

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

ЛАБОРАТОРНА РОБОТА № 4
з дисципліни «Методи наукових досліджень»
на тему «ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ
ВИКОРИСТАННІ РІВНЯННЯ РЕГРЕСІЇ З УРАХУВАННЯМ ЕФЕКТУ
ВЗАЄМОДІЇ»

ВИКОНАВ:
студент 2 курсу
групи ІВ-91
Бойко М. І.
Залікова – 9102

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Лабораторна робота №4

Мета: Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

Завдання:

1. Скласти матрицю планування для повного трьохфакторного експерименту.
2. Провести експеримент, повторивши N раз досліди у всіх точках факторного простору і знайти значення відгуку Y. Знайти значення Y шляхом моделювання випадкових чисел у певному діапазоні відповідно варіанту. Варіанти вибираються за номером в списку в журналі викладача.

$$y_{i\max} = 200 + x_{cp\max}$$

$$y_{i\min} = 200 + x_{cp\min}$$

$$\text{де } x_{cp\max} = \frac{x_{1\max} + x_{2\max} + x_{3\max}}{3}, \quad x_{cp\min} = \frac{x_{1\min} + x_{2\min} + x_{3\min}}{3}$$

3. Знайти коефіцієнти рівняння регресії і записати його.
4. Провести 3 статистичні перевірки – за критеріями Кохрена, Стюдента, Фішера.
5. Зробити висновки по адекватності регресії та значимості окремих коефіцієнтів і записати скореговане рівняння регресії.
6. Написати комп'ютерну програму, яка усе це моделює.

Група: ІВ-91

Номер у списку: 2

Варіант – 102

№ _{варіанта}	x_1		x_2		x_3	
	min	max	min	Max	min	max
102	-10	50	20	60	50	55

Роздруківка коду програми:

```
import math
import numpy as np
from numpy import transpose
from numpy.linalg import solve
from prettytable import PrettyTable
from scipy.stats import f
from scipy.stats import t as t_criterium
from functools import partial
from random import randint

m = 3
N = 8
d = 8

x1_min = -10
x1_max = 50

x2_min = 20
x2_max = 60

x3_min = 50
x3_max = 55

x_max_average = (x1_max + x2_max + x3_max) / 3
x_min_average = (x1_min + x2_min + x3_min) / 3

y_max = int(200 + x_max_average)
y_min = int(200 + x_min_average)

y_matrix = [[randint(y_min, y_max) for _ in range(m)] for _ in range(N)]

average_y = [round(sum(y_matrix[k1]) / m, 3) for k1 in range(N)]

F1 = m - 1
F2 = N
F3 = F1 * F2
F4 = N - d

x1_list = []
x2_list = []
x3_list = []
x1x2_list = []
x1x3_list = []
x2x3_list = []
x1x2x3_list = []

x0_f = [1, 1, 1, 1, 1, 1, 1, 1]
x1_f = [-1, -1, 1, 1, -1, -1, 1, 1]
x2_f = [-1, 1, -1, 1, -1, 1, -1, 1]
x3_f = [-1, 1, 1, -1, 1, -1, -1, 1]

x1x2_f = []
x1x3_f = []
x2x3_f = []
x1x2x3_f = []

for i in range(len(x0_f)):
    x1x2_f.append(x1_f[i] * x2_f[i])
    x1x3_f.append(x1_f[i] * x3_f[i])
    x2x3_f.append(x2_f[i] * x3_f[i])
    x1x2x3_f.append(x1_f[i] * x2_f[i] * x3_f[i])
```

```

x_list = [x0_f, x1_list, x2_list, x3_list, x1x2_list, x1x3_list, x2x3_list,
x1x2x3_list]
x_f_list = [x0_f, x1_f, x2_f, x3_f, x1x2_f, x1x3_f, x2x3_f, x1x2x3_f]

for i in range(len(x0_f)):
    if x1_f[i] == 1:
        x1_list.append(x1_max)
    else:
        x1_list.append(x1_min)

    if x2_f[i] == 1:
        x2_list.append(x2_max)
    else:
        x2_list.append(x2_min)

    if x3_f[i] == 1:
        x3_list.append(x3_max)
    else:
        x3_list.append(x3_min)

    x1x2_list.append(x1_list[i] * x2_list[i])
    x2x3_list.append(x2_list[i] * x3_list[i])
    x1x3_list.append(x1_list[i] * x3_list[i])

    x1x2x3_list.append(x1_list[i] * x2_list[i] * x3_list[i])

dispersion = [round(sum([(k1 - average_y[j]) ** 2) for k1 in y_matrix[j]])) /
m, 3) for j in range(N)]

y_matrix_trans = transpose(y_matrix).tolist()

list_to_solve_1 = list(zip(*x_list))
list_to_solve_2 = x_f_list

list_i_2 = []
for k in range(N):
    S = 0
    for i in range(N):
        S += (list_to_solve_2[k][i] * average_y[i]) / N
    list_i_2.append(round(S, 5))

column_titles = ["X0", "X1", "X2", "X3", "X1X2", "X1X3", "X2X3", "X1X2X3",
"Y1", "Y2", "Y3", "Y", "S^2"]

table = PrettyTable()
cols = x_f_list
[cols.extend(ls) for ls in [y_matrix_trans, [average_y], [dispersion]]]
for i in range(13):
    table.add_column(column_titles[i], cols[i])
print(table, "\n")
print('Рівняння регресії з коефіцієнтами нормованих значень:')
print("y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1x2 + {}*x1x3 + {}*x2x3 +
{}*x1x2x3 \n".format(*list_i_2))

table = PrettyTable()
cols = x_list
[cols.extend(ls) for ls in [y_matrix_trans, [average_y], [dispersion]]]
for i in range(13):
    table.add_column(column_titles[i], cols[i])
print(table, "\n")

list_i_1 = []

```

```

for i in solve(list_to_solve_1, average_y):
    list_i_1.append(round(i, 5))
print('Рівняння регресії з коефіцієнтами натуральних значень:')
print("y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1x2 + {}*x1x3 + {}*x2x3 + {}*x1x2x3".format(*list_i_1))

Gp = max(dispersion) / sum(dispersion)
q = 0.05
q1 = q / F1
fisher = f.ppf(q=1 - q1, dfn=F2, dfd=(F1 - 1) * F2)
Gt = fisher / (fisher + F1 - 1)

if Gp < Gt:
    print("-" * 100)
    print("Дисперсія однорідна\n")
    dispersion_b = sum(dispersion) / N
    dispersion_beta = dispersion_b / (m * N)
    S_beta = math.sqrt(abs(dispersion_beta))
    beta_list = np.zeros(8).tolist()
    for i in range(N):
        beta_list[0] += (average_y[i] * x0_f[i]) / N
        beta_list[1] += (average_y[i] * x1_f[i]) / N
        beta_list[2] += (average_y[i] * x2_f[i]) / N
        beta_list[3] += (average_y[i] * x3_f[i]) / N
        beta_list[4] += (average_y[i] * x1x2_f[i]) / N
        beta_list[5] += (average_y[i] * x1x3_f[i]) / N
        beta_list[6] += (average_y[i] * x2x3_f[i]) / N
        beta_list[7] += (average_y[i] * x1x2x3_f[i]) / N
    t_list = [abs(beta_list[i]) / S_beta for i in range(0, N)]

    for i, j in enumerate(t_list):
        if j >= t_criterium.ppf(q=0.975, df=F3):
            print(f'Значний: {beta_list[i]}')
        else:
            print(f'Незначний: {beta_list[i]}')
            beta_list[i] = 0
            d -= 1
    print("-" * 100)
    print("y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1x2 + {}*x1x3 + {}*x2x3 + {}*x1x2x3".format(*beta_list))

    y_counted = [sum([beta_list[0], *[beta_list[i] * x_list[1:][j][i] for i
in range(N)]]))
        for j in range(N)]
    dispersion_ad = 0
    for i in range(len(y_counted)):
        dispersion_ad += ((y_counted[i] - average_y[i]) ** 2) * m / (N - d)
    Fp = dispersion_ad / dispersion_beta
    fisher = partial(f.ppf, q=1 - 0.05)
    Ft = fisher(dfn=F4, dfd=F3)

    if Ft > Fp:
        print("Рівняння регресії адекватне")
    else:
        print("Рівняння регресії неадекватне")
else:
    print("Дисперсія неоднорідна!")

```

Приклад роботи програми:

X0	X1	X2	X3	X1X2	X1X3	X2X3	X1X2X3	Y1	Y2	Y3	Y	S^2
1	-1	-1	-1	1	1	1	-1	232	245	238	238.333	28.222
1	-1	1	1	-1	-1	1	-1	253	251	249	251.0	2.667
1	1	-1	1	-1	1	-1	-1	223	222	225	223.333	1.556
1	1	1	-1	1	-1	-1	-1	231	252	255	246.0	114.0
1	-1	-1	1	1	-1	-1	1	241	249	234	241.333	37.556
1	-1	1	-1	-1	1	-1	1	255	243	245	247.667	27.556
1	1	-1	-1	-1	-1	1	1	244	246	226	238.667	80.889
1	1	1	1	1	1	1	1	252	231	248	243.667	82.889

Рівняння регресії з коефіцієнтами нормованих значень:

$y = 241.25 + -3.33325 \cdot x_1 + 5.8335 \cdot x_2 + -1.41675 \cdot x_3 + 1.08325 \cdot x_1 x_2 + -3.0 \cdot x_1 x_3 + 1.66675 \cdot x_2 x_3 + 1.5835 \cdot x_1 x_2 x_3$

X0	X1	X2	X3	X1X2	X1X3	X2X3	X1X2X3	Y1	Y2	Y3	Y	S^2
1	-10	20	50	-200	-500	1000	-10000	232	245	238	238.333	28.222
1	-10	60	55	-600	-550	3300	-33000	253	251	249	251.0	2.667
1	50	20	55	1000	2750	1100	55000	223	222	225	223.333	1.556
1	50	60	50	3000	2500	3000	150000	231	252	255	246.0	114.0
1	-10	20	55	-200	-550	1100	-11000	241	249	234	241.333	37.556
1	-10	60	50	-600	-500	3000	-30000	255	243	245	247.667	27.556
1	50	20	50	1000	2500	1000	50000	244	246	226	238.667	80.889
1	50	60	55	3000	2750	3300	165000	252	231	248	243.667	82.889

Рівняння регресії з коефіцієнтами натуральних значень:

$y = 246.66675 + 4.13358 \cdot x_1 + -0.38607 \cdot x_2 + -0.25557 \cdot x_3 + -0.05362 \cdot x_1 x_2 + -0.08223 \cdot x_1 x_3 + 0.01222 \cdot x_2 x_3 + 0.00106 \cdot x_1 x_2 x_3$

Дисперсія однорідна

Значний: 241.24999999999997
Значний: -3.3332499999999996
Значний: 5.8335000000000001
Незначний: -1.4167500000000004
Незначний: 1.0832499999999996
Значний: -3.0000000000000036
Незначний: 1.6667500000000004
Незначний: 1.5835000000000008

$y = 241.24999999999997 + -3.3332499999999996 \cdot x_1 + 5.8335000000000001 \cdot x_2 + 0 \cdot x_3 + 0 \cdot x_1 x_2 + -3.0000000000000036 \cdot x_1 x_3 + 0 \cdot x_2 x_3 + 0 \cdot x_1 x_2 x_3$
Рівняння регресії неадекватне

Process finished with exit code 0