Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 6

з дисципліни «Методи наукових досліджень»

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

ВИКОНАВ:
студент 2 курсу
групи ІО-93
Сукач Артем

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Київ – 2021

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

Завдання:

325	-20	15	25	45	10	20	9,9+9,0*x1+6,3*x2+5,3*x3+9,7*x1*x1+0,9*x2*x2+9,5*x3*x3+6,3*x1*x2+0,8*x1*x3+3,7*x2*x3+5,9*x1*x2*x3

Програмний код

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

m = 3
n = 15

x1min = -20
x1max = 15
x2min = 25
x2max = 45
x3min = 10
x3max = 20

x01 = (x1max + x1min) / 2
x02 = (x2max + x2min) / 2
x03 = (x3max + x3min) / 2
deltax1 = x1max - x01
deltax2 = x2max - x02
deltax3 = x3max - x03
```

```
def function(X1, X2, X3):
    y = 9.9 + 9.0 * X1 + 6.3 * X2 + 5.3 * X3 + 9.7 * X1 * X1 + 0.9 * X2 * X2 + 9.5 * X3 * X3 + 6.3 * X1 * X2 + \
        0.8 * X1 * X3 + 3.7 * X2 * X3 + 5.9 * X1 * X2 * X3 + randrange(0, 10) - 5
    return y


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]
x1x2 = [0] * 15
x1x3 = [0] * 15
x2x3 = [0] * 15
```

```
x1x2x3 = [0] * 15
x1kv = [0] * 15
x2kv = [0] * 15
x3kv = [0] * 15
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))

for i in range(len(list_for_a)):
    list_for_a[i] = list(list_for_a[i])
    for j in range(len(list_for_a[i])):
        list_for_a[i][j] = round(list_for_a[i][j], 3)



planning_matrix_x = PrettyTable()
planning_matrix_x.field_names = ['X1', 'X2', 'X3', 'X1X2', 'X1X3', 'X2X3', 'X1X2X3', 'X1X1', 'X2X2', 'X3X3']
print("Матриця планування з натуралізованими коефіцієнтами X:")
planning_matrix_x.add_rows(list_for_a)
print(planning_matrix_x)



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)]

planing_matrix_y = PrettyTable()
planing_matrix_y.field_names = ['Y1', 'Y2', 'Y3']
print("Матриця планування Y:")
```

```
planing_matrix_y.add_rows(Y)
print(planing_matrix_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(15):
    print("{:.3f}".format(Y_average[i]), end=" ")

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(15):
        a += Y_average[j] * list_for_a[j][num - 1] / 15
    return a


def a(first, second):
    a = 0
    for j in range(15):
        a += list_for_a[j][first - 1] * list_for_a[j][second - 1] / 15
    return a


my = sum(Y_average) / 15
```

```
mx = []
for i in range(10):
    number_lst = []
    for j in range(15):
        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))
y_i = [0] * 15
print("Експериментальні значення:")
for k in range(15):
    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(15):
    print("{:.3f}".format(y_i[i]), end=" ")
print("\n\n------------------------------ Перевірка за критерієм Кохрена ------------------------------")
Gp = max(dispersions) / sum(dispersions)
Gt = 0.3346
print("Gp =", Gp)
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")

print("\n------------------ Перевірка значущості коефіцієнтів за критерієм Стьюдента ------------------")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (15 * m)) ** 0.5

F3 = (m - 1) * n
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])
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(15):
    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(15):
    print("{:.3f}".format(y_st[i]), end=" ")


print("\n\n------------------------ Перевірка адекватності за критерієм Фішера ------------------------")
Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(15)]) / (n - d)
Fp = Sad / sb
F4 = n - d
print("Fp =", Fp)
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")
```

Результат роботи програми

```
Матриця планування з натуралізованими коефіцієнтами X:
+----------+------+--------+------------+-----------+---------+-------------+-----------+----------+----------+
|    X1    |  X2  |   X3   |    X1X2    |   X1X3    |  X2X3   |   X1X2X3    |   X1X1    |   X2X2   |   X3X3   |
+----------+------+--------+------------+-----------+---------+-------------+-----------+----------+----------+
|   -20    |  25  |   10   |    -500    |   -200    |   250   |    -5000    |    400    |   625    |   100    |
|   -20    |  25  |   20   |    -500    |   -400    |   500   |   -10000    |    400    |   625    |   400    |
|   -20    |  45  |   10   |    -900    |   -200    |   450   |    -9000    |    400    |   2025   |   100    |
|   -20    |  45  |   20   |    -900    |   -400    |   900   |   -18000    |    400    |   2025   |   400    |
|    15    |  25  |   10   |    375     |    150    |   250   |     3750    |    225    |   625    |   100    |
|    15    |  25  |   20   |    375     |    300    |   500   |     7500    |    225    |   625    |   400    |
|    15    |  45  |   10   |    675     |    150    |   450   |     6750    |    225    |   2025   |   100    |
|    15    |  45  |   20   |    675     |    300    |   900   |    13500    |    225    |   2025   |   400    |
| -32.775  | 35.0 |  15.0  | -1147.125  | -491.625  |  525.0  |  -17206.875 |  1074.201 |  1225.0  |  225.0   |
|  27.775  | 35.0 |  15.0  |  972.125   |  416.625  |  525.0  |  14581.875  |  771.451  |  1225.0  |  225.0   |
|   -2.5   | 17.7 |  15.0  |   -44.25   |   -37.5   |  265.5  |   -663.75   |    6.25   |  313.29  |  225.0   |
|   -2.5   | 52.3 |  15.0  |  -130.75   |   -37.5   |  784.5  |   -1961.25  |    6.25   |  2735.29 |  225.0   |
|   -2.5   | 35.0 |  6.35  |   -87.5    |  -15.875  |  222.25 |   -555.625  |    6.25   |  1225.0  |  40.322  |
|   -2.5   | 35.0 | 23.65  |   -87.5    |  -59.125  |  827.75 |  -2069.375  |    6.25   |  1225.0  |  559.322 |
|   -2.5   | 35.0 |  15.0  |   -87.5    |   -37.5   |  525.0  |   -1312.5   |    6.25   |  1225.0  |  225.0   |
+----------+------+--------+------------+-----------+---------+-------------+-----------+----------+----------+
Матриця планування Y:
+--------------------+--------------------+--------------------+
|         Y1         |         Y2         |         Y3         |
+--------------------+--------------------+--------------------+
|      -26448.1      |      -26450.1      |      -26456.1      |
|      -52283.1      |      -52284.1      |      -52281.1      |
|      -50442.1      |      -50443.1      |      -50445.1      |
|      -99137.1      |      -99141.1      |      -99139.1      |
|      29577.9       |      29579.9       |      29586.9       |
|      55653.9       |      55658.9       |      55651.9       |
|      51295.9       |      51301.9       |      51293.9       |
|      95815.9       |      95806.9       |      95810.9       |
|   -93522.5789375   |   -93523.5789375   |   -93519.5789375   |
|   105717.1960625   |   105717.1960625   |   105713.1960625   |
|  -583.0540000000001 |  -586.0540000000001 |  -589.0540000000001 |
|  -4463.174000000001 |  -4469.174000000001 |  -4469.174000000001 |
|     -1230.06875    |     -1232.06875    |     -1231.06875    |
| -2939.2537500000017 | -2934.2537500000017 | -2940.2537500000017 |
| -2795.4750000000004 | -2797.4750000000004 | -2790.4750000000004 |
+--------------------+--------------------+--------------------+
Середні значення відгуку за рядками:
-26451.433 -52282.767 -50443.433 -99139.100 29581.567 55654.900 51297.233 95811.233 -93521.912 105715.863 -586.054 -4467.174 -1231.069 -2937.920 -2794.475
Отримане рівняння регресії:
0.167 + 9.231 * X1 + 6.714 * X2 + 5.691 * X3 + 6.290 * X1X2 + 0.781 * X1X3 + 3.693 * X2X3+ 5.901 * X1X2X3 + 9.701 * X11^2 + 0.896 * X22^2 + 9.492 * X33^2 = ŷ
Експериментальні значення:
-26451.593 -52283.655 -50443.310 -99139.706 29581.804 55654.408 51297.753 95811.024 -93521.207 105715.650 -585.480 -4467.255 -1231.665 -2936.831 -2794.479

------------------------------ Перевірка за критерієм Кохрена ------------------------------
Gp = 0.14502164502164502
Дисперсія однорідна

------------------ Перевірка значущості коефіцієнтів за критерієм Стьюдента ------------------
Значущі коефіцієнти регресії: [0.167, 9.231, 6.714, 5.691, 6.29, 0.781, 3.693, 5.901, 9.701, 0.896, 9.492]
Незначущі коефіцієнти регресії: []
Значення з отриманими коефіцієнтами:
-26451.593 -52283.655 -50443.310 -99139.706 29581.804 55654.408 51297.753 95811.024 -93521.211 105715.647 -585.480 -4467.255 -1231.660 -2936.827 -2794.479

------------------------- Перевірка адекватності за критерієм Фішера -------------------------
Fp = 0.46352958735320415
Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0
```