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 наукових досліджень»
# Проведення трьохфакторного експерименту при використанні рівняння регресії з квадратичними членами

ВИКОНАВ:
Студент ІІ курсу ФІОТ
Групи ІВ-91
Микитенко В.О.
Номер заліковки: 9119

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Київ 2021 р.

**Мета роботи:** Провести трьохфакторний експеримент і отримати адекватну модель – рівняння регресії, використовуючи **рототабельний** композиційний план.

**Завдання до лабораторної роботи:**

1. Ознайомитися з теоретичними відомостями.
2. Вибрати з таблиці варіантів і записати в протокол інтервали значень x1, x2, x3. Обчислити і записати значення, відповідні кодованим значенням факторів +1; -1;+ ; - ; 0 для 1, 2, 3.
3. Значення функції відгуку знайти за допомогою підстановки в формулу:
yi = f(x1, x2, x3) + random(10)-5,
де f(x1, x2, x3) вибирається по номеру в списку в журналі викладача.
4. Провести експерименти і аналізуючи значення статистичних перевірок, отримати адекватну модель рівняння регресії. При розрахунках використовувати натуральні значення факторів.
5. Зробити висновки по виконаній роботі.

*Алгоритм отримання адекватної моделі рівняння регресії*

*1) Вибір рівняння регресії (лінійна форма, рівняння з урахуванням ефекту взаємодії і з урахуванням квадратичних членів);*
*2) Вибір кількості повторів кожної комбінації (m = 2);*
*3) Складення матриці планування експерименту і вибір кількості рівнів (N)*
*4) Проведення експериментів;*
*5) Перевірка однорідності дисперсії. Якщо не однорідна – повертаємося на п. 2 і збільшуємо m на 1);*
*6) Розрахунок коефіцієнтів рівняння регресії. При розрахунку використовувати **натуральні** значення x1, x2 и x3.*
*7) Перевірка нуль-гіпотези. Визначення значимих коефіцієнтів;*
*8) Перевірка адекватності моделі рівняння оригіналу. При неадекватності – повертаємося на п.1, змінивши при цьому рівняння регресії;*

Таблиця варіантів

| № варіанту | $x_1$ | | $x_2$ | | $x_3$ | | $f(x_1, x_2, x_3)$ |
|---|---|---|---|---|---|---|---|
| | min | max | min | max | min | max | |
| 118 | -20 | 30 | 5 | 40 | 5 | 10 | 0,6+8,0*x1+8,8*x2+9,2*x3+4,7*x1*x1+0,1*x2*x2+1,1*x3*x3+6,3*x1*x2+0,4*x1*x3+1,3*x2*x3+2,9*x1*x2*x3 |

## Лістинг програми

```
from math import fabs
from random import randrange
import numpy as np
from numpy.linalg import solve
from scipy.stats import f, t

m = 3
n = 15

x1_min = -20
x1_max = 30
x2_min = 5
x2_max = 40
x3_min = 5
x3_max = 10

coef_1 = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

while True:
    def function(x_1, x_2, x_3, c):
```

```
        y = 0.6*c[0] + 8.0 * x_1*c[1] + 8.8 * x_2*c[2] + 9.2 * x_3*c[3] + 4.7 *
x_1 * x_1*c[4] + 0.1 * x_2 * x_2*c[5] +\
            1.1 * x_3 * x_3*c[6] + 6.3 * x_1 * x_2*c[7] + 0.4 * x_1 * x_3*c[8] +
1.3 * x_2 * x_3*c[9] + 2.9 * x_1 * x_2\
            * x_3*c[10] + randrange(0, 10) - 5
        return y

    x01 = (x1_max + x1_min) / 2
    x02 = (x2_max + x2_min) / 2
    x03 = (x3_max + x3_min) / 2
    x1_delt = x1_max - x01
    x2_delt = x2_max - x02
    x3_delt = x3_max - x03

    xn = [[-1, -1, -1, +1, +1, +1, -1, +1, +1, +1],
          [-1, -1, +1, +1, -1, -1, +1, +1, +1, +1],
          [-1, +1, -1, -1, +1, -1, +1, +1, +1, +1],
          [-1, +1, +1, -1, -1, +1, -1, +1, +1, +1],
          [+1, -1, -1, -1, -1, +1, +1, +1, +1, +1],
          [+1, -1, +1, -1, +1, -1, -1, +1, +1, +1],
          [+1, +1, -1, +1, -1, -1, -1, +1, +1, +1],
          [+1, +1, +1, +1, +1, +1, +1, +1, +1, +1],
          [-1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],
          [+1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0],
          [0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0],
          [0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0],
          [0, 0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929],
          [0, 0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929],
          [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

    x1 = [x1_min, x1_min, x1_min, x1_min, x1_max, x1_max, x1_max, x1_max, -1.73
* x1_delt + x01, 1.73 * x1_delt + x01,
          x01, x01, x01, x01, x01]
    x2 = [x2_min, x2_min, x2_max, x2_max, x2_min, x2_min, x2_max, x2_max, x02,
x02, -1.73 * x2_delt + x02, 1.73 *
          x2_delt + x02, x02, x02, x02]
    x3 = [x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, x3_min, x3_max, x03,
x03, x03, x03, -1.73 * x3_delt + x03,
          1.73 * x3_delt + x03, x03]

    x1x2, x1x3, x2x3, x1x2x3 = [0] * n, [0] * n, [0] * n, [0] * n
    x1kv, x2kv, x3kv = [0] * n, [0] * n, [0] * n

    for i in range(15):
        x1x2[i] = x1[i] * x2[i]
        x1x3[i] = x1[i] * x3[i]
        x2x3[i] = x2[i] * x3[i]
        x1x2x3[i] = x1[i] * x2[i] * x3[i]
        x1kv[i] = x1[i] ** 2
        x2kv[i] = x2[i] ** 2
        x3kv[i] = x3[i] ** 2

    list_for_a = list(zip(x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1kv, x2kv,
x3kv))
    print("")

    print("Матриця планування з натуралізованими коефіцієнтами X")
    print("     X1          X2          X3        X1X2        X1X3        X2X3
X1X2X3       X1X1"
          "        X2X2         X3X3")
    for i in range(n):
        for j in range(len(list_for_a[0])):
            print("{:^12.3f}".format(list_for_a[i][j]), end='')
        print("")
    # вивід матриці планування
    Y = [[function(list_for_a[j][0], list_for_a[j][1], list_for_a[j][2], coef_1)
for i in range(m)] for j in range(15)]
    print("")
```

```
    print("Матриця планування Y")
    print("     Y1          Y2          Y3")
    for i in range(n):
        print(end='')
        for j in range(len(Y[0])):
            print("{:^12.3f}".format(Y[i][j]), end='')
        print("")
    # середні y
    aver_y = []
    for i in range(len(Y)):
        aver_y.append(np.mean(Y[i], axis=0))
    print("")

    print("Середні значення відгуку за рядками")
    for i in range(n):
        print("{:.3f}".format(aver_y[i]))
    # розрахунок дисперсій
    dispersions = []
    for i in range(len(Y)):
        a = 0
        for k in Y[i]:
            a += (k - np.mean(Y[i], axis=0)) ** 2
        dispersions.append(a / len(Y[i]))


    def find_kn(num):
        aa = 0
        for o in range(n):
            aa += aver_y[o] * list_for_a[o][num - 1] / n
        return aa


    def a(first, second):
        aaa = 0
        for p in range(n):
            aaa += list_for_a[p][first - 1] * list_for_a[p][second - 1] / n
        return aaa


    my = sum(aver_y) / n
    mx = []
    for i in range(10):
        number_lst = []
        for j in range(n):
            number_lst.append(list_for_a[j][i])
        mx.append(sum(number_lst) / len(number_lst))

    det_1 = [
        [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)]]

    det_2 = [my, find_kn(1), find_kn(2), find_kn(3), find_kn(4), find_kn(5),
find_kn(6), find_kn(7), find_kn(8),
             find_kn(9), find_kn(10)]

    beta = solve(det_1, det_2)
    print("")

    print("Отримане рівняння регресії")
    print("{:.3f} + {:.3f} * X1 + {:.3f} * X2 + {:.3f} * X3 + {:.3f} * X1X2 +
{:.3f} * X1X3 +\n + {:.3f} * X2X3 + "
          "{:.3f} * X1X2X3 + {:.3f} * X11^2 + {:.3f} * X22^2 + {:.3f} * X33^2 =
y".format(beta[0], beta[1], beta[2],

beta[3], beta[4], beta[5],

beta[6], beta[7], beta[8],

beta[9], beta[10]))
    y_i = [0] * n
    print("")
    print("Експериментальні значення")
    for k in range(n):
        y_i[k] = beta[0] + beta[1] * list_for_a[k][0] + beta[2] *
list_for_a[k][1] + beta[3] * list_for_a[k][2] + \
                 beta[4] * list_for_a[k][3] + beta[5] * list_for_a[k][4] +
beta[6] * list_for_a[k][5] + beta[7] * \
                 list_for_a[k][6] + beta[8] * list_for_a[k][7] + beta[9] *
list_for_a[k][8] + beta[10] *\
                 list_for_a[k][9]
    for i in range(n):
        print("{:.3f}".format(y_i[i]))
    print("")

    print("\033[1m\033[30m\033[43m{}\033[0m".format("Перевірка за критерієм
Кохрена"))
    Gp = max(dispersions) / sum(dispersions)
    GT = 0.3346
    print("Gp =", Gp)
    if Gp < GT:
        print("\033[1m\033[30m\033[42mGp={} < GT={} - Дисперсія
однорідна\033[0m".format(round(Gp, 3), GT))
    else:
        print("\033[1m\033[30m\033[41m{}\033[0m".format("Дисперсія
неоднорідна"))
    print("")
    print("\033[1m\033[30m\033[43m{}\033[0m".format("Перевірка значущості
коефіцієнтів за критерієм Стьюдента"))
    sb = sum(dispersions) / len(dispersions)
    sbs = (sb / (n * m)) ** 0.5

    F3 = (m - 1) * n
    coef_1 = []
    coef_2 = []
    d = 11
    res = [0] * 11
    for j in range(11):
        pract_t = 0
        for i in range(15):
            if j == 0:
                pract_t += aver_y[i] / 15
            else:
                pract_t += aver_y[i] * xn[i][j - 1]
            res[j] = beta[j]
        if fabs(pract_t / sbs) < t.ppf(q=0.975, df=F3):
```

```
            coef_2.append(beta[j])
            res[j] = 0
            d -= 1
        else:
            coef_1.append(beta[j])
    print("Значущі коефіцієнти регресії", [round(i, 3) for i in coef_1])
    print("Незначущі коефіцієнти регресії", [round(i, 3) for i in coef_2])
    st_y = []
    for i in range(n):
        st_y.append(res[0] + res[1] * x1[i] + res[2] * x2[i] + res[3] * x3[i] +
res[4] * x1x2[i] + res[5] *
                    x1x3[i] + res[6] * x2x3[i] + res[7] * x1x2x3[i] + res[8] *
x1kv[i] + res[9] *
                    x2kv[i] + res[10] * x3kv[i])
    print("")

    print("Значення з отриманими коефіцієнтами")
    for i in range(n):
        print("{:.3f}".format(st_y[i]))
    print("")

    print("\033[1m\033[30m\033[43m{}\033[0m".format("Перевірка адекватності за
критерієм Фішера"))
    Sad = m * sum([(st_y[i] - aver_y[i]) ** 2 for i in range(n)]) / (n - d)
    Fp = Sad / sb
    F4 = n - d
    print("Fp =", Fp)
    if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
        print("\033[1m\033[30m\033[42m{}\033[0m".format("При рівні значимості
0.05 рівняння регресії адекватне"))
        break
    else:
        print("\033[1m\033[30m\033[41m{}\033[0m".format("При рівні значимості
0.05 рівняння регресії неадекватне"))
```

## Результат роботи програми

```
Матриця планування з натуралізованими коефіцієнтами X
     X1         X2         X3        X1X2        X1X3        X2X3       X1X2X3      X1X1        X2X2        X3X3
-20.000      5.000      5.000    -100.000    -100.000      25.000    -500.000   400.000      25.000      25.000
-20.000      5.000     10.000    -100.000    -200.000      50.000   -1000.000   400.000      25.000     100.000
-20.000     40.000      5.000    -800.000    -100.000     200.000   -4000.000   400.000    1600.000      25.000
-20.000     40.000     10.000    -800.000    -200.000     400.000   -8000.000   400.000    1600.000     100.000
 30.000      5.000      5.000     150.000     150.000      25.000     750.000   900.000      25.000      25.000
 30.000      5.000     10.000     150.000     300.000      50.000    1500.000   900.000      25.000     100.000
 30.000     40.000      5.000    1200.000     150.000     200.000    6000.000   900.000    1600.000      25.000
 30.000     40.000     10.000    1200.000     300.000     400.000   12000.000   900.000    1600.000     100.000
-38.250     22.500      7.500    -860.625    -286.875     168.750   -6454.688  1463.062     506.250      56.250
 48.250     22.500      7.500    1085.625     361.875     168.750    8142.188  2328.062     506.250      56.250
  5.000     -7.775      7.500     -38.875      37.500     -58.312    -291.562    25.000      60.451      56.250
  5.000     52.775      7.500     263.875      37.500     395.812    1979.062    25.000    2785.201      56.250
  5.000     22.500      3.175     112.500      15.875      71.438     357.188    25.000     506.250      10.081
  5.000     22.500     11.825     112.500      59.125     266.062    1330.312    25.000     506.250     139.831
  5.000     22.500      7.500     112.500      37.500     168.750     843.750    25.000     506.250      56.250
```

Матриця планування Y

| Y1 | Y2 | Y3 |
|---|---|---|
| -250.900 | -246.900 | -244.900 |
| -1577.900 | -1577.900 | -1574.900 |
| -14115.900 | -14118.900 | -14116.900 |
| -25367.400 | -25367.400 | -25363.400 |
| 7807.100 | 7804.100 | 7799.100 |
| 10202.100 | 10195.100 | 10201.100 |
| 30340.100 | 30340.100 | 30335.100 |
| 48187.600 | 48180.600 | 48184.600 |
| -17085.412 | -17085.412 | -17085.412 |
| 42521.900 | 42522.900 | 42525.900 |
| -921.650 | -926.650 | -924.650 |
| 8964.165 | 8963.165 | 8964.165 |
| 2291.836 | 2292.836 | 2289.836 |
| 5606.516 | 5603.516 | 5600.516 |
| 3928.600 | 3931.600 | 3931.600 |

Середні значення відгуку за рядками

-247.567
-1576.900
-14117.233
-25366.067
7803.433
10199.433
30338.433
48184.267
-17085.412
42523.567
-924.317
8963.832
2291.503
5603.516
3930.600

Отримане рівняння регресії

-7.939 + 7.982 * X1 + 8.892 * X2 + 12.074 * X3 + 6.306 * X1X2 + 0.405 * X1X3 +
+ 1.304 * X2X3 + 2.899 * X1X2X3 + 4.698 * X11^2 + 0.097 * X22^2 + 0.896 * X33^2 = y

Експериментальні значення
-246.753
-1576.710
-14116.230
-25365.687
7803.273
10198.649
30338.463
48183.672
-17086.685
42524.546
-924.244
8963.466
2290.635
5604.091
3930.602

**Перевірка за критерієм Кохрена**
Gp = 0.16896551724137934
**Gp=0.169 < GT=0.3346 - Дисперсія однорідна**

**Перевірка значущості коефіцієнтів за критерієм Стьюдента**
Значущі коефіцієнти регресії [-7.939, 7.982, 8.892, 12.074, 6.306, 0.405, 1.304, 2.899, 4.698, 0.097, 0.896]
Незначущі коефіцієнти регресії []

Значення з отриманими коефіцієнтами
-246.753
-1576.710
-14116.230
-25365.687
7803.273
10198.649
30338.463
48183.672
-17086.685
42524.546
-924.244
8963.466
2290.635
5604.091
3930.602

**Перевірка адекватності за критерієм Фішера**
Fp = 1.1603158746685944
**При рівні значимості 0.05 рівняння регресії адекватне**