

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
Королевич Б.В.
Варіант: 310

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

310	0	3	-6	3	-4	1
-----	---	---	----	---	----	---

Код програми

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

m = 3
n = 15

# Варіант №310

x1min = 0
x1max = 3
x2min = -6
x2max = 3
x3min = -4
x3max = 1

# Максимальне та мінімальне значення
ymax = 200 + (x1max + x2max + x3max) / 3
ymin = 200 + (x1min + x2min + x3min) / 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, -1.215, 1.215, 0, 0, 0, 0, 0],
      [-1, -1, 1, 1, -1, -1, 1, 1, 0, 0, -1.215, 1.215, 0, 0, 0],
      [-1, 1, -1, 1, -1, 1, -1, 1, 0, 0, 0, 0, -1.215, 1.215, 0]]

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(ymin), int(ymax)) 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 = (x1max + x1min) / 2
x02 = (x2max + x2min) / 2
x03 = (x3max + x3min) / 2

delta_x1 = x1max - x01
delta_x2 = x2max - x02
delta_x3 = x3max - x03
x0 = [1] * n
x1 = [0, 0, 0, 0, 3, 3, 3, 3, -1.215 * delta_x1 + x01, 1.215 * delta_x1 + x01, x01, x01, x01, x01, x01]
x2 = [-6, -6, 3, 3, -6, -6, 3, 3, x02, x02, -1.215 * delta_x2 + x02, 1.215 * delta_x2 + x02, x02, x02, x02]
x3 = [-4, 1, -4, 1, -4, 1, -4, 1, 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 Стюдента
print(" Перевірка значущості коефіцієнтів за критерієм Стюдента")
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 у 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:
[197, 196, 202]
[202, 198, 197]
[202, 199, 201]
[199, 202, 200]
[199, 198, 201]
[200, 199, 201]
[198, 197, 202]
[197, 202, 198]
[199, 199, 202]
[196, 199, 201]
[197, 200, 198]
[198, 196, 198]
[199, 202, 201]
[197, 196, 201]
[200, 201, 198]

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 = 198.851 + -0.239*x1 + 0.102*x2 + -0.205*x3 + -0.625*x1x2 + 0.042*x1x3 + -0.208*x2x3 + 0.042*x1x2x3 0.508*x1^2 + -0.508*x2^2 + 0.508*x3^2
Перевірка за критерієм Кохрена
Середні значення відгуку за рядками: |
 198.333 199.0 200.667 200.333 199.333 200.0 199.0 199.0
Дисперсія однорідна
Перевірка значущості коефіцієнтів за критерієм Стюдента
Значущі коефіцієнти регресії: [198.851, -0.625]
Незначущі коефіцієнти регресії: [-0.239, 0.102, -0.205, 0.042, -0.208, 0.042, 0.508, -0.508, 0.508]
Значення з отриманими коефіцієнтами:
 [198.226, 198.226, 199.476, 199.476, 199.476, 199.476, 198.226, 198.226, 198.851, 198.851, 198.851, 198.851, 198.851, 198.851, 198.851]
Перевірка адекватності за критерієм Фішера
Рівняння регресії адекватне при рівні значимості 0.05
Process finished with exit code 0
```