

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №5
з дисципліни МОПЕ
на тему:

Проведення трьохфакторного експерименту при
використанні рівняння регресії з урахуванням квадратичних
членів (центральний ортогональний композиційний план)

Виконав:
студент групи ІВ-93
Підгайний Данило
Варіант: 318

Перевірив:
Регіда П.Г

Київ 2021

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

Варіант

318	-2	3	-8	9	-10	5
-----	----	---	----	---	-----	---

Код програми

```
import random
import numpy as np
from scipy.stats import f, t
from sklearn import linear_model

m = 3
n = 15
# варіант 18
x1_min = -2
x1_max = 3
x2_min = -8
x2_max = 9
x3_min = -10
x3_max = 5

# максимальне та мінімальне значення
y_max = 200 + (x1_max + x2_max + x3_max) / 3
y_min = 200 + (x1_min + x2_min + x3_min) / 3

# матриця ПФЕ
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, -1.215, 1.215, 0, 0, 0, 0],
      [-1, -1, 1, 1, -1, -1, 1, 1, 1, 0, 0, -1.215, 1.215, 0, 0],
      [-1, 1, -1, 1, -1, 1, -1, 1, 1, 0, 0, 0, 0, -1.215, 1.215]]

x1x2_norm, x1x3_norm, x2x3_norm, x1x2x3_norm, x1kv_norm, x2kv_norm,
x3kv_norm = [0] * n, [0] * n, [0] * n, [0] * n, \
[0] * n, [0] * n, [0] * n

for i in range(n):
    x1x2_norm[i] = xn[1][i] * xn[2][i]
    x1x3_norm[i] = xn[1][i] * xn[3][i]
    x2x3_norm[i] = xn[2][i] * xn[3][i]
    x1x2x3_norm[i] = xn[1][i] * xn[2][i] * xn[3][i]
    x1kv_norm[i] = round(xn[1][i] ** 2, 3)
    x2kv_norm[i] = round(xn[2][i] ** 2, 3)
    x3kv_norm[i] = round(xn[3][i] ** 2, 3)
```

```

# заповнення у(генерація)
Y_matrix = [[random.randint(int(y_min), int(y_max)) for i in range(m)] for
j in range(n)]
# вивід даних за допомогою цикла
print("Матриця планування у:")
for i in range(15):
    print(Y_matrix[i])

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

delta_x1 = x1_max - x01
delta_x2 = x2_max - x02
delta_x3 = x3_max - x03
x0 = [1] * n
x1 = [-4, -4, -4, -4, 4, 4, 4, 4, -1.215 * delta_x1 + x01, 1.215 * delta_x1
+ x01, x01, x01, x01, x01, x01]
x2 = [-10, -10, 4, 4, -10, -10, 4, 4, x02, x02, -1.215 * delta_x2 + x02,
1.215 * delta_x2 + x02, x02, x02, x02]
x3 = [-5, 6, -5, 6, -5, 6, -5, 6, x03, x03, x03, x03, -1.215 * delta_x3 +
x03, 1.215 * delta_x3 + x03, x03]
# заповнення нулями x1x2, x1x3, x1x2x3
x1x2, x1x3, x2x3, x1x2x3 = [0] * n, [0] * n, [0] * n, [0] * n
# заповнення нулями x1kv, x2kv, x3kv
x1kv, x2kv, x3kv = [0] * 15, [0] * 15, [0] * 15

for i in range(n):
    x1x2[i] = round(x1[i] * x2[i], 3)
    x1x3[i] = round(x1[i] * x3[i], 3)
    x2x3[i] = round(x2[i] * x3[i], 3)
    x1x2x3[i] = round(x1[i] * x2[i] * x3[i], 3)
    x1kv[i] = round(x1[i] ** 2, 3)
    x2kv[i] = round(x2[i] ** 2, 3)
    x3kv[i] = round(x3[i] ** 2, 3)
# середні у
Y_average = []
for i in range(len(Y_matrix)):
    Y_average.append(np.mean(Y_matrix[i], axis=0))
    Y_average = [round(i,3) for i in Y_average]
# формуємо списки b і a
list_for_b = list(zip(xn[0], xn[1], xn[2], xn[3], x1x2_norm, x1x3_norm,
x2x3_norm, x1x2x3_norm, x1kv_norm,
x2kv_norm, x3kv_norm))
list_for_a = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3, x1kv, x2kv,
x3kv))
# вивід матриці планування X
print("Матриця планування з нормованими коефіцієнтами X:")
for i in range(15):
    print(list_for_b[i])

skm = linear_model.LinearRegression(fit_intercept=False)
skm.fit(list_for_b, Y_average)
b = skm.coef_
b = [round(i, 3) for i in b]

print("Рівняння регресії зі знайденими коефіцієнтами: \n" "y = {} + {}*x1 +
{}*x2 + {}*x3 + {}*x1x2 + {}*x1x3 + "
" {}*x2x3 + {}*x1x2x3 {}*x1^2 + {}*x2^2 + {}*x3^2".format(b[0], b[1],
b[2], b[3], b[4], b[5], b[6], b[7], b[8],
b[9],

```

```

b[10]))

print("Перевірка за критерієм Кохрена")
print("Середні значення відгуку за рядками:", "\n", +Y_average[0],
Y_average[1], Y_average[2], Y_average[3],
      Y_average[4], Y_average[5], Y_average[6], Y_average[7])
# розрахунок дисперсій
dispersions = []
for i in range(len(Y_matrix)):
    a = 0
    for k in Y_matrix[i]:
        a += (k - np.mean(Y_matrix[i], axis=0)) ** 2
    dispersions.append(a / len(Y_matrix[i]))
# експериментально
Gp = max(dispersions) / sum(dispersions)
# теоретично
Gt = 0.3346
# перевірка однорідності дисперсій
if Gp < Gt:
    print("Дисперсія однорідна")
else:
    print("Дисперсія неоднорідна")

# критерій Стьюдента
print("Перевірка значущості коефіцієнтів за критерієм Стьюдента")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (n * m)) ** 0.5

t_list = [abs(b[i]) / sbs for i in range(0, 11)]

d = 0
res = [0] * 11
coef_1 = []
coef_2 = []
F3 = (m - 1) * n
# перевірка значущості коефіцієнтів(scipy)
for i in range(n-4):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coef_2.append(b[i])
        res[i] = 0
    else:
        coef_1.append(b[i])
        res[i] = b[i]
        d += 1

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

# значення y з коефіцієнтами регресії
y_st = []
for i in range(n):
    y_st.append(res[0] + res[1] * xn[1][i] + res[2] * xn[2][i] + res[3] *
xn[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:\n", y_st)

# критерій Фішера
print("\nПеревірка адекватності за критерієм Фішера\n")
Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(n)]) / (n - d)

```

```
Fp = Sad / sb
F4 = n - d

# перевірка за допомогою scipy
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("Рівняння регресії адекватне при рівні значимості 0.05")
else:
    print("Рівняння регресії неадекватне при рівні значимості 0.05")
```

Приклад роботи програми

Матриця планування у:

[199, 198, 202]

[201, 202, 196]

[204, 204, 197]

[202, 195, 204]

[196, 205, 196]

[195, 200, 197]

[194, 200, 200]

[199, 203, 195]

[204, 199, 196]

[197, 198, 194]

[201, 200, 197]

[203, 197, 197]

[205, 194, 204]

[201, 200, 203]

[195, 198, 200]

Матриця планування з нормованими коефіцієнтами X:

(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, 0, -0.0, 1.476, 0, 0)
(1, 1.215, 0, 0, 0.0, 0.0, 0, 0.0, 1.476, 0, 0)
(1, 0, -1.215, 0, -0.0, 0, -0.0, -0.0, 0, 1.476, 0)
(1, 0, 1.215, 0, 0.0, 0, 0.0, 0.0, 0, 1.476, 0)
(1, 0, 0, -1.215, 0, -0.0, -0.0, -0.0, 0, 0, 1.476)
(1, 0, 0, 1.215, 0, 0.0, 0.0, 0.0, 0, 0, 1.476)
(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

Рівняння регресії зі знайденими коефіцієнтами:

$$y = 198.736 + -1.1*x_1 + 0.267*x_2 + -0.146*x_3 + -0.25*x_1x_2 + 0.083*x_1x_3 + 0.167*x_2x_3 + 0.5*x_1x_2x_3 - 0.736*x_1^2 + 0.054*x_2^2 + 1.409*x_3^2$$

Перевірка за критерієм Кохрена

Середні значення відгуку за рядками:

199.667 199.667 201.667 200.333 199.0 197.333 198.0 199.0

Дисперсія однорідна

Перевірка значущості коефіцієнтів за критерієм Стьюдента

Значущі коефіцієнти регресії: [198.736, -1.1, 1.409]

Незначущі коефіцієнти регресії: [0.267, -0.146, -0.25, 0.083, 0.167, 0.5, -0.736, 0.054]

Значення з отриманими коефіцієнтами:

[199.83599999999998, 199.83599999999998, 199.83599999999998, 199.83599999999998, 197.636, 197.636, 197.636, 200.0725, 197.3995, 198.736, 198.736, 198.736, 198.736]

Перевірка адекватності за критерієм Фішера

Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0