M4L3 Homework Assignment

Joshua Conte

October 15, 2017

1 M4L3 Homework Assignment

R studio was configured with the following parameters before beginning the project:

```
# clears the console in RStudio
cat("\014")
```

```
# clears environment
rm(list = ls())

# Load required packages
require(ggplot2)
require(cluster)
require(amap)
require(useful)
require(mclust)
```

1.1 Load Data.

I opened the Wholesale customers Data Set using read.csv2 and downloaded it directly from the UC Irvine Machine Learning Repository.

To format the data, the data is separated by ',', stringsAsFactors = FALSE so that the strings in a data frame will be treated as plain strings and not as factor variables. I set na strings for missing data. Once the data was loaded I added the column names and changed the data types to numeric and finally removed the text data type.

Below is my R code:

```
# Some csv files are really big and take a while to open.  This command checks to
# see if it is already opened, if it is, it does not open it again.
# I also omitted the first column
if (!exists("dfWCD")) {
dfWCD <-
  read.csv2("Wholesale customers data.csv",
    sep = ",",
    stringsAsFactors = FALSE,
    na.strings=c("","NA")
  )
# Add a column so I know which study the data is refereing to
study <- sprintf("study_%s",seq(1:440))
dfWCD$study<-study
}

# Download directly from site (unreliable from Ecuador)
# if (!exists("dfWCD")) {
# dfWCD <-
#   read.csv2(
#     url(
#       "https://archive.ics.uci.edu/ml/machine-learning-databases/00292/Wholesale customers data.csv"
#     ),
#     sep = ",",
#     stringsAsFactors = FALSE,
#     na.strings=c("","NA")
#   )
```

```
# # Add a column so I know which study the data is refereing to
# study <- sprintf("study_%s",seq(1:440))
# dfWCD$study<-study
# }

# change 2 to 24 to numeric
dfWCD[1:8] <- sapply(dfWCD[1:8], as.numeric)

# Print first lines
str(dfWCD)
```

```
## 'data.frame':    440 obs. of  9 variables:
##  $ Channel         : num  2 2 2 1 2 2 2 2 1 2 ...
##  $ Region          : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ Fresh           : num  12669 7057 6353 13265 22615 ...
##  $ Milk            : num  9656 9810 8808 1196 5410 ...
##  $ Grocery         : num  7561 9568 7684 4221 7198 ...
##  $ Frozen          : num  214 1762 2405 6404 3915 ...
##  $ Detergents_Paper: num  2674 3293 3516 507 1777 ...
##  $ Delicassen      : num  1338 1776 7844 1788 5185 ...
##  $ study           : chr  "study_1" "study_2" "study_3" "study_4" ...
```

```
# Select the first 8 lines for plotting
dfWCD2<-dfWCD[1:8]

# Print first lines
str(dfWCD2)
```

```
## 'data.frame':    440 obs. of  8 variables:
##  $ Channel         : num  2 2 2 1 2 2 2 2 1 2 ...
##  $ Region          : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ Fresh           : num  12669 7057 6353 13265 22615 ...
##  $ Milk            : num  9656 9810 8808 1196 5410 ...
##  $ Grocery         : num  7561 9568 7684 4221 7198 ...
##  $ Frozen          : num  214 1762 2405 6404 3915 ...
##  $ Detergents_Paper: num  2674 3293 3516 507 1777 ...
##  $ Delicassen      : num  1338 1776 7844 1788 5185 ...
```

1.2 Expectation-maximization (EM)

The expectation-maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. Expectation Maximization (EM) is perhaps most often used algorithm for unsupervised learning.

EM clustering probabilistically assigns data to different clusters. This is sometimes called "soft-clustering" (as oppossed to "hard-clustering" in which data only belongs to one cluster).

R has the Mclust function from the mclust library to provide the estimating the parameters in a statistical model using the EM algorthm.

1.2.1 Mclust Process

Given a model (a mixture Guassians, Binomial, etc.), we have the following parameters:

- X: This is a set of observed data. (Doesn't change)
- Z: This is a set of estimates for unobserved values
- T: This is a set of unknown parameters for our model

The expectation-maximization steps:

- Initialize the unknown parameters T to random values.
- Compute the best fit for the missing values Z using the existing parameter values.
- Use the best fit missing values Z to generate a better estimate for the unknown parameters T

Iterate until we have a convergence, usually when Z and T don't change much or after a fixed number of steps.

1.2.2 Mclust Example with Wholesale customers Data Set

Note that the summary command with an mclust object generates:

- log.likelihood: This is the log likelihood of the BIC value
- n: This is the number of X points
- df: This is the degrees of freedom
- BIC: This is the Bayesian information criteria; low is good
- ICL: Integrated Complete X Likelihood-a classification version of the BIC.

This data has 2 outputs, channel and region. Below, I am going to use region as my output to analyze and remove both channel and region and cluster the remaining data.

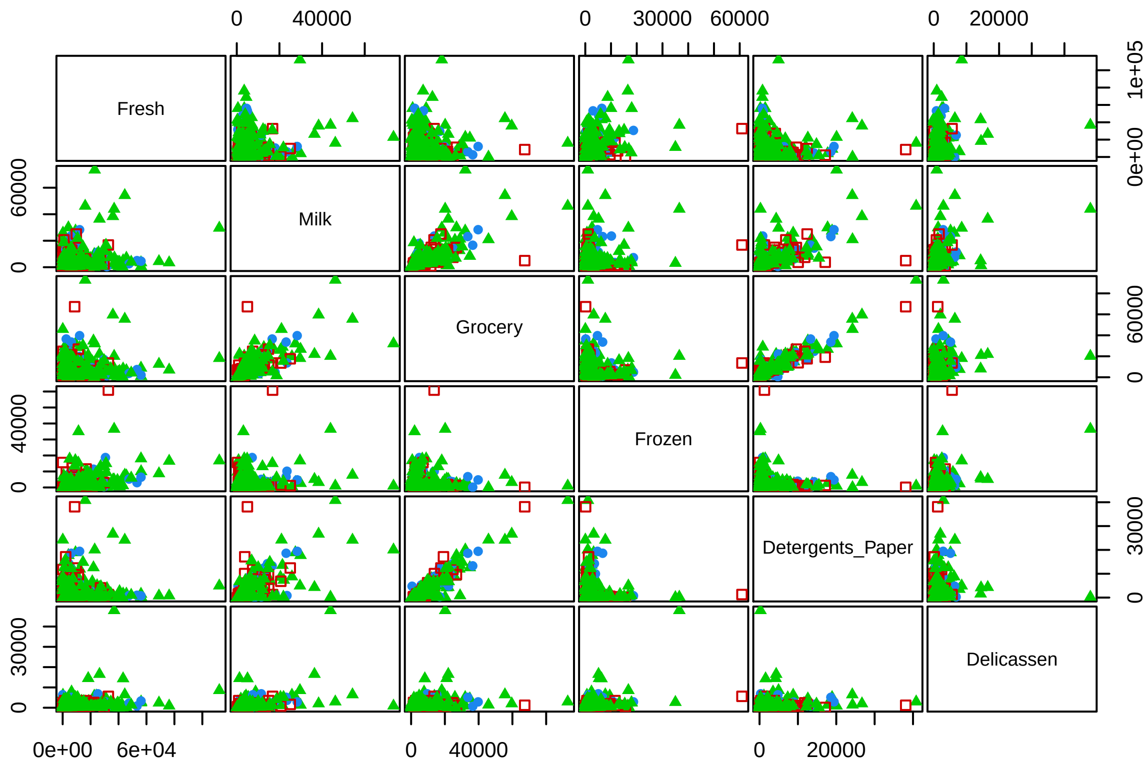
```
# I am going to use region
region.d = dfWCD2$Region
table(region.d)
```

```
## region.d
##   1   2   3
##  77  47 316
```

```
# I am going to remove region and channel from the data since they are not needed.
X = dfWCD2[3:8]
head(X)
```

```
##   Fresh Milk Grocery Frozen Detergents_Paper Delicassen
## 1 12669 9656    7561    214             2674       1338
## 2  7057 9810    9568   1762             3293       1776
## 3  6353 8808    7684   2405             3516       7844
## 4 13265 1196    4221   6404              507       1788
## 5 22615 5410    7198   3915             1777       5185
## 6  9413 8259    5126    666             1795       1451
```

```
# Cluster Plot
clPairs(X, region.d)
```




```r
# This fits the data
fit <- Mclust(X)
fit
```

```
## 'Mclust' model object:
##  best model: diagonal, varying volume and shape (VVI) with 8 components
```

```r
summary(fit)
```

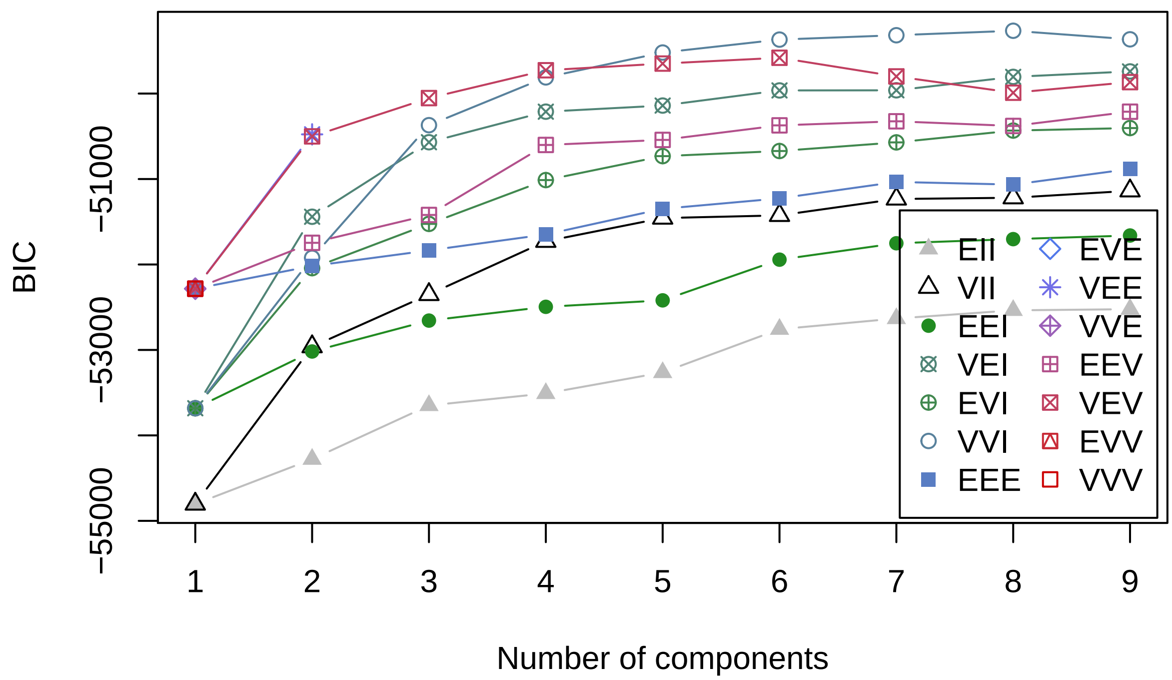
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVI (diagonal, varying volume and shape) model with 8 components:
##
##  log.likelihood   n  df       BIC       ICL
##        -24319.1 440 103 -49265.15 -49349.14
##
## Clustering table:
##   1   2   3   4   5   6   7   8
##  66  17  55  87  14  36 124  41
```

The plot command for EM produces the following four plots:

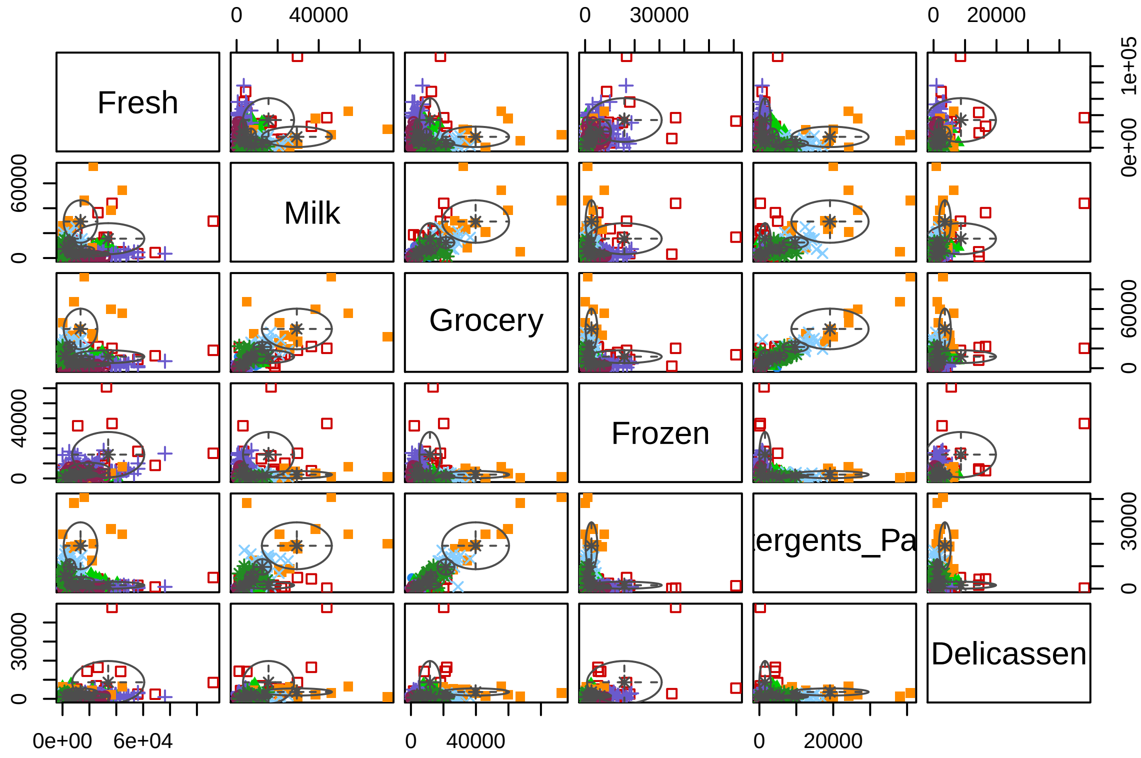
- The BIC values used for choosing the number of clusters
- A plot of the clustering

- A plot of the classification uncertainty
- The orbital plot of clusters

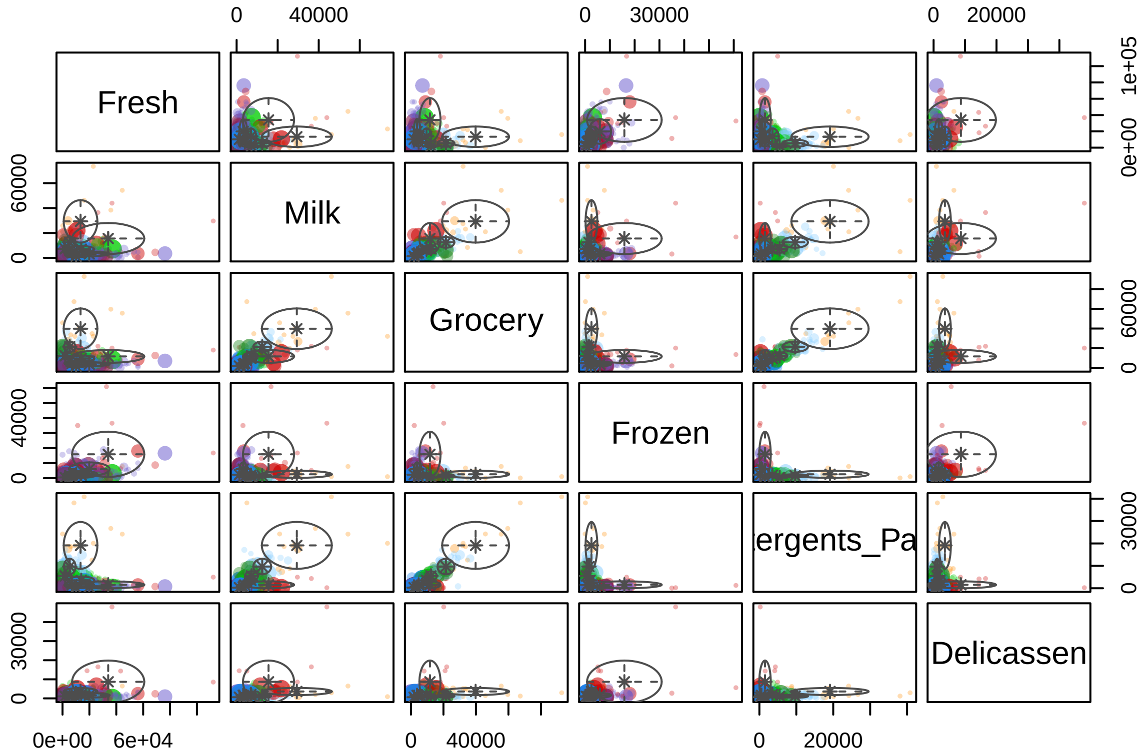
```
# 1: BIC
plot(fit, what = "BIC")
```



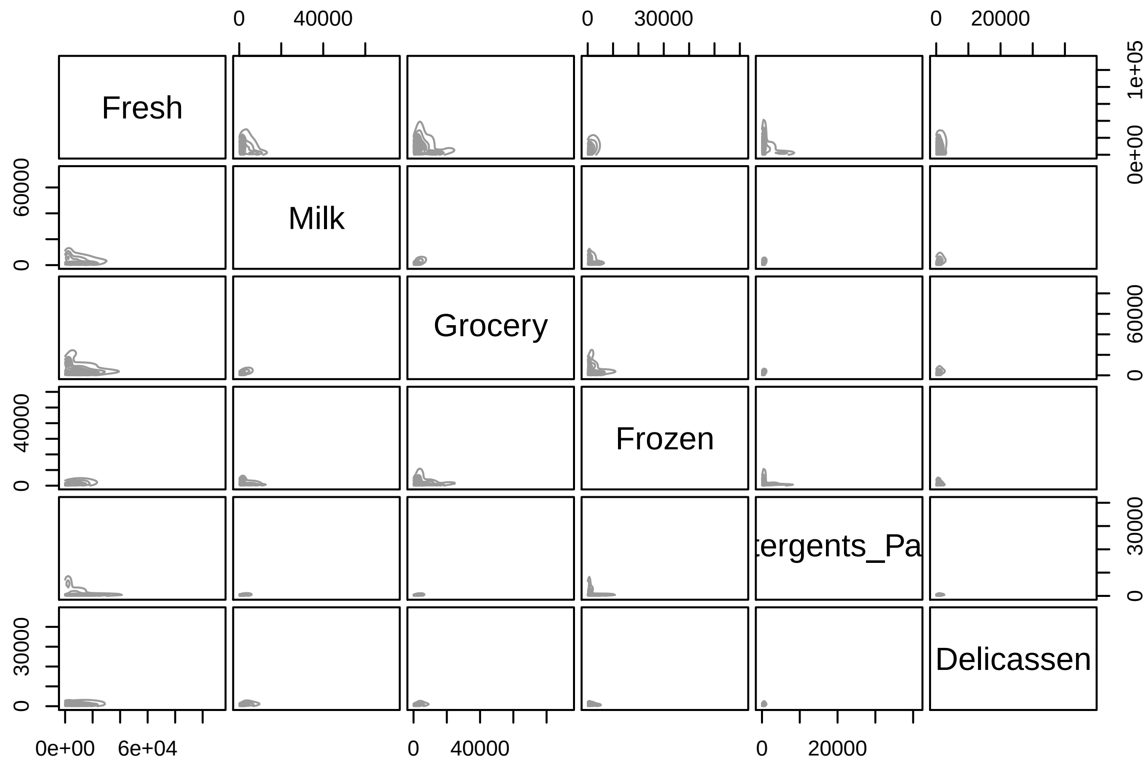

```
#table(class, fit$BIC)
#2: classification
plot(fit, what = "classification")
```




```
#table(class, fit$classification)
# 3: uncertainty
plot(fit, what = "uncertainty")
```




```
#table(class, fit$uncertainty)
# 4: density
plot(fit, what = "density")
```



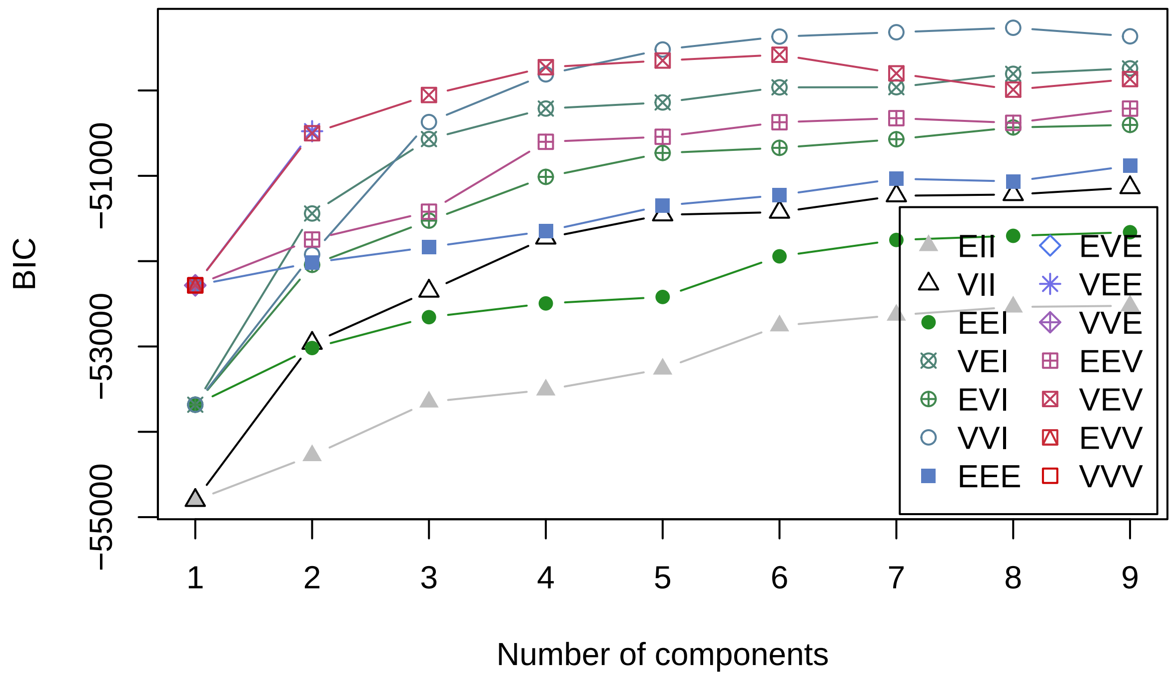

```r
#table(class, fit$density)
```

The mclustBIC from mclust uses BIC for EM initialized by model-based hierarchical clustering for parameterized Gaussian mixture models.

```r
BIC = mclustBIC(X)
summary(BIC)
```

```
## Best BIC values:
##              VVI,8        VVI,7       VVI,9
## BIC      -49265.15 -49317.38677 -49365.1714
## BIC diff      0.00    -52.24061   -100.0252
```

```r
plot(BIC) # Only BIC plot
```



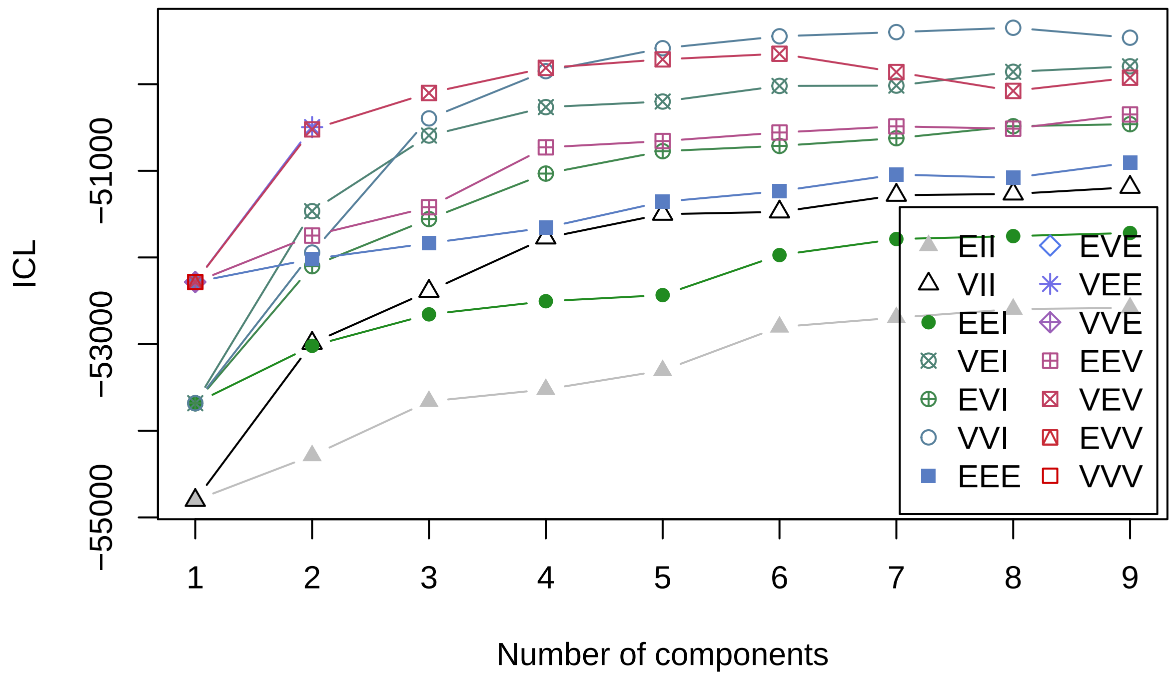

```
# plot(BIC,what = "BIC")
# The following just get BIC plot: plot(BIC,what = "classification"), plot(BIC,what = "uncertainty"), p
```

mclustICL from mclust uses ICL (Integrated Complete-data Likelihood) for parameterized Gaussian mixture models fitted by EM algorithm initialized by model-based hierarchical clustering.

```
ICL = mclustICL(X)
summary(ICL)
```

```
## Best ICL values:
##             VVI,8        VVI,7        VVI,6
## ICL      -49349.14 -49399.15142 -49447.53960
## ICL diff      0.00    -50.01335    -98.40153
```

```
plot(ICL)  # Only ICL plot
```



##M4L2 Analysis ###K-means This calculation makes the k-means calculation more stable, it performs this analysis 1000 times and takes the ones with the least error:

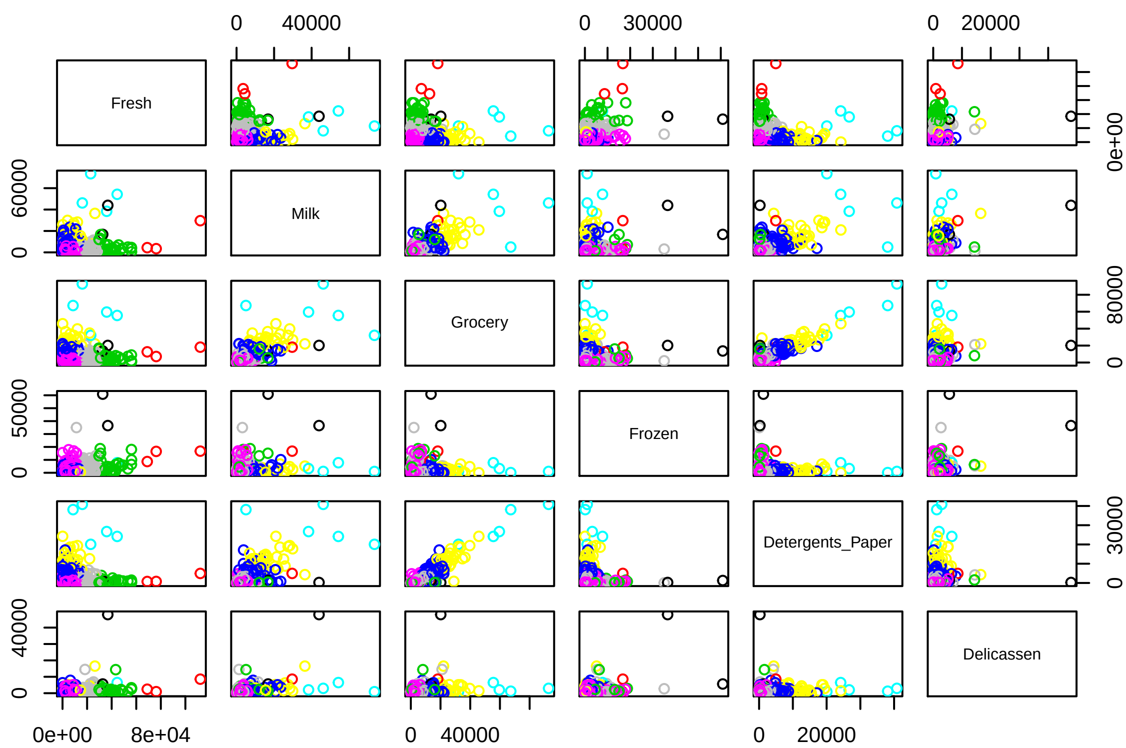
```
# Clustering with 4 clusters
k <- 8
trails<-1000
X.8.cluster <- kmeans(X,k, nstart = trails)
X.8.cluster
```

```
## K-means clustering with 8 clusters of sizes 2, 3, 32, 88, 5, 174, 30, 106
##
## Cluster means:
##       Fresh      Milk   Grocery    Frozen Detergents_Paper Delicassen
## 1 34782.000 30367.000 16898.000 48701.500          755.500 26776.0000
## 2 85779.667 12503.667 12619.667 13991.667         2159.000  3958.0000
## 3 38912.094  4767.750  5452.344  5081.000          811.625  2129.5312
## 4  4398.580  8712.114 12551.375  1445.636         5331.125  1507.0568
## 5 25603.000 43460.600 61472.200  2636.000        29974.200  2708.8000
## 6  5708.356  2463.247  3061.029  2699.379          794.454   846.1092
## 7  6683.067 17468.033 26658.933  1986.300        11872.900  2531.2000
## 8 18860.425  3423.858  4793.991  3584.934         1136.462  1585.3585
##
## Clustering vector:
##   [1] 8 4 4 8 8 6 8 4 6 4 4 8 3 8 8 6 4 6 8 6 8 6 3 7 8 8 6 8 7 3 8 6 8 3 6
##  [36] 4 3 4 4 3 8 8 4 7 4 7 7 5 4 7 6 6 3 4 8 6 7 4 8 4 6 5 6 4 6 7 6 8 6 6
##  [71] 8 8 6 8 4 8 6 7 6 6 6 4 4 8 6 5 5 3 6 8 6 8 7 8 4 6 6 6 6 6 4 4 4 3 8
## [106] 8 4 4 4 7 6 4 8 8 8 6 6 6 8 6 8 6 6 4 3 2 8 8 6 3 6 6 8 6 6 6 4 4 8 6
## [141] 8 3 3 6 8 7 6 6 6 3 8 6 8 6 6 4 4 8 4 4 4 6 8 7 4 7 4 6 6 6 4 7 6 4 6
```

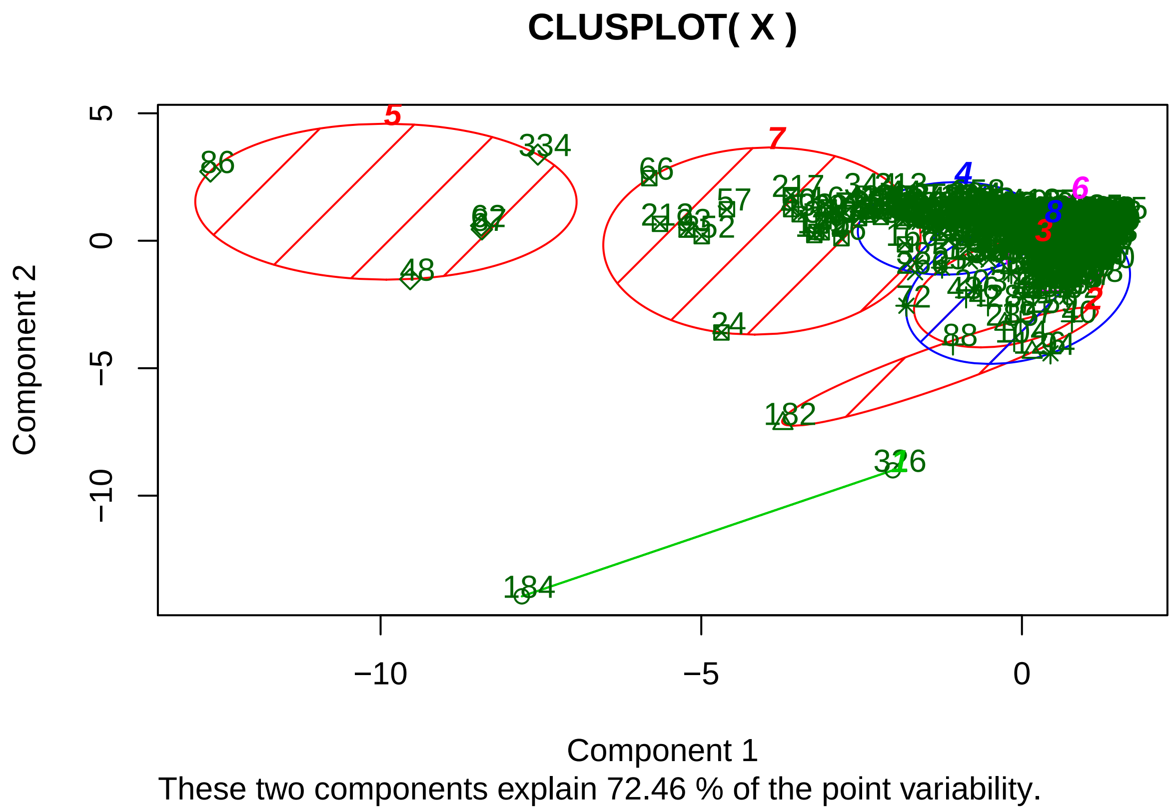
```
## [176] 4 3 8 6 6 8 2 4 1 6 6 6 4 4 4 8 8 6 4 6 8 3 4 6 6 7 7 8 6 6 7 6 6 6 4
## [211] 8 7 6 4 4 4 7 8 4 6 8 4 6 6 6 8 8 6 6 6 6 6 8 6 8 6 6 8 6 3 8 8 8 6 4
## [246] 4 6 8 8 6 6 7 6 3 4 3 6 6 3 3 6 6 8 6 4 4 4 8 4 8 6 6 4 3 6 6 8 6 6 8
## [281] 6 8 3 8 2 3 6 8 8 3 6 6 6 4 8 6 8 6 4 6 8 7 4 4 7 4 7 8 6 4 6 3 4 6 6
## [316] 4 6 6 6 7 6 6 8 8 8 1 6 6 8 6 6 7 8 5 8 8 8 6 6 6 4 4 6 7 6 6 4 8 6 7
## [351] 6 7 6 4 8 6 8 4 4 6 8 6 6 6 6 4 6 6 8 6 3 8 6 8 6 6 4 3 6 4 8 8 3 6 4
## [386] 6 6 8 6 6 6 6 6 8 6 6 4 6 6 6 6 8 8 8 8 6 8 4 6 6 6 6 4 6 6 4 4 4 4 6
## [421] 4 8 8 8 8 6 4 3 6 6 4 6 8 6 8 3 3 7 6 6
##
## Within cluster sum of squares by cluster:
## [1] 1591649631 1657529737 4300328413 5126690922 5682449098 5813885754
## [7] 5004238144 6732045153
##  (between_SS / total_SS =  77.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

These are the plots:

```
plot(X,col=X.8.cluster$cluster)      # Plot Clusters
```




```
# Centroid Plot against 1st two discriminant functions
clusplot(X, X.8.cluster$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```




```
# library(fpc)
# plotcluster(dfWCD2,dfWCD2.4.cluster$cluster)
```

1.2.3 PAM

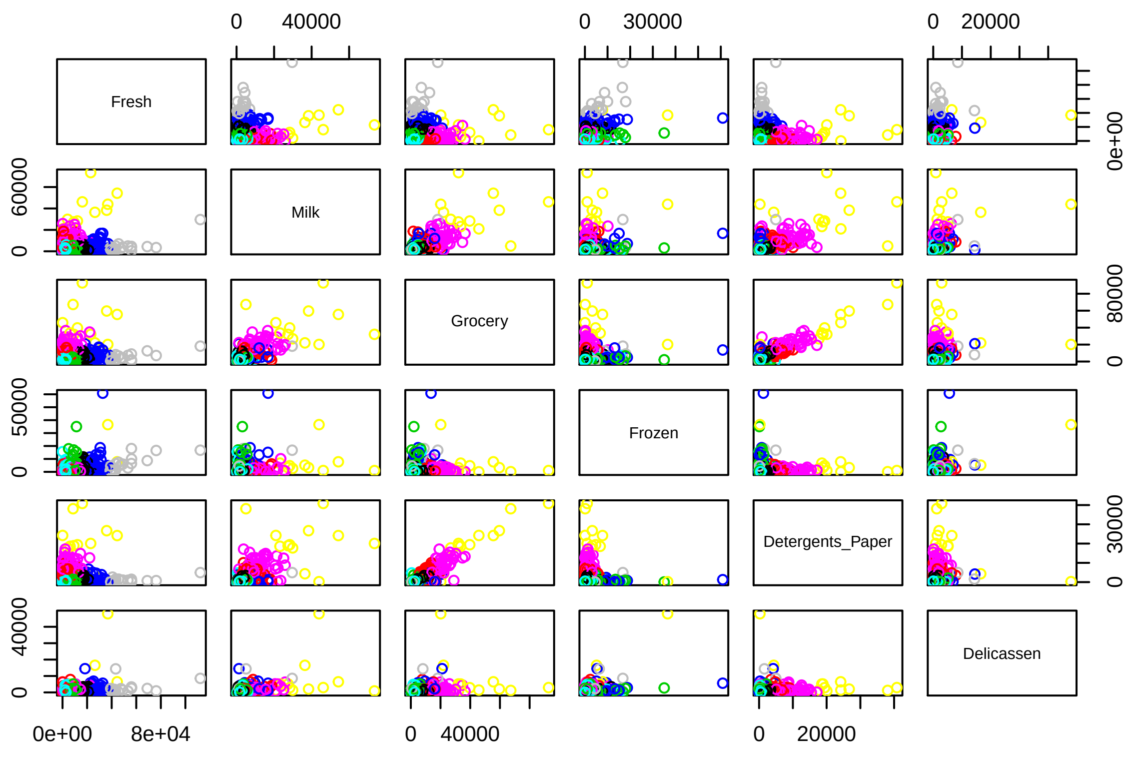
PAM Analysis:

```
# PAM
k<-8
X.pam.8.clust<- pam(X,k, keep.diss = TRUE, keep.data = TRUE)
X.pam.8.clust
```

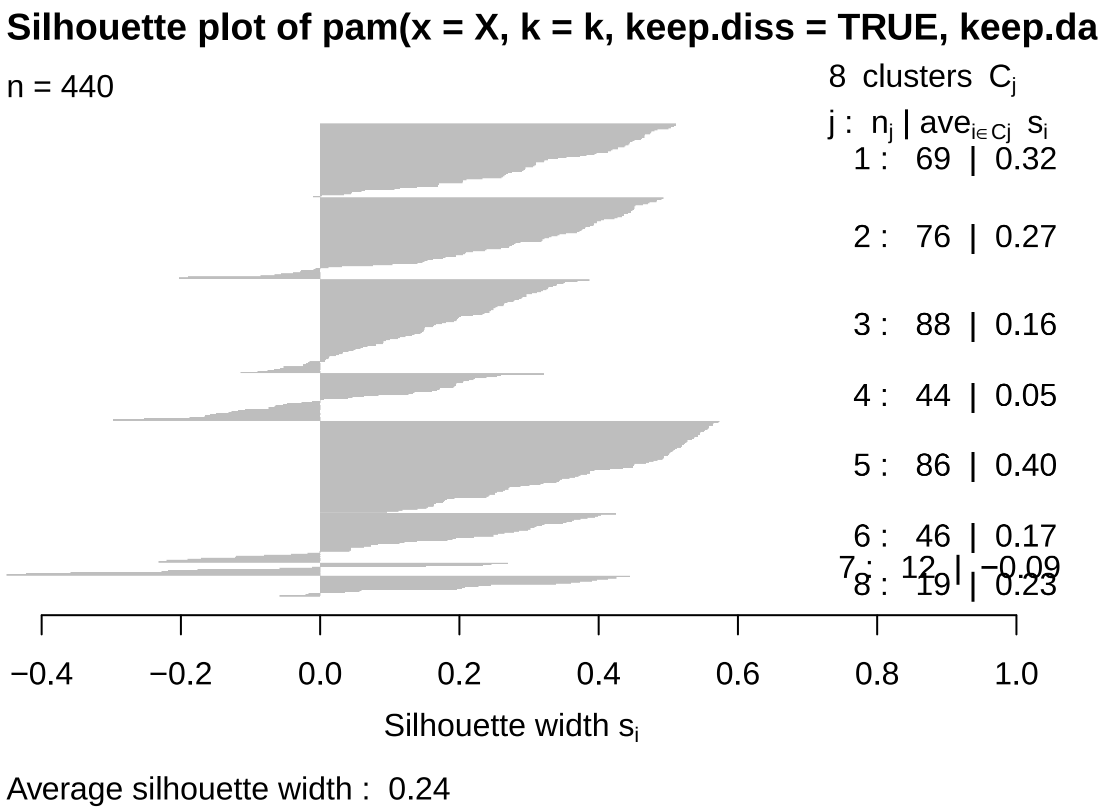
```
## Medoids:
##       ID Fresh  Milk Grocery Frozen Detergents_Paper Delicassen
## [1,] 121 17160  1200    3412   2417              174       1136
## [2,] 198  2427  7097   10391   1127             4314       1468
## [3,]  27  9898   961    2861   3151              242        833
## [4,] 277 27901  3749    6964   4479              603       2503
## [5,] 251  3191  1993    1799   1730              234        710
## [6,] 302  5283 13316   20399   1809             8752        172
## [7,] 212 12119 28326   39694   4736            19410       2870
## [8,] 240 47493  2567    3779   5243              828       2253
## Clustering vector:
##   [1] 1 2 2 3 4 3 3 2 5 6 2 3 4 4 4 3 2 5 1 2 1 5 4 7 4 1 3 1 6 8 1 5 1 4 5
##  [36] 2 4 6 6 8 4 1 6 6 2 6 6 7 2 6 3 2 8 2 4 5 7 2 1 2 3 7 2 6 5 7 2 4 5 3
##  [71] 1 4 5 1 2 1 5 6 3 5 3 2 2 1 3 7 7 8 3 4 3 3 7 3 2 5 2 5 5 3 2 6 2 8 1
```

```
## [106] 1 2 6 2 6 3 6 1 1 1 3 3 3 1 3 1 5 3 2 4 8 1 4 5 8 3 5 1 3 3 5 2 2 1 5
## [141] 1 4 4 3 1 6 3 3 5 4 1 5 1 5 5 6 2 1 2 2 2 3 1 6 2 6 2 5 5 5 2 6 2 6 5
## [176] 2 8 3 3 5 3 8 2 7 5 3 5 2 2 2 1 1 5 2 3 1 4 2 3 3 6 6 4 5 5 6 5 2 5 6
## [211] 1 7 3 2 2 2 6 1 2 5 1 2 5 5 3 3 4 5 5 3 3 2 4 5 1 5 3 1 3 8 4 4 1 3 2
## [246] 2 3 3 1 3 5 7 3 4 2 4 3 3 8 8 3 3 1 5 2 6 6 1 6 1 5 5 2 4 5 5 4 3 3 1
## [281] 5 3 8 4 8 8 3 1 1 8 5 5 5 2 1 3 1 3 2 5 1 6 2 2 6 2 6 1 3 6 3 4 6 3 3
## [316] 2 3 5 3 6 5 3 1 1 4 4 5 5 1 5 3 6 1 7 1 4 1 3 5 5 2 2 2 6 5 2 2 4 5 6
## [351] 5 6 5 6 1 5 1 6 2 5 1 5 5 5 5 2 3 5 1 5 8 1 5 1 5 5 2 8 5 2 4 1 4 5 6
## [386] 3 5 1 3 3 5 5 5 4 3 3 2 3 3 3 5 4 4 4 1 3 4 6 3 3 3 5 2 3 3 2 2 2 6 3
## [421] 2 1 4 1 1 3 6 4 5 3 2 3 1 5 1 4 8 6 3 5
## Objective function:
##    build     swap
## 7099.316 6952.244
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```r
plot(X,col=X.pam.8.clust$clustering)
```

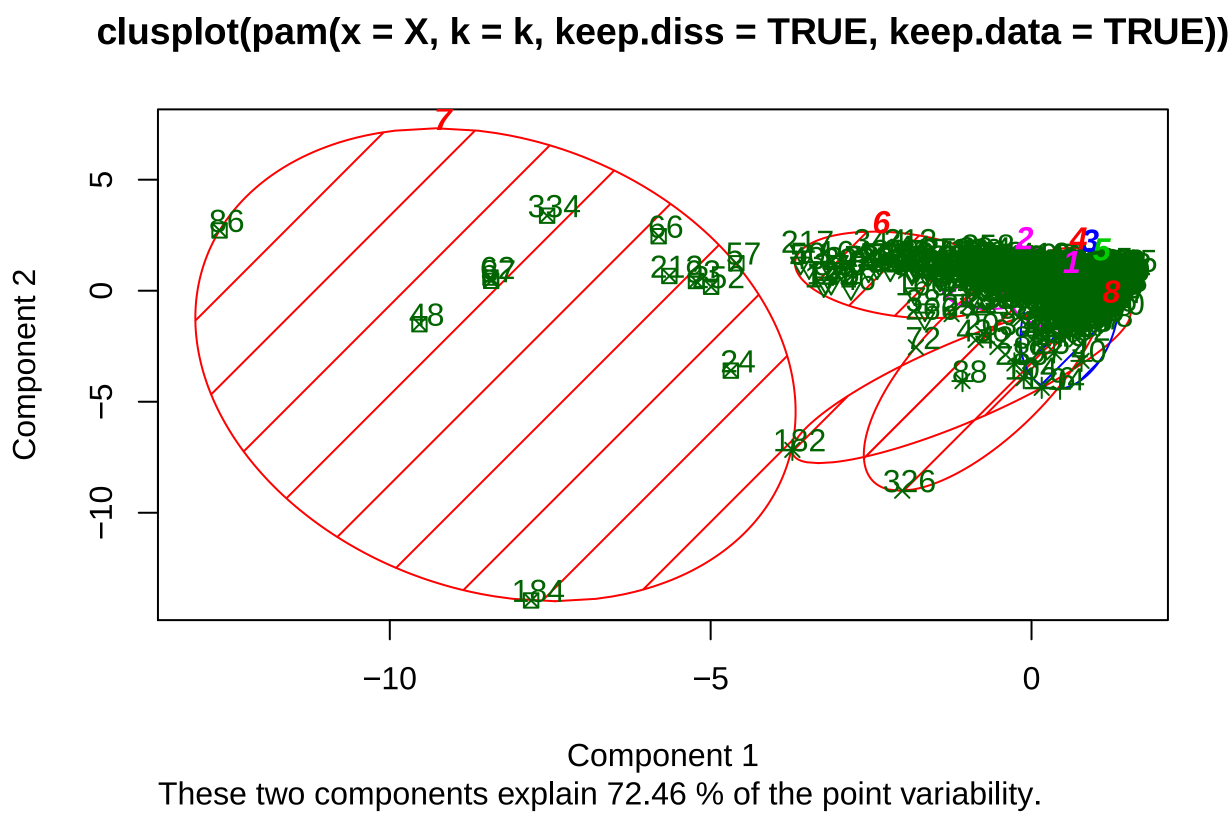



```r
plot(X.pam.8.clust, which.plots = 2)
```



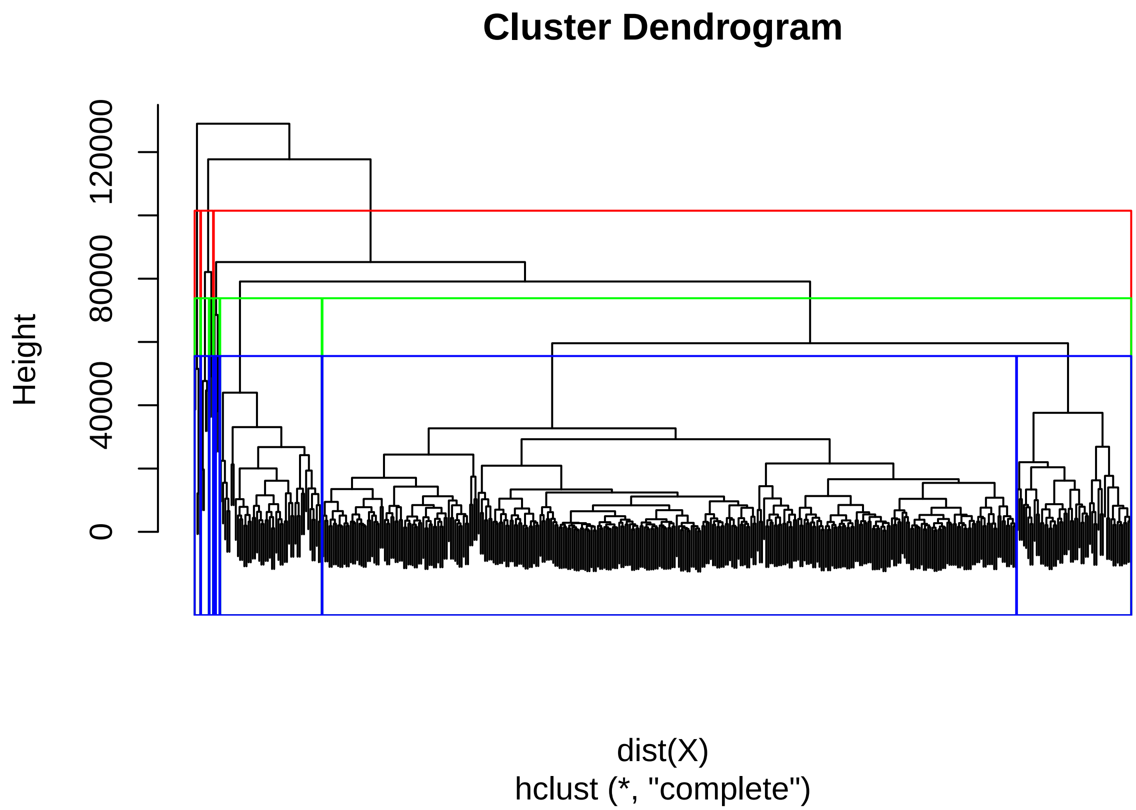

```
# long lines good - means greater within cluster similarity

# Centroid Plot against 1st two discriminant functions
clusplot(X.pam.8.clust, color=TRUE, shade=TRUE, labels=2, lines=0)
```



1.2.4 Hierarchical Clustering

```
dfWCD2.h.clust<- hclust(d=dist(X))
plot(dfWCD2.h.clust, labels = FALSE)
rect.hclust(dfWCD2.h.clust, k=3, border="red")
rect.hclust(dfWCD2.h.clust, k=6, border="green")
rect.hclust(dfWCD2.h.clust, k=8, border="blue")
```



1.3 Questions:

1. How did you choose a model for EM? Evaluate the model performance. I used the same data from the previous assignment, Wholesale Customers Data Set. To format the data, I removed the text columns and the output columns. The output data is channel and region. This left the data Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen to be analyzed.

The cluster plot (clPairs) doesn't appear to cluster very well. The green is overlapping everything in all of the plots, it doesn't seem too separated from the rest of the data. The Mclust plot ended up clustering in 8 different tables instead of 3, like I expected from the region data. The classification table looks acceptable, from what I can tell, the cylinders (probability distributions) are showing the clusters well. In the density plots, it appears there is more variance in the estimates because the density plots are so large.

2. Cluster some of your data using EM based clustering that you also used for k-means, PAM, and hierarchical clustering. How do the clustering approaches compare on the same data? When I set the clusters to 8 and re-run the M4L2 clustering, the results look similar. The k-means, pam, and EM cluster plots look identical. However, I do not think this data clusters well. Looking at the clPairs plot, the green is overlapping everything and when I look at the hierarchical clustering plot, a majority of the data falls into one cluster, even when set to 8 clusters.