



K2 Analytics
Building Skills, Building Individuals

Clustering Techniques

- Rajesh Jakhotia

4-Feb-2016

Earning is in Learning
- Rajesh Jakhotia



K2 Analytics
Building Skills, Building Individuals

Agenda

R Setup

Clustering Overview

Hierarchical Clustering

Non-Hierarchical Clustering

About K2 Analytics

At K2 Analytics, we believe that skill development is very important for the growth of an individual, which in turn leads to the growth of Society & Industry and ultimately the Nation as a whole. For this it is important that access to knowledge and skill development trainings should be made available easily and economically to every individual.

Our Vision: *“To be the preferred partner for training and skill development”*

Our Mission: *“To provide training and skill development training to individuals, make them skilled & industry ready and create a pool of skilled resources readily available for the industry”*

*We have chosen Business Intelligence and Analytics as our focus area. With this endeavour we make this presentation on “**Clustering Techniques**” accessible to all those who wish to learn Analytics. We hope it is of help to you. For any feedback / suggestion or if you are looking for job in analytics then feel free to write back to us at ar.jakhotia@k2analytics.co.in*

Welcome to Analytics!!!

R Setup for Clustering

- R 3.2.3 or higher version should be installed
- Following Libraries are installed. Check by running the below command

it is okay if you get Warning Message, but you should not get Error Message

```
library(NbClust)
```

```
library(fpc)
```

```
library(cluster)
```



K2 Analytics
Building Skills, Building Individuals

Clustering Overview

Intro: Data – Information – Knowledge – Wisdom (DIKW)

Where is the Life... we have lost in Living?

Where is the wisdom... we have lost in Knowledge?

Where is the Knowledge...we have lost in Information?

- T. S. Eliot

- Where is the Information... we have lost in Data?


In order to go from

Data to Information to Knowledge and to Wisdom

We need to simplify Data

How do we simplify data

- Simplify data equals to eliminating data complexity
 - too many variables to a few variables
 - too many records to a few records
- Dimension Reduction
 - Reduce number of variable by using techniques like Principal Component Analysis, Collinearity Check, Business Rule Based, etc
- **Clustering / Segmentation**
 - Case reduction
 - Reduce the number of records by identifying similar groups and representing them as a cluster



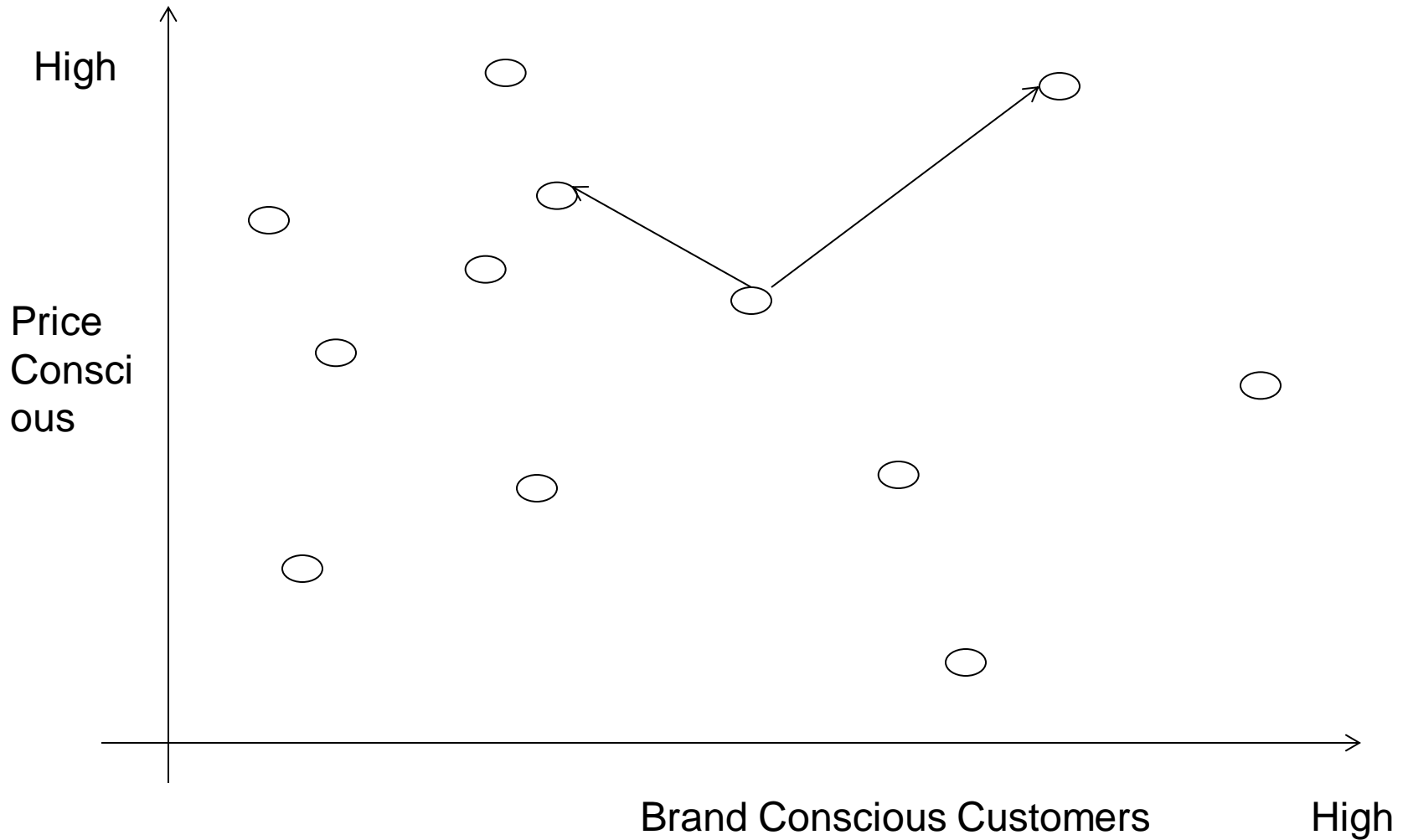
Our focus area
in this
presentation

Clustering

- Clustering is a technique for finding similar groups in data, called clusters.
- It groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- A *cluster* is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters
- How do we define “Similar” in clustering?

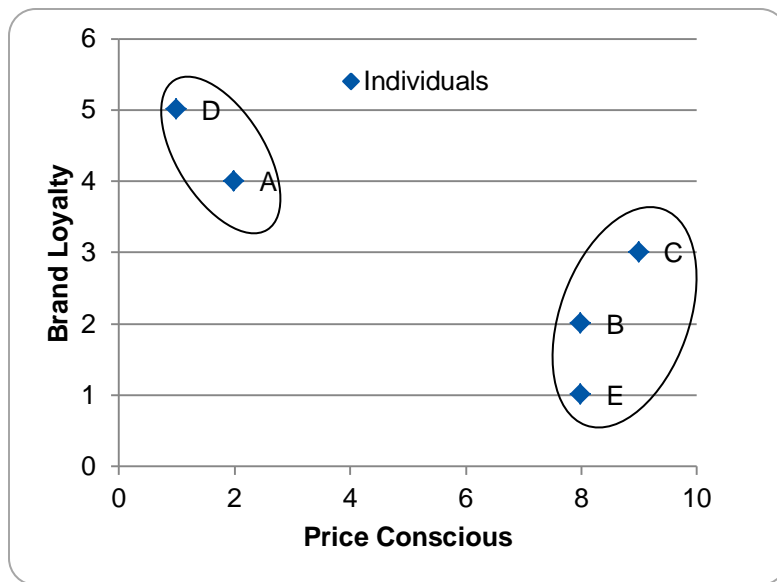


Note: Clustering is an unsupervised learning technique



Simple Clustering e.g.

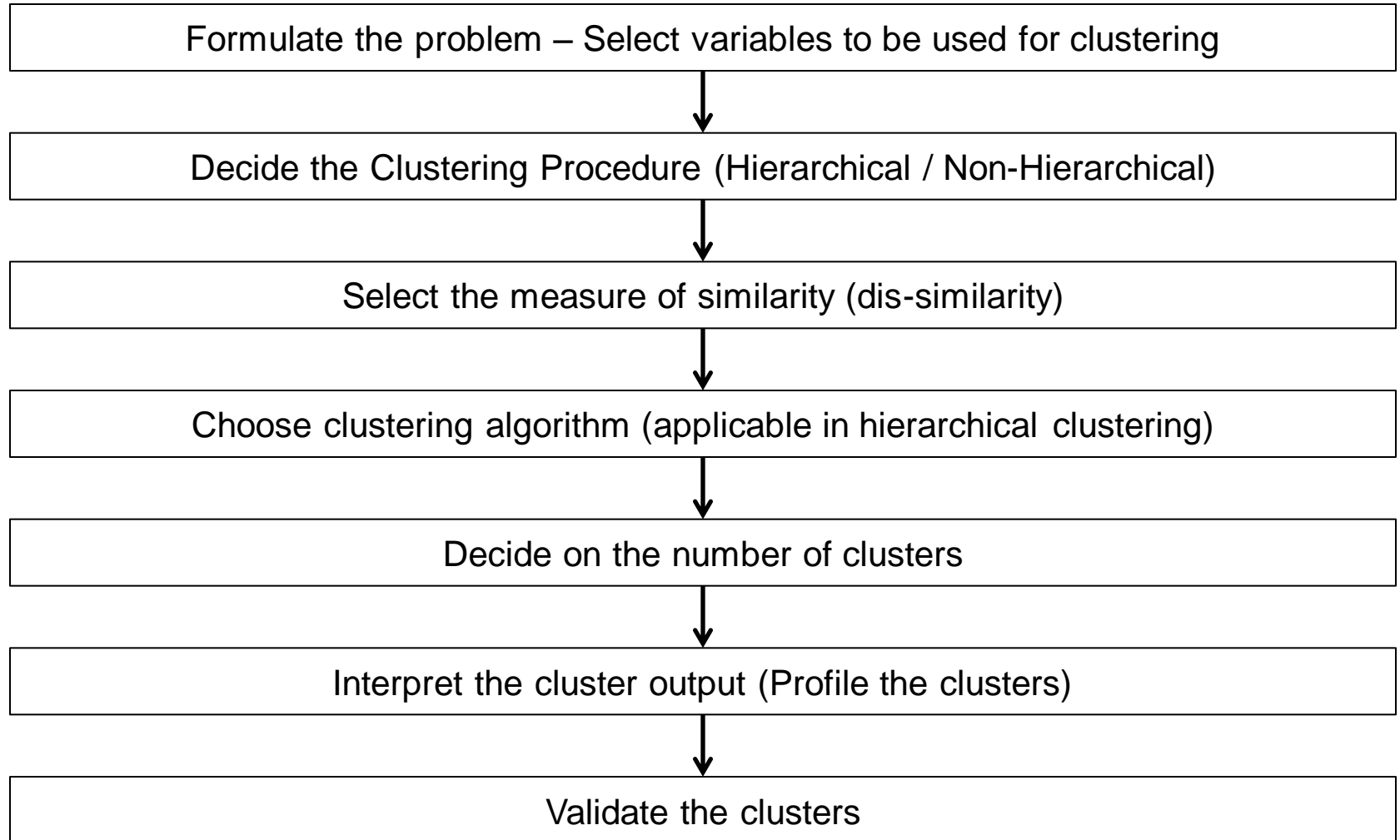
Individuals	Price Conscious	Brand Loyalty
A	2	4
B	8	2
C	9	3
D	1	5
E	8	1



- From Scatter Plot we can see that
 - A & D form one segment of customers who are very high on Brand Loyalty
 - B, C, & E is another segment which is very Price Conscious

Note: The above e.g. is just an hypothetical data to introduce the subject of clustering. It is not related to the below url:
http://globalbizresearch.org/chennai_conference/pdf/pdf/ID_C405_Formatted.pdf

Steps involved in Clustering Analysis



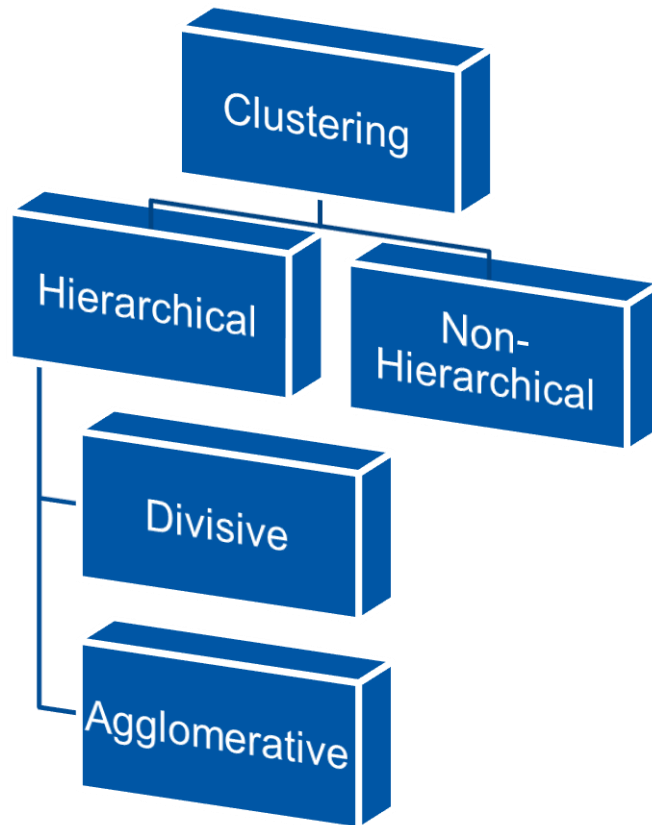
Formulate the clustering problem

- Formulate the problem
 - understand the business problem
 - hypothesize variables that will help solve the clustering problem at hand
- Applications of Clustering Technique
 - Store Clustering
 - Customer Clustering
 - Village Affluency Categorization



Note: It is preferable to do Factor Analysis / Principal Component Analysis before clustering

Decide the clustering procedure



- Types of Clustering Procedures
 - Hierarchical Clustering
 - Non-Hierarchical Clustering
- Hierarchical clustering is characterized by a tree like structure and uses distance as a measure of (dis)similarity
- Non-hierarchical clustering techniques uses partitioning methods and within cluster variance as a measure to form homogeneous groups

Hierarchical Vs. Non-Hierarchical clustering

Hierarchical Clustering

- Relatively very slower
- Agglomerative clustering is most used algorithm
- Uses distance as a measure for (dis)similarity
- Helps suggest optimal numbers of clusters in data.
- Object assigned to a cluster remains in that cluster

Non-Hierarchical Clustering

- Fast and preferable to use with large datasets
- K-means is a very popular non-hierarchical clustering technique
- Uses within cluster variance as a measure of similarity
- Non-hierarchical clustering require number of clusters as an input parameter for starting
- Objects can be reassigned to other clusters during the clustering process



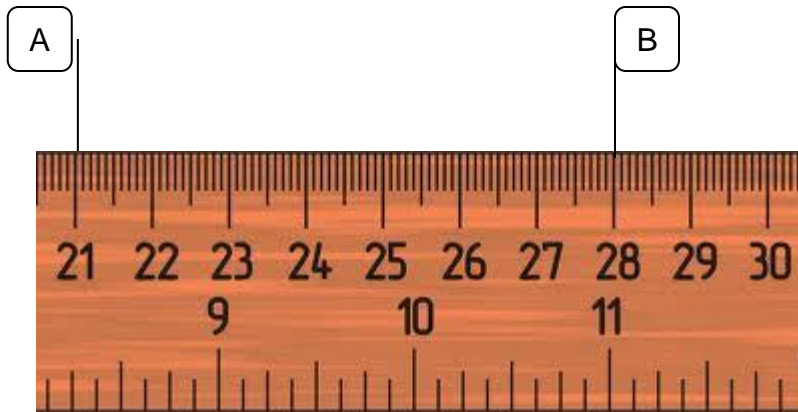
K2 Analytics
Building Skills, Building Individuals

Hierarchical Clustering

Hierarchical Clustering - (dis)similarity measure

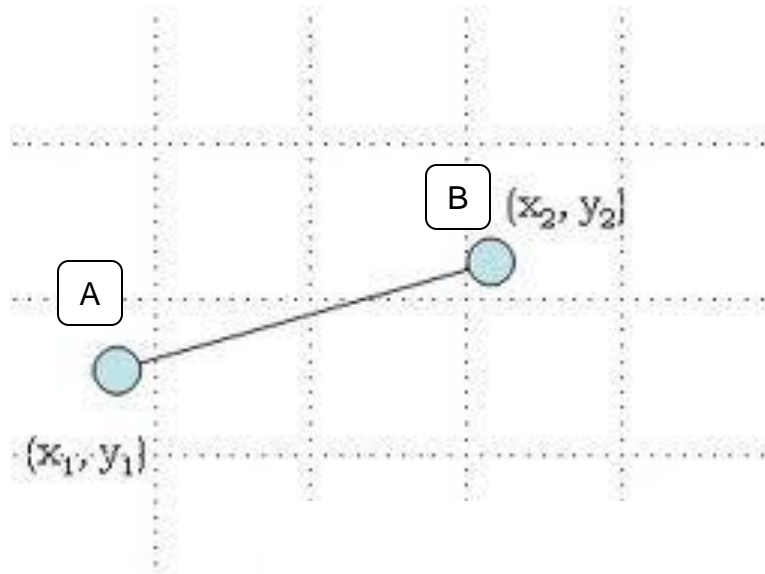
- Hierarchical Clustering is based on dis(similarity) measure
- Most software package calculate a measure of dissimilarity by estimating the distance between pair of objects
- Objects with shorter distance are considered similar, whereas objects with larger distance are dissimilar
- Measures of similarity
 - Euclidean distance (most commonly used)
 - City Block or Manhattan distance
 - Chebyshev distance

Distance Computation



What is the distance between Point A and B?

Ans: 7

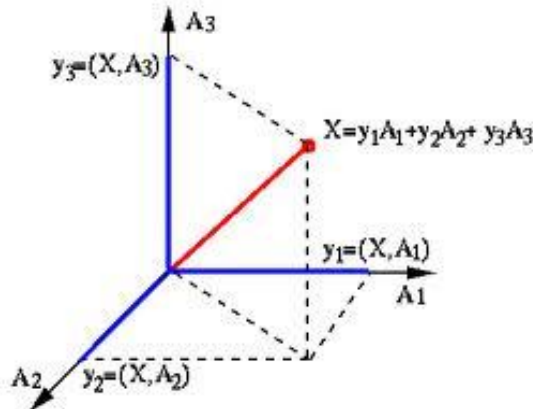


What is the distance between Point A and B?

Ans: $\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$

(Remember the Pythagoras Theorem)

Distance Computation Contd...



- What is the distance between Point A and B in n-Dimension Space?

- If A (a_1, a_2, \dots, a_n) and B (b_1, b_2, \dots, b_n) are cartesian coordinates
- By using Euclidean Distance (which is an extension of Pythagoras Theorem), we get Distance AB as
- $D_{AB} = \sqrt{[(a_1-b_1)^2 + (a_2-b_2)^2 + \dots + (a_n-b_n)^2]}$

Chebyshev Distance

- In mathematics, **Chebyshev distance** is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension
- Assume two vectors: A ($a_1, a_2, \dots a_n$) & B ($b_1, b_2, \dots b_n$)
- Chebyshev Distance
$$= \text{Max} (| a_1 - b_1 | , | a_2 - b_2 | , \dots | a_n - b_n |)$$
- Application: Survey / Research Data where the responses are Ordinal

Manhattan Distance

- Manhattan Distance also called City Block Distance
- Assume two vectors: A (x1, x2, ...xn) & B (y1, y2,...yn)
- Manhattan Distance

$$= |a_1 - b_1| + |a_2 - b_2| + \dots + |a_n - b_n|$$

A

Block

Manhattan Distance = 8 + 4 = 12

Block

Chebyshev Distance = Max (8, 4) = 8

Block

Eucledian Distance = sqrt (8^2 + 4^2) = 8.94

Block

Block

Block

Block

Block

Block

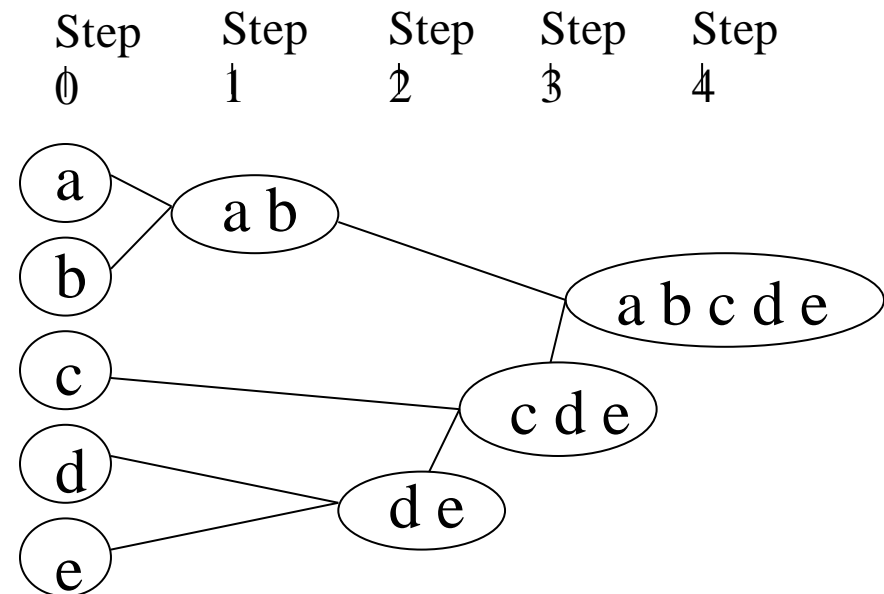
Block

Block

B

Hierarchical Clustering | Agglomerative Clustering

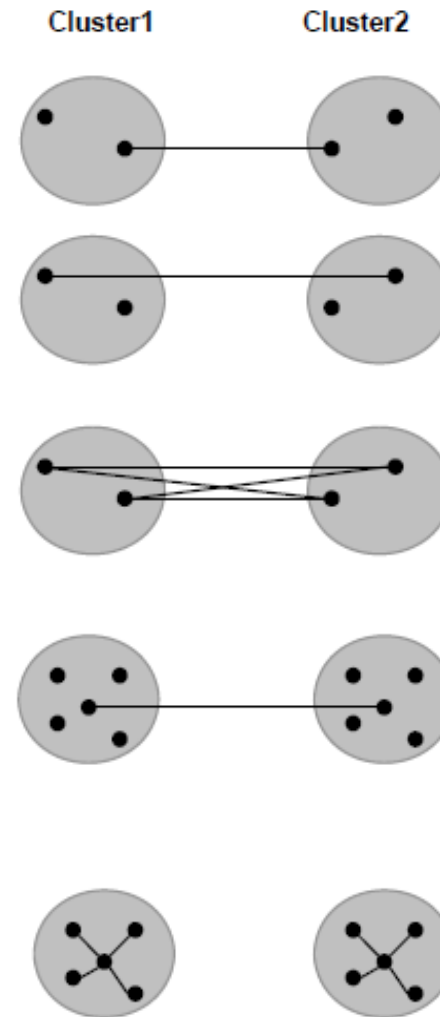
- Starts with each record as a cluster of one record each
- Sequentially merges 2 closest records by distance as a measure of (dis)similarity to form a cluster. This reduces the number of records by 1
- Repeat the above step with new cluster and all remaining clusters till we have one big cluster



How do you measure the distance between cluster (a,b) and (c) or the cluster (a,b) and (d,e) ????

Agglomerative Clustering Procedures

- Single linkage – Minimum distance or Nearest neighbour rule
- Complete linkage – Maximum distance or Farthest distance
- Average linkage – Average of the distances between all pairs
- Centroid method – combine cluster with minimum distance between the centroids of the two clusters
- Ward's method – Combine clusters with which the increase in within cluster variance is to the smallest degree



http://sites.stat.psu.edu/~ajw13/stat505/fa06/19_cluster/09_cluster_wards.html

Clustering e.g. 1 : Clustering for Retail Customers

Let us find the clusters in given Retail Customer Spends data

We will use Hierarchical Clustering technique

Let us first set the working directory path and import the data

```
setwd("D:/K2Analytics/Clustering/")
```

```
RCDF <- read.csv("datafiles/Cust_Spend_Data.csv", header=TRUE)
```

```
View(RCDF)
```

Cust_ID	Name	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
1	A	10000	2	1	1	0
2	B	7000	3	0	10	9
3	C	7000	7	1	3	4
4	D	6500	5	1	1	4
5	E	6000	6	0	12	3
6	F	4000	3	0	1	8
7	G	2500	5	0	11	2
8	H	2500	3	0	1	1
9	I	2000	2	0	2	2
10	J	1000	4	0	1	7

HyperMarket Customer Spend

Data - Metatdata

AVG_Mthly_Spend: The average monthly amount spent by customer

No_of_Visits: The number of times a customer visited the HyperMarket i

Count of **Apparel, Fruits and Vegetable, Staple Items** purchased in a month

Building the hierarchical clusters (without variable scaling)

?dist ## to get help on distance function

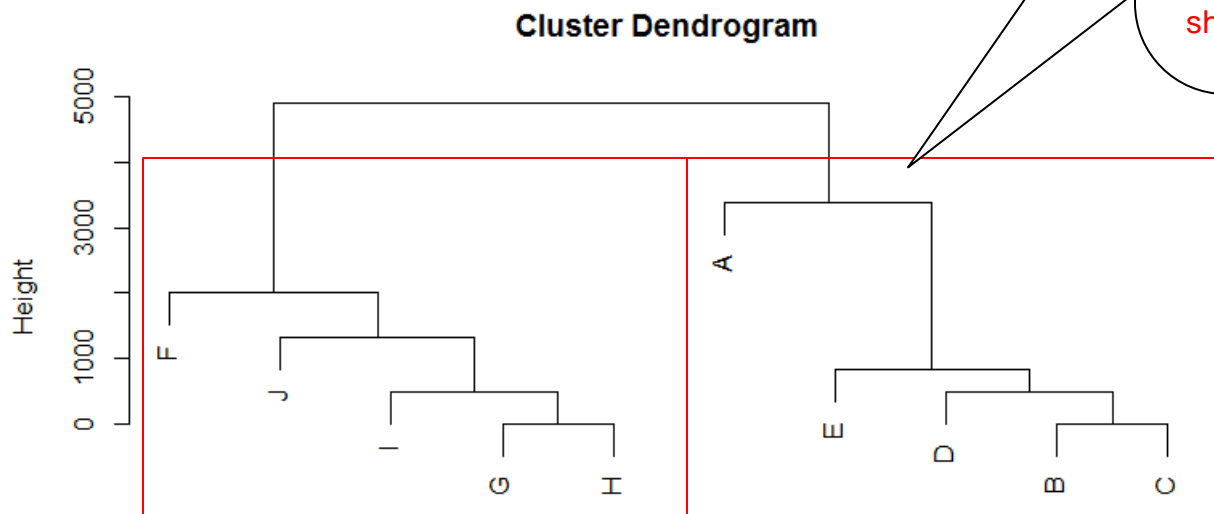
```
d.euc <- dist(x=RCDF[,3:7], method = "euclidean")
```

we will use the hclust function to build the cluster

?hclust ## to get help on hclust function

```
clus1 <- hclust(d.euc, method = "average")
```

```
plot(clus1, labels = as.character(RCDF[,2]))
```



Note: The two clusters formed are primarily on the basis of AVG_MTHLY_SPEND

Euclidean Distance computation in this case is influenced by AVG_MTHLY_SPEND variable as the range of this variable is too large compared to the other variables

To avoid this problem, we should scale the variables used for clustering

Building the hierarchical clusters (with variable scaling)

```
## scale function standardizes the values
```

```
scaled.RCDF <- scale(RCDF[,3:7])
```

```
head(scaled.RCDF, 10)
```

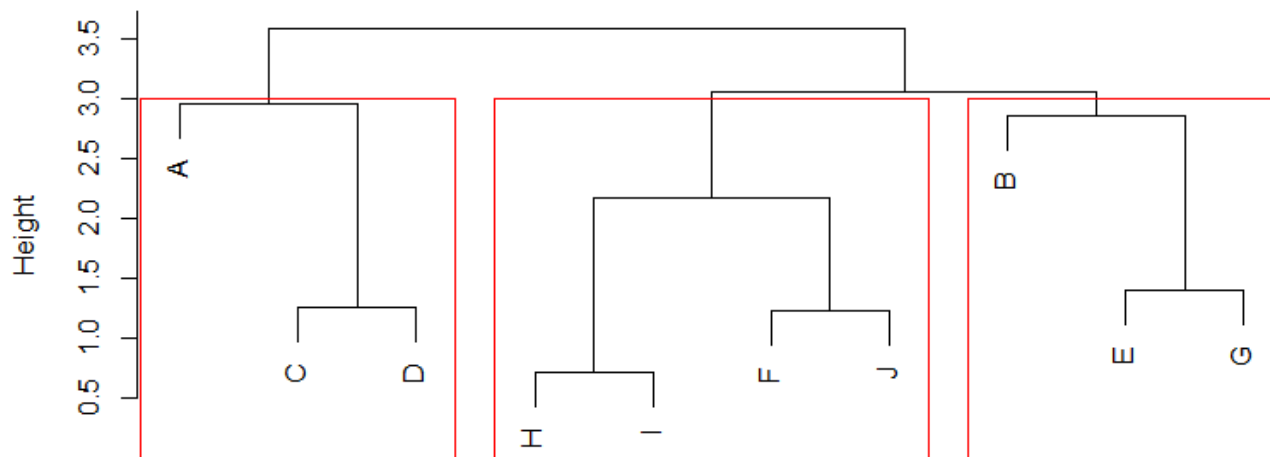
```
d.euc <- dist(x=scaled.RCDF, method = "euclidean")
```

```
clus2 <- hclust(d.euc, method = "average")
```

```
plot(clus2, labels = as.character(RCDF[,2]))
```

```
rect.hclust(clus2, k=3, border="red")
```

Cluster Dendrogram



Understanding the Height Calculation in Clustering

Let us see the distance matrix

d.euc

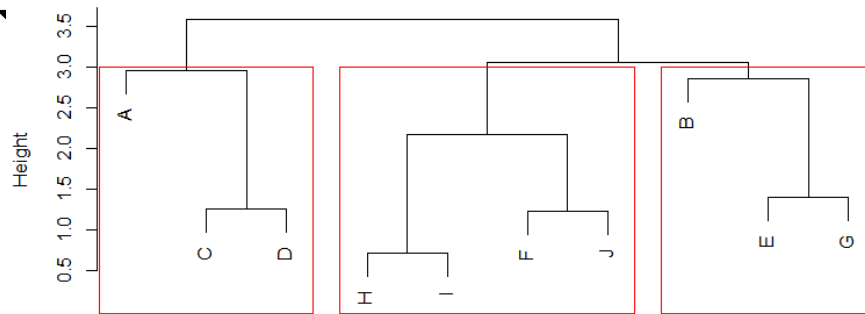
Dist.	A	B	C	D	E	F	G	H	I
B	4.25								
C	3.41	3.84							
D	2.51	3.47	1.26						
E	4.27	2.70	2.92	3.20					
F	3.98	2.21	3.58	2.85	3.43				
G	4.38	3.02	3.38	3.35	1.41	3.17			
H	3.40	3.60	3.66	2.93	3.24	2.35	2.46		
I	3.53	3.39	4.05	3.21	3.48	2.18	2.61	0.73	
J	4.55	2.97	3.59	3.04	3.41	1.24	2.80	2.12	2.06

Let us see the height for clusters

clus2\$height

```
[1] 0.7272685 1.2410400 1.2635399 1.4064486 2.1742679 2.8590372 2.9615235 3.0647590 3.5925234
```

Cluster Dendrogram



Dist.	A	B	C	D	E	F	G	H	I
B	4.25								
C	3.41	3.84							
D	2.51	3.47	1.26						
E	4.27	2.70	2.92	3.20					
F	3.98	2.21	3.58	2.85	3.43				
G	4.38	3.02	3.38	3.35	1.41	3.17			
H	3.40	3.60	3.66	2.93	3.24	2.35	2.46		
I	3.53	3.39	4.05	3.21	3.48	2.18	2.61	0.73	
J	4.55	2.97	3.59	3.04	3.41	1.24	2.80	2.12	2.06

C, D	1.26	H, I	0.73	F, J	1.24	E, G	1.41
A, (C,D)		(H,I), (F,J)				B, (E,G)	
A, C	3.41		F	J		B, E	2.70
A, D	2.51	H	2.35	2.12		B, G	3.02
A, (C,D)	2.96	I	2.18	2.06		B, (E,G)	2.86
		((H,I), (F,J))			2.17		
		(H,I, F,J) , (B, E,G)					
			B	E	G		
		H	3.60	3.24	2.46		
		I	3.39	3.48	2.61		
		F	2.21	3.43	3.17		
		J	2.97	3.41	2.80		
		(H,I, F,J) , (B, E,G)			3.06		
(A, C, D) , (H, I, F, J, B, E, G)							
	H	I	F	J	B	E	G
A	3.40	3.53	3.98	4.55	4.25	4.27	4.38
C	3.66	4.05	3.58	3.59	3.84	2.92	3.38
D	2.93	3.21	2.85	3.04	3.47	3.20	3.35
(A, C, D) , (H, I, F, J, B, E, G)					3.59		

Profiling the clusters

```
## profiling the clusters
```

```
RCDF$Clusters <- cutree(clus2, k=3)
```

```
aggr = aggregate(RCDF[, -c(1,2, 8)], list(RCDF$Clusters), mean )
```

```
clus.profile <- data.frame( Cluster = aggr[,1] ,  
                           Freq = as.vector(table(RCDF$Clusters)) ,  
                           aggr[,-1]  
                           )
```

```
View(clus.profile)
```

Cluster	Freq	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
1	3	7833.333	4.666667	1	1.666667	2.666667
2	3	5166.667	4.666667	0	11.000000	4.666667
3	4	2375.000	3.000000	0	1.250000	4.500000



K2 Analytics
Building Skills, Building Individuals

Non-Hierarchical Clustering (K Means)

K Means Clustering

- K-Means is the most used, non-hierarchical clustering technique
- It is not based on Distance...
- It is based on within cluster Variation, in other words Squared Distance from the Centre of the Cluster
- The algorithm aims at segmenting data such that within cluster variation is reduced

K Means Algorithm

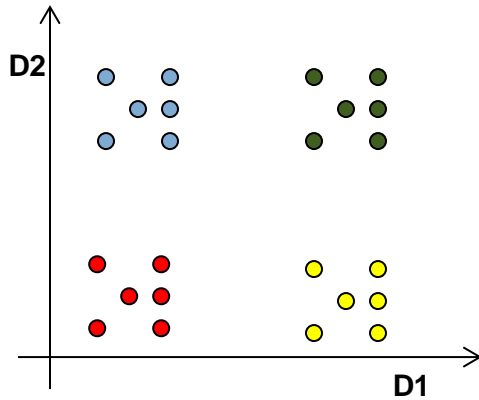
- Input Required : No of Clusters to be formed. (Say K)
- Steps
 1. Assume K Centroids (for K Clusters)
 2. Compute Euclidean distance of each objects with these Centroids.
 3. Assign the objects to clusters with shortest distance
 4. Compute the new centroid (mean) of each cluster based on the objects assigned to each clusters. The K number of means obtained will become the new centroids for each cluster
 5. Repeat step 2 to 4 till there is convergence
 - i.e. there is no movement of objects from one cluster to another
 - Or threshold number of iterations have occurred

K-means advantages

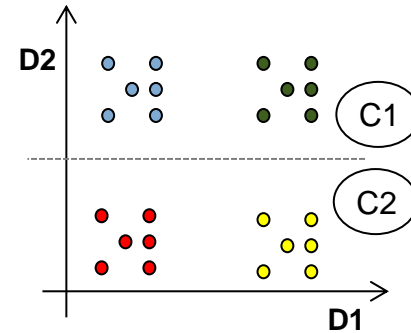
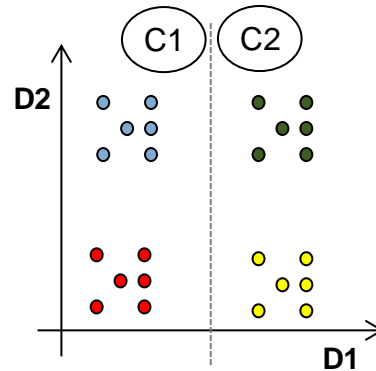
- K-means is superior technique compared to Hierarchical technique as it is less impacted by outliers
- Computationally it is more faster compared to Hierarchical
- Preferable to use on interval or ratio-scaled data as it uses Euclidean distance... desirable to avoid using on ordinal data
- **Challenge – Number of clusters are to be pre-defined and to be provided as input to the process**

Why find optimal No. of Clusters?

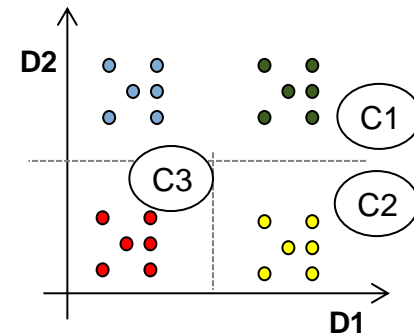
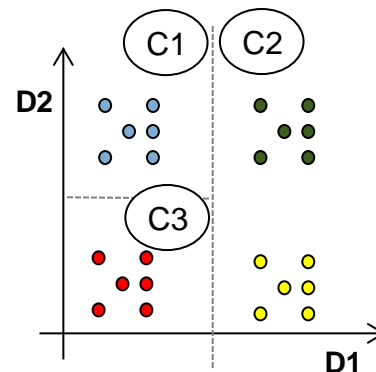
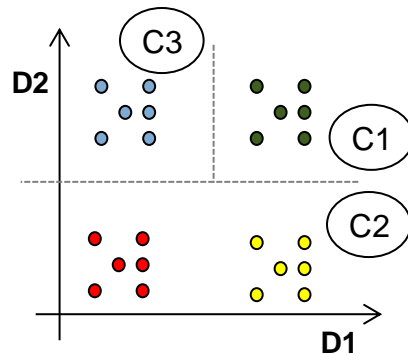
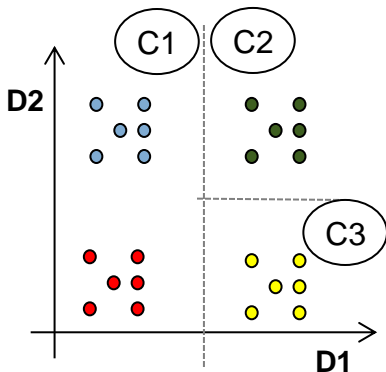
Data to be clustered



■ Two Clusters – 2 possible solution



■ Three Clusters – Multiple possible solution



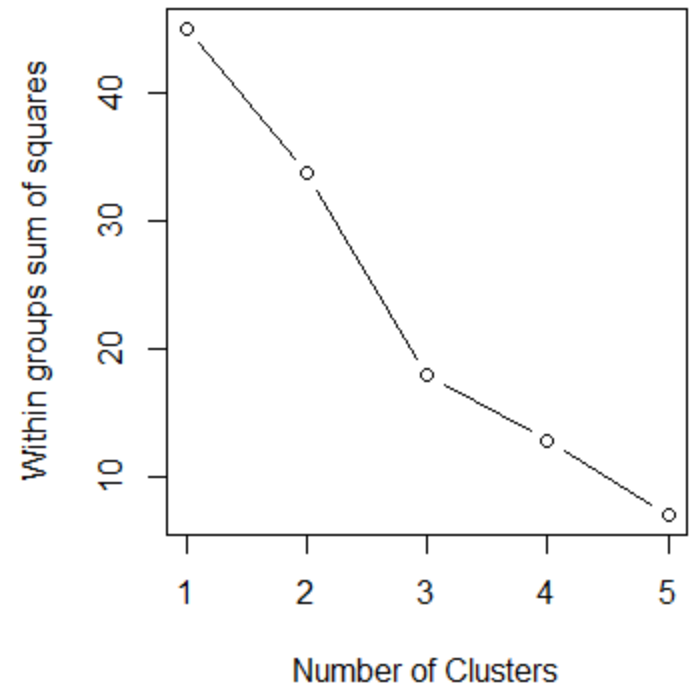
R code to get Optimal No. of Clusters

code taken from the R-statistics blog <http://www.r-statistics.com/2013/08/k-means-clustering-from-r-in-action/>

Identifying the optimal number of clusters form WSS

```
wssplot <- function(data, nc=15, seed=1234) {  
  wss <- (nrow(data)-1)*sum(apply(data,2,var))  
  for (i in 2:nc) {  
    set.seed(seed)  
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}  
  plot(1:nc, wss, type="b", xlab="Number of Clusters",  
       ylab="Within groups sum of squares")}
```

```
wssplot(scaled.RCDF, nc=5)
```



Using NbClust to get optimal No. of Clusters

```
## Identifying the optimal number of clusters
```

```
## install.packages("NbClust")
```

```
library(NbClust)
```

```
set.seed(1234)
```

```
nc <- NbClust(KRCDF[,c(-1,-2)], min.nc=2, max.nc=4, method="kmeans")
```

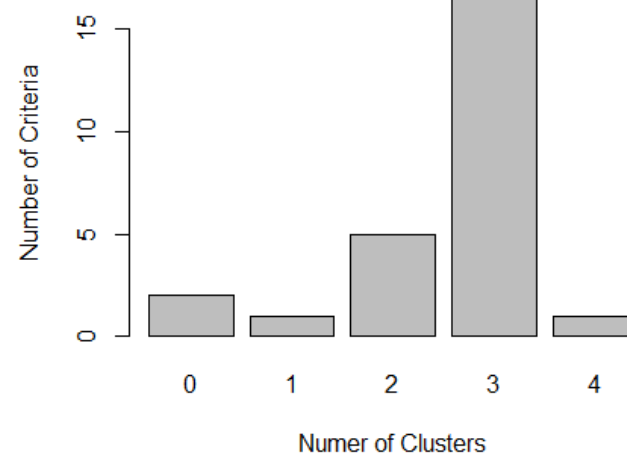
```
table(nc$Best.n[1,])
```

```
barplot(table(nc$Best.n[1,]),
```

```
  xlab="Nuner of Clusters", ylab="Number of Criteria",
```

```
  main="Number of Clusters Chosen by 26 Criteria")
```

Number of Clusters Chosen by 26 Criteria



K Means Clustering R Code

?kmeans

```
kmeans.clus = kmeans(x=scaled.RCDF, centers = 3, nstart = 25)
```

```
## x = data frame to be clustered
```

```
## centers = No. of clusters to be created
```

```
## nstart = No. of random sets to be used for clustering
```

kmeans.clus

```
K-means clustering with 3 clusters of sizes 4, 3, 3
```

```
Cluster means:
```

	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
1	-0.8600931	-0.5883484	-0.621059	-0.6500980	0.1636634
2	1.0367452	0.3922323	1.449138	-0.5612868	-0.4364358
3	0.1100456	0.3922323	-0.621059	1.4280842	0.2182179

```
Clustering vector:
```

```
[1] 2 3 2 2 3 1 3 1 1 1
```

```
within cluster sum of squares by cluster:
```

```
[1] 5.256752 6.514105 6.126217
```

```
(between_SS / total_SS = 60.2 %)
```

```
Available components:
```

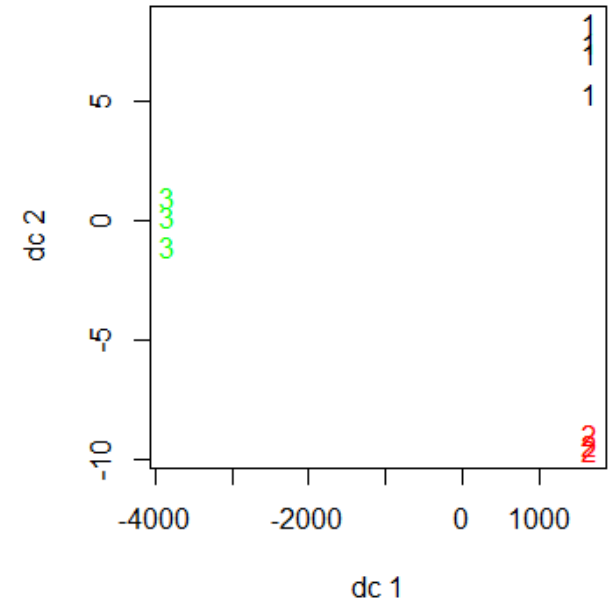
```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

Plotting the clusters

```
## plotting the clusters
## install.packages("fpc")

library(fpc)

plotcluster( scaled.RCDF, kmeans.clus$cluster )
```

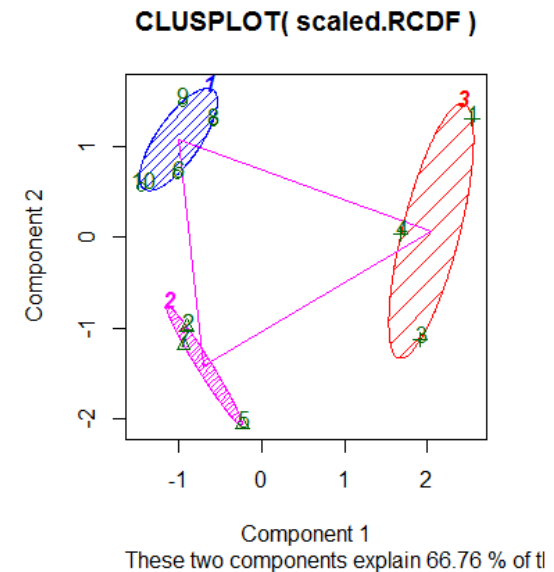


```
# More better plot

library(cluster)

?clusplot

clusplot(scaled.RCDF, kmeans.clus$cluster,
         color=TRUE, shade=TRUE, labels=2, lines=1)
```



Profiling the clusters

```
## profiling the clusters
```

```
KRCDF$Clusters <- kmeans.clus$cluster
```

```
aggr = aggregate(KRCDF[,-c(1,2, 8)],list(KRCDF$Clusters),mean)
```

```
clus.profile <- data.frame( Cluster=aggr[,1],  
                           Freq=as.vector(table(KRCDF$Clusters)),  
                           aggr[,-1])
```

```
View(clus.profile)
```

Cluster	Freq	Avg_Mthly_Spend	No_Of_Visits	Apparel_Items	FnV_Items	Staples_Items
1	4	2375.000	3.000000	0	1.250000	4.500000
2	3	7833.333	4.666667	1	1.666667	2.666667
3	3	5166.667	4.666667	0	11.000000	4.666667

Next steps after clustering

- Clustering provides you with clusters in the given dataset
- Clustering does not provide you rules to classify future records
- To be able to classify future records you may do the following
 - Build Discriminant Model on Clustered Data
 - Build Classification Tree Model on Clustered Data

References

- Chapter 9 : Cluster Analysis (<http://www.springer.com>)
 - Google search : “www.springer.com cluster analysis chapter 9”



K2 Analytics
Building Skills, Building Individuals

Questions???

Thank you

Contact Us
ar.jakhotia@k2analytics.co.in