

```
clc;
close all;
clear;
```

```
threshold = 5.99;
```

۱. مقدار آستانه مشخص می‌گردد.

```
fID = fopen('Iris.txt');
fData = textscan(fID, '%f, %f, %f, %f, %s');
fclose(fID);
a = cell2mat(fData{1, 1});
b = cell2mat(fData{1, 2});
p = cell2mat(fData{1, 3});
q = cell2mat(fData{1, 4});
class = fData{1, 5};
```

۲. فایل Iris.txt خوانده شده و مقادیر خوانده شده به متغیر سلولی fData انتساب داده می‌شوند.

```
[row, ~] = size(class);
c = zeros(row);
for i = 1:row
    if strcmp(class(i, 1), 'Iris-setosa') == 1
        c(i, 1) = 1;
    elseif strcmp(class(i, 1), 'Iris-versicolor') == 1
        c(i, 1) = 2;
    elseif strcmp(class(i, 1), 'Iris-virginica') == 1
        c(i, 1) = 3;
    end
end
```

۳. تعداد سطرهای متغیر class مشخص گردیده و در یک حلقه تکرار مقادیر ۱، ۲ و ۳ به ترتیب جایگزین مقادیر نامی سه کلاس Iris-virginica، Iris-versicolor و Iris-setosa می‌شوند.

```
for col = 1:4
    data = [sortrows([a, b, p, q, c], col), (1:numel(a(:, 1)))];
```

۴. متغیرهای a، b، p، q و c به یکدیگر ملحق شده، برحسب ستون مشخص شده توسط متغیر col به صورت صعودی مرتب می‌شوند و پس از آن به یک بردار ستونی ۱۵۰ عضوی دیگر که مقادیر درایه‌های این بردار اعداد ۱ تا ۱۵۰ هستند ملحق می‌گردند و درنهایت ماتریس به دست آمده به متغیر data انتساب داده می‌شود.

```
uniqueValue = unique(data(:, col));
for i = 1:numel(uniqueValue(:, 1))
    dataIndex = find(uniqueValue(i) == data(:, col));
    data(dataIndex, 6) = i;
end
```

۵. مقادیر یکتای ستون موردنظر (ستون انتخابی جهت گروه‌بندی) مشخص می‌شوند و اعداد یکسان در بازه‌های یکسانی جای خواهند گرفت؛ بنابراین مقادیر ستون ۶-ام بر اساس گروه‌بندی‌ها مجدداً تعیین خواهد شد. به عنوان مثال، اگر سطر ۲-ام تا ۴-ام ستون ۱-ام اعداد یکسانی باشند آنگاه همگی در بازه ۲ جای خواهند گرفت و بنابراین مقادیر متناظر آن‌ها در ستون ۶-ام، ۲ خواهد شد.

```
table.Set = [];
t = repmat(table, [numel(uniqueValue(:, 1)), 1]);
```

۶. استراکچر table که دارای فیلد Set است تعریف شده و متغیری به نام t به تعداد بازه‌های مشخص شده، از نوع table ایجاد می‌گردد.

```
for it = 1:100
```

```
    UniqueInterval = unique(data(:, 6));
```

۷. تعداد بازه‌های یکتا، با توجه به ستون ۶-ام متغیر UniqueInterval به متغیر data انتساب داده می‌شوند.

```
if numel(UniqueInterval(:, 1)) == 6
    break;
end
```

۸. اگر تعداد بازه‌های یکتا برابر با ۶ شد آنگاه برنامه از حلقه اصلی محاسباتی خارج شده و خروجی در خط فرمان چاپ خواهد شد.

```
UniqueInterval = [UniqueInterval, zeros(numel(UniqueInterval(:, 1)), 1)];
A = zeros(numel(UniqueInterval(:, 1)), 3);
test = zeros(numel(UniqueInterval(:, 1))-1, 1);
for i = 1:numel(UniqueInterval(:, 1))-1
    for j = i:i + 1
        for k = 1:3
            A(j, k) = sum(data(find(UniqueInterval(j, 1) == data(:, 6)), 5) == k);
        end
    end
    C1 = sum(A(i:i+1, 1));
    C2 = sum(A(i:i+1, 2));
    C3 = sum(A(i:i+1, 3));
    R1 = sum(A(i, :));
    R2 = sum(A(i+1, :));

    E11 = (R1 * [C1, C2, C3]) ./ sum([C1, C2, C3]);
    index = find(E11 == 0);
    E11(index) = 0.1;

    E21 = (R2 * [C1, C2, C3]) ./ sum([C1, C2, C3]);
    index = find(E21 == 0);
    E21(index) = 0.1;

    UniqueInterval(i, 2) = sum(sum(((A(i:i+1, :) - [E11; E21]).^2)./[E11; E21]));
end
```

۹. یک ستون ۰ (به تعداد سطرهای ماتریس UniqueInterval) تولید شده و به ماتریس UniqueInterval الحاق می‌گردد. همچنین، ماتریس A به تعداد سطرهای ماتریس UniqueInterval و با سه ستون و ماتریس test به اندازه یک کمتر از تعداد سطرهای ماتریس ایجاد می‌گردد.

در ادامه، دوستهٔ مجاور با یکدیگر مورد مقایسه قرار گرفته و  $A_{11}$  تا  $A_{23}$  محاسبه شده و با جمع هر یک از ستون‌های ماتریس A، تعداد اعضاء هر کلاس محاسبه شده و با جمع هر یک از سطرهای ماتریس A تعداد اعضاء هر دسته مشخص می‌شود. مقادیر  $E_{11}$  و  $E_{21}$  محاسبه شده و در صورتی‌که برابر با ۰ باشند مقدار ۱/۰ به آنها اختصاص داده می‌شود. در انتها در ستون دوم ماتریس UniqueInterval مقدار کای-۲ به دست آمده حاصل از مقایسه هر دسته با دستهٔ بعد از خود، در سطر مربوط به دستهٔ کوچک‌تر درج می‌گردد و بدین شکل ما مقدار کای-۲ ناشی از مقایسه هر دسته با دستهٔ بعدی را در اختیار خواهیم داشت.

```
if min(UniqueInterval(1:numel(UniqueInterval(:, 1))-1, 2)) < threshold
    test = UniqueInterval(1:numel(UniqueInterval(:, 1))-1, 2) ==
min(UniqueInterval(1:numel(UniqueInterval(:, 1))-1, 2));
end
```

۱۰. اگر در میان درایه‌های ستون دوم ماتریس UniqueInterval عضوی یافت شود که مقدار آن از مقدار آستانه کمتر باشد، آنگاه آرایه test مقداردهی می‌گردد. نحوه مقداردهی بدین شکل است که تمامی درایه‌های ستون دوم

ماتریس UniqueInterval با مقدار کمینه ستون دوم ماتریس UniqueInterval مقایسه شده و مقادیر برابر باعث می‌شوند که سطرهای متناظر آرایه test با ۱ شوند و مقادیر نابرابر سطرهای متناظر آرایه test را ۰ می‌کنند.

```
j = 1;
t(j).Set = [t(j).Set, 1];
for i = 1:numel(test(:))
    if test(i) == 1
        t(j).Set = [t(j).Set, i + 1];
    else
        j = j + 1;
        t(j).Set = [t(j).Set, i + 1];
    end
end
```

۱۱. ابتدا در فیلد مجموعه درایه اول آرایه سلولی t مقدار ۱ درج می‌گردد و در یک حلقه تکرار از مقدار اولیه ۱ تا تعداد اعضای آرایه test، این موارد بررسی می‌گردد: اگر مقدار درایه i-ام آرایه test برابر ۱ باشد آنگاه به فیلد مجموعه درایه j-ام آرایه سلولی t مقدار ۱+۱ نیز اضافه می‌گردد و در غیر این صورت زیک واحد اضافه شده و در فیلد مجموعه درایه j-ام آرایه سلولی t مقدار ۱+۱ درج می‌گردد.

```
for i = 1:numel(t)
    if ~isempty(t(i).Set)
        for j = 1:numel(t(i).Set)
            [I, J] = find(t(i).Set(j) == data(:, 6));
            data(I, 6) = i;
        end
    end
end
```

۱۲. در یک حلقه تکرار از مقدار اولیه ۱ تا تعداد آرایه سلولی t، تا جایی که به مجموعه تهی نرسیم اندیس‌های درج شده در هر مجموعه را شناسایی کرده و مقادیر ستون ۶-ام و سطور مربوط به اندیس‌های یافته شده را در ماتریس data برابر با شمارنده آ قرار می‌دهیم.

```
for i = 1:numel(t)
    t(i).Set = [];
end

g = zeros(numel(unique(data(:, 6))), 2);
for i = 1:numel(unique(data(:, 6)))
    g(i, 1) = min(data(find(data(:, 6) == i), col));
    g(i, 2) = max(data(find(data(:, 6) == i), col));
end
```

۱۳. در انتهای فرایند، کمینه و بیشینه هر دسته را یافته و در ستون اول و دوم سطر آ-ام ماتریس و درج می‌نماییم (از این ماتریس در نمایش بازه‌ها استفاده می‌کنیم).

```
display('-----');
switch col
    case 1
        disp('Grouping based on Sepal Length');
        disp('-----');
    case 2
        disp('Grouping based on Sepal Width');
        disp('-----');
    case 3
        disp('Grouping based on Petal Length');
        disp('-----');
    case 4
        disp('Grouping based on Petal Width');
```

```
    disp('-----');
end
disp(g);
end
```

خروجی برنامه:

-----  
Grouping based on Sepal Length  
-----

```
4.3000 4.8000
4.9000 4.9000
5.0000 5.4000
5.5000 5.7000
5.8000 7.0000
7.1000 7.9000
```

-----  
Grouping based on Sepal Width  
-----

```
2.0000 2.2000
2.3000 2.4000
2.5000 2.8000
2.9000 2.9000
3.0000 3.3000
3.4000 4.4000
```

-----  
Grouping based on Petal Length  
-----

```
1.0000 1.9000
3.0000 4.4000
4.5000 4.7000
4.8000 4.9000
5.0000 5.1000
5.2000 6.9000
```

-----  
Grouping based on Petal Width  
-----

```
0.1000 0.6000
1.0000 1.3000
1.4000 1.6000
1.7000 1.7000
1.8000 1.8000
1.9000 2.5000
```