CRISA Consumer Segmentation Report

Fabrizio Fiorini

5/8/2021

# Kent State University

# Fundamentals of Machine Learning – Spring 2021

# Abstract

The objective of this assignment is to apply the appropriate machine learning technique to the business problem, and then present the solution to top-level management.

# Company’s profile

CRISA is an Asian market research agency that specializes in tracking consumer purchase behavior in consumer goods (both durable and nondurable). In one major research project, CRISA tracks numerous consumer product categories (e.g., “detergents”), and, within each category, perhaps dozens of brands. To track purchase behavior, CRISA constituted household panels in over 100 cities and towns in India, covering most of the Indian urban market. The households were carefully selected using stratified sampling to ensure a representative sample; 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 has two categories of clients:

* advertising agencies that subscribe to the database services, obtain updated data every month, and use the data to advise their clients on advertising and promotion strategies;
* consumer goods manufacturers, which monitor their market share using the CRISA database.

# Introduction to the study

CRISA has traditionally segmented markets on the basis of purchaser demographics. They 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) and 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. More effective market segmentation would enable CRISA’s clients (in this case, a firm called IMRB) to design more cost-effective promotions targeted at appropriate segments. Thus, multiple promotions could be launched, each targeted at different market segments at different times of the year. This would result in a more cost-effective allocation of the promotion budget to different market segments. It would also enable IMRB to design more effective customer reward systems and thereby increase brand loyalty. In this study we are going to identify clusters of households applying the unsupervised learning algorithm K-means using different sets of variables, as stated above. Then, we are going to comment on the characteristics of these clusters such as demographic, brand loyalty, and purchase behavior. This information would be used to guide the development of advertising and promotional campaigns. Finally, we are going to develop a model that classifies data into these clusters and identifies which one should be used in targeting direct-mail promotions.

# Data exploration

#3.data exploration  
dim(dataset)

## [1] 600 46

head(dataset)

## Member.id SEC FEH MT SEX AGE EDU HS CHILD CS Affluence.Index No..of.Brands  
## 1 1010010 4 3 10 1 4 4 2 4 1 2 3  
## 2 1010020 3 2 10 2 2 4 4 2 1 19 5  
## 3 1014020 2 3 10 2 4 5 6 4 1 23 5  
## 4 1014030 4 0 0 0 4 0 0 5 0 0 2  
## 5 1014190 4 1 10 2 3 4 4 3 1 10 3  
## 6 1017020 4 3 10 2 3 4 5 2 1 13 3  
## Brand.Runs Total.Volume No..of..Trans Value Trans...Brand.Runs Vol.Tran  
## 1 17 8025 24 818.0 1.41 334.38  
## 2 25 13975 40 1681.5 1.60 349.38  
## 3 37 23100 63 1950.0 1.70 366.67  
## 4 4 1500 4 114.0 1.00 375.00  
## 5 6 8300 13 591.0 2.17 638.46  
## 6 26 18175 41 1705.5 1.58 443.29  
## Avg..Price Pur.Vol.No.Promo.... Pur.Vol.Promo.6.. Pur.Vol.Other.Promo..  
## 1 10.19 100% 0% 0%  
## 2 12.03 89% 10% 2%  
## 3 8.44 94% 2% 4%  
## 4 7.60 100% 0% 0%  
## 5 7.12 61% 14% 24%  
## 6 9.38 100% 0% 0%  
## Br..Cd..57..144 Br..Cd..55 Br..Cd..272 Br..Cd..286 Br..Cd..24 Br..Cd..481  
## 1 38% 13% 0% 0% 0% 0%  
## 2 2% 8% 0% 0% 0% 6%  
## 3 3% 55% 0% 3% 0% 0%  
## 4 40% 60% 0% 0% 0% 0%  
## 5 5% 14% 0% 0% 0% 0%  
## 6 8% 7% 0% 0% 0% 0%  
## Br..Cd..352 Br..Cd..5 Others.999 Pr.Cat.1 Pr.Cat.2 Pr.Cat.3 Pr.Cat.4  
## 1 0% 0% 49.2% 23% 56% 13% 7%  
## 2 0% 14% 69.9% 29% 55% 9% 6%  
## 3 0% 2% 37.9% 12% 32% 56% 0%  
## 4 0% 0% 0.0% 0% 40% 60% 0%  
## 5 0% 0% 80.7% 0% 5% 14% 81%  
## 6 0% 0% 85.7% 22% 45% 7% 27%  
## PropCat.5 PropCat.6 PropCat.7 PropCat.8 PropCat.9 PropCat.10 PropCat.11  
## 1 50% 0% 0% 0% 0% 0% 0%  
## 2 46% 35% 3% 2% 1% 0% 6%  
## 3 24% 12% 3% 1% 1% 0% 0%  
## 4 40% 0% 0% 0% 0% 0% 0%  
## 5 81% 0% 0% 5% 0% 0% 0%  
## 6 49% 10% 0% 1% 7% 0% 0%  
## PropCat.12 PropCat.13 PropCat.14 PropCat.15  
## 1 3% 0% 13% 34%  
## 2 0% 0% 8% 0%  
## 3 2% 0% 56% 0%  
## 4 0% 0% 60% 0%  
## 5 0% 0% 14% 0%  
## 6 0% 0% 7% 27%

str(dataset)

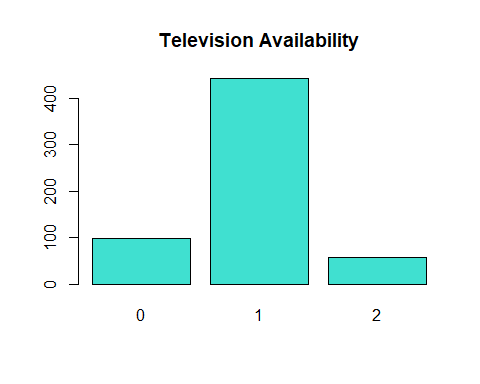
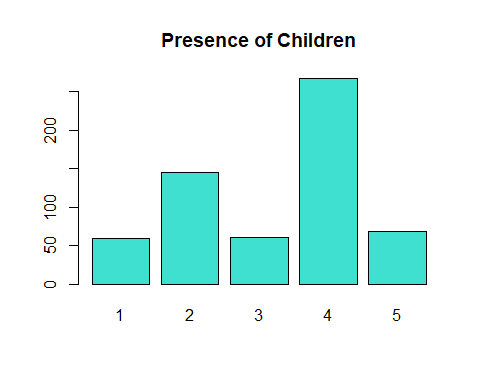
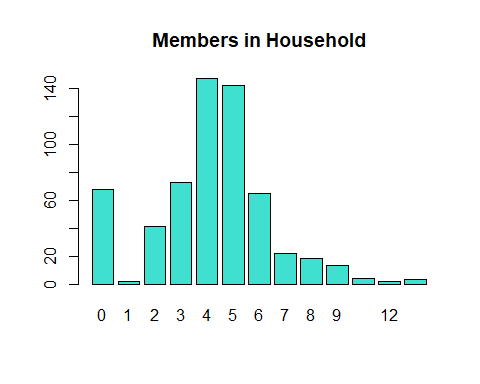
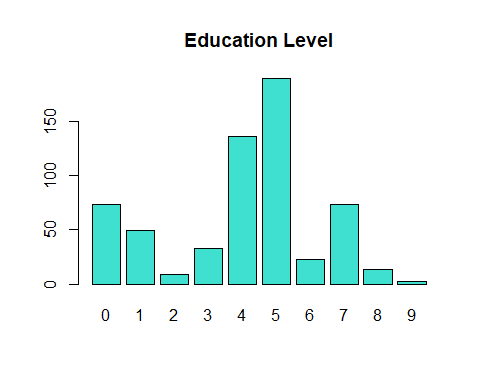
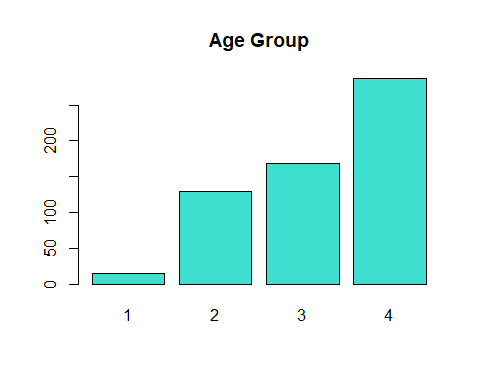
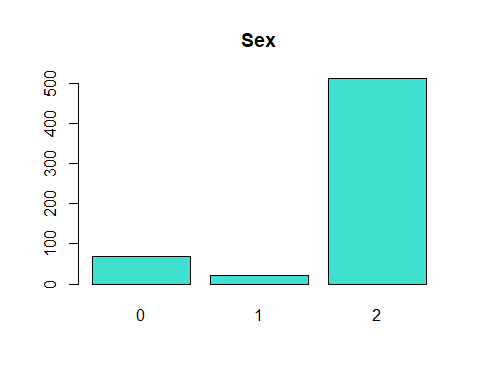
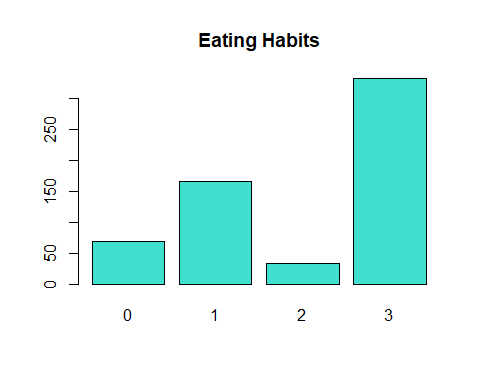
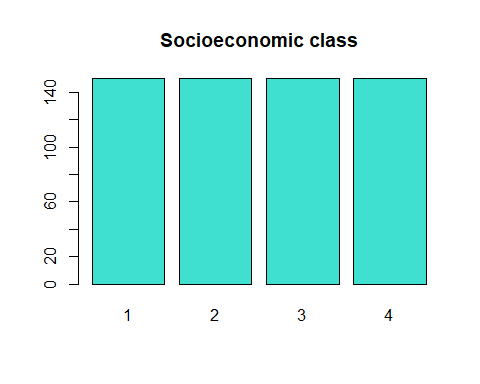
## 'data.frame': 600 obs. of 46 variables:  
## $ Member.id : int 1010010 1010020 1014020 1014030 1014190 1017020 1017110 1017160 1017360 1017460 ...  
## $ SEC : int 4 3 2 4 4 4 4 4 4 1 ...  
## $ FEH : int 3 2 3 0 1 3 2 3 3 3 ...  
## $ MT : int 10 10 10 0 10 10 10 10 10 5 ...  
## $ SEX : int 1 2 2 0 2 2 2 2 2 1 ...  
## $ AGE : int 4 2 4 4 3 3 4 2 4 4 ...  
## $ EDU : int 4 4 5 0 4 4 1 4 4 7 ...  
## $ HS : int 2 4 6 0 4 5 3 5 6 3 ...  
## $ CHILD : int 4 2 4 5 3 2 2 3 4 4 ...  
## $ CS : int 1 1 1 0 1 1 1 0 1 1 ...  
## $ Affluence.Index : int 2 19 23 0 10 13 11 0 17 6 ...  
## $ No..of.Brands : int 3 5 5 2 3 3 4 3 2 4 ...  
## $ Brand.Runs : int 17 25 37 4 6 26 17 8 12 13 ...  
## $ Total.Volume : int 8025 13975 23100 1500 8300 18175 9950 9300 26490 7455 ...  
## $ No..of..Trans : int 24 40 63 4 13 41 26 25 27 18 ...  
## $ Value : num 818 1682 1950 114 591 ...  
## $ Trans...Brand.Runs : num 1.41 1.6 1.7 1 2.17 1.58 1.53 3.13 2.25 1.38 ...  
## $ Vol.Tran : num 334 349 367 375 638 ...  
## $ Avg..Price : num 10.19 12.03 8.44 7.6 7.12 ...  
## $ Pur.Vol.No.Promo.... : chr "100%" "89%" "94%" "100%" ...  
## $ Pur.Vol.Promo.6.. : chr "0%" "10%" "2%" "0%" ...  
## $ Pur.Vol.Other.Promo..: chr "0%" "2%" "4%" "0%" ...  
## $ Br..Cd..57..144 : chr "38%" "2%" "3%" "40%" ...  
## $ Br..Cd..55 : chr "13%" "8%" "55%" "60%" ...  
## $ Br..Cd..272 : chr "0%" "0%" "0%" "0%" ...  
## $ Br..Cd..286 : chr "0%" "0%" "3%" "0%" ...  
## $ Br..Cd..24 : chr "0%" "0%" "0%" "0%" ...  
## $ Br..Cd..481 : chr "0%" "6%" "0%" "0%" ...  
## $ Br..Cd..352 : chr "0%" "0%" "0%" "0%" ...  
## $ Br..Cd..5 : chr "0%" "14%" "2%" "0%" ...  
## $ Others.999 : chr "49.2%" "69.9%" "37.9%" "0.0%" ...  
## $ Pr.Cat.1 : chr "23%" "29%" "12%" "0%" ...  
## $ Pr.Cat.2 : chr "56%" "55%" "32%" "40%" ...  
## $ Pr.Cat.3 : chr "13%" "9%" "56%" "60%" ...  
## $ Pr.Cat.4 : chr "7%" "6%" "0%" "0%" ...  
## $ PropCat.5 : chr "50%" "46%" "24%" "40%" ...  
## $ PropCat.6 : chr "0%" "35%" "12%" "0%" ...  
## $ PropCat.7 : chr "0%" "3%" "3%" "0%" ...  
## $ PropCat.8 : chr "0%" "2%" "1%" "0%" ...  
## $ PropCat.9 : chr "0%" "1%" "1%" "0%" ...  
## $ PropCat.10 : chr "0%" "0%" "0%" "0%" ...  
## $ PropCat.11 : chr "0%" "6%" "0%" "0%" ...  
## $ PropCat.12 : chr "3%" "0%" "2%" "0%" ...  
## $ PropCat.13 : chr "0%" "0%" "0%" "0%" ...  
## $ PropCat.14 : chr "13%" "8%" "56%" "60%" ...  
## $ PropCat.15 : chr "34%" "0%" "0%" "0%" ...

summary(dataset)

## 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   
##   
##   
##

The dataset we are using is called BathSoap and it is provided in an Excel CSV file. It contains 600 observations and 46 variables. These variables are of different nature: Demographics, Purchase summary over the period, Purchase within promotion, Brandwise purchase, Price category, Selling proposition. After carried out some data exploration activity, numerous variables resulted to be classified as character due to the presence of the “%” symbol within the data points. We will take care of this later on. Moreover, it is important to point out that some variables that should be, according to their nature, categorical, are presented as numeric variables. For example, the variable “SEC” (socioeconomic class), “FEH” (eating habits), “SEX” (gender), “AGE”, “EDU” (education), and “CS” (television availability) are expressed with a range of number indicating the category in which the data point falls. For example, the variable “SEC” has a range that goes from 1 (high) to 5 (low), while “SEX” can be 1 for male and 2 for female.

When we checked for missing values, we notice that there are not any. However, from the descriptive statistics analysis, we saw that some of the demographic variables have zero as minimum value, meaning that the dataset contains zero values that can be considered as “missing values” or “values on available”. In fact, the categorical variables have levels that start from 1. If we try to filter for zeros using three demographic variables (for example FEH, EDU, HS) we see that numerous observations are lacking more than one of those variables. Nonetheless, since the K-means clustering algorithm does not consider the categorical variables regarding the households’ demographic, the observations that contain zero values are still considered useful for the purpose of the study. The following visuals offer an easy and effective understanding of the distribution of the observations through the demographic characteristics.



# Data preparation

#6.removing % and preparing data  
dataset.mod <- dataset  
#dataset.mod[,12:19] <- apply(dataset.mod[,12:19],2,as.numeric)  
dataset.mod[,20:46] <- data.frame(lapply(dataset.mod[,20:46], function(x) as.numeric(sub("%", "", x))))  
dataset.mod[,20:46] <- lapply(dataset.mod[,20:46], function(x) as.numeric(x)/100)

As we have seen, the original database form prevented us from carrying out the clustering analysis on the data because of the presence of the “%” symbol next to the value. An essential step in the process was to remove the symbol and have those values set as numeric. Moreover, we decided to show percentage values as decimal points. To express the Brand Loyalty of each household based on their purchase data available in the dataset, we decided to create a metric that tells the maximum value recorded for the percentage spent. A loyal household has a high percentage of spending concentrated on a type of brand, whichever it is.

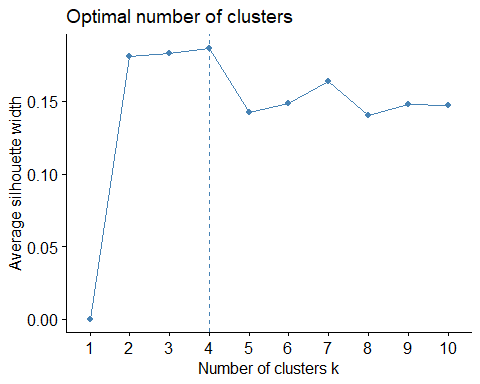
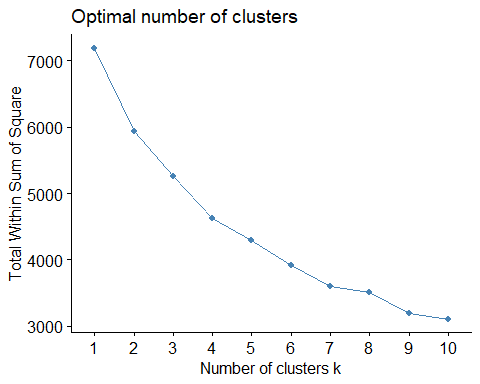
#7.creating brand loyalty metrics  
dataset.mod <- mutate(dataset.mod, Brand\_Loyalty = apply(dataset.mod[,23:30],1,max))

Last step before applying the K-means clustering algorithm was to normalize the data, since one of the main disadvantages of it is its sensibility to variables’ scale, and to select three different sets of variables. First, we used variables that express the Purchase Behavior (see the Introduction section), then we applied the algorithm to a set of variables describing the Basis for Purchase (see the Introduction section), and finally we combined those sets of variables and run again the clustering analysis. Not all the variables of the original dataset have been utilized, limiting the analysis on the variables from 12th to 47th (the last variables is the Brand Loyalty metric we created).

#8.normalizing and partitioning  
dataset.norm <- dataset.mod  
dataset.norm[,12:47] <- scale(dataset.mod[,12:47])  
purchasebehavior <- dataset.norm[, c(12:22,47)]  
basispurchase <- dataset.norm[, c(32:46)]  
behavior.basis <- dataset.norm[, c(12:22,32:47)]

# Clustering analysis – Purchase behavior

When applying the K-means algorithm, the user has to decide the number of clusters desired as result. In fact, the number of cluster k is an hyperparameter of the algorithm. The decision can be based on several factor (such as external considerations or domain knowledge), however an effective way to know in advance what level of k is optimal is to resort to the Elbow and the Silhouette methods.



According to both the charts, we decided to use four clusters for running the clustering analysis with variables describing Purchase Behavior. The result, visualized below, is clusters with size of, respectively, 167, 80, 73, 280. The output of the algorithm also shows us the centers, the within cluster sum of squares, and the series of cluster number for each observation (the first household is assigned to cluster 4, the second and third to cluster 1, the fourth to cluster 4, and so on).

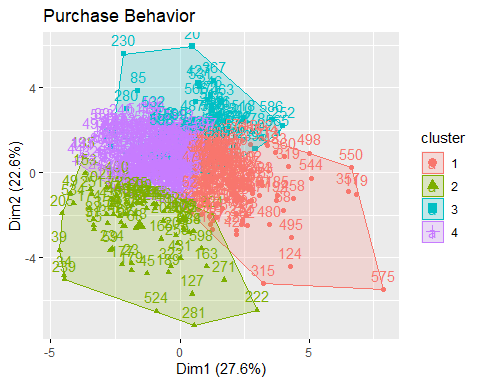
#10.running kmeans for purchase behavior  
set.seed(123)  
kmeansPB4 <- kmeans(purchasebehavior, centers = 4, nstart = 30)  
print(kmeansPB4)

## K-means clustering with 4 clusters of sizes 167, 80, 73, 280  
##   
## Cluster means:  
## No..of.Brands Brand.Runs Total.Volume No..of..Trans Value  
## 1 0.93883973 1.0133966 0.5205349 1.0622059 0.6668780  
## 2 -0.69580314 -0.7973051 1.1018748 -0.3358723 0.5493651  
## 3 -0.05616316 0.2544701 -0.4434796 -0.1070543 -0.3431819  
## 4 -0.34650740 -0.4429613 -0.5096618 -0.5096559 -0.4652341  
## Trans...Brand.Runs Vol.Tran Avg..Price Pur.Vol.No.Promo....  
## 1 -0.2210892 -0.3061882 0.1525072 0.05961055  
## 2 1.2880999 1.6821523 -0.8898174 0.39254646  
## 3 -0.3499923 -0.3786967 0.2355096 -2.04129086  
## 4 -0.1449167 -0.1992639 0.1018732 0.38448412  
## Pur.Vol.Promo.6.. Pur.Vol.Other.Promo.. Brand\_Loyalty  
## 1 -0.06744593 -0.006705676 -0.4876751  
## 2 -0.47360110 -0.035320952 1.1145603  
## 3 1.89088461 0.942426139 -0.4915982  
## 4 -0.31743935 -0.231612800 0.1005842  
##   
## Clustering vector:  
## [1] 4 1 1 4 3 1 4 4 2 4 1 1 2 4 4 1 1 4 4 3 3 2 2 4 4 1 2 4 4 2 2 4 4 2 2 1 4  
## [38] 4 2 4 4 2 2 4 2 4 4 4 2 1 2 2 4 1 2 4 2 1 2 4 1 1 2 4 1 4 1 1 4 2 4 1 2 3  
## [75] 4 4 4 4 1 4 4 1 2 3 3 1 3 2 4 2 1 1 4 4 1 1 2 1 4 4 4 4 2 3 4 1 1 4 1 3 2  
## [112] 4 4 3 3 1 1 4 1 1 3 1 2 1 2 2 2 4 4 1 1 1 4 4 2 3 4 1 4 2 2 3 4 4 1 4 4 3  
## [149] 1 3 4 4 2 4 1 1 4 4 4 1 4 2 2 1 2 1 1 4 1 4 1 4 4 2 1 1 2 2 2 4 4 4 1 3 1  
## [186] 2 1 1 2 3 4 1 4 4 1 4 1 3 1 4 4 2 2 4 2 4 4 2 4 1 4 4 1 2 1 4 4 3 4 1 4 2  
## [223] 2 4 4 4 2 1 4 3 4 1 2 2 4 2 1 2 2 4 1 1 3 4 4 4 4 4 4 4 2 3 4 1 4 4 4 3 2  
## [260] 1 4 4 1 4 1 4 4 4 4 1 2 2 4 3 4 3 1 1 4 3 2 4 1 2 1 4 4 4 1 1 3 4 4 3 4 4  
## [297] 4 1 1 4 4 4 4 1 4 4 1 4 3 4 4 3 1 1 1 1 4 4 1 4 2 3 2 1 4 4 1 4 4 4 1 4 4  
## [334] 3 4 3 1 1 4 4 4 4 2 4 1 3 3 4 1 4 4 4 4 2 1 1 1 1 1 1 4 4 1 1 1 3 3 1 4 4  
## [371] 4 4 4 4 2 3 4 4 1 1 4 4 1 4 1 4 4 1 1 1 1 1 4 4 3 4 4 4 4 2 4 1 4 1 4 4 4  
## [408] 4 1 1 4 4 3 4 4 3 1 4 4 4 1 4 4 4 4 1 3 4 1 4 2 4 4 4 4 2 4 1 4 4 4 4 1 4  
## [445] 4 4 4 4 4 4 4 4 2 4 4 4 4 1 1 1 3 3 3 4 1 1 4 4 4 3 3 4 1 1 2 1 1 3 4 1 4  
## [482] 3 1 3 4 4 3 1 3 1 3 1 4 2 1 1 4 1 1 1 4 1 4 4 4 4 4 4 1 3 4 4 4 1 3 4 4 3  
## [519] 4 1 4 4 4 2 4 4 1 4 1 4 3 3 1 2 4 4 4 4 4 4 4 3 1 1 3 4 4 4 1 1 4 4 4 1 1  
## [556] 4 4 3 3 4 3 2 1 1 3 3 1 3 4 1 4 1 4 1 1 4 1 4 4 3 3 4 4 3 4 3 1 1 4 1 4 4  
## [593] 3 4 4 4 4 2 3 2  
##   
## Within cluster sum of squares by cluster:  
## [1] 1140.3539 1048.8405 963.1328 1466.2982  
## (between\_SS / total\_SS = 35.7 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

table(kmeansPB4$cluster)

##   
## 1 2 3 4   
## 167 80 73 280

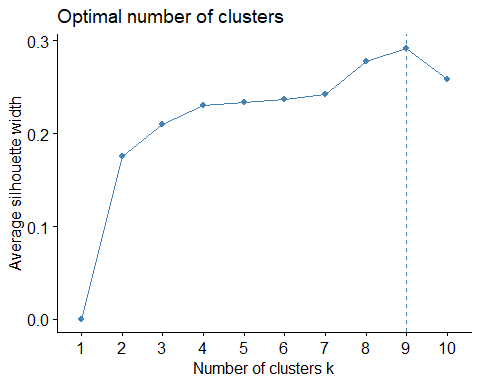
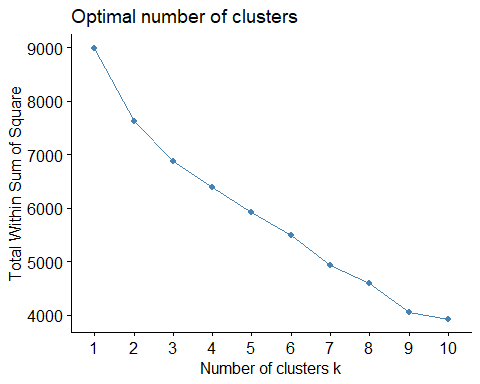
fviz\_cluster(kmeansPB4, data = purchasebehavior) + labs(title = "Purchase Behavior")



#fviz\_cluster(kmeansPB, purchasebehavior, main = "Purchase Behavior Cluster Plot")  
kmeansPB4$centers

## No..of.Brands Brand.Runs Total.Volume No..of..Trans Value  
## 1 0.93883973 1.0133966 0.5205349 1.0622059 0.6668780  
## 2 -0.69580314 -0.7973051 1.1018748 -0.3358723 0.5493651  
## 3 -0.05616316 0.2544701 -0.4434796 -0.1070543 -0.3431819  
## 4 -0.34650740 -0.4429613 -0.5096618 -0.5096559 -0.4652341  
## Trans...Brand.Runs Vol.Tran Avg..Price Pur.Vol.No.Promo....  
## 1 -0.2210892 -0.3061882 0.1525072 0.05961055  
## 2 1.2880999 1.6821523 -0.8898174 0.39254646  
## 3 -0.3499923 -0.3786967 0.2355096 -2.04129086  
## 4 -0.1449167 -0.1992639 0.1018732 0.38448412  
## Pur.Vol.Promo.6.. Pur.Vol.Other.Promo.. Brand\_Loyalty  
## 1 -0.06744593 -0.006705676 -0.4876751  
## 2 -0.47360110 -0.035320952 1.1145603  
## 3 1.89088461 0.942426139 -0.4915982  
## 4 -0.31743935 -0.231612800 0.1005842

# Clustering analysis – Basis for purchase



We applied again the Elbow and the Silhouette methods, this time using the Basis for Purchase set of variables. From the Elbow chart we can see that k=2 already ensure a good performance, even if the line keeps declining steadily until k=7. The Silhouette method suggests the use of 7 clusters and we can notice that after k=3 the line starts to flatten. One of the guidelines of the client was the intention to support two to five different promotional approaches that will be based on the clustering analysis. Therefore, k=3 has been considered the best solution for our study, considering the output of the two methods.

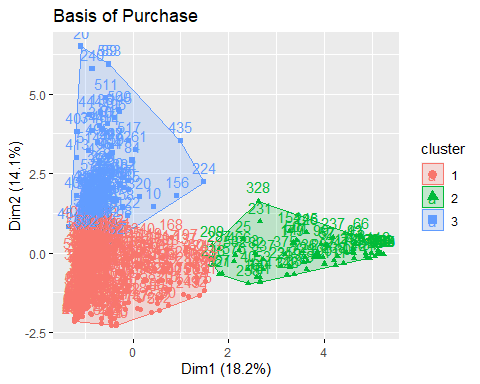
#12.running kmeans for basis for purchase  
set.seed(123)  
kmeansBfP3 <- kmeans(basispurchase, centers = 3, nstart = 30)  
print(kmeansBfP3)

## K-means clustering with 3 clusters of sizes 376, 79, 145  
##   
## Cluster means:  
## Pr.Cat.1 Pr.Cat.2 Pr.Cat.3 Pr.Cat.4 PropCat.5 PropCat.6 PropCat.7  
## 1 -0.4145980 0.5195713 -0.3131789 0.2006057 0.3913062 -0.0626521 -0.04672354  
## 2 -0.7811336 -1.1192162 2.3533589 -0.3222704 -1.0855323 -0.1711945 -0.44395337  
## 3 1.5006786 -0.7375224 -0.4700697 -0.3446096 -0.4232694 0.2557349 0.36303702  
## PropCat.8 PropCat.9 PropCat.10 PropCat.11 PropCat.12 PropCat.13  
## 1 -0.008182177 0.01903935 -0.1545182 0.1095443 -0.09946207 -0.2013688  
## 2 -0.458786014 -0.16641740 -0.2571876 -0.2304083 -0.16393967 -0.2328841  
## 3 0.271176507 0.04129779 0.5408044 -0.1585269 0.34723429 0.6490519  
## PropCat.14 PropCat.15  
## 1 -0.3162083 0.03460623  
## 2 2.3559150 -0.21596239  
## 3 -0.4636068 0.02792474  
##   
## Clustering vector:  
## [1] 1 1 2 2 1 1 1 2 1 3 1 1 1 1 1 1 1 1 1 3 2 2 2 2 2 3 1 1 1 2 2 1 2 2 2 1 1  
## [38] 1 2 2 1 2 2 2 1 1 1 1 2 3 2 1 2 1 2 1 2 1 2 1 1 2 2 1 1 2 2 3 1 1 1 1 2 1  
## [75] 1 3 1 2 1 1 1 2 2 3 1 1 1 1 3 2 3 1 2 3 1 1 2 1 2 1 1 1 1 1 1 1 3 1 1 1 2  
## [112] 1 1 1 1 3 1 2 1 1 1 3 1 3 1 2 1 1 1 3 1 1 3 1 2 3 1 1 1 1 1 2 1 2 3 2 2 1  
## [149] 1 1 1 1 2 2 1 3 3 2 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1  
## [186] 1 1 1 1 3 1 3 1 1 1 1 1 1 3 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 3 1 3 2 3 1 1  
## [223] 2 3 1 1 1 1 3 1 2 1 2 1 2 2 2 2 2 3 1 1 1 1 2 2 2 3 3 1 1 1 1 1 1 1 1 2 1  
## [260] 1 3 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 2 1 1 3 1 3 1 1 1 1  
## [297] 2 1 1 3 1 1 1 1 3 1 1 1 1 3 3 1 1 3 1 1 1 1 1 3 2 3 1 1 1 2 1 2 1 1 1 1 1  
## [334] 1 1 1 1 1 1 1 1 1 1 3 1 1 3 3 1 1 1 3 3 1 3 1 3 1 3 3 3 3 1 3 3 3 1 1 3 1  
## [371] 1 3 1 1 2 3 1 3 1 1 3 3 1 3 1 3 1 1 1 1 3 3 3 3 1 3 3 1 3 1 1 3 1 3 1 3 3  
## [408] 1 1 1 1 3 3 1 3 3 1 3 1 1 1 1 3 1 1 1 1 3 3 1 1 1 1 3 3 1 3 3 1 3 1 3 1 3  
## [445] 3 1 1 1 1 1 3 1 1 3 1 1 3 3 3 3 1 3 1 3 3 2 1 1 3 3 3 3 1 1 1 1 1 3 3 3 3  
## [482] 3 1 3 3 1 3 1 3 3 1 1 3 1 3 3 3 3 1 1 2 3 1 1 3 3 1 1 1 1 3 1 1 3 1 1 3 1  
## [519] 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 3 1 3 1 1 1 1 1 3 1 3 3 1 1  
## [556] 1 1 1 3 3 3 1 3 1 1 1 1 3 1 3 1 1 3 1 1 2 3 3 3 1 1 3 1 1 1 1 1 1 1 1 1 3  
## [593] 1 1 1 1 3 1 1 1  
##   
## Within cluster sum of squares by cluster:  
## [1] 3760.399 240.155 2779.040  
## (between\_SS / total\_SS = 24.5 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

table(kmeansBfP3$cluster)

##   
## 1 2 3   
## 376 79 145

fviz\_cluster(kmeansBfP3, data = basispurchase) + labs(title = "Basis of Purchase")

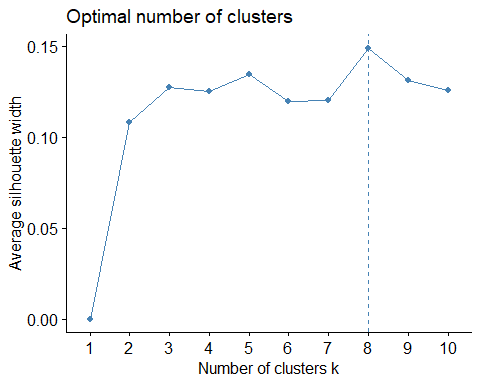
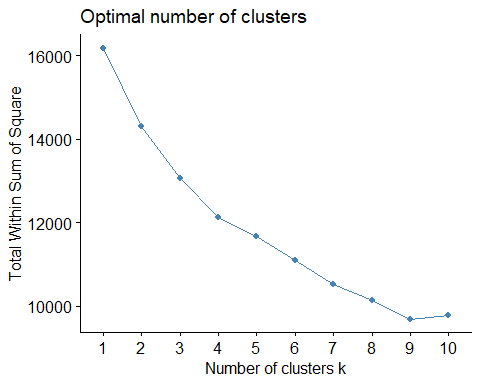


#fviz\_cluster(kmeansBfP, basispurchase, main = "Basis of Purchase Cluster Plot")  
kmeansBfP3$centers

## Pr.Cat.1 Pr.Cat.2 Pr.Cat.3 Pr.Cat.4 PropCat.5 PropCat.6 PropCat.7  
## 1 -0.4145980 0.5195713 -0.3131789 0.2006057 0.3913062 -0.0626521 -0.04672354  
## 2 -0.7811336 -1.1192162 2.3533589 -0.3222704 -1.0855323 -0.1711945 -0.44395337  
## 3 1.5006786 -0.7375224 -0.4700697 -0.3446096 -0.4232694 0.2557349 0.36303702  
## PropCat.8 PropCat.9 PropCat.10 PropCat.11 PropCat.12 PropCat.13  
## 1 -0.008182177 0.01903935 -0.1545182 0.1095443 -0.09946207 -0.2013688  
## 2 -0.458786014 -0.16641740 -0.2571876 -0.2304083 -0.16393967 -0.2328841  
## 3 0.271176507 0.04129779 0.5408044 -0.1585269 0.34723429 0.6490519  
## PropCat.14 PropCat.15  
## 1 -0.3162083 0.03460623  
## 2 2.3559150 -0.21596239  
## 3 -0.4636068 0.02792474

The three clusters formed have size of, respectively, 376, 79, 145.

# Clustering analysis – Combined



Now we run the algorithm combining the two set of variables used previously. The hyperparameter has been set on 3 since Silhouette chart shows a slight decrease for k=4 compared to k=3.

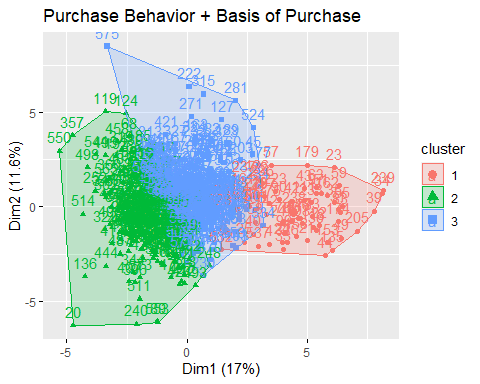
#14.running kmeans for both purchase behavior and basis for purchase  
set.seed(123)  
kmeansPB.BfP3 <- kmeans(behavior.basis, centers = 3, nstart = 30)  
print(kmeansPB.BfP3)

## K-means clustering with 3 clusters of sizes 68, 219, 313  
##   
## Cluster means:  
## No..of.Brands Brand.Runs Total.Volume No..of..Trans Value  
## 1 -0.598521651 -0.8120867 0.09190381 -0.42903264 -0.55575082  
## 2 0.189532593 0.3915028 -0.34789649 0.22334716 0.02008816  
## 3 -0.002581999 -0.0974991 0.22345007 -0.06306329 0.10668291  
## Trans...Brand.Runs Vol.Tran Avg..Price Pur.Vol.No.Promo....  
## 1 1.07696718 0.5271963 -1.3330597 0.2413117  
## 2 -0.22834629 -0.5299567 0.8870873 -0.3717739  
## 3 -0.07420425 0.2562657 -0.3310673 0.2076974  
## Pur.Vol.Promo.6.. Pur.Vol.Other.Promo.. Pr.Cat.1 Pr.Cat.2 Pr.Cat.3  
## 1 -0.4871882 0.23045474 -0.7960149 -1.2358341 2.5300303  
## 2 0.4270467 0.06483572 1.0136848 -0.3572921 -0.4477498  
## 3 -0.1929535 -0.09543113 -0.5363193 0.5184782 -0.2363734  
## Pr.Cat.4 PropCat.5 PropCat.6 PropCat.7 PropCat.8 PropCat.9  
## 1 -0.3579166 -1.1538285 -0.24964584 -0.45522491 -0.4787781 -0.12103046  
## 2 -0.2788596 -0.3512634 0.15706153 0.23968178 0.4577418 0.16357708  
## 3 0.2728708 0.4964442 -0.05565673 -0.06880197 -0.2162573 -0.08815753  
## PropCat.10 PropCat.11 PropCat.12 PropCat.13 PropCat.14 PropCat.15  
## 1 -0.2558533 -0.2835592699 -0.1746531 -0.2404776 2.5320827 -0.252771665  
## 2 0.3397891 0.0003590712 0.2614396 0.3908748 -0.4442971 0.067763353  
## 3 -0.1821591 0.0613526957 -0.1449804 -0.2212431 -0.2392350 0.007502552  
## Brand\_Loyalty  
## 1 1.45329573  
## 2 -0.47543909  
## 3 0.01692349  
##   
## Clustering vector:  
## [1] 3 3 3 1 3 3 3 1 3 2 3 3 3 3 3 3 3 3 3 2 1 1 1 1 1 2 3 3 3 1 1 3 1 1 1 3 3  
## [38] 3 1 1 3 1 1 1 3 3 3 3 1 2 1 3 1 3 1 3 1 3 1 3 3 1 1 3 2 1 3 2 3 3 3 3 1 3  
## [75] 3 2 3 1 3 3 3 3 1 2 3 3 3 3 2 1 2 3 1 2 2 3 1 3 1 3 3 3 3 3 3 3 2 3 3 3 1  
## [112] 3 3 2 3 2 2 1 2 3 3 2 3 2 3 1 3 3 3 2 3 3 2 3 1 2 3 3 3 3 3 1 3 1 2 1 1 3  
## [149] 3 2 3 3 1 1 3 2 2 1 3 3 1 1 3 3 3 3 3 3 3 3 3 2 3 1 3 3 3 1 1 3 3 3 2 2 2  
## [186] 3 3 3 3 2 3 2 3 3 3 3 2 3 2 3 3 3 3 3 1 3 3 3 3 2 3 3 2 3 3 2 3 2 1 2 3 3  
## [223] 1 2 3 3 3 3 2 3 1 3 1 3 1 1 1 1 1 2 3 2 3 3 1 3 1 2 2 3 3 2 3 3 3 3 3 3 3  
## [260] 3 2 3 2 3 2 3 3 3 3 2 3 3 3 3 3 2 2 2 3 3 3 3 3 3 3 3 3 3 3 2 2 2 3 3 3 3  
## [297] 1 3 3 2 2 3 3 3 2 3 2 3 2 2 2 2 3 2 3 2 3 3 2 2 3 2 3 3 2 1 3 1 3 3 3 3 3  
## [334] 2 3 2 2 2 2 3 3 3 3 2 3 2 2 2 3 3 3 2 2 3 2 3 2 3 2 2 2 2 3 2 2 2 2 2 2 3  
## [371] 3 2 3 3 1 2 3 2 3 3 2 2 2 2 3 2 3 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 3 2 2 2 2  
## [408] 3 2 3 3 2 2 3 2 2 2 2 3 3 3 3 2 3 2 3 2 2 2 3 3 3 3 2 2 3 2 2 3 2 3 2 2 2  
## [445] 2 3 3 3 3 3 2 3 3 2 3 3 2 2 2 2 2 2 2 2 2 3 3 3 2 2 2 2 2 2 3 3 3 2 2 2 2  
## [482] 2 2 2 2 3 2 2 2 2 2 2 2 3 2 2 2 2 3 3 1 2 3 3 2 2 3 3 3 2 2 3 3 2 2 3 2 2  
## [519] 2 3 3 2 3 3 3 3 3 3 3 3 2 1 3 3 2 3 1 3 3 3 2 2 3 2 2 3 3 3 3 2 3 2 2 3 3  
## [556] 3 3 2 2 2 2 3 2 3 2 2 3 2 3 2 3 3 2 2 3 3 2 2 2 2 2 2 3 2 3 2 3 3 2 3 3 2  
## [593] 2 3 3 3 2 3 3 3  
##   
## Within cluster sum of squares by cluster:  
## [1] 879.652 6257.391 5929.508  
## (between\_SS / total\_SS = 19.2 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

table(kmeansPB.BfP3$cluster)

##   
## 1 2 3   
## 68 219 313

fviz\_cluster(kmeansPB.BfP3, data = behavior.basis) + labs(title = "Purchase Behavior + Basis of Purchase")



#fviz\_cluster(kmeansPB.BfP, behavior.basis, main = "Combined Cluster Plot")  
kmeansPB.BfP3$centers

## No..of.Brands Brand.Runs Total.Volume No..of..Trans Value  
## 1 -0.598521651 -0.8120867 0.09190381 -0.42903264 -0.55575082  
## 2 0.189532593 0.3915028 -0.34789649 0.22334716 0.02008816  
## 3 -0.002581999 -0.0974991 0.22345007 -0.06306329 0.10668291  
## Trans...Brand.Runs Vol.Tran Avg..Price Pur.Vol.No.Promo....  
## 1 1.07696718 0.5271963 -1.3330597 0.2413117  
## 2 -0.22834629 -0.5299567 0.8870873 -0.3717739  
## 3 -0.07420425 0.2562657 -0.3310673 0.2076974  
## Pur.Vol.Promo.6.. Pur.Vol.Other.Promo.. Pr.Cat.1 Pr.Cat.2 Pr.Cat.3  
## 1 -0.4871882 0.23045474 -0.7960149 -1.2358341 2.5300303  
## 2 0.4270467 0.06483572 1.0136848 -0.3572921 -0.4477498  
## 3 -0.1929535 -0.09543113 -0.5363193 0.5184782 -0.2363734  
## Pr.Cat.4 PropCat.5 PropCat.6 PropCat.7 PropCat.8 PropCat.9  
## 1 -0.3579166 -1.1538285 -0.24964584 -0.45522491 -0.4787781 -0.12103046  
## 2 -0.2788596 -0.3512634 0.15706153 0.23968178 0.4577418 0.16357708  
## 3 0.2728708 0.4964442 -0.05565673 -0.06880197 -0.2162573 -0.08815753  
## PropCat.10 PropCat.11 PropCat.12 PropCat.13 PropCat.14 PropCat.15  
## 1 -0.2558533 -0.2835592699 -0.1746531 -0.2404776 2.5320827 -0.252771665  
## 2 0.3397891 0.0003590712 0.2614396 0.3908748 -0.4442971 0.067763353  
## 3 -0.1821591 0.0613526957 -0.1449804 -0.2212431 -0.2392350 0.007502552  
## Brand\_Loyalty  
## 1 1.45329573  
## 2 -0.47543909  
## 3 0.01692349

The clusters’ size is 68, 219, 313.

# Selection and clusters’ profiles

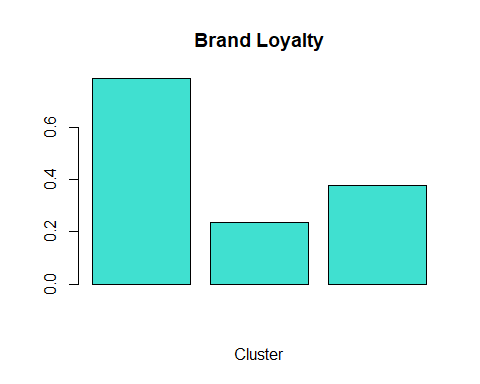
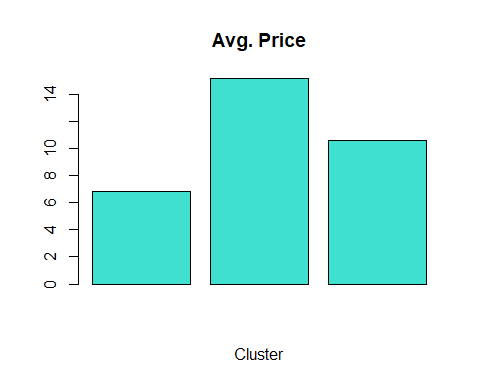
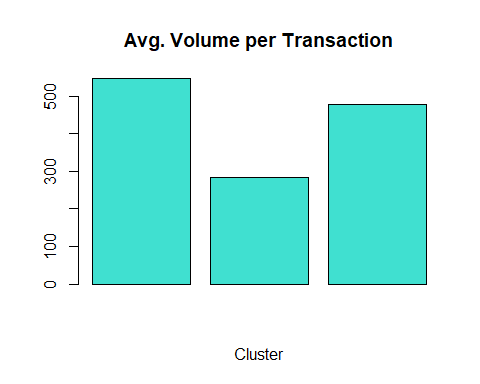
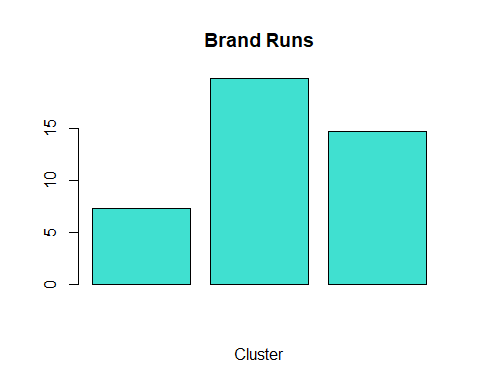
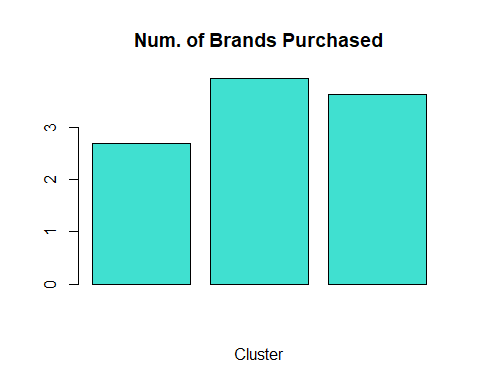
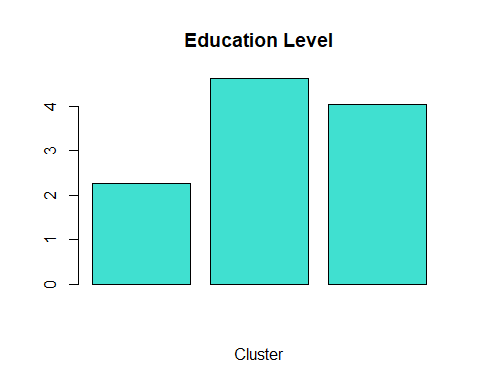
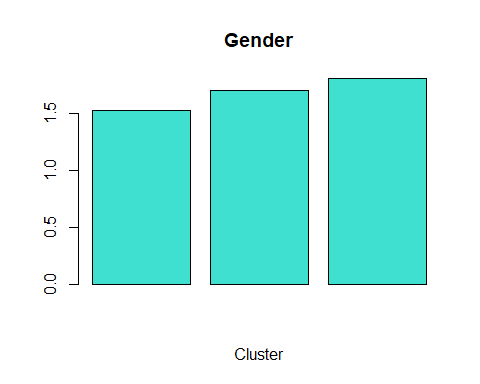
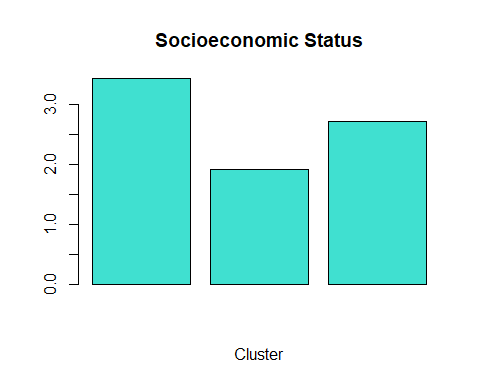
Since all of the segmentation we performed during the previous section of the study offered good results, we decided to select the combined sets of variables for completeness. In fact, an effective promotional approach is based on factors like volume, frequency, brand loyalty, as well as price and selling proposition.

#15.selecting the segmentation  
dataset.segm <- data.frame(dataset.mod, cluster = kmeansPB.BfP3$cluster)  
head(dataset.segm[, 43:48])

## PropCat.12 PropCat.13 PropCat.14 PropCat.15 Brand\_Loyalty cluster  
## 1 0.03 0 0.13 0.34 0.38 3  
## 2 0.00 0 0.08 0.00 0.14 3  
## 3 0.02 0 0.56 0.00 0.55 3  
## 4 0.00 0 0.60 0.00 0.60 1  
## 5 0.00 0 0.14 0.00 0.14 3  
## 6 0.00 0 0.07 0.27 0.08 3

dataset.segm %>% group\_by(dataset.segm$cluster) %>% summarise\_all(mean) -> dataset.clust

We added a Cluster variable at the right-end of the unnormalized dataset that indicates to which cluster each household belong to. Not normalized data is preferred at this point because it is more intuitive to describe the characteristics of the clusters.



* Cluster 1: household from this cluster come from a low socioeconomic environment, have the lowest education level, record the highest average volume per transaction and a low average price, show the lowest number of brand runs, tend to buy very few numbers of brands, and resulted as the most brand loyal.
* Cluster 2: household from this cluster are quite the opposite with respect to cluster 1 since they come from a high socioeconomic environment, have the highest education level, record the lowest average volume per transaction and a high average price (expensive items), show the highest number of brand runs, tend to buy the highest number of brands, and resulted as the least brand loyal.
* Cluster 3: household from this cluster are balanced, compared to the first two clusters, but is characterized by being a female majority, have a high level of education, and show a high number of brands purchased.

# Predictive model

In this section, we report the results of a predicted model we developed to help classifying the data into the three clusters. The model output will be used in the future to target direct-mail promotions. We decided to use a KNN classification model to determine how well the model classify customers and determine the success rate of the promotion. First, we selected a subset of the non-normalized dataset that contains the new column showing the number of cluster. Then, we created a new column that considers all the household in cluster 1, characterized by the highest level of Brand Loyalty, as successful (1) while all the others are considered non successful (0). Finally, we normalized all the data except for the last 3 variables: Brand Loyalty, Cluster, Success.

#17.splitting data for the predictive model  
set.seed(1234)  
dataset.knn <- select(dataset.segm, 12:22, 32:48)  
dataset.knn$Success = ifelse(dataset.knn$cluster == 1,1,0)  
norm\_set <- preProcess(dataset.knn[,1:27], method=c("center", "scale"))  
dataset.knn[,1:27] <- predict(norm\_set, dataset.knn[,1:27])  
Index = createDataPartition(dataset.knn$Brand\_Loyalty, p=0.60, list=FALSE)  
train.df = dataset.knn[Index, ]  
valid.df = dataset.knn[-Index, ]

We needed to prepare the data for the KNN model, identifying the both the lables for the training set and the validation set. Moreover, we tried different values of k in order to find the optimal level, that is at k=15 (highest level of accuracy).

#18.preparing data for knn algorithm and testing the optimal k  
train\_predictors <- train.df[ ,1:28]   
valid\_predictors <- valid.df[ ,1:28]  
train\_labels <- train.df[ ,29]   
valid\_labels <- valid.df[ ,29]   
set.seed(1234)  
Search\_grid <- expand.grid(k=c(1:15))  
model <- train(factor(Success)~ . ,   
 data = dataset.knn, method="knn",  
 tuneGrid=Search\_grid)  
model

## k-Nearest Neighbors   
##   
## 600 samples  
## 28 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 600, 600, 600, 600, 600, 600, ...   
## Resampling results across tuning parameters:  
##   
## k Accuracy Kappa   
## 1 0.9956535 0.9788001  
## 2 0.9952742 0.9765245  
## 3 0.9952745 0.9767959  
## 4 0.9952751 0.9765852  
## 5 0.9952796 0.9765559  
## 6 0.9960007 0.9799673  
## 7 0.9967265 0.9837547  
## 8 0.9976146 0.9881593  
## 9 0.9976325 0.9884226  
## 10 0.9961806 0.9812622  
## 11 0.9974838 0.9875884  
## 12 0.9969215 0.9850531  
## 13 0.9974758 0.9878391  
## 14 0.9978284 0.9893594  
## 15 0.9981885 0.9912951  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was k = 15.

Finally, we were ready to run the algorithm for k=15. The model successfully identified 23 households. The following Confusion Matrix is helpful to visualize the model’s result and measure some performance metrics such as Accuracy, Recall, Precision, and Specificity.

#19.running the model with k=15 and creating confusion matrix  
set.seed(1234)  
knn15 <- knn(train\_predictors,   
 valid\_predictors,   
 cl=train\_labels,   
 k=15 )  
head(knn15)

## [1] 0 0 1 0 0 1  
## Levels: 0 1

summary(knn15)

## 0 1   
## 216 23

c.matrix <- CrossTable(x=valid\_labels,y=knn15, prop.chisq = FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 239   
##   
##   
## | knn15   
## valid\_labels | 0 | 1 | Row Total |   
## -------------|-----------|-----------|-----------|  
## 0 | 214 | 0 | 214 |   
## | 1.000 | 0.000 | 0.895 |   
## | 0.991 | 0.000 | |   
## | 0.895 | 0.000 | |   
## -------------|-----------|-----------|-----------|  
## 1 | 2 | 23 | 25 |   
## | 0.080 | 0.920 | 0.105 |   
## | 0.009 | 1.000 | |   
## | 0.008 | 0.096 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 216 | 23 | 239 |   
## | 0.904 | 0.096 | |   
## -------------|-----------|-----------|-----------|  
##   
##