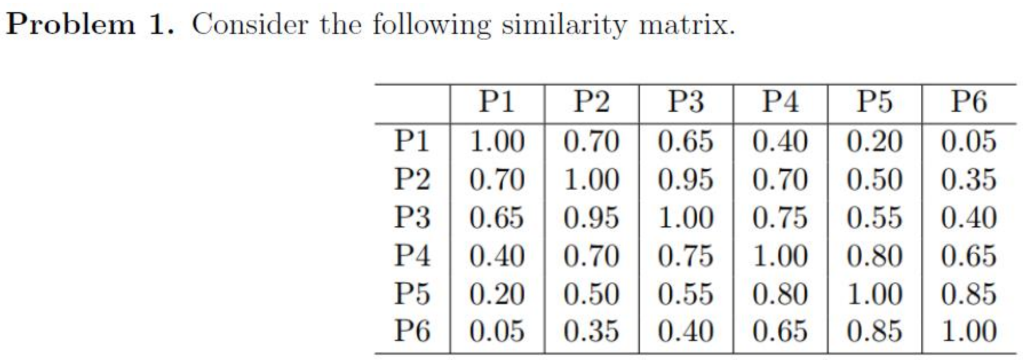
TEAM MEMBERS

1. Nithyadharshni Sampathkumar
2. Saai Krishnan Udayakumar
3. Sibi Senthur Muthusamy

Problem #1



Problem 1. Consider the following similarity matrix.

	P1	P2	P3	P4	P5	P6
P1	1.00	0.70	0.65	0.40	0.20	0.05
P2	0.70	1.00	0.95	0.70	0.50	0.35
P3	0.65	0.95	1.00	0.75	0.55	0.40
P4	0.40	0.70	0.75	1.00	0.80	0.65
P5	0.20	0.50	0.55	0.80	1.00	0.85
P6	0.05	0.35	0.40	0.65	0.85	1.00

Agglomerative clustering takes the assumption that initially every individual points are a clusters of their own and we combine individual clusters until we end up with a single cluster.

We are given with the Similarity matrix, we know that similarity is inversely proportional to distance involved, more the distance between the data points less similar they are. Hence, from the similarity matrix we are subtracting every observation by 1 so that they are converted to the distance measure.

Single Linkage Clustering Technique

"Single Link distance between clusters C (i) and C (j) is the minimum distance between any object in C (i) and C(j)"

Converting the Similarity Matrix to Distance matrix we yield the following Distance table, **(Iteration #1)**

Distance	P1	P2	P3	P4	P5	P6
P1	0	0.30	0.35	0.6	0.80	0.95
P2	0.30	0	0.05	0.3	0.50	0.65
P3	0.35	0.05	0	0.25	0.45	0.60
P4	0.60	0.30	0.25	0	0.20	0.35
P5	0.80	0.50	0.45	0.20	0	0.15
P6	0.95	0.65	0.6	0.35	0.15	0

From the above distance matrix, we infer that the lower distance is observed between the **Points P3 and Point P2 (0.05)** and they form a single cluster after the first iteration.

Iteration #2

Distance	P1	(P2, P3)	P4	P5	P6
P1	0	0.30	0.60	0.80	0.95
(P2, P3)	0.30	0	0.25	0.45	0.60
P4	0.60	0.25	0	0.20	0.35
P5	0.80	0.45	0.20	0	0.15
P6	0.95	0.60	0.35	0.15	0

Next set of clusters is formed between P5 and P6 as they form a single separate cluster on their own after the second iteration. (They have the minimum distance of 0.15 from the above table)

Iteration #3

Distance	P1	(P2, P3)	P4	(P5, P6)
P1	0	0.30	0.60	0.80
(P2, P3)	0.30	0	0.25	0.45
P4	0.60	0.25	0	**0.20**
(P5, P6)	0.80	0.45	0.20	0

Next set of clusters is formed between P4 and (P5, P6) as they form a cluster after the third iteration and they have a distance of 0.20.

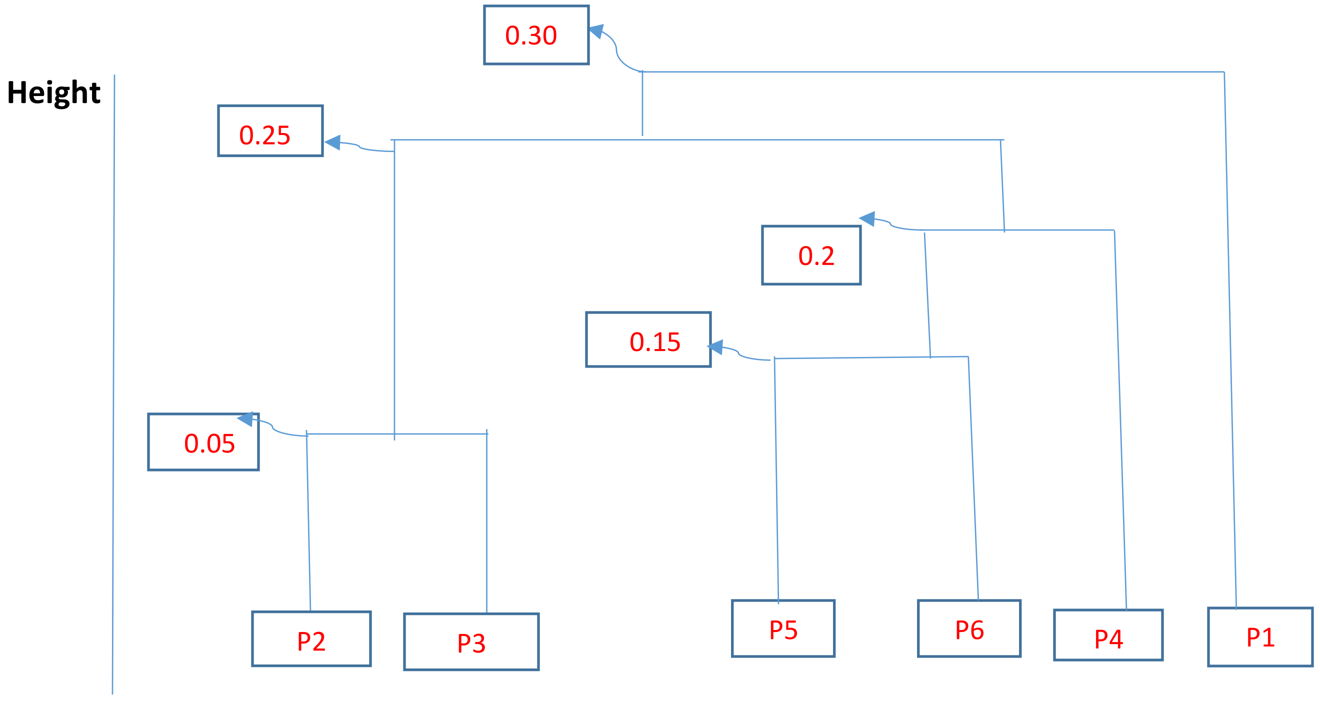
Iteration #4

Distance	P1	(P2, P3)	(P5, P6), P4
P1	0	0.30	0.60
(P2, P3)	0.30	0	0.25
(P5, P6), P4	0.60	0.45	0

Final merging is between clusters {(P2, P3) and (P5, P6), P4} as a single cluster which will finally be merged with cluster containing data point (P1)

Distance	P1	{(P2, P3), (P5, P6), P4 }
P1	0	0.30
{(P2, P3), (P5, P6), P4 }	0.30	0

DENDOGRAM FOR SINGLE LINK CLUSTERING



- Dendograms can be cut at various levels to obtain the number of clusters desired which is an advantage Agglomerative Clustering holds compared to k-means.

Complete Linkage Clustering Technique

Based on the distance table, the points having the minimum distance is chosen to be the first set of clusters. Thereby we form cluster including P1 and P6 initially, after which we compute the distance from each point to the cluster(s) formed.

"Complete-link distance between clusters C(i) and C(j) is the maximum distance between any object in C(i) and any object in C(j). The distance is defined by the two most dissimilar objects"

→ D(C(i),C(j)) → max {d(x, y)| x, y belongs Ci, Cj} (Iteration #1)

Distance	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**
P1	0	0.30	0.35	0.6	0.80	0.95
P2	0.30	0	0.05	0.3	0.50	0.65
P3	0.35	0.05	0	0.25	0.45	0.60
P4	0.60	0.30	0.25	0	0.20	0.35
P5	0.80	0.50	0.45	0.20	0	0.15
P6	0.95	0.65	0.6	0.35	0.15	0

Based on our first iteration the minimum distance is between P2 and P3 is 0.05, they form a single cluster at the end of first iteration.

(Iteration #2)

Distance	**P1**	**(P2, P3)**	**P4**	**P5**	**P6**
P1	0	0.35	0.60	0.80	0.95
(P2, P3)	0.35	0	0.3	0.50	0.65
P4	0.60	0.3	0	0.20	0.35
P5	0.80	0.5	0.20	0	0.15
P6	0.95	0.65	0.35	0.15	0

At the end of second iteration, cluster is formed between points P5 and P6 as they have a minimum distance of 0.15 among other clusters and points.

(Iteration #3)

Distance	**P1**	**(P2, P3)**	**P4**	**(P5, P6)**
P1	0	0.35	0.60	0.95
(P2, P3)	0.35	0	0.3	0.65
P4	0.60	0.3	0	0.35
(P5, P6)	0.95	0.65	0.35	0

Next set of Clusters is formed between points P4 and (P2, P3) as they have a distance of 0.3, so the clusters formed in Iteration #1 and Iteration #3 will be merged together.

(Iteration #4)

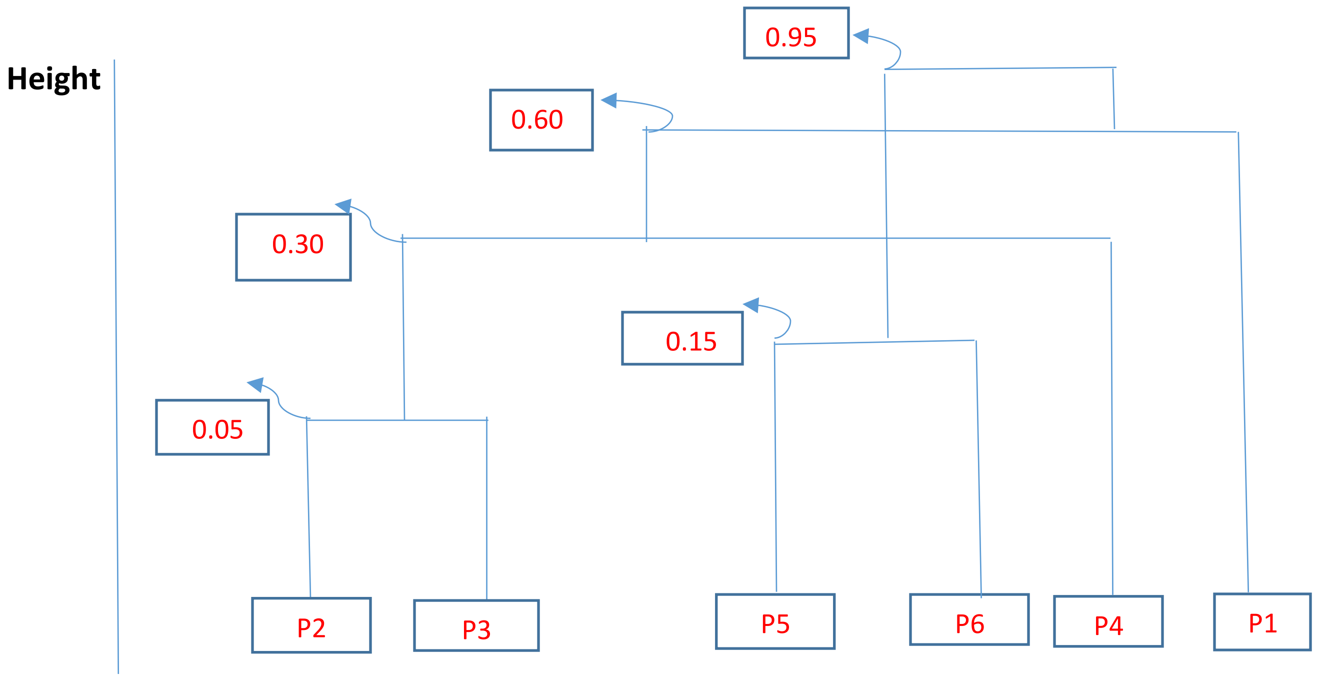
Distance	**P1**	**(P2, P3), P4**	**(P5, P6)**
P1	0	0.60	0.95
(P2, P3), P4	**0.60**	0	0.65
(P5, P6)	0.95	0.65	0

P1 and ((P2, P3), P4) form the next set of clusters and (P5, P6) will merged to form a single cluster.

(Iteration #5)

Distance	**(P5, P6)**	**((P2, P3), P4), P1)**
(P5, P6)	0	0.95
((P2, P3), P4), P1)	**0.95**	0

DENDOGRAM FOR COMPLETE LINKAGE CLUSTERING



##Problem 2

Kmeans Clustering assigns points to clusters based on their distance to the nearest initial centroid. After the initial centroids are assigned, the points are re-assigned to clusters again in the second iteration based on their nearest distance to the newly formed centroids which are the average of points in each cluster during the Initial Iteration. However, K-means Clustering is not suitable for all types of data as they cannot handle non-global clusters or clusters of different sizes and densities. We recommend that Agglomerative Clustering with Single Linkage Distance can perform a better job than K-means Clustering. Agglomerative Clustering (Bottom Up approach) initially assumes that all the individual points are clusters after which they are merged to form a single cluster. Also, single Linkage Agglomerative takes the minimum distance between cluster data points leaving points far way to be clusters at be combined last, which leads to clustering of single data point with the remaining cluster. Although it brings in the chaining effect, where clusters are spread out, there is a high likelihood that we will end up getting a single data point as a Cluster. We represent the Agglomerative Clusters in the form of a Dendrogram where we can extract the desired number of clusters. Clusters containing a single data point irrespective of their dimensionality is easier to be represented through Agglomerative Clustering. Upon making a cut at the desired height level we can get the required number of clusters. In that way we would yield a cluster containing many number of data points (Blue Points) as specified in the questions and another cluster with only single data point (Red Point). Hence, agglomerative Clustering is better than K-means for such type of clustering.

So, Preferred Clustering technique → Agglomerative Clustering ***(Hierarchical Clustering)***

Note: *There could be other types of Clustering models which are more robust and would perform well than Agglomerative in this scenario, for example the Gaussian Mixture models or the spherical k-means: However, we are restricting the comparison to the two clustering models under study and we infer that agglomerative would be the best.*

##Problem 3

Silhouette Coefficients

Silhouette Coefficient is a measure of how similar is an object is to its own cluster (cohesion) compared to other clusters (separation).

S = (b-a)/max (a, b)

a → average distance of point i to the points in the same cluster.

b → min (average distance of i to points in another cluster)

Cluster C1

a - (0,0)

b - (0,1)

c – (2,3)

d (a, b) – 1

d (a, c) – 5

Average Distance of ‘a’ to ‘b’ and ‘c’ would be (1+5)/2 →3 (a)

Distance of ‘a’ to points in the other cluster C2 →

Cluster C2

x - (3,3)

y- (3,4)

d (a, x) → 6

d(a, y) → 7

Average Distance of ‘a’ to points in the other cluster C2 → 6.5 (b)

Silhouette Coefficients of Point (a) → (6.5 – 3) /max(6.5, 3) → 0.538

Similarly repeating the procedure for all data points across cluster c1 and cluster c2,

d (b, a) – 1

d (b, c) – 4

Average Distance of ‘b’ to ‘a’ and ‘c’ would be (1+4)/2 →2.5 (a)

Distance of ‘b’ to points in the other cluster C2 →

d (b, x) → 5

d(b, y) → 6

Average Distance of ‘b’ to points in the other cluster C2 → 5.5 (b)

Silhouette Coefficients of Point (b) → (5.5 – 2.5) /max(2.5, 5.5) → 0.545

d (c, b) – 4

d (c, a) – 5

Average Distance of ‘c’ to ‘a’ and ‘b’ would be (4+5)/2 → 4.5 (a)

Distance of ‘b’ to points in the other cluster C2 →

d (c, x) → 1

d (c, y) → 2

Average Distance of ‘b’ to points in the other cluster C2 → (1+2)/2 →1.5 (b)

Silhouette Coefficients of Point (c) → (1.5 – 4.5) /max(1.5, 4.5) → (-) 0.666

Now we calculating the Silhouette Co-efficient for points in the other cluster2,

Distance between x and y is d(x, y) → 1 (a)

Distance of point x to cluster 1 (a, b, c) → 1, 5 ,6 → Average Distance being (1+ 5+ 6)/3 → 4 (b)

Silhouette Coefficients of Point (x) → (4 – 1) /max(4, 1) →0.75

Distance between x and y is d(x, y) → 1 (a)

Distance of point y to cluster 2 (a, b, c) → 7, 6, 2→ Average Distance being (7 + 6+ 2)/3 → 5 (b)

Silhouette Coefficients of Point (y) → (5 – 1) /max(5, 1) → 0.8

DATA POINTS	SILHOUETTE RATIO
a (0,0)	0.538
b (0,1)	0.545
c (2, 3)	(-) 0.667
x (3, 3)	0.75
y (3, 4)	0.80

The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. This measure has a range of [-1, 1].

Any positive value close to +1 indicates that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster.

From our Silhouette Ratio values, we infer that the points ‘x’ and ‘y’ in cluster 2 are far away from the nearby clusters. Points ‘a’ and ‘b’ in cluster 1 also seem to be well away from the decision boundary. However, point ‘c’ seem to be wrongly assigned to cluster 1 as they have a negative silhouette ratio and therefore must be assigned to cluster 2.

Clusters	**Silhouette Ratio**
Cluster #1	0.13866
Cluster #2	0.775

We understand that from Cluster #2 has a higher Silhouette Ratio than Cluster #1. This could be because a certain point c (2, 3) has a negative ratio and has high likelihood of being in the wrong cluster. Points in Cluster #2 are tightly packed and they have high probability of being close centers wherein their inter cluster distance could be large as well. Average silhouette width of 0.71 -1 indicates that a strong structure has been found and hence Cluster #2 has a strong structure. A value less than 0.25 shows no substantial structure has been found.

Henceforth, Cluster #2 (0.775) is better represented than Cluster #1 (0.13866) from the Silhouette values we had obtained.

Average Silhouette of the Entire Dataset would be → *(0.538 + 0.545 - 0.667 + 0.75 + 0.8)/5 →0.393*

The Average Silhouette of the Entire Dataset would be 0.393, hence we could say that the data points could be Clustered better and Clustering needs to be improved particularly by classifying the point c(2, 3) into Cluster #2.

#Assignment #5 (problem #3) Customer Segmentation

##Loading the library

```
install. packages("xlsx")

library(xlsx)

purchases<-read.xlsx(file.choose(), sheetName="DM_Sheet", header = T)
```

##Reading the data file with the corresponding sheet name on which we perform the analysis

```
View(purchases)
```

##We are reading the CRISA Purchase data containing information on all the transactions.

#Once we load the data, we are try to understand the data types of each variable and the ensuing classes.

#Hence, we use the str function as follows,

```
str(purchases)
```

##We are trying to filter out columns making sure that there are no import errors and also that we are working on data points only needed for our cluster construction purposes, we also make sure that all the variables we cluster on are numeric and if there are any nominal or ordinal variables we handle them appropriate.

```
purchases_df<-purchases [,1:46]

View(purchases_df)

str(purchases_df) ##Analyzing the segmented data frame

purchases_df_cluster<-purchases_df[1:600,]
```

##Filtering out the exact 600 rows needed so that we don't add any extraneous data points.

##Question 1 (a)

#Variables that describe Purchase Behavior given to us are the #Number of Brands, Brand Runs, Total Volume, #No of Transactions, #Value,

#Average Price, Share to Other brands, Max to one brand

#We have the attribute 'Others 999' which we could use to depict the share to other brands.

#We also create a new attribute **'brand_loyalty_new'** to illustrate brand loyalty. 'brand_loyalty_new' is the maximum value of the brand columns (Not including Others999) for a given row and is basically the inclination of a household for a given brand. Also, we are not considering the individual brand purchase percentages to account for the fact that we are considering brand loyalty and not brand loyalty to a given brand.

#Factors governing brand loyalty also include **#Number of different brands -** The more the number of brands purchased the less loyal they are to one of them.

#the lesser the number of brands purchased, they are most loyal to that one.

#Number of brand runs - A measure of how often a particular household switch from one brand to another. If the value is high, it implicitly means that the customer had a long streak of purchasing a particular brand and henceforth is more loyal than a customer who has a smaller value for this attribute.

#Variable Selection for unsupervised learning is a crucial step as it is very sensitive to the variables that are being considered.

#Any superfluous attribute included has the tendency to increase the distance value unnecessarily.

Deriving a new attribute to measure brand loyalty

```
purchases_df_cluster$brand_loyalty_new<-with(purchases_df_cluster,pmax(Br..Cd..55,
Br..Cd..57..144, Br..Cd..24,Br..Cd..272,Br..Cd..286,Br..Cd..352,Br..Cd..481,Br..Cd..5))
```

#deriving the maximum of the above attributes

```
View(purchases_df_cluster)
```

#k-means clustering on brands, brand runs, total volume, #Transactions, Value, #Avg.price, #Share to other brands, #max to one brand

#Filtering out the above columns from the Summary transaction data given to us,

```
Vars<-c('No..of.Brands','Brand.Runs','Total.Volume',
'No..of..Trans','Value','Avg..Price.','Others.999','brand_loyalty_new')
```

Variables to be included for measuring purchase behavior and brand loyalty.

```
vars_in_df<-which(names(purchases_df_cluster) %in% e)

brand_cluster<-purchases_df_cluster[,vars_in_df]

View(brand_cluster)

str(brand_cluster)
```

#We understand that all the variables taken into consideration by us are numeric, and there are no special methods required to handle the categorical variables.

```
sapply(brand_cluster, function(x) sum(is.na(x))) ##We infer that there are no missing values
```

##We don't seem to be having NA values nor any outliers,

#k-means algorithm requires us to specify the value of k (i.e.) number of clusters to be performed on the data,

#Before, we perform clustering, we do the summary function to understand the need for normalizing the data,

```
summary(brand_cluster)
```

- #As we could see, there is a vast difference between the scale of values (For, Example) in Brand. Runs compared to the 'Value' variable,
- #Such a scenario would expect us to normalize the data (Typically, normalization standard like min-max) so that we can get all values in the range [0,1]

#Normalization Function

```
normalize<-function(x)

{

  num<-x-min(x)

  denom<-max(x)-min(x)

  return (num/denom)

}

 brand_cluster_norm<-as.data.frame(lapply(brand_cluster[1:8],normalize))

summary(brand_cluster_norm) #All our attribute values are between [0,1]
```

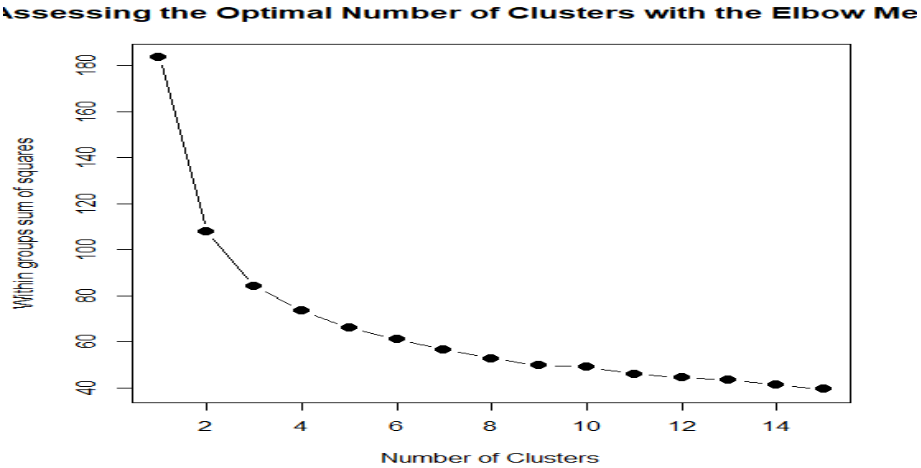
#We have normalized our data set using the min-max normalization technique after which we evaluate the summary function to determine the minimum and maximum values,

Using the ScreePlot to measure the Optimal Number of Clusters

```
set.seed(1234)

mydata <- brand_cluster_norm

nrow(mydata)

wss <- (nrow (mydata)-1)*sum(apply(mydata,2, var))

sum(apply(mydata,2,var))

for (i in 2:15)

  wss[i] <- sum(kmeans(mydata, centers=i)$withinss)

plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares",

     main="Assessing the Optimal Number of Clusters with the Elbow Method", pch=20, cex=2)
```

- The major evaluation function for choosing the optimal number of clusters would be the within cluster sum of squares.
- Clusters which minimize the within cluster “sum of squared Euclidean Distance” (Intra Cluster Distance) and maximize the between cluster sum of squares (Inter Cluster Distance) are likely to be good clusters.
- Hence, continuing our observation when k=number of data points the within cluster sum of squares will basically be zero,

- **We chose the optimal cluster value at points where there is a sudden drop in the 'Scree Plot', the point is called the "Elbow Point".**



#Now we perform k-means with k=4, as we observe a drop in the scree plot from k=4

#perform K-Means with the optimal number of clusters identified from the Elbow method

For now we assume K=4 to be giving us the optimal number of clusters, we also perform k-means clustering for other values of K to check the inter cluster and intra cluster distances.

set.seed(7)

km4=kmeans(mydata,4,nstart=100)

km4$cluster ##we get individual of assignment of cluster for each data point

# "nstart" option attempts multiple initial configurations and reports on the best one.

#nstart=100 will generate 100 initial configurations

km4$withinss

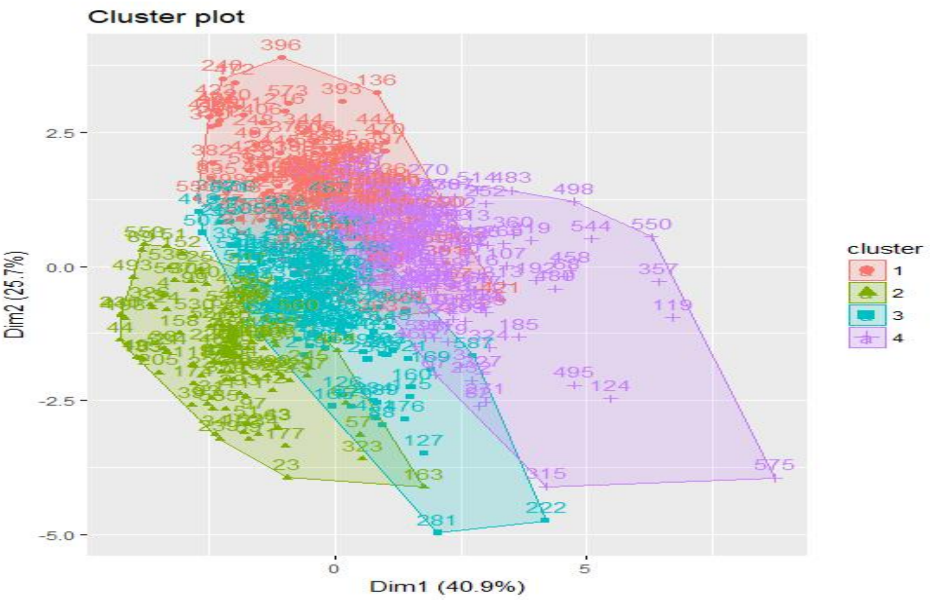
#We can see that the within sum of square distances are '21.20124247' '10.84584799' '17.93006048' '23.45211503'

km4$betweenss

#110.0280985

- # We want them to as high as possible for a cluster (Inter Cluster distance between the clusters) to be regarded as good and
- km4$size ##Size of the other four clusters **##157 152 105 186 being the size of the four clusters**.
- # The ratio of between_ss/total_ss should approach 1 as a measure of the goodness of the classification k-means has found.
- ## We also plot the results using fviz_cluster which provides a nice illustration of the clusters. If there are more than #2 dimensions (variables)
- # fviz_cluster will perform principal component analysis(PCA) and plot the data points according to the first two principal components that explain the majority of the variance.

```
install.packages("factoextra")

library(factoextra)

fviz_cluster(km4, data=mydata)
```



## To Evaluate the data points in Cluster #1 among the four clusters formed, we use the following code

#Cluster 1 Inference

```
Cluster1Instances<-mydata[km4$cluster==1,]

Cluster1Instances

summary(Cluster1Instances)
```

- Households in cluster #1 seem to have switched a fairly higher number of brands and they don't seem to be preferring a particular brand for a longer period of time as the,
- Average #Brand.Runs value based on Cluster #1 is 0.3634936.
- Also, the households in Cluster #1 don't seem to be frequent shoppers as the average value of Total Purchases and Total Transactions seem to be low.

#Cluster 2 Inference

```
Cluster2Instances<-mydata[km4$cluster==2,]

Cluster2Instances

Summary(Cluster2Instances)
```

- Households in Cluster #2 seem to be more loyal customers, as the average value of No of Brands seem to be comparatively less than what we observed in Cluster #1.
- Also, the purchasing power of people in Cluster #2 is comparatively less than households in Cluster #1

#Cluster 3 Inference

Cluster3Instances<-mydata[km4$cluster==3,]

Cluster3Instances

summary(Cluster3Instances)

- Customers in Cluster #3 seem to have purchased the least number of brands, and as a result they can be regarded as the most loyal customers.
- Also, Customers in Cluster #3 have a higher brand loyalty (~0.8480) (i.e., newly created brand loyalty variable) which are supported by lower mean values in Number of Brands and Brand. Runs.

#Cluster 4 Inference

Cluster4Instances<-mydata[km4$cluster==4,]

Cluster4Instances

summary(Cluster4Instances)

- Households in Cluster #4 have a higher #Others.999 value denoting they have higher other brand purchases and they also seem to have low brand loyalty.
- Also, these households are not frequent shoppers (low "Volume" and "No of Transactions")

General, overview of the Between Cluster Sum of Squares, and the between cluster by total Cluster Sum of Squares metric, we need the between cluster metric and the ratio value to be high as well. However, from the table and the scree plot for k=4, we find that the drop in inter cluster distance is comparatively significant compared to other values of 'K'. **Hence, we prefer "K=4" as the preferred choice for Clustering based on Purchase Behavior and Brand Loyalty**

'K'	**Between Cluster (SS)**	**(Between Cluster/Total) %**
2	75.5224	41.2 %
3	99.215	54.1 %
4	110.0289	60.0 %
5	117.43084	64.0 %
6	122.6201	66.8 %
7	126.886	69.2 %

##Part (b)

## We consider the following variables to describe basis-for-purchase (Pur-vol-no-promo, Pur-vol-promo-6, Pur-vol-other) which are percentage of volume purchased under no-promotion, percentage of volume purchased under Promotion Code 6 and percentage of purchased under other promotion.

# Further we take all price categories (indicated by variables #Pr Cat 1, #Pr Cat 2, #Pr Cat 3, #Pr Cat 4) and we create a new variable called the **"Maximum Selling Proportion"** obtained by taking the

maximum of **max([PropCat15], [PropCat11], ... [PropCat5]))** Which is the maximum of percentage of volume purchased under the product proposition category.

```
purchases_df_cluster$MaximumSellingProportion<-with(purchases_df_cluster,pmax(PropCat.5, PropCat.6, PropCat.7, PropCat.8, PropCat.9, PropCat.10, PropCat.11, PropCat.12, PropCat.13, PropCat.14, PropCat.15))
```

#Some of the proposition Categories (From PropCat.10-13) have their distributions tending to zero, but we decided to keep them in tact so as to avoid any information loss considering that some of the data points have valid data and that we are only working with a total of 600 data points. We all ran the clustering models without the PropCat. 10-13 and there wasn't significant difference the two clustering models.

```
var_df1<-
c('Pur.Vol.No.Promo....','Pur.Vol.Promo.6..','MaximumSellingProportion','Pur.Vol.Other.Promo..','Pr.Cat.1','Pr.Cat.2','Pr.Cat.3','Pr.Cat.4')

vars_include<-which(names(purchases_df_cluster) %in% var_df1)

purchase_basis<-purchases_df_cluster[,vars_include]

View(purchase_basis)

View(purchases_df_cluster)

str(purchase_basis)
```

#We understand that all the variables are numeric and hence there is no need to handle ordinal or nominal variables except that we need to check if normalizing is necessary.

```
summary(purchase_basis)
```

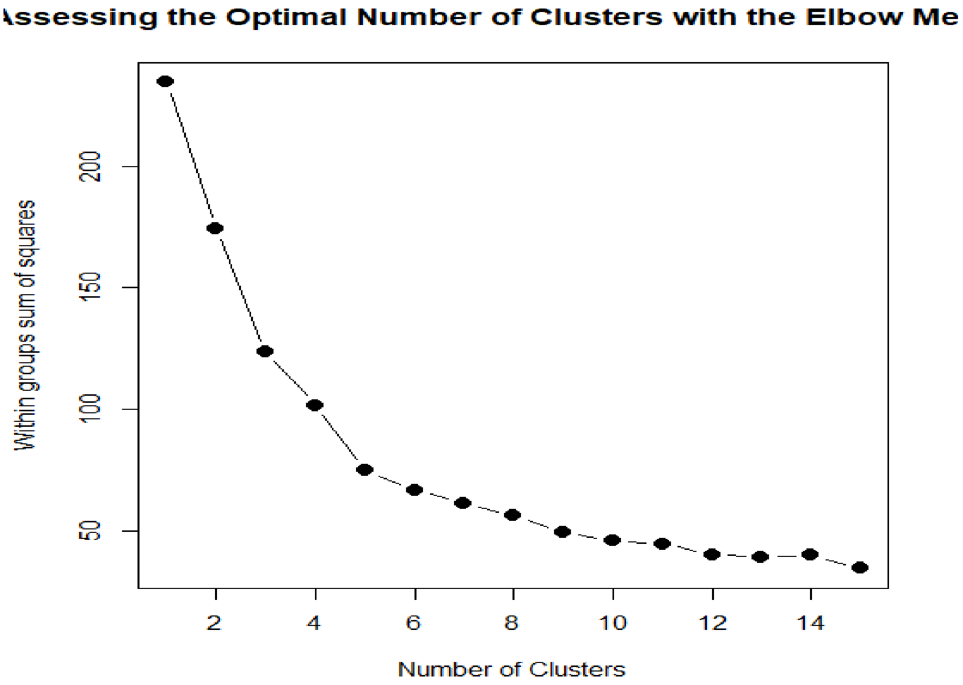
Normalization Function

```
normalize<-function(x)

{

  num<-x-min(x)

  denom<-max(x)-min(x)

  return (num/denom)

}

purchase_cluster_norm<-as.data.frame(lapply(purchase_basis[1:8],normalize))

summary(purchase_cluster_norm)
```

Using the Scree Plot to measure the Optimal Number of Clusters

```
set.seed(1234)

mydata <- purchase_cluster_norm
```

```
nrow(mydata)

wss <- (nrow(mydata)-1)*sum(apply(mydata, 2,var))

sum(apply(mydata,2, var))

for (i in 2:15)

  wss[i] <- sum(kmeans(mydata, centers=i)$withinss)

plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares",

     main="Assessing the Optimal Number of Clusters with the Elbow Method", pch=20, cex=2)
```



#the major evaluation function for choosing the optimal number of clusters would be the within cluster sum of squares.

#Clusters which minimize the within cluster sum of squares and maximize the between cluster sum of squares are likely to be a good cluster.

#Hence, continuing our observation when k=number of data points the within cluster sum of squares will basically be zero,

#We chose the optimal cluster value at points where there is a sudden drop in the 'Scree Plot'

#Now we perform k-means with k=3 and then continue the clustering process to observe which clustering efforts give us the best results as we observe a drop in the scree plot from k=3 to k=n.

#Perform K-Means with the optimal number of clusters identified from the Elbow method

## 3 Means Clustering

set.seed (7)

km3=kmeans(mydata, 3, nstart=100)

km3$cluster ##We get individual of assignment of cluster for each data point

#nstart option attempts multiple initial configurations and reports on the best one. nstart=100 will generate 100 initial configurations

Km3$withinss

#We can see that the within cluster sum of square distances are **35.74424 49.32898 38.64882**

Km3$betweenss

#**110.6793**

#We want them to as high as possible for a cluster (Inter Cluster distance between the clusters) to be regarded as good and

km3$size ##Size of the other three clusters ## 177 311 112 being the size of the three clusters.

#the ratio of between_ss/total_ss should approach 1 as a measure of the goodness of the classification k-means has found.

Km3 # (between_SS / total_SS = 47.2 %)

## 4 Means Clustering

set.seed (7)

km4=kmeans(mydata, 4, nstart=100)

km4$cluster ##We get individual of assignment of cluster for each data point

#nstart option attempts multiple initial configurations and reports on the best one. nstart=100 will generate 100 initial configurations.

Km4$withinss

#We can see that the within sum of square distances are **#7.659503 #10.927 #27.36 #51.37**

Km4$betweenss

#**137.0752**

#We want them to as high as possible for a cluster (Inter Cluster distance between the clusters) to be regarded as good and

Km4$size ##Size of the other four cluster ## 58 322 150 70 being the size of the three clusters.

#the ratio of between_ss/total_ss should approach 1 as a measure of the goodness of the classification k-means has found.

***Km4* # (between_SS / total_SS = 58.5 %)**

Now, we examine different values of ‘k’ and try to find which ‘k’ value minimizes the variance (i.e. the intra cluster distance)

Not Much Difference observed by dropping the #4 variables.

Between Cluster Values and the Ratio’s for different values of ‘K’

‘K’	**Between Cluster (SS)**	**(Between Cluster/Total) %**	**Between Cluster (SS)**	**Between Cluster/Total) %**
2	59.93	25.6 %	59.74	26.7%
3	110.67	47.2 %	110.21	49.3%
4	137.075	58.5 %	136.85	61.2%
5	159.2	67.9 %	153.2	68.5%
6	168	71.7 %	160.42	71.8%
7	174.57	74.5 %	166.64	74.5%

From the table, we conclude that for **‘k=5’ we observe the sharp increase in Inter Cluster Distance (159.2) and the ratio of Between Cluster to Total Cluster Distance from partitioning the cluster with K=4 to K=5 or from K=6 to other Values of K.** We prefer higher ‘K’ because of the high dimensionality of features we are taking into consideration. Also, we could observe that the between cluster sum of squares increases as we increase ‘K’ which could be attributed to the decreasing within cluster sum of squares as data points converge near their centroid for higher ‘k’ values and Between Sum of Squares is essentially **‘Total Sum of Squares – Within Sum of Squares’.**

FIVE PARTITION CLUSTERING

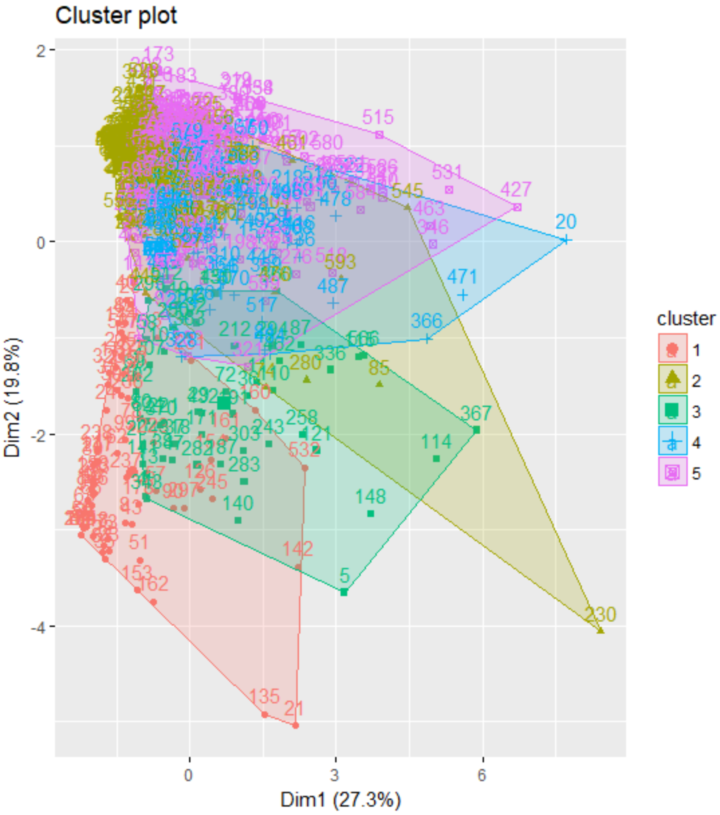
```
set.seed(7)
km5=kmeans (mydata,5, nstart=100)
km5$cluster

##We get individual of assignment of cluster for each data point
#nstart option attempts multiple initial configurations and reports on the best one. nstart=100 will
generate 100 initial configurations

km5$withinss
#We can see that the within sum of square distances are #9.937765 #17.129889 #15.559522
#24.919258 #7.659503

km5$betweenss
#159.1954
#We want them to as high as possible for a cluster (Inter Cluster distance between the clusters) to be
regarded as good and
```

km5$size ##Size of the other four cluster **## '55' #'108' #'173' #'194' #'70'** being the size of the five clusters.

#The ratio of between_ss/total_ss should approach 1 as a measure of the goodness of the classification k-means has found.
km5 # (between_SS / total_SS = 67.9 %)

#fviz_cluster will perform principal component analysis(PCA) and plot the data points according to the first two principal components that explain the majority of the variance.

```
install.packages("factoextra")
library(factoextra)
fviz_cluster(km5, data=mydata)
```



#Cluster 1 Inference

```
Cluster1Instances<-mydata[km5$cluster==1,]
Cluster1Instances
summary(Cluster1Instances)
```

#Customers in Cluster 1 are suitable targets for economical and carbolic products. Hence the selling propositions for cheaper products will work the best for this cluster.

#Cluster 2 Inference

```
Cluster2Instances<-mydata[km5$cluster==2,]
Cluster2Instances
summary(Cluster2Instances)
```

#Cluster 2 households would be the most suitable target for selling premium soaps and beauty products. Selling propositions for premium brands will work for this cluster.

#Cluster 3 Inference

```
Cluster3Instances<-mydata[km5$cluster==3,]
Cluster3Instances
summary(Cluster3Instances)
```

#Cluster 3 customers would be the best target for beauty products and popular soap brands. Cluster 2 customers are among the highest users of beauty products.

#Cluster 4 Inference

```
Cluster4Instances<-mydata[km5$cluster==4,]
Cluster4Instances
summary(Cluster4Instances)
```

#Cluster 4 customers form a good target for popular brands in beauty and health products. Since Cluster 3 and 4 constitute 25% of the customer base, focus can be given on targeting selling propositions on this customer base.

#Cluster 5 Inference

```
Cluster5Instances<-mydata[km5$cluster==5,]
Cluster5Instances
summary(Cluster5Instances)
```

#Cluster 5 constitute one of the highest consumer of beauty products and lesser-popular soap brands. Promotion code 6 works best for cluster 5.

PART (c)

a<-cbind (brand_cluster_norm, purchase_cluster_norm) *#We are using the normalized variables used by us in constructing clusters for demonstrating purchase behavior and basis-for-purchase.*

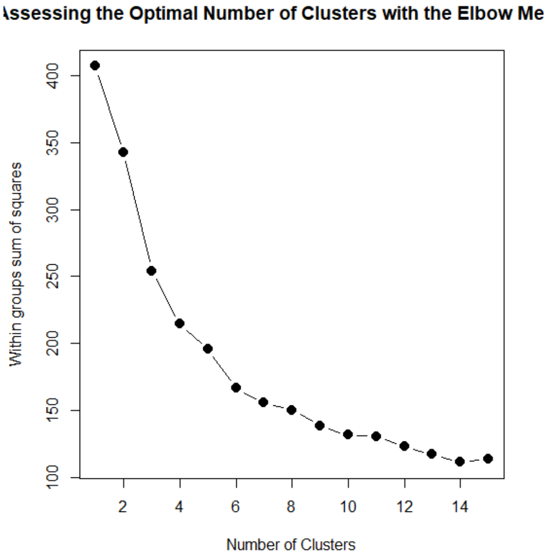
View(a)

```
df1<-a[,-6]

View(df1)
```

The variables used by us would be the following:
No. of. Trans, Value, BrandRuns, No. of. Brands, Others999, All Price Categories (1-4),
Pur_vol_no_promo, Pur_vol_promo_6, Pur_vol_other, MaximumSellingProposition, Brand_Loyalty_new

Using the Scree Plot to measure the Optimal Number of Clusters

```
set.seed(1234)

mydata <- purchase_cluster_norm

nrow(mydata)

wss <- (nrow(mydata)-1)*sum(apply(mydata, 2,var))

sum(apply(mydata,2, var))

for (i in 2:15)

  wss[i] <- sum(kmeans(mydata, centers=i)$withinss)

plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares",

     main="Assessing the Optimal Number of Clusters with the Elbow Method", pch=20, cex=2)
```



#the major evaluation function for choosing the optimal number of clusters would be the within cluster sum of squares.

#Clusters which minimize the within cluster sum of squares and maximize the between cluster sum of squares are likely to be a good cluster.

#Hence, continuing our observation when k=number of data points the within cluster sum of squares will basically be zero,

#We chose the optimal cluster value at points where there is a sudden drop in the 'Scree Plot'

#Now we perform k-means with k=3 and then continue the clustering process to observe which clustering efforts give us the best results as we observe a drop in the scree plot from k=3

#Perform K-Means with the optimal number of clusters identified from the Elbow method

## 3 Means Clustering

set.seed (7)

km3=kmeans(mydata, 3, nstart=100)

km3$cluster ##We get individual of assignment of cluster for each data point

#nstart option attempts multiple initial configurations and reports on the best one. nstart=100 will generate 100 initial configurations

Km3$withinss

#We can see that the within sum of square distances are **81.6260 142.02287 30.45367**

Km3$betweenss

#**152.8213**

#We want them to as high as possible for a cluster (Inter Cluster distance between the clusters) to be regarded as good and

km3$size ##Size of the other four cluster **## 226 293 81 being the size of the three clusters.**

#the ratio of between_ss/total_ss should approach 1 as a measure of the goodness of the classification k-means has found.

Km3 # (between_SS / total_SS = 37.6 %)

## 4 Means Clustering

set.seed (7)

km4=kmeans(mydata, 4, nstart=100)

km4$cluster ##We get individual of assignment of cluster for each data point

#nstart option attempts multiple initial configurations and reports on the best one. nstart=100 will generate 100 initial configurations.

Km4$withinss

#We can see that the within sum of square distances are **#81.88 #17.11 #45.678 #69.76**

Km4$betweenss

#**192.4864**

#We want them to as high as possible for a cluster (Inter Cluster distance between the clusters) to be regarded as good and

Km4$size ##Size of the other four cluster **## 250 71 138 141 being the size of the three clusters.**

#the ratio of between_ss/total_ss should approach 1 as a measure of the goodness of the classification k-means has found.

Km4 **# (between_SS / total_SS = 47.3 %)**

Now, we examine different values of ‘k’ and try to under which ‘k’ value tries to minimize the variance (i.e. the intra cluster distance)

## 5 Means Clustering

set.seed (7)

km5=kmeans(mydata, 5, nstart=100)

km5$cluster ##We get individual of assignment of cluster for each data point

#nstart option attempts multiple initial configurations and reports on the best one. nstart=100 will generate 100 initial configurations.

Km5$withinss

#We can see that the within sum of square distances are **#17.11 #29.99 #38.1 #14.639 #85.35**

Km5$betweenss

#221.7126

#We want them to as high as possible for a cluster (Inter Cluster distance between the clusters) to be regarded as good and

Km5$size ##Size of the other four cluster **## 71 123 98 53 255 being the size of the three clusters.**

#the ratio of between_ss/total_ss should approach 1 as a measure of the goodness of the classification k-means has found.

Km5**# (between_SS / total_SS = 54.5 %)**

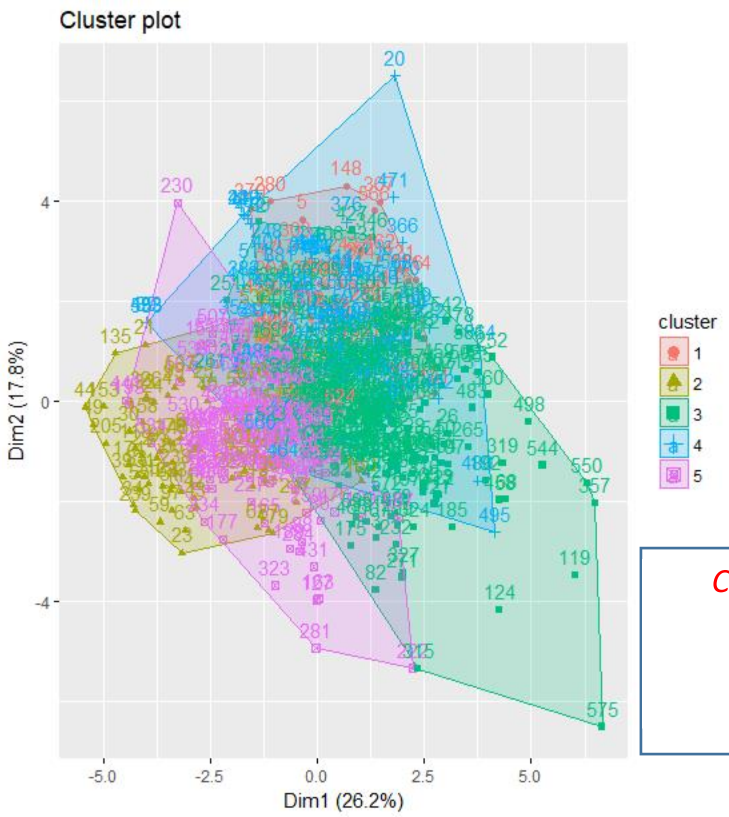
Now, we examine different values of ‘k’ and try to under which ‘k’ value tries to minimize the variance (i.e. the intra cluster distance)

Between Cluster Values and the Ratio’s for different values of ‘K’

‘K’	Between Cluster (SS)	(Between Cluster/Total) %
2	100.78	24.9 %
3	152.82	37.6 %
4	192.48	47.3 %
5	221.8	54.5 %
6	240	59 %
7	252.02	61.9 %

From the table, we conclude that for **‘k=4’** and **‘k=5’ we observe a Considerable increase in Inter Cluster Distance and also an increase in the ratio of Between Cluster to Total Cluster Distance from partitioning the cluster with K=3 to K=4 or from K=6 to other Values of K.** Also, as the features considered in our clustering process increases we observe a considerable drop in inter cluster distance even for higher values of K. So, we consider a five means clusters, and try to infer the data points within each cluster to begin with.

FIVE PARTITION CLUSTERING



```
Cluster1Instances<-mydata[km5$cluster==1,]
Cluster1Instances
summary(Cluster1Instances)
```

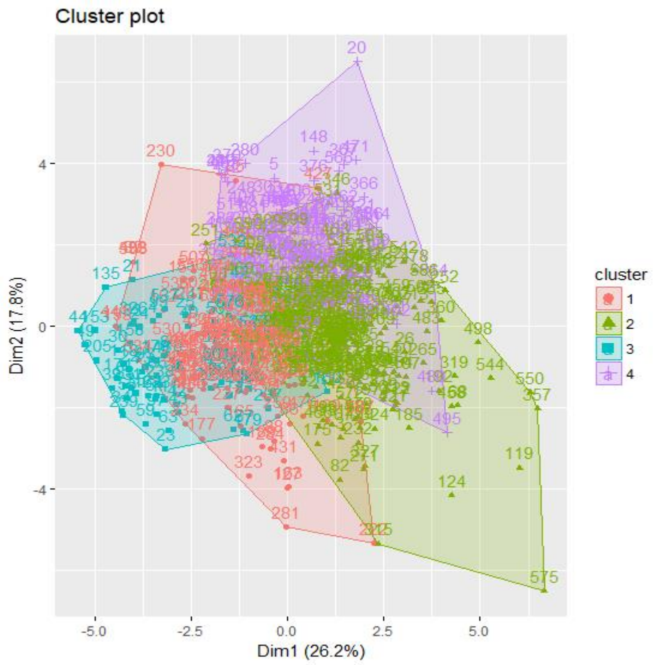
Fetching the Observation from Each Cluster

```
install.packages("factoextra")
library(factoextra)
fviz_cluster(km5, data=mydata)
```

- ***Cluster 1 and Cluster 5***: *Customers in Cluster 1 and Cluster 5 do not have good brand loyalty because they buy from several brands.*
- ***Cluster 2:*** *Cluster 2 customers would have the least loyalty, prefer sub-popular brands and also buy the maximum of the beauty products.*
- ***Cluster 3:*** *Cluster 3 customers who mainly buy soap and beauty products have the best brand loyalty as indicated by high brand loyalty and high total volume variables.*
- ***Cluster 4***: *High market penetration indicated buy high brand loyalty, decent volume of purchases and number of transactions.*

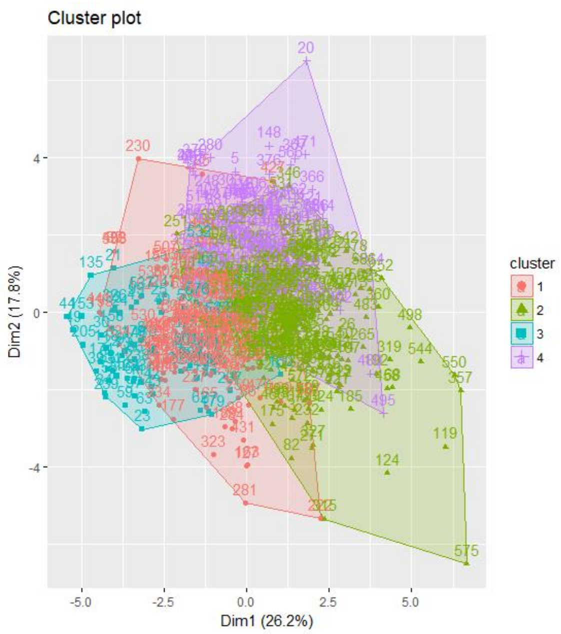
FOUR PARTITION CLUSTERING

```
install.packages("factoextra")

library(factoextra)

fviz_cluster(km5, data=mydata)
```



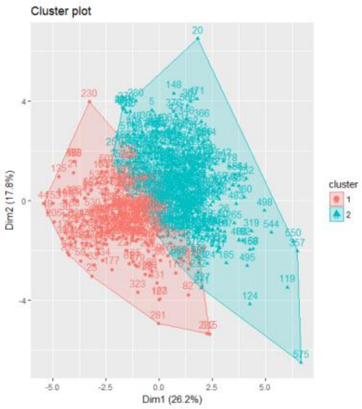
- ***Cluster 1***: *Customers in Cluster 1 buy from a decent number of brands and hence brand loyalty is medium. They purchase high volume of products, high value and higher average price. Targeting this cluster would yield profitability.*
- ***Cluster 2:*** *Cluster 2 customers are best in terms of brand loyalty and profitability with good volume and value of products.*
- ***Cluster 3:*** *Cluster 3 buy from the least number of brands, high volume of products but the value is low. Maximum Brand Loyalty indicates that cluster 3 is the most loyal cluster and can be targeted for economical brands and carbolic categories of products.*
- ***Cluster 3***: *Customers belonging to cluster 3 are the least loyal.*

THREE PARTITION CLUSTERING



- ***Cluster 1***: *Cluster 1 buys higher number of brands than cluster 2, has medium brand loyalty and the value is good. It is suitable target for selling propositions for popular brands and for beauty products.*
- ***Cluster 2:*** *Cluster 2 has the least brand loyalty*
- ***Cluster 3:*** *Uses least number of brands, high total volume, highest brand loyalty but the value is far less than the other 2 clusters. Targeting achieves market penetration but not as much profitability due to its smaller size*.

TWO PARTITION CLUSTERING



#Two partition clustering have 214 members in the first cluster and 386 in the second cluster. These two clusters vary significantly in brand loyalty and volume of purchases.

Question #2

Based on our analysis using the constructed clustering models and multiple iterations our best model is a k-means clustering model with k = 4. For this model, the intra cluster distance is small meaning that the data points in the cluster are similar and the inter cluster distance between the clusters is good and the size of the clusters is also considerably good. Hence we concluded that this was our best model.

Households in cluster 1

- *Customers in Cluster 1 have comparatively lesser number of purchases from other brands (lower "Others 999") and higher brand loyalty value ("Maximum Brand Loyalty").*
- *Customers are frequent shoppers (comparatively high "Volume" and "No of Transactions")*
- *Brand Runs is comparatively high and no. of brands purchased is low.*
- *Customers in Cluster 1 are suitable targets for soap (popular soap brands) and beauty products.*
- *Cluster 1 customers are also a good option to target for cheaper brands.*
- *Low percent of volume purchased on promo code 6 and on products purchased on promo code other than 6.*

Households in cluster 2

- *Customers have more other-brand purchases (high "Others 999") and lowest brand loyalty (low "Maximum_Brand_Loyalty").*
- *Customers are not frequent shoppers (low "Volume" and "No of Trans")*
- *Brand Runs is less meaning the number of streaks of purchasing the same brand is less.*
- *Cluster2 customers would be the most suitable target for selling popular soaps.*
- *Promotion code 6 work best for customers in Cluster 2.*
- *Highest percent of volume on promo code 6 and lowest percent of volume on promo code other than 6.*

Households in cluster 3

- *Customers in Cluster 2 make more purchases in other brands (high "Others999) and have the low brand loyalty (low maxBrCd).*
- *They also are not frequent shoppers (lower "Volume" and "No of Trans")*
- *Brand Runs and No. of Brands used are less.*
- *Cluster 3 customers would be the best target for economic products.*
- *Lowest percent of volume on promo code 6 and highest percent of volume on promo code other than 6.*

Households in cluster 4

- *Customers make least purchases in other brands (high "Others999) and have the best brand loyalty (high Maximum_Brand_Loyalty).*
- *They also are not frequent shoppers (low "Volume" and "No of Trans")*
- *Brand Runs and No. of Brands used are less.*
- *Cluster 4 customers form a good target for premium brands in soap, beauty and hair products and health products.*

DEMOGRAPHICS ATTRIBUTES AND OTHER CHARACTERISITICS

Cluster 1

- *High socio-economic status.*
- *High value in no education category.*

- *Low Affluence index.*
- *Household size of approximately 3 people.*
- *Major Marathi Speaking People.*
- *Households with Television and Broadcast TV Cable.*

Cluster 2

- *High number of Non-vegetarian.*
- *High number of Females.*
- *Households mostly aged above 45 years.*
- *Primarily Speaking Marathi.*
- *Females belonging to a Socio Economic class 'A'*
- *With low level of education (10-12 years).*
- *And with a house size of around 4 people.*

Cluster 3

- *High number of vegetarians.*
- *Greater value of Affluence index.*
- *With television or broadcast TV cable.*
- *Majorly* Speaking Marathi and Gujarati
- *Females belonging to a Socio Economic class 'A'*

Cluster 4

- *Females belonging to a Socio Economic class 'D/E'*
- *House size of around 4 people.*
- *High number of homemakers with age 45+*
- *High number of households with no children.*
- *High number of households Speaking Marathi and Assamese*