HW GMM

Group F

1/22/2020

## Problem 1

## Reading in the dataset - tradeshow.csv

tds <- read.csv("/Users/shreyashiganguly/Documents/Northwestern\_MSiA/Winter 2020/Data Mining/HW2/tradeshow.csv")  
colnames(tds) <- c("buy","social","educ")

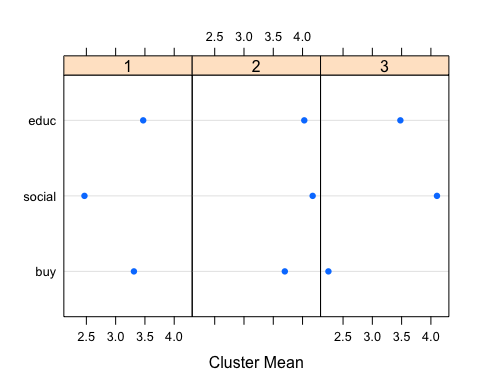
## Part(a) - KMeans clustering

set.seed(12345)  
fit.tds3 = kmeans(tds, 3, nstart=100)  
  
#Cluster sizes, means, RMSE  
summary(fit.tds3)

## n Pct buy social educ RMSE  
## 1 169 0.38 3.31 2.47 3.47 0.6076  
## 2 170 0.38 3.70 4.17 4.03 0.5578  
## 3 106 0.24 2.25 4.10 3.48 0.6534  
## 445 1.00 3.21 3.51 3.69 0.6006  
## SSE= 478.3618 ; SSB= 467.9116 ; SST= 946.2733   
## R-Squared = 0.4944782   
## Pseudo F = 216.1721

plot(fit.tds3)

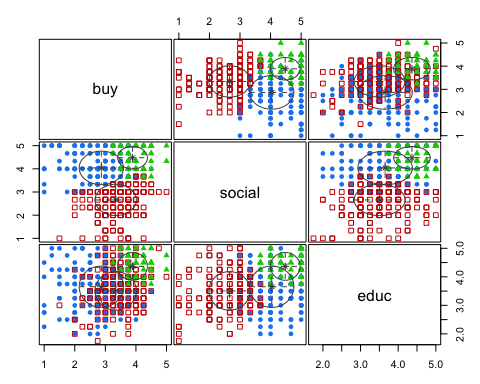
## Loading required package: lattice

 # Cluster Descriptions

* Non social - here to educate themselves and buy some
* Ambitious - here to do everything
* Non buyer - here to network and educate themselves, not to buy

## Part(b) - Gaussian Mixture (VII)

fit.tds.gmm = Mclust(tds,G=3,modelNames="VII")   
plot(fit.tds.gmm, what = "classification")



fit.tds.gmm$parameters$pro

## [1] 0.4395187 0.4265838 0.1338976

fit.tds.gmm$parameters$mean

## [,1] [,2] [,3]  
## buy 2.868500 3.339815 3.888867  
## social 4.038811 2.657831 4.478752  
## educ 3.652959 3.506621 4.365602

sqrt(fit.tds.gmm$parameters$variance$sigmasq)

## [1] 0.7245096 0.6759707 0.4863842

# Observations

* Though the three clusters have the same descriptions, there is better distinction in their values now
  + Cluster 1 - Non buyer
  + Cluster 2 - Non social
  + Cluster 3 - Ambitious
* K-means churned out almost equal sized clusters. However GMM has made the ‘Ambitious’ cluster almost one-third the size of the other two clusters. This solution makes more sense as there must be only a handful of ‘ambitious’ people, intuitively.
* Both K-Means and GMM clusters have the same ordering of within cluster variances Ambitious < Non Social < Non Buyer K-Means gives very small difference in the RMSE values, GMM depicts larger differences
* Number of variance parameters estimated = 3

## Part (c) - Gaussian Mixture (VVE)

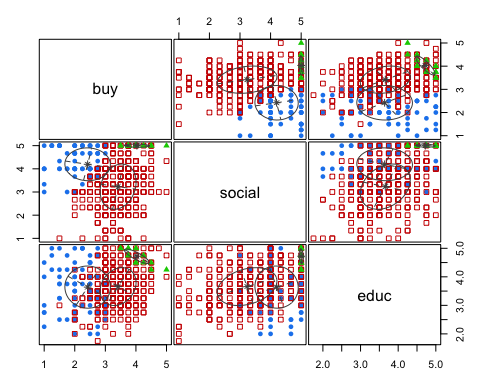
fit.tds.gmm1 = Mclust(tds,G=3)   
fit.tds.gmm1$parameters$mean

## [,1] [,2] [,3]  
## buy 2.420752 3.417206 4.030997  
## social 4.195782 3.231188 4.999999  
## educ 3.639689 3.657019 4.721012

fit.tds.gmm1$parameters$pro

## [1] 0.23084477 0.73819961 0.03095562

plot(fit.tds.gmm1, what = "classification")



fit.tds.gmm1$parameters$variance$sigma

## , , 1  
##   
## buy social educ  
## buy 0.553753113 -0.01041070 -0.006476061  
## social -0.010410703 0.50153507 -0.017107224  
## educ -0.006476061 -0.01710722 0.546659790  
##   
## , , 2  
##   
## buy social educ  
## buy 0.34510202 0.1169826 0.08611469  
## social 0.11698262 0.9540970 0.19174011  
## educ 0.08611469 0.1917401 0.44021197  
##   
## , , 3  
##   
## buy social educ  
## buy 0.167057935 -0.003870663 -0.088292319  
## social -0.003870663 0.004554370 -0.003208518  
## educ -0.088292319 -0.003208518 0.065283215

## Observations

* Though the three clusters have the same descriptions, there is better distinction in their values now
  + Cluster 1 - Non buyer
  + Cluster 2 - Non social
  + Cluster 3 - Ambitious
* Variance Model = VVE (BIC largest?) Class-conditional distributions : Variable volume, variable shape, equal orientation (classification plot?)
* Number of variance parameters estimated = 18

## Part (d) - Solution Preferred

GMM with VII model - proportionate clusters, simpler model

# Problem 2

## Part (a) - KMeans with 5 clusters

nuoqi <- read.csv("/Users/shreyashiganguly/Documents/Northwestern\_MSiA/Winter 2020/Data Mining/HW2/nuoqi.csv")  
colnames(nuoqi)

## [1] "impress" "selfexpress" "functional" "cross" "fashethus"   
## [6] "xbar" "age" "educ" "income"

nuo <- nuoqi[c("impress","selfexpress","functional","cross","fashethus")]  
set.seed(12345)  
fit.nuoqi = kmeans(nuo, 5, nstart=100)  
  
#Cluster sizes, means, RMSE  
summary(fit.nuoqi)

## n Pct impress selfexpress functional cross fashethus RMSE  
## 1 147 0.15 4.16 4.46 4.17 3.05 4.25 0.4443  
## 2 303 0.30 4.07 4.33 4.18 4.12 4.21 0.3704  
## 3 240 0.24 4.68 4.82 4.65 4.54 4.68 0.3038  
## 4 158 0.16 3.53 4.13 3.98 3.61 3.41 0.4500  
## 5 146 0.15 3.20 3.23 3.32 3.24 3.48 0.5158  
## 994 1.00 4.02 4.27 4.13 3.85 4.09 0.4056  
## SSE= 813.4389 ; SSB= 1156.095 ; SST= 1969.534   
## R-Squared = 0.5869891   
## Pseudo F = 351.4025

plot(fit.nuoqi)



## Observation

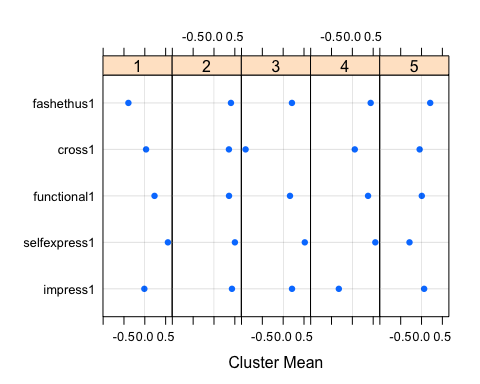
There seems to be some overlap between the clusters with three distinct clusters (non enthusiasts/average/high enthusiasts) Cluster 1 - Non cross fashion Cluster 2 - scores average on all 5 sections Cluster 3 - scores high on all 5 sections Cluster 4 - functional and self expression Cluster 5 - scores low on all 5 sections - not bothered by fashion

## Part (b) - Ipsatization

nuoqi$impress1 <- nuoqi$impress-nuoqi$xbar  
nuoqi$selfexpress1 <- nuoqi$selfexpress-nuoqi$xbar  
nuoqi$functional1 <- nuoqi$functional-nuoqi$xbar  
nuoqi$cross1 <- nuoqi$cross-nuoqi$xbar  
nuoqi$fashethus1 <- nuoqi$fashethus-nuoqi$xbar  
  
nuo1 <- nuoqi[c("impress1","selfexpress1","functional1","cross1","fashethus1")]  
set.seed(12345)  
fit.nuoqi1 = kmeans(nuo1, 5, nstart=100)  
  
#Cluster sizes, means, RMSE  
summary(fit.nuoqi1)

## n Pct impress1 selfexpress1 functional1 cross1 fashethus1 RMSE  
## 1 172 0.17 -0.01 0.55 0.23 0.03 -0.39 0.3720  
## 2 310 0.31 0.43 0.50 0.36 0.36 0.41 0.3252  
## 3 173 0.17 0.21 0.52 0.16 -0.90 0.21 0.4127  
## 4 165 0.17 -0.32 0.55 0.38 0.06 0.44 0.3510  
## 5 174 0.18 0.06 -0.29 0.01 -0.05 0.21 0.3988  
## 994 1.00 0.13 0.38 0.24 -0.04 0.20 0.3672  
## SSE= 666.8154 ; SSB= 444.9373 ; SST= 1111.753   
## R-Squared = 0.4002125   
## Pseudo F = 164.9793

plot(fit.nuoqi1)



##Observations

Still overlapping clusters Cluster 1 - Functional & SelfExpression Cluster 2 - Fashion conscious Cluster 3 - Non cross fashion Cluster 4 - Fashion enthusiast & self expression Cluster 5 - Fashion enthusiast

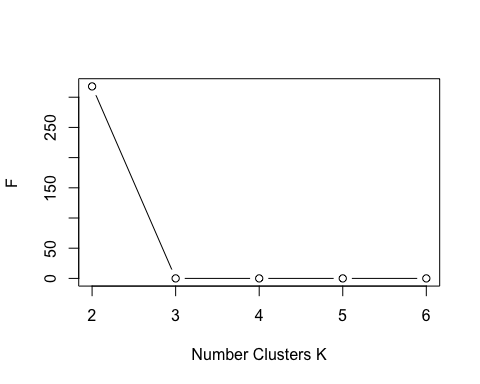
Even with ipsatization, the clustering is not clear. In fact R-square and pseudo-F both decrease.

## Part (c) - KMeans with 2-6 clusters

#Pseudo F - raw data  
set.seed(12345)  
nuo <- nuoqi[c("impress","selfexpress","functional","cross","fashethus")]  
F = double(5)  
i = 1  
for(K in 2:6)  
 set.seed(12345)  
 F[i] = summary(kmeans(nuo, K, nstart=100))$F

## n Pct impress selfexpress functional cross fashethus RMSE  
## 1 123 0.12 3.57 4.28 3.97 3.61 3.31 0.4301  
## 2 106 0.11 3.48 3.33 3.79 3.76 3.90 0.4443  
## 3 95 0.10 3.07 3.34 3.20 2.93 3.29 0.4946  
## 4 231 0.23 4.69 4.82 4.66 4.55 4.70 0.2993  
## 5 294 0.30 4.10 4.39 4.20 4.12 4.20 0.3521  
## 6 145 0.15 4.17 4.47 4.18 3.06 4.26 0.4390  
## 994 1.00 4.02 4.27 4.13 3.85 4.09 0.3909  
## SSE= 754.938 ; SSB= 1214.596 ; SST= 1969.534   
## R-Squared = 0.616692   
## Pseudo F = 317.9124

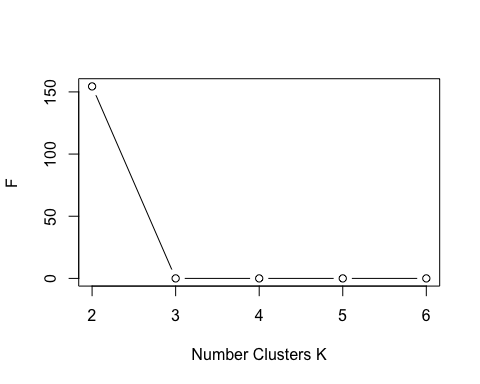
i = i+1  
plot(2:6, F, type="b", xlab="Number Clusters K")



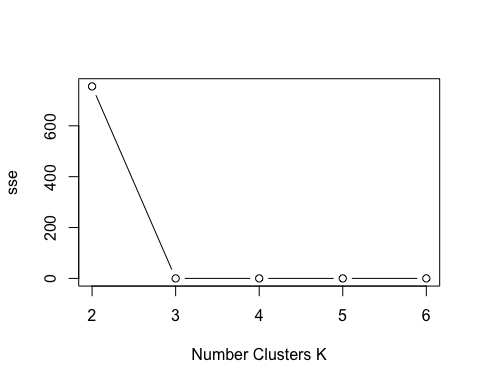
#Pseudo F - 2 clusters  
  
#Pseudo F - ipsatized data  
set.seed(12345)  
nuo1 <- nuoqi[c("impress1","selfexpress1","functional1","cross1","fashethus1")]  
F = double(5)  
i = 1  
for(K in 2:6)  
 set.seed(12345)  
 F[i] = summary(kmeans(nuo1, K, nstart=100))$F

## n Pct impress1 selfexpress1 functional1 cross1 fashethus1 RMSE  
## 1 122 0.12 0.36 0.57 0.73 0.57 0.58 0.3744  
## 2 139 0.14 -0.04 -0.35 -0.07 -0.09 0.20 0.4213  
## 3 321 0.32 0.33 0.38 0.16 0.19 0.26 0.2758  
## 4 133 0.13 -0.38 0.70 0.25 -0.13 0.41 0.3457  
## 5 136 0.14 -0.03 0.57 0.39 0.01 -0.47 0.3712  
## 6 143 0.14 0.26 0.47 0.17 -0.98 0.21 0.4165  
## 994 1.00 0.13 0.38 0.24 -0.04 0.20 0.3554  
## SSE= 624.0503 ; SSB= 487.7024 ; SST= 1111.753   
## R-Squared = 0.4386788   
## Pseudo F = 154.4266

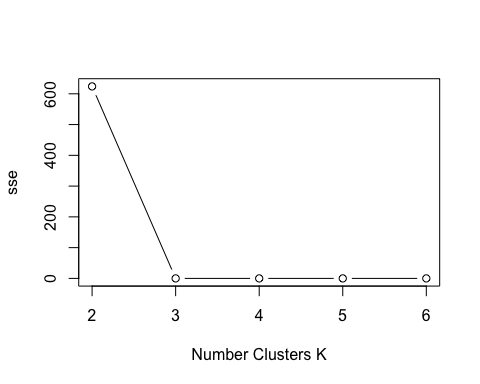
i = i+1  
plot(2:6, F, type="b", xlab="Number Clusters K")



#Pseudo F - 2 clusters  
  
#SSE - raw data  
sse = double(5)  
i = 1  
for(K in 2:6)  
 set.seed(12345)  
 fit = kmeans(nuo, K, nstart=100)  
 sse[i] = fit$tot.withinss  
 i = i+1  
plot(2:6, sse, type="b", xlab="Number Clusters K")



#SSE suggests 3 clusters  
  
#SSE - ipsatized data  
sse = double(5)  
i = 1  
for(K in 2:6)  
 set.seed(12345)  
 fit = kmeans(nuo1, K, nstart=100)  
 sse[i] = fit$tot.withinss  
 i = i+1  
plot(2:6, sse, type="b", xlab="Number Clusters K")

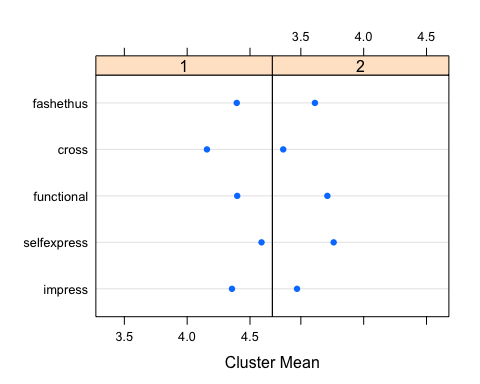


#SSE suggests 3 clusters

#K-Means : 2 clusters  
set.seed(12345)  
fit.nuo2 = kmeans(nuo, 2, nstart=100)  
summary(fit.nuo2)

## n Pct impress selfexpress functional cross fashethus RMSE  
## 1 613 0.62 4.36 4.59 4.40 4.16 4.39 0.4416  
## 2 381 0.38 3.47 3.76 3.71 3.36 3.61 0.5718  
## 994 1.00 4.02 4.27 4.13 3.85 4.09 0.4955  
## SSE= 1217.932 ; SSB= 751.6017 ; SST= 1969.534   
## R-Squared = 0.381614   
## Pseudo F = 612.1761

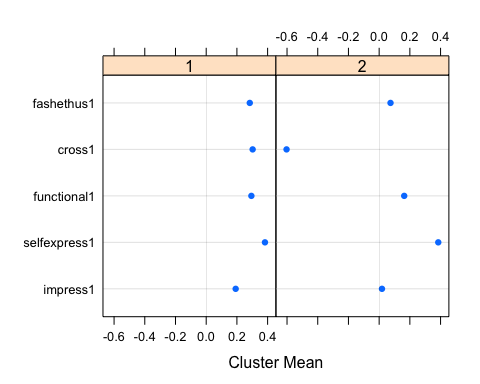
plot(fit.nuo2)



#Cluster 1 - highly fashion conscious  
#Cluster 2 - not fashion conscious  
  
fit.nuo21 = kmeans(nuo1, 2, nstart=100)  
summary(fit.nuo21)

## n Pct impress1 selfexpress1 functional1 cross1 fashethus1 RMSE  
## 1 620 0.62 0.19 0.38 0.29 0.30 0.28 0.3933  
## 2 374 0.38 0.02 0.38 0.16 -0.60 0.07 0.4748  
## 994 1.00 0.13 0.38 0.24 -0.04 0.20 0.4258  
## SSE= 899.1399 ; SSB= 212.6127 ; SST= 1111.753   
## R-Squared = 0.191241   
## Pseudo F = 234.5707

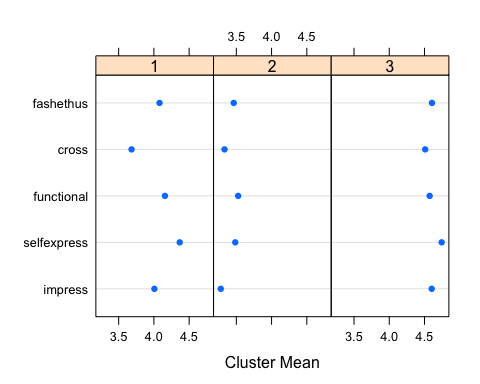
plot(fit.nuo21)



#Confusing clusters  
  
#K-Means : 3 clusters  
set.seed(12345)  
fit.nuo3 = kmeans(nuo, 3, nstart=100)  
summary(fit.nuo3)

## n Pct impress selfexpress functional cross fashethus RMSE  
## 1 458 0.46 4.01 4.37 4.16 3.68 4.08 0.4606  
## 2 234 0.24 3.28 3.48 3.53 3.33 3.46 0.5424  
## 3 302 0.30 4.60 4.74 4.57 4.51 4.61 0.3359  
## 994 1.00 4.02 4.27 4.13 3.85 4.09 0.4486  
## SSE= 997.3443 ; SSB= 972.1895 ; SST= 1969.534   
## R-Squared = 0.493614   
## Pseudo F = 483.0026

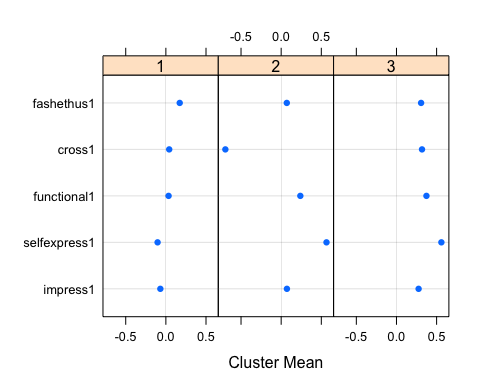
plot(fit.nuo3)



#Cluster 1 - average  
#Cluster 2 - low  
#Cluster 3 - high  
  
fit.nuo31 = kmeans(nuo1, 3, nstart=100)  
summary(fit.nuo31)

## n Pct impress1 selfexpress1 functional1 cross1 fashethus1 RMSE  
## 1 266 0.27 -0.07 -0.10 0.03 0.04 0.17 0.4245  
## 2 277 0.28 0.07 0.56 0.24 -0.70 0.07 0.4451  
## 3 451 0.45 0.28 0.56 0.37 0.32 0.31 0.3557  
## 994 1.00 0.13 0.38 0.24 -0.04 0.20 0.4010  
## SSE= 796.8534 ; SSB= 314.8993 ; SST= 1111.753   
## R-Squared = 0.2832458   
## Pseudo F = 195.8109

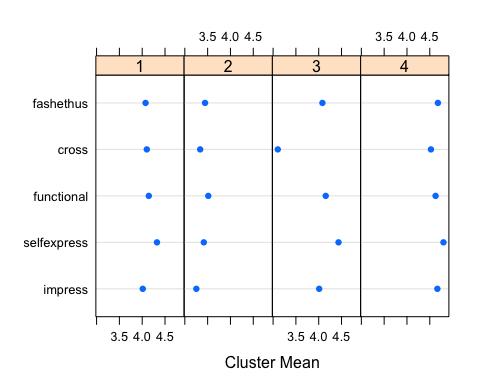
plot(fit.nuo31)



#Confusing clusters  
  
#K-Means : 4 clusters  
set.seed(12345)  
fit.nuo4 = kmeans(nuo, 4, nstart=100)  
summary(fit.nuo4)

## n Pct impress selfexpress functional cross fashethus RMSE  
## 1 337 0.34 4.01 4.32 4.14 4.10 4.07 0.4040  
## 2 211 0.21 3.25 3.42 3.51 3.34 3.44 0.5218  
## 3 194 0.20 4.01 4.43 4.15 3.10 4.08 0.4671  
## 4 252 0.25 4.67 4.80 4.63 4.53 4.68 0.3092  
## 994 1.00 4.02 4.27 4.13 3.85 4.09 0.4242  
## SSE= 890.6949 ; SSB= 1078.839 ; SST= 1969.534   
## R-Squared = 0.5477636   
## Pseudo F = 399.7068

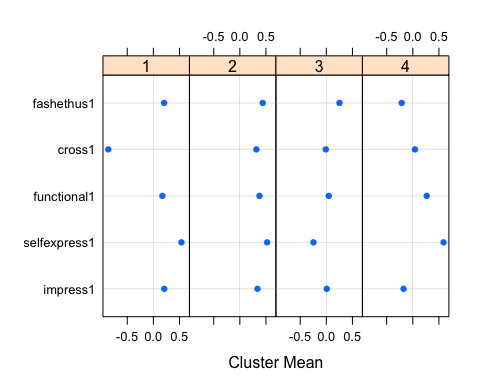
plot(fit.nuo4)



#Cluster 1 - average  
#Cluster 2 - low  
#Cluster 3 - high on all except cross  
#Cluster 4 - high  
  
fit.nuo41 = kmeans(nuo1, 4, nstart=100)  
summary(fit.nuo41)

## n Pct impress1 selfexpress1 functional1 cross1 fashethus1 RMSE  
## 1 189 0.19 0.21 0.54 0.17 -0.86 0.21 0.4168  
## 2 371 0.37 0.34 0.52 0.38 0.32 0.44 0.3349  
## 3 203 0.20 0.01 -0.25 0.05 -0.01 0.25 0.4098  
## 4 231 0.23 -0.18 0.59 0.27 0.04 -0.21 0.3890  
## 994 1.00 0.13 0.38 0.24 -0.04 0.20 0.3799  
## SSE= 714.3881 ; SSB= 397.3645 ; SST= 1111.753   
## R-Squared = 0.3574217   
## Pseudo F = 183.5561

plot(fit.nuo41)



#Confusing clusters - stark distinction on only few attributes  
  
#K-Means : 5 clusters  
set.seed(12345)  
fit.nuo5 = kmeans(nuo, 5, nstart=100)  
summary(fit.nuo5)

## n Pct impress selfexpress functional cross fashethus RMSE  
## 1 147 0.15 4.16 4.46 4.17 3.05 4.25 0.4443  
## 2 303 0.30 4.07 4.33 4.18 4.12 4.21 0.3704  
## 3 240 0.24 4.68 4.82 4.65 4.54 4.68 0.3038  
## 4 158 0.16 3.53 4.13 3.98 3.61 3.41 0.4500  
## 5 146 0.15 3.20 3.23 3.32 3.24 3.48 0.5158  
## 994 1.00 4.02 4.27 4.13 3.85 4.09 0.4056  
## SSE= 813.4389 ; SSB= 1156.095 ; SST= 1969.534   
## R-Squared = 0.5869891   
## Pseudo F = 351.4025

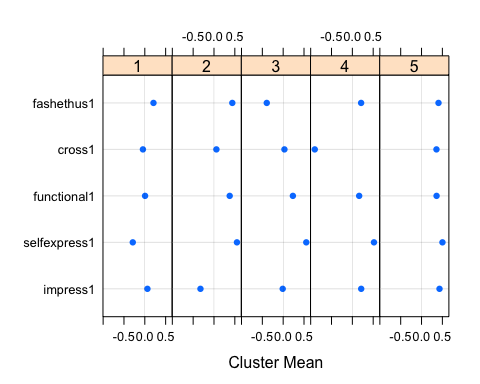
plot(fit.nuo5)



#Cluster 1 - average & low on cross  
#Cluster 2 - average  
#Cluster 3 - high on all  
#Cluster 4 - high on functional and selfexpress  
#Cluster 5 - low  
  
fit.nuo51 = kmeans(nuo1, 5, nstart=100)  
summary(fit.nuo51)

## n Pct impress1 selfexpress1 functional1 cross1 fashethus1 RMSE  
## 1 174 0.18 0.06 -0.29 0.01 -0.05 0.21 0.3988  
## 2 165 0.17 -0.32 0.55 0.38 0.06 0.44 0.3510  
## 3 172 0.17 -0.01 0.55 0.23 0.03 -0.39 0.3720  
## 4 173 0.17 0.21 0.52 0.16 -0.90 0.21 0.4127  
## 5 310 0.31 0.43 0.50 0.36 0.36 0.41 0.3252  
## 994 1.00 0.13 0.38 0.24 -0.04 0.20 0.3672  
## SSE= 666.8154 ; SSB= 444.9373 ; SST= 1111.753   
## R-Squared = 0.4002125   
## Pseudo F = 164.9793

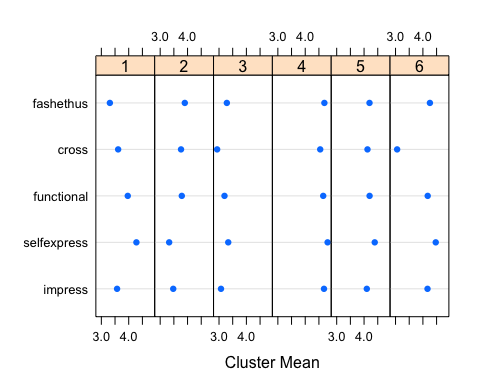
plot(fit.nuo51)



#Confusing clusters - stark distinction on only few attributes  
  
#K-Means : 6 clusters  
set.seed(12345)  
fit.nuo6 = kmeans(nuo, 6, nstart=100)  
summary(fit.nuo6)

## n Pct impress selfexpress functional cross fashethus RMSE  
## 1 123 0.12 3.57 4.28 3.97 3.61 3.31 0.4301  
## 2 106 0.11 3.48 3.33 3.79 3.76 3.90 0.4443  
## 3 95 0.10 3.07 3.34 3.20 2.93 3.29 0.4946  
## 4 231 0.23 4.69 4.82 4.66 4.55 4.70 0.2993  
## 5 294 0.30 4.10 4.39 4.20 4.12 4.20 0.3521  
## 6 145 0.15 4.17 4.47 4.18 3.06 4.26 0.4390  
## 994 1.00 4.02 4.27 4.13 3.85 4.09 0.3909  
## SSE= 754.938 ; SSB= 1214.596 ; SST= 1969.534   
## R-Squared = 0.616692   
## Pseudo F = 317.9124

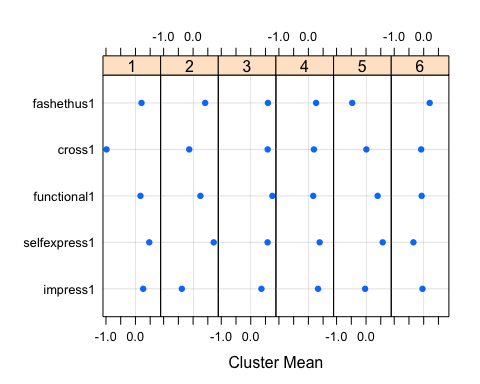
plot(fit.nuo6)



#Cluster 1 - average & low on cross  
#Cluster 2 -   
#Cluster 3 - low on all  
#Cluster 4 - high on all  
#Cluster 5 - average  
#Cluster 6 -   
  
fit.nuo61 = kmeans(nuo1, 6, nstart=100)  
summary(fit.nuo61)

## n Pct impress1 selfexpress1 functional1 cross1 fashethus1 RMSE  
## 1 143 0.14 0.26 0.47 0.17 -0.98 0.21 0.4165  
## 2 133 0.13 -0.38 0.70 0.25 -0.13 0.41 0.3457  
## 3 122 0.12 0.36 0.57 0.73 0.57 0.58 0.3744  
## 4 321 0.32 0.33 0.38 0.16 0.19 0.26 0.2758  
## 5 136 0.14 -0.03 0.57 0.39 0.01 -0.47 0.3712  
## 6 139 0.14 -0.04 -0.35 -0.07 -0.09 0.20 0.4213  
## 994 1.00 0.13 0.38 0.24 -0.04 0.20 0.3554  
## SSE= 624.0503 ; SSB= 487.7024 ; SST= 1111.753   
## R-Squared = 0.4386788   
## Pseudo F = 154.4266

plot(fit.nuo61)



#Confusing clusters - stark distinction on only few attributes  
  
#Recommendation - 3 clusters with raw data

## Recommendation

3 clusters with raw data

## Part (d) - GMM

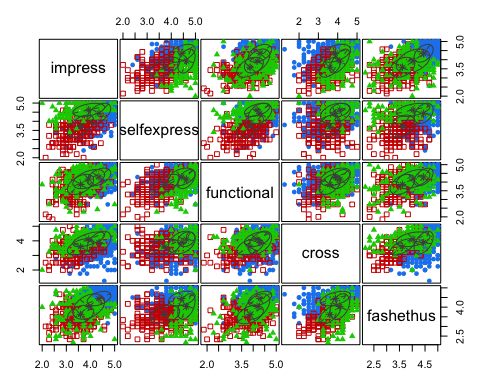
fit.nuo.gmm = Mclust(nuo,G=3)   
fit.nuo.gmm$parameters$mean

## [,1] [,2] [,3]  
## impress 4.350528 3.587571 3.926259  
## selfexpress 4.435210 3.730724 4.626003  
## functional 4.294184 3.864028 4.169933  
## cross 3.882437 3.588585 4.107619  
## fashethus 4.445661 3.615950 4.033125

fit.nuo.gmm$parameters$pro

## [1] 0.4504905 0.2977975 0.2517120

plot(fit.nuo.gmm, what = "classification")



## Observations

Cluster 1 - fashion enthusiasts, impressive, functional Cluster 2 - not enthusiastic Cluster 3 - selfexpression, cross

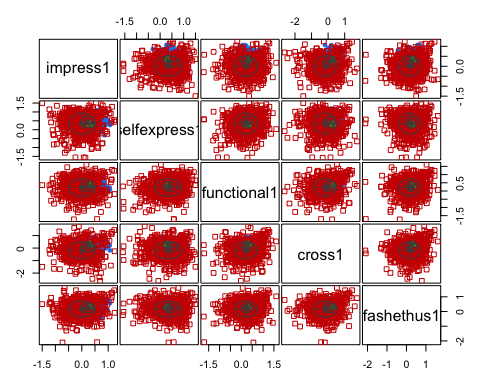
fit.nuo.gmm1 = Mclust(nuo1,G=3)   
fit.nuo.gmm1$parameters$mean

## [,1] [,2] [,3]  
## impress1 0.23622066 0.008213263 0.3087305  
## selfexpress1 0.42486042 0.343125766 0.3087305  
## functional1 0.27387143 0.213234610 0.3087305  
## cross1 0.08377218 -0.174377955 0.3087305  
## fashethus1 0.23892462 0.166699900 0.3087305

fit.nuo.gmm1$parameters$pro

## [1] 0.50100031 0.48491568 0.01408401

plot(fit.nuo.gmm1, what = "classification")



## Observation

Confused cluster solution - last cluster too small

## Underlying problem

cor(nuo)

## impress selfexpress functional cross fashethus  
## impress 1.0000000 0.5453501 0.5265643 0.4621602 0.5716577  
## selfexpress 0.5453501 1.0000000 0.5316097 0.3783569 0.4719896  
## functional 0.5265643 0.5316097 1.0000000 0.4505158 0.5120007  
## cross 0.4621602 0.3783569 0.4505158 1.0000000 0.4547321  
## fashethus 0.5716577 0.4719896 0.5120007 0.4547321 1.0000000

summary(nuoqi)

## impress selfexpress functional cross   
## Min. :2.000 Min. :2.000 Min. :1.857 Min. :1.250   
## 1st Qu.:3.571 1st Qu.:4.000 1st Qu.:3.857 1st Qu.:3.500   
## Median :4.000 Median :4.400 Median :4.286 Median :4.000   
## Mean :4.017 Mean :4.273 Mean :4.135 Mean :3.852   
## 3rd Qu.:4.429 3rd Qu.:4.800 3rd Qu.:4.571 3rd Qu.:4.250   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
## fashethus xbar age educ   
## Min. :2.167 Min. :2.356 Min. : 1.000 Min. :1.000   
## 1st Qu.:3.667 1st Qu.:3.633 1st Qu.: 3.000 1st Qu.:3.000   
## Median :4.167 Median :3.911 Median : 4.000 Median :3.000   
## Mean :4.095 Mean :3.890 Mean : 4.213 Mean :3.285   
## 3rd Qu.:4.500 3rd Qu.:4.167 3rd Qu.: 5.000 3rd Qu.:4.000   
## Max. :5.000 Max. :4.767 Max. :13.000 Max. :6.000   
## income impress1 selfexpress1 functional1   
## Min. :1.000 Min. :-1.5000 Min. :-1.5778 Min. :-1.72540   
## 1st Qu.:2.000 1st Qu.:-0.1329 1st Qu.: 0.1333 1st Qu.:-0.01071   
## Median :3.000 Median : 0.1825 Median : 0.4111 Median : 0.27857   
## Mean :3.232 Mean : 0.1267 Mean : 0.3836 Mean : 0.24496   
## 3rd Qu.:5.000 3rd Qu.: 0.4028 3rd Qu.: 0.7111 3rd Qu.: 0.51389   
## Max. :5.000 Max. : 1.2444 Max. : 1.5000 Max. : 1.60476   
## cross1 fashethus1   
## Min. :-2.62778 Min. :-2.11111   
## 1st Qu.:-0.35000 1st Qu.:-0.04444   
## Median : 0.03333 Median : 0.23333   
## Mean :-0.03824 Mean : 0.20488   
## 3rd Qu.: 0.33333 3rd Qu.: 0.47778   
## Max. : 1.74444 Max. : 1.61111

cov(nuo)

## impress selfexpress functional cross fashethus  
## impress 0.4084858 0.2239324 0.1918425 0.2023184 0.2216619  
## selfexpress 0.2239324 0.4127682 0.1946932 0.1664980 0.1839721  
## functional 0.1918425 0.1946932 0.3249449 0.1759013 0.1770688  
## cross 0.2023184 0.1664980 0.1759013 0.4691464 0.1889628  
## fashethus 0.2216619 0.1839721 0.1770688 0.1889628 0.3680723

There is not much variation in the answers.

library(Rtsne)  
## Curating the database for analysis with both t-SNE and PCA  
Labels<-as.factor(fit.nuo3$cluster)  
## for plotting  
colors = rainbow(length(unique(Labels)))  
names(colors) = unique(Labels)  
  
## Executing the algorithm on curated data  
set.seed(12345)  
tsne<- Rtsne(nuo, dims = 2, perplexity=30, verbose=TRUE, max\_iter = 500, check\_duplicates=FALSE)

## Performing PCA  
## Read the 994 x 5 data matrix successfully!  
## OpenMP is working. 1 threads.  
## Using no\_dims = 2, perplexity = 30.000000, and theta = 0.500000  
## Computing input similarities...  
## Building tree...  
## Done in 0.08 seconds (sparsity = 0.122031)!  
## Learning embedding...  
## Iteration 50: error is 68.932887 (50 iterations in 0.12 seconds)  
## Iteration 100: error is 65.624282 (50 iterations in 0.10 seconds)  
## Iteration 150: error is 65.623435 (50 iterations in 0.09 seconds)  
## Iteration 200: error is 65.623877 (50 iterations in 0.09 seconds)  
## Iteration 250: error is 65.623949 (50 iterations in 0.09 seconds)  
## Iteration 300: error is 1.402034 (50 iterations in 0.10 seconds)  
## Iteration 350: error is 1.257805 (50 iterations in 0.10 seconds)  
## Iteration 400: error is 1.212500 (50 iterations in 0.10 seconds)  
## Iteration 450: error is 1.196012 (50 iterations in 0.10 seconds)  
## Iteration 500: error is 1.184320 (50 iterations in 0.10 seconds)  
## Fitting performed in 0.99 seconds.

## Plotting  
plot(tsne$Y, t='n', main="tsne")  
text(tsne$Y, labels=Labels, col=colors[Labels])

