CRISA Customer Segmentation

Final Assignment

Surbhi Khandelwal
Machine Learning

17 December 2020

Abstract

Customer Data Segmentation Using KMeans. Finding the best customer Segment for running Marketing Campaigns. And assigning success and failure against the customers for which we should run promotions and campaigns.

Contents

I. Introduction

Project Introduction and Goal:

CRISA is an Asian market research agency that specializes in tracking consumer purchase behavior in consumer goods (both durable and nondurable). To track purchase behavior, CRISA constituted household panels in over 100 cities and towns in India, covering most of the Indian urban market. A subset of 600 records is analyzed here. The strata were defined on the basis of socioeconomic status and the market (a collection of cities).

CRISA would now like to segment the market based on two key sets of variables more directly related to the purchase process and to brand loyalty:

- Purchase behavior (volume, frequency, susceptibility to discounts, and brand loyalty)
- Basis of purchase (price, selling proposition)

Doing so would allow CRISA to gain information about what demographic attributes are associated with different purchase behaviors and degrees of brand loyalty, and thus deploy promotion budgets more effectively.

This project is aimed at determining at segmenting the customers on purchase behavior and basis of purchase and then suggesting how to target the various segments via advertisements.

II. Data Exploration

We are using the historical dataset SoapData.csv to build our analysis. This dataset consists of 600 observations and around 46 variables which consists of:

Demographic data like age, gender, education, no. of children, native language, eating habits, etc.

Purchase behaviour data like No. of brands, brand runs, total volume, no. of transactions, value, etc.

Basis of Purchase data like price categorywise purchase, selling propositions, etc.

Missing Data Handling - There are quite a few customers whose data we don't have. For eg. SEX = 0, or education level is not between 1-9, etc. Many of the demographics are not specified across many of the some columns and since k-Means uses continuous variables, they are not important to the clustering algorithm.

For this assignment I am leaving that data as it is, because I did not find these variables to be extremely important. If the value of a variable is NA or NULL, then we will handle those data for the 3 models seperately.

Loading Packages

Install packages that you need and call the libraries for them.

```
library(tidyverse)  # data manipulation
library(factoextra) # clustering algorithms & visualization
library(ISLR)
library(tidyr)
library(caret)
library(dplyr)
library(flexclust)
library(ggplot2)
library(esquisse)
library(hrbrthemes)
library(GGally)
library(viridis)
library(corrplot)
library(ggpubr)
library(gmodels)
library(e1071)
library(FNN)
library(fastDummies)
```

Loading the Dataset

```
SoapData <- read.csv("C:\\Users\\akash\\Desktop\\Kent - !st Sem\\RPractice\\ML\\BathSoap.csv")

summary(SoapData)
```

```
##    Member.id            SEC             FEH              MT
##  Min.   :1010010   Min.   :1.00   Min.   :0.000   Min.   : 0.000
##  1st Qu.:1065295   1st Qu.:1.75   1st Qu.:1.000   1st Qu.: 4.000
##  Median :1106235   Median :2.50   Median :3.000   Median :10.000
##  Mean   :1104188   Mean   :2.50   Mean   :2.048   Mean   : 8.178
##  3rd Qu.:1148293   3rd Qu.:3.25   3rd Qu.:3.000   3rd Qu.:10.000
##  Max.   :1167670   Max.   :4.00   Max.   :3.000   Max.   :19.000
##       SEX              AGE             EDU              HS
##  Min.   :0.000   Min.   :1.000   Min.   :0.000   Min.   : 0.000
```

```
##  1st Qu.:2.000   1st Qu.:3.000   1st Qu.:3.000   1st Qu.: 3.000
##  Median :2.000   Median :3.000   Median :4.500   Median : 4.000
##  Mean   :1.738   Mean   :3.213   Mean   :4.043   Mean   : 4.192
##  3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:5.000   3rd Qu.: 5.000
##  Max.   :2.000   Max.   :4.000   Max.   :9.000   Max.   :15.000
##      CHILD            CS          Affluence.Index No..of.Brands
##  Min.   :1.000   Min.   :0.0000   Min.   : 0.00   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:1.0000   1st Qu.:10.00   1st Qu.:2.000
##  Median :4.000   Median :1.0000   Median :15.00   Median :3.000
##  Mean   :3.233   Mean   :0.9317   Mean   :17.02   Mean   :3.637
##  3rd Qu.:4.000   3rd Qu.:1.0000   3rd Qu.:24.00   3rd Qu.:5.000
##  Max.   :5.000   Max.   :2.0000   Max.   :53.00   Max.   :9.000
##    Brand.Runs     Total.Volume   No..of..Trans        Value
##  Min.   : 1.00   Min.   :  150   Min.   :  1.00   Min.   :  20.0
##  1st Qu.: 8.00   1st Qu.: 6825   1st Qu.: 22.00   1st Qu.: 789.6
##  Median :15.00   Median :10360   Median : 28.00   Median :1216.0
##  Mean   :15.75   Mean   :11915   Mean   : 31.15   Mean   :1337.4
##  3rd Qu.:21.00   3rd Qu.:15344   3rd Qu.: 40.00   3rd Qu.:1675.8
##  Max.   :74.00   Max.   :50895   Max.   :138.00   Max.   :6371.9
##  Trans...Brand.Runs    Vol.Tran         Avg..Price    Pur.Vol.No.Promo....
##  Min.   : 1.000     Min.   :  94.43   Min.   : 5.62   Length:600
##  1st Qu.: 1.420     1st Qu.: 250.51   1st Qu.: 9.76   Class :character
##  Median : 1.845     Median : 361.52   Median :11.25   Mode  :character
##  Mean   : 2.618     Mean   : 415.05   Mean   :11.83
##  3rd Qu.: 2.690     3rd Qu.: 490.89   3rd Qu.:13.42
##  Max.   :23.000     Max.   :2525.00   Max.   :33.33
##  Pur.Vol.Promo.6..  Pur.Vol.Other.Promo.. Br..Cd..57..144     Br..Cd..55
##  Length:600         Length:600            Length:600         Length:600
##  Class :character   Class :character      Class :character   Class :character
##  Mode  :character   Mode  :character      Mode  :character   Mode  :character
##
##
##
##  Br..Cd..272        Br..Cd..286         Br..Cd..24        Br..Cd..481
##  Length:600         Length:600         Length:600         Length:600
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Br..Cd..352         Br..Cd..5          Others.999          Pr.Cat.1
##  Length:600         Length:600         Length:600         Length:600
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Pr.Cat.2           Pr.Cat.3           Pr.Cat.4          PropCat.5
##  Length:600         Length:600         Length:600         Length:600
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

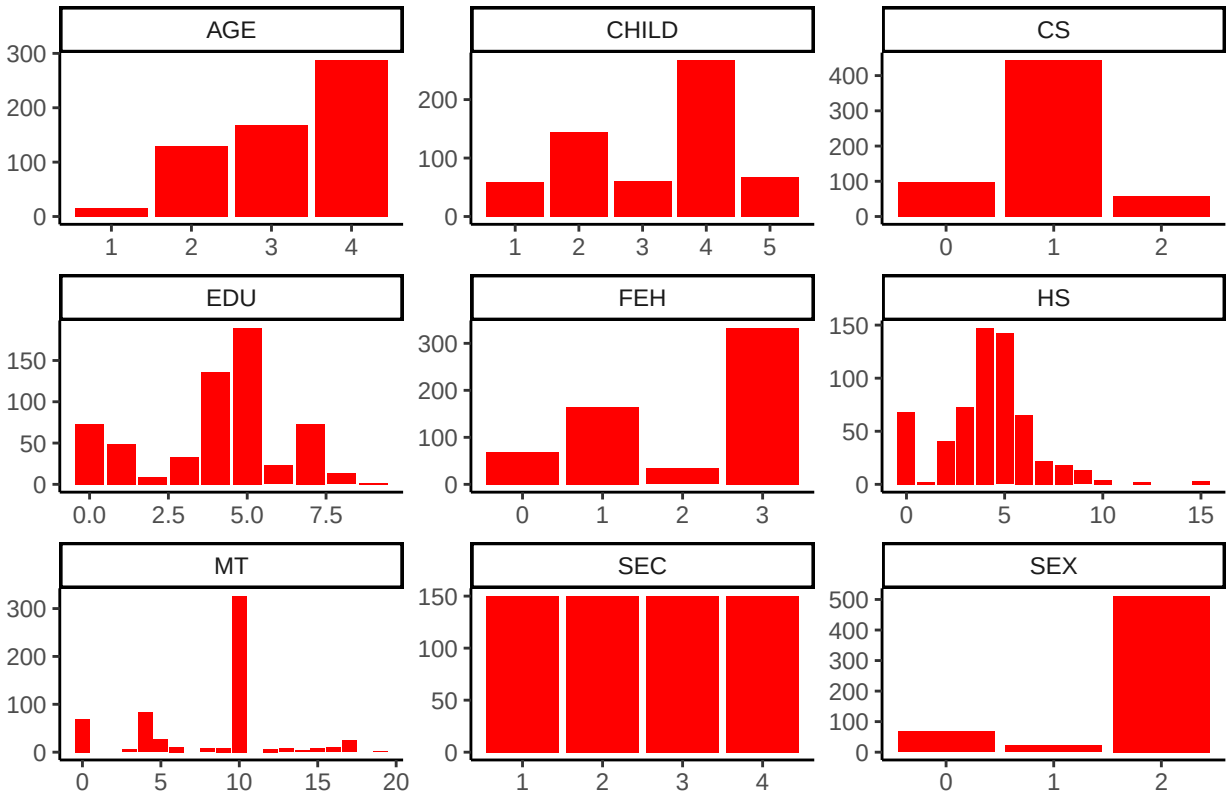
```
##   PropCat.6          PropCat.7          PropCat.8          PropCat.9
##  Length:600         Length:600         Length:600         Length:600
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   PropCat.10         PropCat.11         PropCat.12         PropCat.13
##  Length:600         Length:600         Length:600         Length:600
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   PropCat.14         PropCat.15
##  Length:600         Length:600
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

Preparing the Dataset

```
SoapData_df <- SoapData

SoapData_df$Others.999 <- as.numeric(gsub("\\%", "", SoapData_df$Others.999))
SoapData_df$Pur.Vol.No.Promo.... <- as.numeric(gsub("\\%", "", SoapData_df$Pur.Vol.No.Promo....))
SoapData_df$Pur.Vol.Promo.6.. <- as.numeric(gsub("\\%", "", SoapData_df$Pur.Vol.Promo.6..))
SoapData_df$Pur.Vol.Other.Promo.. <- as.numeric(gsub("\\%", "", SoapData_df$Pur.Vol.Other.Promo..))
SoapData_df$Pr.Cat.1 <- as.numeric(gsub("\\%", "", SoapData_df$Pr.Cat.1))
SoapData_df$Pr.Cat.2 <- as.numeric(gsub("\\%", "", SoapData_df$Pr.Cat.2))
SoapData_df$Pr.Cat.3 <- as.numeric(gsub("\\%", "", SoapData_df$Pr.Cat.3))
SoapData_df$Pr.Cat.4 <- as.numeric(gsub("\\%", "", SoapData_df$Pr.Cat.4))
SoapData_df$PropCat.5 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.5))
SoapData_df$PropCat.6 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.6))
SoapData_df$PropCat.7 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.7))
SoapData_df$PropCat.8 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.8))
SoapData_df$PropCat.9 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.9))
SoapData_df$PropCat.10 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.10))
SoapData_df$PropCat.11 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.11))
SoapData_df$PropCat.12 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.12))
SoapData_df$PropCat.13 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.13))
SoapData_df$PropCat.14 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.14))
SoapData_df$PropCat.15 <- as.numeric(gsub("\\%", "", SoapData_df$PropCat.15))



# Get the actual Volume rather than the percentage number.
SoapData_df$Others.999 <- (SoapData_df$Others.999*SoapData_df$Total.Volume)/100
SoapData_df$Pur.Vol.No.Promo.... <- (SoapData_df$Pur.Vol.No.Promo....*SoapData_df$Total.Volume)/100
SoapData_df$Pur.Vol.Promo.6.. <- (SoapData_df$Pur.Vol.Promo.6..*SoapData_df$Total.Volume)/100
SoapData_df$Pur.Vol.Other.Promo.. <- (SoapData_df$Pur.Vol.Other.Promo..*SoapData_df$Total.Volume)/100
SoapData_df$Pr.Cat.1 <- (SoapData_df$Pr.Cat.1*SoapData_df$Total.Volume)/100
```

```
SoapData_df$Pr.Cat.2 <- (SoapData_df$Pr.Cat.2*SoapData_df$Total.Volume)/100
SoapData_df$Pr.Cat.3 <- (SoapData_df$Pr.Cat.3*SoapData_df$Total.Volume)/100
SoapData_df$Pr.Cat.4 <- (SoapData_df$Pr.Cat.4*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.5 <- (SoapData_df$PropCat.5*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.6 <- (SoapData_df$PropCat.6*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.7 <- (SoapData_df$PropCat.7*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.8 <- (SoapData_df$PropCat.8*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.9 <- (SoapData_df$PropCat.9*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.10 <- (SoapData_df$PropCat.10*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.11 <- (SoapData_df$PropCat.11*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.12 <- (SoapData_df$PropCat.12*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.13 <- (SoapData_df$PropCat.13*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.14 <- (SoapData_df$PropCat.14*SoapData_df$Total.Volume)/100
SoapData_df$PropCat.15 <- (SoapData_df$PropCat.15*SoapData_df$Total.Volume)/100
```

Data Exploration

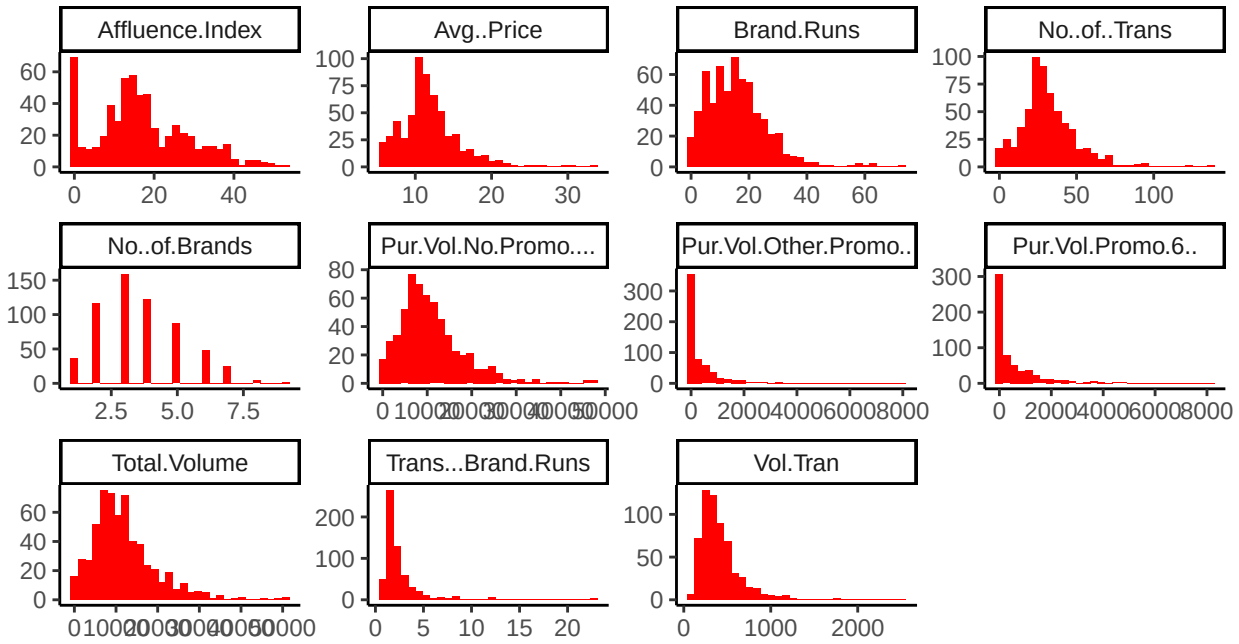


Looking at this data, we can see that most of the customers were

- age level is 3 and 4.
- no. of children is 4 and then 2
- Most of them have Television
- Education level is average at levels 4 and 5 mostly

- Eating Habits is mostly either veg or non-veg
- No. of people in the household is between 3 to 6
- Native language was 10
- socioeconomic level is evenly distributed
- Most of the shoppers are females.

```
‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

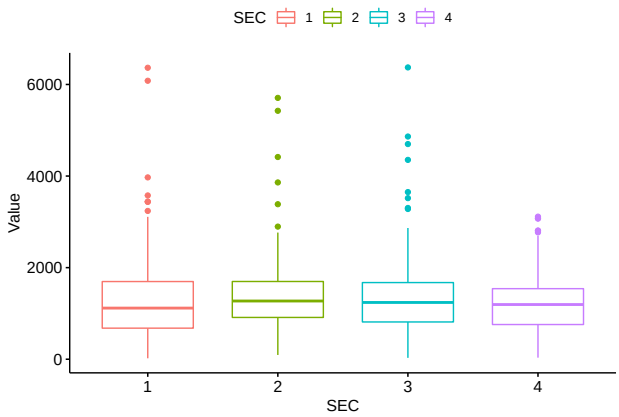


Affluence.index, avg_price, num_trans look to be normally distributed.

- Affluence.index has many 0's which looks like missing values. If we use this variable then we will deal with it accordingly.
- A few of the variables seem to be negatively skewed.

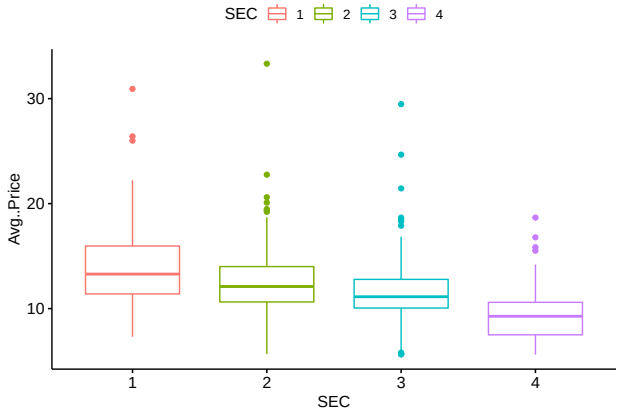
Looking at a few Interrelated Relationships

Let's look at the relation between Socioeconomic level and Total Value.



We can see here that irrespective of the Socioeconomic level, customers spend the almost similar amount of money.

Let's look at the relation between Socioeconomic level and Average price of purchase.



Now we can see a difference. Higher the socioeconomic level, higher is the avg. price of purchase.

III Model Building

a. The variables that describe purchase behavior (including brand loyalty)

For brand loyalty indicators, we have data on

- percent of purchases devoted to major brands
- a catch-all variable for percent of purchases devoted to other smaller brands (to reduce complexity of analysis) - Others.999, and
- a derived variable that indicates the maximum share devoted to any one brand - max.brand.ind

Since CRISA is compiling this data for general marketing use, and not on behalf of one particular brand, we can say a customer who is fully devoted to brand A is similar to a customer fully devoted to brand B - both are fully loyal customers in their behavior. But if we include all the brand shares in the clustering, the analysis will treat those two customers as very different.

So we will use only the derived variable for maximum purchase share for a brand, any brand, ie. "max.brand.ind" and the "other999" along with the purchase information.

So the columns we will be using for this analysis are:

- Average Price
- Brand Runs
- No. of transactions
- No. of Brands
- Other999
- Total Volume
- Value
- Max.brand.ind

Let's store the above variables in one dataset

```
# Building a dataset that has all the variables needed to describe purchase behaviours including brand

purbehdf <- SoapData_df[, c(12:19,31)]
purbehdf <- purbehdf[,-6]

# Finding the max value for each of the brands for brand loyalty.

purbehdf$Max.brand.ind <-apply(SoapData_df[,23:30], MARGIN=1, FUN=max)


purbehdf$Max.brand.ind <- as.numeric(gsub("\\%", "", purbehdf$Max.brand.ind))

purbehdf$Max.brand.ind <- (purbehdf$Max.brand.ind*SoapData_df$Total.Volume)/100


# Adding the max.brand.ind to the dataset.
```

```r
SoapData_df <- cbind(SoapData_df, purbehdf$Max.brand.ind)
purchasebeh <- purbehdf

# Now let's just look at the final dataset before moving ahead with the actual KMeans.

head(purchasebeh)
```

```
##   No..of.Brands Brand.Runs Total.Volume No..of..Trans  Value Vol.Tran
## 1             3         17         8025            24  818.0   334.38
## 2             5         25        13975            40 1681.5   349.38
## 3             5         37        23100            63 1950.0   366.67
## 4             2          4         1500             4  114.0   375.00
## 5             3          6         8300            13  591.0   638.46
## 6             3         26        18175            41 1705.5   443.29
##   Avg..Price Others.999 Max.brand.ind
## 1      10.19   3948.300        3049.5
## 2      12.03   9768.525        1118.0
## 3       8.44   8754.900       12705.0
## 4       7.60      0.000         900.0
## 5       7.12   6698.100         415.0
## 6       9.38  15575.975        1454.0
```

Scaling the Data I am scaling the entire dataset without dividing it into train and test because it's an unsupervised model. Since we won't be able to automatically calculate the accuracy/effectiveness of your model. We can only calculate the distance value and understand how our model is doing.

```r
# Scaling the data frame (z-score)
purchasebeh_scaled <- scale(purchasebeh)
```

```r
# Let's look at the percentage of missing values for each variable in the dataset.

navalues<- colMeans(is.na(purchasebeh_scaled))*100
as.data.frame(navalues)
```

Handling missing records.

```
##                navalues
## No..of.Brands         0
## Brand.Runs            0
## Total.Volume          0
## No..of..Trans         0
## Value                 0
## Vol.Tran              0
## Avg..Price            0
## Others.999            0
## Max.brand.ind         0
```

There are no missing values that we need to handle or manage in the dataset. That's great!

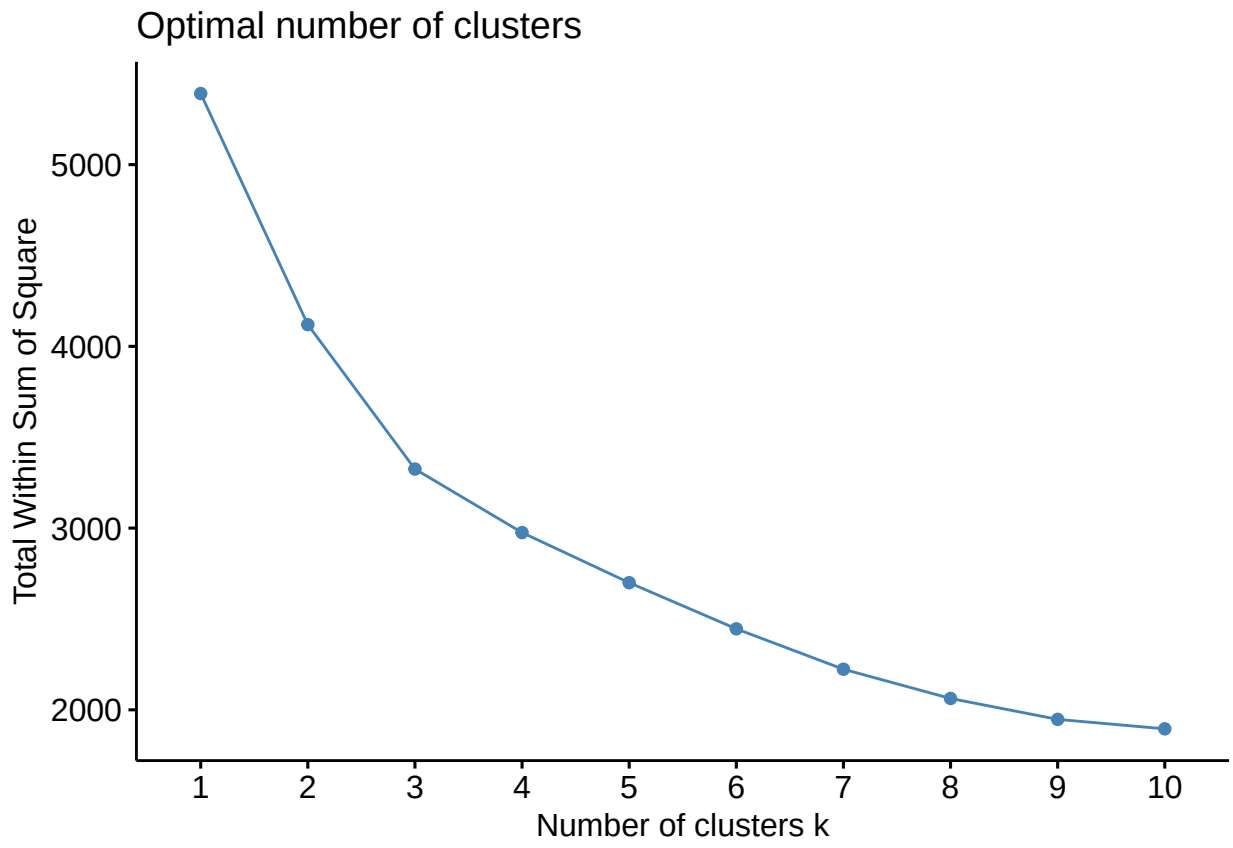
```
distance <- get_dist(purchasebeh_scaled)
#fviz_dist(distance)
```

Getting the distance.

K Means Implementation

Determining the optimum value of k using wss method. WSS Method

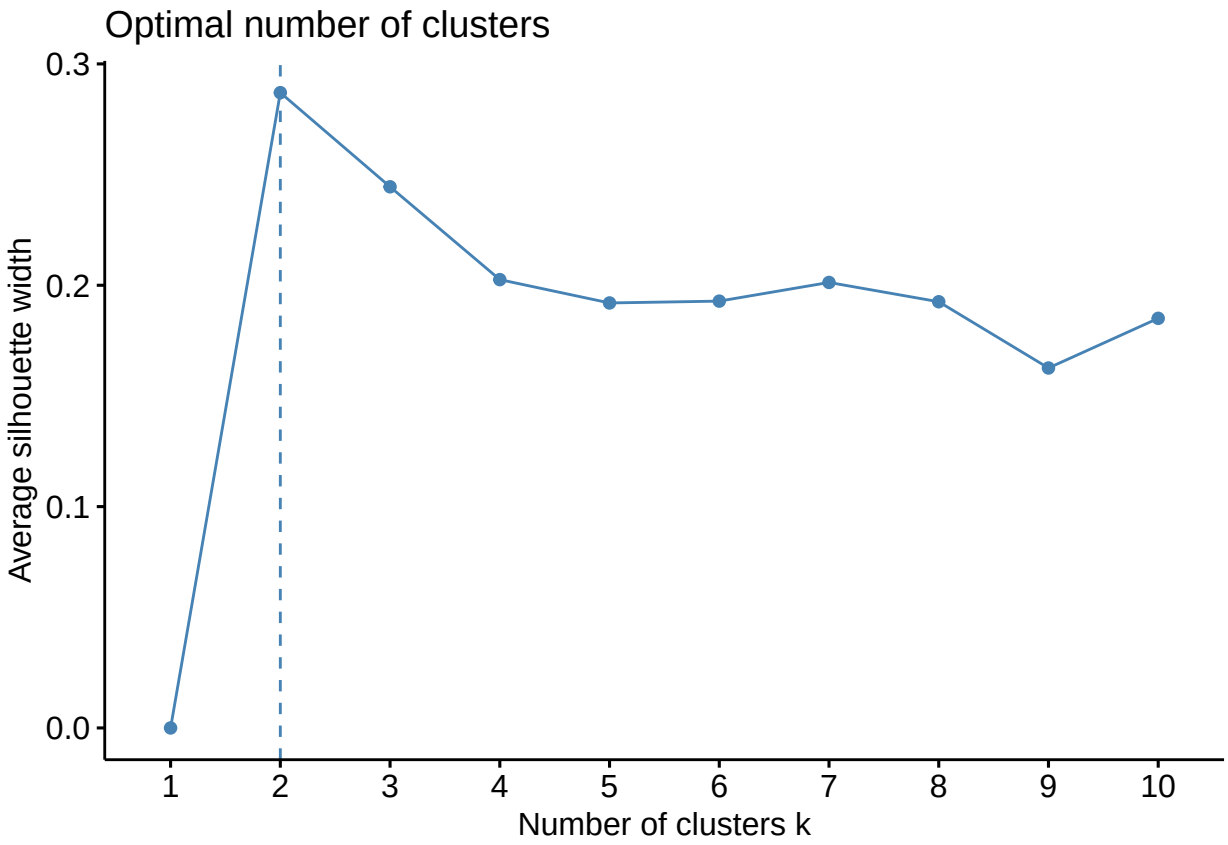
```
fviz_nbclust(purchasebeh_scaled, kmeans, method = "wss")
```



Optimal Value of K=3. Meaning our data can be easily put into 3 clusters. This should align well with loyal customers, completely unloyal customers and others.

Silhouette Method

```
set.seed(123)
fviz_nbclust(purchasebeh_scaled, kmeans, method = "silhouette")
```



Even Accoring to this method, Optimal k=2. But I feel that with our purpose it ll be better to go with k=3. As this aligns well with loyal customers, completely disloyal customers and others. So let's cluster the data and analyze our clusters.

```r
set.seed(123)
kopt <- kmeans(purchasebeh_scaled, centers = 3, nstart = 25) # k = 3, number of restarts = 25

# Visualize the output

kopt$centers # output the centers
```

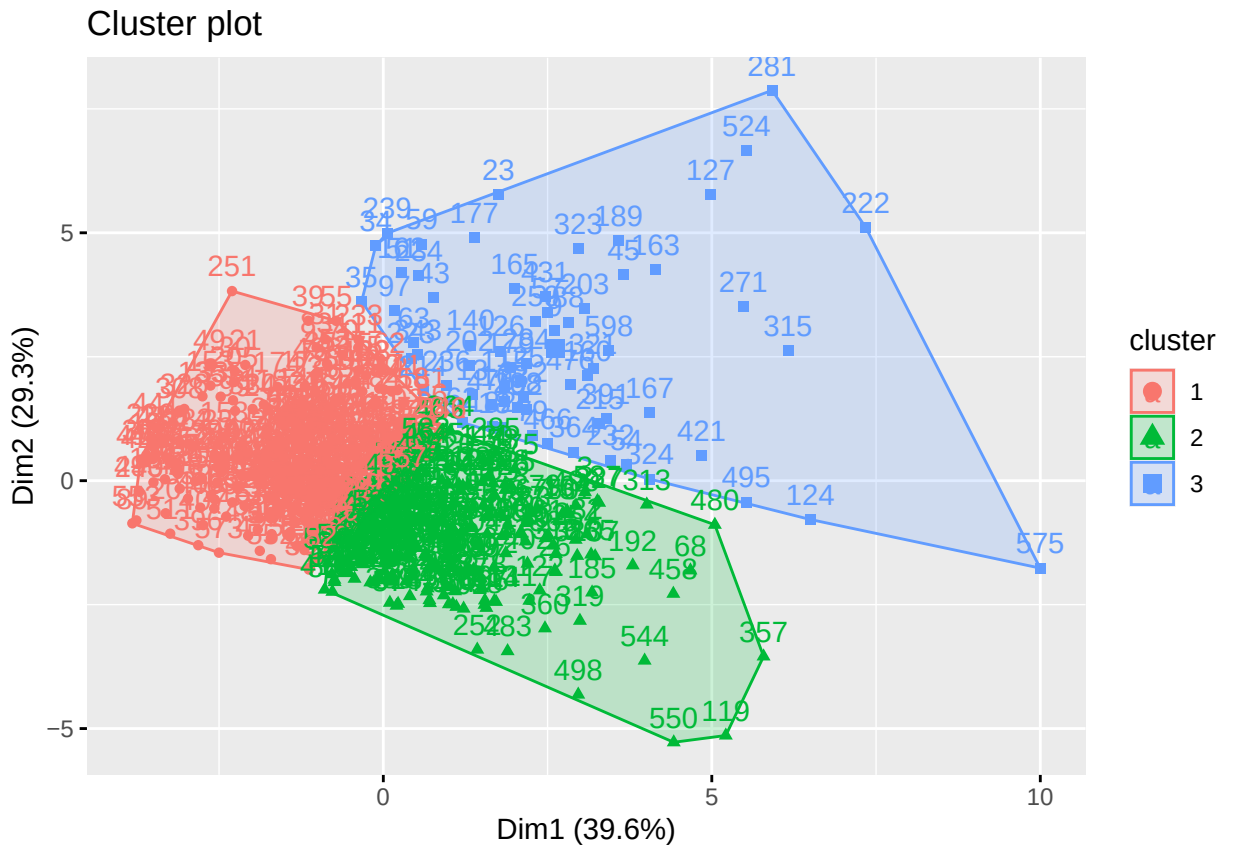
Running K-Means for Optimal k=3

```
##   No..of.Brands Brand.Runs Total.Volume No..of..Trans      Value    Vol.Tran
## 1    -0.5145044 -0.5807571   -0.5003166    -0.5822513 -0.5381477 -0.08434532
## 2     0.8190677  0.9193994    0.1274229     0.7849772  0.3271368 -0.43965822
## 3    -0.2016098 -0.2107502    1.9935960     0.2363765  1.5222636  1.84527253
##    Avg..Price Others.999 Max.brand.ind
## 1 -0.08652417 -0.4486483   -0.07487151
## 2  0.31690024  0.3088685   -0.37529079
## 3 -0.62023886  1.1508268    1.58896898
```

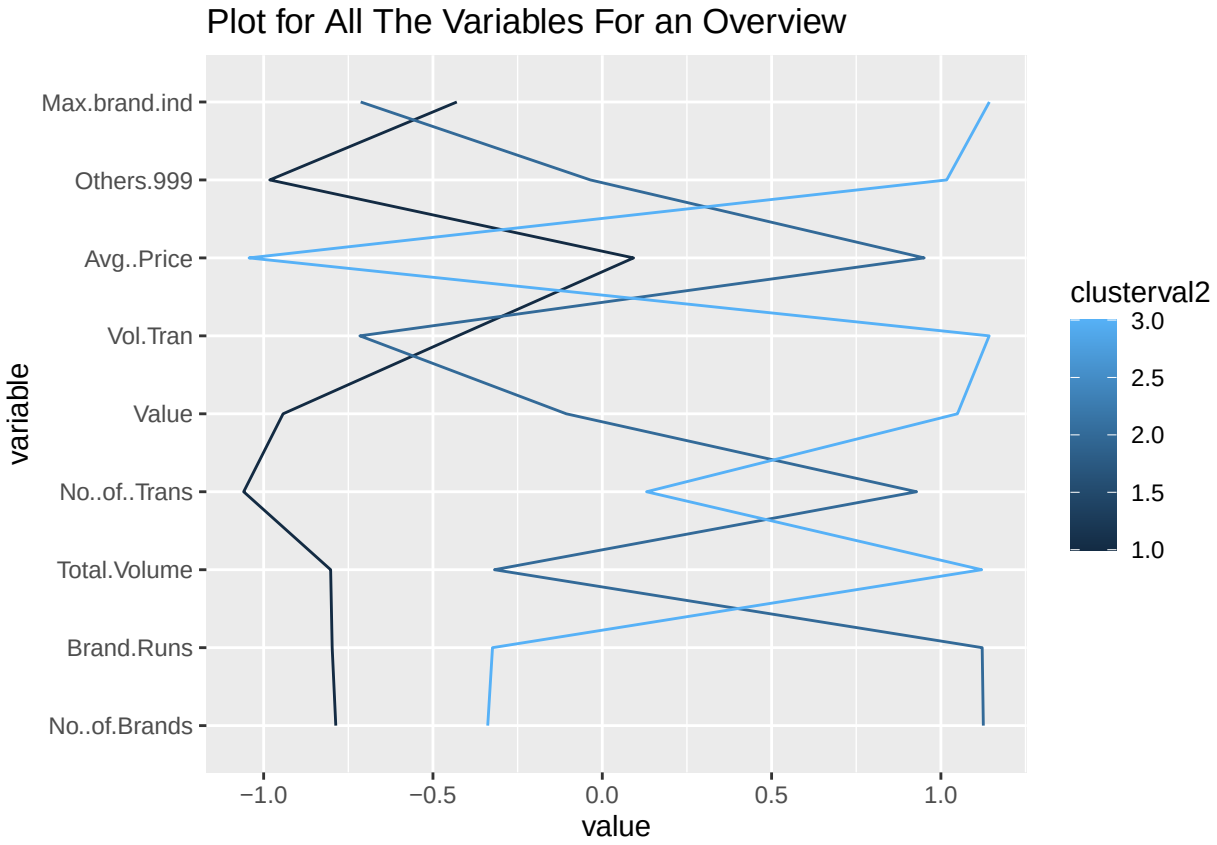
```r
kopt$size # Number of Universities in each cluster
```

```
## [1] 318 216  66
```

```r
fviz_cluster(kopt, data = purchasebeh_scaled) # Visualize the output
```



Determing The Meaning of The Clusters.



Cluster 1 (Sometimes use other brands, but overall high Brand Loyalty - Mostly Loyal Customers of Low-End Brands) - 76 Low average price, low no. of transactions, low no. of brands, high total volume, high value, vol.trans is also high, sometimes use other brands but high brand loyalty.

Cluster 2 (High Brand Loyalty with High Avg Price - Loyal Customers of Pricey Brands) - 330 - High avg price, low no of brand runs, low no. of transactions, low no. of brands, total volume is low, and value is also low, they rarely try other brands.

Cluster 3 (Low Brand Loyalty) - 194 High avg price, high no of brand runs, no of transactions, total volume is average, value is average, Volume per transaction is low, they try different brands.

b. The variables that describe the basis for purchase.

The variables that I have used for describing basis for purchase are:

- Pur.Vol.No.Promo. . . .,
- Pur.Vol.Promo.6..,
- Pur.Vol.Other.Promo..,
- Pr.Cat.1,
- Pr.Cat.2,
- Pr.Cat.3,
- Pr.Cat.4,
- PropCat.5 , ..to.. PropCat.15

Building a dataset that has all the variables needed to describe basis for purchase.

```
purchasedf <- SoapData_df[,c(20:22,32:46)]

# Now let's look at the dataset

head(purchasedf)
```

```
##   Pur.Vol.No.Promo.... Pur.Vol.Promo.6.. Pur.Vol.Other.Promo.. Pr.Cat.1
## 1              8025.00               0.0                   0.0  1845.75
## 2             12437.75            1397.5                 279.5  4052.75
## 3             21714.00             462.0                 924.0  2772.00
## 4              1500.00               0.0                   0.0     0.00
## 5              5063.00            1162.0                1992.0     0.00
## 6             18175.00               0.0                   0.0  3998.50
##   Pr.Cat.2 Pr.Cat.3 Pr.Cat.4 PropCat.5 PropCat.6 PropCat.7 PropCat.8 PropCat.9
## 1  4494.00  1043.25   561.75   4012.50      0.00      0.00      0.00      0.00
## 2  7686.25  1257.75   838.50   6428.50   4891.25    419.25    279.50    139.75
## 3  7392.00 12936.00     0.00   5544.00   2772.00    693.00    231.00    231.00
## 4   600.00   900.00     0.00    600.00      0.00      0.00      0.00      0.00
## 5   415.00  1162.00  6723.00   6723.00      0.00      0.00    415.00      0.00
## 6  8178.75  1272.25  4907.25   8905.75   1817.50      0.00    181.75   1272.25
##   PropCat.10 PropCat.11 PropCat.12 PropCat.13 PropCat.14 PropCat.15
## 1          0        0.0     240.75          0    1043.25    2728.50
## 2          0      838.5       0.00          0    1118.00       0.00
## 3          0        0.0     462.00          0   12936.00       0.00
## 4          0        0.0       0.00          0     900.00       0.00
## 5          0        0.0       0.00          0    1162.00       0.00
## 6          0        0.0       0.00          0    1272.25    4907.25
```

```
# Scaling the data frame (z-score)
purchase_scaled <- scale(purchasedf)
```

Scaling the Data Let's check if there are any missing values in the dataset.

```
# Let's look at the percentage of missing values for each variable in the dataset.

navalues<- colMeans(is.na(purchase_scaled))*100
as.data.frame(navalues)
```

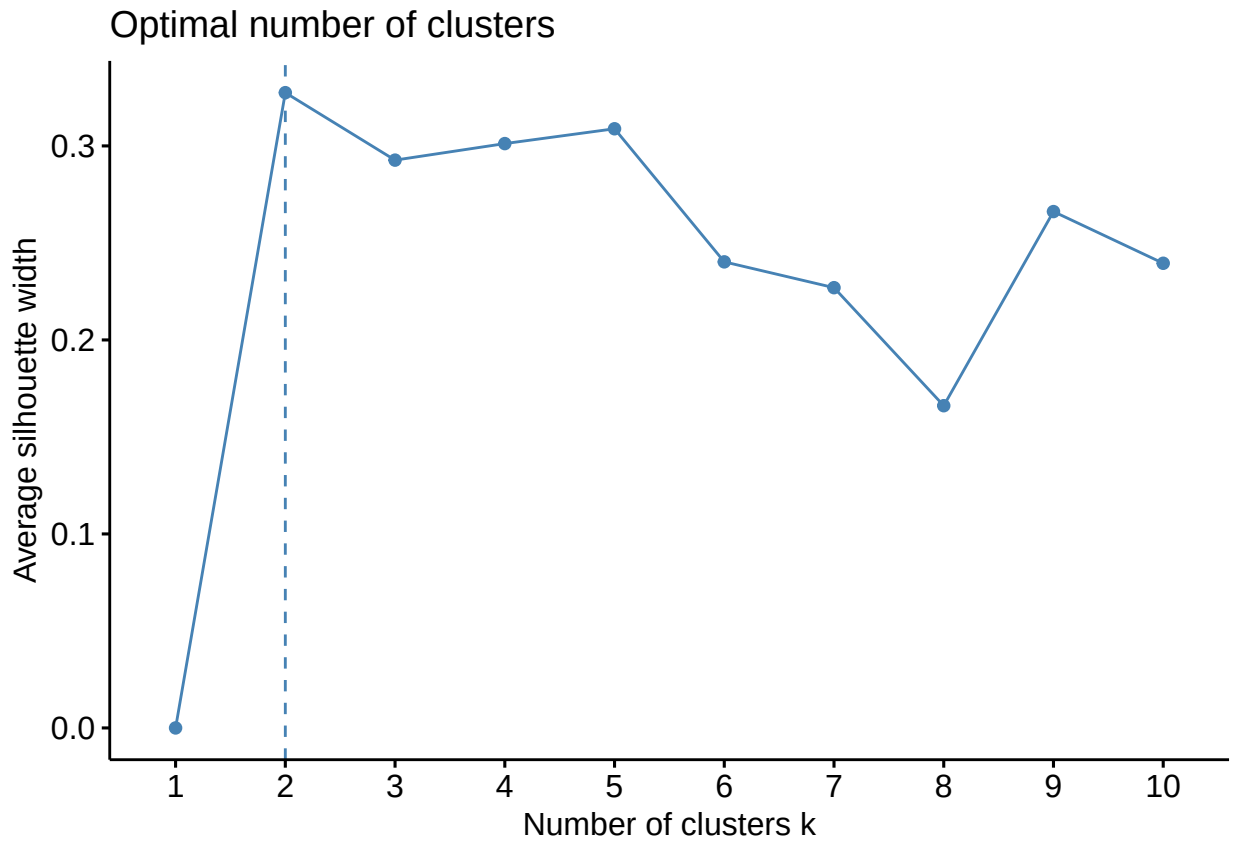
```
##                         navalues
## Pur.Vol.No.Promo....           0
## Pur.Vol.Promo.6..              0
## Pur.Vol.Other.Promo..          0
## Pr.Cat.1                       0
## Pr.Cat.2                       0
## Pr.Cat.3                       0
## Pr.Cat.4                       0
## PropCat.5                      0
## PropCat.6                      0
## PropCat.7                      0
## PropCat.8                      0
## PropCat.9                      0
## PropCat.10                     0
## PropCat.11                     0
## PropCat.12                     0
## PropCat.13                     0
## PropCat.14                     0
## PropCat.15                     0
```

Again, there are no missing values. So let's proceed.

```
distance <- get_dist(purchase_scaled)
```

Getting the distance. Silhouette Method

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



We see that the optimal K value is 2. The two clusters could be - customers who don't rely on any promotions or selling propositions, those who highly rely on promotions.

K Means Implementation Running K-Means for k=2 as that looks like the optimal value of k.

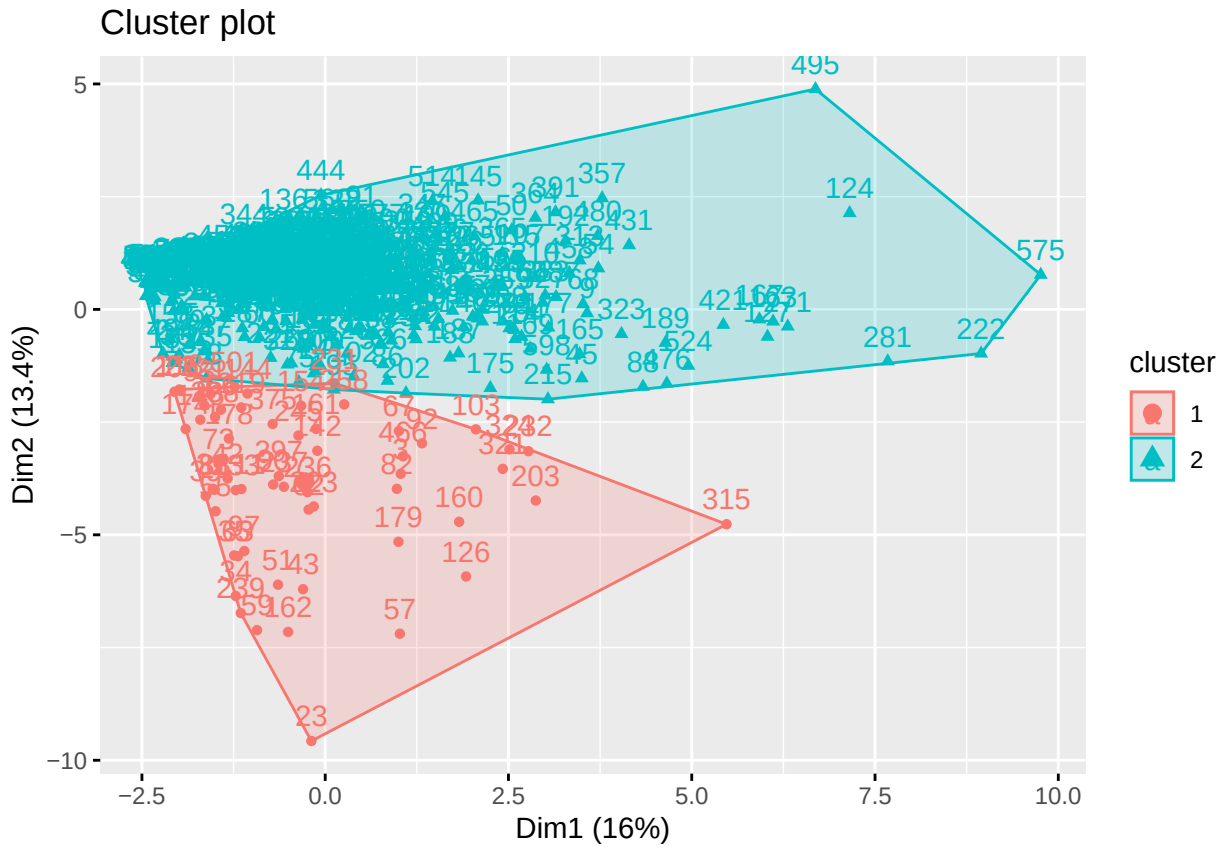
```
set.seed(123)
k2p <- kmeans(purchase_scaled, centers = 2, nstart = 25)

# Visualize the output

k2p$centers # output the centers
```

```
##   Pur.Vol.No.Promo.... Pur.Vol.Promo.6.. Pur.Vol.Other.Promo..    Pr.Cat.1
## 1           0.85906252      -0.084526641            0.64257036 -0.53921622
## 2          -0.09545139       0.009391849           -0.07139671  0.05991291
##      Pr.Cat.2   Pr.Cat.3     Pr.Cat.4   PropCat.5    PropCat.6   PropCat.7
## 1 -0.43354364  2.6246271 -0.054373664 -0.40553926  0.073027094 -0.33035507
## 2  0.04817152 -0.2916252  0.006041518  0.04505992 -0.008114122  0.03670612
##     PropCat.8   PropCat.9  PropCat.10 PropCat.11  PropCat.12  PropCat.13
## 1 -0.39352785  0.02842722 -0.24707926 -0.1918323 -0.13662726 -0.18010935
## 2  0.04372532 -0.00315858  0.02745325  0.0213147  0.01518081  0.02001215
##   PropCat.14  PropCat.15
## 1  2.6215703 -0.16578478
## 2 -0.2912856  0.01842053
```

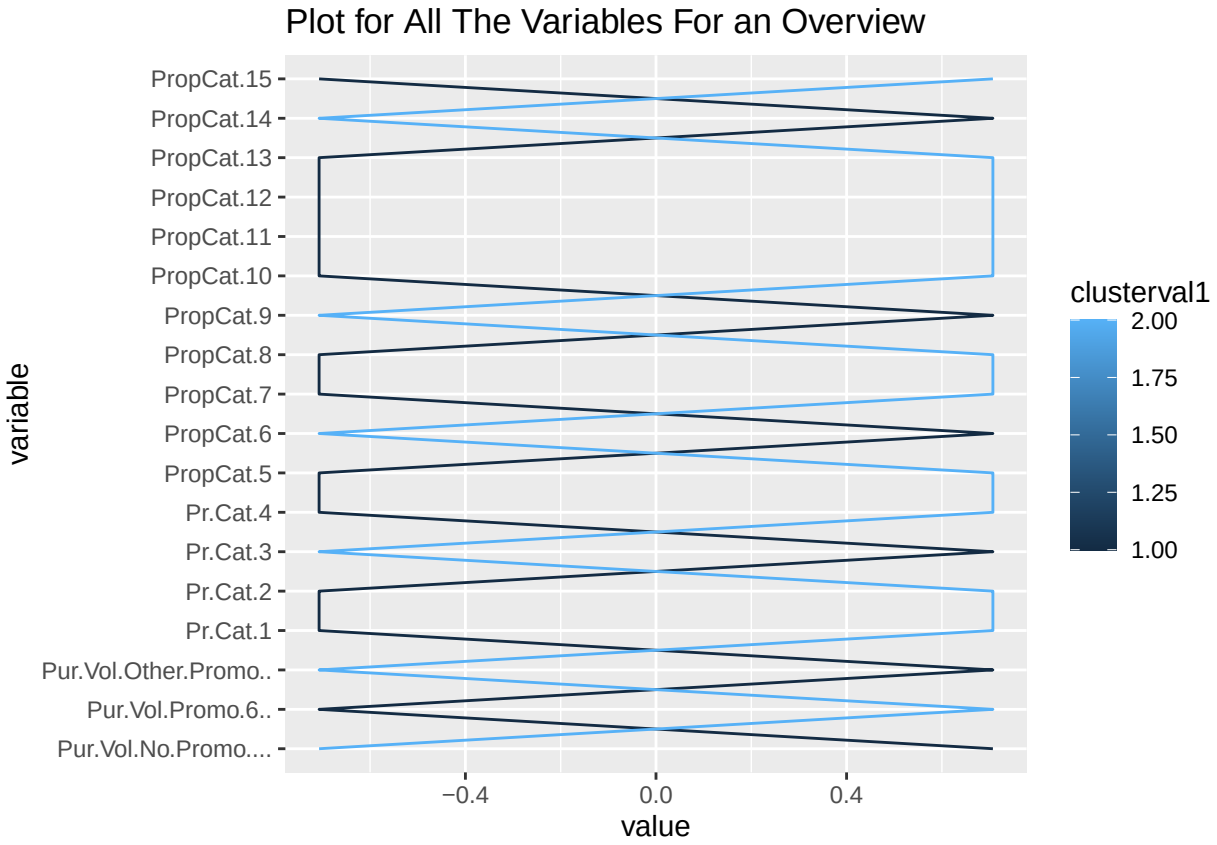
```
k2p$size # Number of Customers in each cluster
```

```
## [1]  60 540
```

```
fviz_cluster(k2p, data = purchase_scaled) # Visualize the output
```



Determining the Meaning of the clusters.



The two clusters are well separated across most variables.

Cluster 1 (60) - purchases without needing promotional offers, likes pricing category 3, and is somewhat responsive to selling propositions 6,9, and 14.

Cluster 2 (540) - Believe in promotions, high Pr.Cat.1,2 and 4. PropCat5, 7,8,10,11,12,13,and 15.

c. The variables that describe both purchase behavior and basis of purchase

```
# We already have the scaled datasets for both the databases, so let's just combine them to form the ta

completedf <- cbind(purchasebeh_scaled, purchase_scaled)

head(completedf)
```

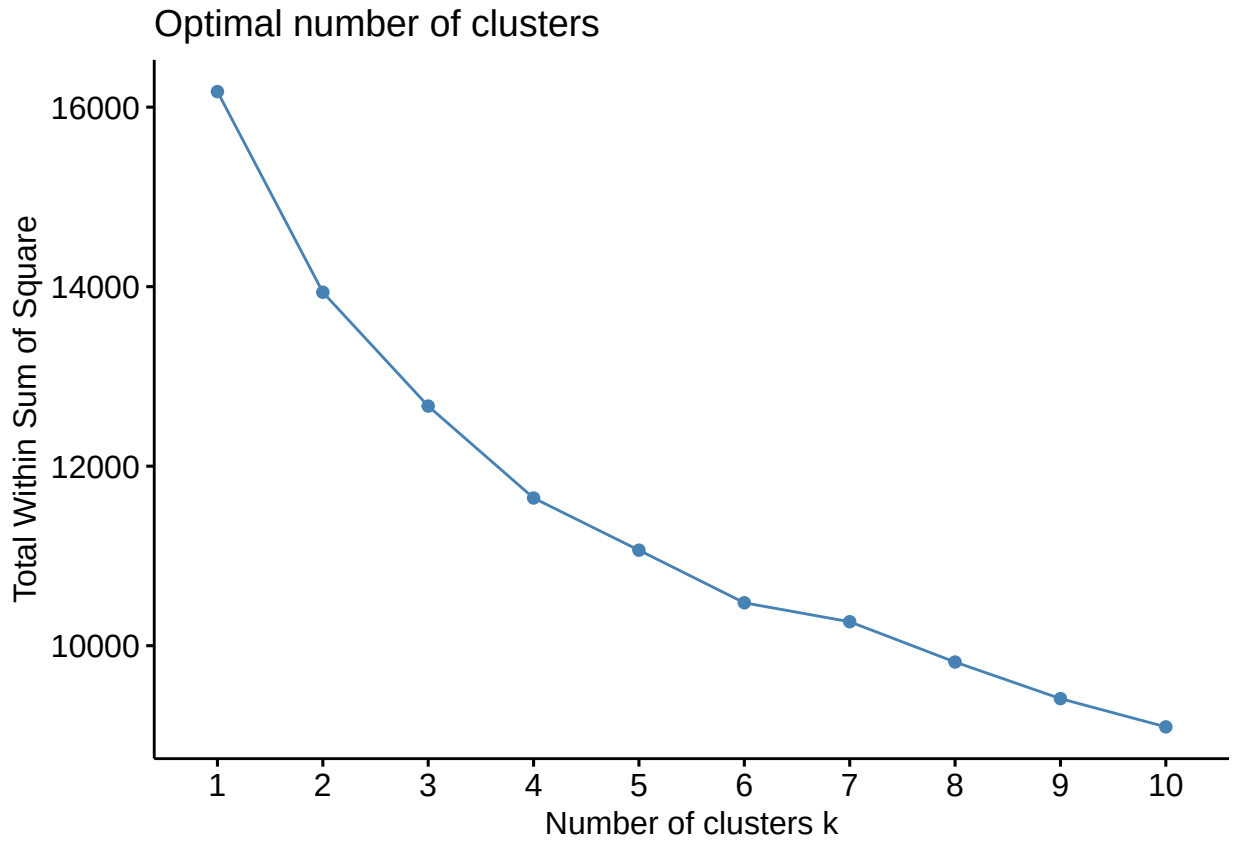
```
##      No..of.Brands Brand.Runs Total.Volume No..of..Trans      Value   Vol.Tran
## [1,]    -0.4030277  0.1200727   -0.5005898    -0.4104681 -0.5881031 -0.3242918
## [2,]     0.8630280  0.8895639    0.2651391     0.5076339  0.3896410 -0.2639930
## [3,]     0.8630280  2.0438006    1.4394712     1.8274054  0.6936645 -0.1944886
## [4,]    -1.0360556 -1.1303505   -1.3403176    -1.5580955 -1.3852447 -0.1610026
## [5,]    -0.4030277 -0.9379777   -0.4651989    -1.0416632 -0.8451360  0.8980852
## [6,]    -0.4030277  0.9857502    0.8056536     0.5650152  0.4168163  0.1135176
##       Avg..Price Others.999 Max.brand.ind Pur.Vol.No.Promo....
## [1,] -0.43944366 -0.3870628    -0.1260909           -0.3943558
## [2,]  0.05217678  0.6565013    -0.4781791            0.1983882
## [3,] -0.90701745  0.4747587     1.6339858            1.4444233
## [4,] -1.13145287 -1.0949914    -0.5179178           -1.2708284
## [5,] -1.25970168  0.1059753    -0.6063272           -0.7922274
## [6,] -0.65586353  1.6977747    -0.4169306            0.9690461
##      Pur.Vol.Promo.6.. Pur.Vol.Other.Promo..      Pr.Cat.1   Pr.Cat.2
## [1,]        -0.5574329            -0.5116939 -0.2794416143 -0.2520866
## [2,]         0.7686897            -0.1252184  0.3854838369  0.2625352
## [3,]        -0.1190296             0.7659568 -0.0003808082  0.2150992
## [4,]        -0.5574329            -0.5116939 -0.8355295851 -0.8798374
## [5,]         0.5452179             2.2427218 -0.8355295851 -0.9096612
## [6,]        -0.5574329            -0.5116939  0.3691393848  0.3419310
##        Pr.Cat.3   Pr.Cat.4    PropCat.5  PropCat.6  PropCat.7  PropCat.8
## [1,] -0.1988988 -0.1925159 -0.246319991 -0.4980207 -0.4173594 -0.5069920
## [2,] -0.1496659 -0.1056810  0.141979507  1.6247065 -0.2567103 -0.3267324
## [3,]  2.5307700 -0.3687742 -0.000177325  0.7049846 -0.1518142 -0.3580118
## [4,] -0.2317780 -0.3687742 -0.794776958 -0.4980207 -0.4173594 -0.5069920
## [5,] -0.1716428  1.7406778  0.189311544 -0.4980207 -0.4173594 -0.2393435
## [6,] -0.1463378  1.1709563  0.540123103  0.2907463 -0.4173594 -0.3897749
##       PropCat.9 PropCat.10 PropCat.11 PropCat.12 PropCat.13 PropCat.14
## [1,] -0.4317488 -0.2850019 -0.2651424  0.8905060 -0.2536688 -0.1912455
## [2,] -0.2787491 -0.2850019  0.3729320 -0.2907978 -0.2536688 -0.1739337
## [3,] -0.1788477 -0.2850019 -0.2651424  1.9761280 -0.2536688  2.5630697
## [4,] -0.4317488 -0.2850019 -0.2651424 -0.2907978 -0.2536688 -0.2244217
## [5,] -0.4317488 -0.2850019 -0.2651424 -0.2907978 -0.2536688 -0.1637435
## [6,]  0.9611232 -0.2850019 -0.2651424 -0.2907978 -0.2536688 -0.1382100
##      PropCat.15
## [1,]  2.1061267
## [2,] -0.2505867
## [3,] -0.2505867
## [4,] -0.2505867
## [5,] -0.2505867
## [6,]  3.9879993
```

```
distance <- get_dist(completedf)
```

Getting the distance.

Determining the optimum value of k wss method.

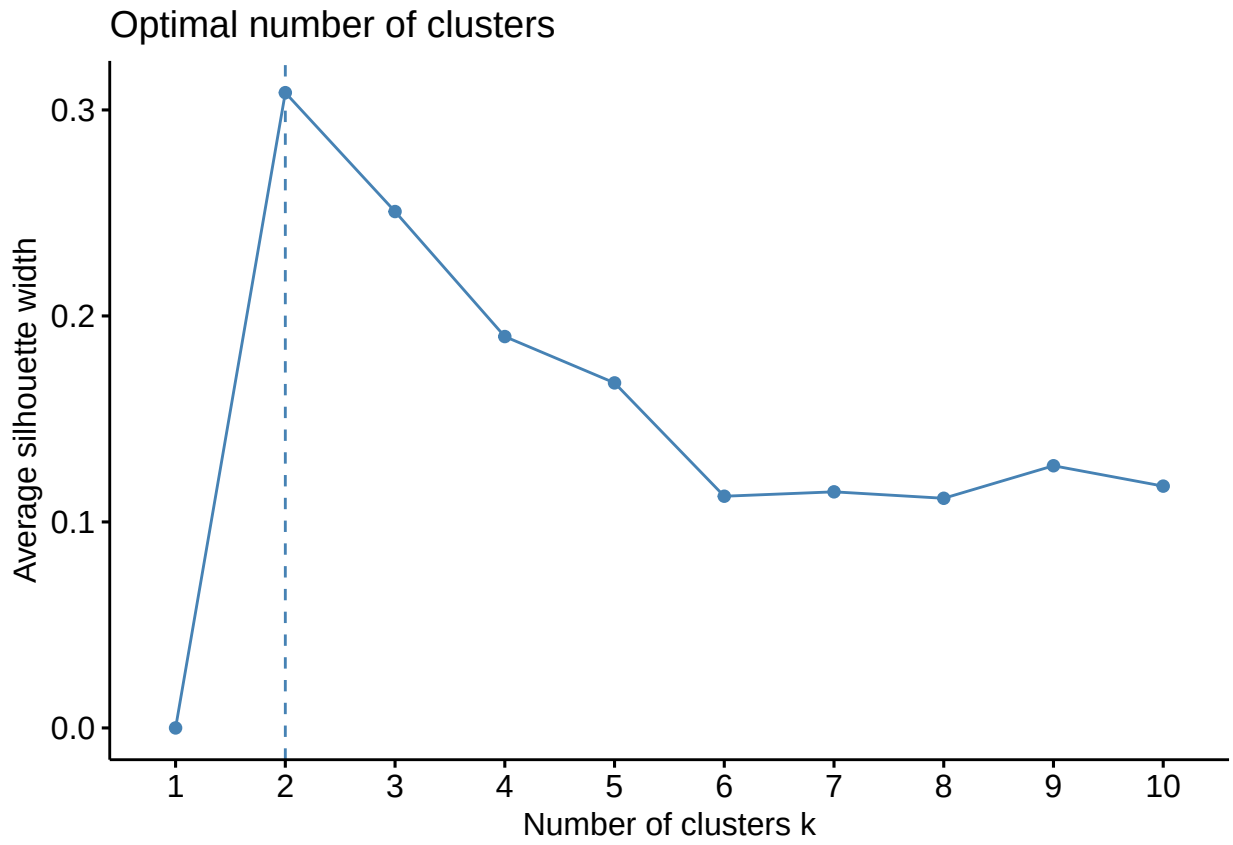
```
fviz_nbclust(completedf, kmeans, method = "wss")
```



Optimal Value of K=2. Meaning our data can be easily put into 2 clusters ie. customers are brand loyal or not.

Silhouette Method

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



K Means Implementation

Running K-Means for k=2

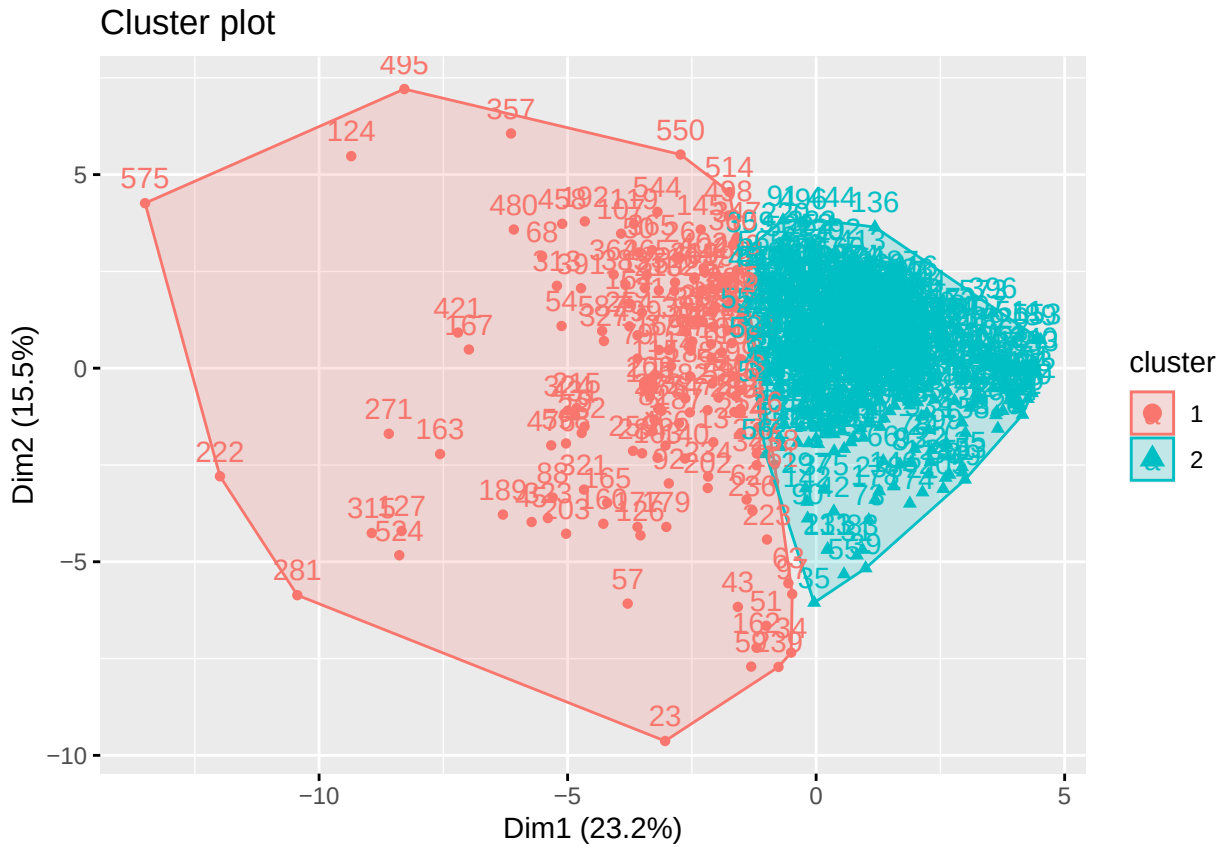
```
set.seed(123)
kcom <- kmeans(completedf, centers = 2, nstart = 25)

# Visualize the output

kcom$centers # output the centers
```

```
##   No..of.Brands Brand.Runs Total.Volume No..of..Trans      Value   Vol.Tran
## 1     0.3653371  0.5399761    1.3512359     0.8103701  1.1986675  0.7649979
## 2    -0.1164261 -0.1720803   -0.4306137    -0.2582498 -0.3819929 -0.2437905
##    Avg..Price Others.999 Max.brand.ind Pur.Vol.No.Promo.... Pur.Vol.Promo.6..
## 1 -0.26230920  1.0266951     0.5968897            1.2989110         0.4824972
## 2  0.08359304 -0.3271885    -0.1902176           -0.4139387        -0.1537629
##   Pur.Vol.Other.Promo..   Pr.Cat.1   Pr.Cat.2   Pr.Cat.3   Pr.Cat.4  PropCat.5
## 1             0.4666014  0.5370899  0.8472105  0.4810428  0.4289618  0.7801568
## 2            -0.1486972 -0.1711605 -0.2699901 -0.1532993 -0.1367021 -0.2486214
##    PropCat.6   PropCat.7   PropCat.8  PropCat.9  PropCat.10 PropCat.11
## 1  0.4935449  0.27595270  0.23915262  0.4665643  0.08452077  0.3050901
## 2 -0.1572835 -0.08794097 -0.07621347 -0.1486853 -0.02693519 -0.0972265
##   PropCat.12  PropCat.13 PropCat.14  PropCat.15
## 1  0.3304546  0.12568727  0.4757911  0.25890883
## 2 -0.1053097 -0.04005419 -0.1516257 -0.08250941
```

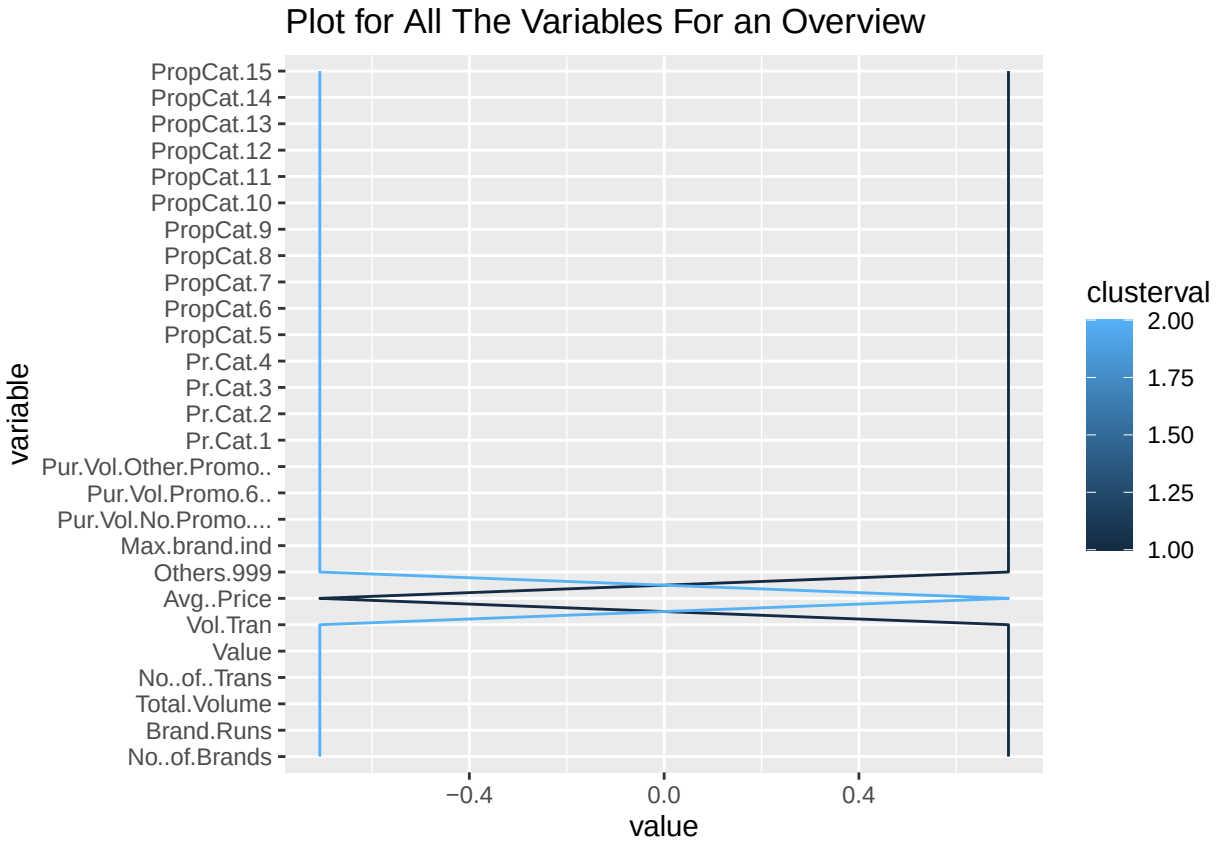
```
kcom$size # Number of Customers in each cluster
```

```
## [1] 145 455
```

```
fviz_cluster(kcom, data =completedf) # Visualize the output
```



Determining the Meaning of the clusters.



The clusters are pretty segregated.

Cluster 1 (145) Look to offers and promotions, not brand loyal. - people who buy low value soaps, they buy often, they look at deals and discounts.

Cluster 2 (455) - Don't believe in offers and discounts and loyal to high-end brands. - people that buy pricey soaps and have less no of transactions, less volume of transactions.

IV. Selecting The Best Segmentation.

Comment on the characteristics (demographic, brand loyalty, and basis for purchase) of these clusters.

I believe that the segmentation based on basis of purchase is of importance.

This segmentation gives us two clusters which are well separated across most variables.

Cluster 1 (60) - purchases without needing promotional offers, likes pricing category 3, and is somewhat responsive to selling propositions 6,9, and 14.

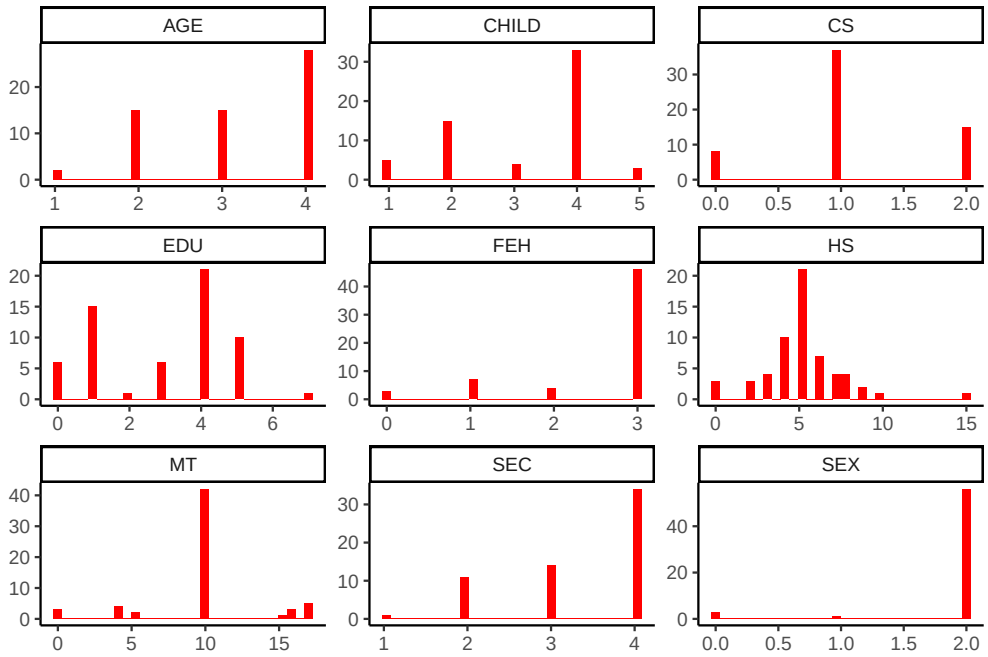
Cluster 2 (540) - Believe in promotions, high Pr.Cat.1,2 and 4. PropCat5, 7,8,10,11,12,13,and 15.

This shows that a huge number of customers are looking at deals and promotions, so it will do us good to understand the demographics of these people and present them with campaigns and promotions that drive them to buy.

Let's look at the various charachteristics:

Looking at demographic data Cluster 1:

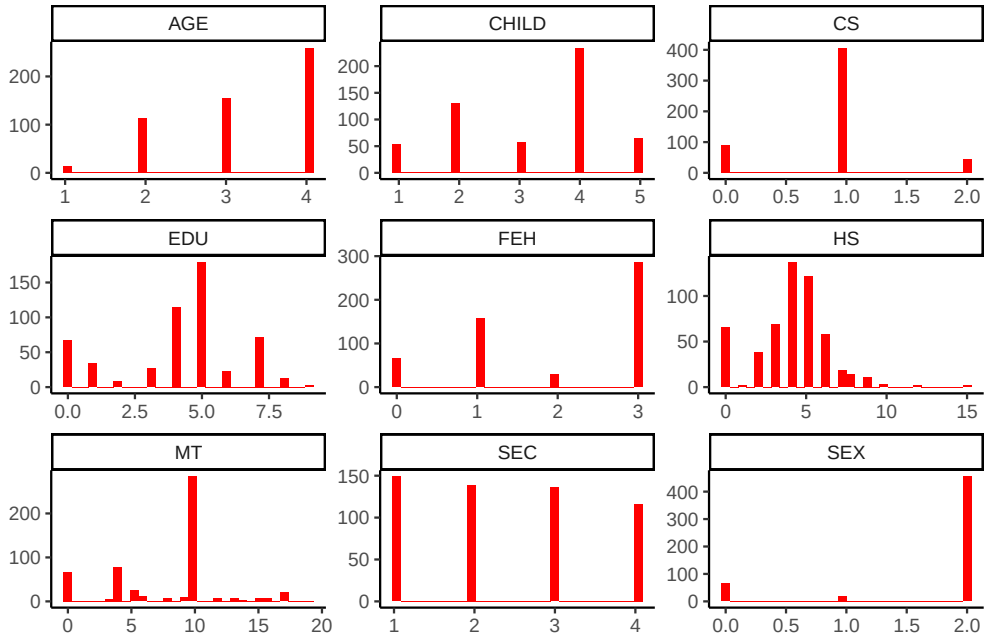
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Most of the shopping is done by females. Socioeconomic level is mostly high. Native language is generally 10, Education level is either very little or average. No. of people is the household is 4-5, so nuclear families. Eating habits are mostly non-veg. Age is higher under level 4. No. of children is also 2-4. Some of them do not have television.

Cluster 2:

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

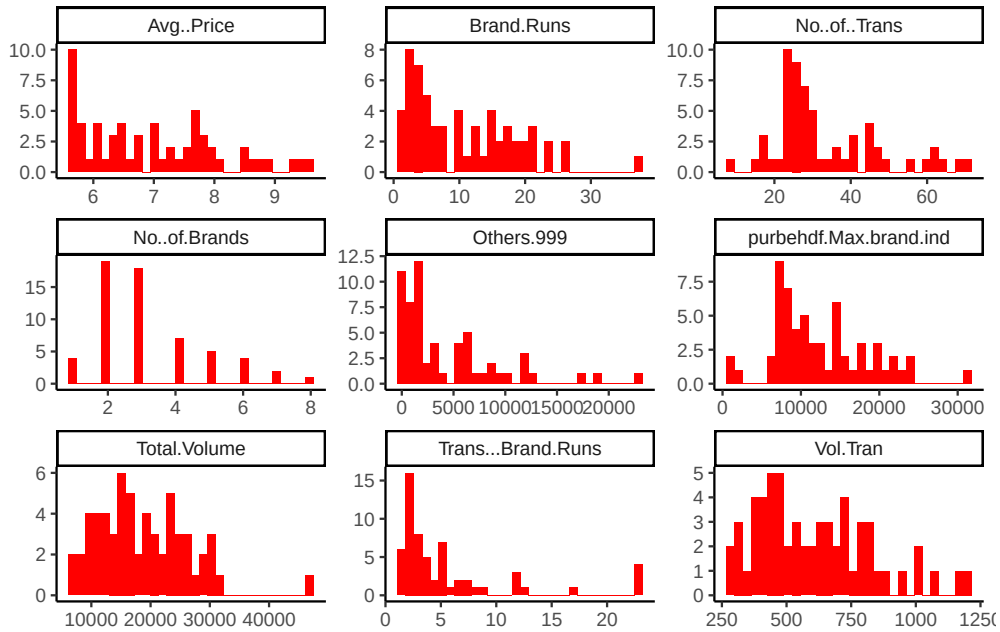


Most of the shopping is done by females. Socioeconomic level is evenly distributed across all levels. Native language is generally 10, Education level is average and higher. No. of people is the household is 5-6. Eating habits are mostly veg and non-veg. Age is higher under level 4 but also includes level 3. No. of children is also 2-4. Most of them have television.

Let's look at Brand Loyalty Charachteristcs:

Cluster 1:

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

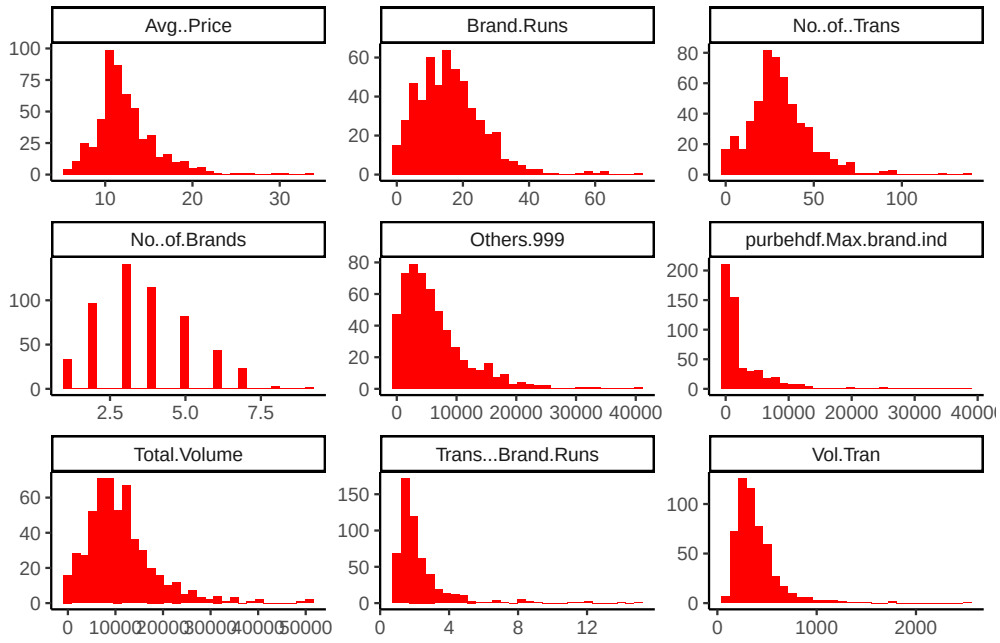


No. of Brands used is

pretty less 2-3. Average Volume per transaction is spread out but peaks on the average. Others.999 is low suggesting they are very loyal to their brand. Average price is pretty spread out.

Cluster 2:

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



No. of Brands used is on the higher side 3-4+. Average Volume per transaction is on the lower side. Others.999 is spread out a little. Max.brand.ind is low suggesting they don't stick with any specif brand. Average price is on the lower end.

V. Model To Classify Data into Success and Failure.

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.

We will create a logistic regression model that will help predict if a customer belongs to cluster 2 on the basis of the demographic data of the customers in cluster 2 and will help us decide if we should advertise to that customer or not basically giving us success or failure for running promotional campaigns.

```
# Preparing the dataset

cluster_val <- k2p$cluster
Soapdata_result <- cbind(SoapData_df, cluster_val)


Soapdata_result[, 49:51] <- dummy_cols(Soapdata_result$cluster_val)

# Partition data 60% train and 40% validation

set.seed(123)
Valid_Index = createDataPartition(Soapdata_result$Brand.Runs,p=0.4, list=FALSE) # 40% reserved for Vali
Valid_Data = Soapdata_result[Valid_Index,]
Train_Data = Soapdata_result[-Valid_Index,] # Validation and Training data is rest

# Choosing variables based on demographics and a few decisive variables that put customers in cluster 1

# Applying logistic regression model
modelreg <- glm(formula = .data_1 ~  SEC+ MT + FEH + SEX + AGE+ EDU + HS + CHILD + CS +  Pr.Cat.3 + Pur


predict_validation<-predict(modelreg, newdata = Valid_Data, type='response')
```

Categorizing the result based on the cutoff value(0.5)

```
resultval <-as.factor(ifelse(predict_validation > 0.5, 1, 0))

CrossTable(x=Valid_Data$.data_1,y=resultval, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  242
##
```

```
##
##                     | resultval
## Valid_Data$.data_1 |         0 |         1 | Row Total |
## -------------------|-----------|-----------|-----------|
##                  0 |       216 |         3 |       219 |
##                    |     0.986 |     0.014 |     0.905 |
##                    |     0.995 |     0.120 |           |
##                    |     0.893 |     0.012 |           |
## -------------------|-----------|-----------|-----------|
##                  1 |         1 |        22 |        23 |
##                    |     0.043 |     0.957 |     0.095 |
##                    |     0.005 |     0.880 |           |
##                    |     0.004 |     0.091 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |       217 |        25 |       242 |
##                    |     0.897 |     0.103 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

This matrix shows the following: 0 is Success , then the misclassifications are 4 false positives, and 21 false negatives. We can identify several measures based on this table. For example

- Accuracy = Number correctly identified / Total = (21 + 216) / 242 = .98
- Recall is the true positive rate or sensitivity = 21/21+ 1 = .95
- Precision is the positive predictive value = 21 / (21 + 4) = 0.84
- Specificity, also called as the true negative rate = 216 / 217 = .99

In simple terms, No. of customers correctly identified is pretty high with an accuracy of 0.98.

High precision means that an algorithm returned substantially more relevant (positive) results than irrelevant (negative) ones, while high recall means that an algorithm returned most of the relevant (positive) results.

So Now using the predict function of this model, we can help people at CRISA/a marketing company to understand if they should run advertisement or give promotions and discounts to a specific customer or if the customer would buy the product anyways.