

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**»

Виконав:

студент групи ІВ-91

Вігор Дмитро

Залікова книжка № 9106

Перевірив:

ас. Регіда П.Г.

Київ – 2021

ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ

Мета: провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.

Завдання на лабораторну роботу:

1. Записати лінійне рівняння регресії.
2. Обрати тип двофакторного експерименту і скласти матрицю планування для нього з використанням додаткового нульового фактору ($x_0=1$).
3. Провести експеримент в усіх точках повного факторного простору (знайти значення функції відгуку y). Значення функції відгуку задати випадковим чином у відповідності до варіанту у діапазоні $u_{\min} \div u_{\max}$.
4. Перевірити однорідності дисперсії за критерієм Романовського
5. Знайти коефіцієнти нормованих рівнянь регресії і виконати перевірку (підставити значення нормованих факторів і коефіцієнтів у рівняння).
6. Провести натуралізацію рівняння регресії й виконати перевірку натуралізованого рівняння.
7. Написати комп'ютерну програму, яка все це виконує.

$u_{\max} = (30 - N_{\text{варіанту}}) \cdot 10 = 240,$

$u_{\min} = (20 - N_{\text{варіанту}}) \cdot 10 = 140.$

Варіант завдання:

| | | | | |
|-----|----|----|----|----|
| 106 | 10 | 40 | 25 | 45 |
|-----|----|----|----|----|

Роздруківка тексту програми:

```
import numpy as np
from typing import List
np.set_printoptions(precision=3)

class Experiment:
    def __init__(self,
                  X1_range: List[int],
                  X2_range: List[int],
                  Y_range: List[int], m: int) -> None:
        self.Rcritical = {
            5: 2,
            6: 2.16,
            7: 2.3,
            8: 2.43,
            9: 2.5
        }
        self.X1_range = X1_range
        self.X2_range = X2_range
        self.Y_range = Y_range
```

```

self.plan_matrix = np.array(
    [np.random.randint(*self.X1_range, size=3),
     np.random.randint(*self.X2_range, size=3)]).T

self.x0 = [np.mean(self.X1_range), np.mean(self.X2_range)]

self.norm_matrix = self.make_norm_plan_matrix()

self.m = m

self.experiment()

self.b = self.find_b()
self.a = self.find_a()

self.check_b = self.check_b_koefs()
self.check_a = self.check_a_koefs()

def experiment(self):
    self.y_matrix = np.random.randint(*self.Y_range, size=(3, self.m))
    self.y_mean = np.mean(self.y_matrix, axis=1)

    self.y_var = np.var(self.y_matrix, axis=1)
    self.sigma = np.sqrt((2 * (2 * self.m - 2)) / (self.m * (self.m - 4)))

    if not self.check_r():
        print(f'\n Дісперсія неоднорідна! Змінимо m={self.m} to m={self.m+1}\n')
        self.m += 1
        self.experiment()

def make_norm_plan_matrix(self) -> np.array:
    self.N = self.plan_matrix.shape[0]
    self.k = self.plan_matrix.shape[1]

    interval_of_change = [self.X1_range[1] - self.x0[0],
                           self.X2_range[1] - self.x0[1]]

    X_norm = [
        [(self.plan_matrix[i, j] - self.x0[j]) / interval_of_change[j]]
        for j in range(self.k)]
    for i in range(self.N)
    ]
    return np.array(X_norm)

def check_r(self) -> bool:
    for i in range(len(self.y_var)):
        for j in range(len(self.y_var)):
            if i > j:
                if self.y_var[i] >= self.y_var[j]:
                    R = (abs((self.m - 2) * self.y_var[i] /
                               (self.m * self.y_var[j]) - 1) / self.sigma)
                else:
                    R = (abs((self.m - 2) * self.y_var[j] /
                               (self.m * self.y_var[i]) - 1) / self.sigma)
                if R > self.Rcritical[self.m]:
                    print('Variance isn\'t stable!')
                    return False

    return True

def find_b(self) -> np.array:
    mx1 = np.mean(self.norm_matrix[:, 0])
    mx2 = np.mean(self.norm_matrix[:, 1])

```

```

a1 = np.mean(self.norm_matrix[:, 0] ** 2)
a2 = np.mean(self.norm_matrix[:, 0] * self.norm_matrix[:, 1])
a3 = np.mean(self.norm_matrix[:, 1] ** 2)

my = np.mean(self.y_mean)
a11 = np.mean(self.norm_matrix[:, 0] * self.y_mean)
a22 = np.mean(self.norm_matrix[:, 1] * self.y_mean)

b = np.linalg.solve([[1, mx1, mx2],
                     [mx1, a1, a2],
                     [mx2, a2, a3]],
                     [my, a11, a22])

return b

def find_a(self) -> np.array:
    delta_x = [abs(self.X1_range[1] - self.X1_range[0]) / 2,
               abs(self.X2_range[1] - self.X2_range[0]) / 2]
    a = [(self.b[0] - self.b[1] * self.x0[0] / delta_x[0] -
          self.b[2] * self.x0[1] / delta_x[1]),
          self.b[1] / delta_x[0],
          self.b[2] / delta_x[1]]
    return np.array(a)

def check_b_koefs(self) -> np.array:
    return np.array([
        (self.b[0] + np.sum(self.b[1:3] * self.norm_matrix[i]))
        for i in range(self.N)])

def check_a_koefs(self) -> np.array:
    return np.array([
        (self.a[0] + np.sum(self.a[1:3] * self.plan_matrix[i]))
        for i in range(self.N)])

def check_results(self) -> None:
    print('Матриця планування:\n', self.plan_matrix)
    print('Нормована матриця:\n', self.norm_matrix)
    print('Матриця Y:\n', self.y_matrix)
    print('\nНормовані коефіцієнти: ', self.b)
    print('Натуралізовані коефіцієнти:', self.a)
    print('\nY середнє: ', self.y_mean)
    print('Перевірка нормованих коефіцієнтів: ', self.check_b)
    print('Перевірка натуралізованих коефіцієнтів: ', self.check_a)

if __name__ == '__main__':
    m = 5
    X1_range = [10, 40]
    X2_range = [25, 45]
    Y_range = [140, 240]
    res = Experiment(X1_range, X2_range, Y_range, m)
    res.check_results()

```

Результати роботи програми:

Матриця планування:

[[24 38]

[36 38]

[28 44]]

Нормована матриця:

[[-0.067 0.3]

[0.733 0.3]

[0.2 0.9]]

Матриця Y:

[[177 187 151 200 239]

[165 218 231 145 235]

[224 220 219 224 177]]

Нормовані коефіцієнти: [181.8 10. 32.222]

Натуралізовані коефіцієнти: [52.356 0.667 3.222]

Y середнє: [190.8 198.8 212.8]

Перевірка нормованих коефіцієнтів: [190.8 198.8 212.8]

Перевірка натуралізованих коефіцієнтів: [190.8 198.8 212.8]

Process finished with exit code 0

Відповіді на контрольні питання:

1) *Що таке регресійні поліноми і де вони застосовуються?*

Регресійні поліноми – це апроксимуючі поліноми, за допомогою яких ми можемо описати функцію. Застосовуються в теорії планування експерименту.

2) *Визначення однорідності дисперсії.*

Опираючись на вимоги регресивного аналізу достовірне оброблення та використання вихідних даних експериментальних досліджень можливе лише тоді, коли дисперсії вимірювання функцій відгуку в кожній точці експерименту є однаковими. Дана властивість називається однорідністю дисперсії.

3) *Що називається повним факторним експериментом?*

ПФЕ – багатофакторний експеримент в якому використовуються всі можливі комбінації рівні факторів. $N_{\text{ПФЕ}} = 2^k$ або 3^k або 5^k .

Висновки:

В ході виконання лабораторної роботи було проведено двофакторний експеримент, перевірено однорідність дисперсії за критерієм Романовського, отримано коефіцієнти рівняння регресії, проведено натуралізацію рівняння регресії.