Data Analysis - Clustering

Varun Bansal

2023-02-15

Setting the work directory.

Loading and attaching all the necessary packages.

```r
#Load packages

if(!require(tinytex)){
  install.packages("tinytex")
  library("tinytex")
}
```

```
## Loading required package: tinytex
```

```r
if (!require(lattice)) {
  install.packages("lattice")
  library(lattice)
}
```

```
## Loading required package: lattice
```

```r
if (!require(gridExtra)) {
  install.packages("gridExtra")
  library(gridExtra)
}
```

```
## Loading required package: gridExtra
```

Data Transformation

```r
# Reading File
df <- read.csv("Expense_Summary.txt", header = TRUE, sep = ",")
head(df, 5)
```

```
##    Food Enter   Edu Trans  Work House   Oth
## 1 0.043 0.085 0.525 0.180 0.005 0.150 0.012
## 2 0.123 0.055 0.002 0.169 0.121 0.266 0.265
## 3 0.043 0.085 0.506 0.193 0.006 0.155 0.012
## 4 0.119 0.038 0.002 0.301 0.139 0.228 0.172
## 5 0.122 0.038 0.002 0.225 0.095 0.354 0.164
```

Renaming all variables with my initials appended

```
# Appending initials to all variables in the data frame

df_VB <- df
colnames(df_VB) <- paste(colnames(df_VB), "VB", sep = "_")
head(df_VB, 5)
```

```
##   Food_VB Enter_VB Edu_VB Trans_VB Work_VB House_VB Oth_VB
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164
```

Standardizing the variables

Before we do standardization, lets look at the summary of data.

```
summary(df_VB)
```

```
##     Food_VB          Enter_VB           Edu_VB          Trans_VB     
##  Min.   :0.0180   Min.   :0.00400   Min.   :0.0010   Min.   :0.0190  
##  1st Qu.:0.0460   1st Qu.:0.03100   1st Qu.:0.0020   1st Qu.:0.1570  
##  Median :0.1190   Median :0.04200   Median :0.0690   Median :0.2020  
##  Mean   :0.1111   Mean   :0.04551   Mean   :0.2271   Mean   :0.1957  
##  3rd Qu.:0.1580   3rd Qu.:0.06100   3rd Qu.:0.5380   3rd Qu.:0.2410  
##  Max.   :0.3080   Max.   :0.11300   Max.   :0.7210   Max.   :0.3710  
##     Work_VB           House_VB          Oth_VB     
##  Min.   :0.00200   Min.   :0.0360   Min.   :0.004  
##  1st Qu.:0.00500   1st Qu.:0.1500   1st Qu.:0.010  
##  Median :0.09000   Median :0.2450   Median :0.118  
##  Mean   :0.08136   Mean   :0.2383   Mean   :0.101  
##  3rd Qu.:0.13750   3rd Qu.:0.3105   3rd Qu.:0.169  
##  Max.   :0.25600   Max.   :0.5090   Max.   :0.305  
```

There doesn't seem to be much of outliers.
Lets do the Shapiro-Wilk test to check for normality.

```
shapiro.test(df_VB$Food_VB)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  df_VB$Food_VB
## W = 0.92316, p-value < 2.2e-16
```

```
shapiro.test(df_VB$Enter_VB)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_VB$Enter_VB
## W = 0.97187, p-value = 0.0000000000001826
```

```
shapiro.test(df_VB$Edu_VB)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_VB$Edu_VB
## W = 0.72942, p-value < 2.2e-16
```

```
shapiro.test(df_VB$Trans_VB)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_VB$Trans_VB
## W = 0.98212, p-value = 0.0000000004057
```

```
shapiro.test(df_VB$Work_VB)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_VB$Work_VB
## W = 0.87438, p-value < 2.2e-16
```

```
shapiro.test(df_VB$House_VB)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_VB$House_VB
## W = 0.97357, p-value = 0.0000000000005649
```

```
shapiro.test(df_VB$Oth_VB)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_VB$Oth_VB
## W = 0.86911, p-value < 2.2e-16
```

Here in all the test, p-value is less than 0.05. So data is not normally distributed.

Also our data has a bounded range, so we will use min-max standardization function.

Using min-max standardization function

```r
# min-max standardization function

norm01 <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
```

```r
# performing min-max standardization function on all the variables

df_VB$Food_Norm01_VB <- norm01(df_VB$Food_VB)
df_VB$Enter_Norm01_VB <- norm01(df_VB$Enter_VB)
df_VB$Edu_Norm01_VB <- norm01(df_VB$Edu_VB)
df_VB$Trans_Norm01_VB <- norm01(df_VB$Trans_VB)
df_VB$Work_Norm01_VB <- norm01(df_VB$Work_VB)
df_VB$House_Norm01_VB <- norm01(df_VB$House_VB)
df_VB$Oth_Norm01_VB <- norm01(df_VB$Oth_VB)
```

```r
head(df_VB,3)
```

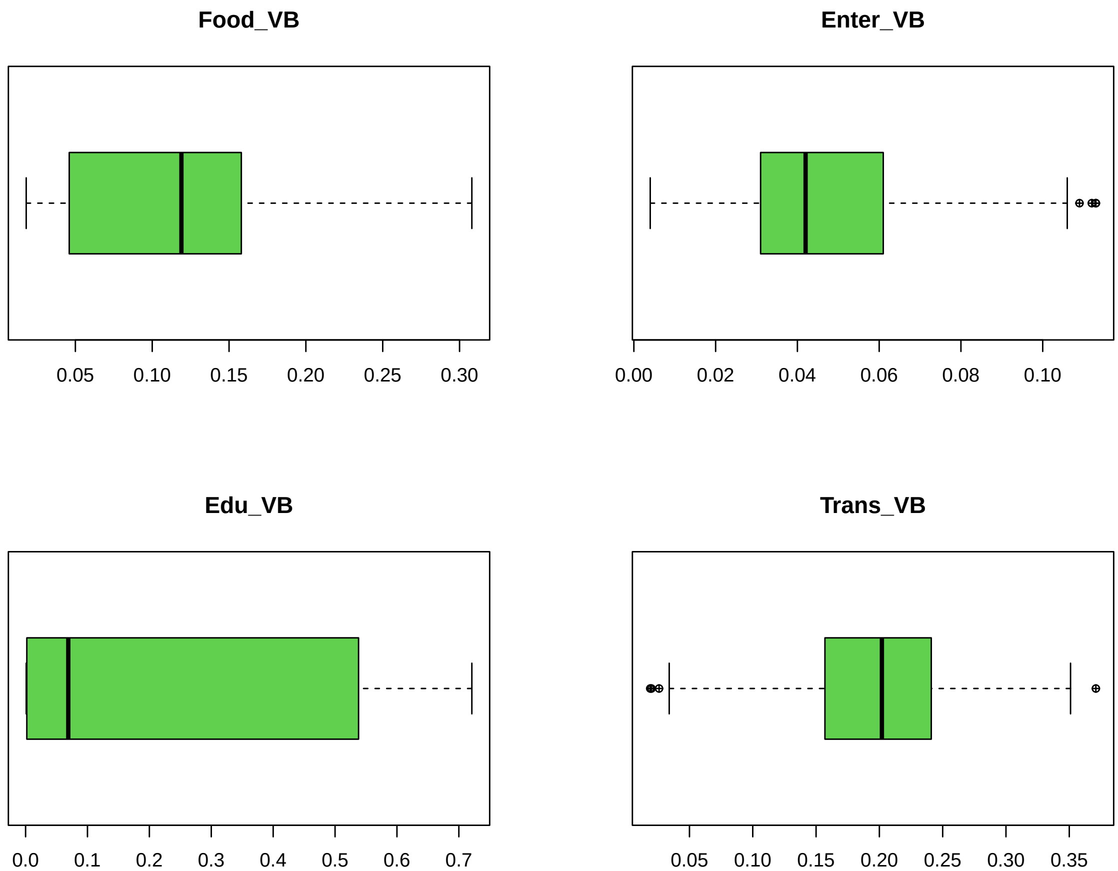
```
##   Food_VB Enter_VB Edu_VB Trans_VB Work_VB House_VB Oth_VB Food_Norm01_VB
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012      0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265      0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012      0.0862069
##   Enter_Norm01_VB Edu_Norm01_VB Trans_Norm01_VB Work_Norm01_VB House_Norm01_VB
## 1       0.7431193   0.727777778       0.4573864     0.01181102       0.2410148
## 2       0.4678899   0.001388889       0.4261364     0.46850394       0.4862579
## 3       0.7431193   0.701388889       0.4943182     0.01574803       0.2515856
##   Oth_Norm01_VB
## 1    0.02657807
## 2    0.86710963
## 3    0.02657807
```

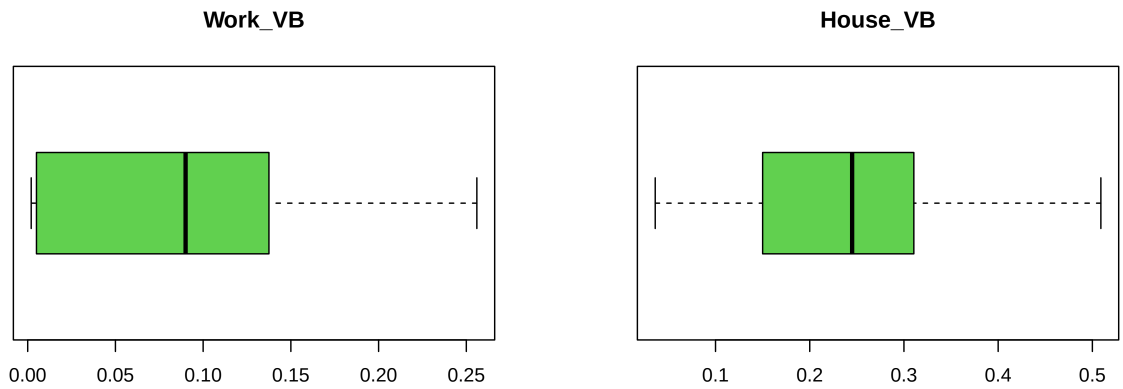
Descriptive Data Analysis

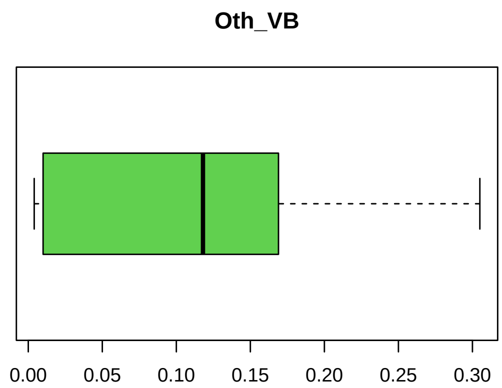
Graphical summaries of the data

Exploring the Data using boxplots.

```r
par(mfrow=c(2,2))

for (i in 1:7) {
  if (is.numeric(df_VB[,i])) {
      boxplot(df_VB[i], main=names(df_VB)[i],
              horizontal=TRUE, pch=10,
              col= 27)
  }
}
```


Food_VB
0.05
0.10
0.15
0.20
0.25
0.30
Enter_VB
0.00
0.02
0.04
0.06
0.08
0.10
Edu_VB
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Trans_VB
0.05
0.10
0.15
0.20
0.25
0.30
0.35



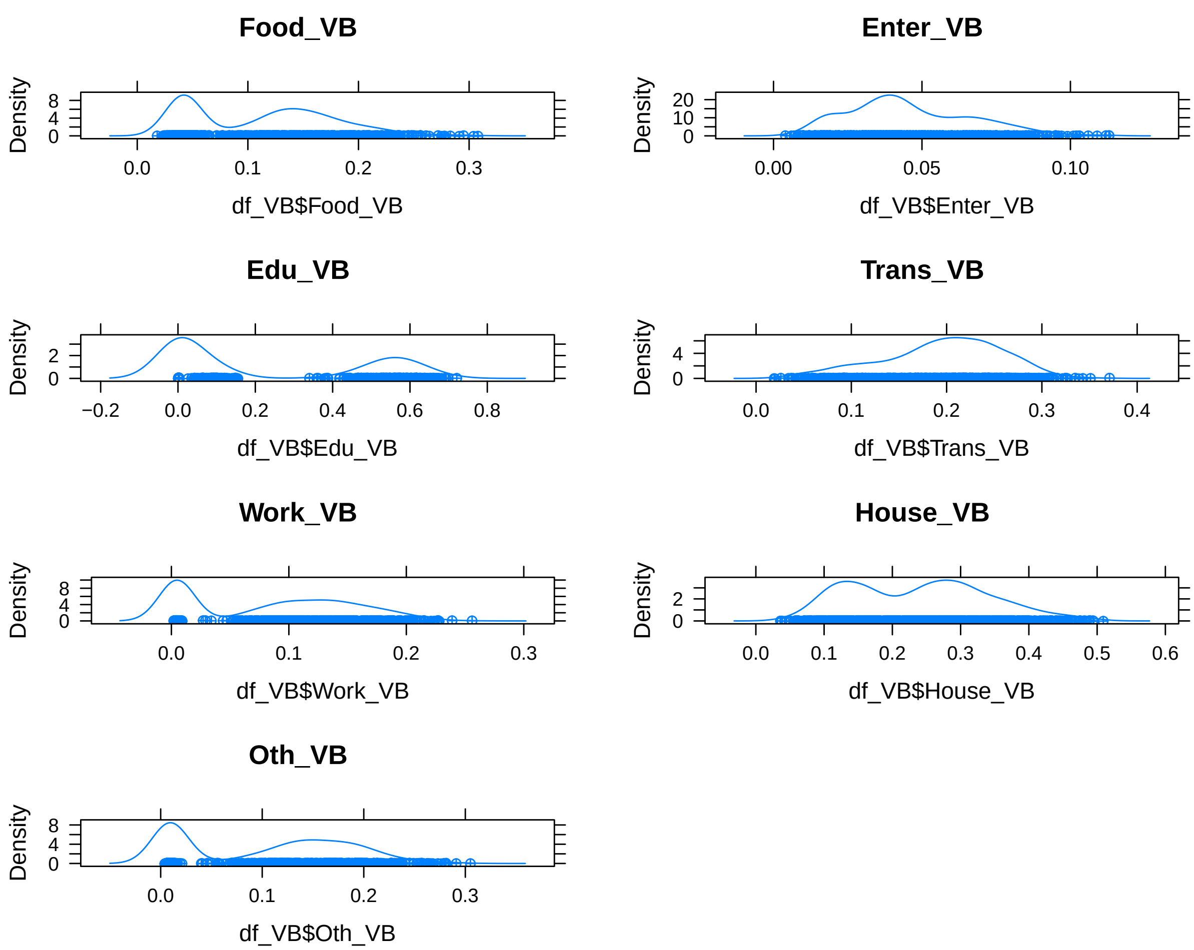



```
par(mfrow=c(1,1))
```

The box plot suggests that there are not many outliners in the dataset.

Now lets density plot and see if data is normally distributed.

```
dp1 <- densityplot( ~ df_VB$Food_VB, pch=10, main='Food_VB')
dp2 <- densityplot( ~ df_VB$Enter_VB, pch=10, main='Enter_VB')
dp3 <- densityplot( ~ df_VB$Edu_VB, pch=10, main='Edu_VB')
dp4 <- densityplot( ~ df_VB$Trans_VB, pch=10, main='Trans_VB')
dp5 <- densityplot( ~ df_VB$Work_VB, pch=10, main='Work_VB')
dp6 <- densityplot( ~ df_VB$House_VB, pch=10, main='House_VB')
dp7 <- densityplot( ~ df_VB$Oth_VB, pch=10, main='Oth_VB')

# Display the plots in a grid
grid.arrange(dp1, dp2, dp3, dp4, dp5, dp6, dp7, ncol = 2)
```



This also suggests data is not normally distributed.

3. Clustering

Using the K-Means procedure clustering

Setting Up for Clusters

```r
# Creating Variable for Elbow Chart
# Creating 2 to 7 Clusters

maxk <- 7    # max number of k

nk <- c(2:maxk) # list of numbers 2 to 7

wss <- rep(0,maxk-1)  # empty list having 7 zeros
```

Creating Clusters

```r
# Setting Clusters 2 to 7

for(k in 2:7){

  ClstrIncome_VB <- kmeans(df_VB[,c(8,13)], iter.max=10, centers=k, nstart=10)

  cat("****************  k = ",k, " ******************\n\n")
  cat("Cluster size: ", ClstrIncome_VB$size, "\n\n")
  cat("Cluster Centers: \n")
  print(ClstrIncome_VB$centers)
  cat("\nRatio of between-cluster variance to total variance ",
      ClstrIncome_VB$betweenss/ClstrIncome_VB$totss)
  cat("\n\n---------------------------------------------------------------\n\n")

  df_with_k <- paste("df_VB", k, sep="_")
  df_VB$cluster <- factor(ClstrIncome_VB$cluster)   # Adding Cluster tags to variables
  assign(df_with_k, df_VB)

  centers <- paste("centers", k, sep="_")

  # the data frame and assign it to the name
  assign(centers, data.frame(cluster=factor(1:k), ClstrIncome_VB$centers))
  wss[k-1] <- ClstrIncome_VB$tot.withinss
}
```

```
## ****************  k =  2  ******************
##
## Cluster size:  417 642
##
## Cluster Centers:
##   Food_Norm01_VB House_Norm01_VB
## 1     0.08833209       0.2114216
## 2     0.47218283       0.5681802
##
## Ratio of between-cluster variance to total variance  0.6967588
##
## ---------------------------------------------------------------
##
## ****************  k =  3  ******************
##
## Cluster size:  409 464 186
##
## Cluster Centers:
##   Food_Norm01_VB House_Norm01_VB
## 1     0.08507714       0.2077981
## 2     0.41252229       0.5016585
## 3     0.61166110       0.7267499
##
## Ratio of between-cluster variance to total variance  0.8161152
##
## ---------------------------------------------------------------
##
## ****************  k =  4  ******************
```

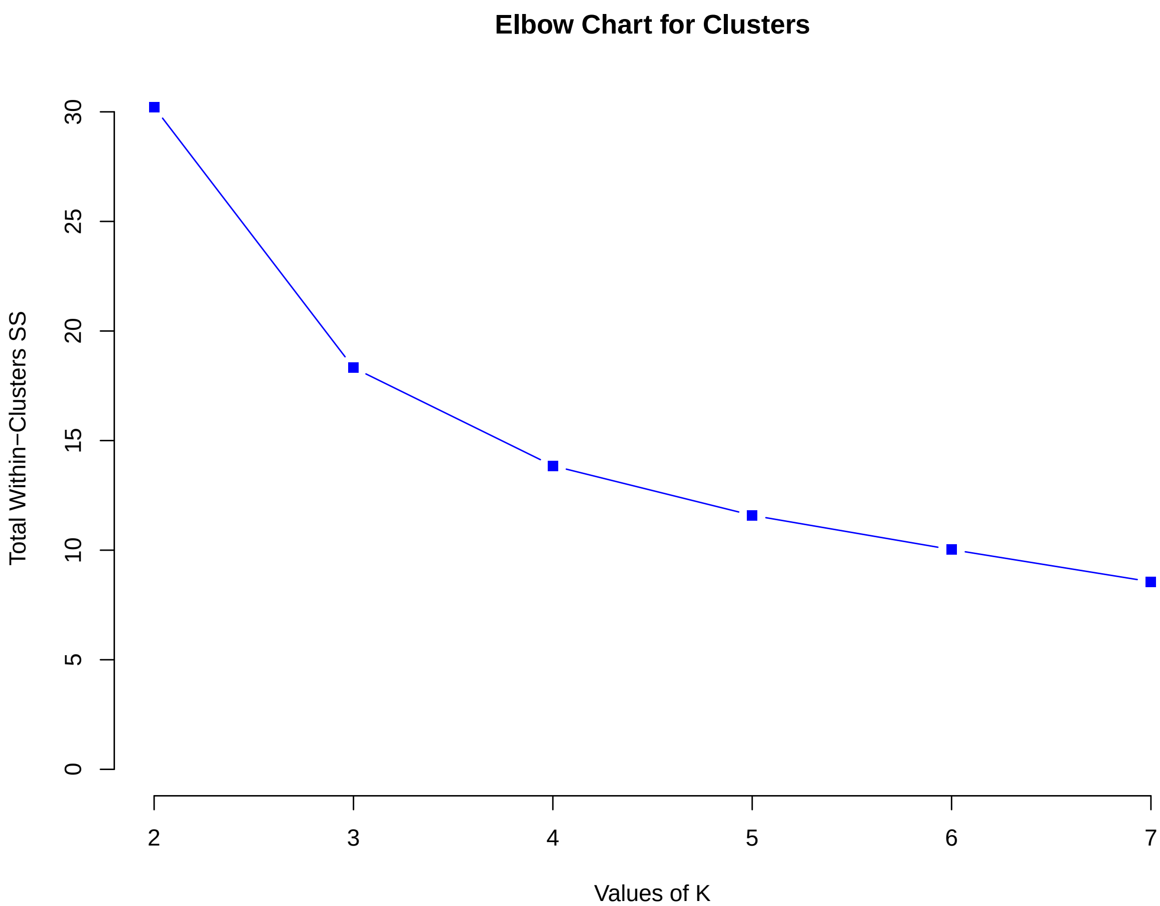
```
##
## Cluster size:  404 228 262 165
##
## Cluster Centers:
##   Food_Norm01_VB House_Norm01_VB
## 1      0.0850717       0.2039803
## 2      0.4941168       0.4409054
## 3      0.3427086       0.5665155
## 4      0.6252038       0.7368057
##
## Ratio of between-cluster variance to total variance  0.8609939
##
## ---------------------------------------------------------------
##
## ****************  k =  5  *******************
##
## Cluster size:  137 214 403 75 230
##
## Cluster Centers:
##   Food_Norm01_VB House_Norm01_VB
## 1     0.49859049       0.7446181
## 2     0.49444086       0.4403983
## 3     0.08470951       0.2035736
## 4     0.74013793       0.6800282
## 5     0.33134933       0.5375402
##
## Ratio of between-cluster variance to total variance  0.8836823
##
## ---------------------------------------------------------------
##
## ****************  k =  6  *******************
##
## Cluster size:  178 61 156 91 402 171
##
## Cluster Centers:
##   Food_Norm01_VB House_Norm01_VB
## 1     0.50896939       0.4347103
## 2     0.75353307       0.6476623
## 3     0.31279841       0.4835475
## 4     0.56411520       0.8033362
## 5     0.08413965       0.2034121
## 6     0.40619076       0.6183623
##
## Ratio of between-cluster variance to total variance  0.8993687
##
## ---------------------------------------------------------------
##
## ****************  k =  7  *******************
##
## Cluster size:  158 179 157 233 99 59 174
##
## Cluster Centers:
##   Food_Norm01_VB House_Norm01_VB
## 1     0.35390659       0.4549870
```

```
## 2     0.37087266       0.6149267
## 3     0.52831100       0.4463312
## 4     0.08052390       0.1521927
## 5     0.56318356       0.7908045
## 6     0.75797779       0.6484395
## 7     0.08902101       0.2809895
##
## Ratio of between-cluster variance to total variance  0.9141097
##
## ---------------------------------------------------------------
```

3. Creating the WSS plots

Plotting 'Elbow' chart

```
plot(2:maxk, wss,
     type="b",
     pch = 15,
     col="blue",
     frame = FALSE,
     main="Elbow Chart for Clusters",
     xlab="Values of K",
     ylab="Total Within-Clusters SS",
     ylim=c(0,max(wss)))
```



Looking at the elbow chart, there seems to be a bend at 4.
So we choose the value of k as 4.

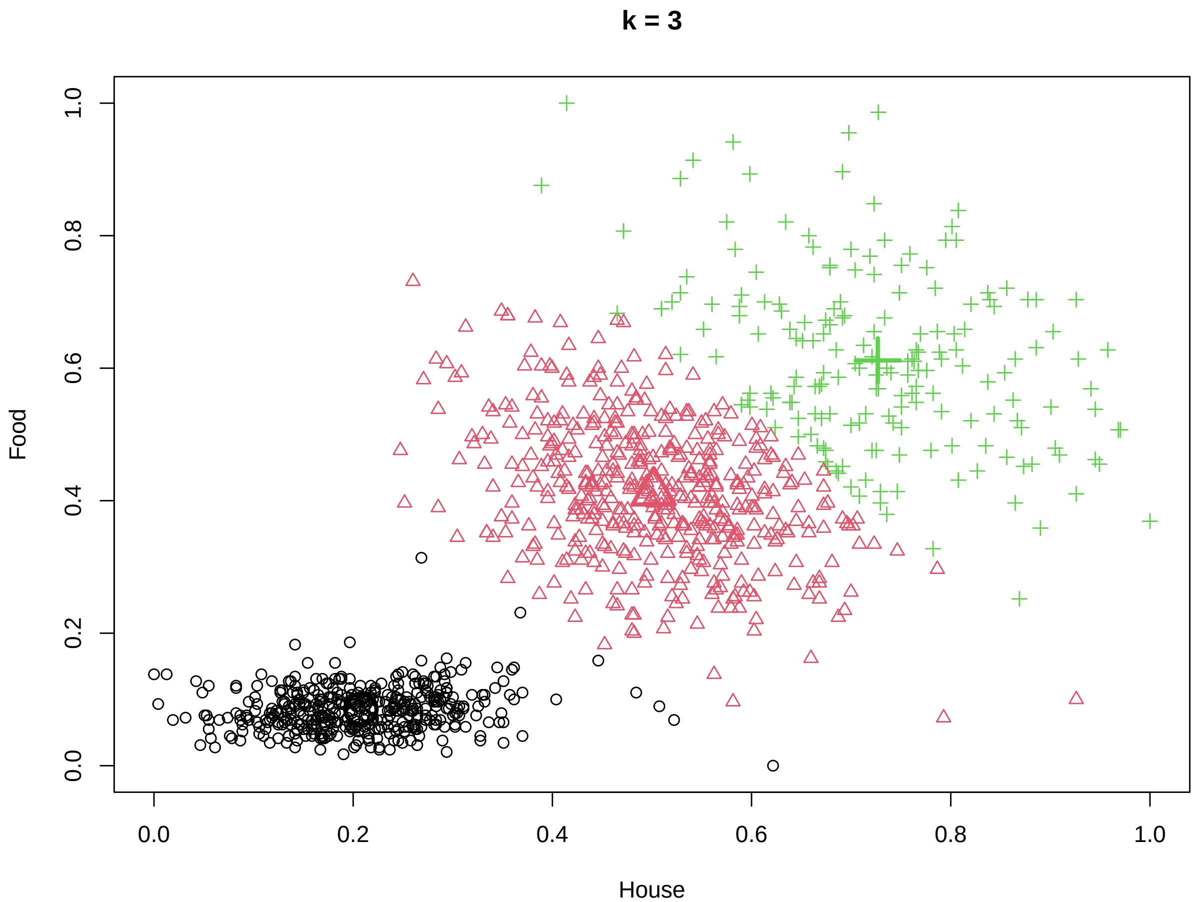
Evaluation of Clusters

Plotting the clusters

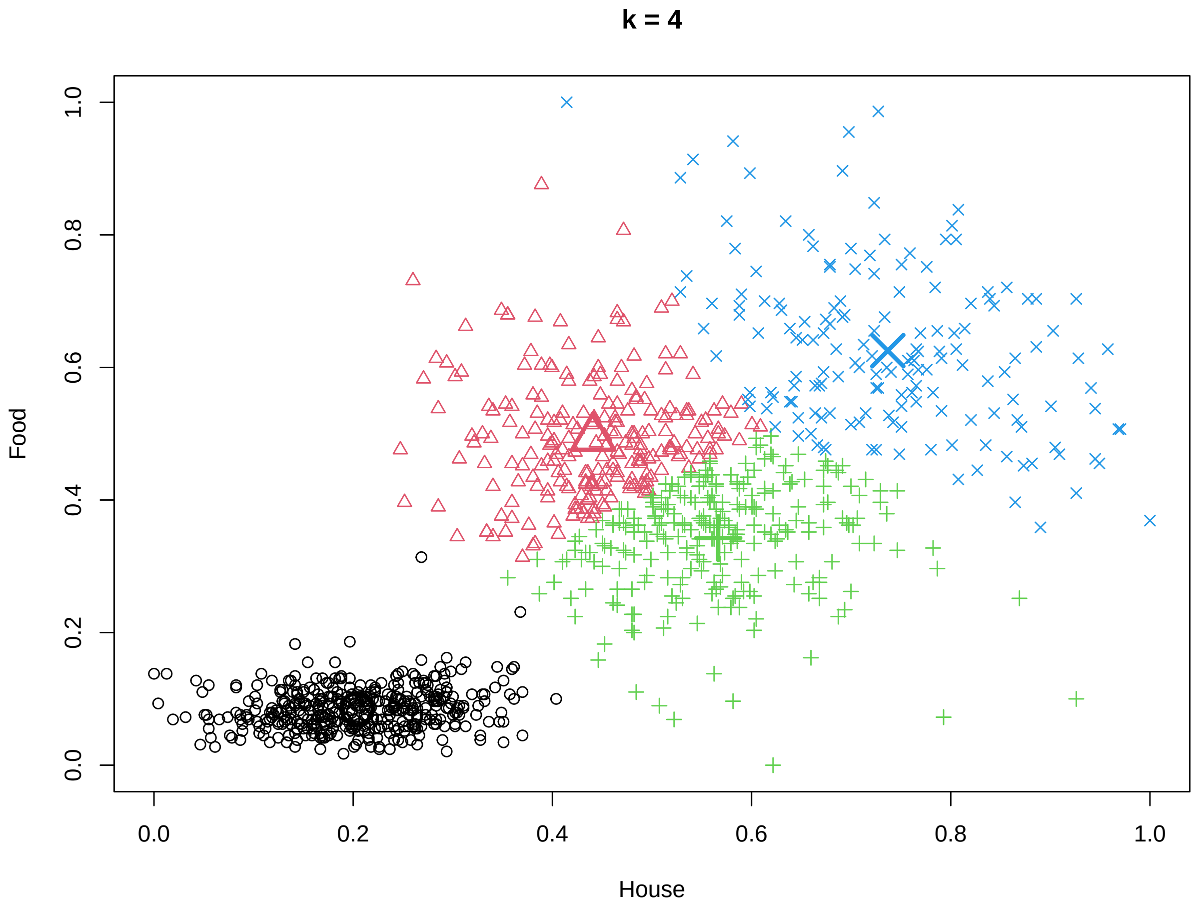
We have choosen k=4.
Plotting the clusters for k=3, k=4, k=5

```
# K=3

plot(df_VB_3$House_Norm01_VB, df_VB_3$Food_Norm01_VB,
     col=df_VB_3$cluster, pch=as.numeric(df_VB_3$cluster),
     main = "k = 3",
     xlab= "House",
     ylab = "Food")

points(centers_3$House_Norm01_VB, centers_3$Food_Norm01_VB,
    col=centers_3$cluster,
```

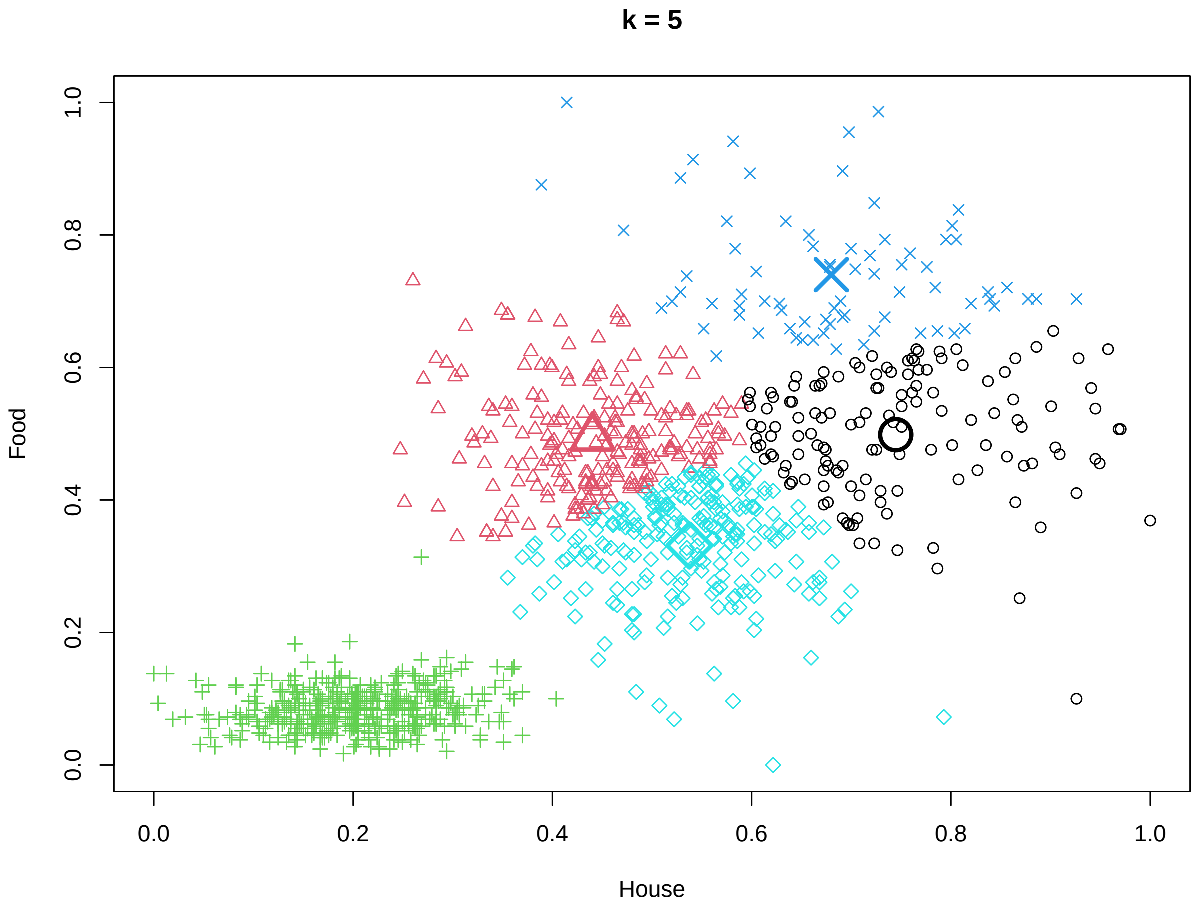
```
    pch=as.numeric(centers_3$cluster),
    cex=3, lwd=3)
```




```
# K=4

plot(df_VB_4$House_Norm01_VB, df_VB_4$Food_Norm01_VB,
     col=df_VB_4$cluster, pch=as.numeric(df_VB_4$cluster),
     main = "k = 4",
     xlab= "House",
     ylab = "Food")

points(centers_4$House_Norm01_VB, centers_4$Food_Norm01_VB,
    col=centers_4$cluster,
    pch=as.numeric(centers_4$cluster),
    cex=3, lwd=3)
```




```r
# K=5

plot(df_VB_5$House_Norm01_VB, df_VB_5$Food_Norm01_VB,
     col=df_VB_5$cluster, pch=as.numeric(df_VB_5$cluster),
     main = "k = 5",
     xlab= "House",
     ylab = "Food")

points(centers_5$House_Norm01_VB, centers_5$Food_Norm01_VB,
    col=centers_5$cluster,
    pch=as.numeric(centers_5$cluster),
    cex=3, lwd=3)
```



Looking at the WSS plot and the charts, at k = 3, clusters look the best and well segregated.

Summarizing the Clusters

```
# Creating summary report

SumClusters_VB <- aggregate(
  cbind(Food_VB, Enter_VB, Edu_VB, Trans_VB, Work_VB, House_VB, Oth_VB) ~ cluster,
  df_VB_3,
  FUN = mean)

SumClusters_VB
```

```
##   cluster    Food_VB   Enter_VB      Edu_VB  Trans_VB     Work_VB  House_VB
## 1       1 0.04267237 0.06595844 0.550488998 0.1877482 0.006486553 0.1342885
## 2       2 0.13763147 0.03792026 0.004112069 0.2357629 0.142295259 0.2732845
## 3       3 0.19538172 0.01949462 0.072290323 0.1131290 0.094005376 0.3797527
##       Oth_VB
```

```
## 1 0.01233741
## 2 0.16910345
## 3 0.12590860
```

Suitable descriptive names for each cluster.

For cluster 1: High on transport and housing, negligible on education and entertainment.

For cluster 2: High on housing, low on entertainment and education.

For cluster 3: High on education, negligible on work.

Uses for this clustering scheme.

There can be many uses of this clustering scheme. Some of them are-

This scheme may come handy in making business strategies. For example, if a company primarily sells products related to housing and transportation, they may want to expand their offerings to appeal to customers in cluster 1. Similarly, if a company is developing a new product related to education, they may want to focus on customers in cluster 3, who are likely to spend more in this area.

This clustering scheme may also help government in there policy-making. They can identify areas where public spending should be prioritized. For example, if cluster 3 represents a large portion of the population, policymakers may want to invest more resources into education to meet the needs of this group. This may also help the government to analyse why people are not willing to spend on education in Cluster 1 and