Kyadav1\_Assignment5

Khushboo Yadav

11/27/2020

#### 1. Importing Libraries and “Cereals” dataset

library(dplyr) # for data manipulation  
library(ggplot2) # for data visualization  
  
# Modeling packages  
library(cluster) # for general clustering algorithms  
library(factoextra) # for visualizing cluster results  
library(NbClust) # for Nbclust ()  
library(clValid) #for cluster Validation  
library(caret) # for data partition  
library(dendextend) # for Visual comparison of dendrogram  
library(fpc) # for stability  
  
Cereals <- read.csv("~/Downloads/Cereals.csv")  
  
  
#removing NA values only for the labeling purpose in dendrogram and last part  
Cereals\_N<- na.omit(Cereals)

#### 2. Data pre-processing :

##### 2.1. Removing columns (mfr, type,shelf) from the cereals dataset

mfr,type,shelf are categorical variables and we can remove them .

summary(Cereals)

## name mfr type calories   
## Length:77 Length:77 Length:77 Min. : 50.0   
## Class :character Class :character Class :character 1st Qu.:100.0   
## Mode :character Mode :character Mode :character Median :110.0   
## Mean :106.9   
## 3rd Qu.:110.0   
## Max. :160.0   
##   
## protein fat sodium fiber   
## Min. :1.000 Min. :0.000 Min. : 0.0 Min. : 0.000   
## 1st Qu.:2.000 1st Qu.:0.000 1st Qu.:130.0 1st Qu.: 1.000   
## Median :3.000 Median :1.000 Median :180.0 Median : 2.000   
## Mean :2.545 Mean :1.013 Mean :159.7 Mean : 2.152   
## 3rd Qu.:3.000 3rd Qu.:2.000 3rd Qu.:210.0 3rd Qu.: 3.000   
## Max. :6.000 Max. :5.000 Max. :320.0 Max. :14.000   
##   
## carbo sugars potass vitamins   
## Min. : 5.0 Min. : 0.000 Min. : 15.00 Min. : 0.00   
## 1st Qu.:12.0 1st Qu.: 3.000 1st Qu.: 42.50 1st Qu.: 25.00   
## Median :14.5 Median : 7.000 Median : 90.00 Median : 25.00   
## Mean :14.8 Mean : 7.026 Mean : 98.67 Mean : 28.25   
## 3rd Qu.:17.0 3rd Qu.:11.000 3rd Qu.:120.00 3rd Qu.: 25.00   
## Max. :23.0 Max. :15.000 Max. :330.00 Max. :100.00   
## NA's :1 NA's :1 NA's :2   
## shelf weight cups rating   
## Min. :1.000 Min. :0.50 Min. :0.250 Min. :18.04   
## 1st Qu.:1.000 1st Qu.:1.00 1st Qu.:0.670 1st Qu.:33.17   
## Median :2.000 Median :1.00 Median :0.750 Median :40.40   
## Mean :2.208 Mean :1.03 Mean :0.821 Mean :42.67   
## 3rd Qu.:3.000 3rd Qu.:1.00 3rd Qu.:1.000 3rd Qu.:50.83   
## Max. :3.000 Max. :1.50 Max. :1.500 Max. :93.70   
##

Cereals<-Cereals[,-c(2,3,13)]

##### 2.2. Normalization of the Cereal dataset

Cereals\_Norm<- sapply(Cereals[,-1], scale)

##### 2.3. Removing NA values from the normalised dataset

Cereals\_Norm<-na.omit(Cereals\_Norm)

#### 3.Calculating the distance using the Euclidean method

d <- dist(Cereals\_Norm, method = "euclidean")

#### 4. Agglomerative Coefficient for each linkage method

Agglomerative coefficient is a measure of the clustering structure of the dataset.Closer to 1 leads to stronger structure.

The coefficient values are basically the mean value of the normalized lengths at which the clusters are formed.

Here, we will use Agglomerative coefficient on “Average”,“Singles”,“Complete” and “ward” methods.

# Using Agnes to compare different methods  
  
# methods to assess  
m <- c( "average", "single", "complete", "ward")  
names(m) <- c( "average", "single", "complete", "ward")  
  
# function to compute coefficient  
ac <- function(x) {  
 agnes(Cereals\_Norm, method = x)$ac  
}  
  
# get agglomerative coefficient for each linkage method  
purrr::map\_dbl(m, ac)

## average single complete ward   
## 0.7888569 0.6091225 0.8508357 0.9088247

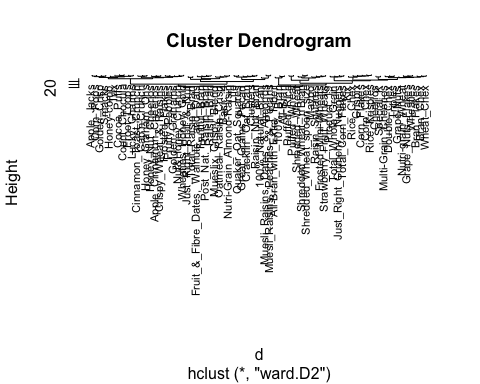
We can see that “ward method” has the highest Agglomerative coefficient. The coefficient value for the ward method is 0.904 , which is relatively closer to 1 which means that the cluster structure is good.

Therefore, ward method is the best fit for it, as it is giving the strongest cluster structure.

Since, we have now finalized ward method , we can apply hierarchical clustering to the Euclidean distance to the normalized measurements:-

#### 5.Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements.

#Applying hierarchical Clustering to the using Euclidean distance to the normalized measurements  
  
hc\_ward<- hclust(d,method="ward.D2")  
  
  
#Creating dendrogram for ward method  
  
plot(hc\_ward,labels=Cereals\_N$name,cex=.7) #ward



#### 6.Comment on differences between hierarchical Clustering and K-means.

K Means clustering needed an advance knowledge of no. of clusters we want to divide our dataset. In hierarchical clustering we can stop at any number of clusters, we find appropriate by interpreting the dendrogram which is a tree like structure.

Hierarchical clustering can’t handle big data well but K Means clustering can handle large dataset quite efficiently.When the dataset is small Hierarchical clustering can be really helpful in understanding the data as we can see the structure and distances in the dendrogram.

#### 

#### 7.How many clusters would you choose?

Lets look at the dendrogram using ward method with height=10.

If we cut the tree closer to the top, we get fewer clusters, and if we cut the tree closer to the bottom, we get more clusters. The y-axis represents the distances between cases. Horizontal lines indicate the positions at which cases/clusters merge with each other. Looking top to bottom we see that the split to 2 clusters causes a small drop in error and the biggest splits occurs at 4 and 6 .

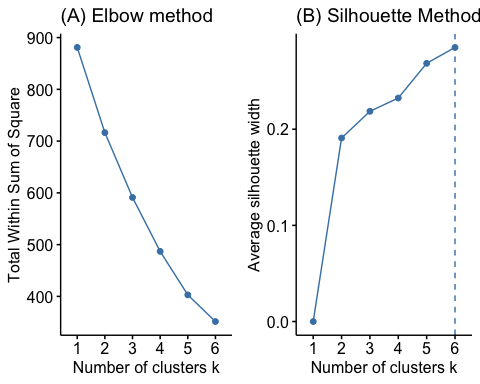
At height 17 , there will be 2 clusters and at height 13 it will be 4 clusters , when the height is 10 it is 6 clusters and similarly we can get more numbers of clusters if we go down in depth.

I want clusters to be interpretable and actionable. This in a way limits number of clusters. Too many clusters will lead to losing the ability to interpret the clusters . Less number of cluster might risk towards generalizing , creating a simplistic treatment and missing the opportunity for a more tailored and effective approach.Also , later we would need to determine a cluster with a specific requirement.

Therefore , I would like to keep the no. of optimal cluster within the range of 4-6.

Let’s use the elbow method and silhouette method to get some insight:

#Creating variable p1 ,p2 to store fviz\_nbclust() output for different methods:  
  
 p1 <- fviz\_nbclust(Cereals\_Norm, FUN = hcut, method = "wss",   
 k.max = 6) +  
 ggtitle("(A) Elbow method")  
 p2 <- fviz\_nbclust(Cereals\_Norm, FUN = hcut, method = "silhouette",   
 k.max = 6) +  
 ggtitle("(B) Silhouette Method")  
   
   
 #Plotting based on Silhouette Method and ELbow method :  
 gridExtra::grid.arrange(p1, p2, nrow = 1)



After applying number of approaches and getting different results,I have realized that with different approaches and methods we are most likely to get different results.However , the most popular measure of cluster goodness is called the silhouette and Dunn’s Index.

Therefore,I have decided to use these 2 indexes i.e silhouette,dunn Index by nbclust() to determine optimal numbers of clusters.

The Silhouette Width and the Dunn Index combine measures of compactness and separation of the clusters.

* The Silhouette Width is the average of each observation’s Silhouette value. The Silhouette value measures the degree of confidence in a particular clustering assignment and lies in the interval [-1,1], with well-clustered observations having values near 1 and poorly clustered observations having values near -1.
* Dunn’s index is the ratio between the minimum inter-cluster distances to the maximum intra-cluster diameter. The diameter of a cluster is the distance between its two furthermost points. In order to have well separated and compact clusters you should aim for a higher Dunn’s index.It has a value between 0 and infinity and should be maximized.

## Dunn's Index  
res\_dunn<-NbClust(data= Cereals\_Norm, distance = "euclidean", min.nc=4, max.nc=6,   
 method = "ward.D2", index = "dunn")  
  
res\_dunn$All.index

## 4 5 6   
## 0.1852 0.1907 0.2464

res\_dunn$Best.nc

## Number\_clusters Value\_Index   
## 6.0000 0.2464

## Silhouette Value  
res\_silhouette <- NbClust(data= Cereals\_Norm, distance = "euclidean", min.nc=4, max.nc=6,   
 method = "ward.D2", index = "silhouette")  
  
res\_silhouette$All.index

## 4 5 6   
## 0.2324 0.2685 0.2850

res\_silhouette$Best.nc

## Number\_clusters Value\_Index   
## 6.000 0.285

Answer:

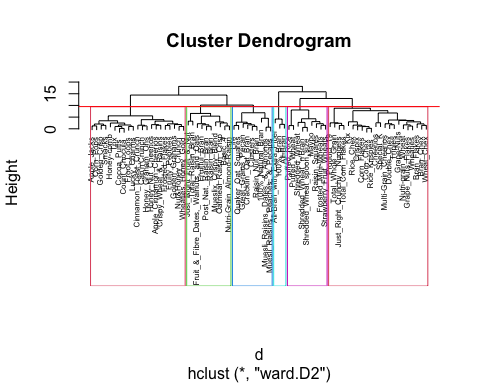
Based on the above output : The optimal number of clusters within the range of 4:8 will be 6.Therefore k=6

Let’s plot the dendrogram with 6 clusters and height = 9.5 .

#cutting   
set.seed(700)  
#Cutting the dendrogram in clusters  
hc\_cut<-cutree(hc\_ward,k=6)  
summary(as.factor(hc\_cut))

## 1 2 3 4 5 6   
## 3 9 21 10 22 9

#plotting Dendrogram  
plot(hc\_ward,labels=Cereals\_N$name,cex=.5) #plotting hierarchical cluster labels=Cereals\_N$name,  
rect.hclust(hc\_ward , k = 6, border = 2:6) # Categorizing dendrogram into 6 clusters  
abline(h = 9.5, col = 'red') # plotting a line with height = 10



##### 8.Understanding of Hierarchical structure of the Cereal dataset:-

Hierarchy of clusters, which is often represented graphically by means of a tree structure called a dendrogram. Clusters of increasing similarity are merged to form larger ones (agglomerative algorithms) or that larger clusters are split into smaller ones of decreasing dissimilarities (divisive procedure).

Such an approach has the advantage of allowing exploration of the data at different levels of similarity/dissimilarity and providing a deeper insight into the relationships among sample objects.

#### Clusters :-

Cluster1:Count= 3, Main Ingredients- All Bran and Extra fiber

Cluster2:Count= 9, Main Ingredients- Raisin, Oatmeal, muesli

Cluster3:Count= 21,Main Ingredients- Apple,Honey,Cinnamon,Raisin

Cluster4:Count= 10,Main Ingredients- Muesli Raisins,Dates and Nuts

Cluster5:Count= 22, Main Ingredients- Rice ,Corn flakes, Chex and Flakes

Cluster6:Count= 9,Main Ingredients- Wheat

#### 9.Comment on the structure of the clusters and on their stability. Hint: To check stability,partition the data and see how well clusters formed based on one part apply to the other part.

#### Assess how consistent the cluster assignments are compared to the assignments based on all the data.

Website used for reference : <http://activisiongamescience.github.io/2016/08/19/Assessing-Stability-of-K-Means-Clusterings/>

* When a new object in inserted/removed from a existing dendrogram it may cause that some two existing partitions will get closer or more distant.
* If there is not much changes in the structure of the dendrogram it is considered to be stable.

##### Steps followed:

##### 9.1.Partionng the data

##### 9.2.Normalization

##### 9.3.Calculating distance using euclidean method

set.seed(666) # to generate the same result  
  
#Data Partitioning :  
# random 80% of data been partitioned from the original data  
  
Index\_Train<-createDataPartition(Cereals$rating, p=0.8, list=FALSE)  
Train\_Data <-Cereals[Index\_Train,]  
Train\_Data\_NN<- na.omit(Train\_Data)  
nrow(Train\_Data)

## [1] 64

#normalization of the partitioned data  
Cereals\_No<- sapply(Train\_Data[,-1], scale)  
  
#removing NA value  
Cereals\_No<- na.omit(Cereals\_No)  
  
#Calculation of distance  
do <- dist(Cereals\_No, method = "euclidean")

##### 9.4.Determining Agglomeretive coefficient:-

# Using Agnes to compare different methods  
  
# methods to assess  
m <- c( "average", "single", "complete", "ward")  
names(m) <- c( "average", "single", "complete", "ward")  
  
# function to compute coefficient  
ac <- function(x) {  
 agnes(Cereals\_No, method = x)$ac  
}  
  
# get agglomerative coefficient for each linkage method  
purrr::map\_dbl(m, ac)

## average single complete ward   
## 0.7644000 0.5751779 0.8301407 0.8923850

ag<-agnes(Cereals\_No, metric = "euclidean", method = "ward")  
ag$ac #0.9620709 will not work unless agnes () is used to determine HC

## [1] 0.892385

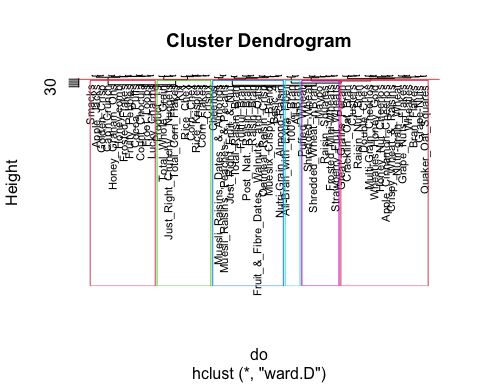
Again “ward method” has the highest Agglomerative coefficient. The coefficient value for the ward method is 0.8923850 , which is lesser than the value with original data. One of the reason is due to the reduction in the data size .

##### 9.5.Hierarcical Clustering and plot

# Determining the hierarchical clustering  
hc\_ward2<- hclust(do,method="ward")

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

#Plot generation of hierarchical clustering  
plot(hc\_ward2,labels=Train\_Data\_NN$name,cex=.7)  
rect.hclust(hc\_ward2 , k = 6, border = 2:6)  
abline(h = 10, col = 'red')



hc\_cut2<-cutree(hc\_ward2,k=6)  
summary(as.factor(hc\_cut2)) #Counting the number of objects in each cluster

## 1 2 3 4 5 6   
## 3 16 12 13 10 7

##### Clusters:-

Cluster1: Count = 3, Main Ingredients- All Bran and Extra fiber

Cluster2: Count = 7, Main Ingredients- Muesli Raisins,Dates and Nuts

Cluster3:Count = 17, Main Ingredients- Apple,fruits

Cluster4:Count = 9, Main Ingredients- Wheat

Cluster5:Count = 16, Main Ingredients- Rice ,Corn flakes, Chex and Flakes

Cluster6:Count = 9, Main Ingredients- Raisin, Oatmeal, meusli multigrain/wheat apple raisin cheerios

Since , we have removed 20 percent of data from the original, we can see that some of cereals are not present in the clusters. Overall on highlevel, I can see some changes in the clusters formed after the removal of slice of data.

We can understand the changes better by doing visual comparison of the dendrogram and later by determing the value for stability of clusters formed in the dendrogram:-

##### Steps followed:-

1. Creating variables for original data and partitioned data
2. Creating a list to hold dendrograms
3. Analyzing the changes occurs in the relation of the variables before /after removing small set of data from the original
4. Comparing the structure dendrograms and Interpretations for stability
5. Calculating Stability

##### 9.6.Visual Comparison of the dendrogram :-

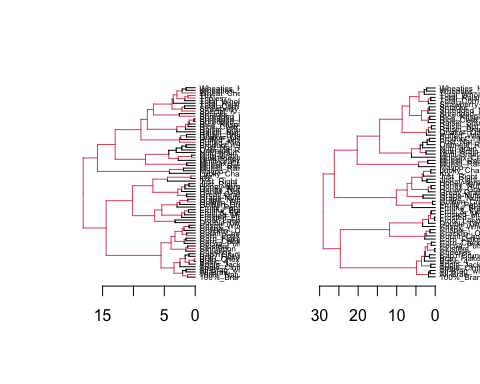
Website used for reference: <http://activisiongamescience.github.io/2016/08/19/Assessing-Stability-of-K-Means-Clusterings/>

#Creating Dendrogram  
dend1 <- as.dendrogram (hc\_ward) # original data hierarchical clustering  
dend2 <- as.dendrogram (hc\_ward2) #partitioned data hierarchical clustering  
  
#Adding labels to the dendrogram  
labels(dend1)<-Cereals\_N$name #full dataset  
labels(dend2)<-Train\_Data\_NN$name # partial dataset  
  
# Adjusting the size of the labels  
labels\_cex(dend1)<-0.5   
  
labels\_cex(dend2)<-0.5  
  
# Create a list to hold dendrograms  
dend\_list <- dendlist(dend2, dend1)

##### 6.1 Analyzing the difference in the Dendrogram branches:

The unique nodes can be explained in more detailed by dend\_diff().

#Plotting trees side by side and highlighting the unique to each tree in red   
 dend\_diff(dend1, dend2)



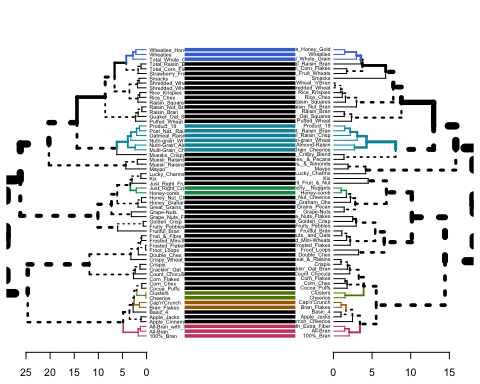
The dend\_diff function plots two trees side by side, highlighting edges unique to each tree in red. We can see lot Unique nodes in the dendrogram above . Here, the difference in data is not pruned .

Left side Dendrogram is based on the original data and right side is based on the partitioned data.We can see that there is lot of unique branches

##### 6.2 Using tanglegram() to compare the Dendrogram side by side

# Align and plot two dendrograms side by side  
 tanglegram(dend\_list,fast=FALSE,highlight\_distinct\_edges = TRUE, # Turn-off dashed lines  
 common\_subtrees\_color\_lines = TRUE, # Turn-on line colors  
 common\_subtrees\_color\_branches = TRUE) # Color common branches )

## Warning in intersect\_trees(dend1, dend2, warn = TRUE): The labels in both tree  
## had different values - trees were pruned.



#### Interpretation of the comparison:-

Since ,we have only used 80 percent of the data,to compare the 2 dendrograms additional data from the original were pruned for comparison.

Note:- We see the horizontal connecting line between the two dendrograms but, however these trees are not identical (or even very similar topologically).

#### Observations:-

1.We can see that there is movement in the positions of the clusters after removing some data from the original.

2.We can some common connecting lines in colors(Blue,green,orange,pink) , which means there are sub-trees which are present in both dendrograms.So,some of the pairs remained same as before. For example : At the bottom we can see pink color horizontal lines for " All-Bran, 100% Bran " items which means they are present in both dendrograms and shows good similarity.

3.“Unique” nodes, with a combination of items not present in the other tree, are highlighted with dashed lines. For example: We can see at the root there are dashed lines which means the combination of items into those branches are unique to each other.

4.The colored branches of the trees to show the two common sub-trees using common\_subtrees\_color\_branches = TRUE. For example: At the bottom we can see pink color lines for " All-Bran, 100% Bran " items. It means their subtree branch is present in both dendrograms.

#### Result for the visual comparison :

With the help of different visual comparison of dendrograms we can say that, the clusters are not highly stable as the correlation of the the variables changes with removal of data.There is some similarity and dissimilarity in the pattern of cluster segmentation which can be seen in the dendrograms.

We can also quantify the result by using clusterboot() on the partitioned data.

##### 6.3 Cluster Stability in calculation

Web site used for reference : <https://www.rdocumentation.org/packages/fpc/versions/2.2-8/topics/clusterboot>

set.seed(11)  
#Input the scaled cereals\_data  
hclust\_stability = clusterboot(Cereals\_No, clustermethod=hclustCBI, method="ward.D2", k=6, count = FALSE)  
hclust\_stability

## \* Cluster stability assessment \*  
## Cluster method: hclust/cutree   
## Full clustering results are given as parameter result  
## of the clusterboot object, which also provides further statistics  
## of the resampling results.  
## Number of resampling runs: 100   
##   
## Number of clusters found in data: 6   
##   
## Clusterwise Jaccard bootstrap (omitting multiple points) mean:  
## [1] 0.9500000 0.9045813 0.7385558 0.6663966 0.4946545 0.7205951  
## dissolved:  
## [1] 5 0 20 36 67 25  
## recovered:  
## [1] 95 93 51 38 21 45

#### Interpretation of the clustering stability results:-

Stability guidelines :-

1.Values > 0.85 =""Highly Stable

2.Values >0.75 =“Stable”

3.Values Between 0.6 and 0.75 = “show some pattern towards stability but needs some more investigation”

4.Below 0.6=“should not be trusted”

* Values > 0.85 denote very stable clusters. Here ,we can see that when the number of cluster =6 , the stability comes 0.72 . which is lower than 0.85. the clusters show some patterns however it needs to be investigated further.
* Clusters were dissolved 25 times out of 100 re-sampling runs, the least the count of dissolved cluster better comes the stability. Here ,the stability is not good enough however it surely shows some patterns, which can be improved.

#### 9. The elementary public schools would like to choose a set of cereals to include in their daily cafeterias. Every day a different cereal is offered, but all cereals should support a healthy diet. For this goal, you are requested to find a cluster of “healthy cereals.”

Websites used for determining “Healthy Cereals” reference :-

<https://www.realmomnutrition.com/how-to-choose-a-healthy-cereal/> <https://www.bestkidstuff.com/food/healthiest-cereal-for-kids/>

Based on the above websites I have considered 5 nutrition elements in cereals dataset .Please find them below in the order of importance:-

1.Sugars(lesser the better)

2.Fat(lower the better)

3.Fiber(higher the better)

4.Protein (higher the better)

5.Calories(lesser the better)

6.Vitamins and minerals(higher the better)

Note: I have not considered cups and weight variable as It will make analysis complex.

#### Should the data be normalized? If not, how should they be used in the cluster analysis?

We don’t need to normalize the data for the analysis as we are determining the values for each feature separately and we do not need to calculate the distance between them.

Also, Scaling will lead to mean of “0” and standard deviation of “1”. I am using mean value for each cluster feature to determine the ‘healthy Cereal’ cluster.Therefore , using normalized values will not help .

# Determining the mean value for each category important for categorizing snacks under "Healthy "  
tapply(Cereals\_N$sugars, hc\_cut, mean) # lower the mean value of Sugar higher for health ##Cluster 6

## 1 2 3 4 5 6   
## 3.666667 7.555556 11.476190 10.900000 3.454545 2.333333

tapply(Cereals\_N$fat, hc\_cut, mean) # least the better ##cluster 6

## 1 2 3 4 5 6   
## 0.6666667 2.6666667 1.0000000 1.4000000 0.5454545 0.1111111

tapply(Cereals\_N$fiber, hc\_cut, mean) # higher in fiber better for health #cluster 1

## 1 2 3 4 5 6   
## 11.0000000 2.6111111 0.5714286 3.6500000 1.6818182 2.1111111

tapply(Cereals\_N$protein, hc\_cut, mean) # higher the mean value of Protein better for health ##cluster 1

## 1 2 3 4 5 6   
## 4.000000 3.444444 1.523810 3.000000 2.681818 2.444444

tapply(Cereals\_N$calories, hc\_cut, mean) # lesser in number good for health ##cluster 1

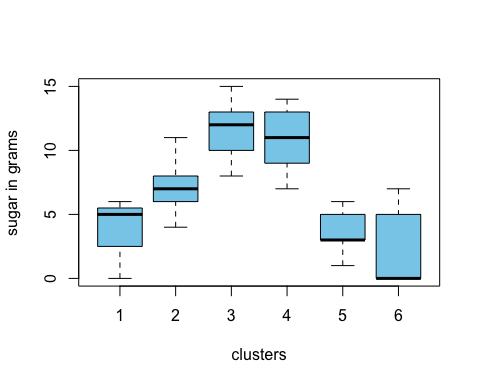
## 1 2 3 4 5 6   
## 63.33333 117.77778 110.95238 132.00000 103.63636 82.22222

tapply(Cereals\_N$vitamins, hc\_cut, mean) #higher in vitamins, good for health ##cluster 4

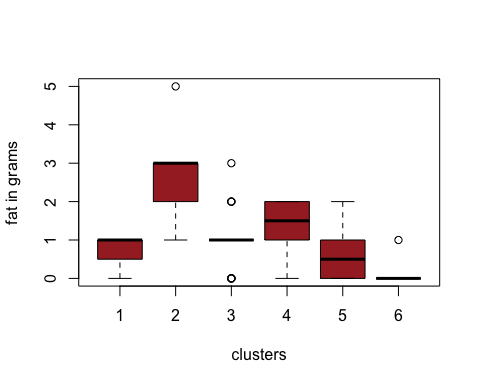
## 1 2 3 4 5 6   
## 25.00000 22.22222 25.00000 40.00000 38.63636 11.11111

#### Using Boxplots to visualize the analysis :-

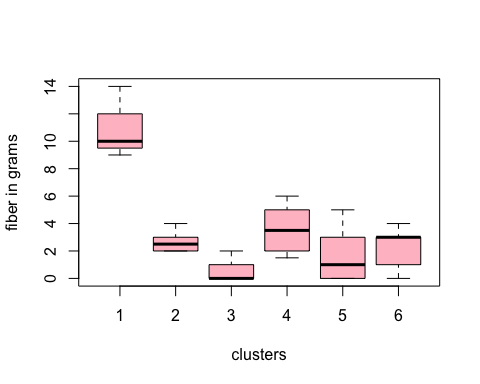
#Visualization of understanding the features of clusters based on the requirement:-  
  
boxplot(Cereals\_N$sugars ~ hc\_cut,xlab="clusters",ylab="sugar in grams",col="sky blue")



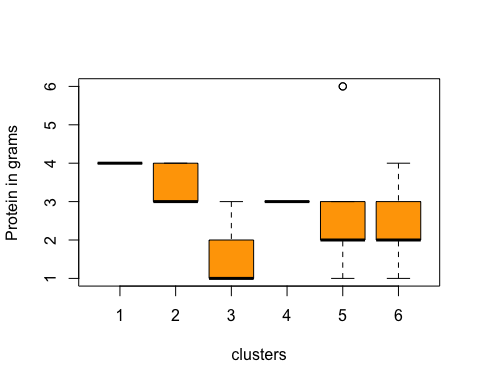
boxplot(Cereals\_N$fat ~ hc\_cut,xlab="clusters",ylab="fat in grams",col=" brown")



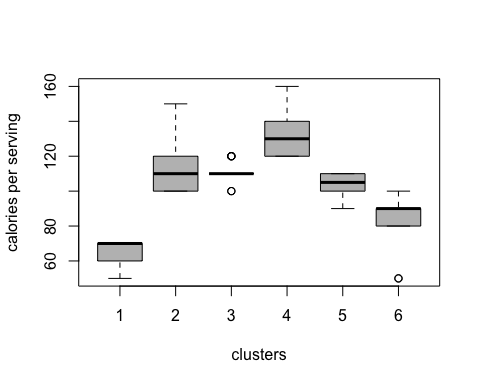
boxplot(Cereals\_N$fiber ~ hc\_cut,xlab="clusters",ylab="fiber in grams",col="pink")



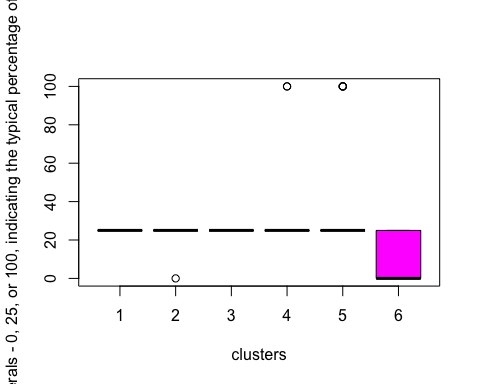
boxplot(Cereals\_N$protein ~ hc\_cut,xlab="clusters",ylab="Protein in grams",col="orange")



boxplot(Cereals\_N$calories ~ hc\_cut,xlab="clusters",ylab="calories per serving",col="grey")



boxplot(Cereals\_N$vitamins ~ hc\_cut,xlab="clusters",ylab="vitamins and minerals - 0, 25, or 100, indicating the typical percentage of FDA recommended",col="magenta")



As per the results,We can see that cluster 1 is lowest in calories, highest in Fiber,Protein and second lowest in sugar content. Therefore ,We can conclude that Cluster 1 seems to be the “healthy Cluster”.

#### Determining the Healthy Cereals cluster :-

#Creating subset with dataset in cluster 1(Healthy Snacks cluster)  
cluster1<-subset(Cereals\_N,hc\_cut==1)  
  
#Cluster 1 cereals names:-  
cluster1$name

## [1] "100%\_Bran" "All-Bran"   
## [3] "All-Bran\_with\_Extra\_Fiber"

#### 

#### 10. How do you compare hierarchical clustering and k-means? What are they main advantages of hierarchical clustering compared to k-means?

Answer:-

##### Comparing the Hierarchical Clustering and k-means:-

1.Directional approach:-

In K means only centroid is considered to form clusters,

Hierarchical Clustering :Top-down, bottom-up approaches are followed

1. Understanding of data :-

K means: the understanding of data is not that clear however,

Hierarchical Clustering : the method shows all the possible linkages between the clusters so we we understand the data much better.

3.Value of K:-

In K means: we need preset the value of k (no. of clusters)

Hierarchical clustering :No need to preset the number of clusters

4.Results:-

K means: the result of running the algorithm multiple times may lead to different results ,

Hierarchical clustering: however in Hierarchical clustering the results remain the same with multiple runs.

5.Knowledge of K:-

K Means : clustering needed advance knowledge of K i.e. no. of clusters one want to divide your data.

Hierarchical Clustering : one can stop at any number of clusters, one find appropriate by interpreting the dendrogram.

6: Volume of Data :

K means: works well with high Volume of data and therefore less computational cost

Hierarchical Clustering: doesn’t work well with high volume of data as compare to K means , as it becomes difficult to understand the data with huge volume

##### Advantages of Hierarchical Clustering over K means:-

1.The main benefit of hierarchical clustering over k-means, is that we get a much finer understanding of the structure of our data, and it is often able to reconstruct real hierarchies in nature.

1. No Apriori information about number of clusters is required and it is easy to implement.