Assignment3Final

Neli Noykova

June 15, 2017

MULTIPLE CORRESPONDANCE ANALYSIS - Assignment 3

The data

As during the Assignment set 1. here we again use **Finnish** sample from ISSP 2012 survey "Family and Changing Gender Roles IV". Original data involve 1171 observations of 8 variables (4 substantive and 4 demographic). All variables are categorical.

The 4 **substantive** variables, which values are measured in 1-5 scale, are:

A: Married people are generally happier than unmarried people.
B: People who want children ought to get married.
C: It is all right for a couple to live together without intending to get married.
D: Divorce is usually the best solution when a couple can't seem to work out their marriage problems.

The **demographic** variables are:
g: gender (1=male, 2=female)
a: age group (1=16-25, 2=26-35, 3=36-45, 4=46-55, 5=56-65, 6= 66+)
e: education (1=Primary, 2=Comprehensive, primary and lower secondary, 3= Post-comprehensive, vocational school or course, 4=General upper secondary education or certificate, 5= Vocational post-secondary non-tertiary education, 6=Polytechnics, 7= University, lower academic degree, BA, 8=University, higher academic degree, MA
p: Living in steady partnership (1=Yes, have partner; live in same household, 2=Yes, have partner; don't live in same household, 3=No partner)

The data wrangling includes the following changes:

1. The missing data ae removed (this has already been provided). The number of observations without missing data is N=924.
2. We again use a combined variable **ga = 6*(g-1) + a**. **The combined variable** ga** describes the interaction of gender and age categories.

Graphical overview of the data and summaries of the variables

The preliminary treated data look as:

```
Finland <- read.table("Finland.txt")
Finland$ga <- 6*(Finland$g-1) + Finland$a
head(Finland)
```

```
##   A B C D g a e p ga
## 1 3 3 1 2 1 2 4 3  2
## 2 3 2 3 2 1 4 2 3  4
## 3 3 3 1 3 1 3 8 1  3
## 4 3 2 2 3 2 2 6 1  8
## 5 2 2 2 3 2 4 5 1 10
## 6 3 3 1 3 1 3 7 3  3
```

```r
dim(Finland)
```

```
## [1] 924   9
```

```r
str(Finland)
```

```
## 'data.frame':    924 obs. of  9 variables:
##  $ A : int  3 3 3 3 2 3 3 2 2 3 ...
##  $ B : int  3 2 3 2 2 3 3 3 3 3 ...
##  $ C : int  1 3 1 2 2 1 1 1 2 3 ...
##  $ D : int  2 2 3 3 3 3 2 2 3 3 ...
##  $ g : int  1 1 1 2 2 1 2 2 2 2 ...
##  $ a : int  2 4 3 2 4 3 1 2 5 4 ...
##  $ e : int  4 2 8 6 5 7 4 8 7 5 ...
##  $ p : int  3 3 1 1 1 3 3 3 1 3 ...
##  $ ga: num  2 4 3 8 10 3 7 8 11 10 ...
```

```r
library(FactoMineR)
library(tidyr)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

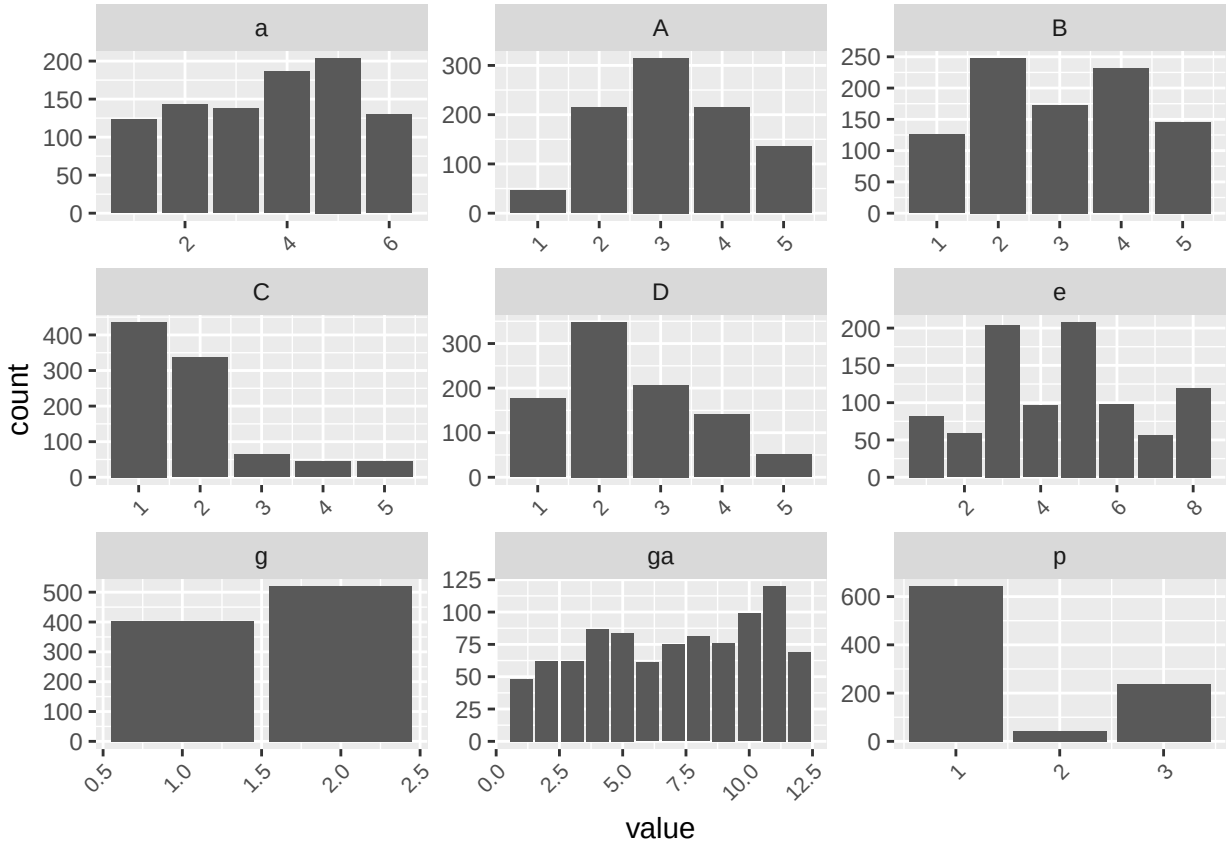
```r
keep_columns <- c("A", "B", "C", "D", "g", "a", "e", "p", "ga")
Finland <- dplyr::select(Finland, one_of(keep_columns))
summary(Finland)
```

```
##        A               B               C               D
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000
##  Median :3.000   Median :3.000   Median :2.000   Median :2.000
##  Mean   :3.193   Mean   :3.025   Mean   :1.835   Mean   :2.503
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##        g               a               e               p
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:1.000
##  Median :2.000   Median :4.000   Median :5.000   Median :1.000
##  Mean   :1.563   Mean   :3.644   Mean   :4.527   Mean   :1.563
##  3rd Qu.:2.000   3rd Qu.:5.000   3rd Qu.:6.000   3rd Qu.:3.000
##  Max.   :2.000   Max.   :6.000   Max.   :8.000   Max.   :3.000
##        ga
##  Min.   : 1.000
##  1st Qu.: 4.000
##  Median : 7.000
##  Mean   : 7.021
##  3rd Qu.:10.000
```

```
##  Max.   :12.000
```

We present data graphically as barplots using ggplot() function.

```
gather(Finland) %>% ggplot(aes(value)) + facet_wrap("key", scales = "free") + geom_bar()+theme(axis.tex
```



We observe that the highest frequency of the answers to question A takes a middle hypothesis, while for questions B and D the highest frequency has the answer 2. Most respondents strongly agree with the hypothesis C. The rest of the barplots show the distribution of demographic variables.

Nonlinear MCA: The package homals in R

There are two main constrained forms of MCA: canonical MCA and nonlinear MCA. Canonical MCA use explanatory variables to explain set of response variables, while in classical MCA there is no restriction imposed by explanatory variables since all variables are assumed to be response variables.

Nonlinear MCA is also assumed to be a constrained form of MCA because it is obtained from common MCA (known also as homogeneity analysis) by imposing restrictions on the variable ranks and levels as well as defined set of variables.

The most important advantages of nonlinear over linear MCA are that except truly numeric nonlinear MCA incorporates nominal and ordinal variables, and additionally can discover nonlinear relationships between variables.

In nonlinear MCA the variables could be nominal, ordinal or true numeric. Nominal variables consist of unordered categories. Since principal components are weighted sums of the original variables, nominal variables could not be analyzed using standard linear PCA. Ordinal variables consist of ordered categories, for example as we have also used, a Likert-type scale. Such scale values are not truly numeric because intervals

between consecutive categories are not equal. Although ordinal variables display more structure than nominal ones, these variables still do not possess traditional numeric properties. The true numeric variables can be viewed as categorical with c categories, where c indicates the number of different observed values. Both ratio and interval variables are considered numeric in nonlinear PCA.

Linear MCA is suitable for variables, all measured as numeric. It is possible to exist nonlinear relationships between some of these variables. Then nonlinear MCA will be more appropriate approach.

Nonlinear PCA converts categories into numeric values because the variance could be established only for numeric values. This process is called quantification. Thus correlations are not computed between observed variables, but between quantified variables. Therefore the correlation matrix in nonlinear PCA is not fixed and depends on the type of quantification chosen for every variable.

The type of quantification is called analysis level. Different analysis levels imply different requirements.

Nonlinear MCA in R could be provided by using the **homals** package. This package performs simple homogeneity analysis, which corresponds to MCA. The package **homals** also provide some extensions to MCA, including nonlinear MCA. Nonlinear PCA could be provided using the function **homals** and setting appropriate options.

In the example below we use "ordinal" analysis level because the substantive variables are ordinal and the categories are given in Likert-type scale.

```
require(homals)
```

```
## Loading required package: homals
```

```
#algebraically, the geometric concept of dimensionality is related to
#the rank of the matrix, which has to be reduced.
#rank - Which quantification ranks. Default is number of dimensions ndim
#level - Which quantification levels. Possible values are "nominal", "ordinal",
#"numerical", and "polynomial" which can be defined as
#single character (if all variable are of the same level) or
#as vector which length corresponds to the number of variables

Finland.nlpca <- homals(Finland[,1:4], rank=1, level="ordinal")
```

1. Transformation plots

Next the transformation plot is drawn. The horizontal axis (x) of the plot displays the categories (1 to 5) of the ordinal substantive variables. On the vertical axis (y) the category quantification for every substantive variable (A-D) are shown.

```
#Transformation plot: Plots the original (categorical) scale against
#the transformed (metric) scale on each dimension over the categories
#of each variable separately.
plot(Finland.nlpca, plot.type="trfplot")    # relationship of transformed scale with 1-to-5 scale
```

From the transformation plot for the variable D we see that the ordinal category quantification are non-strictly increasing with the original category labels. The original spacing between categories is not maintained in the quantifications. Between categories 2 and 4 we see something similar to plateau, which means that some consecutive categories obtain almost the same quantification, called ties. There are two possible reasons for such ties: 1. The persons scoring in the tied categories do not structurally differ from each other considering their scores on the other variables, and therefore categories cannot be distinguished from each other. 2. The ordinal quantifications are obtained by placing an order restriction on nominal quantifications.

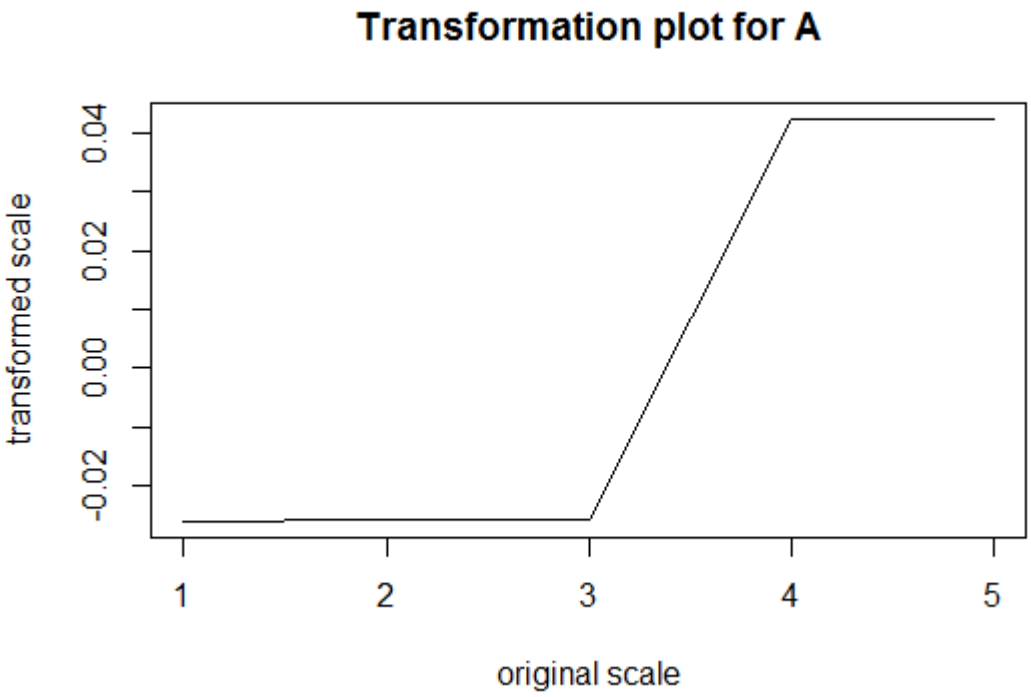


Figure 1: Figure 1A

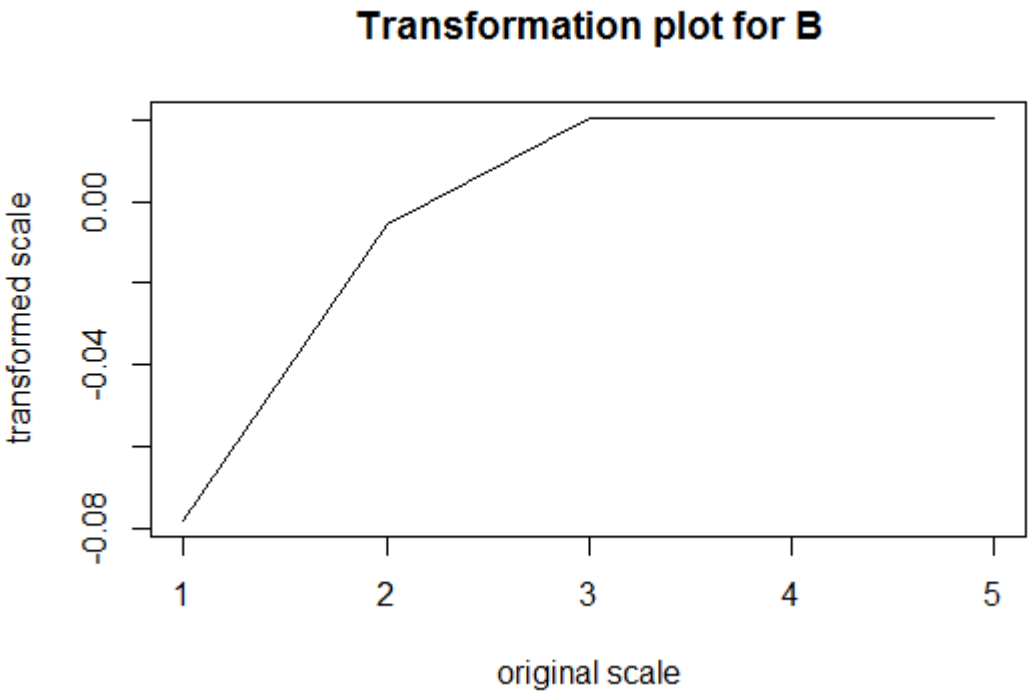


Figure 2: Figure 1B

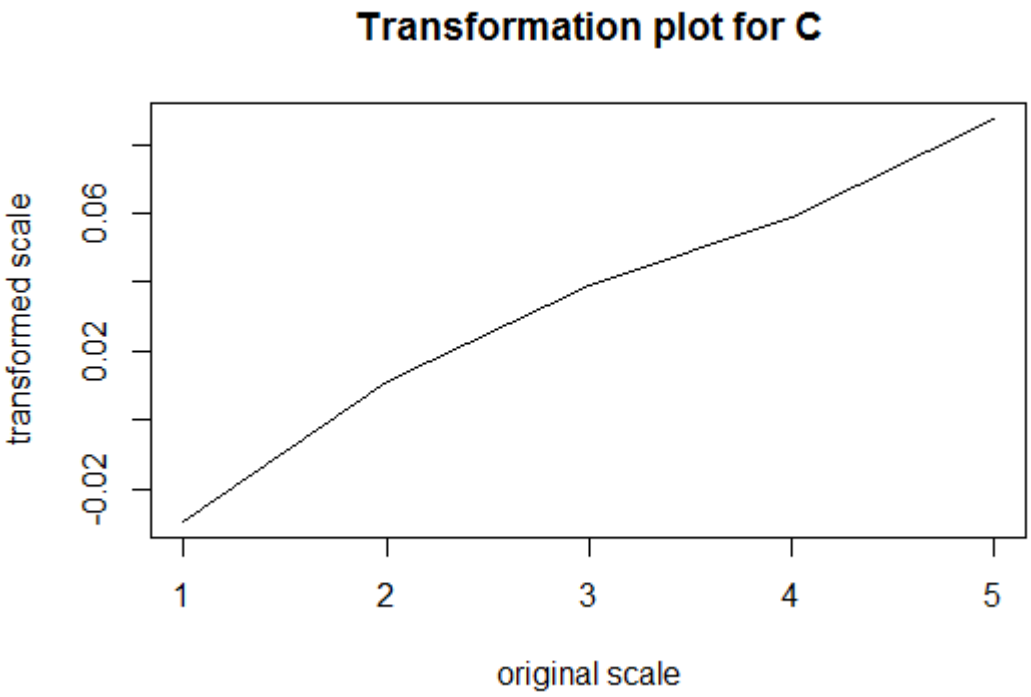


Figure 3: Figure 1C

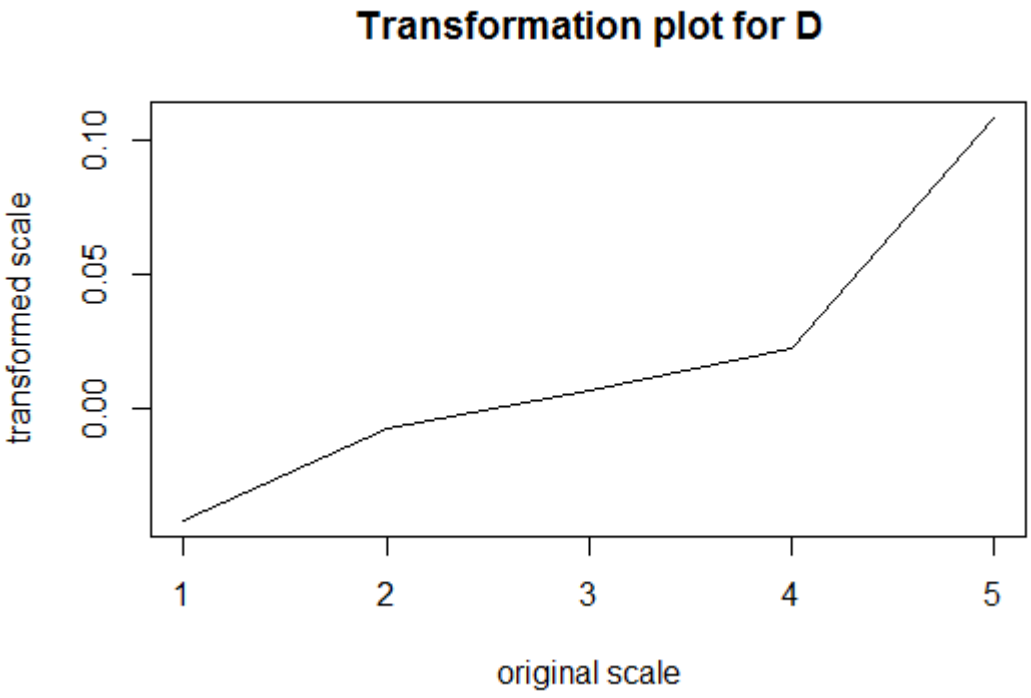


Figure 4: Figure 1D

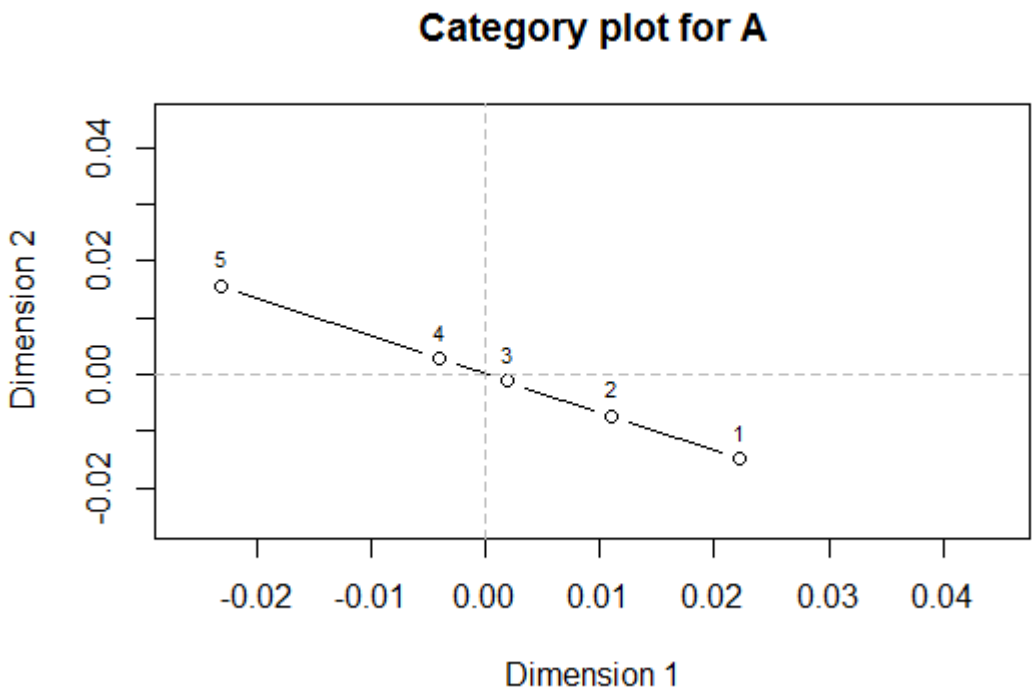


Figure 5: Figure 2A

2.Category plots

This plot represent a quantified variable by displaying its category points in principal component space, where the axes are given by the principal components. Here a variable is represented by a vector going through the origin (0,0) (which is also the mean of the quantified variable) and the point with as coordinates the component loadings for the variable. The component loadings are correlations between the quantified variables and the principal components, and the sum of squared component loadings indicates the variance accounted for (VAF) by the principal component. The category points are also positioned on the variable vector, and their coordinates are found by multiplying the category quantifications by the corresponding loadings on the first (for the x-coordinate) and the second (for the y-coordinate) component.
Categories with quantifications above the mean lie on the side of the origin on which the component loadings point is positioned, and categories with quantifications below the mean lie in the opposite direction. The total length of the variable vector does not indicate the importance of the variable.

```
#2. Category plot: Plots the rank-restricted category quantifications for
#each variable separately.
plot(Finland.nlpca, plot.type="catplot")
```

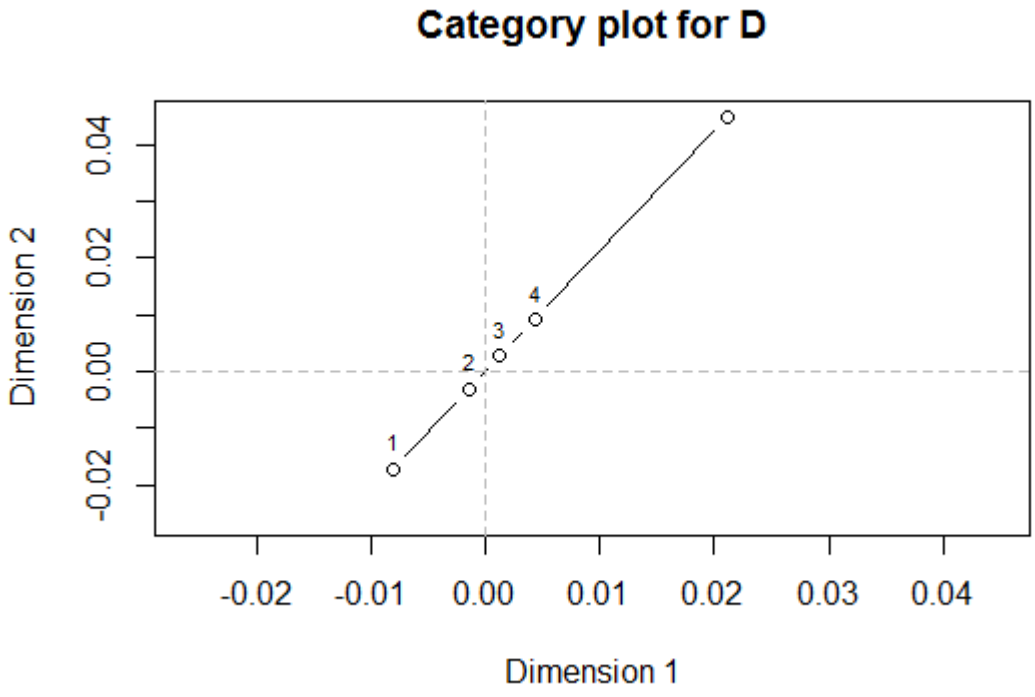
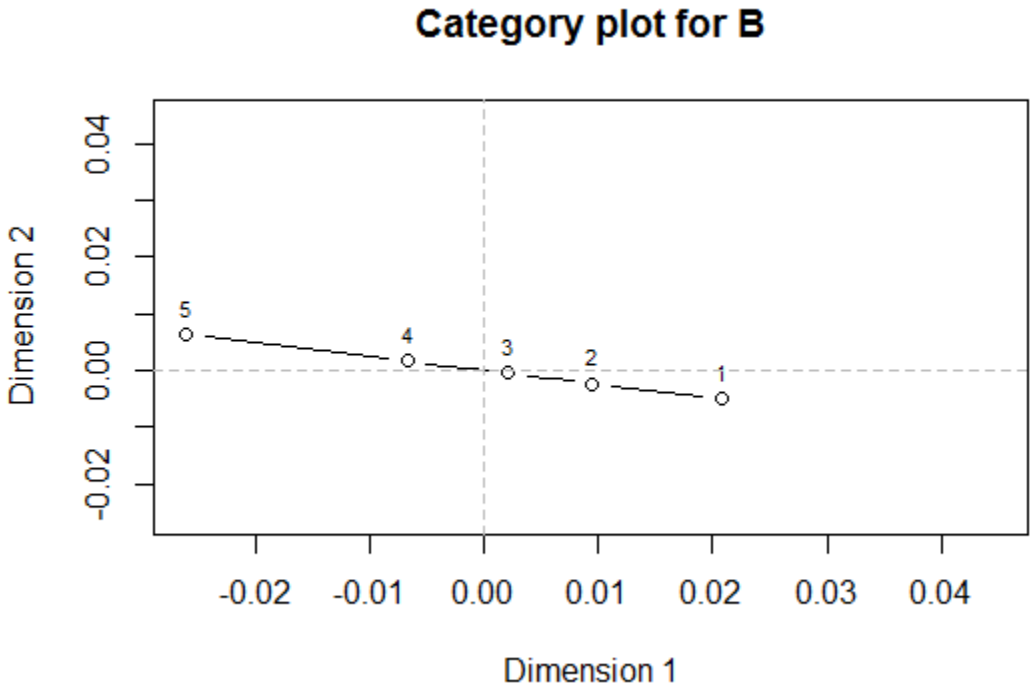


Figure 6: Figure 2D



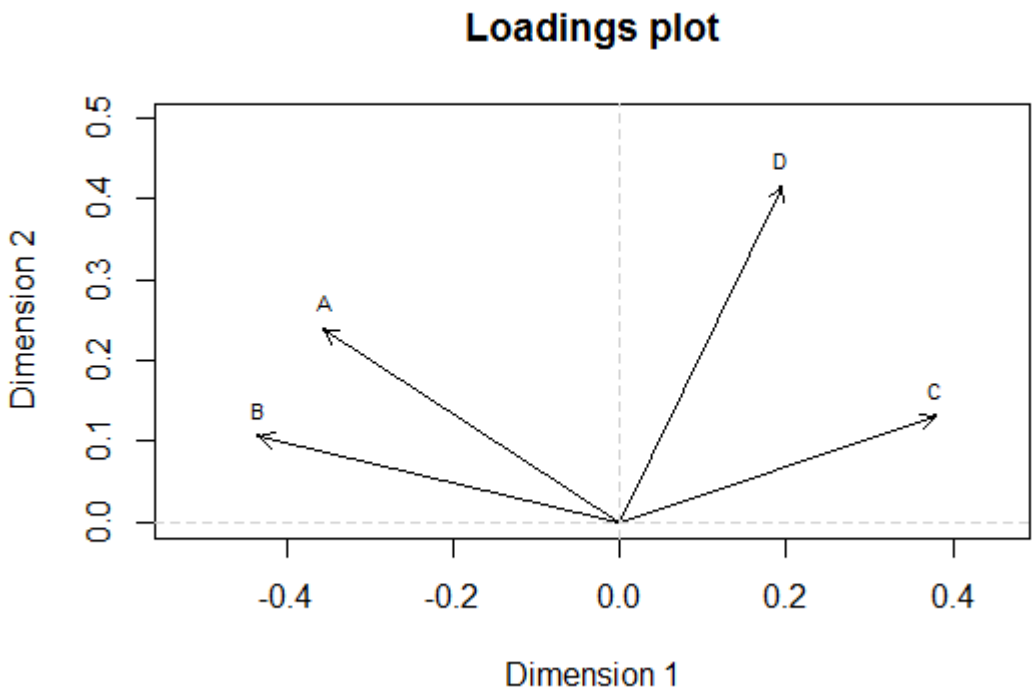


Figure 7: Figure 3

In these plots the loading point is not shown. I do not know how to interpret the nonlinear transformations, which lead to these 4 category plots.

3.Component loadings plot

In these plots only the loading vectors (origin and loading point) are displayed. Here the variables with relatively long vectors fit well into the solution and variables with relatively short vectors fit badly. When vectors are long, the cosines of the angles between the vectors indicate the correlation between the quantified variables. The length of the variable vector from the origin up to the component loading point is an indication of the variable's total variance accounted for (VAF). Thus VAF can be interpreted as the amount of information retained when variables are represented in a lower dimensional space. Nonlinear transformations reduce the dimensionality necessary to represent the variables satisfactory.

```
#3. Loadings plot: Plots the loadings of the variables and connects them with the origin.
plot(Finland.nlpca, plot.type="loadplot")
```

In this example the longest length has a variable vector for D. Other variable vector lengths are very similar and not much shorter than D loading vector.

4.Vector plot

In this plot all cases (object scores) are projected onto a straight line defined by each rank restricted category quantified variable. Here except the straight line, plotted in the category plots, also all object scores are shown.

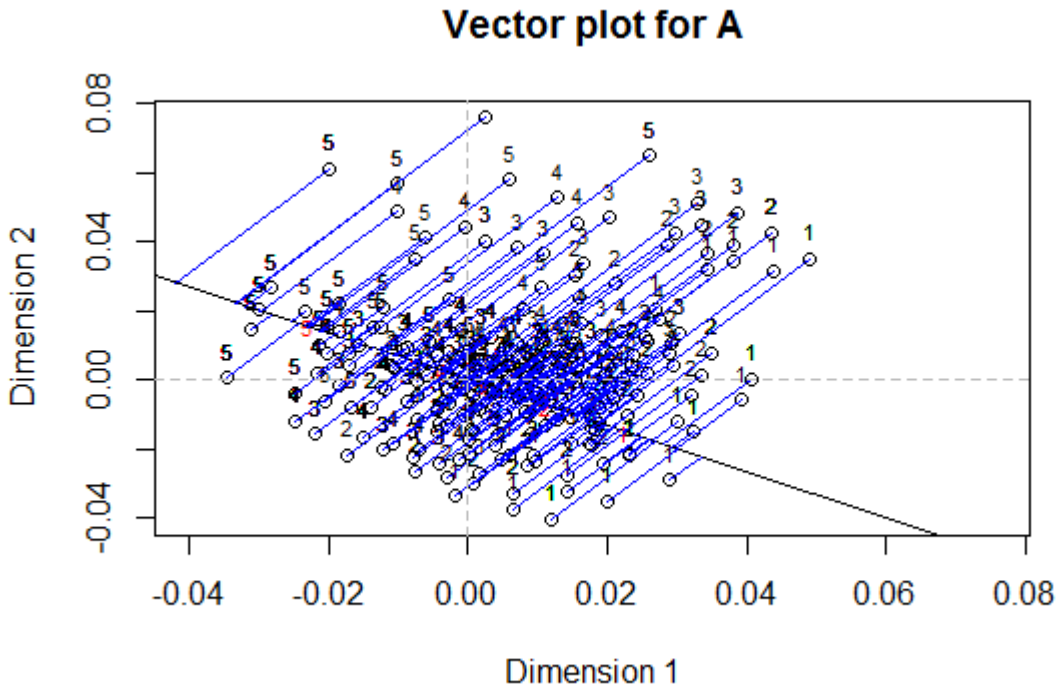
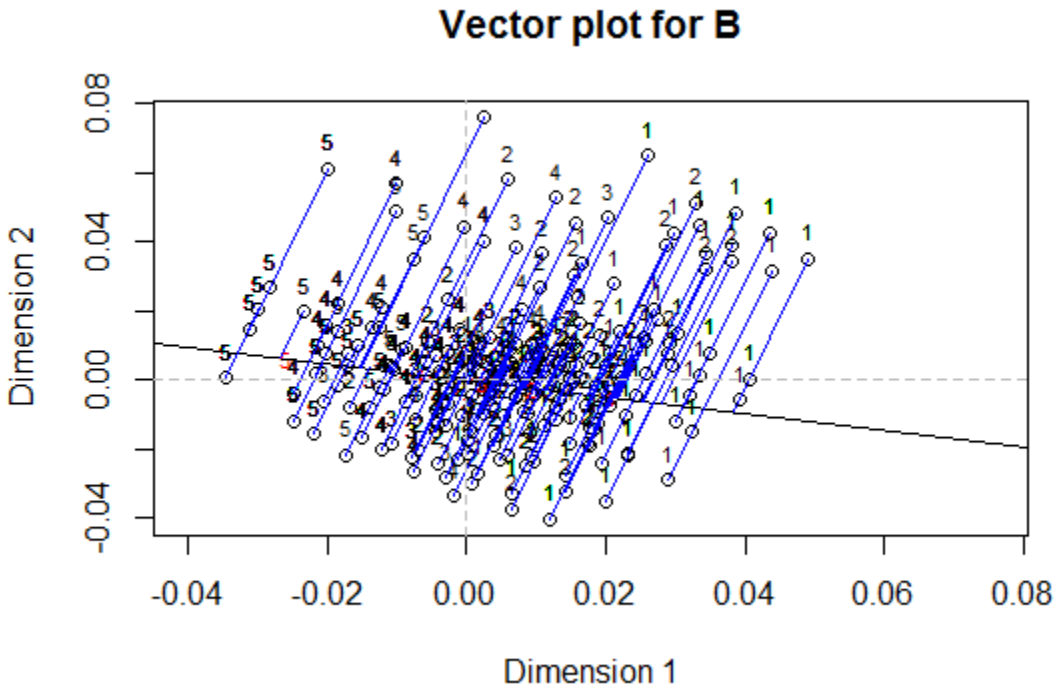
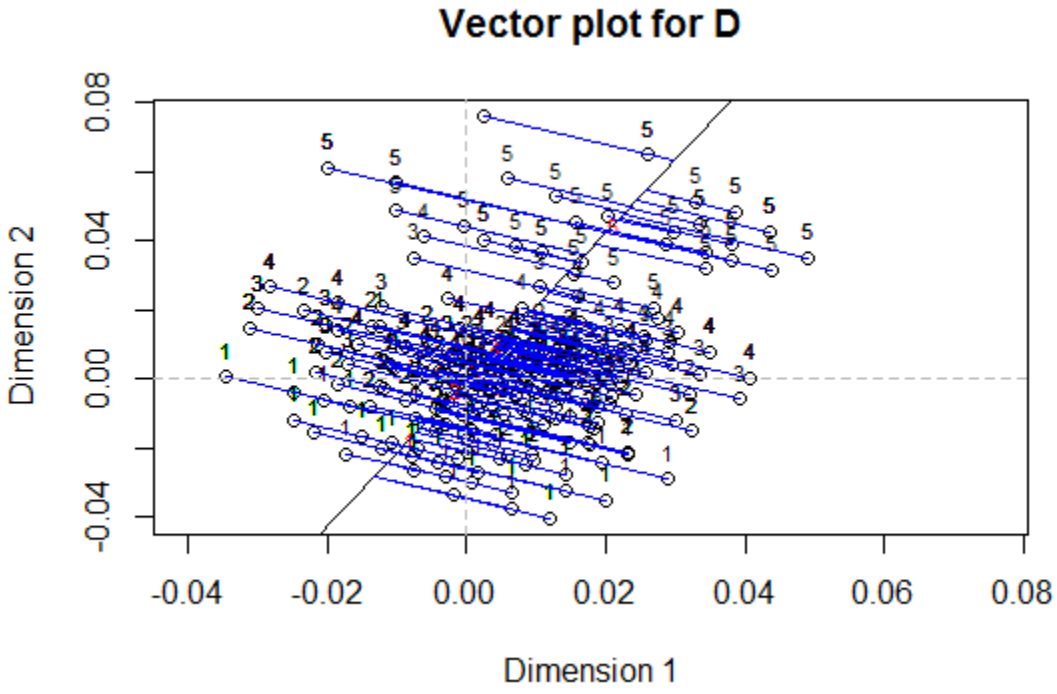


Figure 8: Figure 4A

```
#4. Vector plot: cases projected onto straight line defined by each variable
plot(Finland.nlpca, plot.type="vecplot")
```





##5.Loss

plot

The loss plot shows the rank-restricted category quantifications against the unrestricted for each variable separately. It actually compares MCA and NLPCA.

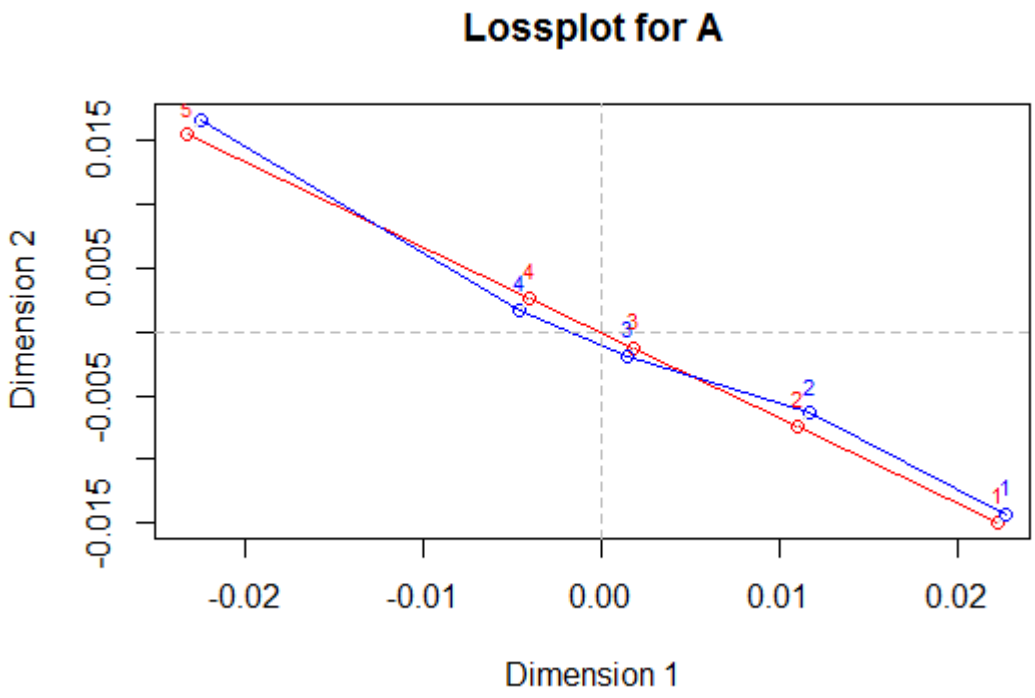
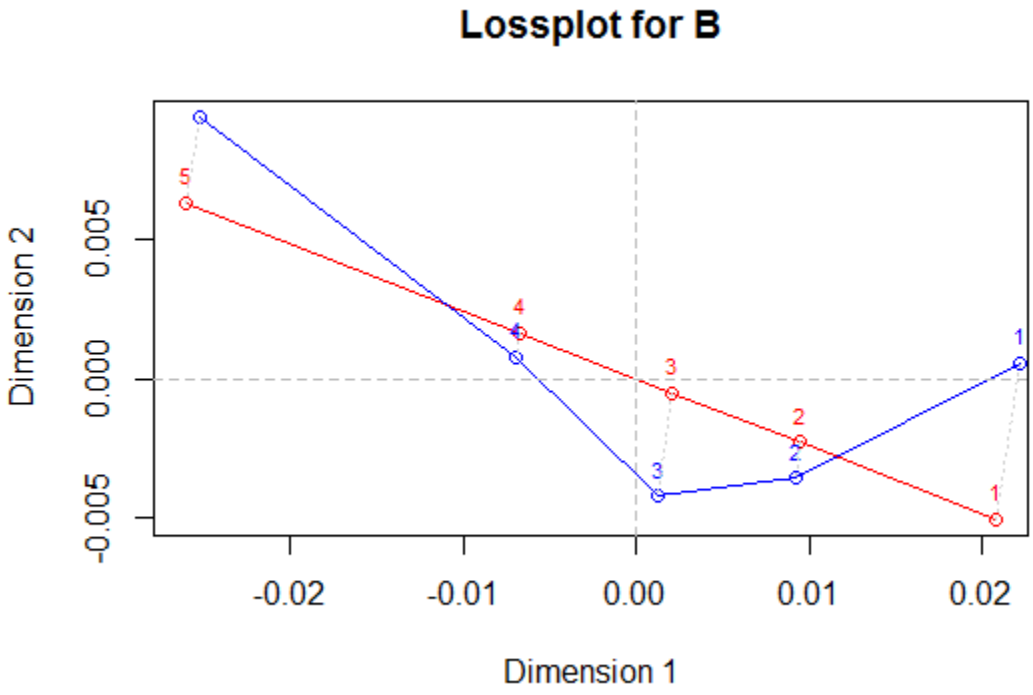
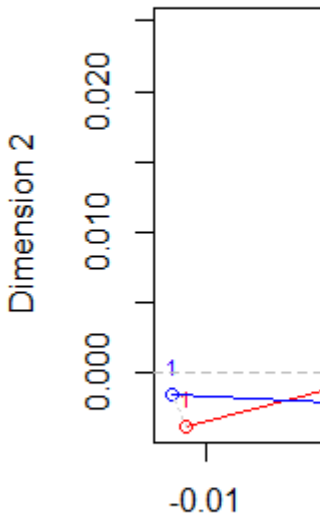
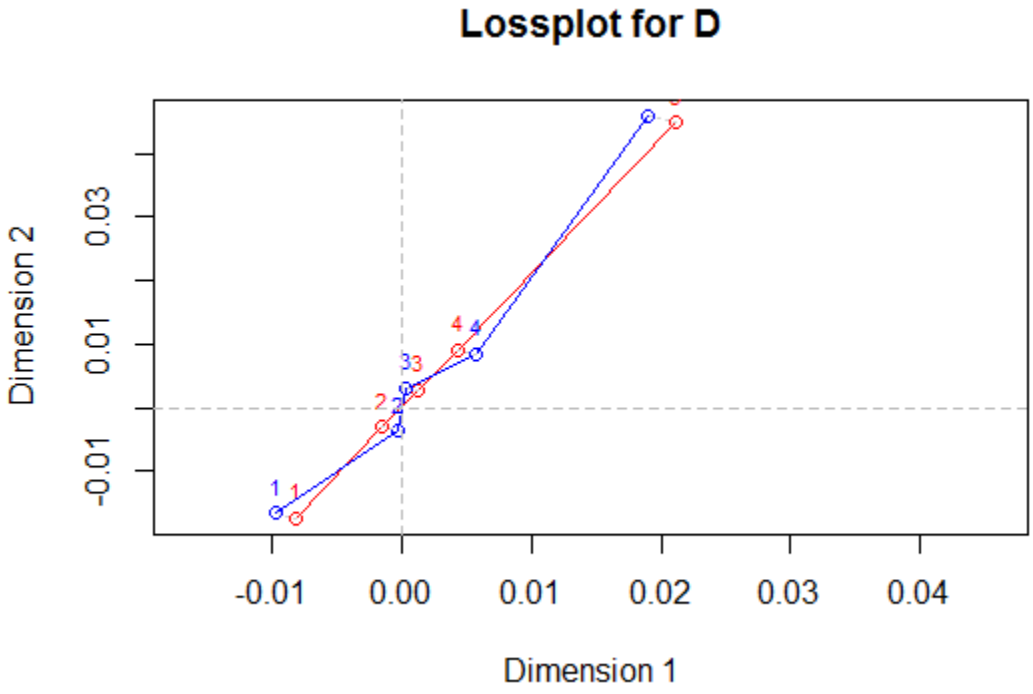


Figure 9: Figure 5A

```
plot(Finland.nlpca, plot.type="lossplot")
```







From these plots we see that the linear and nonlinear MCA results are very near for variables A and D, while both model outputs differ quite a lot for the variables B and C. This could mean that linear model assumption for variables A and D is appropriate.

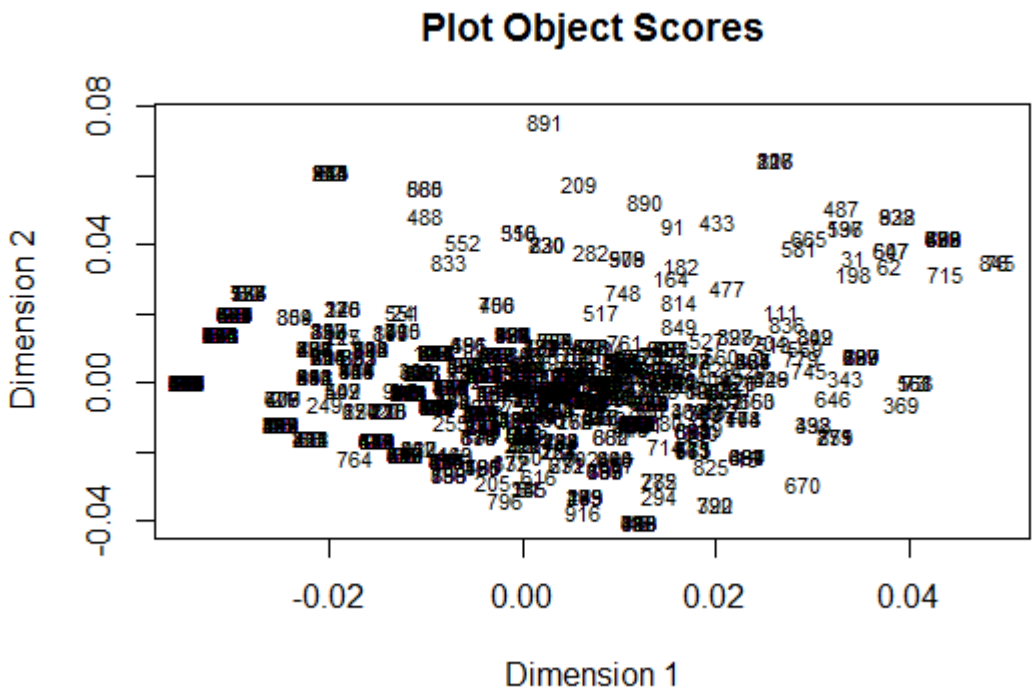


Figure 10: Figure 6

6. Object plot

It plots the scores (cases) of the objects (rows in data set) on two dimensions. Therefore this plot is presented only by dots.

```
plot(Finland.nlpca, plot.type="objplot")
```

7. Label plot

It is similar to the previous object plot. In the label plot the object scores are also plotted, but now for each variable separately with the corresponding category labels. It looks like contour lines forced to be straight & parallel

```
plot(Finland.nlpca, plot.type="labplot")
```

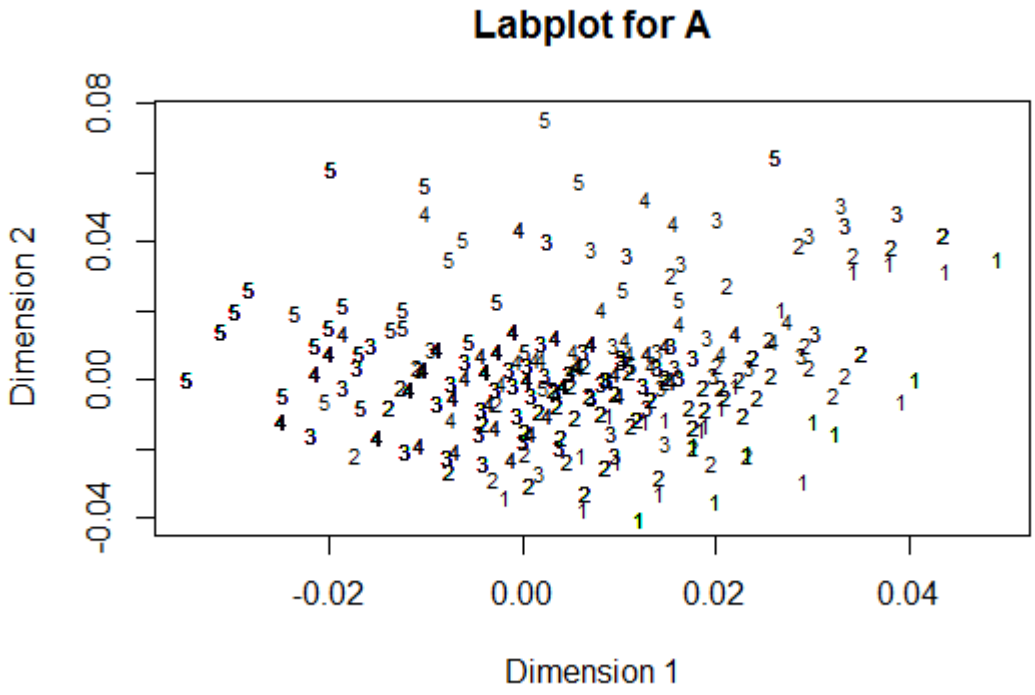
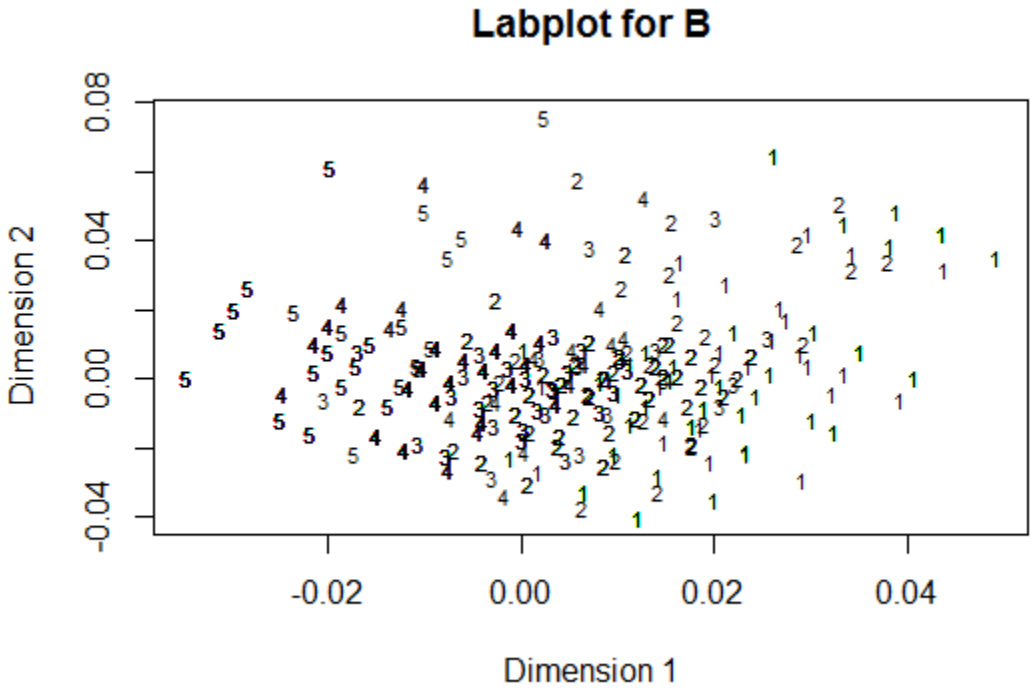
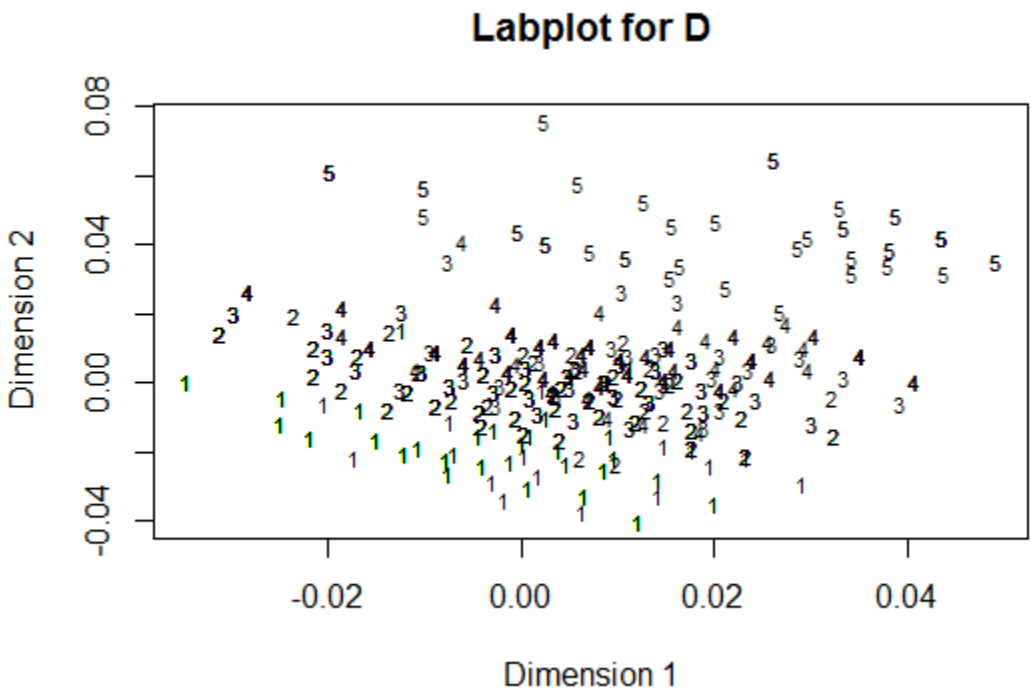


Figure 11: Figure 7A

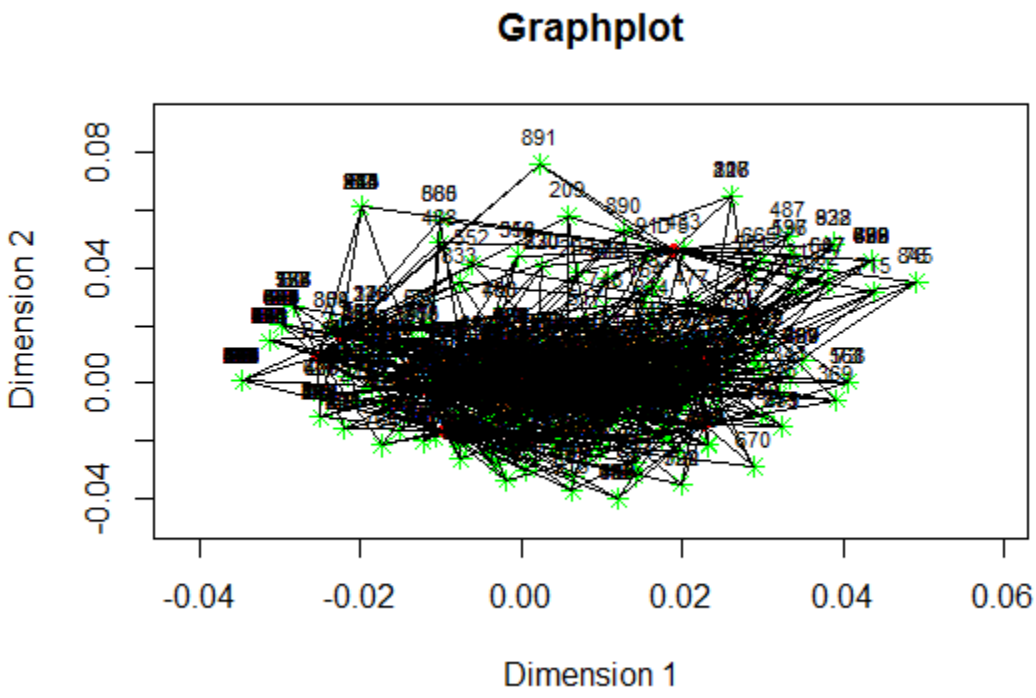




#8. Graph plot.

It is a joint plot with connections between scores/quantifications. So, except the object scores, plotted on object plot, the variable quantifications are also shown. It should be noted that this plot works only for small data sets. Even in the provided example ot is already quite difficult to observe the plotted relationships.

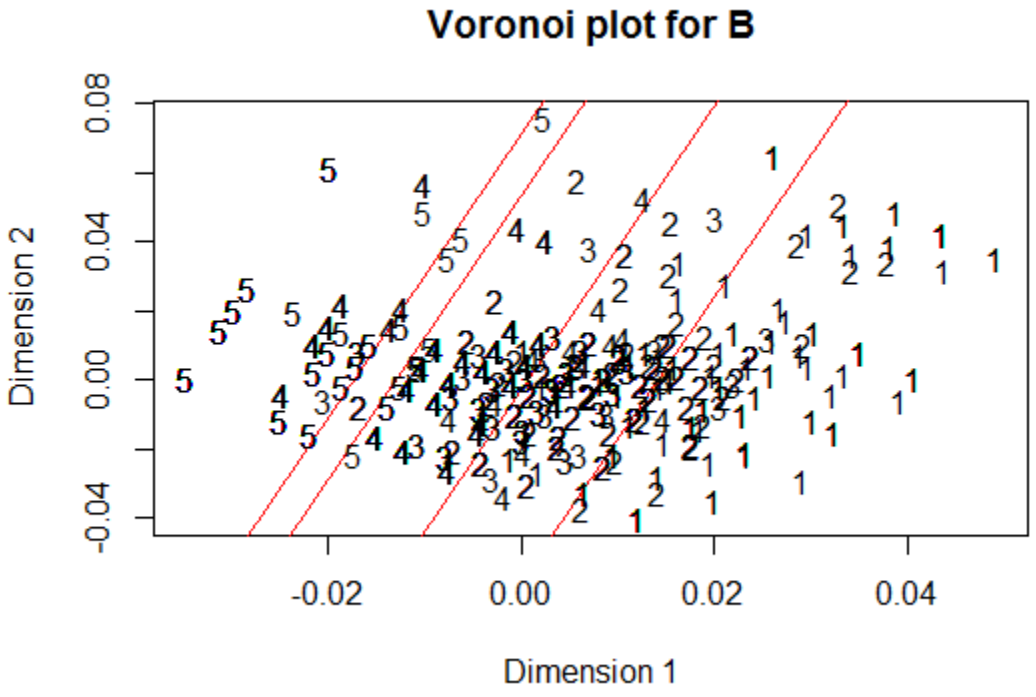
```
plot(Finland.nlpca, plot.type="graphplot")
```



##9.

Voronoi plot. It produces a category plot with Voronoi regions. Looks like contour lines forced to be straight and parallel.

```
plot(Finland.nlpca, plot.type="vorplot")
```



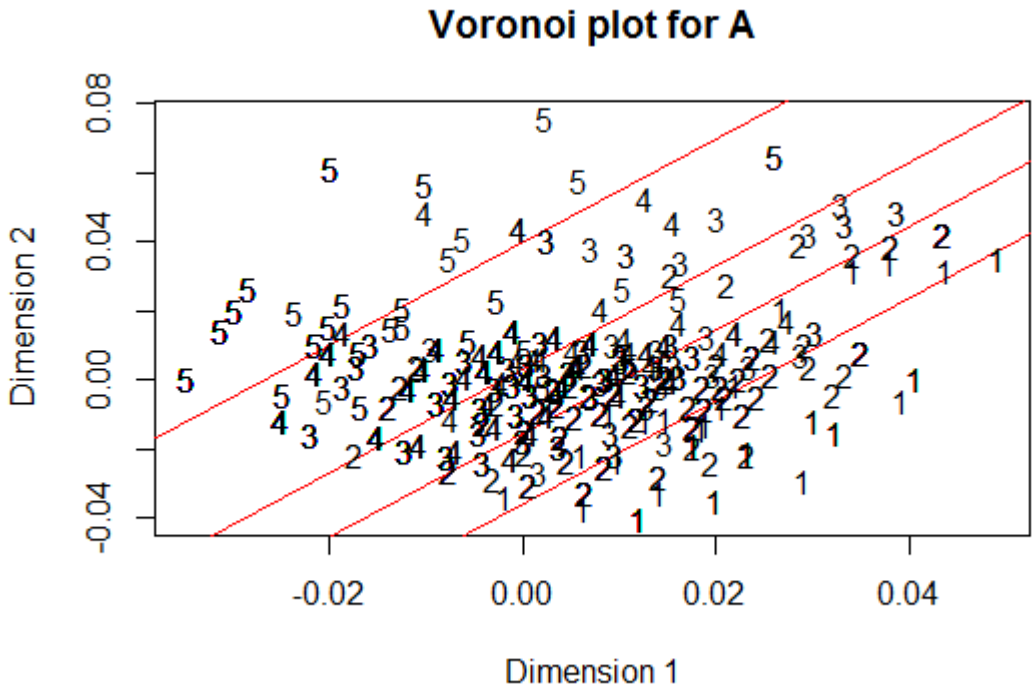


Figure 12: Figure 9A

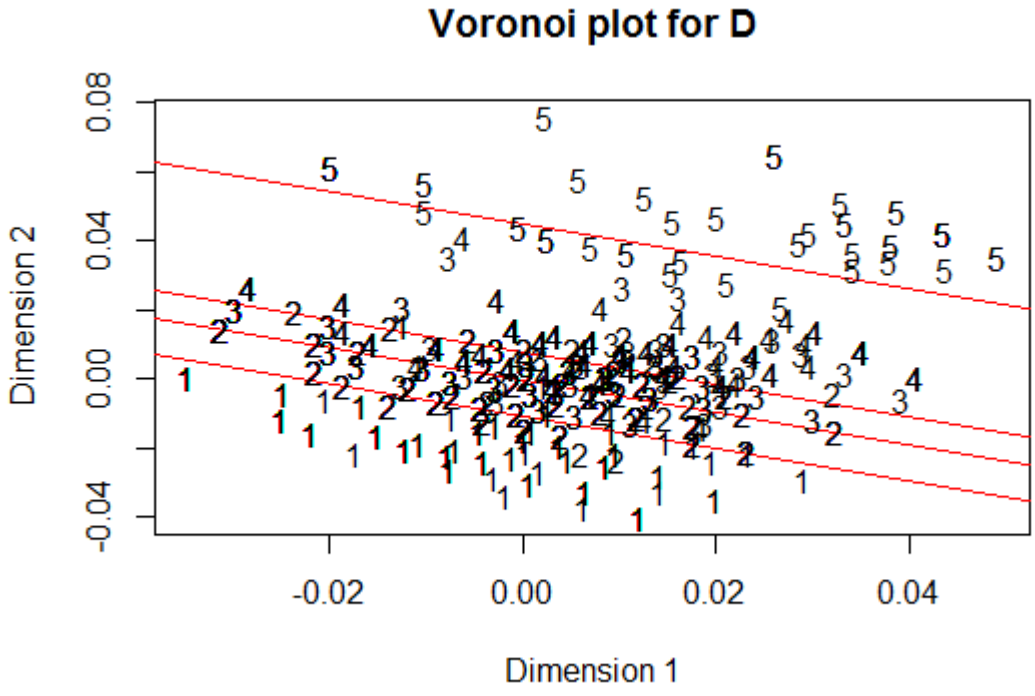


Figure 13: Figure 9D

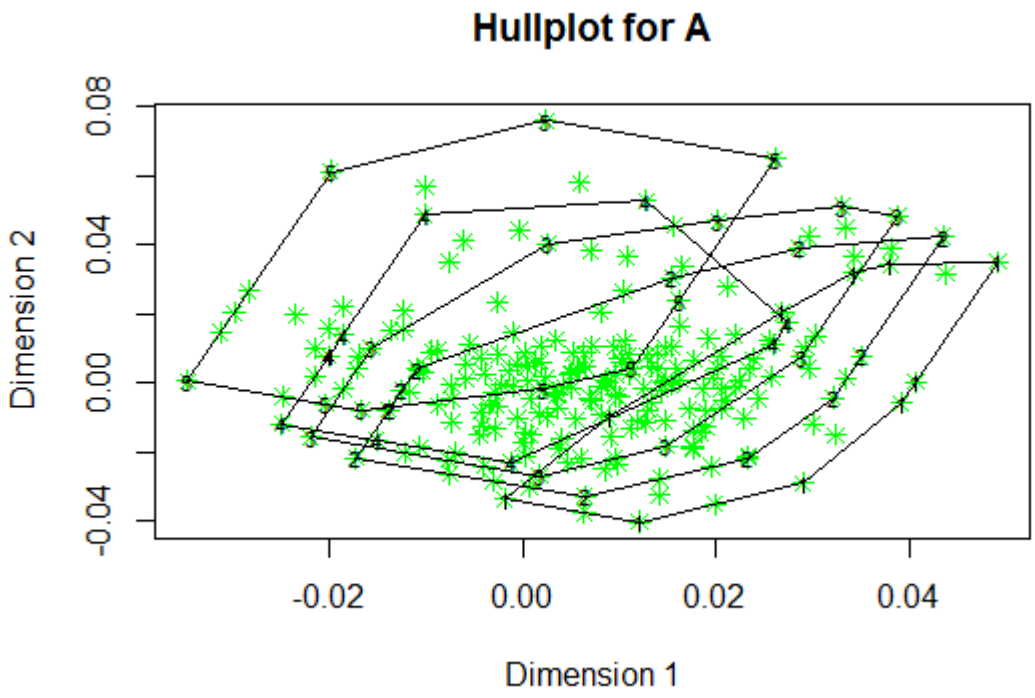
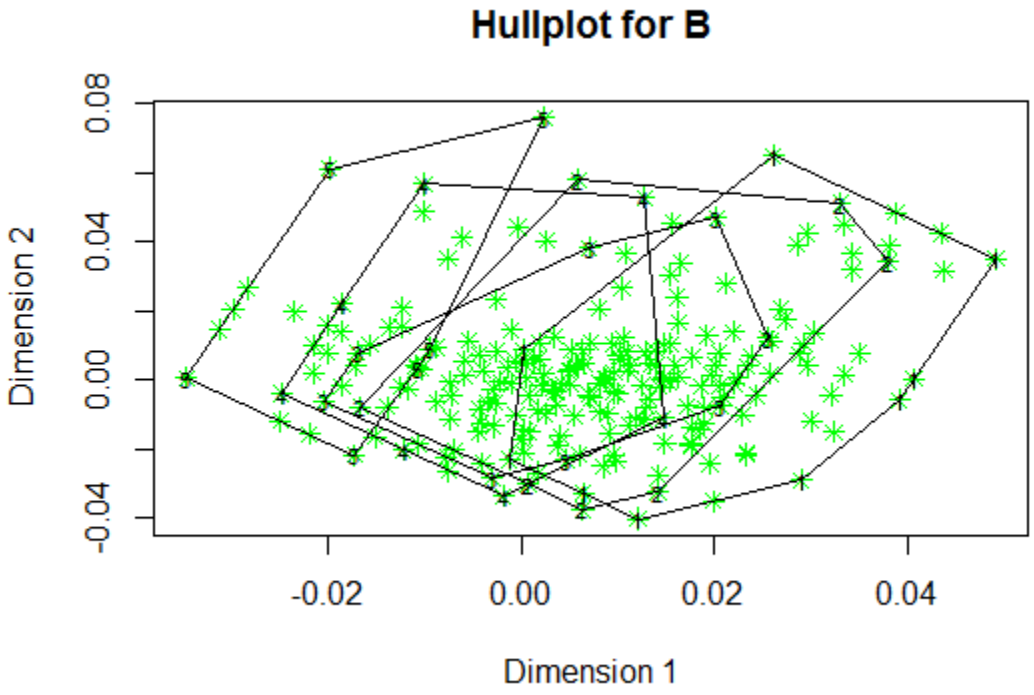


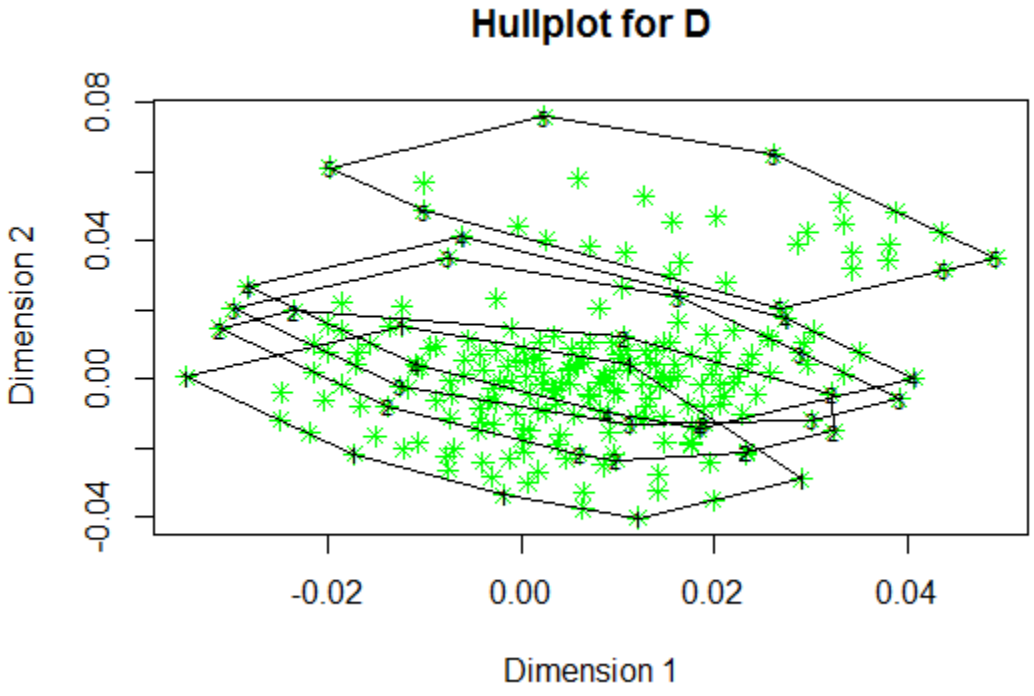
Figure 14: Figure 10A

10. Hull plot

For each single variable the object scores are mapped onto two dimensions and the convex hull for each response category is drawn.

```
# would be better to show confidence ellipses
plot(Finland.nlpca, plot.type="hullplot")
```



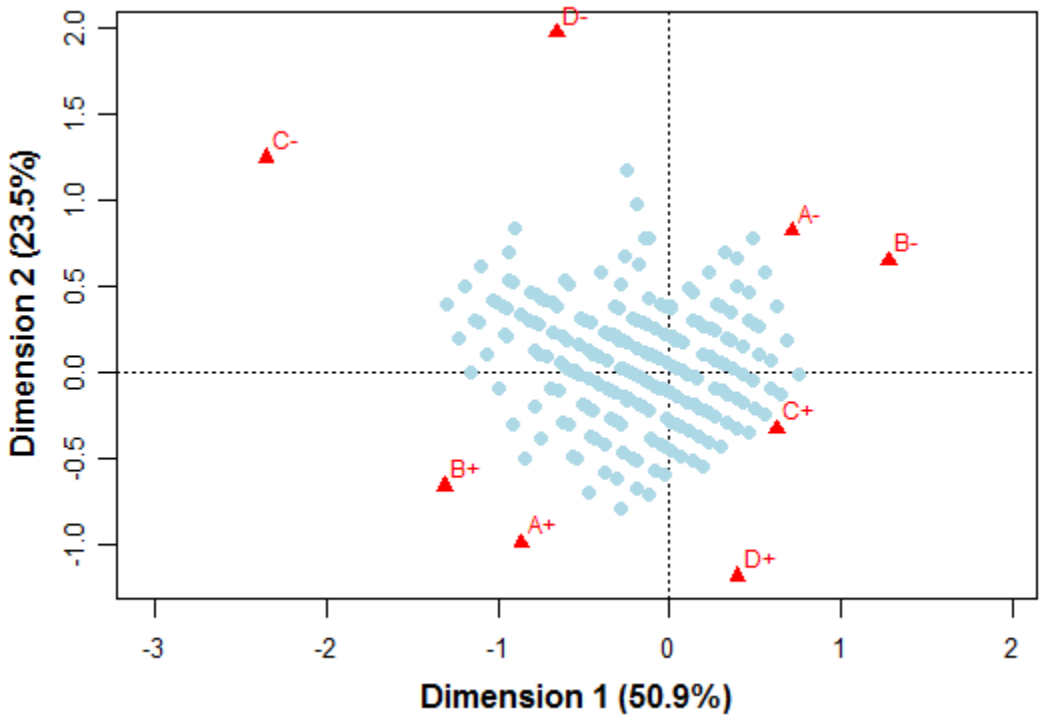


CA of doubled data

The CA method could be applied on data, which have been preprocessed in different ways. One kind of preprocessing is so called doubling, which in this example is provided on data type ratings. We assume 1-to-5

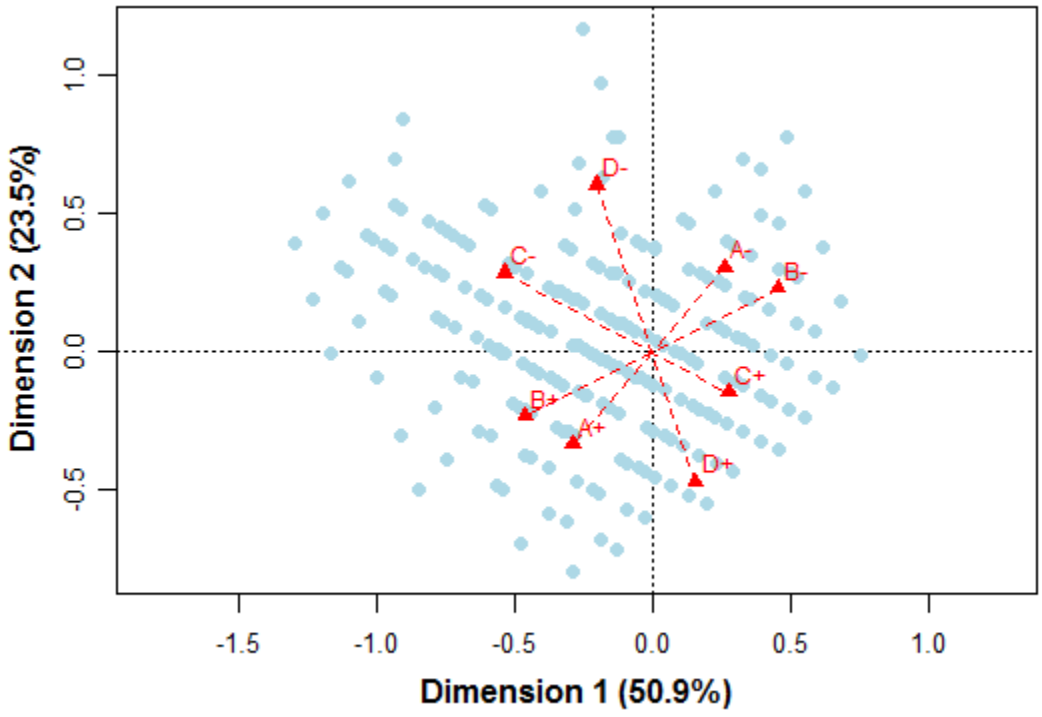
Linkert scale, which starts at one. Then the first column consists of all ratings minus 1. Since rating 1 = "strongly agree", the transformed first column measure the strength of disagreements and therefore is called *negative pole*. The second column is 4 minus the first column and measure the strength of disagreements. The second column is called *positive pole*. This kind of data transformation is called doubling. The CA of doubled data is performed as before when we have used nontransformed data.
Next plot shows the asymmetric map.

```
require(ca)
Finland.doubled <- cbind( (Finland[,1:4]-1), (5-Finland[,1:4]) )
colnames(Finland.doubled) <- paste( rep(LETTERS[1:4],2), rep(c("-","+"), each=4), sep="" )
head(Finland.doubled)
Finland.doubled.ca <- ca(Finland.doubled)
par(mar=c(4.2,4,1,1), mgp=c(2,0.7,0), font.lab=2, cex.axis=0.8)
plot(Finland.doubled.ca, col=c("lightblue","red"), labels=c(0,2), font=2, map="rowprincipal")
```



Next we connect opposite categories in column standard coordinates and draw the contribution biplot.

```
# connect opposite categories in column standard coordinates
Finland.doubled.csc <- Finland.doubled.ca$colcoord
for(j in 1:4) segments(Finland.doubled.csc[j,1],Finland.doubled.csc[j,2],
                       Finland.doubled.csc[4+j,1],Finland.doubled.csc[4+j,2], col="red", lty=2)
par(mar=c(4.2,4,1,1), mgp=c(2,0.7,0), font.lab=2, cex.axis=0.8)
plot(Finland.doubled.ca, col=c("lightblue","red"), labels=c(0,2), font=2, map="rowgreen")
Finland.doubled.ccc <- Finland.doubled.ca$colcoord * sqrt(Finland.doubled.ca$colmass)
for(j in 1:4) segments(Finland.doubled.ccc[j,1],Finland.doubled.ccc[j,2],
                       Finland.doubled.ccc[4+j,1],Finland.doubled.ccc[4+j,2], col="red", lty=2)
```



The polar positions in contribution coordinates are shown on Figure 13.

```
# just the polar positions in contribution coordinates
par(mar=c(4.2,4,1,1), mgp=c(2,0.7,0), font.lab=2, cex.axis=0.8)
plot(Finland.doubled.ca, what=c("none","all"), labels=c(0,2), font=2, map="rowgreen")
for(j in 1:4) segments(Finland.doubled.ccc[j,1],Finland.doubled.ccc[j,2],
                       Finland.doubled.ccc[4+j,1],Finland.doubled.ccc[4+j,2], col="red", lty=2)
```

The demographic group averages are shown on Figure 14

```
Finland.doubled.rpc <- Finland.doubled.ca$rowcoord %*% diag(Finland.doubled.ca$sv)
```

The demographic group averages and confidence regions are shown on Figure 15

```
# add centroids of demographic categories
Finland.doubled.rpc <- Finland.doubled.ca$rowcoord %*% diag(Finland.doubled.ca$sv)
# with confidence ellipses
source("confidenceplots.R")
require(ellipse)
par(mar=c(4.2,4,1,1), mgp=c(2,0.7,0), font.lab=2, cex.axis=0.8)
plot(Finland.doubled.ca, labels=c(0,2), what=c("none","all"))
Finland.doubled.cpc <- Finland.doubled.ca$colcoord %*% diag(Finland.doubled.ca$sv)
for(j in 1:4) segments(Finland.doubled.cpc[j,1],Finland.doubled.cpc[j,2],
                       Finland.doubled.cpc[4+j,1],Finland.doubled.cpc[4+j,2], col="red", lty=2)
confidenceplots(Finland.doubled.rpc[,1], Finland.doubled.rpc[,2], group=Finland$ga, groupcols=c(rep("blu
                groupnames=c("ma1","ma2","ma3","ma4","ma5","ma6","fa1","fa2","fa3","fa4","fa5","fa6"),sl
```

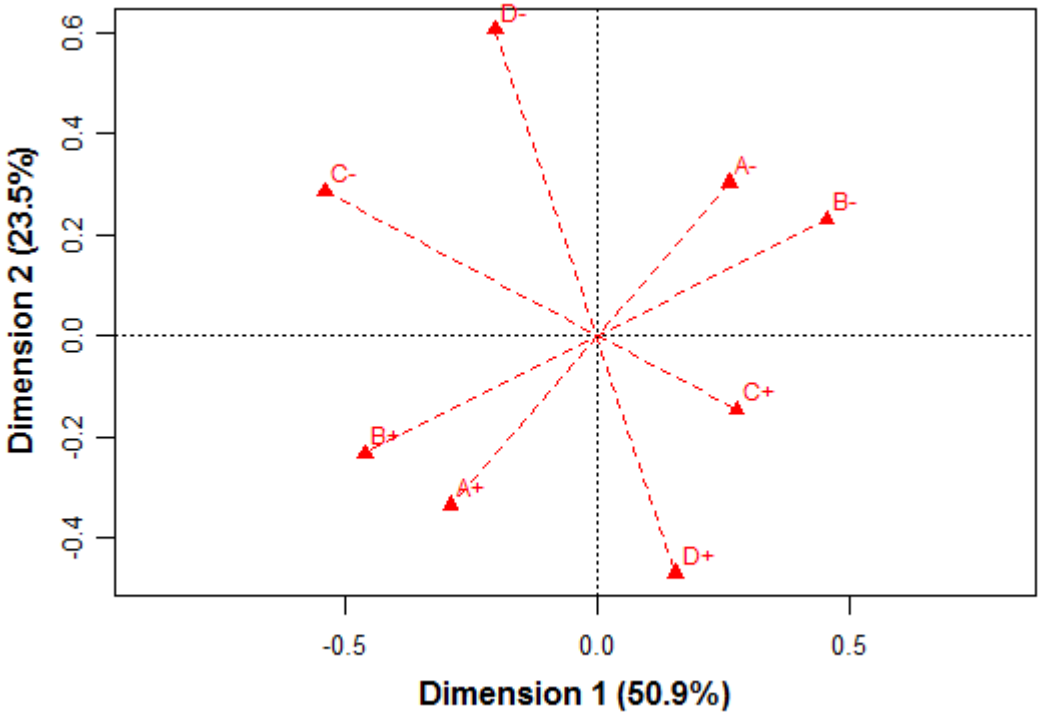


Figure 15: Figure 13

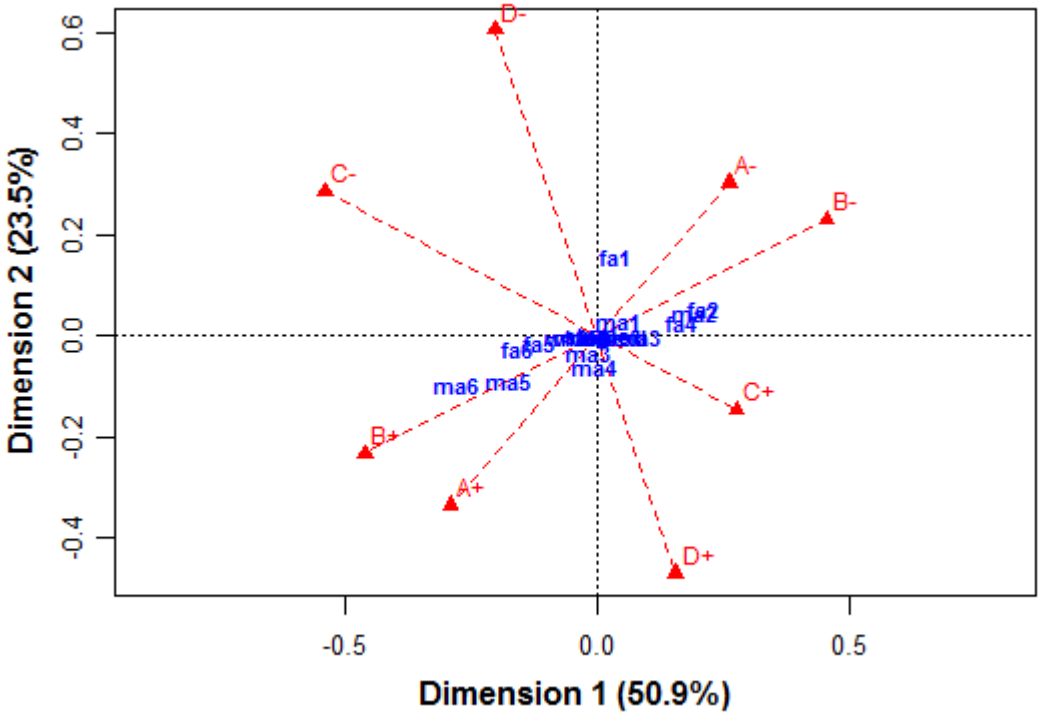
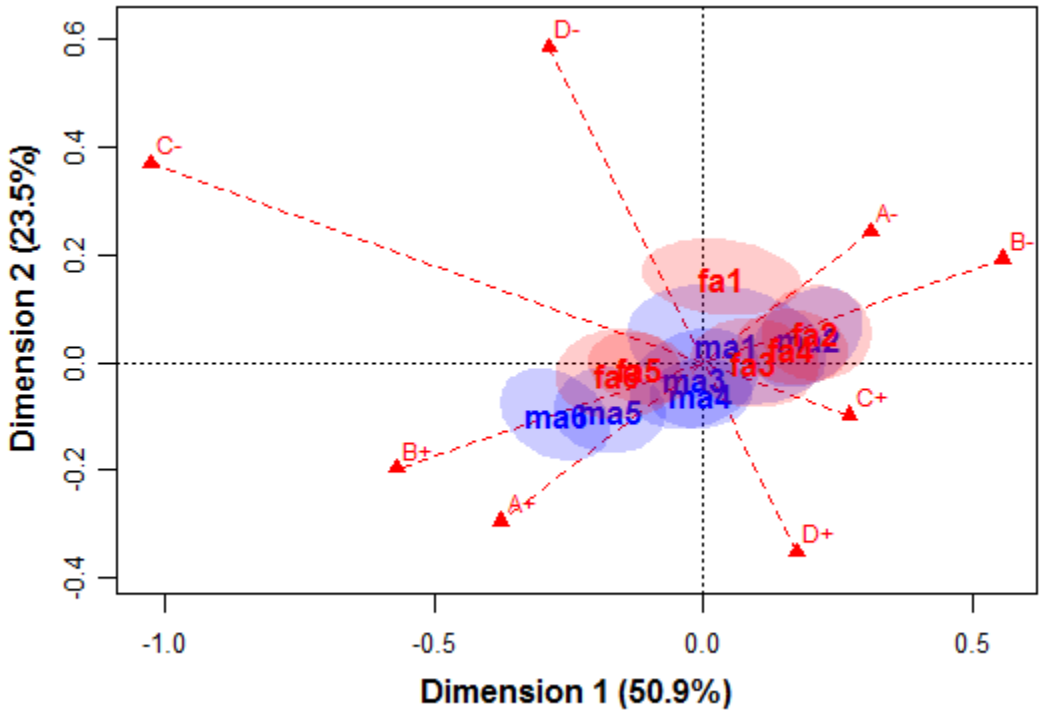


Figure 16: Figure 14



Next we add confidence ellipses to the plot and show the resultt onFigure 14. Only the confidence ellipses are shown on Figure 16.

```
#ellipses just by themselves (watch out for aspect ratio distortion! I have to correct this in next ver
confidenceplots(Finland.doubled.rpc[,1], Finland.doubled.rpc[,2], group=Finland$ga, groupcols=c(rep("bl
                groupnames=c("ma1","ma2","ma3","ma4","ma5","ma6","fa1","fa2","fa3","fa4","fa5","fa6"),s
```

Regular PCA on non-missing data

The regular PCA plot is shown on the next Figure 17:

```
Finland.pca <- prcomp(Finland[,1:4])
names(Finland.pca)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
# [1] "sdev"     "rotation" "center"   "scale"    "x"

par(mar=c(4.2,4,1,1), mgp=c(2,0.7,0), font.lab=2, cex.axis=0.8)
plot(rbind(Finland.pca$x, 10.5*Finland.pca$rotation), type="n", asp=1, xlab="PCA dim 1 (37.5%)", ylab="
abline(h=0, col="gray", lty=2)
abline(v=0, col="gray", lty=2)
points(Finland.pca$x, pch=19, col="lightblue", cex=0.9)
arrows(0, 0, 10*Finland.pca$rotation[,1], 10*Finland.pca$rotation[,2], length=0.1, angle=10, col="pink"
text(10.3*Finland.pca$rotation, labels=colnames(Finland[,1:4]), col="red", font=4)
```

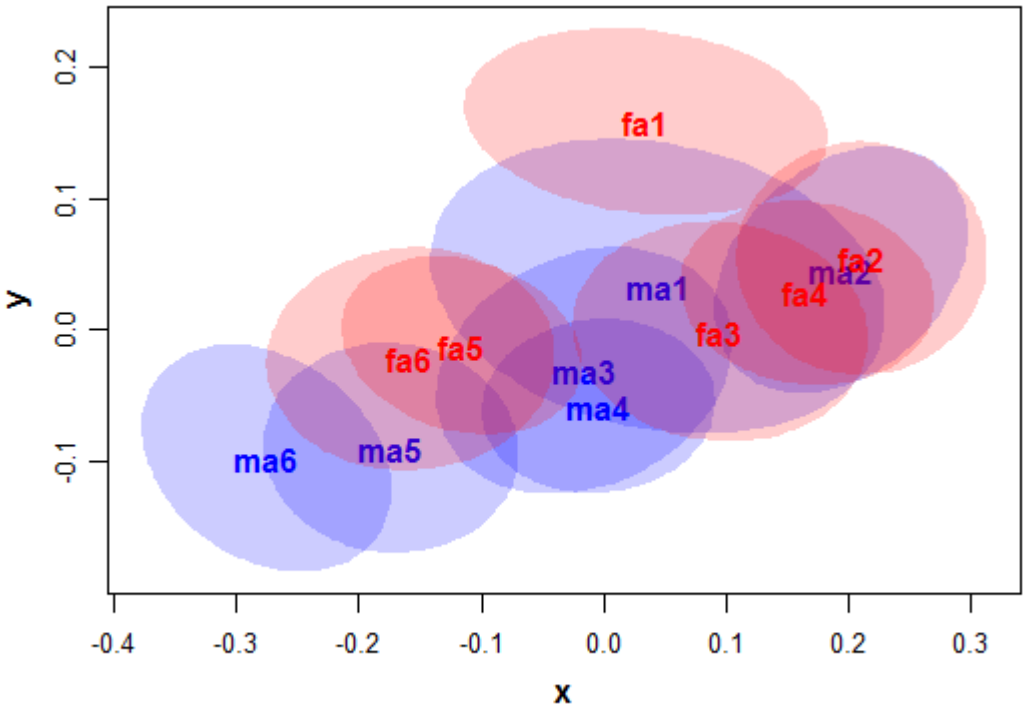
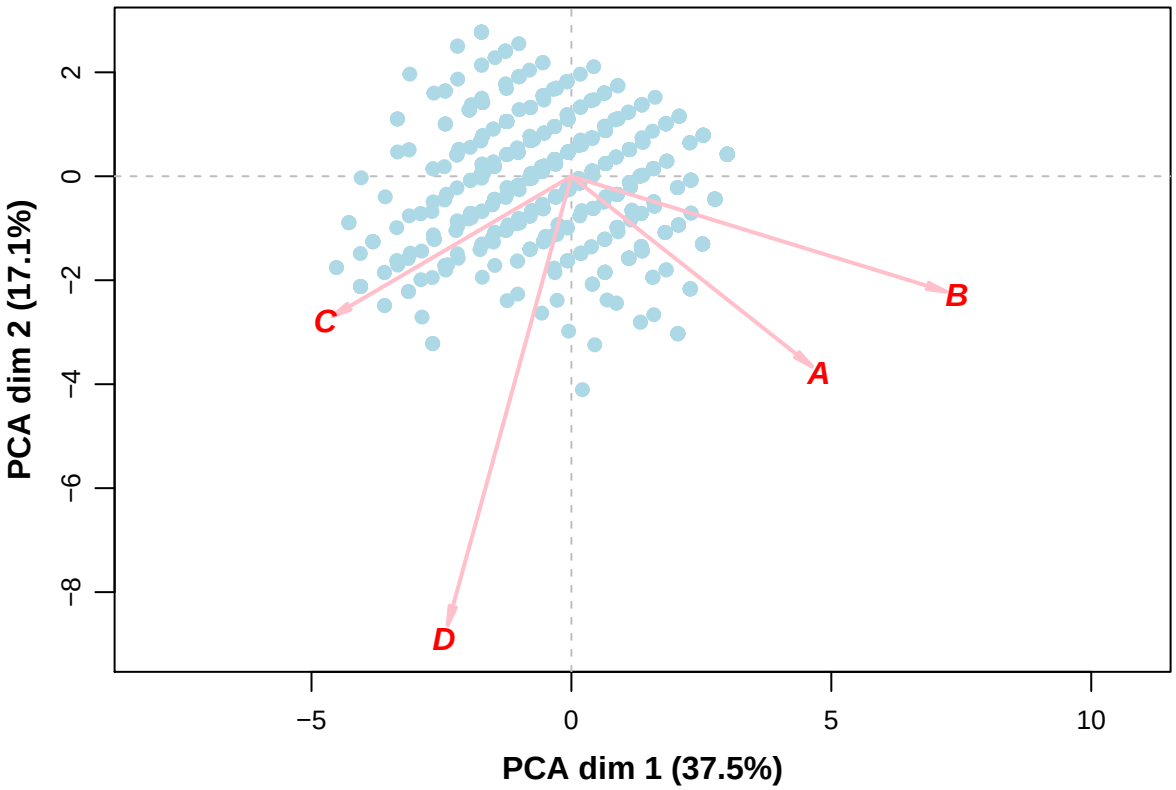


Figure 17: Figure 16



Factor analysis

The R function factanal() performs the analysis on standardized variables

The correlation matrix is:

```
round(cor(Finland[,1:4]),3)
```

```
##        A      B      C      D
## A  1.000  0.527 -0.236 -0.057
## B  0.527  1.000 -0.508 -0.138
## C -0.236 -0.508  1.000  0.291
## D -0.057 -0.138  0.291  1.000
```

Since 2 factors are too many for 4 variables, we have to set factors = 1.

```
Finland.fa <- factanal(Finland[,1:4], factors=1, rotation="none", scores="regression")
names(Finland.fa)
```

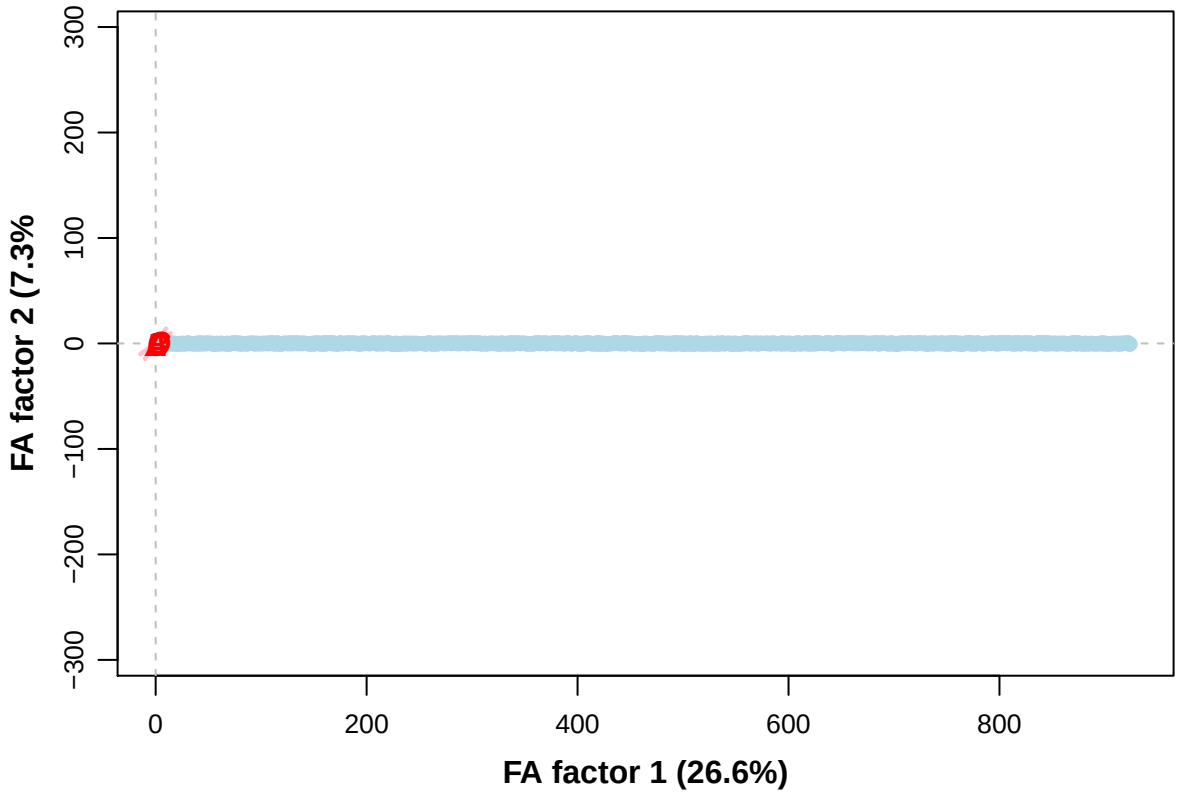
```
##  [1] "converged"    "loadings"     "uniquenesses" "correlation"
##  [5] "criteria"     "factors"      "dof"          "method"
##  [9] "scores"       "STATISTIC"    "PVAL"         "n.obs"
## [13] "call"
```

```
print(Finland.fa)
```

```
##
## Call:
## factanal(x = Finland[, 1:4], factors = 1, scores = "regression",     rotation = "none")
##
## Uniquenesses:
##     A     B     C     D
## 0.715 0.028 0.735 0.979
##
## Loadings:
##   Factor1
## A  0.534
## B  0.986
## C -0.515
## D -0.144
##
##                Factor1
## SS loadings      1.543
## Proportion Var   0.386
##
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 65.38 on 2 degrees of freedom.
## The p-value is 6.35e-15
```

Factor analysis results:

```
par(mar=c(4.2,4,1,1), mgp=c(2,0.7,0), font.lab=2, cex.axis=0.8)
plot(rbind(-Finland.fa$loadings, -0.5*Finland.fa$scores), asp=1, type="n", xlab="FA factor 1 (26.6%)", y
abline(h=0, col="gray", lty=2)
abline(v=0, col="gray", lty=2)
points(-0.5*Finland.fa$scores, pch=19, col="lightblue", cex=0.9)
arrows(0, 0, -Finland.fa$loadings[,1], -Finland.fa$loadings[,1], length=0.1, angle=10, col="pink", lwd=
text(-1.05*Finland.fa$loadings, labels=colnames(Finland[,1:4]), col="red", font=4)
```



FA with rotation:

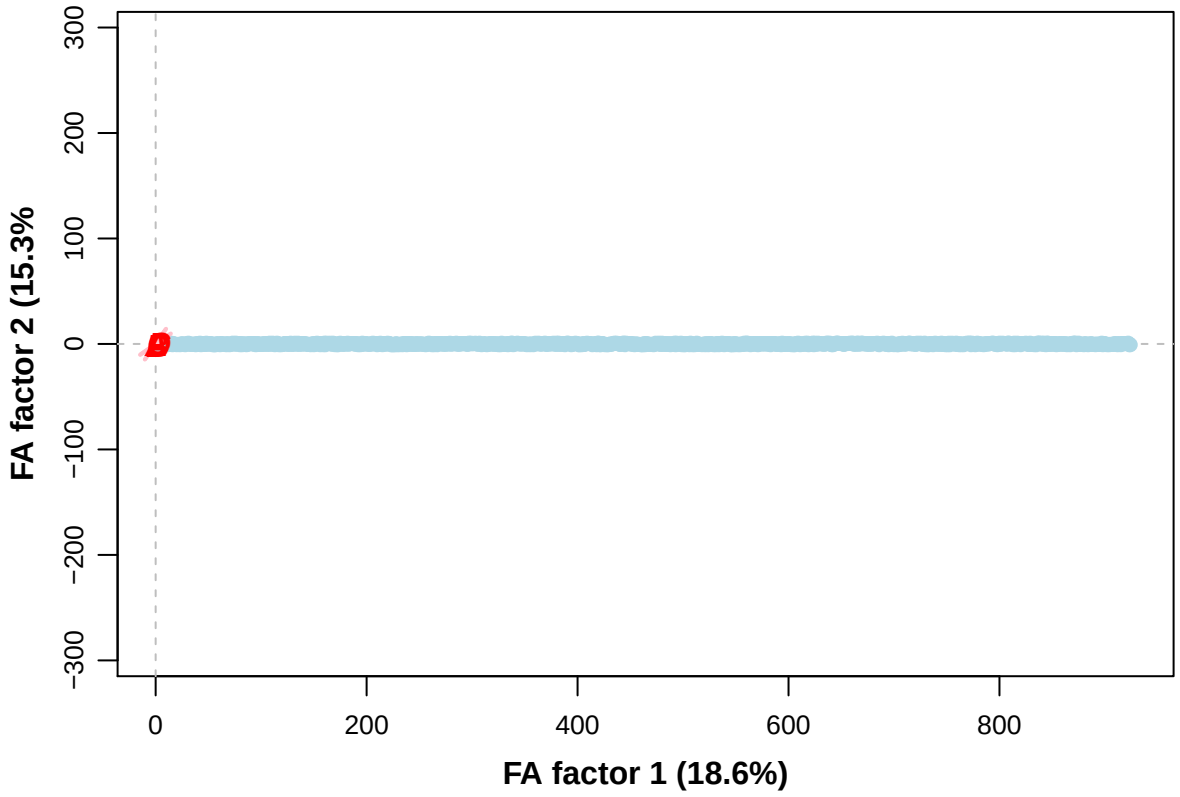
```
Finland.fa <- factanal(Finland[,1:4], factors=1, rotation="varimax", scores="regression")

print(Finland.fa)
```

```
##
## Call:
## factanal(x = Finland[, 1:4], factors = 1, scores = "regression",     rotation = "varimax")
##
## Uniquenesses:
##     A     B     C     D
## 0.715 0.028 0.735 0.979
##
## Loadings:
##   Factor1
## A  0.534
## B  0.986
## C -0.515
## D -0.144
##
##                Factor1
## SS loadings      1.543
## Proportion Var   0.386
##
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 65.38 on 2 degrees of freedom.
```

```
## The p-value is 6.35e-15
```

Plot FA results:

```
par(mar=c(4.2,4,1,1), mgp=c(2,0.7,0), font.lab=2, cex.axis=0.8)
plot(rbind(-Finland.fa$loadings, -0.5*Finland.fa$scores), asp=1, type="n", xlab="FA factor 1 (18.6%)", y
abline(h=0, col="gray", lty=2)
abline(v=0, col="gray", lty=2)
points(-0.5*Finland.fa$scores, pch=19, col="lightblue", cex=0.9)
arrows(0, 0, -Finland.fa$loadings[,1], -Finland.fa$loadings[,1], length=0.1, angle=10, col="pink", lwd=
text(-1.05*Finland.fa$loadings, labels=colnames(Finland[,1:4]), col="red", font=4)
```



K-means clustering of respondents using MCA coordinates

Finland is the set of complete cases

```
require(ca)
```

```
## Loading required package: ca
```

```
Finland.B <- mjca(Finland[,1:4], ps="")$Burt
Finland.Z <- mjca(Finland[,1:4], ps="", reti=T)$indmat
Finland.csc <- mjca(Finland[,1:4], ps="")$colcoord
rownames(Finland.Z) <- 1:nrow(Finland.Z)
colnames(Finland.Z) <- colnames(Finland.B)
# notice that the division by 8 in the next line is becasue of the 9 variables in Finland
```

```
Finland.rpc <- Finland.Z %*% Finland.csc/9
summary(mjca(Finland[,1:4]))
```

```
##
## Principal inertias (eigenvalues):
##
##  dim    value      %   cum%   scree plot
##  1      0.116786  44.9  44.9  *************
##  2      0.090602  34.8  79.7  **********
##  3      0.009457   3.6  83.3  *
##  4      0.006314   2.4  85.7  *
##  5      0.001032   0.4  86.1
##  6      0.000624   0.2  86.4
##  7      00000000   0.0  86.4
##         -------- -----
##  Total: 0.260281
##
##
## Columns:
##      name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
## 1  |  A:1 |   12  848   57 | -374 159  15 |  778 689  83 |
## 2  |  A:2 |   58  847   46 | -351 769  61 |  112  78   8 |
## 3  |  A:3 |   85  545   38 |  -90 122   6 | -168 423  26 |
## 4  |  A:4 |   58  375   44 |   68  45   2 | -184 329  22 |
## 5  |  A:5 |   37  830   63 |  785 759 193 |  240  71  23 |
## 6  |  B:1 |   34  834   65 | -333 121  32 |  807 713 245 |
## 7  |  B:2 |   67  738   47 | -368 666  78 | -120  71  11 |
## 8  |  B:3 |   47  543   46 | -115 100   5 | -242 443  30 |
## 9  |  B:4 |   62  460   45 |  135 127  10 | -219 333  33 |
## 10 |  B:5 |   40  804   66 |  836 782 236 |  141  22   9 |
## 11 |  C:1 |  118  902   39 |  379 899 145 |  -25   4   1 |
## 12 |  C:2 |   91  898   42 | -312 611  76 | -214 288  46 |
## 13 |  C:3 |   17  636   52 | -429 588  27 |  122  47   3 |
## 14 |  C:4 |   12  756   54 | -542 510  30 |  376 246  19 |
## 15 |  C:5 |   12  845   66 | -207  20   4 | 1327 825 231 |
## 16 |  D:1 |   48  970   47 |  382 924  60 |   85  46   4 |
## 17 |  D:2 |   94  829   36 | -119 227  11 | -193 602  39 |
## 18 |  D:3 |   56  277   42 |  -11   4   0 |  -89 273   5 |
## 19 |  D:4 |   38  557   46 | -149 305   7 |  135 252   8 |
## 20 |  D:5 |   14  960   59 |  -59   3   0 | 1008 956 155 |
```

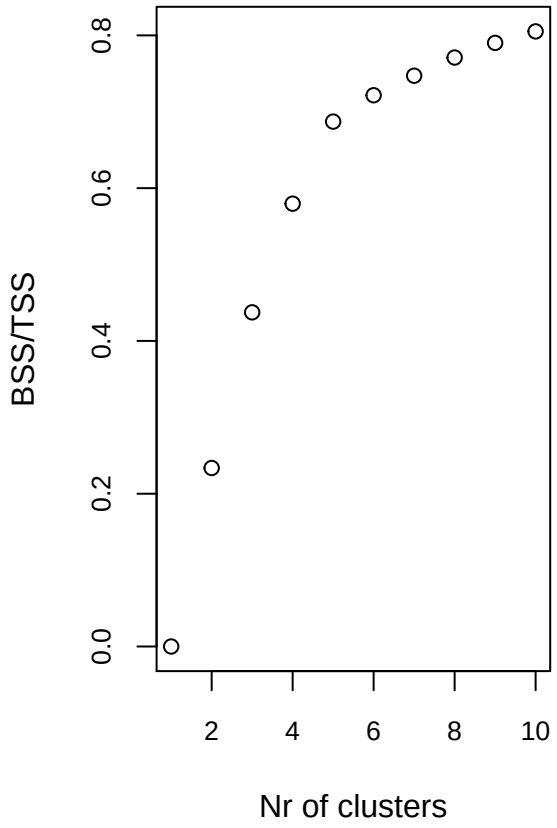
We use 4 dimensions (which is also the maximum factor analysis will allow) and loop on k-means algorithm to decide how many clusters

```
Finland.BW <- rep(0, 10)
for(nc in 2:10) {
  Finland.km <- kmeans(Finland.rpc[,1:4], centers=nc, nstart=20, iter.max=200)
  Finland.BW[nc] <- Finland.km$betweenss/Finland.km$totss
}
Finland.BW
```

```
##  [1] 0.0000000 0.2336672 0.4374201 0.5796749 0.6871295 0.7216313 0.7471689
##  [8] 0.7709886 0.7901674 0.8051961
```

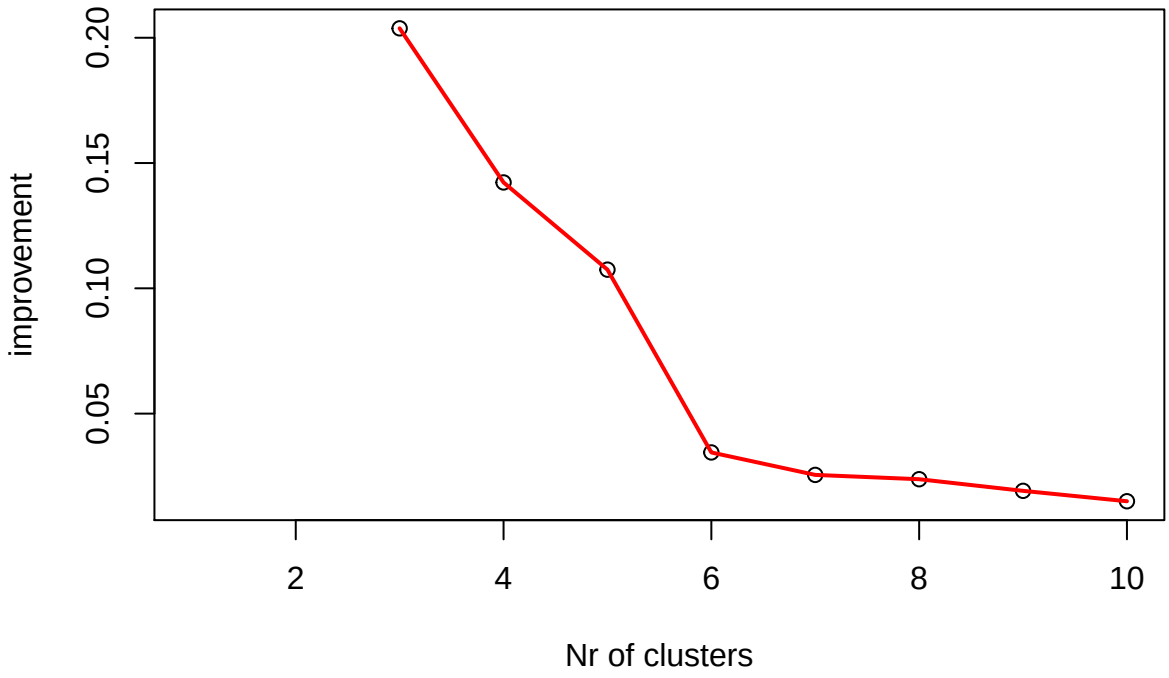
Plot the proportion of between-cluster variance:

```
par(mar=c(4.2,4,1,2), cex.axis=0.8, mfrow=c(1,2))
plot(Finland.BW, xlab="Nr of clusters", ylab="BSS/TSS")
```



Plot the increments in between-cluster variance:

```
Finland.BWinc <- Finland.BW[2:10]-Finland.BW[1:9]
plot(1:10, c(NA,NA, Finland.BWinc[2:9]), xlab="Nr of clusters", ylab="improvement")
lines(3:10, Finland.BWinc[2:9], col="red", lwd=2)
```



If it looks like 5-cluster solution, than it is a good choice

```
Finland.km5 <- kmeans(Finland.rpc[,1:4], centers=5, nstart=20, iter.max=200)
Finland.km5$betweenss/Finland.km5$totss
```

```
## [1] 0.6871295
```

Cluster sizes:

```
Finland.km5$size
```

```
## [1] 167 230 105 210 212
```

Relate clusters to 9 variables in Finland and average those in a cluster by the original Finland counts:

```
Finland.means <- matrix(0,nrow=5,ncol=4)
rownames(Finland.means) <- c("clus1","clus2","clus3","clus4","clus5")
colnames(Finland.means) <- colnames(Finland)[1:4]
for(j in 1:4) Finland.means[,j] <- tapply(Finland[,j], Finland.km5$cluster, mean)
round(Finland.means,1)
```

```
##         A   B   C   D
## clus1 4.4 4.8 1.1 2.2
## clus2 3.6 3.7 1.5 2.5
## clus3 2.2 1.2 3.4 3.3
## clus4 2.6 2.1 2.1 2.4
## clus5 2.8 2.8 1.8 2.5
```

Used and useful links

Linting, M., Meulman, J.J., Groenen, P.J.F., & Van der Kooij, A.J. (2007). Nonlinear principal components analysis: Introduction and application. Psychological Methods

Homogeneity Analysis in R:The Package homals

Package 'homals'

Package 'ca'

Oleg Nenadic and Michael Greenacre, Computation of Multiple Correspondence Analysis, with code in R

Michael Greenacre, Biplots in practice

Multiple Correspondence Analysis Essentials: Interpretation and application to investigate the associations between categories of multiple qualitative variables - R software and data mining

Mike Bendixen, A Practical Guide to the Use of Correspondence Analysis in Marketing Research, Marketing Bulletin, 2003, 14, Technical Note 2.

Michael Greenacre, Correspondence Analysis in Practice, Third Edition

An Example R Markdown

Writing Mathematic Formulas in Markdown