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

Методи наукових
досліджень

Лабораторна
робота №5

«Проведення трьохфакторного експерименту при
використаннірівняння регресії з квадратичними
членами»

Виконала:
Студентка групи ІВ-93
Баранчук І. М.
Перевірив:
Регіда П. Г.

Київ - 2021 р.

Мета: провести трьохфакторний експеримент і отримати адекватну модель

– рівняння регресії, використовуючи рототабельний композиційний план.

Завдання до лабораторної роботи:

- Ознайомитися з теоретичними відомостями.
- 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.
- 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 вписку в журналі викладача.
- Провести експерименти і аналізуючи значення статистичних перевірок, отримати адекватну модель рівняння регресії. При розрахунках використовувати натуральні значення факторів.
- Зробити висновки по виконаній роботі.

301	-10	50	20	60	-10	5	7,1+9,5*x1+7,9*x2+4,9*x3+1,5*x1*x1+0,9*x2*x2+9,7*x3*x3+1,6*x1*x2+0,1*x1*x3+3,8*x2*x3+4,9*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
from time import perf_counter


counter_1 = 0
counter_2 = 0
counter_3 = 0

for i in range(10):
    m = 3
    n = 15

    # варіант 201
    x1min = -10
```

```
    x1max = 50
    x2min = 20
    x2max = 60
    x3min = -10
    x3max = 5


    def function(X1, X2, X3):
        y = 7.1 + 9.5 * X1 + 7.9 * X2 + 4.9 * X3 + 1.5 * X1 * X1 + 0.9 * X2 *
X2 + 9.7 * X3 * X3 + 1.6 * X1 * X2 + \
            0.1 * X1 * X3 + 3.8 * X2 * X3 + 4.9 * X1 * X2 * X3 + randrange(0,
10) - 5
        return y


    x01 = (x1max + x1min) / 2
    x02 = (x2max + x2min) / 2
    x03 = (x3max + x3min) / 2
    deltax1 = x1max - x01
    deltax2 = x2max - x02
    deltax3 = x3max - x03
    # матриця ПФЕ
    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 = [x1min, x1min, x1min, x1min, x1max, x1max, x1max, x1max, -1.73 *
deltax1 + x01, 1.73 * deltax1 + x01, x01, x01,
          x01, x01, x01]
    x2 = [x2min, x2min, x2max, x2max, x2min, x2min, x2max, x2max, x02, x02, -
1.73 * deltax2 + x02, 1.73 * deltax2 + x02,
          x02, x02, x02]
    x3 = [x3min, x3max, x3min, x3max, x3min, x3max, x3min, x3max, x03, x03,
x03, x03, -1.73 * deltax3 + x03,
          1.73 * deltax3 + x03, x03]
    # заповнення нулями x1x2, x1x3, x1x2x3
    # заповнення нулями x1kv, x2kv, x3kv
    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
    # формуємо список a
    list_for_a = list(zip(x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1kv, x2kv,
x3kv))

    print("Матриця планування з натуралізованими коефіцієнтами X:")
    print("      X1           X2           X3          X1X2         X1X3
X2X3        X1X2X3       X1X1"
          "         X2X2         X3X3")
    for i in range(n):
        print(end=' ')
        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]) for
i in range(m)] for j in range(15)]
    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
    Y_average = []
    for i in range(len(Y)):
        Y_average.append(np.mean(Y[i], axis=0))
    print("Середні значення відгуку за рядками:")
    for i in range(n):
        print("{:.3f}".format(Y_average[i]), end=" ")
    # 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_known(num):
        a = 0
        for j in range(n):
            a += Y_average[j] * list_for_a[j][num - 1] / n
        return a


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


    my = sum(Y_average) / 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))

    det1 = [
        [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)]]

    det2 = [my, find_known(1), find_known(2), find_known(3), find_known(4),
find_known(5), find_known(6), find_known(7),
            find_known(8), find_known(9), find_known(10)]

    beta = solve(det1, det2)
    print("\nОтримане рівняння регресії:")
    print("{:.3f} + {:.3f} * X1 + {:.3f} * X2 + {:.3f} * X3 + {:.3f} * X1X2 +
{:.3f} * X1X3 + {:.3f} * X2X3"
          "+ {:.3f} * X1X2X3 + {:.3f} * X11^2 + {:.3f} * X22^2 + {:.3f} *
X33^2 = ŷ"
          .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("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]), end=" ")
    print("\n------------------------------ Перевірка за критерієм Кохрена -
------------------------------")
    start_1 = perf_counter()
```

```
    Gp = max(dispersions) / sum(dispersions)
    Gt = 0.3346
    print("Gp =", Gp)
    if Gp < Gt:
        print("Дисперсія однорідна")
    else:
        print("Дисперсія неоднорідна")

    counter_1 += perf_counter() - start_1



    print("------------------ Перевірка значущості коефіцієнтів за критерієм
Стьюдента -------------------")
    sb = sum(dispersions) / len(dispersions)
    sbs = (sb / (n * m)) ** 0.5
    F3 = (m - 1) * n
    start_2 = perf_counter()
    coefs1 = []
    coefs2 = []
    d = 11
    res = [0] * 11
    for j in range(11):
        t_pract = 0
        for i in range(15):
            if j == 0:
                t_pract += Y_average[i] / 15
            else:
                t_pract += Y_average[i] * xn[i][j - 1]
            res[j] = beta[j]
        if fabs(t_pract / sbs) < t.ppf(q=0.975, df=F3):
            coefs2.append(beta[j])
            res[j] = 0
            d -= 1
        else:
            coefs1.append(beta[j])
    counter_2 += perf_counter() - start_2
    print("Значущі коефіцієнти регресії:", [round(i, 3) for i in coefs1])
    print("Незначущі коефіцієнти регресії:", [round(i, 3) for i in coefs2])
    y_st = []
    for i in range(n):
        y_st.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("Значення з отриманими коефіцієнтами:")
    for i in range(n):
        print("{:.3f}".format(y_st[i]), end=" ")

    print("\n------------------------- Перевірка адекватності за критерієм
Фішера -------------------------")
    Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(n)]) / (n -
d)
    Fp = Sad / sb
    F4 = n - d
    print("Fp =", Fp)
```

```
    start_3 = 0
    if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
        print("Рівняння регресії адекватне при рівні значимості 0.05")
    else:
        print("Рівняння регресії неадекватне при рівні значимості 0.05")
    counter_3 += perf_counter() - start_3
```

Результати роботи програми:

Матриця планування з натуралізованими коефіцієнтами X:

X1	X2	X3	X1X2	X1X3	X2X3	X1X2X3	X1X1	X2X2	X3X3
-10.000	20.000	-10.000	-200.000	100.000	-200.000	2000.000	100.000	400.000	100.000
-10.000	20.000	5.000	-200.000	-50.000	100.000	-1000.000	100.000	400.000	25.000
-10.000	60.000	-10.000	-600.000	100.000	-600.000	6000.000	100.000	3600.000	100.000
-10.000	60.000	5.000	-600.000	-50.000	300.000	-3000.000	100.000	3600.000	25.000
50.000	20.000	-10.000	1000.000	-500.000	-200.000	-10000.000	2500.000	400.000	100.000
50.000	20.000	5.000	1000.000	250.000	100.000	5000.000	2500.000	400.000	25.000
50.000	60.000	-10.000	3000.000	-500.000	-600.000	-30000.000	2500.000	3600.000	100.000
50.000	60.000	5.000	3000.000	250.000	300.000	15000.000	2500.000	3600.000	25.000
-31.900	40.000	-2.500	-1276.000	79.750	-100.000	3190.000	1017.610	1600.000	6.250
71.900	40.000	-2.500	2876.000	-179.750	-100.000	-7190.000	5169.610	1600.000	6.250
20.000	5.400	-2.500	108.000	-50.000	-13.500	-270.000	400.000	29.160	6.250
20.000	74.600	-2.500	1492.000	-50.000	-186.500	-3730.000	400.000	5565.160	6.250
20.000	40.000	-15.475	800.000	-309.500	-619.000	-12380.000	400.000	1600.000	239.476
20.000	40.000	10.475	800.000	209.500	419.000	8380.000	400.000	1600.000	109.726
20.000	40.000	-2.500	800.000	-50.000	-100.000	-2000.000	400.000	1600.000	6.250

Матриця планування Y:

Y1	Y2	Y3
10226.100	10228.100	10229.100
-3993.900	-4002.900	-4000.900
30871.100	30863.100	30867.100
-10485.900	-10486.900	-10477.900
-42535.900	-42537.900	-42541.900
31525.100	31523.100	31517.100
-135660.900	-135662.900	-135662.900
87678.100	87678.100	87673.100
16250.215	16248.215	16254.215
-20780.435	-20779.435	-20775.435
-294.121	-294.121	-293.121
-10163.041	-10156.041	-10156.041
-56967.964	-56963.964	-56963.964
47619.916	47619.916	47623.916
-6299.525	-6300.525	-6305.525

```
Середні значення відгуку за рядками:
10227.767 -3999.233 30867.100 -10483.567 -42538.567 31521.767 -135662.233 87676.433 16250.882 -20778.435 -293.788 -10158.374 -56965.297 47621.249 -6301.858
Отримане рівняння регресії:
4.257 + 9.578 * X1 + 8.049 * X2 + 4.998 * X3 + 1.599 * X1X2 + 0.096 * X1X3 + 3.794 * X2X3+ 4.900 * X1X2X3 + 1.499 * X11^2 + 0.899 * X22^2 + 9.681 * X33^2 = ŷ
Експериментальні значення:
10227.808 -3999.484 30867.752 -10483.206 -42538.766 31521.276 -135661.822 87676.553 16250.495 -20778.264 -293.190 -10159.188 -56965.743 47621.479 -6301.857
------------------------------ Перевірка за критерієм Кохрена ------------------------------
Gp = 0.15665236051502146
Дисперсія однорідна
------------------ Перевірка значущості коефіцієнтів за критерієм Стьюдента ------------------
Значущі коефіцієнти регресії: [4.257, 9.578, 8.049, 4.998, 1.599, 0.096, 3.794, 4.9, 1.499, 0.899, 9.681]
Незначущі коефіцієнти регресії: []
Значення з отриманими коефіцієнтами:
10227.808 -3999.484 30867.752 -10483.206 -42538.766 31521.276 -135661.822 87676.553 16250.495 -20778.264 -293.190 -10159.188 -56965.743 47621.479 -6301.857
------------------------- Перевірка адекватності за критерієм Фішера -------------------------
Fp = 0.27524000884956057
Рівняння регресії адекватне при рівні значимості 0.05
```

Висновки: в даній лабораторній роботі проведено трьохфакторний експеримент та отримано адекватну модель – рівняння регресії, використовуючи рототабельний композиційний план. Кінцевої мети досягнуто.