

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

Лабораторна робота №4

з дисципліни

«Методи оптимізації та планування експерименту»

**на тему: «ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ
ВИКОРИСТАННІ РІВНЯННЯ РЕГРЕСІЇ З УРАХУВАННЯМ ЕФЕКТУ
ВЗАЄМОДІЇ»**

Виконав:

Студент 2-го курсу ФІОТ

групи ІО-82

Шендріков Є.О.

Залікова книжка № 8227

Номер у списку групи 25

Перевірив:

Регіда П. Г.

ВАРІАНТ

225	-20	15	25	45	-20	-15
-----	-----	----	----	----	-----	-----

ФРАГМЕНТ КОДУ

```
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 = [-20, 15], [25, 45], [-20, -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
      [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
      [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)
    xnmat = transpose(x)
    a = [[sum([comb(xnat[j])[i] * comb(xnat[j])[k] for j in range(n))
for i in range(n)] for k in range(n)]
    c = [sum([comb(xnat[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 = {}\n".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}|\n".format(i + 1), end="")
    for j in range(4): print("{: ^4}|\n".format(xn[j][i]), end="")
    for j in range(m): print("{: ^10.4f}|\n".format(y[i][j]), end="")
    print("{0: ^10.4f}|{1: ^10.4f}|\n".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("\nt 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(dfn=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 = {}\n".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}|\n".format(i + 1), end="")
    for j in range(4): print("{: ^4}|\n".format(xn[j][i]), end="")
    for j in range(3): print("{: ^8}|\n".format(xx[j][i]), end="")
    print("{: ^8}|\n".format(xxx[i]), end="")
    for j in range(m): print("{: ^10.4f}|\n".format(y[i][j]), end="")
    print("{0: ^10.4f}|{1: ^10.4f}|\n".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("\ntабличне = ", 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 повторємо все заново
else:
    print("Рівняння регресії адекватне (Ft > Fp)!")
    break

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

```

РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ

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	208.8644	197.0582	206.7634	204.2287	26.4435	
2	+1	-1	+1	+1	205.7587	208.2223	208.6869	207.5560	1.6511	
3	+1	+1	-1	+1	203.1445	206.6903	214.8707	208.2352	24.1106	
4	+1	+1	+1	-1	210.8916	210.0859	202.1944	207.7240	15.3962	

Отримане рівняння регресії при m=3:
 $Y = 206.9360 + 1.0436 \cdot X1 + 0.7040 \cdot X2 + 0.9596 \cdot X3$

Матриця ПФЕ (натуральні значення факторів)										
#	X0	X1	X2	X3	Y1	Y2	Y3	Y	S^2	
1	+1	-20	25	-20	208.8644	197.0582	206.7634	204.2287	26.4435	
2	+1	-20	45	-15	205.7587	208.2223	208.6869	207.5560	1.6511	
3	+1	15	25	-15	203.1445	206.6903	214.8707	208.2352	24.1106	
4	+1	15	45	-20	210.8916	210.0859	202.1944	207.7240	15.3962	

```

Отримане рівняння регресії при m=3:
Y = -39018.3126 + -3.9713*X1 + 263.7154*X2 + -1660.7739*X3

Однорідність дисперсій (критерій Кохрена):
Gr = 0.3911678660054457
Gt = 0.9057006759497539
Дисперсія однорідна (Gr < Gt)

t табличне = 2.3060041350333704
Гіпотеза не підтверджена, beta0 = 206.93597499999998
Гіпотеза підтверджена, beta1 = 0
Гіпотеза підтверджена, beta2 = 0
Гіпотеза підтверджена, beta3 = 0

Fr = 7.116589362821517
Ft = 4.06618055135116
Рівняння регресії неадекватне (Ft < Fr).
-----
Початок виконання алгоритму з N = 8

Поточне m = 3

+====+====+====+====+====+====+====+====+====+====+====+====+====+====+====+
|
| Матриця ПФЕ (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 | 207.9655 | 203.6527 | 213.4556 | 208.3579 | 16.0931 |
| 2 | +1 | -1 | +1 | +1 | -1 | -1 | +1 | -1 | 206.8271 | 212.5394 | 195.0749 | 204.8138 | 52.8615 |
| 3 | +1 | +1 | -1 | +1 | -1 | +1 | -1 | -1 | 205.2460 | 201.4337 | 212.0007 | 206.2268 | 19.0912 |
| 4 | +1 | +1 | +1 | -1 | +1 | -1 | -1 | -1 | 206.4326 | 200.2641 | 202.9086 | 203.2018 | 6.3847 |
| 5 | +1 | -1 | -1 | +1 | +1 | -1 | -1 | +1 | 200.4809 | 199.1002 | 204.8218 | 201.4676 | 5.9429 |
| 6 | +1 | -1 | +1 | -1 | -1 | +1 | -1 | +1 | 200.5350 | 213.8254 | 213.6922 | 209.3509 | 38.8627 |
| 7 | +1 | +1 | -1 | -1 | -1 | -1 | +1 | +1 | 204.6481 | 212.9645 | 199.5464 | 205.7197 | 30.5817 |
| 8 | +1 | +1 | +1 | +1 | +1 | +1 | +1 | +1 | 204.5978 | 201.0534 | 210.3758 | 205.3423 | 14.7617 |
|
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+

Отримане рівняння регресії при m=3:
Y = 205.5601 + -0.4375*X1 + 0.1171*X2 + -1.0975*X3 + -0.9677*X1X2 + 1.7594*X1X3 + 0.4983*X2X3 + -0.0900*X1X2X3

+====+====+====+====+====+====+====+====+====+====+====+====+====+====+====+
|
| Матриця ПФЕ (натуральні значення факторів)
|
+====+====+====+====+====+====+====+====+====+====+====+====+====+====+====+
| # | X0 | X1 | X2 | X3 | X1X2 | X1X3 | X2X3 | X1X2X3 | Y1 | Y2 | Y3 | Y | S^2 |
|
+====+====+====+====+====+====+====+====+====+====+====+====+====+====+====+
| 1 | +1 | -20 | 25 | -20 | -500 | 400 | -500 | +10000 | 207.9655 | 203.6527 | 213.4556 | 208.3579 | 16.0931 |
| 2 | +1 | -20 | 45 | -15 | -900 | 300 | -675 | +13500 | 206.8271 | 212.5394 | 195.0749 | 204.8138 | 52.8615 |
| 3 | +1 | 15 | 25 | -15 | 375 | -225 | -375 | -5625 | 205.2460 | 201.4337 | 212.0007 | 206.2268 | 19.0912 |
| 4 | +1 | 15 | 45 | -20 | 675 | -300 | -900 | -13500 | 206.4326 | 200.2641 | 202.9086 | 203.2018 | 6.3847 |
| 5 | +1 | -20 | 25 | -15 | -500 | 300 | -375 | +7500 | 200.4809 | 199.1002 | 204.8218 | 201.4676 | 5.9429 |
| 6 | +1 | -20 | 45 | -20 | -900 | 400 | -900 | +18000 | 200.5350 | 213.8254 | 213.6922 | 209.3509 | 38.8627 |
| 7 | +1 | 15 | 25 | -20 | 375 | -300 | -500 | -7500 | 204.6481 | 212.9645 | 199.5464 | 205.7197 | 30.5817 |
| 8 | +1 | 15 | 45 | -15 | 675 | -225 | -675 | -10125 | 204.5978 | 201.0534 | 210.3758 | 205.3423 | 14.7617 |
|
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+

Отримане рівняння регресії при m=3:
Y = 187.7546 + 0.9983*X1 + 0.3377*X2 + -1.0181*X3 + -0.0091*X1X2 + 0.0474*X1X3 + 0.0194*X2X3 + -0.0002*X1X2X3

Однорідність дисперсій (критерій Кохрена):
Gr = 0.28638854467624036
Gt = 0.815948432359917
Дисперсія однорідна (Gr < Gt)

t табличне = 2.119905299221011
Гіпотеза не підтверджена, beta0 = 205.5601
Гіпотеза підтверджена, beta1 = 0
Гіпотеза підтверджена, beta2 = 0
Гіпотеза підтверджена, beta3 = 0
Гіпотеза підтверджена, beta4 = 0
Гіпотеза підтверджена, beta5 = 0
Гіпотеза підтверджена, beta6 = 0
Гіпотеза підтверджена, beta7 = 0

Fr = 20.320642157079885
Ft = 2.6571966002210865
Рівняння регресії неадекватне (Ft < Fr).
-----

```


Початок виконання алгориту 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	195.1936	209.7470	203.7992	202.9133	35.6927	
2	+1	-1	+1	+1	203.9373	208.8400	201.4537	204.7437	9.4180	
3	+1	+1	-1	+1	195.3800	214.3620	199.4334	203.0585	66.6233	
4	+1	+1	+1	-1	205.8344	195.7340	212.5477	204.7054	47.7541	

Отримане рівняння регресії при m=3:

$$Y = 203.8552 + 0.0267 \cdot X_1 + 0.8693 \cdot X_2 + 0.0459 \cdot X_3$$

Матриця ПФЕ (натуральні значення факторів)										
#	X0	X1	X2	X3	Y1	Y2	Y3	Y	S^2	
1	+1	-20	25	-20	195.1936	209.7470	203.7992	202.9133	35.6927	
2	+1	-20	45	-15	203.9373	208.8400	201.4537	204.7437	9.4180	
3	+1	15	25	-15	195.3800	214.3620	199.4334	203.0585	66.6233	
4	+1	15	45	-20	205.8344	195.7340	212.5477	204.7054	47.7541	

Отримане рівняння регресії при m=3:

$$Y = -38806.5459 + -3.9590 \cdot X_1 + 261.7676 \cdot X_2 + -1652.8335 \cdot X_3$$

Однорідність дисперсії (критерій Кохрена):

Gp = 0.4177319925365374

Gt = 0.9057006759497539

Дисперсія однорідна (Gp < Gt)

t табличне = 2.3060041350333704

Гіпотеза не підтверджена, beta0 = 203.85522500000002

Гіпотеза підтверджена, beta1 = 0

Гіпотеза підтверджена, beta2 = 0

Гіпотеза підтверджена, beta3 = 0

Fp = 0.9131753216671814

Ft = 4.06618055135116

Рівняння регресії адекватне (Ft > Fp)!

Process finished with exit code 0