Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Методи оптимізації та планування експерименту

ЛАБОРАТОРНА РОБОТА № 2

«ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАЛА:

студентка II курсу ФІОТ

групи ІО-92

Варіант №211

Карнаухова Анастасія

ПЕРЕВІРИВ:

Доц. Порєв В.М.

Київ – 2021

ЛАБОРАТОРНА РОБОТА № 2

Тема: 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.

1. Записати лінійне рівняння регресії.
2. Обрати тип двофакторного експерименту і скласти матрицю планування для нього з використанням додаткового нульового фактору (x_o=1).
3. Провести експеримент в усіх точках повного факторного простору (знайти значення функції відгуку у). Значення функції відгуку задати випадковим чином у відповідності до варіанту у діапазоні $y_{min} \div y_{max}$

   $y_{max} = (30 - N_{варіанту})*10,$
   $y_{min} = (20 - N_{варіанту})*10.$

   Варіанти обираються по номеру в списку в журналі викладача.
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.

Завдання на лабораторну роботу:

211	10	60	-35	15

Варіант завдання:

Код програми:

```
import random as ran
import math as ma
import numpy
from prettytable import PrettyTable

table0 = PrettyTable()
table0.field_names = (["Студент", "Группа"])
name = "Карнаухова Анастасія"
group = "IO-92"
table0.add_row([name, group])
print(table0)

def inf(count, mean):
    array_x1 = [10, 60]
```

```python
m = 5 + count
array_x2 = [-35, 15]
array_y = [(30 - 11) * 10, (20 - 11) * 10]
teta0 = ma.sqrt(2 * (2 * m - 2) / (m * (m - 4)))
matr_x = []
matr_x2 = []
matr_y = [[ran.randint(array_y[1], array_y[0]) for j in range(m)] for i in range(3)]
avey = []
sigma = []
array_fuv = []
array_a = []
array_aij = []
for i in range(3):

    if i == 0:
        matr_x.append([-1, -1])
        matr_x2.append([min(array_x1), min(array_x2)])
    elif i == 1:
        matr_x.append([-1, 1])
        matr_x2.append([min(array_x1), max(array_x2)])
    elif i == 2:
        matr_x.append([1, -1])
        matr_x2.append([max(array_x1), min(array_x2)])
for i in range(len(matr_y)):
    sumer = 0
    temp = sum(matr_y[i]) / len(matr_y[i])
    avey.append(temp)
    for j in range(len(matr_y[i])):
        sumer += (matr_y[i][j] - temp) ** 2
    sigma.append(sumer / len(matr_y[i]))
array_fuv.append(sigma[0] / sigma[1])
array_fuv.append(sigma[2] / sigma[0])
array_fuv.append(sigma[2] / sigma[1])
array_tetas = [(m - 2) / m * array_fuv[i] for i in range(len(array_fuv))]
array_ruv = [ma.fabs(i - 1) / teta0 for i in array_tetas]
for i in range(len(array_ruv)):
    if array_ruv[i] < 2:
        mean = True
        print("R{0}uv > 2".format(i))
    else:
        mean = False
        print("R{0}uv < 2".format(i))
if mean:
    trans = numpy.array(matr_x).transpose()
    array_mx = [sum(trans[i]) / len(trans[i]) for i in range(2)]
    my = sum(avey) / len(avey)
    for i in range(2):
        temp = 0
        if i == 1:
            for j in matr_x:
                temp += numpy.array(j).prod()
            array_a.append(temp / 3)
            temp = 0
        for j in range(len(trans[i])):
            temp += (trans[i][j] ** 2)
        array_a.append(temp / 3)
    for i in range(2):
        temp = 0
        for j in range(len(trans[i])):
```

```
            temp += trans[i][j] * avey[j]
        array_aij.append(temp / 3)
    first = numpy.array(
        [[1, array_mx[0], array_mx[1]], [array_mx[0], array_a[0], array_a[1]],
         [array_mx[1], array_a[1], array_a[2]]])
    second = numpy.array([my, array_aij[0], array_aij[1]])
    res = numpy.linalg.solve(first, second)
    array_delx = [(max(array_x1) - min(array_x1)) / 2, (max(array_x2) - min(array_x2)) / 2]
    array_zerx = [sum(array_x1) / 2, sum(array_x2) / 2]
    a0 = res[0] - res[1] * (array_zerx[0] / array_delx[0]) - res[2] * (array_zerx[1] / array_delx[1])
    a1 = res[1] / array_delx[0]
    a2 = res[2] / array_delx[1]
    ta = PrettyTable()
    ta.field_names = ["X1", "X2", "Y1", "Y2", "Y3", "Y4", "Y5"]
    ta.add_rows(
        [
            [matr_x[0][0], matr_x[0][1], matr_y[0][0], matr_y[0][1], matr_y[0][2], matr_y[0][3], matr_y[0][4]],
            [matr_x[1][0], matr_x[1][1], matr_y[1][0], matr_y[1][1], matr_y[1][2], matr_y[1][3], matr_y[1][4]],
            [matr_x[2][0], matr_x[2][1], matr_y[2][0], matr_y[2][1], matr_y[2][2], matr_y[2][3], matr_y[2][4]],
        ]
    )
    print("m = ", m)
    print("Матриця планування для m = 5")
    print(ta)
    print("Нормованні значення X1 та X2:\n", matr_x)
    print("Значення функції відгуку при m = {0}:\n".format(m), numpy.array(matr_y))
    print("Середнє значення функції відгуку:\n", avey)
    print("Матиматичне очікування X1 та X2:\n", array_mx)
    print("Значення a:\n", array_a)
    print("Значення aij:\n", array_aij, "\n")
    print("Нормоване рівняння регресії")
    print("y = {0} + {1}*x1 + {2}*x2\n".format(res[0], res[1], res[2]))
    print("Значення дисперсій:\n", sigma)
    print("Значення Fuv:\n", array_fuv)
    print("Значення θuv:\n", array_tetas)
    print("Значення Ruv:\n", array_ruv)

    print("Зробимо перевірку:")
    for i in range(len(matr_x)):
        check = res[0] + res[1] * matr_x[i][0] + res[2] * matr_x[i][1]
        print("y{0} = {1}".format(i, check))
    print("\n")
    print("Натуралізоване рівняння регресії")
    print("y = {0} + {1}*x1 + {2}*x2\n".format(a0, a1, a2))
    print("Зробимо перевірку:")
    for i in range(len(matr_x2)):
        check = a0 + a1 * matr_x2[i][0] + a2 * matr_x2[i][1]
        print("y{0} = {1}".format(i, check))
    return mean
else:
    return mean


a = 0
while True:
    b = False
    if inf(a, b):
        n = input("Введіть \"Кінець\" щоб зупинити програму: ")
        if n == "Кінець":
```

```
        break
    else:
        n = input("Введіть \"Кінець\" щоб зупинити програму:")
        if n == "Кінець":
            break
        print("Збільшуємо m на 1")
        a += 1
```

Результат роботи програми:

```
+----------------------+--------+
|       Студент        | Группа |
+----------------------+--------+
| Карнаухова Анастасія | ІО-92  |
+----------------------+--------+
R0uv > 2
R1uv > 2
R2uv > 2
m =  5
Матриця планування для m = 5
+----+----+-----+-----+-----+-----+-----+
| X1 | X2 |  Y1 |  Y2 |  Y3 |  Y4 |  Y5 |
+----+----+-----+-----+-----+-----+-----+
| -1 | -1 | 132 |  98 | 155 | 138 | 148 |
| -1 | 1  | 125 | 141 | 154 | 166 |  96 |
| 1  | -1 |  93 | 152 |  99 | 158 | 157 |
+----+----+-----+-----+-----+-----+-----+
Нормованні значення X1 та X2:
 [[-1, -1], [-1, 1], [1, -1]]
Значення функції відгуку при m = 5:
 [[132  98 155 138 148]
 [125 141 154 166  96]
 [ 93 152  99 158 157]]
Середнє значення функції відгуку:
 [134.2, 136.4, 131.8]
Матиматичне очікування X1 та X2:
 [-0.3333333333333333, -0.3333333333333333]
Значення a:
 [1.0, -0.3333333333333333, 1.0]
Значення aij:
 [-46.26666666666667, -43.199999999999996]

Нормоване рівняння регресії
y = 134.10000000000002 + -1.1999999999999957*x1 + 1.100000000000012*x2

Значення дисперсій:
 [390.56, 593.8399999999999, 862.1600000000001]
Значення Fuv:
 [0.6576855718712112, 2.207496927488734, 1.45183887915937]
Значення θuv:
 [0.3946113431227267, 1.3244981564932405, 0.8711033274956219]
Значення Ruv:
 [0.3384225473962197, 0.18139998412306266, 0.07205543044832936]
Зробимо перевірку:
y0 = 134.2
y1 = 136.40000000000003
y2 = 131.8


Натуралізоване рівняння регресії
y = 136.22000000000003 + -0.04799999999999983*x1 + 0.04400000000000048*x2

Зробимо перевірку:
y0 = 134.20000000000002
y1 = 136.40000000000003
y2 = 131.8
```

Висновок:

Пробивши лабораторну роботу, було проведено трьохфакторний експеримент з урахуванням квадратичних членів, використовуючи центральний ортогональний композиційний план. Та знайдено рівняння регресії адекватне об'єкту. У ході виконання лабораторної роботи проблем не виникло. Результати виконання лабораторної висвітлені на роздруківках.

Контрольні запитання

1) **Що таке регресійні поліноми і де вони застосовуються?**
   В теорії планування експерименту найважливішою частиною є оцінка результатів вимірів. При цьому використовують апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. В ТПЕ ці поліноми отримали спеціальну назву - регресійні поліноми
2) **Визначення однорідності дисперсії.**
   Для цього необхідно спочатку знайти середньоарифметичне значення дослідів $\overline{yj}$ (j=1,$\overline{m}$) (математичне сподівання myj) в кожній точці факторного простору: $\overline{y}$j=(1/m)∑ **y m 1** js(i=1,$\overline{N}$). Оскільки теоретичні значення дисперсії σ 2 j (j=1,$\overline{N}$) невідомі, то перевірка однорідності дисперсії виконується на основі аналізу статистичних оцінок дисперсії S 2 j (i=1,$\overline{N}$) для усіх точок факторного простору. Статистичні оцінки дисперсії S 2 j (j=1,$\overline{N}$) для кожної точки факторного простору розрахо-вують за формулою: S 2 j ={1/(m-1)}{∑ (**y m 1** js-$\overline{y}$j) 2 } (j=1,$\overline{N}$). Отже, перевірка однорідності дисперсії – це перевірка гіпотези стосовно належності N значень статистичних оцінок дисперсії S 2 j (i=1,$\overline{N}$) одній генеральній сукупності.
3) **Що називається повним факторним експериментом?**
   ПФЕ – повний факторний експеримент,- це коли використовуються усі можливі комбінації рівнів факторів; при ПФЕ кількість комбінацій Nп=r k .