

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 ІВ-91  
Красновський О.В.  
Залікова – 9116

ПЕРЕВІРИВ:  
ас. Регіда П. Г.

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

### Завдання:

1. Ознайомитися з теоретичними відомостями.
2. Вибрати з таблиці варіантів і записати в протокол інтервали значень  $x_1, x_2, x_3$ . Обчислити і записати значення, відповідні кодованим значенням факторів +1; -1; +  $l$ ; -  $l$ ; 0 для  $\bar{x}_1, \bar{x}_2, \bar{x}_3$ .
3. Значення функції відгуку знайти за допомогою підстановки в формулу:

$$y_i = f(x_1, x_2, x_3) + \text{random}(10) - 5,$$

де  $f(x_1, x_2, x_3)$  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.

Таблиця варіантів

№ варіанту	$x_1$		$x_2$		$x_3$		$f(x_1, x_2, x_3)$
	min	max	min	max	min	max	
115	10	50	-20	60	-20	20	$8,8+8,0*x_1+5,4*x_2+8,0*x_3+0,2*x_1*x_1+0,2*x_2*x_2+2,9*x_3*x_3+3,4*x_1*x_2+0,9*x_1*x_3+3,5*x_2*x_3+0,3*x_1*x_2*x_3$

### Лістинг програми

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

```
def main(n, m):
    x1_min = 10
    x1_max = 50
    x2_min = -20
    x2_max = 60
    x3_min = -20
    x3_max = 20
    x01 = (x1_max + x1_min) / 2
    x02 = (x2_max + x2_min) / 2
    x03 = (x3_max + x3_min) / 2

    dx1 = x1_max - x01
```

$$dx2 = x2\_max - x02$$

$$dx3 = x3\_max - x03$$

$$\begin{aligned} xn = & [[-1, -1, -1, +1, +1, +1, -1, +1, +1, +1], \\ & [-1, -1, +1, +1, -1, -1, +1, +1, +1, +1], \\ & [-1, +1, -1, -1, +1, -1, +1, +1, +1, +1], \\ & [-1, +1, +1, -1, -1, +1, -1, +1, +1, +1], \\ & [+1, -1, -1, -1, -1, +1, +1, +1, +1, +1], \\ & [+1, -1, +1, -1, +1, -1, -1, +1, +1, +1], \\ & [+1, +1, -1, +1, -1, -1, -1, +1, +1, +1], \\ & [+1, +1, +1, +1, +1, +1, +1, +1, +1, +1], \\ & [-1.73, 0, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0], \\ & [+1.73, 0, 0, 0, 0, 0, 0, 0, 2.9929, 0, 0], \\ & [0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0], \\ & [0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929, 0], \\ & [0, 0, -1.73, 0, 0, 0, 0, 0, 0, 2.9929], \\ & [0, 0, +1.73, 0, 0, 0, 0, 0, 0, 2.9929], \\ & [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]] \end{aligned}$$

$$x1 = [x1\_min, x1\_min, x1\_min, x1\_min, x1\_max, x1\_max, x1\_max, x1\_max, -1.73 * dx1 + x01, 1.73 * dx1 + x01, x01, x01, x01, x01, x01, x01]$$

$$x2 = [x2\_min, x2\_min, x2\_max, x2\_max, x2\_min, x2\_min, x2\_max, x2\_max, x02, x02, -1.73 * dx2 + x02, 1.73 * dx2 + x02, x02, x02, x02]$$

$$x3 = [x3\_min, x3\_max, x3\_min, x3\_max, x3\_min, x3\_max, x3\_min, x3\_max, x03, x03, x03, -1.73 * dx3 + x03, 1.73 * dx3 + x03, x03]$$

$$x1x2 = [0] * 15$$

$$x1x3 = [0] * 15$$

$$x2x3 = [0] * 15$$

$$x1x2x3 = [0] * 15$$

$$x1kv = [0] * 15$$

$$x2kv = [0] * 15$$

$$x3kv = [0] * 15$$

```
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
```

```
tmp_list_a = list(zip(x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1kv, x2kv, x3kv))
```

```
plan_table = BeautifulTable()
```

```
plan_table.columns.header = ['X1', 'X2', 'X3', 'X1X2', 'X1X3', 'X2X3', 'X1X2X3', 'X1X1',  
'X2X2', 'X3X3']
```

```
print("Матриця планування з натуралізованими коефіцієнтами X:")
```

```
for i in range(len(tmp_list_a)):
```

```
    plan_table.rows.append(tmp_list_a[i])
```

```
print(plan_table)
```

```
def func(X1, X2, X3):
```

```
    y = 8.8 + 8*X1 + 5.4*X2 + 8*X3 + 0.2*X1*X1 + 0.2*X2*X2 + 2.9*X3*X3 + 3.4*X1*X2  
    + 0.9*X1*X3 + 3.5*X2*X3 + 0.3*X1*X2*X3 + random.randint(0, 10) - 5
```

```
    return y
```

```
y = [[func(tmp_list_a[j][0], tmp_list_a[j][1], tmp_list_a[j][2]) for _ in range(m)] for j in  
range(15)]
```

```
plan_y = BeautifulTable()
```

```
plan_y.columns.header = ['y1', 'y2', 'y3']
```

```
print("Матриця планування y:")
```

```
for i in range(len(y)):
```

```
    plan_y.rows.append(y[i])
```

```
print(plan_y)
```

```
aver_y = []
```

```
for i in range(len(y)):
```

```
    aver_y.append(np.mean(y[i], axis=0))
```

```
print("Середні значення ф-цій відгуку:\n{}".format(aver_y))
```

```
disp = []
```

```
for i in range(len(y)):
```

```
    a = 0
```

```
    for k in y[i]:
```

```
        a += (k - np.mean(y[i], axis=0)) ** 2
```

```
    disp.append(a / len(y[i]))
```

```
print("Дисперсії:\n{}".format(disp))
```

```
def finds_value(num):
```

```
    a = 0
```

```
    for j in range(15):
```

```
        a += aver_y[j] * tmp_list_a[j][num - 1] / 15
```

```
    return a
```

```
def a(f, s):
```

```
    a = 0
```

```
    for j in range(15):
```

```
        a += tmp_list_a[j][f - 1] * tmp_list_a[j][s - 1] / 15
```

```
    return a
```

```
my = sum(aver_y) / 15
```

```
mx = []
```

```
for i in range(10):
```

```
    number_lst = []
```

```
    for j in range(15):
```

```

        number_lst.append(tmp_list_a[j][i])

    mx.append(sum(number_lst) / len(number_lst))

determinant1 = [[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)]]

determinant2 = [my, finds_value(1), finds_value(2), finds_value(3), finds_value(4),
finds_value(5), finds_value(6), finds_value(7),
                 finds_value(8), finds_value(9), finds_value(10)]

beta = solve(determinant1, determinant2)

print("Рівняння регресії:")

print("y = {} + {} * X1 + {} * X2 + {} * X3 + {} * X1X2 + {} * X1X3 + {} * X2X3"
      "+ {} * X1X2X3 + {} * X11^2 + {} * X22^2 + {} * X33^2"
      .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] * 15

```

```

for k in range(15):

    y_i[k] = beta[0] + beta[1] * tmp_list_a[k][0] + beta[2] * tmp_list_a[k][1] + beta[3] *
tmp_list_a[k][2] + \

        beta[4] * tmp_list_a[k][3] + beta[5] * tmp_list_a[k][4] + beta[6] * tmp_list_a[k][5] +
beta[7] * \

        tmp_list_a[k][6] + beta[8] * tmp_list_a[k][7] + beta[9] * tmp_list_a[k][8] + beta[10] *
tmp_list_a[k][9]


print("Експериментальні значення:\n{}".format(y_i))


gp = max(displ) / sum(displ)
gt = 0.3346
print("\nПеревірка за критерієм Кохрена\nGp = {}".format(gp))
if gp < gt:
    print("Дисперсії однорідні")
else:
    print("Дисперсії неоднорідні")


sb = sum(displ) / len(displ)
sbs = (sb / (15 * m)) ** 0.5


f3 = (m - 1) * n
sign_coef = []
insign_coef = []
d = 11
res = [0] * 11


for j in range(11):
    t_pract = 0
    for i in range(15):
        if j == 0:

```

```

        t_pract += aver_y[i] / 15
    else:
        t_pract += aver_y[i] * xn[i][j - 1]
    res[j] = beta[j]
    if math.fabs(t_pract / sbs) < t.ppf(q=0.975, df=f3):
        insign_coef.append(beta[j])
        res[j] = 0
        d-=1
    else:
        sign_coef.append(beta[j])
print("\nКритерій Стьюдента:")
print("Значущі коефіцієнти регресії :", [round(i, 3) for i in sign_coef])
print("Незначущі коефіцієнти регресії :", [round(i, 3) for i in insign_coef])
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 коефіцієнтами:\n{}".format(y_st))

print("\nПеревірка адекватності Фішера:")
sad = m * sum([(y_st[i] - aver_y[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("Математична модель адекватна експериментальним даним")
else:
    print("Математична модель неадекватна експериментальним даним ")

```



main(15, 3)

## Результат роботи програми

```
demian@pc:~/mnd/lab6mnd$ python3 lab6.py
Матриця планування з натуралізованими коефіцієнтами X:
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  X1  |  X2  |  X3  |  X1X2 |  X1X3 |  X2X3 |  X1X2X3 |  X1X1 |  X2X2 |  X3X3 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  10  |  -20 |  -20 |  -200 |  -200 |  400  |  4000  |  100  |  400  |  400  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  10  |  -20 |  20  |  -200 |  200  |  -400 |  -4000 |  100  |  400  |  400  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  10  |  60  |  -20 |  600  |  -200 |  -120 |  -12000 |  100  |  3600 |  400  |
|      |      |      |      |      |  0    |         |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  10  |  60  |  20  |  600  |  200  |  1200 |  12000 |  100  |  3600 |  400  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  50  |  -20 |  -20 |  -1000 |  -1000 |  400  |  20000 |  2500 |  400  |  400  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  50  |  -20 |  20  |  -1000 |  1000  |  -400 |  -20000 |  2500 |  400  |  400  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  50  |  60  |  -20 |  3000  |  -1000 |  -120 |  -60000 |  2500 |  3600 |  400  |
|      |      |      |      |      |  0    |         |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  50  |  60  |  20  |  3000  |  1000  |  1200 |  60000  |  2500 |  3600 |  400  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| -4.6 | 20.  |  0.0 | -92.0 | -0.0  |  0.0  |  -0.0  | 21.16 | 400.0 |  0.0  |
|      |  0   |      |      |      |      |         |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 64.6 | 20.  |  0.0 | 1292.0 |  0.0  |  0.0  |  0.0   | 4173. | 400.0 |  0.0  |
|      |  0   |      |      |      |      |         | 16    |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 30.0 | -49  |  0.0 | -1476. |  0.0  | -0.0  |  -0.0  | 900.0 | 2420. |  0.0  |
|      | .2   |      |  0     |      |      |         |      | 64    |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 30.0 | 89.  |  0.0 | 2676.0 |  0.0  |  0.0  |  0.0   | 900.0 | 7956. |  0.0  |
|      | 2    |      |        |      |      |         |      | 64    |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 30.0 | 20.  | -34. | 600.0  | -1038 | -692  | -20760.0 | 900.0 | 400.0 | 1197. |
|      | 0    | 6    |        | .0    | .0    |         |      |      | 16    |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 30.0 | 20.  | 34.6 | 600.0  | 1038. | 692.  | 20760.0  | 900.0 | 400.0 | 1197. |
|      | 0    |      |        | 0     | 0     |         |      |      | 16    |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 30.0 | 20.  |  0.0 | 600.0  |  0.0  |  0.0  |  0.0   | 900.0 | 400.0 |  0.0  |
|      | 0    |      |        |      |      |         |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Матриця планування у:						
+	+	+	+			
	y1		y2		y3	
+	+	+	+	+	+	+
	2815.8		2822.8		2822.8	
+	+	+	+	+	+	+
	-1701.2		-1699.2		-1699.2	
+	+	+	+	+	+	+
	-3789.2		-3791.2		-3792.2	
+	+	+	+	+	+	+
	12494.8		12494.8		12496.8	
+	+	+	+	+	+	+
	4978.8		4981.8		4982.8	
+	+	+	+	+	+	+
	-7698.2		-7703.2		-7699.2	
+	+	+	+	+	+	+
	-9942.2		-9951.2		-9948.2	
+	+	+	+	+	+	+
	36568.8		36575.8		36571.8	
+	+	+	+	+	+	+
	-149.568		-153.568		-153.568	
+	+	+	+	+	+	+
	5936.032		5936.032		5937.032	
+	+	+	+	+	+	+
	-4366.152		-4374.152		-4367.152	
+	+	+	+	+	+	+
	11598.208		11595.208		11603.208	
+	+	+	+	+	+	+
	-3735.436		-3731.436		-3734.436	
+	+	+	+	+	+	+
	15987.564		15992.564		15984.564	

	2653.8		2660.8		2661.8	
+	+	+	+	+	+	+

Середні значення ф-цій відгуку:  
[2820.466666666667, -1699.866666666667, -3790.866666666667, 12495.466666666667, 4981.133333333334, -7700.2, -9947.2, 36572.133333333334, -152.23466666666675, 5936.365333333332, -4369.152, 11598.874666666665, -3733.769333333333, 15988.230666666668, 2658.8]

Дисперсії:  
[10.888888888888888, 0.8888888888888888, 1.5555555555555554, 0.8888888888888888, 2.8888888888888893, 4.666666666666667, 14.0, 8.222222222222222, 1, 3.555555555555556, 0.2222222222222222, 12.666666666666666, 10.888888888888889, 2.888888888888889, 10.888888888888889, 12.666666666666666]

Рівняння регресії:  
 $y = 6.970416368577726 + 8.267099575898667 * X_1 + 5.397253140201164 * X_2 + 8.03825855212773 * X_3 + 3.4000000000000004 * X_1X_2 + 0.89843749999999734 * X_1X_3 + 3.502552083333329 * X_2X_3 + 0.29995312500000015 * X_1X_2X_3 + 0.19548498248786622 * X_1^2 + 0.19981096963546527 * X_2^2 + 2.8978516948181587 * X_3^2$

Експериментальні значення:  
[2820.6905756445603, -1700.0707489370054, -3791.4674036391852, 12494.437938445904, 4981.038516651409, -7700.722807930204, -9948.119462632347, 36570.78587945272, -151.85232879290362, 5937.484609148905, -4369.3545410748875, 11600.57882143089, -3733.513314058924, 15989.476261081592, 2658.789338542827]

Перевірка за критерієм Кохрена  
Gr = 0.14318181818182  
Дисперсії однорідні

Критерій Стьюдента:  
Значущі коефіцієнти регресії : [6.97, 8.267, 5.397, 8.038, 3.4, 0.898, 3.503, 0.3, 0.195, 0.2, 2.898]  
Незначущі коефіцієнти регресії : []  
Значення з коефіцієнтами:  
[2820.6905756445603, -1700.0707489370054, -3791.4674036391852, 12494.437938445904, 4981.038516651409, -7700.722807930204, -9948.119462632347, 36570.78587945272, -151.85232879290362, 5937.484609148905, -4369.3545410748875, 11600.57882143089, -3733.513314058924, 15989.476261081592, 2658.789338542827]

Перевірка адекватності Фішера:  
fr = 1.198386396515704  
Математична модель адекватна експериментальним даним