

Національний технічний університет України «Київський політехнічний
інститут імені 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 (центральный
ортогональний композиційний план)»

Виконав:
студен групи ІВ-82
Мотора В. С.
Залікова книжка No ІВ-8217
Номер у списку - 015
Перевірів Регіда П.Г.

Київ
2020 р.

Мета:

Провести трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Знайти рівняння регресії, яке буде адекватним для опису об'єкту.

Варіант 232

-9	8	0	4	-2	7
----	---	---	---	----	---

Результат виконання

X_1	X_2	X_3	X_{12}	X_{13}	X_{23}	X_{123}	X_1^2	X_2^2	X_3^2	Y_1	Y_2	Y_3	Y_{cp}	S
-9	0	-2	0	18	0	0	81	0	4	202	197	198	199.00	4.67
-9	0	7	0	-63	0	0	81	0	49	204	201	198	201.00	6.00
-9	4	-2	-36	18	-8	72	81	16	4	201	204	201	202.00	2.00
8	0	-2	0	-16	0	0	64	0	4	201	206	202	203.00	4.67
8	4	-2	32	-16	-8	-64	64	16	4	206	196	196	199.33	22.22
8	0	7	0	56	0	0	64	0	49	206	196	202	201.33	16.89
-9	4	7	-36	-63	28	-252	81	16	49	201	203	196	200.00	8.67
8	4	7	32	56	28	224	64	16	49	204	205	199	202.67	6.89
-10.83	2.00	2.50	-21.66	-27.07	5.00	-54.14	117.23	4.00	6.25	200	197	206	201.00	14.00
9.83	2.00	2.50	19.66	24.57	5.00	49.14	96.58	4.00	6.25	198	202	206	202.00	10.67
-0.50	-0.43	2.50	0.22	-1.25	-1.08	0.54	0.25	0.18	6.25	197	198	198	197.67	0.22
-0.50	4.43	2.50	-2.21	-1.25	11.07	-5.54	0.25	19.62	6.25	197	204	200	200.33	8.22
-0.50	2.00	-2.97	-1.00	1.48	-5.94	2.97	0.25	4.00	8.81	206	200	199	201.67	9.56
-0.50	2.00	7.97	-1.00	-3.98	15.94	-7.97	0.25	4.00	63.48	203	196	197	198.67	9.56
-0.50	2.00	2.50	-1.00	-1.25	5.00	-2.50	0.25	4.00	6.25	201	198	197	198.67	2.89

Система адекватна при рівнянні з лінійними коефіцієнтами.

Критерій Кохрена		Критерій Стюдента		Критерій Фішера
$b_0 =$	200.420	$b_1 =$	0.060	$t = 2.04$
$G_p = 0.17$		$t_0 = 462.16$	$t_1 = -223.84$	$F_p = 1.05$
$b_2 =$	0.133	$b_3 =$	-0.040	
$G_t = 0.33$		$t_2 = 925.22$	$t_3 = 1154.04$	$F_t = 2.09$

Лістинг програми:

```
def first(y,m):
```

```
    average = []
```

```
    dispersion = []
```

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

```
        average.append(sum(y[i]) / m)
```

```
        dispersion_current = 0
```

```
        for k in range(m):
```

```
            dispersion_current += (average[i] - y[i][k]) ** 2
```

```
        dispersion.append(dispersion_current / m)
```

```
    return average, dispersion
```

```
def kohren(dispertion,f1,f2):
```

```

Gp = max(disperction) / sum(disperction)

Gt = Criteries.get_cohren_value(f2, f1, 0.05)

return Gp, Gt

def add_y(y,m):

    m += 1

    for i in range(15):

        y[i].append(randint(ymin,ymax))

    return y,m

def start():

    y = [[randint(ymin,ymax) for i in range(m)] for j in range(15)]

    average, disperction = first(y,m)

    return y, average,disperction

def get_nat_table(x_k):

    x_n = []

    for i in range(len(x_k)):

        temp = [1,]

        for j in range(3):

            if j == 0:

                if x_k[i][j] == -1:

                    temp.append(x_nat['x1min'])

                else:

                    temp.append(x_nat['x1max'])

            if j == 1:

                if x_k[i][j] == -1:

                    temp.append(x_nat['x2min'])

                else:

                    temp.append(x_nat['x2max'])

            if j == 2:

                if x_k[i][j] == -1:

                    temp.append(x_nat['x3min'])

                else:

                    temp.append(x_nat['x3max'])

        x_n.append(temp)

```

```

x_n[i].append(temp[1] * temp[2])
x_n[i].append(temp[1] * temp[3])
x_n[i].append(temp[2] * temp[3])
x_n[i].append(temp[1] * temp[2] * temp[3])
x_n[i].append(temp[1]**2)
x_n[i].append(temp[2]**2)
x_n[i].append(temp[3]**2)

```

```
l = 1.215
```

```
x01 = (x_nat['x1min'] + x_nat['x1max']) / 2
```

```
x02 = (x_nat['x2min'] + x_nat['x2max']) / 2
```

```
x03 = (x_nat['x3min'] + x_nat['x3max']) / 2
```

```
deltax1 = x_nat['x1max'] - x01
```

```
deltax2 = x_nat['x2max'] - x02
```

```
deltax3 = x_nat['x3max'] - x03
```

```
x_zor = [[1, -l*deltax1+x01, x02, x03],
```

```
         [1, l*deltax1+x01, x02, x03],
```

```
         [1, x01, -l*deltax2+x02, x03],
```

```
         [1, x01, l*deltax2+x02, x03],
```

```
         [1, x01, x02, -l*deltax3+x03],
```

```
         [1, x01, x02, l*deltax3+x03],
```

```
         [1, x01, x02, x03]]
```

```
for i in range(7):
```

```
    x_zor[i].append(x_zor[i][1] * x_zor[i][2])
```

```
    x_zor[i].append(x_zor[i][1] * x_zor[i][3])
```

```
    x_zor[i].append(x_zor[i][2] * x_zor[i][3])
```

```
    x_zor[i].append(x_zor[i][1] * x_zor[i][2] * x_zor[i][3])
```

```
    x_zor[i].append(x_zor[i][1] ** 2)
```

```
    x_zor[i].append(x_zor[i][2] ** 2)
```

```
    x_zor[i].append(x_zor[i][3] ** 2)
```

```
x_n += x_zor
```

```
return x_n
```

```
def coef_with_inter(average, x_n):
```

```
coefs_nat = lstsq(x_n, average, rcond=None)[0]
```

```
return coefs_nat
```

```
def fisher(m, d, sum_st, y_average, S2b, f1, f2):
```

```
    Sad = 0
```

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

```
        Sad += (sum_st[i] - y_average[i]) ** 2
```

```
    Sad *= m / (15 - d)
```

```
    Fp = Sad / S2b
```

```
    f3 = f1 * f2
```

```
    f4 = f2 - d
```

```
    Ftable = Criteries.get_fisher_value(f3, f4, 0.05)
```

```
    return Fp, Ftable
```

Висновок:

Був проведений трьохфакторний експеримент з урахуванням квадратичних членів з використанням центрального ортогонального композиційного плану.

Створена програма працює!