RS-Final Exam

Niharika Pillanagoyala

5/5/2021

## 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.

# Loading Required Libraries

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(factoextra)

## Warning: package 'factoextra' was built under R version 4.0.4

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

library(hrbrthemes)

## Warning: package 'hrbrthemes' was built under R version 4.0.5

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

## Please use hrbrthemes::import\_roboto\_condensed() to install Roboto Condensed and

## if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow

library(GGally)

## Warning: package 'GGally' was built under R version 4.0.5

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(viridis)

## Warning: package 'viridis' was built under R version 4.0.5

## Loading required package: viridisLite

## Warning: package 'viridisLite' was built under R version 4.0.5

library(readr)  
library(Hmisc)

## Warning: package 'Hmisc' was built under R version 4.0.5

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:caret':  
##   
## cluster

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.4

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

## v tibble 3.0.5 v dplyr 1.0.4  
## v tidyr 1.1.2 v stringr 1.4.0  
## v purrr 0.3.4 v forcats 0.5.1

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.4

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x purrr::lift() masks caret::lift()  
## x dplyr::src() masks Hmisc::src()  
## x dplyr::summarize() masks Hmisc::summarize()

library(dplyr)  
library(ggplot2)

# Data Importing

### Loading dataset

setwd("C:/Users/nihar/Desktop/Assignment/ML/Recommended system-Final Exam")  
Soapdata<- read\_csv("BathSoap.csv", col\_types = cols(`Member id` = col\_number(),   
 SEC = col\_number(), FEH = col\_number(),   
 MT = col\_number(), SEX = col\_number(),   
 AGE = col\_number(), EDU = col\_number(),   
 HS = col\_number(), CHILD = col\_number(),   
 CS = col\_number(), `Affluence Index` = col\_number(),   
 `No. of Brands` = col\_number(), `Brand Runs` = col\_number(),   
 `Total Volume` = col\_number(), `No. of Trans` = col\_number(),   
 Value = col\_number(), `Trans / Brand Runs` = col\_number(),   
 `Vol/Tran` = col\_number(), `Avg. Price` = col\_number(),   
 `Pur Vol No Promo - %` = col\_number(),   
 `Pur Vol Promo 6 %` = col\_number(), `Pur Vol Other Promo %` = col\_number(),   
 `Br. Cd. 57, 144` = col\_number(), `Br. Cd. 55` = col\_number(),   
 `Br. Cd. 272` = col\_number(), `Br. Cd. 286` = col\_number(),   
 `Br. Cd. 24` = col\_number(), `Br. Cd. 481` = col\_number(),   
 `Br. Cd. 352` = col\_number(), `Br. Cd. 5` = col\_number(),   
 `Others 999` = col\_number(), `Pr Cat 1` = col\_number(),   
 `Pr Cat 2` = col\_number(), `Pr Cat 3` = col\_number(),   
 `Pr Cat 4` = col\_number(), `PropCat 5` = col\_number(),   
 `PropCat 6` = col\_number(), `PropCat 7` = col\_number(),   
 `PropCat 8` = col\_number(), `PropCat 9` = col\_number(),   
 `PropCat 10` = col\_number(), `PropCat 11` = col\_number(),   
 `PropCat 12` = col\_number(), `PropCat 13` = col\_number(),   
 `PropCat 14` = col\_number(), `PropCat 15` = col\_number()))

# Converting Binary variables from numeric to factor(i.e. Binary variables)  
Soapdata$SEC <- factor(Soapdata$SEC)  
Soapdata$FEH <- factor(Soapdata$FEH)  
Soapdata$MT <- factor(Soapdata$MT)  
Soapdata$SEX <- factor(Soapdata$SEX)  
Soapdata$AGE <- factor(Soapdata$AGE)  
Soapdata$EDU <- factor(Soapdata$EDU)  
Soapdata$HS <- factor(Soapdata$HS)  
Soapdata$CHILD <- factor(Soapdata$CHILD)  
Soapdata$CS <- factor(Soapdata$CS)  
Soapdata$`Affluence Index` <- factor(Soapdata$`Affluence Index`)  
  
# Converting distinct number variables from numeric to integer  
Soapdata$`No. of Brands`<- as.integer(Soapdata$`No. of Brands`)  
Soapdata$`Brand Runs`<- as.integer(Soapdata$`Brand Runs`)  
Soapdata$`Total Volume`<- as.integer(Soapdata$`Total Volume`)  
Soapdata$`No. of Trans`<- as.integer(Soapdata$`No. of Trans`)  
  
# Converting percentages in character to floating numericals  
Soapdata$`Pur Vol No Promo - %`<- as.numeric(str\_replace(Soapdata$`Pur Vol No Promo - %`,"%",""))/100  
Soapdata$`Pur Vol Promo 6 %`<- as.numeric(str\_replace(Soapdata$`Pur Vol Promo 6 %`,"%",""))/100  
Soapdata$`Pur Vol Other Promo %`<- as.numeric(str\_replace(Soapdata$`Pur Vol Other Promo %`,"%",""))/100  
Soapdata$`Br. Cd. 24` <- as.numeric(str\_replace(Soapdata$`Br. Cd. 24`,"%",""))/100  
Soapdata$`Br. Cd. 57, 144`<- as.numeric(str\_replace(Soapdata$`Br. Cd. 57, 144`,"%",""))/100  
Soapdata$`Br. Cd. 55` <- as.numeric(str\_replace(Soapdata$`Br. Cd. 55`,"%",""))/100  
Soapdata$`Br. Cd. 272` <- as.numeric(str\_replace(Soapdata$`Br. Cd. 272`,"%",""))/100  
Soapdata$`Br. Cd. 286` <- as.numeric(str\_replace(Soapdata$`Br. Cd. 286`,"%",""))/100  
Soapdata$`Br. Cd. 481` <- as.numeric(str\_replace(Soapdata$`Br. Cd. 481`,"%",""))/100  
Soapdata$`Br. Cd. 352` <- as.numeric(str\_replace(Soapdata$`Br. Cd. 352`,"%",""))/100  
Soapdata$`Br. Cd. 5` <- as.numeric(str\_replace(Soapdata$`Br. Cd. 5`,"%",""))/100  
Soapdata$`Others 999` <- as.numeric(str\_replace(Soapdata$`Others 999`,"%",""))/100  
Soapdata$`Pr Cat 1` <- as.numeric(str\_replace(Soapdata$`Pr Cat 1`,"%",""))/100  
Soapdata$`Pr Cat 2` <- as.numeric(str\_replace(Soapdata$`Pr Cat 2`,"%",""))/100  
Soapdata$`Pr Cat 3` <- as.numeric(str\_replace(Soapdata$`Pr Cat 3`,"%",""))/100  
Soapdata$`Pr Cat 4` <- as.numeric(str\_replace(Soapdata$`Pr Cat 4`,"%",""))/100  
Soapdata$`PropCat 5` <- as.numeric(str\_replace(Soapdata$`PropCat 5`,"%",""))/100  
Soapdata$`PropCat 6` <- as.numeric(str\_replace(Soapdata$`PropCat 6`,"%",""))/100  
Soapdata$`PropCat 7` <- as.numeric(str\_replace(Soapdata$`PropCat 7`,"%",""))/100  
Soapdata$`PropCat 8` <- as.numeric(str\_replace(Soapdata$`PropCat 8`,"%",""))/100  
Soapdata$`PropCat 9` <- as.numeric(str\_replace(Soapdata$`PropCat 9`,"%",""))/100  
Soapdata$`PropCat 10`<- as.numeric(str\_replace(Soapdata$`PropCat 10`,"%",""))/100  
Soapdata$`PropCat 11`<- as.numeric(str\_replace(Soapdata$`PropCat 11`,"%",""))/100  
Soapdata$`PropCat 12`<- as.numeric(str\_replace(Soapdata$`PropCat 12`,"%",""))/100  
Soapdata$`PropCat 13`<- as.numeric(str\_replace(Soapdata$`PropCat 13`,"%",""))/100  
Soapdata$`PropCat 14`<- as.numeric(str\_replace(Soapdata$`PropCat 14`,"%",""))/100  
Soapdata$`PropCat 15`<- as.numeric(str\_replace(Soapdata$`PropCat 15`,"%",""))/100

# Finding the total null values  
  
sum(is.na(Soapdata))

## [1] 0

Soapdata <- data.frame(Soapdata)  
Soapdata[, c(5,8,7,10)][Soapdata[,c(5,8,7,10)] == 0] <- NA  
head(Soapdata)

## Member.id SEC FEH MT SEX AGE EDU HS CHILD CS Affluence.Index  
## 1 1010010 4 3 10 1 4 4 2 4 1 2  
## 2 1010020 3 2 10 2 2 4 4 2 1 19  
## 3 1014020 2 3 10 2 4 5 6 4 1 23  
## 4 1014030 4 0 0 <NA> 4 <NA> <NA> 5 <NA> 0  
## 5 1014190 4 1 10 2 3 4 4 3 1 10  
## 6 1017020 4 3 10 2 3 4 5 2 1 13  
## No..of.Brands Brand.Runs Total.Volume No..of..Trans Value Trans...Brand.Runs  
## 1 3 17 8025 24 818.0 1.41  
## 2 5 25 13975 40 1681.5 1.60  
## 3 5 37 23100 63 1950.0 1.70  
## 4 2 4 1500 4 114.0 1.00  
## 5 3 6 8300 13 591.0 2.17  
## 6 3 26 18175 41 1705.5 1.58  
## Vol.Tran Avg..Price Pur.Vol.No.Promo.... Pur.Vol.Promo.6..  
## 1 334.38 10.19 1.00 0.00  
## 2 349.38 12.03 0.89 0.10  
## 3 366.67 8.44 0.94 0.02  
## 4 375.00 7.60 1.00 0.00  
## 5 638.46 7.12 0.61 0.14  
## 6 443.29 9.38 1.00 0.00  
## Pur.Vol.Other.Promo.. Br..Cd..57..144 Br..Cd..55 Br..Cd..272 Br..Cd..286  
## 1 0.00 0.38 0.13 0 0.00  
## 2 0.02 0.02 0.08 0 0.00  
## 3 0.04 0.03 0.55 0 0.03  
## 4 0.00 0.40 0.60 0 0.00  
## 5 0.24 0.05 0.14 0 0.00  
## 6 0.00 0.08 0.07 0 0.00  
## Br..Cd..24 Br..Cd..481 Br..Cd..352 Br..Cd..5 Others.999 Pr.Cat.1 Pr.Cat.2  
## 1 0 0.00 0 0.00 0.492 0.23 0.56  
## 2 0 0.06 0 0.14 0.699 0.29 0.55  
## 3 0 0.00 0 0.02 0.379 0.12 0.32  
## 4 0 0.00 0 0.00 0.000 0.00 0.40  
## 5 0 0.00 0 0.00 0.807 0.00 0.05  
## 6 0 0.00 0 0.00 0.857 0.22 0.45  
## Pr.Cat.3 Pr.Cat.4 PropCat.5 PropCat.6 PropCat.7 PropCat.8 PropCat.9  
## 1 0.13 0.07 0.50 0.00 0.00 0.00 0.00  
## 2 0.09 0.06 0.46 0.35 0.03 0.02 0.01  
## 3 0.56 0.00 0.24 0.12 0.03 0.01 0.01  
## 4 0.60 0.00 0.40 0.00 0.00 0.00 0.00  
## 5 0.14 0.81 0.81 0.00 0.00 0.05 0.00  
## 6 0.07 0.27 0.49 0.10 0.00 0.01 0.07  
## PropCat.10 PropCat.11 PropCat.12 PropCat.13 PropCat.14 PropCat.15  
## 1 0 0.00 0.03 0 0.13 0.34  
## 2 0 0.06 0.00 0 0.08 0.00  
## 3 0 0.00 0.02 0 0.56 0.00  
## 4 0 0.00 0.00 0 0.60 0.00  
## 5 0 0.00 0.00 0 0.14 0.00  
## 6 0 0.00 0.00 0 0.07 0.27

# Counting the total number of zero values in the categorical data.  
  
colSums(is.na(Soapdata))

## Member.id SEC FEH   
## 0 0 0   
## MT SEX AGE   
## 0 68 0   
## EDU HS CHILD   
## 73 68 0   
## CS Affluence.Index No..of.Brands   
## 99 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

NAValues <- colnames(Soapdata)[apply(Soapdata, 2, anyNA) ]  
NAValues

## [1] "SEX" "EDU" "HS" "CS"

# Imputing Zero insignificant values in categorical variables with their respective variable mode.   
  
Soapdata$MT <- impute(Soapdata$MT, mode)  
Soapdata$EDU <- impute(Soapdata$EDU, mode)  
Soapdata$HS <- impute(Soapdata$HS, mode)  
Soapdata$CS <- impute(Soapdata$CS, mode)  
Soapdata$SEX <- impute(Soapdata$SEX, mode)

#Data Preparation. #Feature Engineering;

### Q1. Use k-means clustering to identify clusters of households based on:

* (a). The variables that describe purchase behavior (including brand loyalty)
* vol/Trans
* Brand Runs
* No. of Trans
* No. of Brands
* Others999
* Value
* Loyality\_Brand

### Lets consider the maximum value in brand defines the loyality of the brand to the customer. We are analysing maximum value to find the loyality.

set.seed(123)  
Loyality<-Soapdata[, 23:30]  
Soapdata$Loyality\_Brand<- as.numeric(apply(Loyality,1,max))

#Clusters based on "purchase behavior" (including brand loyalty)  
  
Data<- Soapdata[,c(12:19,31,47)]  
Omit<- na.omit(Data)  
head(Data)

## No..of.Brands Brand.Runs Total.Volume No..of..Trans Value Trans...Brand.Runs  
## 1 3 17 8025 24 818.0 1.41  
## 2 5 25 13975 40 1681.5 1.60  
## 3 5 37 23100 63 1950.0 1.70  
## 4 2 4 1500 4 114.0 1.00  
## 5 3 6 8300 13 591.0 2.17  
## 6 3 26 18175 41 1705.5 1.58  
## Vol.Tran Avg..Price Others.999 Loyality\_Brand  
## 1 334.38 10.19 0.492 0.38  
## 2 349.38 12.03 0.699 0.14  
## 3 366.67 8.44 0.379 0.55  
## 4 375.00 7.60 0.000 0.60  
## 5 638.46 7.12 0.807 0.14  
## 6 443.29 9.38 0.857 0.08

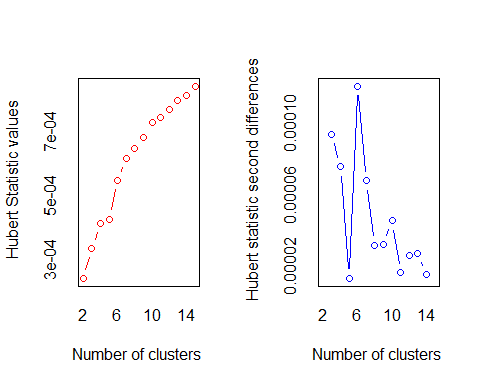
# Normalizing the data  
Normalize <- function(x){  
 (x- mean(x)) /(max(x)-min(x))  
}  
  
# Outline Eliminator function  
  
remove\_outliers <- function(x, na.rm = TRUE) {  
 qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm)  
 H <- 1.5 \* IQR(x, na.rm = na.rm)  
 y <- x  
 y[x < (qnt[1] - H)] <- NA  
 y[x > (qnt[2] + H)] <- NA  
 y  
}  
Data\_Normalize <- data.frame(lapply(Data, Normalize))  
DataScaling1<- as.data.frame(scale(Data\_Normalize))

# Plotting the visualization using Elbow method, silhoutte and Gap\_stat to find the best number of clusters

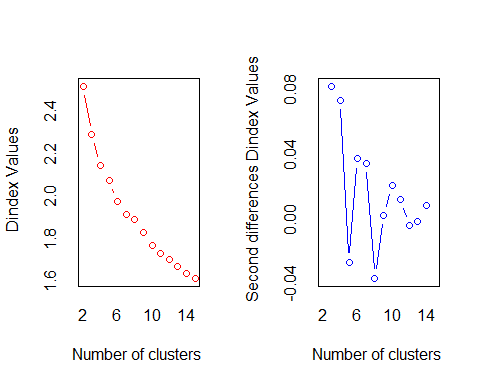
## According to the majority rule, the best number of clusters is 2

silhouette = 2 Elbow = 4 Gap\_stat = 7 Nbclust = 2

library(NbClust)  
NbClust(data = DataScaling1, diss = NULL, distance = "euclidean",  
 min.nc = 2, max.nc = 15, method = "kmeans")



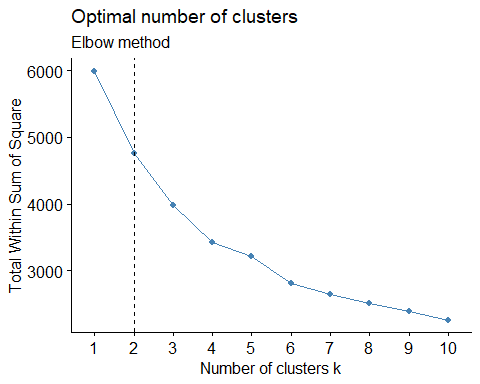
## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



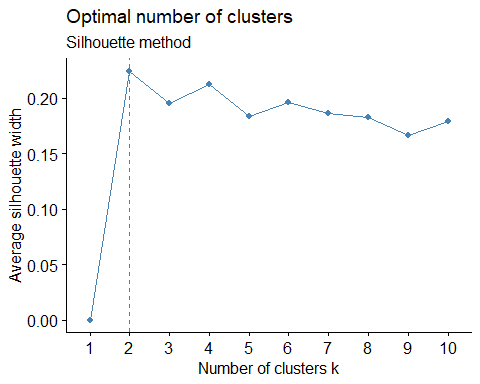
## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 5 proposed 2 as the best number of clusters   
## \* 5 proposed 3 as the best number of clusters   
## \* 5 proposed 4 as the best number of clusters   
## \* 1 proposed 6 as the best number of clusters   
## \* 1 proposed 8 as the best number of clusters   
## \* 1 proposed 10 as the best number of clusters   
## \* 5 proposed 15 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## $All.index  
## KL CH Hartigan CCC Scott Marriot TrCovW TraceW  
## 2 1.0232 155.1692 117.9995 -5.7626 585.108 6.827413e+23 439149.65 4755.930  
## 3 1.2120 151.6396 94.7498 -8.1233 1159.051 5.902081e+23 285033.15 3972.135  
## 4 4.5122 148.4717 40.4491 -6.2598 1654.055 4.598187e+23 187871.79 3428.066  
## 5 0.2546 128.8069 78.3177 -7.7013 1938.808 4.469877e+23 159943.31 3210.198  
## 6 2.3715 132.0502 43.0731 -3.4053 2443.393 2.776048e+23 117223.46 2836.800  
## 7 4.1836 124.9894 21.9225 -4.5404 2645.471 2.698060e+23 102333.18 2645.002  
## 8 0.2336 114.0336 40.2062 -5.0004 2865.735 2.441199e+23 96029.87 2550.705  
## 9 1.0851 111.3933 37.5239 -2.8539 3074.239 2.182670e+23 82934.71 2388.489  
## 10 2.1047 109.2871 23.3960 -0.8545 3268.846 1.948237e+23 73571.15 2245.892  
## 11 1.3940 104.4210 19.0745 -0.3610 3434.779 1.787808e+23 67939.73 2160.230  
## 12 0.4539 99.5671 29.3110 -0.2022 3570.682 1.696399e+23 63333.32 2092.467  
## 13 1.4688 98.0950 22.5326 1.2684 3956.436 1.046729e+23 57625.40 1993.113  
## 14 1.2492 95.5952 19.3406 2.0675 4063.687 1.015249e+23 52146.51 1919.433  
## 15 13.0305 92.9193 8.4584 2.6018 4267.698 8.295277e+22 48404.49 1858.108  
## Friedman Rubin Cindex DB Silhouette Duda Pseudot2 Beale Ratkowsky  
## 2 17.3870 1.2595 0.2411 1.9631 0.1999 0.9877 5.1705 0.0833 0.2911  
## 3 33.5637 1.5080 0.2142 1.7241 0.1962 1.4647 -69.4845 -2.1205 0.3197  
## 4 42.3645 1.7473 0.2362 1.5148 0.2124 1.2802 -58.4458 -1.4636 0.3161  
## 5 44.3679 1.8659 0.2295 1.6361 0.1851 1.1164 -21.7990 -0.6970 0.3005  
## 6 48.4899 2.1115 0.2139 1.5277 0.1968 1.4089 -58.6306 -1.9403 0.2951  
## 7 50.9919 2.2646 0.2341 1.5130 0.1847 1.6640 -76.2192 -2.6642 0.2818  
## 8 53.3964 2.3484 0.2678 1.5135 0.1871 0.7513 29.4643 2.2116 0.2672  
## 9 56.6645 2.5079 0.2617 1.5199 0.1794 2.1015 -69.1863 -3.4698 0.2579  
## 10 59.4723 2.6671 0.2515 1.4515 0.1847 1.9603 -72.9921 -3.2481 0.2493  
## 11 61.5262 2.7729 0.2464 1.4738 0.1724 1.6424 -57.8888 -2.6018 0.2404  
## 12 63.2099 2.8627 0.2405 1.4969 0.1701 1.4805 -36.3521 -2.1516 0.2324  
## 13 69.4822 3.0053 0.1983 1.4793 0.1698 1.7308 -60.7994 -2.7896 0.2262  
## 14 67.9719 3.1207 0.2296 1.4663 0.1704 1.7018 -56.4983 -2.7426 0.2200  
## 15 73.5787 3.2237 0.1900 1.4201 0.1701 1.4260 -36.1474 -1.9859 0.2143  
## Ball Ptbiserial Frey McClain Dunn Hubert SDindex Dindex SDbw  
## 2 2377.9651 0.2558 0.0460 0.7950 0.0379 2e-04 1.8227 2.5196 2.1415  
## 3 1324.0449 0.3381 0.0473 1.3441 0.0261 3e-04 1.6719 2.2927 0.9809  
## 4 857.0166 0.3761 1.3659 1.6253 0.0487 4e-04 1.6212 2.1479 1.0135  
## 5 642.0396 0.3384 -0.1204 2.3062 0.0324 4e-04 1.7139 2.0768 0.7263  
## 6 472.8001 0.3767 0.1640 2.3337 0.0339 5e-04 1.8166 1.9789 0.8337  
## 7 377.8574 0.3828 -0.1046 2.6285 0.0426 6e-04 1.7987 1.9185 0.8650  
## 8 318.8382 0.3906 0.5308 2.6212 0.0494 6e-04 1.9625 1.8928 0.8822  
## 9 265.3877 0.3715 0.3048 3.1789 0.0523 7e-04 1.9699 1.8304 0.6829  
## 10 224.5892 0.3631 0.5651 3.5873 0.0636 7e-04 1.9265 1.7703 0.6697  
## 11 196.3846 0.3495 0.2768 4.0208 0.0619 7e-04 1.9080 1.7310 0.6418  
## 12 174.3722 0.3436 0.2793 4.3226 0.0444 8e-04 1.9222 1.7042 0.6281  
## 13 153.3164 0.3372 0.1991 4.6466 0.0500 8e-04 2.0152 1.6732 0.5875  
## 14 137.1024 0.3337 0.2629 4.8882 0.0444 8e-04 1.9743 1.6405 0.5700  
## 15 123.8738 0.3283 0.0190 5.1781 0.0376 8e-04 1.9686 1.6160 0.5594  
##   
## $All.CriticalValues  
## CritValue\_Duda CritValue\_PseudoT2 Fvalue\_Beale  
## 2 0.8575 69.1160 0.9999  
## 3 0.8286 45.3157 1.0000  
## 4 0.8326 53.6686 1.0000  
## 5 0.8272 43.6599 1.0000  
## 6 0.8308 41.1361 1.0000  
## 7 0.8200 41.9329 1.0000  
## 8 0.8251 18.8702 0.0150  
## 9 0.7662 40.2833 1.0000  
## 10 0.7747 43.3417 1.0000  
## 11 0.7941 38.3785 1.0000  
## 12 0.7735 32.7911 1.0000  
## 13 0.7562 46.4248 1.0000  
## 14 0.7925 35.8645 1.0000  
## 15 0.7901 32.1457 1.0000  
##   
## $Best.nc  
## KL CH Hartigan CCC Scott Marriot  
## Number\_clusters 15.0000 2.0000 4.0000 15.0000 3.0000 6.000000e+00  
## Value\_Index 13.0305 155.1692 54.3007 2.6018 573.9427 1.615842e+23  
## TrCovW TraceW Friedman Rubin Cindex DB Silhouette  
## Number\_clusters 3.0 4.0000 3.0000 4.0000 15.00 15.0000 4.0000  
## Value\_Index 154116.5 326.1998 16.1767 -0.1207 0.19 1.4201 0.2124  
## Duda PseudoT2 Beale Ratkowsky Ball PtBiserial Frey  
## Number\_clusters 2.0000 2.0000 2.0000 3.0000 3.00 8.0000 1  
## Value\_Index 0.9877 5.1705 0.0833 0.3197 1053.92 0.3906 NA  
## McClain Dunn Hubert SDindex Dindex SDbw  
## Number\_clusters 2.000 10.0000 0 4.0000 0 15.0000  
## Value\_Index 0.795 0.0636 0 1.6212 0 0.5594  
##   
## $Best.partition  
## [1] 1 2 2 1 1 2 1 1 2 1 2 2 2 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1  
## [38] 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 2 2 1 1 2 2 1 1 2 1 2 2 1 1 1 2 1 1  
## [75] 2 1 1 1 2 1 1 2 1 2 1 2 1 2 1 1 2 2 1 2 2 2 1 2 1 1 1 1 2 2 2 2 2 2 2 2 1  
## [112] 2 2 2 2 2 2 1 2 2 2 2 1 2 2 1 2 1 1 2 2 2 1 1 1 2 2 2 2 1 1 1 1 1 2 1 1 1  
## [149] 2 1 1 1 1 1 2 2 1 1 1 2 1 1 2 2 1 2 2 1 2 2 2 2 1 1 2 1 1 1 1 1 1 2 2 2 2  
## [186] 1 2 2 2 2 1 2 1 1 2 2 2 2 2 1 1 2 2 1 1 1 1 2 1 2 1 1 2 1 2 1 1 2 1 2 1 2  
## [223] 1 1 2 1 1 2 2 1 1 2 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 2 2 1 2 1 2  
## [260] 2 1 1 2 2 2 1 2 1 2 2 2 2 1 2 1 1 2 2 1 1 2 2 2 2 2 2 2 1 2 2 2 1 1 2 1 1  
## [297] 1 2 2 2 2 1 1 2 2 1 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 1 2 1 2 1 2 1 2  
## [334] 2 1 2 2 2 1 2 1 1 1 1 2 1 2 2 2 1 1 1 2 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 1  
## [371] 2 1 1 1 1 1 1 2 2 2 1 1 2 2 2 2 1 2 2 2 2 2 1 1 2 1 2 1 1 1 1 2 2 2 1 1 1  
## [408] 1 2 2 2 1 2 1 2 2 2 1 1 2 2 1 1 1 1 2 1 1 2 1 2 2 2 1 1 1 2 2 1 1 1 2 2 2  
## [445] 1 1 1 2 1 1 1 1 1 1 1 2 1 2 2 2 2 2 1 1 2 2 2 1 1 2 1 1 2 2 2 2 2 2 1 2 1  
## [482] 2 2 1 2 1 2 2 2 2 1 2 1 1 2 2 1 2 2 2 1 2 1 2 1 1 1 1 2 2 1 1 1 2 1 1 1 2  
## [519] 1 2 1 1 1 2 1 1 2 1 2 1 1 1 2 1 2 1 1 1 1 1 2 2 2 2 2 2 1 1 2 2 1 1 1 2 2  
## [556] 1 2 1 2 1 2 1 2 2 2 1 2 2 1 2 1 2 1 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 1 2 1 1  
## [593] 1 1 1 1 2 2 1 1

fviz\_nbclust(DataScaling1, kmeans, method = 'wss') +  
 geom\_vline(xintercept = 2, linetype = 2)+  
 labs(subtitle = 'Elbow method')

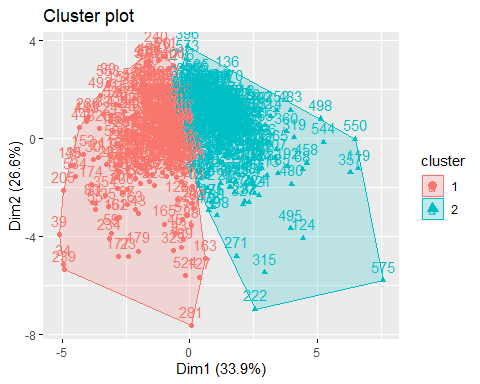


fviz\_nbclust(DataScaling1, kmeans, method = "silhouette")+  
 labs(subtitle = "Silhouette method")



# By considering the value of k=2. ### Now we will run Kmeans with k=2 and nstart = 30 and plot the clusters.

# Generating 2 clusters based on the available data set along with those added manually like Brand Loyality.  
clust1<- kmeans(DataScaling1,2,nstart = 30)  
  
#Visualize the output of the cluster  
fviz\_cluster(clust1, DataScaling1)



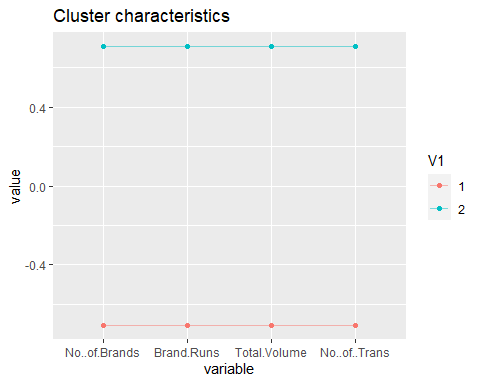
# we will store the centers of the model in Output and print the size of the 2 clusters.  
  
Output1<- as.data.frame(cbind(1:nrow(clust1$centers),clust1$centers))  
Output1$V1<- as.factor(Output1$V1)  
clust1$size

## [1] 283 317

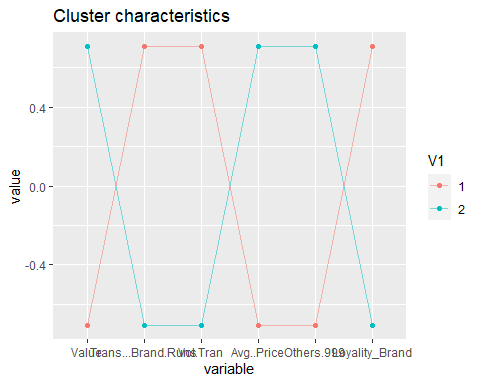
head(Output1)

## V1 No..of.Brands Brand.Runs Total.Volume No..of..Trans Value  
## 1 1 -0.5417123 -0.7088977 -0.1772315 -0.5848426 -0.3438382  
## 2 2 0.4836107 0.6328645 0.1582224 0.5221150 0.3069597  
## Trans...Brand.Runs Vol.Tran Avg..Price Others.999 Loyality\_Brand  
## 1 0.2926739 0.3196693 -0.3132908 -0.5477087 0.6584652  
## 2 -0.2612830 -0.2853830 0.2796886 0.4889639 -0.5878412

# Visualize the cluster  
ggparcoord(Output1,   
 columns = 2:5, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.5)



ggparcoord(Output1,   
 columns = 6:11, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.5)



Cluster No..of.Brands Brand.Runs Total.Volume No..of..Trans Value Trans…Brand.Runs Vol.Tran 1 -0.5417123 -0.7088977 -0.1772315 -0.5848426 -0.3438382 0.2926739 0.3196693  
2 0.4836107 0.6328645 0.1582224 0.5221150 0.3069597 -0.2612830 -0.2853830

Avg..Price Others.999 Loyality\_Brand -0.3132908 -0.5477087 0.6584652 0.2796886 0.4889639 -0.5878412

The two clusters are well-separated on everything. Cluster 1 (n=283) is high activity & value, with low loyalty. Cluster 2 (n=317) is the reverse.

\*cluster 1: Customers in this cluster have high brand loyalty; they buy the least number of brands with high volume transaction in the limited transaction they do. They have high brand runs and high vol. transactions. They donot buy from other999.

\*cluster 2: Customers in this cluster buy from others999 brands which indicate they are not brand loyal.They buy the highest number of brands and the volume of transaction is the least.

# Q1. Use k-means clustering to identify clusters of households based on:

## b) Now considering the variables that describes the purchases.

###Variables used are: *All price categories* selling proportions \*purchase volume with no promotion, promotion 6 and other promotions

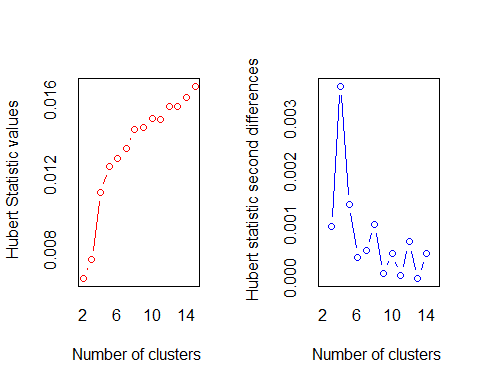
### We will follow same steps as we did previously which finds the maximum (from 36: 46). which will also give us the value for the basis of customers purchase.

###Then we will scale the data and then find the number of clusters using .

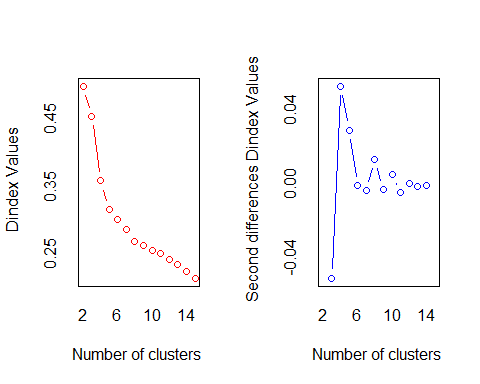
#Clusters based on "basis for purchase"  
  
Loyality2<-Soapdata[, 36:46]  
Soapdata$purchase\_on <- as.numeric(apply(Loyality2,1,which.max))  
Soapdata$purchase1\_on <- as.numeric(apply(Soapdata[,23:30],1,max))  
Data2<- Soapdata[,c(20:22,32:35,49)]  
Loyality2\_normalized <- data.frame(lapply(Data2, Normalize))

#Plotting the visualization using Elbow method, silhoutte to find the best number of clusters ## According to the majority rule, the best number of clusters is 3 silhouette = 6 Elbow = 3 Nbclust = 4

NbClust(data = Loyality2\_normalized, diss = NULL, distance = "euclidean",  
 min.nc = 2, max.nc = 15, method = "kmeans")



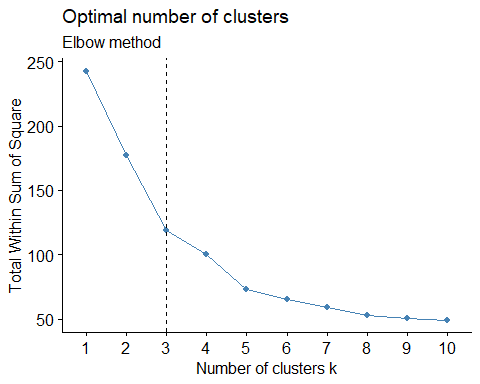
## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



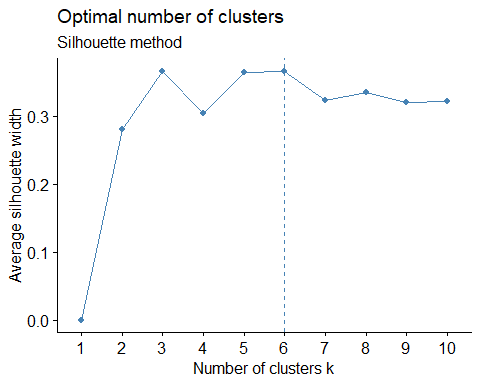
## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 4 proposed 2 as the best number of clusters   
## \* 2 proposed 3 as the best number of clusters   
## \* 10 proposed 4 as the best number of clusters   
## \* 1 proposed 5 as the best number of clusters   
## \* 2 proposed 6 as the best number of clusters   
## \* 1 proposed 8 as the best number of clusters   
## \* 3 proposed 15 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 4   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## $All.index  
## KL CH Hartigan CCC Scott Marriot TrCovW TraceW  
## 2 2.1260 197.6914 102.9959 -2.4068 801.2465 1782.2835 1292.2047 182.2448  
## 3 0.1608 167.0883 418.7278 -10.5824 1710.3620 881.2915 1113.0999 155.4679  
## 4 3.7003 328.5459 152.1280 13.7446 2821.3790 245.9326 268.3839 91.3772  
## 5 2.9483 346.7542 69.9552 22.5202 3370.3393 153.9166 153.5739 72.7961  
## 6 1.3676 323.4647 55.5274 25.8815 3761.1217 115.5554 123.9888 65.1378  
## 7 0.5984 303.4947 78.6582 21.2904 4174.3315 78.9937 107.2519 59.5692  
## 8 21.9972 305.3653 20.3484 25.8485 4427.7598 67.6299 76.3977 52.5930  
## 9 0.1278 278.4512 37.1926 24.6719 4617.5397 62.3845 74.2893 50.8453  
## 10 5.3873 266.7688 18.2123 25.6345 4897.7222 48.2824 66.0273 47.8350  
## 11 0.2500 248.9018 29.2657 24.8956 4981.9440 50.7706 61.6912 46.4026  
## 12 0.3502 239.7700 62.1950 25.5230 5102.6616 49.4096 55.5114 44.2062  
## 13 1.8163 247.7979 39.9243 29.5104 5406.0213 34.9749 44.6198 39.9776  
## 14 0.6233 246.9435 60.3978 31.4435 5570.3299 30.8457 37.8317 37.4317  
## 15 4.0056 256.8138 22.2814 35.4054 5923.2992 19.6623 31.3896 33.9342  
## Friedman Rubin Cindex DB Silhouette Duda Pseudot2 Beale Ratkowsky  
## 2 5172.388 1.3306 0.3444 1.6758 0.2634 1.1038 -41.9266 -0.4955 0.2386  
## 3 8184.685 1.5598 0.3296 1.2912 0.2965 0.5124 383.4986 5.0164 0.2494  
## 4 14097.373 2.6538 0.2889 0.9397 0.4122 0.5023 219.9421 5.2204 0.2829  
## 5 14690.207 3.3311 0.2908 1.0082 0.3645 3.6021 -197.9334 -3.7445 0.2732  
## 6 15688.165 3.7228 0.2700 1.0535 0.3679 0.9886 2.8478 0.0607 0.2992  
## 7 16508.738 4.0708 0.2580 1.0747 0.3533 2.0180 -105.9341 -2.6389 0.2810  
## 8 17112.329 4.6107 0.2313 1.0816 0.3282 1.4734 -49.4831 -1.6805 0.2672  
## 9 17357.473 4.7692 0.2340 1.1613 0.3246 2.5757 -96.6569 -3.1802 0.2541  
## 10 17437.451 5.0694 0.2293 1.1614 0.3077 1.1306 -8.0862 -0.5992 0.2390  
## 11 17526.099 5.2258 0.2269 1.1794 0.3098 2.4034 -62.4792 -3.0244 0.2308  
## 12 17604.892 5.4855 0.2295 1.2521 0.2794 2.6950 -69.8131 -3.2519 0.2224  
## 13 17596.217 6.0657 0.2173 1.2345 0.3016 1.2802 -13.7883 -1.1331 0.2203  
## 14 17709.071 6.4783 0.2148 1.2313 0.2942 0.8310 16.0620 1.0531 0.2108  
## 15 17912.535 7.1460 0.2006 1.2251 0.2940 1.4934 -19.8233 -1.7036 0.2055  
## Ball Ptbiserial Frey McClain Dunn Hubert SDindex Dindex SDbw  
## 2 91.1224 0.3456 -0.2923 0.6545 0.0320 0.0065 8.0960 0.4982 1.9058  
## 3 51.8226 0.3980 -0.2808 0.7265 0.0436 0.0075 5.6538 0.4539 0.9760  
## 4 22.8443 0.6443 0.8925 0.8289 0.0309 0.0111 4.4753 0.3581 0.5227  
## 5 14.5592 0.5933 0.3798 1.3152 0.0316 0.0125 4.8965 0.3152 0.4824  
## 6 10.8563 0.5895 0.8297 1.4819 0.0474 0.0129 4.9812 0.3013 0.4744  
## 7 8.5099 0.5631 0.7331 1.7278 0.0365 0.0134 5.0467 0.2865 0.3222  
## 8 6.5741 0.5109 1.1745 2.3071 0.0110 0.0145 5.9291 0.2680 0.2589  
## 9 5.6495 0.4821 0.9312 2.6562 0.0465 0.0146 6.3672 0.2626 0.2786  
## 10 4.7835 0.4610 0.0573 2.9605 0.0239 0.0150 6.5037 0.2541 0.2832  
## 11 4.2184 0.4621 1.1761 2.9630 0.0239 0.0150 6.6798 0.2506 0.2287  
## 12 3.6838 0.4357 0.0218 3.3907 0.0271 0.0157 6.7944 0.2420 0.2183  
## 13 3.0752 0.4421 0.8213 3.3569 0.0271 0.0157 6.7358 0.2338 0.1929  
## 14 2.6737 0.4141 0.1551 3.8885 0.0307 0.0161 7.0428 0.2241 0.1773  
## 15 2.2623 0.4094 0.2208 4.0296 0.0332 0.0168 7.0975 0.2133 0.1605  
##   
## $All.CriticalValues  
## CritValue\_Duda CritValue\_PseudoT2 Fvalue\_Beale  
## 2 0.8417 83.8956 1.0000  
## 3 0.8432 74.9295 0.0000  
## 4 0.8371 43.2162 0.0000  
## 5 0.7121 110.7899 1.0000  
## 6 0.8200 54.2300 0.9999  
## 7 0.7680 63.4444 1.0000  
## 8 0.7664 46.9350 1.0000  
## 9 0.7278 59.0952 1.0000  
## 10 0.7159 27.7790 1.0000  
## 11 0.7080 44.1233 1.0000  
## 12 0.6992 47.7579 1.0000  
## 13 0.7059 26.2455 1.0000  
## 14 0.7080 32.5770 0.3955  
## 15 0.6864 27.4153 1.0000  
##   
## $Best.nc  
## KL CH Hartigan CCC Scott Marriot TrCovW  
## Number\_clusters 8.0000 5.0000 3.0000 15.0000 4.000 4.0000 4.000  
## Value\_Index 21.9972 346.7542 315.7319 35.4054 1111.017 543.3429 844.716  
## TraceW Friedman Rubin Cindex DB Silhouette Duda  
## Number\_clusters 4.0000 4.000 4.0000 15.0000 4.0000 4.0000 2.0000  
## Value\_Index 45.5097 5912.689 -0.4166 0.2006 0.9397 0.4122 1.1038  
## PseudoT2 Beale Ratkowsky Ball PtBiserial Frey McClain  
## Number\_clusters 2.0000 2.0000 6.0000 3.0000 4.0000 1 2.0000  
## Value\_Index -41.9266 -0.4955 0.2992 39.2998 0.6443 NA 0.6545  
## Dunn Hubert SDindex Dindex SDbw  
## Number\_clusters 6.0000 0 4.0000 0 15.0000  
## Value\_Index 0.0474 0 4.4753 0 0.1605  
##   
## $Best.partition  
## [1] 1 1 4 4 3 3 1 4 1 2 3 1 3 1 1 1 1 1 1 2 4 4 4 4 4 2 1 1 3 4 4 3 4 4 4 3 3  
## [38] 3 4 4 1 4 4 4 3 1 3 1 4 2 4 1 4 1 4 1 4 3 4 1 3 4 4 1 2 4 1 2 3 1 1 3 4 1  
## [75] 3 1 1 4 1 3 1 4 4 2 1 1 3 1 2 4 2 1 4 2 1 1 4 1 4 1 1 1 3 1 3 1 2 1 1 3 4  
## [112] 1 1 3 3 2 1 4 1 1 3 2 1 2 1 4 1 1 1 2 3 1 2 1 4 2 1 1 1 3 3 4 1 4 2 4 4 3  
## [149] 1 1 1 1 4 4 1 2 2 4 1 4 4 4 1 1 1 1 1 1 1 1 3 1 1 4 1 1 1 4 4 1 1 1 1 2 1  
## [186] 1 3 3 1 2 3 2 1 1 1 1 1 1 2 1 1 3 1 3 4 1 3 1 1 1 1 3 1 1 1 2 1 2 4 2 3 1  
## [223] 4 2 1 1 1 1 2 1 4 1 4 1 4 4 4 4 4 2 1 1 3 1 4 1 4 2 2 3 1 2 1 1 1 1 1 4 1  
## [260] 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 1 1 1 1 2 1 2 1 3 3 1  
## [297] 4 1 1 2 1 1 3 1 2 1 1 1 1 2 2 1 1 2 1 1 1 3 1 2 4 2 1 1 1 4 1 4 1 1 1 1 1  
## [334] 1 1 3 2 1 1 1 1 1 3 2 1 1 1 2 1 1 2 2 2 1 2 1 2 1 2 3 2 2 3 2 1 2 3 1 2 3  
## [371] 1 2 1 1 4 2 1 2 1 1 2 2 1 2 1 2 1 1 1 1 2 2 2 2 1 2 2 1 2 1 2 2 1 1 1 2 2  
## [408] 1 1 1 1 2 2 1 2 2 1 2 1 1 1 1 2 1 1 1 1 2 2 1 1 3 1 2 2 1 2 1 1 2 1 1 1 2  
## [445] 2 1 1 1 1 3 2 1 1 1 1 1 2 1 2 2 1 3 1 2 1 1 1 1 1 2 2 2 1 2 1 1 1 2 2 2 2  
## [482] 1 1 2 2 1 2 1 2 2 1 1 2 1 2 2 2 2 1 1 4 2 1 1 2 2 1 1 2 1 2 3 1 2 1 1 2 1  
## [519] 2 1 1 1 1 3 1 1 1 1 1 1 1 4 1 1 1 1 4 1 1 1 1 2 1 2 1 1 1 1 1 2 1 1 2 1 1  
## [556] 1 1 1 2 2 1 1 2 1 1 3 1 2 1 2 1 1 2 2 1 4 2 2 2 1 1 2 1 1 1 1 1 1 1 1 1 2  
## [593] 1 1 1 1 2 1 1 1

fviz\_nbclust(Loyality2\_normalized, kmeans, method = 'wss') +  
 geom\_vline(xintercept = 3, linetype = 2)+  
 labs(subtitle = 'Elbow method')



fviz\_nbclust(Loyality2\_normalized, kmeans, method = "silhouette")+  
 labs(subtitle = "Silhouette method")



# By considering the value of k=3 .

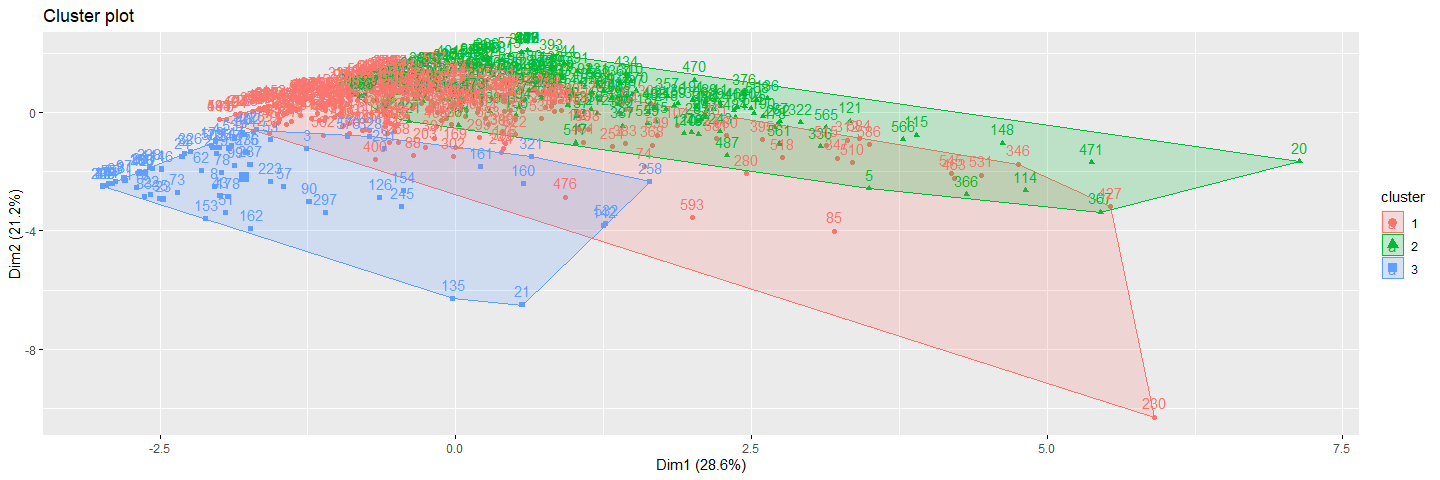
### Now we will run Kmeans with k=3 and nstart = 30 and plot the clusters using fviz\_cluster.

### store the centers of the model in Output, print the size of the 3 clusters and visualize it.

#Generating 3 clusters based on the available dataset  
  
clust2<- kmeans(Loyality2\_normalized,3,nstart = 30)  
Loyality2$cluster <- clust2$cluster  
head(Loyality2)

## PropCat.5 PropCat.6 PropCat.7 PropCat.8 PropCat.9 PropCat.10 PropCat.11  
## 1 0.50 0.00 0.00 0.00 0.00 0 0.00  
## 2 0.46 0.35 0.03 0.02 0.01 0 0.06  
## 3 0.24 0.12 0.03 0.01 0.01 0 0.00  
## 4 0.40 0.00 0.00 0.00 0.00 0 0.00  
## 5 0.81 0.00 0.00 0.05 0.00 0 0.00  
## 6 0.49 0.10 0.00 0.01 0.07 0 0.00  
## PropCat.12 PropCat.13 PropCat.14 PropCat.15 cluster  
## 1 0.03 0 0.13 0.34 1  
## 2 0.00 0 0.08 0.00 1  
## 3 0.02 0 0.56 0.00 3  
## 4 0.00 0 0.60 0.00 3  
## 5 0.00 0 0.14 0.00 2  
## 6 0.00 0 0.07 0.27 2

# Vizualizing Scatterplot for the k=3 clusters   
  
fviz\_cluster(clust2, Loyality2\_normalized)



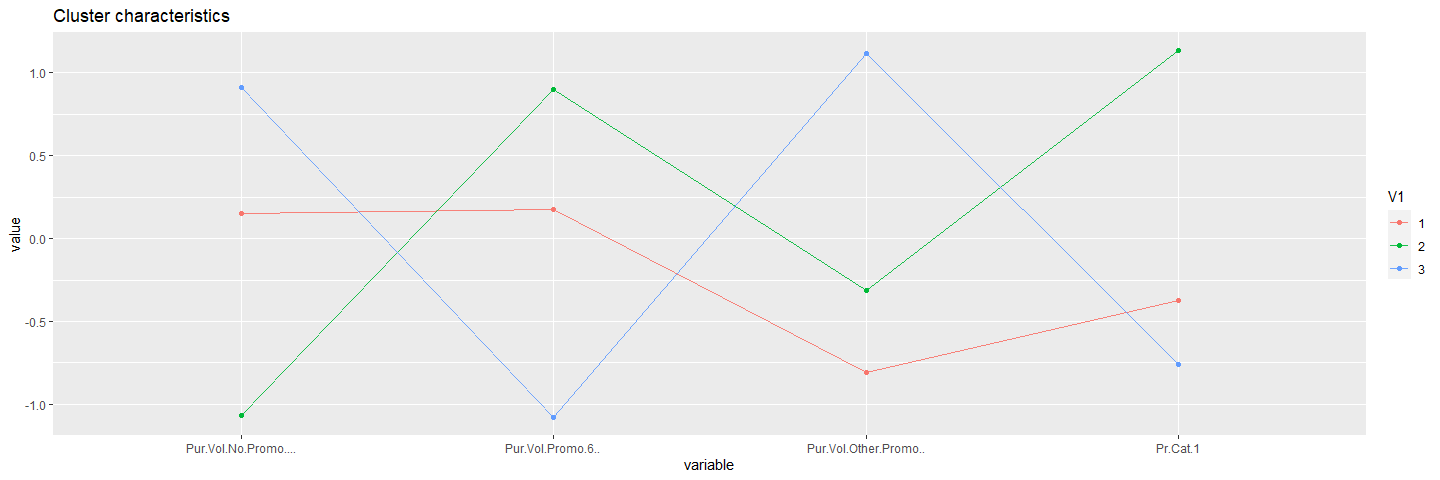
Output2<- as.data.frame(cbind(1:nrow(clust2$centers),clust2$centers))  
Output2$V1<- as.factor(Output2$V1)  
clust2$size

## [1] 321 205 74

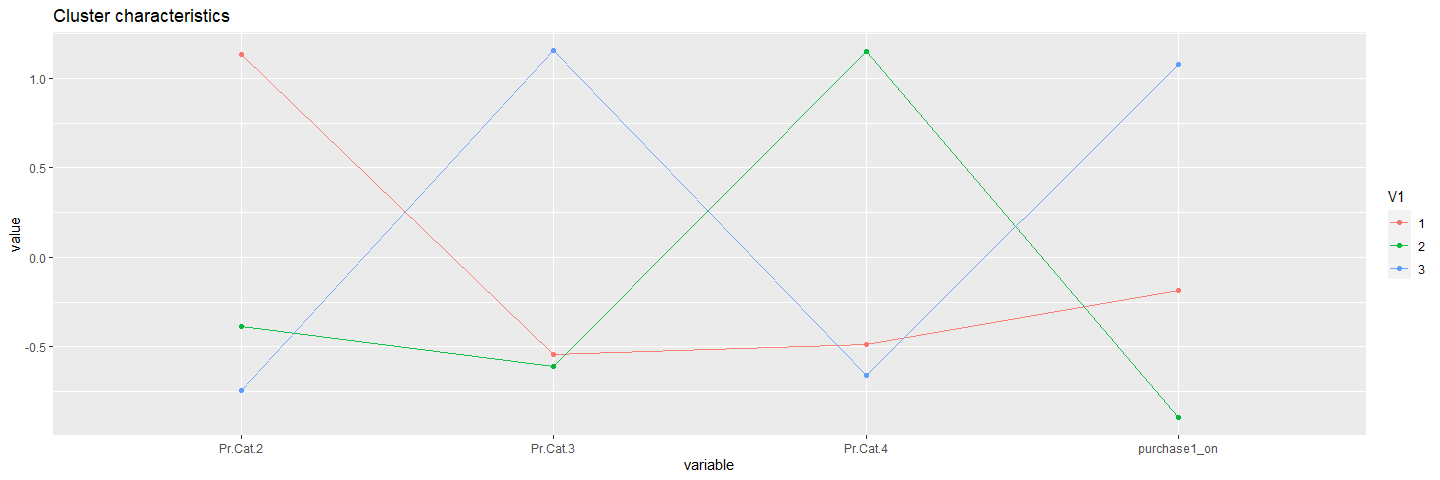
head(Output2)

## V1 Pur.Vol.No.Promo.... Pur.Vol.Promo.6.. Pur.Vol.Other.Promo.. Pr.Cat.1  
## 1 1 0.006545016 -0.003720882 -0.0039774143 -0.1214132  
## 2 2 -0.018171951 0.025594588 0.0008760163 0.2702850  
## 3 3 0.021950000 -0.054763345 0.0148265766 -0.2220914  
## Pr.Cat.2 Pr.Cat.3 Pr.Cat.4 purchase1\_on  
## 1 0.2471883 -0.0800433 -0.04561153 0.02464361  
## 2 -0.2532167 -0.1107455 0.09322033 -0.17959512  
## 3 -0.3707842 0.6540099 -0.06039009 0.39062703

ggparcoord(Output2,   
 columns = 2:5, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.9)



ggparcoord(Output2,   
 columns = 6:9, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.9)



Pur.Vol.No.Promo Pur.Vol.Promo.6 Pur.Vol.Other.Promo Pr.Cat.1 Pr.Cat.2 Pr.Cat.3 Pr.Cat.4 purchase1\_on 0.006545016 -0.003720882 -0.0039774143 -0.1214132 0.2471883 -0.0800433 -0.04561153 0.02464361 -0.018171951 0.025594588 0.0008760163 0.2702850 -0.2532167 -0.1107455 0.09322033 -0.17959512 0.021950000 -0.054763345 0.0148265766 -0.2220914 -0.3707842 0.6540099 -0.06039009 0.39062703

* cluster3: The behavior of Customers in this cluster is that they purchase products from a single price category(pr.cat 4 and pr.cat 1). They purchase based on the promotional(Pur.Vol.Promo 6) and they doesnt buy when there is no promo. We could periodically send the discount offers by email or show the message
* Cluster1: The behavior of Customers purchase products from a single price category(pr.cat 3). Their purchases are affected based on the promotional offers.They purchase products of a specific price category mostly.Customers in this cluster have a high brand loyalty.
* cluster2: The behavior of Customers in this cluster evidently purchase products from a single price category(pr.cat 2). They purchase almost similarly both during price offs and no price offers. We could periodically send the discount offers by email or show the message.

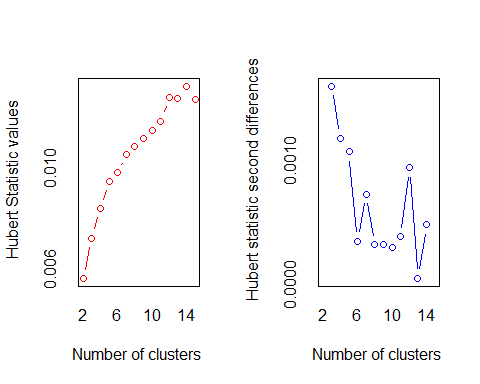
# Q1. Use k-means clustering to identify clusters of households based on:

##(c). By taking the variables that describe the purchase behavior and basis of purchase and forming the cluster.

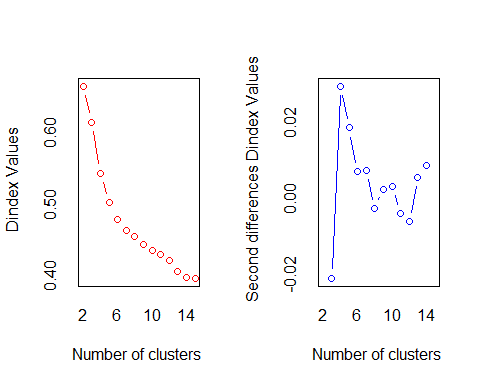
# Now considering the different variables that describes the purchases behavior based on purchase and forming the cluster.  
#Clusters based on all of the above variables.  
  
Loyality3<- Soapdata[,c(12:22,31:35,49)]  
Loyality3\_normalized <- data.frame(lapply(Loyality3, Normalize))  
Loyality3\_normalized<- na.omit(Loyality3\_normalized)

# Plotting the clusters using Elbow, silhoutte method.

NbClust(data = Loyality3\_normalized, diss = NULL, distance = "euclidean",  
 min.nc = 2, max.nc = 15, method = "kmeans")



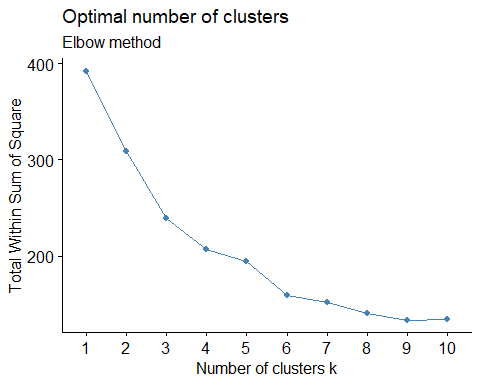
## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



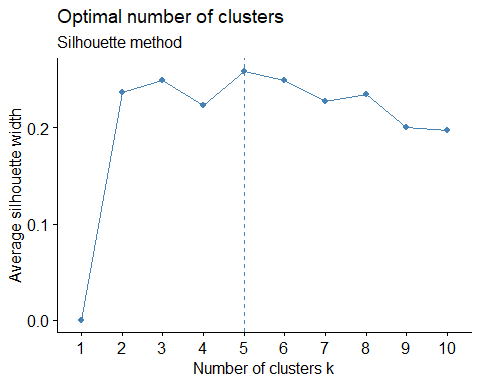
## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 5 proposed 2 as the best number of clusters   
## \* 2 proposed 3 as the best number of clusters   
## \* 5 proposed 4 as the best number of clusters   
## \* 6 proposed 5 as the best number of clusters   
## \* 1 proposed 8 as the best number of clusters   
## \* 1 proposed 12 as the best number of clusters   
## \* 2 proposed 14 as the best number of clusters   
## \* 1 proposed 15 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 5   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## $All.index  
## KL CH Hartigan CCC Scott Marriot TrCovW TraceW  
## 2 2.2266 190.7424 96.4872 -2.2198 809.3644 299235295 867.4687 297.0997  
## 3 0.6264 158.7370 148.1321 -4.0656 1479.6969 220288849 625.6659 255.8228  
## 4 1.6613 181.1561 102.7473 5.3091 2491.5839 72517324 342.8575 204.9652  
## 5 2.0909 184.6663 57.1989 12.5600 3402.4356 24829350 242.3155 174.8261  
## 6 1.6484 173.0834 38.8471 15.8580 3839.6037 17254209 199.6509 159.4936  
## 7 1.5585 159.8741 27.8135 17.5050 4190.3829 13088387 166.4043 149.7031  
## 8 0.9508 147.1870 28.2117 17.7412 4477.7604 10589106 153.1871 142.9962  
## 9 1.1146 138.2186 25.7640 15.3764 4660.0373 9890720 135.5047 136.4917  
## 10 2.6535 130.8578 13.6456 12.0920 4879.5689 8469165 123.7626 130.7900  
## 11 0.4004 121.6539 24.0049 7.6363 5128.1220 6772005 120.5154 127.8335  
## 12 0.4126 117.0849 52.3717 2.9032 5521.5106 4183600 112.4120 122.8276  
## 13 1.7573 121.0454 32.9608 9.0353 5724.7934 3498915 84.6319 112.7824  
## 14 5.5295 120.3383 -0.2016 12.3922 6000.0263 2564972 74.1518 106.7862  
## 15 0.1727 111.4993 34.6885 10.8752 5999.4304 2947408 73.7567 106.8229  
## Friedman Rubin Cindex DB Silhouette Duda Pseudot2 Beale Ratkowsky  
## 2 2361.838 1.3190 0.3678 1.7343 0.2347 1.0736 -19.6141 -0.8036 0.2181  
## 3 7236.922 1.5318 0.3593 1.6931 0.2038 0.7827 66.0902 3.2483 0.2334  
## 4 12343.275 1.9119 0.3377 1.5310 0.2284 1.4874 -76.0217 -3.8335 0.2299  
## 5 14915.908 2.2415 0.3265 1.3492 0.2586 1.4288 -49.2186 -3.5018 0.2345  
## 6 15157.637 2.4569 0.3097 1.4411 0.2496 1.3845 -50.2663 -3.2399 0.2323  
## 7 16101.753 2.6176 0.3164 1.5450 0.2325 2.2693 -118.0212 -6.5147 0.2257  
## 8 16749.433 2.7404 0.3087 1.4797 0.2285 2.0795 -82.5408 -6.0326 0.2239  
## 9 16868.507 2.8710 0.3003 1.5028 0.2158 1.6325 -56.5693 -4.5102 0.2175  
## 10 16879.159 2.9961 0.3283 1.4982 0.2125 1.2370 -19.7322 -2.2276 0.2159  
## 11 17112.207 3.0654 0.3491 1.5661 0.1988 1.5453 -41.2886 -4.1028 0.2076  
## 12 17500.908 3.1904 0.2645 1.5843 0.1806 1.8998 -56.8369 -5.4982 0.1992  
## 13 17489.539 3.4745 0.2726 1.5386 0.1870 1.3510 -25.1998 -3.0169 0.1925  
## 14 17191.791 3.6696 0.3213 1.4994 0.2019 2.2439 -78.1616 -6.4448 0.1927  
## 15 17292.952 3.6684 0.3246 1.4892 0.2039 1.1710 -14.1675 -1.6961 0.1875  
## Ball Ptbiserial Frey McClain Dunn Hubert SDindex Dindex SDbw  
## 2 148.5499 0.3901 0.3575 0.6674 0.0745 0.0056 6.0703 0.6649 1.8694  
## 3 85.2743 0.4380 0.2974 1.1177 0.0726 0.0072 5.8694 0.6147 1.0966  
## 4 51.2413 0.4629 -0.0591 1.5937 0.0661 0.0084 4.7508 0.5435 0.5049  
## 5 34.9652 0.5213 0.9542 1.7335 0.1051 0.0095 4.2136 0.5019 0.4275  
## 6 26.5823 0.4643 0.5758 2.5988 0.0989 0.0098 4.8871 0.4789 0.3929  
## 7 21.3862 0.4370 0.1592 3.2554 0.0957 0.0106 5.0445 0.4630 0.3855  
## 8 17.8745 0.4376 0.6063 3.4193 0.0710 0.0109 4.7941 0.4546 0.3799  
## 9 15.1657 0.4130 0.0910 4.0702 0.0721 0.0112 5.3293 0.4436 0.3653  
## 10 13.0790 0.4162 0.3980 4.2854 0.0818 0.0115 5.2519 0.4352 0.3729  
## 11 11.6212 0.4111 0.2853 4.4597 0.0877 0.0119 5.4184 0.4300 0.3641  
## 12 10.2356 0.4056 0.1828 4.6949 0.0776 0.0128 5.8030 0.4209 0.3603  
## 13 8.6756 0.4005 -0.0564 5.0325 0.0933 0.0128 5.2279 0.4058 0.2966  
## 14 7.6276 0.4128 -0.7851 4.9375 0.0878 0.0133 5.5432 0.3964 0.2947  
## 15 7.1215 0.3935 0.2519 5.3941 0.0878 0.0128 5.4015 0.3958 0.2938  
##   
## $All.CriticalValues  
## CritValue\_Duda CritValue\_PseudoT2 Fvalue\_Beale  
## 2 0.8915 34.8220 1  
## 3 0.8789 32.7892 0  
## 4 0.8797 31.7361 1  
## 5 0.8635 25.9185 1  
## 6 0.8627 28.8135 1  
## 7 0.8538 36.1312 1  
## 8 0.8428 29.6572 1  
## 9 0.8509 25.5868 1  
## 10 0.8457 18.7948 1  
## 11 0.8450 21.4644 1  
## 12 0.8380 23.1915 1  
## 13 0.8397 18.5191 1  
## 14 0.8450 25.8673 1  
## 15 0.8397 18.5191 1  
##   
## $Best.nc  
## KL CH Hartigan CCC Scott Marriot TrCovW  
## Number\_clusters 14.0000 2.0000 3.000 8.0000 4.000 4 4.0000  
## Value\_Index 5.5295 190.7424 51.645 17.7412 1011.887 100083552 282.8084  
## TraceW Friedman Rubin Cindex DB Silhouette Duda  
## Number\_clusters 4.0000 4.000 14.0000 12.0000 5.0000 5.0000 2.0000  
## Value\_Index 20.7184 5106.353 -0.1964 0.2645 1.3492 0.2586 1.0736  
## PseudoT2 Beale Ratkowsky Ball PtBiserial Frey McClain  
## Number\_clusters 2.0000 2.0000 5.0000 3.0000 5.0000 1 2.0000  
## Value\_Index -19.6141 -0.8036 0.2345 63.2756 0.5213 NA 0.6674  
## Dunn Hubert SDindex Dindex SDbw  
## Number\_clusters 5.0000 0 5.0000 0 15.0000  
## Value\_Index 0.1051 0 4.2136 0 0.2938  
##   
## $Best.partition  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 3 3 4 4 2 3 3 4 3 5 2 3 2 1 3 3 3 1 3 5   
## 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40   
## 4 4 4 4 4 3 1 1 2 4 4 2 4 4 4 2 2 2 4 4   
## 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60   
## 3 4 4 4 2 1 2 1 4 5 4 1 4 3 4 1 4 2 4 3   
## 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80   
## 3 4 4 1 3 4 3 3 2 1 1 2 4 3 3 3 3 4 3 2   
## 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100   
## 3 4 4 5 3 1 2 1 5 4 5 3 4 5 3 3 4 3 4 1   
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120   
## 3 3 2 3 2 3 5 3 3 2 4 3 3 2 2 5 3 4 3 3   
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140   
## 2 3 1 3 3 4 1 1 3 5 2 3 5 3 4 5 3 3 1 2   
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160   
## 2 4 3 4 5 4 4 2 3 1 1 1 4 4 3 5 5 4 3 4   
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180   
## 4 4 1 3 1 3 3 3 1 3 2 3 3 4 1 1 1 4 4 3   
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200   
## 3 3 3 3 3 1 2 3 1 5 2 3 1 1 3 3 3 3 5 1   
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220   
## 1 2 3 2 4 3 2 3 1 3 1 2 3 1 3 5 1 5 4 5   
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240   
## 2 1 4 5 3 1 1 3 5 1 4 3 4 1 4 4 4 4 4 5   
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260   
## 3 3 2 1 4 3 4 5 5 2 3 3 3 3 3 1 1 4 1 3   
## 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280   
## 5 3 3 1 3 1 3 1 3 5 3 3 3 3 3 3 3 3 1 2   
## 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300   
## 1 2 2 1 3 3 3 1 3 5 3 5 1 2 2 3 4 3 1 5   
## 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320   
## 3 1 2 3 5 1 3 3 3 5 5 3 3 5 3 3 3 2 3 5   
## 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340   
## 3 5 1 3 3 4 3 4 3 3 3 1 3 3 3 2 3 3 1 3   
## 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360   
## 1 1 2 5 3 3 3 3 3 3 1 5 3 1 5 3 3 3 5 3   
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380   
## 5 1 2 5 3 5 2 3 5 2 3 5 1 1 4 5 3 5 3 1   
## 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400   
## 5 5 3 5 3 5 1 3 3 3 5 5 5 5 3 5 5 3 5 1   
## 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420   
## 5 5 1 3 3 5 5 1 3 3 3 5 5 1 5 5 3 5 1 3   
## 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440   
## 3 3 5 1 3 3 1 5 5 1 1 2 3 5 5 1 5 3 1 5   
## 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460   
## 1 3 3 5 5 3 1 3 1 3 5 1 1 3 1 3 5 3 3 5   
## 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480   
## 3 2 3 5 3 3 3 1 1 5 5 5 3 3 3 1 3 5 5 5   
## 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500   
## 5 3 3 5 5 1 5 3 5 5 1 3 5 1 5 5 5 5 3 3   
## 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520   
## 4 3 3 1 5 5 1 3 3 3 5 2 1 5 3 1 5 3 5 3   
## 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540   
## 1 1 1 2 1 1 3 3 3 1 3 4 3 1 3 1 4 1 1 1   
## 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560   
## 3 3 3 3 3 3 3 1 3 5 1 3 5 3 3 3 3 3 5 5   
## 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580   
## 3 1 5 3 3 2 3 5 1 5 3 3 5 3 3 4 5 5 5 3   
## 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600   
## 3 5 1 3 1 3 3 3 3 3 3 5 1 3 1 1 5 3 3 1

fviz\_nbclust(Loyality3\_normalized, kmeans, method = 'wss') +  
 geom\_vline(xintercept = NULL, linetype = 4)+  
 labs(subtitle = 'Elbow method')



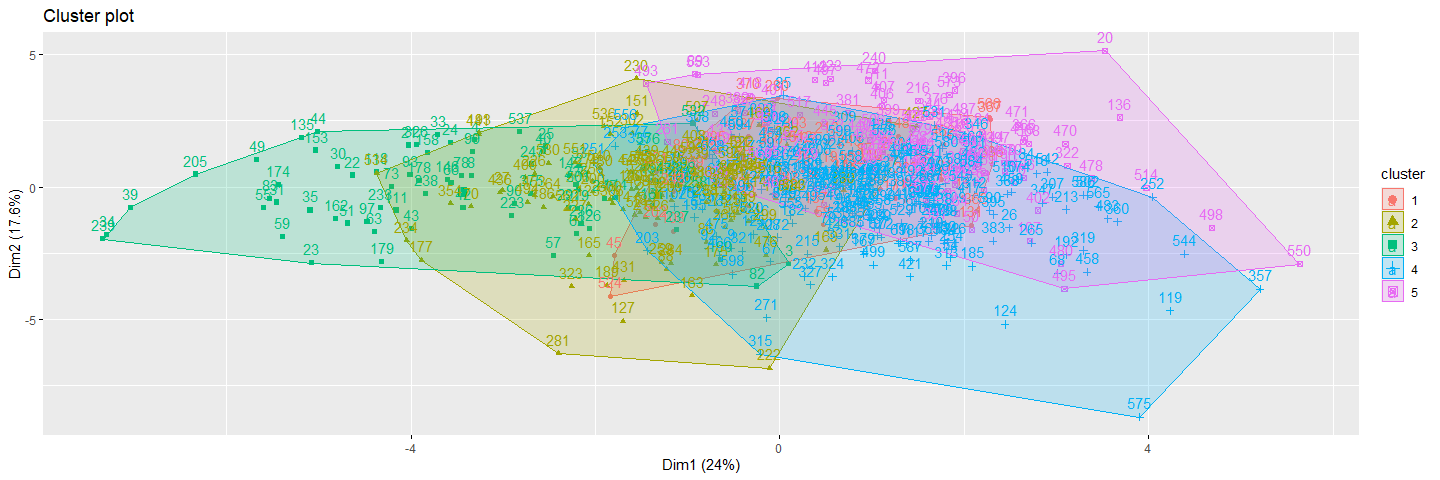
fviz\_nbclust(Loyality3\_normalized, kmeans, method = "silhouette")+  
 labs(subtitle = "Silhouette method")



##Plotting the visualization using Elbow method, silhoutte to find the best number of clusters.

##According to the majority rule, the best number of clusters is 5 silhouette = 5 Elbow = 4 Nbclust = 5

clust3<- kmeans(Loyality3\_normalized, 5,nstart = 50)  
fviz\_cluster(clust3, Loyality3\_normalized)

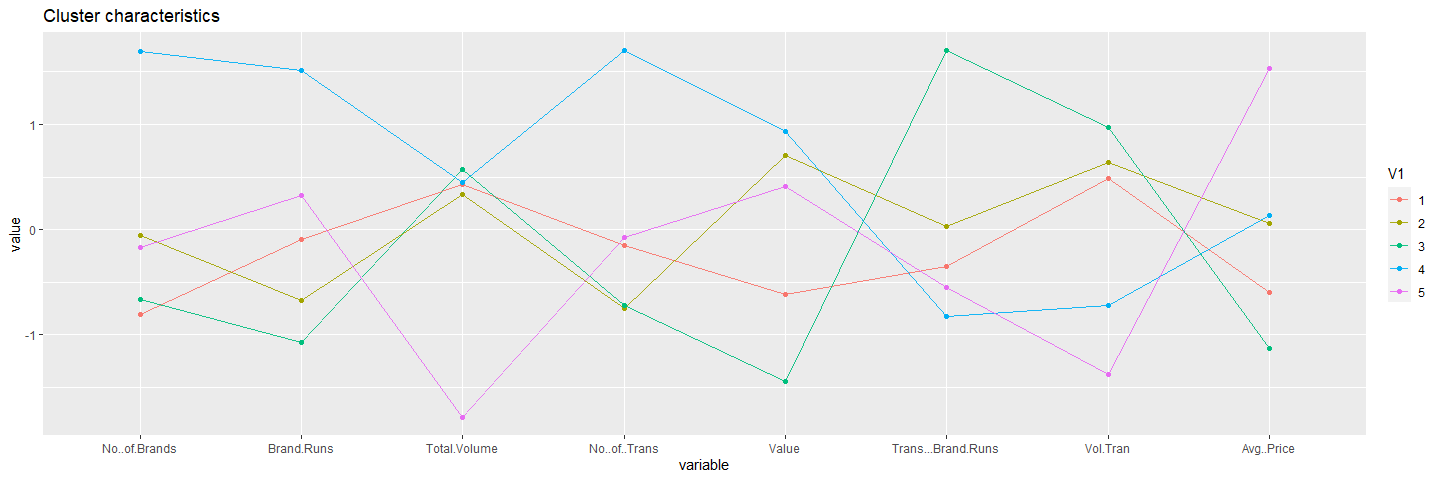


Output3<- as.data.frame(cbind(1:nrow(clust3$centers),clust3$centers))  
Output3$V1<- as.factor(Output3$V1)  
clust3$size

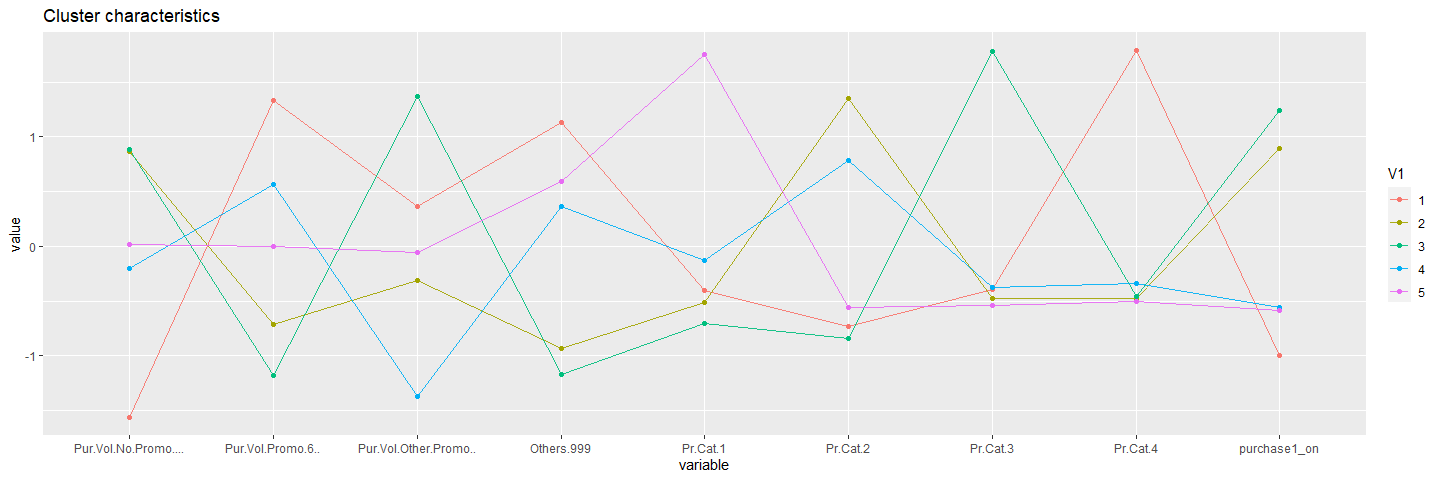
## [1] 52 117 73 241 117

## Visualizing the cluster

ggparcoord(Output3,   
 columns = 2:9, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.9)



ggparcoord(Output3,   
 columns = 10:18, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.9)



* Cluster1: The behavior of Customers in this cluster evidently purchase products from a single price category(pr.cat 4) and with other999 brands, they purchase based on promotional(promo 6).We could periodically send the discount offers by email or show the message.
* Cluster4: They are least brand loyal customers.They are neither least nor highest in other characteristics when compared to other clusters but they have the highest no of transactions and brand runs.
* Cluster3: The cluster has least number of brands, brand runs,highest transaction brand runs and they buy least from other999. They highly purchase product from single category pr.cat 3 when other promo is available. Brand loyal.We could periodically send the discount offers by email or show the message when promo is available.
* Cluster2: The cluster has moderate transactions and They buy products from Pr.Cat 2 and they are brand loyal. They buy products even though with No promos available.
* Cluster5: This cluster have least total volume of transactions, high Avg.price and highest peak in brand loyality (pr.cat1)

# Comparing the Cluster Sizes.

clust1$size

## [1] 283 317

clust2$size

## [1] 321 205 74

clust3$size

## [1] 52 117 73 241 117

#How should K be chosen?

Ans) The value of ‘K’ can be choosen based on below: >>The intra-cluster distances are minimum in all clusters >>The clusters are well apart. That is, the inter cluster distances are maximum.

* In all above segmentation, we observe that for k= 3, distance within clusters is minimum and distance between clusters is maximum. we conclude that K-means algorithm with K=3 is the best model.

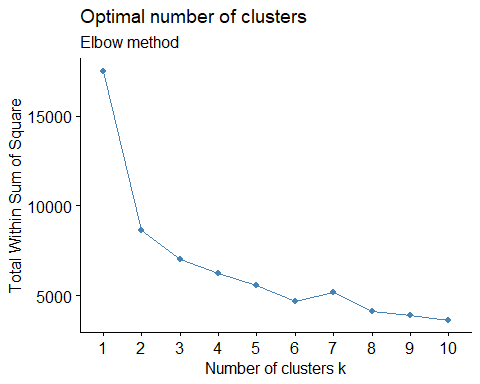
#How should the percentages of total purchases comprised by various brands be treated? Isn’t a customer who #buys all brand A just as loyal as a customer who buys all brand B? What will be the effect on any distance #measure of using the brand share variable as is?

* The percentages of total purchases should not be considered individually as they increase the inter cluster distances and the effectiveness of the clustering drops. Instead, consider MaxBrCode(Max proportion of purchase) which give the brand loyalty of the customer.

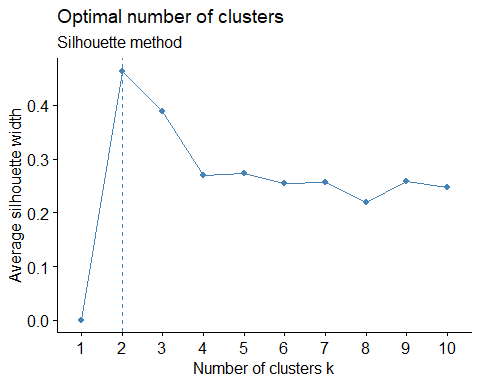
# 2Q)

##Adding demographics to describe the purchase behaviour variables.(which includes such as gender, age, familial and marital status and education)

Loyality4<- Soapdata[,c(2:11,12:19,31:35,47)]  
Loyality4\_normalized <- data.frame(Soapdata[,2:10],lapply(Loyality4[,11:19], Normalize))  
  
fviz\_nbclust(Loyality4\_normalized, kmeans, method = 'wss') +  
 geom\_vline(xintercept = NULL, linetype = 4)+  
 labs(subtitle = 'Elbow method')



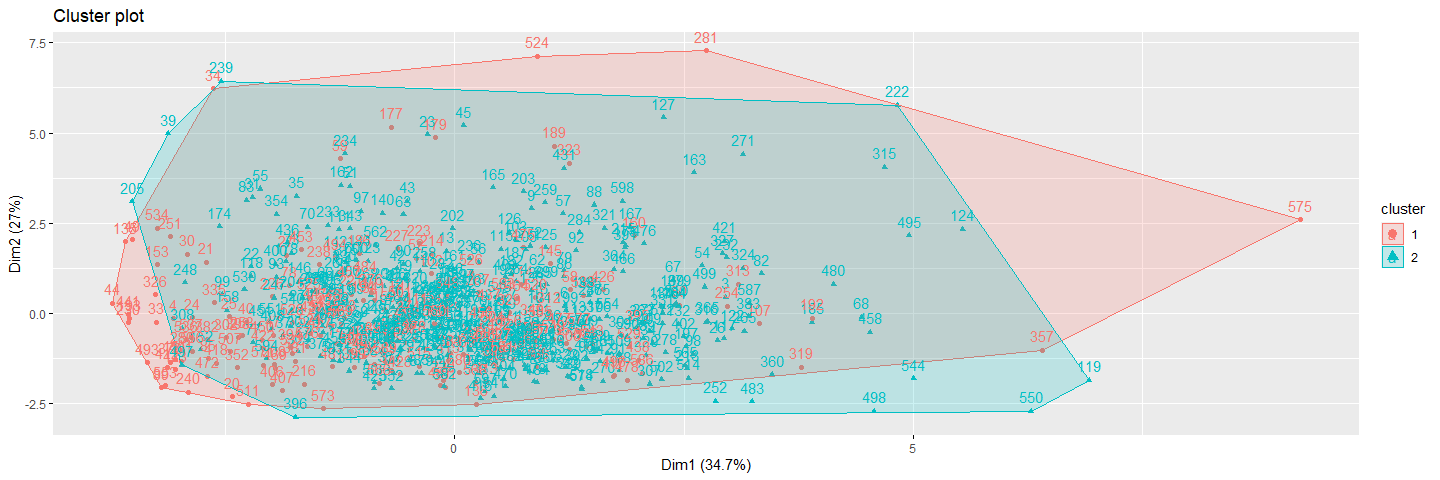
fviz\_nbclust(Loyality4\_normalized, kmeans, method = "silhouette")+  
 labs(subtitle = "Silhouette method")



### According to the majority rule, the best number of clusters is 2.

Elbow 2 Silhoutte 2

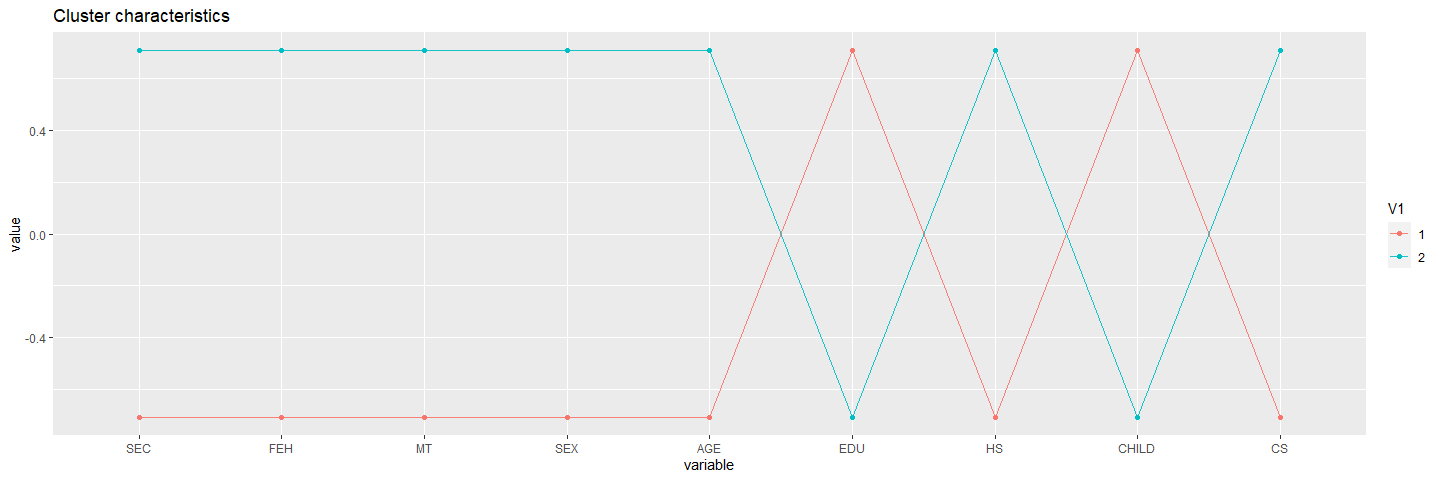
clust4<- kmeans(Loyality4\_normalized, 2,nstart = 50)  
fviz\_cluster(clust4, Loyality4\_normalized[,10:18])



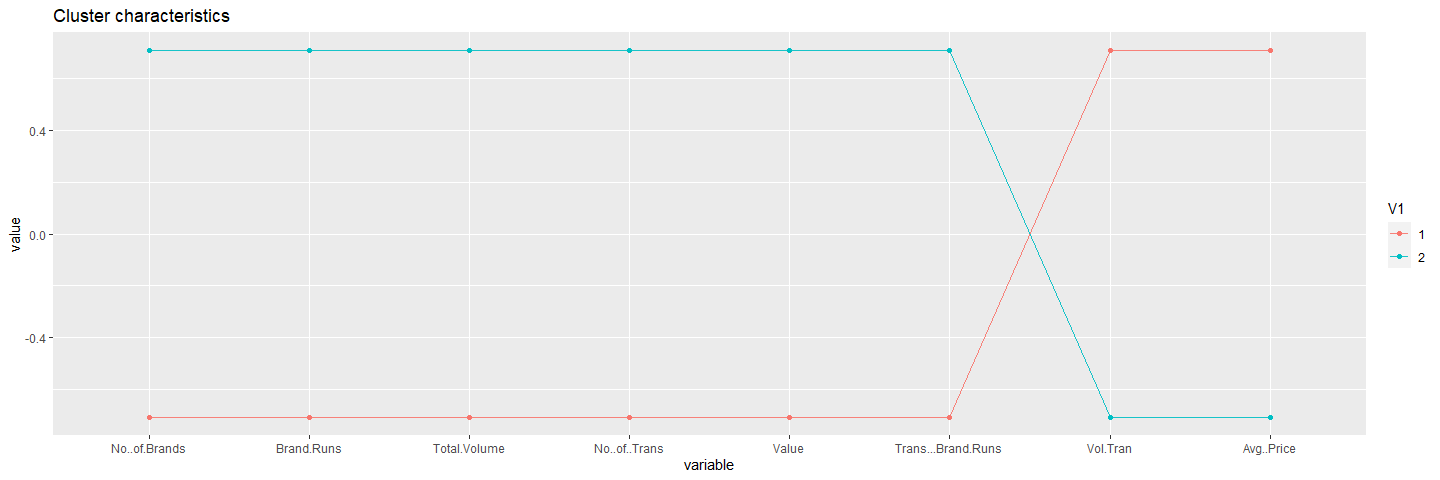
Output4<- as.data.frame(cbind(1:nrow(clust4$centers),clust4$centers))  
Output4$V1<- as.factor(Output4$V1)  
clust4$size

## [1] 195 405

ggparcoord(Output4,   
 columns = 2:10, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.9)



ggparcoord(Output4,   
 columns = 11:18, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.9)

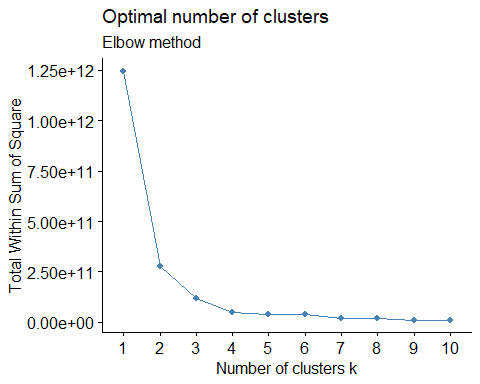


### We have considered three criteria to choose K:

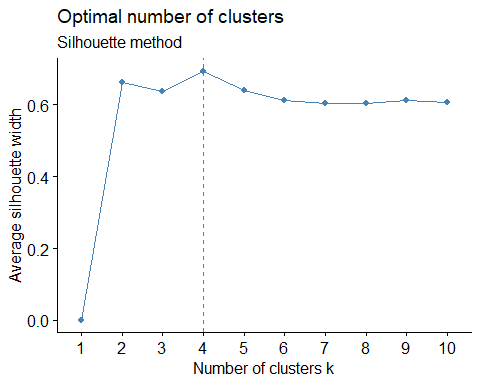
* Minimum distance within cluster
* Maximum distance between clusters
* Information from centroid plot of clusters

## Forming clusters by using all variables

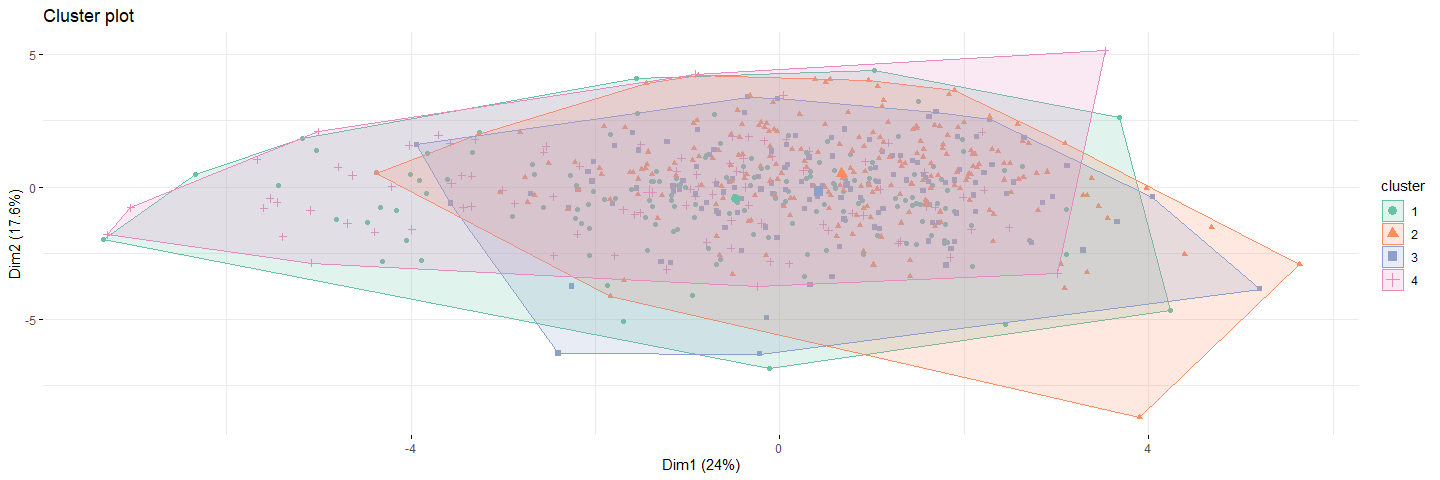
## adding the demographics to the basis of purchase variables  
  
Loyality5<- Soapdata[,c(2:11,12:22,31:35,47,49)]  
Loyality5\_normalized <- data.frame(Soapdata[,1:10],lapply(Loyality5[,11:27],Normalize))  
  
  
fviz\_nbclust(Loyality5\_normalized, kmeans, method = 'wss') +  
 geom\_vline(xintercept = NULL, linetype = 4)+  
 labs(subtitle = 'Elbow method')



fviz\_nbclust(Loyality5\_normalized, kmeans, method = "silhouette")+  
 labs(subtitle = "Silhouette method")

 ## According to the majority rule, the best number of clusters is 4 silhoutte = 4 Elbow = 2

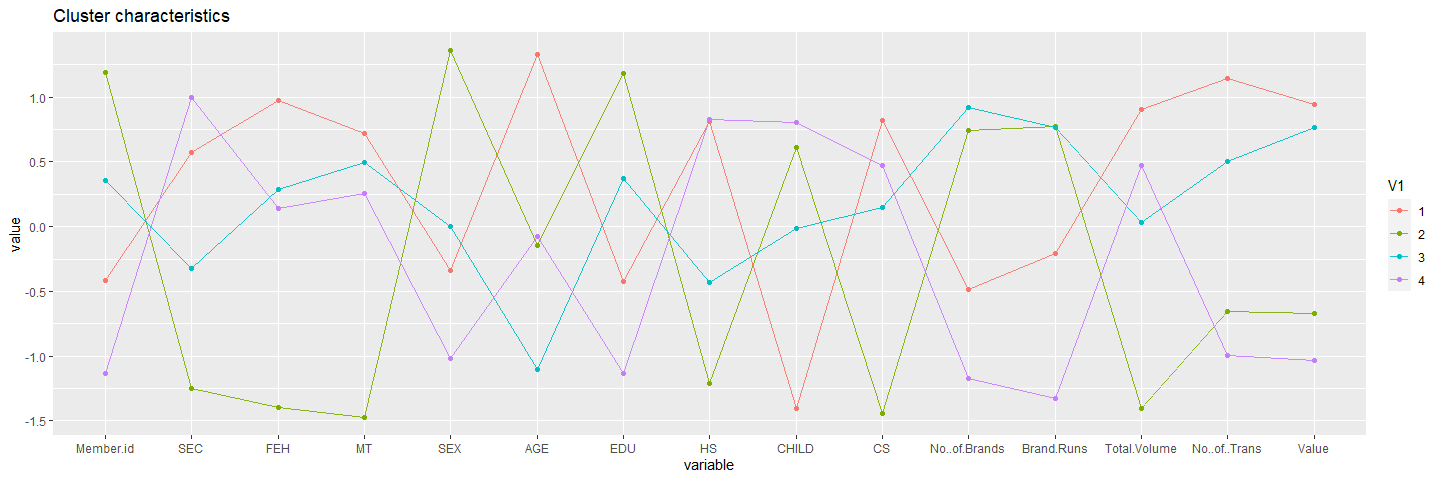
clust5<- kmeans(Loyality5\_normalized, 4, nstart = 50)  
fviz\_cluster(clust5,Loyality5\_normalized[,11:27], palette = "Set2", ggtheme = theme\_minimal(), geom = "point")



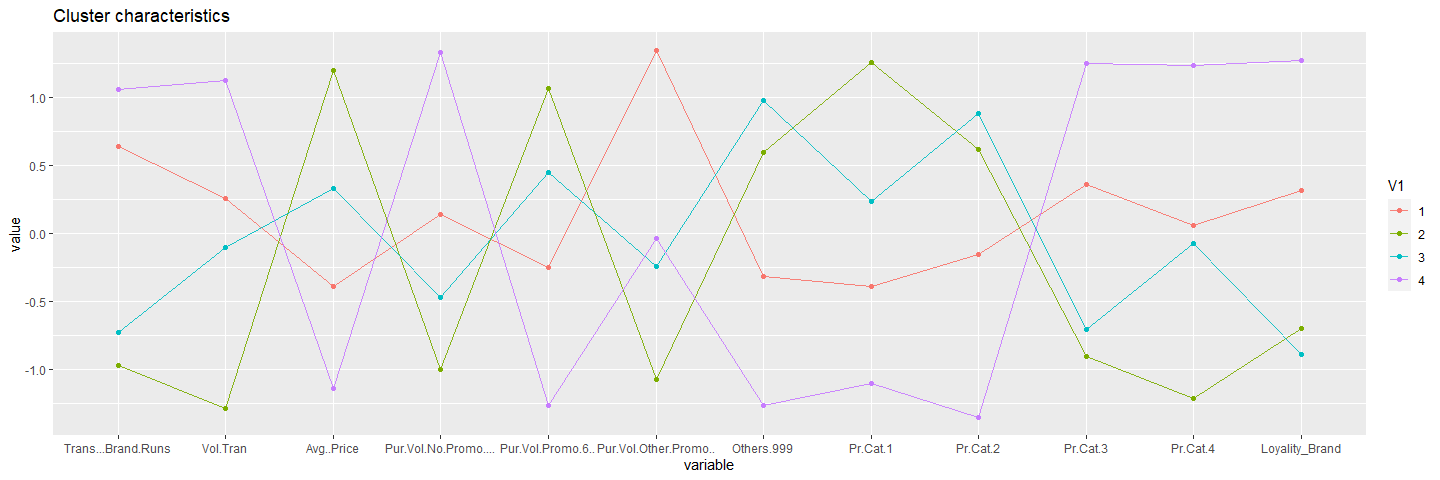
clust5$size

## [1] 158 223 128 91

Output5<- as.data.frame(cbind(1:nrow(clust5$centers),clust5$centers))  
Output5$V1<- as.factor(Output5$V1)  
ggparcoord(Output5,   
 columns = 2:16, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.9)



ggparcoord(Output5,   
 columns = 17:28, groupColumn = 1, showPoints = TRUE,  
 title = "Cluster characteristics", alphaLines = 0.9)



clust4$size

## [1] 195 405

clust5$size

## [1] 158 223 128 91

### Q2. Select what you think is the best segmentation and comment on the characteristics (demographic, brand loyalty, and basis for purchase) of these clusters. (This information would be used to guide the development of advertising and promotional campaigns.)

Cluster 3(n=158): have the high value of CS (Television Availability), Number of transactions, Total volume and value are high.We can easily promote the product through advertisement. the purchase is high during the promo and they are not brand loyal as they are buying products from different categories.

cluster 1(n=91): They are brand loyal. They are highly buying products which fall under category 3 and 4.The Purchase is high irrespective of the promotions.The volume transactions are high too.

cluster 2(n=128): Customers buying more products from other999, we can say least loyal.They have the highest number of brands purchased. Number of instances of consecutive purchase of brands is high so the number of transaction is also high.

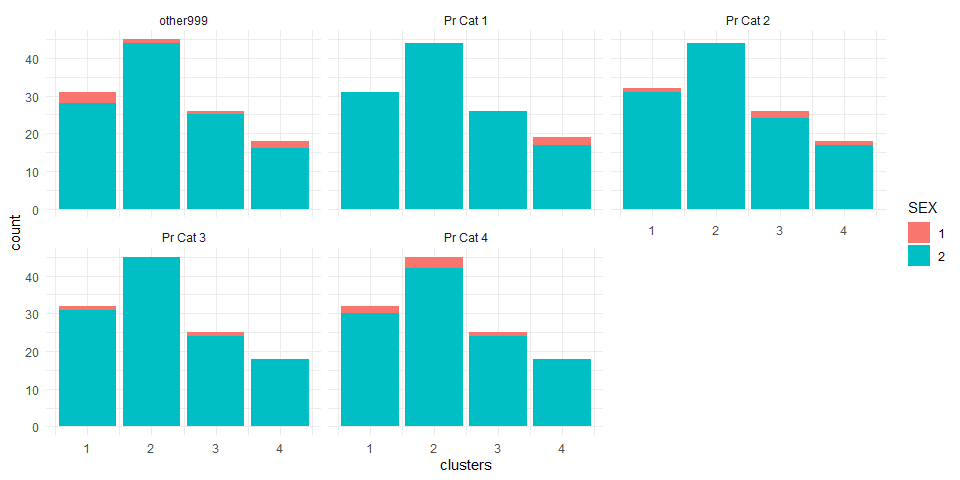
Cluster 4(n=223): They are loyal to brand(pr.cat 1), they tend to buy more during the promotion code 6 is on.The SEC is low.Cluster 2 customers have a higher degree of House hold members but low availability of Television.

# Q3. Develop a model that classifies the data into these segments. Since this information would most likely be used in targeting direct-mail promotions, it would be useful to select a market segment that would be defined as a success in the classification model.

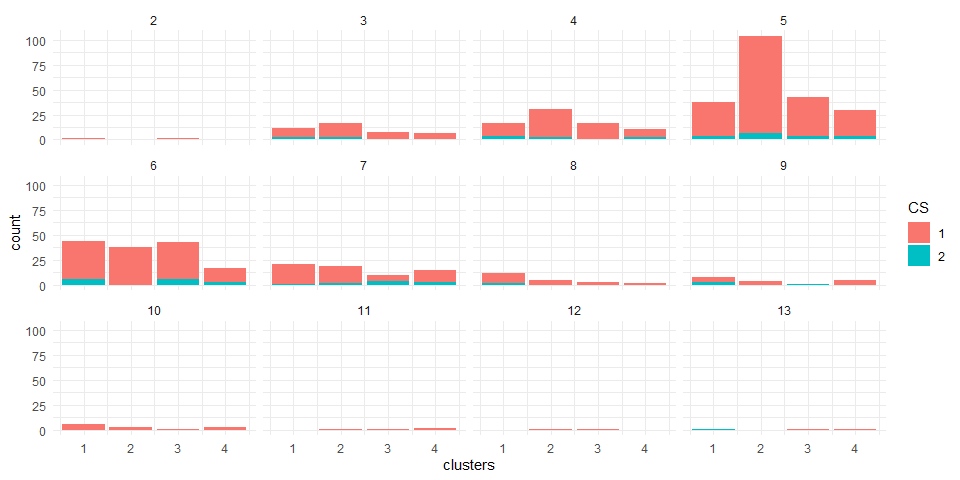
final\_data<-Soapdata[,23:31]  
Soapdata$Loyality<-as.numeric(apply(final\_data,1,which.max))  
  
Loyality6 <- Soapdata[,c(2:11,19,20:22,31:35,47,48,50)]  
Loyality6$clusters <- clust5$cluster  
head(Loyality6)

## SEC FEH MT SEX AGE EDU HS CHILD CS Affluence.Index Avg..Price  
## 1 4 3 10 1 4 4 2 4 1 2 10.19  
## 2 3 2 10 2 2 4 4 2 1 19 12.03  
## 3 2 3 10 2 4 5 6 4 1 23 8.44  
## 4 4 0 0 2 4 5 4 5 1 0 7.60  
## 5 4 1 10 2 3 4 4 3 1 10 7.12  
## 6 4 3 10 2 3 4 5 2 1 13 9.38  
## Pur.Vol.No.Promo.... Pur.Vol.Promo.6.. Pur.Vol.Other.Promo.. Others.999  
## 1 1.00 0.00 0.00 0.492  
## 2 0.89 0.10 0.02 0.699  
## 3 0.94 0.02 0.04 0.379  
## 4 1.00 0.00 0.00 0.000  
## 5 0.61 0.14 0.24 0.807  
## 6 1.00 0.00 0.00 0.857  
## Pr.Cat.1 Pr.Cat.2 Pr.Cat.3 Pr.Cat.4 Loyality\_Brand purchase\_on Loyality  
## 1 0.23 0.56 0.13 0.07 0.38 1 9  
## 2 0.29 0.55 0.09 0.06 0.14 1 9  
## 3 0.12 0.32 0.56 0.00 0.55 10 2  
## 4 0.00 0.40 0.60 0.00 0.60 10 2  
## 5 0.00 0.05 0.14 0.81 0.14 1 9  
## 6 0.22 0.45 0.07 0.27 0.08 1 9  
## clusters  
## 1 4  
## 2 4  
## 3 4  
## 4 4  
## 5 4  
## 6 4

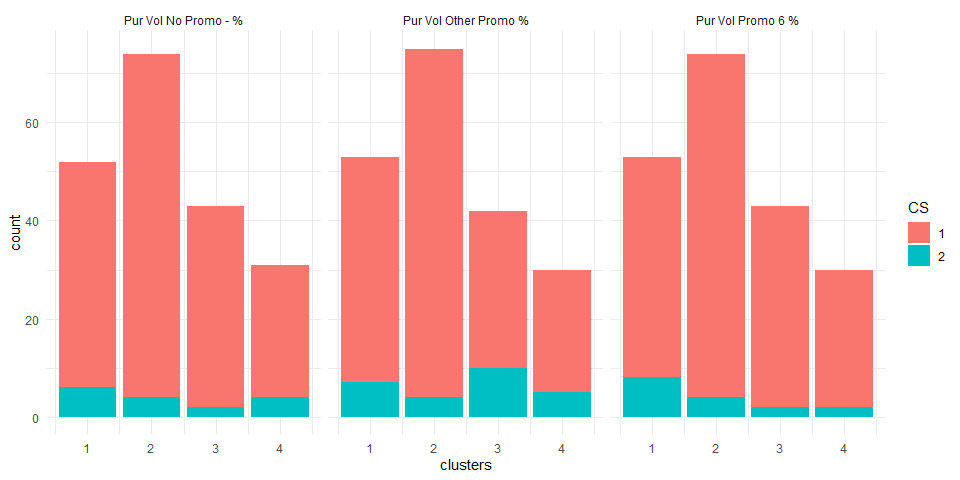
ggplot(Loyality6) +  
 aes(x =clusters,fill= SEX) +  
 geom\_bar() +  
 scale\_fill\_hue() +  
 theme\_minimal() +  
 facet\_wrap(vars(c("Pr Cat 1","Pr Cat 2", "Pr Cat 3","Pr Cat 4","other999")))



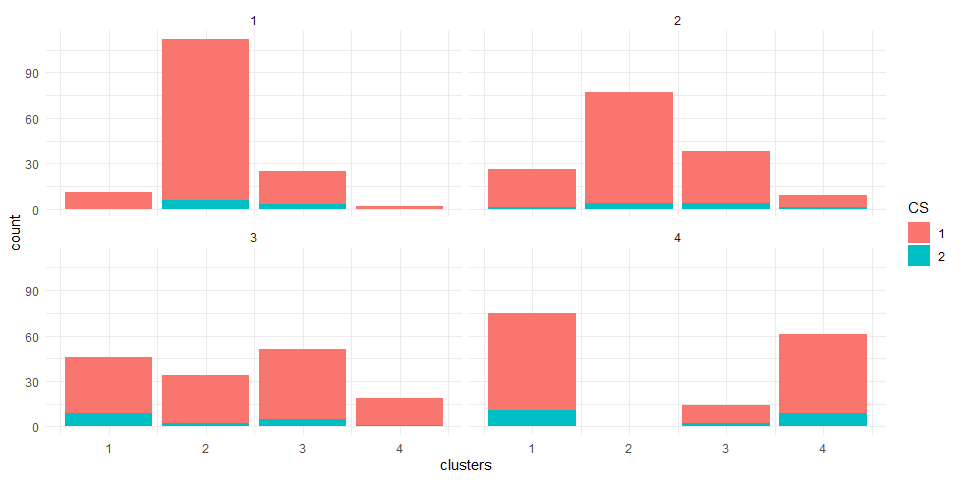
ggplot(Loyality6) +  
 aes(x =clusters,fill= CS) +  
 geom\_bar() +  
 scale\_fill\_hue() +  
 theme\_minimal() +  
 facet\_wrap(vars(c(HS)))



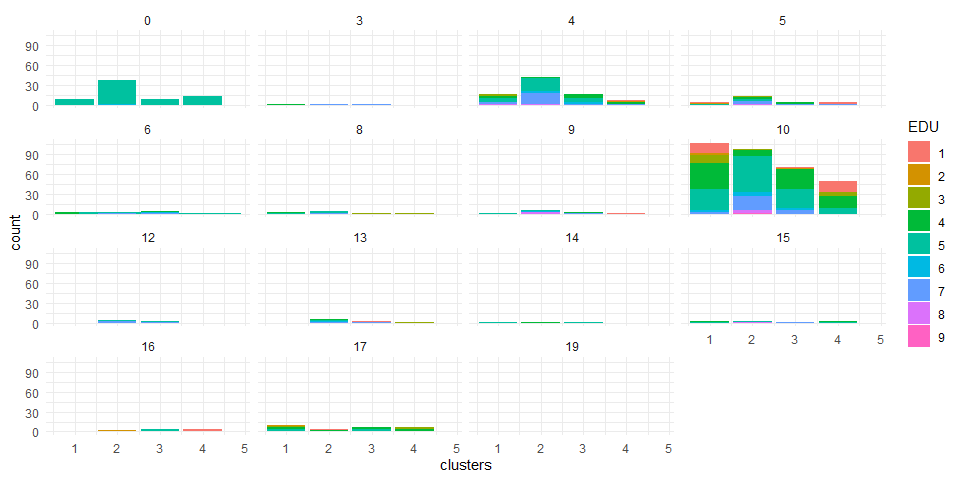
ggplot(Loyality6) +  
 aes(x =clusters,fill= CS) +  
 geom\_bar() +  
 scale\_fill\_hue() +  
 theme\_minimal() +  
 facet\_wrap(vars(c("Pur Vol No Promo - %","Pur Vol Promo 6 %","Pur Vol Other Promo %")))



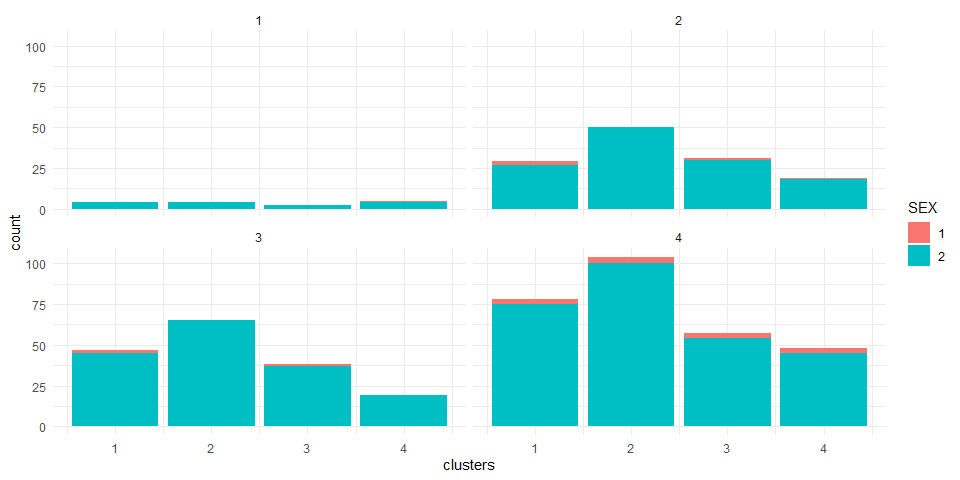
ggplot(Loyality6) +  
 aes(x =clusters,fill= CS) +  
 geom\_bar() +  
 scale\_fill\_hue() +  
 theme\_minimal() +  
 facet\_wrap(vars(c(SEC)))



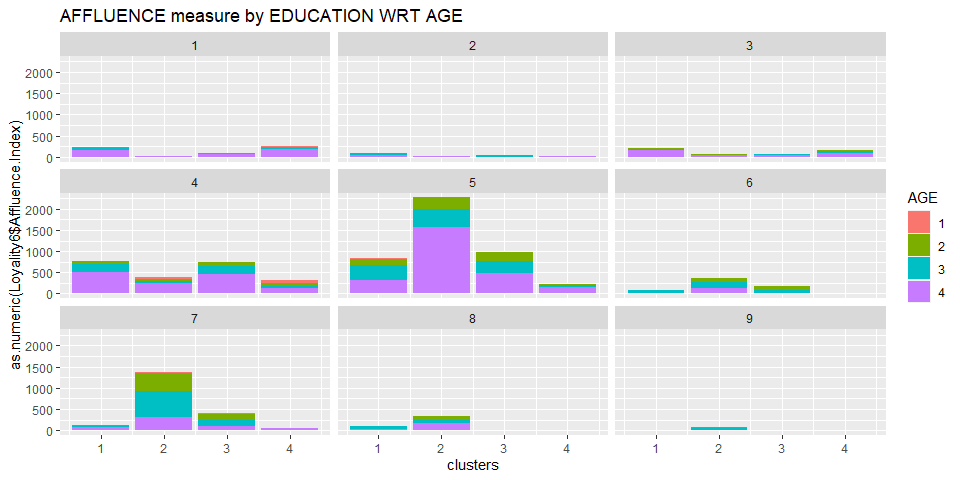
ggplot(Loyality6) +  
 aes(x =clusters,fill= EDU) +  
 geom\_bar() +  
 scale\_fill\_hue() +  
 theme\_minimal() +  
 facet\_wrap(vars(MT))



ggplot(Loyality6) +  
 aes(x =clusters,fill= SEX) +  
 geom\_bar() +  
 scale\_fill\_hue() +  
 theme\_minimal() +   
 facet\_wrap(vars(AGE))



ggplot(Loyality6, aes(x =clusters, y=as.numeric(Loyality6$Affluence.Index), fill= AGE)) + geom\_bar(stat = 'identity') + facet\_wrap(~EDU) + ggtitle("AFFLUENCE measure by EDUCATION WRT AGE")



* As most customers from cluster 4 have access to TV/cable, Add promotions can be telecasted through television which is the best approach for a brand promotions. cluster 1 have more CS = 1. With household people 4,5,7 and 10 and customers fall in cluster 4 have the highest CS = 1.
* Considering education as demographics. There is a high proportion of college graduates in cluster 4 which buys value added packs and premium soaps which shows high brand Loyalty. It looks like most of the people are in 4th and 5th level.
* SEC = 1(high socioeconomic class) with Cluster 4 customers who show a high tendency to buy premium soaps. There are high percentage of customers from other SEC sections in cluster 4, indicating that they prefer to buy any kind of soap. So, we can say that customers with high social economic status don’t care about premium or popular soaps, but their brand royalty is high.
* Most of the SocioEconomic class are Native speakers.The most clusters are dominated by a customer base who speak a common Native language.
* Most of the customers in each cluster are women. It is clearly seen that all the clusters have the highest number of women . It is suggested that more different types products should be released based on Women than men.
* Cluster 4 consists customers with highly affluent people across all education levels. People of Age group 4 are most affluent customer.Customers Potential to be converted into brand loyal customers.

## Conclusion:

* From the above plot we can conclude that most customers are female and they belong to Age group 4 in cluster 4. So based on this company should plan manufacturing more based on this and also should create more add promotions if any new product is released and future product development plans should be built accordingly. Also most all of the customers from age Group 4 in most cluster are not brand loyal but prefer to buy value added packs and premium packs and premium soaps.
* As most of the customers have TV/Cable at home ; It is best option to focus on generating more Add promotions as a effective means of promoting the products.