Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Методи оптимізації та планування експерименту

Лабораторна робота №3

“ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ”

Виконав:

студент групи ІВ-83

Чорноморець В.О.

Перевірив:

ас. Регіда П.Г.

Київ

2020 р.

**Мета:** провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести

натуралізацію рівняння регресії.

Номер у списку: 21

Варіант завдання: 321

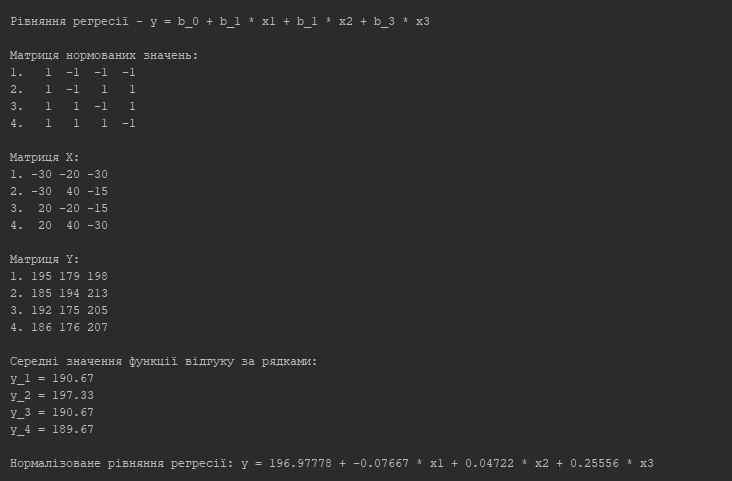


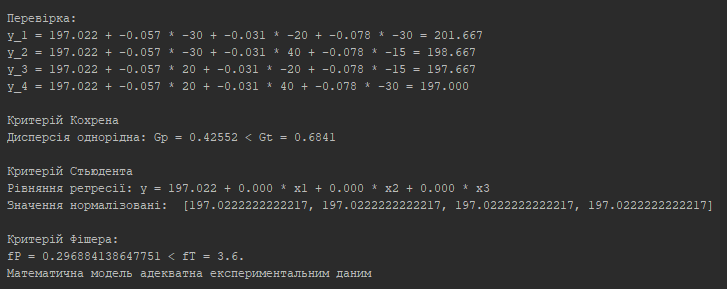


1. Код програми:

import numpy as np  
  
x1\_min = -30  
x1\_max = 20  
x2\_min = -20  
x2\_max = 40  
x3\_min = -30  
x3\_max = -15  
  
x\_average\_max = (x1\_max + x2\_max + x3\_max) / 3  
x\_average\_min = (x1\_min + x2\_min + x3\_min) / 3  
y\_max = 200 + x\_average\_max  
y\_min = 200 + x\_average\_min  
  
m, n = 3, 4  
  
def main(m, n):  
 print("\nМатриця нормованих значень:")  
 norm\_x = np.array([  
 [+1, -1, -1, -1],  
 [+1, -1, +1, +1],  
 [+1, +1, -1, +1],  
 [+1, +1, +1, -1]  
 ])  
 for i in range(len(norm\_x)):  
 print("{}.".format(i + 1), end="")  
 for j in range(len(norm\_x[i])):  
 print("{:4}".format(norm\_x[i][j]), end="")  
 print()  
  
 print("\nМатриця Х:")  
 x = np.array([  
 [x1\_min, x2\_min, x3\_min],  
 [x1\_min, x2\_max, x3\_max],  
 [x1\_max, x2\_min, x3\_max],  
 [x1\_max, x2\_max, x3\_min]  
 ])  
  
 for i in range(len(x)):  
 print("{}.".format(i + 1), end="")  
 for j in range(len(x[i])):  
 print("{:4}".format(x[i][j]), end="")  
 print()  
  
 print("\nМатриця Y:")  
 y = np.random.randint(y\_min, y\_max, size=(n, m))  
 for i in range(len(y)):  
 print("{}.".format(i + 1), end="")  
 for j in range(len(y[i])):  
 print("{:4}".format(y[i][j]), end="")  
 print()  
  
 print("\nСередні значення функції відгуку за рядками:")  
 y\_average = np.sum(y, axis=1) / len(y[0])  
 y\_1, y\_2, y\_3, y\_4 = y\_average  
 print(f"y\_1 = {y\_1:.2f}\ny\_2 = {y\_2:.2f}\ny\_3 = {y\_3:.2f}\ny\_4 = {y\_4:.2f}")  
 mx\_1, mx\_2, mx\_3 = [i / len(x) for i in np.sum(x, axis=0)]  
 my = sum(y\_average) / len(y\_average)  
  
 a\_1 = sum([x[i][0] \* y\_average[i] for i in range(len(x))]) / len(x)  
 a\_2 = sum([x[i][1] \* y\_average[i] for i in range(len(x))]) / len(x)  
 a\_3 = sum([x[i][2] \* y\_average[i] for i in range(len(x))]) / len(x)  
  
 a\_11 = sum([x[i][0] \*\* 2 for i in range(len(x))]) / len(x)  
 a\_22 = sum([x[i][1] \*\* 2 for i in range(len(x))]) / len(x)  
 a\_33 = sum([x[i][2] \*\* 2 for i in range(len(x))]) / len(x)  
 a\_12 = sum([x[i][0] \* x[i][1] for i in range(len(x))]) / len(x)  
 a\_13 = sum([x[i][0] \* x[i][2] for i in range(len(x))]) / len(x)  
 a\_23 = a\_32 = sum([x[i][1] \* x[i][2] for i in range(len(x))]) / len(x)  
  
 det = np.linalg.det([[1, mx\_1, mx\_2, mx\_3], [mx\_1, a\_11, a\_12, a\_13], [mx\_2, a\_12, a\_22, a\_32], [mx\_3, a\_13, a\_23, a\_33]])  
 det\_0 = np.linalg.det([[my, mx\_1, mx\_2, mx\_3], [a\_1, a\_11, a\_12, a\_13], [a\_2, a\_12, a\_22, a\_32], [a\_3, a\_13, a\_23, a\_33]])  
 det\_1 = np.linalg.det([[1, my, mx\_2, mx\_3], [mx\_1, a\_1, a\_12, a\_13], [mx\_2, a\_2, a\_22, a\_32], [mx\_3, a\_3, a\_23, a\_33]])  
 det\_2 = np.linalg.det([[1, mx\_1, my, mx\_3], [mx\_1, a\_11, a\_1, a\_13], [mx\_2, a\_12, a\_2, a\_32], [mx\_3, a\_13, a\_3, a\_33]])  
 det\_3 = np.linalg.det([[1, mx\_1, mx\_2, my], [mx\_1, a\_11, a\_12, a\_1], [mx\_2, a\_12, a\_22, a\_2], [mx\_3, a\_13, a\_23, a\_3]])  
  
 b\_0 = det\_0 / det  
 b\_1 = det\_1 / det  
 b\_2 = det\_2 / det  
 b\_3 = det\_3 / det  
 b = [b\_0, b\_1, b\_2, b\_3]  
  
 print(f"\nНормалізоване рівняння регресії: y = {b\_0:.5f} + {b\_1:.5f} \* x1 + {b\_2:.5f} \* x2 + {b\_3:.5f} \* x3\n")  
 print("Перевірка:")  
 y\_1\_ch = b\_0 + b\_1 \* x[0][0] + b\_2 \* x[0][1] + b\_3 \* x[0][2]  
 y\_2\_ch = b\_0 + b\_1 \* x[1][0] + b\_2 \* x[1][1] + b\_3 \* x[1][2]  
 y\_3\_ch = b\_0 + b\_1 \* x[2][0] + b\_2 \* x[2][1] + b\_3 \* x[2][2]  
 y\_4\_ch = b\_0 + b\_1 \* x[3][0] + b\_2 \* x[3][1] + b\_3 \* x[3][2]  
 print(f"y\_1 = {b\_0:.3f} + {b\_1:.3f} \* {x[0][0]} + {b\_2:.3f} \* {x[0][1]} + {b\_3:.3f} \* {x[0][2]} = {y\_1\_ch:.3f}"  
 f"\ny\_2 = {b\_0:.3f} + {b\_1:.3f} \* {x[1][0]} + {b\_2:.3f} \* {x[1][1]} + {b\_3:.3f} \* {x[1][2]} = {y\_2\_ch:.3f}"  
 f"\ny\_3 = {b\_0:.3f} + {b\_1:.3f} \* {x[2][0]} + {b\_2:.3f} \* {x[2][1]} + {b\_3:.3f} \* {x[2][2]} = {y\_3\_ch:.3f}"  
 f"\ny\_4 = {b\_0:.3f} + {b\_1:.3f} \* {x[3][0]} + {b\_2:.3f} \* {x[3][1]} + {b\_3:.3f} \* {x[3][2]} = {y\_4\_ch:.3f}")  
  
 print("\nКритерій Кохрена")  
 f\_1 = m - 1  
 f\_2 = n  
 s\_1 = sum([(i - y\_1) \*\* 2 for i in y[0]]) / m  
 s\_2 = sum([(i - y\_2) \*\* 2 for i in y[1]]) / m  
 s\_3 = sum([(i - y\_3) \*\* 2 for i in y[2]]) / m  
 s\_4 = sum([(i - y\_4) \*\* 2 for i in y[3]]) / m  
 s\_array = np.array([s\_1, s\_2, s\_3, s\_4])  
 gP = max(s\_array) / sum(s\_array)  
  
 # Розподіл Кохрена  
 table = {3: 0.6841, 4: 0.6287, 5: 0.5892, 6: 0.5598, 7: 0.5365, 8: 0.5175, 9: 0.5017, 10: 0.4884,  
 range(11, 17): 0.4366, range(17, 37): 0.3720, range(37, 145): 0.3093}  
 gT = table.get(m)  
  
 if (gP < gT):  
 print(f"Дисперсія однорідна: Gp = {gP:.5} < Gt = {gT}")  
 else:  
 print(f"Дисперсія неоднорідна: Gp = {gP:.5} > Gt = {gT}")  
 print(f'Збільшуємо m на 1')  
 m = m + 1  
 main(m + 1, n)  
 return  
  
 print("\nКритерій Стьюдента")  
 s2\_B = s\_array.sum() / n  
 s2\_b\_S = s2\_B / (n \* m)  
 s\_b\_S = pow(s2\_b\_S, 1 / 2)  
  
 b\_0 = sum([norm\_x[i][0] \* y\_average[i] for i in range(len(norm\_x))]) / n  
 b\_1 = sum([norm\_x[i][1] \* y\_average[i] for i in range(len(norm\_x))]) / n  
 b\_2 = sum([norm\_x[i][2] \* y\_average[i] for i in range(len(norm\_x))]) / n  
 b\_3 = sum([norm\_x[i][3] \* y\_average[i] for i in range(len(norm\_x))]) / n  
  
 t = [abs(b\_0) / s\_b\_S, abs(b\_1) / s\_b\_S, abs(b\_2) / s\_b\_S, abs(b\_3) / s\_b\_S]  
  
 f3 = f\_1 \* f\_2  
 # Розподіл Стьюдента  
 t\_table = {8: 2.306, 9: 2.262, 10: 2.228, 11: 2.201, 12: 2.179, 13: 2.160, 14: 2.145, 15: 2.131, 16: 2.120,  
 17: 2.110, 18: 2.101, 19: 2.093, 20: 2.086, 21: 2.08, 22: 2.074, 23: 2.069, 24: 2.064, 25: 2.06}  
 d = 4  
  
 for i in range(len(t)):  
 if (t\_table.get(f3) > t[i]):  
 b[i] = 0  
 d -= 1  
  
 print(f"Рівняння регресії: y = {b[0]:.3f} + {b[1]:.3f} \* x1 + {b[2]:.3f} \* x2 + {b[3]:.3f} \* x3")  
 ch\_0 = b[0] + b[1] \* x[0][0] + b[2] \* x[0][1] + b[3] \* x[0][2]  
 ch\_1 = b[0] + b[1] \* x[1][0] + b[2] \* x[1][1] + b[3] \* x[1][2]  
 ch\_2 = b[0] + b[1] \* x[2][0] + b[2] \* x[2][1] + b[3] \* x[2][2]  
 ch\_3 = b[0] + b[1] \* x[3][0] + b[2] \* x[3][1] + b[3] \* x[3][2]  
 ck\_list = [ch\_0, ch\_1, ch\_2, ch\_3]  
 print("Значення нормалізовані: ", ck\_list)  
  
 print("\nКритерій Фішера:")  
 f\_4 = n - d  
 s2\_ad = m / f\_4 \* sum([(ck\_list[i] - y\_average[i]) \*\* 2 for i in range(len(y\_average))])  
 fP = s2\_ad / s2\_B  
 # Розподіл Фішера  
 fT = [  
 [164.4, 199.5, 215.7, 224.6, 230.2, 234],  
 [18.5, 19.2, 19.2, 19.3, 19.3, 19.3],  
 [10.1, 9.6, 9.3, 9.1, 9, 8.9],  
 [7.7, 6.9, 6.6, 6.4, 6.3, 6.2],  
 [6.6, 5.8, 5.4, 5.2, 5.1, 5],  
 [6, 5.1, 4.8, 4.5, 4.4, 4.3],  
 [5.5, 4.7, 4.4, 4.1, 4, 3.9],  
 [5.3, 4.5, 4.1, 3.8, 3.7, 3.6],  
 [5.1, 4.3, 3.9, 3.6, 3.5, 3.4],  
 [5, 4.1, 3.7, 3.5, 3.3, 3.2],  
 [4.8, 4, 3.6, 3.4, 3.2, 3.1],  
 [4.8, 3.9, 3.5, 3.3, 3.1, 3],  
 [4.7, 3.8, 3.4, 3.2, 3, 2.9],  
 [4.6, 3.7, 3.3, 3.1, 3, 2.9],  
 [4.5, 3.7, 3.3, 3.1, 2.9, 2.8],  
 [4.5, 3.6, 3.2, 3, 2.9, 2.7],  
 [4.5, 3.6, 3.2, 3, 2.8, 2.7],  
 [4.4, 3.6, 3.2, 2.9, 2.8, 2.7],  
 [4.4, 3.5, 3.1, 2.9, 2.7, 2.6],  
 [4.4, 3.5, 3.1, 2.9, 2.7, 2.6]  
 ]  
 if (fP > fT[f3][f\_4]):  
 print(f"fp = {fP} > ft = {fT[f3][f\_4]}.\n Математична модель неадекватна експериментальним даним\n")  
 else:  
 print(f"fP = {fP} < fT = {fT[f3][f\_4]}.\n Математична модель адекватна експериментальним даним\n")  
  
  
print("\nРівняння регресії - y = b\_0 + b\_1 \* x1 + b\_1 \* x2 + b\_3 \* x3")  
main(m, n)

Результати виконання роботи





**Відповіді на контрольні запитання**

1. Що називається дробовим факторним експериментом?

Скорочена кількість дослідів, використовуючи для планування так звані регулярні дробові репліки від повного факторного експерименту, що містять відповідну кількість дослідів і зберігають основні властивості матриці планування.

2. Для чого потрібно розрахункове значення Кохрена?

Для перевірки однорідності дисперсії

3. Для чого перевіряється критерій Стьюдента?

Для перевірки значущості коефіціентів

4. Чим визначається критерій Фішера і як його застосовувати?

Визначається відношенням дисперсії адекватності до дисперсії відтворюваності.

Знайдене шляхом розрахунку Fp порівнюють з табличним значенням Fт, що визначається при рівні значимості q та кількості ступенів свободи f4 = N - d і

f3 = f1 \* f2

Якщо Fp< Fт то отримана математична модель з прийнятим рівнем статистичної значимості q адекватна експериментальним даним.