

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 № 5

З дисципліни «Методи наукових досліджень» на тему
«Проведення трьохфакторного експерименту
При використанні рівняння регресії з урахуванням квадратичних членів»

Виконав:
Студент II курсу ФІОТ
Групи ІВ-92
Гаргаєв Кирило
Варіант: 205

Перевірив:
Регіда П. Г.

Варіант

№	X1		X2		X3	
	min	max	min	max	min	max
205	-7	4	-6	10	-8	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 205
x1min = -7
x1max = 4
x2min = -6
x2max = 10
x3min = -8
x3max = 1
ymax = 200 + (x1max + x2max + x3max) / 3
ymin = 200 + (x1min + x2min + x3min) / 3
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)

Y_matrix = [[random.randint(int(ymin), int(ymax)) for i in range(m)] for j in
range(n)]

print("Матриця планування y:")
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 = [-4, -4, -4, -4, 4, 4, 4, 4, -1.215 * delta_x1 + x01, 1.215 * delta_x1 +
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]
x1x2, x1x3, x2x3, x1x2x3 = [0] * n, [0] * n, [0] * n, [0] * n
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)
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]
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("Півняння 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])
# дисперсія
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
# однорідність дисперсії
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
# перевірка значущості коефіцієнтів(scipy)
coef_1 = []
coef_2 = []
F3 = (m - 1) * n
for i in range(n - 4):
    if t_list[i] > t.ppf(q=0.975, df=F3):
        coef_1.append(b[i])
        res[i] = b[i]
        d += 1
    else:
        coef_2.append(b[i])
        res[i] = 0

# 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")

```

Результати роботи програми

```

C:\Users\User\PycharmProjects\NLPproject\venv\Scripts\python.exe C:/Users/User/PycharmProjects/NLPproject/Lab5.py
Матриця планування у:
[199, 197, 196]
[198, 193, 200]
[203, 204, 201]
[199, 205, 198]
[203, 201, 200]
[197, 194, 197]
[203, 197, 204]
[202, 193, 201]
[199, 205, 198]
[199, 196, 199]
[196, 195, 202]
[196, 199, 205]
[193, 198, 204]
[200, 198, 194]
[202, 194, 203]

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 = 198.711 + -0.326*x1 + 1.324*x2 + -1.054*x3 + -0.792*x1x2 + -0.708*x1x3 + 0.125*x2x3 + 0.542*x1x2x3 0.635*x1^2 + 0.296*x2^2 + -0.382*x3^2

Перевірка за критерієм Кохрена
Середні значення відгуку за рядками:
197.333 197.0 202.667 200.667 201.333 196.0 201.333 198.667
Дисперсія однорідна
Перевірка значущості коефіцієнтів за критерієм Стюдента
Значущі коефіцієнти регресії: [198.711, 1.324, -1.054]
Незначущі коефіцієнти регресії: [-0.326, -0.792, -0.708, 0.125, 0.542, 0.635, 0.296, -0.382]
Значення з отриманими коефіцієнтами:
[198.441, 196.333, 201.08900000000003, 198.98100000000002, 198.441, 196.333, 201.08900000000003, 198.98100000000002, 198.711, 198.711, 197.10234000000003, 200.31966, 199.99161, 198.711]

Рівняння регресії адекватне при рівні значимості 0.05

```

Висновки

Проведено трьохфакторний експеримент з урахуванням квадратичних членів, використаний центральний ортогональний композиційний план. Знайдено рівняння регресії, яке є адекватним для опису об'єкту. Складено матрицю планування для повного трьохфакторного експерименту, знайдено значення відгуку Y. Складено матрицю планування для ОЦКП. Знайдено значення Y шляхом моделювання випадкових чисел у певному діапазоні відповідно варіанту. Знайдено коефіцієнти рівняння регресії. Проведені 3 статистичні перевірки – за критеріями Кохрена, Стюдента, Фішера. На екран виведені

висновки по адекватності регресії та значимості окремих коефіцієнтів і скореговане рівняння регресії.