K-means Clustering

The k-means clustering algorithm represents each cluster by its corresponding cluster centroid. The algorithm would partition the input data into *k* disjoint clusters by iteratively applying the following two steps:

- Form *k* clusters by assigning each instance to its nearest centroid.
- Recompute the centroid of each cluster.

Here, we perform k-means clustering on a toy example of movie ratings dataset. Consider the following dataset

	user	Jaws	Star Wars	Exorcist	Omen
0	john	5	5	2	1
1	mary	4	5	3	2
2	bob	4	4	4	3
3	lisa	2	2	4	5
4	lee	1	2	3	4
5	harry	2	1	5	5

In this example dataset, the first 3 users liked action movies (Jaws and Star Wars) while the last 3 users enjoyed horror movies (Exorcist and Omen). Our goal is to apply k-means clustering on the users to identify groups of users with similar movie preferences. (Note that each data has 4 attributes)

1- Assume k=2 and apply K-means algorithm to group the users to 2 clusters (use MATLAB or any other programming language). Determine a cluster ID (0 or 1) for each user and fill in the following table.

	Cluster ID
user	
john	
mary	
bob	
lisa	
lee	
harry	

2- Determine the centroids of each cluster.

The cluster centroids can be applied to other users to determine their cluster assignments. Consider the following dataset and complete the table

	user	Jaws	Star Wars	Exorcist	Omen	Cluster ID
0	paul	4	5	1	2	
1	kim	3	2	4	4	
2	liz	2	3	4	1	
3	tom	3	2	3	3	
4	bill	5	4	1	4	

3- To determine the number of clusters in the data, apply k-means with varying number of clusters (k) from 1 to 6 and compute their corresponding sum-of-squared errors (SSE) and plot the SSE versus k. How many clusters do you think is enough?