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 №3**

З дисципліни «Методи оптимізації та планування»

**ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ**

ВИКОНАВ:

Студент ІІ курсу ФІОТ

Групи ІВ-91

Каптур Максим Ігорович

Номер залікової книжки - 9112

ПЕРЕВІРИВ:

Регіда П.Г.

Київ 2021 р.

**Мета:**

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

**Варіант завдання:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Варіант | Х1 | | Х2 | | Х3 | |
| min | max | min | max | min | max |
| 112 | -40 | 20 | -35 | 15 | 20 | 25 |

**Лістинг програми:**

from random import \*  
import numpy as np  
from numpy.linalg import solve  
from scipy.stats import f, t  
from functools import partial  
  
  
class Laba\_3:  
  
 def \_\_init\_\_(self, n, m):  
 self.n = n  
 self.m = m  
 self.X\_Min = (-40 - 35 + 20) / 3  
 self.X\_Max = (20 + 15 + 25) / 3  
 self.Y\_Max = round(200 + self.X\_Max)  
 self.Y\_Min = round(200 + self.X\_Min)  
 self.x\_Norm = [[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]]  
 self.X\_range = [(-40, 20), (-35, 15), (20, 25)]  
 self.Y = np.zeros(shape=(self.n, self.m))  
 self.Y\_new = []  
 for i in range(self.n):  
 for j in range(self.m):  
 self.Y[i][j] = randint(self.Y\_Min, self.Y\_Max)  
 self.Y\_av = [round(sum(i) / len(i), 2) for i in self.Y]  
 self.x\_Norm = self.x\_Norm[:len(self.Y)]  
 self.x = np.ones(shape=(len(self.x\_Norm), len(self.x\_Norm[0])))  
 for i in range(len(self.x\_Norm)):  
 for j in range(1, len(self.x\_Norm[i])):  
 if self.x\_Norm[i][j] == -1:  
 self.x[i][j] = self.X\_range[j - 1][0]  
 else:  
 self.x[i][j] = self.X\_range[j - 1][1]  
 self.f1 = m - 1  
 self.f2 = n  
 self.f3 = self.f1 \* self.f2  
 self.q = 0.05  
  
 def regres(self, x, b):  
 Y = sum([x[i] \* b[i] for i in range(len(x))])  
 return Y  
  
 def koefs(self):  
 mx1 = sum(self.x[:, 1]) / self.n  
 mx2 = sum(self.x[:, 2]) / self.n  
 mx3 = sum(self.x[:, 3]) / self.n  
 my = sum(self.Y\_av) / self.n  
 a12 = sum([self.x[i][1] \* self.x[i][2] for i in range(len(self.x))]) / self.n  
 a13 = sum([self.x[i][1] \* self.x[i][3] for i in range(len(self.x))]) / self.n  
 a23 = sum([self.x[i][2] \* self.x[i][3] for i in range(len(self.x))]) / self.n  
 a11 = sum([i \*\* 2 for i in self.x[:, 1]]) / self.n  
 a22 = sum([i \*\* 2 for i in self.x[:, 2]]) / self.n  
 a33 = sum([i \*\* 2 for i in self.x[:, 3]]) / self.n  
 a1 = sum([self.Y\_av[i] \* self.x[i][1] for i in range(len(self.x))]) / self.n  
 a2 = sum([self.Y\_av[i] \* self.x[i][2] for i in range(len(self.x))]) / self.n  
 a3 = sum([self.Y\_av[i] \* self.x[i][3] for i in range(len(self.x))]) / self.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('\nНаше рівняння регресії')  
 print(f'Y = {B[0]} + {B[1]}\*x1 + {B[2]}\*x2 + {B[3]}\*x3')  
  
 return B  
  
 def dispers(self):  
 result = []  
 for i in range(self.n):  
 s = sum([(self.Y\_av[i] - self.Y[i][j]) \*\* 2 for j in range(self.m)]) / self.m  
 result.append(s)  
 return result  
  
 def KOHREN(self):  
 q1 = self.q / self.f1  
 fisher\_value = f.ppf(q=1 - q1, dfn=self.f2, dfd=(self.f1 - 1) \* self.f2)  
 G\_cr = fisher\_value / (fisher\_value + self.f1 - 1)  
 s = self.dispers()  
 Gp = max(s) / sum(s)  
 return Gp, G\_cr  
  
 def Student(self):  
 def bs():  
 result = [sum(1 \* Y for Y in self.Y\_av) / self.n]  
 for i in range(3):  
 b = sum(j[0] \* j[1] for j in zip(self.x[:, i], self.Y\_av)) / self.n  
 result.append(b)  
 return result  
  
 S\_kv = self.dispers()  
 s\_kv\_aver = sum(S\_kv) / self.n  
  
 s\_Bs = (s\_kv\_aver / self.n / self.m) \*\* 0.5  
 Bs = bs()  
 ts = [abs(B) / s\_Bs for B in Bs]  
 return ts  
  
 def FISHER(self, d):  
  
 S\_ad = self.m / (self.n - d) \* sum([(self.Y\_new[i] - self.Y\_av[i]) \*\* 2 for i in range(len(self.Y))])  
 S\_kv = self.dispers()  
 S\_kv\_aver = sum(S\_kv) / self.n  
 F\_p = S\_ad / S\_kv\_aver  
 return F\_p  
  
 def all\_toogether(self):  
  
 Student = partial(t.ppf, q=1 - 0.025)  
 t\_student = Student(df=self.f3)  
  
 print('\nКритерій Кохрена, перевірка')  
 Gp, G\_kr = self.KOHREN()  
 print(f'Gp = {Gp}')  
 if Gp < G\_kr:  
 print(f'З ймовірністю {1-self.q} дисперсії однорідні.')  
 else:  
 print("Либо больше, либо ничего (дослідів)")  
 self.m += 1  
 FractionalExperiment(self.n, self.m)  
  
 ts = self.Student()  
 print('\nПеревірка значущості коефіцієнтів за критерієм Стьюдента')  
 print('Критерій Стьюдента:\n', ts)  
 result = [t for t in ts if t > t\_student]  
 B = self.koefs()  
 final\_k = [B[ts.index(i)] for i in ts if i in result]  
 print('Коефіцієнти {} статистично незначущі, тому ми виключаємо їх з рівняння.'.format(  
 [i for i in B if i not in final\_k]))  
  
 for j in range(self.n):  
 self.Y\_new.append(self.regres([self.x[j][ts.index(i)] for i in ts if i in result], final\_k))  
  
 print(f'\nЗначення "Y" з коефіцієнтами {final\_k}')  
 print(self.Y\_new)  
  
 d = len(result)  
 f4 = self.n - d  
 F\_p = self.FISHER(d)  
  
 FISHER = partial(f.ppf, q=1 - 0.05)  
 f\_t = FISHER(dfn=f4, dfd=self.f3) # табличне знач  
 print('\nПеревірка адекватності за критерієм Фішера')  
 print('Fp =', F\_p)  
 print('F\_t =', f\_t)  
 if F\_p < f\_t:  
 print('Математична модель адекватна експериментальним даним')  
 else:  
 print('Математична модель не адекватна експериментальним даним')  
  
  
experiment = Laba\_3(7, 8)  
experiment.all\_toogether()

**Контрольні запитання:**

1. Що називається дробовим факторним експериментом?

Дробовим факторним експериментом називається експеримент з використанням частини повного факторного експерименту.

1. Для чого потрібно розрахункове значення Кохрена?

Розрахункове значення Кохрена використовують для перевірки однорідності дисперсій.

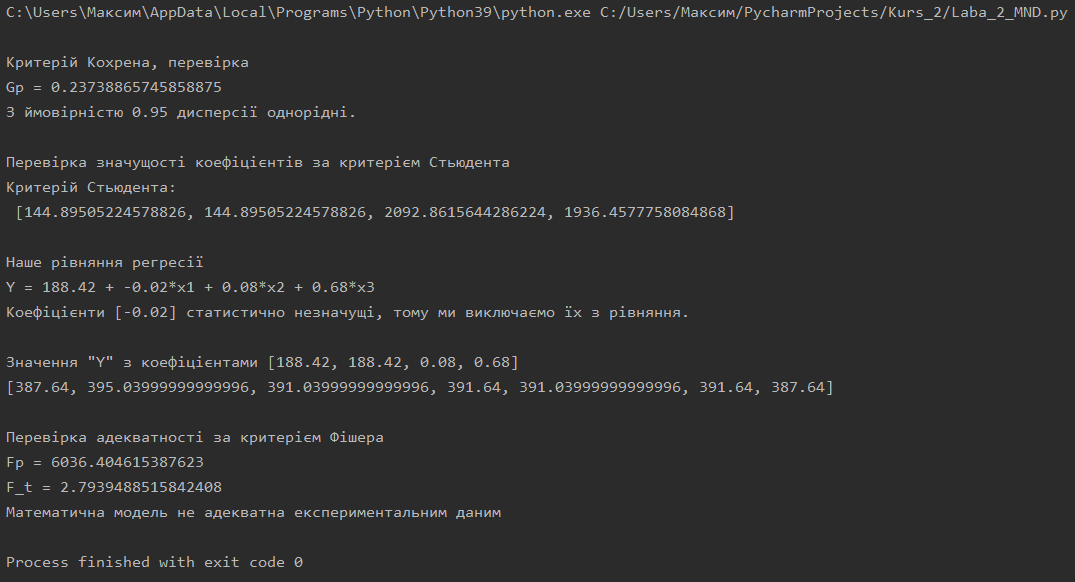
1. Для чого перевіряється критерій Стьюдента?

За допомогою критерію Стьюдента перевіряється значущість коефіцієнтів рівняння регресії.

1. Чим визначається критерій Фішера і як його застосовувати?

Критерій Фішера використовують при перевірці отриманого рівняння регресії досліджуваному об’єкту.

**Результат виконання роботи:**

****

**Висновок:**

В даній лабораторній роботі я провів дробовий трьохфакторний експеримент з трьома статистичними перевірками і отримав коефіцієнти рівняння регресії.