МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський Політехнічний Інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №6

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

«Проведення трьохфакторного експерименту при використанні рівняння регресії з квадратичними членами (рототабельний композиційний план)»

Виконав:
Студент 2-го курсу ФІОТ
групи ІО-82
Шендріков Є.О.
Залікова книжка № 8227
Номер у списку групи: 25

Перевірив:
Регіда П. Г.

Київ – 2020

ВАРІАНТ

225	-25	-5	15	50	-25	-15	3,5+6,6*x1+5,3*x2+5,0*x3+5,1*x1*x1+0,1*x2*x2+7,2*x3*x3+1,4*x1*x2+0,7*x1*x3+4,2*x2*x3+7,7*x1*x2*x3

ФРАГМЕНТ КОДУ

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

x1, x2, x3 = [-25, -5], [15, 50], [-25, -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(-1 * delta_x[0] + xo[0], 3),
      round(1 * 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(-1 * delta_x[1] + xo[1], 3), round(1 * 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(-1 * delta_x[2] + xo[2], 3), round(1 * 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:
    # формування Y
    y = [[round(3.5 + 6.6 * x[0][j] + 5.3 * x[1][j] + 5 * x[2][j] + 5.1 *
x[0][j] * x[0][j] + 0.1 * x[1][j] * x[1][j] +
```

```
                7.2 * x[2][j] * x[2][j] + 1.4 * x[0][j] * x[1][j] + 0.7 *
x[0][j] * x[2][j] + 4.2 * x[1][j] * x[2][j] +
                7.7 * 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]],
               [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)]

    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)!")
            break

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

РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ

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	-25	15	-25	-375	625	-375	9375	625	225	625	78028.0	78031.0	78029.50
2	-25	15	-15	-375	375	-225	5625	625	225	225	46776.0	46777.0	46776.50
3	-25	50	-25	-1250	625	-1250	31250	625	2500	625	241973.5	241980.5	241977.00
4	-25	50	-15	-1250	375	-750	18750	625	2500	225	144820.5	144824.5	144822.50
5	-5	15	-15	-75	75	-225	1125	25	225	225	9413.0	9405.0	9409.00
6	-5	15	-25	-75	125	-375	1875	25	225	625	17423.0	17418.0	17420.50
7	-5	50	-15	-250	75	-750	3750	25	2500	225	27582.5	27582.5	27582.50
8	-5	50	-25	-250	125	-1250	6250	25	2500	625	47603.5	47602.5	47603.00
9	-32.3	32.5	-20.0	-1049.75	646.0	-650.0	20995.0	1043.29	1056.25	400.0	166078.02	166084.02	166081.02
10	2.3	32.5	-20.0	74.75	-46.0	-650.0	-1495.0	5.29	1056.25	400.0	-11063.52	-11067.52	-11065.52
11	-15.0	97.375	-20.0	-1460.625	300.0	-1947.5	29212.5	225.0	9481.891	400.0	220216.15	220219.15	220217.65
12	-15.0	-32.375	-20.0	485.625	300.0	647.5	-9712.5	225.0	1048.141	400.0	-67414.65	-67416.65	-67415.65
13	-15.0	32.5	-45.95	-487.5	689.25	-1493.375	22400.625	225.0	1056.25	2111.403	182314.84	182317.84	182316.34
14	-15.0	32.5	5.95	-487.5	-89.25	193.375	-2900.625	225.0	1056.25	35.403	-20657.09	-20651.09	-20654.09
15	-15.0	32.5	-20.0	-487.5	300.0	-650.0	9750.0	225.0	1056.25	400.0	75979.38	75977.38	75978.38

Отримане рівняння регресії при m=2:

$$\hat{y} = 8.975 + 7.165*X1 + 5.020*X2 + 5.405*X3 + 1.380*X1X2 + 0.723*X1X3 + 4.183*X2X3 + 7.699*X1X2X3 + 5.104*X11^2 + 0.100*X22^2 + 7.203*X33^2$$

Перевірка:

$\hat{y}1 = 78027.323 \approx 78029.500$

$\hat{y}2 = 46775.977 \approx 46776.500$

$\hat{y}3 = 241976.040 \approx 241977.000$

$\hat{y}4 = 144823.188 \approx 144822.500$

$\hat{y}5 = 9408.798 \approx 9409.000$

$\hat{y}6 = 17418.666 \approx 17420.500$

$\hat{y}7 = 27583.527 \approx 27582.500$

$\hat{y}8 = 47602.372 \approx 47603.000$

$\hat{y}9 = 166082.176 \approx 166081.020$

$\hat{y}10 = -11065.138 \approx -11065.520$

$\hat{y}11 = 220217.162 \approx 220217.650$

$\hat{y}12 = -67414.827 \approx -67415.650$

$\hat{y}13 = 182317.061 \approx 182316.340$

$\hat{y}14 = -20654.641 \approx -20654.090$

$\hat{y}15 = 75980.944 \approx 75978.380$

Однорідність дисперсії (критерій Кохрена):

Gp = 0.22939068100358423

Gt = 1.702247191011236

Дисперсія однорідна (Gp < Gt)

```
t табличне =  2.131449545559323
    t0 = 121251.93432319495 => коефіцієнт значимий
    t1 = 31708.732832748687 => коефіцієнт значимий
    t2 = 76456.50535694358 => коефіцієнт значимий
    t3 = 36175.98617600426 => коефіцієнт значимий
    t4 = 26490.44849046173 => коефіцієнт значимий
    t5 = 9125.80579247702 => коефіцієнт значимий
    t6 = 13192.85785953112 => коефіцієнт значимий
    t7 = 182470.71609814066 => коефіцієнт значимий
    t8 = 181348.88071163805 => коефіцієнт значимий
    t9 = 185839.3898524828 => коефіцієнт значимий

Отже, кіл-ть значимих коеф. d = 10

    Рів-ня регресії з урахуванням критерія Стьюдента:
ŷ = 8.975 + 7.165*X2 + 5.020*X3 + 5.405*X1X2 + 1.380*X1X3 + 0.723*X2X3 + 4.183*X1X2X3 + 7.699*X1^2 + 5.104*X2^2 + 0.100*X3^2
```

```
    Перевірка при підстановці в спрощене рів-ня регресії:
y`1 = 78027.323 ≈ 78029.500
y`2 = 46775.977 ≈ 46776.500
y`3 = 241976.040 ≈ 241977.000
y`4 = 144823.188 ≈ 144822.500
y`5 = 9408.798 ≈ 9409.000
y`6 = 17418.666 ≈ 17420.500
y`7 = 27583.527 ≈ 27582.500
y`8 = 47602.372 ≈ 47603.000
y`9 = 166082.176 ≈ 166081.020
y`10 = -11065.138 ≈ -11065.520
y`11 = 220217.162 ≈ 220217.650
y`12 = -67414.827 ≈ -67415.650
y`13 = 182317.061 ≈ 182316.340
y`14 = -20654.641 ≈ -20654.090
y`15 = 75980.944 ≈ 75978.380

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