Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №3

з дисципліни «Методи оптимізації та планування експерименту»

# «ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконав:

студент групи ІО-82

Мартинюк Н.О.

Залікова книжка № IO-8213

Номер у списку: 12

Варіант: 212

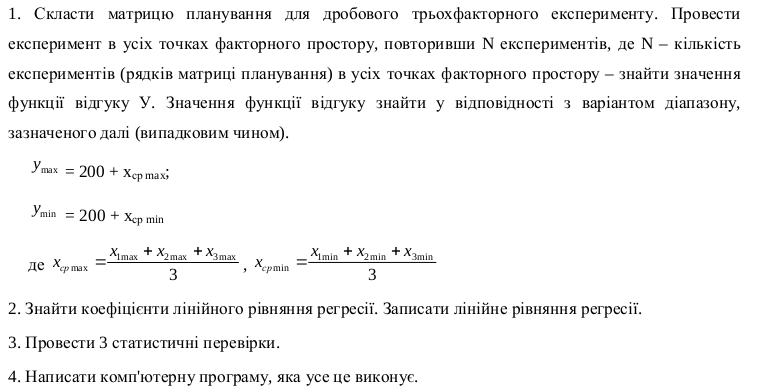
Перевірив:

ас. Регіда П.Г.

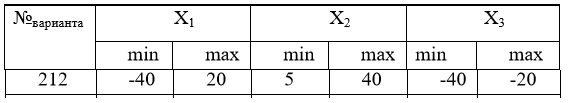
Київ 2020

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

**Завдання на лабораторну роботу:**

****

**Варіант:**



**Лістинг програми:**

import numpy as np  
from scipy.stats import t,f  
  
def table\_student(prob, f3):  
 x\_vec = [i\*0.0001 for i in range(int(5/0.0001))]  
 par = 0.5 + prob/0.1\*0.05  
 for i in x\_vec:  
 if abs(t.cdf(i, f3) - par) < 0.000005:  
 return i  
  
  
def table\_fisher(prob, d, f3):  
 x\_vec = [i\*0.001 for i in range(int(10/0.001))]  
 for i in x\_vec:  
 if abs(f.cdf(i, 4-d, f3)-prob) < 0.0001:  
 return i  
  
def make\_norm\_plan\_matrix(plan\_matrix, matrix\_of\_min\_and\_max\_x):  
 X0 = np.mean(matrix\_with\_min\_max\_x, axis=1)  
 interval\_of\_change = np.array([(matrix\_of\_min\_and\_max\_x[i, 1] - X0[i]) for i in range(len(plan\_matrix[0]))])  
 X\_norm = np.array(  
 [[round((plan\_matrix[i, j] - X0[j]) / interval\_of\_change[j], 3) for j in range(len(plan\_matrix[i]))]  
 for i in range(len(plan\_matrix))])  
 return X\_norm  
  
  
def cochran\_check(Y\_matrix):  
 fisher = table\_fisher(0.95, 1, (m - 1) \* 4)  
 mean\_Y = np.mean(Y\_matrix, axis=1)  
 dispersion\_Y = np.mean((Y\_matrix.T - mean\_Y) \*\* 2, axis=0)  
 Gp = np.max(dispersion\_Y) / (np.sum(dispersion\_Y))  
 if Gp < fisher/(fisher+(m-1)-2):  
 return True  
 return False  
  
  
def students\_t\_test(norm\_matrix, Y\_matrix):  
 mean\_Y = np.mean(Y\_matrix, axis=1)  
 dispersion\_Y = np.mean((Y\_matrix.T - mean\_Y) \*\* 2, axis=0)  
 mean\_dispersion = np.mean(dispersion\_Y)  
 sigma = np.sqrt(mean\_dispersion / (N \* m))  
 betta = np.mean(norm\_matrix.T \* mean\_Y, axis=1)  
 f3 = (m - 1) \* 4  
 t = np.abs(betta) / sigma  
 return np.where(t > table\_student(0.95, f3))  
  
  
def phisher\_criterion(Y\_matrix, d):  
 if d == N:  
 return False  
 Sad = m / (N - d) \* np.mean(check1 - mean\_Y)  
 mean\_dispersion = np.mean(np.mean((Y\_matrix.T - mean\_Y) \*\* 2, axis=0))  
 Fp = Sad / mean\_dispersion  
 f3 = (m - 1) \* 4  
 if Fp > table\_fisher(0.95, d, f3):  
 return False  
 return True  
  
  
matrix\_with\_min\_max\_x = np.array([[-40, 20], [5, 40], [-40, -20]])  
m = 6  
N = 4  
plan\_matr = np.array(  
 [np.random.randint(-40, 20, size=N), np.random.randint(5, 40, size=N), np.random.randint(-40, -20, size=N)]).T  
norm\_matrix = make\_norm\_plan\_matrix(plan\_matr, matrix\_with\_min\_max\_x)  
plan\_matr = np.insert(plan\_matr, 0, 1, axis=1)  
norm\_matrix = np.insert(norm\_matrix, 0, 1, axis=1)  
Y\_matrix = np.random.randint(200 + np.mean(matrix\_with\_min\_max\_x, axis=0)[0],  
 200 + np.mean(matrix\_with\_min\_max\_x, axis=0)[1], size=(N, m))  
mean\_Y = np.mean(Y\_matrix, axis=1)  
if cochran\_check(Y\_matrix):  
 b\_natura = np.linalg.lstsq(plan\_matr, mean\_Y, rcond=None)[0]  
 b\_norm = np.linalg.lstsq(norm\_matrix, mean\_Y, rcond=None)[0]  
 check1 = np.sum(b\_natura \* plan\_matr, axis=1)  
 check2 = np.sum(b\_norm \* norm\_matrix, axis=1)  
 indexes = students\_t\_test(norm\_matrix, Y\_matrix)  
 print("Матриця плану експерименту: \n", plan\_matr)  
 print("Нормована матриця: \n", norm\_matrix)  
 print("Матриця відгуків: \n", Y\_matrix)  
 print("Середні значення У: ", mean\_Y)  
 print("Натуралізовані коефіціенти: ", b\_natura)  
 print("Нормовані коефіціенти: ", b\_norm)  
 print("Перевірка 1: ", check1)  
 print("Перевірка 2: ", check2)  
 print("Індекси коефіціентів, які задовольняють критерію Стьюдента: ", np.array(indexes)[0])  
 print("Критерій Стьюдента: ", np.sum(np.sum(b\_natura[indexes] \* plan\_matr[:, indexes], axis=1), axis=1))  
 if phisher\_criterion(Y\_matrix, np.size(indexes)):  
 print("Рівняння регресії адекватно оригіналу.")  
 else:  
 print("Рівняння регресії неадекватно оригіналу.")  
else:  
 print("Дисперсія неоднорідна!")

**Результат підготовки та виконання програми:**

Матриця плану експерименту:

[[ 1 6 36 -29]

[ 1 2 39 -39]

[ 1 12 27 -32]

[ 1 -17 10 -23]]

Нормована матриця:

[[ 1. 0.533 0.771 0.1 ]

[ 1. 0.4 0.943 -0.9 ]

[ 1. 0.733 0.257 -0.2 ]

[ 1. -0.233 -0.714 0.7 ]]

Матриця відгуків:

[[193 187 191 187 180 193]

[206 196 209 209 181 207]

[191 176 175 201 206 187]

[201 187 176 181 205 176]]

Середні значення У: [188.5 201.33333333 189.33333333 187.66666667]

Натуралізовані коефіціенти: [ 1.55368843e+02 -3.31180017e-01 6.38556104e-02 -1.13170464e+00]

Нормовані коефіціенти: [194.07047092 -9.94250206 1.11644346 -11.31895228]

Перевірка 1: [188.5 201.33333333 189.33333333 187.66666667]

Перевірка 2: [188.5 201.33333333 189.33333333 187.66666667]

Індекси коефіціентів, які задовольняють критерію Стьюдента: [0 1 2 3]

Критерій Стьюдента: [188.5 201.33333333 189.33333333 187.66666667]

Рівняння регресії неадекватно оригіналу.

**Відповіді на контрольні запитання:**

1. 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 означає дробовий факторний експеримент (ДФЕ). Репліка, що включає тільки половину експериментів ПФЕ, називається напівреплікою, що включає четверту частину дослідів - чвертьреплікою і т. д.
2. Розрахункове значення Кохрена потрібне для порівняння цього значення з табличним, задля перевірки однорідності дисперсій.
3. Критерій Стьюдента перевіряється для того, щоб залишити в рівняння регресії лише суттєві коефіцієнти задля спрощення моделі обʼєкта. Спочатку розраховується експериментальне значення критерія для кожного коефіцієнта, після чого вони порівнюються з табличним значенням. Якщо виконується нерівність , то то приймається нуль - гіпотеза , тобто вважається , що знайдений коефіцієнт βs є статистично незначущим і його слід виключити з рівняння регресії .

Якщо , то гіпотеза не підтверджується , тобто βs – значимий коефіцієнт і він залишається в рівнянні регресії .

1. 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

**Висновок:**

Отже, ми провели дробовий трьохфакторний експеримент. Була складена матриця планування, знайдені коефіцієнти рівняння регресії та проведені 3 статистичні перевірки. Під час виконання роботи проблем не виникало. Отримані результати збігаються з очікуваними.