ML CRISCA Customer Segmenation

Obianuju Anumnu

12/14/2020

```
library(class)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.0.2
```

```
library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
library(FNN)
```

```
## Warning: package 'FNN' was built under R version 4.0.2
```

```
##
## Attaching package: 'FNN'
```

```
## The following objects are masked from 'package:class':
##
##     knn, knn.cv
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggvis)
```

```
##
## Attaching package: 'ggvis'
```

```
## The following object is masked from 'package:ggplot2':
##
##     resolution
```

```r
library(ggplot2)
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.2
```

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.1     v purrr   0.3.4
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'readr' was built under R version 4.0.2
```

```
## Warning: package 'forcats' was built under R version 4.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```r
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.0.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(flexclust)
```

```
## Warning: package 'flexclust' was built under R version 4.0.3
```

```
## Loading required package: grid
```

```
## Loading required package: modeltools
```

```
## Warning: package 'modeltools' was built under R version 4.0.3
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'flexclust'
```

```
## The following object is masked from 'package:e1071':
##
##     bclust
```

```
library(imputeTS)
```

```
## Warning: package 'imputeTS' was built under R version 4.0.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(stats)
```

```
#Read Data
bathsoap<- read.csv("C:\\Users\\Obianuju\\OneDrive\\D\\Machine\\BathSoap.csv")
head(bathsoap)
```

```
##   Member.id SEC FEH MT SEX AGE EDU HS CHILD CS Affluence.Index No..of.Brands
## 1   1010010   4   3 10   1   4   4  2     4  1               2             3
## 2   1010020   3   2 10   2   2   4  4     2  1              19             5
## 3   1014020   2   3 10   2   4   5  6     4  1              23             5
## 4   1014030   4   0  0   0   4   0  0     5  0               0             2
## 5   1014190   4   1 10   2   3   4  4     3  1              10             3
## 6   1017020   4   3 10   2   3   4  5     2  1              13             3
##   Brand.Runs Total.Volume No..of..Trans  Value Trans...Brand.Runs Vol.Tran
## 1         17         8025            24  818.0               1.41   334.38
## 2         25        13975            40 1681.5               1.60   349.38
## 3         37        23100            63 1950.0               1.70   366.67
## 4          4         1500             4  114.0               1.00   375.00
## 5          6         8300            13  591.0               2.17   638.46
## 6         26        18175            41 1705.5               1.58   443.29
##   Avg..Price Pur.Vol.No.Promo.... Pur.Vol.Promo.6.. Pur.Vol.Other.Promo..
## 1      10.19                100%                0%                    0%
## 2      12.03                 89%               10%                    2%
## 3       8.44                 94%                2%                    4%
## 4       7.60                100%                0%                    0%
## 5       7.12                 61%               14%                   24%
## 6       9.38                100%                0%                    0%
##   Br..Cd..57..144 Br..Cd..55 Br..Cd..272 Br..Cd..286 Br..Cd..24 Br..Cd..481
## 1             38%        13%          0%          0%         0%          0%
## 2              2%         8%          0%          0%         0%          6%
## 3              3%        55%          0%          3%         0%          0%
## 4             40%        60%          0%          0%         0%          0%
```

```
## 5              5%        14%          0%          0%         0%          0%
## 6              8%         7%          0%          0%         0%          0%
##   Br..Cd..352 Br..Cd..5 Others.999 Pr.Cat.1 Pr.Cat.2 Pr.Cat.3 Pr.Cat.4
## 1          0%        0%     49.20%      23%      56%      13%       7%
## 2          0%       14%     69.90%      29%      55%       9%       6%
## 3          0%        2%     37.90%      12%      32%      56%       0%
## 4          0%        0%      0.00%       0%      40%      60%       0%
## 5          0%        0%     80.70%       0%       5%      14%      81%
## 6          0%        0%     85.70%      22%      45%       7%      27%
##   PropCat.5 PropCat.6 PropCat.7 PropCat.8 PropCat.9 PropCat.10 PropCat.11
## 1       50%        0%        0%        0%        0%         0%         0%
## 2       46%       35%        3%        2%        1%         0%         6%
## 3       24%       12%        3%        1%        1%         0%         0%
## 4       40%        0%        0%        0%        0%         0%         0%
## 5       81%        0%        0%        5%        0%         0%         0%
## 6       49%       10%        0%        1%        7%         0%         0%
##   PropCat.12 PropCat.13 PropCat.14 PropCat.15
## 1         3%         0%        13%        34%
## 2         0%         0%         8%         0%
## 3         2%         0%        56%         0%
## 4         0%         0%        60%         0%
## 5         0%         0%        14%         0%
## 6         0%         0%         7%        27%
```

```
#check for missing data
colMeans(is.na(bathsoap))
```

```
##             Member.id                   SEC                   FEH
##                     0                     0                     0
##                    MT                   SEX                   AGE
##                     0                     0                     0
##                   EDU                    HS                 CHILD
##                     0                     0                     0
##                    CS       Affluence.Index         No..of.Brands
##                     0                     0                     0
##            Brand.Runs          Total.Volume         No..of..Trans
##                     0                     0                     0
##                 Value    Trans...Brand.Runs              Vol.Tran
##                     0                     0                     0
##            Avg..Price  Pur.Vol.No.Promo....     Pur.Vol.Promo.6..
##                     0                     0                     0
## Pur.Vol.Other.Promo..       Br..Cd..57..144           Br..Cd..55
##                     0                     0                     0
##           Br..Cd..272           Br..Cd..286           Br..Cd..24
##                     0                     0                     0
##           Br..Cd..481           Br..Cd..352            Br..Cd..5
##                     0                     0                     0
##            Others.999             Pr.Cat.1              Pr.Cat.2
##                     0                     0                     0
##              Pr.Cat.3              Pr.Cat.4             PropCat.5
##                     0                     0                     0
##             PropCat.6             PropCat.7             PropCat.8
##                     0                     0                     0
##             PropCat.9            PropCat.10            PropCat.11
```

```
##                   0                   0                   0
##         PropCat.12          PropCat.13          PropCat.14
##                   0                   0                   0
##         PropCat.15
##                   0
```

```
#no missing data
```

Data Pre-processing

```
#add row names
row.names(bathsoap)<- bathsoap[,1]
#remove the Member_id column
bathsoap1<-bathsoap[,-1]
#convert percentages to numbers.
bathsoap_num <- bathsoap1[19:45] %>% mutate_each(funs(as.numeric(gsub("%", "", ., fixed = TRUE))/100))
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

## Warning: `mutate_each_()` is deprecated as of dplyr 0.7.0.
## Please use `across()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
bathsoap_total<-cbind(bathsoap1[1:18],bathsoap_num)
```

effect of using percentages of total purchases comprised by various brands

#Percentage of total purchase is an indication of brand loyalty, the data as it is shows multiple volumes of brand purchases and this may confuse the model. It is better to have a derived variable from "Purchase volume and percentage of volume purchased of the brand"

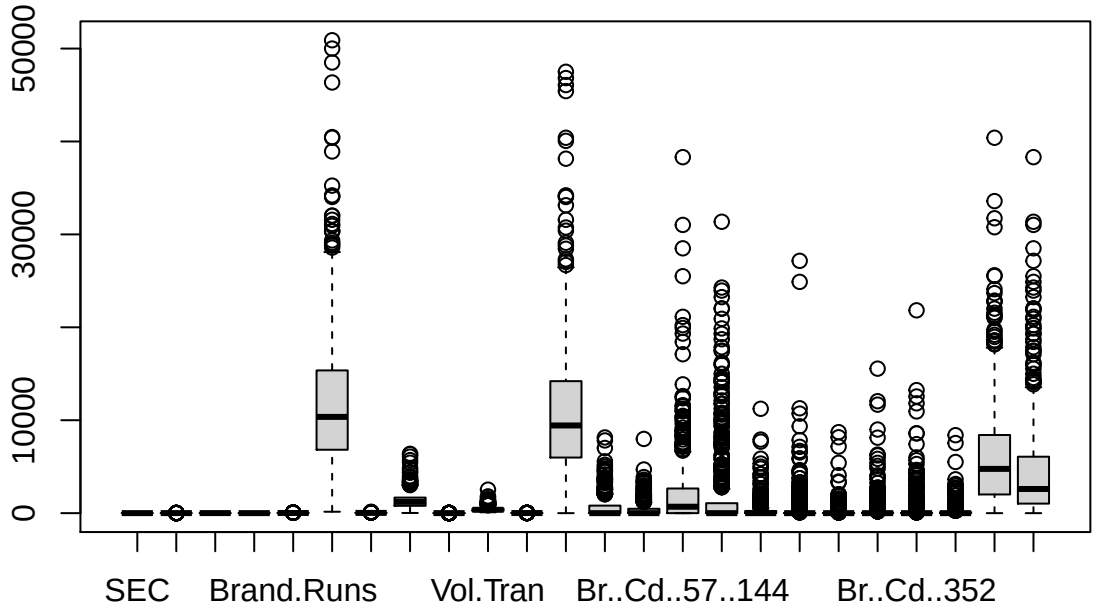
#Clustering - Purchase Behaviour

```r
# Subsetting purchase behavior variables
str(bathsoap_total)
```

```
## 'data.frame':    600 obs. of  45 variables:
##  $ SEC                  : int  4 3 2 4 4 4 4 4 4 1 ...
##  $ FEH                  : int  3 2 3 0 1 3 2 3 3 3 ...
##  $ MT                   : int  10 10 10 0 10 10 10 10 10 5 ...
##  $ SEX                  : int  1 2 2 0 2 2 2 2 2 1 ...
##  $ AGE                  : int  4 2 4 4 3 3 4 2 4 4 ...
##  $ EDU                  : int  4 4 5 0 4 4 1 4 4 7 ...
##  $ HS                   : int  2 4 6 0 4 5 3 5 6 3 ...
##  $ CHILD                : int  4 2 4 5 3 2 2 3 4 4 ...
##  $ CS                   : int  1 1 1 0 1 1 1 0 1 1 ...
##  $ Affluence.Index      : int  2 19 23 0 10 13 11 0 17 6 ...
##  $ No..of.Brands        : int  3 5 5 2 3 3 4 3 2 4 ...
##  $ Brand.Runs           : int  17 25 37 4 6 26 17 8 12 13 ...
##  $ Total.Volume         : int  8025 13975 23100 1500 8300 18175 9950 9300 26490 7455 ...
##  $ No..of..Trans        : int  24 40 63 4 13 41 26 25 27 18 ...
##  $ Value                : num  818 1682 1950 114 591 ...
##  $ Trans...Brand.Runs   : num  1.41 1.6 1.7 1 2.17 1.58 1.53 3.13 2.25 1.38 ...
##  $ Vol.Tran             : num  334 349 367 375 638 ...
##  $ Avg..Price           : num  10.19 12.03 8.44 7.6 7.12 ...
##  $ Pur.Vol.No.Promo.... : num  1 0.89 0.94 1 0.61 1 0.98 0.94 0.9 1 ...
##  $ Pur.Vol.Promo.6..    : num  0 0.1 0.02 0 0.14 0 0.02 0 0.1 0 ...
##  $ Pur.Vol.Other.Promo..: num  0 0.02 0.04 0 0.24 0 0 0.06 0 0 ...
##  $ Br..Cd..57..144      : num  0.38 0.02 0.03 0.4 0.05 0.08 0.45 0.04 0.39 0.07 ...
##  $ Br..Cd..55           : num  0.13 0.08 0.55 0.6 0.14 0.07 0.05 0.79 0 0.12 ...
##  $ Br..Cd..272          : num  0 0 0 0 0 0 0.01 0 0 0 ...
##  $ Br..Cd..286          : num  0 0 0.03 0 0 0 0 0 0 0 ...
##  $ Br..Cd..24           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Br..Cd..481          : num  0 0.06 0 0 0 0 0 0 0 0 ...
##  $ Br..Cd..352          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Br..Cd..5            : num  0 0.14 0.02 0 0 0 0 0 0 0.4 ...
##  $ Others.999           : num  0.492 0.699 0.379 0 0.807 0.857 0.495 0.167 0.615 0.41 ...
##  $ Pr.Cat.1             : num  0.23 0.29 0.12 0 0 0.22 0.07 0.04 0.11 0.61 ...
##  $ Pr.Cat.2             : num  0.56 0.55 0.32 0.4 0.05 0.45 0.66 0.04 0.89 0.1 ...
##  $ Pr.Cat.3             : num  0.13 0.09 0.56 0.6 0.14 0.07 0.05 0.9 0 0.12 ...
##  $ Pr.Cat.4             : num  0.07 0.06 0 0 0.81 0.27 0.23 0.02 0 0.17 ...
##  $ PropCat.5            : num  0.5 0.46 0.24 0.4 0.81 0.49 0.82 0.06 0.7 0.24 ...
##  $ PropCat.6            : num  0 0.35 0.12 0 0 0.1 0 0 0.28 0.46 ...
##  $ PropCat.7            : num  0 0.03 0.03 0 0 0 0.02 0 0 0.15 ...
##  $ PropCat.8            : num  0 0.02 0.01 0 0.05 0.01 0.01 0 0 0 ...
##  $ PropCat.9            : num  0 0.01 0.01 0 0 0.07 0 0 0.02 0 ...
##  $ PropCat.10           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PropCat.11           : num  0 0.06 0 0 0 0 0 0 0 0 ...
##  $ PropCat.12           : num  0.03 0 0.02 0 0 0 0 0.01 0 0 ...
##  $ PropCat.13           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PropCat.14           : num  0.13 0.08 0.56 0.6 0.14 0.07 0.05 0.9 0 0.12 ...
##  $ PropCat.15           : num  0.34 0 0 0 0 0.27 0.1 0.03 0 0.03 ...
```

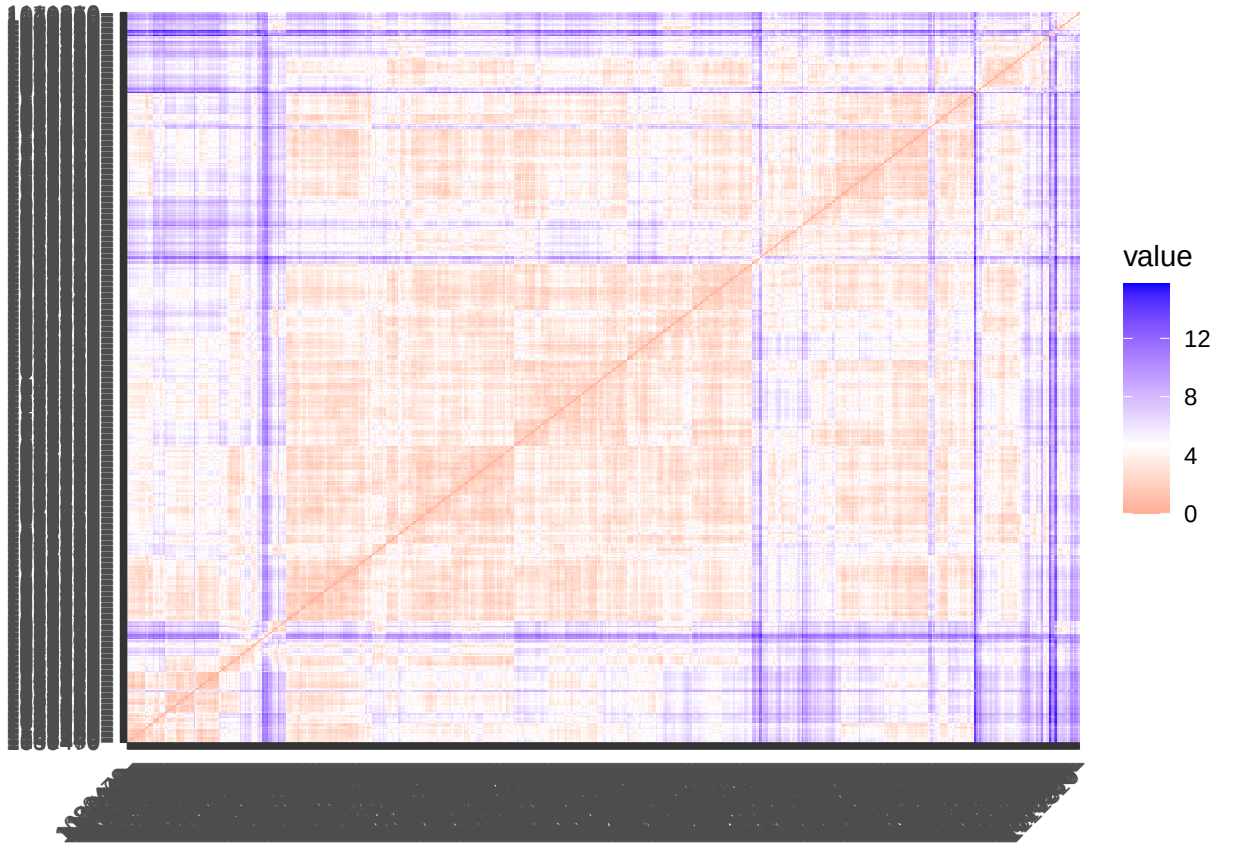
```r
Purchase_B<-bathsoap_total[,c(1,7,8,11:30)]
# total volumes of each brand
volume <- function(x){
```

```
return(x*Purchase_B$Total.Volume)
}
vol<-as.data.frame(lapply(Purchase_B[12:23],volume))
Purchase_Behav <- Purchase_B[,1:11]
Purchase_Behaviour <- cbind(Purchase_Behav,vol)
Purchase_Behaviour$max <- apply(Purchase_Behaviour[,15:22], 1, max)
boxplot(Purchase_Behaviour)
```




```
# data requires normalisation.
PB_norm <- scale(Purchase_Behaviour[c(1:11,23,24)])
```

```
#Visualizing distance matrix
set.seed(100)
distance_matrix <- get_dist(PB_norm)
fviz_dist(distance_matrix)
```




```
#computing kmeans clusterring
#first start with k=4 and number of restarts = 25 (random selection based on fviz_distance)
k4 <- kmeans(PB_norm, centers = 4, nstart = 25)
# Visualize the output
k4$centers
```

```
##          SEC         HS       CHILD No..of.Brands Brand.Runs Total.Volume
## 1  0.6436635  0.2686230 -0.07433688    -0.5499806 -0.6087683   0.09424811
## 2  0.3214118  1.1370645 -0.26378266     0.1300484  0.1335725   2.16373574
## 3 -0.2529287  0.1361035 -0.26922204     0.7943505  0.9004529   0.03873005
## 4 -0.4468408 -0.8515048  0.51901399    -0.5117687 -0.5904084  -0.90415576
##   No..of..Trans      Value Trans...Brand.Runs   Vol.Tran Avg..Price Others.999
## 1    -0.2427642 -0.2364996         0.66168007  0.3302989 -0.7372436 -0.2133999
## 2     0.4744131  1.9263725         0.05109792  1.7117993 -0.3520226  1.6178264
## 3     0.7284980  0.2546817        -0.29390112 -0.4716990  0.3400056  0.2363029
## 4    -0.8631825 -0.7611277        -0.31759384 -0.3255374  0.4407425 -0.6531358
##          max
## 1  0.4379557
## 2  1.2129507
## 3 -0.3026326
## 4 -0.4819425
```

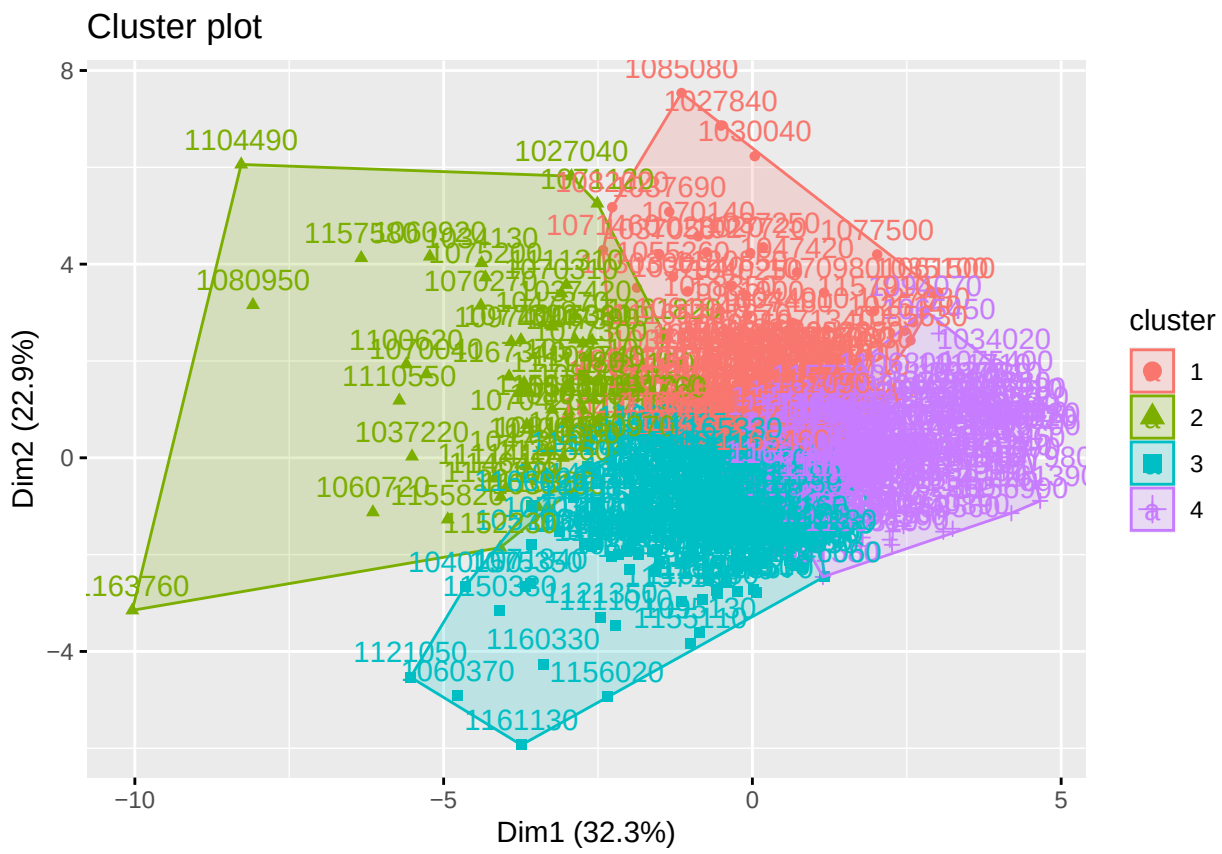
```
k4$size
```

```
## [1] 168  57 212 163
```

```
k4$cluster[150]
```

```
## 1065160
##       4
```

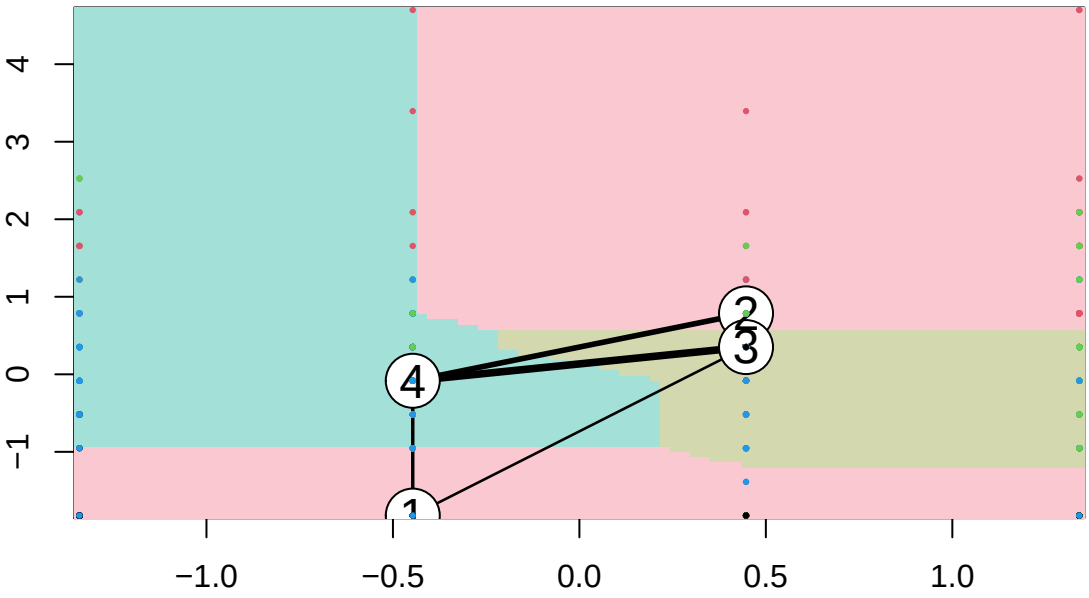
```
fviz_cluster(k4, data = PB_norm)
```



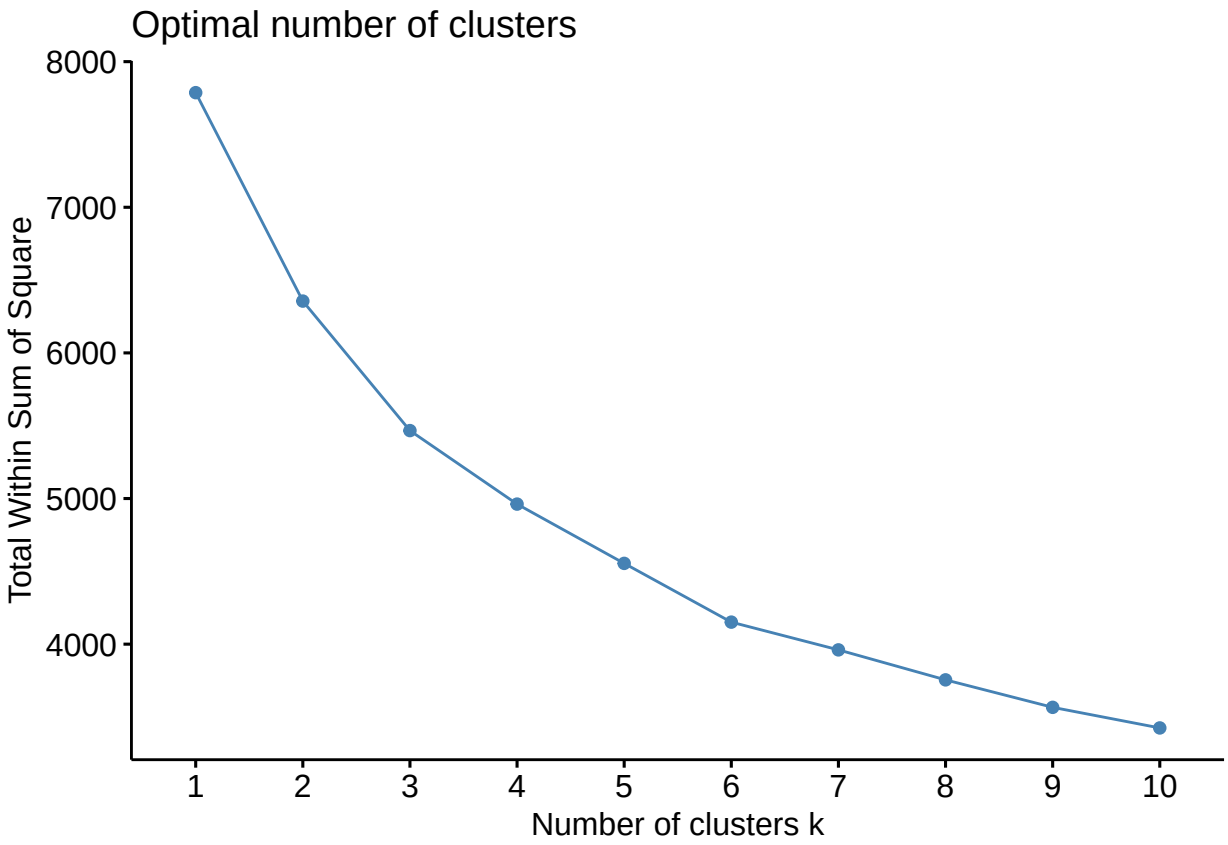

```
#rerun model with other distances
set.seed(120)
#kmeans clustering, using manhattan distance
k4 = kcca(PB_norm, k=4, kccaFamily("kmedians"))
#Apply the predict() function
clusters_index <- predict(k4)
dist(k4@centers)
```

```
##          1        2        3
## 2 5.951159
## 3 3.766939 3.164505
## 4 3.723403 2.992371 2.493683
```

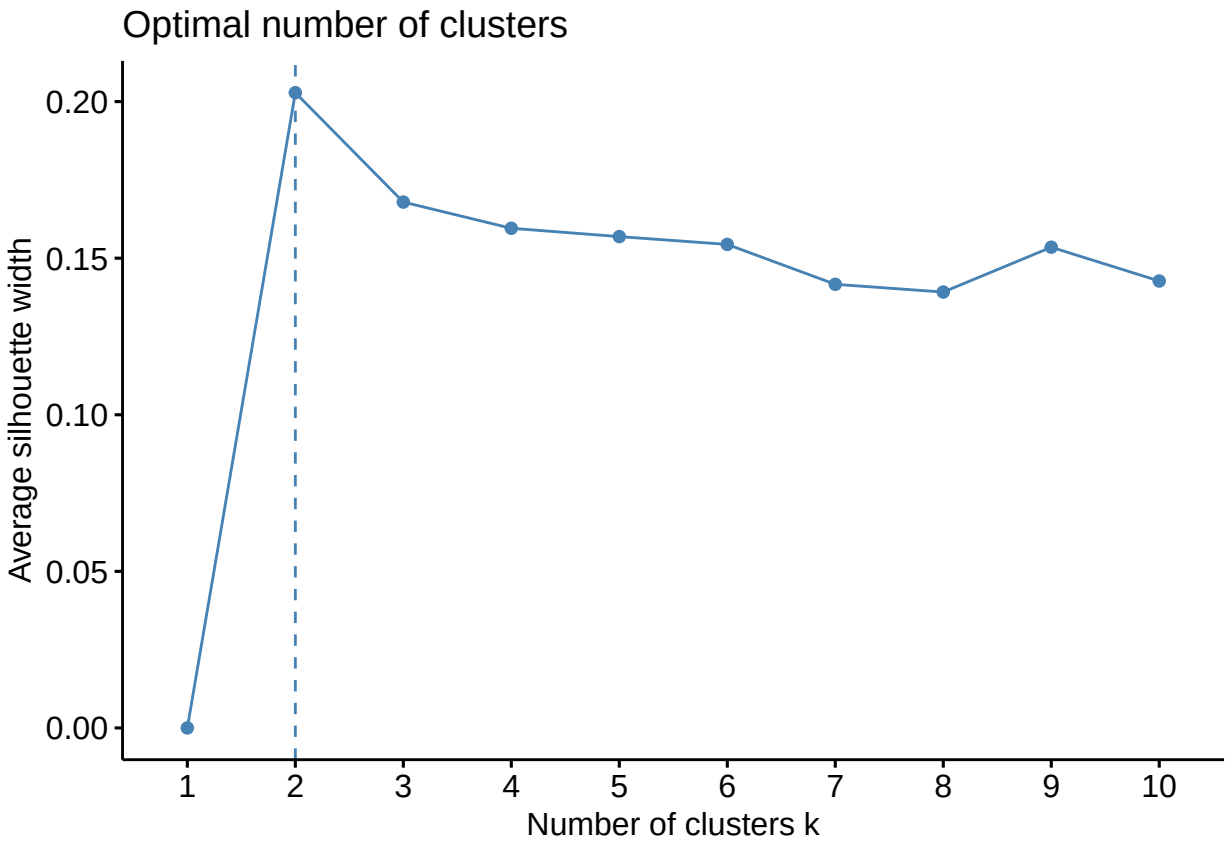
```
image(k4)
points(PB_norm, col=clusters_index, pch=19, cex=0.3)
```




```
#using elbow chart to determine k
set.seed(100)
# Scaling the data frame (z-score)
fviz_nbclust(PB_norm, kmeans, method = "wss")
```



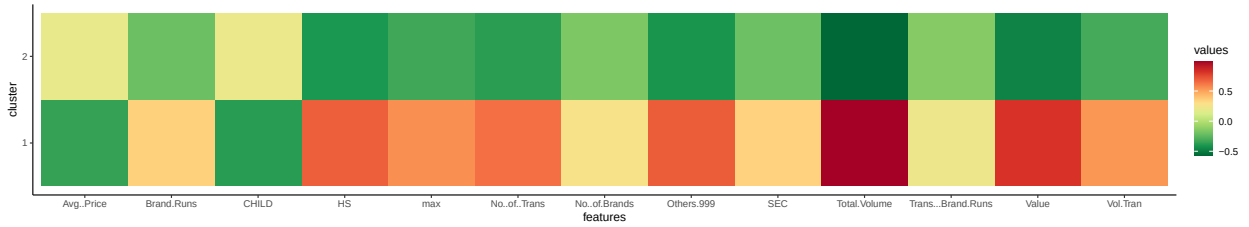

```
fviz_nbclust(PB_norm, kmeans, method = "silhouette")
```




```
set.seed(123)
k2 <- kmeans(PB_norm, centers = 2, nstart = 25) # k = 2, number of restarts = 25
center <- k2$centers
cluster <- c(1:2)
center_PB <- data.frame(cluster, center)
center_reshape <- gather(center_PB, features, values,SEC,HS,CHILD,No..of.Brands, Brand.Runs, Total.Volu

set.seed(123)
library(RColorBrewer)
# create the palette of colors we will use to plot the heat map
hm.palette <-colorRampPalette(rev(brewer.pal(10, 'RdYlGn')),space='Lab')

# Plot the heat map
ggplot(data = center_reshape, aes(x = features, y = cluster, fill = values)) +
    scale_y_continuous(breaks = seq(1, 2, by = 1)) +
    geom_tile() +
    coord_equal() +
    scale_fill_gradientn(colours = hm.palette(90)) +
    theme_classic()
```




```
k2$centers
```

```
##          SEC         HS      CHILD No..of.Brands Brand.Runs Total.Volume
## 1  0.3480004  0.6860247 -0.3658788     0.2708406  0.3523385    0.9886332
## 2 -0.1971699 -0.3886876  0.2072995    -0.1534528 -0.1996278   -0.5601394
##   No..of..Trans      Value Trans...Brand.Runs  Vol.Tran Avg..Price Others.999
## 1     0.6356181  0.8111733          0.2293300  0.534518 -0.3426291  0.6969886
## 2    -0.3601283 -0.4595943         -0.1299337 -0.302847  0.1941267 -0.3948995
##          max
## 1  0.5572348
## 2 -0.3157179
```

```
k2$size
```

```
## [1] 217 383
```

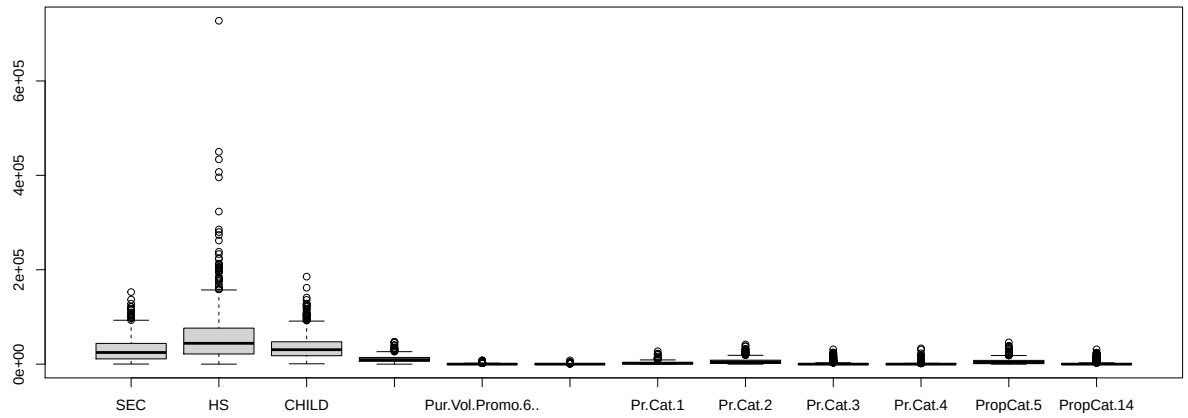
Cluster 1 has 217cstomers and cluster 2 has 383customers Cluster 1 has highEr values for all variables except Avg.Price and children cluster 1 has higher brand loyalty, they have higher volumes and value. Cluster 1 has higher financial capacity and more people in the house which can explain the higher spending. cluster 2 has more children, (maybe the brands represented in the data are not child friendly) Advertising agencies can use this clustering to target customers effectively. manufacturers can monitor their market share (child product manufacturers will clearly favour cluster 2)

#Clustering - Basis of Purchase

```
#Subsetting basis of purchase variables
Basis_P<-bathsoap_total[,c(1,7,8,13,19:21,31:35,44)]
```

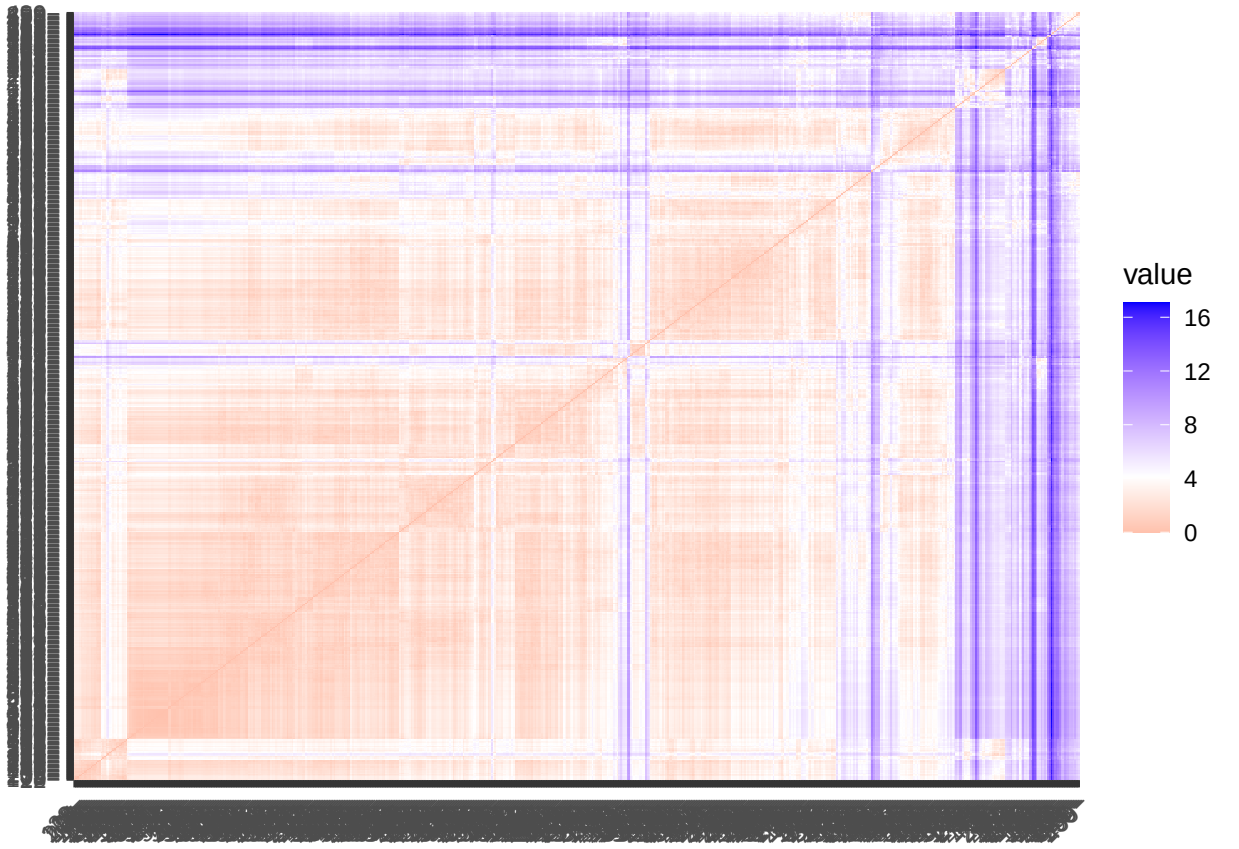
```r
# Finding out the total volumes for each brand category
volume2 <- function(x){
return(x*Basis_P$Total.Volume)
}
Basis_Pur<-as.data.frame(lapply(Basis_P[c(1:3,5:13)],volume2))
```

```r
boxplot(Basis_Pur)
```




```r
#normalise data
BoP_norm <- scale(Basis_Pur)
```

```r
#Visualizing distance matrix
set.seed(100)
distance_matrix <- get_dist(BoP_norm)
fviz_dist(distance_matrix)
```




```
#computing kmeans clusterring
#first start with k=4 and number of restarts = 25 (random selection based on fviz_distance)
k4 <- kmeans(BoP_norm, centers = 4, nstart = 25)
# Visualize the output
k4$centers
```

```
##          SEC         HS      CHILD Pur.Vol.No.Promo.... Pur.Vol.Promo.6..
## 1  1.3398853  0.8388349  0.9640276            1.0045709        0.01047265
## 2 -0.6126846 -0.5185481 -0.5523469           -0.6319346       -0.16451135
## 3  2.0816603  2.7795932  2.1468818            2.7940308        0.21561831
## 4  0.4547148  0.3295837  0.4401655            0.4853518        0.26250666
##   Pur.Vol.Other.Promo..   Pr.Cat.1   Pr.Cat.2   Pr.Cat.3    Pr.Cat.4  PropCat.5
## 1            0.71005073 -0.5382112 -0.4101528  2.8404932 -0.00639854 -0.4156015
## 2           -0.18973984 -0.1538657 -0.4076420 -0.2877099 -0.22564155 -0.4279788
## 3            0.83912023  0.5578547  2.9364848 -0.1586362  0.68563489  3.1523025
## 4            0.03490422  0.3468152  0.4493264 -0.2352592  0.31383519  0.4584521
##   PropCat.14
## 1  2.8318568
## 2 -0.2832380
## 3 -0.1531577
## 4 -0.2416372
```

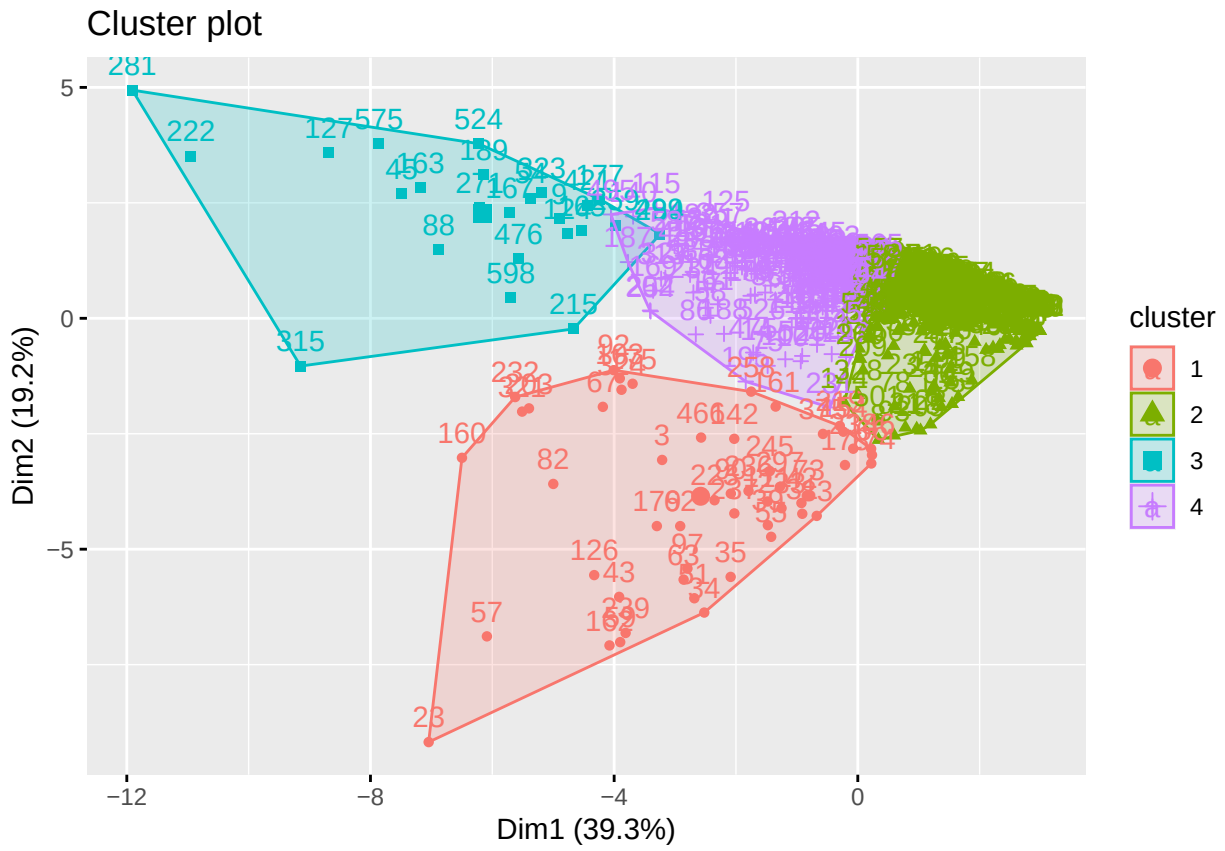
```
k4$size
```

```
## [1]  51 336  25 188
```

```
k4$cluster[150]
```

```
## [1] 2
```

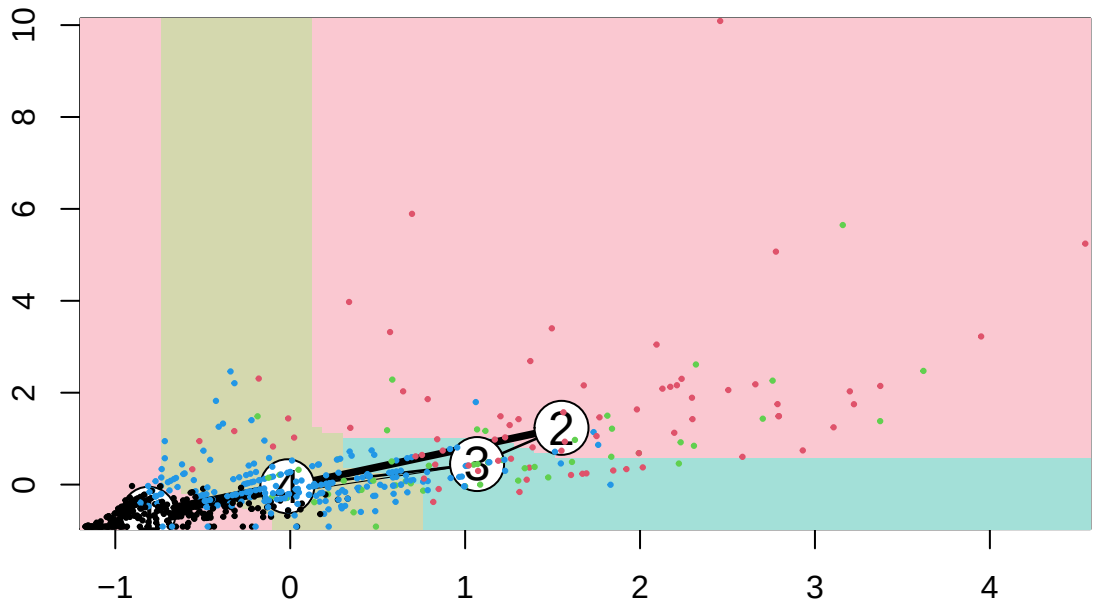
```
fviz_cluster(k4, data = BoP_norm)
```



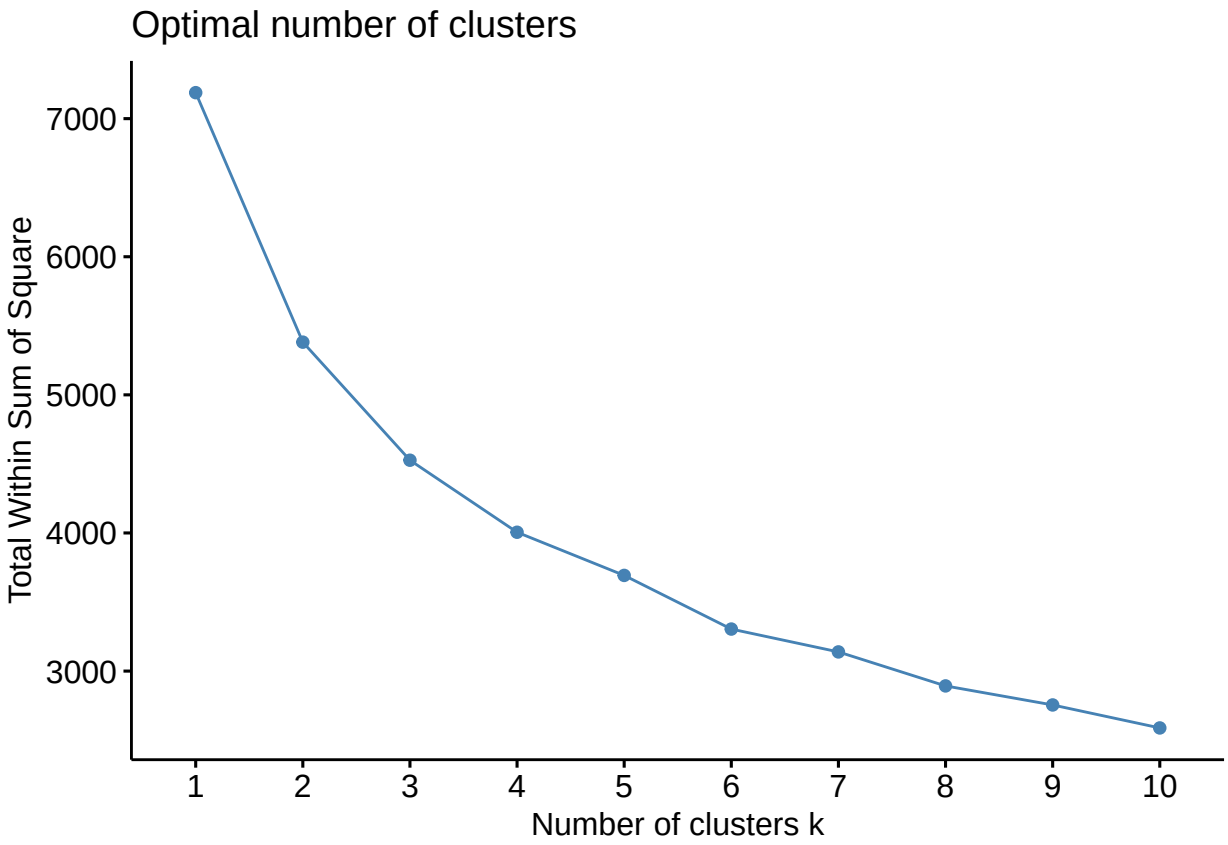

```
#rerun model with other distances
set.seed(120)
#kmeans clustering, using manhattan distance
k4 = kcca(BoP_norm, k=4, kccaFamily("kmedians"))
#Apply the predict() function
clusters_index <- predict(k4)
dist(k4@centers)
```

```
##          1        2        3
## 2 5.244569
## 3 5.568244 5.854421
## 4 1.934697 3.372444 5.136172
```

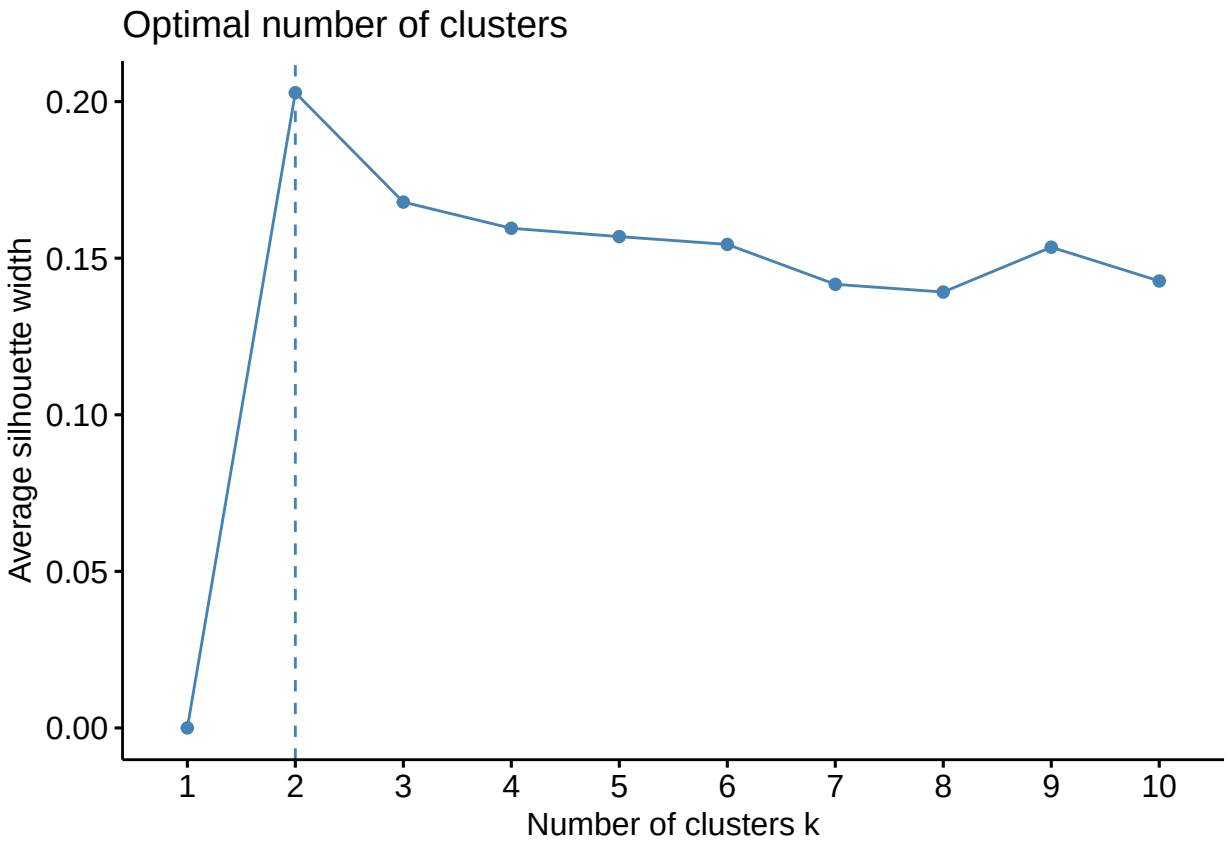
```
image(k4)
points(BoP_norm, col=clusters_index, pch=19, cex=0.3)
```




```
#using elbow chart to determine k
set.seed(100)
# Scaling the data frame (z-score)
fviz_nbclust(BoP_norm, kmeans, method = "wss")
```




```
fviz_nbclust(PB_norm, kmeans, method = "silhouette")
```



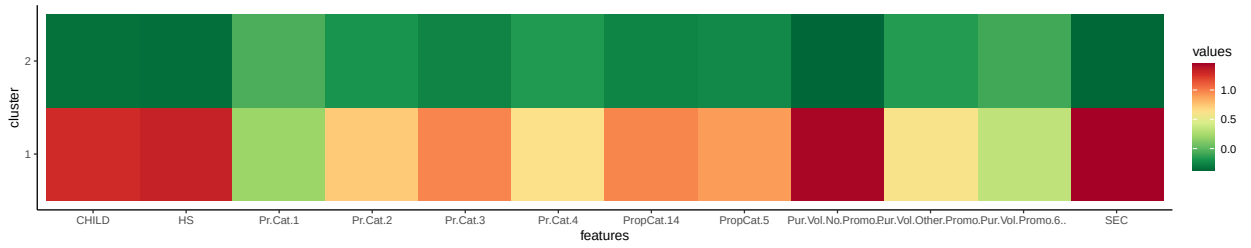

```
set.seed(123)
k2 <- kmeans(BoP_norm, centers = 2, nstart = 25) # k = 2, number of restarts = 25
center <- k2$centers
cluster <- c(1:2)
center_BoP <- data.frame(cluster, center)
str(center_BoP)
```

```
## 'data.frame':    2 obs. of  13 variables:
##  $ cluster              : int  1 2
##  $ SEC                  : num  1.443 -0.376
##  $ HS                   : num  1.316 -0.343
##  $ CHILD                : num  1.271 -0.331
##  $ Pur.Vol.No.Promo.... : num  1.423 -0.371
##  $ Pur.Vol.Promo.6..    : num  0.3324 -0.0866
##  $ Pur.Vol.Other.Promo..: num  0.607 -0.158
##  $ Pr.Cat.1             : num  0.1999 -0.0521
##  $ Pr.Cat.2             : num  0.724 -0.189
##  $ Pr.Cat.3             : num  0.968 -0.252
##  $ Pr.Cat.4             : num  0.629 -0.164
##  $ PropCat.5            : num  0.894 -0.233
##  $ PropCat.14           : num  0.964 -0.251
```

```
center_reshape <- gather(center_BoP, features, values,SEC,HS,CHILD, Pur.Vol.No.Promo....,Pur.Vol.Promo.
set.seed(123)
library(RColorBrewer)
```

```
# create the palette of colors we will use to plot the heat map
hm.palette <-colorRampPalette(rev(brewer.pal(10, 'RdYlGn')),space='Lab')

# Plot the heat map
ggplot(data = center_reshape, aes(x = features, y = cluster, fill = values)) +
    scale_y_continuous(breaks = seq(1, 2, by = 1)) +
    geom_tile() +
    coord_equal() +
    scale_fill_gradientn(colours = hm.palette(90)) +
    theme_classic()
```




```
k2$centers
```

```
##          SEC         HS      CHILD Pur.Vol.No.Promo.... Pur.Vol.Promo.6..
## 1  1.4429256  1.3164639  1.2712372            1.4229841        0.33238359
## 2 -0.3758882 -0.3429444 -0.3311626           -0.3706933       -0.08658732
##   Pur.Vol.Other.Promo..    Pr.Cat.1   Pr.Cat.2   Pr.Cat.3   Pr.Cat.4  PropCat.5
## 1             0.6066892  0.19987748  0.7236554  0.9675907  0.6286412  0.8940245
## 2            -0.1580451 -0.05206892 -0.1885153 -0.2520615 -0.1637637 -0.2328971
##   PropCat.14
## 1  0.9637185
## 2 -0.2510527
```

```
k2$size
```

```
## [1] 124 476
```

Cluster 1 has 124customers aand cluster2 has 476customers Cluster 1 is higher in all variables Cluster 1 does not seem to be affected by price discounts. Cluster 1 is buying more, has more available income, more people in the household and more children

#Clustering with both Purchase behaviour and basis of purchase

```
PB_BoP<-cbind(BoP_norm, PB_norm)
# we will use k=2
set.seed(123)
k2 <- kmeans(PB_BoP, centers = 2, nstart = 25) # k = 2, number of restarts = 25
k2$centers
```

```
##          SEC         HS      CHILD Pur.Vol.No.Promo.... Pur.Vol.Promo.6..
## 1 -0.4285458 -0.3894409 -0.3586244           -0.4329829       -0.08554222
## 2  1.2971419  1.1787774  1.0855007            1.3105724        0.25892310
##   Pur.Vol.Other.Promo..    Pr.Cat.1   Pr.Cat.2   Pr.Cat.3   Pr.Cat.4  PropCat.5
## 1            -0.1585231 -0.07750288 -0.2483276 -0.2498533 -0.1688090 -0.2634564
## 2             0.4798249  0.23458926  0.7516494  0.7562673  0.5109586  0.7974420
##   PropCat.14        SEC         HS       CHILD No..of.Brands  Brand.Runs
## 1 -0.2491663 -0.1654599 -0.2655267  0.06880217   -0.04651094 -0.05630459
## 2  0.7541880  0.5008215  0.8037084 -0.20825354    0.14078143  0.17042529
##   Total.Volume No..of..Trans      Value Trans...Brand.Runs   Vol.Tran
## 1   -0.4406481    -0.1822149 -0.3322173         -0.1275549 -0.3024582
## 2    1.3337737     0.5515364  1.0055704          0.3860890  0.9154942
##   Avg..Price Others.999        max
## 1  0.1779729 -0.2787031 -0.2927942
## 2 -0.5386966  0.8435914  0.8862427
```

```
k2$size
```

```
## [1] 451 149
```

Cluster 1 has 451cstomers and cluster 2 has 149customers Cluster 1 has highEr values for all variables except Avg.Price and children cluster 1 has higher brand loyalty, they have higher volumes and value. Cluster 1 has higher financial capacity and more people in the house which can explain the higher spending. cluster 2 has more children, (maybe the brands represented in the data are not child friendly) Advertising agencies can use this clustering to target customers effectively. manufacturers can monitor their market share (child product manufacturers will clearly favour cluster 2) Cluster 1 are high value customers and should be targeted with high value products Cluster 2 seem to be lower valus customers that respond to price changes, this charateristics can also be used as a marketing approach.

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

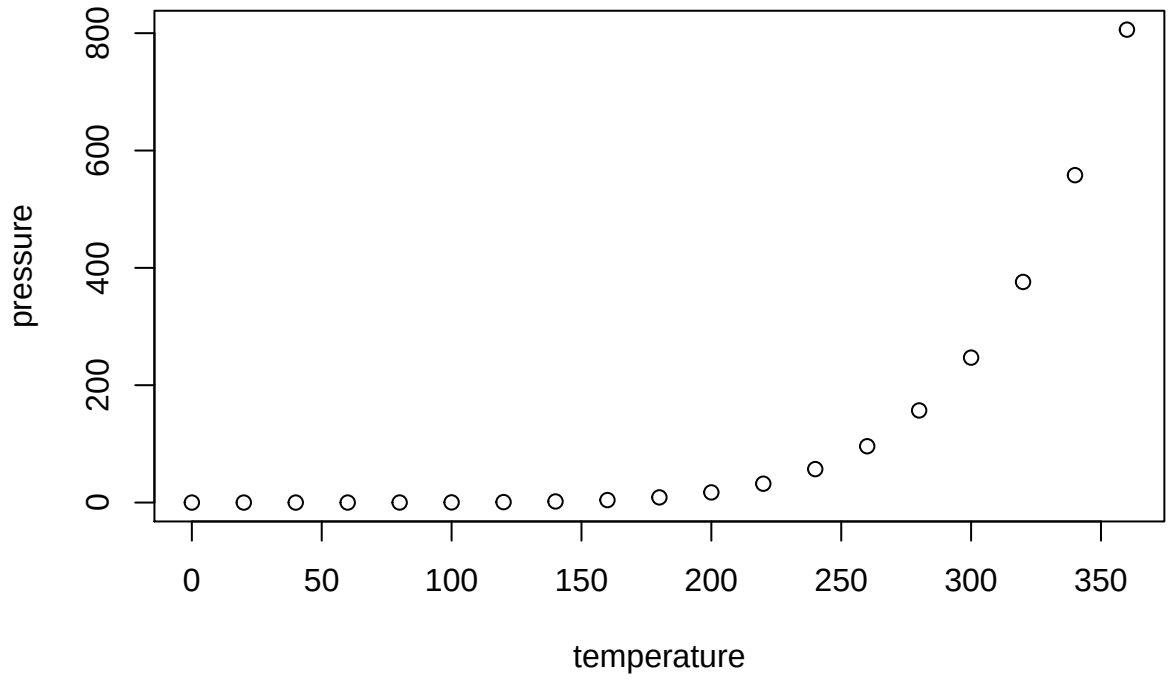
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.