CRISA Asian Market Research Agency

IMRB Consumers Segmentation

Melissa Paniagua

MIS64060:Fundamentals of Machine Learning
Professor Murali Shanker
Kent State University

15 December 2020

Abstract

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

Contents

I. Introduction

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, as well as 46 selected variables. The strata were defined based on socioeconomic status and the market (a collection of cities).

Nevertheless, it is essential to considere the following assumption of the BathSoap dataset.

- AGE: Is defined as categories from 1 to 4. 1 refers to a group of age less than 15 years old. 2 has ages from 15 to 35. The 3th group has ages from 35 to 50, and the 4th group has over 50 years old.

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.

Traditionally, CRISA has selected its segment market based on demographic information. Nevertheless, in this scenario, this agency wants to select its segmentation based on purchase behavior and basis of purchase.

In the following sections, we will implement an unsupervised learning algorithm, K-means clustering, to select form groups based on their similarities. As CRISA requires, wi will use purchase behavior and basis of purchase.

II. Data Exploration

The Bath Soap dataset has 600 observations and 46 variables, as shown below. By performing the "str" function, we can determine that most variables are numeric. However, some percentage variables are classified as a character due to the percent (%) sign. In the next section, I will give a solution to solve this matter.

```
# To get the total number of rows and columns
dim(BathSoap)
```

```
[1] 600  46
```

Additionally, this dataset has many categorical variables that its data type is numeric. For instance: age is a categorical variable where customers' age was grouped into categories. The same analogy happens with socioeconomic groups, eating habits, until the CS variable, which refers to television availability. Therefore, it is essential to take into consideration that even though these variables are numeric, their main goal is to represent a category.

After analyzing the descriptive statistics shown above and reviewing the description of categorical variables, we can see that the dataset has missing values infiltrated as zeros. Let's see how many zeros each categorical variable has:

Note: socioeconomic level, age, and the number of children were removed because these variables do not have missing data.

```
# Count the 0's on each variable
table(BathSoap$FEH)
```

```
  0   1   2   3
 69 165  34 332
```

```
table(BathSoap$MT)
```

```
  0   3   4   5   6   8   9  10  12  13  14  15  16  17  19
 69   5  83  27  11   8   9 326   7   8   3   8  10  25   1
```

```
table(BathSoap$SEX)
```

```
  0   1   2
 68  21 511
```

```
table(BathSoap$EDU)
```

```
  0   1   2   3   4   5   6   7   8   9
 73  49   9  33 136 189  23  73  13   2
```

```
table(BathSoap$HS)
```

```
  0   1   2   3   4   5   6   7   8   9  10  12  15
 68   2  41  73 147 142  65  22  18  13   4   2   3
```

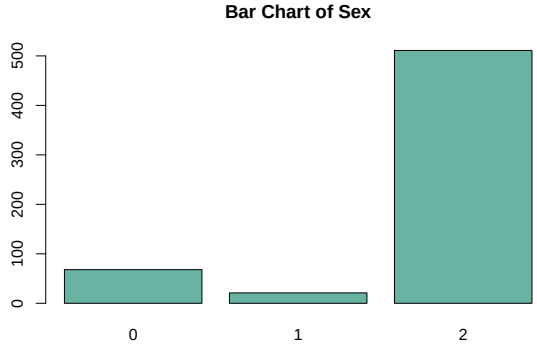
```
table(BathSoap$CS)
```

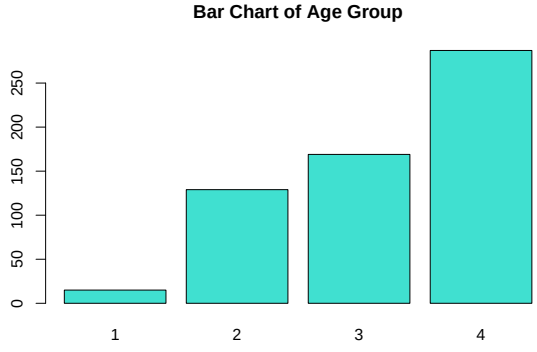
```
  0   1   2
 99 443  58
```

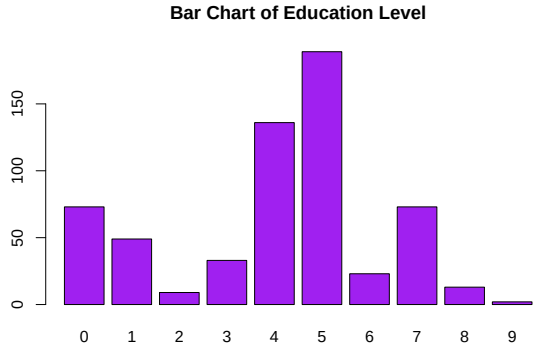
Even though there are some missing values (0's) on almost every categorical variable, I decided not to remove these observations. For our purposes, I will consider them as another category in the data frame.

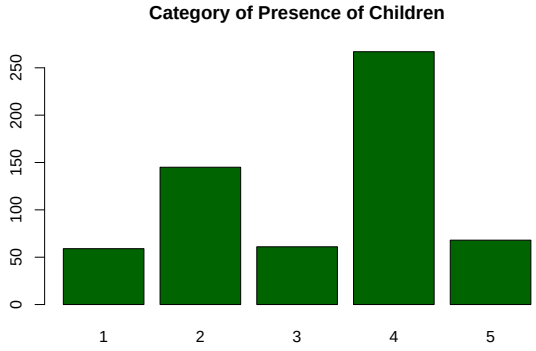
If we decide to remove the missing values (0's) we could lose 105 observations, but the main reason is that it is not a wise decision in terms of trade-off. For example, the observations that have missing values have the same 6 variables missed, but the other variables have data, which could be valuable to the analysis. Additionally, the k-means model will not include the categorical data to run the models. It will not be harmful to conserve the zeros, and it will add relevant data (**more data points**) in the purchase behavior and based on purchase for the data analysis.

Now, let's see some visuals to have a better understanding of the data.









From these plots, we can see the following:

- As mentioned in the introduction, the dataset was defined based on socioeconomic status and the market (a collection of cities), and we can confirm it in the first graph.
- 85.16% of the data points are female.
- More than 250 of the customers are in the age group 4, which based on our assumptions, their age is over 50 years old.
- Most of the customer's education level in the classification of 4 and 5. Remember that 1 means minimum education and 9 is the maximum.

III. Data Preparation

As we saw in the previous section, it is essential to change the data type of the last 16 variables from character to numeric by removing the % sign.

The following output shows the new dataset in which all variables are numeric now. Additionally, it is important to clarify that the categorical variables (the first nine variables) saved as a numeric number, will not be changed to "as.factor" because these variables will not be included in the K-means models we will perform.

```
'data.frame':   600 obs. of  46 variables:
 $ Member.id            : num  1010010 1010020 1014020 1014030 1014190 ...
 $ SEC                  : num  4 3 2 4 4 4 4 4 4 1 ...
 $ FEH                  : num  3 2 3 0 1 3 2 3 3 3 ...
 $ MT                   : num  10 10 10 0 10 10 10 10 10 5 ...
 $ SEX                  : num  1 2 2 0 2 2 2 2 2 1 ...
 $ AGE                  : num  4 2 4 4 3 3 4 2 4 4 ...
 $ EDU                  : num  4 4 5 0 4 4 1 4 4 7 ...
 $ HS                   : num  2 4 6 0 4 5 3 5 6 3 ...
 $ CHILD                : num  4 2 4 5 3 2 2 3 4 4 ...
 $ CS                   : num  1 1 1 0 1 1 1 0 1 1 ...
 $ Affluence.Index      : num  2 19 23 0 10 13 11 0 17 6 ...
 $ No..of.Brands        : num  3 5 5 2 3 3 4 3 2 4 ...
 $ Brand.Runs           : num  17 25 37 4 6 26 17 8 12 13 ...
 $ Total.Volume         : num  8025 13975 23100 1500 8300 ...
 $ No..of..Trans        : num  24 40 63 4 13 41 26 25 27 18 ...
 $ Value                : num  818 1682 1950 114 591 ...
```

```
 $ Trans...Brand.Runs   : num  1.41 1.6 1.7 1 2.17 1.58 1.53 3.13 2.25 1.38 ...
 $ Vol.Tran             : num  334 349 367 375 638 ...
 $ Avg..Price           : num  10.19 12.03 8.44 7.6 7.12 ...
 $ Pur.Vol.No.Promo.... : num  1 0.89 0.94 1 0.61 1 0.98 0.94 0.9 1 ...
 $ Pur.Vol.Promo.6..    : num  0 0.1 0.02 0 0.14 0 0.02 0 0.1 0 ...
 $ Pur.Vol.Other.Promo..: num  0 0.02 0.04 0 0.24 0 0 0.06 0 0 ...
 $ Br..Cd..57..144      : num  0.38 0.02 0.03 0.4 0.05 0.08 0.45 0.04 0.39 0.07 ...
 $ Br..Cd..55           : num  0.13 0.08 0.55 0.6 0.14 0.07 0.05 0.79 0 0.12 ...
 $ Br..Cd..272          : num  0 0 0 0 0 0 0.01 0 0 0 ...
 $ Br..Cd..286          : num  0 0 0.03 0 0 0 0 0 0 0 ...
 $ Br..Cd..24           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Br..Cd..481          : num  0 0.06 0 0 0 0 0 0 0 0 ...
 $ Br..Cd..352          : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Br..Cd..5            : num  0 0.14 0.02 0 0 0 0 0 0 0.4 ...
 $ Others.999           : num  0.492 0.699 0.379 0 0.807 0.857 0.495 0.167 0.615 0.41 ...
 $ Pr.Cat.1             : num  0.23 0.29 0.12 0 0 0.22 0.07 0.04 0.11 0.61 ...
 $ Pr.Cat.2             : num  0.56 0.55 0.32 0.4 0.05 0.45 0.66 0.04 0.89 0.1 ...
 $ Pr.Cat.3             : num  0.13 0.09 0.56 0.6 0.14 0.07 0.05 0.9 0 0.12 ...
 $ Pr.Cat.4             : num  0.07 0.06 0 0 0.81 0.27 0.23 0.02 0 0.17 ...
 $ PropCat.5            : num  0.5 0.46 0.24 0.4 0.81 0.49 0.82 0.06 0.7 0.24 ...
 $ PropCat.6            : num  0 0.35 0.12 0 0 0.1 0 0 0.28 0.46 ...
 $ PropCat.7            : num  0 0.03 0.03 0 0 0 0.02 0 0 0.15 ...
 $ PropCat.8            : num  0 0.02 0.01 0 0.05 0.01 0.01 0 0 0 ...
 $ PropCat.9            : num  0 0.01 0.01 0 0 0.07 0 0 0.02 0 ...
 $ PropCat.10           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PropCat.11           : num  0 0.06 0 0 0 0 0 0 0 0 ...
 $ PropCat.12           : num  0.03 0 0.02 0 0 0 0 0.01 0 0 ...
 $ PropCat.13           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PropCat.14           : num  0.13 0.08 0.56 0.6 0.14 0.07 0.05 0.9 0 0.12 ...
 $ PropCat.15           : num  0.34 0 0 0 0 0.27 0.1 0.03 0 0.03 ...
```

Here we can see that the variables were changed.

Additionally, it is essential to determine the measure of brand loyalty.

To better handle the percentages of volume purchased of the brand, which will be used as a measure of brand loyalty, we will add another column at the end of the data frame to classify if a customer is loyal or not. It will be based on the column named "Others.999." If the customer has values in "Others.999" greater than 50%, we will classify the customer as "0" meaning the customer is NOT loyal to any brand. If the customer, on the other hand, has an "Others.999" value lower than 50%, the function will assign "1" affirming its loyalty to a brand.

```
# Create loyalty vector based on some loyalty variables
BathSoap4$Loyalty = 1*(BathSoap4$Others.999<0.5)

# Show the last columns to see Loyalty column
head(BathSoap4[41:47])
```

```
  PropCat.10 PropCat.11 PropCat.12 PropCat.13 PropCat.14 PropCat.15 Loyalty
1          0       0.00       0.03          0       0.13       0.34       1
2          0       0.06       0.00          0       0.08       0.00       0
3          0       0.00       0.02          0       0.56       0.00       1
4          0       0.00       0.00          0       0.60       0.00       1
5          0       0.00       0.00          0       0.14       0.00       0
6          0       0.00       0.00          0       0.07       0.27       0
```

```r
# Summary of loyalty
table(BathSoap4$Loyalty)
```

```
  0   1
318 282
```

The previous summary shows that 318 customers are not loyal to any brand, and 282 customers are loyal.

For our purposes, we will select three different datasets to run the K-means models based on:

- BathSoap Purchase Behavior
- BathSoap Basis Purchase
- BathSoap both

Normalization is an essential step in the data preparation. It will allow the dataset to have the same scale, and it will help to reduce the bias and its spread.

The following output shows the first 6 rows and the first 6 variables of the select three different datasets: BathSoap Purchase Behavior, BathSoap Basis Purchase, and BathSoap both together.

```r
#To see the first 6 rows and the first 6 variables
head(BathSoap_Purchase_Behavior)[1:6, 1:5]
```

```
     No..of.Brands Brand.Runs Total.Volume No..of..Trans      Value
[1,]    -0.4030277  0.1200727   -0.5005898    -0.4104681 -0.5881031
[2,]     0.8630280  0.8895639    0.2651391     0.5076339  0.3896410
[3,]     0.8630280  2.0438006    1.4394712     1.8274054  0.6936645
[4,]    -1.0360556 -1.1303505   -1.3403176    -1.5580955 -1.3852447
[5,]    -0.4030277 -0.9379777   -0.4651989    -1.0416632 -0.8451360
[6,]    -0.4030277  0.9857502    0.8056536     0.5650152  0.4168163
```

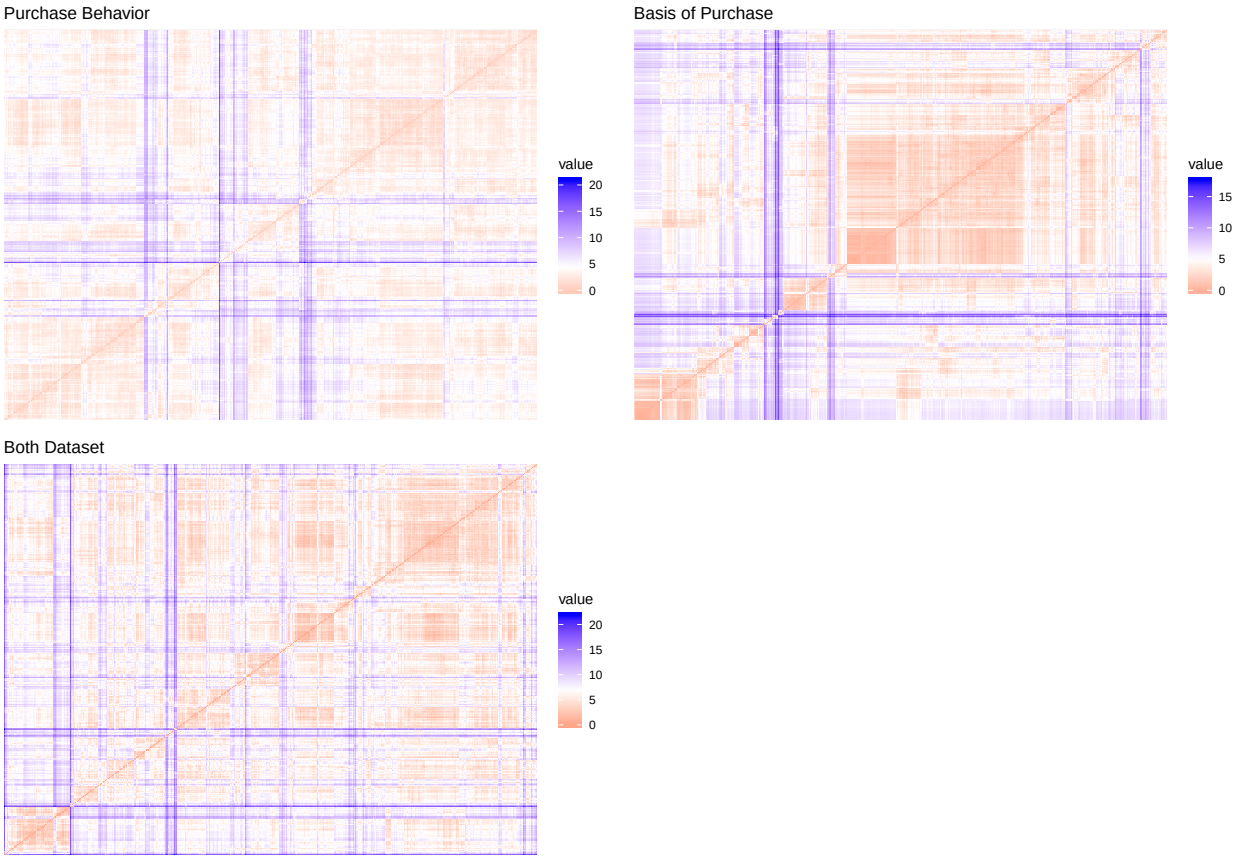
```r
head(BathSoap_Basis_Purchase)[1:6, 1:6]
```

```
        Pr.Cat.1   Pr.Cat.2     Pr.Cat.3    Pr.Cat.4    PropCat.5   PropCat.6
[1,] -0.17442555  0.2143002 -0.034449105 -0.09717115  0.135304773 -0.55572458
[2,]  0.03922943  0.1822114 -0.183687103 -0.14932024  0.008900937  1.54967253
[3,] -0.56612636 -0.5558329  1.569859381 -0.46221482 -0.686320160  0.16612586
[4,] -0.99343633 -0.2991219  1.719097380 -0.46221482 -0.180704817 -0.55572458
[5,] -0.99343633 -1.4222327  0.002860395  3.76186195  1.114934500 -0.55572458
[6,] -0.21003472 -0.1386774 -0.258306103  0.94581077  0.103703814  0.04581745
```

```r
head(BathSoap_both)[1:6, 1:5]
```

```
     No..of.Brands Brand.Runs Total.Volume No..of..Trans      Value
[1,]    -0.4030277  0.1200727   -0.5005898    -0.4104681 -0.5881031
[2,]     0.8630280  0.8895639    0.2651391     0.5076339  0.3896410
[3,]     0.8630280  2.0438006    1.4394712     1.8274054  0.6936645
[4,]    -1.0360556 -1.1303505   -1.3403176    -1.5580955 -1.3852447
[5,]    -0.4030277 -0.9379777   -0.4651989    -1.0416632 -0.8451360
[6,]    -0.4030277  0.9857502    0.8056536     0.5650152  0.4168163
```

Now, we computed the distance using the euclidean distance, and the following are the 6 first observations of those data frames.



This graph is a distance matrix. As we can see, the diagonal values are zeros (pink line) because it is showing the distance between any point against itself. The purple section represents the furthest distance between any pair of observations. For instance, on the Basis of Purchase visual, we can see that there are concentrated pink areas, which means those data points are very close. On the other hand, we can also see that there are strong purple lines that indicate there is a big distance between those observations.

In the following section, we will run K-means model utilizing three different datasets.

Question 1

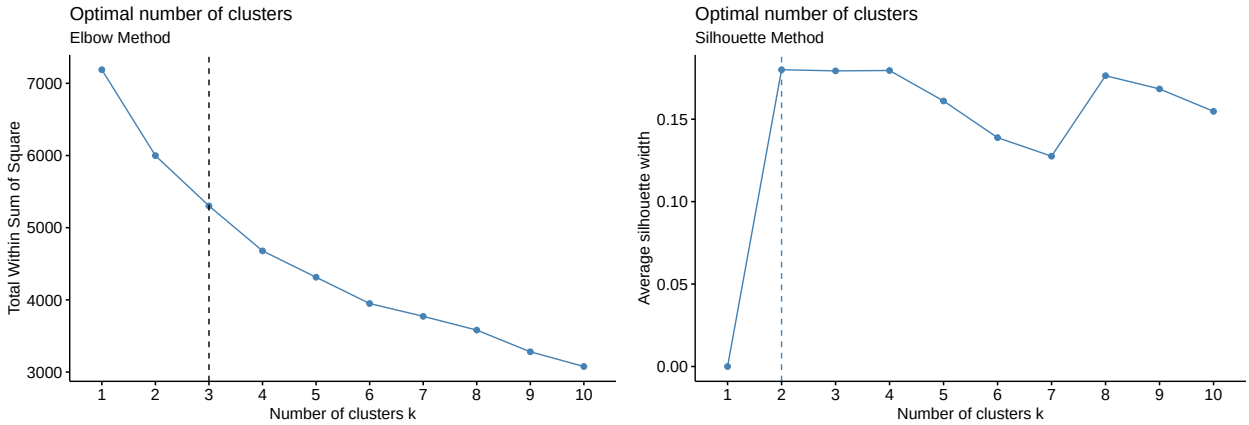
1. Use k-means clustering to identify clusters of households based on:

- BathSoap Purchase Behavior
- BathSoap Basis Purchase
- BathSoap both

IV.I K-means Model based on Purchase Behavior

Before running the K-means model, it is necessary to run heuristic methods that help to identify the "best" k, which is the number of clusters that the model should group the data points.

Choosing optimal k: Elbow Method & Silhouette Method



Due to the nature of these methods compute the distance, it is common to receive different ks. Domain knowledge is the best way of determining the number of clusters, but for our purposes, utilizing K=2 proposed by the Silhouette Method is a good option to run the K-means model.

Run K-means Model

```
set.seed(123)
# To run the kmeans model
kmeans_pur_beh <- kmeans(BathSoap_Purchase_Behavior, centers = 2, nstart = 30)

# To see the results
print(kmeans_pur_beh)
```

```
K-means clustering with 2 clusters of sizes 268, 332

Cluster means:
  No..of.Brands Brand.Runs Total.Volume No..of..Trans      Value
1     0.6858747  0.7890706    0.3184304     0.6838461  0.4915745
2    -0.5536579 -0.6369606   -0.2570462    -0.5520203 -0.3968132
  Trans...Brand.Runs   Vol.Tran Avg..Price Pur.Vol.No.Promo....
1         -0.2759896 -0.2407916  0.2948865           -0.3118945
2          0.2227868  0.1943739 -0.2380409            0.2517703
  Pur.Vol.Promo.6.. Pur.Vol.Other.Promo..    Loyalty
1         0.3727957            0.04067052 -0.3059680
2        -0.3009314           -0.03283042  0.2469863

Clustering vector:
  [1] 2 1 1 2 2 1 2 2 1 2 1 1 2 2 2 1 1 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2
 [38] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 1 1 2 2 1 1 2 2 1 2 1 1 2 2 2 1 2 2
 [75] 2 2 2 2 2 1 2 2 1 2 1 2 1 2 1 2 2 2 1 1 2 1 1 1 2 1 2 2 2 2 2 1 2 1 1 2 1 1 2
[112] 2 2 1 1 1 1 2 1 1 1 1 2 1 1 2 1 2 2 1 1 1 2 2 2 1 2 1 1 2 2 2 2 2 1 2 2 2
[149] 1 2 2 2 2 2 1 1 1 2 2 1 2 2 1 1 2 1 1 2 1 2 1 1 2 2 1 1 2 2 2 2 2 2 1 1 1
[186] 2 1 1 2 1 2 1 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 1 2 1 2 1
[223] 2 2 1 2 2 1 1 2 2 1 2 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2
[260] 1 2 2 1 1 1 2 2 2 2 1 1 2 2 1 2 2 1 1 2 2 1 2 1 1 1 1 1 2 1 1 1 2 2 2 2 2
[297] 2 1 1 1 1 2 2 1 1 2 1 2 2 2 1 1 1 1 1 1 2 2 1 2 1 1 2 1 2 2 1 2 1 2 1 2 1
```

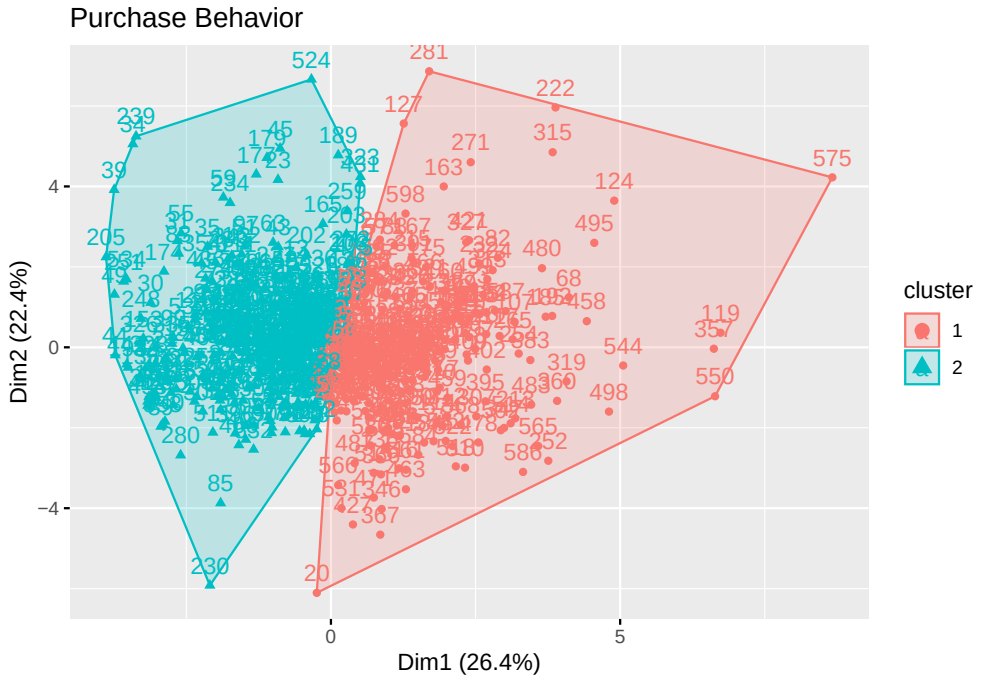
```
[334] 1 2 1 1 1 2 2 2 2 2 2 1 1 1 2 1 2 2 2 1 2 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2
[371] 1 2 2 2 2 2 2 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 2 1 2 1 2 2 2 2 2 1 2 1 2 2 2
[408] 2 1 1 2 2 1 2 1 1 1 2 2 1 1 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 1 1 1
[445] 2 2 2 1 2 2 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 2 1 1 1 2 1 2
[482] 1 1 2 2 2 1 1 1 1 2 1 2 2 1 1 2 1 1 1 2 1 2 1 2 2 2 2 2 1 1 2 2 2 1 1 2 2 1
[519] 2 1 2 2 2 2 2 2 2 1 2 1 2 1 2 1 2 1 2 2 2 2 2 1 1 1 1 1 2 2 2 1 1 2 2 2 1 1
[556] 2 2 1 1 2 1 2 1 1 1 1 1 1 2 1 2 1 2 1 1 2 1 2 2 1 1 1 1 1 2 1 1 1 2 1 2 2
[593] 2 2 2 2 2 1 1 2 2

Within cluster sum of squares by cluster:
[1] 2876.644 3121.640
 (between_SS / total_SS =  16.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

```r
# Run table function to identify to which cluster those sizes belong
table(kmeans_pur_beh$cluster)
```

```
  1   2
268 332
```

This output shows 2 clusters of sizes 268, 332. We also see the clusters means of each variable based on each cluster, and how each data point is assigned. For example the first row was assigned to cluster 2, and so on. The last table determines the 268 data points belong to cluster 1, and 332 to cluster 2.

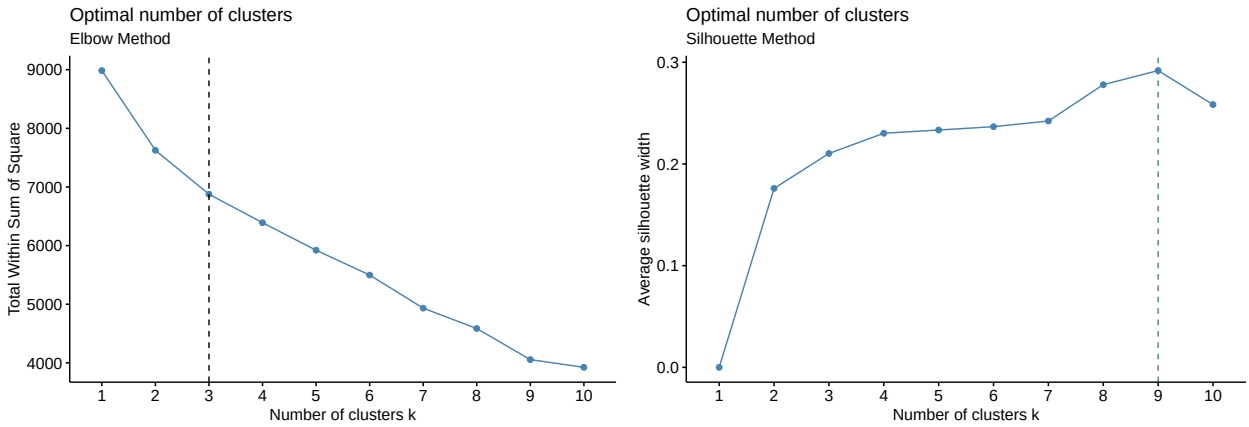
Visualize



IV.II K-means Model based on Basis for Purchase

Basis for Purchase includes data based on the percent of volume purchased under the price category, and the percent of volume purchased under the product proposition category.

Choosing optimal k: Elbow Method & Silhouette Method



Based on Basis for Purchase, we will use k=3 given by the Elbow Method because the marketing efforts would support two to five different promotional approaches, and nine exceeds this projection.

Run K-means Model

```
set.seed(123)
# To run the kmeans model
kmeans_bas_pur <- kmeans(BathSoap_Basis_Purchase, centers = 3, nstart = 30)

# To see the results
print(kmeans_bas_pur)
```

```
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 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 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"
```

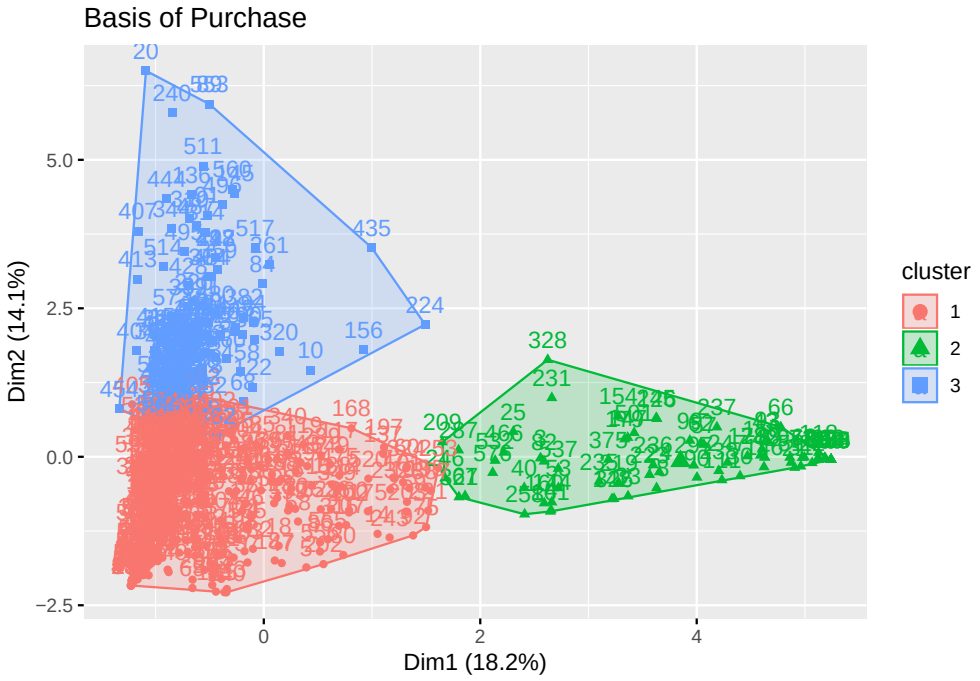
```r
# Let's run table function to identify to which cluster those sizes belong
table(kmeans_bas_pur$cluster)
```

```
  1   2   3
376  79 145
```

The output of Basis of purchase model shows 3 clusters of sizes 376, 79, and 145. It gives the clusters mean of each variable based on each cluster, and clustering vector.

In the following table we can see that cluster 1 has 376, cluster 2: 79, and cluster 3: 145.

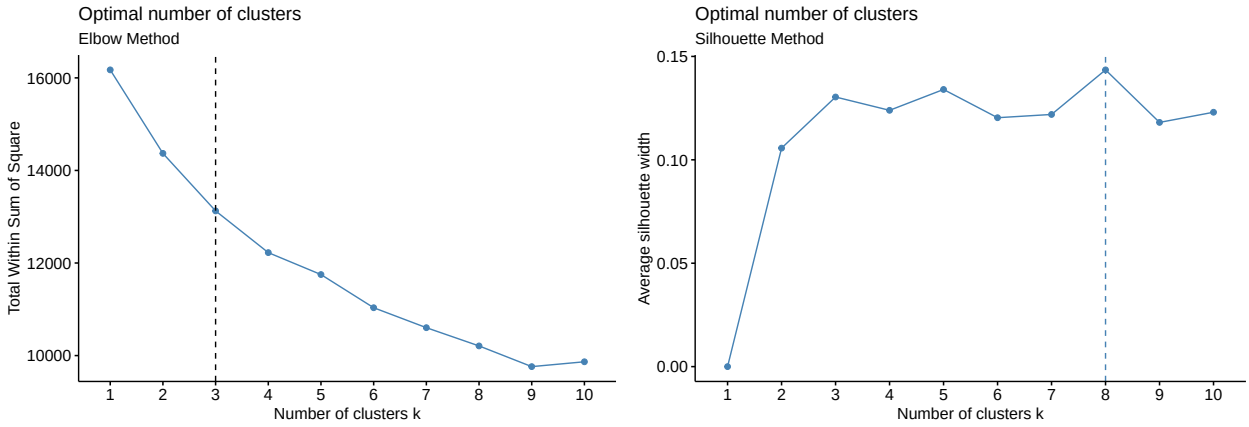
Visualize



IV.III K-means Model based on both: Purchase Behavior & Basis for Purchase

Here, we will combine both subsets (Purchase Behavior and Basis for Purchase), and we will run the K-means to see how the model forms the groups by providing more data points

Choosing optimal k: Elbow Method & Silhouette Method



Like the previous section, we will select k=3 like the Elbow method recommends due to the like number of the Silhouette method provides.

Run K-means Model

```
set.seed(123)
# To run the kmeans model
kmeans_both <- kmeans(BathSoap_both, centers = 3, nstart = 30)

# To see the results
print(kmeans_both)
```

```
K-means clustering with 3 clusters of sizes 71, 323, 206

Cluster means:
  No..of.Brands  Brand.Runs Total.Volume No..of..Trans       Value
1   -0.55459780 -0.78218281    0.1130446   -0.42339909 -0.53680015
2    0.11044997  0.02835315    0.2498161    0.08109308  0.14735016
3    0.01796652  0.22513064   -0.4306640    0.01877801 -0.04602569
  Trans...Brand.Runs   Vol.Tran Avg..Price Pur.Vol.No.Promo....
1         1.01770867  0.5285946 -1.3178426            0.1899996
2        -0.07186868  0.2021054 -0.3118897            0.2574253
3        -0.23807638 -0.4990790  0.9432388           -0.4691181
  Pur.Vol.Promo.6.. Pur.Vol.Other.Promo..   Pr.Cat.1   Pr.Cat.2   Pr.Cat.3
1        -0.4122432             0.2187110 -0.7978367 -1.2125251  2.4826284
2        -0.1968194            -0.1722504 -0.4954593  0.5151211 -0.2434054
3         0.4506890             0.1947009  1.0518434 -0.3897807 -0.4740130
    Pr.Cat.4  PropCat.5   PropCat.6   PropCat.7  PropCat.8   PropCat.9
1 -0.3263334 -1.1327393 -0.25071735 -0.45474864 -0.4761555 -0.13663506
2  0.2288011  0.4662832 -0.05679854 -0.05399766 -0.2246298  0.01404653
3 -0.2462771 -0.3407038  0.17547020  0.24139999  0.5163227  0.02506825
  PropCat.10  PropCat.11 PropCat.12 PropCat.13 PropCat.14  PropCat.15
1 -0.2562582 -0.22559639 -0.1772583 -0.2413875  2.4854563 -0.24306514
2 -0.1815094  0.08413924 -0.1383678 -0.2144097 -0.2474814  0.01705480
3  0.3729216 -0.05417297  0.2780493  0.4193827 -0.4685966  0.05703361
     Loyalty
1  1.0328315
2  0.1065428
3 -0.5230310

Clustering vector:
  [1] 2 2 2 1 2 2 2 1 2 3 2 2 2 2 2 2 2 2 2 3 1 1 1 1 1 3 2 2 2 1 1 2 1 1 1 2 2
 [38] 2 1 1 2 1 1 1 2 2 2 2 1 3 1 2 1 2 1 2 1 2 1 2 2 1 1 2 2 1 2 2 2 2 2 2 1 2
 [75] 2 3 2 1 2 2 2 2 1 3 3 2 2 2 3 1 3 2 1 3 3 2 1 2 1 2 2 2 2 2 2 2 3 2 2 2 1
[112] 2 2 3 2 3 2 1 2 2 2 2 2 3 2 1 2 2 2 3 2 2 3 2 1 3 2 2 2 2 2 1 2 1 3 1 1 2
[149] 2 3 2 2 1 1 2 3 3 1 2 1 1 1 2 2 2 2 2 2 2 2 2 3 2 1 2 2 2 1 1 2 2 2 3 3 2
[186] 2 2 2 2 3 2 3 2 2 2 2 2 2 3 2 2 2 2 2 1 2 2 2 2 2 2 2 3 3 2 2 3 2 3 1 3 2 2
[223] 1 3 2 2 2 2 3 3 1 2 1 2 1 1 1 1 1 3 2 2 2 2 1 2 1 3 3 2 2 3 2 2 2 2 2 1 2
[260] 2 3 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 3 3 3 2 3 2 2 2 2 2 2 2 2 2 3 3 3 2 3 2 2
[297] 1 2 2 3 3 2 2 2 3 2 3 2 3 3 3 3 2 3 2 2 2 2 2 3 2 3 2 2 3 1 2 1 2 2 2 2 2
[334] 3 2 3 2 3 3 2 2 2 2 3 2 3 3 3 2 2 2 3 3 2 3 2 3 2 3 3 3 3 2 3 3 3 3 3 3 2
[371] 2 3 2 2 1 3 2 3 2 2 3 3 3 3 2 3 2 2 2 3 3 3 3 3 3 3 3 3 2 3 3 2 3 3 3 3
[408] 2 3 2 2 3 3 2 3 3 2 3 2 2 2 3 3 2 3 2 3 3 3 2 2 2 2 3 3 2 3 3 2 3 2 3 2 3
[445] 3 2 2 2 2 2 3 2 2 3 2 2 3 3 3 3 3 3 3 3 2 2 2 3 3 3 3 2 2 2 2 3 3 3 3
[482] 3 2 3 3 2 3 3 3 3 3 3 3 2 3 3 3 3 2 2 1 3 2 2 3 3 2 3 2 3 3 3 2 2 3 3 2 3 3
[519] 3 2 2 3 2 2 2 2 2 2 2 2 3 1 2 2 3 2 1 2 2 2 3 3 2 3 3 2 2 2 2 3 2 3 3 2 2
```

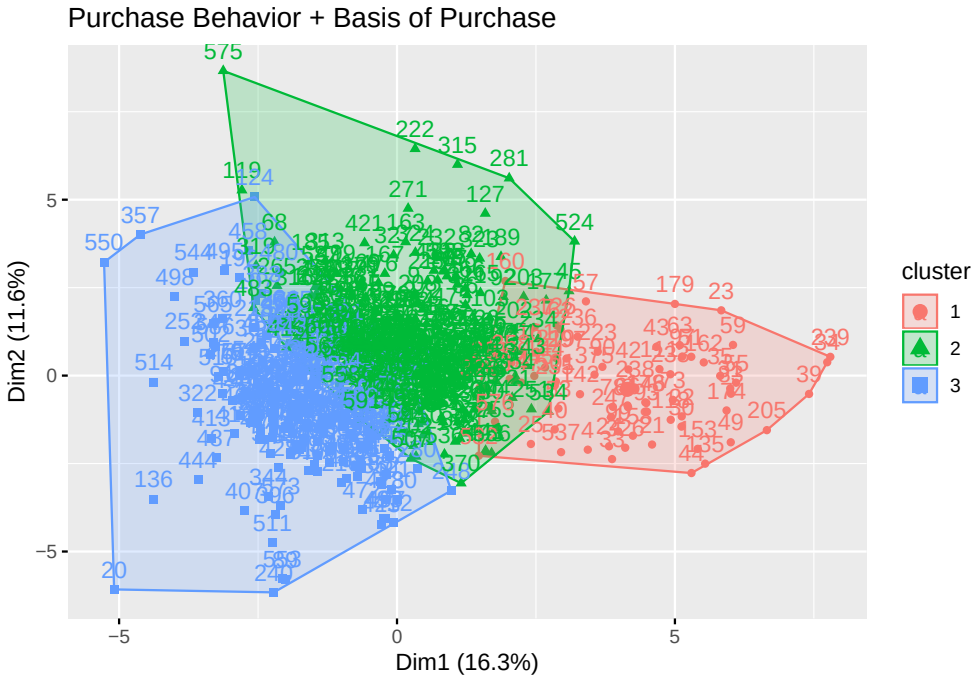
```
[556] 2 2 2 3 3 3 2 3 2 3 3 2 3 2 3 3 2 3 3 2 1 3 3 3 3 3 3 2 3 2 3 2 2 3 2 2 3
[593] 3 2 2 2 3 2 2 2

Within cluster sum of squares by cluster:
[1]  924.1034 5965.5254 6233.8628
 (between_SS / total_SS =  18.9 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

```r
# Let's run table function to identify to which cluster those sizes belong
table(kmeans_both$cluster)
```

```
  1   2   3
 71 323 206
```

In this case, the models groups the data as follows: cluster 1: 71, cluster 2: 323, and cluster 3: 206.

Visualize



Question 2

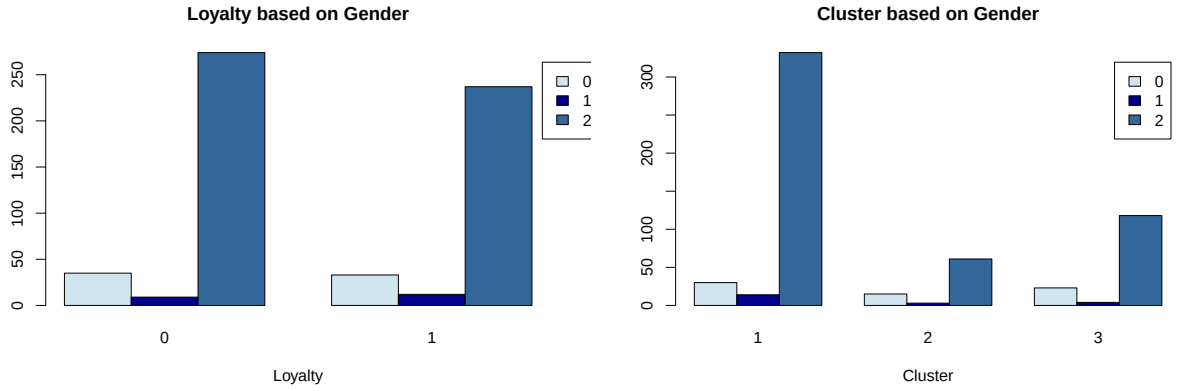
2. Select what you think is the best segmentation and comment on the characteristics (demographic, brand loyalty, and basis for purchase) of these clusters. (This information would be used to guide the development of advertising and promotional campaigns.)

Based on the results of the previous section, in my opinion, the model that best segments the customers is the K-means model based on Basis of Purchase. I believe that model formed the groups utilizing only the basis of the purchase dataset.

In order to analyze the demographic, brand loyalty, and basis for purchase characteristics of these clusters, we will add the number of the cluster the observation belongs to using the unnormalized data frame. We will use unnormalized data because it is more meaningful and easier to compare. Here is an example of the dataset with the new cluster column:

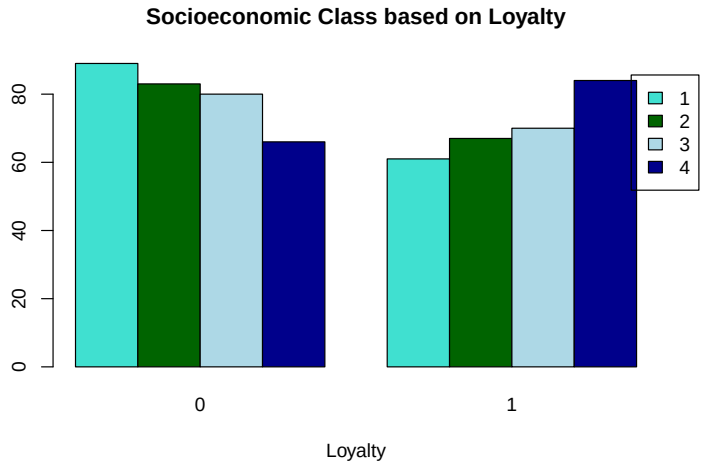
	PropCat.12	PropCat.13	PropCat.14	PropCat.15	Loyalty	cluster
1	0.03	0	0.13	0.34	1	1
2	0.00	0	0.08	0.00	0	1
3	0.02	0	0.56	0.00	1	2
4	0.00	0	0.60	0.00	1	2
5	0.00	0	0.14	0.00	0	1
6	0.00	0	0.07	0.27	0	1

Now, let's see some plot to get more insights.



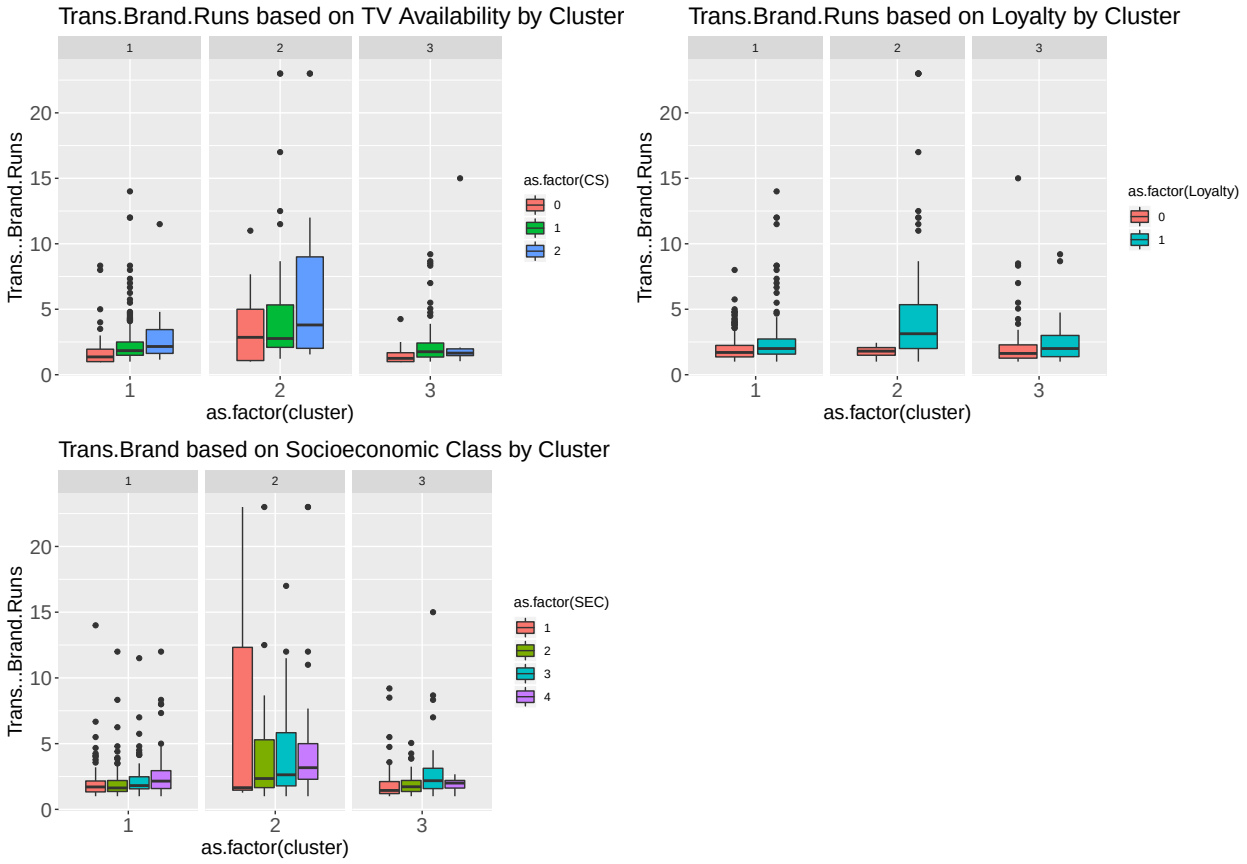
As we saw in the data exploration section, 85.16% of the data points are female. Based on these plots, we can determine that gender does not influence the level of loyalty. Additionally, if we compare the "Loyalty based on Gender" chart VS the "Cluster-based on Gender," there is not much of a difference. Both behave the same!

The following ourput shows the relationship between the socioeconomic class based on loyalty. Remember that in our dataset 1 refers to people in a high class, and 4 is the lower class.



Based on this chart, we can determine that people with a high socioeconomic class (more income) tend to be less loyal to a brand, and people with less income tend to be more loyal to a brand. It might be because there could be a correlation between income and the freedom to select the best products in the market. Nevertheless, correlation does not mean causation. It is something that the market team could take into account when developing advertising and promotional campaigns.

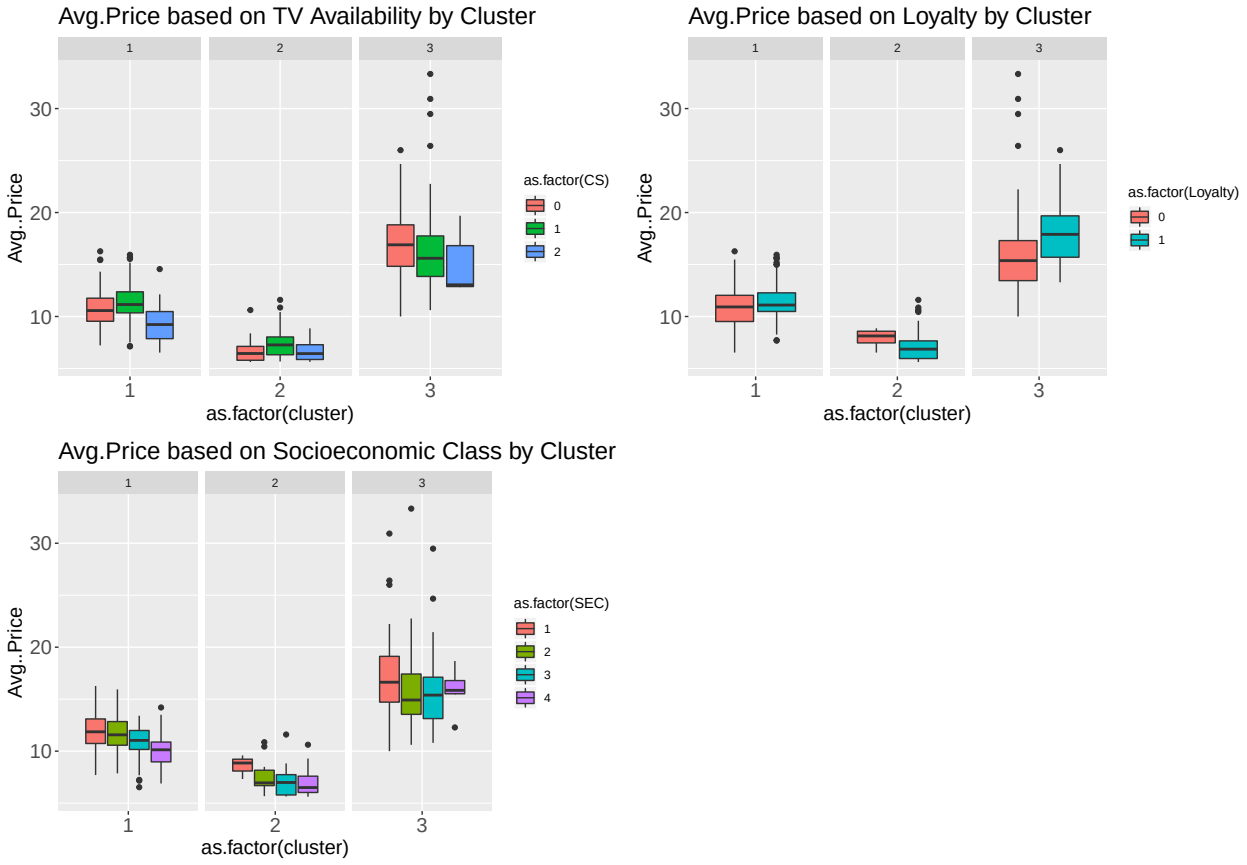
The following charts show the average of transactions per brand run by TV Availability, Loyalty, and Socioeconomic Class.



Here we can see that:

- Cluster 1 and 3 tend to behave similarly.
- Cluster 2: This is the most loyal cluster found on the average of transactions per brand run. It also has the highest range of socioeconomic class but based on TV availability, many data points do not have television availability. Nevertheless, its median has a similar level compared to other clusters. In conclusion, cluster 2 seems to perform more transactions per brand compared to other clusters.

Lastly, here we can see the average price of purchase by TV Availability, Loyalty, and Socioeconomic Class.



From these visuals, we can determine that:

- Cluster 3 is the outstanding cluster out of the 3. Based on the average price of the purchase, this cluster spends more money on average acquirements. It is the most loyal cluster even though it spends more on average purchases.
- Cluster 2 seems to have the lowest scores stand on the average price of purchase.
- Cluster 1: It behaves uniformly across the box plots.

In summary, cluster 3 seems to spend more money on their purchases.

Question 3

3. 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.

In this section, we will run a KNN classification model to determine how well the model will classify its customers to determine if the promotion will succeed or fail based on different characteristics. In order to determine that and based on the previous section, we will use cluster 3, which spends on average more money on their purchases to develop the model.

```
k-Nearest Neighbors

361 samples
 28 predictor
  2 classes: '0', '1'

Pre-processing: re-scaling to [0, 1] (28)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 361, 361, 361, 361, 361, 361, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.9997183  0.9991841
   2  0.9988630  0.9965607
   3  0.9982825  0.9949492
   4  0.9982714  0.9951142
   5  0.9985675  0.9959276
   6  0.9988472  0.9967073
   7  0.9985675  0.9959276
   8  0.9988472  0.9967073
   9  0.9991549  0.9974963
  10  0.9988472  0.9967073

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 1.
```

After performing some data preparation and by performing the grid search, we can see that the optimal k is 1.

Now, we will develop the KNN classification model to determine the level of success and failure of the marketing campaign based on the purchase.

```
#Run the model using k = 2
set.seed(1234)
my_knn <-knn(train_predictors,
                            valid_predictors,
                            cl=train_labels,
                            k=1 )

# See the 6 first values of predicted class in the validation set
head(my_knn)
```

```
[1] 0 0 0 0 1 0
Levels: 0 1
```

```r
# To summarized the model
summary(my_knn)
```

```
  0   1
179  60
```

This output shows the summary of KNN model. Here we can see the levels of the model, and ny seing the summary we can identify that 60 customers are classified as success and 179 as failure.

Now, let's see the models performance analyzing the confusion matrix.

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  239


             | my_knn
valid_labels |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |       176 |         0 |       176 |
             |     1.000 |     0.000 |     0.736 |
             |     0.983 |     0.000 |           |
             |     0.736 |     0.000 |           |
-------------|-----------|-----------|-----------|
           1 |         3 |        60 |        63 |
             |     0.048 |     0.952 |     0.264 |
             |     0.017 |     1.000 |           |
             |     0.013 |     0.251 |           |
-------------|-----------|-----------|-----------|
Column Total |       179 |        60 |       239 |
             |     0.749 |     0.251 |           |
-------------|-----------|-----------|-----------|
```

```r
print(accuracy)
```

```
[1] 0.9874477
```

```
print(recall)
```

```
[1] 0.952381
```

```
print(precision)
```

```
[1] 1
```

```
print(specificity)
```

```
[1] 1
```

It proves the KNN classification model did an excellent good job to classify the customer to target for a direct-mail promotions.

VII. Conclusions

In previous years, CRISA has traditionally segmented its market based on demographic characteristics. In order to help to improve it is market segmentation in India, we developed three K-means clustering models based on the following characteristics: Purchase Behavior, Basis of Purchase, and combining both characteristics (Purchase Behavior + Basis of Purchase).

After analyzing those results, we determined that market the segmentation based on Basis of Purchase is the most effective strategy to follow. We saw that the K-means clustering model worked very well, and it will help CRISA's clients, IMRB, to target its marketing promotions more efficiently, save time and money, and provide accurate rewards to the clients that are loyal to IMRB.

Additionally, by developing the supervised classification model, KNN, the IMRB can target mail promotions with high accuracy and IMRB could implement a successful marketing campaign to enhance its business revenue.