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 експерименту»

з теми «Проведення трьохфакторного експерименту з використанням лінійного рівняння регресії»

Виконала:

студентка ІІ курсу ФІОТ

групи ІВ-93

Трибунська Кароліна

Варіант: 326

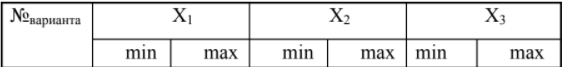
Перевірив:

Регіда П. Г.

Київ 2021

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

**Варіант**

****

****

**Програмний код**

import itertools  
import math  
import random  
import numpy as np  
import xlrd  
  
  
variant = 326  
x1\_min = -25  
x1\_max = -5  
x2\_min = -15  
x2\_max = 35  
x3\_min = -5  
x3\_max = 60  
y\_min = 200 + sum([x1\_min, x2\_min, x3\_min])/3  
y\_max = 200 + sum([x1\_max, x2\_max, x3\_max])/3  
factors\_table = [[1, -1, -1, -1],  
 [1, -1, +1, +1],  
 [1, +1, -1, +1],  
 [1, +1, +1, -1]]  
naturalized\_factors\_table = [np.array([-25, -15, -5]),  
 np.array([-25, 35, 60]),  
 np.array([-5, -15, 60]),  
 np.array([-5, 35, -5])]  
  
  
def experiment(m):  
 return [np.array([random.randint(y\_min, y\_max) for \_ in range(m)]) for i in range(4)]  
  
  
def coefficients(m, natural\_x\_table, normalized\_x\_table, y\_table):  
 def m\_i(array):  
 return np.average(array)  
  
 def a\_ij(arr1, arr2):  
 return np.average(arr1\*arr2)  
  
 def a\_jj(array):  
 return np.average(array\*\*2)  
  
 y\_averages = [np.average(i) for i in y\_table]  
  
 x1 = np.array([i[0] for i in natural\_x\_table])  
 x2 = np.array([i[1] for i in natural\_x\_table])  
 x3 = np.array([i[2] for i in natural\_x\_table])  
  
 mx1 = m\_i(x1)  
 mx2 = m\_i(x2)  
 mx3 = m\_i(x3)  
 my = m\_i(y\_averages)  
  
 a11 = a\_jj(x1)  
 a22 = a\_jj(x2)  
 a33 = a\_jj(x3)  
  
 a12 = a\_ij(x1, x2)  
 a13 = a\_ij(x1, x3)  
 a23 = a\_ij(x2, x3)  
 a1 = a\_ij(y\_averages, x1)  
 a2 = a\_ij(y\_averages, x2)  
 a3 = a\_ij(y\_averages, x3)  
  
 equations\_sys\_coefficients = [[1, mx1, mx2, mx3],  
 [mx1, a11, a12, a13],  
 [mx2, a12, a22, a23],  
 [mx3, a13, a23, a33]]  
 equations\_sys\_free\_members = [my, a1, a2, a3]  
 b\_coefficients = np.linalg.solve(equations\_sys\_coefficients,equations\_sys\_free\_members)  
  
 # normalized regression coefficients  
 b\_normalized\_coefficients = np.array([np.average(y\_averages),  
 np.average(y\_averages\*np.array([i[1] for i in normalized\_x\_table])),  
 np.average(y\_averages\*np.array([i[2] for i in normalized\_x\_table])),  
 np.average(y\_averages\*np.array([i[3] for i in normalized\_x\_table]))])  
 return b\_coefficients, b\_normalized\_coefficients  
  
  
def cochran\_criteria(m, N, y\_table):  
 # y\_name = globals()  
 # print(y\_name["y\_table"])  
 print("Перевірка рівномірності дисперсій за критерієм Кохрена: m = {}, N = {} для таблиці y\_table".format(m, N))  
 cochran\_table = xlrd.open\_workbook("Cochran.xls").sheet\_by\_index(0)  
 y\_averages = [np.average(i) for i in y\_table]  
 y\_variations = [np.var(i) for i in y\_table]  
 max\_y\_variation = max(y\_variations)  
 gp = max\_y\_variation/sum(y\_variations)  
 f1 = m - 1 # column  
 f2 = N # row  
 p = 0.95  
 q = 1-p  
 column = f1 if f1 <= 6 else 7 if f1 in [7,8] \  
 else 8 if f1 in [9, 10] \  
 else 9 if f1 in range(10, 17) \  
 else 10 if f1 in range(17, 37) \  
 else 11  
 row = f2-1 # 3rd row with value 4  
 gt = cochran\_table.row\_values(row-1)[column-1]/math.pow(10,4)  
 print("Gp = {}; Gt = {}; f1 = {}; f2 = {}; q = {:.2f}".format(gp, gt, f1, f2, q))  
 if gp < gt:  
 print("Gp < Gt => дисперсії рівномірні - все правильно")  
 return True  
 else:  
 print("Gp > Gt => дисперсії нерівномірні - треба провести додаткові експерименти")  
 return False  
  
  
def student\_criteria(m, N, y\_table, normalized\_x\_table):  
 print("\nПеревірка значимості коефіцієнтів регресії за критерієм Стьюдента: m = {}, N = {} для таблиці y\_table".format(m, N))  
 student\_table = xlrd.open\_workbook("Student.xls").sheet\_by\_index(0)  
 average\_variation = np.average(list(map(np.var, y\_table)))  
 y\_averages = np.array(list(map(np.average, y\_table)))  
 variation\_beta\_s = average\_variation/N/m  
 standard\_deviation\_beta\_s = math.sqrt(variation\_beta\_s)  
 x\_i = np.array([[el[i] for el in normalized\_x\_table] for i in range(len(normalized\_x\_table))])  
 coefficients\_beta\_s = np.array([np.average(y\_averages\*x\_i[i]) for i in range(len(x\_i))])  
 print("Оцінки коефіцієнтів βs: " + ", ".join(list(map(str,coefficients\_beta\_s))))  
 t\_i = np.array([abs(coefficients\_beta\_s[i])/standard\_deviation\_beta\_s for i in range(len(coefficients\_beta\_s))])  
 print("Коефіцієнти ts: " + ", ".join(list(map(lambda i: "{:.2f}".format(i), t\_i))))  
 f3 = (m-1)\*N  
 p = 0.95  
 q = 0.05  
 t = float(student\_table.col\_values(3)[f3].replace(",", "."))  
 importance = [True if el > t else False for el in list(t\_i)]  
  
 # print result data  
 print("f3 = {}; q = {}; tтабл = {}".format(f3, q, t))  
 beta\_i = ["β{}".format(i) for i in range(N)]  
 importance\_to\_print = ["важливий" if i else "неважливий" for i in importance]  
 to\_print = list(zip(beta\_i, importance\_to\_print))  
 x\_i\_names = [""] + list(itertools.compress(["x{}".format(i) for i in range(N)], importance))[1:]  
 betas\_to\_print = list(itertools.compress(coefficients\_beta\_s, importance))  
 print("{0[0]} {0[1]}; {1[0]} {1[1]}; {2[0]} {2[1]}; {3[0]} {3[1]}".format(\*to\_print))  
 # print(list(zip(list(map(lambda x: "{:+.2f}".format(x), betas\_to\_print)), x\_i\_names)))  
 equation = " ".join(["".join(i) for i in zip(list(map(lambda x: "{:+.2f}".format(x), betas\_to\_print)),x\_i\_names)])  
 print("Рівняння регресії без незначимих членів: y = " + equation)  
 return importance  
  
  
def calculate\_theoretical\_y(x\_table, b\_coefficients, importance):  
 x\_table = [list(itertools.compress(row, importance)) for row in x\_table]  
 b\_coefficients = list(itertools.compress(b\_coefficients, importance))  
 y\_vals = np.array([sum(map(lambda x, b: x\*b, row, b\_coefficients)) for row in x\_table])  
 return y\_vals  
  
  
def fisher\_criteria(m, N, d, naturalized\_x\_table, y\_table, b\_coefficients, importance):  
 print("\nПеревірка адекватності моделі за критерієм Фішера: m = {}, N = {} для таблиці y\_table".format(m, N))  
 fisher\_table = xlrd.open\_workbook("Fisher.xls").sheet\_by\_index(0)  
 f3 = (m - 1) \* N  
 f4 = N - d  
  
 theoretical\_y = calculate\_theoretical\_y(naturalized\_x\_table, b\_coefficients, importance)  
 theoretical\_values\_to\_print = list(zip(map(lambda x: "x1 = {0[1]}, x2 = {0[2]}, x3 = {0[3]}".format(x),naturalized\_x\_table),theoretical\_y))  
 # print(theoretical\_values\_to\_print)  
 print("Теоретичні значення y для різних комбінацій факторів:")  
 print("\n".join(["{arr[0]}: y = {arr[1]}".format(arr = el) for el in theoretical\_values\_to\_print]))  
 y\_averages = np.array(list(map(np.average, y\_table)))  
 s\_ad = m/(N-d)\*(sum((theoretical\_y-y\_averages)\*\*2))  
 y\_variations = np.array(list(map(np.var, y\_table)))  
 s\_v = np.average(y\_variations)  
 f\_p = float(s\_ad/s\_v)  
 f\_t = float((fisher\_table.row\_values(f3) if f3 <= 30 else fisher\_table.row\_values(30))[f4].replace(",","."))  
 print("Fp = {}, Ft = {}".format(f\_p, f\_t))  
 print("Fp < Ft => модель адекватна" if f\_p < f\_t else "Fp > Ft => модель неадекватна")  
 return True if f\_p < f\_t else False  
  
  
m = 4  
y\_table = experiment(m)  
while not cochran\_criteria(m, 4, y\_table):  
 m += 1  
 y\_table = experiment(m)  
print("Матриця планування:")  
labels\_table = ["x1", "x2", "x3"] + ["y{}".format(i+1) for i in range(m)]  
rows\_table = [list(naturalized\_factors\_table[i]) + list(y\_table[i]) for i in range(m)]  
rows\_normalized\_table = [factors\_table[i] + list(y\_table[i]) for i in range(m)]  
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))]))  
print("\t")  
  
print("Нормована матриця планування:")  
labels\_table = ["x{}".format(i) for i in range(4)] + ["y{}".format(i+1) for i in range(m)]  
print((" "\*4).join(labels\_table))  
print("\n".join([" ".join(map(lambda j: "{:<+5}".format(j), rows\_normalized\_table[i])) for i in range(len(rows\_normalized\_table))]))  
print("\t")  
  
  
naturalized\_coefficients, normalized\_coefficients = coefficients(m, naturalized\_factors\_table, factors\_table, y\_table)  
  
print("Рівняння регресії для нормованих факторів: y = {0} {1:+}\*x1 {2:+}\*x2 {3:+}\*x3".format(\*normalized\_coefficients))  
print("\nРівняння регресії для натуралізованих факторів: y = {0:.3f} {1:+.3f}\*x1 {2:+.3f}\*x2 {3:+.3f}\*x3".format(\*naturalized\_coefficients))  
  
importance = student\_criteria(m, 4, y\_table, factors\_table)  
factors\_table = [np.array([1]+list(i)) for i in naturalized\_factors\_table]  
correctness = fisher\_criteria(m, 4, 1, factors\_table, y\_table, naturalized\_coefficients, importance)

**Результати**

Перевірка рівномірності дисперсій за критерієм Кохрена: m = 4, N = 4 для таблиці y\_table

Gp = 0.3197177868548088; Gt = 0.6841; f1 = 3; f2 = 4; q = 0.05

Gp < Gt => дисперсії рівномірні - все правильно

Матриця планування:

x1 x2 x3 y1 y2 y3 y4

-25 -15 -5 +225 +203 +190 +196

-25 +35 +60 +225 +227 +200 +194

-5 -15 +60 +225 +209 +222 +188

-5 +35 -5 +228 +208 +227 +215

Нормована матриця планування:

x0 x1 x2 x3 y1 y2 y3 y4

+1 -1 -1 -1 +225 +203 +190 +196

+1 -1 +1 +1 +225 +227 +200 +194

+1 +1 -1 +1 +225 +209 +222 +188

+1 +1 +1 -1 +228 +208 +227 +215

Рівняння регресії для нормованих факторів: y = 211.375 +3.875\*x1 +4.125\*x2 -0.125\*x3

Рівняння регресії для натуралізованих факторів: y = 215.643 +0.387\*x1 +0.165\*x2 -0.004\*x3

Перевірка значимості коефіцієнтів регресії за критерієм Стьюдента: m = 4, N = 4 для таблиці y\_table

Оцінки коефіцієнтів βs: 211.375, 3.875, 4.125, -0.125

Коефіцієнти ts: 65.17, 1.19, 1.27, 0.04

f3 = 12; q = 0.05; tтабл = 2.1788

β0 важливий; β1 неважливий; β2 неважливий; β3 неважливий

Рівняння регресії без незначимих членів: y = +211.38

Перевірка адекватності моделі за критерієм Фішера: m = 4, N = 4 для таблиці y\_table

Теоретичні значення y для різних комбінацій факторів:

x1 = -25, x2 = -15, x3 = -5: y = 215.64326923076922

x1 = -25, x2 = 35, x3 = 60: y = 215.64326923076922

x1 = -5, x2 = -15, x3 = 60: y = 215.64326923076922

x1 = -5, x2 = 35, x3 = -5: y = 215.64326923076922

Fp = 1.5927514902761242, Ft = 3.49

Fp < Ft => модель адекватна

**Відповіді на 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?** Розрахункове значення Кохрена показує, яку частку в загальній сумі дисперсій у рядках має максимальна з них.
3. **Для чого перевіряється критерій Стьюдента?** Критерій Стьюдента використовується для перевірки значущості коефіцієнтів.
4. **Чим визначається критерій Фішера і як його застосовувати?**  Критерій Фішера використовується для перевірки адекватності рівняння регресії.

**Висновки**

Проведено дробовий трьохфакторний експеримент. Скласдено матрицю планування, знайдені коефіцієнти рівняння регресії, проведені 3 статистичні перевірки. Застосовані критерій Фішера, критерій Стьюдента, розрахункове значення Кохрена.