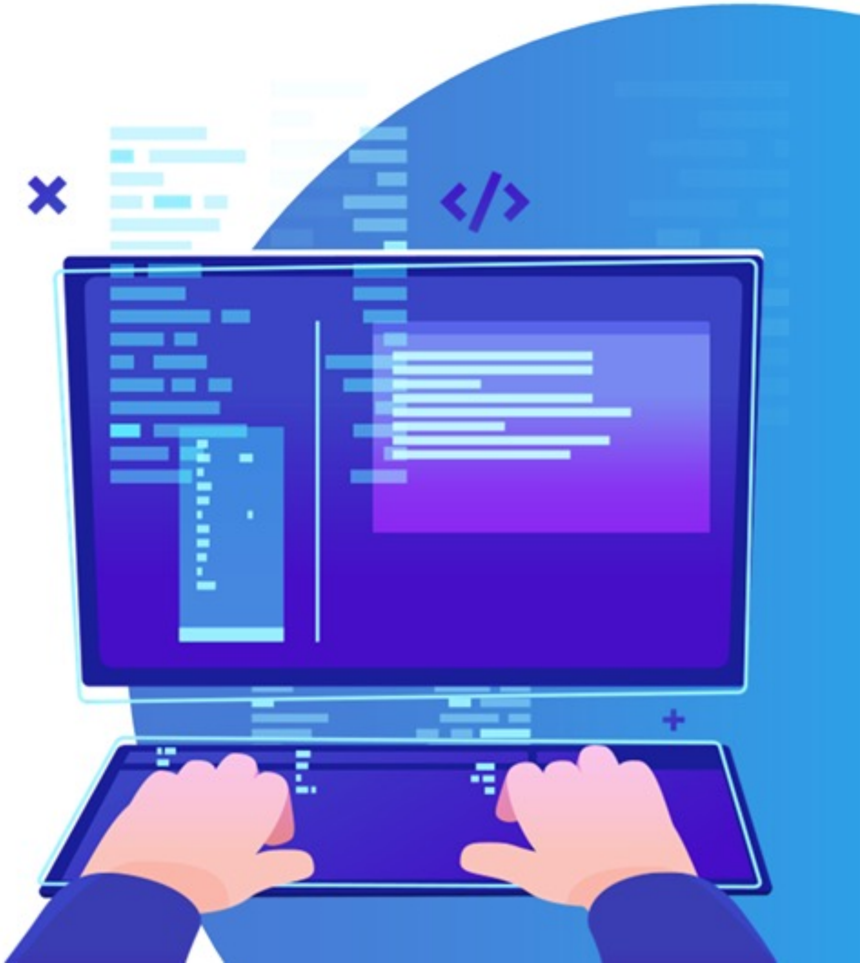
Topic 2

K-means algorithm implemented from the scratch in Matlab

2022.04.25 / Jueun Park (PAR0161)

CONTENTS



CHPATER 01

What is k-means algorithm?

The k-means clustering algorithm is an algorithm that combines a given data into k clusters, and operates in a way that minimizes the variance of distance differences with each cluster. This algorithm is a kind of autonomous learning, which serves to label unlabeled input data.

What is k-means algorithm?

If the center of the i -th cluster is μ_i and the set of points belonging to the cluster is S_i, the overall variance is calculated as follows.

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} |x_j - \mu_i|^2$$

Finding S_i that minimizing this value becomes the goal of the algorithm.
The algorithm first begins by setting the initial μ_i.
Then repeat the following two steps.

What is k-means algorithm?

1. Setting the Cluster :

Euclidean distance from each data to μ_i of each cluster is calculated, and data is allocated by finding the nearest cluster from the corresponding data.

$$S_i^{(t)} = \{x_p : |x_p - \mu_i^{(t)}|^2 \leq |x_p - \mu_j^{(t)}|^2 \forall j, 1 \leq j \leq k\}$$

2. Cluster-centroid rebalancing:

Reset μ_i to the center of gravity value of the data in each cluster.

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

If the cluster does not change, stop repeating.

What is k-means algorithm?

Euclidean distance

Euclidean distance is a common method of calculating the distance between two points. This distance can be used to define the Euclidean space, and the norm corresponding to this distance is called the Euclidean norm.

When points p = (p1, p2,..., pn) and q = (q1, q2,..., qn) represented by an orthogonal coordinate system, the distance between two points p, q is calculated using two Euclidean norms.

$$\|\mathbf{p}-\mathbf{q}\| = \sqrt{(\mathbf{p}-\mathbf{q})\cdot(\mathbf{p}-\mathbf{q})} = \sqrt{\|\mathbf{p}\|^2 + \|\mathbf{q}\|^2 - 2\mathbf{p}\cdot\mathbf{q}}.$$

What is k-means algorithm?

Complexity

Two factors that greatly affect the computational complexity of the k-mean algorithm are **Euclidean space d** and **the number of clusters k**.
Even if the number of clusters is small, it is NP-hard to find the optimal solution of the k-means algorithm in the general Euclidean space d. Conversely, even in a low-dimensional Euclidean space, finding the optimal solution for k clusters is also NP-hard.
(*NP = Non-deterministic Polynomial time)

The time complexity is proportional to the number of clusters, the dimension of the data vector, and the number of data vectors, respectively. In addition, since the algorithm repeats until the solution converges, the algorithm is proportional to the number of iterations. That is, the time taken to group n d-dimensional data vectors into k clusters through i-times of repetition is $O(nkdi)$

For space complexity, the algorithm must store additional information about the center of gravity vectors of each cluster, and since these center of gravity vectors are independent every iteration, the k-means algorithm has a space complexity of $O((n+k)d)$

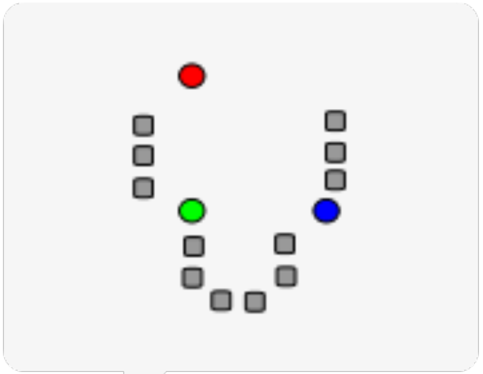
What is k-means algorithm?

Limitation

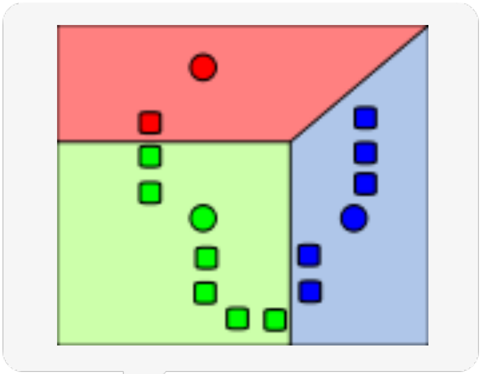
1. The value of the number of clusters k should be specified as an input parameter.
2. There is a possibility that the error convergence of the algorithm will converge to a local minimum rather than a global minimum.
3. It is sensitive to outliers.
4. It is not suitable for finding clusters that are not spherical.

Implement k-means algorithm

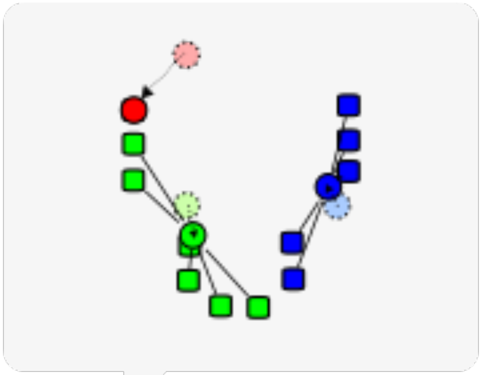
The execution process of a standard algorithm



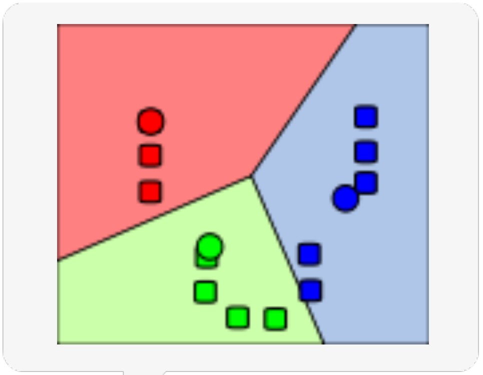
1) The initial k "average value" (k=3 in this case) is randomly selected from the data objects. (shown in colored circles).



2) Each data object is grouped based on the nearest average value.



3) The average value is readjusted based on the center point of k clusters.



4) Repeat steps 2) and 3) until convergence.

Implement k-means algorithm

Main Idea

Input
k: Number of Clusters
D: A set containing n data objects

Output
K clusters

Algorithm
1. From the data object set D, k data objects are randomly extracted, and these data objects are set to the center of each cluster. (Set initial value)
2. For each data object in the set D, obtain the distance from each of the k cluster-centered objects, and find which center is most similar each data object. Then, each data object is assigned to the center point found in that way.
3. Recalculate the center point of the cluster. That is, the center point is recalculated based on the reallocated clusters in (2).
4. Steps 2 and 3 are repeated until the cluster belonging to each data object is not changed.

Implement k-means algorithm

Code

Two-dimensional spatial data, one of the most basic cluster data, is used as an example. K-Means algorithm initialization step. Optionally, select an element of input data D to initialize uk.

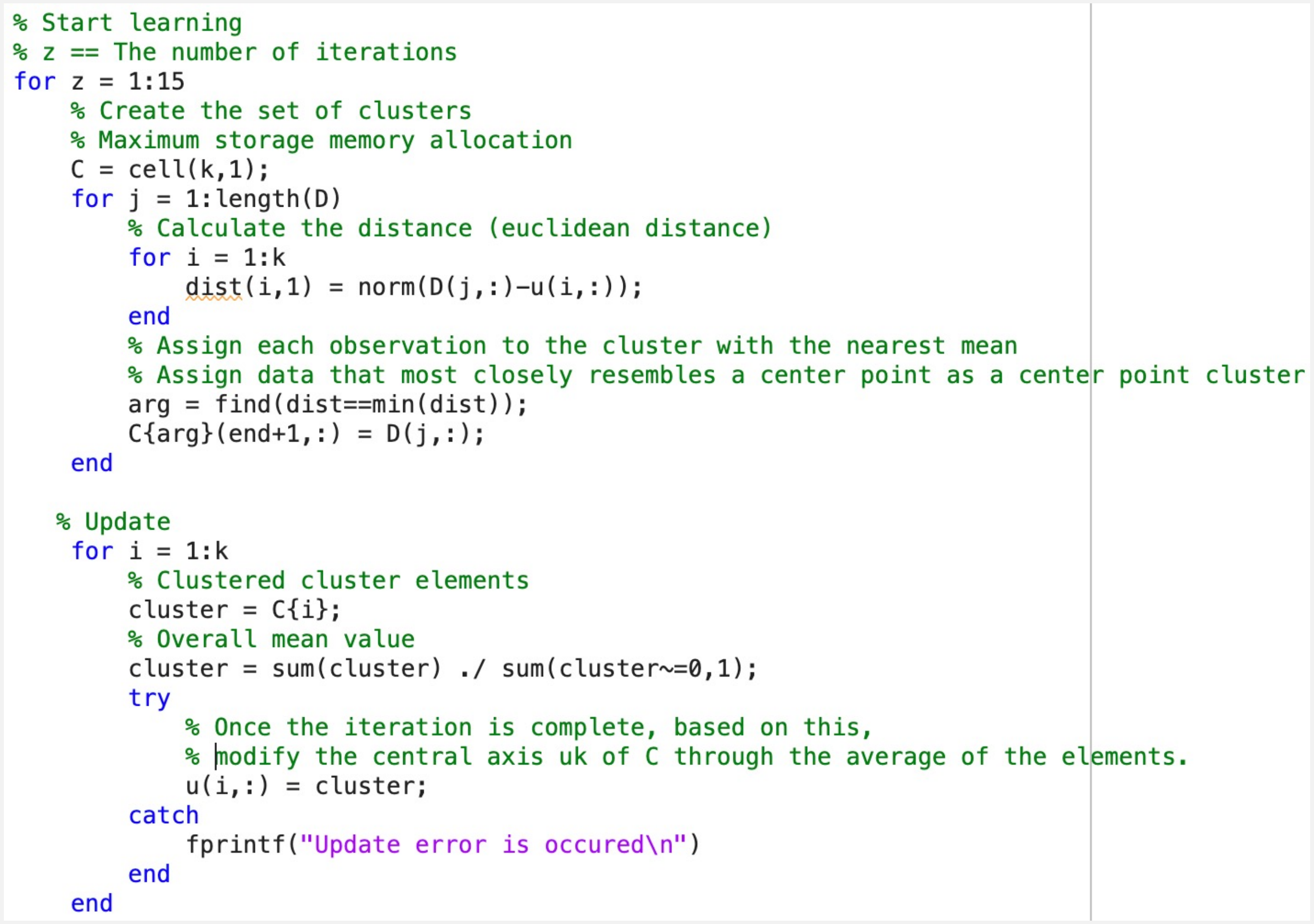
```
D = rand(500, 2); % 500 samples with 2 features

% The number of clusters
k = 2;

% Initialization the centroid
% To initialize 'u', the element of input data D is randomly selected.
random = randperm(length(D),k);
u = D(random ,:);
```

***Random Partition**

After each data is allocated to an arbitrary cluster, the average value of the points allocated to each cluster is set to the initial u.

Implement k-means algorithm

Code



```
% Start learning
% z == The number of iterations
for z = 1:15
    % Create the set of clusters
    % Maximum storage memory allocation
    C = cell(k,1);
    for j = 1:length(D)
        % Calculate the distance (euclidean distance)
        for i = 1:k
            dist(i,1) = norm(D(j,:)-u(i,:));
        end
        % Assign each observation to the cluster with the nearest mean
        % Assign data that most closely resembles a center point as a center point cluster
        arg = find(dist==min(dist));
        C{arg}(end+1,:) = D(j,:);
    end

   % Update
    for i = 1:k
        % Clustered cluster elements
        cluster = C{i};
        % Overall mean value
        cluster = sum(cluster) ./ sum(cluster~=0,1);
        try
            % Once the iteration is complete, based on this,
            % modify the central axis uk of C through the average of the elements.
            u(i,:) = cluster;
        catch
            fprintf("Update error is occured\n")
        end
    end
```

Initialize the cluster set C so that it can be updated every time the learning is repeated. Then, scan all the data D to find the distance from Ck, and set it as an element of the set C with the nearest Centroid value.

Once repeated, the center axis uk of C is modified through the average of the elements based on .

Working for 1D data

[setting]

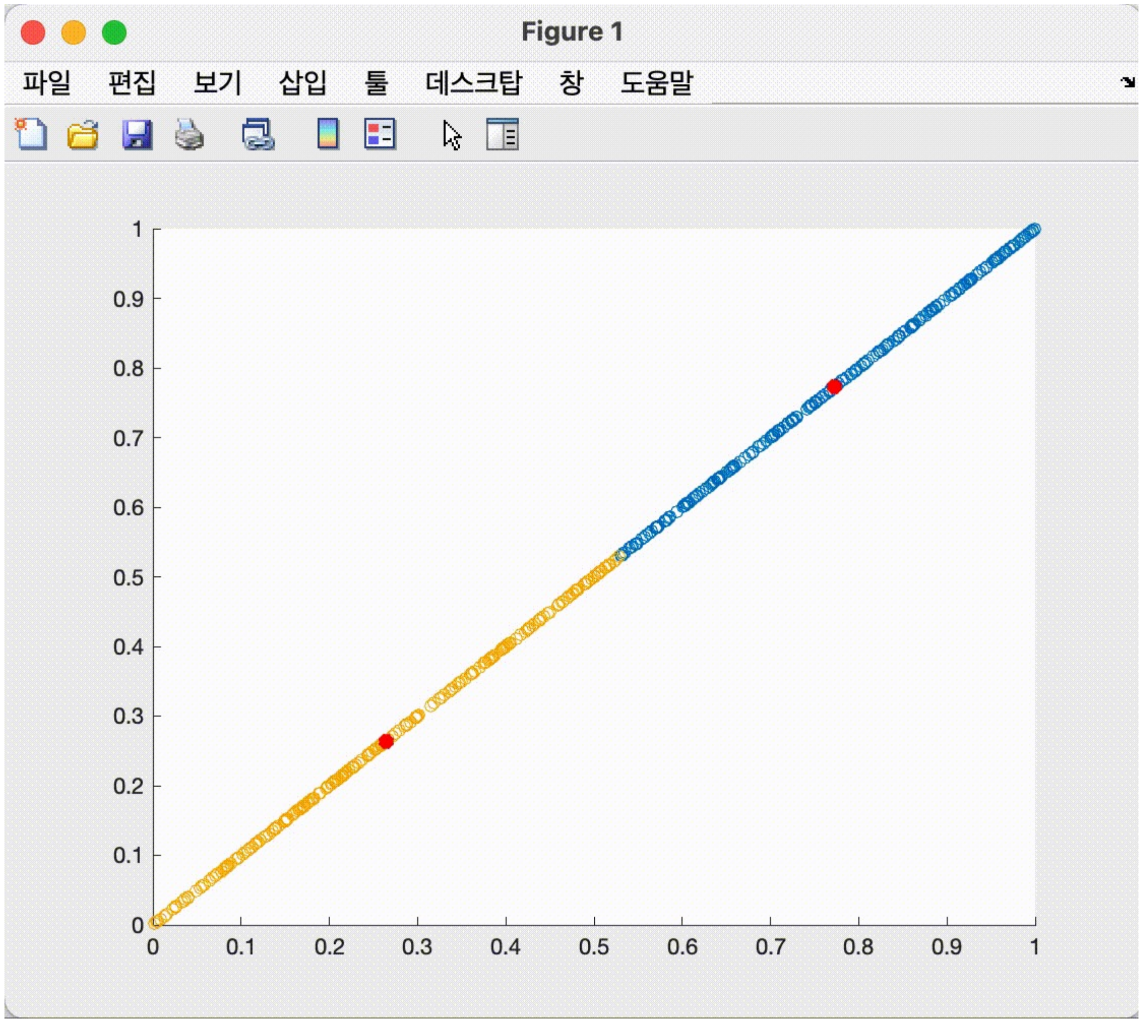
```
D = rand(500, 1); % 500 samples with 1 feature

% Number of Clusters
k = 2;
```

[plot code]

```
% plot
% Checking cluster results in real time
cla;
hold on;
for i = 1: k
    cluster = C{i};
    try
        scatter(cluster(:,1),cluster(:,1)) % cluster data
        scatter(u(:,1),u(:,1),'*r','LineWidth',5) % centroid
    catch
        fprintf("Plot error is occured\n")
    end
end
pause(0.5)
```

Working for 1D data

[result]



Working for 2D data

[setting]

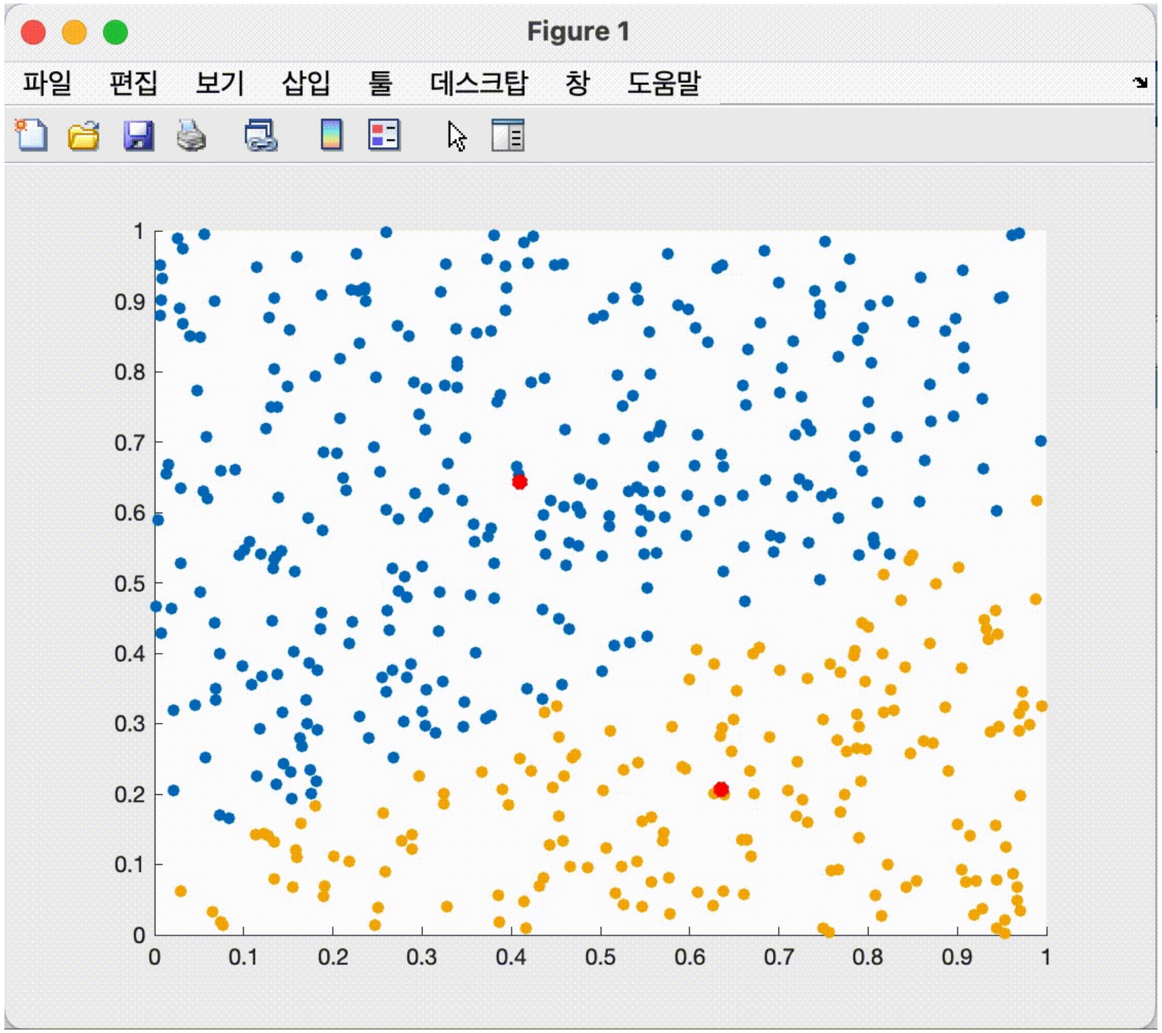
```
D = rand(500, 2); % 500 samples with 2 features

% The number of clusters
k = 2;
```

[plot code]

```
% plot
% Checking cluster results in real time
cla;
hold on;
for i = 1: k
    cluster = C{i};
    try
        scatter(cluster(:,1),cluster(:,2),'filled') % cluster data
        scatter(u(:,1),u(:,2),'*r','LineWidth',5) % centroid
    catch
        fprintf("Plot error is occured\n")
    end
end
pause(0.5)
```

Working for 2D data

[result]



CHPATER 05

Working for 3D data

[setting]

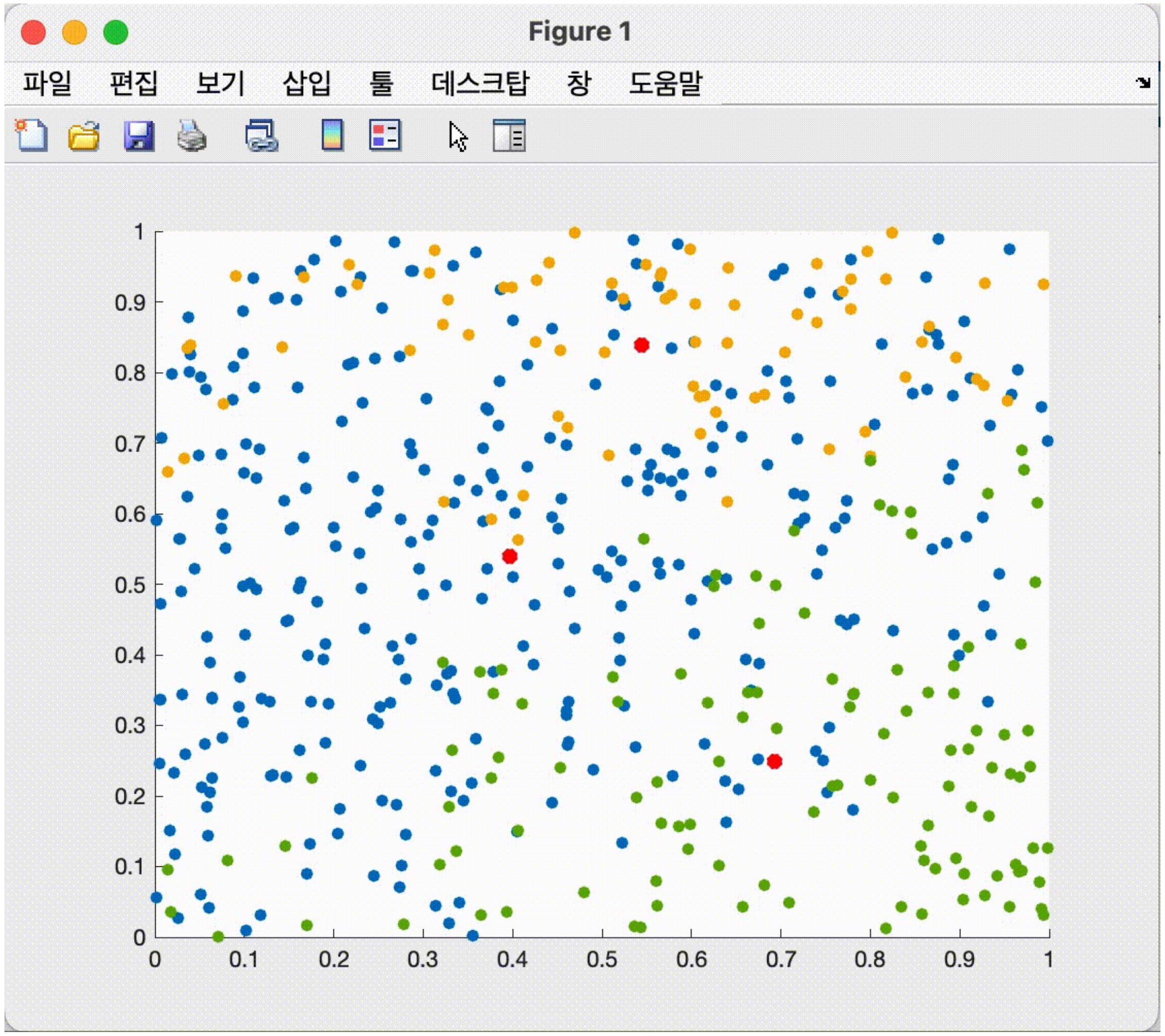
```
D = rand(500, 3); % 500 samples with 3 features

% Number of Clusters
k = 3;
```

[plot code]

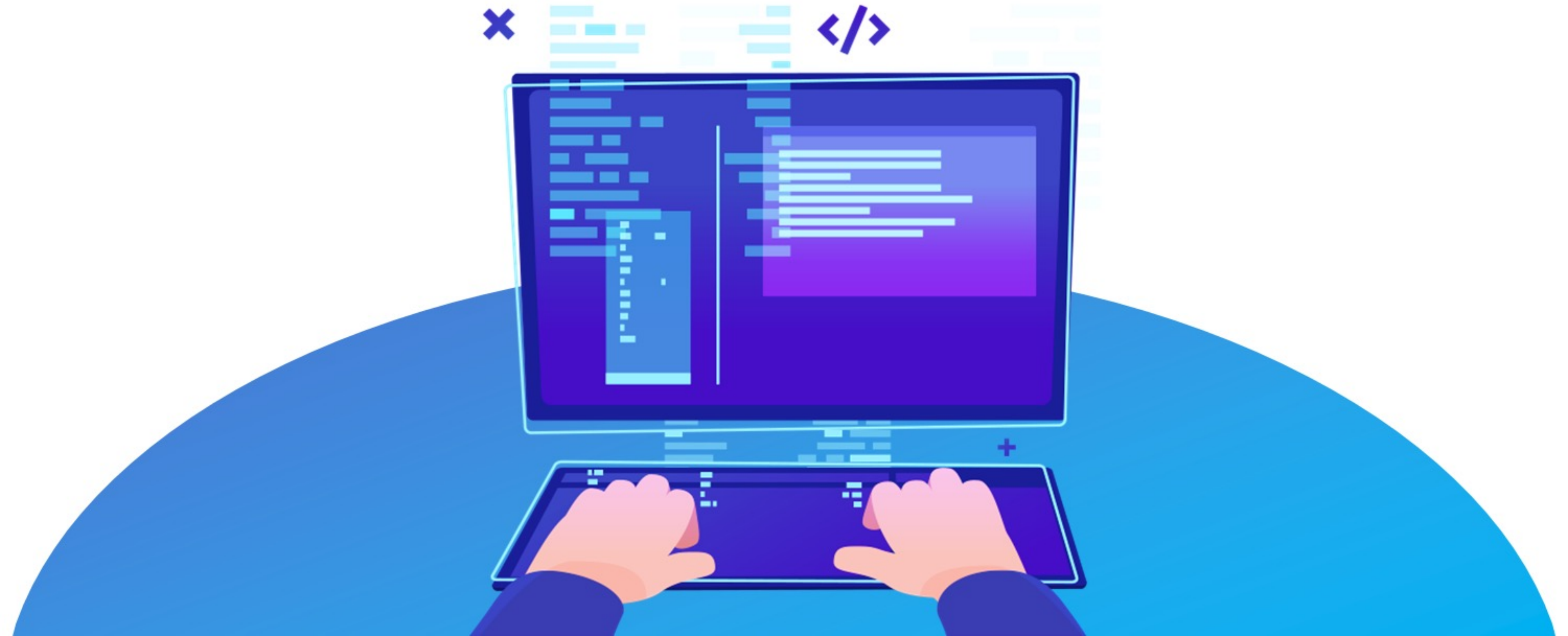
```
%plot
% Checking cluster results in real time
cla;
hold on;
for i = 1: k
    cluster = C{i};
    try
        scatter3(cluster(:,1),cluster(:,2),cluster(:,3),'filled') % cluster data
        scatter3(u(:,1),u(:,2),u(:,3),'*r','LineWidth',5) % centroid
    catch
        fprintf("Plot error is occured\n")
    end
end
pause(0.5)
```

Working for 3D data

[result]



QnA



Thank you