PROG8430-Assignment 3

Mohammed Saifullah

2023-03-06

Loading necessary packages

```
#Load packages
if(!require(pastecs)){install.packages("pastecs")}
```

```
## Loading required package: pastecs
```

```
library("pastecs")

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

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

```
library("lattice")

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

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

```
library("tinytex")
```

Part 1 - Data Transformation

1. Appending initials to all column names

```
getwd() #verify working directory
```

```
## [1] "E:/Big Data Solution Architecture/PROG8430 - Data Analysis Mathematics, Algorithms and Modeling/
```

```
#Read the text data file into a Data Frame
IncomeEx_MS <- read.table("PROG8430-23W-Assign03.txt", sep=',', header = TRUE)
#concatenating initial 'MS' to all column names
colnames(IncomeEx_MS) <- paste(colnames(IncomeEx_MS), "MS", sep = "_")
#Display first 5 rows of the dataset just to verify loaded and name transformation is  successful
head(IncomeEx_MS, 5)
```

```
##   Food_MS Enter_MS Edu_MS Trans_MS Work_MS House_MS Oth_MS
## 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
```

```
#Checking Data Structure
str(IncomeEx_MS)
```

```
## 'data.frame':    1059 obs. of  7 variables:
##  $ Food_MS : num  0.043 0.123 0.043 0.119 0.122 0.084 0.089 0.03 0.134 0.035 ...
##  $ Enter_MS: num  0.085 0.055 0.085 0.038 0.038 0.05 0.042 0.041 0.036 0.067 ...
##  $ Edu_MS  : num  0.525 0.002 0.506 0.002 0.002 0.002 0.002 0.647 0.002 0.559 ...
##  $ Trans_MS: num  0.18 0.169 0.193 0.301 0.225 0.285 0.215 0.156 0.281 0.168 ...
##  $ Work_MS : num  0.005 0.121 0.006 0.139 0.095 0.079 0.204 0.003 0.141 0.003 ...
##  $ House_MS: num  0.15 0.266 0.155 0.228 0.354 0.264 0.254 0.116 0.242 0.159 ...
##  $ Oth_MS  : num  0.012 0.265 0.012 0.172 0.164 0.237 0.195 0.006 0.164 0.01 ...
```

Following 2 functions are for data normalization Ref: PROG8430-K-Means-Example.Rmd, David Marsh, 12/01/2022

```
#Min-Max standardization function
norm01 <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

#SD standardization function
normn <- function(x) {
  return ((x-mean(x))/sd(x))
}
```

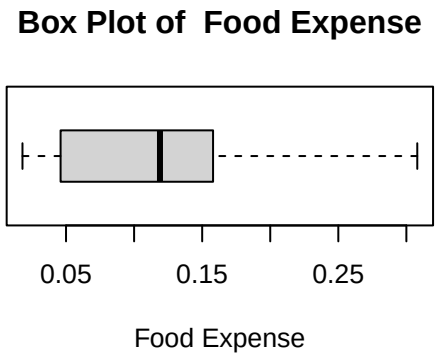
list of expenses to be used for chart labeling

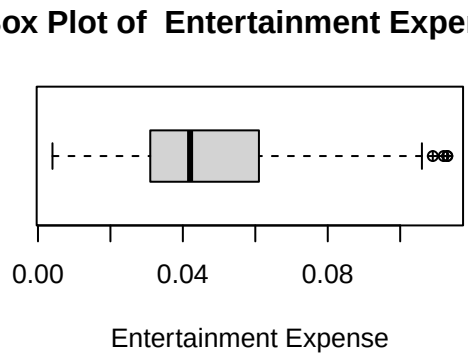
```
expenses <- list("Food", "Entertainment", "Education", "Transpotation", "Work", "Housing", "Other")
```

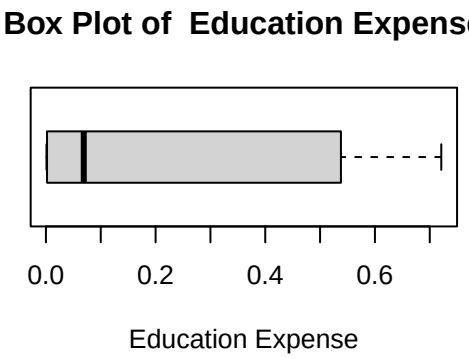
2. Standarizing all of the variables to have similar scaling

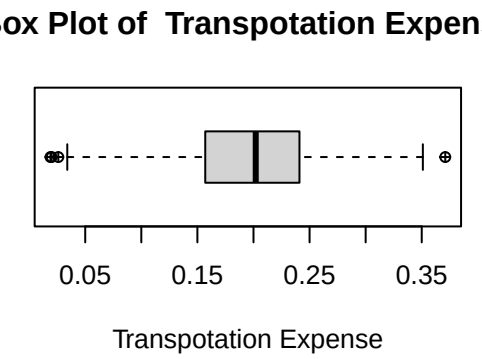
Exploring data to determine which standardization method to use.
If data contain Outlier in a variable, I will use standard deviation standardization If data is tightly clustered in a variable, I will use min-max standardization A small box with short whisker indicates tight cluster.

```
par(mfrow=c(2,2)) # setting up 2 by 2 chart

for (i in 1:ncol(IncomeEx_MS)) { # this loop will continue from 1 to number of columns(nco) in the data
  if (is.numeric(IncomeEx_MS[,i])) { # if column i is numeric execute, data type check here
    #box plot for different expense category
    boxplot(IncomeEx_MS[i], horizontal=TRUE, pch=10,
            main=paste("Box Plot of ", expenses[i], "Expense"), #constructing main chart header
            xlab = paste(expenses[i], "Expense")) #constructing x label
  }
}
```

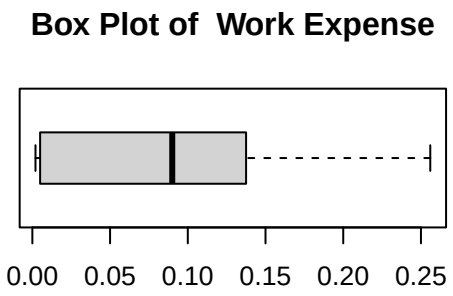


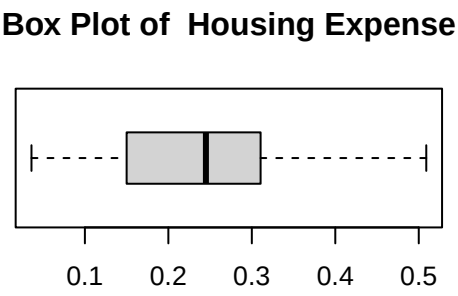


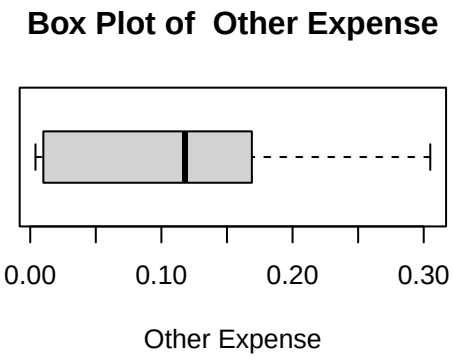





```
par(mfrow=c(1,1)) # resetting to 1 by 1 chart
```







Variables Entr_MS(Entertainment expense) and Trans_MS(Transportation expense) has outliars, so I am going to use standard deviation standardization method. For other variables I will use mim-max standardization

```
#applying standard deviation standardization method to following variables
IncomeEx_MS$EnterNorm_MS <- norm01(IncomeEx_MS$Enter_MS)
IncomeEx_MS$TransNorm_MS <- norm01(IncomeEx_MS$Trans_MS)
IncomeEx_MS$FoodNorm_MS <- norm01(IncomeEx_MS$Food_MS)
IncomeEx_MS$EduNorm_MS <- norm01(IncomeEx_MS$Edu_MS)
IncomeEx_MS$WorkNorm_MS <- norm01(IncomeEx_MS$Work_MS)
IncomeEx_MS$HouseNorm_MS <- norm01(IncomeEx_MS$House_MS)
IncomeEx_MS$OthNorm_MS <- norm01(IncomeEx_MS$Oth_MS)

head(IncomeEx_MS)
```

```
##   Food_MS Enter_MS Edu_MS Trans_MS Work_MS House_MS Oth_MS EnterNorm_MS
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012    0.7431193
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265    0.4678899
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012    0.7431193
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172    0.3119266
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164    0.3119266
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237    0.4220183
##   TransNorm_MS FoodNorm_MS  EduNorm_MS WorkNorm_MS HouseNorm_MS OthNorm_MS
## 1    0.4573864   0.0862069 0.727777778  0.01181102    0.2410148 0.02657807
## 2    0.4261364   0.3620690 0.001388889  0.46850394    0.4862579 0.86710963
## 3    0.4943182   0.0862069 0.701388889  0.01574803    0.2515856 0.02657807
## 4    0.8011364   0.3482759 0.001388889  0.53937008    0.4059197 0.55813953
```

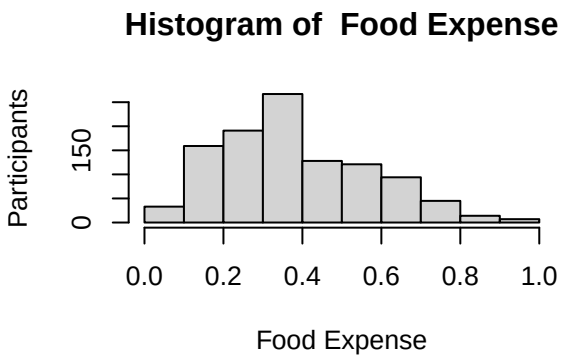
```
## 5    0.5852273   0.3586207 0.001388889  0.36614173    0.6723044 0.53156146
## 6    0.7556818   0.2275862 0.001388889  0.30314961    0.4820296 0.77408638
```

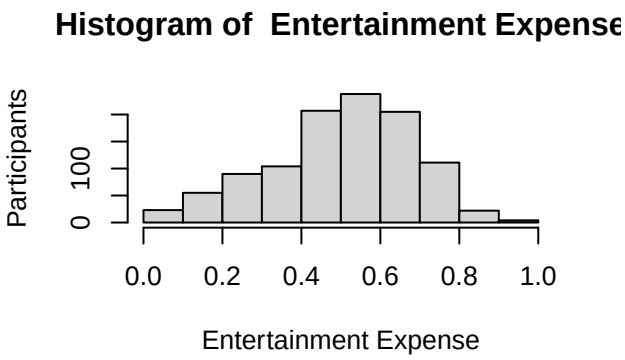
Part 2 - Descriptive Data Analysis

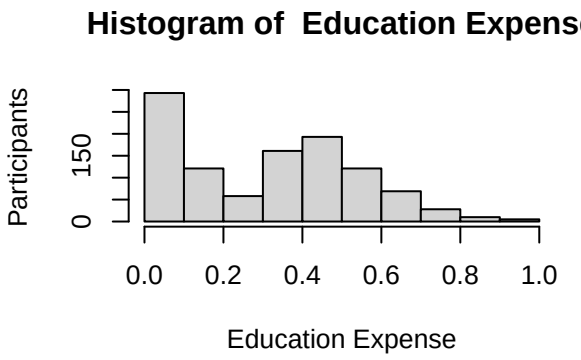
1. Creating graphical summaries of the data

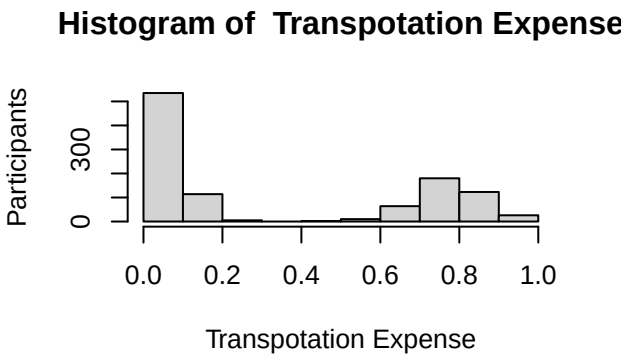
Creating histograms for data observation

```
par(mfrow=c(2,2)) # setting up 2 by 2 chart

for (j in 8:ncol(IncomeEx_MS)) { # this loop will continue from 1 to number of columns(nco) in the data
  if (is.numeric(IncomeEx_MS[,j])) { # if column i is numeric execute, data type check here
      #Histogram for different ecpense category
       hist(IncomeEx_MS[,j],
            main=paste("Histogram of ", expenses[j-7], "Expense"), #constructing main chart header
            xlab = paste(expenses[j-7], "Expense"), #constructing x label
            ylab = "Participants")
  }
}
```

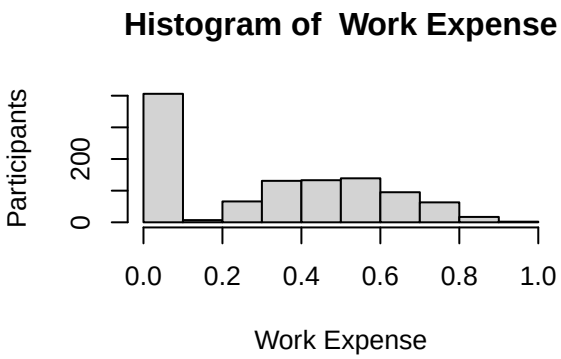


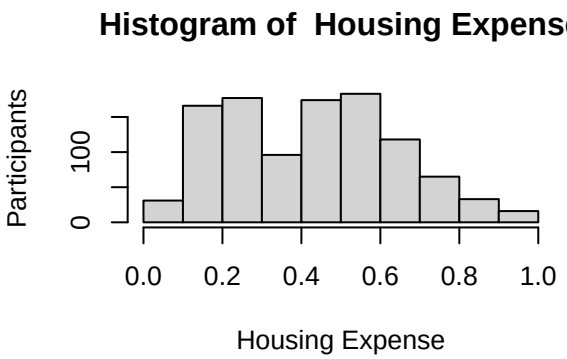







```
par(mfrow=c(1,1)) # resetting to 1 by 1 chart
```





Interpretation: From histogram of different variables, we can interpret that data is not normally distributed.

Part 3 - Clustering

1. Creating segmentation/cluster schemes for k=2,3,4,5,6,7

Clusters Setup

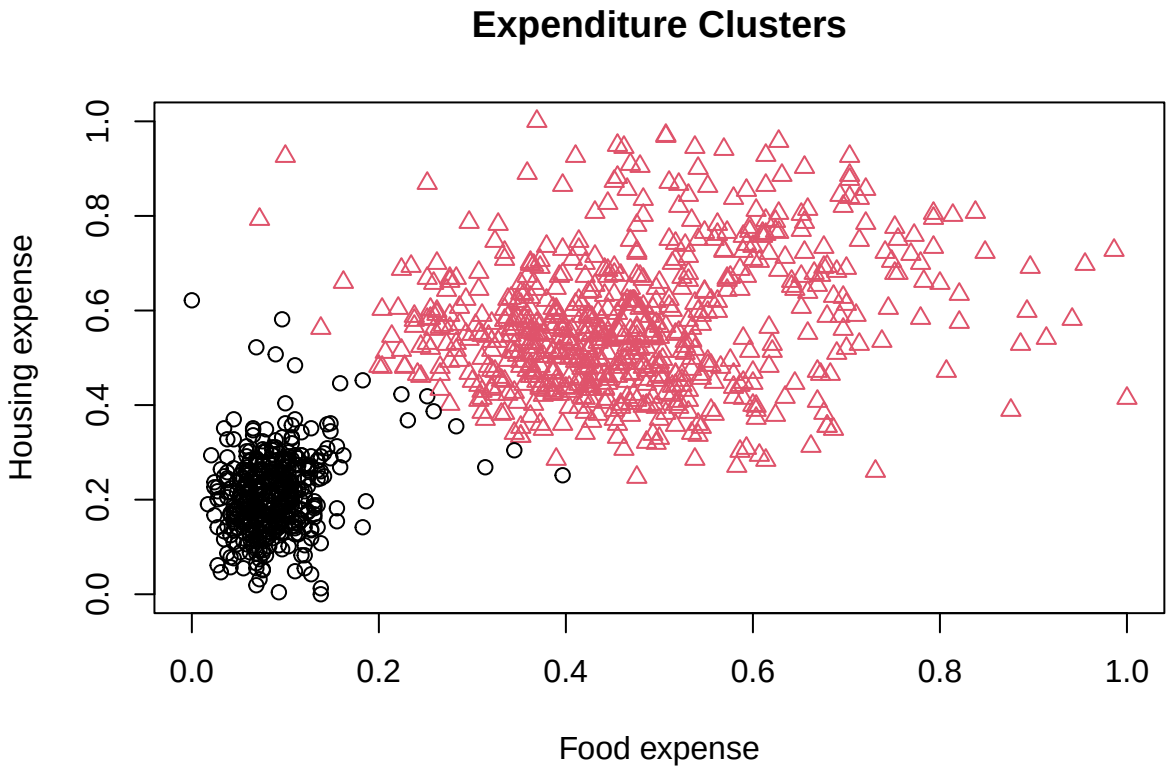
```
# Elbow Chart variables
# Trying for 2 to 7 Clusters
maxk_MS <- 7 #setting maximum k to d
nk_MS <- c(2:maxk_MS) # vector from 2 to 7
wss_MS <- rep(0,maxk_MS-1) #  Vector of 0s initially. It will hold within sum of square(WSS) values in
```

Creating Clusters

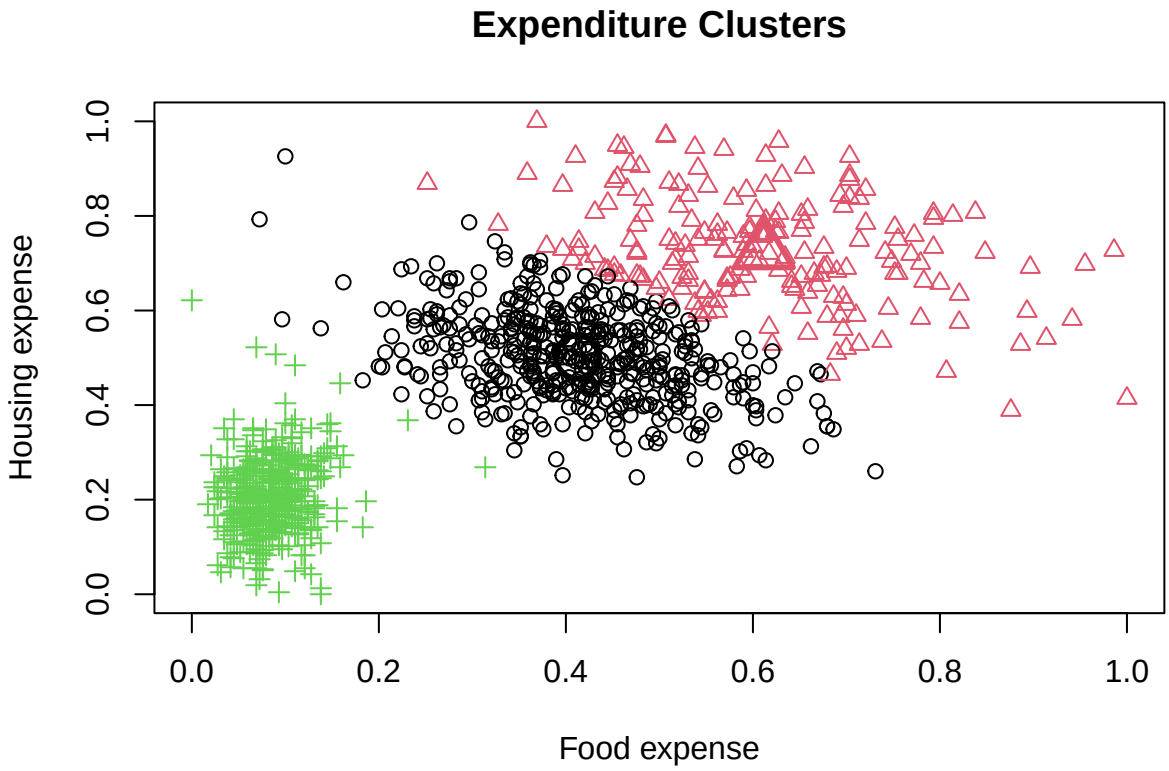
```
#for loop to create clusters for all k values 2-7
for(kval in 2:7){
  #Setting Number of Clusters
  k=kval
  # max iteration is 10.
  # Column 10 and 13
  # Considering 10 different starting centroids - nstart
```

```
  # Output one with highest percentage
  ClusterExp_MS <- kmeans(IncomeEx_MS[,c(10,13)], iter.max=10, centers=k, nstart=10)
  print(ClusterExp_MS$size)
  print(ClusterExp_MS$centers)
  print(ClusterExp_MS$betweenss/ClusterExp_MS$totss)

  IncomeEx_MS$cluster <- factor(ClusterExp_MS$cluster)   # Preparing clusters for summary

  centers_MS <- data.frame(cluster=factor(1:k), ClusterExp_MS$centers)

  wss_MS[k-1] <- ClusterExp_MS$tot.withinss # using sum of square value. Set value in WSS vector(replac

  #Creating a scatter plot showing the clusters and color-coded datapoints.
  plot(IncomeEx_MS$FoodNorm_MS, IncomeEx_MS$HouseNorm_MS, # plotting Food and Housing
       col=IncomeEx_MS$cluster, pch=as.numeric(IncomeEx_MS$cluster), #using cluster number for color an
       main = "Expenditure Clusters",
       xlab = "Food expense",
       ylab = "Housing expense")
  points(centers_MS$FoodNorm_MS, centers_MS$HouseNorm_MS, # superimposing points on plot
         col=centers_MS$cluster, pch=as.numeric(centers_MS$cluster), #using cluster number for color an
         cex=3, lwd=3) # marker size

}
```

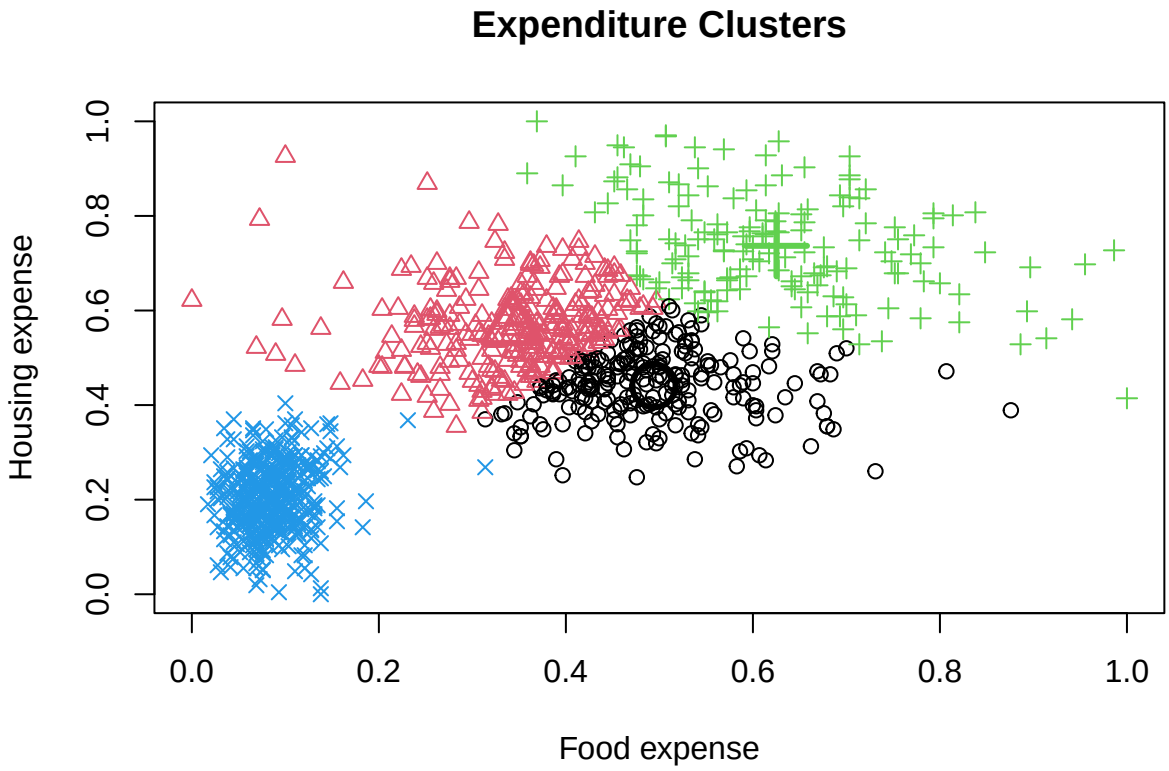
```
## [1] 417 642
##   FoodNorm_MS HouseNorm_MS
## 1  0.08833209    0.2114216
## 2  0.47218283    0.5681802
## [1] 0.6967588
```



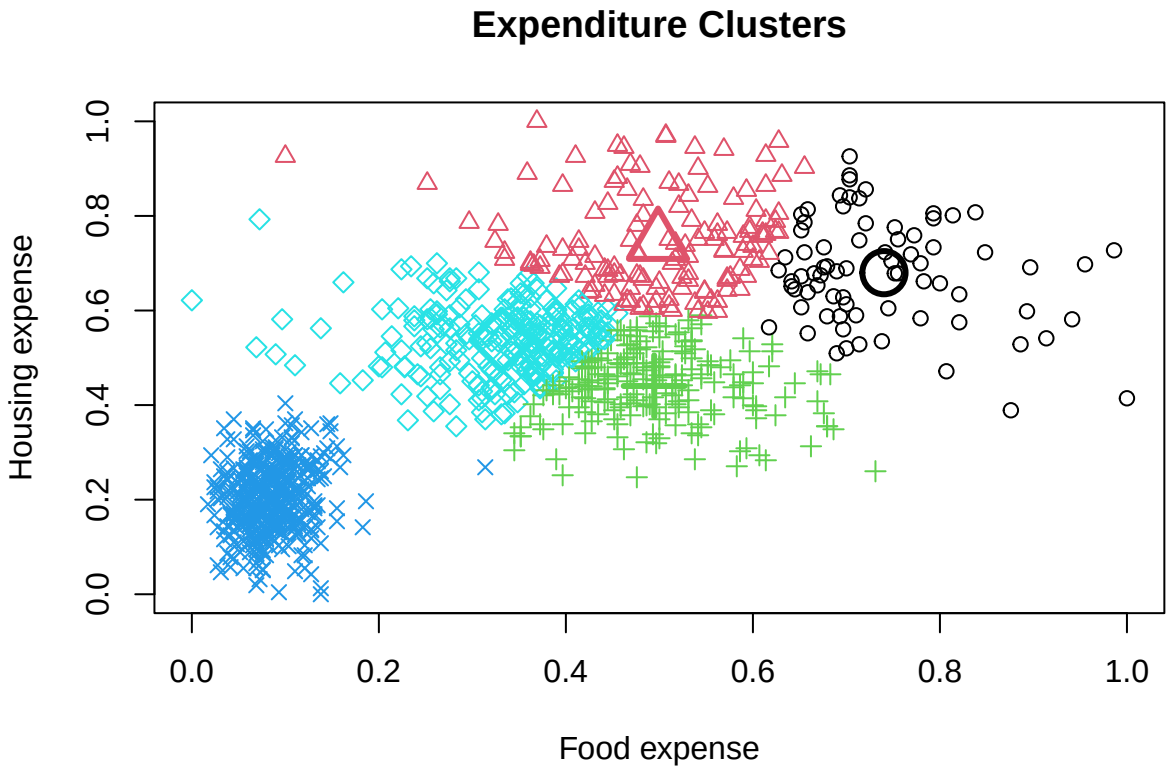

```
## [1] 464 186 409
##   FoodNorm_MS HouseNorm_MS
## 1  0.41252229    0.5016585
## 2  0.61166110    0.7267499
## 3  0.08507714    0.2077981
## [1] 0.8161152
```



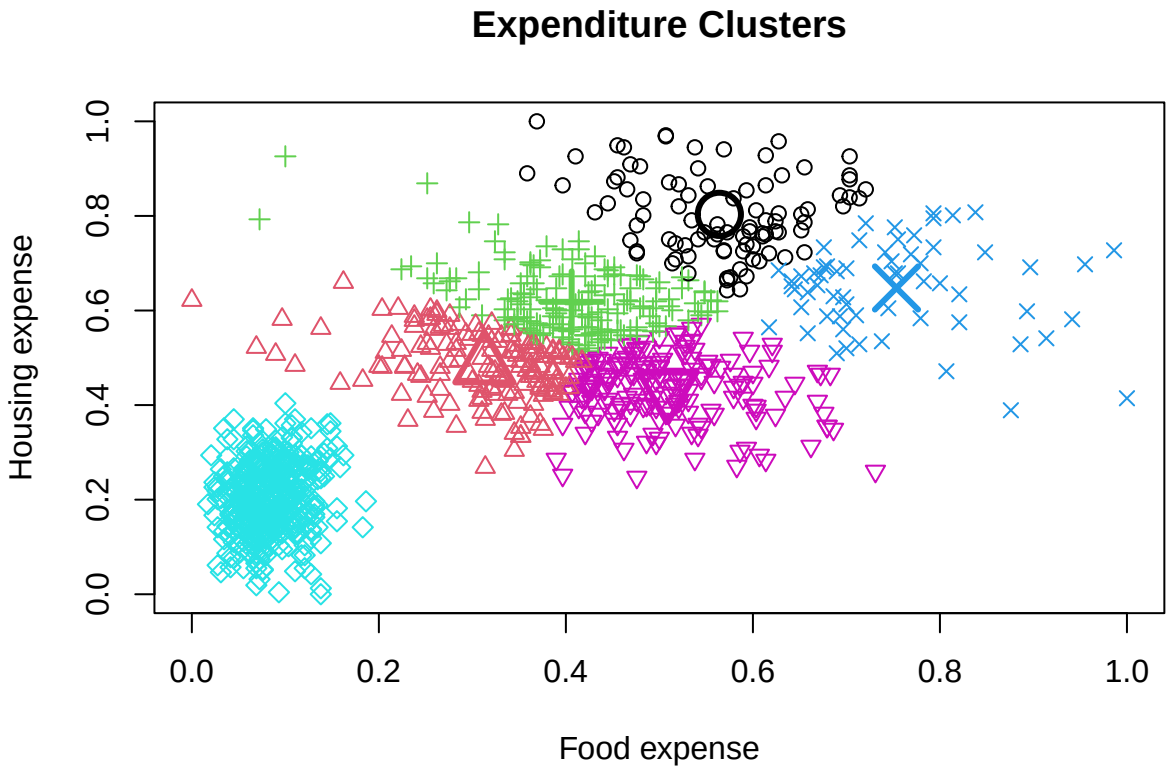

```
## [1] 228 262 165 404
##   FoodNorm_MS HouseNorm_MS
## 1   0.4941168    0.4409054
## 2   0.3427086    0.5665155
## 3   0.6252038    0.7368057
## 4   0.0850717    0.2039803
## [1] 0.8609939
```



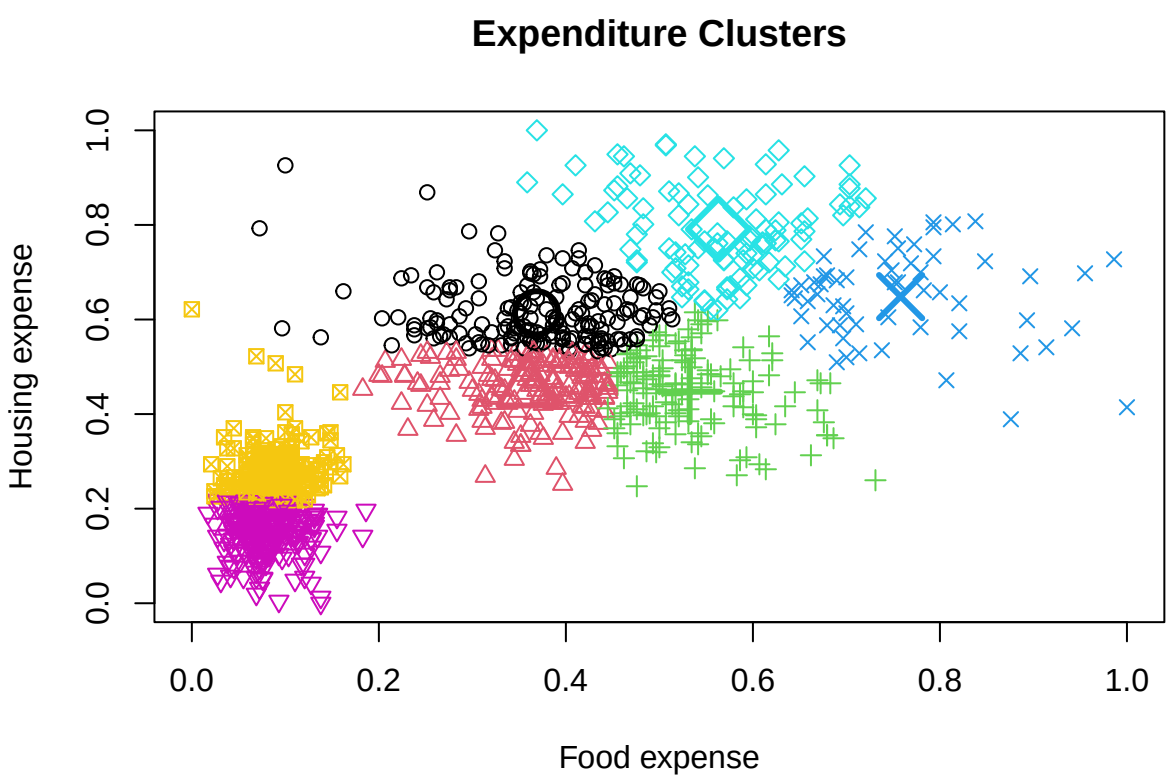

```
## [1]  75 137 214 403 230
##   FoodNorm_MS HouseNorm_MS
## 1  0.74013793    0.6800282
## 2  0.49859049    0.7446181
## 3  0.49444086    0.4403983
## 4  0.08470951    0.2035736
## 5  0.33134933    0.5375402
## [1] 0.8836823
```




```
## [1]  91 156 171  61 402 178
##   FoodNorm_MS HouseNorm_MS
## 1  0.56411520    0.8033362
## 2  0.31279841    0.4835475
## 3  0.40619076    0.6183623
## 4  0.75353307    0.6476623
## 5  0.08413965    0.2034121
## 6  0.50896939    0.4347103
## [1] 0.8993687
```




```
## [1] 180 163 151  59  99 233 174
##   FoodNorm_MS HouseNorm_MS
## 1  0.36865900    0.6141179
## 2  0.35895917    0.4536764
## 3  0.53231331    0.4473069
## 4  0.75797779    0.6484395
## 5  0.56318356    0.7908045
## 6  0.08052390    0.1521927
## 7  0.08902101    0.2809895
## [1] 0.9141005
```




```
head(IncomeEx_MS)
```

```
##   Food_MS Enter_MS Edu_MS Trans_MS Work_MS House_MS Oth_MS EnterNorm_MS
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012    0.7431193
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265    0.4678899
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012    0.7431193
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172    0.3119266
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164    0.3119266
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237    0.4220183
##   TransNorm_MS FoodNorm_MS  EduNorm_MS WorkNorm_MS HouseNorm_MS OthNorm_MS
## 1    0.4573864   0.0862069 0.727777778  0.01181102    0.2410148 0.02657807
## 2    0.4261364   0.3620690 0.001388889  0.46850394    0.4862579 0.86710963
## 3    0.4943182   0.0862069 0.701388889  0.01574803    0.2515856 0.02657807
## 4    0.8011364   0.3482759 0.001388889  0.53937008    0.4059197 0.55813953
## 5    0.5852273   0.3586207 0.001388889  0.36614173    0.6723044 0.53156146
## 6    0.7556818   0.2275862 0.001388889  0.30314961    0.4820296 0.77408638
##   cluster
## 1       7
## 2       2
## 3       7
## 4       2
## 5       1
## 6       2
```

2. Creating the WSS plots to select a suitable k value based on the “elbow”.

```
plot(2:maxk_MS, wss_MS, # 2 to max of cluster
     type="b", pch = 19, frame = FALSE,
     main="Elbow Chart for Clusters",
     xlab="Values of K",
     ylab="Total Within-Clusters SS",
     ylim=c(0,max(wss_MS)))
```

Interpretation: Selecting k value of 3 based on the elbow.

Part 4 - Cluster Evaluation

1. Creating a scatter plot showing the clusters and color-coded datapoints. Clusters for all K values created in loop above.

Interpretation: 2. Based on the WSS plot and the charts I can conclude that the Red cluster best describes the data.

3. Summarize the Clusters

```
#Naming cluster
#grouping by cluster
SummClusters_MS <- aggregate(cbind(Food_MS, Enter_MS, Edu_MS, Trans_MS, Work_MS, House_MS, Oth_MS) ~ clu
SummClusters_MS
```

```
##   cluster    Food_MS   Enter_MS      Edu_MS  Trans_MS     Work_MS  House_MS
## 1       1 0.12491111 0.03348889 0.008227778 0.2232667 0.134561111 0.3264778
## 2       2 0.12209816 0.04163804 0.004564417 0.2448712 0.148656442 0.2505890
## 3       3 0.17237086 0.03735762 0.004788079 0.2308675 0.139516556 0.2475762
## 4       4 0.23781356 0.01766102 0.081101695 0.1034068 0.094779661 0.3427119
## 5       5 0.18132323 0.01850505 0.077020202 0.0999596 0.087686869 0.4100505
## 6       6 0.04135193 0.06372532 0.585012876 0.1881030 0.004665236 0.1079871
## 7       7 0.04381609 0.06908046 0.510563218 0.1864195 0.007379310 0.1689080
##        Oth_MS
## 1 0.149122222
## 2 0.187711656
## 3 0.167635762
## 4 0.122525424
## 5 0.125424242
## 6 0.009081545
## 7 0.013879310
```

4. Naming Clusters

a) Black Cluster - Minimalist
b) Red Cluster - Middle Class

C) Green - Beverly Hills

5. Possible use of this Clustering scheme

a) This clustering scheme can be used to distribute public funding to areas mostly needed
b) Can be used for target marketing
c) Realtor can use this scheme to channel their clients