

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

Лабораторна робота №4
З дисципліни «Методи наукових досліджень»
Проведення трьохфакторного експерименту
при використанні рівняння регресії з урахуванням ефекту взаємодії

ВИКОНАВ:
Студент II курсу ФІОТ
Групи ІВ-91
Липчак Дмитро Олександрович
Номер заліковки: 9118
Номер у списку: 17

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Київ 2021 р.

Мета роботи

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

Завдання на лабораторну роботу

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.

$$y_{i\max} = 200 + x_{\text{ср}\max}$$

$$y_{i\min} = 200 + x_{\text{ср}\min}$$

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

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.

Виконання роботи

№ _{варіанта}	x_1		x_2		x_3	
	min	max	min	Max	min	max
117	-10	50	-20	60	-20	5

Код програми

```
import random
import numpy as np
from numpy.linalg import solve
from pprint import pprint
from scipy.stats import f, t

N = 8
m = 8
x1_min = -10
x1_max = 50
x2_min = -20
x2_max = 60
x3_min = -20
x3_max = 5

print("x1_min = "+str(x1_min), "x2_min = "+str(x2_min), "x3_min = "+str(x3_min))
print("x1_max = "+str(x1_max), "x2_max = "+str(x2_max), "x3_max = "+str(x3_max))
```

```

print("")

y_min = 200 + round((x1_min + x2_min + x3_min)/3)
y_max = 200 + round((x1_max + x2_max + x3_max)/3)
print("y_min = "+str(y_min), "y_max = "+str(y_max))
print("")

mp = [[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]]
print("Матриця планування експерименту")
pprint(mp)
print("")

x1x2_norm = [0,0,0,0,0,0,0,0]
x1x3_norm = [0,0,0,0,0,0,0,0]
x2x3_norm = [0,0,0,0,0,0,0,0]
x1x2x3_norm = [0,0,0,0,0,0,0,0]
for i in range(N):
    x1x2_norm[i] = mp[1][i] * mp[2][i]
    x1x3_norm[i] = mp[1][i] * mp[3][i]
    x2x3_norm[i] = mp[2][i] * mp[3][i]
    x1x2x3_norm[i] = mp[1][i] * mp[2][i] * mp[3][i]
print("x1x2_norm= "+str(x1x2_norm), "x1x3_norm= "+str(x1x3_norm),
      "\nx2x3_norm= "+str(x2x3_norm), "x1x2x3_norm= "+str(x1x2x3_norm))
print("")

y1 = [random.randint(int(y_min), int(y_max)) for i in range(m)]
y2 = [random.randint(int(y_min), int(y_max)) for i in range(m)]
y3 = [random.randint(int(y_min), int(y_max)) for i in range(m)]

mat_y = [[y1[0], y2[0], y3[0]],
          [y1[1], y2[1], y3[1]],
          [y1[2], y2[2], y3[2]],
          [y1[3], y2[3], y3[3]],
          [y1[4], y2[4], y3[4]],
          [y1[5], y2[5], y3[5]],
          [y1[6], y2[6], y3[6]],
          [y1[7], y2[7], y3[7]]]

print("Матриця планування Y")
pprint(mat_y)
print("")

x0 = [1, 1, 1, 1, 1, 1, 1, 1]
x1 = [-10, -10, 50, 50, -10, -10, 50, 50]
x2 = [-20, 60, -20, 60, -20, 60, -20, 60]
x3 = [-20, 5, 5, -20, 5, -20, -20, 5]

x1x2 = [0,0,0,0,0,0,0,0]
x1x3 = [0,0,0,0,0,0,0,0]
x2x3 = [0,0,0,0,0,0,0,0]
x1x2x3 = [0,0,0,0,0,0,0,0]

for i in range(N):
    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]

aver_y = []
for i in range(len(mat_y)):
    aver_y.append(np.mean(mat_y[i], axis=0))

```

```

b_list = [mp[0], mp[1], mp[2], mp[3], x1x2_norm, x1x3_norm, x2x3_norm,
x1x2x3_norm]
a_list = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3))

print("Матриця планування X")
pprint(a_list)
print("")

ai = [round(i, 3) for i in solve(a_list, aver_y)]
print("Рівняння регресії: \n" "y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1x2 +
{}*x1x3 + {}*x2x3 + {}*x1x2x3".format(ai[0],

ai[1],

ai[2],

ai[3],

ai[4],

ai[5],

ai[6],

ai[7]))
print("")

bi = []
for j in range(N):
    S = 0
    for i in range(N):
        S += (b_list[j][i] * aver_y[i]) / N
    bi.append(round(S, 3))
print("Рівняння регресії для нормованих факторів: \n" "y = {} + {}*x1 + {}*x2
+ {}*x3 + {}*x1x2 + {}*x1x3 +
{}*x2x3 + {}*x1x2x3".format(bi[0], bi[1], bi[2], bi[3], bi[4], bi[5],
bi[6], bi[7]))
print("")

print("Перевірка за Кохреном")
print("")

print("Середні значення відгуку за рядками:", "\n", +aver_y[0], aver_y[1],
aver_y[2], aver_y[3],
aver_y[4], aver_y[5], aver_y[6], aver_y[7])
print("")

disp = []
for i in range(len(mat_y)):
    a = 0
    for j in mat_y[i]:
        a += (j - np.mean(mat_y[i], axis=0)) ** 2
    disp.append(a / len(mat_y[i]))
Gp = max(disp) / sum(disp)
Gt = 0.5157

if Gp < Gt:
    print("\033[1m\033[30m\033[43m{}\033[0m".format("Дисперсія однорідна"))
else:
    print("\033[1m\033[30m\033[41m{}\033[0m".format("Дисперсія неоднорідна"))
print("")

print("Перевірка за Стьюдентом")

```

```

sb = sum(displ) / len(displ)
sbs = (sb / (m * 3)) ** 0.5

t_list = [abs(bi[i]) / sbs for i in range(0, 8)]

d = 0
res = [0,0,0,0,0,0,0,0]
coef_1 = []
coef_2 = []
mm = 3
F3 = (mm - 1) * N

for i in range(N):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coef_2.append(bi[i])
        res[i] = 0
    else:
        coef_1.append(bi[i])
        res[i] = bi[i]
        d += 1

print("Значущі коефіцієнти регресії:", coef_1)
print("Незначущі коефіцієнти регресії:", coef_2)

y_st = []
for i in range(N):
    y_st.append(res[0] + res[1] * mp[1][i] + res[2] * mp[2][i] + res[3] *
mp[3][i] + res[4] * x1x2_norm[i] \
        + res[5] * x1x3_norm[i] + res[6] * x2x3_norm[i] + res[7] *
x1x2x3_norm[i])
print("Значення з отриманими коефіцієнтами:", y_st)
print("")

print("Перевірка за Фішером")
S_ad = mm * sum([(y_st[i] - aver_y[i]) ** 2 for i in range(8)]) / (N - d)
Fp = S_ad / sb
F4 = N - d
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("\033[1m\033[30m\033[42m{}\033[0m".format("При рівні значимості
0.05 рівняння регресії адекватне"))
else:
    print("\033[1m\033[30m\033[41m{}\033[0m".format("При рівні значимості
0.05 рівняння регресії неадекватне"))

```

Результат роботи програми

```
C:\Users\38096\anaconda3\python.exe "C:/Z Крі/А МНД/Лаб/Лаб 4/main.py"
```

```
x1_min = -10 x2_min = -20 x3_min = -20
```

```
x1_max = 50 x2_max = 60 x3_max = 5
```

```
y_min = 183 y_max = 238
```

Матриця планування експерименту

```
[[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]]
```

```
x1x2_norm= [1, -1, -1, 1, 1, -1, -1, 1] x1x3_norm= [1, -1, 1, -1, -1, 1, -1, 1]
```

```
x2x3_norm= [1, 1, -1, -1, -1, -1, 1, 1] x1x2x3_norm= [-1, -1, -1, -1, 1, 1, 1, 1]
```

Матриця планування Y

```
[[235, 220, 232],  
 [230, 211, 186],  
 [235, 196, 233],  
 [190, 229, 211],  
 [187, 187, 218],  
 [227, 231, 235],  
 [183, 215, 198],  
 [200, 190, 236]]
```

Матриця планування X

```
[(1, -10, -20, -20, 200, 200, 400, -4000),  
 (1, -10, 60, 5, -600, -50, 300, -3000),  
 (1, 50, -20, 5, -1000, 250, -100, -5000),  
 (1, 50, 60, -20, 3000, -1000, -1200, -60000),  
 (1, -10, -20, 5, 200, -50, -100, 1000),  
 (1, -10, 60, -20, -600, 200, -1200, 12000),  
 (1, 50, -20, -20, -1000, -1000, 400, 20000),  
 (1, 50, 60, 5, 3000, 250, 300, 15000)]
```

Рівняння регресії:

```
y = 207.556 + 0.146*x1 + 0.085*x2 + -0.864*x3 + -0.004*x1x2 + 0.031*x1x3 + 0.002*x2x3 + -0.0*x1x2x3
```

Рівняння регресії для нормованих факторів:

```
y = 213.125 + -3.458*x1 + 1.542*x2 + -4.042*x3 + -1.875*x1x2 + 9.375*x1x3 + -1.792*x2x3 + -4.208*x1x2x3
```

Перевірка за Кохреном

Середні значення відгуку за рядками:

```
229.0 209.0 221.33333333333334 210.0 197.33333333333334 231.0 198.66666666666666 208.66666666666666
```

Дисперсія однорідна

Перевірка за Стьюдентом

Значущі коефіцієнти регресії: [213.125, 9.375]

Незначущі коефіцієнти регресії: [-3.458, 1.542, -4.042, -1.875, -1.792, -4.208]

Значення з отриманими коефіцієнтами: [222.5, 203.75, 222.5, 203.75, 203.75, 222.5, 203.75, 222.5]

Перевірка за Фішером

При рівні значимості 0.05 рівняння регресії адекватне

Висновок

У ході виконання лабораторної роботи я провів трьофакторний експеримент та знайшов рівняння регресії адекватне об'єкту.

Результати роботи програми наведені вище. Під час виконання роботи проблем не виникло.