

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Методи оптимізації та планування експерименту

Лабораторна робота №6

**«Проведення трьохфакторного експерименту при
використанні рівняння регресії з квадратичними
членами»**

Виконала:

студентка групи ІВ-81

Базова Л.Г.

Номер залікової книжки № 8101

Перевірив: Регіда П.Г.

Київ 2020 р.

Код програми:

```
import numpy as np
from scipy.stats import f, t
from tabulate import tabulate
import sklearn.linear_model as lm
import random

def mult(x1, x2, x3 = np.ones(15)):
    x = np.ones(15)
    for i in range(15):
        x[i] *= x1[i] * x2[i] * x3[i];
    return x

def f_x_func(x, b):
    global f_x
    f_x = np.zeros(N)
    for i in range(N):
        f_x[i] += b[0]
        for k in range(len(x[0])):
            f_x[i] += b[k + 1] * x[i][k]

def find_x(x_min, x_max):
    #величини для значень матриці планування
    x01 = (x_max[0] + x_min[0]) / 2
    x02 = (x_max[1] + x_min[1]) / 2
    x03 = (x_max[2] + x_min[2]) / 2
    delta_x1 = x_max[0] - x01
    delta_x2 = x_max[1] - x02
    delta_x3 = x_max[2] - x03
    X1 = np.array([x_min[0], x_min[0], x_min[0], x_min[0], x_max[0], x_max[0], x_max[0],
x_max[0], -l*delta_x1+x01, l*delta_x1+x01, x01, x01, x01, x01, x01])
    X2 = np.array([x_min[1], x_min[1], x_max[1], x_max[1], x_min[1], x_min[1], x_max[1],
x_max[1], x02, x02, -l*delta_x2+x02, l*delta_x2+x02, x02, x02, x02])
    X3 = np.array([x_min[2], x_max[2], x_min[2], x_max[2], x_min[2], x_max[2], x_min[2],
x_max[2], x03, x03, -l*delta_x3+x03, l*delta_x3+x03, x03])
    return np.array(list(zip(X1, X2, X3, mult(X1, X2), mult(X1, X3), mult(X2, X3),
mult(X1, X2, X3), mult(X1, X1), mult(X2, X2), mult(X3, X3))))

def find_y(x, b):
    y_val = np.zeros((N, m))
    f_x_func(x, b)
    for i in range(N):
        for j in range(m):
            y_val[i][j] += f_x[i] + random.random()*10 - 5
    return y_val

def find_b(X, y):
    x = list(X)
    for i in range(len(x)):
        x[i] = np.array([1, ] + list(x[i]))
    X = np.array(x)
    model = lm.LinearRegression(fit_intercept=False)
    model.fit(X, y)
    coefs = model.coef_
    print("Коефіцієнти рівняння регресії:")
    for i in range(len(coefs)):
        print(f"b{i:} = {coefs[i]:}")
    print("\nРівняння регресії")
    print(f"y = {coefs[0]:.3f} + x1 * {coefs[1]:.3f} + x2 * {coefs[2]:.3f} + x3 *
{coefs[3]:.3f}"
        f" + x1x2 * {coefs[4]:.3f} + x1x3 * {coefs[5]:.3f} + x2x3 * {coefs[6]:.3f}"
        f" + x1x2x3 * {coefs[7]:.4f} + x1^2 * {coefs[8]:.4f} + x2^2 * {coefs[9]:.4f} +
x3^2 * {coefs[10]:.4f}")
```

```

    return coefs
def factor_val_check():
    print("Виконаємо перевірку, підставивши значення факторів з матриці планування і порівняємо результат з середніми значеннями функцій відгуку")
    for i in range(N):
        print(f"y{i+1:} = {f_x[i]:} => {y_mean[i]:} = avgY{i+1:}")
    print("Так як значення співдають, то коефіцієнти рівняння розраховані правильно.\n")
def find_disper(y, y_mean):
    disper = np.zeros(N)
    for i in range(N):
        for j in range(m):
            disper[i] += (y[i][j] - y_mean[i]) ** 2
        disper[i] /= m
    return disper
def matrix_print(y, x_list, y_mean, disper):
    global header_table
    header_table = ["№", "x1", "x2", "x3", "x1x2", "x1x3", "x2x3", "x1x2x3", "x1^2", "x2^2", "x3^2"]
    table = []
    for i in range(N):
        table.append([i + 1])
    for i in range(N):
        for _ in range(len(x_list[0])):
            table[i].append(x_list[i][_])
        for j in range(m):
            table[i].append(y[i][j])
        table[i].append(y_mean[i])
        table[i].append(disper[i])
    for i in range(m):
        header_table.append("Y" + str(i + 1))
    header_table.append("Y")
    header_table.append("S^2")
    print(tabulate(table, headers=header_table, tablefmt="fancy_grid"))
def print_eq(b, numb):
    if(numb == 1):
        print((f"f(x1, x2, x3) = {b[0]:.1f} + x1 * {b[1]:.1f} + x2 * {b[2]:.1f} + x3 * {b[3]:.1f}"
            f"+ x1x2 * {b[4]:.1f} + x1x3 * {b[5]:.1f} + x2x3 * {b[6]:.1f}"
            f"+ x1x2x3 * {b[7]:.1f} + x1^2 * {b[8]:.1f} + x2^2 * {b[9]:.1f} + x3^2 * {b[10]:.1f}"))
    else:
        print((f"f(x1, x2, x3) = {b[0]:.7f} + x1 * {b[1]:.7f} + x2 * {b[2]:.7f} + x3 * {b[3]:.7f}"
            f"+ x1x2 * {b[4]:.7f} + x1x3 * {b[5]:.7f} + x2x3 * {b[6]:.7f}"
            f"+ x1x2x3 * {b[7]:.7f} + x1^2 * {b[8]:.7f} + x2^2 * {b[9]:.7f} + x3^2 * {b[10]:.7f}"))
def initial_print():
    for i in range(len(x_min)):
        print(f"x{i:}min = {x_min[i]:.1f}          x{i:}max = {x_max[i]:.1f}")
def kohren_check(disper):
    global Gp, Gt, f1, f2
    print("Критерій Кохрена")
    Gp = max(disper) / sum(disper)
    f1 = m - 1
    f2 = N
    fisher = f.isf([q / f2, f1, (f2 - 1) * f1])
    Gt = round(fisher / (fisher + (f2 - 1)), 4)
    print(f"Gp = {Gp:}\nКількість степенів свободи: F1 = m - 1 = {f1:}; F2 = N = {f2:}")
    print(f"Рівень значимості: q = 1 - p = {q:}\nТабличне значення коефіцієнту Кохрена: Gt = {Gt:}")

```

```

def student_check():
    global sb, d, f3, t_exp
    d = len(x_code[0])
    f3 = f1 * f2
    print(f"Критерій Ст`юдента\нЧисло степенів свободи: F3 = F1*F2 = {f3:}")
    sb = sum(disper) / N
    ssbs = sb / N * m
    sbs = ssbs ** 0.5
    beta = np.zeros(d)
    t_exp = []
    for j in range(d):
        for i in range(N):
            if (j == 0):
                beta[j] += y_mean[i]
            else:
                beta[j] += y_mean[i] * x_code[i][j]
        beta[j] /= N
        t_exp.append(abs(beta[j]) / sbs)

    ttabl = round(abs(t.ppf(q / 2, f3)), 4)
    print(f"табл = {ttabl:}")
    string_eq = f"y = {b[0]:.7f}"
    for i in range(len(t_exp)):
        if (t_exp[i] < ttabl):
            print(f"Коефіцієнт t{i:} = {t_exp[i]:.7f} = > коефіцієнт не значимий")
            b[i] = 0
            d = d - 1
        else:
            print(f"Коефіцієнт t{i:} = {t_exp[i]:.7f} = > коефіцієнт значимий")
            if(i != 0): string_eq += f" + {b[i]:.7f} * " + header_table[i]
    print(f"Значимих коефіцієнтів: d = ", d, "\n\nРівняння регресії після виключення коефіцієнтів:\n", string_eq)
    print(f"\nПеревірка при підстановці в рівняння регресії:")
    f_x_func(x_list, b)
    for i in range(len(f_x)):
        print(f"y'{i+1:} = {f_x[i]:} => {y_mean[i]:} = avgY{i+1:}")
def fisher_check():
    global Fp, Ft
    print(f"\nКритерій Фішера")
    f4 = N - d
    sad = 0
    for i in range(N):
        sad += (f_x[i] - y_mean[i]) ** 2
    sad *= (m / (N - d))
    Fp = sad / sb
    print(f"Fp = {Fp:}")
    print(f"Кількість степенів свободи: F4 = N - d = {f4:}")
    Ft = round(abs(f.isf(q, f4, f3)), 4)
    print(f"Табличне значення коефіцієнту Фішера: Ft = {Ft:}")

#величини за варіантом:
x_min = np.array([-10, 20, 50])
x_max = np.array([50, 60, 55])
initial_print()
b_initial = np.array([1.5, 1.5, 6.8, 3.2, 6.2, 0.7, 2.4, 6.1, 2.7, 0.1, 1.2])
#рівняння з урахуванням квадратичних членів
print_eq(b_initial, 1)
print(f"\nРівняння регресії:")
print(f"y = b0 + b1*x1 + b2*x2 + b3*x3 + b12*x1*x2 + b13*x1*x3 + b23*x2*x3 + b123*x1*x2*x3 + b11*x1^2 + b22*x2^2 + b33*x3^2\n")

```

```

#константи для початкових умов
m = 3
k = 3 #const
p = 0
N = 15 #2^(k - p)+2k + N0
l = k*(1/2)
q = 0.05
x_code = find_x(np.array([-1, -1, -1]), np.array([1, 1, 1]))
#матриця планування
x_list = find_x(x_min, x_max)
y = find_y(x_list, b_initial)
while 1:
    if(m > 3):
        next_int = np.random.randint(y_min, y_max, size=(N, 1))
        y = np.append(y, next_int, axis=1)
        y_mean = np.sum(y, axis=1) / m
        disper = find_disper(y, y_mean)
        print("Матриця планування:")
        matrix_print(y, x_code, y_mean, disper)
        print("Натуралізована матриця:")
        matrix_print(y, x_list, y_mean, disper)

        b = find_b(x_list, y_mean)
        factor_val_check()
        print("Статистичні перевірки:")
        kohren_check(disper)          #find Gp, Gt, f1, f2
        if Gp > Gt:
            print("Дисперсія неоднорідна , потрібно збільшити m\n")
            m = m + 1
            continue
        print("Gp < Gt -> Дисперсія однорідна з ймовірністю 0.95\n")
        student_check()
        fisher_check()
        if Fp > Ft:
            print("Fp > Ft = > отримана математична модель з ймовірністю 0.95 неадекватна
експериментальним даним\n\n\n\n")
            y = find_y(x_list, b_initial)
            continue
        else:
            print("Fp < Ft = > отримана математична модель з ймовірністю 0.95 адекватна
експериментальним даним")
            break

```

Результат роботи програми

x0min = -10.0 x0max = 50.0
x1min = 20.0 x1max = 60.0
x2min = 50.0 x2max = 55.0
f(x1, x2, x3) = 1.5 + x1 * 1.5 + x2 * 6.8 + x3 * 3.2+ x1x2 * 6.2 + x1x3 * 0.7 + x2x3 * 2.4+ x1x2x3 * 6.1 + x1^2 * 2.7 + x2^2 * 0.1 + x3^2 * 1.2

Рівняння регресії:
y = b0 + b1*x1 + b2*x2 + b3*x3 + b12*x1*x2 + b13*x1*x3 + b23*x2*x3 + b123*x1*x2*x3 + b11*x1^2 + b22*x2^2 + b33*x3^2

Матриця планування:

№	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	-1.73205	0	0	-0	-0	0	-0	3	0	0
10	1.73205	0	0	0	0	0	0	3	0	0
11	0	-1.73205	0	-0	0	-0	-0	0	3	0
12	0	1.73205	0	0	0	0	0	0	3	0
13	0	0	-1.73205	0	-0	-0	-0	0	0	3
14	0	0	1.73205	0	0	0	0	0	0	3
15	0	0	0	0	0	0	0	0	0	0

Натуралізована матриця:

№	x1	x2	x3	x1x2	x1x3	x2x3	x1x2x3	x1^2	x2^2	x3^2	Y1	Y2	Y3	Y	S^2
1	-10	20	50	-200	-500	1000	-10000	100	400	2500	-56597.7	-56592.6	-56596.1	-56595.5	4.63969
2	-10	20	55	-200	-550	1100	-11000	100	400	3025	-61842.7	-61843.1	-61849.4	-61845.1	9.364
3	-10	60	50	-600	-500	3000	-30000	100	3600	2500	-175682	-175686	-175683	-175684	2.45841
4	-10	60	55	-600	-550	3300	-33000	100	3600	3025	-192653	-192657	-192651	-192654	6.99978
5	50	20	50	1000	2500	1000	50000	2500	400	2500	325509	325508	325510	325509	0.625879
6	50	20	55	1000	2750	1100	55000	2500	400	3025	357075	357076	357069	357073	9.79001
7	50	60	50	3000	2500	3000	150000	2500	3600	2500	953306	953302	953303	953304	3.51806
8	50	60	55	3000	2750	3300	165000	2500	3600	3025	1.04634e+06	1.04635e+06	1.04635e+06	1.04635e+06	3.66522
9	-31.9615	40	52.5	-1278.46	-1677.98	2100	-67119.2	1021.54	1600	2756.25	-406871	-406867	-406866	-406868	5.01362
10	71.9615	40	52.5	2878.46	3777.98	2100	151119	5178.46	1600	2756.25	965362	965359	965360	965360	1.93415
11	20	5.35898	52.5	107.18	1050	281.347	5626.93	400	28.7187	2756.25	41027.1	41021.8	41022.3	41023.7	5.60509
12	20	74.641	52.5	1492.82	1050	3918.65	78373.1	400	5571.28	2756.25	503121	503119	503118	503119	2.24599
13	20	40	48.1699	800	963.397	1926.79	38535.9	400	1600	2320.34	249813	249813	249807	249811	7.39862
14	20	40	56.8301	800	1136.6	2273.21	45464.1	400	1600	3229.66	294139	294144	294147	294143	9.44994
15	20	40	52.5	800	1050	2100	42000	400	1600	2756.25	271949	271957	271958	271955	15.0363

Коефіцієнти рівняння регресії:

$$b_0 = 13.860076696963734$$

$$b_1 = 0.8517156996965269$$

$$b_2 = 7.126177361885454$$

$$b_3 = 2.7372795008970656$$

$$b_4 = 6.199702735565298$$

$$b_5 = 0.7108669474683843$$

$$b_6 = 2.3973208812049505$$

$$b_7 = 6.100026558583673$$

$$b_8 = 2.7005514536131274$$

$$b_9 = 0.0974403035771031$$

$$b_{10} = 1.2035705963924348$$

Рівняння регресії

$$y = 13.860 + x_1 * 0.852 + x_2 * 7.126 + x_3 * 2.737 + x_1x_2 * 6.200 + x_1x_3 * 0.711 + \\ + x_2x_3 * 2.397 + x_1x_2x_3 * 6.1000 + x_1^2 * 2.7006 + x_2^2 * 0.0974 + x_3^2 * 1.2036$$

Виконаємо перевірку, підставивши значення факторів з матриці

планування і порівняємо результат з середніми значеннями функцій відгуку

$$y_1 = -56597.5 \Rightarrow -56595.48983688911 = \text{avg}Y_1$$

$$y_2 = -61846.5 \Rightarrow -61845.09496549647 = \text{avg}Y_2$$

$$y_3 = -175685.5 \Rightarrow -175683.63445425394 = \text{avg}Y_3$$

$$y_4 = -192654.5 \Rightarrow -192653.82852378805 = \text{avg}Y_4$$

$$y_5 = 325512.5 \Rightarrow 325509.2120446984 = \text{avg}Y_5$$

$$y_6 = 357073.5 \Rightarrow 357073.0263518332 = \text{avg}Y_6$$

$$y_7 = 953304.5 \Rightarrow 953303.5410227313 = \text{avg}Y_7$$

$$y_8 = 1046345.4999999999 \Rightarrow 1046347.085091944 = \text{avg}Y_8$$

$$y_9 = -406868.95627524034 \Rightarrow -406867.9867415265 = \text{avg}Y_9$$

$$y_{10} = 965356.9562752404 \Rightarrow 965360.3042281662 = \text{avg}Y_{10}$$

$$y_{11} = 41025.350473542065 \Rightarrow 41023.74441381568 = \text{avg}Y_{11}$$

$$y_{12} = 503122.6495264579 \Rightarrow 503119.451951898 = \text{avg}Y_{12}$$

$$y_{13} = 249809.71376473352 \Rightarrow 249811.03732673964 = \text{avg}Y_{13}$$

$$y_{14} = 294143.2862352665 \Rightarrow 294143.43620775366 = \text{avg}Y_{14}$$

$$y_{15} = 271954.0 \Rightarrow 271954.7769767775 = \text{avg}Y_{15}$$

Так як значення співдають, то коефіцієнти рівняння розраховані правильно.

Статистичні перевірки:

Критерій Кохрена

$$G_p = 0.1713641928179019$$

$$\text{Кількість степенів свободи: } F_1 = m - 1 = 2; F_2 = N = 15$$

$$\text{Рівень значимості: } q = 1 - p = 0.05$$

$$\text{Табличне значення коефіцієнту Кохрена: } G_t = 0.3346$$

$$G_p < G_t \rightarrow \text{Дисперсія однорідна з ймовірністю } 0.95$$

Критерій Ст'юдента

$$\text{Число степенів свободи: } F_3 = F_1 * F_2 = 30$$

$$t_{\text{табл}} = 2.0423$$

$$\text{Коефіцієнт } t_0 = 253567.1416287 = > \text{коефіцієнт значимий}$$

$$\text{Коефіцієнт } t_1 = 115106.4737700 = > \text{коефіцієнт значимий}$$

$$\text{Коефіцієнт } t_2 = 11043.3941630 = > \text{коефіцієнт значимий}$$

$$\text{Коефіцієнт } t_3 = 96580.2002825 = > \text{коефіцієнт значимий}$$

Коефіцієнт $t_4 = 9049.7196039 = >$ коефіцієнт значимий
Коефіцієнт $t_5 = 3066.9140478 = >$ коефіцієнт значимий
Коефіцієнт $t_6 = 4511.7154764 = >$ коефіцієнт значимий
Коефіцієнт $t_7 = 238585.6111080 = >$ коефіцієнт значимий
Коефіцієнт $t_8 = 235932.3787322 = >$ коефіцієнт значимий
Коефіцієнт $t_9 = 235897.4828301 = >$ коефіцієнт значимий
Значимих коефіцієнтів: $d = 10$

Рівняння регресії після виключення коефіцієнтів:

$$y = 13.8600767 + 0.8517157 * x_1 + 7.1261774 * x_2 + 2.7372795 * x_3 + 6.1997027 * x_1x_2 + 0.7108669 * x_1x_3 + 2.3973209 * x_2x_3 + 6.1000266 * x_1x_2x_3 + 2.7005515 * x_1^2 + 0.0974403 * x_2^2$$

Перевірка при підстановці в рівняння регресії:

$y'_1 = -56595.63052572324 \Rightarrow -56595.48983688911 = \text{avg}Y_1$
 $y'_2 = -61845.90738294931 \Rightarrow -61845.09496549647 = \text{avg}Y_2$
 $y'_3 = -175684.54496329077 \Rightarrow -175683.63445425394 = \text{avg}Y_3$
 $y'_4 = -192655.4107614432 \Rightarrow -192653.82852378805 = \text{avg}Y_4$
 $y'_5 = 325510.633545034 \Rightarrow 325509.2120446984 = \text{avg}Y_5$
 $y'_6 = 357073.77612355043 \Rightarrow 357073.0263518332 = \text{avg}Y_6$
 $y'_7 = 953304.192702864 \Rightarrow 953303.5410227313 = \text{avg}Y_7$
 $y'_8 = 1046347.0650434581 \Rightarrow 1046347.085091944 = \text{avg}Y_8$
 $y'_9 = -406866.0757226373 \Rightarrow -406867.9867415265 = \text{avg}Y_9$
 $y'_{10} = 965358.607525703 \Rightarrow 965360.3042281662 = \text{avg}Y_{10}$
 $y'_{11} = 41022.96266022651 \Rightarrow 41023.74441381568 = \text{avg}Y_{11}$
 $y'_{12} = 503120.44802191335 \Rightarrow 503119.451951898 = \text{avg}Y_{12}$
 $y'_{13} = 249810.36883955554 \Rightarrow 249811.03732673964 = \text{avg}Y_{13}$
 $y'_{14} = 294144.3190113639 \Rightarrow 294143.43620775366 = \text{avg}Y_{14}$
 $y'_{15} = 271954.77697677736 \Rightarrow 271954.7769767775 = \text{avg}Y_{15}$

Критерій Фішера

$$F_p = 1.6801545280927717$$

Кількість степенів свободи: $F_4 = N - d = 5$

Табличне значення коефіцієнту Фішера: $F_t = 2.5336$

$F_p < F_t = >$ отримана математична модель з ймовірністю 0.95 адекватна експериментальним даним