format_norm.format(other[0], *table_factors_1, *table_factors_2)
          + "|\n" + header_format_norm.format("=") + "+")

    for i in range(N):
        print("|{:^3}|".format(i + 1), end="")
        for j in range(3): print("{:^+8}|".format(xn[j][i]), end="")
        for j in range(3): print("{:^+8}|".format(xx[j][i]), end="")
        print("{:^+8}|".format(xxx[i]), end="")
        for j in range(3): print("{:^+8}|".format(x_xn[j][i]),
end="")
        print()
```

```
    print(separator_format_norm.format("-") + "+\n\n")

    # ========================================= Натуральні значення
=================================================
    print(header_format.format("=") + "+\n" + my_sep.format("Матриця
ПФЕ (натуральні значення факторів)") +
          header_format.format("=") + "+\n" +
row_format.format(other[0], *table_factors_1, *table_factors_2,
*table_y,

other[1]) + "|\n" + header_format.format("=") + "+")

    for i in range(N):
        print("|{:^3}|".format(i + 1), end="")
        for j in range(3): print("{:^ 8}|".format(x[j][i]), end="")
        for j in range(3): print("{:^ 10}|".format(xx2[j][i]),
end="")
        print("{:^ 10}|".format(xxx2[i]), end="")
        for j in range(3): print("{:^ 10}|".format(x_x[j][i]),
end="")
        for j in range(m): print("{:^ 10}|".format(y[i][j]), end="")
        print("{:^10.2f}|".format(y_avg[i]))


    def a(first, second): return sum([x_matrix[first - 1][j] *
x_matrix[second - 1][j] / N for j in range(N)])
    def find_a(num): return sum([y_avg[j] * x_matrix[num - 1][j] / N
for j in range(N)])
    def check(b_lst, k):
        return b_lst[0] + b_lst[1] * x_matrix[0][k] + b_lst[2] *
x_matrix[1][k] + b_lst[3] * x_matrix[2][k] + \
               b_lst[4] * x_matrix[3][k] + b_lst[5] * x_matrix[4][k]
+ b_lst[6] * x_matrix[5][k] + \
               b_lst[7] * x_matrix[6][k] + b_lst[8] * x_matrix[7][k]
+ b_lst[9] * x_matrix[8][k] + \
               b_lst[10] * x_matrix[9][k]

    unknown = [[1, mx[0], mx[1], mx[2], mx[3], mx[4], mx[5], mx[6],
mx[7], mx[8], mx[9]],  # ліва частина рівнянь з невідомими для
пошуку коефіцієнтів b (приклад в методі)
               [mx[0], a(1, 1), a(1, 2), a(1, 3), a(1, 4), a(1, 5),
a(1, 6), a(1, 7), a(1, 8), a(1, 9), a(1, 10)],
               [mx[1], a(2, 1), a(2, 2), a(2, 3), a(2, 4), a(2, 5),
a(2, 6), a(2, 7), a(2, 8), a(2, 9), a(2, 10)],
               [mx[2], a(3, 1), a(3, 2), a(3, 3), a(3, 4), a(3, 5),
a(3, 6), a(3, 7), a(3, 8), a(3, 9), a(3, 10)],
```

```
                [mx[3], a(4, 1), a(4, 2), a(4, 3), a(4, 4), a(4, 5),
a(4, 6), a(4, 7), a(4, 8), a(4, 9), a(4, 10)],
                [mx[4], a(5, 1), a(5, 2), a(5, 3), a(5, 4), a(5, 5),
a(5, 6), a(5, 7), a(5, 8), a(5, 9), a(5, 10)],
                [mx[5], a(6, 1), a(6, 2), a(6, 3), a(6, 4), a(6, 5),
a(6, 6), a(6, 7), a(6, 8), a(6, 9), a(6, 10)],
                [mx[6], a(7, 1), a(7, 2), a(7, 3), a(7, 4), a(7, 5),
a(7, 6), a(7, 7), a(7, 8), a(7, 9), a(7, 10)],
                [mx[7], a(8, 1), a(8, 2), a(8, 3), a(8, 4), a(8, 5),
a(8, 6), a(8, 7), a(8, 8), a(8, 9), a(8, 10)],
                [mx[8], a(9, 1), a(9, 2), a(9, 3), a(9, 4), a(9, 5),
a(9, 6), a(9, 7), a(9, 8), a(9, 9), a(9, 10)],
                [mx[9], a(10, 1), a(10, 2), a(10, 3), a(10, 4), a(10,
5), a(10, 6), a(10, 7), a(10, 8), a(10, 9), a(10, 10)]]
    known = [my, find_a(1), find_a(2), find_a(3), find_a(4),
find_a(5), find_a(6), find_a(7), find_a(8), find_a(9), find_a(10)]
# знаходення відомих значень a1, a2, ...

    b = solve(unknown, known)
    print(separator_format.format("-") + f"+\n\n\tОтримане рівняння
регресії при m={m}:\n"
                                        f"ŷ = {b[0]:.3f} +
{b[1]:.3f}*X1 + {b[2]:.3f}*X2 + "
                                        f"{b[3]:.3f}*X3 +
{b[4]:.3f}*X1X2 + {b[5]:.3f}*X1X3 + "
                                        f"{b[6]:.3f}*X2X3 +
{b[7]:.3f}*X1X2X3 + {b[8]:.3f}*X11^2 + "
                                        f"{b[9]:.3f}*X22^2 +
{b[10]:.3f}*X33^2\n\n\tПеревірка:")
    for i in range(N): print("ŷ{} = {:.3f} ≈ {:.3f}".format((i + 1),
check(b, i), y_avg[i]))

    # ========================================= Критерій Кохрена
=========================================================
    def table_fisher(prob, n, m, d):
        x_vec = [i * 0.001 for i in range(int(10 / 0.001))]
        f3 = (m - 1) * n
        for i in x_vec:
            if abs(f.cdf(i, n - d, f3) - prob) < 0.0001:
                return i

    f1, f2 = m - 1, N
    f3 = f1 * f2
    fisher = table_fisher(0.95, N, m, 1)
    Gp = max(dispersions) / sum(dispersions)
    Gt = fisher / (fisher + (m - 1) - 2)
```

```
    print("\nОднорідність дисперсії (критерій Кохрена): ")
    print(f"Gp = {Gp}\nGt = {Gt}")
    if Gp < Gt:
        print("\nДисперсія однорідна (Gp < Gt)")

        D_beta = sum(dispersions) / (N * N * m)
        Sb = sqrt(abs(D_beta))
        beta = [sum([(y_avg[j] * norm_matrix[i][j]) / N for j in
range(N)]) for i in range(len(norm_matrix))]

        t_list = [abs(i) / Sb for i in beta]
        student = partial(t.ppf, q=1-0.025)
        d, T = 0, student(df=f3)
        print("\nt табличне = ", T)

        for i in range(len(t_list)):
            if t_list[i] < T:
                b[i] = 0
                print("\tt{} = {} => коефіцієнт незначимий, його
слід виключити з рів-ня регресії".format(i, t_list[i]))
            else:
                print("\tt{} = {} => коефіцієнт значимий".format(i,
t_list[i]))
                d += 1

        print("\nОтже, кіл-ть значимих коеф. d =", d, "\n\n\tРів-ня
регресії з урахуванням критерія Стьюдента:\nŷ = ", end="")
        print("{:.3f}".format(b[0]), end="") if b[0] != 0 else None
        for i in range(1, 10):
            print(" + {:.3f}*{}".format(b[i], (table_factors_1 +
table_factors_2)[i]), end="") if b[i] != 0 else None
        print("\n\n\tПеревірка при підстановці в спрощене рів-ня
регресії:")
        for i in range(N): print("y`{} = {:.3f} ≈ {:.3f}".format((i
+ 1), check(b, i), y_avg[i]))

        f4 = N - d
        fisher_sum = sum([(check(b, i) - y_avg[i]) ** 2 for i in
range(N)])
        D_ad = (m / f4) * fisher_sum

        fisher = partial(f.ppf, q=1-0.05)
        Fp = D_ad / sum(dispersions) / N
        Ft = fisher(dfn=f4, dfd=f3)
        print("\nКритерій Фішера:")
```

```
        if Fp > Ft:
            print("\tРівняння регресії неадекватне (Ft < Fp).")
            break
        else:
            print("\tРівняння регресії адекватне (Ft > Fp)!")
            print("\n\n--- Час виконання програми: %s секунд ---" %
(process_time() - start_time))
            break

    else:
        print("Дисперсія неоднорідна (Gp > Gt), збільшуємо m,
повторюємо операції")
        m += 1
```

Результат виконання роботи:

```
D:\Anaconda2019\python.exe "C:/Users/User/Desktop/КПІ курс 2 (3-4 сем.)/КПІ курс 2 (4 сем.)/МОПЕ/код/lab6.py"
+===+========+========+========+========+========+========+========+========+========+========+
|                        Матриця ПФЕ (нормальні значення факторів)                          |
+===+========+========+========+========+========+========+========+========+========+========+
| # |   X1   |   X2   |   X3   |  X1X2  |  X1X3  |  X2X3  | X1X2X3 |  X1^2  |  X2^2  |  X3^2  |
+===+========+========+========+========+========+========+========+========+========+========+
| 1 |   -1   |   -1   |   -1   |   +1   |   +1   |   +1   |   -1   |   +1   |   +1   |   +1   |
| 2 |   -1   |   -1   |   +1   |   +1   |   -1   |   -1   |   +1   |   +1   |   +1   |   +1   |
| 3 |   -1   |   +1   |   -1   |   -1   |   +1   |   -1   |   +1   |   +1   |   +1   |   +1   |
| 4 |   -1   |   +1   |   +1   |   -1   |   -1   |   +1   |   -1   |   +1   |   +1   |   +1   |
| 5 |   +1   |   -1   |   -1   |   -1   |   -1   |   +1   |   +1   |   +1   |   +1   |   +1   |
| 6 |   +1   |   -1   |   +1   |   -1   |   +1   |   -1   |   -1   |   +1   |   +1   |   +1   |
| 7 |   +1   |   +1   |   -1   |   +1   |   -1   |   -1   |   -1   |   +1   |   +1   |   +1   |
| 8 |   +1   |   +1   |   +1   |   +1   |   +1   |   +1   |   +1   |   +1   |   +1   |   +1   |
| 9 | -1.73  |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   | +2.993 |   +0   |   +0   |
|10 | +1.73  |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   | +2.993 |   +0   |   +0   |
|11 |   +0   | -1.73  |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   | +2.993 |   +0   |
|12 |   +0   | +1.73  |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   | +2.993 |   +0   |
|13 |   +0   |   +0   | -1.73  |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   | +2.993 |
|14 |   +0   |   +0   | +1.73  |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   | +2.993 |
|15 |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   |   +0   |
+---+--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+
```

Матриця ПФЕ (натуральні значення факторів)

#	X1	X2	X3	X1X2	X1X3	X2X3	X1X2X3	X1^2	X2^2	X3^2	Y1	Y2	Y
1	-10	25	-10	-250	100	-250	2500	100	625	100	8893.6	8896.6	8895.10
2	-10	25	15	-250	-150	375	-3750	100	625	225	-8187.9	-8180.9	-8184.40
3	-10	65	-10	-650	100	-650	6500	100	4225	100	23485.6	23491.6	23488.60
4	-10	65	15	-650	-150	975	-9750	100	4225	225	-22591.9	-22589.9	-22590.90
5	50	25	15	1250	750	375	18750	2500	625	225	77671.1	77676.1	77673.60
6	50	25	-10	1250	-500	-250	-12500	2500	625	100	-18942.4	-18940.4	-18941.40
7	50	65	15	3250	750	975	48750	2500	4225	225	176070.1	176067.1	176068.60
8	50	65	-10	3250	-500	-650	-32500	2500	4225	100	-71554.4	-71552.4	-71553.40
9	-31.9	45.0	2.5	-1435.5	-79.75	112.5	-3588.75	1017.61	2025.0	6.25	-5527.38	-5522.38	-5524.88
10	71.9	45.0	2.5	3235.5	179.75	112.5	8088.75	5169.61	2025.0	6.25	64381.92	64380.92	64381.42
11	20.0	36.35	2.5	727.0	50.0	90.875	1817.5	400.0	1321.323	6.25	11126.65	11126.65	11126.65
12	20.0	53.65	2.5	1073.0	50.0	134.125	2682.5	400.0	2878.322	6.25	16095.94	16094.94	16095.44
13	20.0	45.0	-79.675	900.0	-1593.5	-3585.375	-71707.5	400.0	2025.0	6348.106	-158678.54	-158679.54	-158679.04
14	20.0	45.0	84.675	900.0	1693.5	3810.375	76207.5	400.0	2025.0	7169.856	303258.0	303256.0	303257.00
15	20.0	45.0	2.5	900.0	50.0	112.5	2250.0	400.0	2025.0	6.25	13533.98	13537.98	13535.98

Отримане рівняння регресії при m=2:

$$\hat{y} = 9.617 + 8.039*X1 + 4.477*X2 + 6.164*X3 + 1.999*X1X2 + 0.793*X1X3 + 1.001*X2X3 + 3.000*X1X2X3 + 5.900*X11^2 + 1.004*X22^2 + 8.701*X33^2$$

Перевірка:

$\hat{y}1 = 8896.589 \approx 8895.100$

$\hat{y}2 = -8185.089 \approx -8184.400$

$\hat{y}3 = 23489.917 \approx 23488.600$

$\hat{y}4 = -22591.763 \approx -22590.900$

$\hat{y}5 = 77672.277 \approx 77673.600$

$\hat{y}6 = -18940.544 \approx -18941.400$

$\hat{y}7 = 176067.101 \approx 176068.600$

$\hat{y}8 = -71552.720 \approx -71553.400$

$\hat{y}9 = -5525.610 \approx -5524.880$

$\hat{y}10 = 64382.160 \approx 64381.420$

$\hat{y}11 = 11125.851 \approx 11126.650$

$\hat{y}12 = 16096.280 \approx 16095.440$

$\hat{y}13 = -158679.703 \approx -158679.040$

$\hat{y}14 = 303257.663 \approx 303257.000$

$\hat{y}15 = 13535.956 \approx 13535.980$

Однорідність дисперсії (критерій Кохрена):

Gp = 0.26063829787234044

Gt = 1.702247191011236

Дисперсія однорідна (Gp < Gt)

t табличне = 2.131449545559323

t0 = 58291.16346610504 => коефіцієнт значимий

t1 = 11256.107726775936 => коефіцієнт значимий

t2 = 80812.54553158405 => коефіцієнт значимий

t3 = 9405.73662888319 => коефіцієнт значимий

t4 = 57981.96421118582 => коефіцієнт значимий

t5 = 37132.60885506132 => коефіцієнт значимий

t6 = 25168.12239845932 => коефіцієнт значимий

t7 = 70345.6981999942 => коефіцієнт значимий

```
    t8 = 50814.31171514851 => коефіцієнт значимий
    t9 = 123270.85365709182 => коефіцієнт значимий

Отже, кіл-ть значимих коеф. d = 10

    Рів-ня регресії з урахуванням критерія Стьюдента:
ŷ = 9.617 + 8.039*X2 + 4.477*X3 + 6.164*X1X2 + 1.999*X1X3 + 0.793*X2X3 + 1.001*X1X2X3 + 3.000*X1^2 + 5.900*X2^2 + 1.004*X3^2

    Перевірка при підстановці в спрощене рів-ня регресії:
y`1 = 8896.589 ≈ 8895.100
y`2 = -8185.089 ≈ -8184.400
y`3 = 23489.917 ≈ 23488.600
y`4 = -22591.763 ≈ -22590.900
y`5 = 77672.277 ≈ 77673.600
y`6 = -18940.544 ≈ -18941.400
y`7 = 176067.101 ≈ 176068.600
y`8 = -71552.720 ≈ -71553.400
y`9 = -5525.610 ≈ -5524.880
y`10 = 64382.160 ≈ 64381.420
y`11 = 11125.851 ≈ 11126.650
y`12 = 16096.280 ≈ 16095.440
y`13 = -158679.703 ≈ -158679.040
y`14 = 303257.663 ≈ 303257.000
y`15 = 13535.956 ≈ 13535.980

Критерій Фішера:
    Рівняння регресії адекватне (Ft > Fp)!
```