

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №2
з дисципліни «Методи оптимізації та планування експерименту»
на тему:
**«ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»**

Виконав:
студент групи ІВ-92
Коптюх Назар Євгенович
Залікова книжка № ІВ-9214
Варіант: 212
Перевірив:
Регіда П.Г.

Варіант:

212

-40

20

5

40

Код програми:

```
import numpy as np
import sys

#Функція для складання нормованого плану експерименту
def make_norm_plan_matrix(plan_matrix, matrix_of_min_and_max_x):
    X0 = np.array([(matrix_of_min_and_max_x[i, 0] + matrix_of_min_and_max_x[i, 1]) / 2] for i
in range(len(plan_matrix[0])))
    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

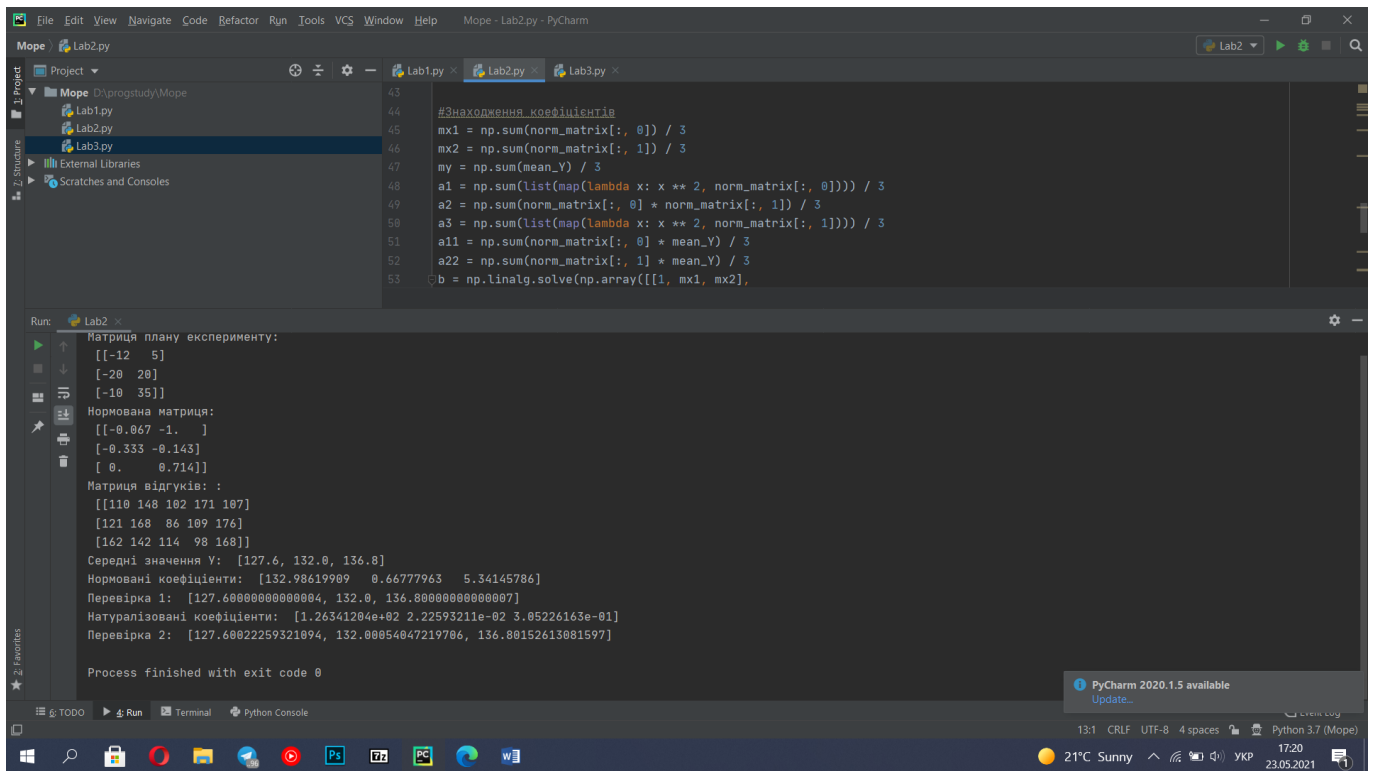
#Підготовка матриць факторів і відгуків з випадковими числами
matrix_with_min_max_x = np.array([[-40, 20], [5, 40]])
m = 5
plan_matr = np.array([np.random.randint(-25, -5, size=3), np.random.randint(5, 40, size=3)]).T
norm_matrix = make_norm_plan_matrix(plan_matr, matrix_with_min_max_x)
Y_matrix = np.random.randint(10 * (20 - 12), 10 * (30 - 12), size=(3, m))
print("Матриця плану експерименту: \n", plan_matr)
print("Нормована матриця: \n", norm_matrix)
print("Матриця відгуків: : \n", Y_matrix)

#Перевірка за критерієм Романовського
mean_Y = [np.mean(Y_matrix[i]) for i in range(len(Y_matrix))]
dispersion_Y = [np.sum([(Y_matrix[i, j] - mean_Y[i]) ** 2 for j in range(m)]) /
np.size(Y_matrix[i])
                for i in range(len(Y_matrix))]
sigma = np.sqrt((2 * (2 * m - 2)) / (m * (m - 4)))
R = []
index = -1
for i in range(len(dispersion_Y)):
    for j in range(len(dispersion_Y)):
        if i > j:
            if dispersion_Y[i] >= dispersion_Y[j]:
                R.append(abs((m - 2) * dispersion_Y[i] / (m * dispersion_Y[j]) - 1) / sigma)
            else:
                R.append(abs((m - 2) * dispersion_Y[j] / (m * dispersion_Y[i]) - 1) / sigma)
        index += 1
    if R[index] > 2.0:
        print("Дисперсія неоднорідна, спробуйте ще раз!")
        sys.exit()
print("Середні значення Y: ", mean_Y)

#Знаходження коефіцієнтів
mx1 = np.sum(norm_matrix[:, 0]) / 3
mx2 = np.sum(norm_matrix[:, 1]) / 3
my = np.sum(mean_Y) / 3
a1 = np.sum(list(map(lambda x: x ** 2, norm_matrix[:, 0]))) / 3
a2 = np.sum(norm_matrix[:, 0] * norm_matrix[:, 1]) / 3
a3 = np.sum(list(map(lambda x: x ** 2, norm_matrix[:, 1]))) / 3
a11 = np.sum(norm_matrix[:, 0] * mean_Y) / 3
a22 = np.sum(norm_matrix[:, 1] * mean_Y) / 3
b = np.linalg.solve(np.array([[1, mx1, mx2],
                              [mx1, a1, a2],
                              [mx2, a2, a3]]),
                    np.array([my, a11, a22]))
pervirkal = [(b[0] + np.sum(b[1:3] * norm_matrix[i])) for i in range(len(norm_matrix))]
print("Нормовані коефіцієнти: ", b)
print("Перевірка 1: ", pervirkal)

#Натуралізація коефіцієнтів
deltaX = [abs(matrix_with_min_max_x[i, 1] - matrix_with_min_max_x[i, 0])/2 for i in range(2)]
X0 = [(matrix_with_min_max_x[i, 1] + matrix_with_min_max_x[i, 0])/2 for i in range(2)]
a = np.array([b[0] - b[1]*X0[0]/deltaX[0] - b[2]*X0[1]/deltaX[1], b[1]/deltaX[0],
b[2]/deltaX[1]])
pervirka2 = [(a[0] + np.sum(a[1:3] * plan_matr[i])) for i in range(len(plan_matr))]
print("Натуралізовані коефіцієнти: ", a)
print("Перевірка 2: ", pervirka2)
```

Результати:



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Mope - Lab2.py - PyCharm
Mope
Project
  Mope
    Lab1.py
    Lab2.py
    Lab3.py
  External Libraries
  Scratches and Consoles
Run: Lab2
  Матриця плану експерименту:
  [[-12  5]
   [-20 20]
   [-10 35]]
  Нормована матриця:
  [[-0.067 -1.  ]
   [-0.333 -0.143]
   [ 0.    0.714]]
  Матриця відгуків:
  [[110 140 102 171 107]
   [121 168  86 109 176]
   [162 142 114  98 168]]
  Середні значення Y: [127.6, 132.0, 136.8]
  Нормовані коефіцієнти: [132.98619909  0.66777963  5.34145786]
  Перевірка 1: [127.60000000000004, 132.0, 136.80000000000007]
  Натуралізовані коефіцієнти: [1.26341284e+02 2.22593211e-02 3.05226163e-01]
  Перевірка 2: [127.60022259321094, 132.00054047219706, 136.80152613081597]
  Process finished with exit code 0
```

Контрольні питання:

1. Що таке регресійні поліноми і де вони застосовуються ?

В теорії планування експерименту найважливішою є оцінка результатів вимірів. При цьому використовують апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. В теорії планування експерименту ці поліноми отримали спеціальну назву – регресійні поліноми, а їх знаходження та аналіз – регресійний аналіз.

2. Визначення однорідності дисперсії

Обирають так названу «довірчу ймовірність» де «Р»- ймовірність з якою вимагається підтвердити гіпотезу про однорідність дисперсій. У відповідності до «Р» і кількості дослідів m обирають з таблиці критичне значення критерію. Кожне експериментальне значення R_{uv} критерію Романовського порівнюється з $R_{кр}$ (значення критерію Романовського за різних довірчих ймовірностей p) і якщо для усіх кожне $R_{uv} < R_{кр}$ то гіпотеза про однорідність дисперсій підтверджується з ймовірністю p .

3. Що називається повним факторним експериментом ?

Для знаходження коефіцієнтів у лінійному рівнянні регресії застосовують повний факторний експеримент. Якщо в багатофакторному експерименті використані всі можливі комбінації рівнів факторів, то такий експеримент називається повним факторним експериментом.

Висновки:

Я провів двофакторний експеримент, перевіряв однорідність дисперсії за критерієм Романовського, отримав коефіцієнти рівняння регресії, провів натуралізацію рівняння регресії.