Assignment 5 Group 14 Multivariate Statistics October 2017

M. Koops [432409], R. van Eeuwijk [454161], P. de Wet [427863], R. Hammink [435001], C. van Veen [432890]

In this report, we apply homogeneity analysis to find relationships between the different categories of different variables in a dataset concerning holiday spending. This data has 708 observations and 8 variables (*destination*, *transport*, *holiday length*, *accommodation*, *big expenses*, *no. children*, *income* and *spending*) which are on a categorical scale. Our research question is: what are the most important relationships between the different categories of these variables?

Methods

The objective of homogeneity analysis is to visualize all the different categories of the variables in a dataset and the observations in a low-dimensional space (in our case 2-dimensional). We hope to find relationships that exist between categories and observations in the m-dimensional setting by plotting them in a 2-dimensional setting, while preserving the variation in the original data as well as possible.

Suppose we have an $n \times m$ data matrix $\mathbf{B}$ containing m categorical variables. Each variable k (with $k = \{1, \dots, m\}$) may have a different number of categories c_k. The kth column of $\mathbf{B}$ can be decomposed as $\mathbf{b}_k = \mathbf{G}_k \mathbf{c}_k$, where $\mathbf{c}_k$ is a $c_k \times 1$ vector containing all categories of variable k and $\mathbf{G}_k$ an $n \times c_k$ indicator matrix, determining the category of each observation. A first step is to find $\mathbf{G}_k$ for each variable, given the input data. Then the goal is to find for each variable k a $c_k \times 2$ matrix $\mathbf{Y}_k$, containing for each category of k the 2-dimensional coordinates to be plotted in a figure, and to find 2-dimensional coordinates for each of the n observations which is stored in an $n \times 2$ matrix $\mathbf{X}$. Those must be chosen such that they minimize

$$L(\mathbf{X}, \mathbf{Y}_1, ..., \mathbf{Y}_m) = m^{-1} \sum_{k=1}^{m} \text{tr}(\mathbf{X} - \mathbf{G}_k \mathbf{Y}_k)'(\mathbf{X} - \mathbf{G}_k \mathbf{Y}_k) \qquad (1)$$

subject to $\mathbf{X}'\mathbf{X} = n\mathbf{I}$ (to ensure orthogonality of columns) and $\mathbf{1}'\mathbf{X} = \mathbf{0}$ (to ensure column mean of zero). After initializing some random $\mathbf{X}$, (1) can be minimized by alternating least-squares: within each iteration each $\mathbf{Y}_k$ is updated given previous $\mathbf{X}$, then $\mathbf{X}$ is updated using the newly computed vectors $\mathbf{Y}_k$.

Given $\mathbf{X}$, $\mathbf{Y}_k$ is updated as $\mathbf{Y}_k = \mathbf{D}_k^{-1} \mathbf{G}_k' \mathbf{X}$, where $\mathbf{D}_k = \mathbf{G}_k' \mathbf{G}_k$ is a $c_k \times c_k$ diagonal matrix representing the relative frequency of observations that belong to each category of variable k. Given all $\mathbf{Y}_k$, the update of $\mathbf{X}$ is computed by first determining $\mathbf{Z} = m^{-1} \sum_{k=1}^{m} \mathbf{G}_k \mathbf{Y}_k$ and to set $\mathbf{Z} = \mathbf{Z} - n^{-1} \mathbf{1} \mathbf{Z}' \mathbf{1}$ (each element is now in deviation-from-column-mean), and then to apply singular value decomposition (SVD) to $\mathbf{Z}$, that is $\mathbf{Z} = \mathbf{P} \mathbf{\Phi} \mathbf{Q}'$ where $\mathbf{P}'\mathbf{P} = \mathbf{I}$; $\mathbf{Q}'\mathbf{Q} = \mathbf{Q}\mathbf{Q}' = \mathbf{I}$ and $\mathbf{\Phi}$ is nonnegative diagonal matrix. The matrix $\mathbf{P}$ is used to update $\mathbf{X}$ as $\mathbf{X} = n^{\frac{1}{2}}\mathbf{P}$. Then each $\mathbf{Y}_k$ should be rotated to principal axes as $\mathbf{Y}_k = n^{-\frac{1}{2}}\mathbf{Y}_k\mathbf{Q}$. Given the updates a new value of the loss function can be computed, and if the improvement is deemed too small the algorithm stops with the final configuration.

After convergence of the loss function (1), the importance of a variable k is computed by the *discrimination measure* with the final obtained $\hat{\mathbf{Y}}_k$. For each dimension s (with $s = \{1, \ldots, p\}$) the discrimination measure of variable k is given by $\eta^2_{ks} = n^{-1}\hat{\mathbf{Y}}'_k(\cdot, s)\mathbf{D}_k\hat{\mathbf{Y}}_k(\cdot, s)$. Here, η^2_{ks} is the weighted sum of squared category scores for dimension s and has a value between 0 and 1. This basically measures the weighted average squared distance of the catigorical quantifications with respect to the origin of the p-dimensional space. The discrimination measure of a variable k then becomes $\boldsymbol{\eta}_k{}' = \text{Diag}(\hat{\mathbf{Y}}'_k\mathbf{D}_k\hat{\mathbf{Y}}_k)$. In homogeneity analysis the maximum number of dimensions allowed is $\sum_{k=1}^{m} c_k - m$ and the importance of a dimension is given by its eigenvalue. Under the assumption of no missing data, the eigenvalues corresponding to the dimensions s can be computed by $\gamma_s = m^{-1}\sum_{k=1}^{m}\eta^2_{ks}$. Large eigenvalues indicate important dimensions. In HOMALS the eigenvalues are divided by the number of variables m such that the eigenvalues obtain values between 0 and 1. Preferably, dimensions are chosen that are easy to interpret.

If the dataset has many observations but not too many categories, a more efficient method exists. Write $\mathbf{G} = [\mathbf{G}_1|...|\mathbf{G}_m]$, then $\mathbf{G}'\mathbf{G}$ is called the Burt-matrix, with on the diagonal the frequency matrices $\mathbf{D}_k$, and on the off-diagonal it has the matrices $\mathbf{G}'_k\mathbf{G}_l$ that count categories that co-occur between variables k and l. We can have all $\mathbf{Y}_k$ is one matrix $\mathbf{Y} = [\mathbf{Y}_1|...|\mathbf{Y}_m]'$ and let $\mathbf{D}$ be a diagonal matrix containing all diagonal matrices $\mathbf{D}_k$. Then perform eigendecomposition on the matrix $\mathbf{D}^{-\frac{1}{2}}\mathbf{G}'\mathbf{G}\mathbf{D}^{\frac{1}{2}} = \mathbf{Q}\boldsymbol{\Phi}^2\mathbf{Q}'$ and compute the coordiantes of the categories as $\mathbf{Y} = \mathbf{D}^{-\frac{1}{2}}\mathbf{Q}_p\boldsymbol{\Phi}_p$ where $\boldsymbol{\Phi}_p$ contains the p largest eigenvalues of the decomposition and $\mathbf{Q}_p$ the accompanying eigenvectors.

Results

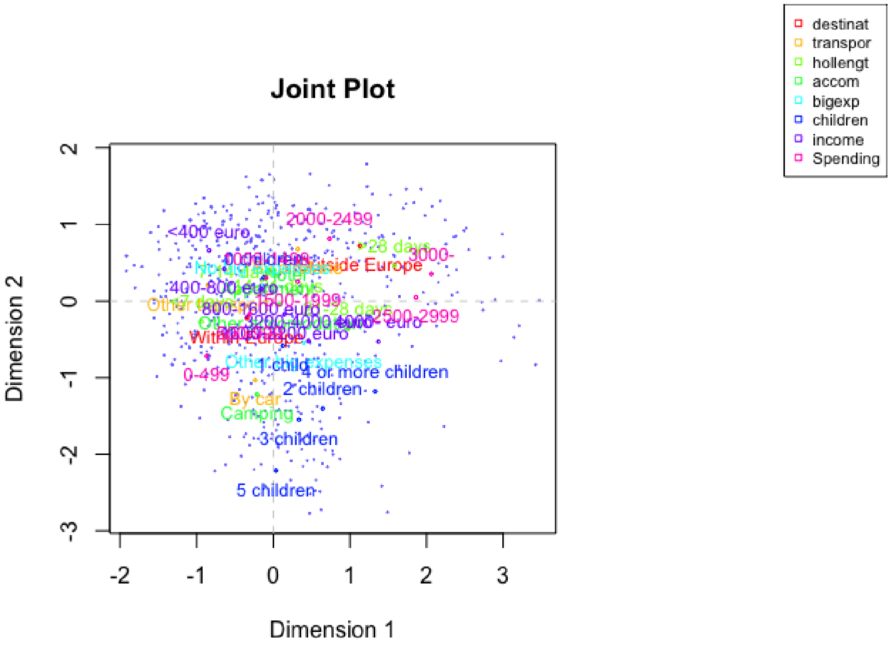

Figure 1: Joint plot created with HOMALS package in R

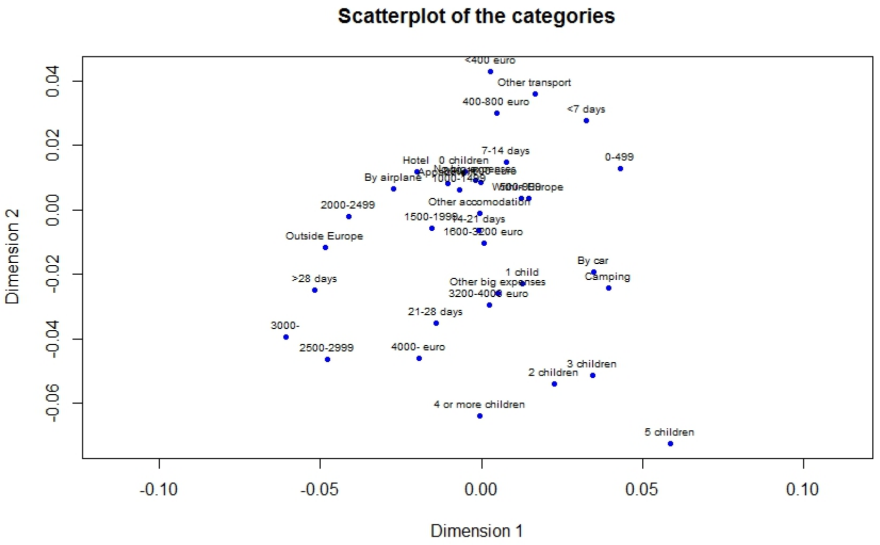

Figure 2: Scatterplot created based on our own code in R

Figures 1 and 2 show the solution of the HOMALS package and our own solution. The locations of many points differ substantially although the conclusions are generally the same. For instance, both imply that people with children tend to earn more than 400 euros, and families also tend to favor camping more than other types of accommodation. Also, the higher the income, the less people tend to favor traveling by car. Also, traveling outside of Europe typically means holidays take longer than 7 days.

```
rm(list = ls())
library(mlr)
library(ParamHelpers)
# 5 MCA through homals
load("//campus.eur.nl/users/home/454161re/Desktop/
\\
holiday.spending(1).RData")
library(homals)
library(ggplot2)
data <- holiday.spending

##### Functions ######
center_scale <- function(s){
  scale(s,scale=FALSE)
}
```

```

# SVD:  X = UDt(V)
SVD <- function(x){
  res<-svd(x)
  d <- res$d   #singular values                            PHI
  u <- res$u   # left singular values   t(u)*u = I         P
  v <- res$v   #right singular values   t(v)*v = v*t(v)=I  Q
  return(res)
  # return(d)
}


homals_own <- function(data){

  n <- dim(data)[1]
  m <- dim(data)[2]
  p <- 2
  #construct random matrix X
  x1 = round(runif(n*p,0,1), 2)
  X <- matrix(x1,ncol=p)

  ###########   Precalculations    ###########
  # take X in its deviation to column mean
  X <- center_scale(X)

  # choose X = n^(1/2*)*u s.t. x is random
  X <- n^(1/2)*SVD(X)$u


  # compute indicator matrix G and D=G'G for each variable k, \\
  as well as the updates for Yk = inv(D_k)G_k'X
  for(i in 1:m) {
    namG <- paste("G", i, sep = "")
    namD <- paste("D", i, sep = "")
    namY <- paste("Y", i, sep = "")
    namZ <- paste("Z", i, sep = "")
    G<- as.matrix(createDummyFeatures(data[,i]))
    D <- t(G)%*%G
    Y <- solve(D)%*%t(G) %*% X
    assign(namG,G)
    assign(namD,D)
    assign(namY,Y)
    Z <- 1/m * G%*%Y
    assign(namZ,Z)
  }

```

```
# #compute loss function
# for(k in 1:m){
#   namG <- paste("G", k, sep = "")
#   namY <- paste("Y", k, sep = "")
#   G <- get(namG)
#   Y <- get(namY)
#   Li[k] <- sum(diag(t(X-G%*%Y)%*%(X-G%*%Y)))
# }
# L <- 1/m*sum(Li) + 1000


# update X given Y
Z <- Z1+Z2+Z3+Z4+Z5+Z6+Z7+Z8
Z <- center_scale(Z)
P <- SVD(Z)$u
Q <- SVD(Z)$v
X <- n^(1/2)*P

# set Y to Y rotated to pc (for all k), using the new X
for(i in 1:m){
  namY <- paste("Y", i, sep = "")
  Y <- get(namY)
  Y <- n^(-1/2)*Y%*%Q
  assign(namY,Y)
}

############ ITERATIVE LOOP #############
eps <- 10^-6
t <-0
Li <- numeric(0)
L <- numeric(0)
repeat{
  # store lagged version of L
  t<-t+1
  for(k in 1:m){
    namG <- paste("G", k, sep = "")
    namD <- paste("D", k, sep = "")
    namY <- paste("Y", k, sep = "")
    namZ <- paste("Z", k, sep = "")
    #given Xt-1, new Yk's
    Y <- solve(get(namD))%*%t(get(namG)) %*% X
    assign(namY,Y)

    # compute update of X (Xt) given new Y
    Z <- 1/m * get(namG)%*%Y
    assign(namZ,Z)
    }
    Z <- Z1+Z2+Z3+Z4+Z5+Z6+Z7+Z8
    Z <- center_scale(Z)
    P  <- SVD(Z)$u
    Q <- SVD(Z)$v
    X <- n^(1/2)*P

    # rotate Y to principal components
    for(i in 1:m){
      namY <- paste("Y", i, sep = "")
      Y <- get(namY)
      Y <- n^(-1/2)*Y%*%Q
      assign(namY,Y)
    }

    # compute the loss function, first for every element \\
    then sum and devide to get Loss
    for(k in 1:m){
      namG <- paste("G", k, sep = "")
      namY <- paste("Y", k, sep = "")
      G <- get(namG)
      Y <- get(namY)
      Li[k] <- sum(diag(t(X-G%*%Y)%*%(X-G%*%Y)))
    }
    L[t] <- 1/m*sum(Li)

    if (t > 1) {
      if (L[t-1] - L[t] < eps){
        break
      }
    }
  }

  listreturn <- list("X"=X,"L"=L)
  return(listreturn)
}

Y <- rbind(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8)

plot(Y, main = "Scatterplot of the categories", xlab = \\
"Dimension 1" , ylab = "Dimension 2", pch = 20, \\
col="blue",asp = 1)
text(Y[,1],Y[,2],labels = row.names(Y), cex = 0.6, \\
adj = c(0.5, -1))

results <- homals_own(data)
```

```

results$L
```