```
################################################################################
# Project:      Credit Scoring Analysis (petit example)
# Description:  Part 5 - Cluster Analysis
# Data:         CleanCreditScoring.csv
# By:           Gaston Sanchez
# url:          www.gastonsanchez.com
#
# Perform a cluster analysis on the results obtained
# from the multiple correspondence analysis
#
# Note:
#     Check Part 4 - Multiple Correspondence Analysis
#     Part4_CredScoring_MCA.R
################################################################################

# remember to change your working directory!!! (don't use mine)
# setwd("/Users/gaston/Documents/Gaston/StatsDataMining")

# load package FactoMineR and ggplot2
require(FactoMineR)
require(ggplot2)

# read cleaned data set
dd = read.csv("CleanCreditScoring.csv", header=TRUE, stringsAsFactors=TRUE)


# ============================================================================
# Apply MCA on categorical variables
# ============================================================================

# select select categorized continuous variables
ddcat = subset(dd, select=c(seniorityR, timeR, ageR, expensesR, incomeR,
    assetsR, debtR, amountR, priceR, finratR, savingsR, Status))

# MCA
mca = MCA(ddcat, ncp=40, quali.sup=12, graph=FALSE)

# get the eigenvalues
eigs = mca$eig$eigenvalue

# calculate significant dimension in MCA
nd = sum(eigs>1/length(eigs))

# get factorial coordinates of individuals
Psi = mca$ind$coord


# ============================================================================
# K-means clustering on factorical coordinates from MCA results
# ============================================================================
# The idea is to take the extracted dimensions (i.e. the factorial cooridnates)
# from the MCA in order to perform a k-means cluster analysis on them

# The first approach is to apply K-means on factorial coordinates
# Let's try k=5 groups
k1 = kmeans(Psi, 5)
# what does k1 contain?
attributes(k1)
# examine the following
k1$size       # size of clusters
k1$withinss   # within cluster variance
k1$centers    # centers coordinates

# between clusters sum of squares
```

```
Bss = sum(rowSums(k1$centers^2) * k1$size)
Bss
# within clusters sum of squares
Wss = sum(k1$withinss)
Wss
# total sum of squares
Tss = sum(rowSums(Psi^2))
Tss
Bss + Wss
# let's calculate the decomposition of inertia
Ib1 = 100 * Bss / (Bss + Wss)
Ib1

# let's repeat kmeans, again with k=5
k2 = kmeans(Psi, 5)
# between clusters sum of squares
Bss = sum(rowSums(k2$centers^2) * k2$size)
Wss = sum(k2$withinss)
# total sum of squares
Tss = sum(rowSums(Psi^2))
Bss + Wss    # Tss = Bss + Wss
# let's calculate the decomposition of inertia
Ib2 = 100 * Bss / (Bss + Wss)
Ib2
# why are we obtaining different results?  (Ib1 != Ib2)
# you can keep playing with different values for k

# let's repeat kmeans, again with k=8
k3 = kmeans(Psi, 8)
# between clusters sum of squares
Bss = sum(rowSums(k3$centers^2) * k3$size)
Wss = sum(k3$withinss)
# total sum of squares
Tss = sum(rowSums(Psi^2))
Bss + Wss    # Tss = Bss + Wss
# let's calculate the decomposition of inertia
Ib3 = 100 * Bss / (Bss + Wss)
Ib3


# ============================================================================
# Hierarchical clustering on factorical coordinates from MCA results
# ============================================================================

# Now, let's apply a hierarchical clustering
# first we calculate a distance matrix between individuals
idist = dist(Psi)

# then we apply hclust with method="ward"
# notice the computation cost! (it takes a while to finish)
h1 = hclust(idist, method="ward")

# plot dendrogram
plot(h1, labels=FALSE)

# after checking the dendrogram, how many groups would you choose?
# where would you cut the dendrogram?
# let's try 8 clusters
nc = 8
# let's cut the tree and see the size of clusters
c1 = cutree(h1, nc)
table(c1)

# prepare data frame for ggplot
df1 = data.frame(Status=dd$Status, mca$ind$coord[,1:2], cluster=as.factor(c1))
```

```
# visualize clusters using the first two factorial coordinates
ggplot(data=df1, aes(x=Dim.1, y=Dim.2)) +
geom_hline(yintercept=0, colour="gray65") +
geom_vline(xintercept=0, colour="gray65") +
geom_point(aes(colour=cluster), alpha=0.5) +
labs(x="Dim 1", y="Dim 2") +
ggtitle("MCA plot with clusters of individuals")

# centers of gravity of the clusters
cog = aggregate(as.data.frame(Psi), list(c1), mean)[,-1]

# what's the quality of the hierarchical partition?
Bss = sum(rowSums(cog^2) * table(c1))
Ib4 = 100 * Bss / Tss
Ib4


# ============================================================================
# Combining k-means and hierarchical clustering with MCA results
# ============================================================================

# let's consolidate the partition
# we'll apply  k-means using the cog's from the hierarchical clustering
k5 = kmeans(Psi, center=cog)
k5$size
Bss = sum(rowSums(k5$centers^2) * k5$size)
Wss = sum(k5$withinss)
Ib5 = 100 * Bss / (Bss + Wss)
Ib5

# clustering of large data sets
# first 2 kmeans with k=14
n1 = 14
km1 = kmeans(Psi, n1)
km2 = kmeans(Psi, n1)

# what's the overlapping between clusters?
table(km2$cluster, km1$cluster)

clas = (k2$cluster - 1)*n1 + k1$cluster
freq = table(clas)
freq[1:10]

# what do we have in freq?
cogclas <- aggregate(as.data.frame(Psi), list(clas), mean)[,2:(ncol(Psi)+1)]

# perform a hierarchical clustering using cogclas
# compare the computational cost (this is way much faster!)
d2 = dist(cogclas)
h2 = hclust(d2, method="ward", members=freq)

# dendrogram
plot(h2)
# barplot
barplot(h2$height)
# cut tree in nc=8 groups
c2 <- cutree(h2, nc)


# ============================================================================
# Probabilistic clustering on MCA results
# ============================================================================

# load package mclust
```

```
library(mclust)

# check the computational cost
emc <- Mclust(Psi, G=7:9)
print(emc)
attributes(emc)

# In this case, we have a probability for each individual
emc$z[1:10,]

# let's see the membership for every individual
emc$classification[1:10]
table(emc$classification)

# what's the quality of the partition?
cog <- aggregate(as.data.frame(Psi), list(emc$classification), mean)[,2:
(ncol(Psi)+1)]
Bss <- sum(rowSums(cog^2)*as.numeric(table(c1)))
Ib7 <- 100*Bss/Tss
Ib7

# add emc classification to data frame
df1$EMC.class = as.factor(emc$classification)

# visualize clusters using the first two factorial coordinates
ggplot(data=df1, aes(x=Dim.1, y=Dim.2)) +
geom_hline(yintercept=0, colour="gray65") +
geom_vline(xintercept=0, colour="gray65") +
geom_point(aes(colour=EMC.class), alpha=0.5) +
labs(x="Dim 1", y="Dim 2") +
ggtitle("MCA plot with clusters of individuals")
```