

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

Методи наукових досліджень
Лабораторна робота №3
«Проведення трьохфакторного експерименту з використанням лінійного
рівняння регресії»

Виконав:
студент групи ІВ-92
Сударєв Артем Анатолійович
Номер у списку групи – 23

Перевірив:
Регіда П. Г.

Київ 2021 р.

Мета: провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

Завдання:

1. Скласти матрицю планування для дробового трьохфакторного експерименту. Провести експеримент в усіх точках факторного простору, повторивши N експериментів, де N – кількість експериментів (рядків матриці планування) в усіх точках факторного простору – знайти значення функції відгуку Y. Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі (випадковим чином).

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

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

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

2. Знайти коефіцієнти лінійного рівняння регресії. Записати лінійне рівняння регресії.
3. Провести 3 статистичні перевірки.
4. Написати комп'ютерну програму, яка усе це виконує.

Варіант:

223		-30		0		-15		35		-30		-25
-----	--	-----	--	---	--	-----	--	----	--	-----	--	-----

Лістинг програми:

```
from random import *
from pprint import pprint
import numpy as np
from math import sqrt
import sys

from scipy.stats import f
from scipy.stats import t as t_check

m, N, d = 3, 4, 4

x = [[-30, 0], [-15, 35], [-30, -25]]
normalizedX = [[1, -1, -1, -1], [1, -1, 1, 1], [1, 1, -1, 1], [1, 1, 1, -1]]

tran1 = [list(i) for i in zip(*normalizedX)]

x_min_max = [round(sum(x[i][k] for i in range(3)) / 3, 3) for k in range(2)]

y_min_max = [int(200 + x_min_max[i]) for i in range(2)]

print('\nЗадана матриця X:\n', x, '\nНормована матриця X:')

```

```

pprint(normalizedX, width=17)

mat_Y = [[randint(y_min_max[0], y_min_max[1]) for i in range(3)] for k in range(4)]
print('\nXcp min and max:\n', x_min_max, '\nY min and max:\n', y_min_max, '\nМатриця Y:')
pprint(mat_Y, width=17)

mat_serY = [round(sum(mat_Y[k1])/3, 3) for k1 in range(4)]
print('\nСередні значення Y:\n', mat_serY, '\nМатриця X:')

mat_X = [[-25, -30, -5], [-25, 45, 5], [-5, -30, 5], [-5, 45, -5]]
pprint(mat_X, width=17)

mx = [round(sum(mat_X[i][k] for i in range(4))/4, 3) for k in range(3)]
my = round(sum(mat_serY)/4, 3)
print('\nЗначення mx:\n', mx, '\nЗначення my:\n', my)

tran = [list(i) for i in zip(*mat_X)]
ai = [round(sum(tran[k][i] * mat_serY[i] for i in range(4)) / 4, 3) for k in range(3)]
aai = [round(sum(tran[k][i]**2 for i in range(4)) / 4, 3) for k in range(3)]
print('\nЗначення ai:\n', ai, '\nЗначення aai:\n', aai)

a12, a21, a13, a31, a23, a32 = 0, 0, 0, 0, 0, 0

for i in range(len(tran[0])):
    a12 += tran[0][i] * tran[1][i] / 4

for i in range(len(tran[0])):
    a13 += tran[0][i] * tran[2][i] / 4

for i in range(len(tran[1])):
    a23 += tran[1][i] * tran[2][i] / 4

a21 = a12
a31 = a13
a32 = a23

denominator = np.linalg.det(np.array([
    [1, mx[0], mx[1], mx[2]],
    [mx[0], aai[0], a12, a13],
    [mx[1], a12, aai[1], a32],
    [mx[2], a13, a23, aai[2]]]))

b = []
b.append(np.linalg.det(np.array([[my, mx[0], mx[1], mx[2]], [ai[0], aai[0], a12, a13], [ai[1], a12, aai[1], a32], [ai[2], a13, a23, aai[2]]])) / denominator)
b.append(np.linalg.det(np.array([[1, my, mx[1], mx[2]], [mx[0], ai[0], a12, a13], [mx[1], ai[1], aai[1], a32], [mx[2], ai[2], a23, aai[2]]])) / denominator)
b.append(np.linalg.det(np.array([[1, mx[0], my, mx[2]], [mx[0], aai[0], ai[0], a13], [mx[1], a12, ai[1], a32], [mx[2], a13, ai[2], aai[2]]])) / denominator)
b.append(np.linalg.det(np.array([[1, mx[0], mx[1], my], [mx[0], aai[0], a12, ai[0]], [mx[1], a12, aai[1], ai[1]], [mx[2], a13, a23, ai[2]]])) / denominator)

checking = [round(b[0] + b[1] * tran[0][i] + b[2] * tran[1][i] + b[3] * tran[2][i], 3) for i in range(4)]

print("\nПеревірка порівнянням з середніми значеннями Y:\n", checking)
mat_disY = [round(sum([(k1 - mat_serY[j]) ** 2) for k1 in mat_Y[j]]) / m, 3) for j in range(4)]
print("\nДисперсії в рядках:\n", mat_disY)
print('\nПеревірка однорідності дисперсії за критерієм Кохрена:\n')

```

```

if max(mat_disY)/sum(mat_disY) < 0.7679:
    print('\nДисперсія однорідна')
else:
    print('\nДисперсія неоднорідна')
    sys.exit(0)

print('\nПеревірка значущості коефіцієнтів за критерієм Стьюдента:')

S2b = sum(mat_disY) / N
S2bs = S2b / (m * N)
Sbs = sqrt(S2bs)

print('\nSbs:\n', round(Sbs, 3))

bb = [round(sum(mat_serY[k] * tran1[i][k] for k in range(N))/N, 3) for i in range(N)]
t = [round(abs(bb[i])/Sbs, 3) for i in range(N)]

print('\nbi:\n', bb, '\nti:\n', t)

f1, f2 = m - 1, N
f3 = f1 * f2

for i in range(N):
    if t[i] < t_check.ppf(q=0.975, df=f3):
        b[i] = 0
        d -= 1
        print('Виключаємо з рівняння коефіцієнт b', i)

y_reg = [round(b[0] + b[1] * mat_X[i][0] + b[2] * mat_X[i][1] + b[3] * mat_X[i][2],
3) for i in range(4)]

print('Значення рівнянь регресій:\n', y_reg)
print('\nПеревірка адекватності за критерієм Фішера:')

f4 = N - d
Sad = (m / (N - d)) * int(sum(y_reg[i] - mat_serY[i] for i in range(N))**2)

Fp = Sad / S2b

print('\nКількість значимих коефіцієнтів:\n', d, '\nFp:\n', round(Fp, 3))

if Fp > f.ppf(q=0.95, dfn=f4, dfd=f3):
    print('Рівняння регресії неадекватно оригіналу при рівні значимості 0.05')
else:
    print('Рівняння регресії адекватно оригіналу при рівні значимості 0.05')

```

Відповіді на контрольні запитання:

Контрольні запитання:

1. Що називається дробовим факторним експериментом?

У деяких випадках немає необхідності проводити повний факторний експеримент (ПФЕ). Якщо буде використовуватися лінійна регресія, то можливо зменшити кількість рядків матриці ПФЕ до кількості коефіцієнтів регресійної моделі. Кількість дослідів слід скоротити, використовуючи для

планування так звані регулярні дробові репліки від повного факторного експерименту, що містять відповідну кількість дослідів і зберігають основні властивості матриці планування – це називають дробовим факторним експериментом (ДФЕ).

2. Для чого потрібно розрахункове значення Кохрена?

За критерієм Кохрена здійснюється перевірка однорідності дисперсії.

3. Для чого перевіряється критерій Стюдента?

Критерій Стюдента застосовується для перевірки значущості коефіцієнтів.

4. Чим визначається критерій Фішера і як його застосовувати?

Отримане рівняння регресії необхідно перевірити на адекватність досліджуваному об'єкту. Для цієї мети необхідно оцінити, наскільки відрізняються середні значення у вихідної величини, отриманої в точках факторного простору, і значення у, отриманого з рівняння регресії в тих самих точках факторного простору. Для цього використовують дисперсію адекватності. Адекватність моделі перевіряють за F-критерієм Фішера, який дорівнює відношенню дисперсії адекватності до дисперсії відтворюваності:

$$F_p = S_{\text{ад}}^2 / S_{\text{в}}^2$$