Final Exam

Aarush Bhardwaj

05/05/2021

Setting working directory to the current file location.

```
setwd("D:/A_Sem_1/ML/Final Exam")
```

Data Importing (Importing required Libraries and dataset)

Including required libraries and setting seed.

```
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(factoextra)

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

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

library(hrbrthemes)

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

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

##       Please use hrbrthemes::import_roboto_condensed() to install Roboto
Condensed and

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

library(GGally)

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

library(viridis)

## Warning: package 'viridis' was built under R version 4.0.4
```

```
## Loading required package: viridisLite

library(NbClust)

library(readr)
library(Hmisc)

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

## Loading required package: survival

##
## Attaching package: 'survival'

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

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

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

library(tidyverse)

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

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

## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1

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

library(dplyr)
library(ggplot2)

library(ggthemes)
library(ggrepel)

## Warning: package 'ggrepel' was built under R version 4.0.4
```

```
library(ggsignif)

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

library(ggpubr)

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

library(cowplot)

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

##
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggpubr':
##
##     get_legend

## The following object is masked from 'package:ggthemes':
##
##     theme_map

set.seed(123)
```

Importing the bath soap data and checking for na values

```
library(readr)
BathSoap <- read_csv("BathSoap.csv", col_types = cols(`Member id` =
col_number(),
    SEC = col_number(), FEH = col_number(),
    MT = col_number(), SEX = col_number(),
    AGE = col_number(), EDU = col_number(),
    HS = col_number(), CHILD = col_number(),
    CS = col_number(), `Affluence Index` = col_number(),
    `No. of Brands` = col_number(), `Brand Runs` = col_number(),
    `Total Volume` = col_number(), `No. of  Trans` = col_number(),
    Value = col_number(), `Trans / Brand Runs` = col_number(),
    `Vol/Tran` = col_number(), `Avg. Price` = col_number(),
    `Pur Vol No Promo - %` = col_number(),
    `Pur Vol Promo 6 %` = col_number(), `Pur Vol Other Promo %` =
col_number(),
    `Br. Cd. 57, 144` = col_number(), `Br. Cd. 55` = col_number(),
    `Br. Cd. 272` = col_number(), `Br. Cd. 286` = col_number(),
    `Br. Cd. 24` = col_number(), `Br. Cd. 481` = col_number(),
    `Br. Cd. 352` = col_number(), `Br. Cd. 5` = col_number(),
    `Others 999` = col_number(), `Pr Cat 1` = col_number(),
    `Pr Cat 2` = col_number(), `Pr Cat 3` = col_number(),
    `Pr Cat 4` = col_number(), `PropCat 5` = col_number(),
    `PropCat 6` = col_number(), `PropCat 7` = col_number(),
    `PropCat 8` = col_number(), `PropCat 9` = col_number(),
    `PropCat 10` = col_number(), `PropCat 11` = col_number(),
```

```
    `PropCat 12` = col_number(), `PropCat 13` = col_number(),
    `PropCat 14` = col_number(), `PropCat 15` = col_number()))

summary(BathSoap)

##    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   Min.   :  0.00
##  1st Qu.: 1.420     1st Qu.: 250.51   1st Qu.: 9.76   1st Qu.: 88.00
##  Median : 1.845     Median : 361.52   Median :11.25   Median : 95.00
##  Mean   : 2.618     Mean   : 415.05   Mean   :11.83   Mean   : 91.31
##  3rd Qu.: 2.690     3rd Qu.: 490.89   3rd Qu.:13.42   3rd Qu.:100.00
##  Max.   :23.000     Max.   :2525.00   Max.   :33.33   Max.   :100.00
##  Pur Vol Promo 6 % Pur Vol Other Promo % Br. Cd. 57, 144    Br. Cd. 55
##  Min.   : 0.000    Min.   :  0.000       Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 0.000    1st Qu.:  0.000       1st Qu.:  0.00   1st Qu.:  0.00
##  Median : 0.000    Median :  0.000       Median :  8.00   Median :  0.00
##  Mean   : 5.358    Mean   :  3.342       Mean   : 18.41   Mean   : 12.94
##  3rd Qu.: 7.000    3rd Qu.:  4.000       3rd Qu.: 28.25   3rd Qu.:  9.25
##  Max.   :67.000    Max.   :100.000       Max.   :100.00   Max.   :100.00
##   Br. Cd. 272      Br. Cd. 286        Br. Cd. 24       Br. Cd. 481
##  Min.   : 0.000   Min.   :  0.000   Min.   :  0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.: 0.000
```

```
##  Median : 0.000   Median :  0.000   Median :  0.000   Median : 0.000
##  Mean   : 3.317   Mean   :  3.397   Mean   :  1.933   Mean   : 2.595
##  3rd Qu.: 2.000   3rd Qu.:  0.000   3rd Qu.:  0.000   3rd Qu.: 1.000
##  Max.   :96.000   Max.   :100.000   Max.   :100.000   Max.   :90.000
##   Br. Cd. 352      Br. Cd. 5        Others 999        Pr Cat 1
##  Min.   : 0.00   Min.   : 0.000   Min.   :  0.00   Min.   :  0.0
##  1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 27.88   1st Qu.:  6.0
##  Median : 0.00   Median : 0.000   Median : 52.55   Median : 18.0
##  Mean   : 3.42   Mean   : 1.815   Mean   : 52.20   Mean   : 27.9
##  3rd Qu.: 0.00   3rd Qu.: 1.000   3rd Qu.: 77.85   3rd Qu.: 42.0
##  Max.   :99.00   Max.   :97.000   Max.   :100.00   Max.   :100.0
##     Pr Cat 2         Pr Cat 3         Pr Cat 4         PropCat 5
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.000   Min.   :  0.00
##  1st Qu.: 21.00   1st Qu.:  0.00   1st Qu.:  0.000   1st Qu.: 16.00
##  Median : 52.50   Median :  0.00   Median :  0.000   Median : 44.00
##  Mean   : 49.32   Mean   : 13.92   Mean   :  8.863   Mean   : 45.72
##  3rd Qu.: 75.00   3rd Qu.: 12.00   3rd Qu.:  7.000   3rd Qu.: 72.00
##  Max.   :100.00   Max.   :100.00   Max.   :100.000   Max.   :100.00
##    PropCat 6        PropCat 7        PropCat 8        PropCat 9
##  Min.   : 0.000   Min.   :  0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.:  0.000   1st Qu.: 0.000   1st Qu.: 0.000
##  Median : 2.000   Median :  1.000   Median : 1.000   Median : 0.000
##  Mean   : 9.238   Mean   :  9.688   Mean   : 8.018   Mean   : 3.085
##  3rd Qu.:10.000   3rd Qu.:  8.000   3rd Qu.: 9.000   3rd Qu.: 3.000
##  Max.   :97.000   Max.   :100.000   Max.   :96.000   Max.   :41.000
##    PropCat 10       PropCat 11       PropCat 12       PropCat 13
##  Min.   :  0.000   Min.   : 0.000   Min.   : 0.00   Min.   :  0.000
##  1st Qu.:  0.000   1st Qu.: 0.000   1st Qu.: 0.00   1st Qu.:  0.000
##  Median :  0.000   Median : 0.000   Median : 0.00   Median :  0.000
##  Mean   :  2.037   Mean   : 2.942   Mean   : 0.62   Mean   :  2.505
##  3rd Qu.:  0.000   3rd Qu.: 1.000   3rd Qu.: 0.00   3rd Qu.:  1.000
##  Max.   :100.000   Max.   :90.000   Max.   :33.00   Max.   :100.000
##    PropCat 14       PropCat 15
##  Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  0.00   1st Qu.: 0.000
##  Median :  0.00   Median : 0.000
##  Mean   : 13.65   Mean   : 2.535
##  3rd Qu.: 12.00   3rd Qu.: 0.000
##  Max.   :100.00   Max.   :84.000
```

Data Prepration

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

a) Considering the variables that describe the purchase behavior:

From the dataset we can see thta the variables that describe the purchase behavior are:

-> vol/Trans

-> Brand Runs

-> No. of Trans

-> No. of Brands

-> Others999

-> Value

-> Loyality_Brand

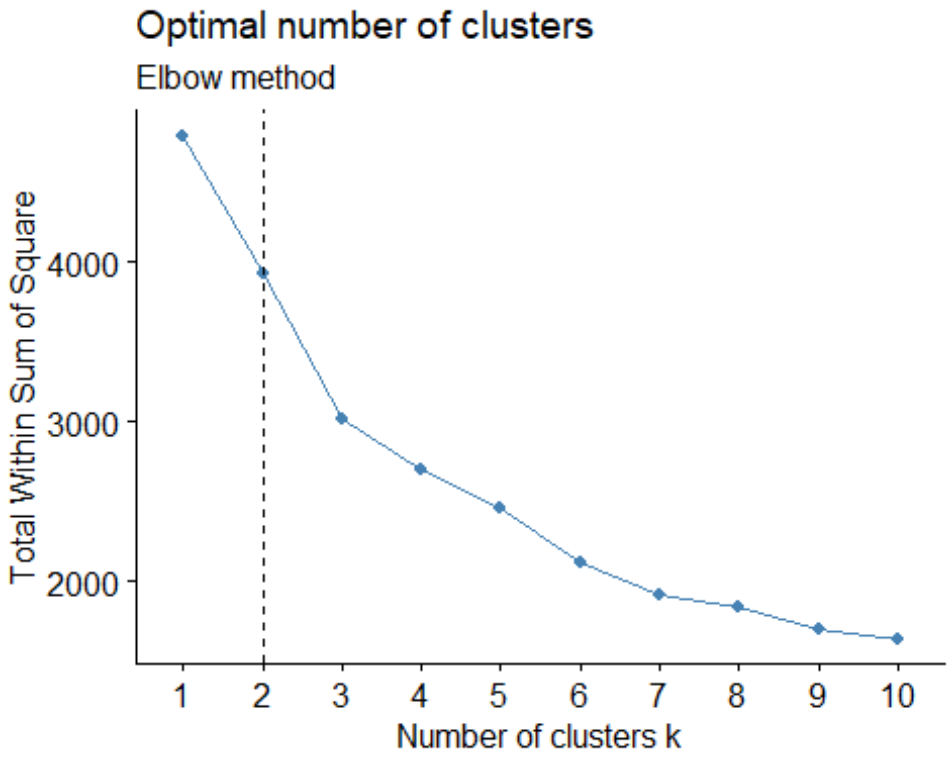
Now in order to find the brand loyalty, we will find the maximum value in brands. This maximum value will correspond to the loyalty of the brand to the customer.

We will do this by creating a new variable named Brand_Loyalty and store in this variable, the max values that correspond to that brand.

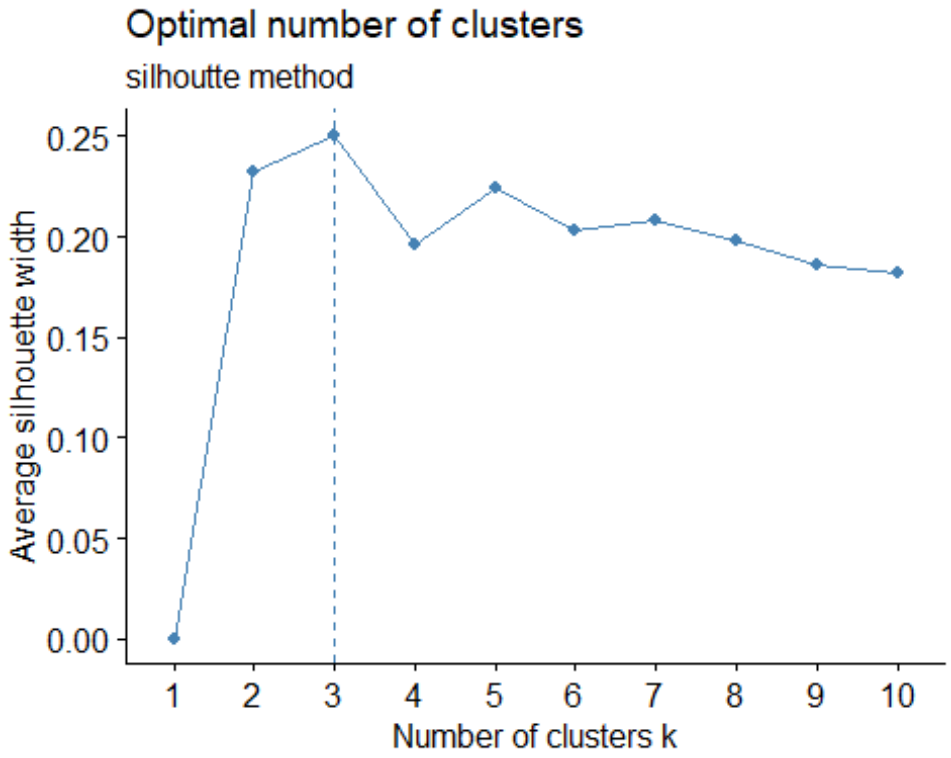
Also a quick summary() review shows us that there are no Na values, So we will just normalize the data.

After normalizing the data, we will find the optimal number of clusters using the fviz_nbclust() methhod and use method as silouhette, euclidean and gap_stat.

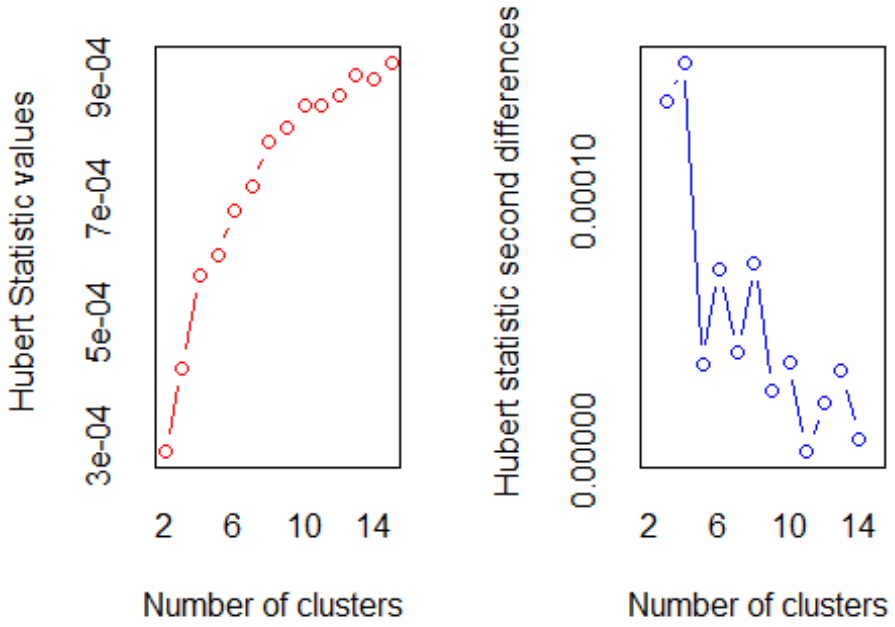
```
cust_loyalty1 <- BathSoap[,23:30]

BathSoap$Brand_Loyalty <- as.numeric(apply(cust_loyalty1,1,max))

Data1 <- BathSoap[,12:19,31,47]
scale_Data1 <- as.data.frame(scale(Data1))

fviz_nbclust(scale_Data1, kmeans, method = 'wss' ) +
geom_vline(xintercept = 2, linetype = 2)+
  labs(subtitle = 'Elbow method')
```



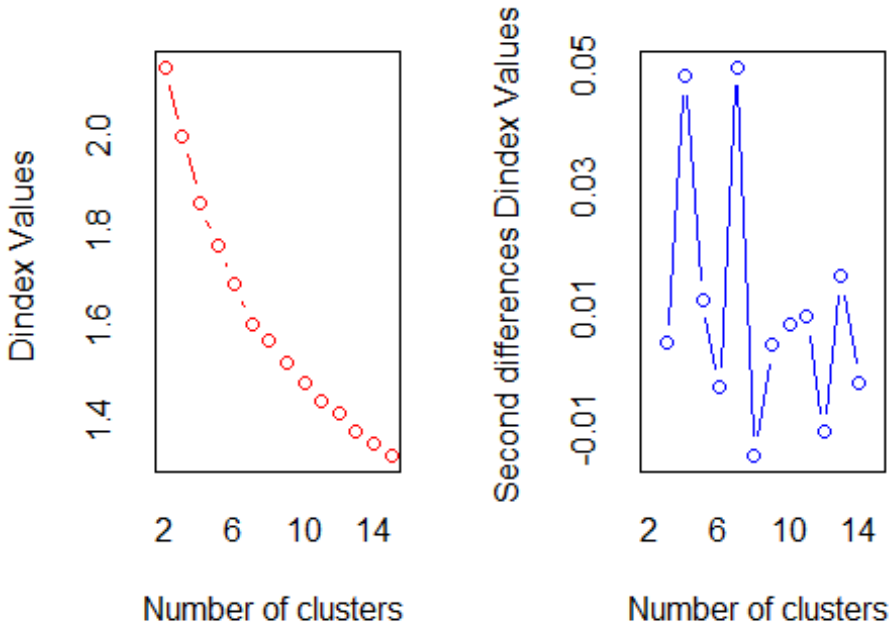

```
fviz_nbclust(scale_Data1, kmeans, method = 'silhouette' ) +
  labs(subtitle = 'silhoutte method')
```



```
NbClust(data = scale_Data1, diss = NULL, distance = "euclidean",
        min.nc = 2, max.nc = 15, method = "kmeans")
```




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




```
## *** : The D index is a graphical method of determining the number of
clusters.
##                 In the plot of D index, we seek a significant knee (the
significant peak in Dindex
##                 second differences plot) that corresponds to a significant
increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 6 proposed 2 as the best number of clusters
## * 4 proposed 3 as the best number of clusters
## * 2 proposed 4 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 4 proposed 6 as the best number of clusters
## * 4 proposed 7 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
## * 1 proposed 15 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

```
## $All.index
##        KL       CH Hartigan      CCC     Scott      Marriot    TrCovW
TraceW
## 2  2.1052 179.5663  94.5206  -6.1861  564.5896 3.316623e+19 371065.15
3685.365
## 3  0.5281 150.9818 106.3561 -12.5718  948.9017 3.932822e+19 237214.07
3182.358
## 4  2.2213 153.7802  62.7806 -10.4164 1410.0983 3.241579e+19 166066.19
2701.147
## 5  0.4748 142.9391  93.5309 -10.5973 1848.9265 2.437486e+19 128043.19
2443.732
## 6  1.7572 150.7790  62.8227  -6.9925 2536.7147 1.115493e+19  86477.77
2111.773
## 7  5.0012 149.1570  28.0729  -3.8640 2701.1542 1.154344e+19  75823.06
1909.789
## 8  0.4081 137.6792  37.6789  -4.0673 2945.9465 1.002611e+19  64330.70
1823.466
## 9  0.8627 132.6222  39.7423  -2.9168 3110.0523 9.652812e+18  57471.80
1714.352
## 10 1.7479 130.0092  28.1982  -1.3397 3358.6057 7.875169e+18  52299.91
1606.333
## 11 1.3600 125.2078  23.3644  -0.7318 3564.2295 6.764096e+18  45900.23
1533.063
## 12 0.6195 120.2600  29.1137  -0.4319 3761.3362 5.795839e+18  41276.95
1474.570
## 13 1.4913 117.9218  22.7189   0.5636 3921.2595 5.210558e+18  38874.17
1405.004
## 14 1.2984 114.6158  19.3505   1.0552 4106.5869 4.437201e+18  35041.07
1352.651
## 15 0.6526 111.1356  23.8411   1.3277 4216.5471 4.240760e+18  33582.68
1309.413
##    Friedman  Rubin Cindex     DB Silhouette   Duda  Pseudot2   Beale
Ratkowsky
## 2   14.0888 1.3003 0.2136 1.8196     0.2348 1.1098  -46.4018 -0.5214
0.2956
## 3   22.6620 1.5058 0.1913 1.7391     0.1887 1.5194 -110.4100 -1.8000
0.3298
## 4   30.6056 1.7741 0.1969 1.6285     0.1875 1.0136   -4.1154 -0.0706
0.3197
## 5   36.4110 1.9609 0.1849 1.4827     0.1857 1.1259  -23.2596 -0.5882
0.3071
## 6   39.4752 2.2692 0.2443 1.3533     0.1999 1.6810  -73.3282 -2.1301
0.3049
## 7   40.4041 2.5092 0.2046 1.3275     0.2080 1.2455  -33.7102 -1.0335
0.2928
## 8   46.9593 2.6280 0.2272 1.3868     0.1775 1.2250  -27.7380 -0.9645
0.2780
## 9   47.9965 2.7952 0.2287 1.3818     0.1732 2.1493 -112.2921 -2.8046
0.2667
## 10  48.5596 2.9832 0.2200 1.3534     0.1871 1.3054  -44.6843 -1.2281
```

```
0.2574
## 11  51.5472 3.1258 0.2170 1.4127     0.1875 1.6984  -67.8519 -2.1542
0.2484
## 12  55.5366 3.2498 0.2192 1.3782     0.1853 1.1293   -8.3584 -0.5941
0.2400
## 13  57.9325 3.4107 0.2069 1.3610     0.1861 1.5098  -21.6098 -1.7489
0.2330
## 14  62.5977 3.5427 0.2090 1.3488     0.1877 1.9608  -49.9795 -2.5346
0.2262
## 15  63.5008 3.6597 0.2069 1.3396     0.1753 1.1473  -11.4250 -0.6666
0.2199
##         Ball Ptbiserial    Frey McClain   Dunn Hubert SDindex Dindex
SDbw
## 2  1842.6827     0.2792  0.1756  0.7215 0.0191  3e-04  2.2303 2.1443
1.7029
## 3  1060.7861     0.3374 -0.1301  1.2017 0.0191  4e-04  2.3291 2.0004
1.4776
## 4   675.2866     0.4000  0.6751  1.3224 0.0246  6e-04  2.3925 1.8606
1.2671
## 5   488.7464     0.3809 -0.0246  1.8271 0.0334  6e-04  2.1624 1.7692
1.2388
## 6   351.9621     0.4001  0.4613  1.8858 0.0422  7e-04  2.2075 1.6893
0.9449
## 7   272.8270     0.3893  0.8032  2.2350 0.0408  7e-04  2.1299 1.6063
1.0145
## 8   227.9332     0.3731  0.2871  2.5647 0.0340  8e-04  2.2970 1.5731
0.8123
## 9   190.4836     0.3683  0.0457  2.8285 0.0360  8e-04  2.4668 1.5255
0.7347
## 10  160.6333     0.3747  1.7856  2.8870 0.0360  9e-04  2.3143 1.4817
0.7308
## 11  139.3693     0.3451  0.3565  3.5057 0.0365  9e-04  2.4020 1.4452
0.6472
## 12  122.8808     0.3392  0.3155  3.7286 0.0324  9e-04  2.3524 1.4172
0.5775
## 13  108.0772     0.3317  0.2742  4.0352 0.0319  9e-04  2.5570 1.3787
0.5581
## 14   96.6179     0.3272  1.4170  4.2381 0.0367  9e-04  2.3773 1.3554
0.5134
## 15   87.2942     0.3075  0.1496  4.8740 0.0367  9e-04  2.5497 1.3298
0.5277
##
## $All.CriticalValues
##    CritValue_Duda CritValue_PseudoT2 Fvalue_Beale
## 2          0.8385            90.3160            1
## 3          0.8288            66.7245            1
## 4          0.8269            64.2689            1
## 5          0.8160            46.9010            1
## 6          0.8115            42.0371            1
## 7          0.7848            46.9016            1
```

```
## 8          0.7978            38.2709            1
## 9          0.7884            56.3609            1
## 10         0.7957            49.0269            1
## 11         0.7784            46.9833            1
## 12         0.7177            28.7089            1
## 13         0.7080            26.3915            1
## 14         0.7015            43.4039            1
## 15         0.7213            34.3952            1
##
## $Best.nc
##                       KL       CH Hartigan     CCC    Scott      Marriot
TrCovW
## Number_clusters 7.0000   2.0000   4.0000 15.0000   6.0000 6.000000e+00
3.0
## Value_Index     5.0012 179.5663  43.5755  1.3277 687.7882 1.360844e+19
133851.1
##                     TraceW Friedman   Rubin Cindex     DB Silhouette   Duda
## Number_clusters   4.0000   3.0000  7.0000 5.0000 7.0000     2.0000 2.0000
## Value_Index     223.7972   8.5732 -0.1212 0.1849 1.3275     0.2348 1.1098
##                   PseudoT2   Beale Ratkowsky     Ball PtBiserial Frey
McClain
## Number_clusters   2.0000  2.0000    3.0000   3.0000     6.0000    1
2.0000
## Value_Index     -46.4018 -0.5214    0.3298 781.8966     0.4001   NA
0.7215
##                   Dunn Hubert SDindex Dindex    SDbw
## Number_clusters 6.0000      0  7.0000      0 14.0000
## Value_Index     0.0422      0  2.1299      0  0.5134
##
## $Best.partition
##   [1] 1 2 2 1 1 2 1 1 2 1 2 2 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1
1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 2 2 1 1 2 2 1 1 2 1 2 2 1 1 1
2 1 1
##  [75] 1 1 1 1 2 1 1 2 1 1 1 2 1 2 1 1 2 2 1 2 2 2 1 2 1 1 1 1 2 2 1 2 2 1
2 1 1
## [112] 1 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 1 1 2 2 2 1 1 1 1 1 2 2 1 1 1 1 1 2
1 1 1
## [149] 2 1 1 1 1 1 2 2 2 1 1 2 1 1 2 2 2 2 2 1 2 1 2 1 1 1 2 2 1 1 1 1 1 2
2 2 2
## [186] 1 2 2 2 2 1 2 1 1 2 2 2 2 2 1 1 1 2 1 1 1 1 2 1 2 1 1 2 1 2 1 1 2 1
2 1 2
## [223] 1 1 1 1 1 2 1 1 1 2 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 2 1 1
2 1 2
## [260] 2 1 1 2 2 2 1 2 1 2 2 2 2 1 1 1 1 2 2 1 1 2 1 2 2 2 2 1 2 2 2 2 1 1
1 1 1
## [297] 2 2 2 2 1 1 1 2 1 1 2 1 1 1 2 2 2 2 2 2 1 1 2 1 2 2 2 2 1 1 2 1 2 1
2 1 1
## [334] 1 1 2 2 2 1 1 1 1 1 1 2 1 2 1 2 1 1 1 2 1 2 2 2 2 2 2 1 1 2 2 2 1 1
2 2 1
```

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

The optimal value of k according to above plots should be

silhouette = 2 Elbow = 4 Nbclust = 2

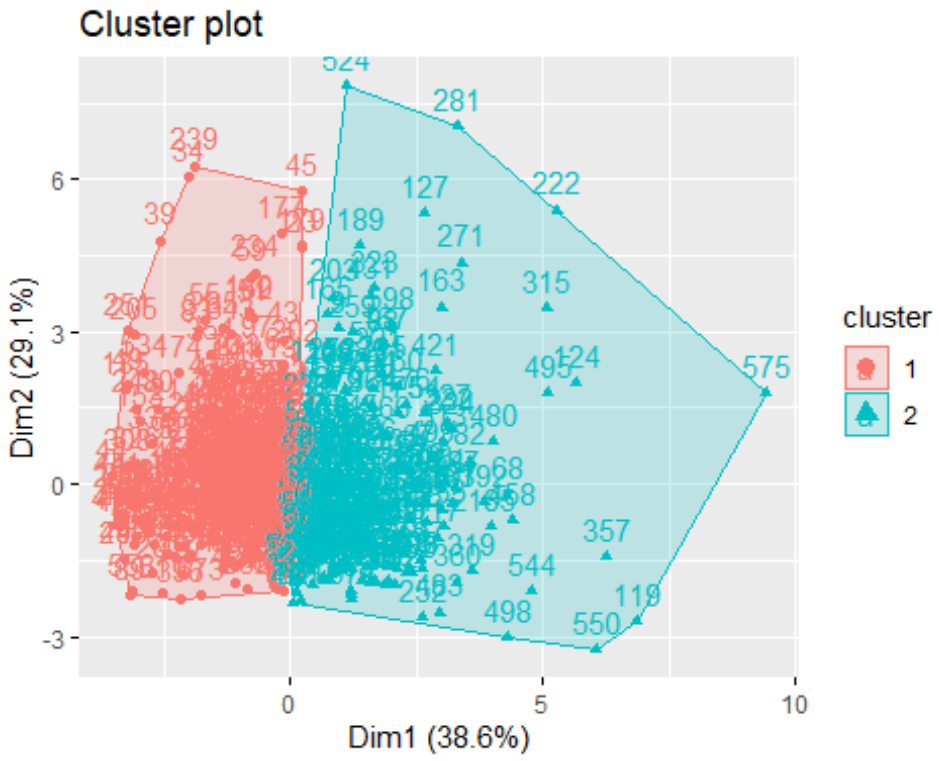
So we will Consider k = 2 and 4 to check how the formation of cluster changes with the change in value of k.

Now we will run kmeans algo with k = 2 and 4 and nstart = 30. After running it, we will plot the clusters using fviz_cluster()

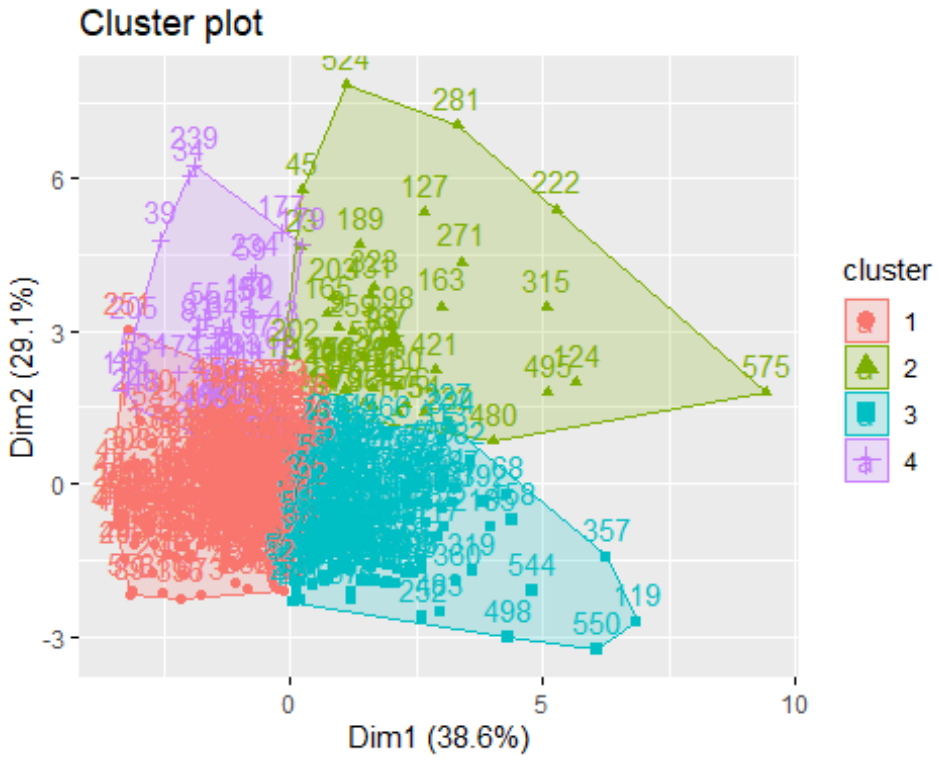
After plotting, we will store the centers of the model in result1 variable in the form of data frame

Also we will print the size of the model.

```
Model_Purchase_Behav <- kmeans(scale_Data1,2, nstart = 30)
fviz_cluster(Model_Purchase_Behav, scale_Data1)
```




```
Model_Purchase_Behav1 <- kmeans(scale_Data1,4, nstart = 30)
fviz_cluster(Model_Purchase_Behav1, scale_Data1)
```



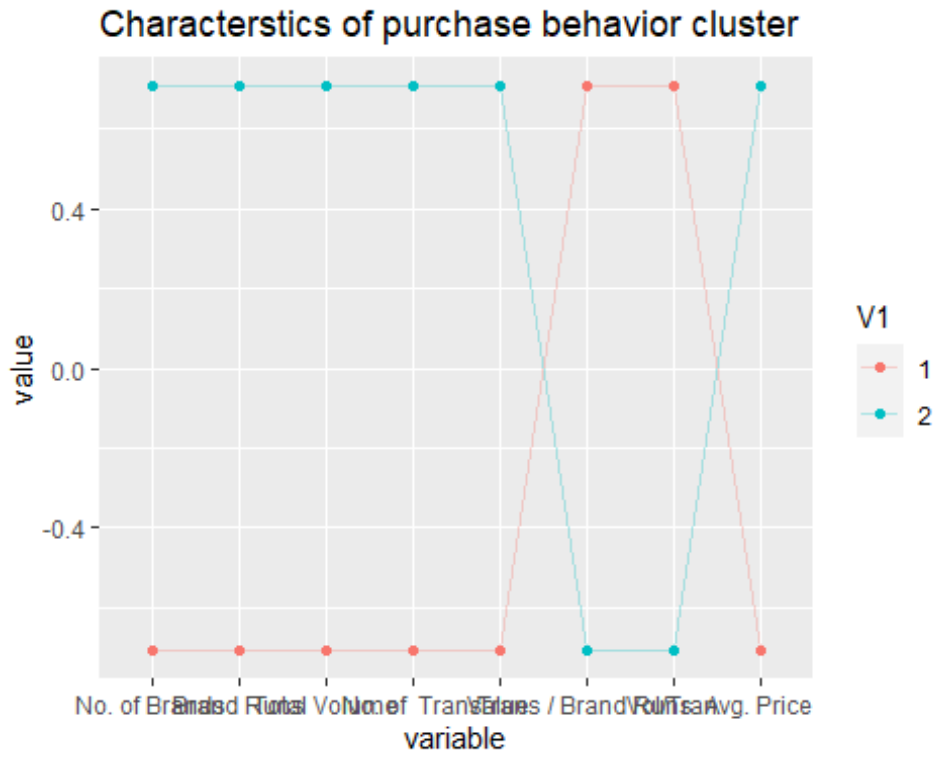
```
result1 <- as.data.frame(cbind(1:nrow(Model_Purchase_Behav$centers),
Model_Purchase_Behav$centers))

result1$V1 <- as.factor(result1$V1)

Model_Purchase_Behav$size

## [1] 334 266
```

After seeing the clusters, we can see that k = 2 is good option as cluster formation is clear.

The size of the model is334, 266

Finally we will visualize the clusters using the ggparcoord() method, which will show us the behavior of the variables within the cluster.

```
ggparcoord(result1,
           columns = 2:ncol(result1), groupColumn = 1,
           showPoints = TRUE,
           title = "Characterstics of purchase behavior cluster",
           alphaLines = 0.3)
```



Cluster Info:

Cluster No..of.Brands Brand.Runs Total.Volume No..of..Trans Value Trans...Brand.Runs Vol.Tran 1 -0.5417123 -0.7088977 -0.1772315 -0.5848426 -0.3438382 0.2926739

0.3196693
2 0.4836107 0.6328645 0.1582224 0.5221150 0.3069597 -0.2612830 -0.2853830

Avg..Price Others.999 Loyality_Brand -0.3132908 -0.5477087 0.6584652 0.2796886
0.4889639 -0.5878412

-> The two clusters are well-separated on almost everything. Cluster 1 (n=283) belongs high activity & value, with low loyalty. Cluster 2 (n=317) belongs to low activity & value, with high loyalty.

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

-> cluster 2: Customers in this cluster buy from others999 brands thus indicating they are not at all brand loyal.They buy the highest number of brands and the volume of transaction is the least.

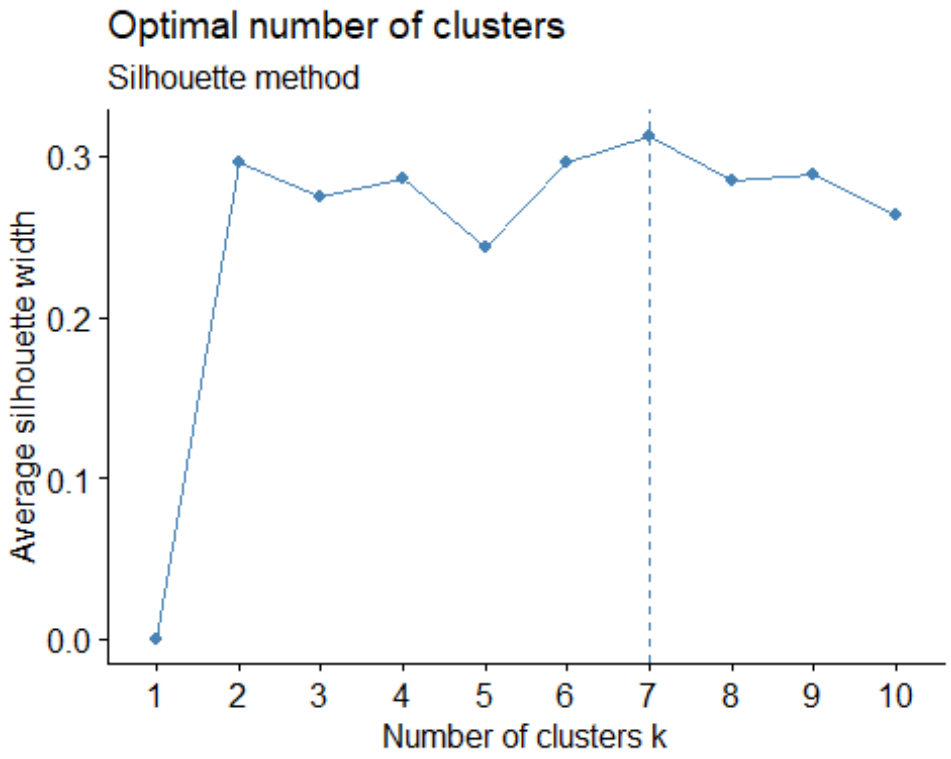
b) Considerinig variables that describe the basis of purchase.

Variables that we willbe considering this time are: -> All price categories -> selling proportions -> purchase volume with no promotion, promotion 6 and other promotions

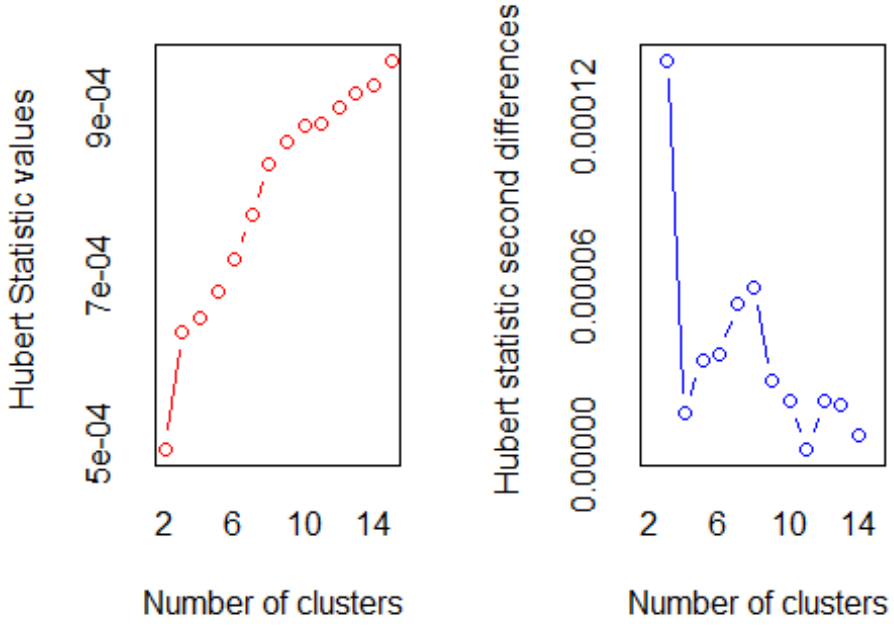
We will follow same steps as previous part, that is we will find maximum for particular columns (from 36: 46) which will give us the value for the basis of customers purchase.

then we will scale the data again and then find the number of clusters using fviz_nbclust() using silhouette, elbow and nbclust method

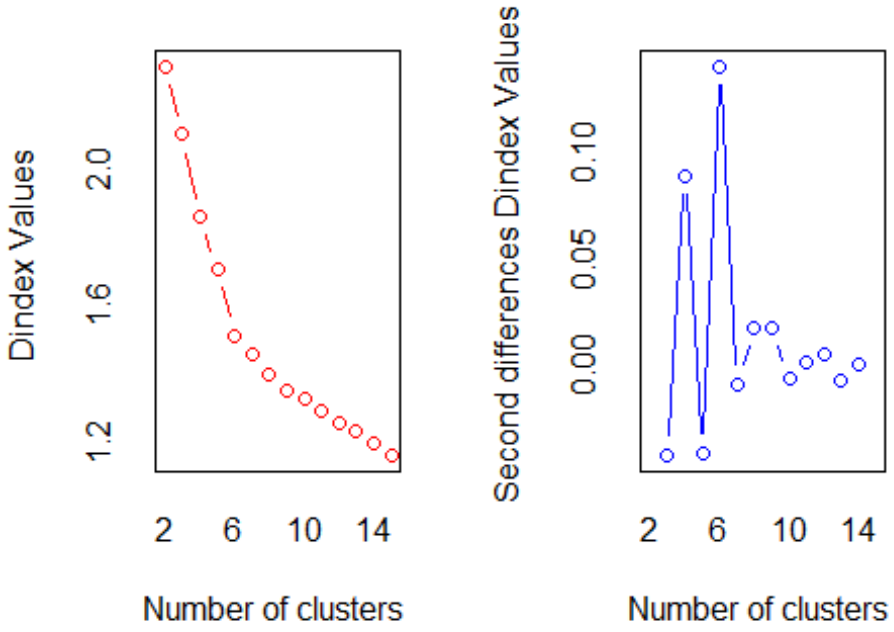
```
cust_loyalty2 <- BathSoap[,36:46]

BathSoap$Purchase_Basis_no <- as.numeric(apply(cust_loyalty2,1,which.max))
BathSoap$Purchase_Basis <- as.numeric(apply(cust_loyalty2,1,max))

Data2 <- BathSoap[,c(20:22,32:35,49)]
scale_Data2 <- as.data.frame(scale(Data2))

fviz_nbclust(scale_Data2, kmeans, method = 'silhouette')+
  labs(subtitle = 'Silhouette method')
```




```
NbClust(data = scale_Data2, diss = NULL, distance = "euclidean",
        min.nc = 2, max.nc = 15, method = "kmeans")
```



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




```
## *** : The D index is a graphical method of determining the number of
clusters.
##                 In the plot of D index, we seek a significant knee (the
significant peak in Dindex
##                 second differences plot) that corresponds to a significant
increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 4 proposed 2 as the best number of clusters
## * 2 proposed 3 as the best number of clusters
## * 6 proposed 4 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 3 proposed 6 as the best number of clusters
## * 4 proposed 7 as the best number of clusters
## * 1 proposed 11 as the best number of clusters
## * 2 proposed 15 as the best number of clusters
```

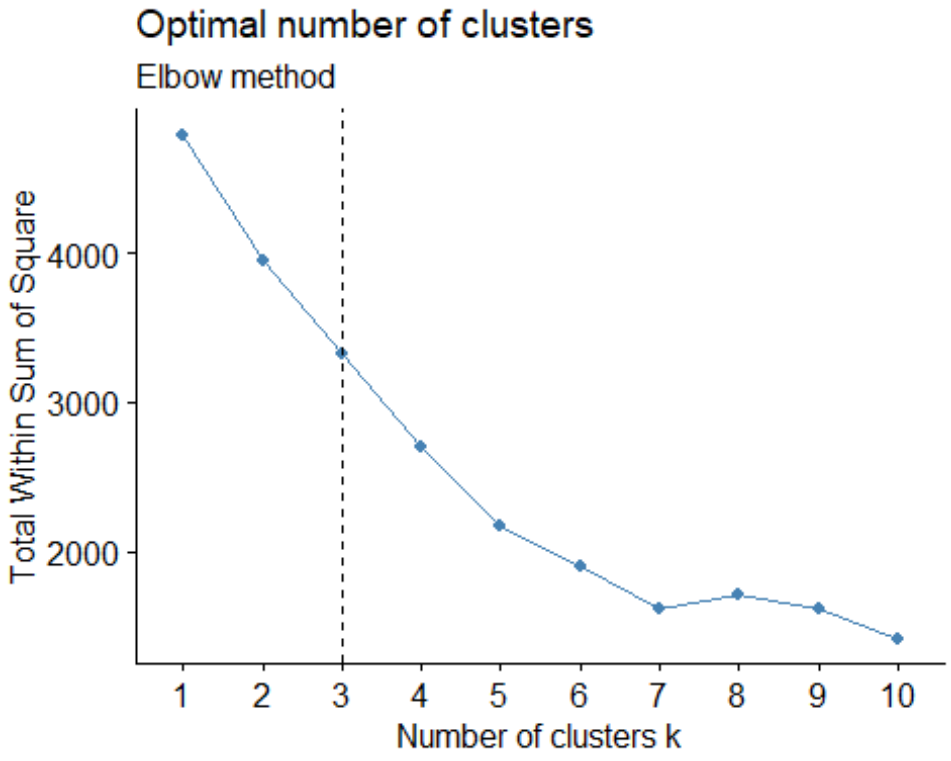
```
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  4
##
##
## *******************************************************************

## $All.index
##        KL       CH Hartigan     CCC     Scott      Marriot    TrCovW
TraceW
## 2  0.1445 103.2967 133.9734 -8.2341  884.4643 9.601040e+14 496806.19
4086.168
## 3  0.8112 129.9884 140.9893 -2.2827 1342.4746 1.006890e+15 296918.54
3338.275
## 4  2.8204 153.8628  69.6861  5.1191 2166.2379 4.535233e+14 196582.44
2700.514
## 5  0.3125 146.0657 164.6792  6.2976 2636.9765 3.233610e+14 167963.40
2417.816
## 6  1.9786 181.8238 100.5192 22.0258 3702.2945 7.887643e+13 110674.23
1893.694
## 7  2.2501 193.5874  56.2013 32.0135 4022.3387 6.297746e+13  72520.90
1619.616
## 8  1.6951 189.3671  39.6119 34.1866 4249.9653 5.628691e+13  57118.12
1479.406
## 9  0.9844 181.4277  38.7765 34.6686 4543.4909 4.367687e+13  52501.38
1386.624
## 10 1.7417 175.8607  27.6713 35.5252 4710.2996 4.083442e+13  44748.18
1301.247
## 11 0.8744 168.1794  28.6700 35.3280 4879.8752 3.724507e+13  40623.29
1242.952
## 12 1.0979 162.6621  26.4993 35.5905 5117.2162 2.984371e+13  37817.66
1185.259
## 13 1.0207 157.7662  25.5984 35.8410 5245.6091 2.827764e+13  33568.61
1134.146
## 14 0.9208 153.6880  26.4585 36.1942 5361.9997 2.701259e+13  30601.47
1086.754
## 15 2.0702 150.7859  17.4536 36.8232 5555.3712 2.246599e+13  28592.38
1039.806
##     Friedman  Rubin Cindex     DB Silhouette   Duda Pseudot2   Beale
Ratkowsky
## 2   5879.278 1.1727 0.1791 2.3773     0.2511 0.9766  10.2005  0.1265
0.2036
## 3   7272.635 1.4355 0.1521 1.8354     0.2641 1.2477 -83.9864 -1.0462
0.2945
## 4  12325.332 1.7745 0.1354 1.4075     0.2807 1.4365 -81.1254 -1.5942
0.3152
## 5  12772.558 1.9820 0.1276 1.3688     0.2432 1.4616 -69.1615 -1.6595
0.3048
## 6  15712.427 2.5305 0.1151 1.2323     0.2966 1.6836 -75.1156 -2.1334
```

```
0.3104
## 7  15887.100 2.9587 0.1393 1.1477     0.3126 1.4011 -67.5645 -1.5044
0.3064
## 8  16039.252 3.2391 0.1325 1.1637     0.3065 1.5398 -97.8080 -1.8410
0.2931
## 9  16455.248 3.4559 0.1510 1.2001     0.2891 1.4011 -46.9483 -1.5027
0.2803
## 10 16532.788 3.6826 0.1507 1.1551     0.2964 1.3972 -51.1701 -1.4923
0.2696
## 11 16917.036 3.8553 0.1426 1.1877     0.2946 1.3962 -57.8937 -1.4896
0.2591
## 12 17021.642 4.0430 0.1389 1.1684     0.2955 1.1161 -14.4607 -0.5460
0.2500
## 13 17449.322 4.2252 0.1343 1.1811     0.2874 1.2577 -31.1482 -1.0737
0.2419
## 14 17442.860 4.4095 0.1311 1.1914     0.2955 1.3745 -40.3239 -1.4281
0.2347
## 15 17816.179 4.6086 0.1286 1.2219     0.2736 1.7467 -76.5215 -2.2334
0.2282
##         Ball Ptbiserial    Frey McClain   Dunn Hubert SDindex Dindex
SDbw
## 2  2043.0840     0.3581 -0.0886  0.5770 0.0143  5e-04  2.7038 2.2987
1.1989
## 3  1112.7583     0.4956  0.3878  0.8291 0.0183  6e-04  2.7372 2.1008
1.2266
## 4   675.1284     0.5162  1.5998  1.1344 0.0116  6e-04  2.3927 1.8606
1.1090
## 5   483.5631     0.4402 -0.0797  1.8783 0.0129  7e-04  2.2619 1.7090
1.0320
## 6   315.6157     0.4810 -0.1817  1.8593 0.0129  7e-04  1.9748 1.5161
0.8237
## 7   231.3737     0.4974  0.5757  1.8119 0.0162  8e-04  2.2167 1.4627
0.9204
## 8   184.9258     0.4790  0.6810  2.0895 0.0228  8e-04  2.1417 1.4006
0.7618
## 9   154.0694     0.4623 -0.0190  2.3292 0.0213  9e-04  2.2273 1.3559
0.7258
## 10  130.1247     0.4680  0.2973  2.3184 0.0218  9e-04  2.5580 1.3283
0.8197
## 11  112.9957     0.4636  0.3242  2.4303 0.0268  9e-04  2.3086 1.2943
0.6922
## 12   98.7716     0.4589  0.7380  2.5306 0.0213  9e-04  2.2779 1.2620
0.6438
## 13   87.2420     0.4356  0.4060  2.8894 0.0267  9e-04  2.4444 1.2348
0.6276
## 14   77.6253     0.4280  1.1933  3.0387 0.0152  9e-04  2.2757 1.2005
0.5839
## 15   69.3204     0.4019  0.4024  3.5049 0.0119  1e-03  2.6164 1.1660
0.5550
##
```

```
## $All.CriticalValues
##    CritValue_Duda CritValue_PseudoT2 Fvalue_Beale
## 2          0.8442            78.4343       0.9982
## 3          0.8376            82.0432       1.0000
## 4          0.7918            70.2187       1.0000
## 5          0.8058            52.7915       1.0000
## 6          0.8048            44.8796       1.0000
## 7          0.8054            57.0091       1.0000
## 8          0.7998            69.8572       1.0000
## 9          0.7957            42.0964       1.0000
## 10         0.7962            46.0843       1.0000
## 11         0.7953            52.5003       1.0000
## 12         0.7940            36.0576       1.0000
## 13         0.7802            42.8228       1.0000
## 14         0.7831            40.9925       1.0000
## 15         0.7587            56.9195       1.0000
##
## $Best.nc
##                       KL       CH Hartigan     CCC    Scott      Marriot
TrCovW
## Number_clusters 4.0000   7.0000   5.0000 15.0000    6.000 4.000000e+00
3.0
## Value_Index     2.8204 193.5874  94.9931 36.8232 1065.318 4.232041e+14
199887.6
##                    TraceW Friedman   Rubin Cindex     DB Silhouette   Duda
## Number_clusters   4.0000    4.000  7.0000 6.0000 7.0000     7.0000 2.0000
## Value_Index     355.0632 5052.698 -0.1478 0.1151 1.1477     0.3126 0.9766
##                   PseudoT2  Beale Ratkowsky     Ball PtBiserial Frey McClain
## Number_clusters   2.0000 2.0000    4.0000   3.0000     4.0000    1   2.000
## Value_Index      10.2005 0.1265    0.3152 930.3257     0.5162   NA   0.577
##                     Dunn Hubert SDindex Dindex   SDbw
## Number_clusters 11.0000      0  6.0000      0 15.000
## Value_Index      0.0268      0  1.9748      0  0.555
##
## $Best.partition
##   [1] 4 4 2 2 3 4 4 2 4 1 3 1 2 4 4 4 4 4 4 3 2 2 2 2 2 1 4 4 2 2 2 2 2 2
2 2 2
##  [38] 2 2 2 4 2 2 2 2 4 2 4 2 1 2 4 2 4 2 4 2 2 2 4 4 2 2 4 1 2 4 1 2 4 4
2 2 3
##  [75] 2 1 4 2 4 2 4 2 2 1 3 4 3 4 1 2 1 2 2 1 4 4 2 4 2 4 4 4 2 3 2 4 1 4
4 3 2
## [112] 4 4 3 3 1 4 2 4 4 3 1 4 1 4 2 4 4 4 1 1 4 1 4 2 1 4 4 4 2 2 3 4 2 1
2 2 3
## [149] 4 3 4 4 2 2 4 1 1 2 4 3 2 2 4 4 4 4 4 4 4 4 2 4 4 2 4 4 4 2 2 4 4 4
4 3 4
## [186] 4 2 1 4 3 2 1 4 4 4 4 4 4 1 4 4 2 2 2 2 4 2 4 4 1 4 2 3 4 4 1 4 1 2
1 2 4
## [223] 2 1 4 4 4 4 1 3 2 4 2 4 2 2 2 2 2 1 4 4 3 4 2 4 2 1 1 2 4 3 4 3 4 4
4 3 4
## [260] 4 1 4 1 4 4 4 4 4 4 1 4 4 4 4 4 3 1 4 4 3 4 2 2 4 4 4 4 4 4 1 3 1 4
```

```
3 4 4
## [297] 2 4 4 1 4 4 2 4 1 4 4 4 3 1 1 3 4 1 4 4 4 2 4 1 3 3 4 4 4 2 4 1 4 4
4 4 4
## [334] 4 4 3 3 1 4 4 4 4 2 1 4 3 3 1 4 4 1 1 1 4 1 4 1 4 1 3 1 1 1 1 1 3 3
3 1 2
## [371] 4 1 4 4 2 1 4 1 4 4 1 1 3 1 4 1 4 4 4 4 1 1 1 1 3 1 1 4 1 4 4 1 4 1
4 1 1
## [408] 4 4 4 4 1 1 4 1 3 4 1 4 4 4 4 1 4 4 4 3 1 1 4 4 2 4 1 1 4 1 1 4 1 4
4 4 1
## [445] 1 4 4 4 4 4 1 4 4 4 4 4 1 4 3 1 3 3 3 1 1 4 4 4 4 1 3 1 1 1 4 3 4 3
1 1 1
## [482] 3 4 1 1 4 1 1 1 1 3 4 1 4 1 1 1 1 4 4 2 3 4 4 1 1 4 4 1 3 1 1 4 3 3
4 1 3
## [519] 1 4 4 4 4 2 4 4 4 4 4 4 3 3 4 4 4 4 2 4 4 4 4 3 4 1 3 4 4 4 4 1 4 4
1 4 4
## [556] 4 4 4 3 1 3 4 1 4 3 3 4 3 4 1 4 4 1 1 4 4 1 1 1 3 3 1 1 3 4 3 4 4 4
4 4 1
## [593] 3 4 4 4 1 4 3 4
```

```
fviz_nbclust(scale_Data2, kmeans, method = 'wss') +
    geom_vline(xintercept = 3, linetype = 2)+
  labs(subtitle = 'Elbow method')
```



The value from the above plots are:

silhouette = 6 Elbow = 3 Nbclust = 4

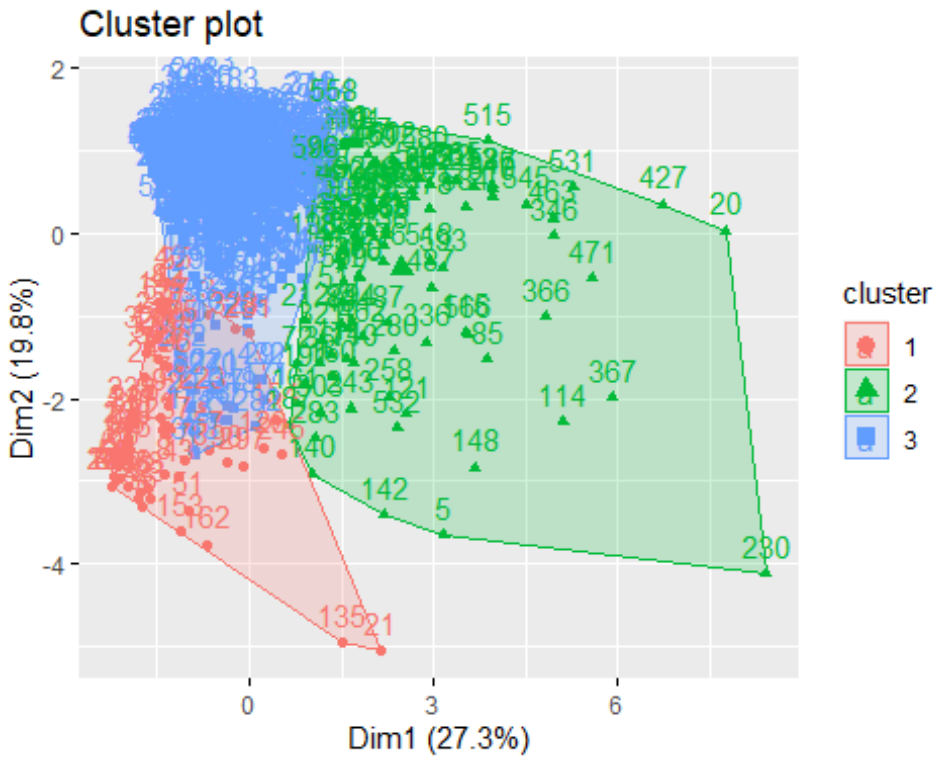
Considering majority rule, the best number of clusters is 3

But we will run kmeans model on scaled data2 , with value of k =3,4 and 7 to check how the formation of cluster changes with the change in value of k.

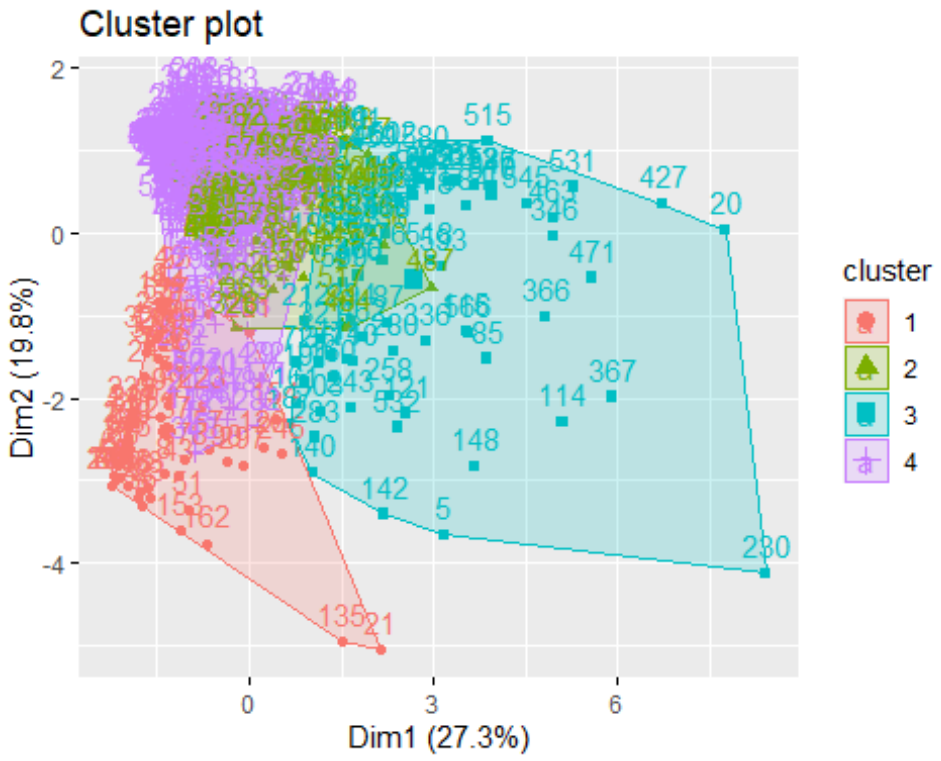
After running kmeans, we will store the centers in a data frame named result2.

And finally, we will show the size of the Model.

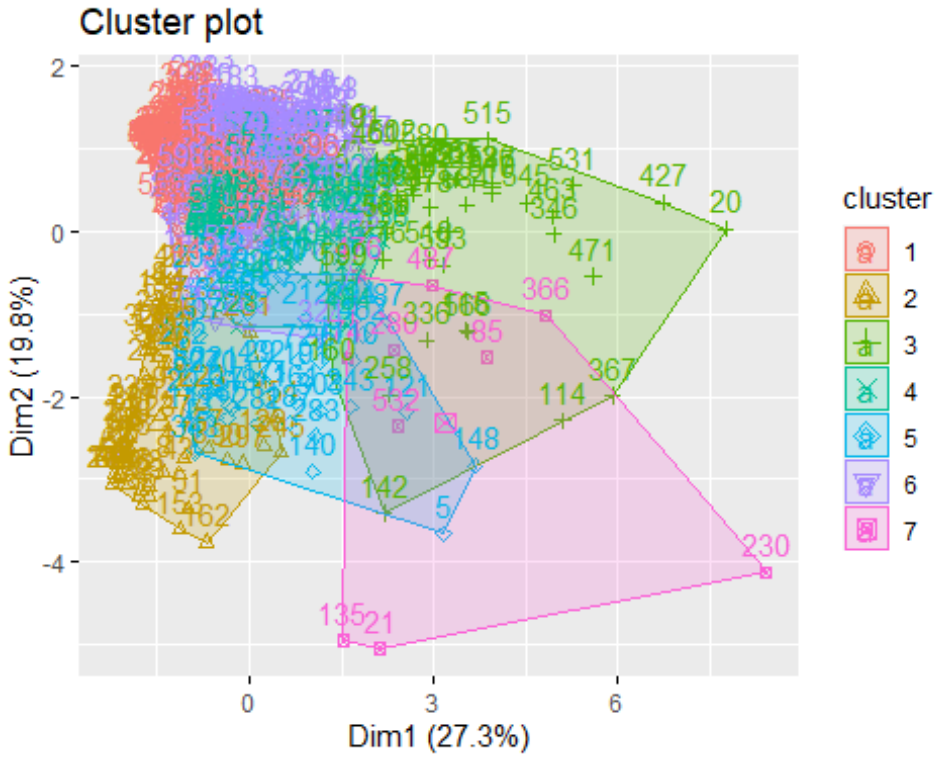
```
Model_Purchase_Basis <- kmeans(scale_Data2, 3, nstart = 30)

fviz_cluster(Model_Purchase_Basis, scale_Data2)
```




```
Model_Purchase_Basis1 <- kmeans(scale_Data2, 4, nstart = 30)

fviz_cluster(Model_Purchase_Basis1, scale_Data2)
```




```
Model_Purchase_Basis2 <- kmeans(scale_Data2, 7, nstart = 30)

fviz_cluster(Model_Purchase_Basis2, scale_Data2)
```



```
result2 <- as.data.frame(cbind(1:nrow(Model_Purchase_Basis$centers),
Model_Purchase_Basis$centers))

result2$V1 <- as.factor(result2$V1)

Model_Purchase_Basis$size
```

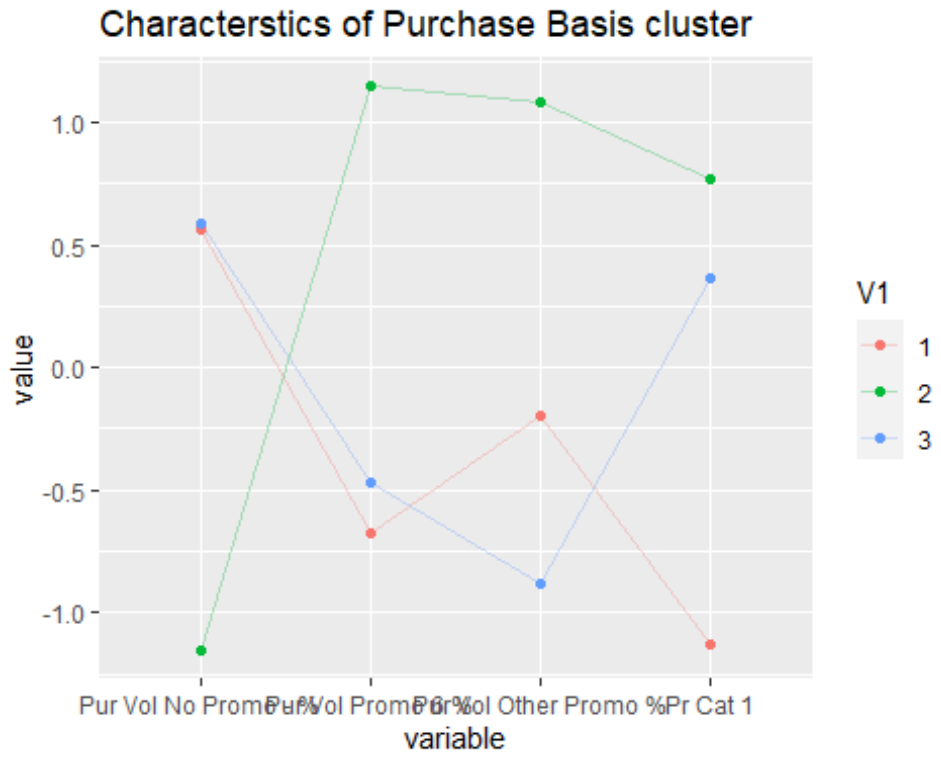
```
## [1]  67 105 428
```

The above comparison shows that the clusters are much more clearly formed for k = 3 but still have minor overlapping as comapred to the cluster formed with k =4 and 7.

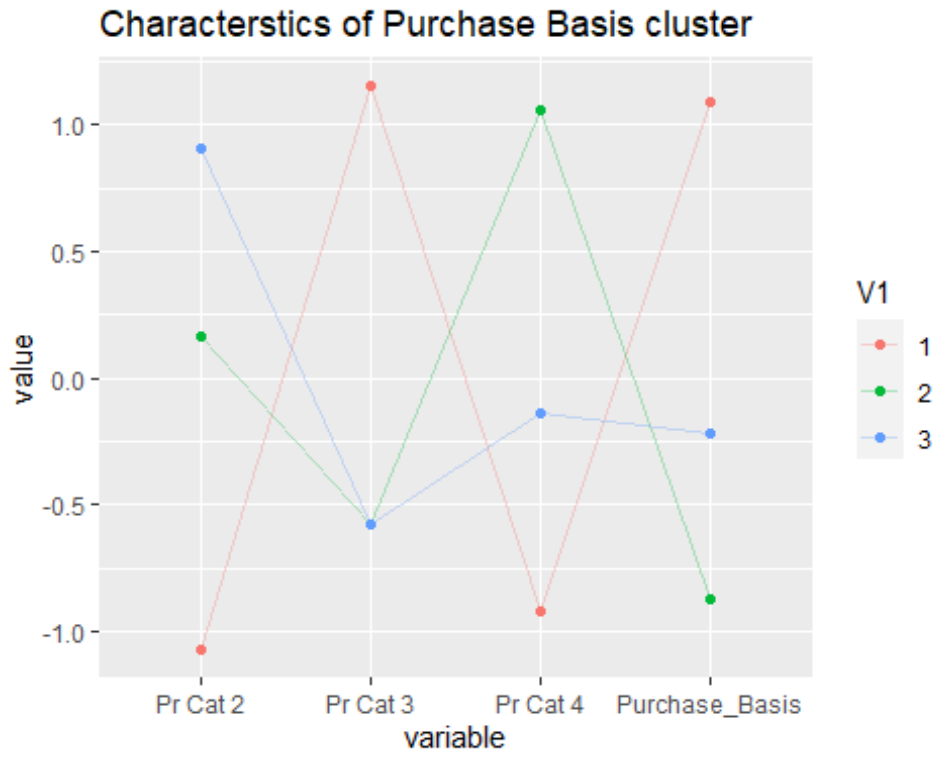
And also The size of the model is67, 105, 428

Finally we will visualize the behavior of the variables within cluster.

```
ggparcoord(result2,
           columns = 2:5, groupColumn = 1,
           showPoints = TRUE,
           title = "Characterstics of Purchase Basis cluster",
           alphaLines = 0.3)
```




```
#ncol(result2)
ggparcoord(result2,
           columns = 6:9, groupColumn = 1,
           showPoints = TRUE,
           title = "Characterstics of Purchase Basis cluster",
           alphaLines = 0.3)
```



Cluster Info:

Pur.Vol.No.Promo Pur.Vol.Promo.6 Pur.Vol.Other.Promo Pr.Cat.1 Pr.Cat.2 Pr.Cat.3 Pr.Cat.4
purchase1_on 0.006545016 -0.003720882 -0.0039774143 -0.1214132 0.2471883 -0.0800433 -0.04561153 0.02464361 -0.018171951 0.025594588 0.0008760163
0.2702850 -0.2532167 -0.1107455 0.09322033 -0.17959512 0.021950000 -0.054763345
0.0148265766 -0.2220914 -0.3707842 0.6540099 -0.06039009 0.39062703

-> Cluster1: I shows the behavior of Customers purchase products from a single price category(pr.cat 3). Their purchases are affected by promotional offers.The customers purchase products of a specific price category mostly and they have a high brand loyalty.

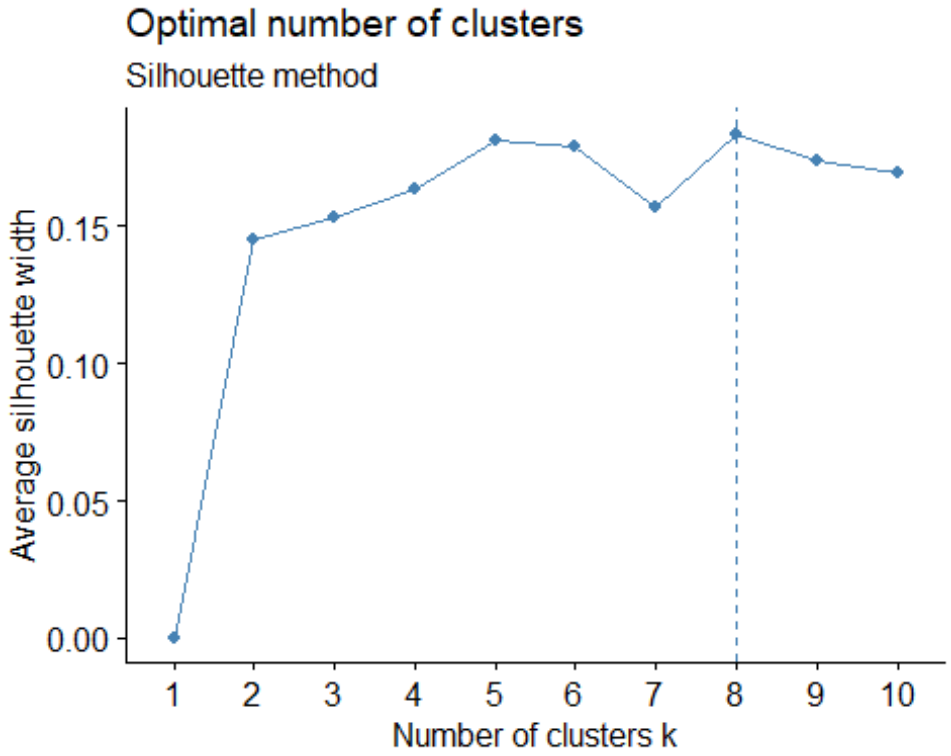
-> cluster2: The behavior of Customers in this cluster is that they purchase products from a single price category(pr.cat 2). They purchase almost similarly all the time (Even if there is price offers or no price offers). We could periodically send the discount offers to them.

-> cluster3: The behavior of Customers in this cluster shows that they purchase products from a single price category(pr.cat 4 and pr.cat 1). They purchase based on the promotions (Pur.Vol.Promo 6) and they doesnt buy when there is no promo. To them as well we could periodically send the discount offers.

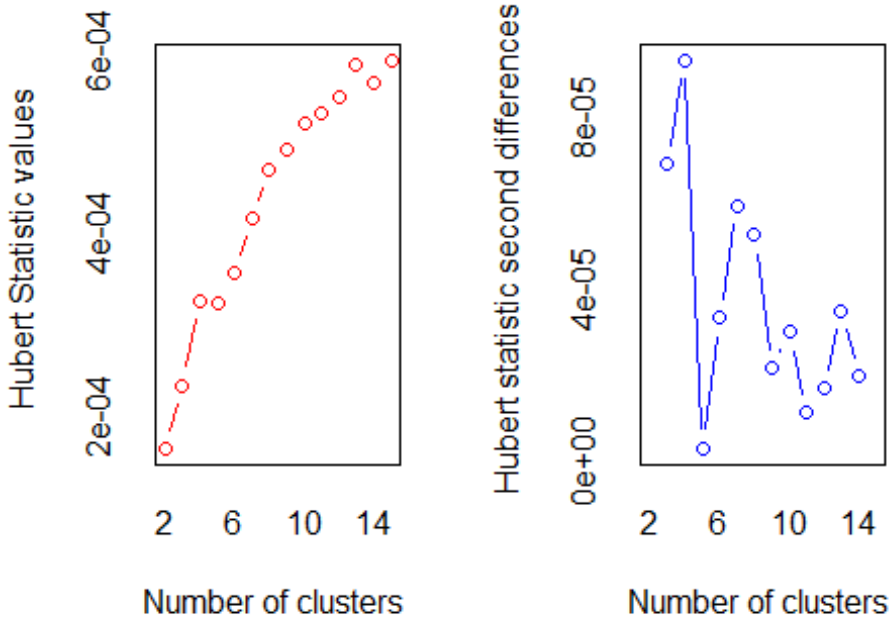
C) Considering variables that describe both purchase bhavior and basis of purchase.

Here we are again scale the required data from BathSoap dataset and the running the fviz_nbcluster() to find the number of clusters, using elbow, silhouette and nbclust.

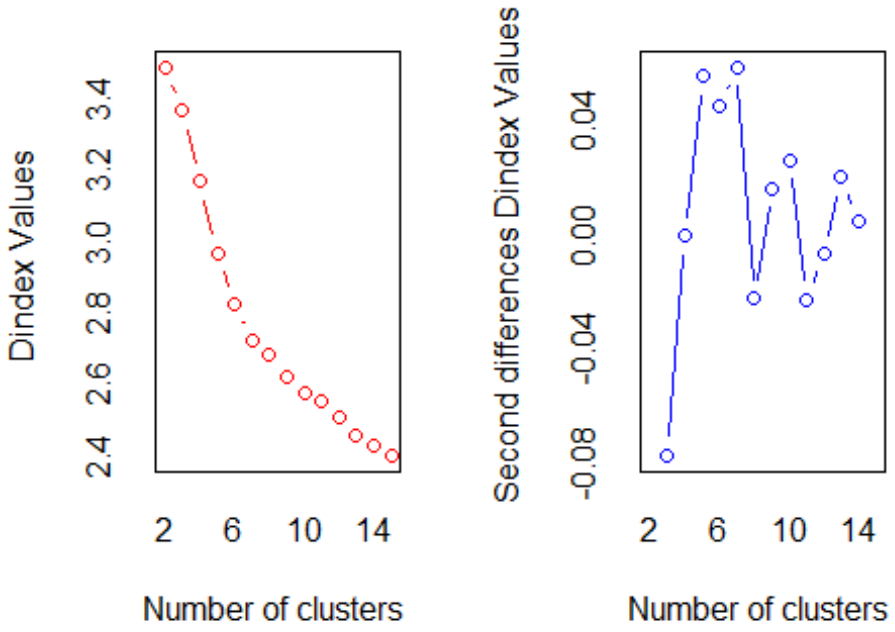
```
Data3<- BathSoap[,c(12:22, 31:35,49)]
scale_Data3 <- as.data.frame(scale(Data3))

fviz_nbclust(scale_Data3, kmeans, method = 'silhouette')+
  labs(subtitle = "Silhouette method")
```




```
NbClust(data = scale_Data3, diss = NULL, distance = "euclidean",
        min.nc = 2, max.nc = 15, method = "kmeans")
```




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




```
## *** : The D index is a graphical method of determining the number of
clusters.
##                 In the plot of D index, we seek a significant knee (the
significant peak in Dindex
##                 second differences plot) that corresponds to a significant
increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 5 proposed 2 as the best number of clusters
## * 1 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 4 proposed 5 as the best number of clusters
## * 7 proposed 7 as the best number of clusters
## * 1 proposed 9 as the best number of clusters
## * 1 proposed 13 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
## * 2 proposed 15 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  7
##
##
## *******************************************************************
```

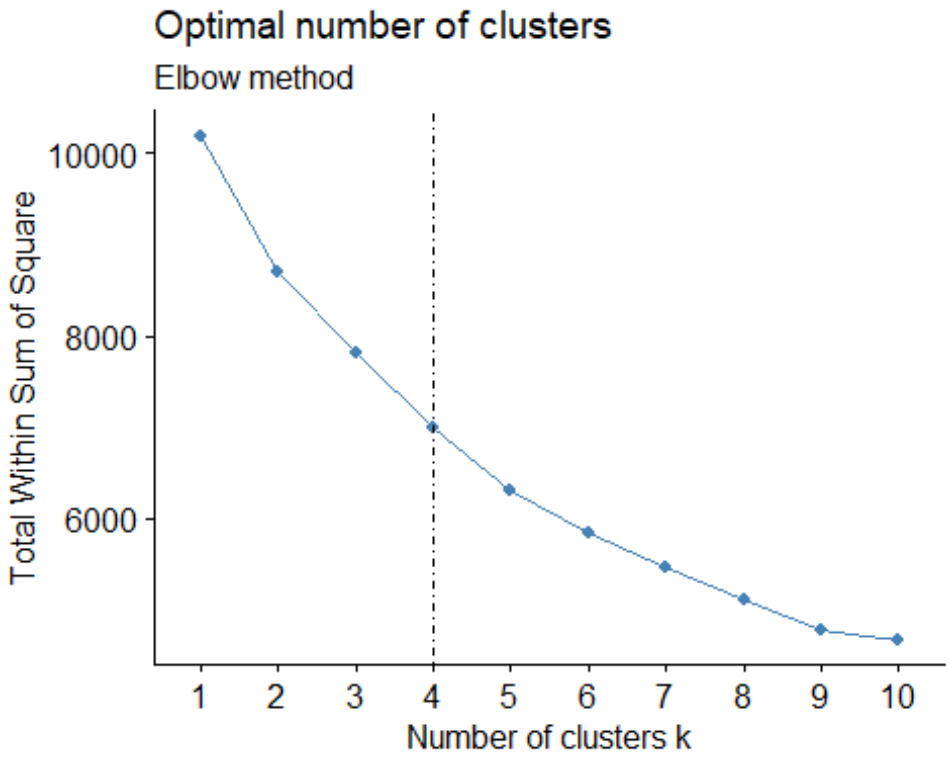
```
## $All.index
##        KL       CH Hartigan      CCC    Scott      Marriot    TrCovW
TraceW
## 2  2.6411 103.6588  49.1774  -5.0087  674.323 7.537337e+35 516064.41
8678.626
## 3  0.3922  80.5454  75.0907 -10.0374 1159.129 7.559385e+35 438221.31
8019.159
## 4  1.0543  85.3380  71.7920  -5.4038 1953.299 3.577042e+35 351520.39
7123.202
## 5  1.2140  89.5107  62.4089   0.7659 2883.610 1.185666e+35 267509.88
6357.411
## 6  1.2473  91.4473  53.1236   6.7295 3694.712 4.418040e+34 222782.41
5753.891
## 7  3.3910  91.7208  22.0594  12.1199 4000.617 3.611618e+34 177408.65
5281.543
## 8  0.6060  84.5509  29.8443  12.6626 4231.666 3.209571e+34 159549.57
5092.118
## 9  1.8107  81.3046  19.4947  15.1296 4370.640 3.222253e+34 139732.70
4847.731
## 10 2.1366  76.6908  12.3648  14.6051 4514.357 3.130752e+34 129535.94
4692.931
## 11 0.4804  71.5832  18.1608  11.5838 4740.052 2.600582e+34 124183.19
4596.598
## 12 0.4411  68.6164  34.2457   9.6146 4938.331 2.223966e+34 115836.98
4459.109
## 13 2.9911  69.2974  14.8636   8.0404 5527.682 9.773869e+33 105001.20
4213.700
## 14 0.5042  66.6162  24.6620   9.0692 5553.660 1.085506e+34  97200.58
4109.639
## 15 0.7516  66.1098 -20.8076  12.0784 6046.782 5.478039e+33  92361.80
3943.668
##     Friedman  Rubin Cindex     DB Silhouette   Duda Pseudot2   Beale
Ratkowsky
## 2   1761.370 1.1733 0.2369 2.3909     0.1450 1.0659 -21.0814 -0.7253
0.2377
## 3   3093.538 1.2698 0.2603 2.1812     0.1380 0.9070  33.2054  1.2017
0.2467
## 4   6831.656 1.4296 0.2364 1.9687     0.1527 1.1858 -48.7323 -1.8370
0.2622
## 5  12497.439 1.6018 0.2290 1.7928     0.1740 0.9034  24.4893  1.2525
0.2627
## 6  15453.372 1.7698 0.2142 1.6172     0.1943 1.4006 -87.2300 -3.3476
0.2590
## 7  15240.394 1.9280 0.1990 1.4900     0.1958 1.3923 -50.9989 -3.2965
0.2567
## 8  15473.038 1.9998 0.1938 1.6021     0.1814 1.6053 -69.3797 -4.4079
0.2460
## 9  15648.100 2.1006 0.1889 1.5823     0.1628 1.3607 -45.8630 -3.0948
0.2378
## 10 15716.237 2.1699 0.1881 1.6945     0.1496 1.5081 -44.4747 -3.9229
```

```
0.2293
## 11 15905.904 2.2153 0.1867 1.6588     0.1492 1.2617 -24.2680 -2.4144
0.2205
## 12 16332.019 2.2836 0.1831 1.7906     0.1377 1.5777 -41.7407 -4.2587
0.2140
## 13 16617.728 2.4166 0.1770 1.6307     0.1543 1.6167 -40.0512 -4.4304
0.2104
## 14 17053.239 2.4778 0.2367 1.7464     0.1406 0.9490   3.3337  0.6249
0.2049
## 15 16972.155 2.5821 0.2522 1.6218     0.1609 1.5808 -41.5190 -4.2444
0.2013
##         Ball Ptbiserial    Frey McClain   Dunn Hubert SDindex Dindex
SDbw
## 2  4339.3130     0.2109 -0.1038  0.8526 0.0564  2e-04  1.5005 3.4923
1.2709
## 3  2673.0530     0.2768 -0.0375  1.2213 0.0470  2e-04  1.5181 3.3708
1.3898
## 4  1780.8004     0.3429  0.3443  1.6394 0.0518  3e-04  1.5522 3.1719
1.0319
## 5  1271.4822     0.3458 -0.0031  2.2644 0.0630  3e-04  1.4010 2.9722
0.9423
## 6   958.9819     0.3789 -0.0271  2.5446 0.0671  4e-04  1.3445 2.8281
0.9080
## 7   754.5062     0.4137  0.2292  2.7313 0.0611  4e-04  1.2731 2.7288
0.8593
## 8   636.5148     0.4131  0.6219  2.9533 0.0611  5e-04  1.5725 2.6880
1.0258
## 9   538.6368     0.3932  4.3828  3.5056 0.0590  5e-04  1.5399 2.6246
0.9527
## 10  469.2931     0.3547  0.0944  4.4040 0.0474  5e-04  1.6382 2.5774
0.9172
## 11  417.8726     0.3553  0.3215  4.4624 0.0474  5e-04  1.7383 2.5562
0.9378
## 12  371.5924     0.3472 -0.0779  4.8982 0.0474  5e-04  1.5859 2.5121
0.8542
## 13  324.1307     0.3616  0.6584  4.8251 0.0483  6e-04  1.6910 2.4613
0.8518
## 14  293.5456     0.3446 -0.7027  5.4806 0.0639  6e-04  1.6067 2.4303
0.7422
## 15  262.9112     0.3617 -1.5016  5.0417 0.0699  6e-04  1.6199 2.4042
0.7469
##
## $All.CriticalValues
##    CritValue_Duda CritValue_PseudoT2 Fvalue_Beale
## 2          0.9018            37.1263       1.0000
## 3          0.8970            37.2035       0.2535
## 4          0.8955            36.3011       1.0000
## 5          0.8877            28.9577       0.2143
## 6          0.8834            40.2673       1.0000
## 7          0.8802            24.6446       1.0000
```

```
## 8          0.8748            26.3330       1.0000
## 9          0.8664            26.6852       1.0000
## 10         0.8521            22.9158       1.0000
## 11         0.8509            20.5045       1.0000
## 12         0.8471            20.5845       1.0000
## 13         0.8405            19.9274       1.0000
## 14         0.8435            11.4998       0.8746
## 15         0.8181            25.1209       1.0000
##
## $Best.nc
##                     KL       CH Hartigan     CCC    Scott      Marriot
TrCovW
## Number_clusters 7.000   2.0000  15.0000  9.0000   5.0000 5.000000e+00
4.00
## Value_Index     3.391 103.6588  45.4696 15.1296 930.3112 1.647514e+35
86700.92
##                    TraceW Friedman   Rubin Cindex   DB Silhouette   Duda
## Number_clusters   7.0000    5.000  7.0000 13.000 7.00     7.0000 2.0000
## Value_Index     282.9227 5665.783 -0.0866  0.177 1.49     0.1958 1.0659
##                  PseudoT2   Beale Ratkowsky    Ball PtBiserial Frey McClain
## Number_clusters   2.0000  2.0000    5.0000    3.00     7.0000    1  2.0000
## Value_Index     -21.0814 -0.7253    0.2627 1666.26     0.4137   NA  0.8526
##                     Dunn Hubert SDindex Dindex    SDbw
## Number_clusters 15.0000      0  7.0000      0 14.0000
## Value_Index      0.0699      0  1.2731      0  0.7422
##
## $Best.partition
##   [1] 1 7 7 4 6 7 1 4 2 5 6 7 6 1 1 7 7 1 6 3 4 4 4 4 1 7 1 1 6 4 4 6 4 4
4 6 6
##  [38] 6 4 1 1 4 4 4 6 1 6 1 4 7 4 1 4 2 4 1 4 6 4 1 7 4 4 1 7 4 7 7 6 1 1
6 4 3
##  [75] 1 5 1 4 2 6 1 7 4 5 3 7 6 2 5 4 5 2 4 5 7 7 4 7 4 1 1 1 6 3 6 7 7 1
7 6 4
## [112] 1 1 3 3 5 7 4 7 7 6 7 1 2 2 4 2 1 1 5 7 7 5 1 4 5 1 7 1 6 6 4 1 4 5
4 4 6
## [149] 7 1 1 1 4 4 1 7 5 4 1 2 6 4 2 7 2 7 2 1 7 1 6 7 1 4 2 1 2 4 4 1 1 1
7 3 7
## [186] 1 6 7 2 3 6 7 1 1 7 7 7 7 5 1 1 6 2 6 4 1 6 2 7 7 1 6 7 1 2 5 1 5 4
7 6 2
## [223] 4 5 1 1 1 7 5 3 4 7 4 2 4 4 4 4 5 7 7 6 1 4 1 4 5 5 6 1 3 1 7 1 1
1 3 2
## [260] 7 5 1 7 1 7 1 1 1 7 5 2 2 1 7 1 3 7 7 1 3 2 6 6 2 7 1 7 1 7 5 7 5 1
6 6 1
## [297] 4 7 1 5 5 1 6 7 5 1 7 1 1 5 5 3 7 5 2 7 1 6 7 5 2 3 2 7 5 4 7 5 1 1
7 1 7
## [334] 7 1 3 7 7 1 1 1 1 6 5 7 3 3 1 7 1 5 5 7 1 5 7 7 7 5 7 5 5 7 5 7 3 3
7 5 6
## [371] 1 5 1 1 4 5 1 5 7 7 5 5 7 5 1 5 7 7 7 7 2 7 5 5 3 5 5 1 5 1 5 7 1 7
1 5 5
## [408] 1 7 7 1 5 5 1 5 3 7 5 1 1 2 1 5 1 5 7 3 5 5 1 2 6 1 5 5 1 5 7 1 5 1
```

```
1 7 5
## [445] 5 1 1 1 1 1 5 1 1 1 1 1 5 7 7 7 3 6 3 5 7 7 1 1 5 5 3 5 7 7 2 2 7 3
5 7 5
## [482] 3 7 5 5 1 5 7 5 7 1 1 5 1 2 5 5 7 7 7 4 7 1 1 5 5 1 1 7 3 5 6 1 7 3
1 5 3
## [519] 5 7 1 1 1 2 1 1 1 1 7 1 3 3 7 1 7 1 4 1 1 1 7 3 7 7 3 1 1 1 7 7 1 5
5 7 7
## [556] 1 1 1 5 5 3 1 7 7 3 3 7 3 1 5 5 7 5 7 2 1 7 5 5 3 3 5 1 3 1 3 7 7 1
7 1 5
## [593] 3 1 1 1 5 2 3 1
```

```
fviz_nbclust(scale_Data3, kmeans, method = 'wss') +
    geom_vline(xintercept = 4, linetype = 4)+
  labs(subtitle = 'Elbow method')
```



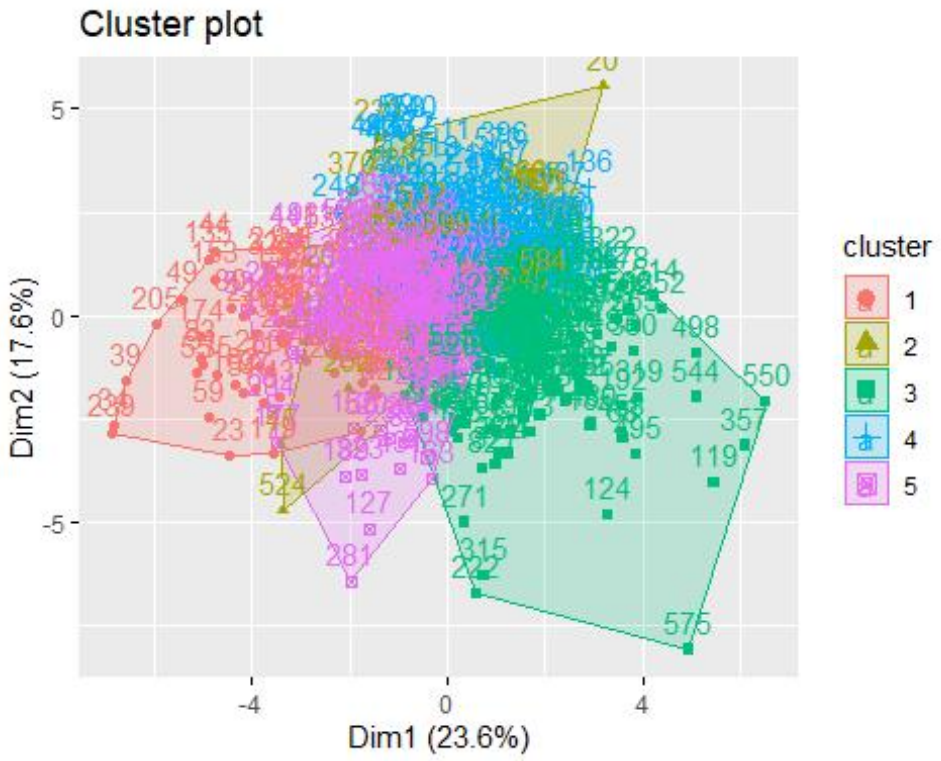
According to the plots, the best number of clusters according to different methods are:

silhouette = 8 Elbow = 4 Nbclust = 5

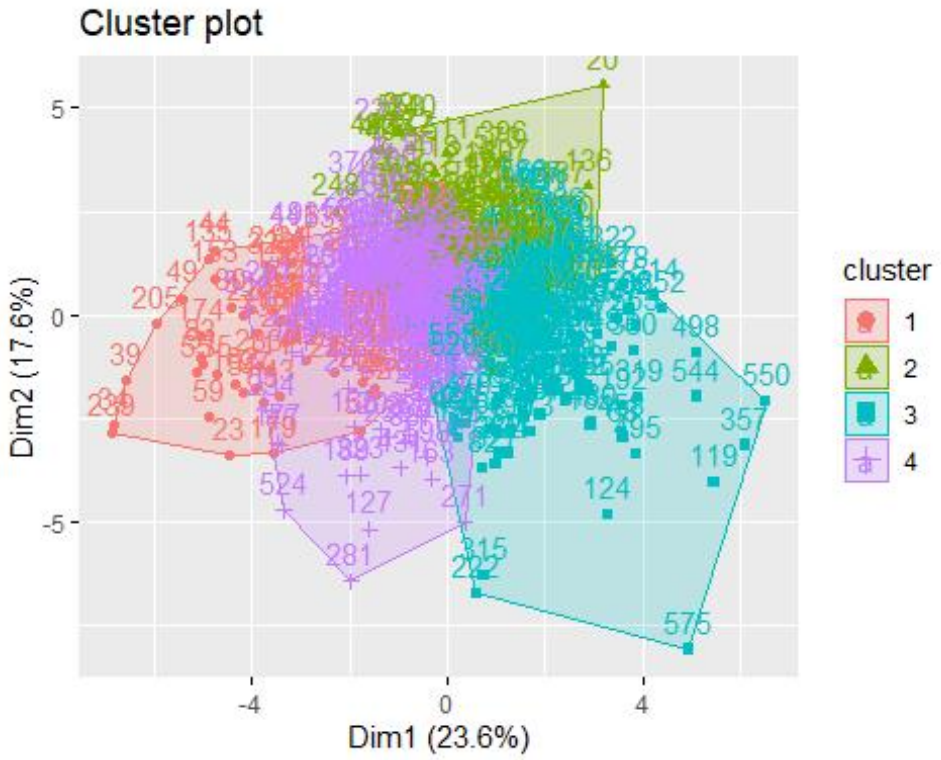
Now, we will consider the value ok = 5 because we don't want to have too many cluster as they might might not capture the realtionship that we want them to show, among the variables.

So we will run kmeans model for k = 5, but will also show the clusters for k = 4 and 8. Then we will store the centers for the model, with best cluster formation (K = 5).

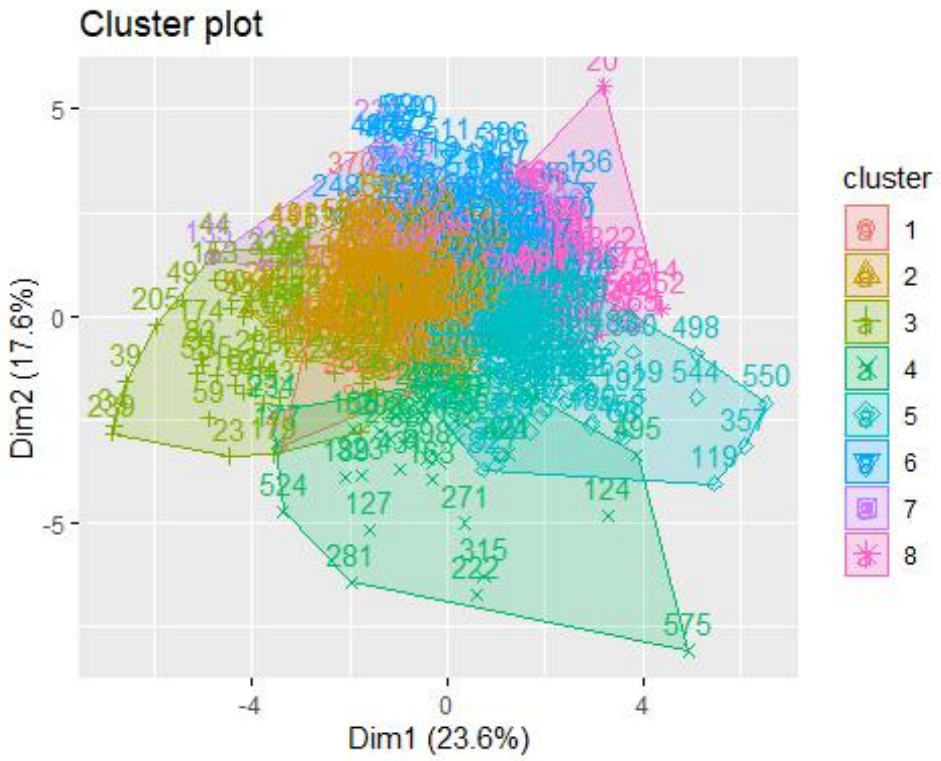
```
Model_Behav_Basis1 <- kmeans(scale_Data3, 5, nstart = 50)
fviz_cluster(Model_Behav_Basis1, scale_Data3)
```




```
Model_Behav_Basis2 <- kmeans(scale_Data3, 4, nstart = 50)
fviz_cluster(Model_Behav_Basis2, scale_Data3)
```



```
Model_Behav_Basis3 <- kmeans(scale_Data3, 8, nstart = 50)
fviz_cluster(Model_Behav_Basis3, scale_Data3)
```



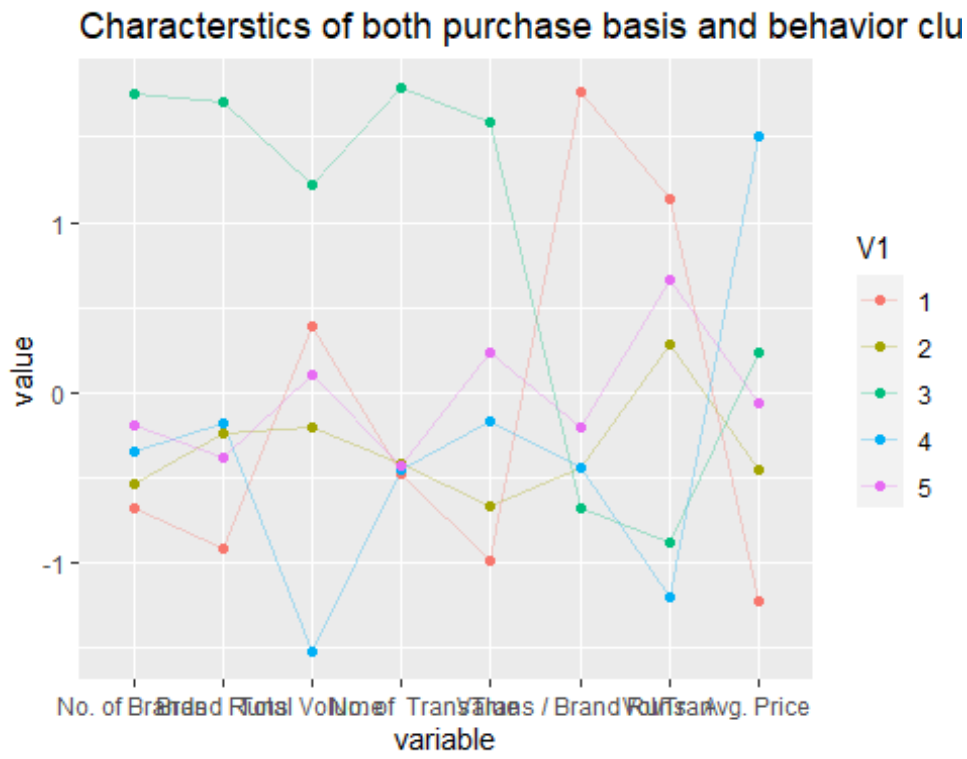

```
result3 <- as.data.frame(cbind(1:nrow(Model_Behav_Basis1$centers),
Model_Behav_Basis1$centers))
result3$V1 <- as.factor(result3$V1)

Model_Behav_Basis1$size

## [1]  64  67 178 109 182
```

As it is clear fro the plot, the model with k = 5 has clear and better cluster formation as compared to k = 4 and 8. So we will consider k = 5 and save it's centers.

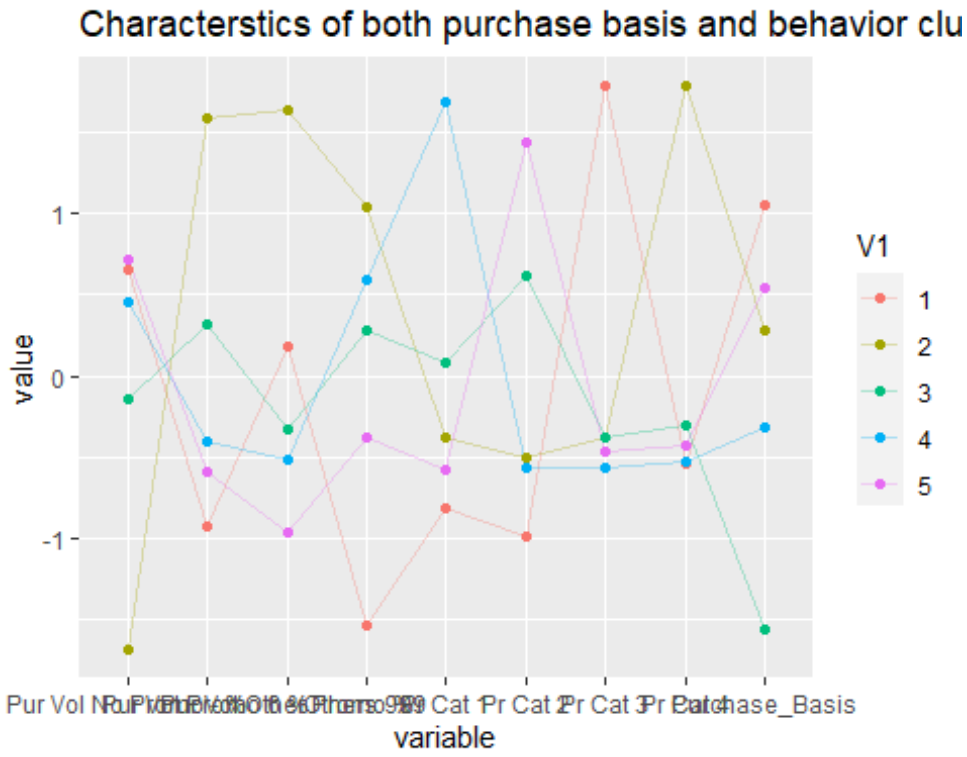
The size of the model is64, 67, 178, 109, 182

Finally we will visualize the variables within the cluster.

```
ggparcoord(result3, columns = 2:9, groupColumn = 1,
           showPoints = TRUE,
           title = "Characterstics of both purchase basis and behavior
cluster (c 2-9).",
           alphaLines = 0.3)
```




```
ggparcoord(result3, columns = 10:18, groupColumn = 1,
           showPoints = TRUE,
           title = "Characterstics of both purchase basis and behavior
cluster (c 10-18).",
           alphaLines = 0.3)
```



Cluster Info:

-> Cluster1: The behavior of Customers in this cluster shows that they purchase products from a single price category(pr.cat 4) and with other999 brands only with the promotion (promo 6).We could periodically send the discount offers to them as well.

-> Cluster2: The cluster has moderate transactions and They buy products from Pr.Cat 2 and also they are brand loyal. They buy products even though with No promos available.

-> Cluster3: The cluster has least number of brands, brand runs,highest transaction brand runs and they buy least from other999. They purchase high volume product from single category pr.cat 3 when other promo is available. Brand loyal.We should periodically send the discount offers when promo is available.

-> Cluster4: They are least brand loyal customers.They are neither least nor highest in other characteristics when compared to other clusters but they have the highest no of transactions and brand runs.

-> Cluster5: This cluster have least total volume of transactions, high Avg.price and highest peak in brand loyality (pr.cat1)

Now, we will compare cluster sizes

```
Model_Purchase_Behav$size

## [1] 334 266

Model_Purchase_Basis$size
```

```
## [1]  67 105 428
```

```
 Model_Behav_Basis1$size
```

```
## [1]  64  67 178 109 182
```

Q- How should K be chosen?

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

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

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

-> The percentages of total purchases should not be considered individually as they increase the inter cluster distances thus decreasing the effectiveness of the clustering. Instead, consider MaxBrCode(Max proportion of purchase) which gives the brand loyalty of the customer.

2. Select which segmentation is best according to you, out of (Demographic, Brand Loyalty and

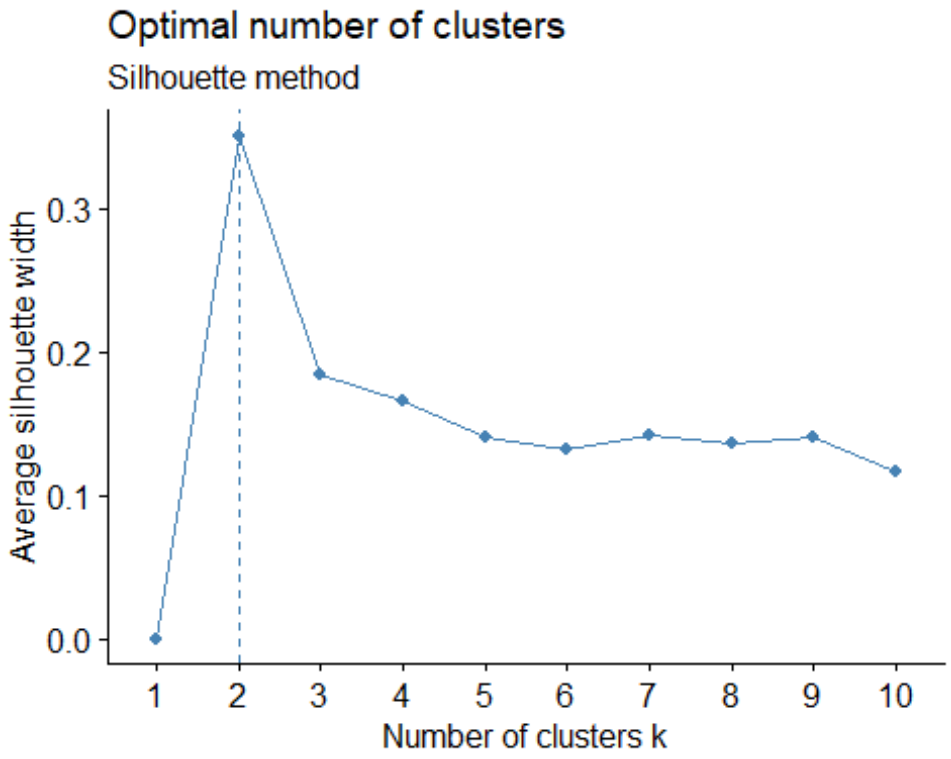
Basis of Purchase)

Now, in order to choose the best segmentation, we first need to add demographic(like (which includes such as gender, age, familial and marital status and education) to first 2 modelling techniques.

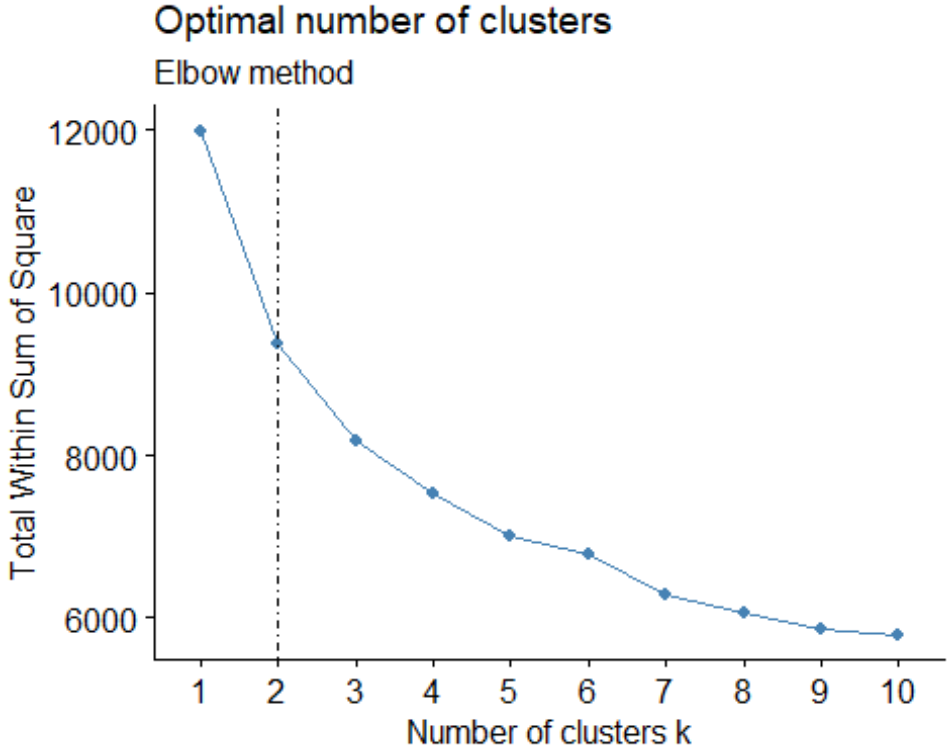
a) Adding demographic to describe the purchase behavior

We will store the necessary variables in Data4, scale the data4 and then use silhouette method to see the best value for k.

```
Data4 <- BathSoap[,c(2:19,31,47)]
scale_Data4 <- as.data.frame(scale(Data4))

fviz_nbclust(scale_Data4, kmeans, method = 'silhouette') +
  labs(subtitle = "Silhouette method")
```



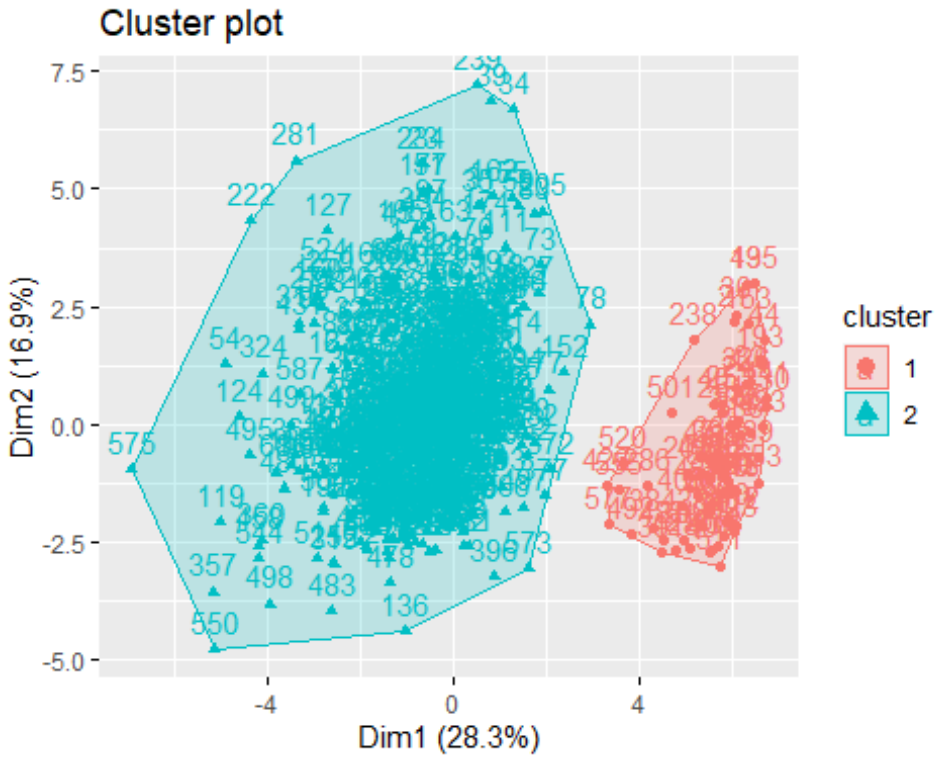

```
fviz_nbclust(scale_Data4, kmeans, method = 'wss') +
    geom_vline(xintercept = 2, linetype = 4)+
  labs(subtitle = 'Elbow method')
```



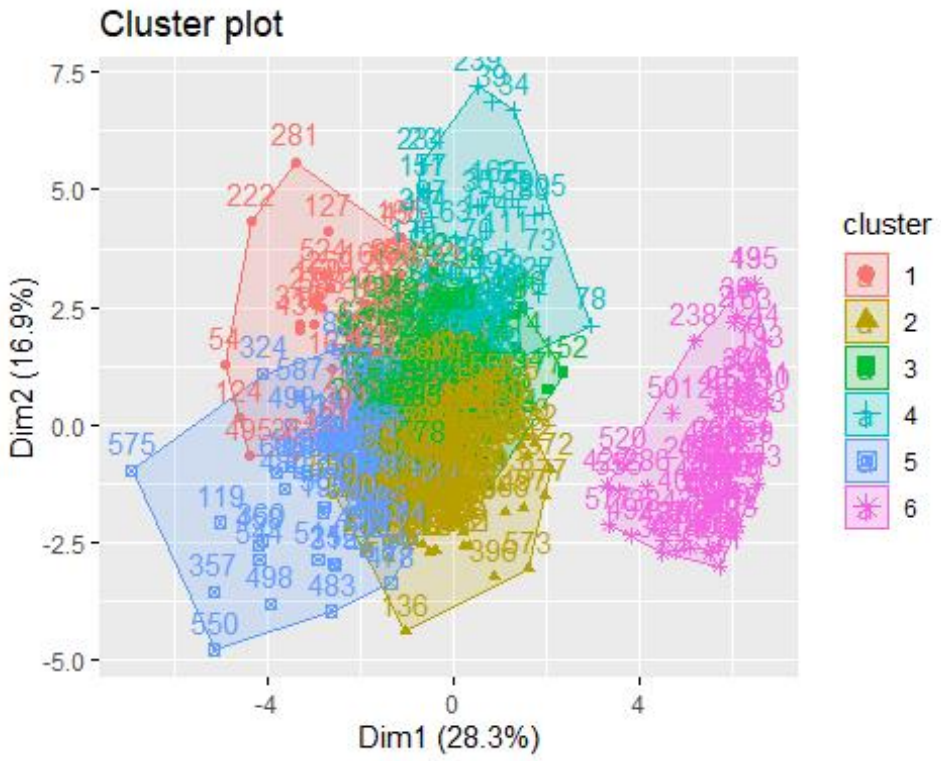
According to our plots, the best number of clusters are: Elbow 2 Silhoutte 2

Here the optimal value is 2, So we will use k = 2 to train our model. Let us visualize the value of k =6 as well (Since elbow method shows bend at that point)

```
Model_Purchase_Behav_demograph1 <- kmeans(scale_Data4, 2, nstart = 50)
fviz_cluster(Model_Purchase_Behav_demograph1, scale_Data4)
```




```
Model_Purchase_Behav_demograph2 <- kmeans(scale_Data4, 6, nstart = 50)
fviz_cluster(Model_Purchase_Behav_demograph2, scale_Data4)
```



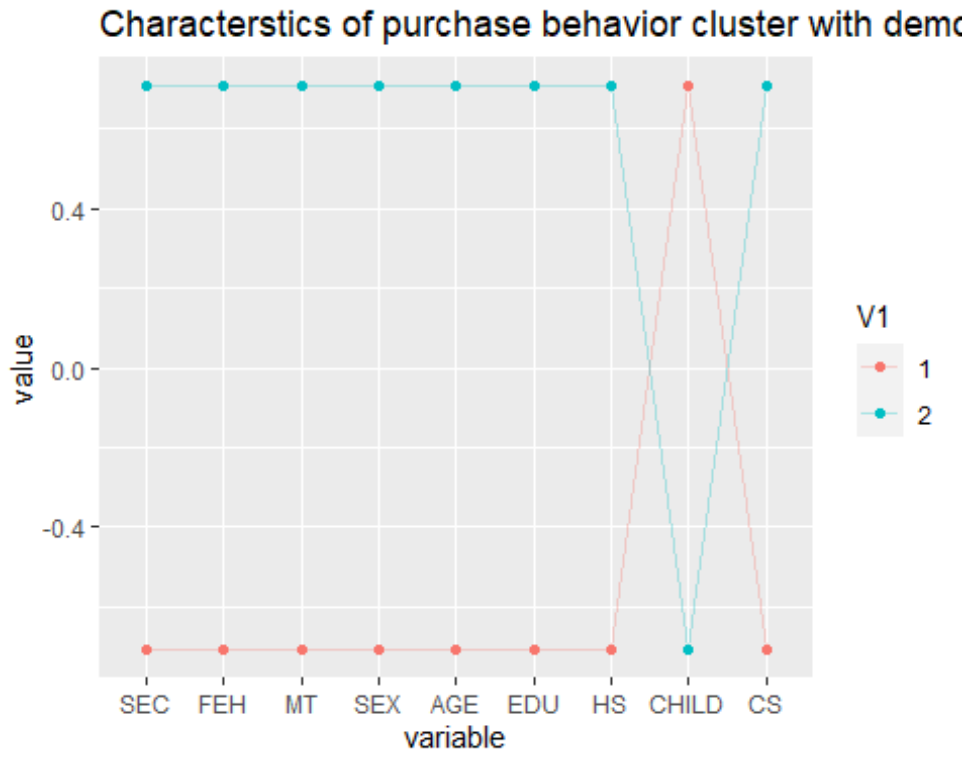

```
result4 <-
as.data.frame(cbind(1:nrow(Model_Purchase_Behav_demograph1$centers),
Model_Purchase_Behav_demograph1$centers))

result4$V1 <- as.factor(result4$V1)
```

The above plot shows us that there are 2 distinct clusters.

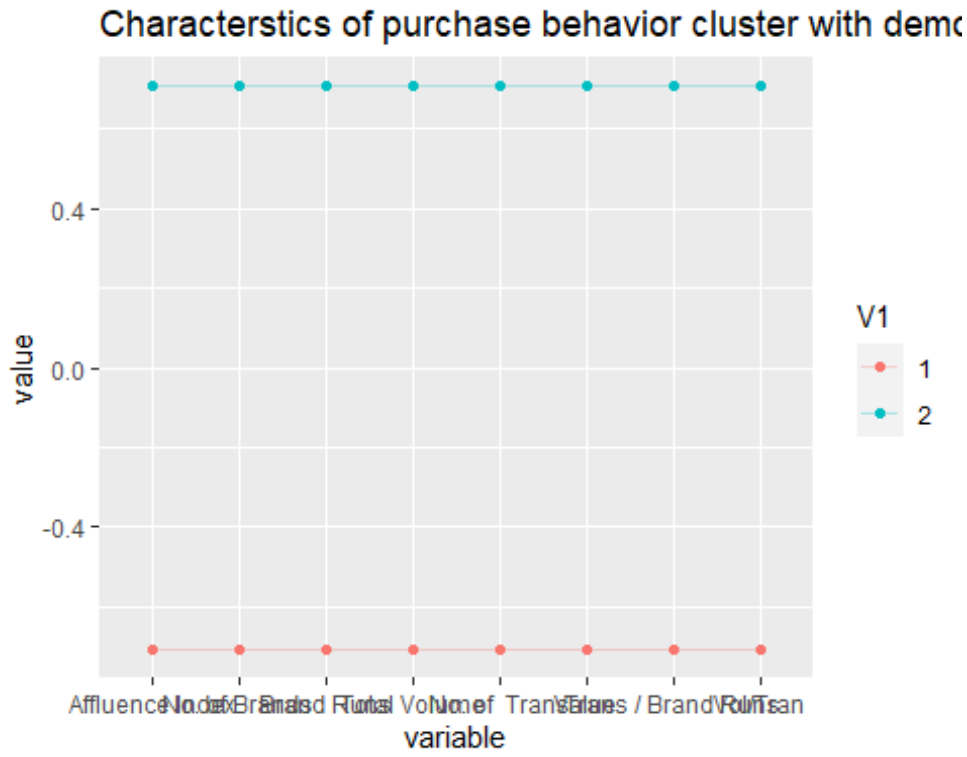
So k = 2 is the best value

Now ultimately, we will visualize this cluster.

```
ggparcoord(result4,
           columns = 2:10,
           groupColumn = 1,
           showPoints = TRUE,
           title = 'Characterstics of purchase behavior cluster with
demographics',
           alphaLines = 0.3)
```




```
#ncol(result4)
ggparcoord(result4,
           columns = 11:18,
           groupColumn = 1,
           showPoints = TRUE,
           title = 'Characterstics of purchase behavior cluster with
demographics',
           alphaLines = 0.3)
```



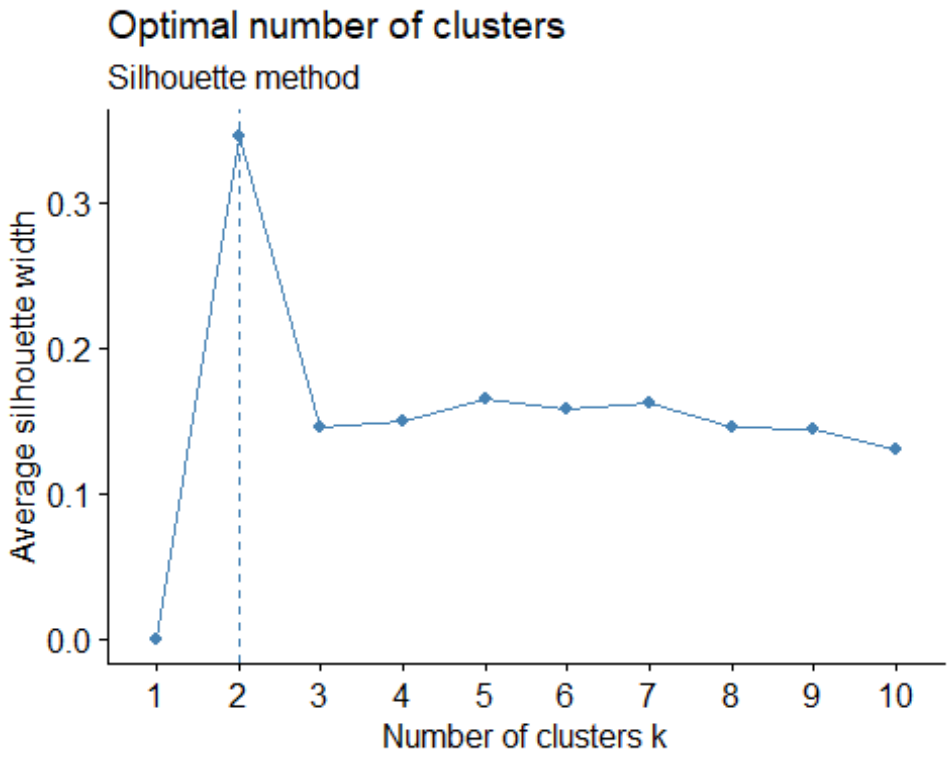
One thing to note. Before running kmean() in the above part, the critieria thta we narrowed to is as follows:

-> Minimum distance within cluster -> Maximum distance between clusters -> Information from centeroid plot of clusters

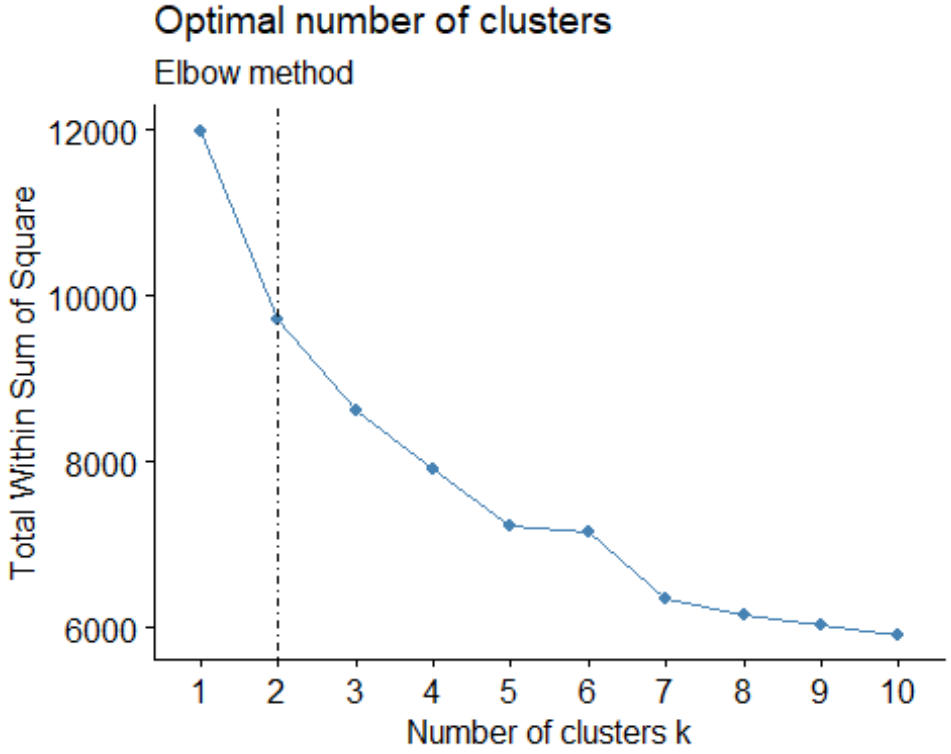
Now, similarly we will add demographic to basis of purchase as well.

We will take the required columns, as a dataframe in Data5, scale it and then find optimal value of k using fviz_nbclust()

```
Data5 <- BathSoap[,c(2:11,20:22,31:35,47,49)]
scale_data5 <- as.data.frame(scale(Data5))

fviz_nbclust(scale_data5, kmeans, method = 'silhouette')+
  labs(subtitle = "Silhouette method")
```



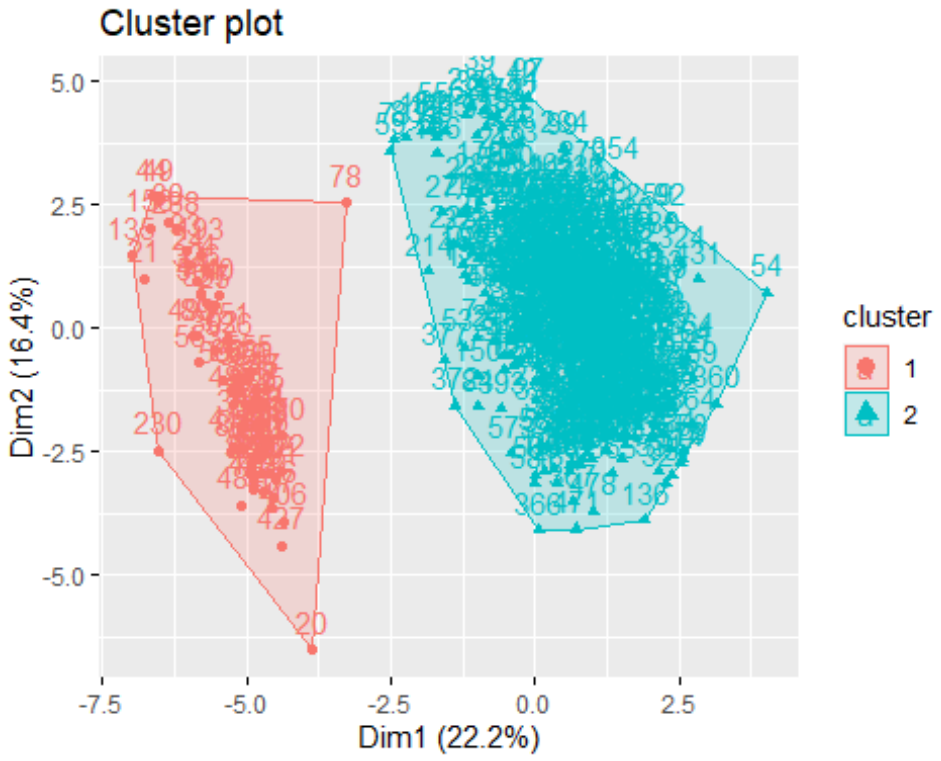

```
fviz_nbclust(scale_data5, kmeans, method = 'wss') +
    geom_vline(xintercept = 2, linetype = 4)+
  labs(subtitle = 'Elbow method')
```



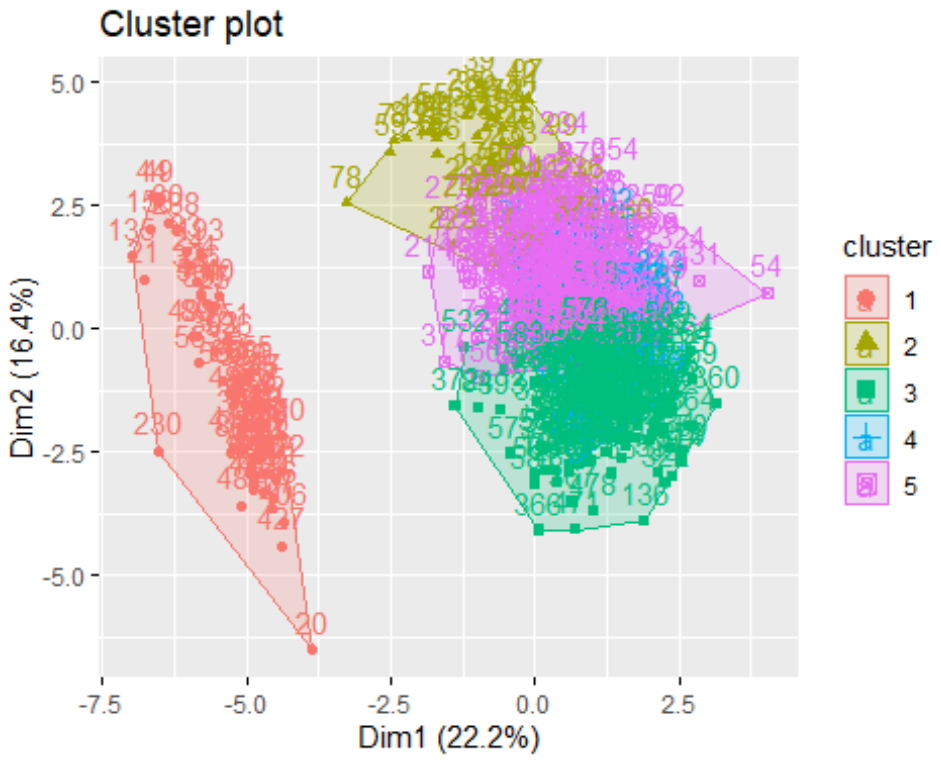
Once again we get the optimal value of k = 2.

We will run the kmeans model with k = 2 and 5 (to look for variations) and store the centers of our desired model in result dataframe

```
Model_Purchase_Basis_Demograph1 <- kmeans(scale_data5, 2, nstart = 50)
fviz_cluster(Model_Purchase_Basis_Demograph1, scale_data5)
```




```
Model_Purchase_Basis_Demograph2 <- kmeans(scale_data5, 5, nstart = 50)
fviz_cluster(Model_Purchase_Basis_Demograph2, scale_data5)
```

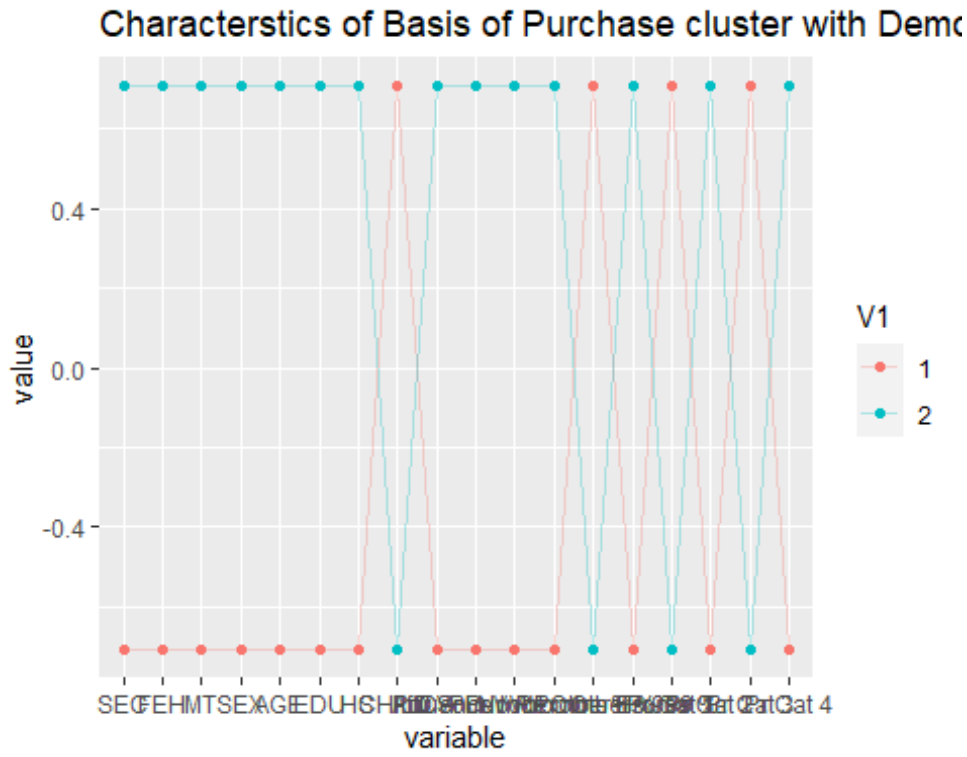



```
result5 <-
as.data.frame(cbind(1:nrow(Model_Purchase_Basis_Demograph1$centers),
Model_Purchase_Basis_Demograph1$centers))

result5$V1 <- as.factor(result5$V1)
```

Since we have similar situation, which we encountered with the Model_Purchase_Behav_demograph1, we will consider the value of k = 2.

Finally we will visualize the plot.

```
ggparcoord(result5,
           columns = 2:19,
           groupColumn = 1,
           showPoints = TRUE,
           title = "Characterstics of Basis of Purchase cluster with
Demographics",
           alphaLines = 0.3)
```

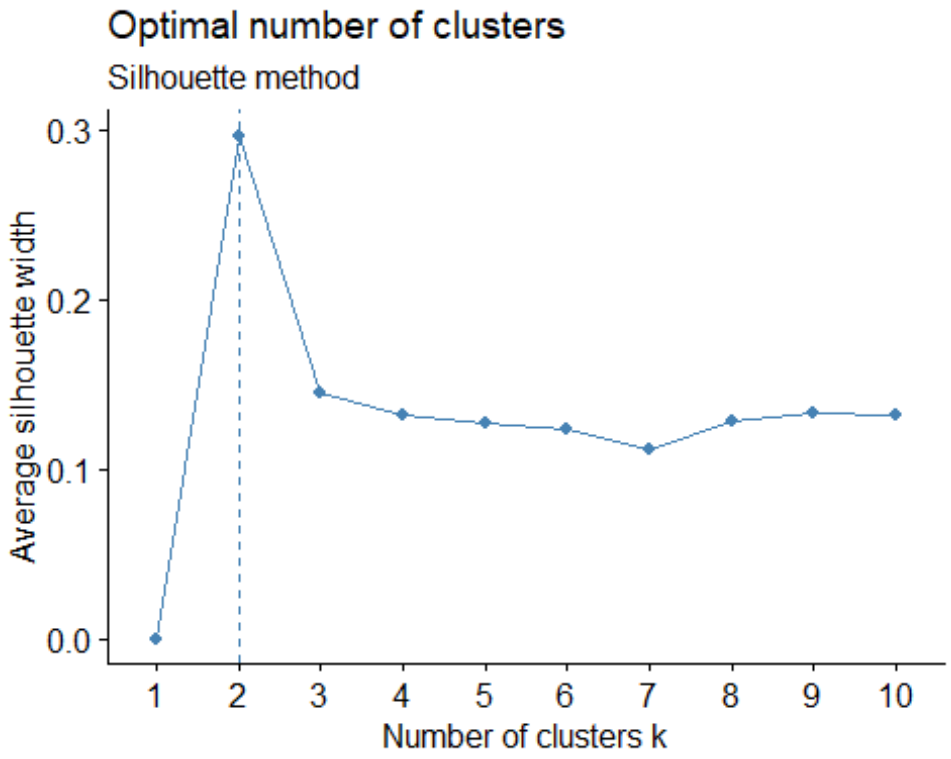


Now we will form cluster using all the variables.

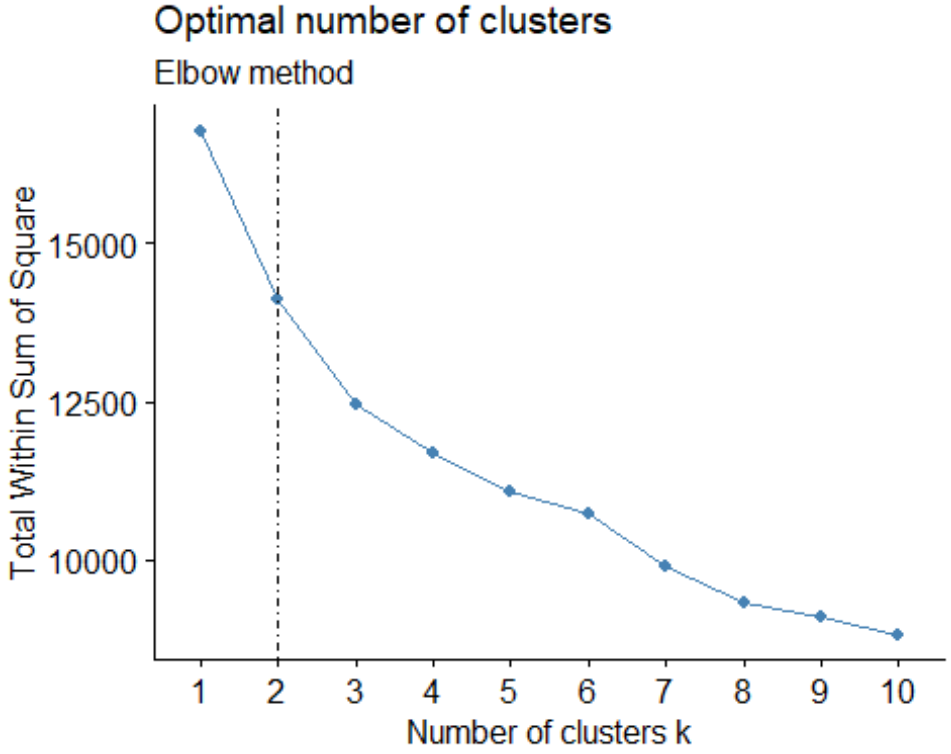
We will take all the variables that we are going to use and put it in Data6

After that we will scale it and find the optimal value of k (using fviz_nbclust()), using silhouette and elbow method

```
Data6 <- BathSoap[,c(2:11,12:22,31:35,47,49)]
scale_Data6 <- as.data.frame(scale(Data6))

fviz_nbclust(scale_Data6, kmeans, method = 'silhouette')+
  labs(subtitle = "Silhouette method")
```

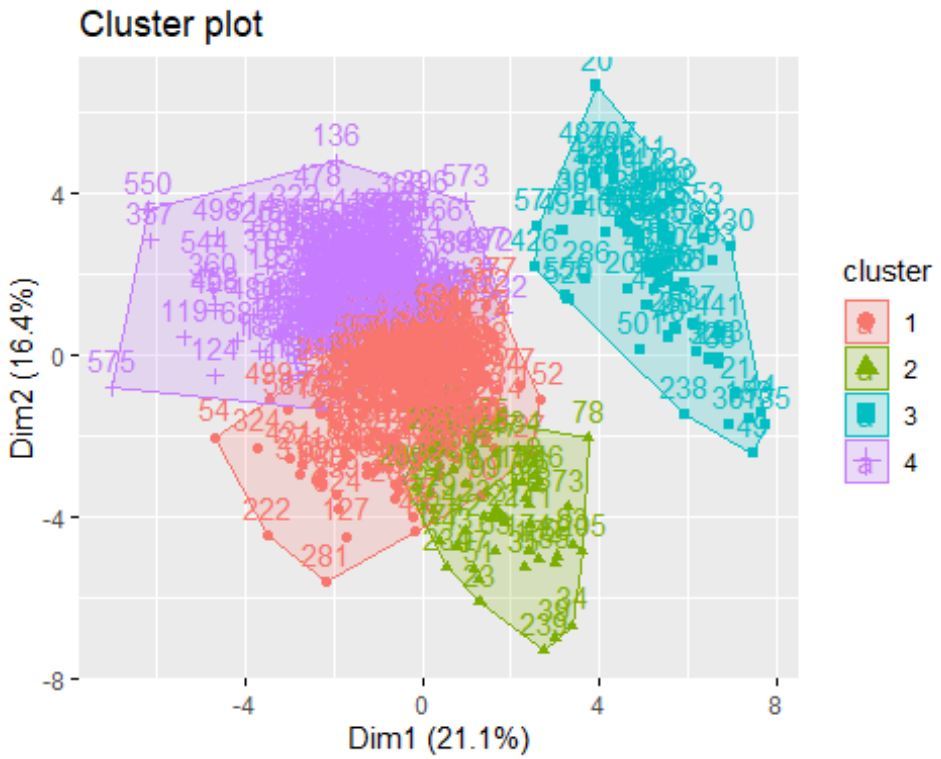



```
fviz_nbclust(scale_Data6, kmeans, method = 'wss') +
    geom_vline(xintercept = 2, linetype = 4)+
  labs(subtitle = 'Elbow method')
```

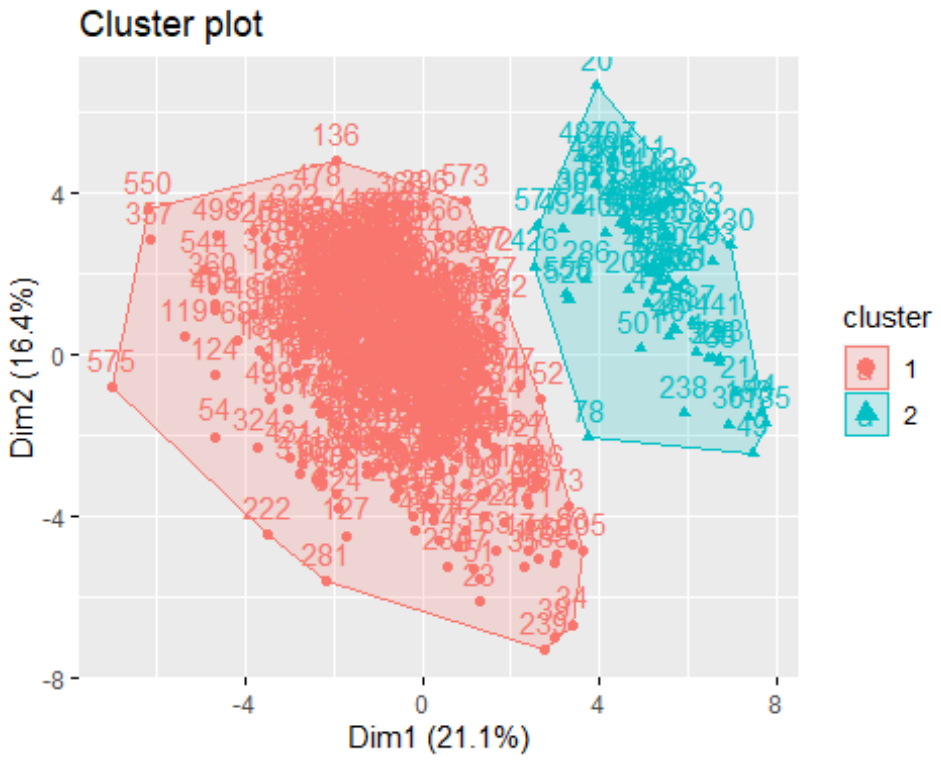


Again, the optimal value shown is 2 but we will train model with both 2 and 4.

```
Model_Behav_Basis_Demograph1 <- kmeans(scale_Data6, 4, nstart = 50)
fviz_cluster(Model_Behav_Basis_Demograph1, scale_Data6)
```



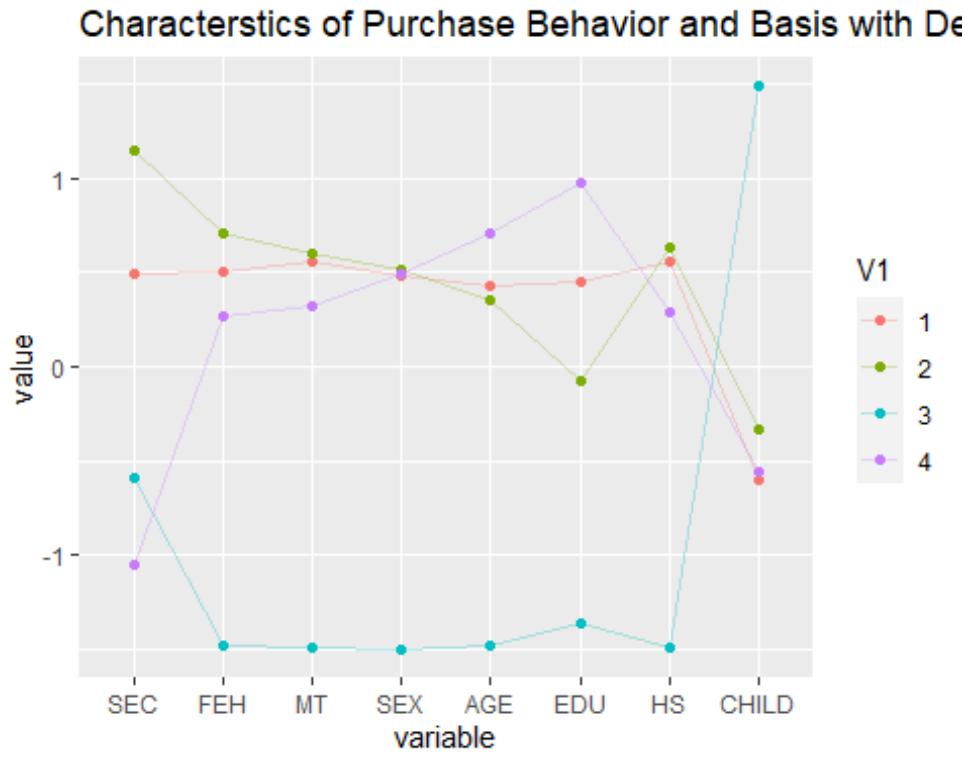

```
Model_Behav_Basis_Demograph2 <- kmeans(scale_Data6, 2, nstart = 50)
fviz_cluster(Model_Behav_Basis_Demograph2, scale_Data6)
```



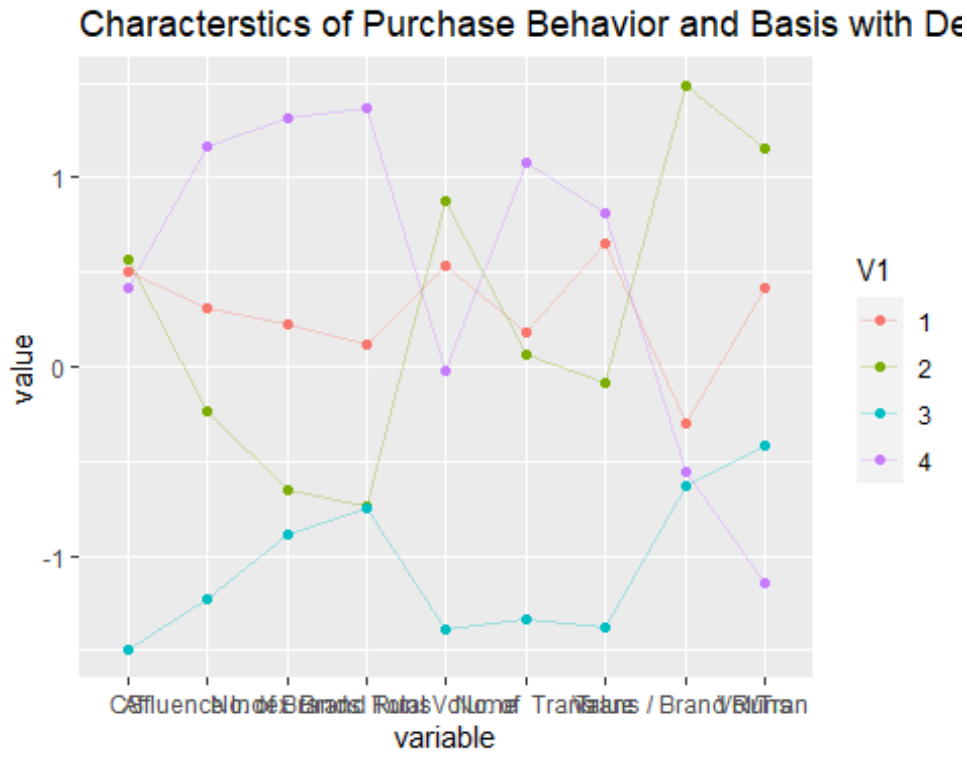

```
result6 <- as.data.frame(cbind(1:nrow(Model_Behav_Basis_Demograph1$centers),
Model_Behav_Basis_Demograph1$centers))

result6$V1 <- as.factor(result6$V1)
```

As shown by the plot, even though k = 2 forms clear and distinct cluster, we will choose k = 4.

```
ggparcoord(result6,
           columns = 2:9, groupColumn = 1,
           showPoints = TRUE,
           title = 'Characterstics of Purchase Behavior and Basis with
Demographics.',
          alphaLines = 0.3)
```




```
ggparcoord(result6,
           columns = 10:18, groupColumn = 1,
           showPoints = TRUE,
           title = 'Characterstics of Purchase Behavior and Basis with
Demographics.',
          alphaLines = 0.3)
```



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

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

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

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

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

We will also display the size of clusters for comparison.

```
Model_Purchase_Behav_demograph1$size
```

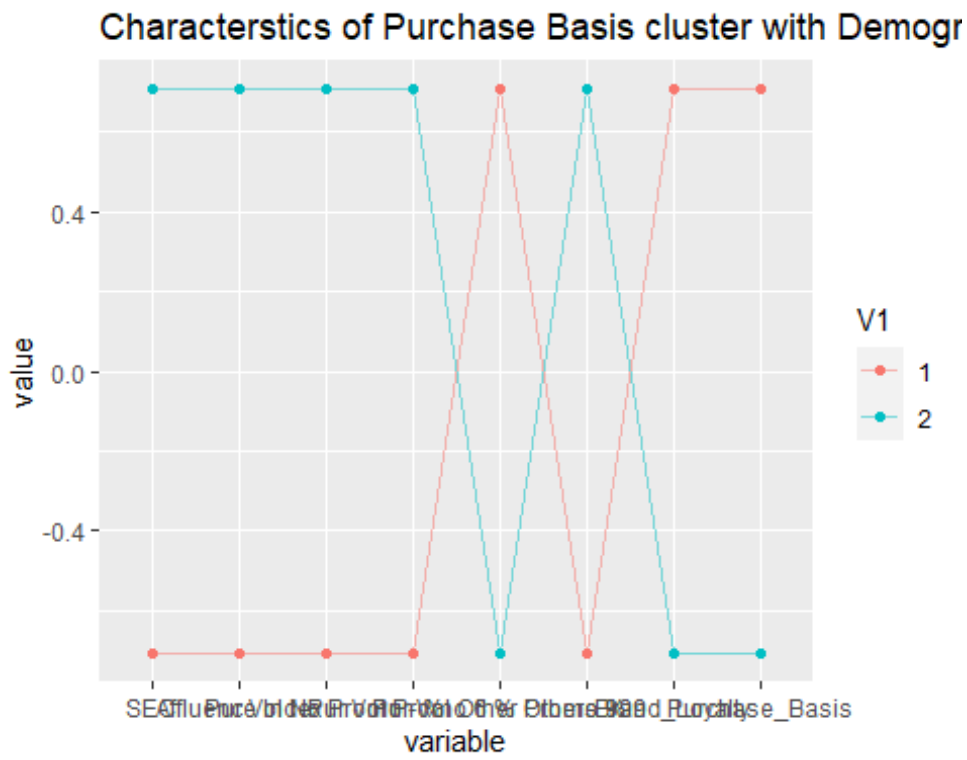
```
## [1]  68 532
```

```
Model_Purchase_Basis_Demograph1$size

## [1]  69 531

Model_Behav_Basis_Demograph1$size

## [1] 256  49  68 227
```

From the above value, it is clear that cluster Model_Purchase_Behav_demograph1 and Model_Purchase_Basis_Demograph1 are almost similar in size, even though Model_Purchase_Basis_Demograph1 has less variables than Model_Purchase_Behav_demograph1.

Due to this, we can say that choosing the cluster with the Purchase Basis with Demographic is the optimum segmentation criteria.

```
ggparcoord(result5, columns = c(2,11:15,20,21),
           groupColumn = 1,
           showPoints = TRUE,
           title = 'Characterstics of Purchase Basis cluster with
Demographics.',
           alphaLines = 0.5)
```



There are a few points that we can derive from the above graph:

1. the customers are buy high quantity of other products and are not loyal to it at all.

2. People in cluster 2 have high socioeconomic and they buy products irrespective of the Promos and stay loyal to it.

3. People with low socioeconomic fall in the cluster 1 and 3 and they buy products with the promo offer and are not at all loyal to the product.

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.

```
# Performing Data Modification so as to display the variables via relevant
plots.



# Converting Binary variables from numeric to factor(i.e. Binary variables)
BathSoap1 <- BathSoap

BathSoap$SEC <- factor(BathSoap$SEC)
BathSoap$FEH <- factor(BathSoap$FEH)
BathSoap$MT <- factor(BathSoap$MT)
BathSoap$SEX <- factor(BathSoap$SEX)
BathSoap$AGE <- factor(BathSoap$AGE)
BathSoap$EDU <- factor(BathSoap$EDU)
BathSoap$HS <- factor(BathSoap$HS)
BathSoap$CHILD <- factor(BathSoap$CHILD)
BathSoap$CS <- factor(BathSoap$CS)
BathSoap$`Affluence Index` <- factor(BathSoap$`Affluence Index`)

# Converting distinct number variables from numeric to integer
BathSoap$`No. of Brands`<- as.integer(BathSoap$`No. of Brands`)
BathSoap$`Brand Runs`<- as.integer(BathSoap$`Brand Runs`)
BathSoap$`Total Volume`<- as.integer(BathSoap$`Total Volume`)
BathSoap$`No. of  Trans`<- as.integer(BathSoap$`No. of  Trans`)


# Converting percentages in character to floating numericals
BathSoap$`Pur Vol No Promo - %`<-  as.numeric(str_replace(BathSoap$`Pur Vol
No Promo - %`,"%",""))/100
BathSoap$`Pur Vol Promo 6 %`<-  as.numeric(str_replace(BathSoap$`Pur Vol
Promo 6 %`,"%",""))/100
BathSoap$`Pur Vol Other Promo %`<-  as.numeric(str_replace(BathSoap$`Pur Vol
Other Promo %`,"%",""))/100
BathSoap$`Br. Cd. 24` <-  as.numeric(str_replace(BathSoap$`Br. Cd.
24`,"%",""))/100
BathSoap$`Br. Cd. 57, 144`<-  as.numeric(str_replace(BathSoap$`Br. Cd. 57,
```

```
144`,"%",""))/100
BathSoap$`Br. Cd. 55` <-  as.numeric(str_replace(BathSoap$`Br. Cd.
55`,"%",""))/100
BathSoap$`Br. Cd. 272` <-  as.numeric(str_replace(BathSoap$`Br. Cd.
272`,"%",""))/100
BathSoap$`Br. Cd. 286` <-  as.numeric(str_replace(BathSoap$`Br. Cd.
286`,"%",""))/100
BathSoap$`Br. Cd. 481` <-  as.numeric(str_replace(BathSoap$`Br. Cd.
481`,"%",""))/100
BathSoap$`Br. Cd. 352` <-  as.numeric(str_replace(BathSoap$`Br. Cd.
352`,"%",""))/100
BathSoap$`Br. Cd. 5` <-  as.numeric(str_replace(BathSoap$`Br. Cd.
5`,"%",""))/100
BathSoap$`Others 999` <-  as.numeric(str_replace(BathSoap$`Others
999`,"%",""))/100
BathSoap$`Pr Cat 1` <-  as.numeric(str_replace(BathSoap$`Pr Cat
1`,"%",""))/100
BathSoap$`Pr Cat 2` <-  as.numeric(str_replace(BathSoap$`Pr Cat
2`,"%",""))/100
BathSoap$`Pr Cat 3` <-  as.numeric(str_replace(BathSoap$`Pr Cat
3`,"%",""))/100
BathSoap$`Pr Cat 4` <-  as.numeric(str_replace(BathSoap$`Pr Cat
4`,"%",""))/100
BathSoap$`PropCat 5` <-  as.numeric(str_replace(BathSoap$`PropCat
5`,"%",""))/100
BathSoap$`PropCat 6` <-  as.numeric(str_replace(BathSoap$`PropCat
6`,"%",""))/100
BathSoap$`PropCat 7` <-  as.numeric(str_replace(BathSoap$`PropCat
7`,"%",""))/100
BathSoap$`PropCat 8` <-  as.numeric(str_replace(BathSoap$`PropCat
8`,"%",""))/100
BathSoap$`PropCat 9` <-  as.numeric(str_replace(BathSoap$`PropCat
9`,"%",""))/100
BathSoap$`PropCat 10`<-  as.numeric(str_replace(BathSoap$`PropCat
10`,"%",""))/100
BathSoap$`PropCat 11`<-  as.numeric(str_replace(BathSoap$`PropCat
11`,"%",""))/100
BathSoap$`PropCat 12`<-  as.numeric(str_replace(BathSoap$`PropCat
12`,"%",""))/100
BathSoap$`PropCat 13`<-  as.numeric(str_replace(BathSoap$`PropCat
13`,"%",""))/100
BathSoap$`PropCat 14`<-  as.numeric(str_replace(BathSoap$`PropCat
14`,"%",""))/100
BathSoap$`PropCat 15`<-  as.numeric(str_replace(BathSoap$`PropCat
15`,"%",""))/100

# Finding the total null values

sum(is.na(BathSoap))
```

```
## [1] 0

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

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

```
0.40
## 5          0        0.00           0      0.00      0.807     0.00
0.05
## 6          0        0.00           0      0.00      0.857     0.22
0.45
##   Pr.Cat.3 Pr.Cat.4 PropCat.5 PropCat.6 PropCat.7 PropCat.8 PropCat.9
## 1     0.13     0.07      0.50      0.00      0.00      0.00      0.00
## 2     0.09     0.06      0.46      0.35      0.03      0.02      0.01
## 3     0.56     0.00      0.24      0.12      0.03      0.01      0.01
## 4     0.60     0.00      0.40      0.00      0.00      0.00      0.00
## 5     0.14     0.81      0.81      0.00      0.00      0.05      0.00
## 6     0.07     0.27      0.49      0.10      0.00      0.01      0.07
##   PropCat.10 PropCat.11 PropCat.12 PropCat.13 PropCat.14 PropCat.15
## 1          0       0.00       0.03          0       0.13       0.34
## 2          0       0.06       0.00          0       0.08       0.00
## 3          0       0.00       0.02          0       0.56       0.00
## 4          0       0.00       0.00          0       0.60       0.00
## 5          0       0.00       0.00          0       0.14       0.00
## 6          0       0.00       0.00          0       0.07       0.27
##   Brand_Loyalty Purchase_Basis_no Purchase_Basis
## 1            38                 1            50
## 2            14                 1            46
## 3            55                10            56
## 4            60                10            60
## 5            14                 1            81
## 6             8                 1            49
```

```
# Counting the total number of zero values in the categorical data.

colSums(is.na(BathSoap))
```

```
##             Member.id                   SEC                   FEH
##                     0                     0                     0
##                    MT                   SEX                   AGE
##                     0                    68                     0
##                   EDU                    HS                 CHILD
##                    73                    68                     0
##                    CS       Affluence.Index         No..of.Brands
##                    99                     0                     0
##            Brand.Runs          Total.Volume         No..of..Trans
##                     0                     0                     0
##                 Value    Trans...Brand.Runs              Vol.Tran
##                     0                     0                     0
##            Avg..Price  Pur.Vol.No.Promo....     Pur.Vol.Promo.6..
##                     0                     0                     0
## Pur.Vol.Other.Promo..       Br..Cd..57..144           Br..Cd..55
##                     0                     0                     0
##           Br..Cd..272           Br..Cd..286           Br..Cd..24
##                     0                     0                     0
##           Br..Cd..481           Br..Cd..352            Br..Cd..5
```

```
##                     0                     0                     0
##            Others.999             Pr.Cat.1             Pr.Cat.2
##                     0                     0                     0
##             Pr.Cat.3             Pr.Cat.4             PropCat.5
##                     0                     0                     0
##             PropCat.6             PropCat.7             PropCat.8
##                     0                     0                     0
##             PropCat.9            PropCat.10            PropCat.11
##                     0                     0                     0
##            PropCat.12            PropCat.13            PropCat.14
##                     0                     0                     0
##            PropCat.15        Brand_Loyalty    Purchase_Basis_no
##                     0                     0                     0
##        Purchase_Basis
##                     0
```

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

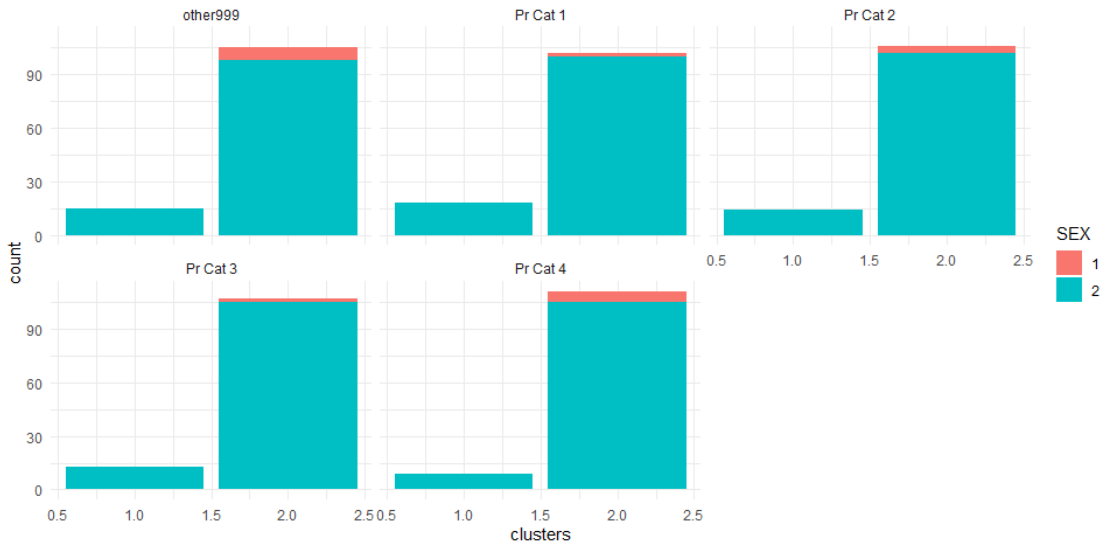
```
## [1] "SEX" "EDU" "HS"  "CS"
```

```
# Imputing Zero insignificant values in categorical variables with their
respective variable mode.

BathSoap$MT <- impute(BathSoap$MT, mode)
BathSoap$EDU <- impute(BathSoap$EDU, mode)
BathSoap$HS <- impute(BathSoap$HS, mode)
BathSoap$CS <- impute(BathSoap$CS, mode)
BathSoap$SEX <- impute(BathSoap$SEX, mode)

Data_final <- BathSoap[,23:31]

BathSoap$Loyality <- as.numeric(apply(Data_final,1,which.max))

Data_final1 <- BathSoap[,c(2:11,19,20:22,31:35,47,48,50)]

Data_final1$clusters <- Model_Purchase_Basis_Demograph1$cluster
head(Data_final1)
```

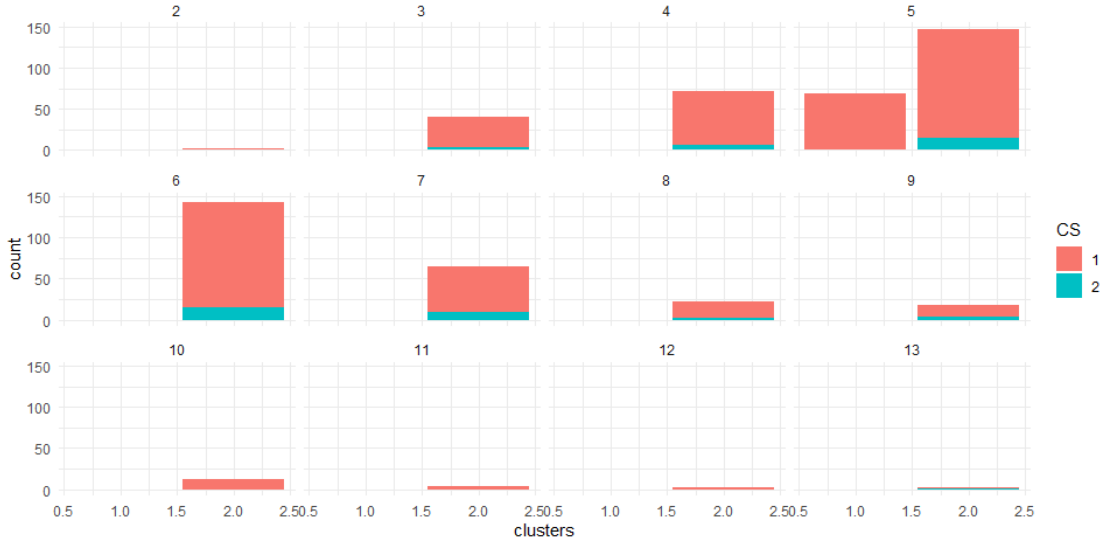
```
##   SEC FEH MT SEX AGE EDU HS CHILD CS Affluence.Index Avg..Price
## 1   4   3 10   1   4   4  2     4  1               2      10.19
## 2   3   2 10   2   2   4  4     2  1              19      12.03
## 3   2   3 10   2   4   5  6     4  1              23       8.44
## 4   4   0  0   2   4   5  4     5  1               0       7.60
## 5   4   1 10   2   3   4  4     3  1              10       7.12
## 6   4   3 10   2   3   4  5     2  1              13       9.38
##   Pur.Vol.No.Promo.... Pur.Vol.Promo.6.. Pur.Vol.Other.Promo.. Others.999
## 1                 1.00              0.00                  0.00      0.492
## 2                 0.89              0.10                  0.02      0.699
## 3                 0.94              0.02                  0.04      0.379
## 4                 1.00              0.00                  0.00      0.000
```

```
## 5                 0.61              0.14                   0.24      0.807
## 6                 1.00              0.00                   0.00      0.857
##   Pr.Cat.1 Pr.Cat.2 Pr.Cat.3 Pr.Cat.4 Brand_Loyalty Purchase_Basis_no
Loyality
## 1     0.23     0.56     0.13     0.07            38                  1
9
## 2     0.29     0.55     0.09     0.06            14                  1
9
## 3     0.12     0.32     0.56     0.00            55                 10
2
## 4     0.00     0.40     0.60     0.00            60                 10
2
## 5     0.00     0.05     0.14     0.81            14                  1
9
## 6     0.22     0.45     0.07     0.27             8                  1
9
##   clusters
## 1        2
## 2        2
## 3        2
## 4        1
## 5        2
## 6        2
```

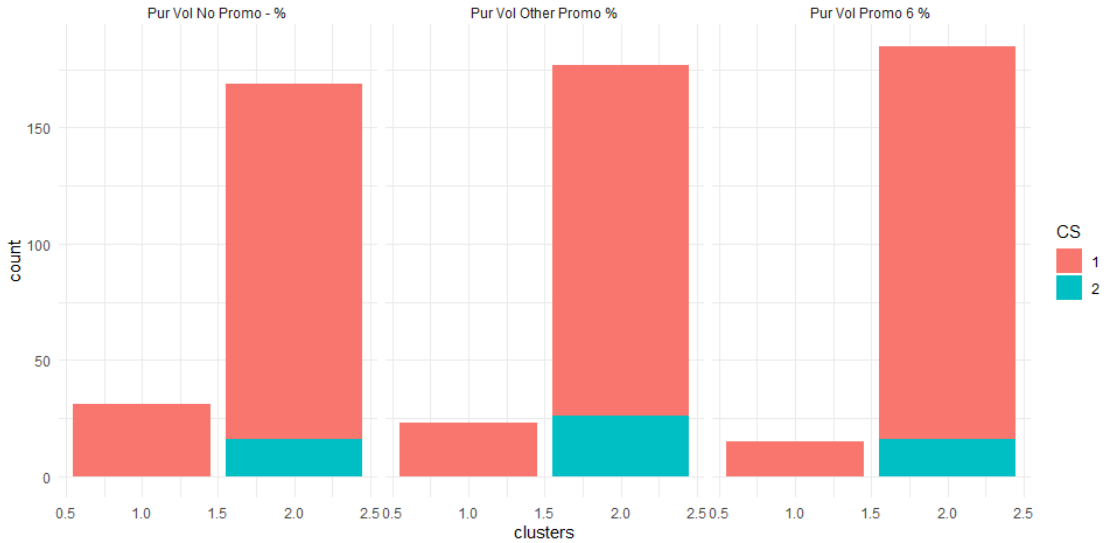
```
ggplot(Data_final1) +
  aes(x =clusters,fill= SEX) +
  geom_bar() +
  scale_fill_hue() +
  theme_minimal() +
  facet_wrap(vars(c("Pr Cat 1","Pr Cat 2", "Pr Cat 3","Pr Cat
4","other999")))
```



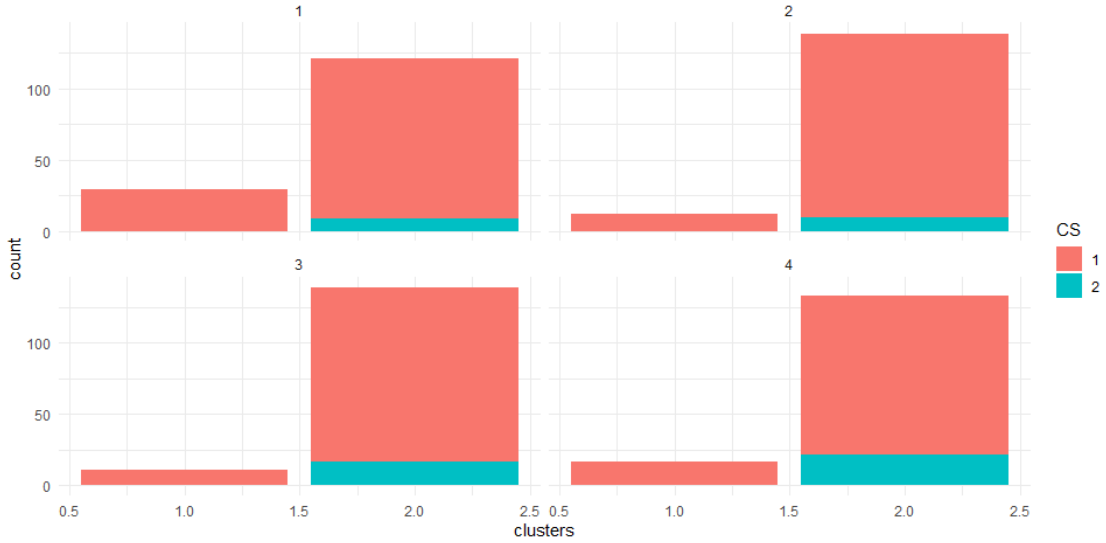
```
ggplot(Data_final1) +
  aes(x =clusters,fill= CS) +
  geom_bar() +
  scale_fill_hue() +
  theme_minimal() +
  facet_wrap(vars(c(HS)))
```



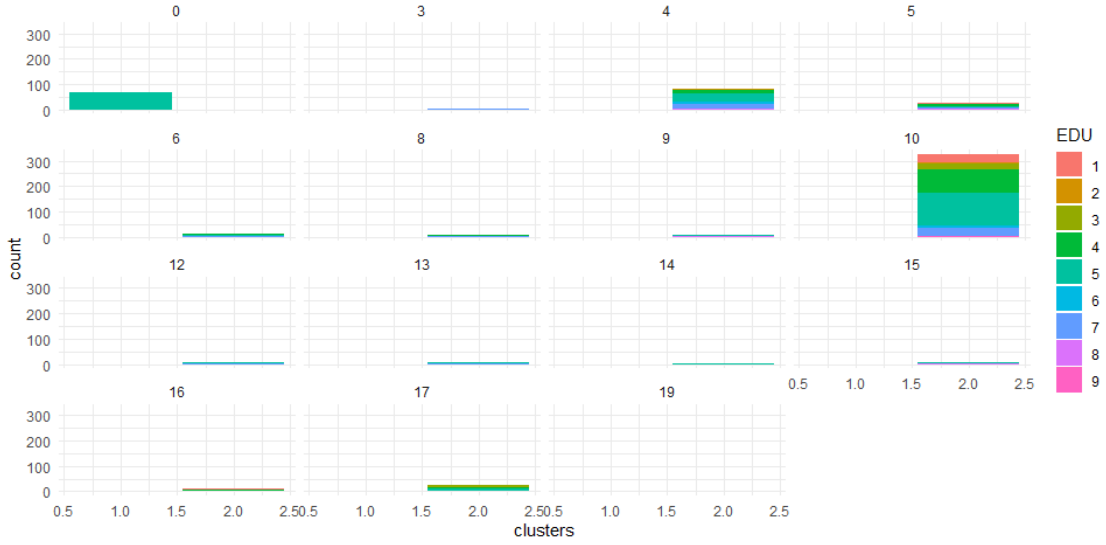

```
ggplot(Data_final1) +
  aes(x =clusters,fill= CS) +
  geom_bar() +
  scale_fill_hue() +
  theme_minimal() +
  facet_wrap(vars(c("Pur Vol No Promo - %","Pur Vol Promo 6 %","Pur Vol Other Promo %")))
```



```
ggplot(Data_final1) +
  aes(x =clusters,fill= CS) +
  geom_bar() +
  scale_fill_hue() +
  theme_minimal() +
  facet_wrap(vars(c(SEC)))
```

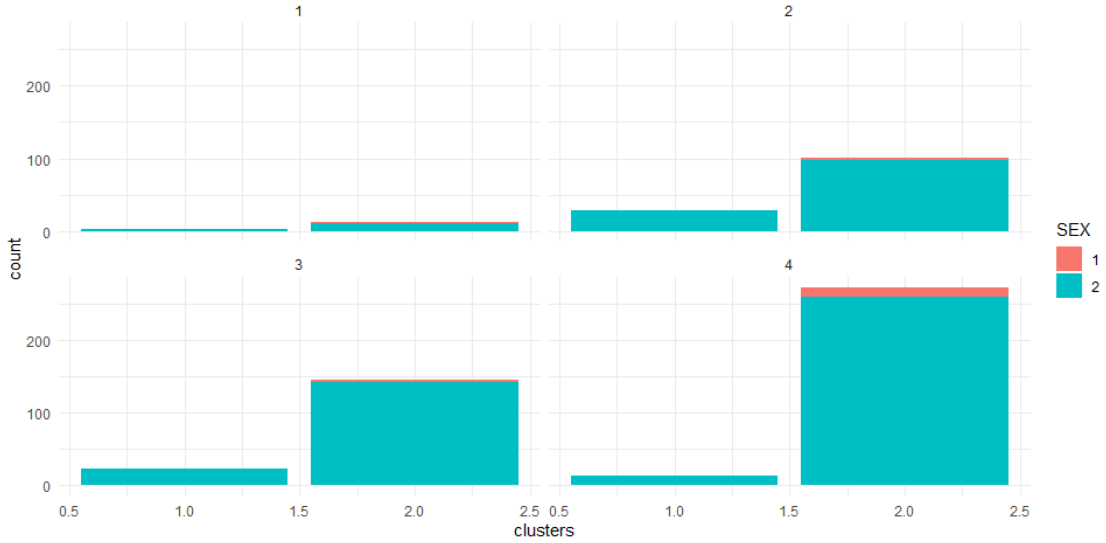



```
ggplot(Data_final1) +
  aes(x =clusters,fill= EDU) +
  geom_bar() +
  scale_fill_hue() +
  theme_minimal() +
  facet_wrap(vars(MT))
```

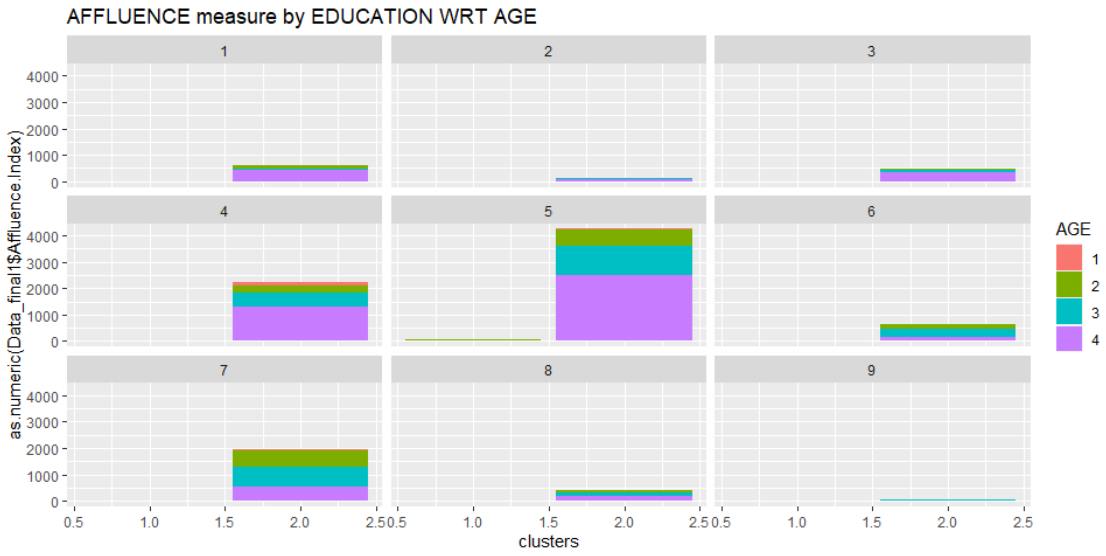



```
ggplot(Data_final1) +
  aes(x =clusters,fill= SEX) +
```

```
  geom_bar() +
  scale_fill_hue() +
  theme_minimal() +
  facet_wrap(vars(AGE))
```




```
ggplot(Data_final1, aes(x =clusters,
y=as.numeric(Data_final1$Affluence.Index), fill= AGE)) +  geom_bar(stat =
'identity') + facet_wrap(~EDU) + ggtitle("AFFLUENCE measure by EDUCATION WRT
AGE")
```



Message Conveyed by the plots:

-> Since most customers from cluster 4 have access to TV/cable, television can be used for the promotions which might prove effective approach for a brand. cluster 1 have more CS =

1. With household people 4,5,7 and 10 and customers falling in cluster 4 have the highest CS = 1.

-> Considering education as demographics, there are a high proportion of college graduates in cluster 4 which buys value added packs and premium soaps which shows high brand Loyalty. It looks like most of the people are in 4th and 5th level.

- SEC = 1(high socioeconomic class) with Cluster 4 customers who show a high tendency to buy premium soaps. There are high percentage of customers from other SEC sections in cluster 4, indicating that they prefer to buy any kind of soap. So, we can say that customers with high social economic status don't care about premium or popular soaps and also their brand royalty is high to the soap brand of their choice.
- Most of the SocioEconomic class are Native speakers.The most clusters are dominated by the customer with a common Native language.
- Most of the customers in each cluster are women. It is clearly seen that all the clusters have the highest number of women. Thus more products should be released that are more appealing to women than men.
- Cluster 4 consists customers with highly affluent people across all education levels. People of Age group 4 are most affluent customer and have potential to be converted into brand loyal customers.

Conclusion:

1) From the above plot we can conclude that most customers are female and they belong to Age group 4 in cluster 4. So based on this company should plan manufacturing of new products and their promotions accordingly. Also almost all of the customers from age Group 4 in most cluster are not brand loyal but prefer to buy value added packs, premium packs and soaps.

2) As most of the customers have TV/Cable at home ; It is the best way to promote the products.