



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра автоматики та управління в технічних системах

**Лабораторна робота №6**  
**Проведення трьохфакторного експерименту при використанні**  
**рівняння регресії з квадратичними членами**

Виконала

студентка групи IT-91:

Луцай Катерина

Перевірила:

Сокульський О. Є.

Київ 2022

**Мета:** 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  $x_1, x_2, x_3$ . 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.

## Варіант 15

```
import numpy as np
import random

x1_min, x1_max = 10, 50
x2_min, x2_max = -20, 60
x3_min, x3_max = -20, 20

x01 = (x1_min + x1_max)/2
x02 = (x2_min + x2_max)/2
x03 = (x3_min + x3_max)/2

l = 1.73 # значення для зоряних точок плану

n = 14 # кількість точок
с = 11 # кількість коефіцієнтів
m = 2 # кількість випробувань

norm_factors = np.array([[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, 0, 0],
                          [1, 0, 0],
                          [0, -1, 0],
                          [0, 1, 0],
                          [0, 0, -1],
                          [0, 0, 1]])

# генерація значень Y
def generate_y(factors, n, m):
    Y = np.zeros((n, m))
    for r in range(n):
        x1, x2, x3 = factors[r, 0:3]
        for i in range(m):
            Y[r, i] = 8.8 + 8.0 * x1 + 5.4 * x2 + 8.0 * 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
```

```
# інтеракції між факторами
```

```
def interact_factors(factors, expanded=False):  
    full_factors = np.zeros((n, 4)) if not expanded else np.zeros((n, 7))  
    full_factors[:, 0] = factors[:, 0] * factors[:, 1]  
    full_factors[:, 1] = factors[:, 0] * factors[:, 2]  
    full_factors[:, 2] = factors[:, 1] * factors[:, 2]  
    full_factors[:, 3] = factors[:, 0] * factors[:, 1] * factors[:, 2]  
    if expanded:  
        for i in range(3):  
            full_factors[:, 4+i] = np.power(factors[:, i], 2)  
    return full_factors
```

```
# отримання натуральних факторів з нормованих
```

```
def natural(norm_factors):  
    natural = np.zeros_like(norm_factors)  
    natural[:, 0][norm_factors[:, 0] == 1] = x1_max  
    natural[:, 1][norm_factors[:, 1] == 1] = x2_max  
    natural[:, 2][norm_factors[:, 2] == 1] = x3_max  
  
    natural[:, 0][norm_factors[:, 0] == -1] = x1_min  
    natural[:, 1][norm_factors[:, 1] == -1] = x2_min  
    natural[:, 2][norm_factors[:, 2] == -1] = x3_min  
  
    natural[:, 0][norm_factors[:, 0] == 0] = x01  
    natural[:, 1][norm_factors[:, 1] == 0] = x02  
    natural[:, 2][norm_factors[:, 2] == 0] = x03  
  
    natural[:, 0][norm_factors[:, 0] == 1] = 1 * (x1_max - x01) + x01  
    natural[:, 1][norm_factors[:, 1] == 1] = 1 * (x2_max - x02) + x02  
    natural[:, 2][norm_factors[:, 2] == 1] = 1 * (x3_max - x03) + x03  
    natural[:, 0][norm_factors[:, 0] == -1] = -1 * (x1_max - x01) + x01  
    natural[:, 1][norm_factors[:, 1] == -1] = -1 * (x2_max - x02) + x02  
    natural[:, 2][norm_factors[:, 2] == -1] = -1 * (x3_max - x03) + x03  
    return natural
```

```
# регресія за факторами та коефіцієнтами
```

```
def regression(matrix, coefs, n, c):  
    def func(factors, coefs, c):  
        y = coefs[0]  
        for i in range(c):  
            y += coefs[i+1] * factors[i]  
        return y  
    values = np.zeros(n)  
    for f in range(n):  
        values[f] = func(matrix[f], coefs, c)  
    return values
```

```
# вирішення натурального СЛР
```

```
def solve_coef(factors, values):  
    A = np.zeros((c, c))  
    x_mean = np.mean(factors, axis=0)  
    for i in range(1, c):
```

```

        for j in range(1, c):
            A[j, i] = np.mean(factors[:, i-1] * factors[:, j-1])
    A[0, 0] = n
    A[0, 1:] = x_mean
    A[1:, 0] = x_mean
    B = np.mean(np.tile(values, (c, 1)).T * np.column_stack((np.ones(n),
factors))), axis=0)
    coef = np.linalg.solve(A, B)
    return coef

print(f"X1 min: {x1_min}\tX1 max: {x1_max}")
print(f"X2 min: {x2_min}\tX2 max: {x2_max}")
print(f"X3 min: {x3_min}\tX3 max: {x3_max}")

factors = natural(norm_factors)

# Перевірка однорідності дисперсії за критерієм Кохрена
# отримання оптимальної кількості випробувань для однієї комбінації факторів
Gt = 0.3346
for i in range(m, 10):
    rand_y = generate_y(factors, n, i)
    std_y = np.std(rand_y, axis=1)**2
    print(f"Max Y dispersion: {std_y.max()}")
    Gp = std_y.max()/std_y.mean()
    print(f"m = {i} Gp: {Gp}\tGt: {Gt}")
    if Gp < Gt:
        m = i # кількість випробувань
        break

print(f"Mean Y dispersion: {std_y.mean()}")
factors = np.concatenate((factors, interact_factors(factors, True)), axis=1)

# перевірка за критерієм студента
coefs_value = np.zeros(4)
coefs_value[0] = np.mean(rand_y.mean(axis=1))
for i in range(3):
    coefs_value[i+1] = np.mean(rand_y.mean(axis=1) * factors[:, i])
stud_crit = np.abs(coefs_value) / np.sqrt(std_y.mean()/(n*m))
ts = 2.048
sig_ind = np.argwhere(stud_crit > ts)

print(f"All coefs are significant: {len(sig_ind.flatten()) ==
c}\t{sig_ind.flatten()}")

# формування планів
plan = np.concatenate((factors, rand_y), axis=1)
plan = np.concatenate((plan, rand_y.mean(axis=1).reshape(n, 1)), axis=1)
plan = np.concatenate((plan, std_y.reshape(n, 1)), axis=1)
print(f"Natural plan:\n{plan}")

coefs = solve_coef(factors, rand_y.mean(axis=1))
print("Natural coefs", coefs)
significant_coefs = np.zeros_like(coefs)
significant_coefs[sig_ind] = coefs[sig_ind]
print("Significant coefs", significant_coefs)

reg_val = regression(factors, significant_coefs, n, len(sig_ind.flatten()))
print("Natural mean Y - regression Y")

```

```

print(np.column_stack((rand_y.mean(axis=1), reg_val)))

# перевірка критерію Фішера
disp = m/(n - len(sig_ind.flatten())) * np.sum((reg_val -
rand_y.mean(axis=1))**2)
Fp = disp / std_y.mean()
Ft = 3.0
print(f"Fisher crit: {Fp < Ft}\t{Fp}\t{Ft}")

```

## Результат виконання:

X1 min: 10 X1 max: 50

X2 min: -20 X2 max: 60

X3 min: -20 X3 max: 20

Max Y dispersion: 20.25

m = 2 Gp: 4.57258064516129 Gt: 0.3346

Max Y dispersion: 13.555555555555555

m = 3 Gp: 2.271276595744681 Gt: 0.3346

Max Y dispersion: 11.0

m = 4 Gp: 1.543859649122807 Gt: 0.3346

Max Y dispersion: 16.64

m = 5 Gp: 1.6949941792782306 Gt: 0.3346

Max Y dispersion: 14.138888888888888

m = 6 Gp: 1.7080536912751678 Gt: 0.3346

Max Y dispersion: 11.387755102040819

m = 7 Gp: 1.4005019720329872 Gt: 0.3346

Max Y dispersion: 11.484374999999998

m = 8 Gp: 1.493252067914671 Gt: 0.3346

Max Y dispersion: 16.444444444444444

m = 9 Gp: 2.1820734846711907 Gt: 0.3346

Mean Y dispersion: 7.536155202821869

All coefs are significant: False [0 1 2 3]

Natural plan:

```
[[ 5.00000000e+01  6.00000000e+01  2.00000000e+01  3.00000000e+03
   1.00000000e+03  1.20000000e+03  6.00000000e+04  2.50000000e+03
   3.60000000e+03  4.00000000e+02  3.65768000e+04  3.65718000e+04
   3.65738000e+04  3.65768000e+04  3.65708000e+04  3.65728000e+04
   3.65758000e+04  3.65748000e+04  3.65718000e+04  3.65739111e+04
   4.54320988e+00]
```

```
[ 5.00000000e+01  6.00000000e+01 -2.00000000e+01  3.00000000e+03
 -1.00000000e+03 -1.20000000e+03 -6.00000000e+04  2.50000000e+03
  3.60000000e+03  4.00000000e+02 -9.94420000e+03 -9.94320000e+03
 -9.95220000e+03 -9.95220000e+03 -9.95020000e+03 -9.95220000e+03
 -9.95020000e+03 -9.94420000e+03 -9.94220000e+03 -9.94786667e+03
  1.64444444e+01]
```

```
[ 5.00000000e+01 -2.00000000e+01 -2.00000000e+01 -1.00000000e+03
 -1.00000000e+03  4.00000000e+02  2.00000000e+04  2.50000000e+03
  4.00000000e+02  4.00000000e+02  4.98480000e+03  4.98380000e+03
  4.97580000e+03  4.97580000e+03  4.97880000e+03  4.98480000e+03
  4.97880000e+03  4.98080000e+03  4.98380000e+03  4.98080000e+03
  1.20000000e+01]
```

```
[ 1.00000000e+01 -2.00000000e+01 -2.00000000e+01 -2.00000000e+02
 -2.00000000e+02  4.00000000e+02  4.00000000e+03  1.00000000e+02
  4.00000000e+02  4.00000000e+02  2.82180000e+03  2.82280000e+03
  2.81880000e+03  2.82480000e+03  2.82280000e+03  2.82280000e+03
  2.82480000e+03  2.82580000e+03  2.81880000e+03  2.82257778e+03
  5.50617284e+00]
```

[ 1.00000000e+01 6.00000000e+01 2.00000000e+01 6.00000000e+02  
2.00000000e+02 1.20000000e+03 1.20000000e+04 1.00000000e+02  
3.60000000e+03 4.00000000e+02 1.24968000e+04 1.24978000e+04  
1.24918000e+04 1.24908000e+04 1.24888000e+04 1.24888000e+04  
1.24968000e+04 1.24948000e+04 1.24908000e+04 1.24930222e+04  
1.12839506e+01]

[ 5.00000000e+01 -2.00000000e+01 2.00000000e+01 -1.00000000e+03  
1.00000000e+03 -4.00000000e+02 -2.00000000e+04 2.50000000e+03  
4.00000000e+02 4.00000000e+02 -7.69820000e+03 -7.70220000e+03  
-7.69920000e+03 -7.69520000e+03 -7.70220000e+03 -7.70020000e+03  
-7.70320000e+03 -7.70020000e+03 -7.69720000e+03 -7.69975556e+03  
6.02469136e+00]

[ 1.00000000e+01 -2.00000000e+01 2.00000000e+01 -2.00000000e+02  
2.00000000e+02 -4.00000000e+02 -4.00000000e+03 1.00000000e+02  
4.00000000e+02 4.00000000e+02 -1.69920000e+03 -1.70320000e+03  
-1.69720000e+03 -1.69720000e+03 -1.69620000e+03 -1.69520000e+03  
-1.69620000e+03 -1.69520000e+03 -1.69520000e+03 -1.69720000e+03  
6.00000000e+00]

[ 1.00000000e+01 6.00000000e+01 -2.00000000e+01 6.00000000e+02  
-2.00000000e+02 -1.20000000e+03 -1.20000000e+04 1.00000000e+02  
3.60000000e+03 4.00000000e+02 -3.78620000e+03 -3.78320000e+03  
-3.79120000e+03 -3.78420000e+03 -3.78820000e+03 -3.78520000e+03  
-3.78220000e+03 -3.78720000e+03 -3.78420000e+03 -3.78575556e+03  
6.91358025e+00]

[-4.60000000e+00 2.00000000e+01 0.00000000e+00 -9.20000000e+01  
-0.00000000e+00 0.00000000e+00 -0.00000000e+00 2.11600000e+01]

4.00000000e+02 0.00000000e+00 -1.44568000e+02 -1.43568000e+02  
-1.46568000e+02 -1.48568000e+02 -1.52568000e+02 -1.47568000e+02  
-1.46568000e+02 -1.46568000e+02 -1.49568000e+02 -1.47345778e+02  
6.39506173e+00]

[ 6.46000000e+01 2.00000000e+01 0.00000000e+00 1.29200000e+03  
0.00000000e+00 0.00000000e+00 0.00000000e+00 4.17316000e+03  
4.00000000e+02 0.00000000e+00 5.93603200e+03 5.93803200e+03  
5.93803200e+03 5.93603200e+03 5.93703200e+03 5.94303200e+03  
5.94003200e+03 5.93603200e+03 5.93703200e+03 5.93792089e+03  
4.76543210e+00]

[ 3.00000000e+01 -4.92000000e+01 0.00000000e+00 -1.47600000e+03  
0.00000000e+00 -0.00000000e+00 -0.00000000e+00 9.00000000e+02  
2.42064000e+03 0.00000000e+00 -4.36715200e+03 -4.36615200e+03  
-4.36715200e+03 -4.36915200e+03 -4.37315200e+03 -4.36615200e+03  
-4.36815200e+03 -4.36815200e+03 -4.37315200e+03 -4.36870756e+03  
6.46913580e+00]

[ 3.00000000e+01 8.92000000e+01 0.00000000e+00 2.67600000e+03  
0.00000000e+00 0.00000000e+00 0.00000000e+00 9.00000000e+02  
7.95664000e+03 0.00000000e+00 1.16002080e+04 1.16042080e+04  
1.16022080e+04 1.16002080e+04 1.15982080e+04 1.16002080e+04  
1.16012080e+04 1.16002080e+04 1.15972080e+04 1.16004302e+04  
3.72839506e+00]

[ 3.00000000e+01 2.00000000e+01 -3.46000000e+01 6.00000000e+02  
-1.03800000e+03 -6.92000000e+02 -2.07600000e+04 9.00000000e+02  
4.00000000e+02 1.19716000e+03 -3.73243600e+03 -3.72843600e+03  
-3.72743600e+03 -3.73243600e+03 -3.73143600e+03 -3.73543600e+03



-3.73343600e+03 -3.73643600e+03 -3.73343600e+03 -3.73232489e+03  
7.65432099e+00]

[ 3.00000000e+01 2.00000000e+01 3.46000000e+01 6.00000000e+02  
1.03800000e+03 6.92000000e+02 2.07600000e+04 9.00000000e+02  
4.00000000e+02 1.19716000e+03 1.59895640e+04 1.59905640e+04  
1.59935640e+04 1.59945640e+04 1.59865640e+04 1.59865640e+04  
1.59925640e+04 1.59935640e+04 1.59905640e+04 1.59908973e+04  
7.77777778e+00]]

Natural coefs [-4.01246646e-05 4.89924406e+00 4.85013431e+00  
8.00245093e+00

3.40048697e+00 9.00104167e-01 3.49925347e+00 3.00029514e-01  
2.50738584e-01 2.13159799e-01 2.95212847e+00]

Significant coefs [-4.01246646e-05 4.89924406e+00 4.85013431e+00  
8.00245093e+00

0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00]

Natural mean Y - regression Y

[[ 3.65739111e+04 6.96019240e+02]  
[-9.94786667e+03 3.75921202e+02]  
[ 4.98080000e+03 -1.20895421e+01]  
[ 2.82257778e+03 -2.08059304e+02]  
[ 1.24930222e+04 5.00049478e+02]  
[-7.69975556e+03 3.08008495e+02]  
[-1.69720000e+03 1.12038733e+02]  
[-3.78575556e+03 1.79951440e+02]  
[-1.47345778e+02 7.44661233e+01]  
[ 5.93792089e+03 4.13493812e+02]

[-4.36870756e+03 -9.16493263e+01]

[ 1.16004302e+04 5.79609262e+02]

[-3.73232489e+03 -3.29048346e+01]

[ 1.59908973e+04 5.20864770e+02]]

Fisher crit: False 55152821.62905287 3.0

**Висновки:** було проведено трьохфакторний експеримент і отримано адекватну модель – рівняння регресії, використовуючи рототабельний композиційний план.