Croissants and Wedges in Multiple Correspondence Analysis

Jan de Leeuw

Version 2.01, December 16, 2015

Contents

Note: This is a working paper which will be expanded/updated frequently. The directory gifi.stat.ucla.edu/croissants has a pdf copy of this article, the complete Rmd file with all code chunks, and R and C files with the code. Unfortunately the Norway data cannot be shared, so that part of the paper is not reproducible. If you want to knit the Rmd file, remove the Norway section.

1 Introduction

In multiple correspondence analysis (MCA) we frequently observe two-dimensional scatter plots in which the points are on or close to a convex function, with shape similar to a parabola. This is often referred to as the *horseshoe effect*, but that name is not quite appropriate, since horseshoes fold back at the endpoints. Thus we suggest to call such plots *croissants*, which seems especially suitable if there is considerable dispersion around the convex function.

Croissants have a complicated history. In data analysis they were first described in the context of scale analysis by Guttman (1950). The MCA eigenvectors of perfect scales satisfy a second order difference equation, whose solution are the discrete orthogonal polynomials. The first and second component define the croissant. In the French data analysis literature the horseshoe or croissant is consequently called the *Effet Guttman*.

The next major contribution was Lancaster (1958), who described a family of bivariate distributions for which the singular value decomposition of the density consisted of orthogonal polynomials. The most famous member of the family is the bivariate normal, which has the Hermite-Chebyshev polynomials as its components. Lancaster's work was generalized and extended by his students and co-workers. It was taken up by Benzécri's school, in particular in the thesis of Naouri (1970).

In ecology horseshoes were first discussed in the influential paper of Hill (1974). They were considered a nuisance, and various techniues were developed to get rid of them (Hill and Gauch 1980). Ecologists work with the Gaussian abundance model for environmental gradients, and Ihm and van Groenewoud (1975) showed this resulted in horseshoes.

In multidimensional scaling quadratic structures were first discussed and analyzed by Levelt, Van De Geer, and Plomp (1966) for musical intervals (a parabola) and by De Gruijter (1967) for political parties (an ellipse). More information on horseshoes in the MDS context is in Diaconis, Goel, and Holmes (2008) and De Leeuw (2007).

The application of Hermite-Chebyshev polynomials to the multivariate normal case in MCA was outlined in section 3.8 of De Leeuw (1974). Gifi (1980) pointed out for the first time that oscillation matrices and total positivity were the relevant mathematical theories, see also section 9.2 of Gifi (1990). Total positivity was used in a general theory by Schriever (1983), Schriever (1985) that covers all previously discussed special cases. The appropriate way to patch the decomposition of the various bivariate marginals together in the multivariate case is due to De Leeuw (1982), see also Bekker and De Leeuw (1988). Much of the work of the Gifi school on horseshoes was summarized in Van Rijckevorsel (1987). #MCA

We give a very brief introduction to MCA to fix the terminology. We start with n measurements on m categorical variables. Variable j has k_j categories. We then expand each variable to an indicator matrix G_j, indicating category membership. Thus G_j is an $n\times k_j$ binary matrix with exactly one element equal to one in each column, and zeroes everywhere else. Define $G:=\begin{bmatrix}G_1\mid\cdots\mid G_m\end{bmatrix}$ and $C=G'G$. The matrix C, which contains the bivariate cross tables of the variables, is called the *Burt Matrix* (*Tableau de Burt*). Also define $D:=\mathbf{diag}(C)$, the diagonal matrix of univariate marginals. Both C and D are of order $K:=\sum k_j$.

MCA is defined as solving the generalized eigenvalue problem $CY=mDY\Lambda$, with $Y'DY=I$. The generalized eigenvectors Y are called the *category quantifications*, and the centroids $X:=\frac{1}{m}GY$ are called the *object scores*. In most cases we do not use all eigenvectors in the data analysis but only a small number of them. The category quantifications are in an $\sum k_j\times p$ matrix, and the object scores are in an $n\times p$ matrix. Note that with the normalization we have chosen $0\leq\Lambda\leq I$ and $X'X=\frac{1}{m}\Lambda$.

In our computations we use the package `geigen` (Hasselman and Lapack authors 2015). Or, alternatively, we define $E:=D^{-\frac12}CD^{-\frac12}$, compute eigenvalues Z such that $EZ=mZ\Lambda$, and then set $Y=D^{-\frac12}Z$.

Also, because of the singularities in G, there is one generalized eigenvalue equal to one (with a corresponding generalized eigenvector y that has all elements equal) and there are $m-1$ generalized eigenvalues equal to zero.

2 Synthetic Data: Normal Distribution

One simple way to reliably generate croissants is to discreticize a sample from a multinormal. Of course there are various parameters to consider. The function `discreteNormal()` allows one to choose the size of the sample n, the number of variables m, the correlation between the variables r, and the discretization points (aka knots).

```
formals ("discreteNormal")
```

```
## $n
##
##
## $m
##
##
## $r
##
##
## $knots
```

Throughout this section we choose $n = 10,000$. Our first normal croissant (category quantifications on the left, object scores on the right) has four variables, knots at the integers from -3 to +3, and all correlations equal to .75. We see somewhat more scatter or filling in the object score croissant, basically because object scores are convex combinations of category quantifications.

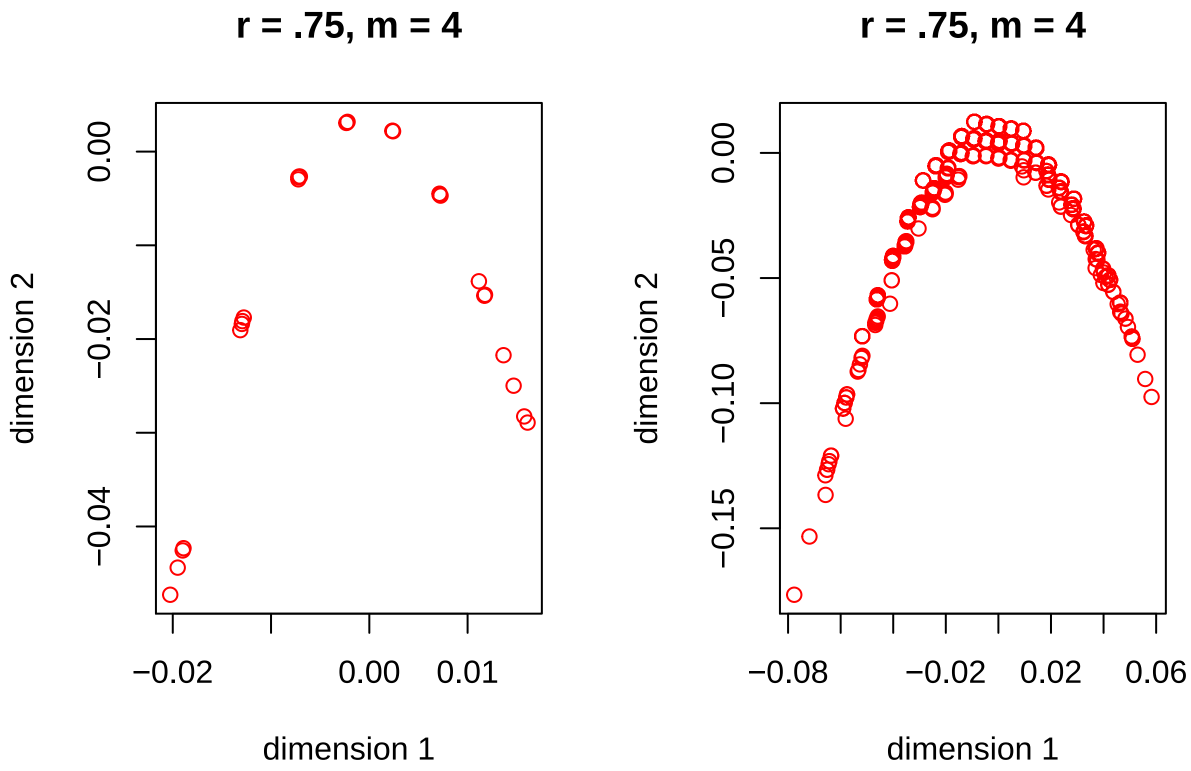

Figure 1: Normal Croissants, m = 4, r = .75

If we decrease the correlation to .25 the croissant crumbles. A correlation of .25 is too close to independence, which means that not enough variation will be captured by the first two eigenvalues. Nevertheless if we discard the outliers the croissant is still there.

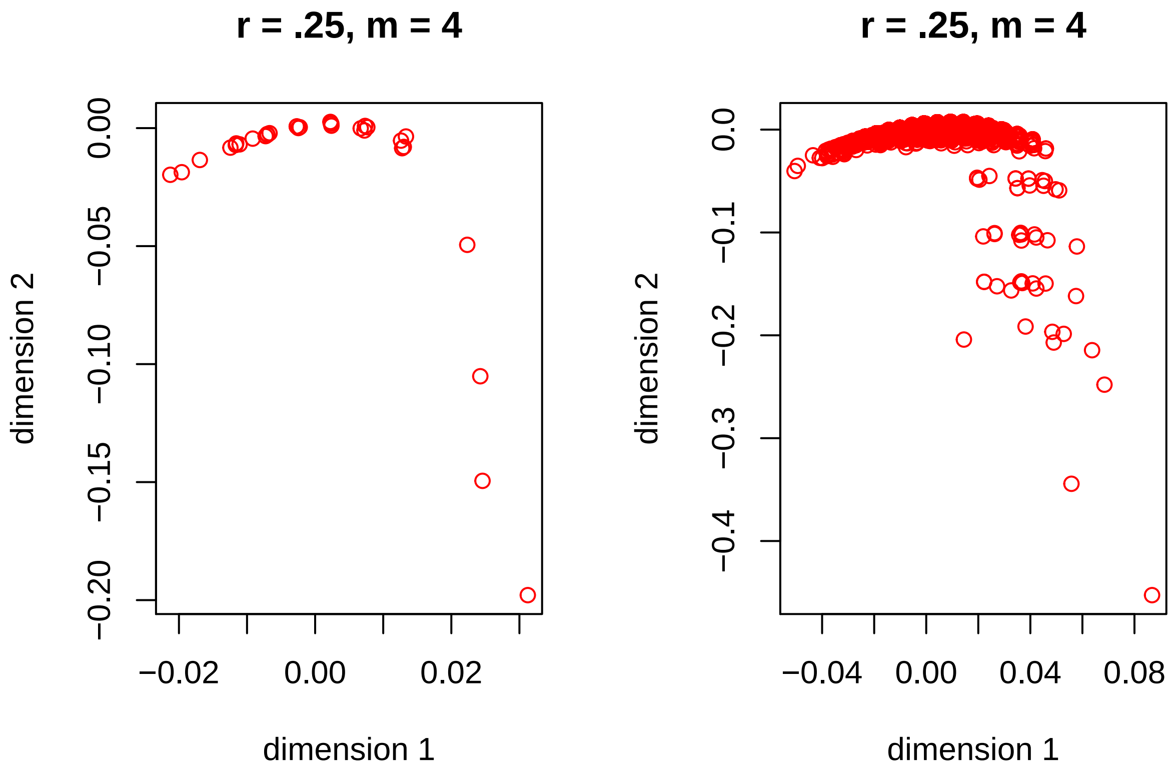

Figure 2: Normal Croissants, m = 4, r = .25

In the third simulation we increase the number of variables to 25, with correlation .75. The croissants are tight and symmetric.

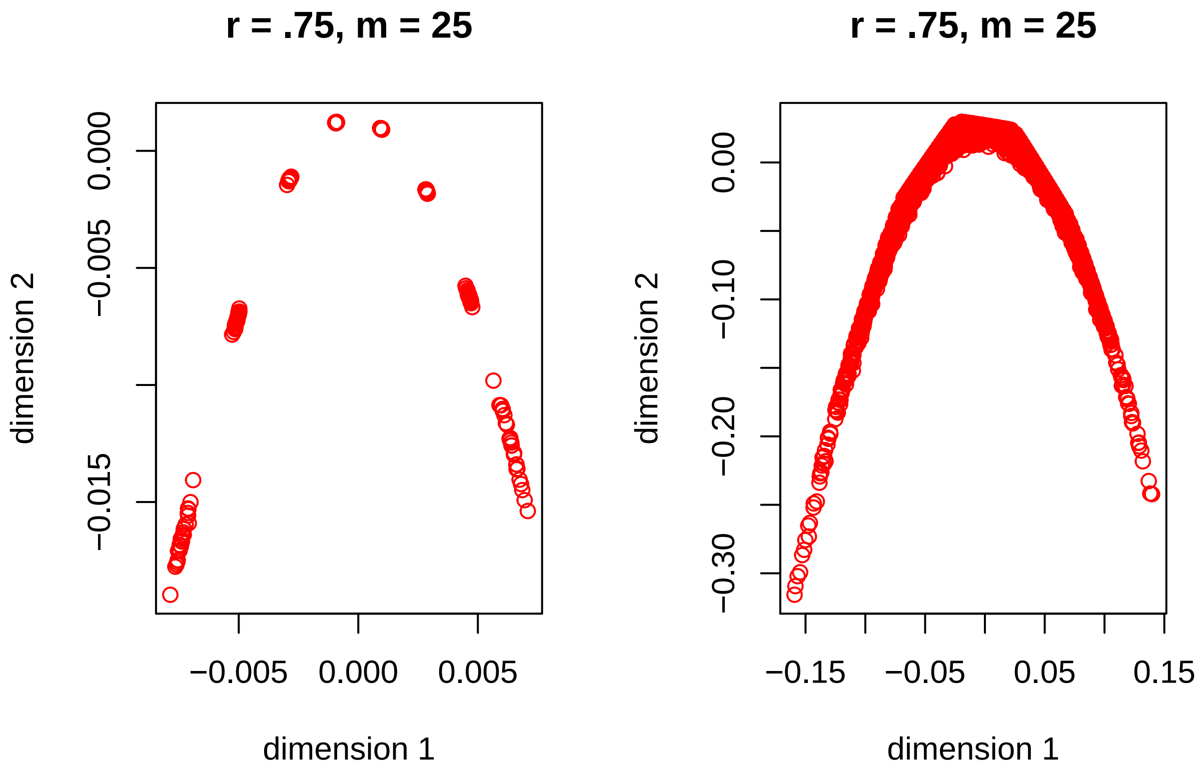

Figure 3: Normal Croissants, m = 25, r = .75

Moreover for 25 variables we still see a clear croissant if the correlation is .25, although there obviously is more filling. Lower correlation means objects scores will be convex combinations of more category quantifications, and thus they will cluster more around the origin, which is the fattest part of the croissant.

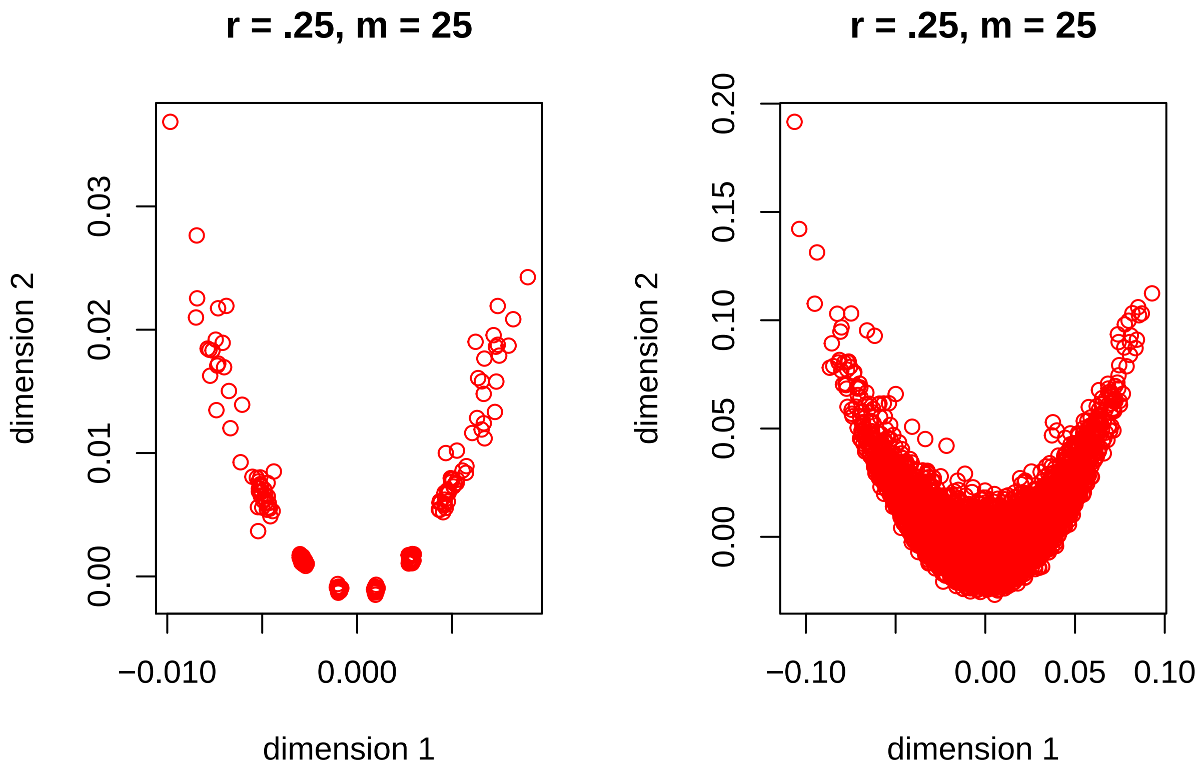

Figure 4: Normal Croissants, m = 25, r = .25

We can also study the influence of skewness by choosing the knots to be `c(0,1,2,3)`, keeping the correlaton between the four variables at .75. A truncated versions of the croissant appears.

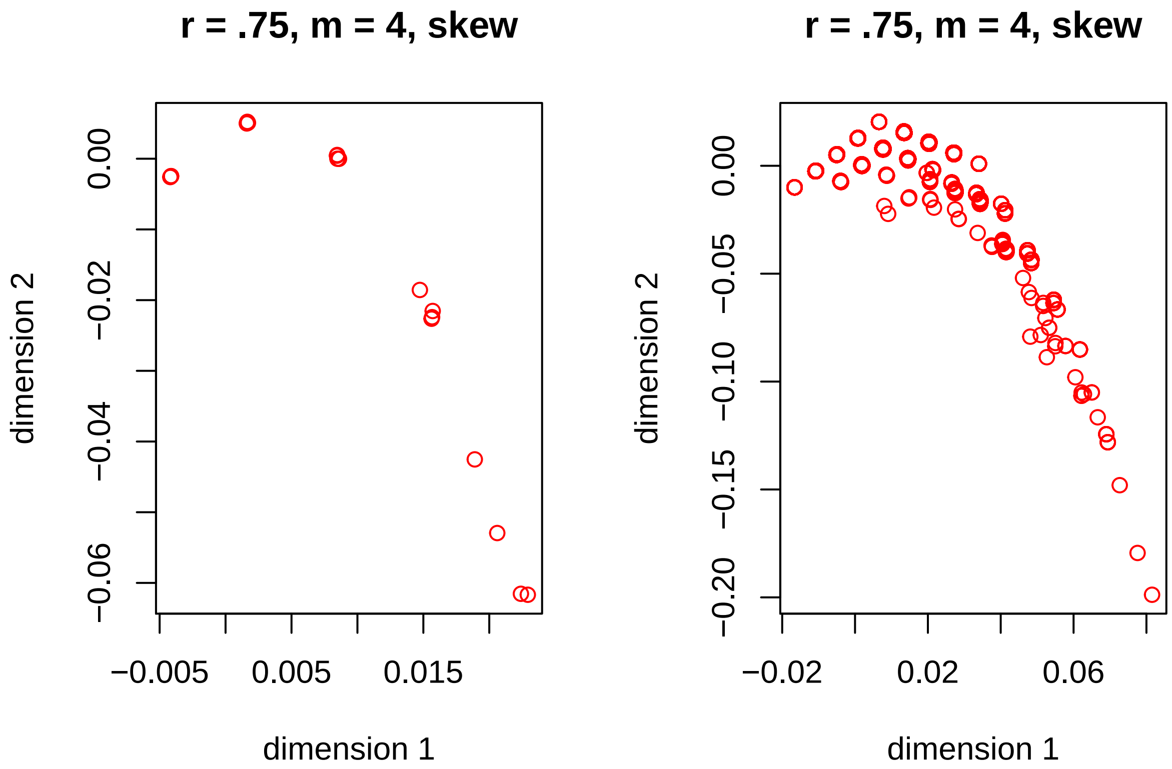

Figure 5: Normal Croissants, m = 4, r = .75, skew

Finally we can choose a correlation matrix which does not have all correlations the same. We use the correlation matrix

```
##        [,1]   [,2]   [,3]   [,4]
## [1,]  1.000 -0.027  0.729  0.008
## [2,] -0.027  1.000  0.001  0.343
## [3,]  0.729  0.001  1.000 -0.027
## [4,]  0.008  0.343 -0.027  1.000
```

This has the effect of perturbing the croissant, because it becomes a mixture of different croissants for the different variables.

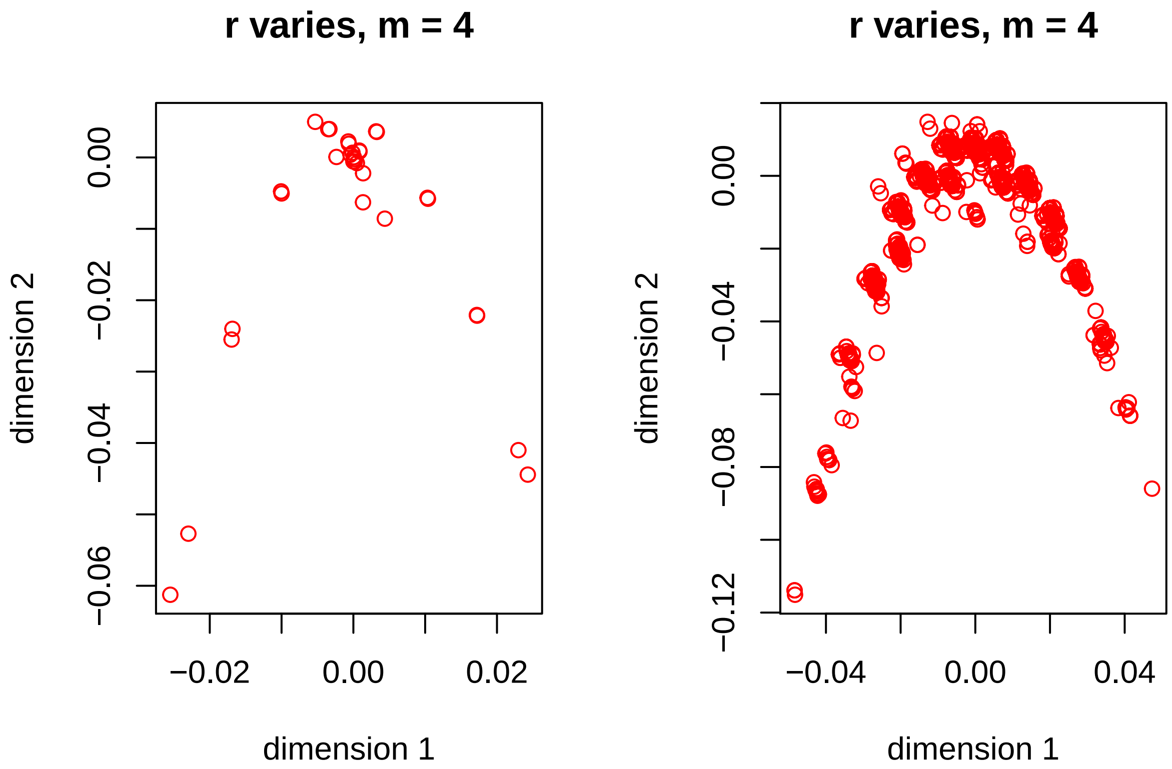

Figure 6: Normal Croissants, m = 4, r is various

In real data we have a combination of all these effects. We will have low correlations and/or skewness for some of the variables. Thus croissants can be asymmetric and have a lot of scatter. And, of course, there may be deviations from underlying normality as well, which can easily destroy the croissant.

3 Real Data: GALO

The GALO example has served us well for almost 40 years. The objects (individuals) are 1290 school children in the sixth grade of elementary school in the city of Groningen (Netherlands) in 1959. There are four variables

- Gender: M/F.
- IQ: The original range (60 to 144) has been categorized into 9 ordered categories.
- SES:
  - LoWC = Lower white collar;
  - MidWC = Middle white collar;
  - Prof = Professional, Managers;
  - Shop = Shopkeepers;
  - Skil = Schooled labor;
  - Unsk = Unskilled labor.
- Teacher's Advice Secondary Education:

- Agr = Agricultural;
- Ext = Extended primary education;
- Gen = General;

- Grls = Secondary school for girls;
- Man = Manual, including housekeeping;

- None = No further education;
- Uni = Pre-University.

3.1 MCA

We use the first two dimensions, which have eigenvalues 0.5391591 and 0.3915176. The category quantifications from MCA for each variable are in Figure 7: GALO Category Quantifications. Small croissants are visible throughout.

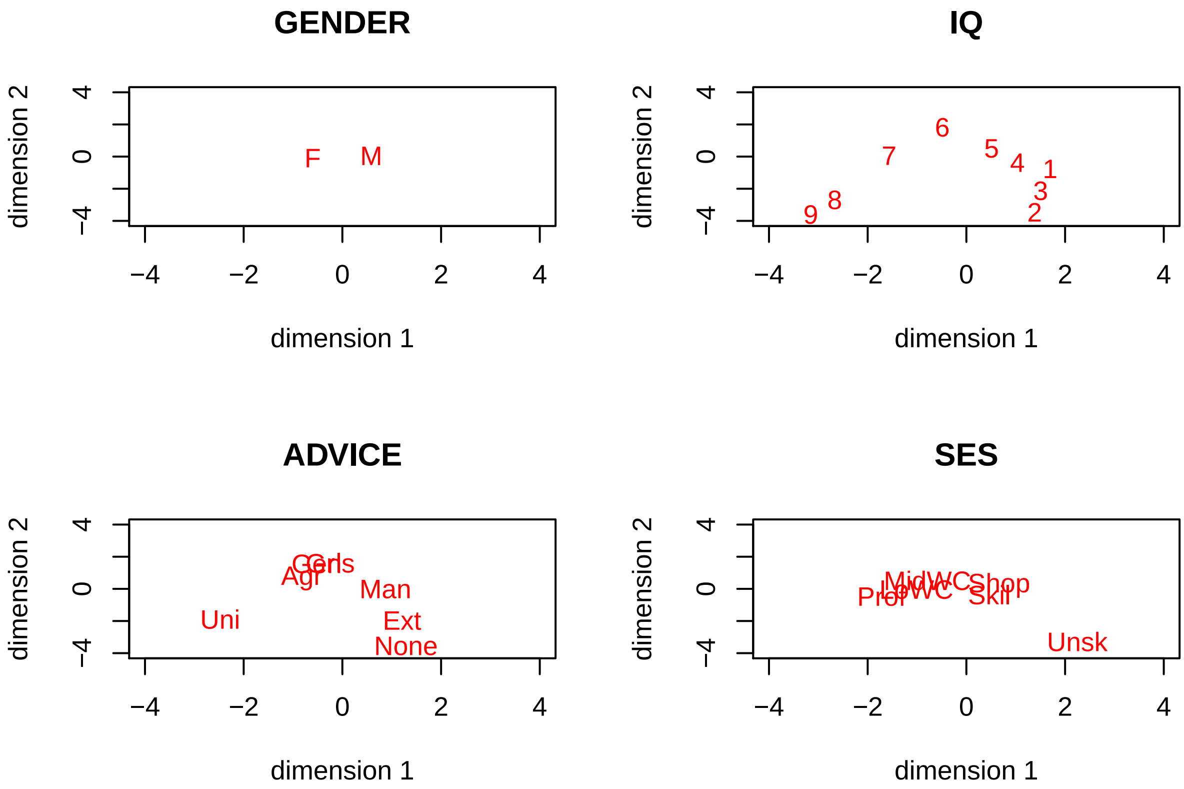

Figure 7: GALO Category Quantifications

The croissant is somewhat clearer if we put the category quantification of all variables in a single plot.

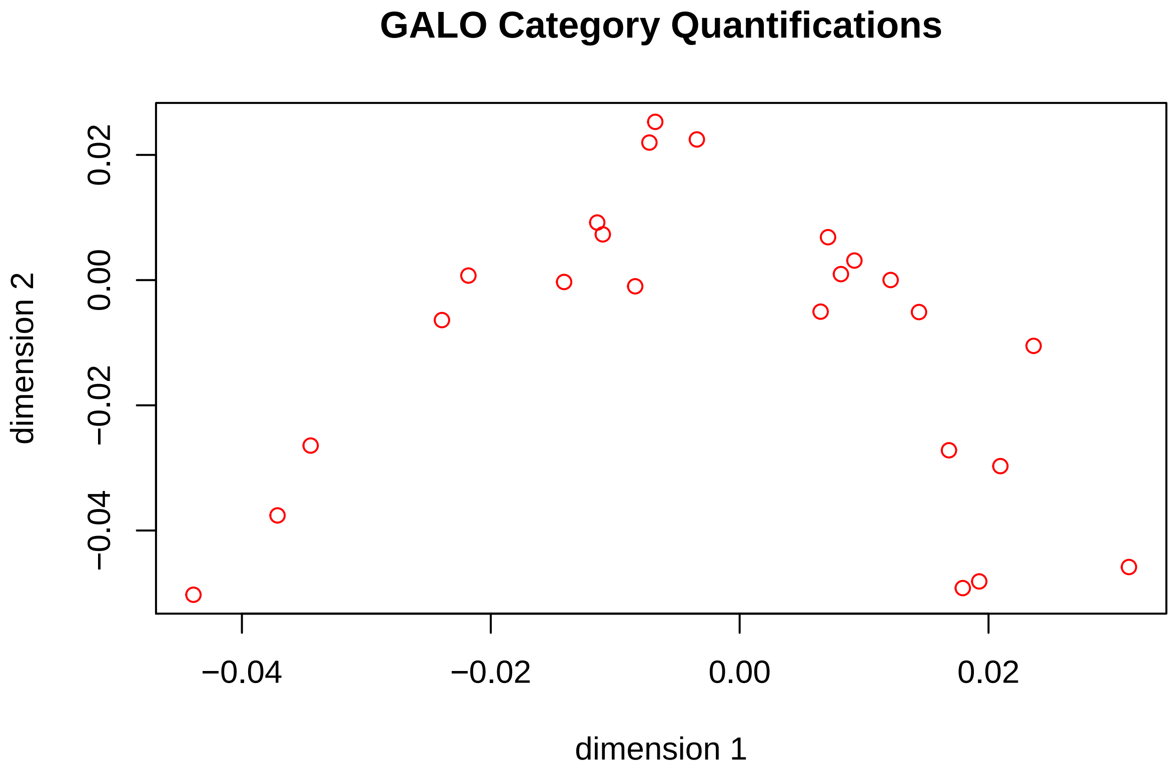

Figure 8: GALO Category Quantifications

We plot the object scores in a somewhat nonstandard way, by putting unnormalized object scores (centroids of category quantifications) within the hull of the category quantifications. This illustrates that, at least to some extent, the shape of the object score plot is determined by the shape of the category quantification plot. We see an obvious croissant for these data.

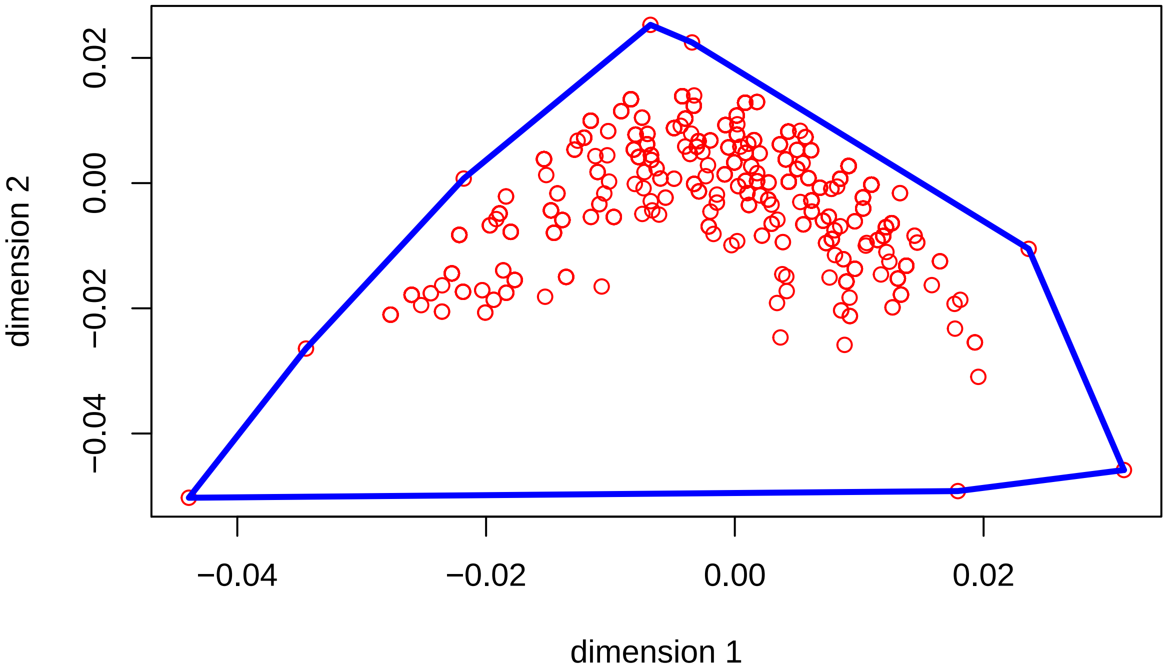

Figure 9: GALO Object Scores in Convex Hull of Quantifications

3.2 Two-step MCA: PRIMALS

One of the standard ways to get rid of MCA croissants is to impose constraints on the category quantifications. In his canonical MCA paper Guttman (1941) already argued that only the first dimension of an MCA should be used in linear multivariate analysis, because the remaining dimensions are components of non-linearity. The MCA does not decompose variance, as PCA does in classical multivariate analysis, but it decomposes chi-square, and thus its interpretation should be completely different.

There are various ways to incorporate this insight. The simplest one is to use the first MCA dimension to quantify the variables, and to use these quantified variables in a standard linear technique such as PCA or regression. The combination of MCA quantification and PCA is sometimes called PRIMALS (Meulman and Gifi 1981).

The correlation matrix of the quantified variables is

```
##           SEX        IQ       ADV       SES
## SEX 1.0000000 0.2250425 0.1530414 0.2624973
## IQ  0.2250425 1.0000000 0.7838499 0.3591835
## ADV 0.1530414 0.7838499 1.0000000 0.3809945
## SES 0.2624973 0.3591835 0.3809945 1.0000000
```

with eigenvalues (divided by 4)

```
## [1] 0.53915911 0.23737936 0.17062083 0.05284071
```

and eigenvectors

```
##             [,1]       [,2]       [,3]        [,4]
## [1,] -0.2968564  0.8447797 -0.4381085  0.07927491
## [2,] -0.5990117 -0.2807262 -0.2625353 -0.70246210
## [3,] -0.5930407 -0.3551810 -0.1553433  0.70570362
## [4,] -0.4487360  0.2852833  0.8455794 -0.04738025
```

Using the first two eigenvectors we can make a PCA plot.

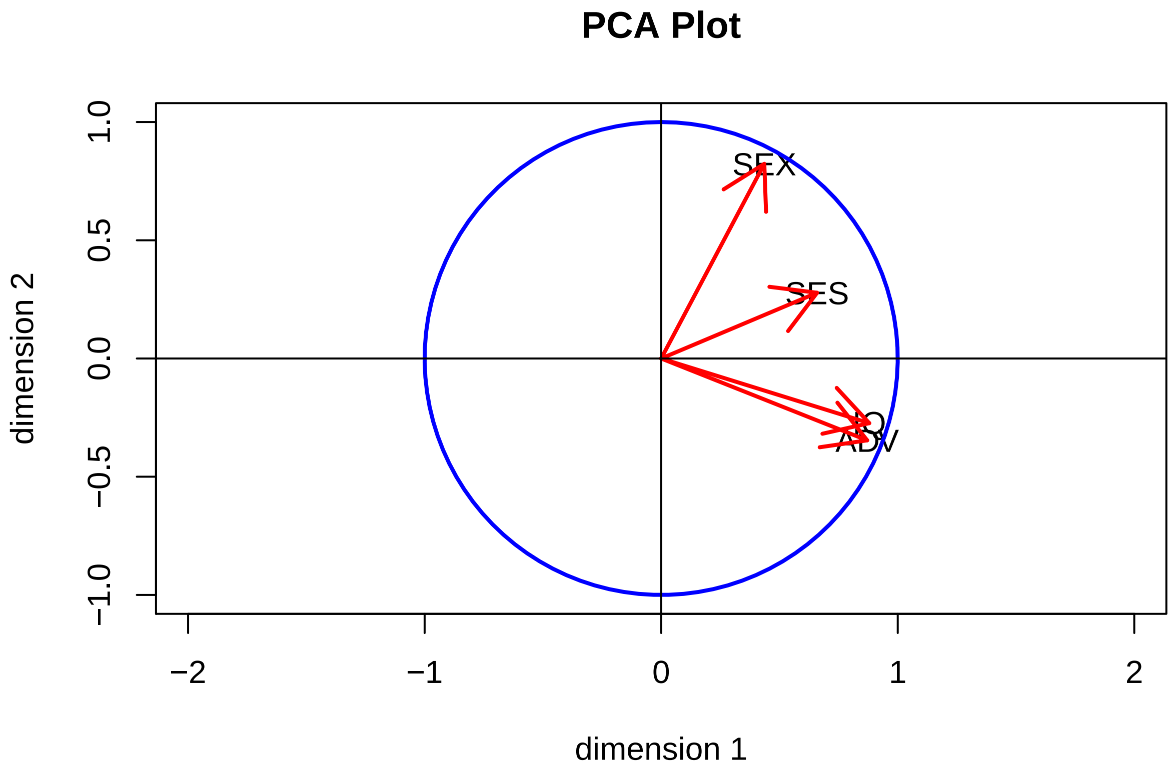

Figure 10: PRIMALS GALO PCA Plot

To show how this analysis is related to MCA, we first normalize the category quantifications of variable j to $y_j'D_jy_j = 1$. We then create the first column of Y by multiplying the y_j by the corresponding entries of the first eigenvector of the correlation matrix. And do the same for the second column of Y. We still have $Y'DY = I$, while **tr** $Y'CY$ is equal to the sum of the first two eigenvalues of the correlation matrix, i.e. 0.7765385. For the MCA the corresponding sum is 0.9306767.

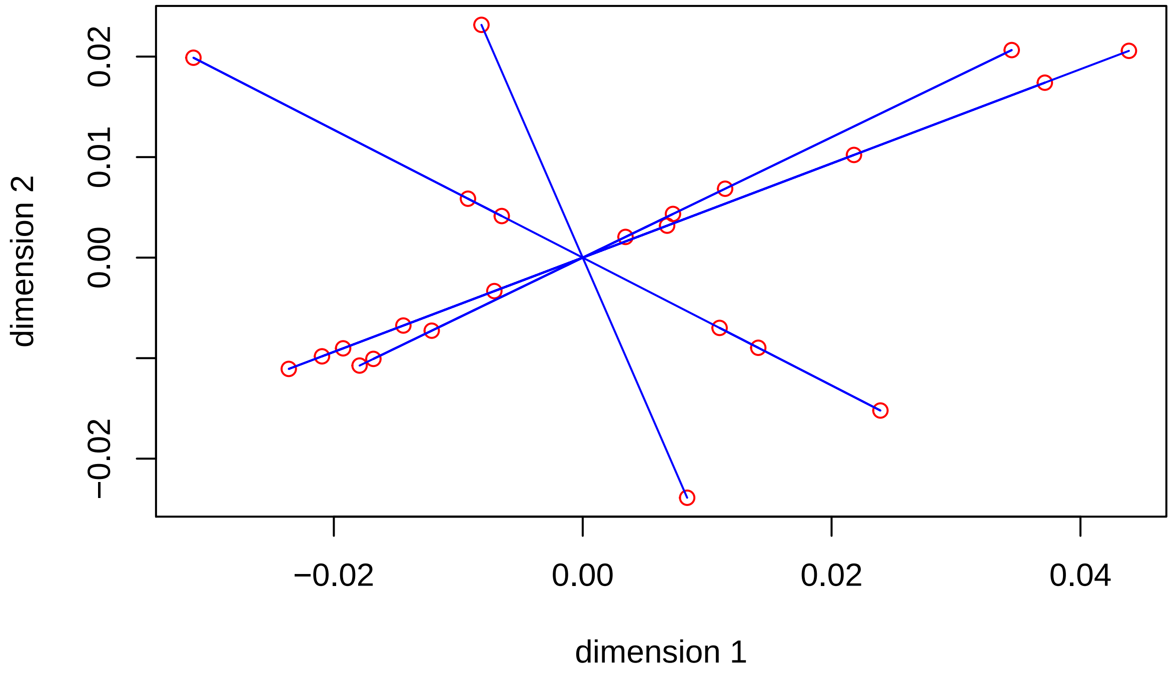

Figure 11: PRIMALS GALO Category Quantifications

The corresponding object scores are the left singular vectors of the matrix of quantified variables. There is no croissant in sight any more.

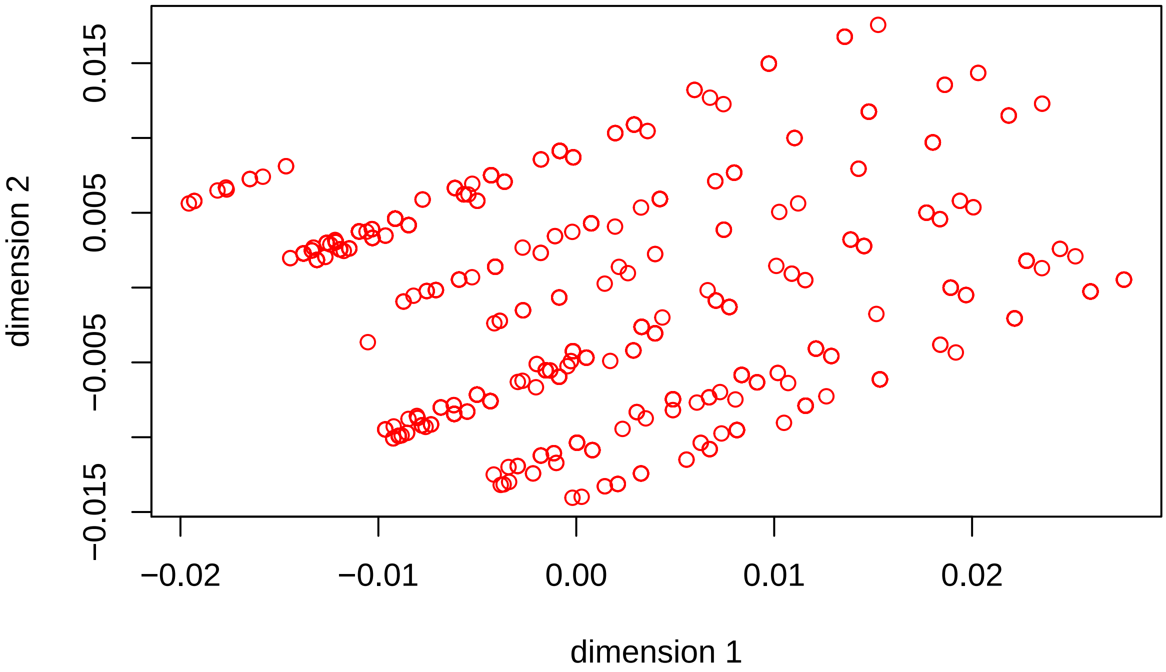

Figure 12: PRIMALS GALO Object Scores

3.3 Nonlinear PCA: PRINCALS

The PRIMALS approach is a two-step approach, in the sense that we first maximize by choosing the largest eigenvalue of the Burt table and then we maximize by choosing the largest eigenvalues of the induced correlation matrix. Thus there are two different criteria involved, and this could be seen as a disadvantage. In the PRINCALS approach we choose quantifications of the variables that maximize the sum of the first p eigenvalues of the induced correlation matrix. For $p = 1$ this is PRIMALS, but for $p > 1$ the two approaches are different.

We perform the necessary computations, in the GALO case for $p = 2$, using the `homals` package (De Leeuw and Mair 2009). Alternatively, the `aspect` package (Mair and De Leeuw 2010) could be used.

The correlation matrix of the quantified variables is

```
##             SEX         IQ         ADV       SES
## SEX  1.00000000 0.01951188 -0.07525007 0.3791437
## IQ   0.01951188 1.00000000  0.58402262 0.4182533
## ADV -0.07525007 0.58402262  1.00000000 0.3333477
## SES  0.37914374 0.41825334  0.33334774 1.0000000
```

with eigenvalues (divided by 4)

```
## [1] 0.4825477 0.3013908 0.1146095 0.1014520
```

and eigenvectors

```
##            [,1]       [,2]       [,3]         [,4]
## [1,] -0.1899217  0.8085833  0.5568847 -0.001470354
## [2,] -0.5980856 -0.2443347  0.1488185 -0.748630276
## [3,] -0.5556153 -0.3817234  0.3664577  0.641317128
## [4,] -0.5454494  0.3752077 -0.7303706  0.168115701
```

The largest eigenvalue is now 0.4825477, while for PRIMALS it was the larger value 0.5391591. But the sum of the two largest eigenvalues for PRINCALS is 0.7839385, while for PRIMALS it is the smaller value 0.7765385.

Using the first two eigenvectors of the induced correlation matrix we can make a PCA plot.

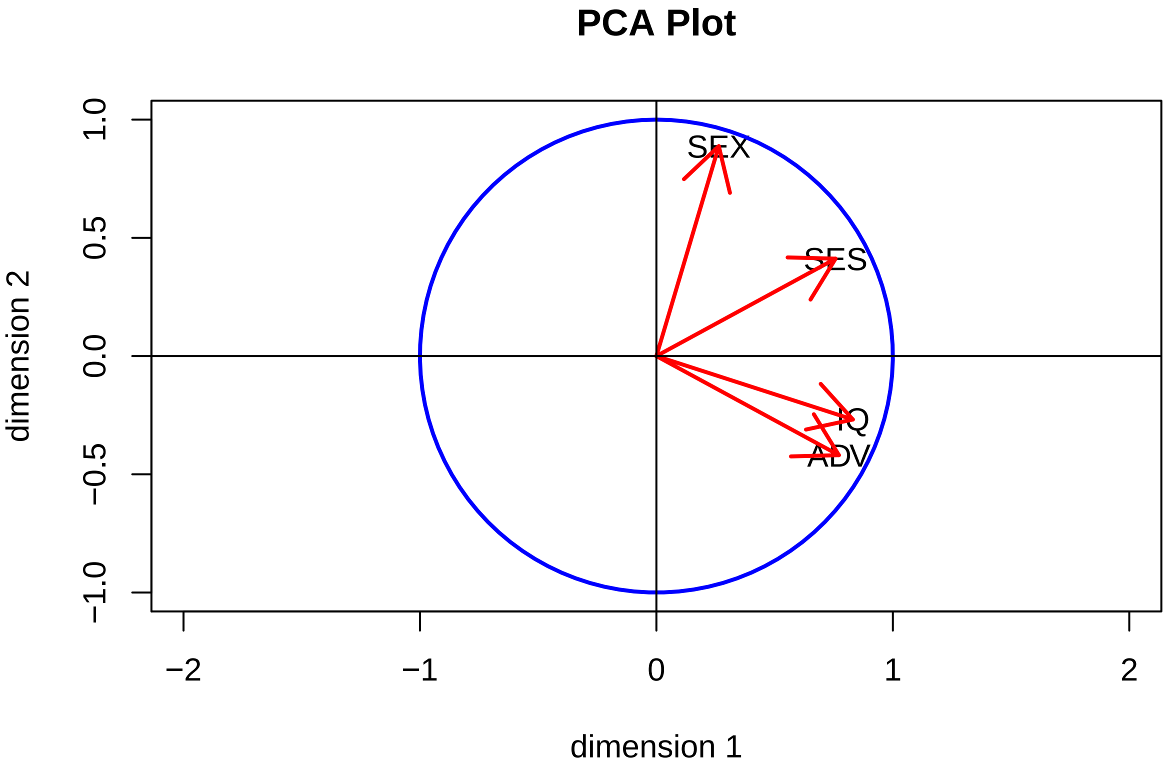

Figure 13: PRINCALS PCA Plot

In the same way as for PRIMALS we can make the PRINCALS category quantifications and object scores.

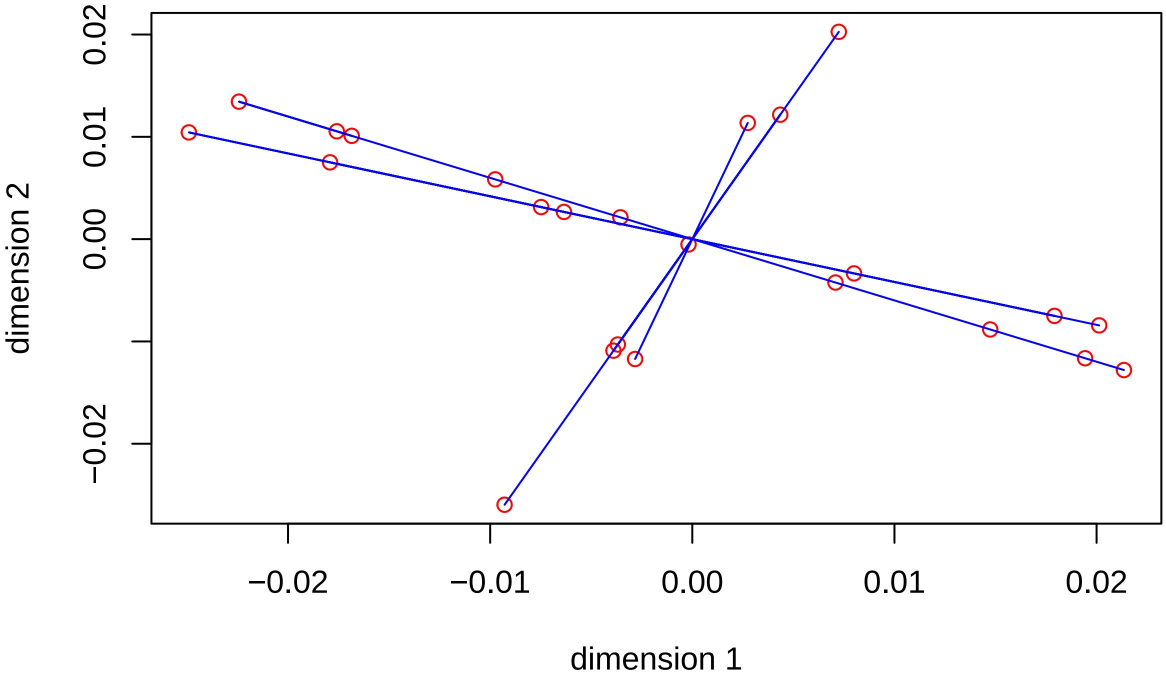

Figure 14: PRINCALS GALO Category Quantifications

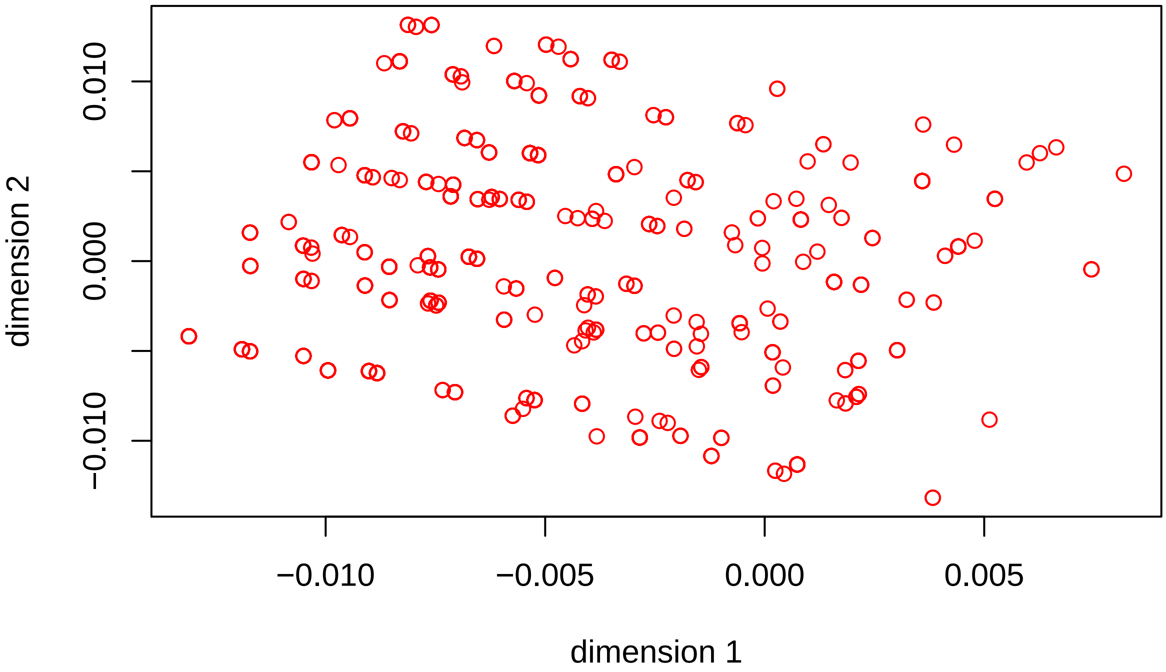

Figure 15: PRINCALS GALO Object Scores

3.4 Simultaneous Bilinearizing MCA

A more complete and more revealing analysis is possible by using the theory in De Leeuw (1982), see also Bekker and De Leeuw (1988) and De Leeuw (2015). We study an alternative way of diagonalizing the matrix $E = D^{-\frac{1}{2}}CD^{-\frac{1}{2}}$.

Suppose we can find square orthonormal K_j such that $K_j'K_j = I$ and $K_jE_{j\ell}K_\ell$ is diagonal for all $j, \ell = 1, \cdots, m$, say equal to the diagonal matrices $\Gamma_{j\ell}$. The condition is that if two submatrices of E have a subscript in common, then then also have the corresponding set of singular vectors in common. De Leeuw (1988) calls this *simultaneous bilinearizability*. Since that name is a bit verbose, we call the condition *SBL*. We know SBL is possible for the continuous multinormal, with the K_j the Hermite-Chebyshev polynomials. It is also possible if all variables are binary, and it is possible if $m = 2$. Those are some important gauges, and we expect SBL to be true approximately in many practical situations.

Define the direct sum $K := K_1 \oplus \cdots \oplus K_m$, and $\Gamma := K'EK$. Under SBL all $\Gamma_{j\ell}$ are diagonal. We can then find a permutation P of rows and columns such that $R := P'\Gamma P$ is the direct sum of a number of correlation matrices. If we start with m variables with k categories each, then $R = R_1 \oplus \cdots \oplus R_k$, where each R_s is of order m. The eigenvectors of R are the direct sum $L := L_1 \oplus \cdots \oplus L_k$, and under SBL we have

$$L'RL = L'P'\Gamma PL = L'P'K'EKPL = m\Lambda.$$

Thus under SBL the eigenvectors of E are given by $Z = KPL$ and the eigenvalues of E are also the eigenvalues of Γ and also the eigenvalues of R. Thus for each eigenvalue of E we can say which is the R_s it belongs to.

If the variables have a different number of categories then under SBL we have k matrices R_s, where k is now the maximum of the k_j. The R_s are generally of different orders, because variable j is represented in R_s if and only if $k_j \geq s$. Thus a binary variable only occurs in R_1 and R_2. In all cases the matrix R_1 can be chosen to be the $m \times m$ matrix with all elements equal to one. For all variables binary the matrix R_2 is the matrix of phi-coefficients (point correlations).

Now SBL will not be precisely be satisfied for empirical data, except in the trivial cases $k_j \equiv 2$ or $m = 2$. To create a suitable dat analysis techniue we choose the K_j in such a way that the sum of squares of the off-diagonal elements of the $\Gamma_{j\ell}$ is minimized, or, equivalently, the sum of squares of the diagonal elements is maximized. The details and the code are in De Leeuw (2008). For this paper the approximate diagonalization procedure is in the function `threeStepApprox()` which uses `kplSVD()` to find the K_j and `kplPerm()` to find the permutation P.

Because the SBL theory is somewhat more complicated than other MCA theory, we analyze a simple normal example in detail.

```
set.seed (12345)
x <- discreteNormal(1000, 4, matrix (.75, 4, 4) + .25 * diag (4), repList (c(-2,2), 4))
b <- burtTable (x)
d <- 1 / sqrt (diag (b$c))
e <- b$c * outer (d, d)
h <- threeStepApprox (e, rep(3, 4))
```

The Burt Table is

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
##  [1,]   25    0    0    9   16    0    9   16    0     9    16     0
##  [2,]    0  943    0   18  913   12   14  920    9    15   915    13
##  [3,]    0    0   32    0   21   11    0   20   12     0    27     5
##  [4,]    9   18    0   27    0    0    8   19    0    12    15     0
##  [5,]   16  913   21    0  950    0   15  920   15    12   929     9
##  [6,]    0   12   11    0    0   23    0   17    6     0    14     9
##  [7,]    9   14    0    8   15    0   23    0    0     9    14     0
##  [8,]   16  920   20   19  920   17    0  956    0    15   927    14
##  [9,]    0    9   12    0   15    6    0    0   21     0    17     4
## [10,]    9   15    0   12   12    0    9   15    0    24     0     0
## [11,]   16  915   27   15  929   14   14  927   17     0   958     0
## [12,]    0   13    5    0    9    9    0   14    4     0     0    18
```

The normalized Burt Table E is

```
##   1.0000  0.0000  0.0000  0.3464  0.1038  0.0000  0.3753  0.1035  0.0000  0.3674  0.1034  0.0000
##   0.0000  1.0000  0.0000  0.1128  0.9646  0.0815  0.0951  0.9690  0.0640  0.0997  0.9627  0.0998
##   0.0000  0.0000  1.0000  0.0000  0.1204  0.4055  0.0000  0.1143  0.4629  0.0000  0.1542  0.2083
##   0.3464  0.1128  0.0000  1.0000  0.0000  0.0000  0.3210  0.1183  0.0000  0.4714  0.0933  0.0000
##   0.1038  0.9646  0.1204  0.0000  1.0000  0.0000  0.1015  0.9654  0.1062  0.0795  0.9738  0.0688
##   0.0000  0.0815  0.4055  0.0000  0.0000  1.0000  0.0000  0.1146  0.2730  0.0000  0.0943  0.4423
##   0.3753  0.0951  0.0000  0.3210  0.1015  0.0000  1.0000  0.0000  0.0000  0.3831  0.0943  0.0000
##   0.1035  0.9690  0.1143  0.1183  0.9654  0.1146  0.0000  1.0000  0.0000  0.0990  0.9687  0.1067
##   0.0000  0.0640  0.4629  0.0000  0.1062  0.2730  0.0000  0.0000  1.0000  0.0000  0.1199  0.2057
##   0.3674  0.0997  0.0000  0.4714  0.0795  0.0000  0.3831  0.0990  0.0000  1.0000  0.0000  0.0000
##   0.1034  0.9627  0.1542  0.0933  0.9738  0.0943  0.0943  0.9687  0.1199  0.0000  1.0000  0.0000
##   0.0000  0.0998  0.2083  0.0000  0.0688  0.4423  0.0000  0.1067  0.2057  0.0000  0.0000  1.0000
```

and its eigenvalues are

```
##  4.0000  2.1028  1.9429  0.9545  0.7356  0.6493  0.6118  0.5338  0.4695 -0.0000 -0.0000 -0.0000
```

After application of `kplSVD()` to E we have

```
##  1.0000  0.0000  0.0000  1.0000  0.0000  0.0000  1.0000 -0.0000 -0.0000  1.0000  0.0000 -0.0000
##  0.0000  1.0000  0.0000 -0.0000  0.3313 -0.0209 -0.0000  0.3622 -0.0212 -0.0000  0.3529 -0.0049
##  0.0000  0.0000  1.0000  0.0000 -0.0226  0.3858  0.0000 -0.0218  0.4459  0.0000 -0.0080  0.1868
##  1.0000 -0.0000  0.0000  1.0000 -0.0000 -0.0000  1.0000 -0.0000 -0.0000  1.0000  0.0000 -0.0000
##  0.0000  0.3313 -0.0226 -0.0000  1.0000 -0.0000  0.0000  0.3054 -0.0141  0.0000  0.4590 -0.0102
##  0.0000 -0.0209  0.3858 -0.0000 -0.0000  1.0000  0.0000 -0.0136  0.2541  0.0000 -0.0117  0.4289
##  1.0000 -0.0000  0.0000  1.0000  0.0000  0.0000  1.0000  0.0000 -0.0000  1.0000  0.0000 -0.0000
## -0.0000  0.3622 -0.0218 -0.0000  0.3054 -0.0136  0.0000  1.0000  0.0000 -0.0000  0.3692 -0.0024
## -0.0000 -0.0212  0.4459 -0.0000 -0.0141  0.2541 -0.0000  0.0000  1.0000 -0.0000 -0.0042  0.1883
##  1.0000 -0.0000  0.0000  1.0000  0.0000  0.0000  1.0000 -0.0000 -0.0000  1.0000  0.0000 -0.0000
##  0.0000  0.3529 -0.0080  0.0000  0.4590 -0.0117  0.0000  0.3692 -0.0042 -0.0000  1.0000  0.0000
## -0.0000 -0.0049  0.1868 -0.0000 -0.0102  0.4289 -0.0000 -0.0024  0.1883 -0.0000 -0.0000  1.0000
```

Permuting rows and columns gives the matrix $R = R_1 \oplus R_2 \oplus R_3$

```
##  1.0000  1.0000  1.0000  1.0000  0.0000  0.0000 -0.0000  0.0000  0.0000  0.0000 -0.0000 -0.0000
##  1.0000  1.0000  1.0000  1.0000 -0.0000 -0.0000 -0.0000  0.0000  0.0000 -0.0000 -0.0000 -0.0000
##  1.0000  1.0000  1.0000  1.0000 -0.0000  0.0000  0.0000  0.0000  0.0000  0.0000 -0.0000 -0.0000
##  1.0000  1.0000  1.0000  1.0000 -0.0000  0.0000 -0.0000  0.0000  0.0000  0.0000 -0.0000 -0.0000
##  0.0000 -0.0000 -0.0000 -0.0000  1.0000  0.3313  0.3622  0.3529  0.0000 -0.0209 -0.0212 -0.0049
##  0.0000 -0.0000  0.0000  0.0000  0.3313  1.0000  0.3054  0.4590 -0.0226 -0.0000 -0.0141 -0.0102
## -0.0000 -0.0000  0.0000 -0.0000  0.3622  0.3054  1.0000  0.3692 -0.0218 -0.0136  0.0000 -0.0024
##  0.0000  0.0000  0.0000 -0.0000  0.3529  0.4590  0.3692  1.0000 -0.0080 -0.0117 -0.0042  0.0000
##  0.0000  0.0000  0.0000  0.0000  0.0000 -0.0226 -0.0218 -0.0080  1.0000  0.3858  0.4459  0.1868
##  0.0000 -0.0000  0.0000  0.0000 -0.0209 -0.0000 -0.0136 -0.0117  0.3858  1.0000  0.2541  0.4289
## -0.0000 -0.0000 -0.0000 -0.0000 -0.0212 -0.0141  0.0000 -0.0042  0.4459  0.2541  1.0000  0.1883
## -0.0000 -0.0000 -0.0000 -0.0000 -0.0049 -0.0102 -0.0024 -0.0000  0.1868  0.4289  0.1883  1.0000
```

And we can now diagonalize the blocks.

```
##  4.0000 -0.0000  0.0000 -0.0000 -0.0000 -0.0000  0.0000 -0.0000 -0.0000  0.0000  0.0000 -0.0000
## -0.0000 -0.0000  0.0000  0.0000 -0.0000 -0.0000  0.0000 -0.0000 -0.0000  0.0000 -0.0000 -0.0000
##  0.0000  0.0000  0.0000 -0.0000 -0.0000  0.0000 -0.0000 -0.0000 -0.0000  0.0000 -0.0000 -0.0000
## -0.0000 -0.0000  0.0000 -0.0000  0.0000  0.0000  0.0000 -0.0000 -0.0000  0.0000 -0.0000 -0.0000
## -0.0000 -0.0000 -0.0000  0.0000  2.0923 -0.0000  0.0000 -0.0000 -0.0395 -0.0077 -0.0066 -0.0015
## -0.0000 -0.0000  0.0000  0.0000 -0.0000  0.7349 -0.0000  0.0000 -0.0037  0.0047 -0.0058 -0.0110
##  0.0000  0.0000 -0.0000  0.0000  0.0000 -0.0000  0.6412 -0.0000  0.0045 -0.0007 -0.0144  0.0158
## -0.0000 -0.0000 -0.0000 -0.0000 -0.0000  0.0000 -0.0000  0.5315 -0.0083 -0.0058  0.0038  0.0126
## -0.0000 -0.0000 -0.0000 -0.0000 -0.0395 -0.0037  0.0045 -0.0083  1.9532 -0.0000 -0.0000 -0.0000
##  0.0000  0.0000  0.0000  0.0000 -0.0077  0.0047 -0.0007 -0.0058 -0.0000  0.9543  0.0000  0.0000
##  0.0000 -0.0000 -0.0000 -0.0000 -0.0066 -0.0058 -0.0144  0.0038 -0.0000  0.0000  0.6185  0.0000
## -0.0000 -0.0000 -0.0000 -0.0000 -0.0015 -0.0110  0.0158  0.0126 -0.0000  0.0000  0.0000  0.4740
```

and we can sort the diagonal elements

```
##  4.0000  2.0923  1.9532  0.9543  0.7349  0.6412  0.6185  0.5315  0.4740  0.0000 -0.0000 -0.0000
```

to find they are really close to the eigenvalues of E. The correlations between the eigenvector of E and the SBL approximations are

```
## -1.0000  0.0000 -0.0000  0.0000  0.0000 -0.0000  0.0000  0.0000  0.0000  0.0000  0.0000 -0.0000
## -0.0000  0.0000  0.0000 -0.0000 -0.9669 -0.0007  0.0007 -0.0014  0.2551  0.0065  0.0043  0.0009
## -0.0000  0.0000  0.0000 -0.0000 -0.2551  0.0029 -0.0033  0.0057 -0.9669  0.0020  0.0013  0.0003
##  0.0000 -0.0000 -0.0000 -0.0000  0.0068  0.0215 -0.0023 -0.0138  0.0002  0.9996 -0.0006 -0.0009
## -0.0000 -0.0000 -0.0000 -0.0000 -0.0003  0.9976  0.0009 -0.0030  0.0030 -0.0216 -0.0500 -0.0420
## -0.0000 -0.0000 -0.0000 -0.0000  0.0019  0.0182 -0.9045  0.0044  0.0032 -0.0022  0.4180 -0.0826
##  0.0000  0.0000  0.0000  0.0000  0.0041  0.0472  0.4154  0.0503 -0.0008  0.0011  0.9057  0.0486
##  0.0000 -0.0000 -0.0000  0.0000  0.0002  0.0091 -0.0356  0.9786  0.0059  0.0134 -0.0490  0.1957
##  0.0000 -0.0000 -0.0000 -0.0000 -0.0009 -0.0404  0.0901  0.1987  0.0007  0.0029  0.0020 -0.9751
## -0.0000 -0.4411 -0.4836  0.7560  0.0000 -0.0000  0.0000 -0.0000 -0.0000 -0.0000 -0.0000  0.0000
##  0.0000  0.0000  0.8424  0.5389  0.0000 -0.0000 -0.0000  0.0000 -0.0000  0.0000 -0.0000 -0.0000
## -0.0000 -0.8975  0.2377 -0.3716  0.0000 -0.0000  0.0000  0.0000 -0.0000  0.0000 -0.0000  0.0000
```

Thus the eight non-trivial eigenvectors of E correspond, repectively, with first of R_2, first of R_3, second of R_3, second of R_2, third of R_2, third of R_3, fourth of R_2, fourth of R_3. In terms of the polynomials underlying the category quantifications this means linear, quadratic, quadratic, linear, linear, quadratic, linear, quadratic.

Back to GALO. For GALO the eigenvalues of $E/4$ are

```
##      1.0000      0.5392      0.3915      0.3833      0.3465      0.3083      0.2764      0.2702      0.2569
```

The direct sum of the eigenvalues of the R_s/m, which have orders 4, 4, 3, 3, 3, 3, 2, 1, 1, is

```
##      1.0000      0.0000      0.0000     -0.0000
##      0.5376      0.2471      0.1636      0.0517
##      0.3908      0.2425      0.1167
##      0.3505      0.2442      0.1553
##      0.3238      0.2527      0.1735
##      0.2688      0.2606      0.2206
##      0.2716      0.2284
##      0.2500
##      0.2500
```

and if we sort these we find

```
##      1.0000      0.5376      0.3908      0.3505      0.3238      0.2716      0.2688      0.2606      0.2527
```

Note that the orders of the matrices R_s in this case are 4, 4, 3, 3, 3, 3, 2, 1, 1.

Clearly the approximate eigenvalues from the KPL diagonalization and the actual eigenvalues are very similar, especially the larger ones and the smaller ones. The correlation between actual eigenvectors and approximate eigenvectors is $Q := Z'KPL$. Let us look at the first three non-trivial eigenvectors of E, which correspond with rows 2, 3, and 4 of Q. Regress them on the non-trivial columns of KPL, which are all columns except the first four. The percentage of variance explained by the colums of KPL are the squares of the elements of rows 2, 3, and 4 of Q.

```
## 0.99 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

## 0.00 0.00 0.00 0.00 0.98 0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## 0.00 0.31 0.01 0.00 0.01 0.01 0.00 0.20 0.07 0.00 0.26 0.02 0.00 0.01 0.08 0.00 0.00 0.00 0.00 0.01
```

Thus we see the first non-trivial eigenvector of E corresponds with the first eigenvector of R_2, which the next eigenvector of E corresponds with the first eigenvector of R_3. The third non-trivial eigenvector of E is most closely related to the second eigenvector of R_2, but there are also contributions of the first eigenvectors of R_4 and R_5.

There is a somewhat more rigorous method to match the eigenvectors in Z and KPL. The Hadamard product $Q * Q$ is doubly stochastic. Finding the permutation matrix closest in the least suares sense to $Q * Q$ is a linear assignment problem that can be solved by using `lp.assign()` from the `lpSolve` package (Berkelaar and others 2015). We indicate for each row the column where the permutation matrix has a one.

```
##  [1]  1  5  9  6 12 15 21 18 23 24 13 16 10 22 20  7 17 14 19 11  8  2  3
## [24]  4
```

Again this indicates that the first three non-trivial eigenvectors of E correspond with the dominant eigenvalue of R_2, the dominant eigenvalue of R_3, and the second eigenvalue of R_2. If the eigenvectors in K correspond with the orthogonal polynomials, as they do in the multinormal case, then the first two nontrivial eigenvectors produce a croissant, because they come from the linear and quadratic transformations, respectively. But if we select the first and the third non-trivial eigenvector of E to plot, then they both come from the linear quantifications and they do not produce a croissant.

We first plot $Y = D^{-\frac{1}{2}}KPL$, using the columns corresponding to the first two non-trivial eigenvectors of E. This shows the croissant.

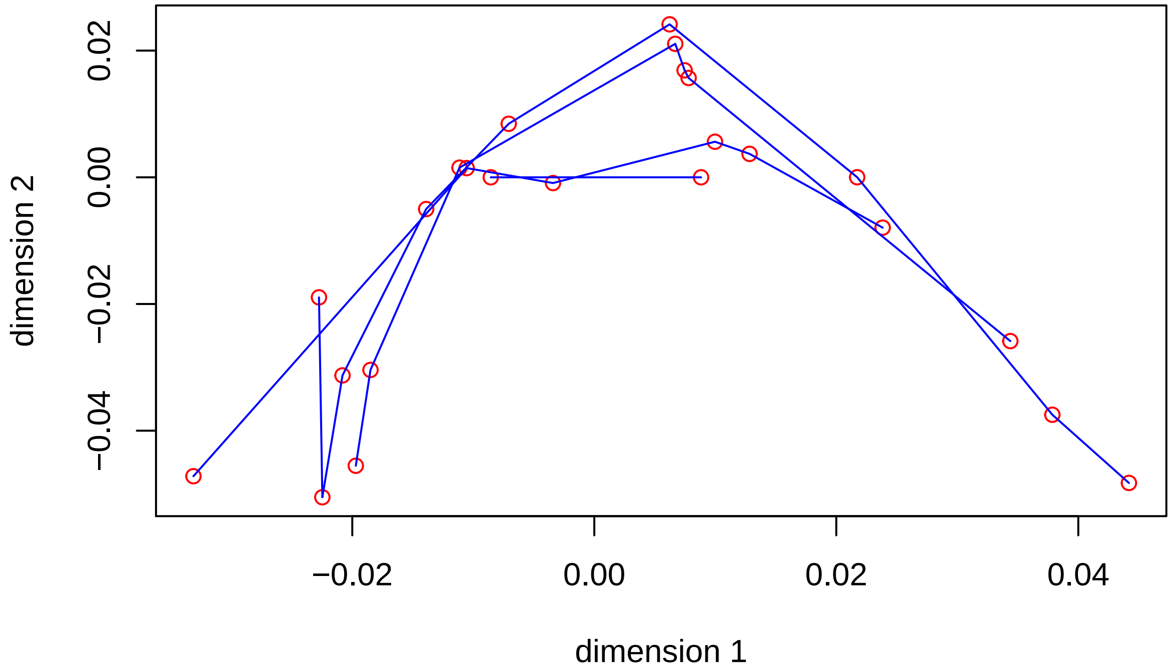

Figure 16: SBL GALO Category Quantifications, Different R

If we choose the first and second eigenvector of R_2, i.e. the solution corresponding to the first and third non-trivial eigenvectors of E, there is no croissant. In fact, the solution is similar to the one given by PRIMALS, consisting of a line through the origin for each variable, with the category quantifications on the line.

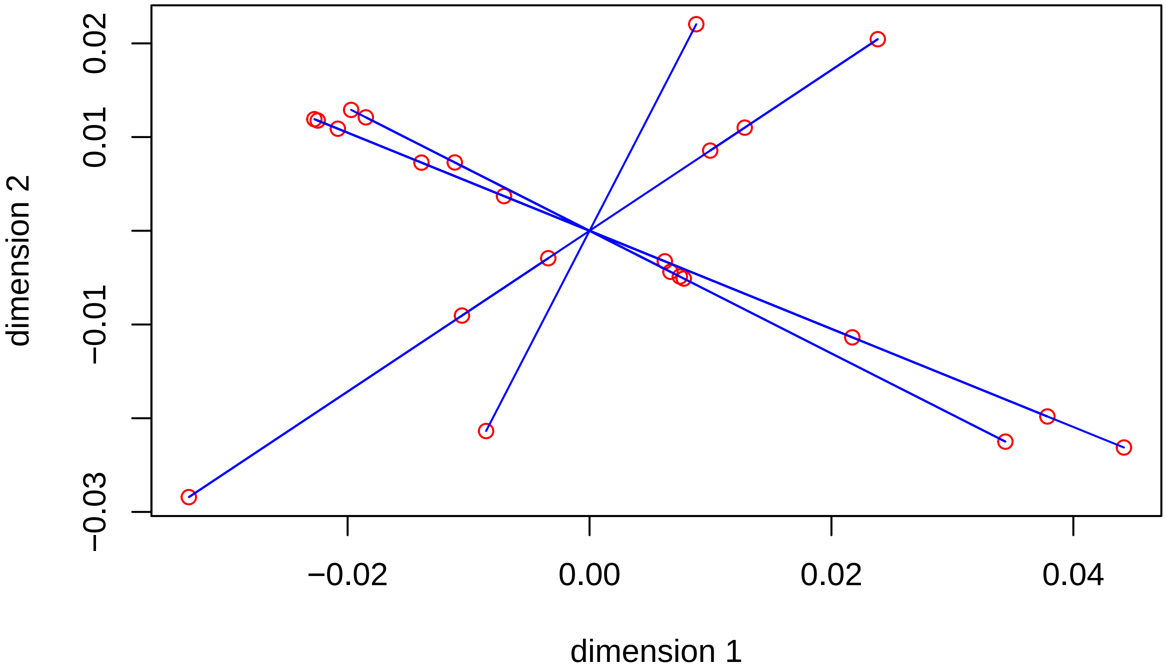

Figure 17: SBL GALO Category Quantifications, Same R

We can compare this to the solution we get by actaully plotting the first and third non-trivial eigenvector of E. There is no croissant in this case either.

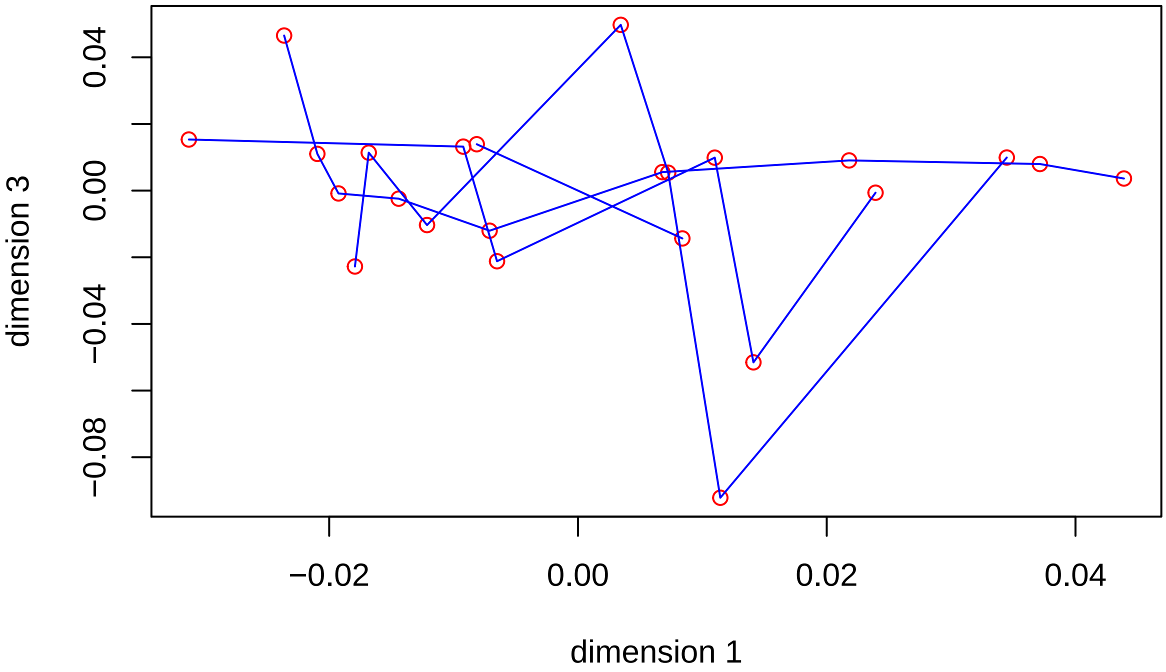

Figure 18: GALO Category Quantifications, Skip One Dimension

4 Real Data: Norway

The data is register data from Norway, with information on a high number (n = 159539) of individuals. The analysis is part of a social mobility study conducted by Arild Danielsen, Department of History, Sociology and Innovation, Faculty of Economics and Social Sciences, University College of Southeast Norway. See Danielsen (2015). The data have four variables with data on class-fractions (based on occupational categories) of the individual's two parents (in 2008) and two grand-fathers (in 1980).

Thus we have information for each of the 159539 individuals about the membership of Mother, Father, Father's Father, and Mother's Father in the following categories.

1. Cultural elite
2. Professional elite
3. Economic elite
4. Cultural upper-middle
5. Professional upper-middle
6. Economic upper-middle
7. Cultural lower-middle
8. Professional lower-middle
9. Economic lower-middle
10. Skilled workers

11. Unskilled and semi-skilled workers
12. Benefits (family / unempl. / social)
13. Not possible to classify
14. Missing observations

In order to simplify our analysis we decided to eliminate all individuals with scores in categories 12-14. This reduces the number of observations to 81259. For the actual sociological analysis this would probably be a wasteful and unwise decision, but for our croissant illustration it makes sense. The marginals of the remaining eleven categories are as follows.

```
##      [,1] [,2] [,3] [,4]  [,5]  [,6] [,7]  [,8] [,9] [,10] [,11]
## [1,]  716 3290 1117 2081  4902  6618 1308  3229 4799 18730 34469
## [2,]  650 3451 1072 1996  4987  6873 1384  3265 5030 18868 33683
## [3,] 1373 3839 3710 4198 11458 11078  860  4534 8630 17506 14073
## [4,]  639 1069  284 6466 10570  2587 3090 14977 7225 17579 16773
```

4.1 MCA

The first two eigenvalues of an MCA on these data are 0.4486663 and 0.3095208. We plot the category quantifications of the four variables. Calling the shape of the plot a croissant is a stretch, it is better to call it a *wedge*.

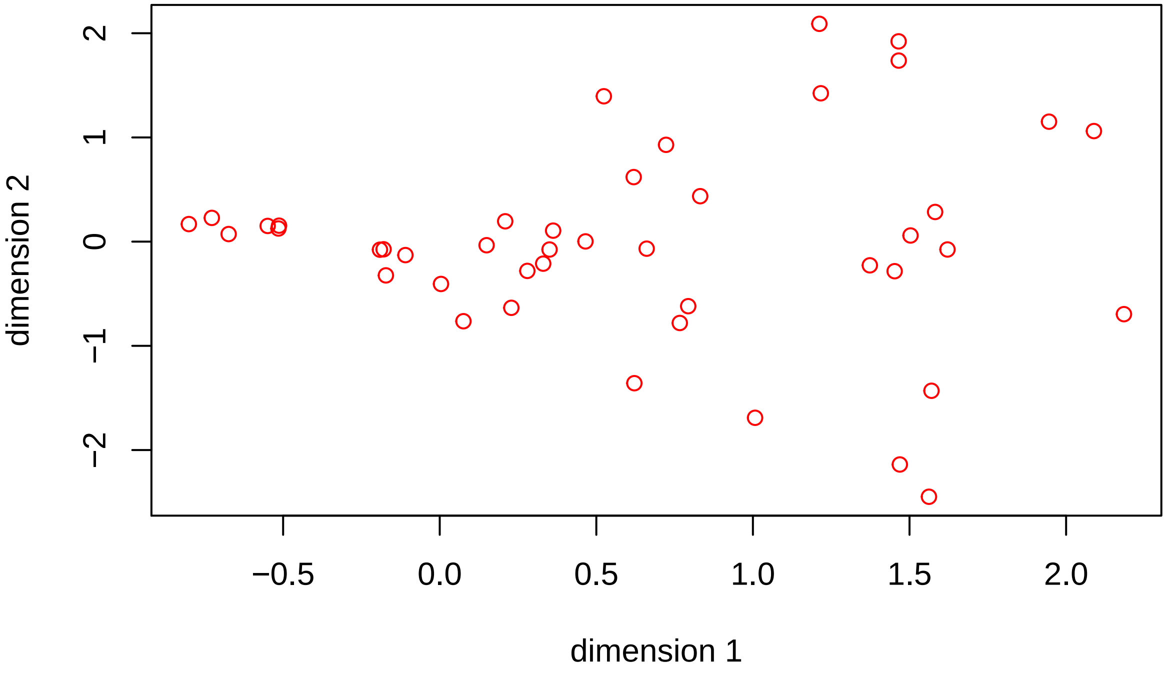

Figure 19: Norway Category Quantifications, Joint

The first nine categories have factorial structure, combining the three possibilities Cultural, Professional, Economic with the three Elite, Upper Middle, and Lower Middle. We use this factorial structure to connect the nine categories by lines. The line “Cultural”" connects points 1, 4, 7, the line “Elite” connects points 1, 2, 3, and so on. In total there are six lines. Categories 10 and 11, skilled and unskilled labour, are not connected. We’ll leave the interpretation to people who know what they are talking about.

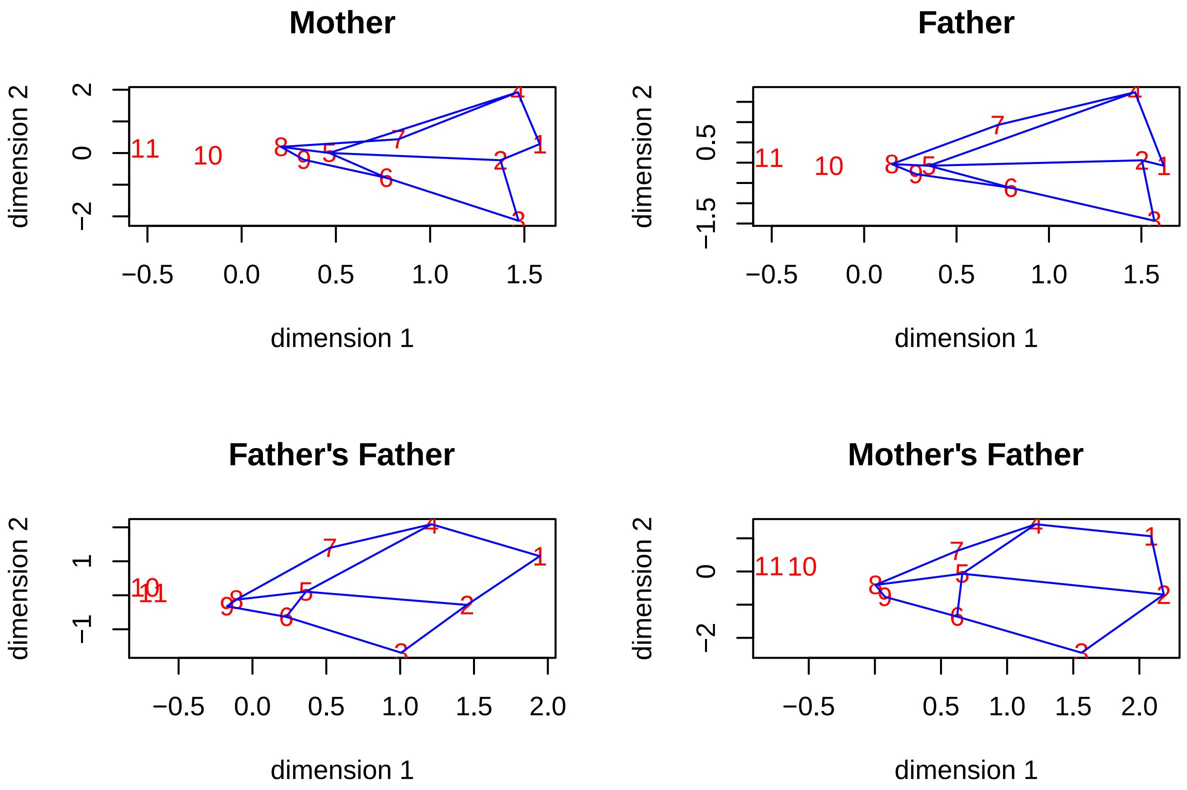

Figure 20: Norway Category Quantifications, Separate

We can replot the separate category quantifications, labeled by the category frequencies. Obviously the sharp point of the wedge corresponds with the categories with the highest frequencies, illustrating the general point that in MCA heavily populated categories tend to be close to the origin of the plot.

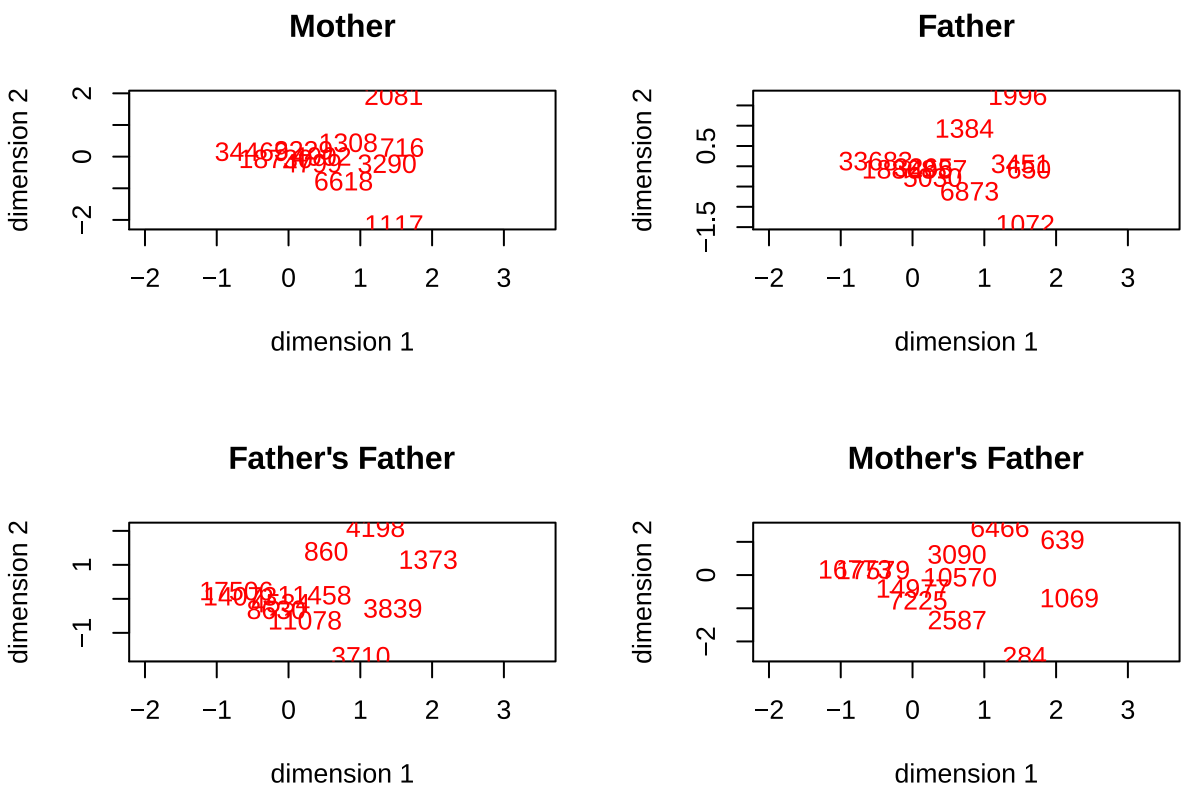

Figure 21: Norway Category Quantifications, Frequencies

The 81259 object scores form a very sharp and well-filled wedge.

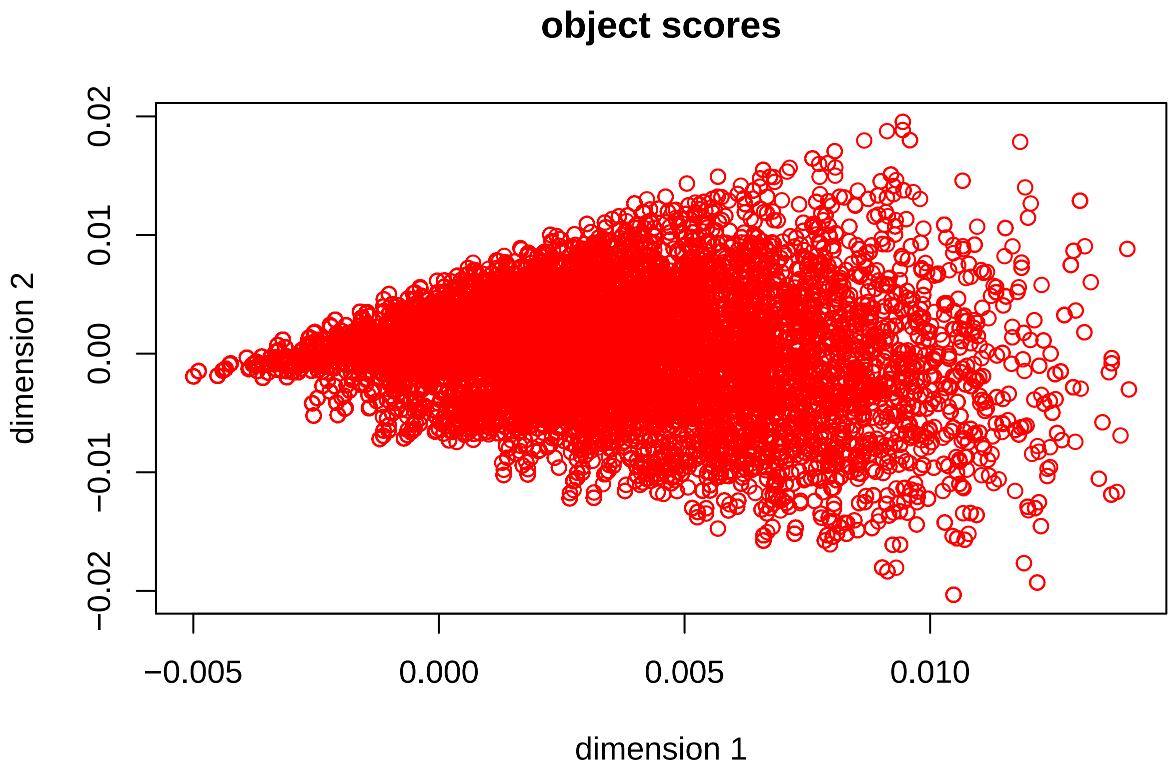

Figure 22: Norway Object Scores

4.2 Additive Constraints

In figure Figure 20 we have used the factorial 3 × 3 structure of the nine first categories to draw a grid in the plot. Instead, we can require that the nine points for each variable are on a regular grid, by using a design matrix that forces the "Cultural Elite" category point to be the sum of a "Cultural" point and an "Elite" point. And so on for all nine combinations. The grids for the four variables are generally different. The MCA under these constraints has first two eigenvalues 0.4470581 and 0.3038511, which is only marginally smaller than the unconstrained eigenvalues 0.4486663 and 0.3095208. The grid lines clearly display the structure within the wedge.

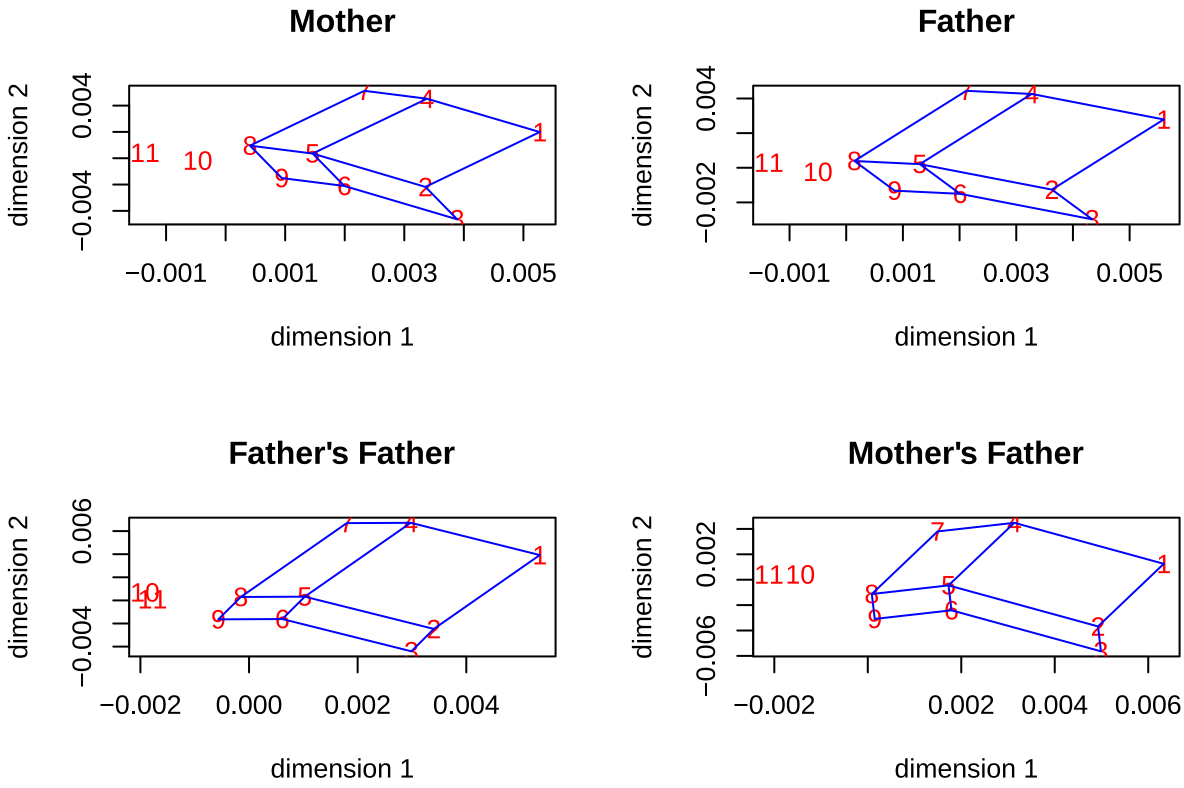

Figure 23: Norway Category Quantifications, Additive

It is somewhat surprising that the object scores show a pattern which is quite different from the unconstrained scores in Figure 22. The wedge is still there, but it is clearly partitioned into five or six parallel clouds of individuals.

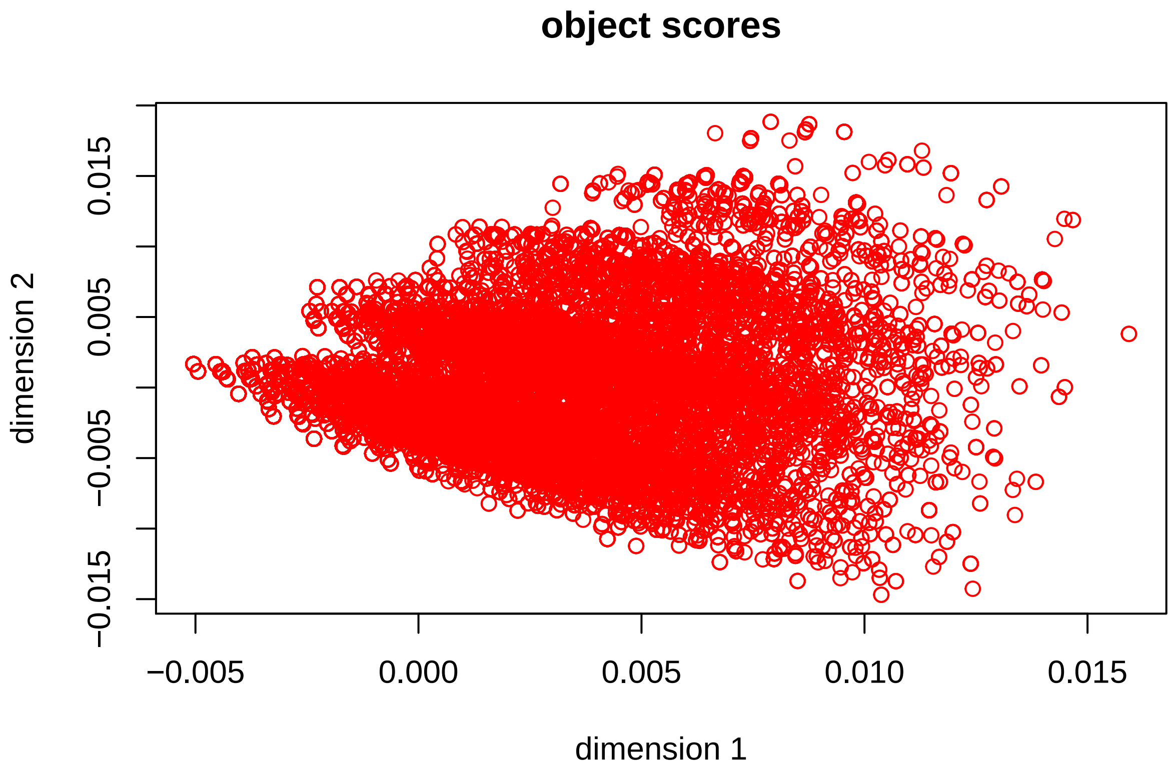

Figure 24: Norway Object Scores, Additive

4.3 Additive and Equality Constraints

We can go a step further with constraining the solution by requiring additivity, as before, and in addition that all four grids are the same. The MCA under these constraints has first two eigenvalues 0.4227509 and 0.2935249, which are somewhat smaller than the previous eigenvalues.

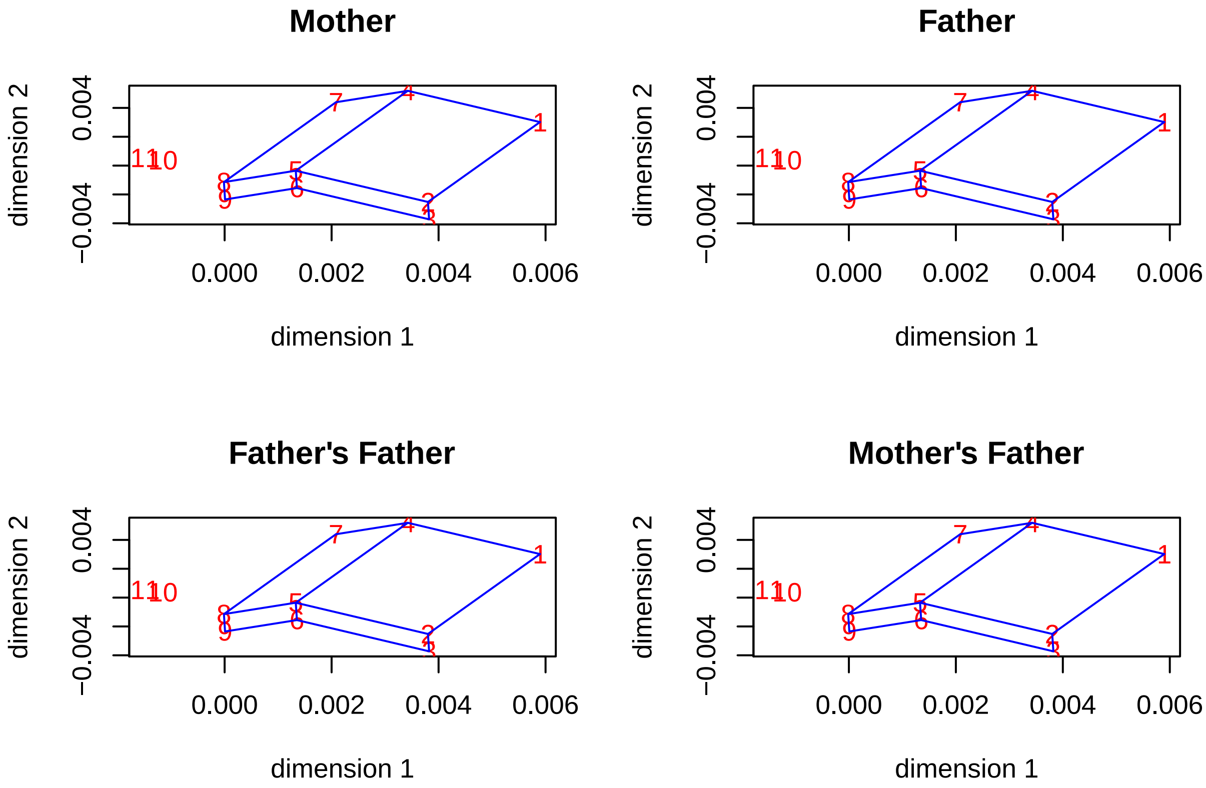

Figure 26: Norway Object Scores, Additive and Equality

The object scores show the same clouds within the wedge, but because many more individuals get the same scores there are fewer points and the clouds are far less dense.

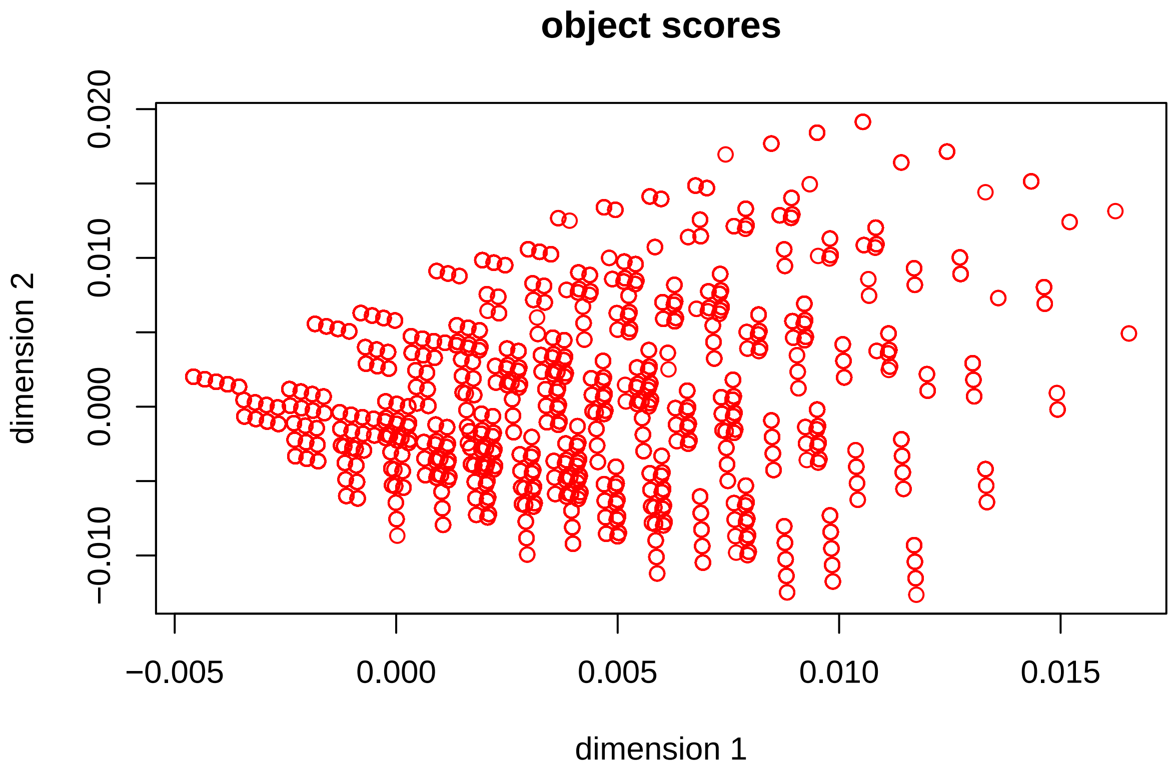

Figure 26: Norway Object Scores, Additive and Equality

4.4 Simultaneous Bilinearizing MCA

Using additivity does not really get rid of the wedge, but allows us to study it in more detail and take it apart. SBL theory gives us another tool to look at the wedge.

For Norway the eigenvalues of $E/4$ are

```
##     1.0000     0.4487     0.3095     0.2912     0.2783     0.2745     0.2620     0.2607     0.2594
```

The direct sum of the eigenvalues of the R_s/m is

```
##     1.0000     0.0000    -0.0000     0.0000
##     0.4486     0.2054     0.1869     0.1591
##     0.3081     0.2430     0.2407     0.2082
##     0.2909     0.2470     0.2413     0.2208
##     0.2742     0.2479     0.2452     0.2328
##     0.2766     0.2497     0.2438     0.2299
##     0.2597     0.2530     0.2482     0.2391
##     0.2577     0.2522     0.2471     0.2430
##     0.2576     0.2500     0.2466     0.2457
##     0.2577     0.2510     0.2488     0.2425
##     0.2551     0.2534     0.2497     0.2418
```

Note that each row here adds up to one. Also note that the orders of the matrices R_s in this case are 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, because all variables have 11 categories. If we sort the eigenvalues of all R_s/m we find

```
##     1.0000     0.4486     0.3081     0.2909     0.2766     0.2742     0.2597     0.2577     0.2577
```

The two sets of sorted eigenvalues are plotted in Figure 27. It is pretty obvious they are close.

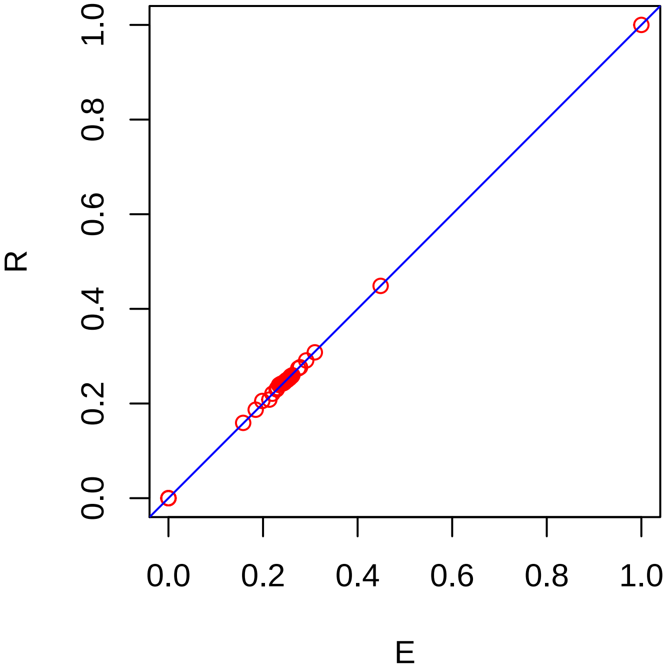

Figure 27: Eigenvalues of E and R for Norway Example

The assignment problem for the eigenvectors gives the solution

```
##  [1]  1  5  9 13 21 17 25 33 29 26 37 42 30 22 41 38 43 31 27 34 19 39 18
## [24] 35 36 14 40 32 10 23 15 28 11 44 20 24 16 12  6  7  8  2  3  4
```

Again we see the familiar pattern: eigenvalues of E are the first eigenvalue of (trivial) R_1, followed by the first eigenvalue of R_2, of R_3, and of R_4. The remaining eigenvalue of R_2 are actually the smallest ones of the whole sequence, except for the trivial zeroes. This tells us that these data are highly one-dimensional. We see the familiar wedge again in Figure 28, which plots the dominant eigenvectors corresponding to R_2 and R_3.

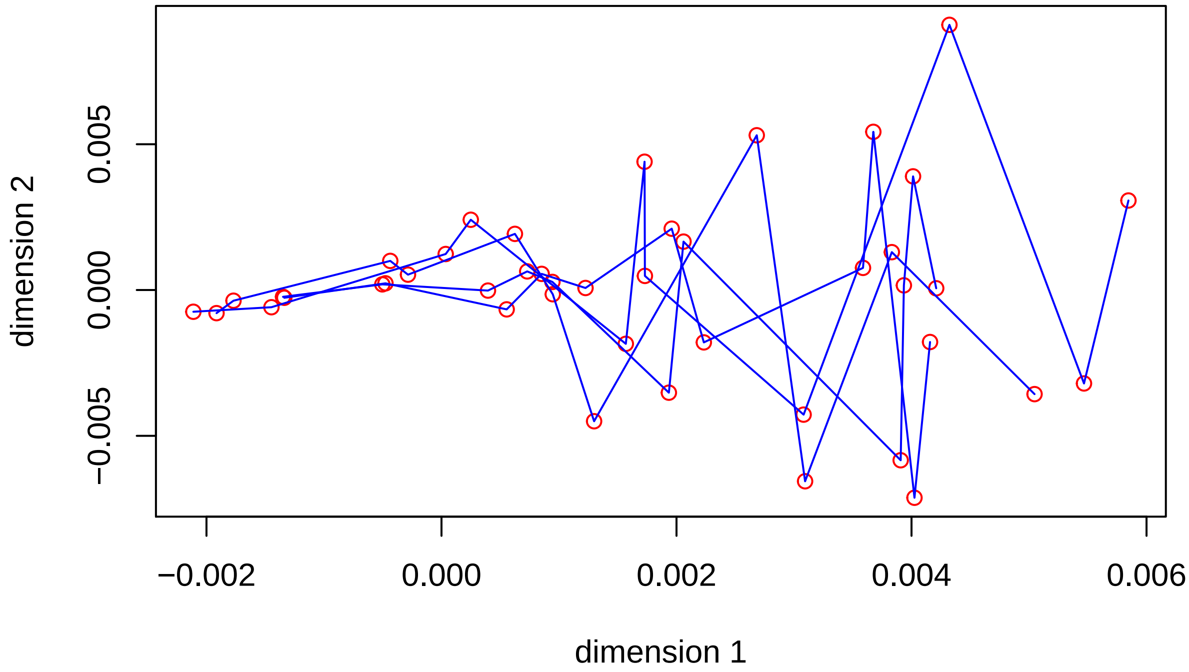

Figure 28: SBL Norway Category Quantifications, Different R

5 Appendix: Code

```
discreteNormal <- function(n, m, r, knots) {
    x <- matrix(rnorm(n * m), n, m)
    x <- x %*% chol(r)
    for (j in 1:m) {
        x[, j] <- apply(outer(x[, j], c(knots[[j]], Inf), ">"), 1, which.min)
    }
    return(x)
}

repList <- function(x, m) {
    z <- vector("list", m)
    return(lapply(z, function(i) x))
}

burtTable <- function(x, degrees = rep(-1, ncol(x)), knots = NULL, center = FALSE,
    standardize = FALSE, orthonormalize = FALSE) {
    n <- nrow(x)
    m <- ncol(x)
    g <- matrix(0, n, 0)
    l <- rep(0, m)
    for (j in 1:m) {
        z <- x[, j]
        if (degrees[j] < 0) {
            h <- ifelse(outer(z, sort(unique(z)), "=="), 1, 0)
        } else {
            h <- bsplineBasis(z, degrees[j], knots[[j]])
        }
        if (center) {
            h <- center(h)[, -1]
        }
        if (standardize) {
            h <- standardize(h)
        }
        if (orthonormalize) {
            h <- gs(h)$q
        }
        g <- cbind(g, h)
        l[j] <- ncol(h)
    }
    return(list(c = crossprod(g), g = g, ord = l))
}

directSum <- function(x) {
    m <- length(x)
    nr <- sum(sapply(x, nrow))
    nc <- sum(sapply(x, ncol))
    z <- matrix(0, nr, nc)
    kr <- 0
    kc <- 0
    for (i in 1:m) {
        ir <- nrow(x[[i]])
        ic <- ncol(x[[i]])
        z[kr + (1:ir), kc + (1:ic)] <- x[[i]]
        kr <- kr + ir
        kc <- kc + ic
    }
    return(z)
}

kplPerm <- function(cc, k) {
    kl <- unlist(sapply(k, function(i) 1:i))
    p <- ifelse(outer(1:sum(k), order(kl), "=="), 1, 0)
    return(list(pcp = t(p) %*% cc %*% p, perm = p, ord = as.vector(table(kl))))
}

makeE <- function(cc, k) {
    dd <- mInvSqrt(blockSelect(cc, k))
    return(dd %*% cc %*% dd)
}

mInvSqrt <- function(x) {
    ex <- eigen(x)
    ew <- abs(ex$values)
    ev <- ifelse(ew == 0, 0, 1/sqrt(ew))
    ey <- ex$vectors
    return(ey %*% (ev * t(ey)))
}

blockSelect <- function(cc, k) {
    l <- unlist(lapply(1:length(k), function(i) rep(i, k[i])))
    return(cc * ifelse(outer(l, l, "=="), 1, 0))
}

kplSVD <- function(e, k, eps = 1e-06, itmax = 500, verbose = TRUE, vectors = TRUE) {
    m <- length(k)
    sk <- sum(k)
    ll <- kk <- ww <- diag(sk)
    itel <- 1
    ossq <- 0
    klw <- 1 + cumsum(c(0, k))[1:m]
    kup <- cumsum(k)
    ind <- lapply(1:m, function(i) klw[i]:kup[i])
    for (i in 1:m) kk[ind[[i]], ind[[i]]] <- t(svd(e[ind[[i]], ])$u)
    kek <- kk %*% e %*% t(kk)
    for (i in 1:m) for (j in 1:m) ww[ind[[i]], ind[[j]]] <- ifelse(outer(1:k[i],
        1:k[j], "=="), 1, 0)
    repeat {
        for (l in 1:m) {
            if (k[l] == 2)
                (next)()
            li <- ind[[l]]
            for (i in (klw[l] + 1):(kup[l] - 1)) for (j in (i + 1):kup[l]) {
                bi <- kek[i, -li]
                bj <- kek[j, -li]
                wi <- ww[i, -li]
                wj <- ww[j, -li]
                acc <- sum(wi * bi^2) + sum(wj * bj^2)
                acs <- sum((wi - wj) * bi * bj)
                ass <- sum(wi * bj^2) + sum(wj * bi^2)
                u <- eigen(matrix(c(acc, acs, acs, ass), 2, 2))$vectors[, 1]
                c <- u[1]
                s <- u[2]
                kek[-li, i] <- kek[i, -li] <- c * bi + s * bj
                kek[-li, j] <- kek[j, -li] <- c * bj - s * bi
                if (vectors) {
                  ki <- kk[i, li]
                  kj <- kk[j, li]
                  kk[i, li] <- c * ki + s * kj
                  kk[j, li] <- c * kj - s * ki
                }
            }
        }
        nssq <- sum(ww * kek^2) - sum(diag(kek)^2)
        if (verbose)
            cat("Iteration ", formatC(itel, digits = 4), "ssq ", formatC(nssq,
                digits = 10, width = 15), "\n")
        if (((nssq - ossq) < eps) || (itel == itmax))
            (break)()
        itel <- itel + 1
        ossq <- nssq
    }
    return(list(kek = kek, kk = kk, itel = itel, ssq = nssq))
}

inducedR <- function(c, y, k) {
    m <- length(k)
    l <- unlist(lapply(1:m, function(i) rep(i, k[i])))
    g <- ifelse(outer(l, 1:m, "=="), 1, 0)
    s <- g * matrix(y, length(y), m)
    r <- crossprod(s, c %*% s)
    e <- abs(diag(r))
    d <- ifelse(e == 0, 0, e)
    return(r/sqrt(outer(d, d)))
}

threeStepApprox <- function(e, k, eps = 1e-06, itmax = 500, verbose = FALSE,
    vectors = TRUE) {
    ee <- eigen(e)
    ev <- ee$vectors
    ea <- ee$values
    f <- kplSVD(e, k, eps = eps, itmax = itmax, verbose = verbose, vectors = TRUE)
    ef <- f$kek
    kf <- f$kk
    g <- kplPerm(ef, k)
    eg <- g$pcp
    kg <- crossprod(g$perm, kf)
    gg <- g$ord
    gl <- cumsum(gg)
    lg <- 1 + c(0, gl)[-length(gl) - 1]
    bb <- lapply(1:length(gl), function(i) eigen(eg[lg[i]:gl[i], lg[i]:gl[i]])$vectors)
    vc <- directSum(bb)
    eh <- crossprod(vc, eg %*% vc)
    kh <- crossprod(kg, vc)
    cc <- crossprod(ev, kh)
    mc <- apply(cc, 1, function(x) max(abs(x)))
    wc <- apply(cc, 1, function(x) which.max(abs(x)))
    yc <- apply(cc, 2, function(x) max(abs(x)))
    vc <- apply(cc, 2, function(x) which.max(abs(x)))
    return(list(eval = ea, evec = ev, aval = diag(eh), avec = kh, bval = ef,
        cval = eg, dval = eh, cc = cc, mc = mc, wc = wc, yc = yc, vc = vc, gg = gg))
}
```

6 Appendix: NEWS

0.01 12/04/15

- First working version posted

0.02 12/06/15

- more runs of examples

0.03 12/07/15

- normal examples

1.00 12/08/15

- added text
- added constrained examples
- first published version

1.01 12/09/15

- GALO text
- GALO single
- GALO PCA

1.02 12/10/15

- subsections
- removed Norway data from repository
- short MCA introduction
- added PRIMALS section to GALO
- added PRINCALS section to GALO

1.03 12/12/15

- code for three-step MCA
- GALO analysis by three-step MCA

1.04 12/13/15

- GALO analysis by three-step MCA
- intro theory for simultaneous bilinearity

2.00 12/14/15

- GALO analysis with SBL completed
- Norway analysis with SBL
- That's it for now

2.01 12/16/15

- small detailed SBL example

References

Bekker, P., and J. De Leeuw. 1988. "Relation Between Variants of Nonlinear Principal Component Analysis." In *Component and Correspondence Analysis*, edited by J.L.A. Van Rijckevorsel and J. De Leeuw, 1–31. Wiley Series in Probability and Mathematcal Statistics. Chichester, England: Wiley. http://www.stat.ucla.edu/~deleeuw/janspubs/1988/chapters/bekker_deleeuw_C_88.pdf.

Berkelaar, M., and others. 2015. *lpSolve: Interface to 'Lp$_s$olve' v. 5.5 to Solve Linear/Integer Programs.* https://CRAN.R-project.org/package=lpSolve.

Danielsen, A. 2015. "An Accumulation Approach to Cultural Capital." Paper presented at the conference Changing elites in Europe LSE, London 26th-27th November 2015.

De Gruijter, D.N.M. 1967. *The Cognitive Structure of Dutch Political Parties in 1966.* Report E019-67. Psychological Institute, University of Leiden.

De Leeuw, J. 1974. *Canonical Analysis of Categorical Data.* Leiden, The Netherlands: Psychological Institute, Leiden University. http://www.stat.ucla.edu/~deleeuw/janspubs/1974/books/deleeuw_B_74.pdf.

———. 1982. "Nonlinear Principal Component Analysis." In *COMPSTAT 1982*, edited by H. Caussinus, P. Ettinger, and R. Tomassone, 77–86. Vienna, Austria: Physika Verlag. http://www.stat.ucla.edu/~deleeuw/janspubs/1982/chapters/deleeuw_C_82.pdf.

———. 1988. "Multivariate Analysis with Linearizable Regressions." *Psychometrika* 53: 437–54. http://www.stat.ucla.edu/~deleeuw/janspubs/1988/articles/deleeuw_A_88a.pdf.

———. 2007. *A Horseshoe for Multidimensional Scaling.* Preprint Series 530. Los Angeles, CA: UCLA Department of Statistics. http://www.stat.ucla.edu/~deleeuw/janspubs/2007/reports/deleeuw_R_07a.pdf.

———. 2008. *Using Jacobi Plane Rotations in R.* Preprint Series 556. Los Angeles, CA: UCLA Department of Statistics. http://www.stat.ucla.edu/~deleeuw/janspubs/2008/reports/deleeuw_R_08a.pdf.

———. 2015. "Multiset Canonical Correlation Analysis." http://rpubs.com/deleeuw/126462.

De Leeuw, J., and P. Mair. 2009. "Homogeneity Analysis in R: the Package `homals`." *Journal of Statistical Software* 31 (4): 1–21. http://www.stat.ucla.edu/~deleeuw/janspubs/2009/articles/deleeuw_mair_A_09a.pdf.

Diaconis, P., S. Goel, and S. Holmes. 2008. "Horseshoes in Multidimensional Scaling and Kernel Methods." *Annals of Applied Statistics* 2: 777–807.

Gifi, A. 1980. *Niet-Lineaire Multivariate Analyse [Nonlinear Multivariate Analysis].* Leiden, The Netherlands: Department of Data Theory FSW/RUL. http://www.stat.ucla.edu/~deleeuw/janspubs/1980/books/gifi_B_80.pdf.

———. 1990. *Nonlinear Multivariate Analysis.* New York, N.Y.: Wiley.

Guttman, L. 1941. "The Quantification of a Class of Attributes: A Theory and Method of Scale Construction." In *The Prediction of Personal Adjustment*, edited by P. Horst, 321–48. New York: Social Science Research Council.

———. 1950. "The Principal Components of Scale Analysis." In *Measurement and Prediction*, edited by S.A. Stouffer and Others. Princeton: Princeton University Press.

Hasselman, B., and Lapack authors. 2015. *geigen: Calculate Generalized Eigenvalues, the Generalized Schur Decomposition and the Generalized Singular Value Decomposition of a Matrix Pair with Lapack.* https://CRAN.R-project.org/package=geigen.

Hill, M.O. 1974. "Correspondence Analysis: a Neglected Multivariate Method." *Applied Statistics* 23: 340–54.

Hill, M.O., and H.G. Gauch. 1980. "Detrended Correspondence Analysis: An Improved Ordination Technique." *Vegetatio* 42: 47–58.

Ihm, P., and H. van Groenewoud. 1975. “A Multivariate Ordering of Vegetation Data Based on Gaussian Type Gradient Response Curves.” *The Journal of Ecology* 63: 767–77.

Lancaster, H.O. 1958. “The Structure of Bivariate Distributions.” *Annals of Mathematical Statistics* 29: 719–36.

Levelt, W.J.M., J.P. Van De Geer, and R. Plomp. 1966. “Triadic Comparions of Musical Intervals.” *British Journal of Mathematical and Statistical Psychology* 19: 163–79.

Mair, P., and J. De Leeuw. 2010. “A General Framework for Multivariate Analysis with Optimal Scaling: The R Package aspect.” *Journal of Statistical Software* 32 (9): 1–23. http://www.stat.ucla.edu/~deleeuw/janspubs/2010/articles/mair_deleeuw_A_10.pdf.

Meulman, J.J., and A. Gifi. 1981. *One-dimensional Homogeneity Analysis, Including Principal Components Analysis with Optimal Scaling.* Department of Data Theory, Leiden University.

Naouri, J.C. 1970. “Analyse Factorielle des Correspondences Continues.” *Publications de L’Institute de Statistique de L’Université de Paris* 19: 1–100.

Schriever, B.F. 1983. “Scaling of Order-dependent Categorical Variables with Correspondence Analysis.” *International Statistical Review* 51: 225–38.

———. 1985. “Order Dependence.” PhD thesis, University of Amsterdam, The Netherlands.

Van Rijckevorsel, J.L.A. 1987. “The Application of Fuzzy Coding and Horseshoes in Multiple Correspondence Analysis.” PhD thesis, University of Leiden, The Netherlands.