IIT ROORKEE

FREE ONLINE EDUCATION swayam

NPTEL ONLINE CERTIFICATION COURSE

K- Means Clustering

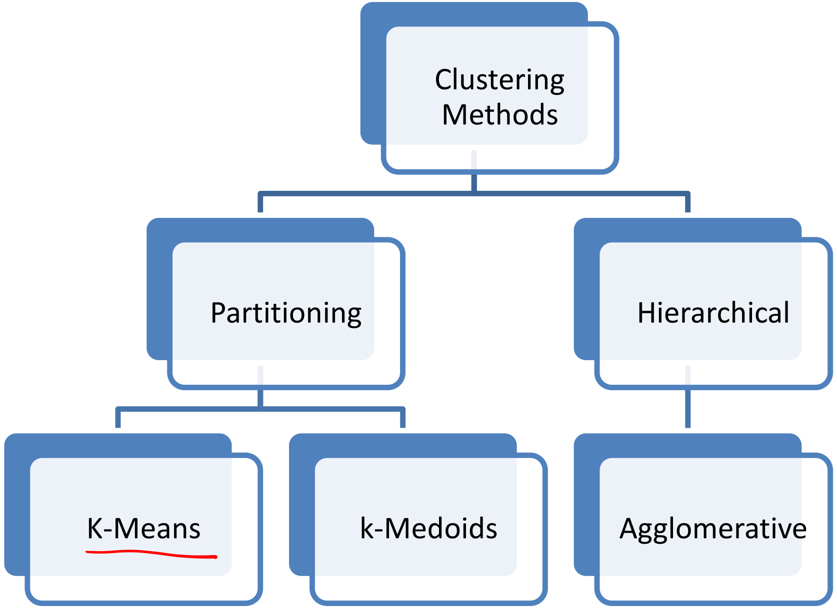
Dr. A. Ramesh

DEPARTMENT OF MANAGEMENT STUDIES

Agenda

- Classification of clustering methods
- Partitioning method: K – means clustering

Classification of Clustering Methods



Which Clustering Algorithm to Choose

- The choice of a clustering algorithm depends on
  - Type of data available
  - Particular purpose
- It is permissible to try several algorithms on the same data, because cluster analysis is mostly used as a descriptive or exploratory tool

Partitioning Method

Given -

- a data set of n objects
- k, the number of clusters
- A partitioning algorithm organizes the objects into k partitions ($k \leq n$), where each partition represents a cluster.
- The clusters are formed to optimize an objective partitioning criterion
- Objective partitioning criterion such as a dissimilarity function based on distance
- Therefore, the objects within a cluster are “similar,” whereas the objects of different clusters are “dissimilar” in terms of the data set attributes.

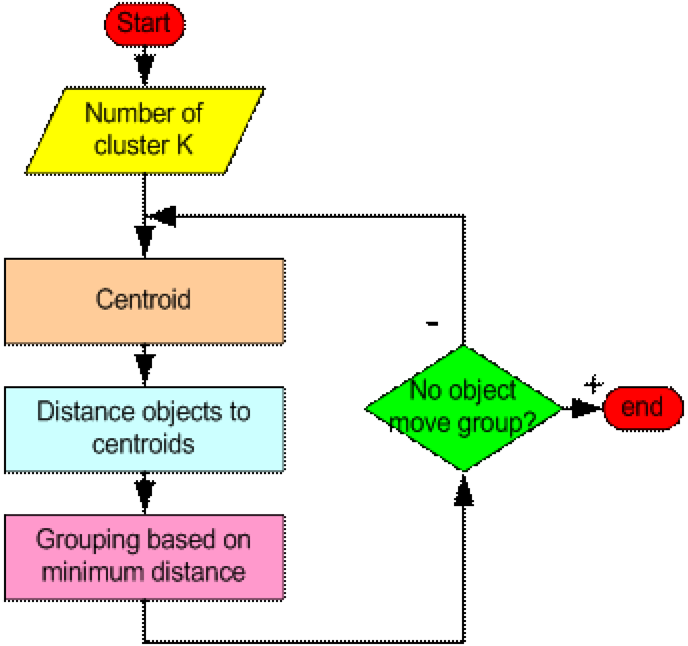
Partitioning Method

- Partitioning methods are applied if one wants to classify the objects into k clusters, where k is fixed.

K-Means Method

- It is a centroid based technique
- The *k*-means algorithm takes the input parameter, *k*, and partitions a set of *n* objects into *k* clusters
- So that the resulting intra-cluster similarity is high but the inter-cluster similarity is low
- Cluster similarity is measured in regard to the *mean* value of the objects in a cluster, which can be viewed as the cluster's *centroid* or *center of gravity*

Working Principle of K-Means Algorithm



Working Principle of K-Means Algorithm

- First it randomly selects *k* of the objects, each of which initially represents a cluster mean or center
- For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean
- It then computes the new mean for each cluster
- This process iterates until the criterion function converges

Working Principle of K-Means Algorithm

- Criterion function

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

where

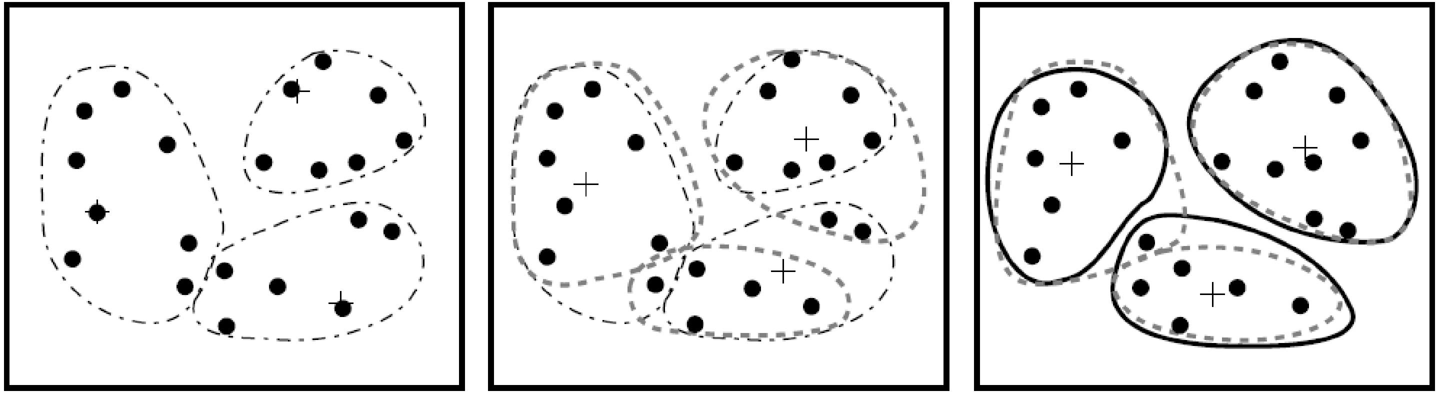
- E is the sum of the square error for all objects in the data set;

-p is the point in space representing a given object;

-m_i is the mean of cluster *Ci* (both p and m_i are multidimensional).

- For each object in each cluster, the distance from the object to its cluster center is squared, and the distances are summed.
- This criterion tries to make the resulting *k* clusters as compact and as separate as possible.

K = 3



(a) (b) (c)

K-Means Clustering Algorithm

Algorithm: k-means. The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

- Input:

  k: the number of clusters,

  D: a data set containing n objects.

- Output: A set of k clusters.

K-Means Clustering Method

- Method:

(1) arbitrarily choose k objects from D as the initial cluster centers;

(2) repeat

(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;

(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;

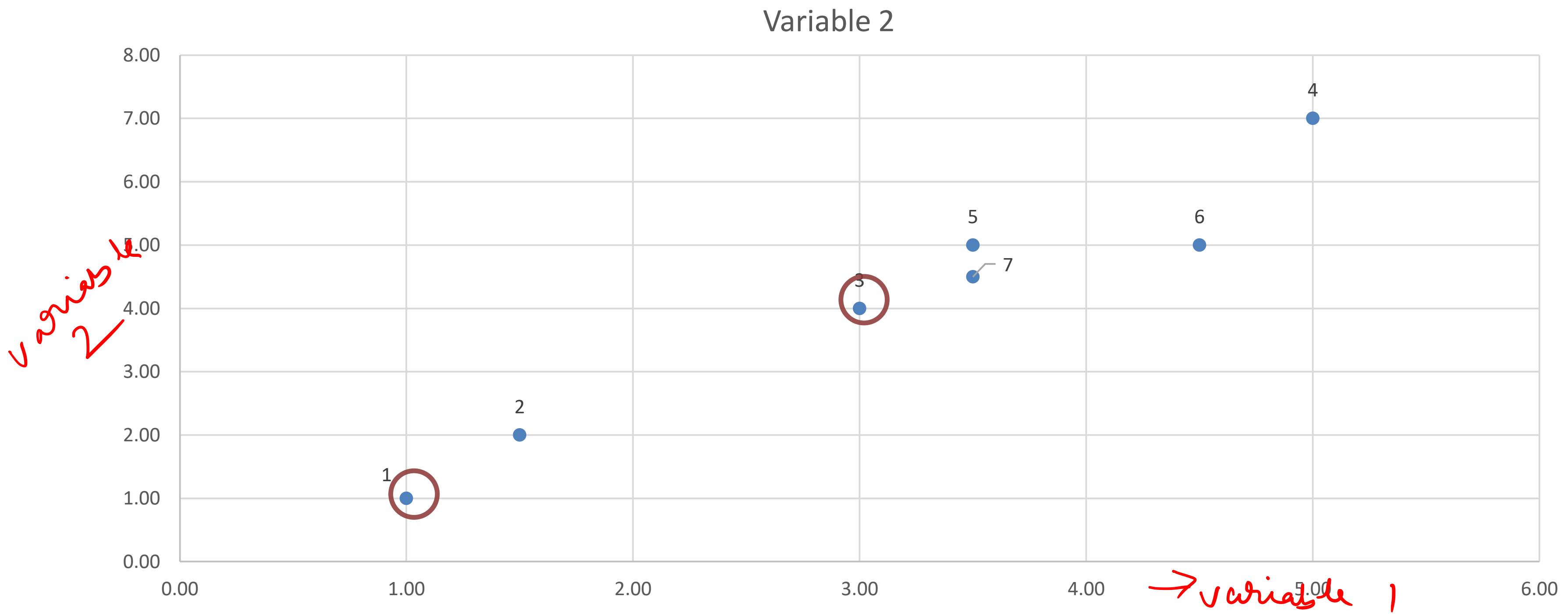
(5) until no change;

K-Means clustering example

$K = 2$

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

K-Means clustering example



K-Means clustering example

- Initialization: Randomly we choose following two centroids (k=2) for two clusters. In this case the 2 centroid are:

Cluster	Var1	Var2
K1	1.0	1.0
K2	3.0	4.0

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

- Calculate Euclidean distance using the given equation

$$\text{Distance}\ [(x_1, y_1), (x_2, y_2)] = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K-Means clustering example

Distance of k1 from k1 (1.0, 1.0) = $\sqrt{(1.0-1.0)^2+(1.0-1.0)^2}$ = 0

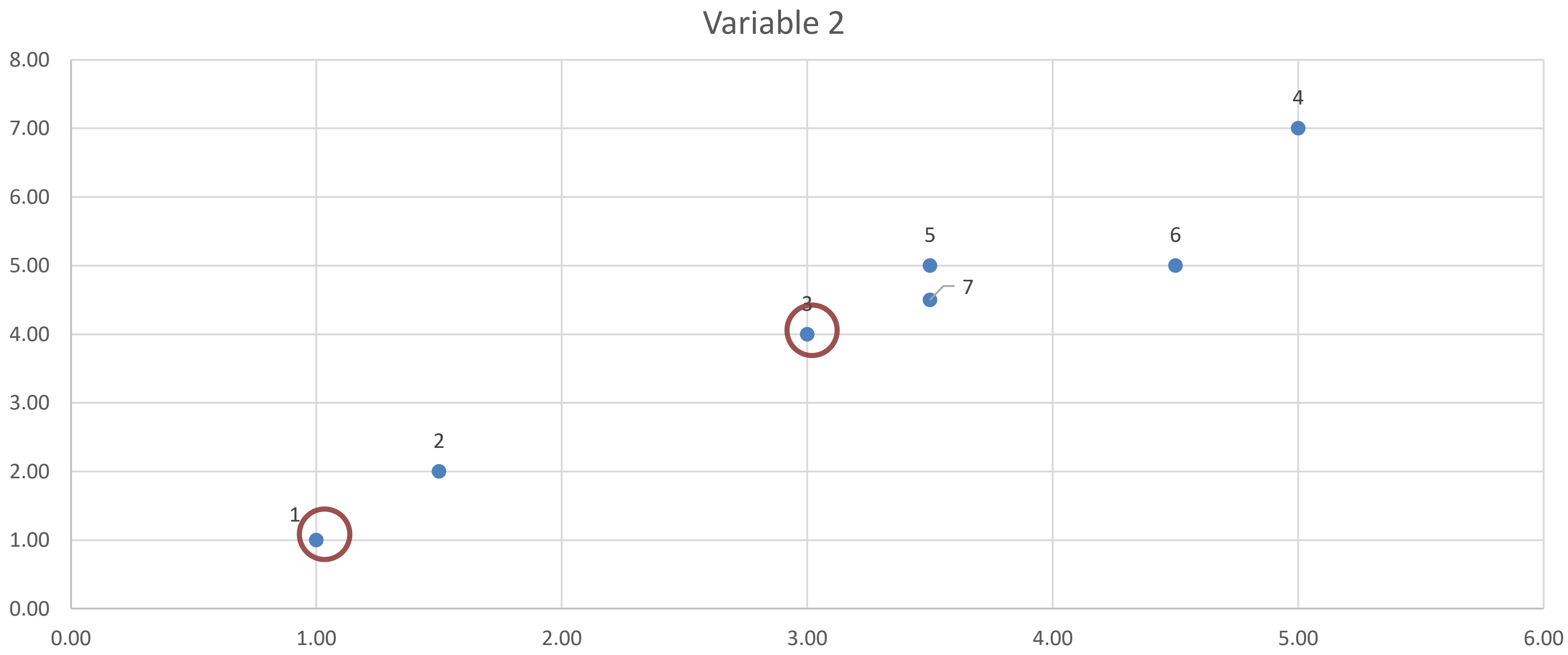
k1 to k2 (1.0, 1.0), (3.0, 4.0) = $\sqrt{(3.0-1.0)^2+(4.0-1.0)^2}$ = 3.61

Distance of k 2 from k2 (3.0, 4.0) = $\sqrt{(3.0-3.0)^2+(4.0-4.0)^2}$ = 0

Cluster	Centroid		
	K1	K2	Assignment
K1	0	3.61	k1
K2	3.61	0	k2

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

At K = 2



Variable 2
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
1
2
3
4
5
6
7

K-Means clustering example

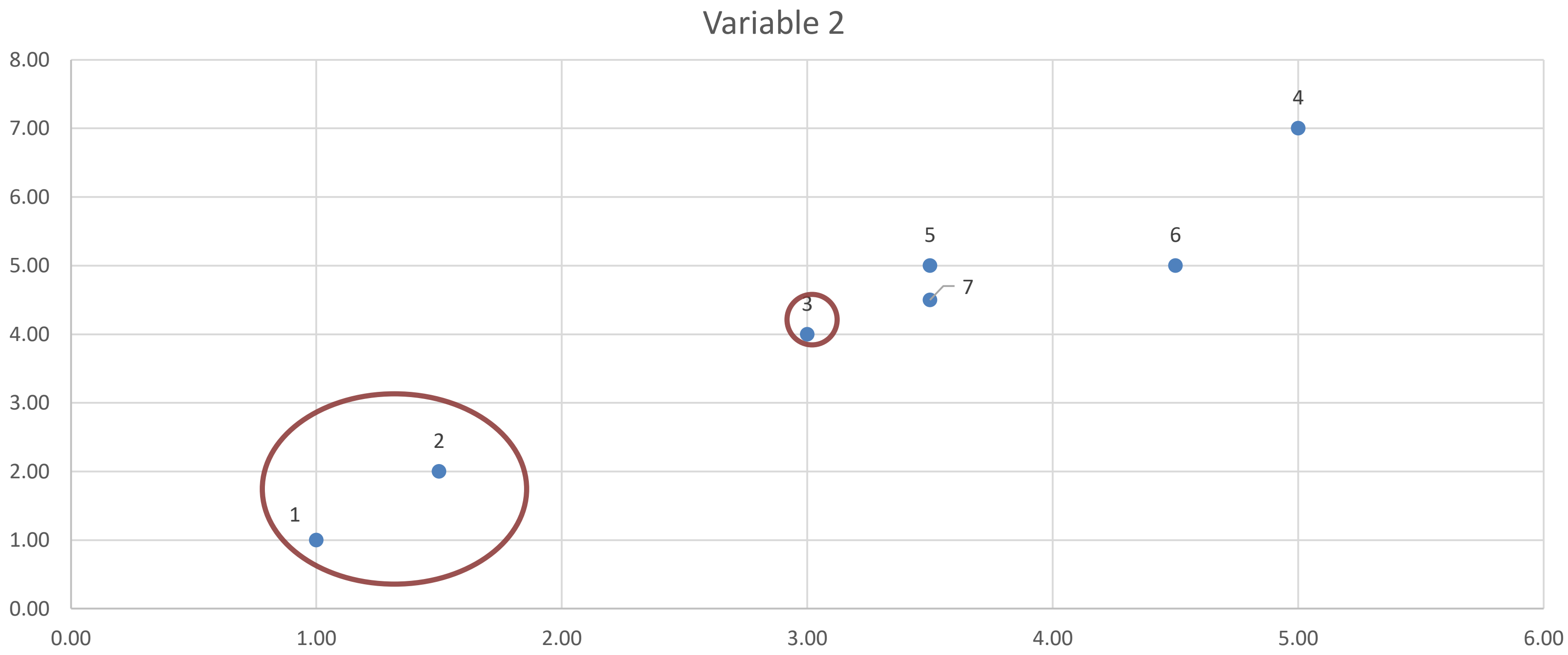
Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

- Calculate Euclidean distance for next dataset (1.5, 2.0)

Distance from cluster1 = $\sqrt{(1.5-1.0)^2+(2.0-1.0)^2}$ = 1.12

Distance from cluster2 = $\sqrt{(1.5-3.0)^2+(2.0-4.0)^2}$ = 2.5

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(1.5, 2.0)	1.12	2.5	k1



Variable 2
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
1
2
3
4
5
6
7

K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	(1.0 + 1.5)/2 = 1.25	(1.0 + 2.0)/2 = 1.5
K2	3.0	4.0

K-Means clustering example

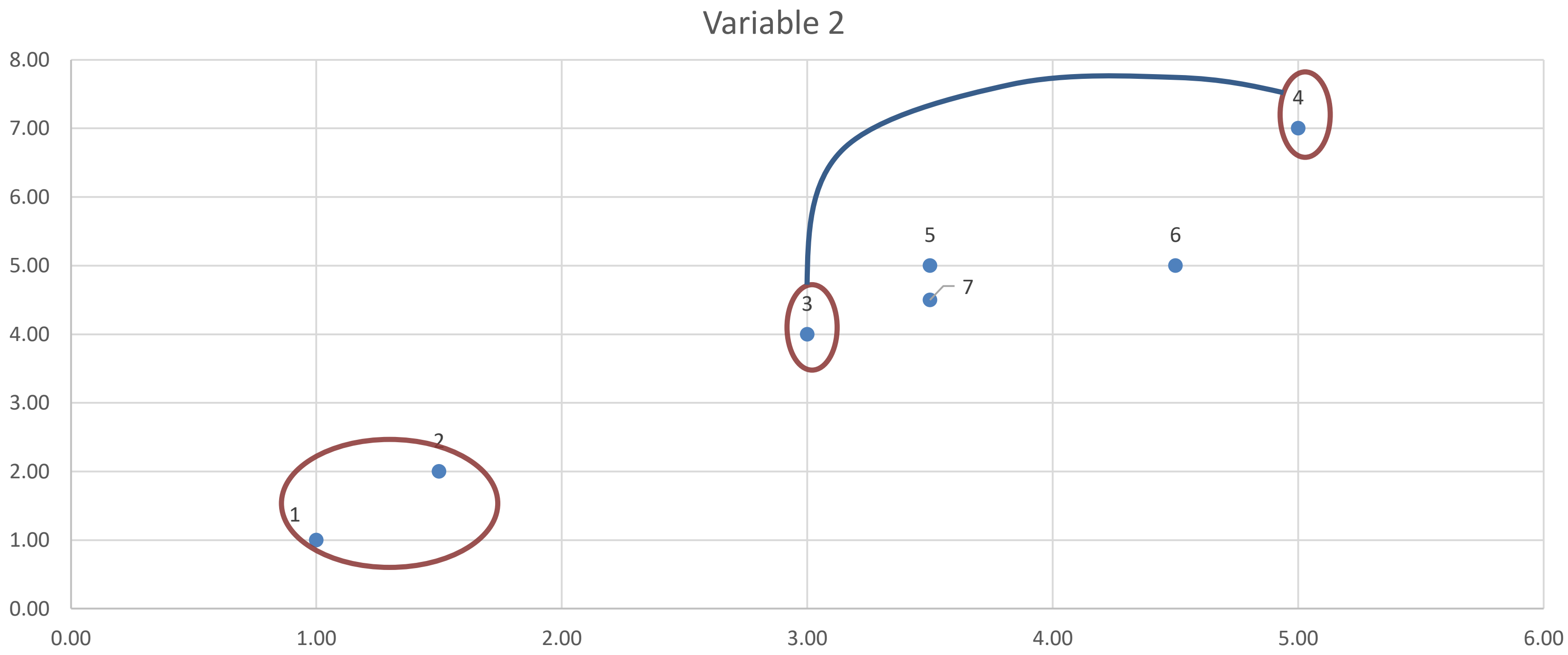
Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

- Calculate Euclidean distance for next dataset (5.0, 7.0)

Distance from cluster1 = $\sqrt{(5.0 - 1.25)^2 + (7.0 - 1.5)^2}$ = 6.66

Distance from cluster2 = $\sqrt{(5.0 - 3.0)^2 + (7.0 - 4.0)^2}$ = 3.61

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(5.0, 7.0)	6.66	3.61	K-2


Variable 2
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
1
2
3
4
5
6
7

K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	1.25	1.5
K2	(3.0 + 5.0)/2 = 4	(4.0 + 7.0)/2 =5.5

K-Means clustering example

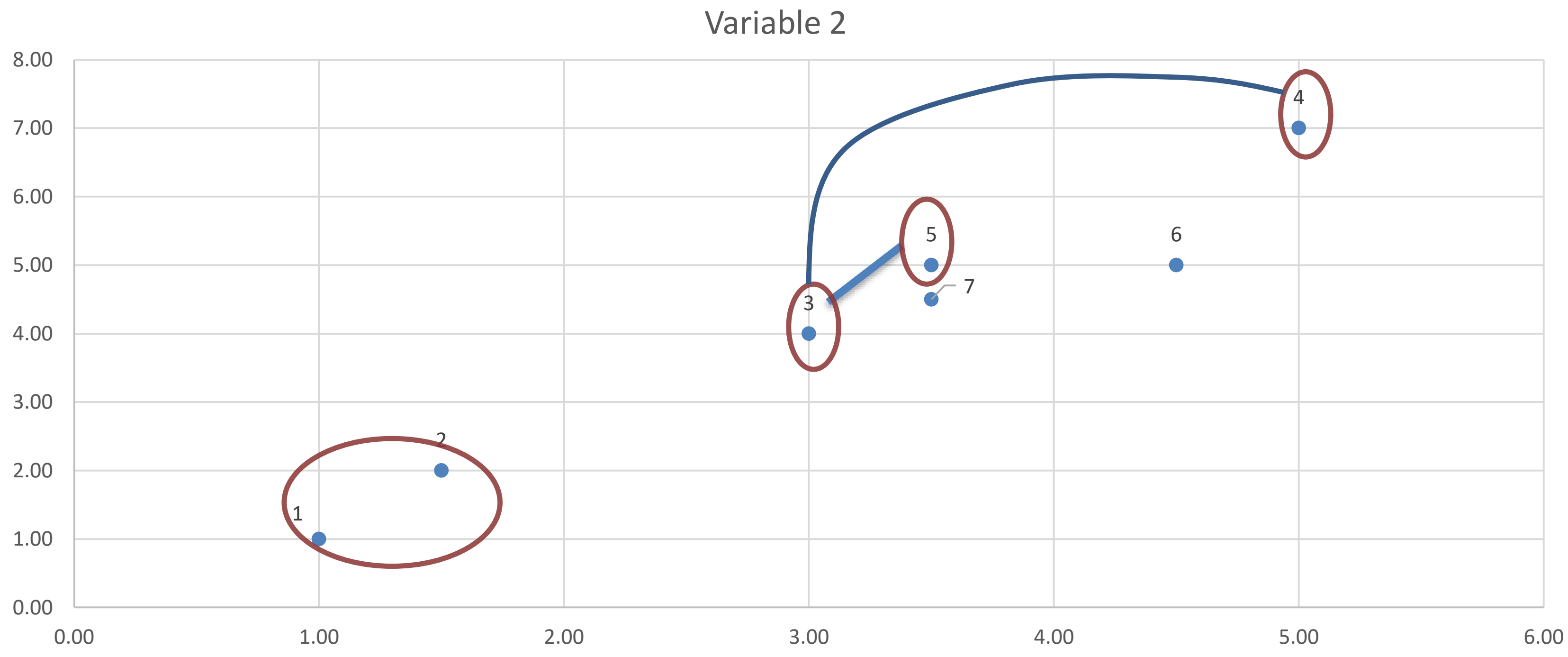
Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

- Calculate Euclidean distance for next dataset (3.5, 5.0)

Distance from cluster1 = $\sqrt{(3.5-1.25)^2+(5.0-1.5)^2}$ = 4.16

Distance from cluster2 = $\sqrt{(3.5-4.0)^2+(5.0-5.5)^2}$ = 0.71

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(3.5, 5.0)	4.16	0.71	K-2



Variable 2
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
1
2
3
4
5
6
7

K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	1.25	1.5
K2	(3.0+5.0+ 3.5)/3 = 3.83	(4.0+7.0 + 5.0)/3 = 5.33

K-Means clustering example

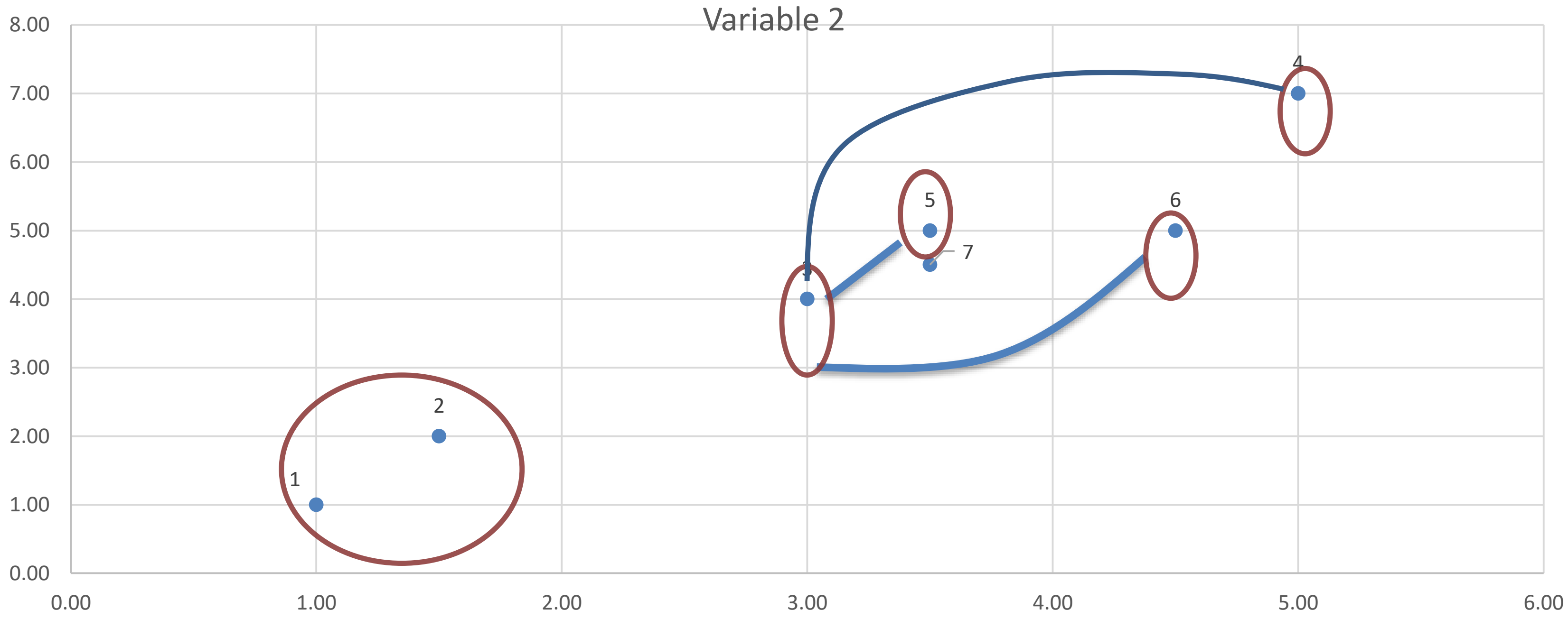
Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

- Calculate Euclidean distance for next dataset (4.5, 5.0)

Distance from cluster1 = $\sqrt{(4.5 - 1.25)^2 + (5.0 - 1.5)^2}$ = 4.78

Distance from cluster2 = $\sqrt{(4.5 - 3.83)^2 + (5.0 - 5.33)^2}$ = 0.75

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(4.5, 5.0)	4.78	0.75	K- 2



Variable 2
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
1
2
4
5
6
7

K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	1.25	1.5
K2	(3.0+5.0+3.5+4.5)/4= 4.00	(4.0+7.0+5.0+5.0)/4= 5.25

K-Means clustering example

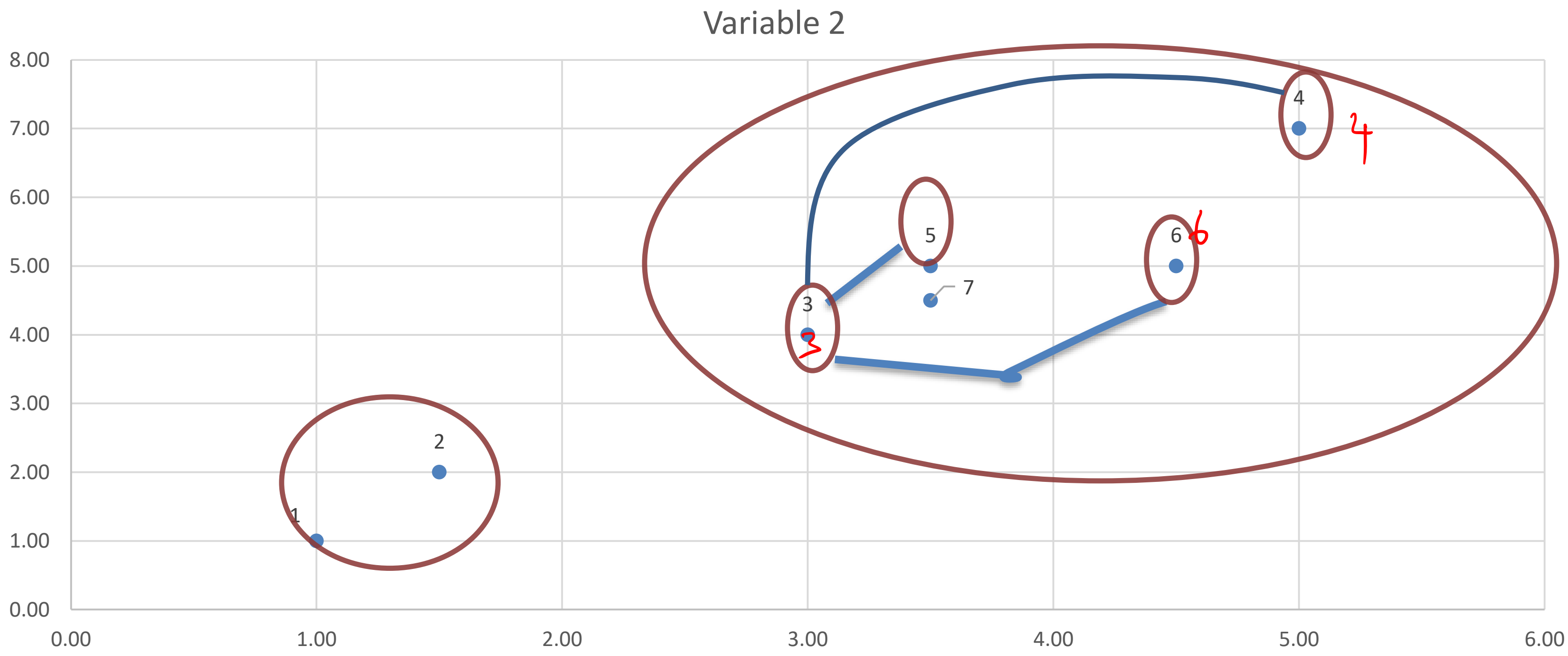
Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

- Calculate Euclidean distance for next dataset (3.5, 4.5)

Distance from cluster1 = $\sqrt{(3.5 - 1.25)^2 + (4.5 - 1.5)^2}$ = 3.75

Distance from cluster2 = $\sqrt{(3.5 - 4.00)^2 + (4.5 - 5.25)^2}$ = 0.86

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(3.5, 4.5)	3.75	0.86	K-2


Variable 2
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
1
2
3
4
5
6
7

K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	1.25	1.5
K2	(3.0+5.0+3.5+4.5+3.5)/5= 3.9	(4.0+7.0+5.0+5.0+4.5)/5= 5.1

K-Means clustering example

Individual	Variable 1	Variable 2	Assignment
1	1.0	1.0	1
2	1.5	2.0	1
3	3.0	4.0	2
4	5.0	7.0	2
5	3.5	5.0	2
6	4.5	5.0	2
7	3.5	4.5	2

Python code for K- Means Clustering

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
data = pd.read_excel('clustering_ex.xlsx')
```

In [3]:
```
data
```

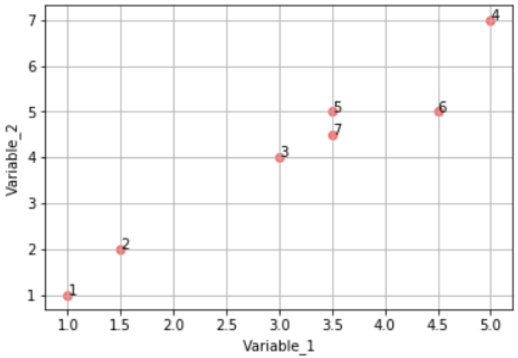
Out[3]:

	Variable_1	Variable_2
0	1.0	1.0
1	1.5	2.0
2	3.0	4.0
3	5.0	7.0
4	3.5	5.0
5	4.5	5.0
6	3.5	4.5

Python code for K- Means Clustering

In [4]:
```
fig = plt.figure(figsize = (5, 5))
x = data['Variable_1']
y = data['Variable_2']
n = range(1,8)
fig, ax = plt.subplots()
ax.scatter(x, y, marker='o', c='red', alpha=0.5)
plt.grid()
plt.xlabel("Variable_1")
plt.ylabel("Variable_2")
for i, txt in enumerate(n):
    ax.annotate(txt, (x[i], y[i]))
```

```
<matplotlib.figure.Figure at 0x20d7a5044a8>
```



Python code

In [5]:
```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=2)
kmeans.fit(data)
```

Out[5]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

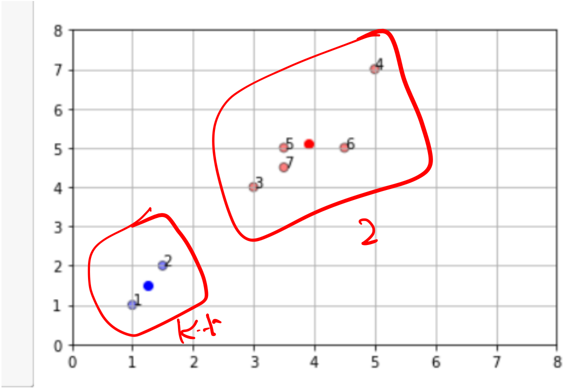
In [6]:
```
labels = kmeans.predict(data)
centroids = kmeans.cluster_centers_
```

In [8]:
```
centroids
```

Out[8]:
```
array([[3.9 , 5.1 ],
       [1.25, 1.5 ]])
```

In [9]:
```
fig = plt.figure(figsize = (5, 5))
colmap = {1:'r', 2:'b'}
colors = map(lambda x: colmap[x+1], labels)
colors1 = list(colors)
fig, ax = plt.subplots()
ax.scatter(x, y, color = colors1, alpha = 0.5, edgecolor = 'k')
for idx, centroid in enumerate(centroids):
    plt.scatter(*centroid, color = colmap[idx+1])

for i, txt in enumerate(n):
    ax.annotate(txt, (x[i], y[i]))
plt.grid()
plt.xlim(0, 8)
plt.ylim(0, 8)
plt.show()
```



Thank you