

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

Методи оптимізації та планування експерименту

ЛАБОРАТОРНА РОБОТА № 5

**«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ
ВИКОРИСТАННІ РІВНЯННЯ РЕГРЕСІЇ З УРАХУВАННЯМ
КВАДРАТИЧНИХ ЧЛЕНІВ(ЦЕНТРАЛЬНИЙ ОРТОГОНАЛЬНИЙ
КОМПОЗИЦІЙНИЙ ПЛАН)»**

ВИКОНАЛА:

студентка II курсу ФІОТ

групи ІО-92

Варіант №211

Карнаухова Анастасія

ПЕРЕВІРИВ:

Доц. Порєв В.М.

Київ – 2021

ЛАБОРАТОРНА РОБОТА № 4

Тема: ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ ВИКОРИСТАННІ РІВНЯННЯ РЕГРЕСІЇ З УРАХУВАННЯМ КВАДРАТИЧНИХ ЧЛЕНІВ(ЦЕНТРАЛЬНИЙ ОРТОГОНАЛЬНИЙ КОМПОЗИЦІЙНИЙ ПЛАН)

Мета: провести трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Знайти рівняння регресії адекватне об'єкту.

Завдання на лабораторну роботу:

1. Взяти рівняння з урахуванням квадратичних членів.
2. Скласти матрицю планування для ОЦКП
3. Провести експеримент у всіх точках факторного простору (знайти значення функції відгуку Y). Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі. Варіанти вибираються по номеру в списку в журналі викладача.

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

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

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

4. Розрахувати коефіцієнти рівняння регресії і записати його.
5. Провести 3 статистичні перевірки.

Варіант завдання:

Код

211	-6	10	-3	5	-4	9
-----	----	----	----	---	----	---

програми:

```
import random as r
import numpy as np
import pprint
from scipy.stats import t, f
import sklearn.linear_model as lm
from functools import partial
from prettytable import PrettyTable

table0 = PrettyTable()
table0.field_names = (["Студент", "Група"])
name = "Карнаухова Анастасія"
group = "ІО-92"
table0.add_row([name, group])
print(table0)
```



```

float(format(x3 ** 2, '.2f'))]
print('Натуралізована матриця: ')
pprint.pprint(x_matrix)

print('Y :')
pprint.pprint(y)

y_avr = np.zeros(n)
for i in range(len(y)):
    for j in range(len(y[0])):
        y_avr[i] += y[i][j] / m
return [x_matrix_norm, x_matrix, y, y_avr]

def find_coefs(x, y):
    skm = lm.LinearRegression(fit_intercept=False) # знаходимо коефіцієнти
    # рівняння регресії
    skm.fit(x, y)
    B = skm.coef_
    print('Коефіцієнти: ')
    print(B)
    return B

def perevirka(x, y, b):
    y_pract = np.zeros(len(y))
    for i in range(len(x)):
        for j in range(len(x[0])):
            y_pract[i] += b[j] * x[i][j]
    print("\nПеревірка:")
    print("\ny - real : ", y)
    print("\ny - found: ", y_pract)

def get_new_y(x, b):
    y_pract = np.zeros(len(y))
    for i in range(len(x)):
        for j in range(len(x[0])):
            y_pract[i] += b[j] * x[i][j]
    return y_pract

def get_cohren_critical(prob, f1, f2):
    f_crit = f.isf((1 - prob) / f2, f1, (f2 - 1) * f1)
    return f_crit / (f_crit + f2 - 1)

def cohren_crit(y, n, m):
    y_var = [np.var(i) for i in y]
    Gp = max(y_var) / sum(y_var)
    Gt = get_cohren_critical(0.95, m-1, n)
    if(Gp < Gt):
        print("\nДисперсії однорідні")
        return True
    else:
        print("\nДисперсії не однорідні")
        return False

fisher_teor = partial(f.ppf, q=1 - 0.05)

student_teor = partial(t.ppf, q=1 - 0.025)

def kriteriy_studenta(x, y, y_aver, n, m, B):
    d = 0
    y_var = [np.var(i) for i in y]

```

```

s_kv_aver = sum(y_var) / n
s_aver = (s_kv_aver/(n * m))** 0.5
b = np.zeros(len(x[0]))
for i in range(len(x[0])):
    for j in range(len(y_aver)):
        b[i] += y_aver[j] * x[j][i] / n
ts = []
for bi in b:
    ts.append(bi/s_aver)
Stud_teor = student_teor(df = (m-1) * n)
for i in range(len(ts)):
    if ts[i] < Stud_teor:
        B[i] = 0
    else:
        d += 1
print("\nКоефіцієнти після перевірки нуля гіпотези: ")
print(B)
return [B, d]

def kriteriy_fishera(m, n, d, new_y_pract, y_avr, y):
    f4 = n - d
    f3 = (m-1)*n
    y_var = [np.var(i) for i in y]
    Sa = (sum(y_var)/ n)
    Sad = m/(n-d)* sum([(new_y_pract[i] - y_avr[i])**2 for i in
range(len(y_avr))])
    pract = Sad/Sa
    teor = fisher_teor(dfn= f4, dfd=f3)
    if pract > teor:
        print("\nПрактичне значення:" , pract)
        print("Теоретичне значення:" , teor)
        print("\nПівняння регресії неадекватне")
        return [False, False]
    else:
        print("\nПівняння регресії адекватне")
        return [True, True]

if __name__ == "__main__":
    odnorid = False
    adekvat = False
    n = 15
    m = 3

    while not adekvat:
        while not odnorid:
            x_matrix_norm, x_matrix, y, y_avr = create_plan_matrix(n, m)

            odnorid = cohren_crit(y, n, m)

            if odnorid == False:
                m+=1
            B = find_coefs(x_matrix, y_avr)
            perevirka(x_matrix, y_avr, B)
            new_B, d = kriteriy_students(x_matrix_norm, y, y_avr, n, m, B)
            new_y_pract = get_new_y(x_matrix, new_B)

            adekvat, odnorid = kriteriy_fishera(m, n, 4, new_y_pract, y_avr, y)

```

Результат роботи програми:

```
+-----+-----+
|      Студент      | Група |
+-----+-----+
| Карнаухова Анастасія | IO-92 |
+-----+-----+

Нормована матриця:
[[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, 1, 1, 1, 1, 1, 1, 1],
 [1, -1.215, 0, 0, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 1.215, 0, 0, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, -1.215, 0, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, 1.215, 0, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, 0, -1.215, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, 0, 1.215, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

Натуралізована матриця:
[[1, -6.0, -3.0, -4.0, 18.0, 24.0, 12.0, -72.0, 36.0, 9.0, 16.0],
 [1, -6.0, -3.0, 9.0, 18.0, -54.0, -27.0, 162.0, 36.0, 9.0, 81.0],
 [1, -6.0, 5.0, -4.0, -30.0, 24.0, -20.0, 120.0, 36.0, 25.0, 16.0],
 [1, -6.0, 5.0, 9.0, -30.0, -54.0, 45.0, -270.0, 36.0, 25.0, 81.0],
 [1, 10.0, -3.0, -4.0, -30.0, -40.0, 12.0, 120.0, 100.0, 9.0, 16.0],
 [1, 10.0, -3.0, 9.0, -30.0, 90.0, -27.0, -270.0, 100.0, 9.0, 81.0],
 [1, 10.0, 5.0, -4.0, 50.0, -40.0, -20.0, -200.0, 100.0, 25.0, 16.0],
 [1, 10.0, 5.0, 9.0, 50.0, 90.0, 45.0, 450.0, 100.0, 25.0, 81.0],
 [1, 5.57, 4.0, 6.5, 22.28, 36.2, 26.0, 144.82, 31.02, 16.0, 42.25],
 [1, 10.43, 4.0, 6.5, 41.72, 67.8, 26.0, 271.18, 108.78, 16.0, 42.25],
 [1, 8.0, 2.79, 6.5, 22.28, 52.0, 18.1, 144.82, 64.0, 7.76, 42.25],
 [1, 8.0, 5.21, 6.5, 41.72, 52.0, 33.9, 271.18, 64.0, 27.2, 42.25],
 [1, 8.0, 4.0, 3.46, 32.0, 27.7, 13.85, 110.8, 64.0, 16.0, 11.99],
 [1, 8.0, 4.0, 9.54, 32.0, 76.3, 38.15, 305.2, 64.0, 16.0, 90.96],
 [1, 8.0, 4.0, 6.5, 32.0, 52.0, 26.0, 208.0, 64.0, 16.0, 42.25]]

Y :
array([[198., 200., 200.],
       [202., 205., 203.],
       [200., 196., 207.],
       [205., 205., 201.],
       [200., 196., 195.],
       [199., 204., 204.],
       [202., 198., 203.],
       [208., 202., 204.],
       [200., 200., 195.],
       [195., 205., 208.],
       [201., 195., 196.],
       [199., 196., 205.],
       [204., 208., 205.],
       [199., 200., 205.],
       [198., 206., 199.]])

Дисперсії однорідні
Коефіцієнти:
[ 1.95452059e+02 -3.98590046e-01 -8.51236492e-04  5.19112977e-01
  1.71269900e-02  2.70940103e-03 -1.96574690e-02 -8.94755497e-04
  8.21587914e-02  1.31457350e-01 -4.80550287e-02]
```

Перевірка:

```
y - real : [199.33333333 203.33333333 201.          203.66666667 197.
202.33333333 201.          204.66666667 198.33333333 202.66666667
197.33333333 200.          205.66666667 201.33333333 201.          ]
```

```
y - found: [199.34349792 203.31432497 201.07515606 203.55993377 197.05692959
202.14963949 201.43895726 203.55674467 199.06329761 203.82032205
199.92024471 202.38301497 201.09308472 199.93444593 200.95707296]
```

Коефіцієнти після перевірки нуль гіпотези:

```
[ 1.95452059e+02  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 8.21587914e-02  1.31457350e-01 -4.80550287e-02]
```

Практичне значення: 6.688717031974071

Теоретичне значення: 2.125558760875511

Рівняння регресії неадекватне

Нормована матриця:

```
[[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, 1, 1, 1, 1, 1, 1, 1],
 [1, -1.215, 0, 0, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 1.215, 0, 0, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, -1.215, 0, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, 1.215, 0, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, 0, -1.215, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, 0, 1.215, 0, 0, 0, 0, 1.4623, 0, 0],
 [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```

Натуралізована матриця:

```
[[1, -6.0, -3.0, -4.0, 18.0, 24.0, 12.0, -72.0, 36.0, 9.0, 16.0],
 [1, -6.0, -3.0, 9.0, 18.0, -54.0, -27.0, 162.0, 36.0, 9.0, 81.0],
 [1, -6.0, 5.0, -4.0, -30.0, 24.0, -20.0, 120.0, 36.0, 25.0, 16.0],
 [1, -6.0, 5.0, 9.0, -30.0, -54.0, 45.0, -270.0, 36.0, 25.0, 81.0],
 [1, 10.0, -3.0, -4.0, -30.0, -40.0, 12.0, 120.0, 100.0, 9.0, 16.0],
 [1, 10.0, -3.0, 9.0, -30.0, 90.0, -27.0, -270.0, 100.0, 9.0, 81.0],
 [1, 10.0, 5.0, -4.0, 50.0, -40.0, -20.0, -200.0, 100.0, 25.0, 16.0],
 [1, 10.0, 5.0, 9.0, 50.0, 90.0, 45.0, 450.0, 100.0, 25.0, 81.0],
 [1, 5.57, 4.0, 6.5, 22.28, 36.2, 26.0, 144.82, 31.02, 16.0, 42.25],
 [1, 10.43, 4.0, 6.5, 41.72, 67.8, 26.0, 271.18, 108.78, 16.0, 42.25],
 [1, 8.0, 2.79, 6.5, 22.28, 52.0, 18.1, 144.82, 64.0, 7.76, 42.25],
 [1, 8.0, 5.21, 6.5, 41.72, 52.0, 33.9, 271.18, 64.0, 27.2, 42.25],
 [1, 8.0, 4.0, 3.46, 32.0, 27.7, 13.85, 110.8, 64.0, 16.0, 11.99],
 [1, 8.0, 4.0, 9.54, 32.0, 76.3, 38.15, 305.2, 64.0, 16.0, 90.96],
 [1, 8.0, 4.0, 6.5, 32.0, 52.0, 26.0, 208.0, 64.0, 16.0, 42.25]]
```

Y :

```
array([[202., 202., 196.],
       [208., 203., 208.],
       [199., 205., 197.],
       [207., 197., 202.],
       [199., 207., 208.],
       [197., 197., 208.],
       [206., 207., 204.],
       [206., 199., 202.],
       [195., 196., 205.]])
```

```
[199., 197., 207.],  
[206., 207., 203.],  
[203., 205., 202.],  
[204., 198., 204.],  
[201., 204., 205.],  
[196., 197., 202.]])
```

Дисперсії однорідні

Коефіцієнти:

```
[ 2.00801416e+02  8.21950784e-02  1.58985212e-01 -1.64106867e-01  
 1.86853417e-02 -3.94315361e-02 -2.61071080e-02  3.25807875e-03  
 1.77353829e-02 -9.51607592e-02  5.55545643e-02]
```

Перевірка:

```
y - real : [200.          206.33333333 200.33333333 202.          204.66666667  
200.66666667 205.66666667 202.33333333 198.66666667 201.  
205.33333333 203.33333333 202.          203.33333333 198.33333333]
```

```
y - found: [200.00073012 206.33461498 200.31412184 201.89982633 204.7031889  
200.80227312 205.74016805 202.51251492 200.98518699 202.29265577  
201.94481439 200.84207501 201.31078882 202.78275814 201.53428262]
```

Коефіцієнти після перевірки нуль гіпотези:

```
[ 2.00801416e+02  0.00000000e+00  0.00000000e+00  0.00000000e+00  
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00  
 1.77353829e-02 -9.51607592e-02  5.55545643e-02]
```

Практичне значення: 3.090475427316699

Теоретичне значення: 2.125558760875511

Рівняння регресії неадекватне

Висновок:

Проробивши лабораторну роботу, було проведено трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Та знайдено рівняння регресії адекватне об'єкту. У ході виконання лабораторної роботи проблем не виникло. Результати виконання лабораторної висвітлені на роздруківках.