Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №3

Тема: «Проведення трьохфакторного експерименту з використанням

Лінійного рівняння регресії.»

Виконав:

студент групи ІО-93

Ващенко Ірина Олегівна

Номер в списку групи: 4

Перевірив:

ас. Регiда П. Г.

Київ 2021

**Лабораторна робота № 3**

«Проведення трьохфакторного експерименту з використанням

Лінійного рівняння регресії.»

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

**Завдання на лабораторну роботу:**

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 У. Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі (випадковим чином).

2.Знайти коефіцієнти лінійного рівняння регресії. Записати лінійне рівняння регресії.

3. Провести 3 статистичні перевірки.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № варианта | X1 | | X2 | | X3 | |
| min | max | min | max | min | max |
| 304 | 15 | 45 | 30 | 80 | 15 | 45 |

4. Написати комп'ютерну програму, яка усе це виконує.

import numpy as np  
import random  
from numpy.linalg import solve  
from scipy.stats import f, t  
from functools import partial  
  
x1\_min, x1\_max = 15, 45  
x2\_min, x2\_max = 30, 80  
x3\_min, x3\_max = 15, 45  
  
aver\_ymin = (45 + 80 + 45)/ 3  
aver\_ymax = (15 + 30 + 15) / 3  
  
y\_max = 200 + int(aver\_ymin)  
y\_min = 200 + int(aver\_ymax)  
y\_min\_max = [y\_min, y\_max]  
  
  
def experiment(n, m):  
 y = np.random.randint(\*y\_min\_max, size=(n, m))  
  
 x\_plan = np.array([[1, -1, -1, -1],  
 [1, -1, 1, 1],  
 [1, 1, -1, 1],  
 [1, 1, 1, -1]])  
  
 xn = np.array([[1, 15, 30, 15],  
 [1, 15, 80, 45],  
 [1, 45, 30, 45],  
 [1, 45, 80, 15]])  
  
 print(**'**\n**Матриця планування'**)  
 labels\_table = [**"x0"**, **"x1"**, **"x2"**, **"x3"**] + [**"y{}"**.format(i + 1) for i in range(4)]  
 rows\_table = [list(xn[i]) + list(y[i]) for i in range(3)]  
 print((**" "** \* 4).join(labels\_table))  
 print(**"**\n**"**.join([**" "**.join(map(lambda j: **"{:<5}"**.format(j), rows\_table[i])) for i in range(len(rows\_table))]))  
 return xn, y, x\_plan  
  
def Regression(x, b):  
 y = sum([x[i]\*b[i] for i in range(len(x))])  
 return y  
  
def coefficient(x, y\_avarg, n):  
 mx1 = sum(x[:, 1]) / n  
 mx2 = sum(x[:, 2]) / n  
 mx3 = sum(x[:, 3]) / n  
 my = sum(y\_avarg) / n  
  
 a12 = sum([x[i][1] \* x[i][2] for i in range(len(x))]) / n  
 a13 = sum([x[i][1] \* x[i][3] for i in range(len(x))]) / n  
 a23 = sum([x[i][2] \* x[i][3] for i in range(len(x))]) / n  
 a11 = sum([i \*\* 2 for i in x[:, 1]]) / n  
 a22 = sum([i \*\* 2 for i in x[:, 2]]) / n  
 a33 = sum([i \*\* 2 for i in x[:, 3]]) / n  
 a1 = sum([y\_avarg[i] \* x[i][1] for i in range(len(x))]) / n  
 a2 = sum([y\_avarg[i] \* x[i][2] for i in range(len(x))]) / n  
 a3 = sum([y\_avarg[i] \* x[i][3] for i in range(len(x))]) / n  
  
  
 X = [[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a23], [mx3, a13, a23, a33]]  
 Y = [my, a1, a2, a3]  
 B = [round(i, 2) for i in solve(X, Y)]  
  
 print(**'**\n**Рівняння регресії'**)  
 print(**f'**{B[0]} **+** {B[1]}**\*x1 +** {B[2]}**\*x2 +** {B[3]}**\*x3'**)  
  
 return B  
  
  
def count\_dispersion(y, y\_avarg, n, m):  
 y\_var = np.var(y, axis=1)  
 return y\_var  
  
  
  
  
def check\_criter\_Cochran(y, y\_avarg, n, m):  
 S\_kv = count\_dispersion(y, y\_avarg, n, m)  
 Gp = max(S\_kv) / sum(S\_kv)  
 print(**'**\n**Перевірка за критерієм Кохрена'**)  
 return Gp  
  
  
# оцінки коефіцієнтів  
def bs(x, y, y\_avarg, n):  
 res = [sum(1 \* y for y in y\_avarg) / n]  
 for i in range(3): # 4 - ксть факторів  
 b = sum(j[0] \* j[1] for j in zip(x[:, i], y\_avarg)) / n  
 res.append(b)  
 return res  
  
  
  
def significance\_student(x, y, y\_avarg, n, m):  
  
 S\_kv = count\_dispersion(y, y\_avarg, n, m)  
 kv\_aver = sum(S\_kv) / n  
  
 s\_Bs = (kv\_aver / n / m) \*\* 0.5  
 Bs = bs(x, y, y\_avarg, n)  
 ts = [abs(B) / s\_Bs for B in Bs]  
 return ts  
  
  
  
def adequacy\_Fishera(y, y\_avarg, y\_new, n, m, d):  
 S\_ad = m / (n - d) \* sum([(y\_new[i] - y\_avarg[i])\*\*2 for i in range(len(y))])  
 S\_kv = count\_dispersion(y, y\_avarg, n, m)  
 S\_kv\_aver = sum(S\_kv) / n  
  
 return S\_ad / S\_kv\_aver  
  
  
def Cochran(f1, f2, q=0.05):  
 q1 = q / f1  
 fisher\_value = f.ppf(q=1 - q1, dfn=f2, dfd=(f1 - 1) \* f2)  
 return fisher\_value / (fisher\_value + f1 - 1)  
  
  
def main(n, m):  
 f1 = m - 1  
 f2 = n  
 f3 = f1 \* f2  
 q = 0.05  
  
 student = partial(t.ppf, q=1-0.025)  
 t\_student = student(df=f3)  
  
 G\_kr = Cochran(f1, f2)  
  
 x, y, x\_plan = experiment(n, m)  
 y\_avarg = [sum(y[i][j] for j in range(4)) / 4 for i in range(4)]  
  
 B = coefficient(x, y\_avarg, n)  
  
 Gp = check\_criter\_Cochran(y, y\_avarg, n, m)  
 print(**f'Gp =** {Gp}**'**)  
 if Gp < G\_kr:  
 print(**f'Дисперсії однорідні** {1-q}**'**)  
 else:  
 print(**"Збільшти кількість дослідів"**)  
 m += 1  
 main(n, m)  
  
 ts = significance\_student(x\_plan[:, 1:], y, y\_avarg, n, m)  
 print(**'**\n**Критерій Стьюдента:**\n**'**, ts)  
 res = [t for t in ts if t > t\_student]  
 final\_k = [B[ts.index(i)] for i in ts if i in res]  
 print(**'Cтатистично незначущі {}, тому виключаємо їх з рівняння.'**.format([i for i in B if i not in final\_k]))  
  
 y\_regr = []  
 for j in range(n):  
 y\_regr.append(Regression([x[j][ts.index(i)] for i in ts if i in res], final\_k))  
  
 print(**f'**\n **"y" =>** {final\_k}**'**)  
 print(y\_regr)  
  
 d = len(res)  
 f4 = n - d  
 F\_p = adequacy\_Fishera(y, y\_avarg, y\_regr, n, m, d)  
  
 fisher = partial(f.ppf, q=1 - 0.05)  
 f\_t = fisher(dfn=f4, dfd=f3) # табличне знач  
  
 print(**'**\n**Перевірка адекватності за критерієм Фішера'**)  
 print(**'Fp ='**, F\_p)  
 print(**'F\_t ='**, f\_t)  
 if F\_p < f\_t:  
 print(**'Математична модель адекватна'**)  
 else:  
 print(**'Математична модель не адекватна'**)  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main(4, 4)

Text, email

Description automatically generated

**Контрольні запитання:**

*1.Що називається дробовим факторним експериментом?*

**Дробовий факторний експеримент** – це частина ПФЕ, який мінімізує число дослідів, за рахунок тієї інформації, яка не дуже істотна для побудови лінійної моделі.

*2. Для чого потрібно розрахункове значення Кохрена?*

**Критерій Кохрена** — використовують для порівняння трьох і більше виборок однакового обсягу n.

Якщо вибіркові дисперсії отримані за вибірками однакових обсягів, для їх порівняння використовують більш зручний і точний критерій Кохрена. Кохрена досліджував розподіл максимальної вибіркової дисперсії до суми всіх дисперсій.

*3. Для чого перевіряється критерій Стьюдента?*

**Критерій Стьюдента** - загальна назва для статистичних тестів, в яких статистика критерію має розподіл Стьюдента. Найбільш часто t-критерії застосовуються для перевірки рівності середніх значень у двох вибірках. Тому перед застосуванням критерію Стьюдента рекомендується виконати перевірку нормальності.

*4. Чим визначається критерій Фішера і як його застосовувати?*

Критерій Фішера застосовується для перевірки рівності дисперсій двох вибірок. Його відносять до критеріїв розсіювання. Критерій Фішера заснований на додаткових припущеннях про незалежність і нормальності вибірок даних. Перед його застосуванням рекомендується виконати перевірку нормальності.

**Висновок:**

В процесі лабораторної роботи було проведено дробовий трьохфакторний експеримент. Після чого склали матрицю планування, знайшли коефіцієнти рівняння регресії та провели перевірку однорідність дисперсії за критерієм Кохрена, нуль-гіпотезу за критерієм Стьюдента та адекватність моделі за критерієм Фішера.