

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 №4

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	-20	60	10	15
-----	----	----	-----	----	----	----

Лістинг програми:

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

x1, x2, x3 = [10, 50], [-20, 60], [10, 15]
m, N = 3, 4 # 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)
y_range = [200 + max(x_avg), 200 + min(x_avg)] # Yi(max) & Yi(min)

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

xx = [[x * y for x, y in zip(xn[1], xn[2])], # нормовані значення
      факторів для ефекту взаємодії
      [x * y for x, y in zip(xn[1], xn[3])],
      [x * y for x, y in zip(xn[2], xn[3])]]

xxx = [x * y * z for x, y, z in zip(xn[1], xn[2], xn[3])]

x = [[min(x1), min(x1), max(x1), max(x1), min(x1), min(x1), max(x1),
max(x1)],
      [min(x2), max(x2), min(x2), max(x2), min(x2), max(x2), min(x2),
max(x2)],
```

```

    [min(x3), max(x3), max(x3), min(x3), max(x3), min(x3), min(x3),
max(x3)]]

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

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

while True:
    flag = True # прапорець на випадок якщо при N=4 рівняння регресії
буде неадекватне, тоді переходимо до N=8

    def cochrans(f1, f2, q=0.05):
        q1 = q / f1
        fisher = f.ppf(q=1-q1, dfn=f2, dfd=(f1-1)*f2)
        return fisher / (fisher + f1 - 1)

    while True: # цикл для виконання алгоритму з N=4 (доходить до
перевірки на однорідність дисперсії,
                # якщо однорідна - виходимо з цього циклу, інакше
збільшуємо m)
        def comb(arr): # формування усіх комбінацій для 3 елементів
            return [1, *arr, arr[0] * arr[1], arr[0] * arr[2], arr[1] *
arr[2], arr[0] * arr[1] * arr[2]]

        def get_b_nat(x, y_avg): # функція для знаходження b (натуральне)
при N=4, для N=8 використовується
                                # спрощена версія, тут це не виходить,
бо матриця буде не квадратна
            n = len(y_avg)
            xnats = transpose(x)
            a = [[sum([comb(xnats[j])[i] * comb(xnats[j])[k] for j in
range(n)))] for i in range(n)] for k in range(n)]
            c = [sum([comb(xnats[j])[i] * y_avg[j] for i in range(n)]) for
j in range(n)]
            return solve(array(a), array(c))

        print("-" * 120, "\nПочаток виконання алгоритму з N =
{}".format(N))
        print("\nПоточне m = {}".format(m))

        y = [[round(uniform(min(y_range), max(y_range)), 4) for i in
range(m)] for j in range(N)] # формування Y
        y_avg = list(map(lambda arr: round(sum(arr) / len(arr), 4), y)) #
середнє значення Y

```

```

        solve_b_norm = [xn[0], xn[1], xn[2], xn[3]] # для нормального
рівняння
        b_norm = [sum([(solve_b_norm[k][i] * y_avg[i]) / N) for i in
range(N)]) for k in range(N)] # b до норм. рівня
        b_nat = get_b_nat(x, y_avg) # b для натурального рівняння

        dispersions = [sum([(y[i][j] - y_avg[i]) ** 2) / m for j in
range(m)]) for i in range(N)] # дисперс. по рядках

# ===== Форматування таблиці
=====
        table_factors_1 = ["#", "X0", "X1", "X2", "X3"]
        table_y = ["Y1", "Y2", "Y3", "Y", "S^2"]

        header_format = "+{0: ^4}" * (len(table_factors_1)) + "+{0: ^10s}"
* (len(table_y))
        row_format = "|{: ^4}" * (len(table_factors_1)) + "|{: ^10}" *
(len(table_y))
        separator_format = "+{0: ^4s}" * (len(table_factors_1)) + "+{0: ^10s}" * (len(table_y))

# ===== Нормальні значення
=====
        print(header_format.format("=") + "+\n" +
"|{: ^79s}|\n".format("Матриця ПФЕ (нормальні значення факторів)") +
        header_format.format("=") + "+\n" +
row_format.format(*table_factors_1, *table_y) + "|\n" +
        header_format.format("=") + "+")

        for i in range(N):
            print("|{: ^4}|".format(i + 1), end="")
            for j in range(4): print("{: ^4}|".format(xn[j][i]), end="")
            for j in range(m): print("{: ^10.4f}|".format(y[i][j]), end="")
            print("{0: ^10.4f}|{1: ^10.4f}|".format(y_avg[i],
dispersions[i]))

        print(separator_format.format("-") + f"+\n\n\tОтримане рівняння
регресії при m={m}:\n"
                                                    f"Y = {b_norm[0]:.4f} +
{b_norm[1]:.4f}*X1 + {b_norm[2]:.4f}*X2 +
                                                    f" {b_norm[3]:.4f}*X3\n")

# ===== Натуральні значення
=====
        print(header_format.format("=") + "+\n" +
"|{: ^79s}|\n".format("Матриця ПФЕ (натуральні значення факторів)") +
        header_format.format("=") + "+\n" +
row_format.format(*table_factors_1, *table_y) + "|\n" +
        header_format.format("=") + "+")

```

```

        for i in range(N):
            print("|{0:^4}|{1:^4}|".format(i + 1, xn[0][i]), end="")
            for j in range(3): print("{: ^ 4}|".format(x[j][i]), end="")
            for j in range(m): print("{:^10.4f}|".format(y[i][j]), end="")
            print("{0:^10.4f}|{1:^10.4f}|".format(y_avg[i],
dispersions[i]))

        print(separator_format.format("-") + f"+\n\n\tОтримане рівняння
регресії при m={m}:\n"

                                f"Y = {b_nat[0]:.4f} +
{b_nat[1]:.4f}*X1 + {b_nat[2]:.4f}*X2 + "
                                f"{b_nat[3]:.4f}*X3\n")

        # ===== Критерій Кохрена
=====
        f1, f2 = m - 1, N
        f3 = f1 * f2
        Gp = max(dispersions) / sum(dispersions)
        Gt = cochrans(f1, f2)

        print("Однорідність дисперсії (критерій Кохрена): ")
        print(f"Gp = {Gp}\nGt = {Gt}")
        if Gp < Gt:
            print("Дисперсія однорідна (Gp < Gt)")
            break
        else:
            print("Дисперсія неоднорідна (Gp > Gt), збільшуємо m,
повторюємо операції")
            m += 1
        D_beta = sum(dispersions) / (N*N*m)
        Sb = sqrt(abs(D_beta))
        beta = [sum([(y_avg[j] * xn[i][j]) / N for j in range(N)]) for i in
range(N)]

        t_list = [abs(i) / Sb for i in beta]

        student = partial(t.ppf, q=1-0.025)
        d, T = 0, student(df=f3)
        print("\n\t Табличне = ", T)
        for i in range(len(t_list)):
            if t_list[i] < T:
                beta[i] = 0
                print("\tГіпотеза підтверджена, beta{} = 0".format(i))
            else:
                print("\tГіпотеза не підтверджена, beta{} = {}".format(i,
beta[i]))
                d += 1

```

```

    yo = [beta[0] + beta[1] * x[0][i] + beta[2] * x[1][i] + beta[3] *
x[2][i] for i in range(N)]

    f4 = N - d
    fisher_sum = sum([(yo[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 / D_beta
    Ft = fisher(df=f4, dfd=f3)
    print(f"\nFp = {Fp}\nFt = {Ft}")
    if Fp > Ft:
        print("Рівняння регресії неадекватне (Ft < Fp).")
        flag = False
    else:
        print("Рівняння регресії адекватне (Ft > Fp)!")
        break

    if not flag:
        m, N = 3, 8
        print("-" * 120, "\nПочаток виконання алгоритму з N =
{}".format(N))
        print("\nПоточне m = {}".format(m))

        y = [[round(uniform(min(y_range), max(y_range)), 4) for i in
range(m)] for j in range(N)] # формування Y
        y_avg = list(map(lambda arr: round(sum(arr) / len(arr), 4), y)) #
середнє значення Y

        solve_b_norm = [xn[0], xn[1], xn[2], xn[3], xx[0], xx[1], xx[2],
xxx] # для нормального рівняння
        solve_b_nat = list(zip(xn[0], x[0], x[1], x[2], xx2[0], xx2[1],
xx2[2], xxx2)) # для натурального рівняння

        b_norm = [sum([((solve_b_norm[k][i] * y_avg[i]) / N) for i in
range(N)]) for k in range(N)] # b до норм. рівня
        b_nat = [round(i, 4) for i in solve(solve_b_nat, y_avg)] # b для
натурального рівняння
        dispersions = [sum([((y[i][j] - y_avg[i]) ** 2) / m for j in
range(m))] for i in range(N)] # дисперс. по рядках

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

        table_factors_1 = ["#", "X0", "X1", "X2", "X3"]
        table_factors_2 = ["X1X2", "X1X3", "X2X3", "X1X2X3"]
        table_y = ["Y1", "Y2", "Y3", "Y", "S^2"]

        header_format = "+{0: ^4}" * (len(table_factors_1)) + "+{0: ^8s}"

```

```

* (len(table_factors_2)) + \
    "{0:=^10s}" * (len(table_y))
    row_format = "|{: ^4}" * (len(table_factors_1)) + "|{: ^8}" *
(len(table_factors_2)) + "|{: ^10}" * (len(table_y))
    separator_format = "+{0:-^4s}" * (len(table_factors_1)) + "+{0:-
^8s}" * (len(table_factors_2)) + \
    "+{0:-^10s}" * (len(table_y))

# ===== Нормальні значення
=====
    print(header_format.format("=") + "+\n" +
"|{: ^115s}|\n".format("Матриця ПФЕ (нормальні значення факторів)") +
    header_format.format("=") + "+\n" +
row_format.format(*table_factors_1, *table_factors_2, *table_y) +
    "\n" + header_format.format("=") + "+")

    for i in range(N):
        print("|{: ^4}|".format(i + 1), end="")
        for j in range(4): print("{: ^4}|".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(m): print("{: ^10.4f}|".format(y[i][j]), end="")
        print("{0:^10.4f}|{1:^10.4f}|".format(y_avg[i],
dispersions[i]))

    print(separator_format.format("-") + f"+\n\n\tОтримане рівняння
регресії при m={m}:\n"
                                f"Y = {b_norm[0]:.4f} +
{b_norm[1]:.4f}*X1 + {b_norm[2]:.4f}*X2 + "
                                f"{b_norm[3]:.4f}*X3 +
{b_norm[4]:.4f}*X1X2 + {b_norm[5]:.4f}*X1X3 + "
                                f"{b_norm[6]:.4f}*X2X3 +
{b_norm[7]:.4f}*X1X2X3\n")

# ===== Натуральні значення
=====
    print(header_format.format("=") + "+\n" +
"|{: ^115s}|\n".format("Матриця ПФЕ (натуральні значення факторів)") +
    header_format.format("=") + "+\n" +
row_format.format(*table_factors_1, *table_factors_2, *table_y) +
    "\n" + header_format.format("=") + "+")

    for i in range(N):
        print("|{0:^4}|{1:^4}|".format(i + 1, xn[0][i]), end="")
        for j in range(3): print("{: ^ 4}|".format(x[j][i]), end="")
        for j in range(3): print("{: ^ 8}|".format(xx2[j][i]), end="")
        print("{: ^8}|".format(xxx2[i]), end="")
        for j in range(m): print("{: ^10.4f}|".format(y[i][j]), end="")
        print("{0:^10.4f}|{1:^10.4f}|".format(y_avg[i],
dispersions[i]))

```

```

print(separator_format.format("-") + f"+\n\n\tОтримане рівняння
регресії при m={m}:\n"

f"Y = {b_nat[0]:.4f} +
{b_nat[1]:.4f}*X1 + {b_nat[2]:.4f}*X2 + "
f"{b_nat[3]:.4f}*X3 +
{b_nat[4]:.4f}*X1X2 + {b_nat[5]:.4f}*X1X3 + "
f"{b_nat[6]:.4f}*X2X3 +
{b_nat[7]:.4f}*X1X2X3\n")

# ===== Критерій Кохрена
=====

f1, f2 = m - 1, N
f3 = f1 * f2
Gp = max(dispersions) / sum(dispersions)
Gt = cochrans(f1, f2)

print("Однорідність дисперсії (критерій Кохрена): ")
print(f"Gp = {Gp}\nGt = {Gt}")
if Gp < Gt:
    print("Дисперсія однорідна (Gp < Gt)")

D_beta = sum(dispersions) / (N * N * m)
Sb = sqrt(abs(D_beta))
full_matrix = xn + xx + [xxx]
beta = [sum([(y_avg[j] * full_matrix[i][j]) / N for j in
range(N)]) for i in range(N)]

t_list = [abs(i) / Sb for i in beta]

student = partial(t.ppf, q=1-0.025)
d, T = 0, student(df=f3)
print("\n\tтабличне = ", T)
for i in range(len(t_list)):
    if t_list[i] < T:
        beta[i] = 0
        print("\tГіпотеза підтверджена, beta{} = 0".format(i))
    else:
        print("\tГіпотеза не підтверджена, beta{} =
{}".format(i, beta[i]))
        d += 1

full_matrix2 = x + xx2 + [xxx2]
yo = [beta[0] + beta[1] * full_matrix2[0][i] + beta[2] *
full_matrix2[1][i] + beta[3] * full_matrix2[2][i] +
beta[4] * full_matrix2[3][i] + beta[5] *
full_matrix2[4][i] + beta[6] * full_matrix2[5][i] +
beta[7] * full_matrix2[6][i] for i in range(N)]

```



```

f4 = N - d
fisher_sum = sum([(yo[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 / D_beta
Ft = fisher(dfn=f4, dfd=f3)
print(f"\nFp = {Fp}\nFt = {Ft}")
if Fp > Ft:
    print("Рівняння регресії неадекватне (Ft < Fp).")
    N = 4 # генеруємо нові Y і повторюємо все заново
else:
    print("Рівняння регресії адекватне (Ft > Fp)!")
    break

else:
    print("Дисперсія неоднорідна (Gr > 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/lab4.py"

Початок виконання алгоритму з N = 4

Поточне m = 3

```

+====+====+====+====+====+====+====+====+====+====+====+====+
|                                     Матриця ПФЕ (нормальні значення факторів)                                     |
+====+====+====+====+====+====+====+====+====+====+====+====+
| # | X0 | X1 | X2 | X3 | Y1 | Y2 | Y3 | Y | S^2 |
+====+====+====+====+====+====+====+====+====+====+====+====+
| 1 | +1 | -1 | -1 | -1 | 227.7954 | 221.5497 | 204.1478 | 217.8310 | 100.1160 |
| 2 | +1 | -1 | +1 | +1 | 208.4305 | 241.0156 | 207.9122 | 219.1194 | 239.7658 |
| 3 | +1 | +1 | -1 | +1 | 208.9756 | 229.3812 | 236.7941 | 225.0503 | 138.3565 |
| 4 | +1 | +1 | +1 | -1 | 240.8182 | 236.7202 | 229.6378 | 235.7254 | 21.3284 |
+----+----+----+----+----+----+----+----+----+----+----+----+

```

Отримане рівняння регресії при m=3:

$Y = 224.4315 + 5.9563 \cdot X1 + 2.9909 \cdot X2 + -2.3467 \cdot X3$

```

+====+====+====+====+====+====+====+====+====+====+====+====+
|                                     Матриця ПФЕ (натуральні значення факторів)                                     |
+====+====+====+====+====+====+====+====+====+====+====+====+
| # | X0 | X1 | X2 | X3 | Y1 | Y2 | Y3 | Y | S^2 |
+====+====+====+====+====+====+====+====+====+====+====+====+
| 1 | +1 | 10 | -20 | 10 | 227.7954 | 221.5497 | 204.1478 | 217.8310 | 100.1160 |
| 2 | +1 | 10 | 60 | 15 | 208.4305 | 241.0156 | 207.9122 | 219.1194 | 239.7658 |
| 3 | +1 | 50 | -20 | 15 | 208.9756 | 229.3812 | 236.7941 | 225.0503 | 138.3565 |
| 4 | +1 | 50 | 60 | 10 | 240.8182 | 236.7202 | 229.6378 | 235.7254 | 21.3284 |
+----+----+----+----+----+----+----+----+----+----+----+----+

```

Отримане рівняння регресії при m=3:

$$Y = -13095.0218 + 7.6933 \cdot X_1 + 0.9368 \cdot X_2 + 1031.9954 \cdot X_3$$

Однорідність дисперсії (критерій Кохрена):

$G_p = 0.4799475903176873$

$G_t = 0.9057006759497539$

Дисперсія однорідна ($G_p < G_t$)

t табличне = 2.3060041350333704

Гіпотеза не підтверджена, $\beta_{\alpha 0} = 224.43152500000002$

Гіпотеза підтверджена, $\beta_{\alpha 1} = 0$

Гіпотеза підтверджена, $\beta_{\alpha 2} = 0$

Гіпотеза підтверджена, $\beta_{\alpha 3} = 0$

$F_p = 19.18975951001107$

$F_t = 4.06618055135116$

Рівняння регресії неадекватне ($F_t < F_p$).

Початок виконання алгоритму з N = 8

Поточне m = 3

Матриця ПДЕ (нормальні значення факторів)													
#	X0	X1	X2	X3	X1X2	X1X3	X2X3	X1X2X3	Y1	Y2	Y3	Y	S^2
1	+1	-1	-1	-1	+1	+1	+1	-1	203.5447	209.0914	203.8501	205.4954	6.4812
2	+1	-1	+1	+1	-1	-1	+1	-1	218.1753	238.9288	211.8769	222.9937	133.5759
3	+1	+1	-1	+1	-1	+1	-1	-1	236.5865	218.6755	236.5645	230.6088	71.2023
4	+1	+1	+1	-1	+1	-1	-1	-1	233.2618	228.9887	222.5155	228.2553	19.5161
5	+1	-1	-1	+1	+1	-1	-1	+1	240.7766	239.7698	238.8705	239.8056	0.6062
6	+1	-1	+1	-1	-1	+1	-1	+1	220.8633	210.7080	205.7590	212.4434	39.5292
7	+1	+1	-1	-1	-1	-1	+1	+1	223.5187	201.4164	215.8536	213.5962	83.9665
8	+1	+1	+1	+1	+1	+1	+1	+1	214.2229	208.1928	220.1882	214.2013	23.9818

Отримане рівняння регресії при m=3:

$$Y = 220.9250 + 0.7404 \cdot X_1 + -1.4515 \cdot X_2 + 5.9774 \cdot X_3 + 1.0144 \cdot X_1X_2 + -5.2377 \cdot X_1X_3 + -6.8533 \cdot X_2X_3 + -0.9133 \cdot X_1X_2X_3$$

Матриця ПДЕ (натуральні значення факторів)													
#	X0	X1	X2	X3	X1X2	X1X3	X2X3	X1X2X3	Y1	Y2	Y3	Y	S^2
1	+1	10	-20	10	-200	100	-200	-2000	203.5447	209.0914	203.8501	205.4954	6.4812
2	+1	10	60	15	600	150	900	+9000	218.1753	238.9288	211.8769	222.9937	133.5759
3	+1	50	-20	15	-1000	750	-300	-15000	236.5865	218.6755	236.5645	230.6088	71.2023
4	+1	50	60	10	3000	500	600	+30000	233.2618	228.9887	222.5155	228.2553	19.5161
5	+1	10	-20	15	-200	150	-300	-3000	240.7766	239.7698	238.8705	239.8056	0.6062
6	+1	10	60	10	600	100	600	+6000	220.8633	210.7080	205.7590	212.4434	39.5292
7	+1	50	-20	10	-1000	500	-200	-10000	223.5187	201.4164	215.8536	213.5962	83.9665
8	+1	50	60	15	3000	750	900	+45000	214.2229	208.1928	220.1882	214.2013	23.9818

Отримане рівняння регресії при m=3:
$$Y = 138.4227 + 1.2069 \cdot X_1 + 0.6111 \cdot X_2 + 6.6303 \cdot X_3 + 0.0070 \cdot X_1 X_2 + -0.0956 \cdot X_1 X_3 + -0.0548 \cdot X_2 X_3 + -0.0005 \cdot X_1 X_2 X_3$$

Однорідність дисперсії (критерій Кохрена):

Gp = 0.35257405906386924

Gt = 0.815948432359917

Дисперсія однорідна (Gp < Gt)

t табличне = 2.1199052992210112

Гіпотеза не підтверджена, beta0 = 220.92496249999996

Гіпотеза підтверджена, beta1 = 0

Гіпотеза підтверджена, beta2 = 0

Гіпотеза не підтверджена, beta3 = 5.977387499999999

Гіпотеза підтверджена, beta4 = 0

Гіпотеза не підтверджена, beta5 = -5.2377375000000015

Гіпотеза не підтверджена, beta6 = -6.853312500000001

Гіпотеза підтверджена, beta7 = 0

Fp = 83566365.9322675

Ft = 3.0069172799243438

Рівняння регресії неадекватне (Ft < Fp).

Початок виконання алгоритму з N = 4

Поточне m = 3

Матриця ПДЕ (нормальні значення факторів)										
#	X0	X1	X2	X3	Y1	Y2	Y3	Y	S^2	
1	+1	-1	-1	-1	213.5726	202.6437	229.8099	215.3421	124.5659	
2	+1	-1	+1	+1	211.0314	230.6363	210.0393	217.2357	89.9525	
3	+1	+1	-1	+1	207.8016	221.9700	217.3449	215.7055	34.8011	
4	+1	+1	+1	-1	229.0587	204.7885	221.3241	218.3904	102.4770	

Отримане рівняння регресії при m=3:
$$Y = 216.6684 + 0.3795 \cdot X_1 + 1.1446 \cdot X_2 + -0.1978 \cdot X_3$$

Матриця ПДЕ (натуральні значення факторів)										
#	X0	X1	X2	X3	Y1	Y2	Y3	Y	S^2	
1	+1	10	-20	10	213.5726	202.6437	229.8099	215.3421	124.5659	
2	+1	10	60	15	211.0314	230.6363	210.0393	217.2357	89.9525	
3	+1	50	-20	15	207.8016	221.9700	217.3449	215.7055	34.8011	
4	+1	50	60	10	229.0587	204.7885	221.3241	218.3904	102.4770	

Отримане рівняння регресії при $m=3$:

$$Y = -12059.6070 + 7.6388 \cdot X_1 + 0.8774 \cdot X_2 + 949.3385 \cdot X_3$$

Однорідність дисперсії (критерій Кохрена):

$$G_p = 0.35408514688234255$$

$$G_t = 0.9057006759497539$$

Дисперсія однорідна ($G_p < G_t$)

$$t_{\text{табличне}} = 2.3060041350333704$$

Гіпотеза не підтверджена, $\beta_0 = 216.668425$

Гіпотеза підтверджена, $\beta_1 = 0$

Гіпотеза підтверджена, $\beta_2 = 0$

Гіпотеза підтверджена, $\beta_3 = 0$

$$F_p = 0.8150204997519017$$

$$F_t = 4.06618055135116$$

Рівняння регресії адекватне ($F_t > F_p$)!

Process finished with exit code 0